STANFORD ARTIFICIAL INTELLIGENCE LABORATORY
MEMO AIM-186

STAN-CS-73-332

MODELS OF LCF

BY

ROBIN MILNER

SUPPORTED BY

ADVANCED RESEARCH PROJECTS AGENCY

ARPA ORDER NO. 457

JANUARY 1973

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD /UNIVERSITY

MODELS OF LCF

1. Introduction

The logic of computable functions proposed by Dana Scott in 1969, in an unpublished note, has since been the subject of an interactive proof-checking program designed as a first step in formally based machine-assisted reasoning about computer programs. This implementation is fully documented in [1], and its subsequent applications are reported in later papers [2,3,4, and 5]. However the model theory of the logic, which scott originally supplied, is not discussed in those papers, and the purpose of this Memorandum is to present that theory. Nothing is added here to Scott's work. The concept of a continuous function, which is central to the theory, has since been developed by him to provide models for the X-calculus and to yield his mathematical theory of continuous lattices; the interested reader can follow these topics in Scott [6]. However, since LCF is only a version of the *typed* X-calculus, these developments are not necessary for the present purpose, and the present paper contains all that is needed to understand LCF.

2. Continuous Function Domains

In this section we define a particular sort of partially ordered domain, called a complete partial order (cpo), and the concept of continuous function. We prove some propositions for later use; in particular, that if D and E are cpo's, then the set of continuous functions from D to E is itself a cpo.

Definition 2.1 A partial order (po) is a pair $(D,\sqsubseteq)$ where D is any set (domain) and $\sqsubseteq$ is a transitive, reflexive , antisymmetric relation over D.

Definition 2.2 For a po $(D,\sqsubseteq)$, a set $X \subseteq D$ is a chain if $X = \{x_i | i \geq 0\}$ and $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \cdots$.

Definition 2.3 A po $(D,\sqsubseteq)$ is a complete partial order (cpo) if

(1) It has a minimum element, which we denote by $\perp_D$, or just $\perp$ if there is no confusion.

(2) Every chain $X \subseteq D$ has a least upper bound (lub) in D, which we denote by $\sqcup X$.

Definition 2.4 If D and E are cpo's, then a function $f : D \to E$ is continuous if every chain $X \subseteq D$ satisfies $\sqcup\{f(x) : x \in X\} = f(\sqcup X)$.

Thus a continuous function is one which preserves the lubs of chains. Note that the set on the lefthand side of the above equation is a chain, since if $X = \{x_0, x_1, \ldots\}$ and $x_0 \sqsubseteq x_1 \sqsubseteq \ldots\ldots$ then we also have $f(x_0) \sqsubseteq f(x_1) \sqsubseteq \ldots$. To see this, we only need to

STANFORD ARTIFICIAL INTELLIGENCE LABORATORY JANUARY 1973
MEMO AIM- 186

COMPUTER SCIENCE DEPARTMENT
REPORT CS-332

MODELS OF LCF

by

Robin Milner

ABSTRACT: LCF is a deductive system for computable functions proposed by D. Scott in 1969 in an unpublished memorandum. The purpose of the present paper is to demonstrate the soundness of the system with respect to certain models, which are partially ordered domains of continuous functions. This demonstration was supplied by Scott in his memorandum; the present paper is merely intended to make this work more accessible.

This research was supported in part by the Advanced Research Projects Agency of the Office of the Secretary of Defense under Contract No. SD-183.

The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.S. Government.

Reproduced in the USA. Available from the National Technical Information Service, Springfield, Virginia 22151.

observe that any continuous function is monotonic - that is, $x \sqsubseteq y \Rightarrow f(x) \sqsubseteq f(y)$, and this is true because if Y is the chain $\{x \sqsubseteq y\}$ then $\sqcup Y = y$, so we have $f(x) \sqsubseteq \sqcup\{f(x), f(y)\} = f(\sqcup Y) = f(y)$.

We should also note that there is an alternative (more restrictive) definition of a cpo which uses the concept of *directed set* (X is directed iff $x, y \in X \Rightarrow \exists z \in X . x, y \sqsubseteq z$) instead of chain, This, in turn, leads to an alternative (more restrictive) definition of continuous function. We have chosen the less restrictive alternative, but we remark that the theory can be done equally well (as far as we are here concerned) with either definition.

