A Semantic Characterization of Full Abstraction
For Typed Lambda Calculi

Ketan Mulmuley

Computer Science Department,
Carnegie-Mellon University,
Pittsburgh, U.S.A.

Abstract
Full abstraction is a well known issue in denotational semantics. For a special case of typed lambda calculus, PCF, Plotkin showed that the classical model consisting of domains of continuous functions is not fully abstract. Milner constructed a fully abstract model of typed lambda calculus syntactically. However, its precise relationship with the classical model was not clear, and hence it remained open whether a fully abstract model can be constructed which is related to the classical model in a pleasant way. In this paper we show that a fully abstract, extensional model of typed lambda calculus can be constructed as a homomorphic retraction of the classical model.

1. Introduction

Full abstraction has been a well known issue for many years. It arises in the context of the Scott-Strachey denotational approach to semantics. In this approach each programming construct is given a denotation in a mathematical model. Of course, if the semantics is to be of any use at all, it must have the property that whenever two constructs have the same denotations they must behave identically in all programming contexts. However, the converse is difficult to ensure. This demands that two programming constructs have the same denotations whenever they behave identically in all programming contexts. The problem of full abstraction is to construct models having this property. For a special case of typed lambda calculus, PCF, it was shown by Plotkin that the classical model consisting of domains of continuous functions is not fully abstract. (See [2]). However, he was able to make the model fully abstract by adding to the language a new programming construct which provided a parallel or facility. On the other hand, Milner was able to obtain a syntactic fully abstract, extensional model of typed lambda calculus. (See [1]). Unfortunately, the precise relationship of Milner's model to the classical model was not clear.

Hence, it remained open if a fully abstract model of typed lambda calculus can be constructed which is related to the classical model in a nice way. Here we shall construct an extensional, fully abstract and algebraic model of typed lambda calculus which is a homomorphic retraction of the classical model. This provides one nice semantic characterization of full abstraction for typed lambda calculi.

2. Typed Lambda Calculus

We assume some familiarity with typed lambda calculus and combinators.

Assume we are given a set of ground types. We shall let \( \kappa, \kappa_1, \kappa_2 \ldots \) range over these ground types. From the given ground types we define the set of functional types, \( \Gamma \):

1. \( \kappa \in \Gamma \), if \( \kappa \) is a given ground type,
2. \( (\alpha \rightarrow \beta) \in \Gamma \) whenever \( \alpha, \beta \in \Gamma \).

We shall omit the braces {} whenever possible. In this case the association of {} is taken as from right to left. Thus \( \alpha \rightarrow \beta \rightarrow \sigma \) denotes \( \alpha \rightarrow (\beta \rightarrow \sigma) \). We write:

\[ (\sigma_1, \ldots, \sigma_n, \tau) = \sigma_1 \rightarrow \cdots \rightarrow \sigma_n \rightarrow \tau \quad \text{for } n \geq 0. \]

Note that each type can be written in the form \( (\sigma_1, \ldots, \sigma_n, \kappa) \). A type is said to be of first order if it is of the form \( (\kappa_1, \ldots, \kappa_n, \kappa) \). We assume that we are given, for each ground type \( \kappa \), a set basic constants \( B^\kappa \) and for each first order type \( \tau \) a set of basic function constants \( F^\tau \). A family \( \{L^\tau\} \), where \( L^\tau \) is intended to be a set of terms of type \( \tau \), is defined to be the family of the smallest sets \( L^\tau \) satisfying the conditions:

1. \( B^\kappa \subseteq L^\kappa \), for all ground types \( \kappa \).
2. \( F^\tau \subseteq L^\tau \), for all first order types \( \tau \).
3. \( s \in L^\tau \) where \( S \) is the usual closed combinator \( \lambda z y z (x z)(y z) \) of type \( \tau \).
4. \( K \in L^\tau \) where \( K \) is the usual closed combinator \( \lambda z y z \) of type \( \tau \).
5. \( Y \in L^\tau \) where \( Y \), which is intended to be a fixed point combinator, is of type \( \tau \).
6. \( (t s) \in L^\delta \) whenever \( t \in L^\alpha \rightarrow \delta \) and \( s \in L^\alpha \).

We shall omit the braces {} whenever possible, the
associating then being from left to right. Thus we shall write \( t \downarrow s \) instead of \( (t \downarrow s) \).

Define \( \mathcal{L} = \bigcup \mathcal{L}^n, \mathcal{F} = \bigcup \mathcal{F}^n \), where \( \tau \) ranges over the first order types, and \( \mathcal{B} = \bigcup \mathcal{B}^n \).

We now give an operational semantics to the above language in terms of reduction rules. Each reduction rule intuitively specifies one step of evaluation. Assume that we are given, for every first order function constant \( f : \{ \kappa_1, \ldots, \kappa_n, \kappa \} \), a set of reduction rules of the form:

\[
f_{t_1} \ldots t_n ^{\downarrow} b_t,
\]

where \( t_j \in \mathcal{L}^n \) and \( b \in \mathcal{B} \). We define a reduction relation \( \rightarrow \) on terms as follows:

1. \( f_{t_1} \ldots t_n ^{\downarrow} b \), where \( f \in \mathcal{F} \).
2. \( (y g) ^{\downarrow} g (y g) \), where \( g \in \mathcal{L}^n \).
3. \( r s t ^{\downarrow} (r t)(s t) \).
4. \( k r s ^{\downarrow} r \).
5. \( r s ^{\downarrow} r' s' \).
6. \( r s ^{\downarrow} r' s' \).

We shall denote by \( \rightarrow^{*} \) the transitive reflexive closure of \( \rightarrow \). Even though \( \rightarrow^{*} \) is not monogenic, it can be shown that the order of evaluation does not matter if \( \rightarrow^{*} \) is 'reasonable'.

