A Game Semantics of Linearly Used Continuations

James Laird

COGS, University of Sussex, UK
jiml@cogs.susx.ac.uk

Abstract. We present an analysis of the "linearly used continuation-passing interpretation" of functional languages, based on game semantics. This consists of a category of games with a coherence condition on moves — yielding a fully abstract model of an affine type-theory — and a *syntax-independent* and *full* embedding of a category of HO-style "well-bracketed" games into it. We show that this embedding corresponds precisely to linear CPS interpretation in its action on a games model of the call-by-value (untyped) λ-calculus, yielding a proof of *full abstraction* for the associated translation.

1 Introduction

Continuation-passing-style (CPS) interpretation is widely used for reasoning about typed and untyped functional languages. However, a limitation of the standard CPS interpretation is its failure to capture the constraints on control flow which typically exist in such languages. This is because continuations become first-class objects in the *target* language which may be duplicated and discarded like any other arguments, although this corresponds to behaviour in the *source* language only if the latter also treats continuations as first-class objects (using a construct such as **call/cc**). If not, the target language contains "junk" contexts which can break equivalences which hold in the source language. One solution to this problem is a finer-grained analysis of CPS using *linear types* to control the duplication of continuations. This paper is a semantic investigation of such a "linear CPS interpretation"[1] with the object of establishing its completeness as a basis for reasoning about program equivalence.

Berdine et. al. [6] have given linear CPS translations of the call-by-value λ-calculus — and other control features such as exceptions, jumps and coroutines — into a linear λ-calculus. Syntactic mehods have been used to show that these translations are complete (in the sense that the target language does not contain any "junk" at translated types) [13, 7, 9] but these rely either on the restriction

[1] Notwithstanding the fact that, as emphasized in [6], "it is continuation transformers rather than the continuations which are linear" it seems reasonable to refer to a *linear CPS interpretation*, meaning a continuation-passing-interpretation based on linear types.

A.D. Gordon (Ed.): FOSSACS 2003, LNCS 2620, pp. 313–327, 2003.

to a simply typed source language, or on a heavy restriction of the types in the target language. We shall achieve a more general result by semantic means.

Game semantics has been used to give models of both purely functional languages [4, 10, 16, 3] and non-functional features, including continuations [11], which are free of "junk" and hence *fully abstract*. Another useful feature of games is that intensional phenomena such as control flow and linear use of resources can be represented concretely in terms of behavioural constraints.

1.1 Contribution

The primary aim of this paper is to give a (game) semantics of linear CPS translation. By this we mean the following.

Definition 1. *Let $\mathcal{L}_1, \mathcal{L}_2$ be programming languages, and $\overline{(\_)} : \mathcal{L}_1 \to \mathcal{L}_2$ a translation between them. A semantic interpretation of $\overline{(\_)}$ consists of models $\mathcal{M}_1$ (of $\mathcal{L}_1$) and $\mathcal{M}_2$ (of $\mathcal{L}_2$), and a map $\phi : \mathcal{M}_1 \to \mathcal{M}_2$ such that for all terms M of $\mathcal{L}_1$, $\phi(\llbracket M \rrbracket_{\mathcal{M}_1}) = \llbracket \overline{M} \rrbracket_{\mathcal{M}_2}$.*

However, these formal requirements fail to capture the semantic character of a satisfactory interpretation; given a model $\mathcal{M}_2$ of $\mathcal{L}_2$, we can use a translation to determine its own interpretation by defining $\llbracket M \rrbracket_{\mathcal{M}_1} = \llbracket \overline{M} \rrbracket_{\mathcal{M}_2}$, so that $\phi : \mathcal{M}_1 \to \mathcal{M}_2$ is just an inclusion. By contrast, what we seek is a semantic interpretation in which both the models, and the mapping between them, are defined independently of the syntax.

We shall define a category of "coherence games", $\mathcal{GC}$, in which we give a fully abstract model of a target language for linear CPS translation, λ_{Aff} (a recursively typed, dual affine/non-linear λ-calculus similar to those used in [13, 6]). We shall then give a semantic interpretation of the linear CPS translation as an embedding into $\mathcal{GC}$ of a standard HO-style category of *well-bracketed* games, $\mathcal{WB}$. Specifically, we shall show that if $\overline{(\_)}$ is the linear CPS translation of λ_v (the untyped call-by-value λ-calculus) into λ_{Aff}, and $\llbracket \_ \rrbracket_{\mathcal{WB}}$ is a standard semantics of λ_v in $\mathcal{WB}$ (described in [16]) obtained by solving the domain equation $D = D \Rightarrow D_\perp$ then the following square commutes:

$$\begin{array}{ccc} \lambda_v & \xrightarrow{\llbracket \_ \rrbracket_{\mathcal{WB}}} & \mathcal{WB} \\ \downarrow{\scriptstyle \overline{(\_)}} & & \downarrow{\scriptstyle \phi} \\ \lambda_{\mathrm{Aff}} & \xrightarrow{\llbracket \_ \rrbracket_{\mathcal{GC}}} & \mathcal{GC} \end{array}$$

We will prove completeness of the linear CPS interpretation from a semantic perspective by showing that ϕ is *full*. Moreover, fullness will be used to prove a syntactic completeness result — *full abstraction* — for the translation of λ_v into λ_{Aff}. This can be seen as a version of the "no-junk" condition studied in [7] — we show that the translation introduces no *observable* (i.e. finite) junk.

Definition 2. *A translation* $\overline{(\_)} : \mathcal{L}_1 \to \mathcal{L}_2$ *is equationally* fully abstract *if for all terms* M_1, M_2 *of* $\mathcal{L}_1$*,* M_1 *is observationally equivalent to* M_2 *if and only if* $\overline{M_1}$ *is observationally equivalent to* $\overline{M_2}$*.*

In terms of game semantics, our interpretation can be seen as an analysis of the *bracketing condition* [4, 10]. This is known to correspond to *local control flow* in games. More precisely, relaxing it leads to models of functional languages with non-local control operators such as **call/cc** [11] which are equivalent to models constructed indirectly by CPS interpretation [12]. A connection between bracketing and linearity is evident in one direction; in a well-bracketed sequence, every question has at most one answer. More surprising is a result proved here — that under appropriate conditions, an innocent strategy which answers each question at most once must be well-bracketed.

