# A Universal Innocent Game Model for the Böhm Tree Lambda Theory

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

Computing Laboratory, Parks Road, Oxford OX1 3QD, UK,
{Andrew.Ker, Hanno.Nickau, Luke.Ong}@comlab.ox.ac.uk.

**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.

## 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 [4]. Although the general approach is innocent in the sense of [3] and [7], 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 an 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 Sect. 2. Section 3 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.* Sections 2 and 3 should be regarded as a survey of [4], and this paper is a sequel to that work. The notion of EAC strategies has a natural extension to *explicitly and effectively almost-everywhere copycat.* However finding an ambient cartesian closed category for these strategies to inhabit proved to be a painful process as we briefly show in Sect. 4 – 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 $\mathcal{M}$. We formulate a new version of the powerful *Exact Correspondence Theorem* of the earlier work, with which we can show that $\mathcal{M}$ is a universal $\lambda$-model which induces the intended equational theory. To our knowledge, $\mathcal{D}_{\mathrm{EAC}}$ and $\mathcal{M}$ are the first syntax-independent universal $\lambda$-models.

In [2], Di Gianantonio et al. have obtained game models of the untyped $\lambda$-calculus using history-free strategies. 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.

## 2 Arenas and Innocent Strategies

This section and the next give a quick introduction to the basic ideas underpinning the main result of the paper. We refer the reader to [4] for further details and to [3] and [5] for proofs of all results quoted. We define an *arena* to be a finite tuple of nonempty trees of *moves.* The root of each tree is called an *initial move.* Our trees are considered "upside-down" with the root at the top, rather like family trees. We can also refer to the child of a node, and say that one node inherits from another, in the same vein. 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 $\circ$.

We will only be interested in countably branching, countably deep trees. Thus we can encode each tree of the arena as a subset of $\mathbb{N}^*$[1] by inductively labelling the root as $\varepsilon$ and the $n^{th}$ child of the move $\boldsymbol{s}$ as $\boldsymbol{s} \cdot n$ (we use the notation $\boldsymbol{s}, \boldsymbol{t}$ etc. to denote sequences). Hence each move of each tree is associated uniquely with a sequence of natural numbers. Conversely, given any subset $A \subseteq \mathbb{N}^*$ which is prefix-closed and has the property that whenever $\boldsymbol{s}\cdot n \in A$ we have $\boldsymbol{s}\cdot m \in A$ for each $m \leq n$, we can form an arena of one tree where the moves are the elements of $A$, ordered by prefix. *Henceforth by arenas, we shall always mean arenas in sequence-subset (of* $\mathbb{N}^*$*) form.* 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.

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 $A \times B = \langle A_1, \ldots, A_m, B_1, \ldots, B_n \rangle$.

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

- 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 $A$. More precisely $A \Rightarrow B = \langle C_1, \ldots, C_n \rangle$ where

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

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 $\boldsymbol{s}$ is said to be *well-formed* if elements of $\boldsymbol{s}$ alternate between P-moves and O-moves and if $m \in \boldsymbol{s}$ is justified by $m^-$ then the move $m$ is directly beneath $m^-$ in the tree of the arena[2]. Henceforth all justified sequences are assumed to be well-formed.

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

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

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

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

[2] The no-dangling-question-mark condition in [3, 7] (equivalently the well-bracketing condition) is redundant for our arenas.

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 [4] 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 $\boldsymbol{s}$ and $\boldsymbol{t}$ and P-moves $m$, $\boldsymbol{s} \cdot m \in \sigma \wedge \boldsymbol{t} \in \sigma \wedge \ulcorner \boldsymbol{s} \urcorner = \ulcorner \boldsymbol{t} \urcorner \to \boldsymbol{t} \cdot m \in \sigma$ and the moves $m$ are justified by moves which are identical in the P-view $\ulcorner \boldsymbol{s} \urcorner = \ulcorner \boldsymbol{t} \urcorner$. 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 [3, §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 $\boldsymbol{s} \cdot m \cdot m' \in \mathrm{dom}(f)$ then $f(\boldsymbol{s}) = m$. 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 1.** *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 recursive arenas as objects and recursive innocent strategies as morphisms.*

