THE SEMANTICS OF NONDETERMINISM

by

T.S.E. Maibaum

Research Report CS-77-30

Department of Computer Science

University of Waterloo
Waterloo, Ontario, Canada

December 1977

§0 Introduction

One of the most intriguing topics in Mathematical semantics in the last few years has been that of non-determinism. Although very few existing languages allow non-determinism, the study of such languages is not without merit. For example, any language which deals with relations as opposed to functions (e.g. query languages for relational data bases) must be in some way nondeterministic. In [4], Dijkstra has introduced a non-deterministic language which he claims facilitates the synthesis of programs. Moreover, many authours have studied parallelism by using the concept of non-determinism.

Non-determinism means, of course, allowing some element of chance to influence how a computation might proceed. As a first approach, we might introduce a choice construct "or" into a simple language of recursive definitions. As in [17], these recursive definitions give rise to evaluation sequences and the application of the evaluation mechanism to a "program segment" T_1 or T_2 would result in a random choice to evaluate either T_1 or to evaluate T_2.

The relevant domains of interpretation for these recursive definitions are non-deterministic domains or structures which are special instances of a class of domains suggested in [5]. The elements of this idea appear in [13] and [6] and were formally pointed out in [8], [9]. A structure is an element of a restricted class of complete partially ordered sets (cpo's). The restriction is a consequence of requiring that we not only have the usual so-called "computational partial order" ($\sqsubseteq$) on data domains, we also order domains by a so-called "results partial order".

The reasoning behind the choice of this restricted class is explained as follows: We assume that our machine is equipped with basic functions which are deterministic (i.e. return at most one output when given some input). The non-determinism results from having a choice construct in a programming language. Any given execution of a nondeterministic program P will result in a deterministic computation. However, many different computations may be executions of P and these computations (call them C_p) may or may not be comparable using the usual ordering of computations. The result of executing P could be the output of any of these computations C_p. What could be the output of an execution of P_1 or P_2? The output could be an output of an execution of P_1 or an output of an execution of P_2. Thus (informally) result (P_1 or P_2) = join (result (P_1), result (P_2)) where join: (sets of results)$^2 \to$ (sets of results). Moreover, even if the computations of P_1 and P_2 (on the same input) are not comparable (using the "computational partial order"), we may be able to show that result (P_1) approximates result (P_2) with respect to the join operation indicated above.

Another important problem in studying computations is how to construct function spaces of given domains. For example, if D is a cpo, then $[D \to D]$ is the set of continuous functions from D to D and is easily shown to be a cpo. The fact that D and $[D \to D]$ have similar properties as domains is vital in studying deterministic computations. Since we restrict the class of cpo's we may use in studying nondeterministic computations, do we also need to restrict the class of functions we allow in order to maintain these special properties? The answer is of course in the affirmative: given a structure D, we let [D,D] be the class of functions which are continuous with respect to the computational partial order and monotonic with respect to the results partial order. This reflects the intuitive idea that if we give "more" inputs to a nondeterministic program, then we should expect "more" outputs.

As to the contents of the paper, in Section 1, we outline some underlying mathematical ideas. In Section 2, we study the class of structures and show that there is a universal structure; that is, we show that there is a domain in which nondeterministic programs can be given meaning symbolically and that interpretations of this symbolic meaning in other structures are consistent with the meaning of these programs in these structures. In Section 3, we show that these ideas can be generalised to give definitions of nondeterministic programs of higher type: i.e. nondeterministic functionals.

§1 Basic Definitions and Ideas

Let S be a set of sorts. A many-sorted alphabet Σ is an indexed family of sets $\{\Sigma_{w,s}\}_{<w,s>\in S^*xS}$ indexed by S^*xS. Given $f \in \Sigma_{w,s}$, f is said to be of type $<w,s>$, arity w, sort s, and rank $|w|$ (where $|w|$ is the length of the string w). f is called an operator symbol or name.

A many sorted Σ-algebra (or Σ-algebra) A_Σ is an indexed family of sets $A = \{A_s\}_{s\in S}$ and an indexed family of assignments $\alpha = \{\alpha_{w,s}\}_{<w,s>\in S^*xS}$ where $\alpha_{w,s}: \Sigma_{w,s} \to (A_{s_1} x \ldots xA_{s_n} \to A_s)$ for $w = s_1 \ldots s_n$ and $(A_{s_1} x \ldots xA_{s_n} \to A)$ the set of functions from $A_{s_1} x \ldots xA_{s_n}$ to A_s. A is said to be the carrier of A_Σ and A_s is the carrier of sort s. We denote $\alpha_{w,s}(f)$ by f_A.

Given $w \in S^*$, we will assume $w = s_1 \ldots s_n$ unless stated otherwise. We will denote $A_{s_1} x \ldots xA_{s_n}$ by A^w. Let A_Σ and B_Σ be Σ-algebras. A Σ-homomorphism $h: A_\Sigma \to B_\Sigma$ is an indexed family of mappings $\{h_s\}_{s\in S}$ such that $h_s: A_s \to B_s$ and for $f \in \Sigma_{w,s}$, $<a_1, \ldots ,a_n> \in A^w$,

$$h_s(f_A(a_1, \ldots ,a_n)) = f_B(h_{s_1}(a_1), \ldots ,h_{s_n}(a_n)).$$

A Σ-homomorphism which is injective, surjective, or bijective is said to be a Σ-monomorphism, Σ-epimorphism, or Σ-isomorphism, respectively.

Let C be a class of Σ-algebras. A Σ-algebra A_Σ is said to be initial in the class C if and only if $A_\Sigma \in C$ and for each $B_\Sigma \in C$ there exists a unique Σ-homomorphism $h : A_\Sigma \to B_\Sigma$. Let Alg_Σ be the class of all Σ-algebras. Then $\underline{Alg_\Sigma}$ has an initial algebra T_Σ defined as follows:

Let T_Σ be the family of sets $\{T_{\Sigma,s}\}_{s\in S}$ defined by:

(i) $\Sigma_{\lambda,s} \subseteq T_{\Sigma,s}$ for each $s \in S$;

(ii) Given $f \in \Sigma_{w,s}$ and $\langle t_1, \ldots, t_n\rangle \in T_\Sigma{}^w$, $ft_1 \ldots t_n \in T_{\Sigma,s}$.

For $f \in \Sigma_{w,s}$ and $\langle t_1, \ldots, t_n\rangle \in T_\Sigma{}^w$, define

$$f_{T_\Sigma}(t_1, \ldots, t_n) = ft_1 \ldots t_n.$$

(As in [13], T_Σ is classically known as the word algebra on the alphabet Σ.)

Let $X = \{X_s\}_{s\in S}$ be an indexed family of sets of "variables". Define $\Sigma(X)$ by $\Sigma(X)_{\lambda,s} = \Sigma_{\lambda,s} \cup X_s$, $\Sigma(X)_{w,s} = \Sigma_{w,s}$ for $w \neq \lambda$. We will generally write $T_{\Sigma(X)}$ as $T_\Sigma(X)$ and then T_Σ is isomorphic to $T_\Sigma(\{\emptyset\}_{s\in S})$. Let A_Σ be a Σ-algebra and let $a: X \to A$ be an <u>assignment</u> of values to variables. Then a extends in a unique way to a Σ-homomorphism $\bar{a}$: $T_\Sigma(X) \to A_\Sigma$. This is a very important result and is proved by showing how A_Σ can be "made into a $\Sigma(X)$ -algebra". (The assignment to Σ is as before and the assignment to X is via the assignment a.) For further details, see [1]. Let $X_w = \{x_{1,s_1}, \ldots, x_{n,s_n}\}$ where x_{i,s_i} is of sort s_i. Define $(X_w)_s$ by $(X_w)_s = \{x_{i,s_i} | s_i = s\}$. Then let $T_\Sigma(X_w) = T_\Sigma(\{(X_w)_s\}_{s\in S})$. The motivation for defining X_w and $T_\Sigma(X_w)$ can be found in [13] or [1].

A <u>congruence</u> q on A_Σ is an indexed family of equivalence relations $q = \{q_s\}_{s\in S}$, q_s defined on A_s, with the <u>substitution property</u>. That is, given $f \in \Sigma_{w,s}$ and $\langle a_1, \ldots, a_n\rangle, \langle b_1, \ldots, b_n\rangle \in A^w$ such that $a_i q_{s_i} b_i$ for $1 \leq i \leq n$, we have $f_A(a_1, \ldots, a_n) q_s f_A(b_1, \ldots, b_n)$. Denote by $[a]_q$ (or just $[a]$) the equivalence class of a. Let A/q be the family of sets

$$\{\{[a] | a \in A_s\}_s\}_{s\in S}.$$

We can make A/q into a Σ-algebra A_Σ/q by defining for $f \in \Sigma_{w,s}$ and $\langle [a_1], \ldots ,[a_n] \rangle \in (A/q)^w$,

$$f_{A/q}([a_1], \ldots ,[a_n]) = [f_A(a_1, \ldots ,a_n)].$$

This definition is independent of representatives from equivalence classes because of the substitution property. Moreover, there exists a unique homomorphism $h: A_\Sigma \to A_\Sigma/q$ defined by $h_s(a) = [a]$ for $a \in A_s$.

A Σ-equation is a pair $\langle t_1,t_2 \rangle \in T_\Sigma(X_w)_s \times T_\Sigma(X_w)_s$ for some $s \in S$. We usually write $t_1 = t_2$ instead of $\langle t_1,t_2 \rangle$. A Σ-algebra A_Σ is said to satisfy $t_1 = t_2$ if for all assignments $a: X_w \to A^w$, $\bar{a}(t_1) = \bar{a}(t_2)$. Define the relation $R^A_{t_1 = t_2}$ on A_Σ by:

$$\bar{a}(t_1) R_{t_1 = t_2} \bar{a}(t_2) \text{ for some assignment } a: X_w \to A.$$

Given $R_{t_1 = t_2}$, there exists a least congruence $q_{t_1 = t_2}$ on T_Σ containing $R_{t_1 = t_2}$. $t_1 = t_2$ is said to generate $q_{t_1 = t_2}$. A system of Σ-equations E is an indexed family of sets of Σ-equations $E = \{E_s\}$. As above, E generates a least congruence q_E on A_Σ.