Notice that we use the same symbol $\sqsubseteq$ for the relation in every po under discussion. This should give no difficulty. We also use names like D and E both for po's and for their domains.

Definition 2.5 We denote the set of continuous functions from D to E, where these are cpo's, by $[D \rightarrow E]$.

Proposition 2.1 If D and E are cpo's then $F = [D \rightarrow E]$ is a cpo under the relation

$$f \sqsubseteq g \quad \text{iff} \quad \forall x .\ f(x) \sqsubseteq g(x)$$

Proof First, F *is* is a po under this relation (check reflexivity, transitivity and antisymmetry). Second, the minimum element $\perp_F$ of F is easily seen to be $\lambda x . \perp_E$. Finally, we need that any chain $Z \subseteq F$ has a lub $\sqcup Z \in F$. Define $\sqcup Z = \lambda x \cdot \sqcup\{f(x) : f \in Z\}$. This is a well-defined function since for each x in D, $\{f(x) : f \in Z\}$ is easily seen to be a chain in E. Next, it bounds above every $f \in Z$, since for each $x \in D$, $f(x) \sqsubseteq \sqcup\{f(x) : f \in Z\} = (\sqcup z)(x)$. Further, it is a lub, since if h is any other upper bound for Z, then for each $x \in D$ and $f \in Z$, we have $f(x) \sqsubseteq h(x)$; it follows that $(\sqcup Z)(x) \sqsubseteq h(x)$, and hence $\sqcup Z \sqsubseteq h$.

But we must aiso show that $\sqcup Z \in F$, i.e., $\sqcup Z$ is *continuous.*

Let $X \subseteq D$ be a chain. We require

$$(uz)\ (\sqcup X) = \sqcup\{(\sqcup Z)\ (x)\ :\ x \in X\},$$

But $(\sqcup Z)\ (\sqcup X)$ $= \sqcup\{f(\sqcup X)\ :\ f \in Z\}$ by the definition of $\sqcup Z$.

$= \sqcup\{f(x)\ :\ f \in Z, x \in X\}$

$= \sqcup\{(\sqcup Z)\ (x)\ :\ x \in X\}$.

This completes the proof. ☒

<u>Proposition 2.2</u> For any cpo D, every $f \in [D \rightarrow D]$ has a minimum fixed-point $Yf \in D$ - i.e. we have $f(Yf) = Yf$ and for all $x \in D$, $f(x) = x$ implies $Yf \sqsubseteq x$.

<u>Remark</u> This proposition ensures the existence of the least fixed-point operator $Y : [D \rightarrow D] \rightarrow D$. The next proposition shows that Y is continuous, i.e. $Y \in [[D \rightarrow D] \rightarrow D]$.

<u>Proof</u> The set $S = \{f^{i}(\perp_D)\ :\ 0 \leq i\}$ is a chain by the monotonicity of f. Define Yf = US. By the continuity of f, we have $f(Yf) = \sqcup\{f^{i+1}(\perp_D)\ :\ 0 \leq i) = Yf$, so Yf is a fixed-point of f. Let x be any other fixed-point. Now by the monotonicity of f we have $f(Q) \sqsubseteq f(x) = x$, and by induction on i we can show $f^{i}(\perp_D) \sqsubseteq x$ for all $i \geq 0$, so $Yf = \sqcup\{f^{i}(\perp_D)\ :\ 0 \leq i) \sqsubseteq x$, and thus Yf is the minimum fixed-point of f. ☒

<u>Propos:2ion. 3</u> Y is continuous, so $Y \in [[D \rightarrow D] \rightarrow D]$

<u>Proof</u> Let Z be any chain $\subseteq [D \rightarrow D]$. We must show that $Y(\sqcup Z) = \sqcup\{Yf\ :\ f \in Z\}$. In one direction ($\sqsupseteq$) proof is easy since for each $f \in Z$, $\sqcup Z \sqsupseteq f$, so $Y(\sqcup Z) \sqsupseteq Yf$ by the monotonicity of Y which in turn follows directly from the definition of Yf. In the other direction we only need to show that $\sqcup\{Yf\ :\ f \in Z\}$ is a fixed-point of $\sqcup Z$, since then

