II. SYNTAX, SEMANTICS AND PROOF THEORY FOR EQUATIONAL LOGIC

1. A Formal Syntax for Expressions

Given an alphabet Σ, the traditional syntax of Σ-terms is developed as follows:

Definition 1

Let $\rho:\Sigma \to \mathbb{N}$ be a rank function assigning to each character $a \in \Sigma$ a nonnegative integer ρa called the rank of a.

(1) If $\rho a = 0$, then a is a Σ-term.
(2) If $\rho b = n > 0$, and if $A_1,\dots,A_n$ are Σ-terms, then $b(A_1,\dots,A_n)$ is a Σ-term.
(3) Nothing is a Σ-term unless required to be by (1) and (2).

The notion of expressions as Σ-terms is very convenient for traditional logic, but not for studies of reduction. For the study of reduction, we need a way to distinguish different parts or subexpressions even if they look the same. For example, if we apply the equation schema $x*(y+z) = (x*y)+(x*z)$ to reduce the expression $A = 1*(2+1)$ to $B = (1*2)+(1*1)$, it is sometimes important to capture formally the intuitive notion that the leftmost two occurrences of 1 in B correspond to the leftmost occurrence in A, while the rightmost occurrence of 1 in B corresponds to the rightmost occurrence in A. To formalize such distinctions we will define expressions as labeled trees, following Rosen [Ro73] and Brainerd [Br69].

Definition 2

$\mathbb{P}$ is the set of positive integers $\{1,2,3,\dots\}$

$\mathbb{P}^*$ is the set of finite sequences of positive integers, such as $(1,7,32,2,5)$.

An element of $\mathbb{P}^*$ may be thought of as a tree address, where the empty string, $()$, is the address of the root, and for each $x \in \mathbb{P}^*$ and $i \in \mathbb{P}$, $x\cdot(i)$ is the address of the ith son of x (for $x,y \in \mathbb{P}^*$, $x\cdot y$ denotes x concatenated with y).

Definition 3 [Ro73] (Def.4.2, p.167), [Br69] (Def.2.2, p.218)

A tree domain is a finite subset $D \subseteq \mathbb{P}^*$ such that $() \in D$ and,

For all $x \in \mathbb{P}^*$, $i \in \mathbb{P}$
$x\cdot(i) \in D \Rightarrow x \in D$,
$x\cdot(i+1) \in D \Rightarrow x\cdot(i) \in D$.

A tree domain is, intuitively, the set of addresses in some tree. The definition above merely insures that there is a root address, (), and that every nonroot address $x\cdot(i)$ has a father x, and that if x has an (i+1)st son $x\cdot(i+1)$, then x also has an ith son, $x\cdot(i)$. For instance, D = {(), (1), (2), (2,1), (2,2), (2,3)} is a tree domain.

Tree domains will be drawn as graphs with the root, (), at the top and descending arcs representing the son relations. D above is shown in Figure 1.

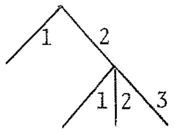

Figure 1

Any tree address in D may be read by following a path from the root to a node, putting the integers on traversed arcs into a sequence.

Definition 4 [Ro73] (Def.4.3, p.167), [Br69] (Def.2.4, p.218)

Let Σ be any set.

A Σ-tree is a function $A:D \to \Sigma$ where $D \subseteq P^*$ is a tree domain.

$\Sigma_* = \{A \mid A \text{ is a } \Sigma\text{-tree}\}$.

Thinking of Σ as a set of labels, a tree is a function from tree addresses to labels. Σ-trees will be drawn by attaching characters in Σ to the addresses of an appropriate tree domain.

Example 1

Let $\Sigma = \mathbb{N} \cup \{a,b,c,\ldots,z\}$.

Let D = {(), (1), (2), (1,1), (2,1), (2,2), (2,3)}.

A = {<(), f>, <(1), 0>, <(2), g>, <(2,1), x>, <(2,2), 1>, <(2,2), 3>} is a Σ-tree with domain D.

A is shown in Figure 2.

