Games Characterizing Levy-Longo Trees

C.-H. L. Ong* P. Di Gianantonio**

Abstract. We present a simple *strongly universal* innocent game model for Levy-Longo trees i.e. every point in the model is the denotation of a unique Levy-Longo tree. The observational quotient of the model then gives a universal, and hence fully abstract, model of the pure Lazy Lambda Calculus.

1 Introduction

This paper presents a *strongly universal* innocent game model for Levy-Longo trees [11, 12] (i.e. every point in the model is the denotation of a unique Levy-Longo tree). We consider arenas in the sense of [8, 14] in which questions may justify either questions or answers, but answers may only justify questions; and we say that an answer (respectively question) is *pending* in a justified sequence if no question (respectively answer) is explicitly justified by it. Plays are justified sequences that satisfy the standard conditions of Visibility and Well-Bracketing, and a new condition, which is a *dual* of Well-Bracketing, called

> *Persistence*: If an odd-length (respectively even-length) play s has a pending O-answer (respectively P-answer) – let a be the last such in s, and if s is followed by a question q, then q must be explicitly justified by a.

We then consider *conditionally copycat* strategies, which are *innocent* strategies (in the sense of [8]) that behave in a *copycat* fashion as soon as an O-answer is followed by a P-answer. Together with a *relevance* condition, we prove that the recursive such strategies give a *strongly universal* model of Levy-Longo trees i.e. every strategy is the denotation of a unique Levy-Longo tree. To our knowledge, this is the first universal model of Levy-Longo trees. The observational quotient of the model then gives a universal and fully abstract model of the pure Lazy Lambda Calculus [15, 3].

Related work. Universal models for the Lazy Lambda Calculus with convergence test were first presented in [2] and [13]. The model studied in the former is in the AJM style [1], while that in the latter, by McCusker, is based on an innocent-strategy [8] universal model for call-by-name FPC, and is obtained via a universal and fully abstract translation from the Lazy Lambda Calculus into

* Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom. Fax: +44 1865 273839. `lo@comlab.ox.ac.uk`

** Dipartimento di Matematica e Informatica, Universitá de Udine, via delle Scienze, 206, 33100 Udine, Italy. Fax +39 0432558499. `digianantonio@dimi.uniud.it`

call-by-name FPC. The present paper considers the *pure* (i.e. without any constant) Lazy Lambda Calculus. Our model is the same as McCusker's, except that it has three additional constraints: Persistence, which is a constraint on plays, and Conditional Copycat and Relevance, which are constraints on strategies. Since Persistence constrains Opponent as well as Player, the model presented here is not simply a submodel of McCusker's.

An AJM-style game model of the the pure Lazy Lambda Calculus was presented by the second author in [6]. The strategies therein are history-free and satisfy a *monotonicity* condition. Though fully abstract for the language, the model is not universal (there are finite monotone strategies that are not denotable). However we believe it is possible to achieve universality by introducing a condition similar to relevance. In [10, 9] game models based on *effectively almost-everywhere copycat* (or EAC) strategies are constructed which are strongly universal for Nakajima trees and Böhm trees respectively. Several local structure results for AJM-style game models can be found in [7].

2 Arenas and nested levels

An ***arena*** is a triple $A = \langle M_A, \lambda_A, \vdash_A \rangle$ where M_A is a set of moves; $\lambda_A : M_A \longrightarrow \{ \text{PQ}, \text{PA}, \text{OQ}, \text{OA} \}$ is a labelling function which, for given a move, indicates which of P or O may make the move and whether it is a question (Q) or an answer (A); and $\vdash_A \subseteq (A + \{ * \}) \times A$, where $*$ is a dummy move, is called the *justification relation* (we read $m_1 \vdash_A m_2$ as "m_1 justifies m_2") satisfying the following axioms: we let m, m', m_i range over M_A

1. Either $* \vdash_A m$ (in which case we call m an *initial move*) or else $m^- \vdash_A m$ for some m^-.
2. Every initial move is an O-question.
3. If $m \vdash_A m'$ then m and m' are moves by different players.
4. If $m_1 \vdash_A m$ and $m_2 \vdash_A m$ then $m_1 = m_2$.
5. If $m \vdash_A m'$ and m is an answer then m' is a question ("Answers may only justify questions.").