**Theorem 1.** $\mathbb{A}$ *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 $\mathbb{C}$; that is, there exist morphisms $\mathit{Fun} : R \rightarrow [R \Rightarrow R]$ and $\mathit{Gr} : [R \Rightarrow R] \rightarrow R$ such that $\mathit{Gr} \,;\, \mathit{Fun} = \mathrm{id}_{[R \Rightarrow R]}$. Thus we may specify a *$\lambda$-algebra* by a 4-tuple $\langle \mathbb{C}, R, \mathit{Fun}, \mathit{Gr} \rangle$ (it is in fact a $\lambda\eta$-algebra if $\mathit{Fun} \,;\, \mathit{Gr} = \mathrm{id}_R$); the underlying set of the $\lambda$-algebra is the set $\mathbb{C}(\mathbf{1}, R)$ of global sections. If the reflexive object $R$ *has enough points* (i.e. $\forall\, f, g : R \rightarrow R.[\forall\, r : \mathbf{1} \rightarrow R.r \,;\, f = r \,;\, g] \rightarrow f = g$) then $\langle \mathbb{C}, R, \mathit{Fun}, \mathit{Gr} \rangle$ is a *$\lambda$-model* (i.e. a weakly extensional $\lambda$-algebra). We refer the reader to [1] for a comprehensive treatment of the model theory of the untyped $\lambda$-calculus.

Recall that the arena $U$ has the key property that $U = U \Rightarrow U$ so that in this case the morphisms $\mathit{Fun}$ and $\mathit{Gr}$ are both the identity on $U$. We can now define the first two of our game $\lambda$-algebras (which are both $\lambda\eta$-algebras): $\langle \mathbb{A}, U, \mathrm{id}_U, \mathrm{id}_U \rangle$ which we shall write simply as $\mathcal{D}$, and $\langle \mathbb{A}_{\mathrm{REC}}, U, \mathrm{id}_U, \mathrm{id}_U \rangle$ 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 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.

## 3 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 $\mathbb{N} \times \mathbb{N}_0$ (where $\mathbb{N}_0 = \{0, 1, 2, \cdots\}$). 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:

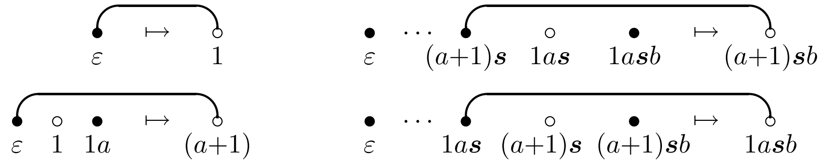
$e_\sigma : \langle v_1, \ldots, v_n \rangle \mapsto (i, p)$ if and only if

$$f_\sigma : \overset{\bullet}{\varepsilon} \quad \overset{\circ}{\boldsymbol{s}_1} \; \overset{\bullet}{\boldsymbol{s}_1 v_1} \; \overset{\circ}{\boldsymbol{s}_2} \; \overset{\bullet}{\boldsymbol{s}_2 v_2} \quad \cdots \quad \overset{\circ}{\boldsymbol{s}_{n-p}} \; \overset{\bullet}{\boldsymbol{s}_{n-p} v_{n-p}} \quad \cdots \quad \overset{\circ}{\boldsymbol{s}_n} \; \overset{\bullet}{\boldsymbol{s}_n v_n} \qquad \mapsto \qquad \overset{\circ}{\boldsymbol{s}_{n-p}(v_{n-p} i)}$$

Justification pointers in the P-view can be deduced from the behaviour of $f_\sigma$ on shorter P-views, and so have been omitted. Note that each $\boldsymbol{s}_i$ is a sequence of natural numbers.

Furthermore, we can expand any partial function $f : \mathbb{N}^* \rightharpoonup \mathbb{N} \times \mathbb{N}_0$ which has prefix-closed domain and satisfies $f(\boldsymbol{v}) = (i, p) \to 0 \leq p \leq |\boldsymbol{v}|$ 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 1.* The following is the innocent function of the "copycat" strategy $\mathrm{id}_U$:

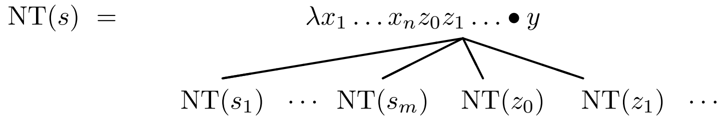


$\mapsto$
$\varepsilon$ $1$
$\cdots$ $\mapsto$
$\varepsilon$ $(a{+}1)\boldsymbol{s}$ $1a\boldsymbol{s}$ $1a\boldsymbol{s}b$ $(a{+}1)\boldsymbol{s}b$
$\mapsto$
$\varepsilon$ $1$ $1a$ $(a{+}1)$
$\cdots$ $\mapsto$
$\varepsilon$ $1a\boldsymbol{s}$ $(a{+}1)\boldsymbol{s}$ $(a{+}1)\boldsymbol{s}b$ $1a\boldsymbol{s}b$

Here $\boldsymbol{s}$ 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 $\boldsymbol{v}$, $\boldsymbol{v}i \mapsto (i, 1)$.