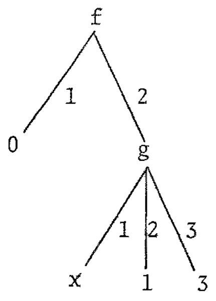

Figure 2

For the purposes of this work, expressions are Σ-trees. Every Σ-term may be represented straightforwardly by a Σ-tree. For instance, the tree in Example 1 represents the term f(0,g(x,1,3)). The notion of occurrences of subexpressions is captured by the use of tree addresses, which may be distinguished even if they are at the roots of subtrees which look the same. Notational conventions for describing trees will be:

Definition 5 [Ro73] (Defs. 4.1, 4.4, 4.5, 4.10, pp.167,168), [Br69] (Defs. 2.7, 2.9, p.219)

A-L vary over Σ_* or $\Sigma_\#$.

a-h vary over Σ.

i-n vary over $\mathbb{N}$ (the nonnegative integers $\{0,1,2,\ldots\}$).

p-z vary over $\mathcal{P}^*$ (tree addresses).

Functional application of f to x is written fx.

$x \cdot y$ means x concatenated with y.

anc is the ancestor (initial substring) relation on $\mathcal{P}^*$,
  x anc y iff $\exists_{w \in \mathcal{P}^*}\ x \cdot w = y$.

anc$\neq$ is the proper ancestor relation,
  x anc$\neq$ y iff x anc y and $x \neq y$.

$\perp$ is the independence relation on $\mathcal{P}^*$,
  $x \perp y$ iff $\neg(x$ anc $y)$ and $\neg(y$ anc $x)$.

$\perp S$ iff $\forall_{x,y \in S}\ x \perp y \vee x = y$.
  Read "$\perp S$" as "S is an independent set."

$S \perp T$ iff $\forall_{x \in S, y \in T}\ x \perp y$.
  Read "$S \perp T$" as "S and T are mutually independent."

domain is the domain operation on functions,
  domainA = $\{x \mid Ax$ is defined$\}$.

If D is a tree domain, then
  $\mathcal{P}^\perp D = \{S \mid S \subseteq D \wedge \perp S\}$.

$\mathcal{P}^\perp(\mathcal{P}^*) = \{S \mid S \subseteq \mathcal{P}^* \wedge \perp S \wedge S$ is finite$\}$.

M-Z vary over $\mathcal{P}^\perp(\mathcal{P}^*)$.

/ is the subtree operator.
  If A is a tree and $x \in$ domainA $\subseteq \mathcal{P}^*$, then
  $A/x = \lambda y.\ A(x \cdot y)$; i.e., A/x is the subtree of

A rooted at the address x.

$A(x\leftarrow B) = \lambda y.$ <u>if</u> $y=x\cdot z$ <u>then</u> Bz <u>else</u> Ay;

i.e., $A(x\leftarrow B)$ is the tree resulting from replacing the subtree at x in A by B.

If $M\in P^{\perp}$(<u>domain</u>A), then $A(x\leftarrow B_x |\ x\in M)$ is the tree obtained from A by replacing each subtree at $x\in M$ by the corresponding B_x. Formally:

$A(x\leftarrow B_x |\ x\in\phi) = A$

$A(x\leftarrow B_x |\ x\in M\cup\{y\}) = A(x\leftarrow B_x |\ x\in M)(y\leftarrow B_y)$.

This notation is well-defined since independent (in the sense of $\perp$) replacements are order-independent.

If $a\in\Sigma$, and $A_1,\ldots,A_n$ are Σ-trees,

then $a(A_1,\ldots,A_n) = \lambda y.$ <u>if</u> $y=()$ <u>then</u> a
<u>else</u> $A_i z$ where $y = (i)\cdot z$;

i.e., $a(A_1,\ldots,A_n)$ is the Σ-tree with a at the root and the subtrees $A_1,\ldots,A_n$ directly below, in order.

The character $a\in\Sigma$ will also be used to denote the one-node Σ-tree

$\lambda y.$ <u>if</u> $y=()$ <u>then</u> a <u>else</u> undefined.

Thus, each Σ-term may be interpreted as a Σ-tree. $\Sigma_{\#}$ is the set of Σ-trees which represent Σ-terms. The special brackets <> and [] will sometimes be used instead of () to improve readability.

A = B may refer to the equation or it may assert the equality of A and B. The correct meaning should be obvious from the context. E.g., $A=B \in \mathcal{A}$ refers to the equation A = B, while $A = B(x\leftarrow C)$ asserts equality. To simplify reading, equations may be enclosed in pointed brackets, e.g., $\langle A=B\rangle \in \mathcal{A}$.

<u>Example 2</u>

Let $\Sigma = \{0,s,p,Cond,F\}$, where

0 represents the usual integer,

s and p represent the successor and predecessor operations on non-negative integers,

Cond represents the special conditional function defined by

$Cond(x,y,z) =$ <u>if</u> $x=0$ <u>then</u> y <u>else</u> z,

F represents an undetermined binary function to be defined recursively.

