**Programming Research Group**

# A UNIVERSAL INNOCENT GAME MODEL FOR THE BÖHM TREE LAMBDA THEORY


Andrew D. Ker, Hanno Nickau, C.-H. Luke Ong

{adk,nickau,lo}@comlab.ox.ac.uk
Oxford University Computing Laboratory,
Parks Road, Oxford. OX1 3QD



PRG-TR-10-00



Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD

**Abstract**

We present a game model of the untyped $\lambda$-calculus, with equational theory equal to the Böhm tree $\lambda$-theory $\mathcal{B}$, which is universal (i.e. every element of the model is definable by some term). This answers a question of Di Gianantonio, Franco and Honsell. We build on our earlier work, which uses the methods of innocent game semantics to develop a universal model inducing the maximal consistent sensible theory $\mathcal{H}^*$. To our knowledge these are the first syntax-independent universal models of the untyped $\lambda$-calculus.

**Keywords:** Game semantics, effectively almost-everywhere copycat, innocent strategies, untyped $\lambda$-calculus.

## Contents

## 1 Introduction

We aim to construct a *universal* model (i.e. every element of the model is the denotation of some term) of the pure untyped $\lambda$-calculus which induces the Böhm tree $\lambda$-theory $\mathcal{B}$, by building on the game models presented in [KNO01]. Although the general approach is innocent in the sense of [HO00] and [Nic96], the two-player games we use are simpler and can be considered a special case where moves are neither questions nor answers but simply "declarations". A notable feature of game semantics is that the $\lambda$-definable strategies are effective methods for copying moves uniformly from one "component" of the game to another. For example, the identity strategy on an arena $A \Rightarrow A$ is *everywhere copycat* i.e. P always plays back every O-move (but in the opposite component of $A$). The key idea is that the innocent strategies definable by untyped $\lambda$-terms are, what we call, *effectively almost-everywhere copycat* (EAC). Informally this means that at every position, except in response to finitely many possible O-moves, the strategy is constrained to behave, from that point onwards, uniformly in a everywhere-copycat fashion, just like the identity strategy. Effectively here means that (not only is the strategy itself recursive but also) at every position, the boundary of that finite part of the game tree in which the strategy is *not* forced to play copycat must be computable.

We find it convenient to introduce innocent strategies in a concrete setting whereby (tree) arenas are defined as subsets of $\mathbb{N}^*$ of a certain kind, and this we do in Section 2.1. Section 2.2 introduces the EAC strategies which give rise to a universal $\lambda$-model $\mathcal{D}_{\mathrm{EAC}}$ whose theory is the maximal consistent sensible $\lambda$-theory $\mathcal{H}^*$. The definition of such strategies uses an efficient encoding of innocent strategies, as partial functions from $\mathbb{N}^*$ to tuples of numbers, which we call *economical form*. Section 2 should be regarded as a survey of [KNO01], and this paper is a sequel to that work. The notion of EAC strategies has a natural extension to *effectively and explicitly almost-everywhere copycat*. However finding an ambient cartesian closed category for these strategies to inhabit proved to be a painful process as we show in Section 3 – the natural analogues fail to work quite as intended. Once this has been overcome we use a reflexive object to describe a $\lambda$-algebra which we call $\mathcal{D}_{\mathrm{XA}}$. In Section 4 we formulate new versions of the powerful *Exact Correspondence Theorem* and *Separation Lemma* of the earlier work, to show that $\mathcal{D}_{\mathrm{XA}}$ is a universal $\lambda$-model which induces the intended equational theory. To our knowledge, $\mathcal{D}_{\mathrm{EAC}}$ and $\mathcal{D}_{\mathrm{XA}}$ are the first syntax-independent universal $\lambda$-models.

In [AM95] Abramsky and McCusker constructed the first game model for the untyped $\lambda$-calculus which was fully abstract for the Lazy Lambda Calculus. More recently in [GFH98], Di Gianantonio *et al* have obtained game models of the untyped $\lambda$-calculus using history-free strategies [AJM94]. They show that all their models induce the same $\lambda$-theory $\mathcal{H}^*$ and have asked for "new techniques for overcoming this apparent rigidity of game $\lambda$-models". This paper answers that question by constructing a universal game model for the Böhm tree $\lambda$-theory.

An extended abstract of this work [KNO99] (in which the model $\mathcal{D}_{\mathrm{XA}}$ was called $\mathcal{M}$) was presented at the European Association for Computer Science Logic conference in Madrid, in September 1999. The work presented in this paper was undertaken as part of the first-named author's EPSRC-funded doctoral research (the results appear in [Ker01]) and also under grant GR/L27787.

## 2 Review of Previous Work

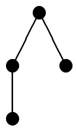
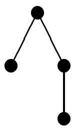
We begin with a brief summary of the results of [KNO01], on which we build in the body of this paper. The definitions in this section are only reviewed in a cursory way; the reader is referred to [KNO01] for a more detailed exposition and proofs of the later results, and to [HO00] and [McC98] for proofs of the initial results regarding categories of games.

### 2.1 Arenas and Innocent Strategies

We begin with a word on trees. Usually, when we talk about a tree we mean a countably-branching *labelled* tree, and presented in a concrete way. Often, the labels will be sequences in $\mathbb{N}^*$[1]; the root is labelled $\varepsilon$ and the descendants of the node labelled $\vec{s}$ are labelled $\vec{s}\cdot 1, \ldots, \vec{s}\cdot n$, if there are $n$ such descendants (we use the notation $\vec{s}$ for sequences and $\cdot$ for concatenation, prefix, or extension). Thus we can talk about the "$m^{th}$ descendant of the node $\vec{s}$" — it is the node labelled $\vec{s} \cdot m$. We can describe a tree by the set of labels of its nodes.

When we draw trees they are illustrated "upside-down" with the root at the top, rather more like family trees than the botanical kind (following this analogy, we can refer to a *child* of a node, and say that one node *inherits from* another, with the obvious meanings). Because of the labelling by sequences of natural numbers, the children come with an order, also like a family tree. When we draw a tree without labels, we intend that the numbering of children goes from left to right.

This means that we do not consider the trees drawn

 and 

to be the same. The first is described by the set $\{\varepsilon, \langle 1\rangle, \langle 1, 1\rangle, \langle 2\rangle\}$ and the second by $\{\varepsilon, \langle 1\rangle, \langle 2\rangle, \langle 2, 1\rangle\}$. In the graph-theoretic sense, we would expect these to be different pictures of the same tree, but for our purposes they are different trees.

Regardless of how a tree is labelled, we can encode each node by an element of $\mathbb{N}^*$ by enumerating the branches at each node (as long as the tree is only countably branching, and there is a well order on the labels of the descendants of each node).

**Definition** We define an *arena* to be a finite tuple of nonempty trees of *moves*.

[1] We *do not* include 0 in the set $\mathbb{N}$, and write $\mathbb{N}_0$ for $\mathbb{N} \cup \{0\}$. The set $\mathbb{N}^*$ is the set of finite sequences in $\mathbb{N}$.

The root of each tree is called an *initial move*. Using the above convention we identify each move with the unique sequence of natural numbers which identifies its position in the tree, and hence every tree of an arena by a subset of $\mathbb{N}^*$. Conversely, given any subset $A \subseteq \mathbb{N}^*$ which is prefix-closed and has the property that whenever $\vec{s} \cdot n \in A$ we have $\vec{s} \cdot m \in A$ for each $m \leq n$, we can form an arena of one tree where the moves are the elements of $A$. *Henceforth by arenas, we shall always mean arenas in sequence-subset (of $\mathbb{N}^*$) form.* We can then say that an arena is *recursively enumerable* when each subset is a recursively enumerable subset of $\mathbb{N}^*$.

For example, the empty sequence $\langle\rangle$ is an arena, which we call the *empty arena*; $\langle\{\,\varepsilon\,\}\rangle$ is the minimal one-tree arena consisting of a root node; the maximal one-tree arena, consisting of an infinitely deep, infinitely branching tree, is $\langle\mathbb{N}^*\rangle$. As the empty arena, the minimal and maximal one-tree arenas are important, we shall name them $E, M$ and $U$ respectively.

We refer to the *depth* of an element of a tree, which is the length of the sequence which encodes it. Thus the root of a tree is at depth zero. We say that moves at an even depth of the trees (including the roots at depth 0) are *O-moves*, and moves at an odd depth are *P-moves*. O-moves are often denoted by $\bullet$ and P-moves by $\circ$.

There are two major constructions for forming arenas. Suppose $A = \langle A_1, \ldots, A_m\rangle$ and $B = \langle B_1, \ldots, B_n\rangle$ are arenas.

- The *product arena* $A \times B$ is the "disjoint union" of the trees of $A$ and $B$, the concatenation of their tuples. Formally $A \times B = \langle A_1, \ldots, A_m, B_1, \ldots, B_n\rangle$.

- The *function space arena* $A \Rightarrow B$ is constructed as follows: the initial moves of $A \Rightarrow B$ are those of $B$; and to the tree below each such initial move, we graft onto it a copy of $A$. Precisely, $A \Rightarrow B = \langle C_1, \ldots, C_n\rangle$ where

$$C_i = \{\varepsilon\} \;\cup\; \{a \cdot \vec{s} \mid 1 \leq a \leq m \wedge \vec{s} \in A_a\} \;\cup\; \{(a+m) \cdot \vec{s} \mid a \cdot \vec{s} \in B_i\}.$$

The reader may wish to check that $M \Rightarrow M = \langle\{\,\varepsilon, 1\,\}\rangle$, and that $U \Rightarrow U$ and $U$ are equal as arenas.

A *justified sequence* of an arena $A$ is a sequence of moves of which each element except the first, which must be an initial move, is equipped with a pointer to some previous move. We call the pointer a *justification pointer* and if the move $m^-$ is pointed to by $m$ we say that $m^-$ *justifies* $m$. We say that a move $m^-$ in a justified sequence *hereditarily justifies* $m$ if one can reach $m^-$ from $m$ by repeatedly following justification pointers. A justified sequence $\vec{s}$ is said to be *well-formed* if it satisfies the following properties[2]:

- *Alternation*: Elements of $\vec{s}$ alternate between P-moves and O-moves.

- *Justification*: If $m$ is justified by $m^-$ then the move $m$ is directly beneath $m^-$ in the tree of the arena.

[2] The no-dangling-question-mark condition in [HO00, Nic96] (equivalently the well-bracketing condition) is redundant for our arenas.

Henceforth all justified sequences are assumed to be well-formed.

The *P-view* of a justified sequence $\vec{s}$, written $\ulcorner\vec{s}\urcorner$, is given recursively by:

$$\begin{array}{rcll} \ulcorner\varepsilon\urcorner & = & \varepsilon & \text{for initial moves } \varepsilon \\ \ulcorner\vec{s}\cdot m\urcorner & = & \ulcorner\vec{s}\urcorner\cdot m & \text{for } m \text{ a P-move} \\ \ulcorner\vec{s}\cdot m^{-}\cdot\vec{t}\cdot m\urcorner & = & \ulcorner\vec{s}\urcorner\cdot m^{-}\cdot m & \text{for } m \text{ an O-move justified by } m^{-} \end{array}$$

The definition of *O-view*, $\llcorner\vec{s}\lrcorner$, is given analagously.

It may be the case that for a P-move $m$ justified by an O-move $m^{-}$ in some justified sequence $\vec{s}$, the move $m^{-}$ is deleted in $\ulcorner\vec{s}\urcorner$. Thus the O- or P-view of a justified sequence may not itself be a justified sequence. This motivates the following:

A *legal position* of an arena $A$ is a well-formed justified sequence $\vec{s}$ satisfying the *visibility condition*: for each non-initial P-move $m$ justified by $m^{-}$, say $\vec{s} = \vec{t_1}\cdot m^{-}\cdot\vec{t_2}\cdot m\cdot\vec{t_3}$, we have that $m^{-} \in \ulcorner\vec{t_1}\cdot m^{-}\cdot\vec{t_2}\cdot m\urcorner$. Similarly all O-moves are justified by P-moves appearing in the O-view up to that point. Then if $\vec{s}$ is a legal position then so are $\ulcorner\vec{s}\urcorner$ and $\llcorner\vec{s}\lrcorner$. By a *P-view* of an arena $A$, we shall mean a justified sequence which is the P-view of some legal position of $A$.

**Lemma 2.1.1 (View Characterisation)** A justified sequence of an arena $A$ is a P-view if and only if it is well-formed and every non-initial O-move is justified by the immediately preceding P-move.

Within arenas there are games played out between two players, called Proponent and Opponent, or just P and O. A *P-strategy* $\sigma$ for a single-tree arena $A$ consists of a prefix-closed subset of legal positions of $A$ which is deterministic (if $\vec{s}\cdot m \in \sigma$ and $\vec{s}\cdot m' \in \sigma$ for P-moves $m$ and $m'$ then $m = m'$ and these moves are justified by the same move in $\vec{s}$) and such that if $\vec{s}\cdot m \in \sigma$ for a P-move $m$ and $\vec{s}\cdot m\cdot m'$ is a legal position of $A$ then $\vec{s}\cdot m\cdot m' \in \sigma$. An *O-strategy* is defined analogously. However we are more often interested in P-strategies which we will usually just refer to as strategies. For a general arena $A = \langle A_1, \ldots, A_n\rangle$ a P-strategy is an $n$-tuple of P-strategies, one for each tree. In contrast, an O-strategy is just a single O-strategy on one of the trees, together with information which selects that tree.

If we have strategies $\sigma$ and $\tau$ on arenas $A \Rightarrow B$ and $B \Rightarrow C$ respectively then we can form their composite strategy $\sigma;\tau$ on $A \Rightarrow C$. Informally we do this by identifying O/P-moves of the $B$ component of $A \Rightarrow B$ with P/O-moves of the $B$ component of $B \Rightarrow C$, and then hiding all the moves in $B$. This is reminiscent of CSP's "parallel composition" and "hiding" operators; see [KNO01] for a formal definition. Similar ideas extend to arenas of multiple trees. An essentially straightforward result, although tedious in proof, is that composition is well-defined and associative. We will only be interested in strategies with a property called *innocence*.

A P-strategy $\sigma$ is *innocent* if for odd-length legal positions $\vec{s}$ and $\vec{t}$ and P-moves $m$, $\vec{s}\cdot m \in \sigma \wedge \vec{t} \in \sigma \wedge \ulcorner\vec{s}\urcorner = \ulcorner\vec{t}\urcorner \implies \vec{t}\cdot m \in \sigma$, where $m$ is justified by the move in $\vec{t}$ which projects under P-view to the whichever move justified $m$ in $\ulcorner\vec{s}\urcorner$, i.e. P's next move, and its justification, at each stage depends only on the P-view up to that point. An important fact is that composition of innocent strategies is well-defined (for a proof see [HO00,

§5.3]): if $\sigma$ is an innocent strategy on $A \Rightarrow B$ and $\tau$ an innocent strategy on $B \Rightarrow C$ then $\sigma;\tau$ is an innocent strategy on $A \Rightarrow C$.

The property of innocence means that such a strategy is determined by a partial function from odd-length P-views to *justified P-moves* i.e. P-moves equipped with a justification pointer back into the P-view. In fact, given an innocent strategy $\sigma$, we can define a canonical such function, which we write $f_\sigma$, that defines it. A function constructed in this way is called *innocent* and we can formalise such functions. We say that $f$ is an *innocent function* if $f$ is a partial function from odd-length P-views in $A$ to justified P-moves of $A$ such that $\mathrm{dom}(f)$ is closed under odd-length prefix, and if $\vec{s} \cdot m \cdot m' \in \mathrm{dom}(f)$ then $f(\vec{s}) = m$. We note that such a function can only encode an innocent strategy. The conditions given are required to make the function "strategic", i.e. the set of legal positions it describes are prefix-closed, deterministic and made up of properly justified sequences of moves. These conditions are sufficient to allow us to define the reverse construction of a unique strategy $\sigma_f$ from an innocent function $f$ such that the construction is invertible (i.e. $f_{\sigma_f} = f$ and $\sigma_{f_\sigma} = \sigma$) and it preserves and reflects inclusion (i.e. $f \subseteq f' \iff \sigma_f \subseteq \sigma_{f'}$). Thus we can identify the representation by innocent function and that by subset of legal positions.

An innocent strategy is said to be *compact* if the graph of its innocent function is finite (i.e. is defined on finitely many P-views). It is said to be *recursive* if the innocent function representing it is recursive. It is easy to see that the composition of two recursive innocent strategies is itself recursive.

**Definition** Objects of the *Category of Arenas and Innocent Strategies*, $\mathbb{A}$, are arenas (in sequence-subset form); morphisms $f : A \to B$ are innocent strategies on the function space arena $A \Rightarrow B$. Composition of morphisms is composition as strategies. The *Category of Arenas and Recursive Innocent Strategies*, $\mathbb{A}_{\mathrm{REC}}$, has recursively enumerable arenas as objects and recursive innocent strategies as morphisms.

**Theorem 2.1.2** $\mathbb{A}$ and $\mathbb{A}_{\mathrm{REC}}$ are both cartesian closed.

The terminal object $\mathbf{1}$ of both $\mathbb{A}$ and $\mathbb{A}_{\mathrm{REC}}$ is the empty arena $E$, and the categorical constructions of product and function space are exactly the respective arena constructs. The category $\mathbb{A}$ is enriched over dI-domains. (One cannot say the same of $\mathbb{A}_{\mathrm{REC}}$, because the computable partial functions do not form a cpo. For example, one can "approximate" the Halting Problem by computable functions.)

Scott has observed that every $\lambda$-algebra arises from a reflexive object $R$ in some cartesian closed category $\mathbb{C}$. We give the construction explicitly as it is needed for analysis later (although the reader is referred to [Bar84] fora comprehensive treatment).