Let $\underline{Alg}_{\Sigma,E}$ be the class of Σ-algebras satisfying E. Then $\underline{Alg}_{\Sigma,E}$ has an initial algebra, namely T_Σ/q_E. (Note that, in general, initial algebras are unique up to isomorphism. That is, if A_Σ and B_Σ are initial in C, then A_Σ is isomorphic to B_Σ.) This result is very important and is the basis of much of the work on abstract data types ([2], [12]).

Let A be a partially ordered set (poset) with partial ordering $\sqsubseteq_A$. If A has a minimal element, then it is denoted by $\bot_A$. A non-empty subset A' of A is an ω-chain (or chain) if and only if A' is totally ordered. We will normally indicate the ordering of the chain explicitly by writing $\{a_i\}$ for $a_0 \sqsubseteq_A a_1 \sqsubseteq_A a_2 \sqsubseteq_A \cdots$. A is said to be an ω-complete (or complete) partially ordered set (cpo) if A has a minimal element and every chain in A has a least upper bound. ($A' \subseteq A$ has a least upper bound b if and only if for each $a \in A'$, $a \sqsubseteq_A b$ and if there exists any $b' \in A$ such that for each $a \in A'$, $a \sqsubseteq_A b'$, then $b \sqsubseteq_A b'$.) We denote the least upper bound of the chain $\{a_i\}$ by $\bigsqcup_{i>0} a_i$ or $\bigsqcup a_i$.

If A and B are posets, then $f: A \to B$ is said to be continuous if $f(\bigsqcup a_i) = \bigsqcup f(a_i)$ for any chain $\{a_i\}$ in A when $\bigsqcup a_i$ exists in A. Note that on the left of the equation, the least upper bound is taken with respect to $\sqsubseteq_A$, whereas on the right it is taken with respect to $\sqsubseteq_B$.

If A, B are posets, then so is $A \times B$ with partial order $\sqsubseteq_{A \times B}$ defined by $(a,b) \sqsubseteq_{A \times B} (a',b')$ if and only if $a \sqsubseteq_A a'$, $b \sqsubseteq_B b'$. If A, B are complete, then so is $A \times B$ (with minimal element $(\bot_A, \bot_B)$).

Let A_Σ be a Σ-algebra. A_Σ is said to be continuous if each A_s is complete (with minimal element denoted by $\bot_s$) and if for each $f \in \Sigma_{w,s}$, $f_A: A^w \to A_s$ is continuous. (Note that A^w is complete by the remark above.) That is, for $\{<a_{s_{1,i}}, \ldots, a_{s_{n,i}}>\}$ a chain in A^w,

$$\bigsqcup f_A(a_{s_{1,i}}, \ldots, a_{s_{n,i}}) = f_A(a_{s_1}, \ldots, a_{s_n}) \text{ where } a_{s_j} = \bigsqcup a_{s_{j,i}} \text{ for}$$

$1 \leq j \leq n$. Note that to demonstrate the continuity of f_A, it is sufficient to show that f_A is continuous with respect to each of its arguments separately. That is, if $\{a_{j,i}\}$ is a chain in A_{s_i}, then it is sufficient to show that for each i, $1 \le i \le n$, $\bigsqcup f_A(a_{s_1}, \ldots, a_{j,i}, \ldots, a_{s_n}) = f_A(a_{s_1}, \ldots, \bigsqcup a_{j,i}, \ldots, a_{s_n})$. A Σ-homomorphism $h: A_\Sigma \to B_\Sigma$ between the continuous Σ-algebras A_Σ and B_Σ is <u>continuous</u> if and only if each $h_s: A_s \to B_s$ is continuous as a function. h is said to be <u>strict</u> if $h_s((\bot_s)_A) = (\bot_s)_B$; i.e., h maps minimal elements to corresponding minimal elements. Let <u>$CAlg_\Sigma$</u> be the class of continuous Σ-algebras together with strict, continuous Σ-homomorphisms between them. Then <u>$CAlg_\Sigma$</u> has an initial algebra which can be defined as follows.

Let $D = [A \text{-}{\circ}\text{->} B]$ be the set of partial functions from the set A to the set B. We can order the elements of D by set inclusion on the graphs of the functions in D (with the graphs considered as subsets of $A \times B$). It is well known that with this ordering D is a cpo. For $f \in D$, let $def(f) = \{a \mid \langle a,b\rangle \in f\}$. $def(f)$ is the <u>domain of definition</u> of f. Let Σ denote (ambiguously) the family $\{\Sigma_{w,s}\}_{\langle w,s\rangle \in S^* \times S}$ and the set formed by taking the disjoint union of the family Σ.

For each $s \in S$, let $CT_{\Sigma,s}$ be the set of partial functions $t: N^* \to \Sigma$ (N the set of natural numbers) such that

(i) If $\lambda \in def(t)$, then $t(\lambda)$ has sort s;

(ii) If $w \in N^*$, $i \in N$, and $wi \in def(t)$, then

(a) $w \in def(t)$ and

(b) if $t(w)$ has arity $s_1 \ldots s_n$, then $i < n$ and $t(wi)$ has sort s_{i+1}.

Let $CT_\Sigma = \{CT_{\Sigma,s}\}_{s\in S}$ and define f_{CT_Σ} for $f \in \Sigma_{w,s}$ by:

(i) If $w = \lambda$ (and so f is a constant or a nullary) then

$f_{CT_\Sigma} = \{\langle\lambda,f\rangle\}$;

(ii) If $t_i \in CT_{\Sigma,s_i}$ for $1 \le i \le n$, then

$$f_{CT_\Sigma}(t_1, \ldots, t_n) = \{\langle\lambda,f\rangle\} \cup \bigcup_{i<n} \{\langle iu,g\rangle \mid \langle u,g\rangle \in t_{i+1}\}.$$

Thus CT_Σ is a Σ-algebra and can in fact be shown to be a continuous Σ-algebra and is initial in $\underline{CAlg}_\Sigma$ ([1]). The order on CT_Σ is that obtained by restricting the order on $[N^*\text{-}\circ\text{->}\Sigma]$ to elements of CT_Σ.

The above is an awkward formalism in which to think about the initial algebra in $\underline{CAlg}_\Sigma$ and so we introduce a more intuitive one. Let $\perp = \{\{\perp_s\}\}_{s\in S}$ and consider $\Sigma(\perp)$. (See above for definition of $\Sigma(X)$.) Consider $CT'_{\Sigma,s}$ to be defined by:

(i) $\Sigma(\perp)_{\lambda,s} \subseteq CT'_{\Sigma,s}$;

(ii) If $f \in \Sigma(\perp)_{w,s}$ and t_i is a finite or infinite polish prefix expression over $\Sigma(\perp)$ already in CT'_{Σ,s_i} for $1 \le i \le n$, then $ft_1 \ldots t_n \in CT'_{\Sigma,s}$.

Let $CT'_\Sigma = \{CT'_{\Sigma,s}\}_{s\in S}$. We can order $CT_{\Sigma,s}$ by: $t \sqsubseteq_{CT'} t'$ if and only if

(i) $t = t'$;

or (ii) $t = \perp_s$;

or (iii) $t = ft_1 \ldots t_n$, $t' = ft'_1 \ldots t'_n$ and $t_i \sqsubseteq_{CT'} t'_i$ for $1 \le i \le n$.

It is straightforward to show that CT_Σ and CT'_Σ are isomorphic and we will generally think of elements of CT_Σ to be polish prefix expressions (or trees denoted by polish prefix expressions).

Let $E = \{E_s\}_{s \in S}$ be a system of Σ-equations over $\Sigma(\bot)$ such that for each $t_1 = t_2$ in E_s, $t_1, t_2 \in CT_{\Sigma,s}$. Congruences over continuous Σ-algebras are defined as before, as are the concepts of satisfiability and the least congruence generated by a system of Σ-equations. Given congruence q over CT_Σ, although the algebra CT_Σ/q is well defined, it is not in general complete. (i.e., the carriers of CT_Σ/q are not cpo's as some chains do not have least upper bounds.) However, the particular application we have in mind will turn out to be complete and we will need the following concept to prove this: Let A be a poset and $R \subseteq A \times A$ a relation on A. Suppose we have chains $\{a_i\}$ and $\{b_i\}$ such that for all i, $a_i R b_i$ and moreover $\bigsqcup a_i$ and $\bigsqcup b_i$ exist. If $\bigsqcup a_i \, R \bigsqcup b_i$ for all such chains $\{a_i\}$ and $\{b_i\}$, then R is said to be <u>$\sqsubseteq_A$-complete</u>.

§2 Σ - structures and Solutions of Equations

Let Σ be a many-sorted alphabet sorted by S such that:

(i) $+ \in \Sigma_{ss,s}$ for each $s \in S$,

and (ii) $\perp_s \in \Sigma_{\lambda,s}$ for each $s \in S$.

$+$ will be used to denote a (finite) join operation with respect to the so-called subset partial order. $\perp_s$ is used to denote the bottom element of the carrier of sort s of a given continuous Σ-algebra. Consider the cpo CT_Σ with the usual order $\sqsubseteq_C$. Define the relation $\subseteq_C$ on CT_Σ as follows:

$t \subseteq_C t'$ for $t,t' \in CT_{\Sigma,s}$ if and only if

(i) $t = t'$,

or (ii) $t' = +t_1t_2$ and $t \subseteq_C t_1$, or $t \subseteq_C t_2$,

or (iii) $t' = ft'_1 \dots t'_n$, $t = ft_1 \dots t_n$ and $t_i \subseteq_C t'_i$ for $1 \le i \le n$.

