Recursion-Closed Algebraic Theories*

JEAN H. GALLIER

Department of Computer and Information Science,
The Moore School of Electrical Engineering D2,
University of Pennsylvania, Philadelphia, Pennsylvania 19104

Received November 7, 1980; revised March 30, 1981

A class of algebraic theories called “recursion-closed,” which generalize the rational theories studied by J. A. Goguen, J. W. Thatcher, E.G. Wagner and J. B. Wright in [in “Proceedings, 17th IEEE Symposium on Foundations of Computer Science, Houston, Texas, October 1976,” pp. 147–158; in “Mathematical Foundations of Computer Science, 1978,” Lecture Notes in Computer Science, Vol. 64, Springer–Verlag, New York/Berlin, 1978; “Free Continuous Theories,” Technical Report RC 6906, IBM T. J. Watson Research Center, Yorktown Heights, N.Y., December 1977; “Notes on Algebraic Fundamentals for Theoretical Computer Science,” IBM Technical Report, 1979], is investigated. This work is motivated by the problem of providing the semantics of arbitrary polyadic recursion schemes in the framework of algebraic theories. It is suggested by Goguen et al. (“Proceedings, 17th IEEE Symposium”) that the semantics of arbitrary polyadic recursion schemes can be handled using algebraic theories. The results show that this is indeed the case, but that “rational theories” are insufficient and that it is necessary to introduce a new class of “recursion-closed” algebraic theories. This new class of algebraic theories, is defined and studied, and “free recursion-closed algebraic theories” are proved to exist.

1. INTRODUCTION

The object of this paper is to define and study a generalization of the concept of a rational algebraic theory investigated by Goguen et al. and Thatcher et al. [21, 36–38]. The fundamental concept of an algebraic theory is due to Lawvere [26]. This work is motivated by the problem of providing the semantics of recursion schemes (in the sense of Courcelle and Nivat [10]) in the framework of algebraic theories.

Recursion-closed theories were forced upon us when we discovered that rational theories are not rich enough in least upper bounds (of chains). Consequently, not all finite recursion schemes have fixpoint semantics in a rational theory.

It is suggested in [21] that the semantics of arbitrary polyadic recursion schemes can be handled using algebraic theories.

The above program is carried out in this paper. The results show that such a
treatment is indeed possible, but that rational theories are insufficient and that it is necessary to introduce a class of theories satisfying stronger conditions.

Our investigations proceed in three steps.

(1) By extending slightly the definition of a recursion scheme, (specifically, allowing "parameters") we define an operation of substitution of schemes which confers an interesting structure on the class of schemes. The idea of allowing parameters in schemes is due to Wagner.

(2) Exploiting a suggestion made by Goguen et al. in [21], we define an extended interpretation $I$ as a function $I: \Sigma \to T$ from the alphabet $\Sigma$ from which the schemes are constructed to an ordered algebraic theory $T$. Then, with every scheme $a$ is associated a functional $a_I$, which is shown to be monotonic and the mapping which assigns the functional $a_I$ to the scheme $a$ is a homomorphism of algebraic theories, substitution of schemes corresponding to the composition of functionals.

(3) We investigate the minimal requirements on an interpretation $I$ for the functional $a_I$ associated with a scheme $a$ to have a least fixpoint. We show that the "rational algebraic theories" of [21] are insufficient for that purpose and we define a new class of ordered algebraic theories called "recursion-closed" algebraic theories which satisfy the desired condition. It is shown that every "recursion-closed" algebraic theory is rational in the sense of [21], and we prove that for every ranked alphabet $\Sigma$ there is a free "recursion-closed" algebraic theory $RCT_{\Sigma}$ generated by $\Sigma$, generalizing the results of [21]. The structure of the free "recursion-closed" algebraic theory $RCT_{\Sigma}$ generated by $\Sigma$ can be described explicitly. Indeed, its elements are $n$-tuples of (usually infinite) trees which are least fixpoints of finite recursion schemes.

One of the features of this paper is that we generalize the notion of an interpretation, taking the notion of an algebraic theory as a key concept. Conventionally, an interpretation is a mapping assigning functions to the symbols of the base alphabet, and since functions can obviously be composed, the role played by composition is obscured. Our more general notion of an interpretation (which includes the standard notion) clarifies the role played by composition and the nature of the axioms that an interpretation should satisfy for the fixpoint approach to hold.

The crucial idea which led to the study of rational theories and to the research presented in this paper is the following. In order for a functional $a_I$ to have a least fixpoint, it is sufficient to know that the chain $a^\omega_I(\bot)$ has a least upper bound, rather than to require all $\omega$-chains to have a least upper bound.