Given a cartesian closed $\mathbb{C}$ with reflexive object $R$, together with morphisms $\mathit{Fun} : R \to [R \Rightarrow R]$ and $\mathit{Gr} : [R \Rightarrow R] \to R$ such that $\mathit{Gr}\,;\mathit{Fun} = \mathrm{id}_{[R \Rightarrow R]}$, define a $\lambda$-algebra $\langle \mathcal{A}, \bullet, [\![-]\!]_{-} \rangle$ as follows:

(i) $\mathcal{A}$ is the homset $\mathrm{Hom}_{\mathbb{C}}(\mathbf{1}, R)$.

(ii) For any object $A$ with $f, g : A \to R$ define $f \bullet g = \langle f; \mathit{Fun}, g\rangle; \mathrm{eval}_{R,R}$. In particular this defines a binary operation on $\mathcal{A}$.

(iii) If $\{x_1, \ldots, x_n\} \supseteq \mathrm{FV}(s)$ define inductively the morphism $[\![s]\!]_\Delta : R^n \to R$, where $\Delta = \langle x_1, \ldots x_n\rangle$, as follows:

$$\begin{array}{rcl} [\![x]\!]_\Delta & = & \Pi^\Delta_x \text{ (the obvious projection morphism)}, \\ [\![st]\!]_\Delta & = & [\![s]\!]_\Delta \bullet [\![t]\!]_\Delta, \\ [\![\lambda x.s]\!]_\Delta & = & \Lambda([\![s]\!]_{\Delta \cdot x}); \mathit{Gr}. \end{array}$$

In the last clause we may assume that $x$ does not appear in $\Delta$ (by renaming if necessary).

(iv) If $\rho$ is a valuation mapping variables to elements of $\mathcal{A}$, and $\Delta$ is as above, define the morphism $\rho^\Delta : \mathbf{1} \to R^n$ by $\rho^\Delta = \langle \rho(x_1), \ldots, \rho(x_n)\rangle$. Then set

$$[\![s]\!]_\rho = \rho^\Delta; [\![s]\!]_\Delta.$$

Thus we may specify a $\lambda$-algebra by a 4-tuple which we shall write as $\mathcal{M}(\mathbb{C}, R, \mathit{Fun}, \mathit{Gr})$ (it is in fact a $\lambda\eta$-algebra if $\mathit{Fun}\,; \mathit{Gr} = \mathrm{id}_R$). If the reflexive object $R$ *has enough points* (i.e. $\forall\, f, g : R \to R.[\forall\, r : \mathbf{1} \to R.r\,;\, f = r\,;\, g] \implies f = g$) then $\mathcal{M}(\mathbb{C}, R, \mathit{Fun}, \mathit{Gr})$ is a $\lambda$-*model* (i.e. a weakly extensional $\lambda$-algebra).

Recall that the arena $U$ has the key property that $U = U \Rightarrow U$ so that in this case the morphisms *Fun* and *Gr* are both the identity on $U$. We can now define the first two of our game $\lambda$-algebras (which are both $\lambda\eta$-algebras): $\mathcal{M}(\mathbb{A}, U, \mathrm{id}_U, \mathrm{id}_U)$ which we shall write simply as $\mathcal{D}$, and $\mathcal{M}(\mathbb{A}_{\mathrm{REC}}, U, \mathrm{id}_U, \mathrm{id}_U)$ which we shall write as $\mathcal{D}_{\mathrm{REC}}$. By abuse of notation, we shall use $\mathcal{D}$ and $\mathcal{D}_{\mathrm{REC}}$ to denote the respective underlying sets. Clearly $\mathcal{D}_{\mathrm{REC}} \subset \mathcal{D}$. By a method of approximation introduced in [KNO01] we can show that both the $\lambda\eta$-algebras are *sensible* i.e. all unsolvable $\lambda$-terms have the same denotation which in this case is given by the everywhere undefined innocent function.

### 2.2 Effectively Almost-Everywhere Copycat Strategies

There are three properties that allow for a more compact representation of an innocent strategy:

(i) Each non-initial O-move in any P-view must be a child of the previous move, and the initial move must be $\varepsilon$.

(ii) Given only the O-moves of a P-view and the value of the innocent function on strictly shorter P-views we can reconstruct the original P-view entirely.

(iii) The P-move to which this P-view is mapped must be a child of the move justifying it.

In view of these redundancies, we encode innocent strategies $\sigma$, over any single-tree arena, as (partial) maps from $\mathbb{N}^*$ to $\mathbb{N} \times \mathbb{N}_0$. We call this encoding the *economical form* of $\sigma$ and sometimes write it $e_\sigma$ (quite often we abuse notation and write it $f_\sigma$ too). It is defined as follows:

$e_\sigma : \langle v_1, \ldots, v_n\rangle \mapsto (i, p)$ if and only if

$$f_\sigma : \begin{array}{cccccccccccccc} \bullet & \circ & \bullet & \circ & \bullet & \cdots & \circ & \bullet & \cdots & \circ & \bullet & \mapsto & \circ \\ \varepsilon & \vec{s}_1 & \vec{s}_1 v_1 & \vec{s}_2 & \vec{s}_2 v_2 & & \vec{s}_{n-p} & \vec{s}_{n-p} v_{n-p} & & \vec{s}_n & \vec{s}_n v_n & & \vec{s}_{n-p}(v_{n-p} i) \end{array}$$

(When the resulting move from a clause of the innocent function is a child of the root node we have $n = p$, and the intention is that the (non-existent) move $\vec{s_0}v_0$ should mean the initial move $\varepsilon$.)

Justification pointers in the P-view can be deduced from the behaviour of $f_\sigma$ on shorter P-views, and they do not affect the value of $e_\sigma\langle v_1, \ldots, v_n\rangle$, and so have been omitted. Note that each $\vec{s_i}$ is a sequence of natural numbers coding a move of the arena.

Furthermore, we can expand any partial function $f : \mathbb{N}^* \rightharpoonup \mathbb{N} \times \mathbb{N}_0$ which has prefix-closed domain and satisfies $f(\vec{v}) = (i, p) \implies 0 \leq p \leq |\vec{v}|$ into an innocent strategy on $U$. Depending on the function, we might not need the whole of $U$ to contain the strategy. We could extend this idea for multiple-tree arenas, but since we will not use it except on the arena $U$ there is no need to do so.

**Example** The following is the innocent function of the "copycat" strategy $\mathrm{id}_U$:

$$\underset{\varepsilon}{\bullet} \;\mapsto\; \underset{1}{\circ} \qquad\qquad \underset{\varepsilon}{\bullet}\ \cdots\ \underset{(a{+}1)\vec{s}}{\bullet}\ \underset{1a\vec{s}}{\circ}\ \underset{1a\vec{s}b}{\bullet} \;\mapsto\; \underset{(a{+}1)\vec{s}b}{\circ}$$

$$\underset{\varepsilon}{\bullet}\ \underset{1}{\circ}\ \underset{1a}{\bullet} \;\mapsto\; \underset{(a{+}1)}{\circ} \qquad\qquad \underset{\varepsilon}{\bullet}\ \cdots\ \underset{1a\vec{s}}{\bullet}\ \underset{(a{+}1)\vec{s}}{\circ}\ \underset{(a{+}1)\vec{s}b}{\bullet} \;\mapsto\; \underset{1a\vec{s}b}{\circ}$$

Here $\vec{s}$ range over sequences of appropriate parity, $a$ and $b$ over positive natural numbers. The reader is invited to check that the economical form of this strategy is given by: $\varepsilon \mapsto (1, 0), i \mapsto (i + 1, 1)$ and for nonempty sequences $\vec{v}$, $\vec{v}i \mapsto (i, 1)$.

A principle of the $\lambda$-calculus is that a term can be applied successively to any other term. So the term $\lambda x.x$ (say) is really more like "$\lambda x z_0 z_1 z_2 \ldots \bullet\, x z_0 z_1 z_2 \ldots$" (we use a large dot $\bullet$ to make the "end" of the infinite chain of abstractions really clear). Thus there is some notion of infinite $\eta$-expansion. If we think about the denotation of $\lambda x.x$ in the game models, it is similarly expanded — it copies the whole of the first subtree to the rest of the arena, as if copying not only the $x$ variable but also all of its arguments. This correspondence turns out to be general, and can be made precise by relating innocent strategies in economical form to a kind of (infinitely) $\eta$-expanded Böhm trees first studied by Nakajima in [Nak75]. We call a formal connexion of this form an *Exact Correspondence Theorem*.

For a $\lambda$-term $s$ the *Nakajima tree* of $s$, written $\mathrm{NT}(s)$, is (informally) the countably branching, countably deep tree labelled as follows. If $s$ is unsolvable then $\mathrm{NT}(s) = \perp$, the empty tree. If $s$ has HNF $\lambda x_1 \ldots x_n \bullet y s_1 \ldots s_m$ then

$$\mathrm{NT}(s) \;=\; \begin{array}{c} \lambda x_1 \ldots x_n z_0 z_1 \ldots \bullet\, y \\ \mathrm{NT}(s_1) \quad \cdots \quad \mathrm{NT}(s_m) \quad \mathrm{NT}(z_0) \quad \mathrm{NT}(z_1) \quad \cdots \end{array}$$

where $z_0, z_1, \ldots$ are countably many fresh variables. The process of finding such fresh variables given in [Nak75] is quite complicated.

In [KNO01] we propose a *variable-free* representation of Nakajima trees so that for a closed term $s$, $\mathrm{NT}(s)$ is represented as $\mathrm{VFF}(s)$, a partial function from $\mathbb{N}^*$ to $\mathbb{N} \times \mathbb{N}_0$.

Note that the "infinitely nested" $\lambda$-abstractions of the form $\lambda z_1 z_2 \ldots \bullet y$, which label the nodes of a Nakajima tree (of a closed term), can be coded as a pair $(i, r)$ whereby the head variable $y$ is the $i^{th}$ in the infinite list of variables bound by the $\lambda$-abstraction situated $r$ levels up in the tree. The map $\mathrm{VFF}(s)$ is just a function that maps occurrences (of nodes) to such labels encoded as pairs of numbers, and a translation from terms to such

**Definition** For a partially $(\mathbb{N} \times \mathbb{N}_0)$-labelled tree $p$ the tree $\{p\}^*$ is the same tree labelled identically, except that nodes at depth $d$ labelled $(i, d+1)$ are relabelled $(i, d+2)$.

Similarly the tree $\{p\}^n$, for $n \in \mathbb{N}_0$, is labelled identically except for nodes of depth $d$ as follows:

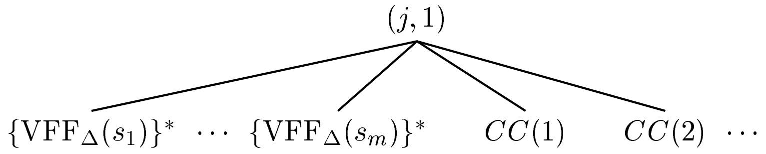
(i) those labelled $(i, d)$ are relabelled $(i+n, d)$;

(ii) those labelled $(i, d+1)$ for $i \leq n$ are relabelled $(n-i+1, d)$;

(iii) those labelled $(i, d+1)$ for $i > n$ are relabelled $(i-n, d+1)$.

For a term $s$ with free variables within $\Delta$ the *variable-free form* of the Nakajima tree of $s$, $\mathrm{VFF}_\Delta(s)$, is the following partially $(\mathbb{N} \times \mathbb{N}_0)$-labelled tree:

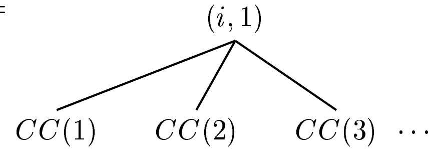
$$\begin{array}{rcl}
\mathrm{VFF}_\Delta(s) & = & \perp \text{, for unsolvable } s. \\
\mathrm{VFF}_\Delta(\lambda x_1 \ldots x_n \bullet s) & = & \{\mathrm{VFF}_{\Delta \cdot \langle x_1, \ldots, x_n \rangle}(s)\}^n \text{, if } s \text{ is of the form } v_j s_1 \ldots s_m. \\
\mathrm{VFF}_\Delta(v_j s_1 \ldots s_m) & = &
\end{array}$$




where $\Delta = \langle v_k, \ldots, v_1 \rangle$ (note the reverse order).

Here $CC(i)$ is the infinite tree defined by

$CC(i) =$




To see $\mathrm{VFF}_\Delta(s)$ as a partial function from $\mathbb{N}^*$ to $\mathbb{N} \times \mathbb{N}_0$, note that we can identify the labelled tree with the labelling function, which maps the encoding of each node in $\mathbb{N}^*$ to the pair labelling that node.

**Lemma 2.2.1** This definition coincides with the informal notion of variable-free form described earlier. Formally, suppose that $s$ is a term with free variables all occurring in the sequence $\Delta = \langle v_k, \ldots, v_1 \rangle$. Construct the Nakajima tree of $s$, and rename all the bound variables so that if $\vec{a} \in \mathbb{N}^*$ is the sequence identifying a node of the Nakajima tree then the abstracted variables at that node are renamed to $x_1^{\vec{a}}, x_2^{\vec{a}}, \ldots$ (in that order). Let this renamed Nakajima tree have labelling function $A$, and consider $\mathrm{VFF}_{\Delta}(s)$ also as a labelling function.

Then for any sequence $\vec{a} = \langle a_1, \ldots, a_p \rangle$ there are three possibilities for $A(\vec{a})$:

(i) If $\vec{a} \notin \mathrm{dom}(A)$ then $\mathrm{VFF}_{\Delta}(s)$ is unlabelled or undefined at $\vec{a}$,

(ii) If $A(\vec{a}) = \lambda x_1^{\vec{a}} x_2^{\vec{a}} \ldots \bullet v_j$ then $\mathrm{VFF}_{\Delta}(s)(\vec{a}) = (j, p+1)$,

(iii) If $A(\vec{a}) = \lambda x_1^{\vec{a}} x_2^{\vec{a}} \ldots \bullet x_j^{\langle a_1, \ldots, a_{p-r} \rangle}$ then $\mathrm{VFF}_{\Delta}(s)(\vec{a}) = (j, r)$.

The following theorem, a key result in [KNO01], connects the economical form of an innocent strategy denoting a term, with the variable free form of the Nakajima tree of that term.

**Theorem 2.2.2 (Exact Correspondence)** For every closed $\lambda$-term $s$, the innocent strategy denotation $[\![s]\!]$ in both $\mathcal{D}$ and $\mathcal{D}_{\mathrm{REC}}$, given in economical form, is exactly $\mathrm{VFF}(s)$, the Nakajima tree of $s$ in variable free form.

**Example** We now introduce example terms and strategies which we will use repeatedly to illustrate many of the concepts in the rest of the paper. Consider the terms $I = \lambda x.x$ and $1 = \lambda xy.xy$. The reader may wish to verify that the following represents the first two levels of the Nakajima trees of those terms:

$$\begin{array}{ccc} \mathrm{NT}(I) & = & \lambda x z_0 z_1 \ldots \bullet x \\ & & \lambda \vec{u} \bullet z_0 \quad \lambda \vec{v} \bullet z_1 \quad \lambda \vec{w} \bullet z_2 \cdots \end{array} \qquad \begin{array}{ccc} \mathrm{NT}(1) & = & \lambda x y z_0 z_1 \ldots \bullet x \\ & & \lambda \vec{u} \bullet y \quad \lambda \vec{v} \bullet z_0 \quad \lambda \vec{w} \bullet z_1 \cdots \end{array}$$

After renaming of bound variables, these are the same. Since $I$ and $1$ differ only by $\eta$-conversion, this should be no surprise. Thus we can calculate their common variable-free form, the first two levels of which is:

$$\begin{array}{c} (1,0) \\ (2,1) \quad (3,1) \quad (4,1) \quad \cdots \end{array}$$

For example, the node labelled $(2,1)$ means that the head variable of the corresponding node in the Nakajima tree is found as the second in the list of variables abstracted at the node one level above. The Exact Correspondence Theorem tells us that $[\![I]\!] = [\![1]\!]$ has the economical form which is given (in part) by $\varepsilon \mapsto (1,0), \langle 1 \rangle \mapsto (2,1), \langle 2 \rangle \mapsto (3,1)$ and so on.

We say that a $\lambda$-algebra is *universal* if every element is the denotation of some $\lambda$-term. By the Exact Correspondence Theorem, it is easy to see that neither $\mathcal{D}$ nor $\mathcal{D}_{\mathrm{REC}}$ is universal, since no *non-trivial* compact innocent strategy can be the denotation of any $\lambda$-term (note that the only finite Nakajima tree is the trivial tree $\perp$).

Our aim in the rest of this section is to characterise the definable parts of $\mathcal{D}_{\mathrm{REC}}$, and we shall do so by capturing the right ambient CCC.

**Notation** For tree-like $A \subseteq \mathbb{N}^*$ (i.e. those subsets which are prefix-closed and satisfy $\vec{s} \cdot n \in A \implies \vec{s} \cdot m \in A$ for all $m < n$) and for any $\vec{s} \in A$ we define

$$\begin{array}{rcl} A\,@\,\vec{s} & = & \text{the subtree of } A \text{ rooted at } \vec{s} \\ A^{>m} & = & \text{the tree obtained from } A \text{ by deleting the first } m \text{ branches.} \end{array}$$

For example, for the maximal single-tree arena $U$, we have $U\,@\,\vec{s} = U = U^{>n}$ for all sequences $\vec{s}$ and numbers $n$. Next fix an innocent strategy in economical form $f$ and let $\vec{v} \in \mathrm{dom}(f)$. We shall use the following shorthand:

$$\begin{array}{rcl} \mathbf{m_o}^f(\vec{v}) & = & \text{the last move of the P-view encoded by } \vec{v} \\ \mathbf{m_p}^f(\vec{v}) & = & \text{the response of } \sigma_f \text{ at the P-view.} \end{array}$$

Note that the former is by definition an O-move and the latter a P-move. We omit the superscript $f$ wherever it is clear which strategy is intended. For example, for any innocent strategy $f$ the O-move $\mathbf{m_o}(\varepsilon)$ is the initial move $\varepsilon$ and $\mathbf{m_p}(\varepsilon)$ is the first P-move made by $\sigma_f$ in response. Now we can define a new property of strategies:

**Definition** Consider an innocent strategy in economical form $f : \mathbb{N}^* \to \mathbb{N} \times \mathbb{N}_0$, over some single-tree arena $A$. We say that $f$ is *everywhere copycat* (EC) at $\vec{v} \in \mathbb{N}^*$ if $f$ is undefined at $\vec{v}$ or the following hold:

(i) The arenas $A\,@\,\mathbf{m_o}(\vec{v})$ and $A\,@\,\mathbf{m_p}(\vec{v})$ are order-isomorphic (with respect to the prefix ordering).