<u>Lemma 2.1</u>: $\subseteq_C$ is $\sqsubseteq_C$-complete. That is, if $\{t_i\}$, $\{t'_i\}$ are chains so that for each i, $t_i \subseteq_C t'_i$, then $\sqcup t_i \sqsubseteq_C \sqcup t'_i$.

<u>Proof</u>: Consider some $t_k \in \{t_i\}$. Since $t_k \subseteq_C t'_k$, one can perform the following sequence of steps on nodes of t'_k, starting at the root, in order to transform it into t_k:

(i) Let the node of t'_k being considered be labelled by $+$ and let the subtree whose root is this node be $+t_1t_2$.

(ii) There is a corresponding subtree of t_k, denoted by r, such that

(a) $r = +r_1 r_2$ and $r_i \subseteq_C t_i$ for $i = 1, 2$;

or (b) $r \subseteq_C t_1$;

or (c) $r \subseteq_C t_2$.

In case (a), perform no transformation at node labelled by +. In case (b) (symmetrically (c)), replace the subtree $+t_1 t_2$ by t_1 (symmetrically t_2).

(iii) Proceed to step (i) at the next node to be considered.

Call the above sequence of choices sq_k. We can think of applying sq_k to any tree t such that $t'_k \sqsubseteq_C t$. We need this restriction on t to ensure that sq_k is well-defined on t.

In particular for all $j > k$, we have that $t'_k \sqsubseteq_C t'_j$ and so we can conclude that

$$t_k = sq_k(t'_k) \sqsubseteq sq_k(t'_j).$$

Since this is true for every $j > k$, we must then have

$$t_k = sq_k(t'_k) \sqsubseteq sq_k(\bigsqcup_{j>k} t'_j) = sq_k(\sqcup t'_i) \qquad (*)$$

since $\bigsqcup_{j>k} t'_j = \sqcup t'_i$.

Since $t_{k+1} \subseteq_C t'_{k+1}$, we have $t_{k+1} = sq_{k+1}(t'_{k+1})$ for some sequence of choices sq_{k+1}. But $t_k \sqsubseteq_C t_{k+1}$ and $t'_k \sqsubseteq t'_{k+1}$ and so $t_k = sq_k(t'_k) \sqsubseteq sq_{k+1}(t'_{k+1}) = t_{k+1}$. In fact $t_k \sqsubseteq_C sq_j(\sqcup t'_i)$ for all $j \geq k$. Thus there must exist a sequence of choices sq such that $sq_j \leq sq$ for all j and $t_j = sq_j(t'_j) \sqsubseteq_C sq(\sqcup t'_i)$. ($sq_j \leq sq$ means that sq_j is a prefix of the sequence sq.)

But then $\sqcup t_j = \sqcup sq_j(t_j') \sqsubseteq_C sq(\sqcup t_i')$. Suppose that $\sqcup t_i \neq sq(\sqcup t_i')$. Now clearly $sq(t_j') = sq_j(t_j') \sqsubseteq_C \sqcup t_i$ for each j. ($sq(t_j')$ makes sense only if we assume that the choices specified by sq at non-existent nodes of t_j' are not used in obtaining $sq(t_j')$.) So we have $\sqcup sq(t_j') \sqsubseteq_C \sqcup t_i$ and, finally, since $\sqcup sq(t_j') = sq(\sqcup t_j')$, that $\sqcup t_i \sqsubseteq_C sq(\sqcup t_i')$. Thus $\sqcup t_i \subseteq_C \sqcup t_i'$.□

Consider CT_Σ/q where q is the least congruence generated by the family of sets of equations

$$E = \{\{+xy = +yx,\ +x+yz = ++xyz,\ +xx = x\}\}_{s \in S}.$$

(These axioms specify the properties of the join operation.) Denote CT_Σ/q by NCT_Σ. Define the relation $\sqsubseteq_N$ on NCT_Σ by: $[t] \sqsubseteq_N [t']$ if and only if $t \sqsubseteq_C r$ for some $r \in [t']$. This definition is independent of the choice of representatives from $[t]$ or $[t']$ for if $\bar{t}$ is any other representative of $[t]$, then $\bar{t}$ can be transformed to t by some sequence of transformations using only the axioms, r can be transformed to $\bar{r}$ using the reverse of the above sequence, and clearly $\bar{t} \sqsubseteq_C \bar{r}$.

<u>Lemma 2.2</u>: $\sqsubseteq_N$ is a partial order on NCT_Σ.

<u>Proof</u>: The only property which is not trivial to check is antisymmetry. So let $[t] \sqsubseteq_N [t']$ and $[t'] \sqsubseteq_N [t]$. Then there exist $r \in [t]$ and $r' \in [t']$ such that $t \sqsubseteq_C r' \sqsubseteq_C r$. Thus $t \sqsubseteq_C r$ and r is obtained from t by replacing some occurrences of $\perp$ in t by some expressions. But $r \in [t]$ and so r must be obtainable from t by use only of the axioms. Clearly $+xy = +yx$ and $+x+yz = ++xyz$ do not lend themselves to this process since the use of either on t gives t' such that $t \not\sqsubseteq_C t'$ and

$t' \not\sqsubseteq_C t$. The only possibility is to use $x = +xx$ repeatedly to replace a subexpression $\bot$ of t by $+\bot\bot$ to obtain r. Since $t \sqsubseteq_C r' \sqsubseteq_C r$, r' is obtained from t by making only some of these replacements. Then using $+xx = x$ on r', we obtain $r'' = t$ and of course $r'' \in [t']$. But then $[t] = [t']$ since they have an element in common. □

Lemma 2.3: NCT_Σ is complete.

Proof: Clearly $[\bot_s]$ is the least element of $NCT_{\Sigma,s}$. Let $\{[t_i]\}$ be a chain in NCT_Σ. Clearly the t_i can be chosen so that $t_i \sqsubseteq_C t_{i+1}$ for all i. Let $t = \sqcup t_i$ and consider $[t]$. Clearly $[t_i] \sqsubseteq_N [t]$ for all i, so $[t]$ is an upper bound for $\{[t_i]\}$. To show that $[t] = \sqcup[t_i]$, assume that there exists $[\bar{t}]$ so that for all i, $[t_i] \sqsubseteq_N [\bar{t}] \not\sqsubseteq_N [t]$. Thus for each t_i, there is $\bar{r}_i \in [\bar{t}]$ so that $t_i \sqsubseteq_C \bar{r}_i$. Since $t_i \sqsubseteq_C t_{i+1}$, we can choose a single representative, say $\bar{t}$, of $[\bar{t}]$ so that $t_i \sqsubseteq_C \bar{t}$. But then $\sqcup t_i = t \sqsubseteq_C \bar{t}$ and so $[t] \sqsubseteq_N [\bar{t}]$. This is a contradiction and so $[t]$ is the least upper bound of $\{[t_i]\}$. □

Lemma 2.4: NCT_Σ is a continuous Σ-algebra (with respect to the partial order $\sqsubseteq_N$).

Proof: Let $\{[t_i]\}$ be a chain in NCT_Σ. Then $f_N([t^{(1)}], \ldots, \sqcup[t_i], \ldots, [t^{(n)}])$

$$= [f_{CT}(t^{(1)}, \ldots, t, \ldots, t^{(n)}]$$

(where $t \in \sqcup[t_i]$)

$$= [f_{CT}(t^{(1)}, \ldots, \sqcup t_i, \ldots, t^{(n)})]$$

(where $t = \sqcup t_i$, $t_i \sqsubseteq_C t_{i+1}$ for all i)

$$= [\sqcup f_{CT}(t^{(1)}, \dots, t_i, \dots, t^{(n)})]$$

(since f_{CT} is continuous)

$$= \sqcup[f_{CT}(t^{(1)}, \dots, t_i, \dots, t^{(n)})]$$

(assuming $\sqcup[s_i] = [\sqcup s_i]$)

$$= \sqcup f_N([t^{(1)}], \dots, [t_i], \dots, [t^{(n)}]).$$

The proof that $\sqcup[s_i] = [\sqcup s_i]$ is the subject of the next lemma. □

Lemma 2.5: Let $\{s_i\}$ be a chain in CT_Σ. Then $\sqcup[s_i] = [\sqcup s_i]$.

Proof: Let $s = \sqcup s_i$. Then $[\sqcup s_i] = [s]$. We have shown above that $\sqcup[s_i] = [t]$ where t is the limit of a chain of representatives. Clearly $[t]$ is independent of the choice of chains of representatives from $\{[s_i]\}$ and so $[t] = [s]$. □

We now define the relation $\subseteq_N$ on NCT_Σ by: $[t] \subseteq_N [t']$ if and only if $t \subseteq_C s$ for some $s \in [t']$.

Lemma 2.6: $\subseteq_N$ is a partial order on NCT_Σ.

Proof: Reflexivity and transitivity are clear. Antisymmetry follows from an analysis similar to that found in Lemma 2.2. □

The join of $[t]$ and $[t']$ with respect to $\subseteq_N$ is defined by $+_N([t],[t']) = [+tt']$. This definition is clearly independent of representatives chosen.

Lemma 2.7: $\subseteq_N$ is $\sqsubseteq_N$-complete.

Proof: Let $\{[t_i]\}, \{[t_i']\}$ be chains in NCT_Σ such that:

(i) $\{t_i\}$ is a chain in CT_Σ;

(ii) $[t_i] \subseteq_N [t_i']$ for all i.

From property (ii), there exist $s_i \in [t_i']$ for all i so that $t_i \subseteq_C s_i$. The s_i can clearly be chosen in such a way that $s_0 \sqsubseteq_C s_1 \sqsubseteq_C s_2 \sqsubseteq_C \cdots$. Thus $\sqcup t_i \subseteq_C \sqcup s_i$ by lemma 2.1. But $[\sqcup t_i] = \sqcup[t_i]$ and $[\sqcup s_i] = \sqcup[s_i](=\sqcup[t_i'])$ by lemma 2.5, and so $\sqcup[t_i] \subseteq_N \sqcup[t_i']$. □

<u>Lemma 2.8</u>: Let $f \in \Sigma_{w,s}$ for some $\langle w,s\rangle \in S^*xS$. Then f_N is $\subseteq_N$-monotonic.

