Models of LCF

ROBIN MILNER

Stanford Artificial Intelligence Laboratory

Computer Science Department

School of Humanities and Sciences, Stanford University

Typeset with typographic corrections by Patryk Zadarnowski, March 2010.

LCF is a deductive system for computable functions proposed by D. Scott in 1969 in an unpublished memorandum.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.

---

1. INTRODUCTION

The logic of computable functions proposed by Dana Scott in 1909, in an unpublished note, has since been the subject of an interactive proof checking program designed as a first step in formally based machine-assisted reasoning about computer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 [2,3,4,5]. 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 λ-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* λ-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 a partially ordered domain, called a complete partial order (CPO), and the concept of a continuous function. We prove some propositions for later use; in particular, that if D and E are CPOs, 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 $X = \{x_i \mid i \geq 0\}$ and $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \dots$.

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, and

(2) every chain $X \subseteq D$ has a least upper bound (LUB) in D, which we denote by $\bigsqcup X$.

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

Thus a continuous function is one which preserves the LUBs of chains. Note that the set on the left-hand side of the above equation is a chain, since if $X = \{x_0, x_1, \dots\}$ and $x_0 \sqsubseteq x_1 \sqsubseteq \dots$ then we also have $f(x_0) \sqsubseteq f(x_1) \sqsubseteq \dots$. To see this, we only need to 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 $\bigsqcup Y = y$, so we have $f(x) \sqsubseteq \bigsqcup \{f(x), f(y)\} = f(\bigsqcup Y) = f(y)$.

We should also note that there is an alternative (more restrictive) definition of a CPO which uses the concept of a *directed set* (X is directed iff $x, y \in X \Rightarrow \exists z \in X.x, y \sqsubseteq z$) instead of a chain. This, in turn, leads to an alternative (more restrictive) definition of a 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 POs and for their domains.

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

PROPOSITION 2.1. *If D and E are CPOs, then $F = [D \to E]$ is a CPO under the relation*

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

PROOF. First, F is 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 $\bigsqcup Z \in F$. Define $\bigsqcup Z = \lambda x . \bigsqcup \{f(x) \mid f \in Z\}$. This is a well-defined function since, for each x in D, $\{f(x) \mid f \in Z\}$ is easily seen to be a chain in E. Next, it bounds above every $f \in Z$ since, for each $x \in D$, $f(x) \sqsubseteq \bigsqcup \{f(x) \mid f \in Z\}$. 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 $(\bigsqcup Z)(x) \sqsubseteq h(x)$, and hence $\bigsqcup Z \sqsubseteq h$.

But we must also show that $\bigsqcup Z \in F$, i.e., $\bigsqcup Z$ is *continuous*. Let $X \subseteq D$ be a chain. We require

$$(\bigsqcup Z)(\bigsqcup X) = \bigsqcup \{(\bigsqcup Z)(x) \mid x \in X\},$$

But

$$(\bigsqcup Z)(\bigsqcup X) = \bigsqcup\{f(\bigsqcup X) \mid f \in Z\} \quad \text{by the definition of } \bigsqcup Z$$
$$= \bigsqcup\{f(x) \mid f \in Z, x \in X\}$$
$$= \bigsqcup\{(\bigsqcup Z)(x) \mid x \in X\}.$$

This completes the proof. $\square$

PROPOSITION 2.2. *For any CPO D, every $f \in [D \to D]$ 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 $Yf \sqsubseteq x$.*

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

PROOF. The set $S = \{f^i(\bot_D) \mid 0 \le i\}$ is a chain by the monotonicity of f. Define $Yf = \bigsqcup S$. By the continuity of f, we have $f(Yf) = \bigsqcup\{f^{i+1}(\bot_D) \mid 0 \le i\} = Yf$, so Yf is a fixed point of f. Let x be any other fixed point. Now by the monotonicity of f we have $f(\bot_D) \sqsubseteq f(x) = x$ and, by induction on i, we can show $f^i(\bot_D) \sqsubseteq x$ for all $i \ge 0$, so $Yf = \bigsqcup\{f^i(\bot_D) \mid 0 \le i\} \sqsubseteq x$, and thus Yf is the minimum fixed point of f. $\square$