This observation was first made by Elgot [12] and exploited by Goguen et al. [21]. Elgot [12] defined a class of algebraic theories, "iterative algebraic theories," where every "ideal" morphism has a unique fixpoint. Goguen et al. [21] introduced the class of "rationally closed" ordered algebraic theories. Intuitively, an ordered algebraic theory $T$ is "rationally closed," if for every "regular recursion scheme" $\alpha$ and every interpretation in $T$, the functional $a_I$ has a least fixpoint. By a "regular" recursion scheme, we mean a recursion scheme where the function symbols $F_i$ have an arity 0, and so, they only occur as leaves. In this paper, we extend the work of
Goguen et al. [21] by introducing the notion of a "recursion-closed" algebraic theory, which is a proper generalization of the notion of "rational" theory. The problem with rational theories is that for some (in fact, most) finite recursion schemes $\alpha$, the functionals $\alpha_I$ fail to have a least fixpoint. Roughly speaking, recursion-closed algebraic theories are obtained by shifting the rationality requirement to the level of functionals. "Recursion-closed" algebraic theories are in a sense "ideal" interpretations for recursion schemes, since they satisfy the "minimal" conditions under which functionals of the form $\alpha_I$ have a least fixpoint. Roughly speaking, an ordered algebraic theory $T$ is "recursion-closed," if for all schemes $\alpha$, for all interpretations $I$ in $T$, the functional $\alpha_I$ has a least fixpoint. We then prove that "free recursion-closed algebraic theories" exist and can be described in a rather simple way. Indeed, every element of the "free recursion-closed algebraic theory $\mathbf{RCT}_\Sigma" generated by a ranked alphabet $\Sigma$ can be described as an $n$-tuple of (infinite) trees obtained as the least fixpoint of a finite recursion scheme. We prove the closure of this theory under the required operations by constructing recursion schemes satisfying certain conditions. The description of the free recursion-closed algebraic theory $\mathbf{RCT}_\Sigma$ also raises a number of questions which appear to be unanswered, and we leave these as open problems.

The importance of algebraic theories in semantics was first recognized by Elgot [12], Wagner [40–42] and Goguen [18, 19]. They realized that a general study of fixpoint solutions as initiated by the work of Scott and others [32–35] could be fruitfully carried out in the unifying framework of algebraic theories. Following Elgot, Ginali [16, 17], Burstall and Thatcher [7], Burstall and Goguen [6], Goguen et al. [20, 21] and Thatcher et al. [36–38] have used algebraic theories in semantic studies. In particular, the semantics of flowchart programs and of monadic recursion schemes is very nicely treated in [37] using the "introduction of variables construction." A brief sketch of the "introduction of variables construction," which is very closely related to our treatment, is also given in [36] for monadic recursion schemes. Related studies of schemes are those of Nivat [30], Courcelle [8, 9], Courcelle and Nivat [10] and Guessarian [25]. Ordered clones and theories have been studied by Wand [43–45]. Recent work of Arnold [1], Arnold and Nivat [3, 4] and Boudol [5] deals with the difficult problem of tackling nondeterminism. The work of Nivat and Arnold [31] is noteworthy since it bases its foundation on the concept of complete metric spaces instead of partial orders. Tiuryn [39] studies classes of rational algebras and the relationship between rational and iterative theories. We also point out that there seems to be very close connections between algebraic theories and the "magmoides" of Arnold and Dauchet [2]. Finally, an extensive study of varieties of chain-complete algebras and a very complete bibliography on this topic can be found in Meseguer [28].

We now describe briefly the contents of this paper.

Section 2 contains a summary of definitions and results used. We define labeled trees, algebraic theories and algebras. In Section 3, we define the class of recursion schemes, starting with the definition of "standard schemes" and then generalizing the definition, which allows us to give an interesting structure to the class of schemes, by
introducing an operation of substitution of schemes. This operation is defined in Section 4 and is shown to be associative and \( \omega \)-continuous. Generalized interpretations and the functionals associated with a scheme are defined and studied in Section 5. After a brief review of “rational theories,” “recursion-closed” algebraic theories are defined and studied in Section 6. It is shown that “free recursion-closed” algebraic theories exist by providing their construction.

2. Preliminaries

In order to minimize the review of definitions and results needed in this work, we will follow as much as possible the definitions and notations found in the works of Thatcher et al. [36–38] and Goguen et al. [20, 21]. We warn the reader who is already familiar with this material and impatient to reach the heart of the subject that our definition of an algebraic theory is the dual of that of [36–38]. This has the advantage of eliminating a number of confusing reversals.

Sorts (or types). By a set of sorts (or types), we understand a set \( S \) of data types in some programming language. For example, \( S = \{\text{integer, real, boolean, character}\} \) is a set of sorts.