In the following we shall refer to standard arena constructions such as product $A \times B$, function space $A \Rightarrow B$ and lifting $A_\perp$; the reader may wish to consult e.g. [8, 14] for a definition. We use square and round parentheses in bold type as meta-variables for moves as follows:

O-question	P-answer	P-question	O-answer
[	**]**	**(**	**)**

A ***justified sequence*** over an arena A is a finite sequence of alternating moves such that, except the first move which is initial, every move m has a *justification pointer* (or simply *pointer*) to some earlier move m^- whereby $m^- \vdash_A m$; we say that m is *explicitly justified* by m^-. A question (respectively answer) in a justified sequence s is said to be ***pending*** just in case no answer (respectively question) in s is explicitly justified by it. This extends the standard meaning of "pending questions" to "pending answers". Recall the definition of the ***P-view*** $\ulcorner s \urcorner$ of a justified sequence s:

$$
\begin{array}{rll}
\ulcorner \epsilon \urcorner = \epsilon & & \\
\ulcorner s\, m \urcorner = \ulcorner s \urcorner\, m & & \text{if } m \text{ is a P-move} \\
\ulcorner s\, m \urcorner = m & & \text{if } m \text{ is initial} \\
\ulcorner s\, m_0\, u\, m \urcorner = \ulcorner s \urcorner\, m_0\, m & & \text{if the O-move } m \text{ is explicitly justified by } m_0
\end{array}
$$

In $\ulcorner s\, m_0\, u\, m \urcorner$ the pointer from m to m_0 is retained, similarly for the pointer from m in $\ulcorner s\, m \urcorner$ in case m is a P-move.

Definition 1. A justified sequence s over A is said to be a ***legal position*** (or ***play***) just in case it satisfies:

1. *Visibility*: Every P-move (respective non-initial O-move) is explicitly justified by some move that appears in the P-view (respectively O-view) at that point.
2. *Well-Bracketing*: Every P-answer (respectively O-answer) is an answer to (i.e. explicitly justified by) the last pending O-question (respectively P-question).
3. ***Persistence***: If an odd-length (respectively even-length) s has a pending O-answer (respectively P-answer) – let a be the last such in s, and if s is followed by a question q, then q must be explicitly justified by a.

Remark 1. Except for Persistence, all that we have introduced so far are standard notions of the *innocent* approach to Game Semantics in the sense of [8]. Note that there can be at most one pending O-answer (respectively P-answer) in a P-view (respectively O-view). It is an immediate consequence of Well-Bracketing that no question may be answered more than once in a legal position.

Persistence may be regarded as a *dual* of Well-Bracketing: it is to questions what Well-Bracketing is to answers. The effect of Persistence is that in certain situations, namely when there is a pending O-answer, a strategy has no choice over which question it can ask, or equivalently over which argument it can interrogate, *at that point* (of course it may decide instead to answer an O-question). An apparently similar restriction on the behaviour of strategies is imposed by the *rigidity* condition introduced by Danos and Harmer [5]. (For any legal position of a rigid strategy, the pointer from a question is to some move that appears in the *R-view* of the play at that point.) However since Persistence is a constraint on *plays* containing answers that justify questions, whereas rigidity is a condition on *strategies* over arenas whose answers do *not* justify any move, it is not immediately obvious how the two notions are related.

Nested levels. Take any set M that is equipped with a function $\lambda : M \longrightarrow \{\, \mathrm{Q}, \mathrm{A} \,\}$ which labels elements as either questions or answers. Let s be a finite sequence of elements from M – call s a *dialogue*. Set $\#_{\mathrm{qn}}(s)$ and $\#_{\mathrm{ans}}(s)$ respectively to be the number of questions and the number of answers in s. Following [6], we define the ***nested level*** at $s\, m$ (or simply the ***level*** of m whenever s is understood) to be

$$\mathsf{NL}(s\,m) = \begin{cases} \delta - 1 & \text{if } m \text{ is a question} \\ \delta & \text{if } m \text{ is an answer} \end{cases}$$

where $\delta = \#_{\text{qn}}(s\,m) - \#_{\text{ans}}(s\,m)$; we define $\mathsf{NL}(\epsilon) = 0$. For example, the nested levels of the moves in the dialogue **[()([][()])()][(** are:

Nested Level																
3								**(**	**)**							
2					**[**	**]**	**[**			**]**						
1		**(**	**)**	**(**							**)**	**(**	**)**			**(**
0	**[**													**]**	**[**	

For $l \geq 0$, we write $s \upharpoonright l$ to mean the subsequence of s consisting of moves at level l. We say that an answer a in a dialogue t is *closed* if it is the last move in t at level l, where l is the level of a in t.

We state some basic properties of nested levels of dialogues.

Lemma 1. *In the following, we let s range over dialogues.*

1. *For any $s = u\,m\,m'$, if m and m' are at different levels l and l' respectively, then m and m' are either both questions (in which case $l' = l + 1$) or both answers (in which case $l' = l - 1$). As a corollary we have:*
2. *If a and b in a dialogue are at levels l_1 and l_2 respectively, then for any $l_1 \leq l \leq l_2$, there is some move between a and b (inclusive) at level l.*
3. *For any $l \geq 0$, if $l < \mathsf{NL}(s)$ (respectively $l > \mathsf{NL}(s)$) then the last move in s at level l, if it exists, is a question (respectively answer).*
4. *Suppose s begins with a question. For each l, if $s \upharpoonright l$ is non-empty, the first element is a question, thereafter the elements alternate strictly between answers and questions.*
5. *Take any dialogue sq, where q is a question. Suppose $\mathsf{NL}(sq) = l$ then an answer a is the last occurring closed answer in s if and only if a is the last move at level l in s.*

The notion of nested level is useful for proving that the composition of strategies is well-defined. Note that Lemma 1 holds for dialogues in general – there is no assumption of justification relation or pointers, nor of the distinction between P and O.