PROPOSITION 2.3. *Y is continuous, so $Y \in [[D \to D] \to D]$.*

PROOF. Let Z be any chain $\subseteq [D \to D]$. We must show that $Y(\bigsqcup Z) = \bigsqcup\{Yf \mid f \in Z\}$. In one direction ($\sqsupseteq$) the proof is easy since, for each $f \in Z$, $\bigsqcup Z \sqsupseteq f$, so $Y(\bigsqcup Z) \sqsupseteq Yf$ 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 $\bigsqcup\{Yf \mid f \in Z\}$ is a fixed point of $\bigsqcup Z$, since then it dominates the least such, which is $Y(\bigsqcup Z)$.

$$\bigsqcup Z(\bigsqcup\{Yf \mid f \in Z\}) = \bigsqcup\{g(\bigsqcup\{Yf \mid f \in Z\}) \mid g \in Z\}$$
$$= \bigsqcup\{g(Yf) \mid g \in Z, f \in Z\} \quad \text{by continuity of } g$$
$$= \bigsqcup\{f(Yf) \mid f \in Z\}$$
$$\text{since } g(Yf) \sqsubseteq h(Yh) \text{ where } h = \max(g, f)$$
$$= \bigsqcup\{Yf \mid f \in Z\}$$

which is the required fixed point property. This completes this proof. $\square$

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 CPOs, 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 λ-calculus.

3.1 Types

(1) *ind* and *tr* are (basic) types.
(2) If β_1, β_2 are types then $(\beta_1 \to \beta_2)$ is a type.
(3) These are all the types.

We use $\beta, \beta_1, \beta_2, \ldots$ to denote types, and frequently omit parentheses, assuming that '$\to$' associates to the right, so that $\beta_1 \to \beta_2 \to \beta_3$ abbreviates $(\beta_1 \to (\beta_2 \to \beta_3))$.

3.2 Terms

Each term has a well defined type. We use s, t, u to denote terms, and write $s : \beta$ 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 $\mathtt{TT} : tr$, $\mathtt{FF} : tr$ and the families (indexed by type) $\mathtt{UU}_\beta$, $\supset_{tr \to \beta \to \beta \to \beta}$ and $\mathsf{Y}_{(\beta \to \beta) \to \beta}$. 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 $s : \beta_1 \to \beta_2$ and $t : \beta_1$ are terms then $s(t) : \beta_2$ is a term. If $x : \beta_1$ is an identifier and $s : \beta_2$ is a term, then $[\lambda x \, . \, s] : \beta_1 \to \beta_2$ 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 $\supset(s)(t)(u)$ and $\mathsf{Y}([\lambda x \, . \, s])$ respectively as $(s \to t, u)$ and $[\alpha x . s]$, 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.

3.3 Semantics

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

$$tt \qquad ff$$
$$\searrow \quad \nearrow$$
$$\perp_{tr}$$
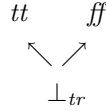
and $D_{\beta_1 \to \beta_2} = [D_{\beta_1} \to D_{\beta_2}]$. Note that D_{ind} completely determines a standard model.

Let $\mathcal{I}$ 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

$$\mathcal{A} : \mathcal{I} \to \bigcup \{D_\beta\}$$

which satisfies the further conditions

(1) $\mathcal{A}[\![x : \beta]\!] \in D_\beta$. We write the (syntactic) arguments of $\mathcal{A}$ in decorated brackets as an aid to the eye.

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

$$\mathcal{A}[\![\mathtt{TT}]\!] = tt,$$
$$\mathcal{A}[\![\mathtt{FF}]\!] = ff,$$
$$\mathcal{A}[\![\mathtt{UU}_\beta]\!] = \perp_\beta,$$
$$\mathcal{A}[\![\supset_{tr \to \beta \to \beta \to \beta}]\!] = \lambda \xi \in D_{tr} \, . \, \lambda \eta \in D_\beta \, . \, \lambda \chi \in D_\beta] \, . \, (\xi \to \eta, \chi), \text{ and}$$
$$\mathcal{A}[\![\mathsf{Y}_{(\beta \to \beta) \to \beta}]\!] = \mathsf{Y}_{(\beta \to \beta) \to \beta}$$

where $(\xi \rightarrow \eta, \chi)$, the conditional, takes the values $\perp, \eta, \chi$ according as $\xi = \perp_{tr}, tt, ff$, and where we have subscripted the fixed point operator Y on the right to indicate that it belongs to $[[D_\beta \rightarrow D_\beta] \rightarrow D_\beta]$. 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 $\mathcal{A}[\![\mathsf{Y}]\!]$ 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) it yields a *continuous* function over the appropriate domains.