2 An Affine Target Language for CPS

Despite being chosen independently, the linear λ-caluli used as target languages for the linear CPS translations in [13] and [6] are very similar; both are presented using dual contexts in a similar style to Barber's DILL [5]. We shall study an *affine* version as this is simpler to model and retains the key property that the affine CPS translation of λ_v is fully abstract.

In the terminology of [6], we shall restrict our attention to λ_{Aff} with only *pointed* types (this means, in essence, without sums, including atomic datatypes). However it is straightforward to extend our semantics of λ_{Aff} to include sum types, using the $\mathsf{Fam}(\mathcal{C})$ construction [3]. So types are generated from type-variables together with a single ground type R — the "answer type" of the CPS translation — by the connectives $\Rightarrow, \multimap, \&$ (intuitionistic implication, linear implication, and linear additive product) and a least fixedpoint operator.

$$T ::= X \mid \mathrm{R} \mid T \multimap T \mid T \& T \mid T \Rightarrow T \mid \mu X.T$$

Using $\Rightarrow$ instead of ! to introduce non-linear behaviour results in a much less complex calculus — terms represent derivations in the negative fragment intuitionistic natural deduction (with no "commuting conversions"), but with a finer-grained type-system.

Terms-in-context of λ_{Aff} have the form $\Gamma; \Sigma \vdash M : T$ — i.e. there are two 'zones' (multisets of typed variables) to the left of the turnstyle of which the first is intuitionistic, and the second is affine. The term-language itself is just the λ-calculus with pairing. Unlike [6], a single, standard notation for λ-abstraction and application is used for the introduction and elmination of both $\Rightarrow$ and $\multimap$, so the type which can be assigned to a term-in-context is not unique. As in [6] we take the "equality approach" [1] to recursive types — i.e. we let type-equality be the least congruence on types such that $\mu X.T = T[\mu X.T/X]$ and extend the typing rules in Table 1 with:

$$\frac{\Gamma; \Delta \vdash M : S}{\Gamma; \Delta \vdash M : T} S = T$$

$$\frac{}{\Gamma,x{:}T;\Delta\vdash x{:}T} \qquad \frac{}{\Gamma;\Delta,x{:}T\vdash x{:}T}$$

$$\frac{\Gamma;\Delta,x{:}S\vdash M{:}T}{\Gamma;\Delta\vdash\lambda x.M{:}S\multimap T} \qquad \frac{\Gamma;\Delta\vdash N{:}S \quad \Gamma;\Delta'\vdash M{:}S\multimap T}{\Gamma;\Delta,\Delta'\vdash M\ N{:}T}$$

$$\frac{\Gamma,x{:}S;\Delta\vdash M{:}T}{\Gamma;\Delta\vdash\lambda x.M{:}S\Rightarrow T} \qquad \frac{\Gamma;\_\vdash N{:}S \quad \Gamma;\Delta\vdash M{:}S\Rightarrow T}{\Gamma;\Delta\vdash M\ N{:}T}$$

$$\frac{\Gamma;\Delta\vdash M{:}S \quad \Gamma;\Delta\vdash N{:}T}{\Gamma;\Delta\vdash\langle M,N\rangle{:}S\&T} \qquad \frac{\Gamma;\Delta\vdash M{:}T_1\&T_2}{\Gamma;\Delta\vdash\pi_i(M){:}T_i}$$

Table 1. Typing judgements for λ_{Aff}

Definition 3. *The equational theory of λ_{Aff}, $=_{\beta\eta\pi}$ is generated by the rules:*

$$(\beta)\quad (\lambda x.M)\ N =_{\beta\eta\pi} M[N/x]$$
$$(\eta)\quad \lambda x.(M\ x) =_{\beta\eta\pi} M,\ x\notin FV(t)$$
$$(\pi)\quad \pi_i(\langle M_1,M_2\rangle) =_{\beta\eta\pi} M_i,\ i=1,2$$
$$(\pi_\eta)\quad \langle\pi_1(M),\pi_2(M)\rangle =_{\beta\eta\pi} M$$

We shall not give an explicit operational semantics of λ_{Aff}, but a notion of convergence based on the existence of a head-normal form

Definition 4. *The head-normal forms of λ_{Aff} are given by the grammar:*
$H ::= B \mid \lambda x.H \mid \langle H,H\rangle$*, where:*
$B ::= x \mid B\ M \mid \pi_i(B)$.
For a closed term M, we shall write $M\Downarrow$ if there exists H such that $M =_{\beta\eta\pi} H$.

There is no canonical notion of observational equivalence for λ_{Aff}, because there is no canonical notion of observation. Basically, we have a choice between observing convergence to head-normal form at either linear or non-linear function type, and these are *not* equivalent in general. In the context of the linear CPS translation, it is the former which is most suitable, since it corresponds to the observations available in the source language. (Allowing observations at non-linear function types causes failure of full abstraction for the translation.)

Definition 5. *Let $\simeq_L$ be contextual equivalence of λ_{Aff} terms defined with respect to observations at the linear type* $\mathrm{R}\multimap\mathrm{R}$ *— i.e. if $M,N:T$, then $M\simeq_L N$ if for all closing contexts $C[\_]:\mathrm{R}\multimap R$, $C[M]\Downarrow$ if and only if $C[N]\Downarrow$.*

The source language for our example of linear CPS translation will be the untyped call-by-value λ-calculus, λ_v, which has terms given by the grammar:
$M ::= x \mid \lambda x.M \mid M\ N$,
and an operational semantics given by:

$$\frac{}{\lambda x.P\Downarrow\lambda x.P} \qquad \frac{M\Downarrow\lambda x.P \quad N\Downarrow\lambda y.Q \quad P[\lambda y.Q/x]\Downarrow V}{M\ N\Downarrow V}$$

A standard CPS translation of λ_v can be given an affine typing in λ_{Aff} [6].

Definition 6. *Let* $D = \mu X.(X \Rightarrow (X \Rightarrow \mathrm{R}) \multimap \mathrm{R})$. *A term* M *of* λ_v *with free variables* $x_1, \ldots x_n$ *is interpreted as a* λ_{Aff} *term* $x_1 : D, \ldots, x_n : D; \_ \vdash \overline{M} : (D \Rightarrow \mathrm{R}) \multimap \mathrm{R}$ *defined as follows:*

- $\overline{x} = \lambda k.k\, x$,
- $\overline{\lambda x.M} = \lambda k.k\, (\lambda x.\overline{M})$,
- $\overline{M\, N} = \lambda k.\overline{M}\, (\lambda m.\overline{N}\, (\lambda n.(m\, n)\, k))$.

