Dialogue Games and Innocent Strategies: An Approach to (Intensional) Full Abstraction for PCF

Preliminary Announcement

Martin Hyland and *Luke Ong*

26th July 1993

This note is (intended to be) released in conjunction with a preliminary announcement of Abramsky, Jagadeesan and Malacaria entitled *Games and Full Abstraction of PCF*. Like Abramsky *et al.* (but independently), we have found an intensionally fully abstract model for PCF [Plo77]. Our model is a Cartesian closed category of Scott domains all of whose compact elements are definable in PCF. Using Stoughton's Theorem [Sto88], the model can be extensionally collapsed by means of a continuous homomorphism to the least fixpoint, order-extensional, fully abstract model which is shown to be unique by Milner [Mil77]. It is unclear at this stage how our model relates to that of Abramsky *et al.*

Our model of computation is based on a kind of game in which each play consists of a dialogue of questions and answers between two players. This approach is very concrete and in nature goes back to Kleene [Kle78] and Gandy in one tradition, and to Kahn and Plotkin [KP78], and Berry and Curien [BC82] in another. This work has also been informed by the game semantical paradigm of [AJ92]. We analyze PCF-style computations *directly* in terms of partial strategies based on the information available to each player when he is about to move. Hence our games are not history-sensitive; but they are not history-free either. Rather each player selects only that part of the "history" that interests him currently: he has a *view* of current concerns. These dialogue games have two important features:

- No question may be answered until all subsidiary questions are answered. This is a version of Gandy's "no dangling question marks" condition.
- Also our players are required to play an "innocent" strategy: they play on the basis of their current views which are continually updated as the play unfolds.

They give expression to what seems to us to be the nub of PCF-style sequentiality.

Dialogue Games over Computational Arenas

A *simple computational arena* consists of the following data:

- A partially order set of questions such that the upper set of each question is a finite linear order. So the questions form an upside down forest (of trees), the root of each tree being a maximal element in the ordering.
- An association to each question of a set of possible answers. Any such answer is said to *match* the question.

Questions of depth 0, 2, 4, *etc.* are associated with Opponent (O), and those of depth 1, 3, 5, *etc.* are associated with Player (P). The questions of depth 0 (corresponding to the roots of trees) are initial questions, and they have a special status. The ordering over questions expresses a dependency relationship. We say that a question q *justifies* another, say q', if and only if q is the unique element immediately above q' in the ordering. The point of separating moves into questions and answers (represented generically as "(" and ")" respectively) is to enable the introduction of a notion of *balanced play.* A play is a sequence of moves (of a certain kind). A balanced play is one with an equal number (possibly zero) of questions and matching answers such that each subplay bounded on the left by a question and on the right by its matching answer is in turn a balanced play. For example, $(()())((())())$ is balanced, but $)($ or $())$ are not; nor is $()$ if the answer $)$ is does not match the question $($. A play is *potentially balanced* if it can be extended to one which is balanced.

In a play of the associated *computation game*, O and P play alternately with O starting. Their moves are questions and answers — strictly justified questions and matching answers.

Conditions on Play More precisely, a play has to satisfy the following:

(1) The pattern of questions and answers is potentially balanced.

(2) An initial question is asked at the start of a play and cannot be repeated. The play stops when the initial answer is answered. We compute a *final* value once. However, in this non-linear setting, all other questions can be repeated subject to the next condition.

(3) A question other than the inital question can only be asked if its unique justifying question has already been asked. And questions must be justified explicitly; that is, the moves include the data as to which occurrence of the unique justifying question is being appealed to.

Views It is now helpful to refine our notations: we write "[" for O's question, "(" for P's question, ")" for O's answer and ")" for P's answer. P's view $\widehat{p}$ of a play p (which is a sequence of moves) is defined according to cases: we omit the cases of p being positions at which O is to move. Let p, q, r range over plays.

- If p is $q(r)$ where the distinguished P-question "(" and O-answer ")" are matching, then $\widehat{p}$ is $\widehat{q}()$.
- If p is $q(r[$ where the distinguished O-question "[" is explicitly justified by the occurence of the P-question "(", then $\widehat{p}$ is $\widehat{q}([$.

For example, a P-view may have the shape: $[()()([()()([()\cdots$. Note that whenever there is a pattern $([$, the O-question $[$ must be justified by the P-question $($. Also, there are no segments of the form $[\cdots]$, *i.e.* we forget about questions which we have already dealt with. There is a dual definition of O's view of a play p.

Innocent Strategies

An *innocent strategy* for P is a (partial) strategy in the sense of Abramsky and Jagadeesan, though in this case, there is no question of whether a strategy is winning or not. Innocent strategies are neither history-free, nor are they history-sensitive. Rather, each such strategy is determined by a partial function of a certain kind from P-views to P-moves (in exactly the same way that a history-free strategy is determined by a partial function from O-moves to P-moves).