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

$$\mathcal{A}[\![s(t)]\!] = \mathcal{A}[\![s]\!](\mathcal{A}[\![t]\!])$$
$$\mathcal{A}[\![\lambda x . s]\!] = \lambda \xi . \mathcal{A}_{\xi/x}[\![s]\!].$$

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.

PROPOSITION 3.1. *For each assignment $\mathcal{A}$ and for each $x : \beta_1, s : \beta_2$, we have* $\lambda \xi \in D_{\beta_1} . \mathcal{A}_{\xi/x}[\![s]\!] \in [D_{\beta_1} \rightarrow D_{\beta_2}]$.

PROOF. First, suppose s is an atomic term, i.e., an identifier. Either $s = x$, in which case $\beta_1 = \beta_2$ and $\lambda \xi . \mathcal{A}_{\xi/x}[\![s]\!]$ is the identity function over D_{β_1}, or $s \neq x$, in which case it is a constant function from D_{β_1} to D_{β_2}. In either case it is a continuous function, hence $\in [D_{\beta_1} \rightarrow D_{\beta_2}]$.

Next, suppose s is $t(u)$, $t : \beta_3 \rightarrow \beta_2$ and $u : \beta_3$. Assume the proposition for t and u. We have to show that, for any chain $X \subseteq D_{\beta_1}$,

$$\bigsqcup \{\mathcal{A}_{\xi/x}[\![t(u)]\!] \mid \xi \in X\} = \mathcal{A}_{\bigsqcup X/x}[\![t(u)]\!];$$

that is, that

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

Now if we denote $\lambda \xi . \mathcal{A}_{\xi/x}[\![t]\!]$ and $\lambda \xi . \mathcal{A}_{\xi/x}[\![u]\!]$ by f and g, the inductive assumption tells us that $f \in [D_{\beta_1} \rightarrow [D_{\beta_3} \rightarrow D_{\beta_2}]]$ and $g \in [D_{\beta_1} \rightarrow D_{\beta_3}]$, and the required equation merely states that, for such f and g, $\lambda \xi . f(\xi)(g(\xi))$ is continuous. The proof of this we leave to the reader; it is hardly more than proving that for a chain X, $\{f(\xi)(g(\xi)) \mid \xi \in X\}$ and $\{f(\xi)(g(\eta)) \mid \xi, \eta \in X\}$ 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

$$\lambda \xi \in D_{\beta_1} \, . \, \mathcal{A}_{\xi/x}[\![\lambda y \, . \, t]\!] \in [D_{\beta_1} \to [D_{\beta_3} \to D_{\beta_4}]]$$

that is, that, for any chain $X \subseteq D_{\beta_1}$,

$$\bigsqcup \{\lambda \eta \in D_{\beta_3} \, . \, (\mathcal{A}_{\xi/x})_{\eta/y}[\![t]\!] \mid \xi \in X\} = \lambda \eta \in D_{\beta_3} \, . \, (\mathcal{A}_{\bigsqcup X/x})_{\eta/y}[\![t]\!].$$

Now, in the case $x = y$, we have $(\mathcal{A}_{\xi/x})_{\eta/y} = (\mathcal{A}_{\bigsqcup X/x})_{\eta/y} = \mathcal{A}_{\eta/y}$ 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 $\lambda \xi \, . \, (\mathcal{A}_{\eta/y})_{\xi/x}[\![t]\!]$ is continuous, hence monotonic, so $\{(\mathcal{A}_{\xi/x})_{\eta/y}[\![t]\!]\}$ is a chain in D_{β_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 $\bigsqcup$ for function spaces (Proposition 2.1) we can replace the left-hand side of the desired equation by

$$\lambda \eta \in D_{\beta_3} \, . \, \bigsqcup \{(\mathcal{A}_{\eta/y})_{\xi/x}[\![t]\!] \mid \xi \in X\}$$
$$. \, \lambda \eta \in D_{\beta_3} \, . \, (\mathcal{A}_{\eta/y})_{\bigsqcup X/x}[\![t]\!]$$
$$. \, \lambda y \in D_{\beta_3} \, . \, (\mathcal{A}_{\bigsqcup X/x})_{\eta/y}[\![t]\!] \quad \text{since } x \neq y$$