it dominates the least such, which is $Y(\sqcup Z)$. ☠ □ ◆

$$\begin{aligned}
\sqcup Z(\sqcup\{Yf : f\in Z\}) &= \sqcup\{g(\sqcup\{Yf : f\in Z\}) : g\in Z\} \\
&= \sqcup\{g(Yf) : g\in Z, f\in Z\} \quad \text{by continuity of } g. \\
&= \sqcup\{f(Yf) : f\in Z\}, \quad \text{since} \\
&\qquad\qquad g(Yf) \sqsubseteq h(Yh) \text{ where } h = \max(g,f). \\
&= \sqcup\{Yf : f\in Z\}
\end{aligned}$$

which is the required fixed-point property. This completes this proof.

☒

3. Pure LCF : Terms

In this section we give the term syntax of Pure LCF, and then after defining a standard interpretation as a function from identifiers into the union of a family of cpo's, we show how such an interpretation is extended uniquely to a function from all terms into the same range. The terms of Pure LCF are just those of a typed X-calculus.

Types (1) ind and tr are (basic) types.

(2) If $\beta1$, $\beta2$ are types then $(\beta1 \rightarrow \beta2)$ is a type.

(3) These are all the types.

We use β, $\beta1$, $\beta2,\ldots$ to denote types, and frequently omit parentheses, assuming that '$\rightarrow$' associates to the right, so that $\beta1 \rightarrow \beta2 \rightarrow \beta3$ abbreviates $(\beta1 \rightarrow (\beta2 \rightarrow \beta3))$.

Terms Each term has a well defined type. We use s,t,u to denote terms, and write s : β to mean that s has type β.

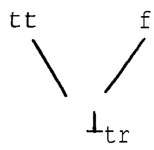
(1) Any identifier is an (atomic) term. We do not need to describe them, except to say that there are infinitely many at each type, that the type of each is determined in some way (perhaps by explicit subscripting), and that they include TT : tr, FF : tr and the families (indexed by type) uu_{β}, $\supset_{tr \rightarrow \beta \rightarrow \beta \rightarrow \beta}$ and $Y_{(\beta \rightarrow \beta) \rightarrow \beta}$. These identifiers are special only in that each standard interpretation will assign a particular element to each of them. We use x,y to denote arbitrary identifiers.

(2) If s : $\beta1 \rightarrow \beta2$ and t : $\beta1$ are terms then s(t) : $\beta2$ is a term. If x : $\beta1$ is an identifier and s : $\beta2$ is a term, then $[\lambda x \cdot s]$: $\beta1 \rightarrow \beta2$ is a term.

(3) These are all the terms.

Remark In the machine implementation of LCF, and often for intelligibility, we have written terms of the form x(s)(t)(u) and $Y([\lambda x \cdot s])$ respectively as $(s \rightarrow t,u)$ and $[\alpha x \cdot s]$, and have dispensed with $\supset$ and Y. It is clear that every term of implemented LCF is then a transcription of a term of Pure LCF, and it therefore suffices to discuss the semantics of the latter.

Semantics A standard model (of LCF) is a family $\{D_\beta\}$ of cpo's, one for each type β, where $D_{\underline{ind}}$ is an arbitrary cpo, D_{tr} is the cpo $\{tt,ff,\perp_{tr}$ 3 under the partial order given by the diagram



and $D_{\beta 1 \rightarrow \beta 2} = [D_{\beta 1} \rightarrow D_{\beta 2}]$. Note that $D_{\underline{ind}}$ completely determines a standard model.

Let $\mathcal{J}$ be the set of identifiers of Pure LCF. A standard interpretation (of LCF) is a standard model $(D_\beta\}$ together with a standard assignment, which is a function

$$\mathcal{A} : \mathcal{J} \rightarrow \bigcup\{D_\beta\}$$

which satisfies the further conditions

(1)* $\mathcal{A}[\![x : \beta]\!] \in D_\beta$

(2) The value of $\mathcal{A}$ for the special identifiers is given by the following:

* We write the (syntactic) arguments of $\mathcal{A}$ in decorated brackets as an aid to the eye.

it yields a <u>continuous</u> function over the appropriate domains.