We turn next to the denotational semantics of \( \mathcal{L} \). Assume that we are given, for each ground type \( \kappa \), a ground domain \( \mathcal{D}^\kappa \) and a type-respecting ground semantics \( G : \mathcal{B} \rightarrow \bigcup \mathcal{D}^\kappa \) such that all the finite elements of the ground domains are definable by basic constants, i.e., for each finite \( d \in \mathcal{D}^\kappa \), there exists a basic constant \( b \in \mathcal{B}^\kappa \) such that \( d = G b \). We also assume that for each \( f \in \mathcal{F} \), \( f = \kappa_1, \ldots, \kappa_n, \kappa \), we are given a first order continuous function \( H^f \) such that for all \( b_1, \ldots, b_n \mathcal{B} \):

\[
( H^f ) (G b_1) \ldots (G b_n) = ( G b ) \text{ if } f b_1 \ldots b_n ^{\downarrow} b.
\]

A model of \( \mathcal{L} \), \( M = ( \mathcal{D}^\kappa, \mathcal{A} ) \), consists of:

1. a cpo \( \mathcal{D}^\kappa \) for each type \( \kappa \) such that for each ground type \( \kappa \), \( \mathcal{D}^\kappa \) is isomorphic to the given ground domain.
2. a continuous application function \( \cdot : \mathcal{D}^\alpha \times \mathcal{D}^\beta \rightarrow \mathcal{D}^\gamma \) for all types \( \alpha \) and \( \beta \).
3. a type preserving map \( A : \mathcal{L} \rightarrow \bigcup \mathcal{D}^\kappa \) such that:
   a) \( A \) is a homomorphism, i.e., \( A ( t s ) = ( A t ) \cdot ( A s ) \), for all \( t, s \in \mathcal{L} \),
   b) \( A \mathcal{B} = \mathcal{B} \), for all \( b \in \mathcal{B} \),
   c) \( A f = H^f \) if \( f \in \mathcal{F} \).

Again we shall omit - whenever possible, the association being assumed to be from left to right. If \( b \in \mathcal{B} \) and \( f \in \mathcal{F} \), we shall ambiguously use the symbol \( b \) to denote \( G b \) and the symbol \( f \) to denote \( H^f \). Whether a symbol \( b \) or \( f \) is playing a syntactic role or a semantic one should be clear from the context.

If a model for \( \mathcal{L} \) is to be of any value, it should be faithful to its operational semantics. Let us define a type-respecting map \( O : \bigcup \mathcal{L}^\kappa \rightarrow \bigcup \mathcal{D}^\kappa \) as:

\[
O t = \begin{cases} 
G b & \text{if } t ^{\rightarrow} b \\
\bot & \text{otherwise}.
\end{cases}
\]

A model, \( M = ( \mathcal{D}^\kappa, \mathcal{A} ) \), is called adequate (or faithful) if, for all ground terms \( t \),

\[
A t = O t.
\]

In this case we say that \( O \) and \( A \) are semantically equivalent.

One can now define precisely, what it means to say that one term is operationally weaker than the other.

We say \( t \sqsubseteq s \), where \( t, s \in \mathcal{L}^\kappa \), if for all \( t_j \in \mathcal{L}^n \),

\[
O ( t_{t_1} \ldots t_n ) \sqsubseteq O ( s_{t_1} \ldots t_n ).
\]

This definition differs from the usual definition found in literature which is given in terms of 'contexts'. We shall see later that the two definitions are equivalent.

A model, \( M = ( \mathcal{D}^\kappa, \mathcal{A} ) \), is called fully abstract if for all terms \( t, s \in \mathcal{L}^\kappa \):

\[
A t \sqsubseteq A s \iff t \sqsubseteq s.
\]

A simple model for \( \mathcal{L} \) is the classical model, \( M = ( \mathcal{D}^\kappa, \mathcal{A} ) \), where the ground domains are the given ones and domains at higher types are inductively defined simply as follows: \( \mathcal{D}^\alpha = \mathcal{D}^\beta \rightarrow \mathcal{D}^\gamma \), where \( \mathcal{D}^\alpha \rightarrow \mathcal{D}^\beta \) is the domain of continuous functions from \( \mathcal{D}^\alpha \) to \( \mathcal{D}^\beta \). The application function \( \cdot \) is the usual function application and \( A \) is defined as follows:

1. \( A ( b ) = b \), for \( b \in \mathcal{B} \)
2. \( A ( f ) = f \), for \( f \in \mathcal{F} \)
3. \( A ( S ) = \lambda z y ( z z ) ( y z ) \)
4. \( A ( K ) = \lambda x y. x \)
5. \( A ( Y ) = \bigcup_{n=0}^\infty \lambda f. f^n ( \lambda ) \)
6. \( A ( t s ) = ( A t ) ( A s ) \).

It can be shown that \( M \) is an adequate model of \( \mathcal{L} \) (for a similar proof see [2]). However, it was shown by Plotkin that \( M \) is not fully abstract for a special case of \( \mathcal{L} \) PCF. But in this case Plotkin made \( M \) fully abstract by adding an extra parallel or facility to \( \mathcal{L} \). On the other hand, Milner demonstrated the existence of fully abstract model for \( \mathcal{L} \) in [1]. Unfortunately, the precise relationship of his model with \( M \) is not clear. In this paper we shall...
show that it is possible to obtain a fully abstract, extensional model of typed lambda calculus as a homomorphic retraction of $M$.

It should be made clear at the outset that we dealing with general typed lambda calculi as in [1]. In particular, we do not assume that $\mathcal{L}$ has any additional properties like sequentiality (see [3]). Our method will provide a semantic characterization of full abstraction for any general typed lambda calculus, but we shall leave it open whether a better and direct semantic characterization can be found when the language $\mathcal{L}$ is known to be sequential.

Henceforth $M$ will refer to the above-mentioned classical model $(D', \cdot, A)$. We shall also assume that $D'$ is $\omega$-algebraic complete lattices. Why we require $D'$ to be consistent complete cpos will become clear in the next section.

3. Inclusive Predicates

The collapsing of $M$ onto a fully abstract model is achieved through some inductively defined inclusive predicates (see [4], [7], [5]). For each type $\tau$ we define an inclusive predicate $\Theta^\tau \subseteq D' \times \mathcal{L}$ as follows:

1. For a ground type $\kappa$,
   \[
   \Theta^\kappa = \{(d, t) \mid d \subseteq O(t)\}.
   \]

2. For a type $\tau = \alpha \rightarrow \beta$,
   \[
   \Theta^\tau = \{(d, t) \mid \forall (c, s) \in \Theta^\kappa. (dc, ts) \in \Theta^\beta\}.
   \]

It is easy to show that $\Theta$'s can be defined equivalently as follows:

1. For a ground type $\kappa$,
   \[
   \Theta^\kappa = \{(d, t) \mid d \subseteq \overline{O(t)}\}.
   \]

2. For a type $\tau = (\sigma_1, \ldots, \sigma_n, \kappa)$
   \[
   \Theta^\tau = \{(d, t) \mid \forall (d_1, t_1) \in \Theta^{\sigma_1} \ldots d_n \subseteq \overline{O(t_1 \ldots t_n)}\}.
   \]

We shall use any of the two equivalent formulations as convenient. It is easy to show that all $\Theta$'s are directed complete. Note that $(d, e) \in \Theta$ can be taken as saying $d$ is weaker than $e$ in some sense. Hence $\Theta^\tau$ can be used to define a natural quasiorder $\sqsubseteq_{\tau}$ on $D'$. We say

\[
\begin{align*}
d_1 &\sqsubseteq_{\tau} d_2 \text{ iff for all } t, (d_1, t) \in \Theta^\tau \text{ implies } (d_1, t) \subseteq \Theta^\tau.
\end{align*}
\]

Let $\simeq$ be the induced equivalence relation. The equivalence class of $d \in D'$ will be denoted by $[d]^\tau$.