$\rho 0 = 0$

$\rho s = \rho p = 1$

$\rho F = 2$

$\rho Cond = 3$

Then $\Sigma_{\#}$ includes

F(0,s(0))

Cond(0,0,F(p(0),F(0,s(0))))

Σ_{*} also contains nonsensical objects such as

F(0)

s(p)

0(F(p,s), Cond(F,p)).

The Σ-trees above are shown in Figure 3.

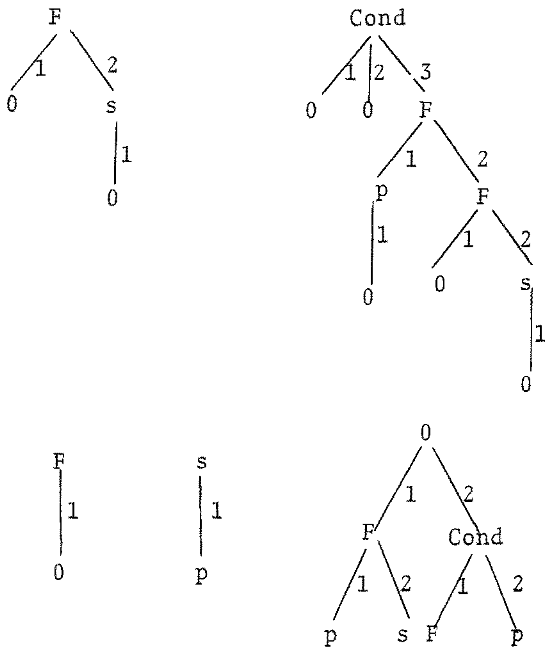

Figure 3

2. Semantics for Expressions and Equations

The traditional semantic treatment of expressions in $\Sigma_{\#}$ is based on the notion of an interpretation of Σ.

Definition 6

An interpretation of Σ is a pair

$$\mathbb{I} = \langle D,v \rangle$$

where D is any set, and v is a function

$$v:\Sigma \to \bigcup_{n\in\mathbb{N}} (D^n \to D)$$

such that, for all $a \in \Sigma$,

$$va \in D^{\rho a} \to D.$$

D is called the domain of $\mathbb{I}$.

v is called a valuation of Σ.

Arbitrary elements in a domain D will be named by underlined lower case letters $\underline{a}$, $\underline{b}$, $\underline{c}$, $\underline{d}$, ...

An interpretation gives meaning to each character in Σ. Now v may be extended to $\hat{v}:\Sigma_{\#} \to D$ straightforwardly.

Definition 7

Let $I = \langle D,v \rangle$ be an interpretation of Σ.
The valuation of $\Sigma_{\#}$ induced by I is the function $\hat{v}:\Sigma_{\#} \to D$ defined by:

$$\hat{v}a = va \text{ if } \rho a = 0$$
$$\hat{v}(b(A_1,\dots,A_n)) = (vb)(\hat{v}A_1,\dots,\hat{v}A_n) \quad \text{if } \rho b = n \text{ and } A_1,\dots,A_n \in \Sigma_{\#}.$$

$\hat{v}$ gives meaning to each expression in $\Sigma_{\#}$. A slight modification of Definitions 6 and 7 provides semantics for all expressions in Σ_{*}.

Definition 8

A pseudointerpretation of Σ is a pair

$$I = \langle D,v \rangle$$

where D is any set, and v is a function

$$v:\Sigma \to (D^* \to D)$$

The valuation of Σ_{*} induced by I is the function $\hat{v}:\Sigma_{*} \to D$ defined by:

$$\hat{v}a = (va)()$$
$$\hat{v}(b(A_1,\dots,A_n)) = (vb)(\hat{v}A_1,\dots,\hat{v}A_n).$$

We will not distinguish between interpretations and pseudointerpretations, since all our results about one will hold for the other.

The important semantic question about an equation A = B is: "Is this equation true?" An equation is true for an interpetation precisely when the value of each side is the same.

Definition 9

Let $I = \langle D,v \rangle$ be a (pseudo) interpretation of Σ.
For all $A,B \in \Sigma_{\#}$ (Σ_{*})

$$I \models A = B \text{ iff } \hat{v}A = \hat{v}B$$

$I \models A = B$ is read "I satisfies A = B" or
"A = B is valid (true) for I."