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 [6]. 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

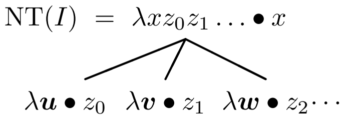


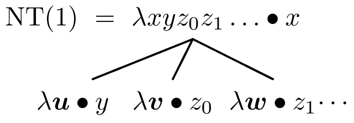
$\mathrm{NT}(s) \;=$
$\lambda x_1 \ldots x_n z_0 z_1 \ldots \bullet y$
$\mathrm{NT}(s_1)$ $\cdots$ $\mathrm{NT}(s_m)$ $\mathrm{NT}(z_0)$ $\mathrm{NT}(z_1)$ $\cdots$

where $z_0, z_1, \ldots$ are countably many fresh variables. The process of finding such fresh variables given in [6] is quite complicated. In [4] 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.

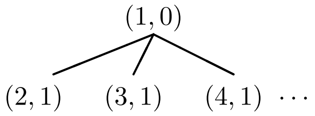
The theorem of key importance in [4] is the *Exact Correspondence Thoerem*, which states that for every closed $\lambda$-term $s$, the innocent strategy denoting $s$ (in both $\mathcal{D}$ and $\mathcal{D}_{\mathrm{REC}}$) given in economical form is exactly $\mathrm{VFF}(s)$, the Nakajima tree of $s$ in variable free form.

*Example 2.* 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:


$\mathrm{NT}(I) \;=\; \lambda x z_0 z_1 \ldots \bullet x$
$\lambda \boldsymbol{u} \bullet z_0$
$\lambda \boldsymbol{v} \bullet z_1$
$\lambda \boldsymbol{w} \bullet z_2 \cdots$


$\mathrm{NT}(1) \;=\; \lambda x y z_0 z_1 \ldots \bullet x$
$\lambda \boldsymbol{u} \bullet y$
$\lambda \boldsymbol{v} \bullet z_0$
$\lambda \boldsymbol{w} \bullet z_1 \cdots$

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:



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 single-node tree $\bot$). 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 $\boldsymbol{s} \cdot n \in A \rightarrow \boldsymbol{s} \cdot m \in A$ for all $m < n$) and for any $\boldsymbol{s} \in A$ we define

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

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

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

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 2.** *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* $\boldsymbol{v} \in \mathbb{N}^*$ *if* $f$ *is undefined at* $\boldsymbol{v}$ *or the following hold:*

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

*(ii) Whenever* $\boldsymbol{w} \geq \boldsymbol{v}$ *we have that for all* $i \in \mathbb{N}$ $f(\boldsymbol{w} \cdot i) = (i, 1)$*.*

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

*We say that* $f$ *is* almost-everywhere copycat *(AC) at* $\boldsymbol{v}$ *if* $f$ *is undefined at* $\boldsymbol{v}$ *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* $(A \mathbin{@} \mathbf{m_o}(\boldsymbol{v}))^{>(t_v - o_v)}$ *and* $(A \mathbin{@} \mathbf{m_p}(\boldsymbol{v}))^{>t_v}$ *are isomorphic.*

*(ii) For all* $i > t_v$*,* $f(\boldsymbol{v} \cdot i) = (i - o_v, 1)$ *and* $f$ *is everywhere copycat at* $\boldsymbol{v} \cdot i$*.*

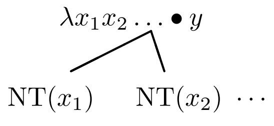
*(iii) For all* $\boldsymbol{w} \geq (\boldsymbol{v} \cdot k)$ *with* $k \leq t_v$*, if* $f(\boldsymbol{w}) = (i,\ |\boldsymbol{w}| - |\boldsymbol{v}|\,)$ *then* $i \leq t_v - o_v$*.*