(ii) Whenever $\vec{w} \geq \vec{v}$ we have that for all $i \in \mathbb{N}$ such that the move coded by $\vec{w} \cdot i$ exists, $f(\vec{w} \cdot i) = (i, 1)$.

(iii) If $f(\vec{v}) = (i, p)$ then $p > 0$.

We say that $f$ is *almost-everywhere copycat* (AC) at $\vec{v}$ if $f$ is undefined at $\vec{v}$ or there exist numbers $t_v \in \mathbb{N}_0$ and $o_v \in \mathbb{Z}$ with $o_v \leq t_v$ called the *copycat threshold* and *offset* respectively, such that

(i) The arenas $(A\,@\,\mathbf{m_o}(\vec{v}))^{>(t_v - o_v)}$ and $(A\,@\,\mathbf{m_p}(\vec{v}))^{>t_v}$ are isomorphic.

(ii) For all $i > t_v$ such that the move coded by $\vec{v} \cdot i$ exists, $f(\vec{v} \cdot i) = (i - o_v, 1)$ and $f$ is everywhere copycat at $\vec{v} \cdot i$.

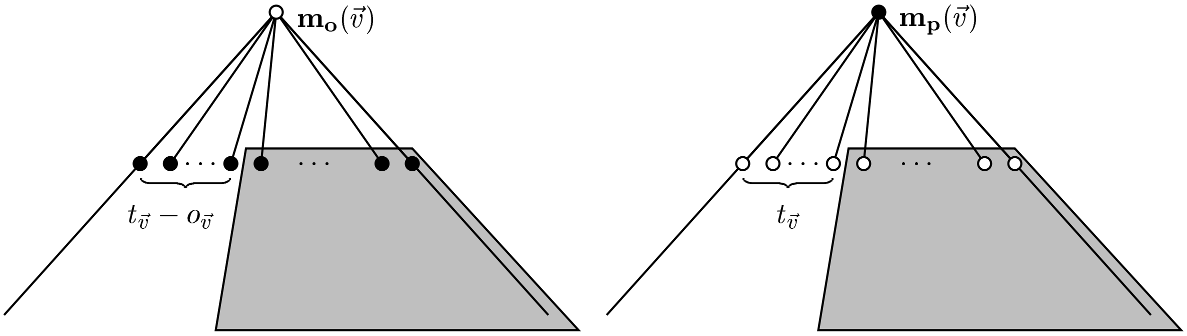
(iii) For all $\vec{w} \geq (\vec{v} \cdot k)$ with $k \leq t_v$, if $f(\vec{w}) = (i, |\vec{w}| - |\vec{v}|)$ then $i \leq t_v - o_v$.

(iv) If $f(\vec{v}) = (i, 0)$ then $i \leq t_v - o_v$.

(Note that $f$ is EC at $\vec{v}$ if and only if $f$ is AC at $\vec{v}$ with $t_v = o_v = 0$.)

Finally, we say that $f$ is *effectively almost-everywhere copycat* (EAC) if $f$ is computable, almost-everywhere copycat at every sequence on which it is defined and the functions $\vec{v} \mapsto t_v$ and $\vec{v} \mapsto o_v$ are computable. A strategy $\sigma$ over an arena $A$ is EAC if its innocent function is EAC, and we can generalise to multiple-tree arenas in the usual way.

We illustrate the definition of EAC. Suppose P plays a strategy which is almost-everywhere copycat at $\vec{v}$. The two arenas $A\,@\,\mathbf{m_o}(\vec{v})$ and $A\,@\,\mathbf{m_p}(\vec{v})$ are shown below.




The idea is that except for finitely many subtrees of the moves in question, P's behaviour is "everywhere copycat" at $\mathbf{m_o}(\vec{v})$ i.e. P simply copies O's move between two isomorphic subarenas (which are shaded in the figure).

Since the notion of EAC is only defined for innocent strategies, we will just say "EAC strategy" instead of "EAC innocent strategy".

For a specific P-view $\vec{v}$ of such a function $f$, we will say that $t_v$ and $o_v$ are *valid* copycat threshold and offset if $f$ satisfies the conditions (i)-(iv) of AC at that P-view with those particular values. Valid copycat thresholds are not unique, as the following lemma shows.

**Lemma 2.2.3** If $f : \mathbb{N}^* \to \mathbb{N} \times \mathbb{N}_0$ is an innocent strategy in economical form, and $f$ is defined and AC at some P-view $\vec{v}$ with copycat threshold and offset $t_v$ and $o_v$ respectively, then for any $t' \geq t_v$, $f$ is also AC at the P-view $\vec{v}$ with threshold and offset $t'$ and $o_v$ respectively. That is, any value larger than a valid copycat threshold is still a valid threshold for a specific P-view (with the same offset).

Thus at each P-view of an EAC strategy there will be a *least copycat threshold*, the least value for $t_v$ which is still a valid threshold. However, the existence of a computable function giving valid thresholds does not imply the computability of the function giving least thresholds.

**Definition** The *category of arenas and EAC strategies*, $\mathbb{A}_{\text{EAC}}$, has recursively enumerable arenas as objects and EAC strategies on $A \Rightarrow B$ as morphisms from $A$ to $B$.

A main result in [KNO01] is that the category $\mathbb{A}_{\text{EAC}}$ is well-defined; the proof that EAC strategies compose is omitted from this reference, but is in fact a simple consequence of Theorem 3.2.2 which appears later in this paper. Additionally,

**Theorem 2.2.4** $\mathbb{A}_{\text{EAC}}$ is cartesian closed.

The arena $U$ is still an object of $\mathbb{A}_{\text{EAC}}$ and still equal to its function space. Thus, in the same way as before, we can define a $\lambda\eta$-algebra $\mathcal{M}\langle\mathbb{A}_{\text{EAC}}, U, \text{id}_U, \text{id}_U\rangle$ which we shall denote by $\mathcal{D}_{\text{EAC}}$. Since $\mathbb{A}_{\text{EAC}}$ is a subcategory of $\mathbb{A}_{\text{REC}}$, with the same class of objects and the same cartesian closed structure, $\mathcal{D}_{\text{EAC}} \subset \mathcal{D}_{\text{REC}}$.

Other properties of $\mathcal{D}$, $\mathcal{D}_{\mathrm{REC}}$ and $\mathcal{D}_{\mathrm{EAC}}$ are explored in [KNO01]; one which is immediate from the Exact Correspondence Theorem is that two terms of the $\lambda$-calculus have the same denotation precisely when they have the same Nakajima tree. This equality is captured by the maximal consistent sensible theory $\mathcal{H}^*$ (see [Bar84]).Also, almost by construction, every EAC strategy on $U$ is the denotation of some closed term of the $\lambda$-calculus. Furthermore, $\mathcal{D}_{\mathrm{EAC}}$ inherits the order-extensionality property which the syntax of the $\lambda$-calculus (modulo the theory $\mathcal{H}^*$) enjoys. Thus:

**Theorem 2.2.5** The equational theory of the models $\mathcal{D}$, $\mathcal{D}_{\mathrm{REC}}$ and $\mathcal{D}_{\mathrm{EAC}}$ is $\mathcal{H}^*$, the maximal consistent sensible theory. Additionally, $\mathcal{D}_{\mathrm{EAC}}$ is a universal and order-extensional $\lambda$-model.

# 3 Effectively and Explicitly Almost-Everywhere Copycat Strategies

Our aim in this sequel paper is to find a game model which does not validate $\eta$-conversion. To do so, we will need to augment EAC strategies with some extra information. By examining the parts of Nakajima trees which use fresh variables, in the light of the Exact Correspondence Result, it becomes clear that the additional information we require is that which specifies copycat thresholds at each P-view. We are thus lead to the definition of an *EXAC strategy*, and we describe the problems involved in the search for a CCC of such strategies, and the solution $\mathbb{XA}$.

## 3.1 Specifying Copycat Thresholds

To find a model in which $\eta$-conversion is not validated, we require the terms $I$ and 1 to be denoted differently. They have the same variable-free form of Nakajima tree, so it is not apparent how this might be achieved. The key is to make use of the fact that the copycat thresholds are not unique — any number greater than a given valid copycat threshold is also a valid copycat threshold (Lemma 2.2.3). Different thresholds (at some P-view) may be used to distinguish $I$ and 1.

This idea is prompted by the observation that when one compares a term with its denotation, the part of the EAC strategy which is specified by the rules of copycat, the part which is not specified *explicitly*, corresponds precisely to the part of the Nakajima tree which has been generated by $\eta$-expansion (i.e. the part of the tree with the fresh variables as the head variables). Recall the Nakajima trees of $I$ and 1 — the former has fresh variables appearing at every node except the root, whereas the latter is similar except that there is not a fresh variable at the first child of the root. Therefore we aim to find a model where $I$ and 1 are represented by the strategy with the same moves, but the copycat threshold of $[\![I]\!]$ at the first P-view is 0, whereas that of $[\![1]\!]$ is 1.

However, the definition of an EAC strategy is stated in terms of the existence of some computable function which associates a pair of numbers to each P-view of the strategy and this function is *not* specified along with the strategy.(A consequence of this is that there is no computable procedure for finding valid thresholds for an EAC strategy, nor for deciding whether a given innocent strategy is EAC.)

**Remark 3.1.1** It is really the thresholds (rather than the offsets) which are important because, for a certain P-view $\vec{v}$ of an EAC strategy $\sigma$, the copycat threshold $t$ usually gives enough information to compute the offset $o$ directly. This is clear since $f_\sigma(\vec{v} \cdot (t+1)) = (t+1-o, 1)$, as long as the move coded by $\vec{v} \cdot (t+1)$ actually exists in the arena in question. When this move does not exist, any value would do for the offset. However, in this case the value of the offset is irrelevant and we will not take this technicality into account. Similarly, when none of the moves coded by $\vec{v} \cdot i$ exist for any $i$, the threshold is irrelevant. We will not distinguish between strategies which only differ in such circumstances. In practice, we only consider strategies over the arena $U$, in which all such moves always exist, so this technicality can be ignored.

This motivates the following definition:

**Definition** An *effectively and explicitly almost-everywhere copycat strategy* (EXAC strategy) is given by a pair $\langle \sigma, t_\sigma \rangle$, where $\sigma$ is an EAC strategy and $t_\sigma$ is an effective function mapping the P-views where $\sigma$ is defined to valid copycat thresholds. We sometimes write the EXAC strategy $\langle \sigma, t_\sigma \rangle$ just as $\sigma$.

We will usually refer to the first and second part of an EXAC strategy as the "(underlying) EAC strategy (part)" and the "threshold function (part)", respectively. In view of our remark above, however, we will sometimes speak of the offsets as if they too are specified by the threshold function.

This definition allows us to make the intended finer distinction between strategies: two strategies with the same moves must be equal as EAC strategies, but may have different copycat thresholds and so can be distinguished as EXAC strategies. There is an obvious forgetful map from EXAC strategies to EAC strategies, which takes only the strategy part (i.e. erasing the threshold information).

In a similar vein to the economical form of innocent strategies, using the same encoding of a P-view as a sequence of natural numbers, we can give an economical form of EXAC strategies over single-tree arenas. We can also take advantage of the fact that parts of the strategy are completely dictated by its copycat nature. Let us say that a P-view is *entirely explicit* if none of the O-moves in it exceed the copycat threshold of the P-view at which they are made. Thus if a P-view is not entirely explicit the ensuing move can be deduced from the threshold and offset of the P-view preceding the first O-move in it which did exceed the copycat threshold.

**Definition** The *economical form* of an EXAC strategy is a map from $\mathbb{N}^*$ to $\mathbb{N} \times \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{Z}$. The domain is the encoding of P-views in the same way as economical form of an EAC strategy. The map is defined at a sequence $\vec{v}$ only if the P-view encoded by $\vec{v}$ is entirely explicit, in which case

$$\vec{v} \mapsto (i, r, t, o)$$

where the resulting P-move is encoded as before — it is the $i^{th}$ child of the move $2r$ from last of the P-view — and the copycat threshold and offset at this P-view are $t$ and $o$ respectively.

**Example** We take the EXAC strategies $\eta_0$ and $\eta_1$ to be $\langle [\![I]\!], t_0\rangle$ and $\langle [\![1]\!], t_1\rangle$, where $t_0$ maps every P-view to the threshold 0 and $t_1$ does likewise except that the minimal P-view is mapped to the threshold 1. Since $[\![I]\!] = [\![1]\!]$, they have the same EAC strategy part, but different threshold functions. These are the suggestions we made for the denotations of $I$ and 1 in a model not supporting $\eta$-conversion. Nearly every P-view of either is not entirely explicit, and the respective economical forms are given by:

$$\begin{array}{rcl} \varepsilon & \mapsto & (1,0,0,-1) \end{array} \quad \text{and} \quad \begin{array}{rcl} \varepsilon & \mapsto & (1,0,1,-1) \\ \langle 1\rangle & \mapsto & (2,1,0,0) \end{array}$$

## 3.2 Composition of EXAC Strategies

We now need a method to compose EXAC strategies. Of course the EAC strategy part will just be the standard composition of innocent strategies, and we give below an algorithm for computing the composition of the threshold functions.

**Algorithm 3.2.1 (The Composition Algorithm)** Let $\langle \sigma, t_\sigma\rangle$ be an EXAC strategy over $A \Rightarrow B$, and $\langle \tau, t_\tau\rangle$ be an EXAC strategy over $B \Rightarrow C$. Take a P-view $\vec{v}$ on which the strategy $\sigma;\tau$ (which is given by the usual composition of innocent strategies) is defined and suppose that the last move of the P-view is $\underline{m}$ and the resulting move is $m$:

$$\sigma;\tau : \underbrace{\overset{\bullet}{\varepsilon} \cdots \overset{\bullet}{\underline{m}}}_{\vec{v}} \mapsto \overset{\circ}{m}$$

We write $\vec{u} = \mathbf{u}(\vec{v}, \sigma, \tau)$ for the *uncovering* of the composition up to the move $m$. A formal definition can be found in [HO00] or [KNO01], but we may describe it as the sequence of moves of the composition which result after the P-view $\vec{v}$, including any relevant $B$-moves which would be hidden by the composition. It will be of the form:

$$\underbrace{\overset{\bullet}{\varepsilon} \cdots \overset{\bullet}{\underline{m}}\ \overset{\star}{m_1}\ \overset{\star}{m_2} \cdots \overset{\star}{m_{p-1}}\ \overset{\star}{m_p}}_{\vec{u}}\ \overset{\circ}{m}$$

The moves $m_i$ are the intermediate interactions which might have taken place between $\sigma$ and $\tau$ before the move $m$ became the visible outcome. They are all in the arena $B$ and they are either O- or P-moves, depending on which strategy is looking at them which is why they are written $\star$. Possibly there are no such intermediate moves, in which case $p = 0$. We do not care about justification pointers, and for tidiness set $m_0 = \underline{m}$ and $m_{p+1} = m$.

For $1 \leq i \leq p+1$ we consider the P-view that the strategies $\sigma$ and $\tau$ are faced with when making the move $m_i$, and set

$$\vec{u_i} = \ulcorner \vec{u}_{\leq m_{i-1}} \upharpoonright X \urcorner^X$$

where $X$ depends on which strategy is to make the move $m_i$. It will be $(A, B, b)$ or $(B, C, c)$, where $b$ or $c$ is the initial $B$- or $C$-move hereditarily justifying $m_i$.

Define $t_i$ and $o_i$ to be the copycat threshold and offset of $\sigma$, or $\tau$ as appropriate, at the P-view $\vec{u_i}$. These are specified by $t_\sigma$ or $t_\tau$. Then set:

$$\begin{array}{rcll}
t'_i & = & t_i + |A|, & \text{if } m_i \text{ is a root of the arena } B \\
 & & t_i, & \text{otherwise} \\
o'_i & = & o_i + |A|, & \text{if } m_i \text{ is a root of the arena } B \\
 & & o_i, & \text{otherwise}
\end{array}$$

$$\begin{array}{rcl}
T_1 & = & t'_1 \\
O_1 & = & o'_1 \\
T_{i+1} & = & \max(T_i + o'_{i+1}, t'_{i+1}) \\
O_{i+1} & = & O_i + o'_{i+1}
\end{array}$$

$$\begin{array}{rcll}
\mathbf{t} & = & T_{p+1} & \\
\mathbf{o} & = & O_{p+1} - |A| + |B|, & \text{if } \underline{\mathrm{m}} \text{ is a root of the arena } C \\
 & & O_{p+1}, & \text{otherwise}
\end{array}$$

Then $\mathbf{t}$ and $\mathbf{o}$ are the copycat threshold and offset of the composition $\langle \sigma, t_\sigma \rangle ; \langle \tau, t_\tau \rangle$ at the P-view $\vec{v}$.

Now we must show that this method does indeed produce an EXAC strategy, i.e. that the composite threshold function specifies valid thresholds and offsets for the composite strategy. In fact it does so only under some restrictions, for which we need an additional definition.