$\models$ is usually extended in several ways, as follows:
Let $\mathcal{I}$ be a set of interpretations of Σ.
$\mathcal{I} \models A=B$ iff $\mathbb{I} \models A=B$ for all $\mathbb{I} \in \mathcal{I}$.
$C=D \models A=B$ iff $\{\mathbb{I} \mid \mathbb{I} \models C=D\} \models A=B$.
Let $\mathcal{A}$ be a set of equations.
$\mathbb{I} \models \mathcal{A}$ iff $\forall_{A=B \in \mathcal{A}}\ \mathbb{I} \models A=B$,
$\mathcal{A} \models A=B$ iff $\{\mathbb{I} \mid \mathbb{I} \models \mathcal{A}\} \models A=B$.
$\mathcal{A} \models A=B$ is read "A=B is a consequence of $\mathcal{A}$".
$\models A=B$ iff $\mathbb{I} \models A=B$ for all interpretations $\mathbb{I}$.
$\models A=B$ is read "A=B is <u>valid</u>."

The simplicity of the above semantic definitions and their close correspondence to the intuitive meanings of equations are great strengths of equational logic. The main purpose of formalizing the definitions here is to provide a precise terminology for dealing with the correctness and completeness of proof techniques and computational strategies. Although this work is not directly concerned with expressions containing variable symbols, it is convenient to have semantics for such expressions.

<u>Definition 9</u>

Let $\langle D,v\rangle$ be an interpretation of Σ.
Let $\mathbb{V}$ be a set of variable symbols, $\mathbb{V} \cap \Sigma = \phi$, $\rho a = 0$ for $a \in \mathbb{V}$.
Let $A,B \in (\Sigma \cup \mathbb{V})_{\#}$.
$\mathbb{I} \models A=B$ iff $\mathbb{I}' \models A=B$
for all $\mathbb{I}' = \langle D,v'\rangle$ where v' extends v to $\Sigma \cup \mathbb{V}$.

<u>Example 3</u>

A natural interpretation of Σ in Example 2 (p. 7) is $\langle\mathbb{N},v\rangle$ where
$\mathbb{N}$ is the set of nonnegative integers and
$v0$ = the number zero
$vs = \lambda n.\ n+1$
$vp = \lambda n.$ <u>if</u> $n=0$ <u>then</u> 0 <u>else</u> $n-1$
$v\,\mathrm{Cond} = \lambda i,j,k.$ <u>if</u> $i=0$ <u>then</u> j <u>else</u> k
$vF = \lambda m,n.\ 0$

With v defined above, and variable symbols x,y,z,
$\langle\mathbb{N},v\rangle \models p(0) = 0$
$\langle\mathbb{N},v\rangle \models p(s(x)) = x$
$\langle\mathbb{N},v\rangle \models \mathrm{Cond}(0,x,y) = x$
$\langle\mathbb{N},v\rangle \models \mathrm{Cond}(s(x),y,z) = z$
$\langle\mathbb{N},v\rangle \models F(x,y) = 0$
$\langle\mathbb{N},v\rangle \models F(x,y) = \mathrm{Cond}(x,0,F(p(x),F(x,y)))$

$\langle \mathbb{N}, v\rangle \models F(s(0),0) = 0$

3. Equational Proofs

Proofs in equational logic (without variable symbols) are generally performed using rules of inference corresponding to the reflexive, symmetric, transitive and substitution laws for equality:

1. Reflexivity $\dfrac{}{A = A}$

2. Symmetry $\dfrac{A = B}{B = A}$

3. Transitivity $\dfrac{A=B,\ B=C}{A=C}$

4. Substitution $\dfrac{A=B}{C(x\leftarrow A) = C(x\leftarrow B)}$

For proofs and computations in this work, an axiom with variable symbols may always be treated as a schema representing all of the axioms (without variables) which result from replacing those variables by expressions.

Definition 10

Let $\mathcal{A}$ be a set of equations.

A proof $\mathcal{D}$ of $A=B$ from $\mathcal{A}$ is a finite sequence of equations ending in $A=B$ such that, if $C=D$ is in $\mathcal{D}$, then either

(1) $C=D \in \mathcal{A}$ or

(2) $C=D$ is derived from (0 or more) previous equations in $\mathcal{D}$ by one application of one of the rules 1,2,3,4.

$\mathcal{A} \vdash A=B$ means that there exists a proof of $A=B$ from $\mathcal{A}$.

$\vdash A=B$ means $\phi \vdash A=B$.