3 Conditionally copycat strategies and relevance

Recall that a *P-strategy* (or simply *strategy*) σ for a game A is defined to be a non-empty, prefix-closed set of legal positions of A satisfying:

1. For any even-length $s \in \sigma$, if sm is a legal position then $sm \in \sigma$.
2. (*Determinacy*). For any odd-length s, if sm and sm' are in σ then $m = m'$.

A strategy is said to be ***innocent*** [8] if whenever even-length $s\,m \in \sigma$ then for any odd-length $s' \in \sigma$ such that $\ulcorner s \urcorner = \ulcorner s' \urcorner$, we have $s'm \in \sigma$. That is to say, σ is completely determined by a partial function f (say) that maps P-views p to *justified P-moves* (i.e. $f(p)$ is a P-move together with a pointer to some move in p). We write f_σ for the minimal such function that defines σ. We say that an innocent strategy σ is ***compact*** just in case f_σ is a finite function (or equivalently σ contains only finitely many P-views).

Definition 2. We say that an innocent strategy σ is ***conditionally copycat*** (or simply ***CC***) if for any odd-length P-view $p \in \sigma$ in which there is an O-answer which is immediately followed by a P-answer (i.e. p has the shape "$\cdots \boldsymbol{)}\,\boldsymbol{]} \cdots$"), then $p\,m \in \sigma$ for some P-move m which is explicitly justified by the penultimate O-move in p.

CC strategies can be characterized as follows.

Lemma 2 (CC). *An innocent strategy σ is CC if and only if for every even-length P-view p in σ that has the shape $u\,\boldsymbol{)}_0\,\boldsymbol{]}_0\,v$*

1. *for any O-move m, if $pm \in \sigma$ then $pmm' \in \sigma$ for some P-move m', and*
2. *if v is a non-empty segment, then v is a **copycat block** of moves. I.e. v has the shape*

$$a_1\, b_1\, a_2\, b_2\, \cdots\, a_n\, b_n$$

where $n \geq 1$ such that

   (a) for each i, the P-move b_i is a question iff the preceding O-move a_i is a question

   (b) $\boldsymbol{]}_0$ explicitly justifies a_1 uniquely, $\boldsymbol{)}_0$ explicitly justifies b_1 uniquely, and for each $i \geq 1$, b_i explicitly justifies a_{i+1} uniquely, and each a_i explicitly justifies b_{i+1} uniquely.

   In other words v is an interleaving of two sequences v_1 and v_2, such that in each v_i, each element (except the first) is explicitly justified by the preceding element in the other sequence.

Composition of strategies. Suppose σ and τ are strategies over arenas $A \Rightarrow B$ and $B \Rightarrow C$ respectively. The set of ***interaction sequences*** arising from σ and τ is defined as follows:

$$\mathbf{ISeq}(\sigma, \tau) \;=\; \{\, u \in \mathcal{L}(A, B, C) : u \restriction (A, B, b) \in \sigma, u \restriction (B, C) \in \tau \,\}$$

where $\mathcal{L}(A, B, C)$ is the set of local sequences (see [8, 14]) over (A, B, C), and where b ranges over occurrences of initial B-moves in u, $u \restriction (A, B, b)$ is the subsequence of u consisting of moves from the arenas A and B that are hereditarily justified by the occurrence b (note that the subsequence inherits the pointers associated with the moves), and similarly for $u \restriction (B, C)$. We can now define the composite strategy $\sigma\,;\tau$ over $A \Rightarrow C$ as $\sigma\,;\tau = \{\, u \restriction (A, C) : u \in \mathbf{ISeq}(\sigma, \tau) \,\}$. In $u \restriction (A, C)$ the pointer of every initial A-move is to the unique initial C-move.

The nested level of an interaction sequence is well-defined, since an interaction sequence is a dialogue. It is useful to establish a basic property about nested levels of interaction sequences.

Lemma 3. *For any m_1 and m_2 in $u \in \mathbf{ISeq}(\sigma, \tau)$ and for any $l \geq 0$, if the segment $m_1 m_2$ appears in $u \upharpoonright l$, then m_1 explicitly justifies m_2 in u.*

Remark 2. The proof of the Lemma appeals to the assumption that $u \upharpoonright (B, C)$ and $u \upharpoonright (A, B, b)$ satisfy Persistence, and to the structure of interaction sequences (in particular, Locality and the Switching Convention). If legal positions are not required to satisfy Persistence, then the Lemma does not hold.

A notion of relevance. We consider a notion of relevance whereby P is not allowed to respond to an O-question by engaging O indefinitely in a dialogue at one level higher, nor is P allowed to "give up"; instead he must eventually answer the O-question.