We define $\mathcal{A}$ by induction on the structure of terms, as follows:

$$\mathcal{A}[\![s(t)]\!] = \mathcal{A}[\![s]\!](\mathcal{A}[\![t]\!])$$

$$a[\![[\lambda x \cdot s]]\!] = \lambda\xi \cdot \mathcal{A}_{\xi/x}[\![s]\!] .$$

That $\mathcal{A}$ respects types is obvious. That $\mathcal{A}[\![s]\!] \in D_{\beta}$ for all β and $s : \beta$ is a corollary of the following

<u>Proposi3.1n</u> For each assignment $\mathcal{A}$ and for each $x : \beta 1$, $s : \beta 2$, $\lambda\xi \in D_{\beta 1} \cdot \mathcal{A}_{\xi/x} [\![s]\!] \in [D_{\beta 1} \to D_{\beta 2}]$.

<u>Proof</u> First, suppose s is an atomic term, i.e. an identifier. Either $s = x$, in which case $\beta 1 = \beta 2$ and $\lambda\xi \cdot \mathcal{A}_{\xi/x}[\![s]\!]$ is the identity function over $D_{\beta 1}$, or $s \neq x$ in which case it is a constant function from $D_{\beta 1}$ to $D_{\beta 2}$. In either case it is a continuous function, hence $\in [D_{\beta 1} \to D_{\beta 2}]$.

Next suppose s is $t(u)$, $t : \beta 3 \to \beta 2$ and $u : \beta 3$. Assume the proposition for t and u. We have to show that for any chain $X \subseteq D_{\beta 1}$,

$$\sqcup\{\mathcal{A}_{\xi/x}[\![t(u)]\!] : \xi \in X\} = \mathcal{A}_{\sqcup X/x}[\![t(u)]\!] \text{; that is, that}$$

$$\sqcup\{\mathcal{A}_{\xi/x}[\![t]\!](\mathcal{A}_{\xi/x}[\![u]\!]) : \xi \in X\} = \mathcal{A}_{\sqcup X/x}[\![t]\!](\mathcal{A}_{\sqcup X/x}[\![u]\!]).$$

Now if we denote $\lambda\xi \cdot \mathcal{A}_{\xi/x}[\![t]\!]$ and $\lambda\xi \cdot \mathcal{A}_{\xi/x}[\![u]\!]$ by f and g, the inductive assumption tells us that $f \in [D_{\beta 1} \to [D_{\beta 3} \to D_{\beta 2}]]$ and $g \in [D_{\beta 1} \to D_{\beta 3}]$, and the required equation merely states that for such f and g, $\lambda\xi \cdot f(\xi)(g(\xi))$ is continuous. The proof of this we leave to the reader; it is hardly more than proving that for a chain X, $\{f(\xi)(g(\xi)) : \xi \in X\}$ and $\{f(\xi)(g(\eta)) : \xi, \eta \in X\}$ are cofinal chains.

Finally, suppose s is $[\lambda y . t]$, $y : \beta 3$, $t : \beta 4$ and $\beta 2 = \beta 3 \to \beta 4$. We need to show that

$$\lambda\xi\in D_{\beta 1},\ \mathcal{A}_{\xi/x}[\![\,[\lambda y\cdot t]\,]\!] \in [D_{\beta 1}\to [D_{\beta 3}\to D_{\beta 4}]]$$

that is, that for any chain $X \subseteq D_{\beta 1}$,

$$\bigsqcup \{\lambda\eta\in D_{\beta 3}\cdot(\mathcal{A}_{\xi/x})_{\eta/y}[\![t]\!] : \xi\in X\} =$$
$$\lambda\eta\in D_{\beta 3}\cdot(\mathcal{A}_{\sqcup X/x})_{\eta/y}[\![t]\!]$$

Now in the case $x = y$, we have $(\mathcal{A}_{\xi/x})_{\eta/y} = (\mathcal{A}_{\sqcup X/x})_{\eta/y} = \mathcal{A}_{\eta/y}$ and the equation reduces to a tautology. If $x \neq y$, then $(\mathcal{A}_{\xi/x})_{\eta/y} = (\mathcal{A}_{\eta/y})_{\xi/x}$, and the inductive hypothesis (that the proposition is true for t) tells us that $\lambda\xi\cdot(\mathcal{A}_{\eta/y})_{\xi/x}[\![t]\!]$ is continuous - hence monotonic - so $\{ (\mathcal{A}_{\xi/x})_{\eta/y}[\![t]\!]\}$ is a chain in $D_{\beta 4}$, for each η. Moreover, the inductive hypothesis also tells us that for each ξ $\lambda\eta\,.\,(\mathcal{A}_{\xi/x})_{\eta/y}[\![t]\!]$ is in $[D_{\beta 3}\to D_{\beta 4}]$, and by the previous remark the set of these functions - as ξ ranges over X - is a chain in $[D_{\beta 3}\to D_{\beta 4}]$. Thus by the definition of $\sqcup$ for function spaces (Proposition 2.1) we can replace the lefthand side of the desired equation by