3 Game Semantics

We shall now present a category of "Hyland-Ong-style" games, containing a semantics of λ_{Aff}, which is universal for finite types. It is based on the addition of a notion of *coherence* to HO arenas in the form of a symmetric relation on moves, and a corresponding refinement of the notion of legal sequence. A second significant feature is that we drop the visibility condition from plays; this actually cuts down the space of innocent and coherent strategies, allowing our universality result to be proved.[2]

First, we shall briefly describe standard notions of *underlying* arenas and innocent strategies which can form a basis for both coherence games (by adding a coherence relation) or the standard "well-bracketed" games (by adding a question/answer labelling to moves).

Definition 7. *An arena* A *is a pair* $\langle M_A, \vdash_A \rangle$ *where* $\vdash_A \subseteq (M_A \cup \{*\}) \times M_A$ *(the* enabling *relation) allows a unique* polarity *to be* inferred *for each move, according to the following rules:*
m is an O-move if it is initial (i.e. $* \vdash a$*), or is enabled by some P-move.*
m is a Player (P)-move if it is enabled by some O-move.

A *justified sequence* over an arena A is a sequence of elements of M_A in which each non-initial move a comes with a pointer to a preceding, enabling move $j_s(a)$. A legal sequence on A is a justified sequence which is *alternating* — Player moves follow Opponent moves and vice-versa. The product and function-space constructions on arenas are standard [10, 16].

Definition 8. *For arenas* A_1, A_2, *define:*

- $A_1 \times A_2 = \langle M_{A_1} + M_{A_2}, \{\langle\langle m, i\rangle, \langle n, i\rangle\rangle \mid m \vdash_{A_i} n\} \cup \{\langle *, \langle m, i\rangle\rangle \mid * \vdash_{A_i} m\}\rangle$,
- $A_1 \to A_2 = \langle M_{A_1} + M_{A_2}, \{\langle\langle m, i\rangle, \langle n, i\rangle\rangle \mid m \vdash_{A_i} n\} \cup \{\langle\langle m, 2\rangle, \langle n, 1\rangle\rangle \mid * \vdash_{A_2} m \wedge * \vdash_{A_1} n\} \cup \{\langle *, \langle m, 2\rangle\rangle \mid * \vdash_{A_2} m\}\rangle$.

The empty arena, with no moves, will be written $\mathbf{1}$.

An innocent strategy will be represented as a set of *Player views* [16].

[2] This departure from existing games models of linear logic is necessary — in particular, although the AJM game semantics of PCF [4] is based on a model of intuitionistic affine type theory which can be used to model λ_{Aff}, this semantics is not fully complete.

Definition 9. *The* Player view *of a non-empty justified sequence s is a sequence with justification pointers, $\ulcorner s \urcorner$, defined by induction as follows:*
$\ulcorner sa \urcorner = \ulcorner s \urcorner a$, *if a is a Player move,*
$\ulcorner sa \urcorner = a$, *if a is an initial Opponent move,*
$\ulcorner satb \urcorner = \ulcorner s \urcorner ab$, *if b is an Opponent move justified by a.*
A legal P-view is a legal sequence s such that $\ulcorner s \urcorner = s$.

Definition 10. *An innocent strategy is a set of even-length legal P-views which is non-empty, even-prefix-closed and even-branching — i.e. $sab, sac \in \sigma$ implies $b = c$.*

From an innocent strategy we can obtain a strategy in the standard form of an even-prefix-closed set of even-length legal sequences by taking its *view-closure*.

Definition 11. *Given a strategy $\sigma : A$, the "view closure" $VC(\sigma)$ is defined to be the least set of legal sequences on A such that $\varepsilon \in VC(\sigma)$, and if $s \in VC(\sigma)$ and $\ulcorner sab \urcorner \in \sigma$, then $sab \in VC(\sigma)$.*

Composition of strategies is by taking the views of the "parallel composition plus hiding" of ther view-closures.

Definition 12. *For $\sigma : A_1 \to A_2, \tau : A_2 \to A_3$: $\sigma; \tau = \{\ulcorner t \upharpoonright A_1, A_3 \urcorner \mid t \upharpoonright A_1, A_2 \in VC(\sigma) \wedge t \upharpoonright A_2, A_3 \in VC(\tau)\}$.*

3.1 Coherence Arenas

Given an arena A, we say that moves $m, n \in M_A$ are *co-enabled* if $\exists l \in (M_A)_*.(l \vdash_A m) \wedge (l \vdash_A n)$.

Definition 13. *A coherence arena (or C-arena) A is a pair $\langle |A|, \sim_A \rangle$ consisting of an underlying arena $|A|$ together with a symmetric relation $\sim_A$ between co-enabled moves of $|A|$.*
A coherent sequence of A is a legal sequence s of $|A|$ which satisfies the following conditions:

- *All initial moves in s are coherent: if $ta, t'a' \sqsubseteq s$ and $* \vdash a, b$ then $a \sim_A b$.*
- *Any two non-initial moves in s with the same justifier are coherent: if $ta, t'a' \sqsubseteq s$ and $j_s(a) = j_s(b)$ then $a \sim_A b$.*

We can now define notions of additive and multiplicative product; both are based on the product of the underlying arenas, but they are differentiated by varying the coherence relations on initial moves. This distinction can be summarised: in $A \otimes B$ every initial move of A is coherent with every initial move of B, whereas in $A\&B$ every initial move of A is incoherent with every initial move of B. Hence a coherent sequence of $A \otimes B$ consists of a pair of interleaved sequences of A and B, and a coherent sequence of $A\&B$ is a sequence wholly from A or wholly from B.

Definition 14. *Given C-Arenas A_1, A_2, define the following C-arenas:*

Tensor Product $A_1 \otimes A_2 = \langle |A_1| \times |A_2|, \sim_{A_1 \otimes A_2} \rangle$ *where* $\langle m, i \rangle \sim_{A_1 \otimes A_2} \langle n, j \rangle$ *iff* $i = j$ *implies* $m \sim_{A_i} n$.
Additive Product $A_1 \& A_2 = \langle |A_1| \times |A_2|, \sim_{A_1 \& A_2} \rangle$ *where* $\langle m, i \rangle \sim_{A_1 \& A_2} \langle n, j \rangle$ *iff* $i = j$ *and* $m \sim_{A_i} n$.
Linear Function Space $A_1 \multimap A_2 = \langle |A_1| \to |A_2|, \sim_{A_1 \multimap A_2} \rangle$, *where* $\langle m, i \rangle \sim_{A_1 \multimap A_2} \langle n, j \rangle$ *iff* $i = j$ *implies* $m \sim_{A_i} n$.