In this paper we adopt the convention of dropping the type subscripts and superscripts whenever no ambiguity arises. Thus we shall often write $\subseteq_{\tau}$, $\simeq$, or $\Theta$ instead of $\subseteq_{\tau}$, $\simeq$, or $\Theta^\tau$. The convention also applies to any definitions we introduce in future.

The inclusive relation $\Theta$ and the induced equivalence relation $\simeq$ have many nice properties. For example,

\[
\text{if } t \simeq s \text{ then } (d, t) \in \Theta \text{ implies } (d, s) \in \Theta.
\]

Secondly, $\subseteq$ is a refinement of $\subseteq_{\tau}$, i.e.,

\[
d_1 \subseteq d_2 \text{ implies } d_1 \subseteq_{\tau} d_2.
\]

As the $D'$'s are assumed to be complete lattices it is easy to see that whenever $d_1 \subseteq [d]$ and $d_2 \subseteq [d]$ then $d_1 \sqcup d_2 \subseteq [d]$. (This where we need the lattice property of $D'$.) This means that $[d]$ is directed. By the directed completeness of the $\Theta$'s it follows that each $[d]$ has a maximum element max$[d] = \bigcup[d]$.

Let us define a monotonic function, $F^\tau$, on the finite elements of $D'$:

\[
F^\tau(d) = \text{max}[d], \text{ for each finite } d \in D'.\tag{2}
\]

Let $Q'$ be the unique continuous extension of $F^\tau$. ($Q'$ can be shown to be the closure of the quotient space of $D'$, hence the mnemonic name $Q$). Define $A^\tau$ as follows:

1. $A^\tau(b) = Q(b)$ where $b$ is a basic ground type constant.
2. $A^\tau(f) = Q(f)$ where $f$ is a basic first order function constant.
3. $A^\tau(s) = Q(\lambda x, y, z. (xz)[yz])$
4. $A^\tau(K) = Q(\lambda x, y, z) xz$
5. $A^\tau(Y) = Q\left(\bigcup_{n=0}^{\infty} A^\tau f^n(\bot)\right)$
6. $A^\tau(rs) = (A^\tau r)A^\tau s$ where $r$ and $s$ are of the appropriate types.

Now we can state the main result of this paper:

**Theorem 3.1**: $M^Q = (Q', \cdot, A^Q)$, where the application $\cdot$ is just the restriction of the application function of $M$, is a fully abstract, extensional, algebraic, $\beta$-model of $\mathcal{L}$. Moreover $Y$ has the standard interpretation in $M^Q$ and the following diagram commutes.

\[
\begin{array}{c}
\mathcal{L} \xrightarrow{A} M
\
\downarrow Q
\
M^Q
\end{array}
\]

As it stand, this theorem is highly nonobvious. It is not even clear that the application in $M^Q$ is well defined! That is to say, why should it be the case that whenever $d \in Q^\kappa$ and $c \in Q^\tau$ then $dc \in Q^\tau$? And, of course, we have to show that it is indeed fully abstract, extensional, algebraic, and homomorphic retraction.
Rest of the paper is devoted to proving this theorem. We cannot prove this theorem directly. What we shall do is to consider a sequence of subsets of \( L, L_1 \subseteq L_2 \subseteq \ldots \). For each \( L_i \) we construct a fully abstract, extensional model \( M_\alpha^i \). The model \( M_\alpha^i \) is then shown to be a limit of the sequence \( M_\alpha^{i-1}, M_\alpha^{i-2}, \ldots \). Of course, the success of the approach depends on choosing each \( L_i \) wisely. Before we do that, we need to extend the notion of a model.

Let \( K \) be a subset of \( L \) which is closed under application, i.e., whenever \( t \in K \) and \( s \in K \), then \( ts \in K \).

A \( K \)-model, \( N = (E^\alpha, B) \), consists of

1. a cpo \( E^\alpha \) for each type \( \tau \) such that, for each ground type \( \kappa \), \( E^\alpha \) is a subdomain of the given ground domain \( D^\kappa \).
2. a continuous application function \( \cdot : E^\alpha \times E^\alpha \rightarrow E^\beta \) for all types \( \alpha, \beta \).
3. a type preserving map \( B : K \rightarrow \bigcup E^\tau \) which is a homomorphism:

\[
B(t s) = B(t) B(s) \quad \text{for } t, s \in K.
\]

It is clear that a model for \( L \) as defined in the previous section is just an \( L \)-model.

A \( K \)-model, \( N = (E^\alpha, B) \), is said to be adequate if \( B(t) = O(t) \) for every ground term \( t \in K \).

Let \( K^* = L \cap K \). Given \( t, s \in K^{(\kappa_1, \ldots, \kappa_n)} \), we say \( t \triangleleft s \), if for all \( t_i \in K^\kappa_i \),

\[
O(t_1 \ldots t_n) \subseteq O(s_1 \ldots s_n).
\]

We say that a \( K \)-model, \( N = (E^\alpha, B) \), is fully abstract if for all \( t, s \in K^* :\

\[
B(t) \subseteq B(s) \iff t \triangleleft s.
\]

We can now address the question of selecting the sequence of subsets of \( L, L_1 \subseteq L_2 \subseteq \ldots \). Suppose that we are given, for each ground type \( \kappa \), a monotone sequence of finite projections, \( \phi_i^\kappa \triangleleft \phi_{i+1}^\kappa \) such that \( \bigcup_i \phi_i^\kappa = \mathbb{I} \), where \( \mathbb{I} \) is the identity function on \( D^\kappa \). Finiteness of \( \phi_i^\kappa \) implies that \( \phi_i^\kappa \) is an \( E^\alpha \)-term, \( \phi_i^\kappa \) is finite, and moreover each \( \phi_i^\kappa \) is a finite element of \( D^\kappa \). For every higher type \( \tau \rightarrow \beta \) we inductively define \( \phi_i^\tau \):

\[
\phi_i^\tau = \lambda f : \tau. \phi_i^\alpha \circ f \circ \phi_i^\beta.
\]

We shall denote \( |\phi_i^\kappa| \) by \( D_i^\kappa \). It follows that each \( \phi_i^\kappa \) is finite--hence \( D_i^\kappa \) is finite--and also that \( \bigcup_i \phi_i^\kappa = \mathbb{I} \). Also \( \bigcup_i \phi_i^\kappa \) is isomorphic to the function space \( D_i^\kappa \rightarrow D_i^\kappa \).