$$\lambda\eta\in D_{\beta 3}\ .\ \bigsqcup \{(\mathcal{A}_{\eta/y})_{\xi/x}[\![t]\!] : \xi\in X\}$$
$$= \lambda\eta\in D_{\beta 3}\ .\ (\mathcal{A}_{\eta/y})_{\sqcup X/x}[\![t$$
$$= \lambda y\in D_{\beta 3}\ .\ (\mathcal{A}_{\sqcup X/x})_{\eta/y}[\![t]\!] \qquad \text{since } x \neq y$$

and we are done. We have therefore proved the proposition by induction on the structure of terms. ☒

Corollary 3.2

For every assignment $\mathcal{A}$, type β, and term $s : \beta$, $\mathcal{A}[\![s]\!] \in D_{\beta}$.

Proof For atomic terms the corollary is assured by the definition of an assignment. For X-terms, the proposition gives the corollary directly, For an application term $s(t) : \beta$, the proposition tells us that

$\lambda\xi\in D_{\beta 1}.\mathcal{A}_{\xi/x}\llbracket s(t)\rrbracket \in [D_{\beta 1}\to D_{\beta}]$, so by application to $\mathcal{A}\llbracket x\rrbracket$ we get

$$\mathcal{A}\llbracket s(t)\rrbracket = \mathcal{A}_{\mathcal{A}\llbracket x\rrbracket/x}\llbracket s(t)\rrbracket \in D_{\beta}$$

as required. ☒

$$\mathcal{A}[\![\, TT \,]\!] = tt, \ \mathcal{A}[\![\, FF \,]\!] = ff,$$

$$\mathcal{A}[\![\, UU_\beta \,]\!] = \bot_\beta,$$

$$\mathcal{A}[\![\, \supset_{\underline{tr} \to \beta \to \beta \to \beta} \,]\!] =$$

$$\lambda\xi\in D_{\underline{tr}} . \lambda\eta\in D_\beta . \lambda\chi\in D_\beta . (\xi \to \eta,\chi), \text{ and}$$

$$\mathcal{A}[\![\, Y_{(\beta \to \beta) \to \beta} \,]\!] = Y_{(\beta \to \beta) \to \beta}$$

where $(\xi \to \eta,\chi)$ - the conditional - takes the values $\bot$, η, χ according as $\xi = \bot_{tr}$, tt, ff, and where we have subscripted the fixed-point operator Y on the right to indicate that it belongs to $[[D_\beta \to D_\beta] \to D_\beta]$. Note that the Y on the left is an identifier, and the Y on the right a function. It is easy to check that $\mathcal{A}[\![\supset]\!]$ is a continuous function, and Proposition 2.3 has assured us that $\mathcal{A}[\![\, Y \,]\!]$ is also continuous.

If $\mathcal{A}$ satisfies condition (1) above, but not necessarily condition (2), we call it just an *assignment*, yielding an *interpretation* (not necessarily standard). We also confuse the terms assignment and interpretation, since we have no occasion to discuss here different standard models.

We write $\mathcal{A}_{\xi/x}$ to indicate the assignment differing from $\mathcal{A}$ only in that its value at x is ξ; clearly we have that

$$(\mathcal{A}_{\xi/x})_{\eta/y} = \begin{cases} \mathcal{A}_{\eta/y} & \text{if } x = y \\ (\mathcal{A}_{\eta/y})_{\xi/x} & \text{otherwise.} \end{cases}$$

We now show how to extend the domain of an assignment $\mathcal{A}$ to all terms, preserving the condition that

$$\mathcal{A}[\![\, s : \beta \,]\!] \in D_\beta$$

which states not only that $\mathcal{A}$ respects types, but also that (for composite types)

4. Pure LCF : Formulae, Sentences, Rules and Validity

In this section we define the remainder of the syntax of Pure LCF, extending the domain of assignments $\mathcal{A}$ still further, and after defining the concept of validity of a sentence we give the rules of inference and show that they preserve validity.