**Definition** Let $\sigma$ be an EAC strategy over a single-tree arena. If $\sigma$ has a first move, then it has a copycat threshold and offset, say $t$ and $o$, at the P-view consisting only of the root O-move. The *l-number* of $\sigma$ is the value $t - o$, and we write it $\mathbf{l}(\sigma)$.

If $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle$ is an EAC strategy over an arena with $n$ trees, which is defined on at least one of the minimal P-views, then we define $\mathbf{l}(\sigma) = \min_{i=1, \sigma_i \neq \perp}^{n} \{\mathbf{l}(\sigma_i)\}$.

We also use the logical notation $|A|$ for the number of trees in the arena $A$.

**Theorem 3.2.2** If $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$ are EAC strategies satisfying $\mathbf{l}(\sigma) \geq |A|$ (or $\sigma$ is everywhere undefined) and $\mathbf{l}(\tau) \geq |B|$ (or $\tau$ is everywhere undefined) then Algorithm 3.2.1 produces valid copycat thresholds and offsets for $\sigma ; \tau$.

The proof of this theorem may be found in the Appendix. We will also need the following:

**Lemma 3.2.3** Composition of EXAC strategies is associative.

**Proof** The proof is straightforward and we omit it. One proceeds by examining the uncovering of three strategies together (that is, the sequence including intermediate moves in both hidden arenas) and relating this to the uncoverings of the two different bracketings of the triple composition. ■

## 3.3 The Category $\mathbb{A}_{\mathrm{EXAC}}$

We first define the "obvious" category, which derives directly from the conditions required for the composition algorithm to work correctly. Perhaps surprisingly we can show that this does not give rise to a CCC.

**Definition** The *category of arenas and EXAC strategies*, written $\mathbb{A}_{\mathrm{EXAC}}$, is defined as follows.

(i) Objects are recursively enumerable arenas.

(ii) The morphisms from $A$ to $B$ are the EXAC strategies on the arena $A \Rightarrow B$ which have l-number greater than or equal to $|A|$, or are everywhere undefined.

(iii) The identity morphism on $A$ is the EXAC strategy $\langle \mathrm{id}_A, 0 \rangle$, i.e. the copycat threshold is zero everywhere. The offset at the minimal P-view is $-|A|$, and 0 everywhere else.

(iv) Composition of morphisms is given by composition of EXAC strategies via algorithm 3.2.1.

One can show that this does indeed specify a category. We already have associativity of composition, and we must also show:

**Proposition 3.3.1**

(i) If $\sigma : A \Rightarrow B$ satisfies $\mathbf{l}(\sigma) \geq |A|$ (or is everywhere undefined) and $\tau : B \Rightarrow C$ satisfies (or is everywhere undefined) $\mathbf{l}(\tau) \geq |B|$ then $\mathbf{l}(\sigma;\tau) \geq |A|$, or $\sigma;\tau$ is everywhere undefined.

(ii) For any morphism $\sigma : A \to B$ we have $\sigma; \mathrm{id}_B = \sigma = \mathrm{id}_A; \sigma$.

**Proof** Both are elementary consequences of the construction of the composition algorithm. ■

Also, $\mathbb{A}_{\mathrm{EXAC}}$ has the obvious terminal object $E$ (the arena consisting of no trees defined in section 2.1) and products given in the usual way. However,

**Proposition 3.3.2** $\mathbb{A}_{\mathrm{EXAC}}$ does not form a CCC with the usual constructions.

**Proof** Suppose that $\sigma : A \times B \to C$. Then we know that $\mathbf{l}(\sigma) \geq |A| + |B|$. We need a morphism $\Lambda(\sigma) : A \to B \Rightarrow C$, which must have l-number at least $|A|$, so we could take $\Lambda(\sigma)$ to be the same EXAC strategy as $\sigma$. However this choice may not be unique.

For example, consider $\eta_0$ and $\eta_1$ as defined earlier in this section. Note that the only difference between the two strategies is their threshold for the initial P-view. One can verify that both $\eta_0$ and $\eta_1$ can be considered as morphisms $U \to U \Rightarrow U$ and that in this case $\eta_0 \times \mathrm{id}_U; \mathrm{eval}_{U,U} = \eta_1 \times \mathrm{id}_U; \mathrm{eval}_{U,U} : U \times U \to U$, and that this is the same as the morphism $U \times U \to U$ described by $\eta_1$. Hence there are two candidates for $\Lambda(\eta_1)$. ■

We should perhaps also say that it is not clear that $\mathbb{A}_{\mathrm{EXAC}}$ forms a CCC with any unusual constructions either.

In order to fix this up, we made another attempt. The problem with $\mathbb{A}_{\mathrm{EXAC}}$ is that the conditions for an EXAC strategy to be a morphism $A \to (B \Rightarrow C)$ are weaker than those to be a morphism on $A \times B \to C$. One solution might be the following:

(i) An object is a pair $(A, n)$ where $n \in \mathbb{N}_0$.

(ii) A morphism $\sigma : (A, n) \to (B, m)$ is an EXAC strategies on $A \Rightarrow B$ such that $\mathbf{l}(\sigma) \geq m + |A|$.

(iii) Composition is composition of EXAC strategies.

(iv) The identity on $(A, n)$ is $\langle \mathrm{id}_A, t \rangle$, where $t$ is the function mapping the minimal P-view to $n$ and the others to zero.

Then we can set $(A, n) \Rightarrow (B, m) = (A \Rightarrow B, m + |A|)$, which gives the same set of morphisms $A \times B \to C$ and $A \to B \Rightarrow C$.

However, in this case, the identity will not work correctly! Sometimes the thresholds of $\sigma; \mathrm{id}$ come out greater than $\sigma$. To fix this, we find we must include information in the objects specifying minimal thresholds for the morphisms. But this breaks the function spaces again, and we have to add information specifying the minimum l-number for some other P-views, whereupon there are again problems with identities...

What we believe to be the least fixed point of the fixing-up process is presented in the next section.

### 3.4 The Category $\mathbb{XA}$

We now present a new category based on EXAC strategies, which does form a CCC. Although it is more complicated than the "almost-CCC" $\mathbb{A}_{\mathrm{EXAC}}$, it seems to be the simplest way to construct a CCC.

Firstly let us write $\mathrm{br}(A)$ for the number of branches of a tree at the root (assuming that this is finite). Then we can write $\mathrm{br}(A \mathbin{@} m)$, for any move $m$ of a finitely-branching forest $A$, to mean the number of direct children of $m$ in $A$. Then we make the following definition:

**Definition** Let $A$ be an arena and $X$ a finitely-branching recursively enumerable sub-arena[3] of $A$. We say that an EXAC strategy $\sigma$ over $A$ is *$X$-explicit* if the following holds:

[3]We say than an arena $A = \langle A_1, \ldots, A_m \rangle$ is a *subarena* of $B = \langle B_1, \ldots, B_n \rangle$ if $m = n$ and for each $i$, $A_i$ is a subset of $B_i$. We say that an arena is *finitely-branching* if every tree in it is finitely branching.

Let $\sigma : \vec{v} \mapsto (i, r, t, o)$ be the economical form of any clause of the innocent function. Suppose that the sequence $\vec{v}$ codes a P-view ending in the O-move $\underline{m}$, and that the consequent P-move encoded by this clause is $m$. Then

(i) if $\underline{m}$ is in the subarena $X$ then $t - o \geq \mathrm{br}(X \mathbin{@} \underline{m})$,

(ii) if $m$ is in the subarena $X$ then $t \geq \mathrm{br}(X \mathbin{@} m)$.

An intuitive description of this definition is the following: The subarena $X$ determines a part of the arena $A$ where the strategy is known to be explicitly defined, i.e. that is neither in the domain nor the range of automatic copycat forced by the threshold information of $\sigma$. This means that given a strategy $\sigma$ over $A$ which is $X$-explicit, any P-view of $\sigma$ with moves only in $X$ is entirely explicit.

The following lemma will be useful later.

**Lemma 3.4.1** If $A$ is an r.e. arena of which $M \Rightarrow M$ is a subarena (i.e. $A$ is a single tree r.e. arena which is not equal to $E$ or $M$) then every EXAC strategy over $A$ is $(M \Rightarrow M)$-explicit.

**Proof** Let $\sigma$ be an EXAC strategy over such an arena $A$. Either $\sigma = \perp$, in which case it is vacuously $(M \Rightarrow M)$-explicit, or it makes a move in response to the initial O-move $\varepsilon$.

The only P-view on which the conditions for $\sigma$ to be $(M \Rightarrow M)$-explicit come into play is the initial P-view (when $\mathrm{br}((M \Rightarrow M) \mathbin{@} \varepsilon) = 1$), as $\mathrm{br}((M \Rightarrow M) \mathbin{@} m) = 0$ for any P-move $m$ and $\underline{m} \notin (M \Rightarrow M)$ for any non-initial move O-move $\underline{m}$. Thus we only need to show that $\mathbf{l}(\sigma) \geq 1$. But suppose that $\mathbf{l}(\sigma) = 0$ — then by the condition (iv) of the definition of AC at the initial P-view we must have that the innocent function of $\sigma$ maps $\varepsilon$ to $(i, 0)$ with $i \leq \mathbf{l}(\sigma)$, a contradiction if $\sigma \neq \perp$. ∎

We now define a category of EXAC strategies which will form a CCC.

**Definition** The category $\mathbb{XA}_{\mathrm{EXAC}}$, or simply $\mathbb{XA}$, is given by the following:

(i) Objects are pairs $(A, X)$ consisting of a r.e. arena $A$ and a finitely-branching r.e. subarena $X$.

(ii) A morphism $\sigma : (A, X) \to (B, Y)$ is an EXAC morphism on $A \Rightarrow B$ which is $(X \Rightarrow Y)$-explicit.

(iii) Composition of morphisms is composition of EXAC strategies via Algorithm 3.2.1.

(iv) The identity strategy on $(A, X)$, $\mathrm{id}_{(A,X)}$, is the EXAC strategy $\langle \mathrm{id}_A, t \rangle$, where $\mathrm{id}_A$ is the EAC identity strategy on $A$, and $t$ is the function that takes the least value on every P-view which still leaves the EXAC strategy $\langle \mathrm{id}_A, t \rangle$ as $(X \Rightarrow X)$-explicit.

**Theorem 3.4.2** This does form a category. We already know that composition is associative so it remains to show that:

> **Composition of morphisms is well-defined:** If $\sigma : (A, X) \to (B, Y)$ and $\tau : (B, Y) \to (C, Z)$ then $\sigma; \tau$ is $(X \Rightarrow Z)$-explicit.

**Identities work as required:** If $\sigma : (A, X) \to (B, Y)$ then $\sigma; \mathrm{id}_{(B,Y)} = \sigma = \mathrm{id}_{(A,X)}; \sigma$.

**Proof**

**Composition of morphisms is well-defined:**

Take any P-view $\vec{v}$ on which the composite strategy $\sigma; \tau$ is defined. Suppose that the last move of this P-view is $\underline{m}$ and the resulting P-move is $m$.

Firstly suppose that $\underline{m} \in X \Rightarrow Z$. We will have to take the special case when $\underline{m}$ is a root of $C$ separately, so first assume that this is not the case. Then we know that either $\underline{m} \in X \Rightarrow Y$ or $\underline{m} \in Y \Rightarrow Z$, depending on whether $\sigma$ or $\tau$ makes the next (possibly hidden) move after $\underline{m}$. In either case, in the notation of the Composition Algorithm, we that $t_1 - o_1 \geq \mathrm{br}(X \Rightarrow Z \,@\, \underline{m})$ by hypothesis. But then if we examine $\mathbf{t}$ and $\mathbf{o}$, the threshold and offset of the composition at this P-view, we see that:

$$\begin{aligned}
\mathbf{t} - \mathbf{o} &= T_{p+1} - O_{p+1} \\
&\geq (t'_1 + \textstyle\sum_{i=2}^{i=p+1} o'_i) - (\sum_{i=1}^{i=p+1} o'_i) \\
&= t'_1 - o'_1 \\
&= t_1 - o_1 \\
&\geq \mathrm{br}(X \Rightarrow Z \,@\, \underline{m})
\end{aligned}$$

In the special case when $\underline{m}$ is the root of $C$, we have that $\mathbf{o} = O_{p+1} - |A| + |B|$, but also we know that the strategy making the first move after $\underline{m}$ is $\tau$, and that $\mathrm{br}(X \Rightarrow Z \,@\, \underline{m}) = \mathrm{br}(Y \Rightarrow Z \,@\, \underline{m}) + |A| - |B|$, so the above reasoning is still sound.

Secondly, suppose that $m \in X \Rightarrow Z$. There are no special cases; we always have that $t'_{p+1} = t_{p+1} \geq \mathrm{br}(X \Rightarrow Z \,@\, m)$ by hypothesis (this is because $m$ cannot be a root of $B$ or $C$). But $\mathbf{t} = T_{p+1} \geq t'_{p+1}$ so that $\mathbf{t} \geq \mathrm{br}(X \Rightarrow Z \,@\, m)$.

Hence $\sigma; \tau$ is $X \Rightarrow Z$-explicit.

**Identities work as required:**

It is simple to verify that the identity on $(A, X)$ has the following economical form. Suppose that $A = \langle A_1, \ldots, A_n \rangle$, so that $\mathrm{id}_{(A,X)} = \langle \mathrm{id}_1, \ldots, \mathrm{id}_n \rangle$, say. Then:

$$\begin{array}{rrcl}
\mathrm{id}_i : & \varepsilon & \mapsto & (i, 0, \mathrm{br}(X_i \,@\, \varepsilon), -n) \\
& t & \mapsto & (t + n, 1, \mathrm{br}(X_i \,@\, t), 0) \\
& \vec{s} \cdot t & \mapsto & (t, 1, \mathrm{br}(X_i \,@\, (\vec{s} \cdot t)), 0) \text{ for all nonempty sequences } \vec{s}
\end{array}$$

Here we have extended the definition of br slightly, to have that $\mathrm{br}(A \,@\, m) = 0$ when $m$ is not in $A$.

That is, the copycat threshold of $\mathrm{id}_{(A,X)}$ at a P-view ending in the O-move coded by $\underline{m}$ in either of the two copies of the arena $A$ (which copy it will be depends on the parity of the length of $\underline{m}$) is the number of children of $\underline{m}$ in $X$, or zero if $\underline{m}$ is not in $X$.

Now we know that for an EAC strategy $\sigma$ on $A \Rightarrow B$, $\mathrm{id}_A; \sigma = \sigma = \sigma; \mathrm{id}_B$, where $\mathrm{id}_A$ and $\mathrm{id}_B$ are the EAC identity strategies. It remains to show that this still holds when one also includes copycat thresholds and offsets. We will only consider $\mathrm{id}_A; \sigma$, as the other proof is very similar and uses no additional techniques.

Let us select some P-view $\vec{v}$ on which $\mathrm{id}_A; \sigma$ is defined, and suppose that it ends in the O-move $\underline{m}$, with $m$ the resulting P-move. There are four cases:

(i) Both $\underline{m}$ and $m$ are in $B$.
(ii) $\underline{m}$ is in $B$, $m$ is in some component $(A, a)$.
(iii) $m$ is in $B$, $\underline{m}$ is in some component $(A, a)$.
(iv) $\underline{m}$ is in $(A, a_1)$, $m$ is in $(A, a_2)$.

Let us write $t_\sigma$ and $o_\sigma$ for the copycat threshold and offset of $\sigma$ at the same P-view $\vec{v}$. In each case we want to show that these match the copycat threshold and offset of the composition. In what follows "condition (i) " (or (ii)) refers to the fact that $\sigma$ is $X \Rightarrow Y$-explicit, at the P-view $\vec{v}$.

(i) Is completely trivial.
(ii) Examining the way the EAC identity strategies work (simply copycat) we see that the uncovering must be of the form

$$\cdots \begin{array}{ccc} \bullet & \star & \circ \\ \underline{m} & m_1 & m \end{array}$$

where $m$ is the same move in $A$ as $m_1$, since it was arrived at by copycat. $\sigma$ made the move $m_1$ in response to the P-view $\vec{v}$.

There are three cases. Let us first assume that $\underline{m}$ is not a root of $B$ and $m$ (equivalently $m_1$) not a root of $A$. Then, in the notation of the Composition Algorithm, we have:

$$\begin{array}{ll} t'_1 = t_\sigma & o'_1 = o_\sigma \\ t'_2 = \mathrm{br}(X @ m_1) & o'_2 = 0 \\ T_1 = t_\sigma & O_1 = o_\sigma \\ T_2 = \max(t_\sigma, \mathrm{br}(X @ m_1)) \stackrel{*}{=} t_\sigma & O_2 = o_\sigma \end{array}$$

(* by condition (ii)) Hence $\mathbf{t} = t_\sigma$ and $\mathbf{o} = o_\sigma$ as required.

In the case when $\underline{m}$ is not a root of $B$ but $m$ is a root of $A$, we have similarly:

$$\begin{array}{ll} t'_1 = t_\sigma + |A| & o'_1 = o_\sigma + |A| \\ t'_2 = \mathrm{br}(X @ m_1) & o'_2 = -|A| \\ T_1 = t_\sigma + |A| & O_1 = o_\sigma + |A| \\ T_2 = \max(t_\sigma, \mathrm{br}(X @ m_1)) \stackrel{*}{=} t_\sigma & O_2 = o_\sigma \end{array}$$

(* by condition (ii))

In the case when $\underline{m}$ is a root of $B$, we must have that $m$ is a root of $A$, and the calculation above applies.
(iii) Very similar to (ii), with no special cases.
(iv) As before, we see that the uncovering must be of the form

$$\cdots \begin{array}{cccc} \bullet & \star & \star & \circ \\ \underline{m} & m_1 & m_2 & m \end{array}$$

where $\underline{m}$ is the same move in $A$ as $m_1$, and $m_2$ is the same move in $A$ as $m$. The move $m_2$ is made by $\sigma$ in response to the P-view $\vec{v}$.