Definition 3. We say that a CC strategy σ is ***relevant*** if whenever $f_\sigma : p\boldsymbol{[} \mapsto \boldsymbol{(}_0$, then there is some $b \geq 0$, and there are moves $\boldsymbol{)}_0, \boldsymbol{(}_1, \boldsymbol{)}_1, \cdots, \boldsymbol{(}_b, \boldsymbol{)}_b$, and $\boldsymbol{]}$ such that

$$f_\sigma : p\boldsymbol{[}\,\boldsymbol{(}_0\,\boldsymbol{)}_0 \cdots \boldsymbol{(}_b\,\boldsymbol{)}_b \mapsto \boldsymbol{]}.$$

We call b the *branching factor* of σ at the P-view $p\boldsymbol{[}$. (The reason behind the name is explained in the proof of Lemma 6.)

Theorem 1. *If σ and τ are relevant CC strategies over arenas $A \Rightarrow B$ and $B \Rightarrow C$ respectively then the composite $\sigma\,;\tau$ is also a relevant CC strategy.*

The category $\mathbb{L}$. We define a category called $\mathbb{L}$ whose objects are arenas and whose maps $A \longrightarrow B$ are relevant CC strategies of the arena $A \Rightarrow B$. It is completely straightforward to verify that $\mathbb{L}$ is cartesian closed (see e.g. [8] for a very similar proof): the terminal object is the empty arena; for any arenas A and B, their cartesian product is given by the standard product construction $A \times B$, and the function space arena is $A \Rightarrow B$. However lifting $(-)_\perp$ is *not* functorial. We write $\mathbb{L}_{\mathrm{rec}}$ for the subcategory whose objects are arenas but whose maps are *recursive* (in the sense of [8, §5.6]), relevant, CC strategies.

Remark 3. (i) There is no way lifting can be functorial in a category of conditionally copycat strategies. Take a CC strategy $\sigma : A \longrightarrow B$. Since $\mathrm{id}_\perp = \mathrm{id} : A_\perp \longrightarrow B_\perp$, $\sigma_\perp$ is forced to respond to the initial move q_B in $B_\perp$ with the initial move q_A in $A_\perp$, and to respond to the P-view $q_B q_A a_A$ with the move a_B. Now almost all P-views in $\sigma_\perp$ contain an O-answer a_A immediately followed by a P-answer a_B, and so, by Lemma 2, $\sigma_\perp$ is almost always constrained to play copycat, whereas σ may not be restricted in the same way. (It is easy to construct concrete instances of σ and $\sigma_\perp$.)

(ii) Functoriality of lifting is not necessary for the construction of our game models of the Lazy Lambda Calculus. The domain equation $D = [D \Rightarrow D]_\perp$ is solved in an auxiliary category of games whose maps are the subgame relations (see e.g. [2]), and lifting *is* functorial in this category. All we need are two (relevant, CC) strategies, $\mathsf{up}_D : D \longrightarrow D_\perp$ and $\mathsf{dn}_D : D_\perp \longrightarrow D$, such that $\mathsf{dn}_D \circ \mathsf{up}_D = \mathrm{id}_D$, which are easily constructible for any arena D.

(iii) Indeed functoriality of lifting is *inconsistent* with our model being fully abstract. A feature of our model is that there are "few" denotable strategies that are compact-innocent; indeed the innocent strategy denoted by a closed term is compact if and only if the term is unsolvable of a finite order. Now we know from [3, Lemma 9.2.8] that projections on the finite approximations $\mathcal{D}_n$ of the fully abstract model $\mathcal{D}$ of the Lazy Lambda Calculus are not λ-definable. If *all* the domain constructions involved in the domain equation $D = [D \Rightarrow D]_\perp$ were functorial, these projections would be maps that are definable *categorically*, which would imply that our model is not fully abstract.

4 Universality and full abstraction

The model. We denote the initial solution of the recursive domain equation $D = [D \Rightarrow D]_\perp$ in the category $\mathbb{L}$ as the arena $\mathcal{D}$. The arena $\mathcal{D}$ satisfies the properties:

1. Every question justifies a unique answer, and at most one question.
2. Every answer justifies a unique question.

With respect to the justification relation, $\mathcal{D}$ has the structure of a finitely-branching tree in which every node has either one or two descendants; see Figure 1 for a picture of $\mathcal{D}$. Note that $[\mathcal{D} \Rightarrow \mathcal{D}]_\perp$ and $\mathcal{D}$ are *identical* (not just