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

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 λ-terms, the proposition gives the corollary directly. For an application term $s(t) : \beta$, the proposition tells us that $\lambda \xi \in D_{\beta_1} \, . \, \mathcal{A}_{\xi/x}[\![s(t)]\!] \in [D_{\beta_1} \to D_\beta]$ so, by application to $\mathcal{A}[\![x]\!]$, we get $\mathcal{A}[\![s(t)]\!] = \mathcal{A}_{\mathcal{A}[\![x]\!]/x}[\![s(t)]\!] \in D_\beta$ as required. $\square$

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.

4.1 Atomic well-formed formulae (AWFFs)

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

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

4.2 Well-formed formulae (WFFs)

A WFF is a set of AWFFs. We use $P, Q, P_1, Q_1, \ldots$ to denote arbitrary WFFs. Extend $\mathcal{A}$ to WFFs by

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

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

4.3 Sentences

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

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

We say that $P \vdash Q$ 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 $\mathcal{A}[\![A]\!]$ is independent of $\mathcal{A}[\![x]\!]$; more precisely, $\mathcal{A}_{\xi/x}[\![A]\!] = \mathcal{A}[\![A]\!]$. 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 $\mathcal{A}[\![A\{t/x\}]\!] = \mathcal{A}_{\mathcal{A}[\![t]\!]/x}[\![A]\!]$.

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

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

CONJ $\qquad\qquad\qquad\qquad\qquad \dfrac{P \vdash Q_1 \quad P \vdash Q_2}{P \vdash Q_1 \cup Q_2}$

Clearly valid.

CUT $\qquad\qquad\qquad\qquad\qquad \dfrac{P_1 \vdash P_2 \quad P_2 \vdash P_3}{P_1 \vdash P_3}$

Clearly valid.

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

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

REFL $\qquad\qquad\qquad\qquad\qquad \vdash s \subset s$

Clearly valid, by reflexivity of $\sqsubseteq$.

TRANS $\qquad\qquad\qquad\qquad s \subset t, t \subset u \vdash s \subset u$

Clearly valid, by transitivity of $\sqsubseteq$.

MIN1 $\qquad\qquad\qquad\qquad\qquad \vdash \mathtt{UU} \subset s$

Clearly valid, by minimality of $\perp_\beta$.

MIN2 $\qquad\qquad\qquad\qquad\qquad \vdash \mathtt{UU}(s) \subset \mathtt{UU}$

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 $\mathtt{UU}$, 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 $\mathtt{UU}_{\beta_1 \to \beta_2}(s : \beta_1) \subset \mathtt{UU}_{\beta_2}$. Similarly we will omit subscripts from $\supset$ and $\mathtt{Y}$.

CONDT $\qquad\qquad\qquad\qquad \vdash \supset(\mathtt{TT})(s)(t) \equiv s$

CONDU $\qquad\qquad\qquad\qquad \vdash \supset(\mathtt{UU})(s)(t) \equiv \mathtt{UU}$

CONDF $\qquad\qquad\qquad\qquad \vdash \supset(\mathtt{FF})(s)(t) \equiv t$

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

ABSTR $\qquad\qquad\qquad \dfrac{P \vdash s \subset t}{P \vdash [\lambda x \; . \; s] \subset [\lambda x \; . \; t]} \quad x$ not free in P

Let $\mathcal{A}$ be such that $\mathcal{A}[\![P]\!] = \mathrm{T}$. Since x is not free in P, we have also $\mathcal{A}_{\xi/x}[\![P]\!] = \mathrm{T}$ 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 $\mathcal{A}[\![[\lambda x \; . \; s] \subset [\lambda x \; . \; t]]\!] = \mathrm{T}$ as required.