We shall say that a strategy is *coherent* if it is never the first participant in a dialogue to violate the coherence condition.

Definition 15. *For any C-arena A, an innocent strategy on A is coherent if whenever $sa \in VC(\sigma)$ and s is coherent then sa is coherent.*

However, we cannot define a category of C-arenas in the standard fashion by taking morphisms from A to B to be coherent strategies on $A \multimap B$ fails; the composition of coherent strategies is not coherent (see [12]). To solve this problem, we place a further restriction on the coherent strategies which are permitted as morphisms.

Definition 16. *A C-strategy from A to B is a coherent strategy on $A \multimap B$ such that if $s \in VC(\sigma)$ is coherent, then every two moves in s which are initial in A are coherent — i.e. if $ta, t'a' \sqsubseteq s$ and $* \vdash_A a, b$ then $a \sim_A b$.*

It is now straightforward to show that the composition of innocent C-strategies is an innocent C-strategy and so we can define category $\mathcal{GC}$ with C-arenas as objects and C-strategies from A to B as morphisms from A to B.

Proposition 1. $(\mathcal{GC}, \mathbf{1}, \otimes)$ *is a symmetric monoidal category with a cartesian product, &.*

The price we have paid is the loss of the symmetric monoidal *closed* structure — $\mathcal{GC}$ does not have all exponentials in the following sense.

Definition 17. *Let A, B be objects in a symmetric monoidal category $(\mathcal{C}, I, \otimes)$. An exponential of B by A is a an object B^A such that for all C in $\mathcal{C}$, there is an isomorphism:* $\mathrm{ev}_{A,B} : \mathcal{C}(C \otimes A, B) \cong \mathcal{C}(C, B^A)$ *which is natural in C.*

We have an isomorphism of arenas — $(A \otimes B) \multimap C \cong A \multimap (B \multimap C)$ — but not, in general $\mathcal{GC}(A \otimes B, C) \cong \mathcal{GC}(A, B \multimap C)$. However, to model λ_{Aff}, we do not require that B^A exists for any arena B, but only for B within a specified collection of *well-opened* arenas, for which the notions of coherent strategy and C-strategy coincide.

Definition 18. *Say that a C-arena A is* well-opened *if for any pair of initial moves $m, n \in M_A$ (not necessarily distinct), $m \not\sim_A n$. We shall write $\mathcal{WO}$ for the full subcategory of $\mathcal{GC}$ consisting of well-opened C-arenas.*

Lemma 1. *If B is well-opened, then for any C-arena A, $A \multimap B$ is an exponential of B by A.*

Proof. If D is any C-arena, then *every* coherent strategy σ on $D \multimap B$ is a C-strategy from D to B, since by well-openedness of B, every coherent sequence on $D \multimap B$ contains at most one initial move, and so any two initial moves of D in s must have the same justifier, and hence be coherent. So for any arena C, $\mathcal{GC}(C \otimes A, B) \cong \mathcal{GC}(\mathbf{1}, (C \otimes A) \multimap B) \cong \mathcal{GC}(\mathbf{1}, C \multimap (A \multimap B)) \cong \mathcal{GC}(C, A \multimap B)$ as required.

Lemma 2. *If B is well-opened, then for any A, $A \multimap B$ is well-opened, and if A and B are well opened, then $A\&B$ is well-opened.*

We can now use coherence to define a monoidal co-monad $! : \mathcal{GC} \to \mathcal{GC}$. The C-arena $!A$ has the same underlying arena as A, but all of the "incoherences" between the initial moves of A are removed.

Definition 19. *For any C-arena A, define $!A = \langle |A|, \sim_{!A} \rangle$, where $m \sim_{!A} n$, if $m \sim_A n$ or $* \vdash m, n$.*

Thus for a well-opened arena A, the coherent sequences of $!A$ consist of multiple interleaved sequences ("threads") of A. The action of $!$ on morphisms is simple: if $\sigma : !A \to B$ is a coherent strategy, then $\sigma^\dagger : !A \to !B$ has the same underlying strategy. For each C-arena A we have $\mathtt{der}_A : !A \to A$ which has the same underlying strategy as the identity. We also have equalities of arenas: $!(A\&B) = !A \otimes !B = !(A \otimes B)$ yielding the monoidal natural transformations and contraction maps $\mathtt{con}_A : !A \to !A \otimes !A$ for each A. Thus we can define $A \Rightarrow B = !A \multimap B$.

3.2 Interpretation of $\boldsymbol{\lambda_{\mathbf{Aff}}}$ in $\mathcal{GC}$

We have defined functors $\_\&\_ : \mathcal{WO} \times \mathcal{WO} \to \mathcal{WO}$, $\_ \multimap \_ : \mathcal{WO}^{OP} \times \mathcal{WO} \to \mathcal{WO}$ and $\_ \Rightarrow \_ : \mathcal{WO}^{OP} \times \mathcal{WO} \to \mathcal{WO}$ which we shall use to interpret the corresponding type-constructors in λ_{Aff}. To interpret R, we must identify a well-opened answer-object, and to eliminate "junk" we choose the smallest such non-terminal arena.

Definition 20. *Let o be the arena with one move which is not coherent with itself, i.e. $M_o = \{o\}$, $\vdash_o = \langle *, o \rangle$ and $o \not\sim_o o$.*

To interpret recursive types in $\mathcal{GC}$, we use the "information-system"-like approach studied in depth in [16], which allows domain equations to be solved up to equality, as required by λ_{Aff}. First, we define an inclusion order on C-arenas.

Definition 21. *$A_1 \trianglelefteq A_2$ if:*

- $M_{A_1} \subseteq M_{A_2}$,
- $\vdash_{A_1} = \vdash_{A_2} \cap ((M_{A_1})_* \times M_{A_1})$
- $\sim_{A_1} = \sim_{A_2} \cap (M_{A_1} \times M_{A_1})$.

If $A \trianglelefteq B$ then there are obvious inclusion and projection maps $A \rightarrow^{\mathtt{in}} B \rightarrow^{\mathtt{out}} A$ such that $\mathtt{in}; \mathtt{out} = \mathsf{id}_{\mathtt{A}}$ and $\mathtt{out}; \mathtt{in} \subseteq \mathsf{id}_{\mathtt{B}}$.