We make an important assumption: We assume that each \( \phi_i^\kappa \) is definable in \( L \). This means that there exists a term \( \phi_i^\kappa \in L \) such that \( A(\phi_i^\kappa) = \phi_i^\kappa \). It follows by induction that \( \phi_i^\kappa \) is definable for every type \( \tau \); we let, for \( \tau = \alpha \rightarrow \beta \),

\[
\Phi_i^\tau = \lambda f : \tau. \Phi_i^\alpha \circ f \circ \Phi_i^\beta.
\]

(Strictly speaking \( \Phi_i^\tau \) is an S-K combinator equivalent to the right-hand side of the above equation). Then it is easily seen that \( A(\Phi_i^\tau) = \phi_i^\tau \).

For each term \( t : \tau \in L \), we define its \( i \)-th syntactic approximant \( [t]_i \in L \) as:

\[
[t]_i = \Phi_i^\tau t.
\]

Let \( L_i \) be the smallest set closed under application which contains \( [t]_i \) for each \( t \in L \). Then

\[
A(s) \in D_i^\tau \text{ for every } s : \tau \in L_i.
\]

and moreover

\[
A(t) = \bigcup_0^\infty A([t]_i) \text{ for every } t \in L.
\]

Let \( M_i = (D_i^\tau, A_i) \), where \( A_i : L_i \rightarrow D_i^\tau \) is simply the restriction of \( A \) to \( L_i \). Then \( M_i \) is an adequate, extensional \( L_i \)-model. We shall collapse \( M_i \) onto a fully abstract, extensional model \( M_i^\alpha \). But before that let us investigate the relationship between the operational preorder w.r.t \( L \) and the operational preorder w.r.t \( L_i \).

Lemma 3.2: For all \( i \),

if \( t, s \in L \) then \( t \not\triangleleft_s s \) implies \( [t]_i \not\triangleleft_s [s]_i \).

Proof: Suppose \( t, s \in L \) and \( t \not\triangleleft_s s \). Then for all \( t_1, \ldots, t_n \in L_i \) we have, as \( L_i \subseteq L \),

\[
O(t_1 \ldots t_n) \subseteq O(s_1 \ldots s_n).
\]

Hence,

\[
\phi_i^\tau(O(t_1 \ldots t_n)) \subseteq \phi_i^\tau(O(s_1 \ldots s_n)) \quad (3)
\]

Note that, because for all \( j, t_j \in L_i \), we know that \( A(t_j) \in D_i^\tau \) which implies \( \phi_i^\tau \circ \sigma \in D_i^\tau \). Hence, remembering that \( O \) and \( A \) are semantically equivalent,

\[
\phi_i^\tau(O(t_1 \ldots t_n)) = \phi_i^\tau(A(t_1 \ldots t_n)) = \phi_i^\tau((A(t_1)) \ldots (A(t_n))) = \phi_i^\tau((A(t_1) \circ \cdots \circ (A(t_n))) = (\phi_i^\tau \circ A(t_1)) \ldots (A(t_n)) = A([t_1]_i \ldots [t_n]_i) = O([s]_1 \ldots [s]_n).
\]

And similarly,

\[
\phi_i^\tau(O(s_1 \ldots s_n)) = O([s]_1 \ldots [s]_n).
\]

From (3) we conclude that:

\[
O([t]_i t_1 \ldots t_n) \subseteq O([s]_i s_1 \ldots s_n) \text{ for all } t_j \in L_i.
\]

Thus indeed \( [t]_i \not\triangleleft_s [s]_i \).
4. A Finite Approximate Model

In this section we describe how one can construct, for each $L_i$, a model $M_i^Q$ which will be a finite approximation to the final fully abstract $L_i$-model, $M_i^Q$.

For each $d \in D_i^r$ define $[d]_i^r = D_i^r \cap [d]^r$. Then it is easy to see that $[d]_i^r$ also has a maximum; $\max[d]_i^r = \phi_i^r(\max[d]^r)$. Thus we have a map $\max_i^r : D_i^r \to D_i^r$.

$$\max_i^r : d \mapsto \max[d]_i^r,$$

which is monotonic and hence trivially continuous as $D_i^r$ is finite. Let

$$Q_i^r = \max_i^r \circ \phi_i^r,$$

Then, for each $d \in D_i^r$, $Q_i^r \circ Q_i^r(d) = Q_i^r(\max[d]_i^r) = \max[\max[d]_i^r]_i = \max[d]_i^r = Q_i^r(d)$. Hence $Q_i^r$ is a retract of $D_i^r$ and also of $D_i$. Also, for all $d \in D_i^r$, $Q_i(d)$ and $d$ belong to the same equivalence class. Hence

for all $d \in D_i^r$, $Q_i(d) \simeq d$.

Let us define $M_i^Q = (Q_i^r, \ldots , A_i^Q)$, where $\phi_i$ is just the restriction of the application function in $M_i$ (or equivalently $M$), and $A_i^Q : L_i \to \bigcup Q_i^r$ is defined as (dropping the type superscripts):

$$A_i^Q = Q_i \circ A_i.$$

Obviously,

$$A_i \subseteq A_i^Q,$$

and

$$A_i^Q(t) \simeq A_i(t)$$

for all $t \in L_i$.

It will turn out that $M_i^Q$ is a fully abstract, extensional $L_i$-model. Of course, a lot of work has to be done in order to prove this.

First we ask: is the application in $M_i^Q$ well defined? That is, if $d \in Q_i^r$ and $c \in Q_i^r$ then does $d \circ c \in Q_i^r$ always? Before we address this question let us prove a general lemma.

**Lemma 4.1:** Suppose we are given $d_1, d_2 : \tau = \alpha \to \beta$. Then if, for all $c : \alpha$, there exists a $c' : c \subseteq c$ such that $d_1 c \subseteq d_2 c'$ then $d_1 \subseteq d_2$.

**Proof:** We have to show that given any $(d_1, t) \in \Theta^r$, $(d_2, t) \in \Theta^r$.

Let $(d_2, t)$ be some arbitrary element in $\Theta^r$. For any $(c, s) \in \Theta^\tau$ we know that $(c, s) \in \Theta^r$ as $c' \subseteq c$. Hence, because $(d_1, t) \in \Theta^r$, $(d_2, c', t) \in \Theta^r$. This implies, as $d_2 c' \subseteq d_2 c$, that $(d_2, c, t) \in \Theta^r$. Thus for every $(c, s) \in \Theta^\tau$ we have $(d_2 c', t) \in \Theta^r$, which means $(d_1, t) \in \Theta^r$. This concludes the proof.