<u>Proof</u>: Suppose $[t_i] \subseteq_N [t_i']$ for some i, $1 \le i \le n$.

Then $f_N([t_1], \ldots, [t_i], \ldots, [t_n])$

$= [f_C(t_1, \ldots, t_i, \ldots, t_n)]$ (definition of f_N)

$= [ft_1 \ldots t_i \ldots t_n]$ (definition of f_C)

$\subseteq_N [ft_1 \ldots t_i' \ldots t_n]$ (by definition of $\subseteq_N$ in terms of $\subseteq_C$)

$= f_N([t_1], \ldots, [t_i'], \ldots, [t_n])$ (by definition of f_N) □

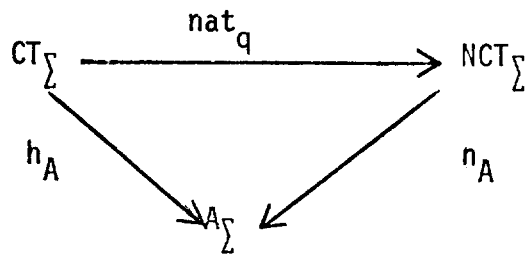
We can summarise the above in the following important result:

<u>Theorem 2.9</u>: NCT_Σ is a Σ-structure.

<u>Proof</u>: By lemmas 2.2, 2.3, 2.6, 2.7 the set NCT_Σ is a structure. Lemmas 2.4 and 2.8 indicate that NCT_Σ is indeed a Σ-structure. □

We now want to prove that NCT_Σ is initial in the class of Σ-structures. Let <u>Str_Σ</u> be the class of Σ-structures together with strict, $\sqsubseteq$-continuous, and $\subseteq$-monotonic Σ-homomorphisms.

Consider the diagram



where: (i) h_A: $CT_\Sigma \to A_\Sigma$ is the unique homomorphism guaranteed by the initiality of CT_Σ in $\underline{CAlg}_\Sigma$;

(ii) $nat_q : CT_\Sigma \to NCT_\Sigma$ is the natural homomorphism mapping t to its equivalence class $[t]$;

(iii) n_A: $NCT_\Sigma \to A_\Sigma$ is a mapping defined by $n_A([t]) = h_A(t)$.

Note that n_A is well defined since if $t, t' \in [t]$, then t' is transformable into t using only the axioms, A_Σ is in $\underline{Str}_\Sigma$ and so obeys the axioms, and so h_A must identify the images of t and t'.

We now proceed to prove that n_A is a unique, $\sqsubseteq$-continuous, $\subseteq$-monotonic homomorphism. In order to prove the $\subseteq$-monotonicity of n_A, we need the following auxilliary result.

<u>Lemma 2.10</u>: Let $A_\Sigma \in \underline{Str}_\Sigma$ and h_A: $CT_\Sigma \to A_\Sigma$. h_A preserves $\subseteq$.

<u>Proof</u>: The proof proceeds by induction on the structure of t' for $t \subseteq_C t'$. If $t' = a \in \Sigma_{\lambda,s}$, then $t = a$ and clearly $h_A(t) \subseteq_N h_A(t')$. If $t' = ft_1' \,.....\, t_n'$, then $t = ft_1 \ldots t_n$ and $t_i \subseteq_C t_i'$ for $1 \leq i \leq n$.

So we assume $h_A(t_i) \sqsubseteq_N h_A(t_i')$ for $1 \le i \le n$ and then

$$\begin{aligned} h_A(t) &= h_A(ft_1 \ldots t_n) \\ &= f_A(h_A(t_1), \ldots, h_A(t_n)) && (h_A \text{ is a } \Sigma\text{-homomorphism}) \\ &\sqsubseteq_A f_A(h_A(t_1'), \ldots, h_A(t_n')) && (\text{by induction since } f_A \text{ is } \subseteq\text{-monotonic}) \\ &= h_A(ft_1' \ldots t_n') && (h_A \text{ is a } \Sigma\text{-homomorphism}) \\ &= h_A(t'). \end{aligned}$$

If $t' = +t_1't_2'$, then either

(i) $t \sqsubseteq_C t_1'$ (symmetrically t_2'), in which case

$$\begin{aligned} h_A(t) &\sqsubseteq_A h_A(t_1') && (\text{by induction}) \\ &\sqsubseteq_A +_A(h_A(t_1'), h_A(t_2')) && (+_A \text{ is a join}) \\ &= h_A(+t_1't_2') && (h_A \text{ is a } \Sigma\text{-homomorphism}) \\ &= h_A(t') \end{aligned}$$

or (ii) $t = +t_1t_2$ and $t_i \sqsubseteq_C t_i'$ for $i = 1,2$, in which case

$$\begin{aligned} h_A(t) &= h_A(+t_1t_2) \\ &= +_A(h_A(t_1), h_A(t_2)) && (h_A \text{ is a } \Sigma\text{-homomorphism}) \\ &\sqsubseteq_A +_A(h_A(t_1'), h_A(t_2')) && (\text{by induction since } +_A \text{ is } \subseteq\text{-monotonic}) \\ &= h_A(+t_1't_2') && (h_A \text{ is a } \Sigma\text{-homomorphism}) \\ &= h_A(t') \end{aligned}$$

□

<u>Theorem 2.11</u>: NCT_Σ is initial in <u>Str_Σ</u>.

<u>Proof</u>: By Theorem 2.9, $NCT_\Sigma \in$ <u>Str_Σ</u>. We must now prove that n_A in the above diagram is a unique, $\sqsubseteq$-continuous, $\subseteq$-monotonic Σ-homomorphism. The fact that n_A is a Σ-homomorphism is proved by:

$$\begin{aligned}
&n_A(f_N([t_1], \ldots , [t_n])) \\
&\quad = n_A([ft_1 \ldots t_n]) && \text{(definition of } f_N) \\
&\quad = h_A(ft_1 \ldots t_n) && \text{(definition of } n_A) \\
&\quad = h_A(f_C(t_1, \ldots , t_n)) && \text{(definition of } f_C) \\
&\quad = f_A(h_A(t_1), \ldots, h_A(t_n)) && (h_A \text{ is a } \Sigma\text{-homomorphism}) \\
&\quad = f_A(n_A([t_1]), \ldots , n_A([t_n]) && \text{(definition of } n_A).
\end{aligned}$$

The fact that n_A is unique is proved as follows: Suppose $\bar{n}_A$: $NCT_\Sigma \to A_\Sigma$ is a homomorphism. Let $t = a \in \Sigma_{\lambda,s}$. Then $a_A = h_A(a) = \bar{n}_A([a])$ by definition. Let $t = ft_1 \ldots . t_n$ and suppose $n_A([t_i]) = \bar{n}_A([t_i])$ for all i. Then $n_A([t])$

$$\begin{aligned}
&\quad = f_A(h_A(t_1), \ldots , h_A(t_n)) && (h_A \text{ is a } \Sigma\text{-homomorphism}) \\
&\quad = f_A(n_A([t_1]), \ldots , n_A([t_n])) && \text{(definition of } n_A) \\
&\quad = f_A(\bar{n}_A([t_1]), \ldots , \bar{n}_A([t_n])) && \text{(induction hypothesis)} \\
&\quad = \bar{n}_A(f_N([t_1] , \ldots , [t_n]) && (\bar{n}_A \text{ is a } \Sigma\text{-homomorphism}) \\
&\quad = \bar{n}_A([t]) && \text{(definition of } f_N).
\end{aligned}$$

Thus we have proved by induction that $n_A = \bar{n}_A$. Let $\{[t_i]\}$ be a chain in NCT_Σ such that $\{t_i\}$ is a chain in CT_Σ. Then

$$\begin{aligned}
n_A(\sqcup[t_i]) &= n_A([\sqcup t_i]) && \text{(lemma 2.5)} \\
&= h_A(\sqcup t_i) && \text{(definition of } n_A) \\
&= \sqcup h_A(t_i) && (h_A \text{ is continuous}) \\
&= \sqcup n_A([t_i]) && \text{(definition of } n_A).
\end{aligned}$$

Thus n_A is $\sqsubseteq$-continuous.

Suppose $[t] \subseteq_N [t']$. Then

$$n_A([t]) = h_A(t) \qquad \text{(definition of } n_A)$$

$\subseteq_A h_A(t')$ (lemma 2.10 and definition of $\subseteq_N$)

$= n_A([t'])$ (definition of n_A).

So n_A is $\subseteq$-monotonic. □

We have now established the result we were after (the initiality of NCT_Σ in $\underline{Str_\Sigma}$). Now we proceed to apply the above to solving recursive systems of equations. Let Σ be as above and suppose Ξ is some S-sorted family of sets of "function variables". We will be interested in systems of equations of the following form:

$$E: \quad F_1(x_{1,1}, \ldots, x_{1,n_1}) = t_1$$
$$\vdots$$
$$F_m(x_{m,1}, \ldots, x_{m,n_m}) = t_m$$

where (i) $F_j \in \Xi_{w_j,s_j}$ for some $\langle w_j, s_j\rangle \in S^* x S$, $1 \le j \le m$;

(ii) $\langle x_{j,1}, \ldots, x_{j,n_j}\rangle = \langle x_{1,s_{1,j}}, \ldots, x_{n_j,s_{n_j,j}}\rangle$ for $w_j = s_{1,j} \ldots s_{n_j,j}$ and $1 \le j \le m$; and (iii) $t_j \in CT_{\Sigma \cup \Xi}(X_{w_j})_{s_j}$ for $1 \le j \le m$. Thus

$F_1, \ldots, F_m$ are names of the functions which we are defining;

$\langle x_{j,1}, \ldots, x_{j,n_j}\rangle$ are the (properly typed) arguments of F_j; and t_j is some expression in the "basic function symbols" in Σ, the function variables in Ξ, and the input variables $\langle x_{j,1}, \ldots, x_{j,n_j}\rangle$.