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

*(Note that* $f$ *is EC at* $\boldsymbol{v}$ *if and only if* $f$ *is AC at* $\boldsymbol{v}$ *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* $\boldsymbol{v} \mapsto t_v$ *and* $\boldsymbol{v} \mapsto o_v$ *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.*

To illustrate the definition of everywhere copycat strategies, suppose $f$ is defined at $\boldsymbol{v}$. Intuitively we say that $f$ is everywhere copycat at $\boldsymbol{v}$ if, from $\mathbf{m_p}(\boldsymbol{v})$ onwards, $f$'s behaviour is simply to play copycat for as long as the arena will allow it. So if O's move is $mi$, the $i^{th}$ child of the justifying move $m$, then P responds with the $i^{th}$ child of the move immediately preceding $m$ in the P-view. Condition (i) in the definition guarantees that P's copycat move will always be available. As before we will primarily be interested in strategies on $U$. Since $U \mathbin{@} \boldsymbol{s} = U = U^{>n}$ for all sequences $\boldsymbol{s}$ and numbers $n$, Condition (i) will always hold. Condition (ii) is best understood with reference to the Exact Correspondence Theorem which relates innocent strategies to Nakajima trees. It specifies that the subtree of the Nakajima tree corresponding to $f$, rooted at $\boldsymbol{v}$, has the following shape:



Condition (iii) of the definition is a technicality, which ensures that the variable $y$ is not one of the $x_i$.

**Definition 3.** *The* category of arenas and EAC strategies*,* $\mathbb{A}_{\mathrm{EAC}}$*, has recursive arenas as objects and EAC strategies on* $A \Rightarrow B$ *as morphisms from* $A$ *to* $B$*.*

A main result in [4] is that the category $\mathbb{A}_{\mathrm{EAC}}$ is well-defined; the proof that EAC strategies compose is highly technical. In fact,

**Theorem 2.** $\mathbb{A}_{\mathrm{EAC}}$ *is cartesian closed.*

The arena $U$ is still an object of $\mathbb{A}_{\mathrm{EAC}}$ and still equal to its function space. Thus we can define a $\lambda\eta$-algebra $\langle \mathbb{A}_{\mathrm{EAC}}, U, \mathrm{id}_U, \mathrm{id}_U \rangle$ which we shall denote by $\mathcal{D}_{\mathrm{EAC}}$. Properties of $\mathcal{D}_{\mathrm{EAC}}$ will be presented later.

## 4 Effectively and Explicitly Almost-Everywhere Copycat Strategies

We wish to find a new game model which invalidates the rule of $\eta$-conversion. To do so, we would 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. 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 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.) It is really the thresholds (rather than the offsets) that are important because, for a certain P-view $\boldsymbol{v}$ of an EAC strategy $\sigma$, the copycat threshold $t$ gives enough information to compute the offset $o$ directly. This motivates the following definition:

**Definition 4.** *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 comments 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 5.** *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 usual way. The map is defined at a sequence* $\boldsymbol{v}$ *only if the P-view encoded by* $\boldsymbol{v}$ *is entirely explicit, in which case*

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

*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 3.* 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:

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

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 (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 $\boldsymbol{v}$ 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 $m$.

We write $\boldsymbol{u} = \mathbf{u}(\boldsymbol{v}, \sigma, \tau)$ for the *uncovering* of the composition up to the move $m$. A formal definition can be found in [3] or [4], but we may describe it as the sequence of moves of the composition which result after the P-view $\boldsymbol{v}$, including any relevant $B$-moves which would be hidden by the composition. It will be of the form $\langle \varepsilon, \ldots, \underline{m}, m_1, m_2, \ldots, m_{p-1}, m_p, m \rangle$.

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, and are all in the arena $B$. 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 $\boldsymbol{u_i}$ that the strategies $\sigma$ or $\tau$ are faced with when the move $m_i$ was made. (For details on how one may define such a P-view precisely see [4] or [5]). Define $t_i$ and $o_i$ to be the copycat threshold and offset of $\sigma$, or $\tau$ as appropriate, at the P-view $\boldsymbol{u_i}$. These are specified by $t_\sigma$ or $t_\tau$. Then set:

$$
\begin{array}{rll}
t'_i = & t_i + |A|\,, & \text{if } m_i \text{ is a root of the arena } B \\
& t_i, & \text{otherwise} \\
o'_i = & o_i + |A|\,, & \text{if } m_i \text{ is a root of the arena } B \\
& o_i, & \text{otherwise}
\end{array}
\qquad
\begin{array}{rl}
T_1 &= t'_1 \\
O_1 &= o'_1 \\
T_{i+1} &= \max(T_i + o'_{i+1}, t'_{i+1}) \\
O_{i+1} &= O_i + o'_{i+1}
\end{array}
$$

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

(By $|A|$ we mean the number of trees in the arena $A$). 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 $\boldsymbol{v}$.