We could already have defined the product $A \times B$ and function space $A \to B$ of two games (as determined by computational arenas). For $A \times B$, we simply take the obvious "disjoint sum" of the computational arenas as directed graphs. For $A \to B$, it is simplest to draw a picture as in Figure 1:

the initial moves are those of B and to the tree "below" each such initial move, we add immediately after the initial move a copy of the forest of questions for A. Note that the net effect is that moves of the new game $A \to B$ are defined in terms of those of A and B in a way similar to $A^{\perp} \propto B$ (A "perp" par B): a P-move (resp. O-move) in A becomes an O-move (resp. P-move) in $A \to B$. The following are consequent features of plays of $A \to B$:

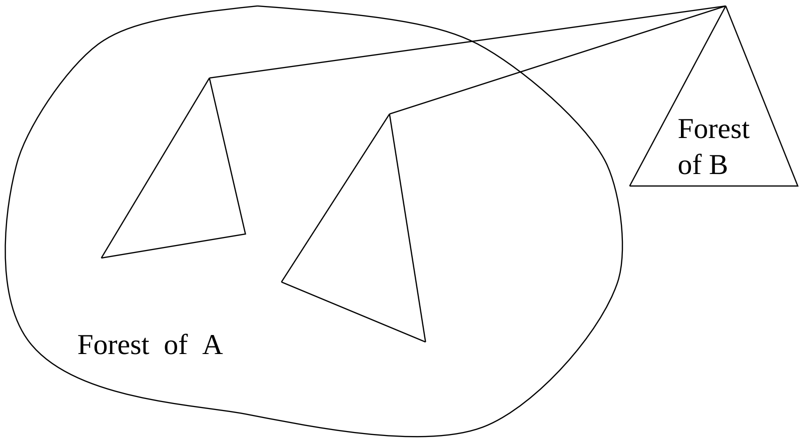

Figure 1: The forest of questions of $A \to B$.

- the projection of a play onto B is a play in B, while the projection onto A can be read as the interleaving of a finite number of plays in A,
- whenever a pairs of successive moves are in different components, it is P who has switched game.

In fact many plays in A may be involved in a play of $A \to B$, the analogue is with $(!A)^{\perp} \propto B$ as expected; but we suppress linear aspects in this treatment.

Intensionally Fully Abstract Model

From the above, we obtain a Cartesian closed category. The objects are computational arenas, and morphisms from A to B are innocent strategies for the game associated with the computational arena $A \to B$. Innocent strategies compose (effectively as in [AJ92]) to give innocent strategies. The collection of innocent strategies represented as partial functions for a (game associated with a) computational arena ordered by inclusion forms a Scott domain. The category gives rise to a model of PCF. Simple types are interpreted by computational arenas, and PCF-terms by innocent strategies.

- The ground type ι (natural numbers) is interpreted as the collection of P-strategies associated with the following computational arena: there is one initial question [, and possible answers $\{ 0, 1, 2, \cdots \}$. Note that the associated domain is just the standard flat CPO of natural numbers.
- The first order constants are straightforwardly interpreted as strategies.
- One can interpret a least fixpoint operator of type $(A \to A) \to A$ as the process of "iteratively unfolding $f : A \to A$" for any f.

Obviously, this model is neither order-extensional nor extensional ("left-add" and "right-add" have different denotations as strategies).

Definability A finite strategy (which is precisely a compact element of the domain associated with the game) is one which is given by a finite partial function from P-views to P-moves. We show that all finite strategies are definable in PCF by induction on the size of the function. As is well known, this is a sufficient condition for (intensional) full abstraction for PCF.

References

[AJ92] S. Abramsky and R. Jagadeesan. Games and full completeness for multiplicative linear logic. Technical Report DoC 92/94, Imperial College, 1992. 32 pp. To appear in *J. Symb. Logic*.

[BC82] G. Berry and P.-L. Curien. Sequential algorithms on concrete data structures. *Theoretical Computer Science*, 20:265–321, 1982.

[Kle78] S. C Kleene. Recursive functionals and quantifiers of finite types revisited i. In G. E. Sacks J. E. Fenstad, R. O. Gandy, editor, *Generalized Recursion Theory II, Proceedings of the 1977 Oslo Symposium*, pages 185–222. North-Holland, 1978.

[KP78] G. Kahn and G. D. Plotkin. Structures de données concrètes. Technical Report Report 338, INRIA-LABORIA, 1978.

[Mil77] R. Milner. Fully abstract models of typed lambda-calculus. *Theoretical Computer Science*, 4:1–22, 1977.

[Plo77] G. D. Plotkin. LCF as a programming language. *Theoretical Computer Science*, 5:223–255, 1977.

[Sto88] A. Stoughton. *Fully Abstract Models of Programming Languages*. Pitman, 1988. Research Notes in Theoretical Computer Science.