**Corollary 4.2:** Let $d_1, d_2 : \tau = \alpha \to \beta$ be given such that

1. $d_1 = (c \Rightarrow b)$ for some finite $c : \alpha$ and $b : \beta$, where $\Rightarrow$ indicates the usual step function, i.e.

$$d_1 a = b$$

if $c \subseteq a$

$$= \bot$$

otherwise.

2. there exists $c' \subseteq c$ such that $b \subseteq d_2 c'$.

Then $d_1 \subseteq d_2$.

**Proof:** For every $a : \alpha$, we show that there exists $a' \subseteq a$ such that $d_1 a = d_2 a'$. The result then follows from the above lemma. Consider two cases.

1. $c \subseteq a$: Then $c \subseteq a$, as $\subseteq$ is a refinement of $\subseteq$. Now $d_1 a = b \subseteq d_2 c'$ and $c' \subseteq c \subseteq a$, hence we can let $a' = c'$.

2. $c \not\subseteq a$: Then $d_1 a = \bot$. Hence we can let $a' = \bot$.

Now we can show that application in $M_i^Q$ is well defined. Let $d \in Q_i^r$, where $\tau = \alpha \to \beta$, and $c \in Q_i^r$. We want to show that $d \circ c \in Q_i^r$. Let $b = \max(d c)$, then this amounts to showing that $b = dc$. Define $a : D_i^r$ as

$$a = (c \Rightarrow b)$$

As $b = \max(d c)$, trivially $b \subseteq dc$. Now from the preceding corollary it immediately follows that $a \subseteq d$. Hence $a \subseteq \max(d c) \subseteq \max[d]_i^r = d$. This implies that $b = ac \subseteq dc$. On the other hand $dc \subseteq \max(d c)_i = b$. Thus $b = dc$ and we have shown that the application in $M_i^Q$ is well defined.

What can we say about the extensionality of $M_i^Q$? Note that this does not follow from the extensionality of $M_i$. The extensionality of $M_i$ says: if $d_1, d_2 \in D_i^r$, where $\tau = \alpha \to \beta$, then $d_1 \subseteq d_2$ whenever

$$d_1 c \subseteq d_2 c$$

for all $c : D_i^\tau$.

On the other hand the extensionality of $M_i^Q$ says: if $d_1, d_2 \in Q_i^r$, then $d_1 \subseteq d_2$ whenever

$$d_1 c \subseteq d_2 c$$

for all $c : Q_i^\tau$,

which is a much stronger statement as $Q_i^\tau$ is just a subset of $D_i^\tau$.

We can prove extensionality as follows. Let $d_1, d_2 : Q_i^r$ be such that $d_1 c \subseteq d_2 c$ for all $c : Q_i^\tau$. Then for all $a : D_i^\tau$ we have

$$d_1 a = d_1 (d_2 a)$$

and $d_2 a \subseteq d_2 (d_2 a)$. Thus

$$d_1 a = d_2 (Q_i^r(a))$$

for all $a : D_i^\tau$ by the assumption, as $Q_i^r(a) \in Q_i^r$.

Remembering that $\subseteq$ is a refinement of $\subseteq$, this implies $d_1 a \subseteq d_2 (Q_i^r(a))$ for all $a : D_i^\tau$. Also, as $a \in D_i^r$, $Q_i^r(a) \subseteq a$. Now we immediately conclude from Lemma 4.1 that $d_1 \subseteq d_2$. Hence

$$d_1 = \max(d_1) \subseteq \max(d_2) = d_2,$$

which proves the extensionality of $M_i^Q$. Algebraicity of $M_i^Q$ follows trivially because $|Q_i^r|$ is finite for all $i$. 

283
Before we turn to the full abstractness of $M^Q$ let us prove some lemmas.

Lemma 4.3: For all $t : \tau$, $(At, t) \in \Theta'$.

Proof:

1. If $t$ is a basic ground constant or a first order function constant then it is obvious.
2. $t = Y$: Let the type of $Y$ be $\tau = \sigma \rightarrow \alpha$, where $\sigma = \alpha \rightarrow \alpha$. We show by induction that

   for all $n$, $(y_n, Y) \in \Theta'$ where $y_n = \lambda f. f^n 1$,

   Then the result follows from the directed completeness of $\Theta'$.

   The basis is clear as $(y_0, Y) = (\bot, Y) \in \Theta'$.

   Assume, as the induction hypothesis that $(y_n, Y) \in \Theta'$.

   We have to show that $(y_{n+1}, Y) \in \Theta'$. For this it suffices to show that for all $(c, s) \in \Theta'$,

   $(y_{n+1}, Y) \in \Theta'$.

   Let $(c, s)$ be an arbitrary element of $\Theta'$. As $(y_n, Y) \in \Theta'$, $(c^{\bot}, Y) \in \Theta'$. Hence, as $(c, s) \in \Theta'$, $(c^{\bot}, y_s) \in \Theta'$. But, as $y_s \Downarrow Y$, this means that $(c^{\bot}, Y) \in \Theta'$.

   Thus, for all $(c, s) \in \Theta'$,

   $(y_{n+1}, Y) = (c^{\bot}, y_s) \in \Theta'$.

   Hence $(y_{n+1}, Y) \in \Theta'$. This concludes the proof of this case.

3. $t = K$ or $S$: this case is easier than the previous one, and hence is deleted.

4. $t : \beta = \alpha s$: where $r : \sigma \rightarrow \beta$, and $s : \alpha$: By the induction hypothesis $(Ar, r) \in \Theta'^{-\beta}$ and $(As, s) \in \Theta'^{-\alpha}$. Hence $(At, t) = ((Ar, r)(As, s)) \in \Theta'$.

Corollary 4.4: For all $t : \tau \in \mathcal{L}_i$, $(A^i(t), t) \in \Theta'$.

Proof: Let $t \in \mathcal{L}_i$. By the above lemma, $(At, t) = \Theta'$. Since $A^i(t) \simeq A_i(t)$, this means $(A^i(t), t) \in \Theta'$.

Lemma 4.5: For $t, s : \tau$, $t \sqsubseteq s$ iff $(At, s) \in \Theta'$.

Proof:

$\Leftarrow$: Suppose $t \sqsubseteq s$. By Lemma 4.3 $(At, t) \in \Theta'$. Hence, as $t \sqsubseteq s$, $(At, s) \in \Theta'$.