Atomic well-formed formulae (awffs)

If s,t : β are terms, then s ⊂ t is an awff. Let us add the truth values T,F (not to be confused with TT, FF) to the range of an assignment, and extend any $\mathcal{A}$ to awffs by

$$\mathcal{A}[\![s \subset t]\!] = \begin{cases} T & \text{if } \mathcal{A}[\![s]\!] \sqsubseteq \mathcal{A}[\![t]\!] \\ F & \text{otherwise} \end{cases}$$

Well-formed formulae (wffs)

A wff is a set of awffs. We use P,Q,P1,Q1,...... to denote arbitrary wffs. Extend $\mathcal{A}$ to wffs by

$$\mathcal{A}[\![P]\!] = \begin{cases} T \text{if} & A \in P \Rightarrow \mathcal{A}[\![A]\!] = T \\ F & \text{otherwise.} \end{cases}$$

We use s ≡ t to abbreviate $\{s \subset t, \ t \subset s\}$.

Sentences

If P,Q are wffs, then P ⊢ Q is a sentence (if P = ∅, we just write ⊢ Q). Extend $\mathcal{A}$ to sentences by

$$\mathcal{A}[\![P \vdash Q]\!] = \begin{cases} F \text{ if } \mathcal{A}[\![P]\!] = T, \ \mathcal{A}[\![Q]\!] = F \\ T \text{ otherwise.} \end{cases}$$

We say that P⊢Q is false in $\mathcal{A}$, true in $\mathcal{A}$ respectively. We say that a sentence is valid iff it is true in all standard interpretations.

We now introduce the rules of inference of Pure LCF, accompanying each by a proof - often very trivial - that it is valid (a rule is valid if whenever its hypotheses are valid its conclusion is valid). The proofs will rely on two facts about assignments which are fairly easy to prove (we omit their proofs). First, if A is any syntactic entity in the domain of an assignment $\mathcal{A}$, and x is not free in A, then $\mathcal{A}\llbracket A \rrbracket$ is independent of $\mathcal{A}\llbracket x \rrbracket$; more precisely, $\mathcal{A}_{\xi/x}\llbracket A \rrbracket = \mathcal{A}\llbracket A \rrbracket$. Second, in specifying the inference rules we use $A\{t/x\}$ to mean: Substitute t for x in A with suitable changes of bound variables so that no identifier free in t becomes bound after the substitution, and we need the fact that $\mathcal{A}\llbracket A\{t/x\} \rrbracket = \mathcal{A}_{\mathcal{A}\llbracket t \rrbracket/x}\llbracket A \rrbracket$.

Rules of Inference

We write the hypotheses of each rule above a solid line. If there are none, we omit the solid line. We use the same names for rules as in [1].

INCL $\quad P \vdash Q \qquad (Q \subseteq P)$

Clearly P true. in $\mathcal{A}$ implies Q true in $\mathcal{A}$.

CONJ

$$\frac{P \vdash Q1 \qquad\qquad P \vdash Q2}{P \vdash Q1 \cup Q2}$$

Clearly valid

CUT

$$\frac{P1 \vdash P2 \qquad\qquad P2 \vdash P3}{P1 \vdash P3}$$

Clearly valid.

APPL $\quad t \subset u \vdash s(t) \subset s(u)$

If $\mathcal{A}\llbracket t \rrbracket \sqsubseteq \mathcal{A}\llbracket u \rrbracket$, then $\mathcal{A}\llbracket s(t) \rrbracket = \mathcal{A}\llbracket s \rrbracket(\mathcal{A}\llbracket t \rrbracket) \sqsubseteq \mathcal{A}\llbracket s \rrbracket(\mathcal{A}\llbracket u \rrbracket) = \mathcal{A}\llbracket s(u) \rrbracket$, using the monotonicity of $\mathcal{A}\llbracket s \rrbracket$.