There are two cases this time. First suppose that $m_2$ is not a root of $A$. Then:

$$\begin{array}{ll}
t'_1 = \mathrm{br}(X \mathbin{@} \underline{m}) = \mathrm{br}(X \mathbin{@} m_1) & o'_1 = 0 \\
t'_2 = t_\sigma & o'_2 = o_\sigma \\
t'_3 = \mathrm{br}(X \mathbin{@} m_2) & o'_3 = 0 \\
T_1 = \mathrm{br}(X \mathbin{@} m_1) & O_1 = 0 \\
T_2 = \max(\mathrm{br}(X \mathbin{@} m_1) + o_\sigma, t_\sigma) \stackrel{*}{=} t_\sigma & O_2 = o_\sigma \\
T_3 = \max(t_\sigma, \mathrm{br}(X \mathbin{@} m_2)) \stackrel{**}{=} t_\sigma & O_3 = o_\sigma
\end{array}$$

(* by condition (i), ** by condition (ii)). If $m_2$ is a root of $A$ then:

$$\begin{array}{ll}
t'_1 = \mathrm{br}(X \mathbin{@} \underline{m}) = \mathrm{br}(X \mathbin{@} m_1) & o'_1 = 0 \\
t'_2 = t_\sigma + |A| & o'_2 = o_\sigma + |A| \\
t'_3 = \mathrm{br}(X \mathbin{@} m_2) & o'_3 = -|A| \\
T_1 = \mathrm{br}(X \mathbin{@} m_1) & O_1 = 0 \\
T_2 = \max(\mathrm{br}(X \mathbin{@} m_1) + o_\sigma + |A|, t_\sigma + |A|) \stackrel{*}{=} t_\sigma + |A| & O_2 = o_\sigma + |A| \\
T_3 = \max(t_\sigma, \mathrm{br}(X \mathbin{@} m_2)) \stackrel{**}{=} t_\sigma & O_3 = o_\sigma
\end{array}$$

(* by condition (i), ** by condition (ii)). So either way the result holds.

■

**Proposition 3.4.3** The following constructions make $\mathbb{XA}$ into a CCC:

**Terminal Object:** The terminal object is $(E, E)$.

**Binary Products:** The product $(A, X) \times (B, Y)$ is $(A \times B, X \times Y)$. The projection strategy $\pi^{(A,X)\times(B,Y)}_{(A,X)}$ is given by the EXAC strategy $\langle \pi^{A\times B}_A, t\rangle$, where $\pi^{A\times B}_A$ is the EAC projection strategy and $t$ is the threshold function specifying the least value at each P-view to make this EXAC strategy $((X \times Y) \Rightarrow X)$-explicit. Similarly for the other projection.

**Exponentials:** The exponential object $(A, X) \Rightarrow (B, Y)$ is $(A \Rightarrow B, X \Rightarrow Y)$. The evaluation map $\mathrm{eval}_{(B,Y),(C,Z)}$ is the same EXAC strategy as $\mathrm{id}_{(B,Y)\Rightarrow(C,Z)}$.

**Proof**

**Terminal Object:**

The only morphisms with the terminal object as codomain are clearly the 0-tuples.

**Binary Products:**

Suppose that $A = \langle A_1, \ldots, A_m\rangle$ and $B = \langle B_1, \ldots, B_n\rangle$, so that $\pi^{(A,X)\times(B,Y)}_{(A,X)}$ is an $m$-tuple, $\langle \pi_1, \ldots, \pi_m\rangle$.

Then it is easy to check that $\pi_i$ has the following economical form:

$$\begin{array}{rrcl}
\pi_i : & \varepsilon & \mapsto & (i, 0, \mathrm{br}(X_i \mathbin{@} \varepsilon), -n-m) \\
& t & \mapsto & (t+n+m, 1, \mathrm{br}(X_i \mathbin{@} t), 0) \\
& \vec{s}\cdot t & \mapsto & (t, 1, \mathrm{br}(X_i \mathbin{@} (\vec{s}\cdot t)), 0) \text{ for all nonempty sequences } \vec{s}
\end{array}$$

We claim that give any morphisms $\sigma = \langle \sigma_1, \ldots, \sigma_m \rangle : (C, Z) \to (A, X)$ and $\tau = \langle \tau_1, \ldots, \tau_n \rangle : (C, Z) \to (B, Y)$, we have $\langle \sigma_1, \ldots, \sigma_m, \tau_1 \ldots, \tau_n \rangle ; \pi^{(A,X)\times(B,Y)}_{(A,X)} = \sigma$. We already know that this holds for the EAC strategy part of the EXAC strategies under consideration, and the fact that it also holds for the threshold functions is almost entirely similar to the proof that identities work as required when composed on the right.

There is only one case where the copycat thresholds and offsets of $\pi^{(A,X)\times(B,Y)}_{(A,X)}$ differ from those of $\mathrm{id}_{(A,X)}$, namely the initial P-views, so this leaves two cases to check.

Either the first move of $\sigma$ is in $A$ or $C$. If the former, the interaction sequence of the first move of the composition is:

$$\ldots \begin{array}{cccc} \bullet & \star & \star & \circ \\ \underline{m} & m_1 & m_2 & m \end{array}$$

where $\underline{m}$ is a root of $A$, $m_1$ the corresponding root of $A \times B$, and $m_2$ and $m$ are the first $A$-move made by $\sigma$.

Then using the same techniques as above we can show that, if $t_\sigma$ and $o_\sigma$ are the threshold and offset of $\sigma$ at the initial P-view, we have:

$$\begin{array}{ll}
t'_1 = \mathrm{br}(X \mathbin{@} \varepsilon) + |C| = \mathrm{br}((Z \Rightarrow X) \mathbin{@} \varepsilon) & o'_1 = -n - m + |C| \\
t'_2 = t_\sigma & o'_2 = o_\sigma \\
t'_3 = \mathrm{br}(X \mathbin{@} m_2) & o'_3 = 0 \\
T_1 = \mathrm{br}(X \mathbin{@} \varepsilon) + |C| & O_1 = -n - m + |C| \\
T_2 = \max(\mathrm{br}(X \mathbin{@} \varepsilon) + |C| + o_\sigma, t_\sigma) \stackrel{*}{=} t_\sigma & O_2 = o_\sigma + |C| - n - m \\
T_3 = \max(t_\sigma, \mathrm{br}(X \mathbin{@} m_2)) \stackrel{**}{=} t_\sigma & O_3 = o_\sigma + |C| - n - m
\end{array}$$

(* by condition (i), ** by condition (ii)). So, because we are in the special case where $\underline{m}$ is the root, $\mathbf{t} = t_\sigma$ and $\mathbf{o} = o_\sigma$.

In the other case, the first move made by $\sigma$ is immediately visible, so the interaction sequence looks like:

$$\ldots \begin{array}{ccc} \bullet & \star & \circ \\ \underline{m} & m_1 & m \end{array}$$

where $\underline{m}$ a the root of $A$, $m_i$ the corresponding root of $A \times B$, and $m$ the root of $C$.

$$\begin{array}{ll}
t'_1 = \mathrm{br}(X \mathbin{@} \varepsilon) + |C| = \mathrm{br}((Z \Rightarrow X) \mathbin{@} \varepsilon) & o'_1 = -n - m + |C| \\
t'_2 = t_\sigma & o'_2 = o_\sigma \\
T_1 = \mathrm{br}(X \mathbin{@} \varepsilon) + |C| & O_1 = -n - m + |C| \\
T_2 = \max(\mathrm{br}(X \mathbin{@} \varepsilon) + |C| + o_\sigma, t_\sigma) \stackrel{*}{=} t_\sigma & O_2 = o_\sigma + |C| - n - m
\end{array}$$

(* by condition (i)). So in either case $\mathbf{t} = t_\sigma$ and $\mathbf{o} = o_\sigma$.

Since the strategy $\langle \sigma_1, \ldots, \sigma_m, \tau_1 \ldots, \tau_n \rangle$ is clearly $Z \Rightarrow (X \times Y)$-explicit, this is sufficient to prove that this works as the tupled morphism, and also the required universal property of the projection. The proof for the other projection is, of course, entirely similar.

**Exponentials:**

If $\sigma$ is any morphism $(A, X) \to ((B, Y) \Rightarrow (C, Z))$ we claim that the composition $\sigma \times \mathrm{id}_{(B,Y)} ; \mathrm{eval}_{(B,Y),(C,Z)}$ is a morphism $(A, X) \times (B, Y) \to (C, Z)$ which is the same

EXAC strategy as $\sigma$. Hence for $\tau : (A, X) \times (B, Y) \to (C, Z)$, the curried morphism $\Lambda(\tau)$ is uniquely given by the same EXAC strategy as $\tau$.

As before, we already know this result to be true as far as the moves of the strategies go, and it remains to check the thresholds and offsets. The proof contains a considerable amount of tedious detail, all in a similar vein to that shown above, of which we give an outline.

Firstly, we note that $\sigma \times \mathrm{id}_{(B,Y)}$ is shorthand for $\langle \pi^{(A,X)\times(B,Y)}_{(A,X)}; \sigma, \pi^{(A,X)\times(B,Y)}_{(B,Y)}; \mathrm{id}_{(B,Y)}\rangle$. Let us write this as $\langle \sigma', \pi^{(A,X)\times(B,Y)}_{(B,Y)}\rangle$. Recall that the thresholds and offsets of the projection morphisms do not differ very much from those of the relevant identity morphisms: routine calculation shows that $\sigma'$ has exactly the same thresholds as $\sigma$, and also the same offsets, except that the offset at any minimal P-view is decreased by $|B|$.

Then it remains to show that, at any P-view $\vec{v}$ of the composition

$$\langle \sigma', \pi^{(A,X)\times(B,Y)}_{(B,Y)}\rangle; \mathrm{eval}_{(B,Y),(C,Z)},$$

the copycat threshold and offset matches that of $\sigma$. As before, suppose that the P-view ends in the move $\underline{m}$ and has resulting move $m$, and write $t_\sigma$ and $o_\sigma$ for the copycat threshold and offset of $\sigma$ at $\vec{v}$.

There are nine cases, corresponding to whether $\underline{m}$ and $m$ are in the arenas $A$, $B$ or $C$. Each gives a different interaction sequence, and requires a separate proof. Some cases have special subcases, when $\underline{m}$ or one of the intermediate moves is a root of some tree. All cases, although sometimes long, are very similar to the proofs already given for identities and projections. ■

# 4 A Universal Model of Böhm Tree Equality

Now that we have described an ambient Cartesian Closed Category based on EXAC strategies, we show how this does give rise to a $\lambda$-algebra $\mathcal{D}_{\mathrm{XA}}$ with precisely the equational theory we intended. Many properties of the model $\mathcal{D}_{\mathrm{EAC}}$ also hold for $\mathcal{D}_{\mathrm{XA}}$, but $\mathcal{D}_{\mathrm{XA}}$ not a $\lambda$-model, i.e. is not weakly extensional.

## 4.1 The Model $\mathcal{D}_{\mathrm{XA}}$

Recall the single-tree arenas $U$ and $M$ defined in Section 2.1. The former is "maximal", in that it is (countably) infinitely branching and infinitely deep. The latter is "minimal", consisting of one a single node.

Let us write $U_0$ for the object $(U, M)$ of $\mathbb{XA}$, and $U_1$ for the object $U_0 \Rightarrow U_0$. The reader may wish to verify that, for example, the concrete representation of $U_1$ is given by $(U, \langle\{\varepsilon, \langle 1\rangle\}\rangle)$.

Recall also the EXAC strategy $\eta_1$ from Section 3.1. We repeat the definition for convenience: the EAC strategy part is the same as the strategy $\mathrm{id}_U$ (since $U = U \Rightarrow U$ this does define an EAC strategy over $U$). The copycat threshold is zero at every P-view except the initial P-view, when it is 1. Thus the economical form is given by:

$$\begin{array}{rcl}\varepsilon & \mapsto & (1,0,1,-1)\\ \langle 1\rangle & \mapsto & (2,1,0,0)\end{array}$$

It is routine to check that $\eta_1$ specifies two morphisms of $\mathbb{XA}$, $f : U_0 \to U_1$ and $g : U_1 \to U_0$. (In fact it is the case that these morphisms could equally be specified as the EXAC strategies which have EAC strategy part $\mathrm{id}_U$, and the least copycat thresholds to make them explicit in the necessary subarenas to be morphisms of that type. The definition of morphisms by EAC strategy and "least threshold function" to make them explicit in the appropriate subarenas seems to be a recurring theme.)

We can now show that $f$ and $g$ form a retraction from $U_1$ into $U_0$: we know that that EAC strategy part of $f$, $g$, $\mathrm{id}_{U_0}$ and $\mathrm{id}_{U_1}$ are all the same as $\mathrm{id}_U$, so that same holds for $f;g$ and $g;f$. Thus it remains only to check threshold functions. Simple applications of the Composition Algorithm show that the thresholds of both $f;g$ and $g;f$ are 1 at the minimal P-view, and zero elsewhere. The same holds for $\mathrm{id}_{U_1}$, whereas $\mathrm{id}_{U_0}$ has threshold which is zero at every P-view. Hence, as EXAC strategies and thus as morphisms in $\mathbb{XA}$,

$$g;f = \mathrm{id}_{U_1}, f;g \neq \mathrm{id}_{U_0}.$$

Thus allows us to identify a $\lambda$-algebra $\mathcal{M}(\mathbb{XA}, U_0, f, g)$, which we shall write as $\mathcal{D}_{\mathrm{XA}}$. To distinguish the denotation of a term as an EAC strategy in $\mathcal{D}_{\mathrm{EAC}}$ (which is the same as the denotation in $\mathcal{D}$ and $\mathcal{D}_{\mathrm{REC}}$), from the denotation as an EXAC strategy in this model we write it $[\![s]\!]^{\mathrm{XA}}$.

We will want to lift some results about EAC strategies and the model $\mathcal{D}_{\mathrm{EAC}}$ to EXAC strategies and the model $\mathcal{D}_{\mathrm{XA}}$. We make that connection precise with the following.

**Theorem 4.1.1** Let $E : \mathbb{XA} \to \mathbb{A}_{\mathrm{EAC}}$ be given on objects by $E((A, X)) = A$ and on morphisms by $E(\langle \sigma, t_\sigma \rangle) = \sigma$. Then

(i) $E$ is a full, exact cartesian closed, functor, and

(ii) $E$ preserves the reflexive object and the retraction morphisms.

Hence for any term $s$ and valuation $\rho$, $[\![s]\!]_\rho = E([\![s]\!]^{\mathrm{XA}}_\rho)$.

**Proof** (i) is straightforward verification (for example the projections in $\mathbb{XA}$ are defined to be precisely the projections in $\mathbb{A}_{\mathrm{EAC}}$, along with the least copycat threshold to make them explicit in the appropriate subarena).

To show that $E$ is full suppose that $\sigma : A \to B$ is an arrow of $\mathbb{A}_{\mathrm{EAC}}$. (Suppose that neither $A$ nor $B$ are equal to the empty arena $E$, these are special cases dealt with below). Then $\sigma$ is an EAC strategy on $A \Rightarrow B$ and hence has some recursive valid threshold function $t_\sigma$, so $\langle \sigma, t_\sigma \rangle$ is an EXAC strategy over $A \Rightarrow B$. Furthermore by Lemma 3.4.1 this EXAC strategy is $(M \Rightarrow M)$-explicit hence it is a morphism $(A, M) \to (B, M)$ of $\mathbb{XA}$. (The special cases are as follows: if $A$ is the empty arena $E$ then the EXAC strategy above will vacuously be $(E \Rightarrow M)$-explicit, symmetrically for $B = E$, and if $A = B = E$ then $\sigma = \perp$, which is vacuously $(E \Rightarrow E)$-explicit.)

For (ii), clearly $E(U_0) = U$, $E(f) = \mathit{Fun}$ and $E(g) = \mathit{Gr}$, and we know that the $\lambda$-algebra $\mathcal{M}(\mathbb{C}, R, \mathit{Fun}, \mathit{Gr})$ (where $R$ is a reflexive object via $\mathit{Fun}$ and $\mathit{Gr}$ in the CCC $\mathbb{C}$)

is completely determined by cartesian closed structure of $\mathbb{C}$, along with $R$, *Fun* and *Gr*. ■

**Corollary 4.1.2** $\mathcal{D}_{\mathrm{XA}}$ is sensible. $[\![s]\!]^{\mathrm{XA}} = \bot$ (the everywhere undefined EXAC strategy) if and only if $s$ is unsolvable.

## 4.2 Böhm Trees in Variable-Free Form and Exact Correspondence

The Exact Correspondence Result proved in [KNO01] showed that the denotation of term, in the models $\mathcal{D}$, $\mathcal{D}_{\mathrm{REC}}$ and $\mathcal{D}_{\mathrm{EAC}}$, has a very close connexion with the term's Nakajima tree. Here we will show that the same connexion exists between the denotation of a term in $\mathcal{D}_{\mathrm{XA}}$ and the Böhm tree of the term.

Recall that the *Böhm tree* of term $s$, written $\mathrm{BT}(s)$ is given (informally) by the following: if $s$ is unsolvable then $\mathrm{BT}(s) = \bot$. If $s$ has HNF $\lambda x_1 \ldots x_n . y s_1 \ldots s_m$ then

$$\mathrm{BT}(s) = \quad \begin{array}{c} \lambda x_1 \ldots x_n . y \\ \diagup \qquad \diagdown \\ \mathrm{BT}(s_1) \;\; \cdots \;\; \mathrm{BT}(s_m) \end{array}$$

Thus each node of the Böhm tree has a finite number of abstractions, a head variable, and a finite number of children. We consider Böhm tree modulo $\alpha$-conversion, so that the the following information is sufficient to describe a Böhm tree of a closed term: for each node we have numbers specifying the number of abstractions and children, and information to describe where head variable was abstracted in the tree in the same was as we did for Nakajima trees.

