1 Introduction

The Full Abstraction Problem for PCF [23, 20, 7, 11] is one of the longest-standing problems in the semantics of programming languages. There is quite widespread agreement that it is one of the most difficult; there is much less agreement as to what exactly the problem is, or more particularly as to the precise criteria for a solution. The usual formulation is that one wants a “semantic characterization” of the fully abstract model (by which we mean the inequationally fully abstract order-extensional model, which Milner proved to be uniquely specified up to isomorphism by these properties [20]). The problem is to understand what should be meant by a “semantic characterization”.

Our view is that the essential content of the problem, what makes it important, is that it calls for a semantic characterization of sequential, functional computation at higher types. The phrase “sequential functional computation” deserves careful consideration. On the one hand, sequentiality refers to a computational process extended over time, not a mere function; on the other hand, we want to capture just those sequential computations in which the different parts or “modules” interact with each other in a purely functional fashion.

There have, to our knowledge, been just four models of PCF put forward as embodying some semantic analysis. Three are domain-theoretic: the “standard model” based on Scott-continuous functions [23]; Berry’s bidomains model based on stable functions [5]; and the Bucciarelli-Ehrhard model based on strongly stable functions [8]. The fourth is the Berry-Curien model based on sequential algorithms [6] (Cartwright and Felleisen’s model turns out to be equivalent to the sequential algorithms model [9, 12]). Of these, we can say that the standard model gives a good account of functional computation at higher types, but fails to capture sequentiality, while the sequential algorithms model gives a good analysis of sequential computation, but fails to capture functional behaviour. In each case, the failure can calibrated in terms of definability: the standard model includes parallel functions; the sequential algorithms model includes algorithms which compute “functionals” which are sensitive to non-functional aspects of the behaviour of their arguments. The bidomains model also contains non-sequential functions; while the strongly stable model, in the light of a recent result by Ehrhard [14], can be seen as the “extensional collapse” of the

* This research was supported by grants from UK SERC and ESPRIT Basic Research Action 6811 “CLICS II”
sequential algorithms model. In short, all these models are unsatisfactory because they contain “junk”. On the other side of the coin, we have Milner’s result that an order-extensional model is fully-abstract iff all its compact elements are definable.

**Intensional Full Abstraction**

This suggests that the key step towards solving the Full Abstraction problem for PCF is to capture PCF definability. This motivates the following definition. A model $\mathcal{M}$ (not necessarily extensional) is *intensionally fully abstract* if it is algebraic, and all its compact elements are definable in PCF. In support of this terminology, we have the fact that the fully abstract model can be obtained from an intensionally fully abstract model $\mathcal{M}$ in the following canonical fashion. Firstly, define a logical relation on $\mathcal{M}$ induced by the ordering on the ground types (which are assumed standard, i.e. isomorphic to the usual flat domains of natural numbers and booleans). Because of the definability properties of $\mathcal{M}$, this relation is a preorder at all types. In particular, it is reflexive at all types. This says that all elements of the model have extensional (functional) behaviour—there is no junk.

We can now apply Theorem 7.2.2 of [24] to conclude that $\mathcal{M}$ can be collapsed by a continuous homomorphism to the fully abstract model. In short, the fully abstract model is the extensional collapse of any intensionally fully abstract model. Moreover, note that the collapsing map is a homomorphism, and in particular preserves application. This contrasts sharply with “collapses” of the standard model to obtain the fully abstract model, as in the work of Mulumley [21] and Stoughton and Jung [19], which are only homomorphic on the “inductively reachable” subalgebra.

Thus we propose that a reasonable factorization of the full abstraction problem is to look for a semantic presentation of an intensionally fully abstract model, which embodies a semantic analysis of sequential functional computation. The construction of such a model is our first main result; it is outlined in Sections 2 and 3.

We have explained how the (order-extensional, inequationally) fully abstract model can be obtained from any intensionally fully abstract model by means of a general construction, described in [24]. However, this description of the fully abstract model leaves something to be desired. Firstly, just because the construction in [24] is very general, it is unlikely to yield any useful information about the fully abstract model. Secondly, it is not entirely syntax-free: it refers to the type structure of PCF.

What would the ideal form of description of the fully abstract model be? We suggest that it should comprise the specification of a cartesian closed category whose objects are certain cpo’s, given together with certain additional “intensional” structure, to be used to characterize sequentiality; and whose morphisms are continuous functions between these cpo’s—not all continuous functions, of course, but only the sequential ones, as determined by the intensional structure. The interpretation of PCF generated from this category should then be the fully