$\mathcal{A} \overset{'}{\vdash} A=B$ means that there exists a proof of $A=B$ from $\mathcal{A}$ which does not require rule 4.

Theorem 1 (Correctness and Completeness)

$\mathcal{A} \models A=B$ iff $\mathcal{A} \vdash A=B$

$\models A=B$ iff $\vdash A=B$

Proof $\mathcal{A} \vdash A=B \Rightarrow \mathcal{A} \models A=B$ by induction on the length of a proof of $A=B$. The converse is proved by constructing an interpretation $\langle D,v\rangle$ defined by

$D = \{\{B \mid \mathcal{A} \vdash A=B\} \mid A \in \Sigma_*\}$

$(va)(\underline{d}_1,\ldots,\underline{d}_n) = \{B \mid \mathcal{A} \vdash a(D_1,\ldots,D_n) = B\}$ where $\forall_i D_i \in \underline{d}_i$.

v is easily proved well-defined.

By induction on the structure of $A \in \Sigma_*$,

$\hat{v}A = \{B \mid A \vdash A{=}B\}$.

So $\langle D,v\rangle \models A{=}B$ iff $A \vdash A{=}B$.

$\langle D,v\rangle \models A$ is straightforward.

So, by Definition 9 (p. 9), $A \models A{=}B \Rightarrow A \vdash A{=}B$. $\models A{=}B \Leftrightarrow \vdash A{=}B$ is merely the special case where $A = \phi$. □

The correctness and completeness theorem shows a great strength of equational proof techniques - they are sufficient to prove all the consequences of a set of equational axioms. In order to develop computational strategies, we would like to simplify the notion of proof even more. Rule 4 (substitution) is the most complicated. Substitution cannot be eliminated, but it may be isolated at the beginning of a proof.

Lemma 1

Let $A' = \{E(x{\leftarrow}C) = E(x{\leftarrow}D) \mid C \in \Sigma_*,\ x \in \underline{\text{domain}}E,\ C = D \in A\}$.

$A \vdash A{=}B$ iff $A' \overset{1}{\vdash} A{=}B$.

Proof Note that A' is the set of all equations which follow from A by one application of rule 4. We may reorder a proof so that all applications of rule 4 occur at the beginning. □

With substitution out of the way, we may reinterpret proofs in a form more suitable for discussing computational strategies.

Lemma 2

Define $\rightarrow$ by

$A \rightarrow B$ iff $\exists_{x\in\underline{\text{domain}}A, C\in\Sigma}\ A/x{=}C \in A \wedge B = A(x{\leftarrow}C)$

Let $\equiv$ be $(\rightarrow \cup \leftarrow)^*$, the reflexive, symmetric, transitive closure of $\rightarrow$.

Then $A \models A{=}B$ iff $A \vdash A{=}B$ iff $A \equiv B$.

Proof Consider A' as in Lemma 1. $A \rightarrow B$ iff $A{=}B \in A'$. Now, since rules 1, 2, 3 state exactly the reflexive, symmetric and transitive properties of equality, we may show that $A' \overset{1}{\vdash} A{=}B$ iff $A \equiv B$. Then by Theorem 1 and Lemma 1, $A \models A{=}B$ iff $A \vdash A{=}B$ iff $A \equiv B$. See [CF58] (Th.1, p.60) for another treatment. □

Our computation theory for equational logic is based on the study of the relation $\rightarrow$. Note that, given an expression E_0, all the expressions F such that $A \models E_0 = F$ may be generated by creating sequences (E_i) such that, for each i, either $E_i \rightarrow E_{i+1}$ or $E_i \leftarrow E_{i+1}$. The study of Subtree Replacement Systems will show, in many cases, that we need consider only the case $E_i \to E_{i+1}$, and that we may effectively choose an appropriate E_{i+1} given E_i.

Example 4

Consider $\Sigma = \{0,s,p,Cond,F\}$ of Example 2 (p. 7), with the interpretation $<\mathbb{N},v>$ of Example 3 (p.10).

Let $A = \{p(s(A)) = A,$
$Cond(0,A,B) = A,$
$Cond(s(A),B,C) = C,$
$F(A,B) = Cond(A,0,F(p(A),F(A,B)))\mid A,B,C \in \Sigma_\#\}$.

A contains only equations which are valid for $<\mathbb{N},v>$.