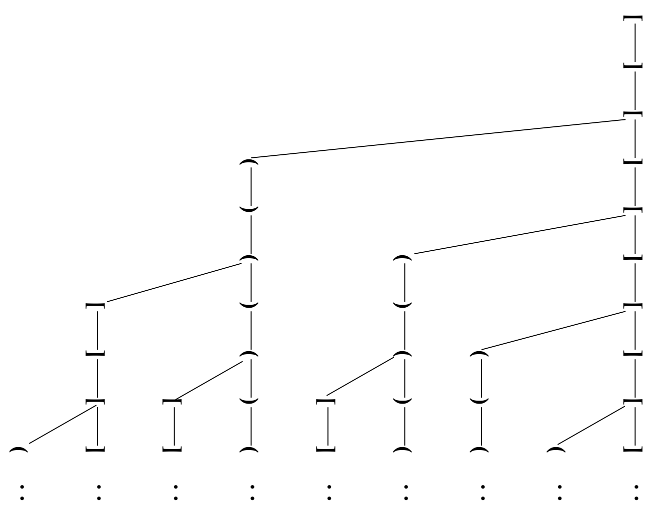


Fig. 1. A picture of $\mathcal{D}$

isomorphic) arenas. For any closed λ-term s, we shall write $[\![s]\!]$ for its denotation in the model given by $\mathcal{D}$ in $\mathbb{L}$ (so that $[\![s]\!]$ is a relevant, CC strategy over $\mathcal{D}$). By adapting a standard method in [4] based on an approximation theorem, we have the following result:

Lemma 4 (Adequacy). *For any closed term s, we have $[\![s]\!] = \bot$, the strategy that has no response to the opening move, if and only if s is strongly unsolvable (i.e. s is not β-convertible to a λ-abstraction).*

Structure of P-views. We aim to describe P-views of $\mathcal{D}$ in terms of blocks (of moves) of two kinds, called α and β respectively.

For $n \geq 0$, an α_n***-block*** is an alternating sequence of O-questions and P-answers of length $2n+1$, beginning with an O-question, such that each element except the first is explicitly justified by the preceding element, as follows:

$$\mathbf{[}_0\, \mathbf{]}\, \mathbf{[}_1\, \mathbf{]} \cdots \mathbf{[}_{n-1}\, \mathbf{]}\, \mathbf{[}_n$$

We call $\mathbf{[}_i$ the i-th question of the block.

For $m \geq 0, i \geq 0$ and $j \geq 1$, a $\beta_m^{(i,j)}$***-block*** is an alternating sequence of P-questions and O-answers of length $2m+1$, beginning with a P-question, such that each element except the first is explicitly justified by the preceding element, as follows:

$$\mathbf{(}_0\, \mathbf{)}\, \mathbf{(}_1\, \mathbf{)} \cdots \mathbf{(}_{m-1}\, \mathbf{)}\, \mathbf{(}_m$$

We call $\mathbf{(}_i$ the i-th question of the block. The superscript (i,j) in $\beta_m^{(i,j)}$ encodes the target of the justification pointer of $\mathbf{(}_0$ relative to the P-view of which the $\beta_m^{(i,j)}$-block is a part (about which more anon). A $\overline{\beta}_m^{(i,j)}$-block is just a $\beta_m^{(i,j)}$-block followed by a $\mathbf{)}$, which is explicitly justified by the last question $\mathbf{(}_m$. An α-block is just an α_n-block, for some n; similarly for a β-block.

Suppose we have a P-view of the form

$$p \quad = \quad A_1\ B_1\ A_2\ B_2\ \cdots\ A_k\ B_k\ \cdots$$

where each A_k is an α_{n_k}-block and each B_k is a $\beta_{l_k}^{(i_k,j_k)}$-block. The superscript (i_k, j_k) encodes the fact that the 0-th question of the block B_k is explicitly justified by the j_k-th question of the block A_{k-i_k}. Thus we have the following constraints: for each $k \geq 1$

$$0 \leq i_k < k \qquad \wedge \qquad 1 \leq j_k \leq n_{k-i_k} \tag{1}$$

The lower bound of j_k is 1 rather than 0 because, by definition of $\mathcal{D}$ (see Figure 1), the only move that the 0-th question of any α-block can justify is an answer. Note that since p is a P-view by assumption, for each $k \geq 2$, the 0-th question of the α-block A_k is explicitly justified by the last question of the preceding β-block.

Remark 4. It is straightforward to see that given any finite alternating sequence γ of α- and β-blocks

$$\gamma \;=\; \alpha_{n_1}\,\beta_{l_1}^{(i_1,j_1)}\;\cdots\;\alpha_{n_k}\,\beta_{l_k}^{(i_k,j_k)}\;\cdots$$

subject to the constraints (1), there is exactly one P-view p of $\mathcal{D}$ that has the shape γ. Therefore there is no harm in referring to the P-view p simply as γ, and we shall often do so in the following.