We encode this information into a *variable-free form* in the following manner:

**Definition** For a $(\mathbb{N} \times \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{Z})$-labelled tree $p$ the tree $p^*$ is the same tree labelled identically, except that nodes at depth $d$ labelled $(i, d+1, t, o)$ are relabelled $(i, d+2, t, o)$.

Similarly the tree $\{p\}^n$, for $n \in \mathbb{N}_0$, is labelled identically except that firstly the node at the root $(i, r, t, o)$ is first relabelled to $(i, r, t, o-n)$, and then nodes of depth $d$ are relabelled as follows:

(i) those labelled $(i, d, t, o)$ are relabelled $(i+n, d, t, o)$;

(ii) those labelled $(i, d+1, t, o)$ for $i \leq n$ are relabelled $(n-i+1, d, t, o)$;

(iii) those labelled $(i, d+1, t, o)$ for $i > n$ are relabelled $(i-n, d+1, t, o)$.

For a term $s$ with free variables within $\Delta$ the *variable-free form* of the Böhm tree of

$s$, $\mathrm{VFBT}_\Delta(s)$, is the following $(\mathbb{N} \times \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{Z})$-labelled tree:

$$\begin{array}{rcl}
\mathrm{VFBT}_\Delta(s) & = & \bot, \qquad \text{the empty tree, for unsolvable } s. \\
\mathrm{VFBT}_\Delta(\lambda x_1 \dots x_n.s) & = & \{\mathrm{VFBT}_{\Delta\cdot\langle x_1,\dots,x_n\rangle}(s)\}^n, \\
& & \qquad \text{if } s \text{ is of the form } v_j s_1 \dots s_m. \\
\mathrm{VFBT}_\Delta(v_j s_1 \dots s_m) & = & \begin{array}{c} (j,1,m,m) \\ \diagup \qquad \diagdown \\ \mathrm{VFBT}_\Delta(s_1)^* \qquad \mathrm{VFBT}_\Delta(s_m)^* \end{array} \\
& & \text{where } \Delta = \langle v_k, \dots, v_1\rangle \text{ (note the reverse order).}
\end{array}$$

**Lemma 4.2.1** The encoding of head variables in VFBT matches that in VFF. Precisely, for any term $s$ with free variables within $\Delta$ and any sequence $\vec{v} \in \mathbb{N}^*$ at which (the labelling function of) $\mathrm{VFBT}_\Delta(s)$ is defined we have $\mathrm{VFBT}_\Delta(s)(\vec{v}) = (i,r,t,o) \implies \mathrm{VFF}_\Delta(s)(\vec{v}) = (i,r)$.

Furthermore, for any $k$, $\{\mathrm{VFBT}_\Delta(s)(\vec{v})\}^k = (i,r,t,o) \implies \{\mathrm{VFF}_\Delta(s)(\vec{v})\}^k = (i,r)$.

**Proof** Trivial when one compares the definitions of VFBT and VFF. ∎

This definition is at first sight rather opaque, and indeed it could have been stated in a clearer fashion but that this would have complicated Theorem 4.2.4 below. The effect of the definition is illustrated by the following lemma, which is analogous to Lemma 4.2.1 of [KNO01].

**Lemma 4.2.2** Let $s$ be a term with all free variables occurring within $\Delta = \langle v_k, \dots, v_1\rangle$. Construct the Böhm tree of $s$, and rename all the bound variables so that if $\vec{a} \in \mathbb{N}^*$ codes a node of $\mathrm{BT}(s)$ then the $i^{th}$ abstracted variable at this node is $x_i^{\vec{a}}$. Let this renamed Böhm tree have labelling function $A$, and consider $\mathrm{VFBT}_\Delta(s)$ also as a labelling function.

Then for any sequence $\vec{a} = \langle a_1, \dots, a_p\rangle$ there are three possibilities for $A(\vec{a})$:

(i) If $\vec{a} \notin \mathrm{dom}(A)$ then $\mathrm{VFBT}_\Delta(s)$ is unlabelled or undefined at $\vec{a}$,

(ii) If $A(\vec{a}) = \lambda x_1^{\vec{a}} \dots x_n^{\vec{a}}.v_j$, and the node coded by $\vec{a}$ has $m$ children, then $\mathrm{VFBT}_\Delta(s) = (j, p+1, m, m-n)$.

(iii) If $A(\vec{a}) = \lambda x_1^{\vec{a}} \dots x_n^{\vec{a}}.x_j^{\langle a_1,\dots,a_{p-r}\rangle}$, and the node coded by $\vec{a}$ has $m$ children, then $\mathrm{VFBT}_\Delta(s) = (j, r, m, m-n)$.

**Proof** Straightforward, and entirely similar to the proof of Lemma 4.2.1 in [KNO01]. ∎

**Example** One may check that $\mathrm{VFBT}(I)$ and $\mathrm{VFBT}(1)$ are:

$$(1,0,0,-1) \qquad \text{and} \qquad \begin{array}{c} (1,0,1,-1) \\ | \\ (2,1,0,0) \end{array}$$

The node $\lambda xy.x$, in the Böhm tree of 1, corresponds to the node of $\mathrm{VFBT}(1)$ labelled $(1,0,1,-1)$, which is so labelled because the head variable is the first abstracted variable zero levels up the tree (namely $x$), the node has one child, and the number of abstractions at this level is $1-(-1)=2$.

The following result will be useful in what follows.

**Lemma 4.2.3** If $\sigma : (A,X) \to U_1$ is a morphism in $\mathbb{XA}$, for any object $(A,X)$, then $\sigma; g : (A,X) \to U_0$ is the same EXAC strategy as $\sigma$.

If $\tau : (A,X) \to U_0$ is a morphism in $\mathbb{XA}$, for any object $(A,X)$, then $\tau; f : (A,X) \to U_1$ is the same EXAC strategy as $\tau$, unless the threshold and offset of $\tau$ at the minimal P-view, $t$ and $o$, satisfy $t-o=|A|$. In this case $\tau; f$ is the same EXAC strategy as $\tau$ except that the threshold at the minimal P-view is $t+1$.

**Proof** Recall that both $f$ and $g$ are the EXAC strategy $\eta_1$, which is the same as the EXAC strategy $\mathrm{id}_{U_0}$, except that the threshold at the minimal P-view is 1 rather than 0. Hence we can use the proof that identities work as required in Theorem 3.4.2 to show that $\sigma; f$ and $\tau; g$ are the same EXAC strategies as $\sigma$ and $\tau$ respectively, except at the minimal P-view.

It remains to examine the minimal P-view. Suppose that the first move of $\sigma$ is in the arena $A$ rather than $U$, with copycat threshold and offset $t_\sigma$ and $o_\sigma$. Then, in the notation of the Composition Algorithm,

$$\begin{array}{ll}
t'_1 = 1+|A| & o'_1 = -1+|A| \\
t'_2 = t_\sigma & o'_2 = o_\sigma \\
T_1 = 1+|A| & O_1 = -1+|A| \\
T_2 = \max(1+|A|+o_\sigma, t_\sigma) \stackrel{*}{=} t_\sigma & O_2 = o_\sigma - 1 + |A|
\end{array}$$

Recall that $U_1 = (U, M \Rightarrow M)$ . Then (*) is because $\sigma$ must be $(X \Rightarrow (M \Rightarrow M))$-explicit, hence $t_\sigma - o_\sigma \geq |A|+1$.

Thus $\mathbf{t} = t_\sigma$ and $\mathbf{o} = o_\sigma$, i.e. the threshold and offset of $\sigma; f$ at the minimal P-view match that of $\sigma$. A similar calculation applies when the first move of $\sigma$ is in the arena $U$.

The same figures occur in the calculation of $\tau; g$ at the minimal P-view, except that in this case we only know that $\tau$, as a morphism $(A,X) \to U_0$, must be $(X \Rightarrow M)$-explicit, so that $t_\tau - o_\tau \geq |A|$. When equality holds, $\mathbf{t} = t_\tau + 1$, when it is a strict inequality, $\mathbf{t} = t_\tau$ as for $\sigma$. ■

We use this to show a powerful connexion between the denotations of terms in $\mathcal{D}_{\mathrm{XA}}$ and Böhm trees.

**Theorem 4.2.4 (Exact Correspondence for $\mathcal{D}_{\mathrm{XA}}$)** If $s \in \Lambda$ with free variables in $\Delta = \langle v_k, \ldots, v_1 \rangle$ then $[\![s]\!]^{\mathrm{XA}}_{\Delta} = \{\mathrm{VFBT}_\Delta(s)\}^k$ when the former is considered as an EXAC strategy in economical form and the latter as a labelling function.

In particular for closed terms $s$, $[\![s]\!]_\varepsilon = \mathrm{VFBT}_\varepsilon(s)$

**Proof** We show by induction on the length of $\vec{\alpha}$, for all terms $s$ and contexts $\Delta$

simultaneously, that

(i) $\vec{\alpha} \cdot i \in \mathrm{dom}([\![s]\!]^{\mathrm{XA}}_{\Delta})$ if and only if $\vec{\alpha} \cdot i \in \mathrm{dom}(\mathrm{VFBT}_{\Delta}(s))$. The latter is trivially equal to $\mathrm{dom}(\{\mathrm{VFBT}_{\Delta}(s)\}^k)$.

(ii) If $[\![s]\!]^{\mathrm{XA}}_{\Delta}(\vec{\alpha}) = (i, r, t, o)$ and $\{\mathrm{VFBT}_{\Delta}(s)\}^k(\vec{\alpha}) = (i', r', t', o')$ then $i' = i, r' = r$ and $t' = t$.

We will be able to use the Exact Correspondence Theorem for $\mathcal{D}$, together with Lemma 4.2.1 and Theorem 4.1.1 to prove that $i' = i$ and $r' = r$, and then show $t' = t$ by considering the composition algorithm. In view of Remark 3.1.1 this will ensure that $o' = o$ too (in the base case $o' = o$ comes for free, but the proof of the inductive step is easier without having to consider offsets).

**Base Case:** If $s$ is unsolvable then both $[\![s]\!]^{\mathcal{M}}_{\Delta}$ and $\mathrm{VFBT}_{\Delta}(s)$ are everywhere undefined.

Otherwise $s$ has a head normal form $\lambda x_1 \dots x_n . y s_1 \dots s_m$. Then in the notation above $t' = m$ and $o' = m - n - k$.

We must return to the definition of $[\![-]\!]^{\mathrm{XA}}$ to discover the copycat thresholds and offsets.

$$[\![s]\!]^{\mathrm{XA}}_{\Delta} = \underbrace{\Lambda(\cdots\Lambda(\Lambda}_{n\ \Lambda\text{'s}}([\![y s_1 \dots s_m]\!]^{\mathrm{XA}}_{\Gamma}); g); g \cdots); g$$

where $\Gamma = \Delta \cdot \langle x_1, \dots, x_n \rangle$. Since $\Lambda(\sigma)$ is the same EXAC strategy as $\sigma$, and by Lemma 4.2.3 so is $\sigma; g$, this has the same thresholds and offsets as $[\![y s_1 \dots s_m]\!]^{\mathrm{XA}}_{\Gamma}$. Let $V = \underbrace{U_0 \times \cdots \times U_0}_{k+n\text{ times}}$.

$$[\![y s_1 \dots s_m]\!]^{\mathrm{XA}}_{\Gamma} = (\Pi^{\Gamma}_{y} \bullet [\![s_1]\!]^{\mathrm{XA}}_{\Gamma}) \cdots \bullet [\![s_m]\!]^{\mathrm{XA}}_{\Gamma}$$

We will show, by induction on $m$, that the threshold of this strategy at the minimal P-view is $m$ and the offset is $m - n - k$. For convenience we also include the fact that the first P-move of the composite strategy is played in the arena $V$ in the induction hypothesis.

*Case $m = 0$:* The strategy is just $\Pi^{\Gamma}_{y}$. Depending on $y$, this is just one of the projections $V \to U_0$, which we know has threshold 0 and offset $-n - k$ at the minimal P-view. We also know that the first P-move is played in the arena $V$.

*Inductive Case* Suppose that we have a strategy $\sigma : V \to U_0$ with threshold $m$ and offset $m - n - k$ at the minimal P-view, and which makes the first P-move in the arena $V$. Then for any strategy $\tau : V \to U_0$,

$$\sigma \bullet \tau = \langle \sigma; f, \tau \rangle; \mathrm{eval}_{U_0,U_0}.$$

First examine $\sigma; f$ — we are in the special case of Lemma 4.2.3, so that the threshold of $\sigma; f$ is $m + 1$ and the offset $m - n - k$.

It is now simple to use the Composition Algorithm to examine the threshold and offset of $\langle \sigma; f, \tau \rangle; \mathrm{eval}_{U_0,U_0}$ at the minimal P-view. Since we know that the first P-move of $\sigma$ is in the arena $V$, there is one intermediate move, which is the root of $V \Rightarrow U_1$ at which $\sigma$ is to play. Thus the strategy $\tau$ is irrelevant for this calculation and in the notation of the Composition Algorithm the resulting threshold and offset is calculated as follows:

$$
\begin{array}{rclrcl}
t_1' & = & 1+k+n & o_1' & = & -1+k+n \\
t_2' & = & m+1 & o_2' & = & m-n-k \\
T_1 & = & 1+k+n & O_1 & = & -1+k+n \\
T_2 & = & \max(m+1, 1+k+n+m-n-k) & O_2 & = & -1+k+n+m-n-k \\
& = & m+1 & & = & m-1
\end{array}
$$

Hence $\mathbf{t} = m+1$ and $\mathbf{o} = m-1-(n+k)+2 = m+1-n-k$. One can also see that the first move of the composition is in the arena $V$, which completes the inductive step of this claim.

This completes the proof that the threshold and offset of $[\![s]\!]_\Delta^{\mathrm{XA}}$ at the minimal P-view are $m$ and $m-n-k$ respectively.

This shows that $\langle i\rangle \in \mathrm{dom}([\![s]\!]_\Delta^{\mathrm{XA}})$ if and only if $1 \le i \le m$ and $s_i$ is solvable. On the other hand, $\langle i\rangle \in \mathrm{dom}(\mathrm{VFBT}_\Delta(s))$ if and only if $1 \le i \le m$ and $s_i$ is solvable. This completes the proof of (i).

Suppose that $[\![s]\!]_\Delta^{\mathrm{XA}}(\varepsilon) = (i,r,t,o)$ and $\{\mathrm{VFBT}_\Delta(s)\}^k(\varepsilon) = (i',r',t',o')$. Now Lemma 4.2.1 means that $\{\mathrm{VFF}_\Delta(\varepsilon)\}^k(\varepsilon) = (i',r')$. On the other hand, Theorem 4.1.1 means that $[\![s]\!]_\Delta(\varepsilon) = (i,r)$, and the Exact Correspondence Theorem for $\mathcal{D}_{\mathrm{EAC}}$ gives that $i'=i$ and $r'=r$.

Finally, by the definition of VFBT, $t'=m=t$ and $o'=m-n-k=o$, completing the base case of the outer induction.

**Inductive Step:** Again if $s$ is unsolvable then both functions are everywhere undefined, so assume that $s$ has HNF $\lambda x_1 \ldots x_n . y s_1 \ldots s_m$ and again write $\Gamma = \Delta \cdot \langle x_1, \ldots, x_n\rangle$. We assume the results (i) and (ii) for each of the terms $s_1, \ldots, s_m$, each with the context $\Gamma$, for all sequences $\vec{\alpha}$ of length up to $l$.

Let $\vec{\alpha}$ be any sequence of length $l$. Take $1 \le j \le m$. We show that (i) and (ii) hold for the term $s$ and context $\Delta$, for the sequence $j \cdot \vec{\alpha}$. We already know it holds for the sequence $\varepsilon$, and (also by the base case) that the domain of both functions is contained in the set $\{j \cdot \vec{\alpha} \mid 1 \le j \le m, \vec{\alpha} \in \mathbb{N}^*\}$. This will therefore establish the inductive step.

Suppose that $[\![s]\!]_\Delta^{\mathrm{XA}}(j\cdot\vec{\alpha}) = (i,r,t,o)$ and $\{\mathrm{VFBT}_\Delta(s)\}^k(j\cdot\vec{\alpha}) = (i',r',t',o')$. We know by result (i) of the inductive hypothesis that one is defined if and only if the other is. Lemma 4.2.1 means that $\{\mathrm{VFF}_\Delta(s)\}^k(j\cdot\vec{\alpha}) = (i',r')$. On the other hand, Theorem 4.1.1 means that $[\![s]\!]_\Delta(j\cdot\vec{\alpha}) = (i,r)$, and the Exact Correspondence Theorem for $\mathcal{D}_{\mathrm{EAC}}$ gives that $i'=i$ and $r'=r$. We next show that $t'=t$, completing the proof of (ii).

Now by the definition of VFBT,

$$
\begin{array}{rl}
& \{\mathrm{VFBT}_\Delta(s)\}^k(j\cdot\vec{\alpha}) = (i',r',t',o') \\
\text{if and only if} & \{\mathrm{VFBT}_\Delta(s)\}^{(k+n)}(s_j)(\vec{\alpha}) = (i'',r'',t',o'')
\end{array}
$$

for some irrelevant numbers $i''$, $r''$, $o''$ (this is simple to verify). Then by the inductive hypothesis $[\![s_j]\!]_\Gamma^{\mathrm{XA}}(\vec{\alpha}) = (i'',r'',t',o'')$

With this fact in hand we examine $[\![s]\!]_\Delta^{\mathrm{XA}}$, aiming to calculate $[\![s]\!]_\Delta^{\mathrm{XA}}(j\cdot\vec{\alpha})$ from this fact.