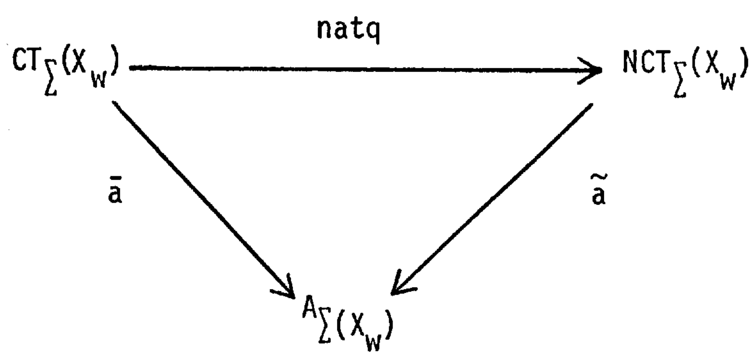
We will show that these equations can be solved by assigning to each t_j a continuous function over a structure whose elements are non-deterministic functions. Thus we consider t_j to be a functional with variables $F_1, \ldots, F_m$. (The $\langle x_{j,1}, \ldots, x_{j,n_j}\rangle$ are essentially ignored in this process.) We proceed by defining this structure of non-deterministic functions.

Let $NCT_\Sigma(X)$ denote $CT_\Sigma(X)/q$. Then $NCT_\Sigma(X)$ is initial in $\underline{Str}_{\Sigma(X)}$. That is, given an assignment $a: X \to A$ where A_Σ is any Σ-structure, there exists a unique, $\sqsubseteq$-continuous, $\subseteq$-monotonic Σ-homomorphism $\tilde{a}: NCT_\Sigma(X) \to A_\Sigma$ which extends a. Let $[t] \in NCT_\Sigma(X_w)_s$ for some $<w,s>$. We use $[t]$ to define a derived operation $[t]_A: A^w \to A_s$ by using the definition $[t]_A(a) = \tilde{a}([t])$ for an assignment $a: X_w \to A$. We call $[t]_A$ a derived operation of type $<w,s>$. Order the set of derived operations of type $<w,s>$ as follows:

(i) $[t]_A \sqsubseteq_F [t']_A$ if and only if for all assignments $a: X_w \to A$, $[t]_A(a) \sqsubseteq_A [t']_A(a)$;

(ii) $[t]_A \subseteq_F [t']_A$ if and only if for all assignments $a: X_w \to A$, $[t]_A(a) \subseteq_A [t']_A(a)$.

(The meaning of F in $\sqsubseteq_F$ and $\subseteq_F$ will become clear in a moment.) Consider the following diagram:



Since $NCT_\Sigma(X_w)$ is initial in $\underline{Str}_{\Sigma(X_w)}$ clearly $\tilde{a}([t]) = \bar{a}(t)$ where $a: X_w \to A$. Thus

$$[t]_A(a) \sqsubseteq_A [t']_A(a) \iff \tilde{a}([t]) \sqsubseteq_A \tilde{a}([t'])$$
$$\iff \bar{a}(t) \sqsubseteq_A \bar{a}(r) \text{ (for some } r \in [t'])$$
$$\iff t \sqsubseteq_C r.$$

Thus $[t]_A \sqsubseteq_F [t']_A$ if and only if $t \sqsubseteq_C r$ for some $r \in [t']$. It can just as easily be shown that $[t]_A \subseteq_F [t']_A$ if and only if $t \subseteq_C r$ for some $r \in [t']$. So the set of derived operations of type $<w,s>$ form a structure with $[\bot_s]$ as least element. (This follows easily from the properties of $CT_\Sigma(X_w)$ and the above analysis.)

Consider the alphabet $F(\Sigma)$ sorted by the set $F(S) = S^* X S$:

(i) $f \in \Sigma_{w,s}$ is an element of $F(\Sigma)_{\lambda,<w,s>}$;

(ii) $\delta_w^i \in F(\Sigma)_{\lambda,<w,s_i>}$ for $w = s_1 \ldots s_n$, $1 \le i \le n$;

(iii) $+' \in F(\Sigma)_{\sigma\sigma,\sigma}$ for each $\sigma \in F(S)$;

(iv) $C_{<w,v,s>} \in F(\Sigma)_{<<w,s><v,s_1>\ldots <v,s_n>,<v,s>>}$ for each $<w,v,s> \in S^* \ X \ S^* \ X \ S$.

That is, the elements of Σ become constants of $F(\Sigma)$; nullaries δ_w^i are introduced to serve as constant functions standing for values of input variables; a new join function $+'$ is introduced to join derived operations; composition functions are added to allow composition of derived operations.

Now define the $F(\Sigma)$-algebra $F(A_\Sigma)$ as follows:

(i) $F(A_\Sigma)_{w,s} = \{[t]_A | [t] \in NCT_\Sigma(X_w)_s\}$;

(ii) $(\delta_w^i)_F := x_{i,s_i}$ for $w = s_1 \ldots s_n$, $1 \le i \le n$;

(iii) $+'_F([t]_A, [t']_A) = [+tt']_A$;

(iv) $(C_{<w,v,s>})_F ([t_0]_A, \ldots , [t_n]_A)$

$= [t_0 \leftarrow <t_1, \ldots , t_n>]_A$.

Theorem 2.12: If A_Σ is a Σ-structure, then $F(A_\Sigma)$ is a $F(\Sigma)$-structure.

Proof: As shown above, each $F(A_\Sigma)_{w,s}$ is a structure. We show that each $C_{w,v,s}$ is $\sqsubseteq_F$-continuous. Let $\{[t_i]_A\}$ be a chain in $F(A_\Sigma)$. Then

$$
\begin{aligned}
&(C_{<w,v,s>})_F\,(\sqcup[t_i]_A,\ [t_1']_A,\ \ldots\ ,\ [t_n']_A) \\
&\quad = (C_{<w,v,s>})_F\,([\sqcup t_i]_A,\ [t_1']_A,\ \ldots\ ,\ [t_n']_A) && \text{(lemma 2.5)} \\
&\quad = [\sqcup t_i \leftarrow <t_1',\ \ldots\ ,\ t_n'>]_A && \text{(definiton of } C_{<w,v,s>}) \\
&\quad = [\sqcup(t_i \leftarrow <t_1',\ \ldots\ ,\ t_n'>)]_A && (\leftarrow \text{ is continuous}) \\
&\quad = \sqcup[t_i \leftarrow <t_1',\ \ldots\ ,\ t_n'>]_A && \text{(lemma 2.5)} \\
&\quad = \sqcup(C_{<w,v,s>})_F\,([t_i]_A,\ [t_1']_A,\ \ldots\ ,\ [t_n']_A) && \text{(definiton of } C_{<w,v,s>}).
\end{aligned}
$$

Similarly, we can show that

$$
\begin{aligned}
&(C_{<w,v,s>})_F\,([t_0']_A,\ [t_1']_A,\ \ldots\ ,\ \sqcup[t_i]_A,\ \ldots\ ,\ [t_n']_A) \\
&\quad = \sqcup(C_{<w,v,s>})_F\,([t_0']_A,\ [t_1']_A,\ \ldots\ ,\ [t_i]_A,\ \ldots\ ,\ [t_n']_A)
\end{aligned}
$$

and so $(C_{<w,v,s>})_F$ is $\sqsubseteq$-continuous. To prove that $+_F'$ is $\sqsubseteq$-continuous, consider

$$
\begin{aligned}
&+_F'(\sqcup[t_i]_A,\ [t']_A) \ \text{(symmetrically } +_F' \ ([t']_A,\ \sqcup[t_i]_A)) \\
&\quad = +_F'([\sqcup t_i]_A,\ [t']_A) && \text{(lemma 2.5)} \\
&\quad = [+\sqcup t_i t']_A && \text{(definition of } +_F') \\
&\quad = [\sqcup + t_i t']_A && (+ \text{ is continuous in } CT_\Sigma) \\
&\quad = \sqcup[+t_i t']_A && \text{(lemma 2.5)} \\
&\quad = \sqcup +_F'([t_i]_A,\ [t']_A) && \text{(definition of } +_F').
\end{aligned}
$$

To show that $(C_{<w,v,s>})_F$ is $\subseteq_F$-monotonic, consider $[t]_A \subseteq_F [t']_A$.

$$(C_{<w,v,s>})_F([t]_A, [t_1]_A, \ldots, [t_n]_A)$$

$$= [t \leftarrow <t_1, \ldots, t_n>]_A \qquad \text{(definition of } C_{<w,v,s>})$$

$$\subseteq_F [r \leftarrow <t_1, \ldots, t_n>]_A \quad \text{for some } r \in [t']$$

$$= (C_{<w,v,s>})_F([r]_A, [t_1]_A, \ldots, [t_n]_A) \quad \text{(definition of } C_{<w,v,s>})$$

$$= (C_{<w,v,s>})_F([t']_A, [t_1]_A, \ldots, [t_n])$$

$$\text{(since } [r] = [t']).$$

It is just as easy to show that

$$(C_{<w,v,s>})_F([t_0]_A, [t_1]_A, \ldots, [t]_A, \ldots, [t_n]_A)$$

$$\subseteq_F (C_{<w,v,s>})_F([t_0]_A, [t_1]_A, \ldots, [t']_A, \ldots, [t_n]_A).$$

Obviously $+_F'$ is $\subseteq_F$-monotonic and so $F(A_\Sigma)$ is a $F(\Sigma)$-structure. □

Now let E be a system of equations as illustrated above. There corresponds to E a system E' over $F(\Sigma) \cup \Xi'$ (where $\Xi'_{\lambda,<w,s>} = \Xi_{w,s}$ and for all $<\omega,\sigma> \in F(S)^+ \times F(S)$, $\Xi_{\omega,\sigma} = \emptyset$) obtained as follows. Define

$F_{w,s} : CT_{\Sigma \cup \Xi}(X_w)_s \to (CT_{F(\Sigma) \cup \Xi'})_{<w,s>}$ by:

(i) $F_{w,s}(x_{i,s}) = \delta_w^i$ for $x_{i,s}$ in X_w;

(ii) $F_{w,s}(a) = C_{<w,\lambda,s>}(a)$ for $a \in (\Sigma \cup \Xi)_{\lambda,s}$;

(iii) $F_{w,s}(\bot_s) = \bot_{<\lambda,s>}$;

(iv) $F_{w,s}(ft_1 \ldots t_n) = C_{<v,w,s>}\, f\, C_{w,v_1}(t_1) \ldots C_{w,v_n}(t_n)$ for $v = v_1 \ldots v_n$, $f \in (\Sigma \cup \Xi)_{v,s}$, and $t_i \in CT_{\Sigma \cup \Xi}(X_w)_{v_i}$ for $1 \le i \le n$.