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 6.** *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 (we call this P-view the minimal P-view). 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, and defined on at least one of the minimal P-views, then we say $\mathbf{l}\,(\sigma) = \min_{i=1,\sigma_i \neq \perp}^{n} \{\mathbf{l}\,(\sigma_i)\}$.*

This is termed the l-number of $\sigma$ because, as will eventually be shown, it corresponds to the number of $\lambda$-abstractions at the root of the Böhm tree of the term whose denotation is $\sigma$. For example $\mathbf{l}\,(\eta_0) = 1$ and $\mathbf{l}\,(\eta_1) = 2$, and we will be able to show that $\eta_0$ is the denotation of $\lambda x.x$ and $\eta_1$ the denotation of $\lambda xy.xy$.

**Theorem 3.** *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 4 produces valid copycat thresholds and offsets for $\sigma ; \tau$.*

There is an "obvious" category, which derives directly from the conditions required for the composition algorithm to work correctly.

**Definition 7.** *The* category of arenas and EXAC strategies*, written* $\mathbb{A}_{\mathrm{EXAC}}$*, has recursive arenas as objects, and 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. The identity morphism on $A$ is the EXAC strategy $\langle id_A, 0\rangle$, i.e. the copycat threshold is zero everywhere.*

One can show that this does indeed specify a category. Also, $\mathbb{A}_{\mathrm{EXAC}}$ has the obvious terminal object — the empty arena $E$ — and products given in the usual way. However, $\mathbb{A}_{\mathrm{EXAC}}$ does not form a CCC with the usual constructions, as the following example shows:

*Example 4.* 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. 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 $\Lambda(\eta_1)$.

It is not clear that $\mathbb{A}_{\mathrm{EXAC}}$ forms a CCC with any unusual function space constructions either.

If we try to fix the definition of $\mathbb{A}_{\mathrm{EXAC}}$, by cutting down the homsets some more, it becomes clear that one must also specify minimum copycat thresholds at the minimal P-views, along with minimum l-numbers. The obvious solution still does not work, and we can repeat the fixing-up process to obtain a sequence of failures — each is either not a category at all because identities fail to work properly, or has a non-uniqueness of curried morphisms as above.

We now present a new category based on EXAC strategies, which does form a CCC. Although it does appear to be much more complicated than the "almost-CCC" $\mathbb{A}_{\mathrm{EXAC}}$, it seems to be the natural limit of the fixing-up process.

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 $\mathrm{br}(A \,@\, m)$, 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 8.** *Let $A$ be an arena and $X$ a finitely-branching subarena*[3] *of $A$. We say that an EXAC strategy $\sigma$ over $A$ is $X$-*explicit *if the following holds:*

*Let $\sigma : \boldsymbol{v} \mapsto (i, r, t, o)$ be the economical form of any clause of the innocent function. Suppose that the sequence $\boldsymbol{v}$ 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 $t - o \geq br(X \,@\, \underline{m})$,*

*(ii) if $m$ is in the subarena $X$ then $t \geq br(X \,@\, m)$.*

[3] We say that the arena $A = \langle A_1, \cdots, A_m\rangle$ is a *subarena* of $B = \langle B_1, \cdots, B_n\rangle$ 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

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. moves in $X$ are 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.

**Definition 9.** *The category $\mathbb{XA}_{\mathrm{EXAC}}$, or simply $\mathbb{XA}$, is given by the following: objects are pairs $(A, X)$ consisting of a recursive arena $A$ and a finitely-branching recursive subarena $X$; a morphism $\sigma : (A, X) \to (B, Y)$ is an EXAC strategy on $A \Rightarrow B$ which is $(X \Rightarrow Y)$-explicit. Composition of morphisms is composition of EXAC strategies, and the identity strategy on $(A, X)$, $id_{(A,X)}$, is the EXAC strategy $\langle id_A, t\rangle$, where $id_A$ 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 $\langle id_A, t\rangle$ as $(X \Rightarrow X)$-explicit.*

The fact that the composition algorithm gives valid thresholds and offsets for EXAC strategies satisfying these conditions comes from Theorem 3 — a morphism $\sigma : (A, X) \to (B, Y)$ is $(X \Rightarrow Y)$-explicit, and $X$ has the same number of trees as $A$, so in particular $\mathbf{l}(\sigma) \geq |A|$.