Lemma 5 (P-view Characterization). *Suppose the even-length P-view*

$$W \quad = \quad \alpha_{n_1}\,\beta_{l_1}\;\cdots\;\alpha_{n_m}\,\beta_{l_m}$$

is in a relevant CC strategy σ over $\mathcal{D}$ for some $m \geq 0$. Then exactly one of the following holds:

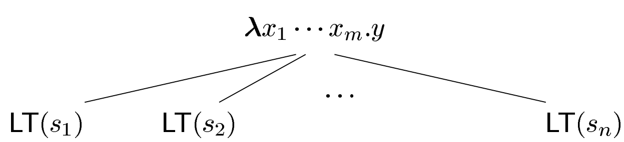
(1) For each $j \geq 0$, $W\,\alpha_j \in \mathrm{dom}(f_\sigma)$.
(2) There is some $n \geq 0$ such that $W\,\alpha_n \in \sigma \setminus \mathrm{dom}(f_\sigma)$.
(3) There are some $n_{m+1} \geq 0$, some $0 \leq i < m+1$ and some $1 \leq j \leq n_{m+1-i}$ such that $f_\sigma : W\,\alpha_{n_{m+1}} \;\mapsto\; \boldsymbol{(}^{(i,j)}$; further by relevance, for some $l \geq 0$, we have

$$f_\sigma \;:\; W\,\alpha_{n_{m+1}}\,\overline{\beta}_l^{(i,j)} \;\;\mapsto\; \boldsymbol{]}.$$

Moreover by CC we have $W\,\alpha_{n_{m+1}}\,\overline{\beta}_l^{(i,j)}\,\boldsymbol{]}\,C \in \mathrm{dom}(f_\sigma)$, for each (odd-length) copycat block C, as defined in Lemma 2.

For any λ-term s, if the set $\{\, i \geq 0 : \exists t.\boldsymbol{\lambda\beta} \vdash s = \boldsymbol{\lambda} x_1 \cdots x_i.t \,\}$ has no supremum in $\mathbb{N}$, we say that s has *order infinity*; otherwise if the supremum is n, we say that s has *order n*. A term that has order infinity is unsolvable (e.g. $\mathbf{yk}$, for any fixpoint combinator $\mathbf{y}$). We give an informal definition of $\mathsf{LT}(s)$, the ***Levy-Longo tree*** [11, 12] of a λ-term s, as follows:

- Suppose s is unsolvable: If s has order infinity then $\mathsf{LT}(s)$ is the singleton tree $\top$; if s has order $n \geq 0$ then $\mathsf{LT}(s)$ is the singleton tree $\bot_n$.
- Suppose $s =_\beta \boldsymbol{\lambda} x_1 \cdots x_m.y s_1 \cdots s_n$ where $m, n \geq 0$. Then $\mathsf{LT}(s)$ is the tree:



It is useful to fix a *variable-free representation* of Levy-Longo trees. We write $\mathbb{N} = \{\,0, 1, 2, \cdots\}$ and $\mathbb{N}_+ = \{\,1, 2, 3, \cdots\}$. A ***Levy-Longo pre-tree*** is a partial function T from the set $(\mathbb{N}_+)^*$ of *occurrences* to the following set of *labels*

$$\mathbb{N} \times (\mathbb{N} \times \mathbb{N}_+) \times \mathbb{N} \quad \cup \quad \{\,\bot_i : i \geq 0\,\} \quad \cup \quad \{\,\top\,\}$$

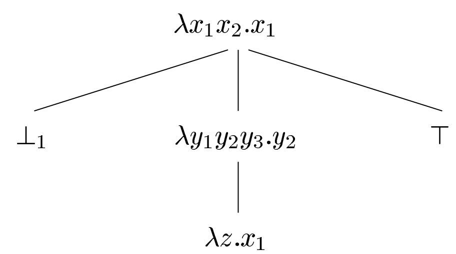
such that

1. dom(T) is prefix-closed.
2. Every occurrence that is labelled by any of $\bot_i$ and $\top$ is maximal in dom(T).
3. If $T(l_1 \cdots l_m) = \langle n, (i,j), b \rangle$ then:
   (a) $l_1 \cdots l_m l \in \mathrm{dom}(T) \iff 1 \leq l \leq b$, and
   (b) $0 \leq i \leq m + 1$, and
   (c) If $i \leq m$ then $T(l_1 \cdots l_{m-i})$ is a triple, the first component of which is at least j.

(The case of $i = m + 1$ corresponds to the head variable at $l_1 \cdots l_m$ being a free variable.) We say that the pre-tree is *closed* if $T(l_1 \cdots l_m) = \langle n, (i,j), b \rangle \implies i \leq m$. A ***Levy-Longo tree*** is the Levy-Longo pre-tree given by $\mathsf{LT}(s)$ for some λ-term s. In the following, we shall only consider closed pre-trees and trees.

To illustrate the variable-free representation, consider the following (running) example.