Then, if $F_j(x_{j,1}, \ldots, x_{j,n_j}) = t_j$ is in E, we place $F_j = F_{w_j,s_j}(t_j)$ in E'. Given any $F(\Sigma)$-structure B, each $F_{w_j,s_j}(t_j)$ defines a $\sqsubseteq_B$-continuous function from $B_{w_1,s_1} X \ldots XB_{w_m,s_m}$ to B_{w_j,s_j} (noting that $\langle\lambda,\langle w_i,s_i\rangle\rangle$ is the type of F_i in E' for $1 \le i \le m$) and so E' defines a $\sqsubseteq_B$-continuous "system" function from $B_{w_1,s_1} X \ldots XB_{w_m,s_m}$ to itself. (See, for example, [6]).

We denote the solution of E' over the structure B by $|E'_B|$. Thus we can find $|E'_{CT_{F(\Sigma)}}|$, $|E'_{F(CT_\Sigma)}|$, $|E'_{F(A_\Sigma)}|$, $|E'_{F(NCT_\Sigma)}|$, etc. where A_Σ is any Σ-structure. We could also find $|E'_{\mathfrak{F}(A)}|$ where $\mathfrak{F}(A)_{w,s} = \{f | f: A^w \to A_s$, f is $\sqsubseteq_A$-continuous, $\subseteq_A$-monotonic$\}$. Thus $F(A_\Sigma)_{w,s} \subseteq \mathfrak{F}(A)_{w,s}$ since $\mathfrak{F}(A)_{w,s}$ contains <u>all</u> $\sqsubseteq_A$-continuous, $\subseteq_A$-monotonic functions from A^w to A_s. Most authors are interested in solving E' over $\mathfrak{F}(A)$, but we will satisfy ourselves with $F(A_\Sigma)$ since the latter contains all program definable functions already.

We will be particularly interested in solving E' over $F(NCT_\Sigma)$ as the following analysis will make clear. $F(NCT_\Sigma)_{w,s}$ is the set of derived operations over NCT_Σ of type $\langle w,s\rangle$. Let $[t]_{F(N)} \in F(NCT_\Sigma)_{w,s}$. Then $[t]_{F(N)}([t_1], \ldots, [t_n]) = \widetilde{[t_1], \ldots, [t_n]}\ ([t])$

(by definition of $[t]_{F(N)}$)

$= \overline{\langle[t_1], \ldots, [t_n]\rangle}\ (t)$

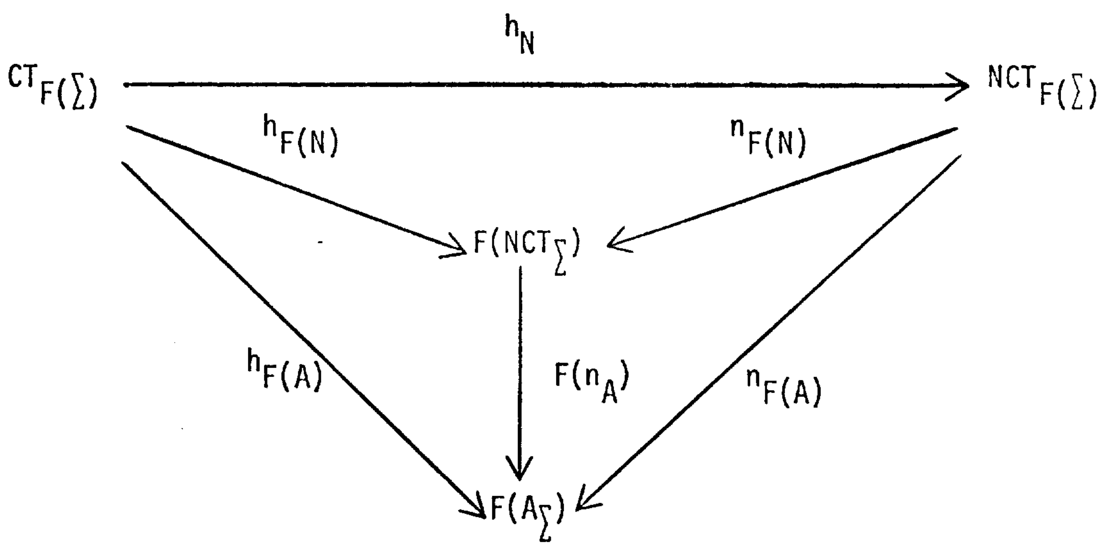
(by definition of $\widetilde{\langle[t_1], \ldots, [t_n]\rangle}$)

Now $\overline{<[t_1], \ldots , [t_n]>}$: $CT_\Sigma(X_w) \to NCT_\Sigma(X_v)$, for some $v \in S^*$, and it is easy to show that

$$\overline{<[t_1], \ldots , [t_n]>}(t) = [t \leftarrow <t_1, \ldots , t_n>].$$

Thus composition of derived operations in $F(NCT_\Sigma)$ corresponds to syntactic substitution in NCT_Σ. This models the usual "body replacement rule" of syntactic or symbolic solutions of systems of equations and motivates our adoption of $F(NCT_\Sigma)$ as our symbolic computation domain.

Consider the following diagram:



Since the least fixed point of a system of equations E' over a structure B is found by taking the least upper bound of an approximating chain and since h_B and n_B are continuous for any continuous algebra B, we must have that the following preserve solutions of equations (i.e., they map the solution to E' in the domain algebra into the solution in the co-domain algebra):

$$h_{F(A)},\ h_{F(N)},\ h_N,\ n_{F(N)},\ n_{F(A)}.$$

It remains to show that $F(n_A)$ is a unique strict, $\sqsubseteq$-continuous, $\subseteq$-monotonic $F(\Sigma)$-homomorphism for any Σ-structure A_Σ, proving that the whole diagram commutes and all mappings in the diagram preserve least fixed points.

Let $[t]$ be in $F(NCT_\Sigma)_{w,s}$. (Note that we have dropped the $F(N)$ from $[t]_{F(N)}$ in $F(NCT_\Sigma)$. This is because of the one to one correspondence between $[t]_{F(N)}$ and $[t]$ in $NCT_\Sigma(X)$ as shown above.) Define

$$F(n_A)([t]) = h_{F(A)}(F_{w,s}(t)).$$

Note that $F_{w,s}(t) \in h^{-1}_{F(N)}(t)$.

<u>Lemma 2.13</u>: $F(n_A)$ is a unique, strict, $\sqsubseteq$-continuous, $\subseteq$-monotonic homomorphism.

<u>Proof</u>: $F(n_A)$ is strict since

$$\begin{aligned} F(n_A)([\bot]) &= h_{F(A)}(F_{w,s}(\bot)) \\ &= h_{F(A)}(\bot) \\ &= \bot_{F(A_\Sigma)}. \end{aligned}$$

We show that $F(n_A)$ is a homomorphism:

$$\begin{aligned} &F(n_A)((C_{<w,v,s>})_{F(N)}([t_0], \ldots , [t_n]) \\ &= h_{F(A)}(F_{v,s}(t_0 \leftarrow <t_1, \ldots , t_n>)) \\ &= h_{F(A)}(F_{v,s}(C_{<w,v,s>}t_0 \cdots t_n)) \\ &= h_{F(A)}(C_{<w,v,s>}F_{w,s}(t_0)F_{v,s_1}(t_1) \cdots F_{v,s_n}(t_n)) \\ &= (C_{<w,v,s>})_{F(A)}(h_{F(A)}(F_{w,s}(t_0)),h_{F(A)}(F_{v,s_1}(t_1)),\ldots,h_{F(A)}(F_{v,s_n}(t_n \\ &= (C_{<w,v,s>})_{F(A)}(F(n_A)([t_0]), \ldots , F(n_A)([t_n])). \end{aligned}$$