$F(s(0),0) = 0$ may be proved from A as follows:

$F(s(0),0) = Cond(s(0), 0, F(p(s(0)), F(s(0),0)))$	A
$Cond(s(0), 0, F(p(s(0)), F(s(0),0))) = F(p(s(0)), F(s(0),0))$	A
$F(s(0),0) = F(p(s(0)), F(s(0)),0))$	Trans.
$p(s(0)) = 0$	A
$F(p(s(0)), F(s(0),0)) = F(0, F(s(0),0))$	Subst.
$F(s(0),0) = F(0, F(s(0),0))$	Trans.
$F(0, F(s(0),0)) = Cond(0, 0, F(p(0), F(0, F(s(0),0))))$	A
$F(s(0),0) = Cond(0, 0, F(p(0), F(0, F(s(0),0))))$	Trans.
$Cond(0, 0, F(p(0), F(0, F(s(0),0)))) = 0$	A
$F(s(0),0) = 0$	Trans.

$F(s(0),0) \to^* 0$ as shown by the following sequence:

$$\begin{aligned} F(s(0),0) &\to Cond(s(0), 0, F(p(s(0)), F(s(0),0)) \\ &\to F(p(s(0)), F(s(0),0)) \\ &\to F(0, F(s(0),0)) \\ &\to Cond(0, 0, F(p(0), F(s(0),0)))) \\ &\to 0. \end{aligned}$$

*4. Continuous Semantics for Equations

A more specialized type of semantics is currently popular in theoretical computer science. This alternate semantics, like the traditional logical semantics, assigns to each expression a value in some domain. The values assigned are intended to represent the available information about an expression. Domains are partially ordered according to information content.

Definition 11

Let D be a set and $\sqsubseteq$ a partial ordering of D.

A chain is a set $C \subseteq D$ such that

$$\forall_{\underline{a},\underline{b}\in C}\ \underline{a} \sqsubseteq \underline{b} \vee \underline{b} \sqsubseteq \underline{a}$$

D is a <u>Complete Partial Ordering</u> (CPO) if every chain $C \subseteq D$ has a least upper bound, written $\sqcup C$.

Intuitively, $\underline{a} \sqsubseteq \underline{b}$ (read "$\underline{a}$ is less defined than $\underline{b}$") means that $\underline{b}$ represents all the information in $\underline{a}$, and possibly some more. A chain is a set of objects with increasing information, so the existence of least upper bounds for chains guarantees the ability to collect all the information presented by an increasing sequence of objects.

Note that $\sqcup\phi$ must be a minimum element of D. $\sqcup\phi$ is often called "bottom" or "undefined", since it represents a complete lack of information. In other literature, $\sqcup\phi$ is generally written $\perp$, but we use ω to avoid confusion with the independence relation. In a CPO D, subsets $C \subseteq D$ may not have greatest lower bounds, but, when they do, we write $\sqcap C$.

<u>Lemma 3</u>

Let D be a CPO.
Let $C \subseteq D$ be such that $\forall_{\underline{a},\underline{b}\in C}\ \exists_{\underline{c}\in C}\ \underline{a} \sqsubseteq \underline{c} \wedge \underline{b} \sqsubseteq \underline{c}$.
(C is called a <u>directed set</u>.)
Then C has a least upper bound, written $\sqcup C$.

<u>Proof</u> See [Ma76] (Corollary 1, p.55) □

<u>Definition 12</u>

Let D be a CPO.
The ordering $\sqsubseteq$ extends to D^* as
$(\underline{a}_i) \sqsubseteq (\underline{b}_i)$ iff $\forall_i\ \underline{a}_i \sqsubseteq \underline{b}_i$
and to $D^* \to D$ and $\Sigma \to (D^* \to D)$ as
$f \sqsubseteq g$ iff $\forall_{(\underline{a}_i)}\ f(\underline{a}_i) \sqsubseteq g(\underline{a}_i)$
$v \sqsubseteq w$ iff $\forall_a\ va \sqsubseteq wa$
D^*, $D^* \to D$ and $\Sigma \to (D^* \to D)$ are CPOs with the relation $\sqsubseteq$.
A function $f:D \to E$ is <u>continuous</u> iff for all nonempty chains
$C \subseteq D$,
$f(\sqcup C) = \sqcup\{f(\underline{a}_i) \mid (\underline{a}_i) \in C\}$.

Continuity means that our information about a function value is proportional to our information about the arguments of the function. Most work on continuous semantics is based on the following well-known Lemma.