REFL $\vdash s \subset s$

Clearly valid, by reflexivity of $\sqsubseteq$

TRANS s c t, $t \subset u$ t- scu

Clearly valid by transitivity of $\sqsubset$

MIN1 $\vdash$ uucs

Clearly valid, by the minimality of $\perp_{\beta}$

MIN2 $\vdash UU(s) \subset UU$

Clearly valid, by the definition $\perp_{\beta 1 \to \beta 2} = \lambda\xi\in\beta 1.\ \perp_{\beta 2}$

Note that in the last two rules we have omitted the type subscripts from UU, intending that they be supplied in such a way as to yield a proper awff - i.e. that the terms on either side should have the same type. We could have written $UU_{\beta 1 \to \beta 2}(s : \beta 1) \subset UU_{\beta 2}$. Similarly we will omit subscripts from $\supset$ and Y.

CONDT $\vdash \supset (TT)(s)(t) \equiv s$

CONDU $\vdash \supset (UU)(s)(t) \equiv uu$

CONDF I- $\supset$ @F)(s)(t) $\equiv t$

These rules are justified by the standard interpretation of $\supset$.

ABSTR $$\frac{P \text{ t- } s \subset t}{P \vdash [\lambda x.s] \subset [\lambda x .t]}$$ x not free in P.

Let $\mathcal{A}$ be such that $\mathcal{A}[\![P]\!] = T$. Since x is not free in P, we have also $\mathcal{A}_{\xi/x}[\![P]\!] = T$ for any ξ. So the hypotheses of the rule assures us that for each ξ in D_{β}, where $x : \beta$, $\mathcal{A}_{\xi/x}[\![s]\!] \sqsubseteq \mathcal{A}_{\xi/x}[\![t]\!]$. Hence $\lambda\xi.\mathcal{A}_{\xi/x}[\![s]\!] \sqsubseteq \lambda\xi.\mathcal{A}_{\xi/x}[\![t]\!]$, which is to say that

$\mathcal{A}[\![[\lambda x.s] \subset [\lambda x.t]]\!] = T$, as required.

CONV $\vdash [\lambda x.s]\ (t) \equiv s\{t/x\}$

We have that $\mathcal{A}\llbracket [\lambda x.s](t)\rrbracket = (\lambda\xi.\mathcal{A}_{\xi/x}\llbracket s\rrbracket)(\mathcal{A}\llbracket t\rrbracket)$ $= \mathcal{A}_{\mathcal{A}\llbracket t\rrbracket/x}\llbracket s\rrbracket$, which is equal to $\mathcal{A}\llbracket s\{t/x\}\rrbracket$ by the second of the facts about assignments which we have assumed.

ETACONV $\qquad \vdash [\lambda x.y(x)] = y$, y distinct from x

$$\mathcal{A}\llbracket [\lambda x.y(x)]\rrbracket = \lambda\xi.\mathcal{A}_{\xi/x}\llbracket y(x)\rrbracket = \lambda\xi.\mathcal{A}_{\xi/x}\llbracket y\rrbracket(\mathcal{A}_{\xi/x}\llbracket x\rrbracket)$$

$= \lambda\xi.\mathcal{A}\llbracket y\rrbracket(\xi)$ (since x is distinct from y, so does not occur free in y), $= \mathcal{A}\llbracket y\rrbracket$.

CASES

$$\frac{P,\ s \equiv TT \text{ i-}Q \qquad P,\ s \equiv UU \vdash Q \qquad P,\ s \equiv FF \vdash Q}{P \vdash Q}$$

Let $\mathcal{A}$ be such that $\mathcal{A}\llbracket P\rrbracket = T$. Since s : tr, $\mathcal{A}\llbracket s\rrbracket$ must take one of the values $\{tt, \bot_{tr}, ff\}$, so that one of $\mathcal{A}\llbracket s \equiv TT\rrbracket$, $\mathcal{A}\llbracket s \equiv UU\rrbracket$, $\mathcal{A}\llbracket s \equiv FF\rrbracket$ takes the value T. The validity of the appropriate hypothesis ensures $\mathcal{A}\llbracket Q\rrbracket = T$.

FIXP. $\qquad \vdash Y(x) \equiv x(Y(x))$

Clearly valid by the standard interpretation of Y.