As we found in the base case, $[\![s]\!]^{\mathrm{XA}}_{\Delta} = [\![ys_1 \dots s_m]\!]^{\mathrm{XA}}_{\Gamma} = (\Pi^{\Gamma}_{y} \bullet [\![s_1]\!]^{\mathrm{XA}}_{\Gamma}) \bullet \cdots \bullet [\![s_m]\!]^{\mathrm{XA}}_{\Gamma}$, which with the $\bullet$'s decoded is

$$\langle \cdots \langle \langle \Pi^{\Gamma}_{y}; f, [\![s_1]\!]^{\mathrm{XA}}_{\Gamma} \rangle; \mathrm{eval}; f, [\![s_2]\!]^{\mathrm{XA}}_{\Gamma} \rangle; \mathrm{eval}; f \cdots, [\![s_m]\!]^{\mathrm{XA}}_{\Gamma} \rangle; \mathrm{eval}$$

where eval is $\mathrm{eval}_{U_0,U_0}$. What follows is not a rigorous analysis, as such a thing would be impossible to typeset as well as extremely tedious.

We already know, from proof of the Exact Correspondence Theorem for $\mathcal{D}_{\mathrm{EAC}}$, that if the first O-move made against this strategy is $j$ then the result of this multiple composition is to copy moves made by and against $\sigma_j$ from the components where they are hidden into ones where they are not; this composite strategy makes moves which are (a small translation of) those of $\sigma_j$. What is important is that between visible moves, except between the initial move and the first P-move, all of the intermediate moves are *not* roots of the arenas they occur in (the reader is invited to draw a picture — on a large piece of paper — and demonstrate this for themselves). The reason this is important is because for $\Pi^{\Gamma}_{y}$ and $\mathrm{eval}_{U_0,U_0}$ the copycat thresholds and offsets are always zero except at the initial P-view (this is very simple to check).

Thus when we work out the threshold and offset of $[\![s]\!]^{\mathrm{XA}}_{\Delta}$ at the P-view coded by $j \cdot \vec{\alpha}$ using the composition algorithm and the fact that the threshold and offset of $[\![s_j]\!]^{\mathrm{XA}}_{\Gamma}$ at the P-view coded by $\vec{\alpha}$ are $t'$ and $o''$, the calculation will be either of the form

$$\begin{array}{llcll}
t'_1 = 0 & o'_1 = 0 & \quad\text{or}\quad & t'_1 = t' & o'_1 = o'' \\
t'_2 = 0 & o'_2 = 0 & & t'_2 = 0 & o'_2 = 0 \\
 & \vdots & & & \vdots \\
t'_p = 0 & o'_p = 0 & & t'_p = 0 & o'_p = 0 \\
t'_{p+1} = t' & o'_{p+1} = o'' & & t'_{p+1} = 0 & o'_{p+1} = 0
\end{array}$$

depending on which component the visible moves appear in. In the first case $T_1 = T_2 = \cdots = T_p = 0$ and $\mathbf{t} = T_{p+1} = \max(t', o'') = t'$, In the second case, $T_1 = t'$ so $t' = T_2 = \cdots = T_{p+1} = \mathbf{t}$.

Thus in either case we have shown that the copycat threshold of $[\![s]\!]^{\mathrm{XA}}_{\Delta}$ at the P-view coded by $j \cdot \vec{\alpha}$ is $t'$, but by assumption it is also $t$. Hence $t = t'$.

Finally, we show (i) as follows: $j \cdot \vec{\alpha} \cdot i \in \mathrm{dom}([\![s]\!]^{\mathrm{XA}}_{\Delta})$ if and only if $\vec{\alpha} \cdot i \in \mathrm{dom}([\![s_j]\!]^{\mathrm{XA}}_{\Gamma})$, this is because of the way the multiple composition which defines $[\![s]\!]^{\mathrm{XA}}_{\Delta}$ copies the moves of $s_j$ after the first P-move $j$. But $\vec{\alpha} \cdot i \in \mathrm{dom}([\![s_j]\!]^{\mathrm{XA}}_{\Gamma})$ if and only if $\vec{\alpha} \cdot i \in \mathrm{dom}(\mathrm{VFBT}_{\Gamma}(s_j))$ (by (i) of the inductive hypothesis) if and only if $j \cdot \vec{\alpha} \cdot i \in \mathrm{dom}(\mathrm{VFBT}_{\Delta}(s))$ by the definition of VFBT.

This completes the inductive step of the outermost induction. ■

**Corollary 4.2.5** For closed terms $s$ and $t$,

$$[\![s]\!]^{\mathrm{XA}} \subseteq [\![t]\!]^{\mathrm{XA}} \iff \mathrm{BT}(s) \subseteq \mathrm{BT}(t).$$

The order on $\mathcal{D}_{\mathrm{XA}}$ is inclusion of EXAC strategies, namely inclusion of both EAC strategy part and threshold function. The order on Böhm trees is inclusion of labelling function, modulo renaming of bound variables, which amounts to inclusion of variable-free form. Thus the local structure of $\mathcal{D}_{\mathrm{XA}}$ is the $\lambda$-theory $\mathcal{B}$, which equates terms with the same Böhm tree.

**Example** Applying the Exact Correspondence Theorem to the variable-free forms of the Böhm trees of the terms $I$ and 1, which we looked at earlier in the section, we can deduce that the economical forms of $[\![I]\!]$ and $[\![1]\!]$ are, as we hoped, the EXAC strategies $\eta_0$ and $\eta_1$ described in Section 3.1.

As with the model $\mathcal{D}_{\mathrm{EAC}}$, the Exact Correspondence Theorem allows us to prove the powerful result of universality holds for $\mathcal{D}_{\mathrm{XA}}$.

**Theorem 4.2.6** $\mathcal{D}_{\mathrm{XA}}$ is a universal $\lambda$-algebra.

**Proof** Suppose that $\sigma' = \langle \sigma, t_\sigma \rangle$ is an EXAC strategy on $U$ (to be an element of the model it must be a member of the homset $\mathrm{Hom}_{\mathbb{XA}}(\mathbf{1}, U_0)$). Let $f$ be the economical form of the innocent function of $\sigma$, and write $t_v$ and $o_v$ for the threshold and offset of $\sigma$ at the P-view coded by $\vec{v}$.

Let the set $X \subseteq \mathbb{N}^*$ be defined inductively by:

$$\begin{array}{l} \varepsilon \in X \\ \vec{v} \in X \implies \forall i.\ 1 \leq i \leq t_v \implies \vec{v} \cdot i \in X. \end{array}$$

This has the property that the domain of the economical form of $\sigma'$ is a subset of $X$ (this is straightforward to verify).

We define the labelling function of Böhm-like tree $A$ as follows. The domain of $A$ is the set $X$. For any sequence $\vec{v} = \langle v_1, \ldots, v_p \rangle \in X$ we define $A(\vec{v})$ by:

(i) If $\vec{v} \notin \mathrm{dom}(f)$ then $A(\vec{v})$ is undefined (the partially-labelled tree has label $\perp$ at this node).

(ii) If $f(\vec{v}) = (i, r)$ then $A(\vec{v}) = \lambda x_1^{\vec{v}} \ldots x_n^{\vec{v}}.x_i^{\langle v_1, \ldots, v_{p-r} \rangle}$, where $n = t_v - o_v$.

Now there are no free variables in the Böhm-like tree $A$ which is also clearly r.e. We make use of the characterisation of those Böhm-like trees which are the Böhm trees of terms, which can for instance be found in [Bar84, 10.1.23]. This states that a Böhm-like tree $A$ is the Böhm tree of some term if and only if $A$ has only finitely many free variables and the labels of $A$ are given recursively. Hence there is some term $s$ with $\mathrm{BT}(s) = A$.

Now we prove that $\mathrm{VFBT}_\varepsilon(s)$, as a labelling function, is the same as the economical form of $\sigma'$. Take any sequence $\vec{v} \in X$. If $f$ is not defined on this sequence, neither is the economical form of $\sigma'$, but on the other hand $A$ is not labelled at $\vec{v}$ by (i) above. If $f(\vec{v}) = (i, r)$ then we know that the node in $X$ coded by $\vec{v}$ has $t_v$ children, hence similarly for $A$, and also that $A(\vec{v}) = \lambda x_1^{\vec{v}} \ldots x_n^{\vec{v}}.x_i^{\langle v_1, \ldots, v_{p-r} \rangle}$, where $n = t_v - o_v$, by (ii) above. Thus $\mathrm{VFBT}_\varepsilon(s)(\vec{v}) = (i, r, t_v, o_v)$ by Lemma 4.2.2. On the other hand, by definition the economical form of $\sigma'$ also maps $\vec{v}$ to $(i, r, t_v, o_v)$.

Finally, we appeal to the Exact Correspondence Theorem for $\mathcal{D}_{\mathrm{XA}}$, which shows that $[\![s]\!]_\varepsilon$ has economical form given by the labelling function of $\mathrm{VFBT}_\varepsilon(s)$, which by the above is the economical form of $\sigma'$. ∎

### 4.3 Non-extensionality of $\mathcal{D}_{\mathrm{XA}}$

Recall that a $\lambda$-algebra $\mathcal{A}$ (more generally, any applicative structure) is called *extensional* if for all $s, t \in \mathcal{A}$,

$$(\forall a \in \mathcal{A}.\ s \bullet a = t \bullet a) \implies s = t.$$

In [KNO01] we showed that $\mathcal{D}_{\mathrm{EAC}}$ was extensional (in fact it was order-extensional, a stronger property). However, since $[\![I]\!]^{\mathrm{XA}} \neq [\![1]\!]^{\mathrm{XA}}$, but for all terms $s$ and $t$, $Ist = 1st$, we can be sure that $\mathcal{D}_{\mathrm{XA}}$ is not extensional.

In fact we will show that $\mathcal{D}_{\mathrm{XA}}$ is not even *weakly extensional*. A $\lambda$-algebra $\mathcal{A}$ (more generally, any combinatory algebra) is weakly extensional if the first-order statement

$$\mathcal{A} \models (\forall x.s = t) \implies \lambda x.s = \lambda x.t$$

is true. A weakly extensional $\lambda$-algebra is called a *$\lambda$-model*.

Scott showed that if we present a $\lambda$-algebra as $\mathcal{M}(\mathbb{C}, R, \mathit{Fun}, \mathit{Gr})$, as described in Section 2.1, then this is weakly extensional if and only if the object $R$ *has enough points* in the category $\mathbb{C}$, i.e. $\forall\, s, t : R \to R.\ (\forall\, r : \mathbf{1} \to R.r\ ; s = r\ ; t) \implies s = t$. The allows a short proof of the following.

**Lemma 4.3.1** $\mathcal{D}_{\mathrm{XA}}$ is not weakly extensional.

**Proof** Recall that $\mathcal{D}_{\mathrm{XA}}$ is the model $\mathcal{M}(\mathbb{XA}, U_0, f, g)$. Define $s = \mathrm{id}_{U_0}$ and $t = f; g$. Both are morphisms $U_0 \to U_0$ and $s \neq t$, as demonstrated in Section 4.1.

Now take any $r : \mathbf{1} \to U_0$, then certainly $r; s = r$. By the comments in the proof of Lemma 3.4.1, we must have $r = \bot$ or $\mathbf{l}(r) \geq 1$. But then $r; t = r$, trivially in the first case and by Lemma 4.2.3 (twice) in the second.

So $U_0$ does not have enough points in $\mathbb{XA}$, hence $\mathcal{D}_{\mathrm{XA}}$ is not weakly extensional. ■

## 5 Further Work

Although we succeeded in our aim to find a game model of $\mathcal{B}$, there are other questions which the work prompts. For example, it is usual in game semantics to make an *extensional collapse* of strategies, i.e. identify those with equivalent applicative behaviour, and this is usually required to make an extensional model. With $\mathcal{D}_{\mathrm{EAC}}$ we avoided the necessity of doing so, by characterising the definable elements of $\mathcal{D}_{\mathrm{REC}}$, and because the $\lambda$-theory it models is itself extensional. A natural direction is therefore to formulate a notion of extensional equivalence and study the extensional collapse of $\mathcal{D}$ and $\mathcal{D}_{\mathrm{REC}}$. We believe that this does *not* lead to the model $\mathcal{D}_{\mathrm{EAC}}$ as there are innocent strategies which do not have the same behaviour as any EAC strategy. A conjecture is that we in fact arrive at a model of (a particular sort of) infinitary untyped $\lambda$-calculus.

Another direction is to examine the Böhm tree composition algorithm which the game model gives: it is quite different from the usual composition of Böhm trees, which turns the finite approximants (of the trees to be composed) into terms, composes them as terms, and then uses these as finite approximants to the answer. The method given by the model is "demand-driven", in that for any node of the answer only the relevant nodes of the composed trees are examined.

Finally, we may try to find game models of other (perhaps non-sensible) $\lambda$-theories. If we hope for an Exact Correspondence result we will need the $\lambda$-theory to have some sort of normal form for terms, and this may be a sticking point. A new idea is needed.

# A Proof of Theorem 3.2.2

Firstly, we may assume that $C$ is a single tree arena, because if $C = \langle C_1, \ldots, C_n \rangle$ then we can write $\tau = \langle \tau_1, \ldots, \tau_n \rangle$ and set $\sigma;\tau = \langle \sigma;\tau_1, \ldots, \sigma;\tau_n \rangle$ and since $\mathbf{l}(\tau) = \min_{i=1}^{n}\{\mathbf{l}(\tau_i)\}$ we can be sure that the condition $\mathbf{l}(\tau_i) \geq |B|$ holds for each $i$ where $\tau_i$ has a first move.

We choose a P-view $\vec{v}$ of the composite strategy $\sigma;\tau$, overloading notation so that $\vec{v}$ refers to both the P-view and the sequence of natural numbers which encodes it; we do likewise for moves.

However we must be very careful when referring to "the $i^{th}$" child of a move $m$. This may well be ambiguous — in the special case where $m$ is the root of $C$, the $i^{th}$ child of the move $m$ in the arena $A \Rightarrow C$ is the $(i-1)^{th}$ child of the same move in the arena $C$. In all other cases there is no difference which arena the move $m$ is looked at in. In what follows $(A\,@\,m)^{>n}$ will unambiguously refer to selecting the subtree rooted at $m$ and deleting the first $n$ subtrees of that, relative to the arena $A$.

Using the notation of the Algorithm, we aim to show:

- (i) The arenas $((A \Rightarrow C)\,@\,\underline{m})^{>(\mathbf{t}-\mathbf{o})}$ and $((A \Rightarrow C)\,@\,m)^{>\mathbf{t}}$ are order-isomorphic;
- (ii) If $i > \mathbf{t}$ then $f(\vec{v} \cdot i) = (i - \mathbf{o}, 1)$ and further $f(\vec{v} \cdot i \cdot \vec{w} \cdot j) = (j, 1)$ for all sequences $\vec{w}$ and numbers $j \geq 1$;
- (iii) For all $\vec{w} \geq (\vec{v} \cdot k)$ with $k \leq \mathbf{t}$, if $f(\vec{w}) = (i, |\vec{w}| - |\vec{v}|)$ then $i \leq \mathbf{t} - \mathbf{o}$;
- (iv) If $f(\vec{v}) = (i, 0)$ then $i \leq \mathbf{t} - \mathbf{o}$.

This is what we need, for the strategy $\sigma\,;\tau$ to be AC at the P-view $\vec{v}$. We note that condition (ii) is sufficient for the composition to be EC where necessary — condition (iii) in the definition of everywhere copycat ("If $f(\vec{v}) = (i, 0)\ldots$") holds vacuously.

We assume that neither $\sigma$ nor $\tau$ is everywhere undefined; these special cases will be dealt with at the end.

**Proof of (i) and (ii):**

Note that $\sigma$'s thresholds and offsets are relative to the arena $A \Rightarrow B$, whereas $\tau$'s are relative to $B \Rightarrow C$. We imagine, therefore, that all these moves take place in the arena $(A \Rightarrow B) \Rightarrow C$; then consideration of the values of $t'_i$ and $o'_i$ shows that they are the same offsets, but always relative to this new arena, which we will henceforth shorten to $ABC$.

We show by induction that all $T_i$ and $O_i$ satisfy:

- (a) $(ABC\,@\,\underline{m})^{>(T_i - O_i)} \cong (ABC\,@\,m_i)^{>T_i}$;
- (b) If $j > T_i$ and the $j^{th}$ child (relative to the arena $ABC$) of the last move in the interaction $\vec{u_i}$ (i.e. $m_i j$) is played the resulting continuation has as the next visible move the P-move $\underline{m}(j - O_i)$ (relative to the arena $ABC$), justified by $\underline{m}$;

(c) Furthermore a subsequent move $\underline{m}(j - O_i)w_1$ will result in an interaction ending in $m_i j w_1$, and in general for any odd-length sequence $\vec{w}$ the move $\underline{m}(j - O_i)\vec{w}$ will result in an interaction ending in $m_i j \vec{w}$ and for even-length sequences $\vec{w}$ the move $m_i j \vec{w}$ will result in an interaction ending in $\underline{m}(j - O_i)\vec{w}$.

For $i = p + 1$ the moves $m_i = m$ and $m_i j$ and their descendants are visible in the arena $C$.

For $i = 1$ the properties we require are just the AC property of $\sigma$ or $\tau$ at $\vec{u_0}$.

Suppose that we have proved the result for $T_i$ and $O_i$.