<u>Lemma 4</u> (Least Fixpoints) [Ma76] (Th.9, p.65)

Let D be a CPO, f a continuous function $f:D \to D$.

f has a least fixpoint $\underline{\mathrm{lfp}}\,f \in D$, which is the least element such that $f(\underline{\mathrm{lfp}}\,f) = \underline{\mathrm{lfp}}\,f$.

$$\begin{aligned}\underline{\mathrm{lfp}}\,f &= \sqcap\{\underline{d} \mid f\underline{d} = \underline{d}\}\\ &= \sqcup\{f^n\omega \mid n \in \mathbb{N}\}\end{aligned}$$

Proof $\{f^n\omega \mid n \in \mathbb{N}\}$ is a chain, so $\sqcup\{f^n\omega \mid n \in \mathbb{N}\}$ exists. It is straightforward to show that the limit of this chain is the least fixpoint of f. □

Definition 13

A continuous interpretation of Σ is a pair $I = \langle D,v\rangle$ such that

D is a CPO and v is a function

$v: \Sigma \to (D^* \to D)$ such that,

for all $a \in \Sigma$, va is continuous.

v extends to $\hat{v}: \Sigma_* \to D$ just as in Definition 8.

It is convenient to extend $\hat{v}$ further to $\hat{v}:(\Sigma \cup D)_\# \to D$ with the conventions that $\underline{d} \in D$ is a constant symbol ($\rho\underline{d} = 0$) and $\hat{v}\underline{d} = \underline{d}$.

The relation $\models$ follows just as in Definition 9.

Some people [Sc70] insist that D be a complete lattice (every set $C \subseteq D$ has a least upper bound and a greatest lower bound). Every complete lattice is a CPO, so all of this discussion applies also to complete lattices.

Example 5

Let $\Sigma = \{0,s,p,\mathrm{Cond},F\}$ as in Example 2.

Let $D = \mathbb{N}^+ = \mathbb{N} \cup \{\omega\}$, and define $\sqsubseteq$ by $i \sqsubseteq j$ iff $i=j \vee i=\omega$.

Let $v0 = 0$

$$(vs)i = \begin{cases} i+1 & \text{if } i \in \mathbb{N}\\ \omega & \text{if } i=\omega\end{cases}$$

$$(vp)i = \begin{cases} i-1 & \text{if } i \in \mathbb{N}-\{0\}\\ 0 & \text{if } i=0\\ \omega & \text{if } i=\omega\end{cases}$$

$$(v\,\mathrm{Cond})(i,j,k) = \begin{cases} j & \text{if } i=0\\ k & \text{if } i \in \mathbb{N}-\{0\}\\ \omega & \text{if } i=\omega\end{cases}$$

$$(vF)(i,j) = \omega$$

Then the continuous interpretation $\langle D,v\rangle$ represents the fact that we can compute the successor, predecessor and conditional functions on $\mathbb{N}$, but we don't yet know what F is.

A richer interpretation of Σ uses the domain $D' = P\mathbb{N}$, with $\underline{a} \sqsubseteq \underline{b}$ iff $\underline{b} \subseteq \underline{a}$.

Note that $\omega = \mathbb{N}$.

Let $v'0 = \{0\}$

$(v's)\underline{a} = \{i+1 \mid i \in \underline{a}\}$

$(v'p)\underline{a} = \{\underline{\max}(i-1,0) \mid i \in \underline{a}\}$

$$(v'\mathrm{Cond})(\underline{a},\underline{b},\underline{c}) = \begin{cases} \underline{b} \text{ if } \underline{a} = \{0\} \\ \underline{c} \text{ if } 0 \notin \underline{a} \neq \phi \\ \phi \text{ if } \underline{a} = \phi \\ \underline{b} \cup \underline{c} \text{ otherwise} \end{cases}$$

$(v'F)(\underline{a},\underline{b}) = \mathbb{N}$

The continuous interpretation $\langle D',v'\rangle$ represents a stronger understanding of s,p and Cond. Notice that $\langle D',v'\rangle \models \mathrm{Cond}(s(A),B,C) = C$ for all $A,B,C \in \Sigma_{\#}$, but $\langle D,v\rangle \models \mathrm{Cond}(s(A),B,C) = C$ only when $\hat{v}A \neq \omega$.

So far, continuous semantics seems to be merely a complicated special case of classical semantics. With the new semantics we may treat a set of equations A as a set of definitions, instead of a set of assertions as in the traditional view. Rather than investigating which equations are consequences of A, we use continuous semantics to study the way in which A extends our information about expressions.