Also $F(n_A)(+'_{F(N)}([t_1], [t_2]))$

$$= h_{F(A)}(+'F_{w,s}(t_1)F_{w,s}(t_2))$$

$$= +'_{F(A)}(h_{F(A)}(F_{w,s}(t_1)), h_{F(A)}(F_{w,s}(t_2)))$$

$$= +'_{F(A)}(F(n_A)([t_1]), F(n_A)([t_2])).$$

To prove that $F(n_A)$ is $\sqsubseteq$-continuous, let $\{[t_i]\}$ be a chain in $NCT_\Sigma(X_w)_s$ such that $\{t_i\}$ is a chain in $CT_\Sigma(X_w)_s$. Then

$$F(n_A)(\sqcup[t_i]) = F(n_A)([\sqcup t_i])$$

$$= h_{F(A)}(F_{w,s}(\sqcup t_i))$$

$$= h_{F(A)}(\sqcup F_{w,s}(t_i))$$

(clearly a property of $F_{w,s}$)

$$= \sqcup h_{F(A)}(F_{w,s}(t_i))$$

$$= \sqcup F(n_A)([t_i]).$$

To show that $F(n_A)$ is $\subseteq$-monotonic, let $[t_i] \subseteq_{F(N)} [t_2]$ and consider

$$F(n_A)([t_1]) = h_{F(A)}(F_{w,s}(t_1))$$

$$\subseteq_{F(A)} h_{F(A)}(F_{w,s}(r)) \quad \text{for some } r \in [t_2]$$

(since if $[t_1] \subseteq_{F(N)} [t_2]$, then there exists $r \in [t_2]$ so that $t_1 \subseteq_C r$ and then clearly $F_{w,s}$ preserves $\subseteq$ so $F_{w,s}(t_1) \subseteq_{F(CT_\Sigma)} F_{w,s}(r))$

$$= F(n_A)([r])$$

$$= F(n_A)([t'])$$

It is easy to show that $F(n_A)$ is unique. □

<u>Theorem 2.14</u>: $|E'_{F(A)}| = |E'_{F(NCT_\Sigma)}|_A$. (The solution of E' over $F(A_\Sigma)$ has as its solution a (tuple of) derived operation(s) over A_Σ. $|E'_{F(NCT_\Sigma)}|$ is a (tuple of) derived operation(s) over NCT_Σ; i.e., $|E'_{F(NCT_\Sigma)}|$ is a (tuple of) expression(s) in $NCT_\Sigma(X)$ and so $|E'_{F(NCT_\Sigma)}|_A$ is the (tuple of) derived operation(s) over A_Σ defined by this (tuple of) expression(s). Thus the theorem states that solving E' symbolically over $F(NCT_\Sigma)$ and then taking the derived operation over A defined by the solution gives the same result as solving E' directly over $F(A_\Sigma)$; i.e. finding directly the derived operation defined by E' over A_Σ.)

<u>Proof</u>: $|E'_{F(NCT_\Sigma)}|_A = (\sqcup(E'_{F(NCT_\Sigma)})^n\ (\beta))_A$

(for $\beta = \langle[\bot_{w_1,s_1}], \ldots, [\bot_{w_m,s_m}]\rangle$)

$= \sqcup((E'_{F(NCT_\Sigma)})^n\ (\beta))_A$

$= \sqcup((\langle[t_1], \ldots, [t_m]\rangle \leftarrow \ldots \leftarrow \langle[t_1], \ldots, [t_m]\rangle)(\beta))_A$

(since $E'_{F(NCT_\Sigma)} = \langle h_{F(N)}(F_{w_1,s_1}(t_1)), \ldots, h_{F(N)}(F_{w_m,w_m}(t_m))\rangle$

$= \langle[t_1], \ldots, [t_m]\rangle$)

$= \sqcup(\langle[t_1]_A, \ldots, [t_m]_A\rangle^n(\beta_F))_{(A)}$

(where $\beta_A = \langle(\bot_{w_1,s_1})_{F(A)}, \ldots, (\bot_{w_m,s_m})_{F(A)}\rangle$ and the use of the homomorphism property of $F(n_A)$)

$= \sqcup(E'_{F(A)}{}^n(\beta_{F(A)}))$

$= |E'_{F(A)}|$. □

§3 Equations of Higher Type

We now want to extend our previous results to define non-deterministic functions of higher type (in the logical sense). Let $\sum$ be as in section 2. Define $F^n(S)$ by: $F^0(S) = S$, $F^{n+1}(S) = F^n(S)^* \times F^n(S)$. Define $F^n(\sum)$ by: $F^0(\sum) = \sum$, $F^{n+1}(\sum) = F(F^n(\sum))$. Denote by $+^{(n)}$ for $n \geq 0$, the join symbol introduced at level n. Thus $+^{(0)}$ is $+ \in \sum_{ss,s}$ for some $s \in S$ and $+^{(1)} = +'$ of section 2. Denote by $C^{(n)}, {}^{(n)}\delta$ for $n > 0$ (with, of course, the appropriate subscripts and superscripts) the composition, projection symbols, respectively, introduced at the n'th level. For each $n \geq 0$, define

$$\Xi^{(n)} = \{\Xi^{(n)}_{\omega,\sigma}\}_{\langle\omega,\sigma\rangle \in F^n(S)^* \times F^n(S)}$$

to be an indexed family of variables of level n. We assume each set in each such family to be countably infinite. Thus in section 2, the system E has $F_1, \ldots, F_m \in \Xi^{(1)}$ while the $x_{i,j} \in \Xi^{(0)}_{\lambda,s}$ for appropriate $s \in S$. Define $F^n(A_\sum)$ for any $\sum$-structure $A_\sum$ by: $F^0(A_\sum) = A_\sum$ and $F^{n+1}(A_\sum) = F(F^n(A_\sum))$. We will call

$$E^{(n)} : \quad \begin{array}{c} F^{(n)}_1(x^{(n-1)}_{1,1}, \ldots, x^{(n-1)}_{1,n_1}) = t_1 \\ \vdots \\ F^{(n)}_m(x^{(n-1)}_{m,1}, \ldots, x^{(n-1)}_{m,n_m}) = t_m \end{array}$$

a level n system of equations for $n > 0$ if:

(i) $F^{(n)}_j \in \Xi^{(n)}_{\omega j,\sigma j}$ for some $\langle\omega_j,\sigma_j\rangle \in F^n(S)$, $1 \leq j \leq m$;

(ii) $x^{(n-1)}_{j,i} \in \Xi^{(n-1)}_{\omega',\sigma'}$ for some $\langle\omega',\sigma'\rangle \in F^{n-1}(S)$, $1 \leq j \leq m$ and $1 \leq i \leq n_j$;

$$\text{(iii)} \quad t_j \in CT_{F^{n-1}(\Sigma)\cup\Xi^{(n)}}(\Xi^{(n-1)}_{\omega_j}, \sigma_j \quad \text{for} \quad \langle\omega_j,\sigma_j\rangle \quad \text{as above.}$$

(In (iii) we further assume that t_j contains only occurrences of $F_1^{(n)}, \ldots, F_m^{(n)}$ and no other elements of $\Xi^{(n)}$.) Note that we have used the convention of using upper case letters for the names of functions we are trying to define and lower case letters (mainly x's) for so-called input variables. We do this for the sake of readability. In general, we will write equations in a more readable form as illustrated by the following example:

Example 3.1: Consider the recursive definition

$$F^{(2)}(F^{(1)},X^{(0)})$$

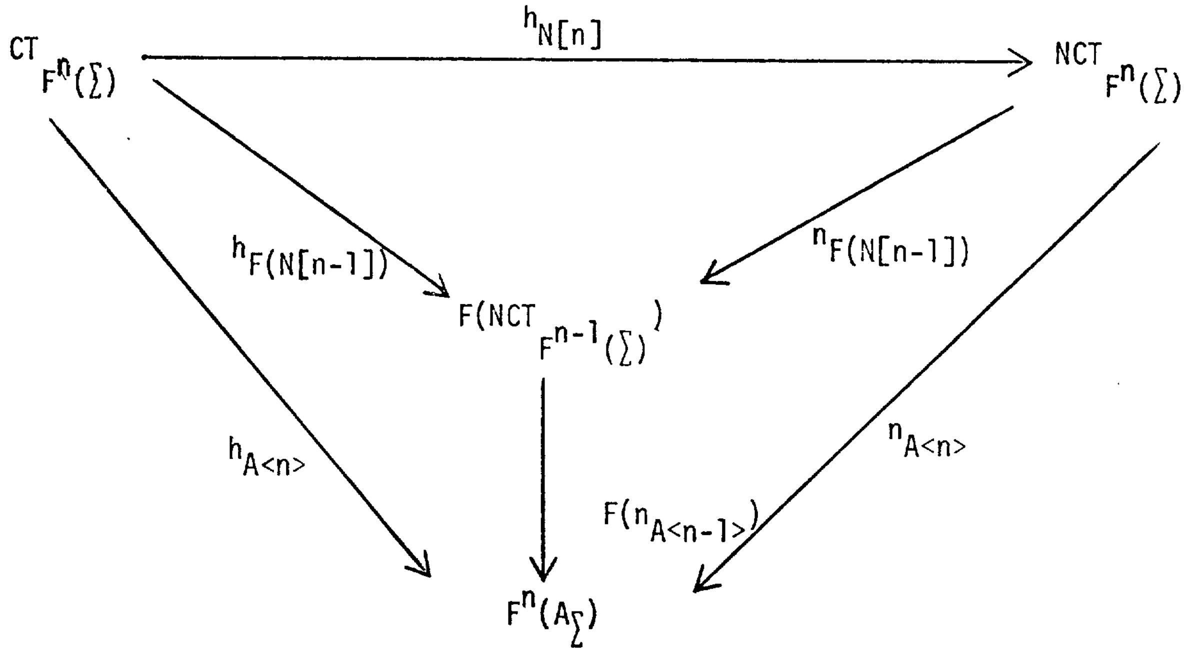
$$= f_1(f_2(F^{(1)}(X^{(0)})), X^{(0)}, F^{(2)}(F^{(1)}, f_3(X^{(0)}))$$

where (i) $\Sigma = \{\Sigma_n\}_{n \in N}$(sorted by $S = \{i,b\}$) and $f_3 \in \Sigma_{i,i}$, $f_2 \in \Sigma_{i,b}$, $f_1 \in \Sigma_{bii,i}$;

(ii) $X^{(0)} \in \Xi_i^{(0)}$, $F^{(1)} \in \Xi_{i,i}^{(1)}$, $F^{(2)} \in \Xi^{(2)}_{\langle i,i\rangle\langle\lambda,i\rangle,\langle\lambda,i\rangle}$.