$\Rightarrow$: Suppose $(At, s) \in \Theta'$. Let $r = (\sigma_1, \ldots, \sigma_n, \alpha)$. Then for all $t_j : \sigma_j$

$$O(t_1, \ldots, t_n) = \Theta(At_1, \ldots, t_n) \subseteq \Theta(A^i(t_1), \ldots, t_n),$$

since $(At, s) \in \Theta'$ by the assumption, and $(At_j, t_j) \in \Theta^i$ by Lemma 4.3. Thus $t \sqsubseteq s$.

Lemma 4.6: $(d, t) \in \Theta'$ iff $d \sqsubseteq At$.

Proof:

$\Leftarrow$: Suppose $d \sqsubseteq At$. By Lemma 4.3 $(At, t) \in \Theta'$. Hence, as $d \sqsubseteq At$, $(d, t) \in \Theta'$.

$\Rightarrow$: Suppose $(d, t) \in \Theta'$. Then for any $(At, s) \in \Theta'$ we conclude from the preceding lemma that $t \sqsubseteq s$ and hence, as $(d, t) \in \Theta'$, $(d, s) \in \Theta'$.

This means $d \sqsubseteq At$.

Corollary 4.7: If $d \in D_1'$ and $t \in L_i$ then $(d, t) \in \Theta'$ implies $Q_i d \sqsubseteq A^Q_i(t)$.

Proof: By the above lemma $d \sqsubseteq At = A_i(t)$. This means

$$Q_i(d) = max[d] \sqsubseteq max[A_i(t)] = Q_i(A_i(t)) = A^Q_i(t).$$

Corollary 4.8: $t \sqsubseteq s$ iff $As \subseteq As$.

Proof:

$$t \sqsubseteq s \text{ iff } (At, s) \in \Theta \text{ by Lemma 4.5} \quad \text{iff } At \sqsubseteq As \text{ by Lemma 4.6}.$$

Now we are in a position to prove the full abstractness of $M^Q$. Let $t, s \in \mathcal{L}_i$ be such that $t \sqsubseteq s$. Then by a proof similar to that of Lemma 3.2 one can show that $t \sqsubseteq s$. Hence by the Corollary 4.8 $A_i(t) = At \sqsubseteq As = A_i(s)$. This means

$$A^Q_i(t) = Q_i A_i(t) = max[A_i(t)] \sqsubseteq max[A_i(s)] = Q_i A_i(s) = A^Q_i(s).$$

Thus indeed $A^Q_i(t) \sqsubseteq A^Q_i(s)$.

On the other hand if $A^Q_i(t) \sqsubseteq A^Q_i(s)$, where $t, s \in \mathcal{L}_i$, then

$$A(t) = A_i(t) \sqsubseteq A^Q_i(t) \sqsubseteq A^Q_i(s) \sqsubseteq A_s = A(s).$$

Then by Corollary 4.8, $t \sqsubseteq s$. This in turn implies, by Lemma 3.2, $[t] \sqsubseteq [s]$. But as $t, s \in \mathcal{L}_i$, we know that $t \simeq [t]$, and $s \simeq [s]$. Thus indeed $t \sqsubseteq s$.

We have then proved that for all $t, s \in \mathcal{L}_i$, $t \sqsubseteq s$ iff $A^Q_i(t) \sqsubseteq A^Q_i(s)$. That is, $M^Q$ is fully abstract.

Finally we need to show that $A^Q_i$ is a homomorphism, i.e., $A^Q_i(rs) = \langle A^Q_i(r) A^Q_i(s) \rangle$, for all $r, s \in \mathcal{L}_i$.

Let $r, s \in \mathcal{L}_i$.

Then

$$A^Q_i(rs) = Q_i \langle A_i(rs) \rangle \sqsubseteq Q_i \langle A_i(r) A_i(s) \rangle \sqsubseteq Q_i \langle A^Q_i(r) A^Q_i(s) \rangle$$

as $A_i \sqsubseteq A^Q_i$.\hspace{1cm}[4]
But note that, as $A^Q_{i}(r), A^Q_{j}(s) \in Q_i$, we conclude, because the application in $M^Q_i$ is well defined, that $(A^Q_{i}(r))(A^Q_{j}(s)) \in Q_i$. This means $Q_i ((A^Q_{i}(r))(A^Q_{j}(s))) = (A^Q_{i}(r))(A^Q_{j}(s))$.

Thus by (4) we conclude that

$$A^Q_{i}(rs) \subseteq (A^Q_{i})(A^Q_{j}) rs.$$  

To prove the other inequality, it suffices to prove that $((A^Q_{i})(A^Q_{j}) rs) \subseteq A^Q_{i}(rs)$. Because then,

$$A^Q_{i}(rs) = Q_i (((A^Q_{i})(A^Q_{j}) rs))$$

as above

by Corollary 4.7.

It is easy to prove that $((A^Q_{i})(A^Q_{j}) rs) \subseteq \Theta$. By Corollary 4.4, $(A^Q_{i}(r), r), (A^Q_{j}(s), s) \subseteq \Theta$. Hence

$$(A^Q_{i})(A^Q_{j}) rs \subseteq \Theta.$$  

We have proved that $A^Q_{i}$ is a homomorphism.

Let us summarize what we have proved in this section. $M^Q_i$ is a fully abstract, extensional, algebraic $\mathcal{L}_i$-model. Further $Q_i$ is a homomorphic retract. i.e., the following diagram commutes.

$$\xymatrix{ \mathcal{L}_i \ar[r]^{A^Q_{i}} & M^Q_i \ar[d]^{Q_i} \ar[r] & M^Q_j \ar[d]^{Q_j} \ar[r] & \ldots}

Of course it is obvious that the above diagram commutes because that is how we defined $A^Q_{i}$. What is not obvious is that $A^Q_{i}$ defined in this fashion is actually a homomorphism. It is possible to take another approach; we can define $A^Q_{i}$ as a homomorphism and then prove that the above diagram commutes. Which approach one takes is a matter of taste but the end result is the same anyway.

It is also interesting to note that not all the finite elements of $M^Q_i$ at higher types need be definable, which is one of the manifestations of its truly semantic nature.

We shall have more to say about this later.

### 5. Limit Construction

Now that we have fully abstract, extensional, algebraic finite models $M^Q_i$ the next natural thing to do is to construct $M^Q$ as their 'limit'. For this to go through the $M^Q_i$'s must bear some relationship to each other. We want that $Q_i$ to be some sense a subretract of $Q_j$ if $i \leq j$.