Consider a continuous interpretation $\langle D,v\rangle$ of Σ, representing preliminary information about Σ, and a set of equations, A, intended to supply additional information about Σ. We would like to represent the resulting state of knowledge about Σ by a new interpretation $\langle D,v^A\rangle$, where $v \sqsubseteq v^A$ and $\langle D,v^A\rangle \models A$. v^A would extend, as usual, to a valuation $\hat{v}^A$ of Σ_* or $\Sigma_{\#}$.

If D is a complete lattice, then an appropriate v^A may always be found. In other cases there may be no reasonable choice.

<u>Definition 14</u>

$$v^A = \sqcup\{w \mid \langle D,w\rangle \models A \text{ and } v \sqsubseteq w\}$$

whenever such a greatest lower bound exists.

When v^A exists, $\hat{v}^A$ represents all the information about expressions which was given by $\hat{v}$, plus all the additional information which may be derived from A. Given preliminary information v and equations A, a reasonable strategy for discovering information about an expression A is to collect all the information about expressions B such that $A \vdash A=B$. In reasonable systems, this strategy should produce the same information as $v^{\mathcal{A}}$. We prove such a result here for complete lattices, and in Chapter VIII for CPOs with special sets $\mathcal{A}$ representing recursive equations.

<u>Definition 15</u>

Given axioms $\mathcal{A}$ and continuous interpretation $\langle D,v \rangle$,

$$\underline{\text{def}}\ A = \sqcup\{\hat{v}B \mid \mathcal{A} \vdash A{=}B\}$$

whenever such a least upper bound exists.

<u>Theorem 2</u>

Assume that $v^{\mathcal{A}}$ and $\underline{\text{def}}$ A exist.

(1) Then $\underline{\text{def}}\ A \sqsubseteq \hat{v}^{\mathcal{A}}A$.

(2) If, in addition, there is a continuous interpretation w such that $v \sqsubseteq w$ and $\hat{w}A = \underline{\text{def}}\ A$, then $\hat{v}^{\mathcal{A}}A = \underline{\text{def}}\ A$.

<u>Proof</u>

(1) Since $v \sqsubseteq v^{\mathcal{A}}$, $\hat{v} \sqsubseteq \hat{v}^{\mathcal{A}}$.
Let $\mathcal{A} \vdash A{=}B$.
Then $\mathcal{A} \models A{=}B$, so $\hat{v}^{\mathcal{A}}A = \hat{v}^{\mathcal{A}}B$.
Thus, $\hat{v}B \sqsubseteq \hat{v}^{\mathcal{A}}A$.
Therefore, $\underline{\text{def}}\ A = \sqcup\{\hat{v}B \mid \mathcal{A} \vdash A{=}B\} \sqsubseteq \hat{v}^{\mathcal{A}}A$.

(2) Note that $\langle D,w \rangle \models \mathcal{A}$ and $v \sqsubseteq w$.
So $v^{\mathcal{A}} \sqsubseteq w$.
Therefore, $\hat{v}^{\mathcal{A}}A \sqsubseteq \hat{w}A = \underline{\text{def}}\ A$. □

<u>Corollary 1</u>

If D is a complete lattice, then $v^{\mathcal{A}}$ and $\underline{\text{def}}$ A exist, and $\hat{v}^{\mathcal{A}}A = \underline{\text{def}}\ A$ for all $A \in \Sigma_*$.

<u>Proof</u> Define w by

$$(wc)(\underline{d}_1,\ldots,\underline{d}_n) = \sqcup\{\underline{\text{def}}(c(D_1,\ldots,D_n)) \mid \forall_i\ \underline{\text{def}}\ D_i \sqsubseteq \underline{d}_i\} \sqcup \underline{(vc)}(\underline{d}_1,\ldots,\underline{d}_n)$$

$v \sqsubseteq w$ and $\hat{w}A = \underline{\text{def}}\ A$.
So, by Theorem 2, $\hat{v}^{\mathcal{A}}A = \underline{\text{def}}\ A$. □

Continuous semantics is usually applied to situations where Σ may be partitioned into primitive symbols Σ_p and defined symbols Σ_d. v gives meaning only to the primitive symbols, so $va = \omega$ for $a \in \Sigma_d$, and $\mathcal{A}$ adds meaning only to the defined symbols, so $v^{\mathcal{A}}b = vb$ for $b \in \Sigma_p$. In such situations, the equations $\mathcal{A}$ are said to define symbols in Σ_d in terms of those in Σ_p. See Section VIII.4 on recursive equations for an example of defining Σ_d from Σ_p.