If this equation is interpreted as an equation over the flat cpo of integers, with f_1 the usual conditional, f_2 a test for zero, and f_3 the successor function, then the resulting least fixed point is the functional which maps the pair $\langle f,a\rangle$(for f: integers $\to$ integers) to the smallest integer b greater than or equal to a for which f(b) is zero, if such a b exists. □

We also use $\underline{or}_n$ as an infix operator to represent $+^{(n)}$ and we use infix notation and other common conventions used in writing systems of equations. It is now up to the reader to show that such an informal system can easily be translated into an equivalent formally defined system. Now given $E^{(n)}$ defined using the formal system, we can use $F_{wj,j}$ (to map t_j to t'_j, an element of $CT_{F^n(\Sigma)\cup\Xi^{(n)}}$ and thus) to define a system $(E^{(n)})'$ corresponding to $E^{(n)}$ which we can then solve using the techniques of the previous section. Now consider the diagram below:



This diagram is the same as the last one in section 2, but with some consistent relabelling of edges and nodes. We have used $N[k]$ to denote $NCT_{F^k(\Sigma)}$ and $B<k>$ to denote $F^k(B_\Sigma)$ for any Σ-structure B_Σ. Thus this clearly gives a relabelling consistent with the new application. Theorem 2.14 becomes in this new setting (dropping the (n) from $E^{(n)}$);

<u>Lemma 3.2</u>:

$$(|E'_{A<n>}| =)\ |E'_{F^n(A_\Sigma)}| = |E'_{F(NCT_{F^{n-1}(\Sigma)})}|\ F^{n-1}(A)$$

$$(= |E'_{F(N[n-1])}|\ A<n-1>). \qquad \square$$

This is not quite good enough even though $F(NCT_{F^{n-1}(\Sigma)})$ is a symbolic domain. However, it is not the "right" symbolic domain. The expressions in $|E'_{F(N[n-1])}|$ contain composition, projection, and join symbols of all levels between one and $n - 1$. Since we are interested mainly in the derived operations defined by these expressions over A_{Σ}, we should use only Σ and the appropriate variables of all levels up to n in our symbolic solution. This means that we must interpret the composition, projection and join symbols of levels between one and $n - 1$ as operations of composition, individual variables of the appropriate level, and join operations, respectively. To put it another way, we should be seeking symbolic solutions in $F^n(NCT_{\Sigma}) = NCT_{\Sigma}\langle n \rangle$. We show that this in fact can be done.

Consider the map $n_{F(N[n-1])} : N[n] \to F(N[n-1])$. Clearly all this map does is interpret the symbols $C^{(n)}$, ${}^{(n)}{}_{\delta}$, and $+^{(n)}$ as composition of functions of level $n-1$, individual variables of level $n - 1$ (i.e., function variables of level $n - 2$), and joins of level $n - 1$ functions, respectively. This is similar to a class of homomorphisms considered in [ ] (see also [ ]) and for that reason will be renamed YIELD. In fact we will ambiguously use YIELD as a name for the homomorphism

$$\text{YIELD}: F^i(N[n-i]) \to F^{i+1}(N[n-i-1])$$

$0 \le i \le n - 1$, which interprets the symbols $C^{(n-i)}$, ${}^{(n-i)}{}_{\delta}$, and $+^{(n-i)}$ as the appropriate compositions, variables, and joins. Clearly, each such YIELD is a homomorphism and clearly it is unique (in the context in which it is found). One can compose such homomorphisms to obtain unique homomorphisms

$$\text{YIELD}^j : F^i(N[n-i]) \to F^{i+j}(N[n-i-j])$$

for $0 \le i \le n-1$ and $0 \le j \le n-i$. Note that again YIELD^j is unique only in context.

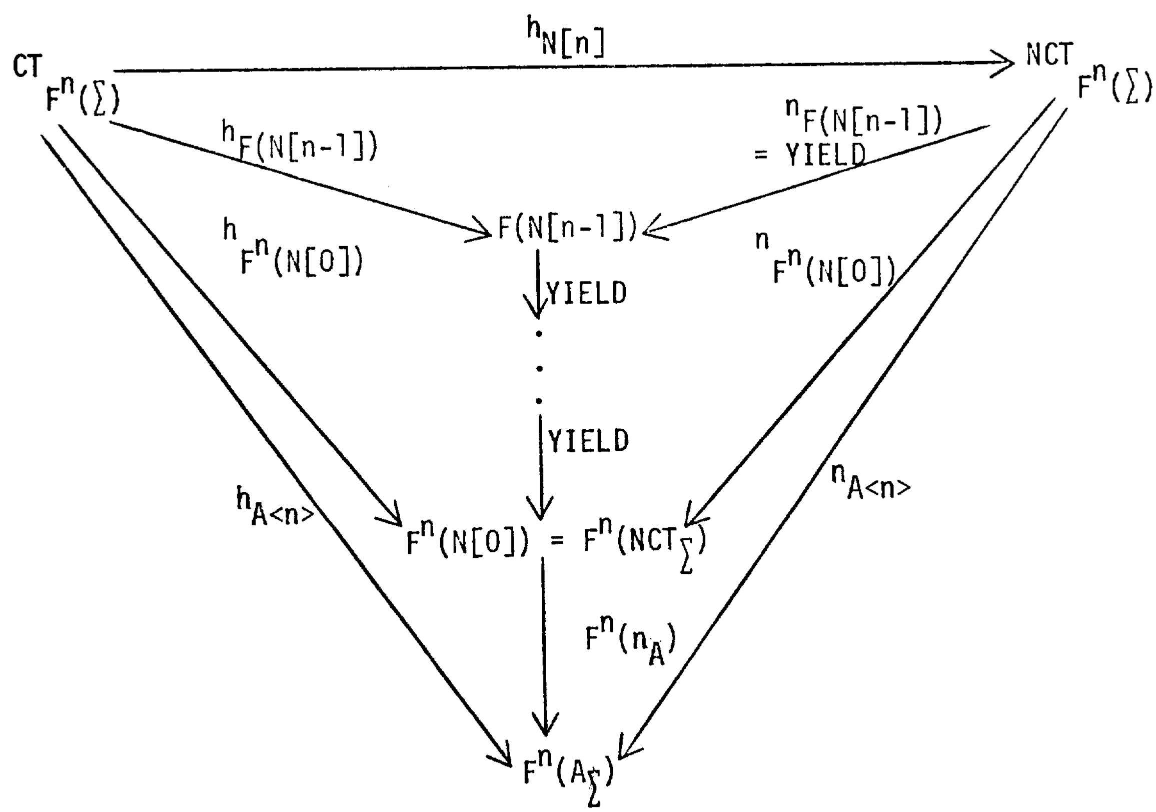
The proof of the following is left to the reader.

<u>Lemma 3.3</u>: For all $n > 0$, $0 \le i \le n-1$, $0 \le j \le n-i$,

$$\text{YIELD}^j : F^i([n-i]) \to F^{i+j}(N[n-i-j])$$

is a strict, $\sqsubseteq$-continuous, $\subseteq$-monotonic, $F^n(\Sigma)$-homomorphism. □

Consider now the following diagram:



Clearly $n_{F(N[n-1])} \circ YIELD^{n-1} \circ F^n(n_A)$ in this diagram must equal $n_{F(N[n-1])} \circ F(n_{A<n-1>})$ of the previous diagram. Paralleling the proof of lemma 2.13, we can define $F^n(n_A)$ in terms of $h_{A<n>}$ and show that it is an unique, strict, $\sqsubseteq$-continuous, $\subseteq$-monotonic $F^n(\Sigma)$-homomorphism. This leads to the following important result for the system E'.

<u>Theorem 3.4</u>: $|E'_{N[n]}|_{A<n>} = |E'_{F^n(N[0])}|_A = |E'_{A<n>}|$. □

(The proof is similar to that for Theorem 2.14.)

BIBLIOGRAPHY

[1] J.A. Goguen, J.W. Thatcher, E.G. Wagner and J.B. Wright,
Initial Algebra Semantics and Continuous Algebras,
J.A.C.M. Vol. 24, No. 1, pp. 68-95, 1977.

[2] J.A. Goguen, J.W. Thatcher, and E.G. Wagner,
An Initial Algebra Approach to the Specification, Correctness,
and Implementation of Abstract Data Types, IBM Research Report RC6487.

[3] E.A. Ashcroft and Z. Manna,
Formalisation of Properties of Parallel Programs,
Machine Intelligence 6, Edinburgh University Press, 1971.

[4] E.W. Dijkstra,
A Discipline of Programming,
Prentice-Hall, 1976.

[5] H. Egli,
A Mathematical Model for Non-deterministic Computations,
Unpublished, 1975.

[6] J. Engelfreit and M. Schmidt, 10 and 01 DAIMI Report PB 47,
University of Aarhus, Danemark (1975).

[7] M.J. Fischer,
Grammars with Macro-Like Productions,
Doctoral Dissertation, Harvard University, 1968.

[8] M. Hennessy,
Call-by-value and Call-by-name in a Non-deterministic Environment,
University of Waterloo, Research Report, CS-77-13, 1977.

[9] M. Hennessy and E.A. Ashcroft,
Parameter-Passing Mechanisms and Non-determinism,
Proc. 9th Annual ACM Conference on Theory of Computing, Boulder, 1977.

[10] P. Hitchcock and D. Park,
Induction Rules and Proofs of Termination,
Proc. of 1st Colloquium on Automata, Languages and Programming,
pp. 225-253, North Holland, 1972.

[11] R. Karp and R. Miller,
Parallel Program Schemata, J.C.S.S.
J.C.S.S., Vol. 3, pp. 147-195, 1969.

[12] M. Levy,
Thesis forthcoming,
University of Waterloo.

Bibliography (Cont'd)

[13] T.S.E. Maibaum,
A Generalized Approach to Formal Languages,
J.C.S.S. Vol. 8, No. 3, June 1974.

[14] T.S.E. Maibaum,
Generalized Syntax and Semantics,
Proc. of Canadian Computer Conference, May, 1976.

[15] M. Nivat,
On the Interpretation of Recursive Polyadic Program Schemes,
Atti dei convegno d'informatica teorica, Rome, 1973.

[16] J. Vuillemin,
Syntaxe, Sémantique et Axiomatique d'un Langage de Programmation Simple,
These de doctoral d'etat, Paris, 1974.

[17] J. Vuillemin,
Correct and Optimal Implementations of Recursion in a Simple
Programming Language, J.C.S.S., Vol. 9, pp. 332-354, 1974.