**Theorem 4.** *$\mathbb{XA}$ forms a CCC with the following constructions: the terminal object $\mathbf{1}$ is $(E, E)$, where $E$ is the empty arena; the product $(A, X) \times (B, Y)$ is $(A \times B, X \times Y)$, with the threshold functions for projections specified in the same style as identities; the exponential object $(A, X) \Rightarrow (B, Y)$ is $(A \Rightarrow B, X \Rightarrow Y)$, and the evaluation map $eval_{(B,Y),(C,Z)}$ is the same EXAC strategy as $id_{(B,Y)\Rightarrow(C,Z)}$.*

Now that we have found an ambient CCC for the EXAC strategies, we can construct another $\lambda$-algebra based on it. In this category, however, the reflexive object is not isomorphic to its function space — exactly as we would hope for a model invalidating $\eta$-conversion.

Let us write $U^0$ for the object $\langle U, M\rangle$ of $\mathbb{XA}$, and $U^1$ for the object $U^0 \Rightarrow U^0$. Here $U$ and $M$ are the maximal and minimal single-tree arenas described in Sect. 2. We define morphisms $F : U^0 \to U^1$ and $G : U^1 \to U^0$ to both be given by the EXAC strategy $\eta_1$ (the definition of which can be found in Sect. 4). It is straightforward to check that this does give proper morphisms, and that they satisfy $G; F = \mathrm{id}_{U^1}$ and $F; G \neq \mathrm{id}_{U^0}$.

Hence we can identify a new $\lambda$-algebra $\langle \mathbb{XA}, U^0, F, G\rangle$ which invalidates the rule of $\eta$-conversion; we denote this $\lambda$-algebra $\mathcal{M}$. By erasing all threshold information, we can reduce $\mathcal{M}$ to (a subset of) $\mathcal{D}$ and deduce that $\mathcal{M}$ is also sensible.

In the same way that the denotation of a term in the model $\mathcal{D}_{\mathrm{EAC}}$ had a strong connection with its Nakajima tree, the denotation in $\mathcal{M}$ corresponds closely to (a variable-free version of) the Böhm tree. The variable-free form of the Böhm tree of a term is similar to the construction VFF mentioned earlier and defined in [4], but it includes extra information describing how many abstractions there are at each node, and how many children.

**Definition 10.** *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* $(n-i+1, d, t, o)$*;*

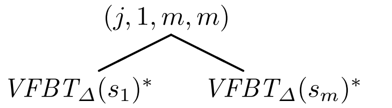
*(iii) those labelled* $(i, d+1, t, o)$ *for* $i > n$ *are relabelled* $(i-n, d+1, t, o)$.

*For a term* $s$ *with free variables within* $\Delta$ *the* variable-free form *of the Böhm tree of* $s$, $VFBT_\Delta(s)$, *is the following* $(\mathbb{N} \times \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{Z})$*-labelled tree:*

$$VFBT_\Delta(s) = \perp, \qquad \text{the empty tree, for unsolvable } s.$$

$$VFBT_\Delta(\lambda x_1 \ldots x_n \bullet s) = \{\, VFBT_{\Delta \cdot \langle x_1, \ldots, x_n \rangle}(s) \,\}^n, \qquad \text{if } s \text{ is of the form } v_j s_1 \ldots s_m.$$

$$VFBT_\Delta(v_j s_1 \ldots s_m) =$$



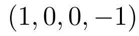
*where* $\Delta = \langle v_k, \ldots, v_1 \rangle$ *(note the reverse order).*

Exactly as before we can show that, at each node of the Böhm tree of $s$, the first two elements of the tuple at the corresponding node of VFBT($s$) encode the head variable by counting how many levels one goes up the tree, and how many abstractions along, to find where the variable is abstracted. The third component just counts the number of children at the node, and the fourth is the number of children minus the number of abstractions. We choose to encode the number of abstractions in this rather elliptic fashion in order to make the following theorem easier to state:

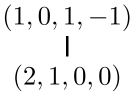
**Theorem 5 (Exact Correspondence for $\mathcal{M}$).** *If* $s \in \Lambda$ *with free variables in* $\Delta = \langle v_k, \ldots, v_1 \rangle$ *then* $[\![ s ]\!]^{\mathcal{M}}_{\Delta} = \{\, VFBT_\Delta(s) \,\}^k$ *when the former is considered as an EXAC strategy in economical form and the latter as a labelling function.*

*In particular for closed terms* $s$, $[\![ s ]\!]_\varepsilon = VFBT_\varepsilon(s)$

*Example 5.* Although it is hard to see directly, the given definition of VFBT does work as intended. One may check that VFBT($I$) and VFBT(1) are:



and



The node $\lambda xy.x$, in the Böhm tree of 1, corresponds to the node of 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 Exact Correspondence Theorem gives us the economical forms of $[\![ I ]\!]$ and $[\![ 1 ]\!]$ which, as we hoped, are the EXAC strategies $\eta_0$ and $\eta_1$ described earlier in this section.

The Exact Correspondence results are of key importance in examining the local structure of the game models. In [4] we use them, and a powerful result which we call the Separation Lemma, to obtain proofs that $\mathcal{D}_{\mathrm{EAC}}$ is a *universal* and *order-extensional* $\lambda$-model, with equational theory given by $\mathcal{H}^*$ (the maximal consistent sensible theory). The models $\mathcal{D}$ and $\mathcal{D}_{\mathrm{REC}}$ are neither universal nor extensional, but do generate the same equational theory on terms.

In a similar way, we can use the Exact Correspondence Theorem for $\mathcal{M}$ to prove the following:

**Theorem 6.** *(i) $\mathcal{M}$ is universal i.e. every element is definable as the denotation of some term of the $\lambda$-calculus. (ii) $\mathcal{M}$ is weakly extensional, so it is a $\lambda$-model (for a discussion of weak extensionality see [1, §5]. (iii) $\mathcal{M}$ equates two terms of the $\lambda$-calculus precisely when they have the same Böhm tree. Thus the local structure of the model is the Böhm tree theory $\mathcal{B}$.*

## 5 Further Work

Although we succeeded in our aim to find a universal game model of $\mathcal{B}$, there are other questions which the work prompts. Firstly, one might ask if there is a less syntactic way to arrive at $\mathcal{D}_{\mathrm{EAC}}$ from $\mathcal{D}_{\mathrm{REC}}$, perhaps by some sort of *extensional collapse*. In fact extensional collapse appears to be insufficient, and further investigation would be of interest. In another direction, we can use $\mathcal{D}$, $\mathcal{D}_{\mathrm{REC}}$, $\mathcal{D}_{\mathrm{EAC}}$ and $\mathcal{M}$ as very natural combinatory algebras which are in some sense *sequential*. Therefore one might wish to study realizability models over them. Finally, and more practically, we could examine the Böhm tree composition algorithm given by the game model: it is quite different from the usual method in that it is "demand-driven" – for every node of the answer only the relevant nodes of the composed trees are examined.

## References

[1] Barendregt, H.: *The Lambda Calculus, Its Syntax and Semantics.* North-Holland, revised edition, 1984.

[2] Di Gianantonio, P., Franco, G., Honsell, F.: Games Semantics for Untyped $\lambda$-Calculus, Preprint, announced on the Types Mailing List. April 1998.

[3] Hyland, J. M. E, Ong, C.-H. L.: On Full Abstraction for PCF: I, II and III. To appear in *Information and Computation*, 133 pages, ftp-able. 1994.

[4] Ker, A., Nickau, H., Ong, C.-H. L.: Game Models of Untyped $\lambda$-Calculus. Submitted for publication, 49 pages, ftp-able. 1998.

[5] McCusker, G.: *Games and Full Abstraction for a Functional Metalanguage with Recursive Types.* Cambridge University Press, 1998.

[6] Nakajima, R.: Infinite Normal Forms for the $\lambda$-Calculus. In Goos, G. and Hartmanis, J., editors, *Symposium on $\lambda$-Calculus and Computer Science*, *LNCS* 37, pages 62–82. Springer-Verlag, 1975.

[7] Nickau, H.: *Hereditarily Sequential Functionals: A Game-Theoretic Approach to Sequentiality.* Dissertation, Universität GH Siegen. Shaker-Verlag, 1996.