Note that if $i \leq j$ then $Q_i \subseteq Q_j$. Moreover, we shall show that there exists an injection-projection pair between $Q_i$ and $Q_j$, and this injection-projection pair is the most natural one: $(Q_i, Q'_j)$. Pictorially:

$$Q_i \subseteq Q'_j \subseteq Q_j$$

More formally, for $(Q'_j, Q'_i)$ to be an injection-projection pair two conditions need to be satisfied:

1. for all $c \in Q'_i$, $Q_i \cap Q'_i \subseteq c$
2. for all $d \in Q'_j$, $Q_j \cap Q'_j = d$

If these conditions are satisfied we can rightfully say that $Q'_i$ is a subretract of $Q'_j$. We denote this by $Q'_i \subseteq Q'_j$.

The first condition easily follows from the condition $Q'_i \subseteq Q'_j$. To prove the second condition, consider an arbitrary $d \in Q'_j$. This implies $d \in Q'_j$ too. Now we calculate:

$$Q'_i \cap Q'_j = d = Q'_j = Q'_j d$$

Thus indeed $Q'_i \subseteq Q'_j$ if $i \leq j$.

We now have, for each $\tau$, a monotone sequence:

$$Q'_i \subseteq Q'_j \subseteq \ldots$$

Note that for each $d \in D'_\tau$, by ??,

$$F'(d) = \max_{j \geq d} \max_{i \geq d} [Q'_i] = [Q'_j],$$

As $Q'$ is the unique continuous extension of $F'$ it follows that

$$Q' = \lim_{i \to 0} Q'_i.$$  

As $Q'_i(c) \approx c$ for every $c \in D'_\tau$, we conclude, from the directed completeness of $\Theta$ predicates:

$$Q'(d) \approx d$$

Now $M^Q = (Q', A^Q)$, as defined in ??, can be looked upon as the limit of the models $M^Q_i, M^Q_j, \ldots$. We shall show that $M^Q$ is a fully abstract, extensional, algebraic $\mathcal{L}$-model.

As before we have to show that the application is well defined in $M^Q$, i.e., if $d \in Q' \alpha \rightarrow \beta$ and $c \in Q''$ then $dc \in Q'$. But this time it is easy:

$$dc = \lim_{i \to 0} (Q'' - \alpha \rightarrow \beta)(Q_i^c)$$

As application is well defined in each $M^Q_i$, we know that each $(Q'' - \alpha \rightarrow \beta)(Q_i^c) \in Q'_i$. Hence
Before we prove the extensionality of $M^Q$ let us prove one lemma.

**Lemma 5.1:** Let $d \in Q'_\tau$ and $c \in Q'_\sigma$, where $\tau = \alpha \rightarrow \beta$, and $i \subseteq j$. Then

$$Q_i(d(dQ,c)) = (QJ)c.$$  

**Proof:** One part of the equality, $(QJ)c \subseteq Q_i(d(QJ)c))$ is obvious as $d \subseteq Q_i(d)$ and $c \subseteq Q_i(c)$. It remains to prove that $Q_i(d(QJ)c)) \subseteq (QJ)c.$

As $c \in Q'_\rho$, we know that $c \in D'_\rho$ and hence $c \in D'_\sigma$ too. This means $Q_j(c) \simeq c$. Let $b = Q_i(d(Q_j,c))$. Then

$$b = Q_i(d(Q_j,c)) \subseteq Q_i((QJ)c) \text{ as } Q_i \subseteq Q_j,$$

$$= d(Q_j,c) \text{ as the application in } M^Q \text{ is well defined.}$$

Hence, as $\subseteq$ is a refinement of $\subseteq$, we conclude that $b \subseteq d(Q_j,c)$. Let $d' : D'_\sigma = (c \Rightarrow b)$. Then because $c \simeq Q_j(c)$ and $b \subseteq (QJ)c)$, we immediately conclude from Lemma 4.1 that $d' \subseteq d$. Therefore $d' \subseteq d$, as $d' \max|d'_j| \subseteq \max|d_j| = d$. But then, as $d' \subseteq D'_j$, $d' \subseteq Q_i(d') \subseteq Q_i(d)$. Hence $Q_i(d(QJ)c)) = b = d'c \subseteq (QJ)c)$. 

**Corollary 5.2:** If $d \in Q'_\tau$ and $c \in Q'_\sigma$, where $\tau = \alpha \rightarrow \beta$, then $Q_i(d(Q_j,c)) = (QJ)c)$. 

**Proof:**

$$Q_i(d(Q_j,c)) \subseteq Q_i((QJ)c))$$

$$= Q_i((QJ)(Q_j \circ c)))$$

$$= \bigcup_{j=0}^\infty Q_i((QJ)(Q_j(c)))$$

$$= \bigcup_{j=0}^\infty \bigcup_{k=0}^\infty Q_i((QJ)(Q_j(c)))$$

$$= (QJ)c) \text{ by Lemma 5.1}.\] 

From the above Lemma 5.1 one easily proves that the following diagram commutes:

$$\begin{array}{ccc}
Q_i^{\alpha \rightarrow \beta} \times Q_j^{\beta} & \rightarrow & Q_j^{\beta} \\
\downarrow & & \downarrow \\
Q_i^{\alpha \rightarrow \beta} \times Q_i^{\beta} & \rightarrow & Q_i^{\beta}
\end{array}$$

This roughly says that the application remains invariant under the injection of the family $\{Q'_i\}$ into $\{Q'_j\}$ which is slightly surprising, as this injection 'increases' the elements: $Q'_i(c) \supseteq c$, for $c \in Q'_i$. Thus in true sense $\{Q'_i\}$ can be embedded into $\{Q'_j\}$, or equivalently we can now say that $M^Q$ can be embedded into $M^Q_i$.

Proving extensionality now is easy. Suppose $a, b \in Q'_\tau$, where $\tau = \alpha \rightarrow \beta$, and that for all $c \in Q'_\sigma$, $ac \subseteq bc$. Then for all $h \in Q'_\tau$,

$$(QJ)a h_0 = Q_i(a(QJ)c)) \quad \text{by Corollary 5.2}$$

$$\subseteq Q_i(b(QJ)c)) \quad \text{by assumption, as } Q_i(c) \in Q^\sigma$$

$$= (QJ)b h_0 \quad \text{by Corollary 5.2.}$$

By extensionality of $M^{Q_i}$, we conclude that $Q_i(a) \subseteq Q_i(b)$. Hence

$$a = Q_i(a) = \bigcup_{i=0}^\infty Q_i(a) \subseteq \bigcup_{i=0}^\infty Q_i(b) = Q_i(b) = b,$$

which proves that $M^Q$ is extensional.