**Either** $T_{i+1} = T_i + o'_{i+1}$, so we must have $T_i + o'_{i+1} \geq t'_{i+1}$, say $T_i = t'_{i+1} - o'_{i+1} + l$. Then

$$\begin{array}{rrcl} & (ABC \,@\, \underline{m})^{>(T_i - O_i)} & \cong & (ABC \,@\, m_i)^{>T_i} \\ \text{and by the AC property of } \sigma \text{ or } \tau, & (ABC \,@\, m_i)^{>(t'_{i+1} - o'_{i+1})} & \cong & (ABC \,@\, m_{i+1})^{>t'_{i+1}} \\ \text{hence} & (ABC \,@\, m_i)^{>(t'_{i+1} - o'_{i+1} + l)} & \cong & (ABC \,@\, m_{i+1})^{>(t'_{i+1} + l)}. \end{array}$$

But $T_{i+1} - O_{i+1} = T_i - O_i$ and $T_{i+1} = t'_i + l$ hence

$$(ABC \,@\, \underline{m})^{>(T_{i+1} - O_{i+1})} \cong (ABC \,@\, m_{i+1})^{>T_{i+1}}.$$

**Or** $T_{i+1} = t'_{i+1}$, so we must have $T_i + o_{i+1} \leq t'_{i+1}$, say $T_i = t'_{i+1} - o'_{i+1} - l$. Then

$$\begin{array}{rrcl} & (ABC \,@\, \underline{m})^{>(T_i - O_i)} & \cong & (ABC \,@\, m_i)^{>T_i} \\ \text{hence} & (ABC \,@\, \underline{m})^{>(T_i - O_i + l)} & \cong & (ABC \,@\, m_i)^{>(T_i + l)} \\ \text{and by the AC property of } \sigma \text{ or } \tau, & (ABC \,@\, m_i)^{>(t'_{i+1} - o'_{i+1})} & \cong & (ABC \,@\, m_{i+1})^{>t'_{i+1}}. \end{array}$$

But $T_{i+1} - O_{i+1} = t'_{i+1} - o'_{i+1} - O_i = T_i - O_i + l$ hence

$$(ABC \,@\, \underline{m})^{>(T_{i+1} - O_{i+1})} \cong (ABC \,@\, m_{i+1})^{>T_{i+1}}.$$

Now for the proof of (b) and (c) suppose that the strategy dictating the first move of the interaction is $\tau$, i.e. $\underline{m}$ is in the arena $C$ (the other case is similar and will require some $\tau$'s to be replaced by $\sigma$'s and one other difference which we note below). Suppose that the move $m_i j$ is made after the move $m_i$, with $j > T_i$. Then the interaction sequence looks like:

$$\underbrace{\overset{\bullet}{\varepsilon}\ \cdots\ \overset{\bullet}{\underline{m}} \xrightarrow{\tau} \overset{\star}{m_1} \xrightarrow{\sigma} \overset{\star}{m_2} \xrightarrow{\tau} \overset{\star}{m_3}\ \cdots\ \overset{\star}{m_{i-2}} \xrightarrow{\sigma} \overset{\star}{m_{i-1}} \xrightarrow{\tau} \overset{\star}{m_i}}_{\vec{u_i}} \longrightarrow \overset{\star}{m_i j}$$

(Here and hereafter we are omitting the justification pointers, because where they should point to are determined by the property of AC.) We can predict how this interaction would continue: since $j > T_i \geq t'_i$ we know by the EAC property of $\tau$ that we must continue:

$$\underbrace{\cdots\ \overset{\star}{m_{i-1}} \xrightarrow{\tau} \overset{\star}{m_i}}_{\vec{u_i}} \longrightarrow \overset{\star}{m_i j} \xrightarrow{\tau} \overset{\star}{m_{i-1}(j - o'_i)}$$

Here $m_i j$ refers to the $j^{th}$ child of $m$ relative to the arena $ABC$, and $m_{i-1}(j - o'_i)$ is the $(j - o'_i)^{th}$ child relative to the arena $ABC$.

But $j - o'_i > T_i - o'_i \geq T_{i-1}$ so by the induction hypothesis we know that this interaction must continue until it reaches the visible P-move $\underline{m}(j - o'_i - O_{i-1}) = \underline{m}(j - O_i)$:

$$\cdots \underbrace{\overset{\star}{m_{i-1}} \xrightarrow{\tau} \overset{\star}{m_i}}_{\vec{u_i}} \to \overset{\star}{m_i j} \xrightarrow{\tau} \overset{\star}{m_{i-1}(j - o'_i)} \overset{\text{I.H.}}{\cdots\cdots\to} \overset{\circ}{\underline{m}(j - O_i)}$$

We have to be a little careful, because the move $\underline{m}(j - O_i)$ has to be visible, i.e. it is in the arena $C$. If the move $\underline{m}$ is a root of $C$ then this will only be true as long as $j - O_i > |B|$. Thankfully in this case we can use that $\mathbf{l}(\tau) \geq |B|$ so that $j - O_i > T_i - O_i \geq t'_1 + \sum_{j=2}^{i-1} o'_j - \sum_{j=1}^{i-1} o'_j = t'_1 - o'_1 = t_1 - o_1 \geq |B|$.

Of course, there will be no such problem in the case that $\sigma$ was the strategy to dictate the move immediately after $\underline{m}$, because $\underline{m}$ would have to be a $A$-move. However, in this case we have to show that the move $m_{i-1}(j - o'_i)$ is not visible, i.e. it is in the arena $B$. There is no problem unless $m_{i-1}$ is a root of $B$. Then similarly to above, we can use the condition that $\mathbf{l}(\sigma) \geq |A|$ giving $j - o'_i > T_i - o'_i \geq t'_i - o'_i = t_i - o_i$. Now since the move $m_{i-1}$ is a root of $B$ this value is the l-number of $\sigma$, hence the required result.

If there is a further P-move $\underline{m}(j - O_i)w_1$, by the inductive hypothesis, the following must result:

$$\cdots \underbrace{\overset{\star}{m_{i-1}} \xrightarrow{\tau} \overset{\star}{m_i}}_{\vec{u_i}} \to \overset{\star}{m_i j} \xrightarrow{\tau} \overset{\star}{m_{i-1}(j - o'_i)} \overset{\text{I.H.}}{\cdots\cdots\to} \overset{\circ}{\underline{m}(j - O_i)} \to \overset{\bullet}{\underline{m}(j - O_1)w_1} \overset{\text{I.H.}}{\cdots\cdots\to} \overset{\star}{m_{i-1}(j - o'_i)w_1}$$

and because $\tau$ must be EC at this view, we have:

$$\cdots \underbrace{\overset{\star}{m_i}}_{\vec{u_i}} \to \overset{\star}{m_i j} \xrightarrow{\tau} \overset{\star}{m_{i-1}(j - o'_i)} \overset{\text{I.H.}}{\cdots\cdots\to} \overset{\circ}{\underline{m}(j - O_i)} \to \overset{\bullet}{\underline{m}(j - O_1)w_1} \overset{\text{I.H.}}{\cdots\cdots\to} \overset{\star}{m_{i-1}(j - o'_i)w_1} \xrightarrow{\tau} \overset{\star}{m_i j w_1}$$

And it is clear how subsequent moves will be copied by $\tau$, and the induction hypothesis will always apply, giving the desired result.

Finally we note that in order to consider the children of the root of $C$ to be labelled with respect to $A \Rightarrow C$ rather than $ABC$ we must set $\mathbf{o} = O_{p+1}$, unless $\underline{m}$ is a root of $C$ in which case $\mathbf{o} = O_{p+1} - |A| + |B|$. We can set $\mathbf{t} = T_{p+1}$ in any case, because the threshold refers to children of $m$, and $m$ could not be a root of $C$.

Property (a) proves (i); (b) and (c) together give (ii).

**Proof of (iii):**

We require a little result for use later:

Directly from the definition we have that $T_{i+1} \geq T_i + o'_{i+1}$ so $T_{p+1} \geq T_n + \sum_{i=n+1}^{p+1} o'_i$. Also we have $O_{p+1} = \sum_{i=1}^{p+1} o'_i$ and $T_n \geq t'_n$. Thus

$$T_{p+1} - O_{p+1} \geq T_n - \sum_{i=1}^{n} o'_i \geq t'_n - \sum_{i=1}^{n} o'_i.$$

Now suppose that we have a sequence $\vec{w}$ such that $f_{\sigma;\tau}(\vec{v} \cdot k \cdot \vec{w}) = (i, |\vec{w}| + 1)$ with $k \leq \mathbf{t}$ (here $f_{\sigma;\tau}$ economical form of the innocent function of the composition). We aim to show that $i \leq \mathbf{t} - \mathbf{o}$. We would prefer to talk in terms of children of moves in the arena $ABC$, in which case we say that the resulting move is $\underline{m}i'$. Here $i' = i$ unless $\underline{m}$ is a root of $C$, in which case $i' = i + |B| - |A|$. Then we want to show that $i' \leq T_{p+1} - O_{p+1}$.

All we know about the interaction which produced this move is that it must be of the form:

$$\underbrace{\cdots\ \overset{\bullet}{\underline{m}}\ \ \overset{\star}{m_1}\ \cdots\ \overset{\star}{m_p}}_{\mathbf{u}(\vec{v},\sigma,\tau)}\ \ \overset{\circ}{m}\ \ \overset{\bullet}{mk}\ \ \overbrace{\cdots \qquad\qquad \cdots \qquad\qquad \cdots}^{\text{some } \star,\ \circ \text{ and } \bullet \text{ moves}}\ \ \star\ \ \overset{\circ}{\underline{mi'}}$$

Let us write $\vec{u} = \mathbf{u}(\vec{v} \cdot k \cdot \vec{w}, \sigma, \tau)$. Since the move $\underline{mi'}$ is justified by $\underline{m}$ it must be in the arena $A$ and it was $\sigma$ which caused it. So we examine the view $\sigma$ saw when making that move:

$$\vec{u_1} = \ulcorner \vec{u} \upharpoonright X \urcorner^X$$

where as before $X$ is the appropriate component. As the move $\underline{m}$ must be in this view (because of the visibility condition) so there cannot be justification pointers from O-moves jumping over it. Hence $\vec{u_1} = \ulcorner \vec{u_1}_{\leq \underline{m}} \urcorner \cdot \vec{x} = \ulcorner \mathbf{u}(\vec{v}, \sigma, \tau)_{\leq \underline{m}} \upharpoonright X \urcorner \cdot \vec{x}$, for some sequence $\vec{x}$, and $f_(\vec{u_1}) = (i, |x|)$, where $f$ is $f_\sigma$ or $f_\tau$ as appropriate. Again, in terms of the arena $ABC$, this resulting move is child number $i'$ of the move justifying it.

Now we look at the first element of $\vec{x}$, call it $k_1$.

**Either** $k_1 \leq t'_1$. But then by the EAC property of $\sigma$ or $\tau$ we must have $i' \leq t'_1 - o'_1 \leq T_{p+1} - O_{p+1}$ by the result shown above. So the result holds in this case.

**Or** $k_1 > t'_1$. But then the EAC property of $\sigma$ or $\tau$ forces that $\vec{x} = \langle k_1 \rangle$ with $i' = k_1 - o'_1$. So the uncovering must look like:

$$\cdots\ \overset{\bullet}{\underline{m}} \overset{\sigma}{\dashrightarrow} \overset{\star}{m_1}\ \cdots\ \star \overset{\tau}{\dashrightarrow} \overset{\star}{m_1 k_1} \overset{\sigma}{\dashrightarrow} \overset{\circ}{\underline{mi'}}$$

Because to whichever strategy was last to act the move of the interaction $m_1$ is a P-move, we can be sure it is in view when the move $\underline{mi'}$ is made.

So we know that the move $m_1 k_1$ was made by either $\sigma$ or $\tau$. Again we look at the view it had:

$$\vec{u_2} = \ulcorner \vec{u}_{\leq m_1 k_1} \upharpoonright X \urcorner^X$$

As before we can deduce that $\vec{u_2} = \ulcorner \mathbf{u}(\vec{v}, \sigma, \tau)_{\leq m_1} \upharpoonright X \urcorner \cdot \vec{x}$ for some sequence $\vec{x}$ and again we examine the first move, call it $k_2$. Either $k_2 \leq t'_2$, whence as before we can deduce $k_1 \leq t'_2 - o'_2$ hence $i' \leq t'_2 - o'_2 - o'_1 \leq T_{p+1} - O_{p+1}$, or $k_2 > t'_2$ so that $\vec{x} = \langle k_2 \rangle$ and $k_1 = k_2 - o'_2$.

We can use the argument inductively: at each stage the interaction looks like:

$$\begin{array}{cccccccccccccccc} & \bullet & \overset{\sigma}{\dashrightarrow} & \star & \overset{\tau}{\dashrightarrow} & \star & & \star & & \star & & \star & \overset{\tau}{\dashrightarrow} & \star & \overset{\sigma}{\dashrightarrow} & \circ \\ \cdots & \underline{m} & & m_1 & & m_2 & \cdots & m_n & \cdots & m_n k_n & \cdots & m_2 k_2 & & m_1 k_1 & & \underline{m} i' \end{array}$$

and because of the EAC property of $\sigma$ or $\tau$ at $\ulcorner \mathbf{u}(\vec{v}, \sigma, \tau)_{\leq m_n} \upharpoonright X \urcorner$ either $k_n \leq t'_n$, whence $i' \leq t'_n - \sum_{i=1}^{n} o'_i \leq T_{p+1} - O_{p+1}$, or $k_{n-1} = k_n - o'_n$ and we can apply the argument again.

At the last stage, when $n = p + 1$, we actually know that the sequence $\vec{x}$ is singleton, because the move $m_{p+1} = m$ is answered directly by $mk$, by hypothesis. So $i' = k - \sum_{i=1}^{p+1} o'_i$ (by chasing back the definitions of $k_n$) and $k \leq T_{p+1}$ by hypothesis, hence $i' \leq T_{p+1} - O_{p+1}$. All cases have now been covered.

**Proof of (iv):**

The move $m$ must be $\underline{m}i$ and be justified by $\underline{m}$. We need to show that the EAC properties of $\sigma$ and $\tau$ force $i \leq \mathbf{t} - \mathbf{o}$. Again, relative to the arena $ABC$ we can say that the move $m$ is the $i'^{th}$ child of $\underline{m}$, where $i'$ is as above. Again, we wish to show that $i' \leq T_{p+1} - O_{p+1}$.

The proof is similar to that of (iii). We consider the sequence which each strategy had in view, when making the move $\underline{m}i'$, $\vec{u_1} = \ulcorner \mathbf{u}(\vec{v}, \sigma, \tau) \upharpoonright X \urcorner^X$. Once again the move $\underline{m}$ must be in view, so $\vec{u_1} = \ulcorner \mathbf{u}(\vec{v}, \sigma, \tau)_{\leq \underline{m}} \urcorner \cdot \vec{x}$. We look at the first element of $\vec{x}$, $k_1$ say, and as before either $k_1 \leq t'_1$ — exactly as before this leads to $i' \leq T_{p+1} - O_{p+1}$ — or we can deduce that the move before $m$ is $m_1 k_1$ and apply the argument inductively. Note that, unlike in (iii), the move before $m$ is actually $m_p$.

After $n$ steps we will be have either $k_n \leq t'_n$, hence $i' \leq T_{p+1} - O_{p+1}$, or $m_{p-n+1} = m_n k_n$. So this time we reach the final stage after $p/2$ steps (we know $p$ must be even because $\underline{m}$ and $m$ are in the same component) and here have no choice but that $m_{p/2+1} = m_{p/2} k_{p/2}$, hence $k_{p/2} \leq t'_{p/2} - o'_{p/2}$ and $i' \leq T_{p+1} - O_{p+1}$ follows directly from the result shown at the start of the proof of (iii).

**Special Cases:**

If $\tau$ is everywhere undefined and then so is the composition. If $\sigma$ is everywhere undefined then for the composition to be defined on the P-view $\vec{v}$ the moves $\underline{m}$ and $m$ must be in $C$, with no intermediate moves in between. Now we can treat this in the same way as the usual cases.

## References


[AJM94] S. Abramsky, R. Jagadeesan, and P. Malacaria. Full abstraction for PCF (extended abstract). In *Theoretical Aspects of Computer Software: TACS'94, Sendai, Japan*, pages 1–15. Springer-Verlag, 1994. LNCS Vol. 789.

[AM95] S. Abramsky and G. A. McCusker. Games and full abstraction for the lazy $\lambda$-calculus. In *Proceedings, Tenth Annual IEEE Symposium on Logic in Computer Science*, pages 234–243. IEEE Computer Society Press, 1995.

[Bar84] H. P. Barendregt. *The Lambda Calculus, Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, 1984.

[GFH98] P. Di Gianantonio, G. Franco, and F. Honsell. Games semantics for untyped $\lambda$-calculus. preprint, 1998.

[HO00] J. M. E. Hyland and C.-H. L. Ong. On full abstraction for PCF: I, II and III. To appear in *Information and Computation*, 2000.

[Ker01] A. D. Ker. *Innocent Game Models of Untyped Lambda Calculus*. PhD thesis, University of Oxford, 2001. In preparation.

[KNO99] A. D. Ker, H. Nickau, and C.-H. L. Ong. A universal innocent game model of the Böhm tree lambda theory. In *Computer Science Logic: Proceedings of the 8th Annual Conference of the EACSL, Madrid, Spain, September 1999*, number 1683 in Lecture Notes in Computer Science, pages 405–419. Springer-Verlag, 1999.

[KNO01] A. D. Ker, H. Nickau, and C.-H. L. Ong. Innocent game models of untyped $\lambda$-calculus. To appear in *Theoretical Computer Science*, 2001.

[McC98] G. A. McCusker. *Games and Full Abstraction for a Functional Metalanguage with Recursive Types*. Distinguished Dissertations in Computer Science. Springer-Verlag, 1998.

[Nak75] R. Nakajima. Infinite normal forms for the $\lambda$-calculus. In C. Böhm, editor, *Proceedings of the Symposium on $\lambda$-calculus and Computer Science Theory*, volume 37 of *Lecture Notes in Computer Science*, pages 62–82. Springer-Verlag, Berlin, 1975.

[Nic96] Hanno Nickau. *Hereditarily Sequential Functionals: A Game-Theoretic Approach to Sequentiality*. Shaker-Verlag, 1996. Dissertation, Universität Gesamthochschule Siegen.