Example 1. Set $s = \boldsymbol{\lambda} x_1 x_2 . x_1 \, \bot_1 \, (\boldsymbol{\lambda} y_1 y_2 y_3 . y_2 \, (\boldsymbol{\lambda} z . x_1)) \, \top$. The Levy-Longo tree $\mathsf{LT}(s)$ as shown in the figure below



In variable-free form, $\mathsf{LT}(s)$ is the following partial function:

$$\begin{cases} \epsilon \mapsto \langle 2, (0,1), 3 \rangle & 2 \mapsto \langle 3, (0,2), 1 \rangle \quad 21 \mapsto \langle 1, (2,1), 0 \rangle \\ 1 \mapsto \bot_1 & 3 \mapsto \top \end{cases}$$

We Take $\mathsf{LT}(s) : 21 \mapsto \langle 1, (2,1), 0 \rangle$ which encodes the label $\boldsymbol{\lambda} z . x_1$ of the tree at occurrence 21: the first component is the *nested depth* of the λ-abstraction: in this case it is a 1-deep λ-abstraction (i.e. of order one); the second component (i,j) says that the head variable (x_1 in this case) is a copy of the j-th (in this case, first) variable bound at the occurrence i (in this case, two) levels up; and the third component is the *branching factor* at the occurrence, which is 0 in this case i.e. the occurrence 21 has 0 children.

Thanks to Lemma 5, we can now explain the correspondence between relevant CC strategies over $\mathcal{D}$ and closed Levy-Longo pre-trees; we shall write the pre-tree corresponding to the strategy σ as T_σ. Using the notation of Lemma 5, the action of the strategy σ on a P-view $p \in \sigma$ of the shape $\alpha_{n_1} \, \beta_{l_1}^{(i_1,j_1)} \cdots \alpha_{n_m} \, \beta_{l_m}^{(i_m,j_m)}$ **[** determines precisely the label of T_σ at the occurrence $l_1 \cdots l_m$. Corresponding to each of the three cases in Lemma 5, the label defined at the occurrence is as follows:

1. $\top$
2. $\bot_n$ where $n \geq 0$, and
3. $\langle n, (i, j), b \rangle$.

It is easy to see the occurrence in question is maximal in $\mathrm{dom}(T_\sigma)$ in cases 1 and 2. Suppose case 3 i.e. $T_\sigma(l_1 \cdots l_m) = \langle n, (i, j), b \rangle$. From the P-view p, we can work out the label of T_σ at each prefix $l_1 \cdots l_k$ (where $k \leq m$) of the corresponding occurrence, which is $\langle n_{k+1}, (i_{k+1}, j_{k+1}), b_{k+1} \rangle$, as determined by

$$f_\sigma : \alpha_{n_1}\, \beta_{l_1}^{(i_1, j_1)} \cdots \alpha_{n_k}\, \beta_{l_k}^{(i_k, j_k)}\, \alpha_{n_{k+1}}\, \overline{\beta}_{b_{k+1}}^{(i_{k+1}, j_{k+1})} \mapsto \mathbf{]}$$

we set $\langle n_{m+1}, (i_{m+1}, j_{m+1}), b_{m+1} \rangle = \langle n, (i, j), b \rangle$. Note that b_{k+1} is well-defined because of relevance. Thus the domain of T_σ is prefix-closed. Take any $k \leq m$. For each $1 \leq l \leq b_{k+1}$, we have the odd-length P-view

$$\alpha_{n_1}\, \beta_{l_1}^{(i_1, j_1)} \cdots \alpha_{n_k}\, \beta_{l_k}^{(i_k, j_k)}\, \alpha_{n_{k+1}}\, \beta_{l}^{(i_{k+1}, j_{k+1})}\, \mathbf{[} \quad \in \quad \sigma$$

and so, we have $l_1 \cdots l_k l \in \mathrm{dom}(T_\sigma) \iff 1 \leq l \leq b_{k+1}$. Finally, we must have $j_{k+1} \leq n_{k-i_{k+1}}$, as the pointer of the 0-th (P-)question of the β-block $\beta_l^{(i_{k+1}, j_{k+1})}$ is to the j_{k+1}-th question of the α-block $\alpha_{n_{k-i_{k+1}}}$.

To summarize, we have shown:

Lemma 6 (Correspondence). *There is a one-to-one correspondence between relevant CC strategies over $\mathcal{D}$ and closed Levy-Longo pre-trees.*

Example 2. Take the term $s = \boldsymbol{\lambda} x_1 x_2 . x_1\, \bot_1\, (\boldsymbol{\lambda} y_1 y_2 y_3 . y_2\, (\boldsymbol{\lambda} z . x_1))\, \top$ in the preceding example. In the following table, we illustrate the exact correspondence between the relevant CC strategy $[\![\, s \,]\!]$ denoted by s on the one hand, and the Levy-Longo tree $\mathsf{LT}(s)$ of the term on the other.