Proposition 2. *C-Arenas form a large cpo, ordered by* $\trianglelefteq$.

Each of the functors $\&, \multimap$ and $!$ are continuous with respect to $\trianglelefteq$. Thus we can define a least fixed point operator by iteration.

Definition 22. *Given a continuous functor* $F : \mathcal{WO}^{OP} \times \mathcal{WO} \rightarrow \mathcal{WO}$, *let* $\Delta(F) \in \mathcal{WO} = \bigsqcup_{i \in \omega} F^i(\mathbf{1})$.

Then $F(\Delta(F), \Delta(F)) = \Delta(F)$ and so we have a sound model of λ_{Aff}, based on the following interpretations of types with n free variables as mixed-variance functors from $(\mathcal{WO}^{OP} \times \mathcal{WO})^n$ to $\mathcal{WO}$:

$$\begin{array}{cc} [\![\mathrm{R}]\!] = o & [\![X]\!] = \pi_r \\ [\![S \multimap T]\!] = [\![S]\!] \multimap [\![T]\!] & [\![S \Rightarrow T]\!] = ![\![S]\!] \multimap [\![T]\!] \\ [\![S \& T]\!] = [\![S]\!] \& [\![T]\!] & [\![\mu X.T(X, Y_1, \ldots Y_n)]\!] = \Delta([\![T(\cdot, Y_1, \ldots, Y_n)]\!]) \end{array}$$

Each term-in-context $x_1 : S_1, \ldots, x_m : S_m; y_1 : T_1, \ldots y_n : T_n \vdash M : U$ is thus interpreted as a natural transformation from $![\![S_1]\!] \otimes \ldots \otimes ![\![S_k]\!] \otimes [\![T_1]\!] \otimes \ldots \otimes [\![T_m]\!]$ to $[\![U]\!]$ following e.g. [5]. (We shall write $[\![S_1, \ldots, S_n; T_1, \ldots T_m \vdash U]\!]$ for the set of such natural transformations.)

Using approximation relations as in [17, 16], we prove the following computational adequacy result.

Proposition 3. *For any program* M *of* λ_{Aff}, $M \Downarrow$ *if and only if* $[\![M]\!] \neq \bot$.

We also have a universality result for *finite types*, and hence a full abstraction result with respect to an intrinsic preorder.

Definition 23. *The (closed) finite types of* λ_{Aff} *are generated by the grammar:* $F ::= \mathrm{R} \mid F \& F \mid F \multimap F \mid F \Rightarrow F$

For an innocent strategy σ, let $\#(\sigma)$ be the cardinality of σ as a set of views, so we may say that σ is finite if $\#(\sigma)$ is finite.

Proposition 4. *Let* F *be a finite type, and* Γ, Δ *contexts of finite types. Then every finite, innocent C-strategy* $\sigma \in [\![\Gamma; \Delta \vdash F]\!]$ *is definable — i.e. there exists a* λ_{Aff}*-term-in-context* $\Gamma; \Delta \vdash M_\sigma : F$ *such that* $\sigma = [\![M_\sigma]\!]$.

Proof. (Sketch) We apply an inductive decomposition similar to the decomposition theorem for the simply-typed λ-calculus, described axiomatically in [2]. Formally, proof is by induction on $\#(\sigma)$. If this is zero then σ is the empty strategy, which is definable as a divergent program. We prove the inductive case by a series of lemmas, all of which are implicitly dependent on the inductive hypothesis. The first is a subcase for *strict* strategies[3] and π*-atomic* types, which are given by the following grammar:
$P ::= \mathrm{R} \mid F \multimap P \mid F \Rightarrow P$.
We shall write π-atomic types in the form $S_1 \Rightarrow \ldots \Rightarrow S_m \Rightarrow T_1 \multimap \ldots \multimap (T_n \multimap \mathrm{R})$, or $\boldsymbol{S} \Rightarrow (\boldsymbol{T} \multimap \mathrm{R})$.

[3] A strategy in $\mathcal{GC}(A, B)$ is *strict* if the first P-move in σ (if any) is in A.

Lemma 3. *If P_1, P_2 are π-atomic types and $\sigma \in [\![\_; P_1 \vdash P_2]\!]$ is strict then σ is definable.*

Proof. By induction on the size of P_2. The base case is $P_2 = \mathrm{R}$ and $P_1 = \boldsymbol{S} \Rightarrow (\boldsymbol{T} \multimap \mathrm{R})$ — using the induction hypothesis on $\#(\sigma)$ we can find $\boldsymbol{M} : \boldsymbol{S}$ and $\boldsymbol{N} : \boldsymbol{T}$ such that $\sigma = [\![x : P_1 \vdash ((x\ \boldsymbol{M})\ \boldsymbol{N}) : \mathrm{R}]\!]$. The induction cases are:

- If $P_2 = C \Rightarrow P$, then by induction (on P_2) we can find a term $\_; x : (\boldsymbol{C \Rightarrow S}) \Rightarrow ((\boldsymbol{C \Rightarrow T}) \multimap \mathrm{R}) \vdash M : P$ such that $\sigma = [\![y : \boldsymbol{S} \Rightarrow (\boldsymbol{T} \multimap \mathrm{R}) \vdash \lambda z.M[(\lambda \boldsymbol{uv}.(y\ \boldsymbol{(u\ z)}\ \boldsymbol{(v\ z)}))/x] : C \Rightarrow P]\!]$
- If $P_2 = C \multimap P$ then there exists $i \leq m$ and $x : \boldsymbol{S} \Rightarrow (\boldsymbol{T'} \multimap \mathrm{R}) \vdash M : P$ (where $T'_i = C \multimap T_i$ and $T'_j = T_j$ for $j \neq i$) such that $\sigma = [\![\_; y : \boldsymbol{S} \Rightarrow (\boldsymbol{T} \multimap \mathrm{R}) \vdash \lambda z.M[(\lambda \boldsymbol{u}.\lambda \boldsymbol{v}.y\ \boldsymbol{u} v_1 \dots v_{i-1}(v_i\ z) v_{i+1} \dots v_m)/x] : C \multimap P]\!]$

Lemma 4. *If F is any finite type, P is a π-atomic type, and $\sigma \in [\![\_; F \vdash P]\!]$ is strict then σ is definable.*

Proof. is by induction on the size of F. If this is an atomic formula then Lemma 3 applies. Otherwise $F = \boldsymbol{S} \Rightarrow (\boldsymbol{T} \multimap U_1 \& U_2)$ and we can find $i \in \{1, 2\}$ and a term-in-context $x : \boldsymbol{S} \Rightarrow (\boldsymbol{T} \multimap U_i) \vdash M : P]\!]$ such that $\sigma = [\![y : \boldsymbol{S} \Rightarrow (\boldsymbol{T} \multimap U_1 \& U_2) \vdash M[\lambda \boldsymbol{u}.\lambda \boldsymbol{v}.\pi_i(y\ \boldsymbol{uv})/x] : P]\!]$.