Note that for every $t \in L$,

$$A^Q(t) = \bigcup_{i=0}^\infty A_i([t])$$

$$\subseteq \bigcup_{i=0}^\infty A^Q_i([t]) \text{ as } A_i \subseteq A^Q$$

$$= A^Q(t).$$

$A^Q$ is a homomorphism, because for every $r, s \in L$ of appropriate type,

$$A^Q(rs) = Q \circ A^Q(r)$$

$$= (\bigcup_{i=0}^\infty A_i([r_i])) \bigcup_{i=0}^\infty A_i([s_i]),$$

$$\text{by syntactic continuity}$$

$$= (\bigcup_{i=0}^\infty A_i([r_i][s_i]))$$

$$= \bigcup_{i=0}^\infty A^Q_i([r_i][s_i])$$

$$= \bigcup_{i=0}^\infty (A^Q_i([r_i]))(A^Q_i([s_i]))$$

$$= (A^Q_i(r))(A^Q_i(s)),$$

as $A^Q_i$ is a homomorphism.

$$A^Q$$ is a homomorphism because it is the inverse limit of $M^{Q_i}$s and each $M^{Q_i}$ is trivially algebraic.
We now address the question of full abstractness of $M^Q$.

Suppose $t \not\equiv s$. Then by Lemma 3.2, we know that for all $i$, $[t]_i \not\equiv [s]_i$. This is because full abstractness of $M^Q_i$ implies $A^Q_i([t]_i) \subseteq A^Q_i([s]_i)$, for all $i$. Hence

$$A^Q t = \bigcup_{i=0}^{\infty} A^Q_i([t]_i) \subseteq \bigcup_{i=0}^{\infty} ([s]_i) = A^Q(s).$$

On the other hand suppose $A^Q(t) \subseteq A^Q(s)$. Then:

$$A(t) \subseteq A^Q(t) \subseteq A^Q(s) = Q(A s) = A(s).$$

Thus $A(T) \subseteq A(S)$, which, by Corollary 4.8, means that $t \not\equiv s$. We have proved: $M^Q$ is fully abstract.

$Y$ has the standard interpretation in $M^Q$, i.e., $A^Q(Y) = \bigcup_{j=0}^{\infty} A^Q(Y_j)$, where $Y_j = \lambda f.f^n(\Omega)$. This is because

$$A^Q(Y) = Q(A Y) = Q\left(\bigcup_{j=0}^{\infty} A(Y_j)\right) = \bigcup_{j=0}^{\infty} Q \circ A(Y_j) = \bigcup_{j=0}^{\infty} A^Q(Y_j).$$

$M^Q$ is also a $\beta$-model (i.e., a model for beta-conversion). For this one has to prove that

1. $A^Q(S w u v) = A^Q((u w)(u v))$ for all $u, v, w \in L$ of appropriate types.
2. $A^Q(K w u) = A^Q(u)$ for all $u, v \in L$ of appropriate types.

These equations say that $S$ behaves like $S$, $K$ behaves like $K$, and they constitute a closed combinator version of the usual beta-conversion equation. But they are obviously true in $M^Q$ because $S w u v \equiv (u w)(u v)$, $K w u \equiv u$, and $M^Q$ is fully abstract.

To summarize: $M^Q$ is a fully abstract, extensional, algebraic $\beta$-model for $L$. $Y$ has the standard interpretation in $M^Q$ and moreover $A^Q = Q \circ A$. This at last finishes the proof of Theorem 3.1.

We can now easily show that $t \not\equiv s$ iff for all ground contexts $C[]$,

$$O(C[t]) \not\subseteq O(C[s]).$$

(5)

Suppose $t \not\equiv s$. Then, because $M^Q$ is fully abstract, $A^Q(t) \subseteq A^Q(s)$. This means, as $A^Q$ is a homomorphism, that $A^Q(C[t]) \subseteq A^Q(C[s])$ for all ground contexts $C[]$. But then,

$$O(C[t]) = A^Q(C[t]) \quad \text{as} \quad M^Q \quad \text{is adequate}$$

$$\subseteq A^Q(C[s]) = O(C[s]) \quad \text{as} \quad M^Q \quad \text{s adequate.}$$

On the other hand if $O(C[t]) \subseteq O(C[s])$ for all ground contexts then $O(t_1 \ldots t_n) \subseteq O(s_1 \ldots s_n)$ for all $t_1, \ldots, t_n \in L$ of appropriate types (let context be $[t_1 \ldots t_n]$. And hence $t \not\equiv s$.

Thus (5) could be used as an alternative definition of $\not\equiv$ instead of the one given in (1), because both of them are now shown to be equivalent. It is surprising how naturally this equivalence follows from the existence of fully abstract $M^Q$. Contrast this with the elaborate efforts taken in [1] to prove this equivalence (Milner's First Context Lemma). Milner could not have taken our approach because, unlike in our case, the construction of his fully abstract model depends upon the validity of the above equivalence.

6. Conclusion

Observe that in our fully abstract model $M^Q$ not all the finite elements at the higher type need be definable. We have not tried to come up with a specific finite element which is not definable. But we conjecture this that is the case. Contrast this with Milner's syntactic construction of a fully abstract model for typed lambda calculus, where every finite element is definable. Even when the classical model of continuous functions was shown fully abstract in [2] for PCF enriched with 'parallel or' it turned out that all the finite elements of the model were definable in the enriched language. To our knowledge, this is the first model which is fully abstract and whose all finite elements need not be definable still. We leave it open to find out the properties which the ground domains should have so that all the finite elements of $M^Q$ are definable. (see articulate domains of [4]).

We also leave open whether a better semantic characterization can be found when $L$ is known to possess the additional properties like sequentiality (see [3]).

Though we did not show it, the $\Theta$ predicates defined in this paper can be used to show that $O$ and $A$ are semantically equivalent. In fact such inclusive predicates were introduced in [4] and [5] with exactly this aim in mind: to show the semantic equivalence between operational and denotational semantics. The techniques developed in these papers were mainly meant for the cases when the domains under consideration were reflexive. It should not then come as a surprise if the technique developed in this paper could be extended to obtain a semantic characterization of full abstraction even when domains under consideration are reflexive. In fact that is the case. (see [8] or [9]).

The same inclusive predicates which are used to show that the operational and denotational semantics are equivalent can used to collapse the model onto a fully abstract
This ties everything together nicely.

7. Acknowledgements

This work was done under the guidance of Prof. Dana Scott. I thank him for the valuable suggestions he gave me. Also thanks to Steve Brookes for his numerous helpful comments and to Roberto Minio for his help in TEX.

REFERENCES