P-views in $[\![\, s \,]\!]$		occurrences	labels of $\mathsf{LT}(s)$
$\boxed{\alpha_2\, \overline{\beta}_3^{(0,1)}}$	$\mapsto \mathbf{]}$	ϵ	$\langle 2, (0, 1), 3 \rangle$
$\alpha_2\, \beta_{\mathbf{1}}^{(0,1)}\, \alpha_1$	$\in \sigma \setminus \mathrm{dom}(f_\sigma)$	1	$\bot_1$
$\alpha_2\, \beta_{\mathbf{2}}^{(0,1)}\, \boxed{\alpha_3\, \overline{\beta}_1^{(0,2)}}$	$\mapsto \mathbf{]}$	2	$\langle 3, (0, 2), 1 \rangle$
$\alpha_2\, \beta_{\mathbf{3}}^{(0,1)}\, \alpha_n$	$\mapsto \mathbf{]}$ for $n \geq 0$	3	$\top$
$\alpha_2\, \beta_{\mathbf{2}}^{(0,1)}\, \alpha_3\, \beta_{\mathbf{1}}^{(0,2)}\, \boxed{\alpha_1\, \overline{\beta}_0^{(2,1)}}$	$\mapsto \mathbf{]}$	21	$\langle 1, (2, 1), 0 \rangle$

For each P-view shown above, note that the subscripts in bold give the corresponding occurrence in the Levy-Longo tree, and the label at that occurrence is specified by the (subscripts and the superscript in the) block that is framed. The first, third and fifth P-views define the "boundary" beyond which the copycat response sets in.

Using an argument similar to the proof of [4, Thm 10.1.23], we can show that every *recursive* closed Levy-Longo pre-tree T is the Levy-Longo tree of some closed λ-term. Thus we have:

Theorem 2 (Universality).

1. *The denotation of a closed* λ*-term* s *is a recursive, relevant, CC strategy which corresponds to* $\mathsf{LT}(s)$ *in the sense of Lemma 6.*
2. *Every recursive, relevant, CC strategy over* $\mathcal{D}$ *is the denotation of a closed* λ*-term. I.e. for every* $\sigma \in \mathbb{L}_{\mathrm{rec}}(\mathbf{1}, \mathcal{D})$ *there is some* $s \in \Lambda^o$ *such that* $[\![\, s \,]\!] = \sigma$*.*

It follows that two closed λ-terms have the same denotation in $\mathcal{D}$ iff they have the same Levy-Longo tree. As a straightforward corollary, the observational quotient of the model then gives a universal, and hence fully abstract, model of the pure Lazy Lambda Calculus.

Acknowledgements. The authors are grateful to EU TMR LINEAR, Merton College, and Oxford University Computing Laboratory for their support.

References

1. S. Abramsky, R. Jagadeesan, and P. Malacaria. Full abstraction for PCF. *Information and Computation*, 163, 2000.
2. S. Abramsky and G. McCusker. Games and full abstraction for the Lazy Lambda Calculus. In *Proc. LICS*, pages 234–243. The Computer Society, 1995.
3. S. Abramsky and C.-H. L. Ong. Full abstraction in the Lazy Lambda Calculus. *Information and Computation*, 105:159–267, 1993.
4. H. Barendregt. *The Lambda Calculus*. North-Holland, revised edition, 1984.
5. V. Danos and R. Harmer. The anatomy of innocence. In *Proc. CSL 2001*, pages 188–202. Springer Verlag, 2001. LNCS Vol. 2142.
6. P. Di Gianantonio. Game semantics for the Pure Lazy Lambda Calculus. In *Proc. TLCA 2001*, pages 106–120. Springer-Verlag, 2001.
7. P. Di Gianantonio, G. Franco, and F. Honsell. Game semantics for the untyped $\lambda\beta\eta$-calculus. In *Proc. TLCA 1999*, pages 114–128. Springer-Verlag, 1999. LNCS Vol. 1591.
8. J. M. E. Hyland and C.-H. L. Ong. On Full Abstraction for PCF: I. Models, observables and the full abstraction problem, II. Dialogue games and innocent strategies, III. A fully abstract and universal game model. *Information and Computation*, 163:285–408, 2000.
9. A. D. Ker, H. Nickau, and C.-H. L. Ong. A universal innocent game model for the Böhm tree lambda theory. In *Proc. CSL 1999*, pages 405 – 419. Springer-Verlag, 1999. LNCS Volume 1683.
10. A. D. Ker, H. Nickau, and C.-H. L. Ong. Innocent game models of untyped λ-calculus. *Theoretical Computer Science*, 272:247–292, 2002.
11. J.-J. Levy. An algebraic interpretation of equality in some models of the lambda calculus. In C. Böhm, editor, *Lambda Calculus and Computer Science Theory*, pages 147–165. Springer-Verlag, 1975. *LNCS* No. 37.
12. G. Longo. Set-theoretical models of lambda calculus: Theories, expansions and isomorphisms. *Annals of Pure and Applied Logic*, 24:153–188, 1983.
13. G. McCusker. Full abstraction by translation. In *Advances in Theory and Formal Methods of Computing*. IC Press, 1996.
14. G. McCusker. *Games for recursive types.* BCS Distinguished Dissertation. Cambridge University Press, 1998.
15. G. D. Plotkin. Call-by-name, call-by-value and the lambda calculus. *Theoretical Computer Science*, 1:125–159, 1975.