Lemma 5. *If $\sigma \in [\![S_1, \dots, S_m; T_1, \dots, T_n \vdash \mathrm{R}]\!]$ is strict then σ is definable.*

Proof. If the first P-move in σ is in some $[\![S_i]\!]$, then using Lemma 4 we can find a term $\_; x_i : S_i \vdash M : \boldsymbol{S} \Rightarrow \boldsymbol{T} \multimap \mathrm{R}$ such that $\sigma = [\![x_1 : S_1, \dots, x_m : S_m; y_1 : B_1, \dots, y_n : B_n \vdash (M\ \boldsymbol{x})\ \boldsymbol{y} : \mathrm{R}]\!]$.
If the first P-move in σ is in some $[\![T_j]\!]$, then we can find a term $\_; y : T_j \vdash M : \boldsymbol{S} \Rightarrow T_1 \multimap \dots \multimap T_{j-1} \multimap T_{j+1} \multimap \dots \multimap T_k \multimap \mathrm{R}$ such that $\sigma = [\![x_1 : S_1, \dots, x_n : S_m; y_1 : T_1, \dots, y_n : T_n \vdash (M\ \boldsymbol{x})\ y_1 \dots y_{j-1} y_{j+1} \dots y_k : \mathrm{R}]\!]$.

Finally, we can complete the induction to prove Proposition 4 by showing that if $\sigma \in [\![\Gamma; \Delta \vdash F]\!]$ then σ is definable, by induction on the size of F. If $F = \mathrm{R}$, then σ must be strict, and Lemma 5 applies. Otherwise $F = F_1 \& F_2$, or $F = F_1 \multimap F_2$ or $F = F_1 \Rightarrow F_2$ and can be decomposed and reconstructed by projection/pairing or uncurrying/currying.

Definition 24. *Define the* intrinsic equivalence $\approx$ *on strategies $\sigma, \tau : A$:*
$\sigma \approx \tau$ if for all C-strategies $\rho : A \to (o \multimap o)$, $\sigma; \rho = \bot$ if and only if $\tau; \rho = \bot$.

The proof of the full abstraction follows a standard pattern [4, 10, 16], based on Proposition 4 and the fact that for any closed λ_{Aff} type, T there is a chain of finite types $F_1, F_2, \dots$ such that $[\![T]\!] = \bigsqcup_{i \in \omega} [\![F_i]\!]$.

Theorem 1 (Full Abstraction). *For all λ_{Aff}-terms $M, N : T$, $M \simeq_L N$ if and only if $[\![M]\!] \approx [\![N]\!]$.*

4 Well-Bracketing and Linear CPS

We can now show that a variant of the original category of HO games [10] (and its extension by McCusker with lifted sums [16]) can be fully embedded in the category of coherence games, and that this embedding preserves denotations with respect to linear CPS translation.

Definition 25. *A bracketed arena (or B-arena) is a pair* $\langle |A|, \lambda_A \rangle$ *consisting of an underlying arena* $|A|$ *(as per Definition 7), with a labelling function* $\lambda_A : M_A \to \{Q, A\}$*, which partitions the set of moves into* questions *and* answers*. Answers must be enabled (and must only be enabled) by questions.*

The product and function-space of bracketed arenas are based on the corresponding constructions on the underlying arenas — i.e. $A \times B = \langle |A| \times |B|, [\lambda_A, \lambda_B] \rangle$ and $A \Rightarrow B = \langle |A| \to |B|, [\lambda_A, \lambda_B] \rangle$. The rôle of answer labelling is to allow the definition of the *bracketing condition*.

Definition 26. *For each justified sequence,* s*, we define a prefix* $\mathsf{pending}(s) \sqsubseteq s$ *as follows:*
$\mathsf{pending}(\varepsilon) = \varepsilon$,
$\mathsf{pending}(sq) = sq$ *if* q *is a question,*
$\mathsf{pending}(sqta) = \mathsf{pending}(s)$*, if* a *is an answer justified by* q*.*

The pending question of s is the final move in $\mathsf{pending}(s)$, if any.

Definition 27. *An alternating justified sequence* s *on a bracketed arena is* well-bracketed *if every answer in* s *is justified by the pending question — i.e. if* $rqta \sqsubseteq s$ *and* a *is an answer justified by* q*, then* $rq = \mathsf{pending}(rqt)$*.*
An innocent and well-bracketed *strategy on a B-arena is an innocent strategy on the underlying arena such that whenever* $sab \in VC(\sigma)$ *and* sa *is well-bracketed then* sab *is well-bracketed.*

The following lemma, which characterizes innocent and well-bracketed strategies in terms of their P-views is proved in [11, 12].

Lemma 6. *An innocent strategy* σ *is well-bracketed if and only if for all* $s \in \sigma$*,* s *is well-bracketed.*

The composition of innocent and well-bracketed strategies (as defined in Definition 12) is well-bracketed and thus we have a cartesian closed category $(\mathcal{WB}, \mathbf{1}, \times, \Rightarrow)$ with B-arenas as objects and innocent well-bracketed strategies on $A \Rightarrow B$ as morphisms from A to B [10, 12]. $\mathcal{WB}$ also has a *lifted sum* construction [16], obtained by adding an initial question and answers corresponding to each summand. To give a semantics of λ_v, we only require the unary version — lifting.

Definition 28. *For any B-arena* A*, the lifting* $A_\perp$ *is defined as follows:*

- $M_{A_\perp} = (\{q\} \cup \{a\}) + M_A$,
- $\vdash_{A_\perp} = \{\langle *, \mathtt{inl}(q) \rangle, \langle \mathtt{inl}(q), \mathtt{inl}(a) \rangle\} \cup \{\langle \mathtt{inl}(a), \mathtt{inr}(b) \rangle \mid * \vdash_A b\} \cup \{\langle \mathtt{inr}(m), \mathtt{inr}(n) \mid m \vdash_A n\}$,
- $\lambda_{A_\perp}(\mathtt{inl}(q)) = Q, \quad \lambda_{A_\perp}(\mathtt{inl}(a)) = A, \quad \lambda_{A_\perp}(\mathtt{inr}(b)) = \lambda_A(b)$.