INDUCT.

$$\frac{P \vdash Q\text{f}\,UU/x\} \qquad P \cup Q \vdash Q\{s(x)/x\}}{P \vdash Q\{Y(s)/x\}}$$

x not free in P or s

For simplicity, we consider just the case that Q is an awff. Moreover we can assume that it is of the form $t(x) \subset u(x)$ where x is not free in t or u, since for any term t', $\mathcal{A}\llbracket t'\rrbracket = \mathcal{A}\llbracket [\lambda y.t'\{y/x\}](x)\rrbracket$, y distinct from x, and then x is not free in $[\lambda y.t'\{y/x\}]$. Let $\mathcal{A}$ be a standard assignment, $\mathcal{A}\llbracket P\rrbracket = T$, and assume that $\mathcal{A}\llbracket s\rrbracket = f$, $\mathcal{A}\llbracket t\rrbracket = g$, $\mathcal{A}\llbracket u\rrbracket = h$. We first show by induction on i that for each $i \geq 0$, $g(f^i(\perp_\beta)) \sqsubseteq h(f^i(\perp_\beta))$, where $x : \beta$. For $i = 0$, the first hypothesis gives that $\mathcal{A}_{\perp_\beta/x}\llbracket Q \rrbracket = T$, that is $\mathcal{A}\llbracket t \rrbracket(\perp_\beta) \sqsubseteq \mathcal{A}\llbracket u \rrbracket(\perp_\beta)$ (since x is not free in t,u), so $g(\perp_\beta) \sqsubseteq h(\perp_\beta)$. Now assume the inequality for i. That is, we assume $\mathcal{A}_{f^i(\perp_\beta)/x}\llbracket Q \rrbracket = T$. Since x is not free in P, we also have $\mathcal{A}_{f^i(\perp_\beta)/x}\llbracket P \rrbracket = T$, and we deduce from the second hypothesis that $\mathcal{A}_{f^i(\perp_\beta)/x}\llbracket Q\,\{s(x)/x\} \rrbracket = T$. Now $\mathcal{A}_{f^i(\perp_\beta)/x}\llbracket s(x) \rrbracket = f(f^i(\perp_\beta))$, since x is not free in s, $= f^{i+1}(\perp_\beta)$, so from the second fact which we assumed for assignments we deduce that $\mathcal{A}_{f^{i+1}(\perp_\beta)/x}\llbracket Q \rrbracket = T$, that is $g(f^{i+1}(\perp_\beta)) \sqsubseteq h(f^{i+1}(\perp_\beta))$. So the induction is complete. Now $\mathcal{A}\llbracket Q\{Y(s)/x\} \rrbracket = \mathcal{A}_{Y(f)/x}\llbracket Q \rrbracket$, which we require to take the value T. That is, we require $g(Y(f)) \sqsubseteq h(Y(f))$. But $g(Y(f)) = \sqcup\{g(f^i(\perp_\beta)) : i \geq 0\}$ (by the continuity of g), $\sqsubseteq \sqcup\{h(f^i(\perp_\beta)) : i > 0\}$ (by what we have proved), $\sqsubseteq h(Y(f))$ by the monotonicity of h, and the justification is complete.

This completes also our justification of the validity of the Rules of LCF.

REFERENCES

1] Milner, R., "Logic for Computable Functions. Description of a Machine Implementation", Artificial Intelligence Laboratory Memo No. AIM-169, Computer Science Department, Stanford University (1972).

2] Milner, R., "Implementation and Applications of Scott's Logic for Computable Functions", Proc. ACM Conference on Proving Assertions about Programs, New Mexico State University, Las Cruees, New Mexico, (1972).

3] Weyhrauch, R. and Milner, R., "Program Semantics and Correctness in a Mechanized Logic", Proc. USA-Japan Computer Conference, Tokyo (1972) (to appear).

4] Milner, R., and Weyhrauch, R., "Proving Compiler Correctness in a Mechanized Logic", Machine Intelligence 7, ed. D. Michie, Edinburgh University Press (1972) (to appear).

5] Milner, R., "A Calculus for the Mathematical Theory of Computation", Proc. Symposium on Theoretical Programming, Novosibirsk, USSR (1972) (to appear in the Springer-Verlag Lecture Notes Series).

6] Scott, D., "Continuous Lattices", Proc. 1971 Dalhousie Conference, Springer Lecture Note Series, Springer Verlag, Heidelberg.