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

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

ETACONV $\qquad\qquad \vdash [\lambda x \; . \; y(x)] \equiv y \quad y$ distinct from x

$\mathcal{A}[\![[\lambda x \; . \; y(x)]]\!] = \lambda \xi \; . \; \mathcal{A}_{\xi/x}[\![y(x)]\!] = \lambda \xi \; . \; \mathcal{A}_{\xi/x}[\![y]\!](\mathcal{A}_{\xi/x}[\![x]\!]) = \lambda \xi \; . \; \mathcal{A}[\![y]\!](\xi)$ (since x is distinct from y, so does not occur free in y) $= \mathcal{A}[\![y]\!]$.

CASES $\qquad\qquad \dfrac{P, s \equiv \mathtt{TT} \vdash Q \quad P, s \equiv \mathtt{UU} \vdash Q \quad P, s \equiv \mathtt{FF} \vdash Q}{P \vdash Q}$

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

FIXP $$\vdash \mathrm{Y}(x) \equiv x(\mathrm{Y}(x))$$

Clearly valid by the standard interpretation of Y.

INDUCT $$\frac{P \vdash Q\{\mathrm{UU}/x\} \quad P \cup Q \vdash Q\{s(x)/x\}}{P \vdash Q\{\mathrm{Y}(s)/x\}} \quad x \text{ not free in } P \text{ or } s$$

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}[\![t']\!] = \mathcal{A}[\]\!]$, y distinct from x, and then x is not free in $[\lambda y . t'\{y/x\}]$. Let $\mathcal{A}$ be a standard assignment, $\mathcal{A}[\![P]\!] = \mathrm{T}$, and assume that $\mathcal{A}[\![s]\!] = f$, $\mathcal{A}[\![t]\!] = g$, $\mathcal{A}[\![u]\!] = h$. We first show by induction on i that, for each $i \geq 0$, $g(f^i(\bot_\beta)) \sqsubseteq h(f^i(\bot_\beta))$, where $x : \beta$. For $i == 0$, the first hypothesis gives that $\mathcal{A}_{\bot_\beta/x}[\![Q]\!] = \mathrm{T}$, that is $\mathcal{A}[\![t]\!](\bot_\beta) \sqsubseteq \mathcal{A}[\![u]\!](\bot_\beta)$ (since x is not free in t, u), so $g(\bot_\beta) \sqsubseteq h(\bot_\beta)$. Now assume the inequality for i. That is, we assume $\mathcal{A}_{f^i(\bot_\beta)/x}[\![Q]\!] = \mathrm{T}$. Since x is not free in P, we also have $\mathcal{A}_{f^i(\bot_\beta)/x}[\![P]\!] = \mathrm{T}$, and we deduce from the second hypothesis that $\mathcal{A}_{f^i(\bot_\beta)/x}[\![Q\{s(x)/x\}]\!] = \mathrm{T}$. Now $\mathcal{A}_{f^i(\bot_\beta)}[\![s(x)]\!] = f(f^i(\bot_\beta))$, since x is not free in s, $= f^{i+1}(\bot_\beta)$ so, from the second fact which we assumed for assignments, we deduce that $\mathcal{A}_{f^{i+1}(\bot_\beta)/x}[\![Q]\!] = \mathrm{T}$, that is $g(f^{i+1}(\bot_\beta)) \sqsubseteq h(f^{f+1}(\bot_\beta))$. So the induction is complete. Now $\mathcal{A}[\![Q\{\mathrm{Y}(s)/x\}]\!] = \mathcal{A}_{\mathrm{Y}(f)/x}[\![Q]\!]$, which we require to take the value T. That is, we require $g(\mathrm{Y}(f)) \sqsubseteq h(\mathrm{Y}(f))$. But $g(\mathrm{Y}(f)) = \bigsqcup\{g(f^i(\bot_\beta)) \mid i \geq 0\}$ (by the continuity of g), $\sqsubseteq \bigsqcup\{h(f^i(\bot_\beta)) \mid i \geq 0\}$ (by what we have proved), $\sqsubseteq h(\mathrm{Y}(f))$ by the monotonicity of h, and the justification is complete.

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

5. 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.