There is an evident operation taking $\sigma : A \Rightarrow B_\perp$ to $\sigma^* : A_\perp \Rightarrow B_\perp$, and well-bracketed strategies $\eta_A : A \to A_\perp$ and $t_{A,B} : A \times B_\perp \to (A \times B)_\perp$, making lifting a strong monad on $\mathcal{WB}$. We can solve domain equations up to equality in $\mathcal{WB}$, just as in $\mathcal{GC}$ — by taking fixed points of mixed variance functors which are minimal with respect to the inclusion order — $A \trianglelefteq B$ if $|A| \trianglelefteq |B|$ and $\lambda_A = \lambda_B \restriction M_B$. Thus a semantics of the call-by-value λ-calculus can be defined in standard fashion.

Definition 29. *Let D be the least solution to $D = D \Rightarrow D_\perp$ in $\mathcal{WB}$ — i.e. $D = \bigsqcup_{i\in\omega} F^i(\mathbf{1})$, where $F(X) = X \Rightarrow X_\perp$. For each term-in-context $x_1, \ldots x_n \vdash M$ of λ_v, let $[\![M]\!]_{\mathcal{WB}} : D^n \to D_\perp$ be the innocent, well-bracketed strategy denoting M, defined as in [16].*

Let $\Sigma = \mathbf{1}_\perp$ be the game with a single question and answer, and define the *intrinsic equivalence* $\approx$ on well-bracketed strategies $\sigma, \tau : A$ as follows:
$\sigma \approx \tau$ if for all $\rho : A \to \Sigma$, $\sigma; \rho = \perp$ if and only if $\tau; \rho = \perp$.
We can now state McCusker's full abstraction result for the model of λ_v in $\mathcal{WB}$.

Proposition 5 (McCusker [16]). *For any λ_v terms M, N: $M \simeq N$ if and only if $[\![M]\!]_{\mathcal{WB}} \approx [\![N]\!]_{\mathcal{WB}}$.*

4.1 Embedding the Well-bracketed Games into $\mathcal{GC}$

From each B-arena A we can define a C-arena $\phi(A)$ with the same underlying structure, and a coherence relation in which moves are incoherent if and only if they are answers to the same question.

Definition 30. *$\phi(A) = \langle |A|, \sim_A \rangle$, where $m \not\sim_A n$ if $\lambda_A(m) = \lambda_A(n) = A$, and $m \sim_A n$ otherwise.*

ϕ acts as the identity on underlying innocent strategies. To show that this defines a functor, we need to show that any innocent and well-bracketed strategy on a B-arena is a coherent strategy on the associated C-arena.

Lemma 7. *If sqt is a coherent sequence on $\phi(A)$ and q is a question such that $\mathsf{pending}(sqt) = \mathsf{pending}(s)$, then q is answered in t.*

Proof. is by induction on the length of sqt. Given $sqtb$ where b is an O move, we have *either* that b is a question — in which case $\mathsf{pending}(sqtb) \neq \mathsf{pending}(s)$ — *or* b is an answer to a question q' in t — i.e. $t = t'q't''b$, where $\mathsf{pending}(sqt') = \mathsf{pending}(s)$ and hence q is answered in t' by induction hypothesis — or b is an answer to a question q' in s — i.e. $s = s'q't'$, and $\mathsf{pending}(s'q't') = \mathsf{pending}(sqtb) = \mathsf{pending}(s')$ and hence q' is already answered in t' which contradicts coherence of $sqtb$.

Proposition 6. *For any B-arena A, if σ is a well-bracketed innocent strategy on A then it is a coherent strategy on $\phi(A)$.*

Proof. It is sufficient to show that if $sbc \in VC(\sigma)$ and sb is coherent, then $\mathsf{pending}(sb)$ has not already been answered and hence sbc is coherent. We show this by induction on sequence length: Given an (odd-length) sequence sb, either b is a question — in which case it is the pending question, and unanswered — or b is an answer, and $sb = s'qtb$, where b answers q and so $\mathsf{pending}(sb) = \mathsf{pending}(s')$. The pending question in sb is not answered in s by inductive hypothesis, and so the only remaining possibility is that it is answered in t. In other words there is a prefix $s'qt'a \sqsubseteq s$, in which a answers $\mathsf{pending}(s) = \mathsf{pending}(s')$. But by assumption we have $\mathsf{pending}(s'qt') = \mathsf{pending}(s')$ and hence by Lemma 7 q is answered in t', which contradicts coherence of sb.

Functoriality and injectivity of ϕ are immediate, so it remains to show *fullness* of the embedding. It is clearly not the case in general that if s is a coherent sequence on $\phi(A)$ then s is well-bracketed; the coherence condition prevents repetition of answers, but not *jumping* — answering a question out of turn. However, we can show that if σ is an innocent and coherent strategy on $\phi(A)$, then σ must be well-bracketed, as otherwise Opponent can force σ to violate coherence because of innocence. A non-well-bracketed strategy on $(o \Rightarrow o) \Rightarrow \Sigma$ which translates to a non-coherent strategy on $(o \Rightarrow o) \Rightarrow (o \multimap o)$ is depicted in Figure 1.

Proposition 7. *For any C-strategy $\sigma : \phi(A) \to \phi(B)$, σ is a well-bracketed strategy on $A \Rightarrow B$.*

Proof. By Lemma 6 it suffices to show that every $s \in \sigma$ is well-bracketed. Suppose for a contradiction that we have a minimal length sequence $sqta \in \sigma$ such that q is the pending question in sqt but a is not justified by q. Then the justified sequence $sqtaqt$ in which the two q moves have the same justifier is a coherent sequence on $\phi(A \Rightarrow B)$. But $\ulcorner sqtat \urcorner = sqt$ and hence $sqtata \in VC(\sigma)$, which violates the coherence condition.

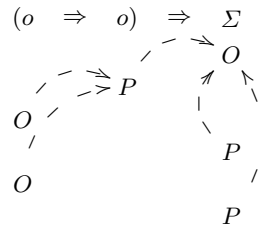

Fig. 1. A non-well-bracketed strategy forced to violate coherence

Theorem 2. *The category of bracketed arenas and innocent, well-bracketed strategies embeds fully in the category of C-arenas and C-strategies.*

It remains to observe that interpreting λ_v in $\mathcal{WB}$ followed by embedding in $\mathcal{GC}$ corresponds to interpretation via linear CPS translation. The embedding preserves the standard cartesian closed structure of $\mathcal{WB}$ — the key point is that it also respects the definitions of lifting in the following sense.

Lemma 8. *If A is a B-arena then $\phi(A_\perp) = (\phi(A) \Rightarrow o) \multimap o$.*

Hence $\phi(D) = [\![\mu X.X \Rightarrow ((X \Rightarrow \mathrm{R}) \multimap \mathrm{R}]\!]_{\mathcal{GC}}$ and it is straightforward to check that the embedding commutes with linear CPS translation as required.

Proposition 8. *For any program M of λ_v, $\phi([\![M]\!]_{\mathcal{WB}}) = [\![\overline{M}]\!]_{\mathcal{GC}}$.*

Theorem 3. *The linear CPS translation from λ_v to λ_{Aff} is fully abstract.*

Proof. Soundess is straightforward. For completeness, suppose $\overline{M} \not\simeq_L \overline{N}$. Then $[\![\overline{M}]\!]_{\mathcal{GC}} \not\simeq [\![\overline{N}]\!]_{\mathcal{GC}}$ by Theorem 1 — i.e. w.l.o.g. there exists a coherent strategy $\rho : [\![(\mu X.(X \Rightarrow X_\perp)_\perp]\!]_{\mathcal{GC}} \to \mathrm{R} \multimap \mathrm{R}$ such that $[\![\overline{M}]\!]; \rho = \perp$ and $[\![\overline{N}]\!]; \rho \neq \perp$. By Proposition 7, ρ is a well-bracketed strategy on $D_\perp \to \Sigma$ such that $[\![M]\!]_{\mathcal{WB}}; \rho = \perp$ and $[\![N]\!]_{\mathcal{WB}}; \rho \neq \perp$ so by Proposition 5 $M \not\simeq N$.

5 Conclusions and Further Directions

There are several possibilities for extending this work. On the logical side, Streicher's domain equation for the designs of Ludics [18] corresponds to a type of λ_{Aff}; the correspondence between designs and the strategies at this type seems to parallel closely the account given in [8]. There is also a close relationship between λ_{Aff}, and proofs in polarized classical linear logic with both lifting and the exponentials, and we can construct a model of the latter along the lines of [15] which is, moreover, fully complete.

As described in [6], several other control features, including exception-handling, `GOTO`-style jumps and coroutines can be given linear CPS translations in λ_{Aff}, and hence modelled in our category of games. However, apart from exception-handling (for which the linear CPS interpretation is essentially equivalent to the standard monadic interpretation using a lifted sum type) these features are not considered in a higher-order setting, which is a significant limitation. One problem is that features such as coroutines and locally declared exceptions are "hybrid effects" which manipulate control flow but have "state-like" features, making them difficult to capture via translation into λ_{Aff} on its own. Game semantics has been proposed as the basis for a detailed semantic account of such hybrid effects [14], since it allows state and control to be combined seamlessly. However, one apparent difficulty in giving a linear continuation passing semantics of state and state-like features is that by storing continuations, a program can perform jumps in the flow of control whilst still using its continuations linearly. In the work described here, this corresponds to the reliance on *innocence* (i.e. purely functional behaviour) to prove fullness of the embedding corresponding to linear CPS translation.

Acknowledgements

This research was supported by UK EPSRC grant GR/N 38824. I would like to thank Guy McCusker, Thomas Streicher, Peter O'Hearn, Russ Harmer and Matthew Wall for useful discussions.

References

[1] M. Abadi and M. P. Fiore. Syntactic considerations on recursive types. In *Proceedings of the Eleventh Annual Symposium on Logic in Computer Science, LICS '96*, 1996.

[2] S. Abramsky. Axioms for full abstraction and full completeness. In *Essays in Honour of Robin Milner*. MIT Press, 1997.

[3] S. Abramsky and G. McCusker. Call-by-value games. In M. Neilsen and W. Thomas, editors, *Computer Science Logic:* 11^{th} *Annual workshop proceedings*, LNCS, pages 1–17. Springer-Verlag, 1998.

[4] S. Abramsky, R. Jagadeesan and P. Malacaria. Full abstraction for PCF. *Information and Computation*, 163:409–470, 2000.

[5] A. Barber. Dual intuitionistic linear logic. Technical Report ECS-LFCS-96-347, LFCS, University of Edinburgh, 1996.

[6] Josh Berdine, Peter O'Hearn, Uday Reddy, and Hayo Thielecke. Linear continuation-passing. *Higher Order Symbolic Computation*, 15(2/3):181–208, September 2002.

[7] Josh Berdine, Peter W. O'Hearn, and Hayo Thielecke. On affine typing and completeness of CPS. Draft, December 2000.

[8] C. Faggian and J. M. E. Hyland. Designs disputes and strategies. In *Proccedings of CSL '02*, 2002.

[9] M. Hasegawa. Linearly used effects: monadic and cps transformations into the linear lambda calculus. In *Proc. 6th International Symposium on Functional and Logic Programming (FLOPS2002), Aizu*, number 2441 in LNCS, pages 167–182. Springer, 2002.

[10] J. M. E. Hyland and C.-H. L. Ong. On full abstraction for PCF: I, II and III. *Information and Computation*, 163:285–408, 2000.

[11] J. Laird. Full abstraction for functional languages with control. In *Proceedings of the Twelfth International Symposium on Logic In Computer Science, LICS '97*. IEEE Computer Society Press, 1997.

[12] J. Laird. *A Semantic Analysis of Control*. PhD thesis, Department of Computer Science, University of Edinburgh, 1998.

[13] J. Laird. Finite models and full completeness. In *Proceedings of CSL '00*, number 1862 in LNCS. Springer, 2000.

[14] J. Laird. A fully abstract game semantics of local exceptions. In *Proceedings of the Sixteenth International Symposium on Logic In Computer Science, LICS '01*. IEEE Computer Society Press, 2001.

[15] O. Laurent. Polarized games. In *Proceedings of the Seventeenth International Symposium on Logic In Computer Science, LICS '02*, 2002.

[16] G. McCusker. *Games and full abstraction for a functional metalanguage with recursive types*. PhD thesis, Imperial College London, 1996.

[17] A. M. Pitts. Relational properties of domains. *Information and Computation*, 127:66–90, 1996.

[18] T. Streicher. A domain equation for the designs of Ludics. 2002.