\( S \)-ranked alphabet. An \( S \)-ranked alphabet \( \Sigma \) is a family \( (\Sigma_{u,s})_{(u,s) \in S^* \times S} \) of sets \( \Sigma_{u,s} \) indexed by the pairs \( (u, s) \) in \( S^* \times S \). Intuitively, if \( u = u_1 \cdots u_n \), each symbol \( f \) in \( \Sigma_{u,s} \) represents an operation taking \( n \) arguments, each of sort \( u_i \), and yielding an element of sort \( s \). Symbols in \( \Sigma_{1,s} \) are called constants of sort \( s \). We say that a symbol \( f \) in \( \Sigma_{u,s} \) is of sort \( s \) and has arity \( u \). In the rest of this paper, we will assume that a special symbol denoted \( \bot_s \) is adjoined to every \( S \)-ranked alphabet \( \Sigma_{1,s} \) (\( \bot_s \) is of arity \( \lambda \)).\footnote{The empty string is denoted \( \lambda \).} We will usually drop the subscript \( s \).

\( \Sigma \)-trees. A \( \Sigma \)-tree \( t \) is a (finite branching, ordered, possibly infinite) tree whose nodes are labeled with symbols from an \( S \)-ranked alphabet \( \Sigma \) in a way which is consistent with the sorts and arities of the symbols in \( \Sigma \).

Formally, \( \Sigma \)-trees are defined using the notion of a tree domain due to Gorn [23]. Let \( \omega \) denote the set of nonnegative integers.

Tree domain. A tree domain \( D \) is a nonempty subset of \( (\omega - \{0\})^* \) satisfying the following conditions:

1. For all strings \( u \) in \( D \), every prefix \( v \) of \( u \) is also in \( D \).
2. For every string \( u \) in \( D \), for every positive integer \( i \), if \( ui \) is in \( D \) then, for every \( j \), \( 1 \leq j \leq i \), \( uj \) is also in \( D \).

\( \Sigma \)-tree. Given an \( S \)-ranked alphabet \( \Sigma \), a \( \Sigma \)-tree (for short, a tree) is a function \( t: D \to \Sigma \) such that the following conditions hold.
RECURSION-CLOSED ALGEBRAIC THEORIES

(1) \( D \) is a tree domain.

(2) For all \( w \) in \( D \), let \( n = \text{card}(\{wi|wi \in D\}) \).

(i) If \( n = 0 \) then \( t(w) \) belongs to some \( \Sigma_{\lambda,s} \), or

(ii) \( n > 0 \) and if each \( t(wi) \) is of sort \( u_i \), then \( t(w) \in \Sigma_{\lambda,s} \) for some sort \( s \) and where \( u = u_1 \cdots u_n \).

\( D \) is called the domain of the tree \( t \) and is denoted \( \text{dom}(t) \). The elements of the domain \( \text{dom}(t) \) are called the nodes of the tree. A node satisfying condition (2i) is called a leaf. The node corresponding to the empty string is the root of the tree. The sort \( s \) of the symbol labeling the root of a tree \( t \) is also called the sort of the tree. The tree with a single node labeled \( \bot \) is also denoted \( \bot \). \( \Sigma \)-trees have an obvious graphical representation as illustrated below.

EXAMPLE 2.1. A \( \Sigma \)-tree. Let \( S = \{\text{int, bool}\} \) and let \( \Sigma_{\text{int-int, int}} = \{f, g\}, \Sigma_{\text{int, bool}} = \{p\}, \Sigma_{\lambda, \text{int}} = \{a, b\} \) and \( \Sigma_{\text{bool-int, int}} = \{c\} \).

\[
\begin{array}{c}
\text{t} \\
\text{\quad g} \\
\text{\quad a} \\
\text{\quad o} \\
\end{array}
\]

\( t \) is a \( \Sigma \)-tree.

For every \( f \in \Sigma_{\lambda, s} \), the tree

\[
\begin{array}{c}
\text{f} \\
\text{\quad x_1} \\
\text{\quad \ldots} \\
\text{\quad x_n}
\end{array}
\]

where \( |u| = n \ (n \geq 1) \) is denoted \( f \), for every \( x_i^u, 1 \leq i \leq n \), the tree \( x_i^u \) is denoted \( x_i^u \) and for every \( a \in \Sigma_{\lambda,s} \), the tree \( a \) is denoted \( a \).

The set of all \( \Sigma \)-trees of sort \( s \) is denoted \( CT_s^\Sigma \) and the set of all \( \Sigma \)-trees is denoted \( CT_\Sigma \). A tree is total if the label \( \bot \) does not occur in the tree; otherwise we say that the tree is partial. A tree is finite if its domain is finite. The set of total finite trees is denoted \( T_s^\Sigma \) and the set of partial and total finite trees is denoted \( FT^\Sigma \).

There is a partial ordering \( \preceq \) defined on \( CT_s^\Sigma \) (and \( FT^\Sigma \)) as follow. For every pair of trees \( t_1, t_2 \) in \( CT_s^\Sigma \), the relation \( t_1 \preceq t_2 \) holds if and only if

1. \( \text{dom}(t_1) \subseteq \text{dom}(t_2) \) and

2. for all \( w \in \text{dom}(t_1) \), \( t_1(w) \neq \bot \) implies that \( t_2(w) = t_1(w) \).

The tree \( \bot \) is the least element of \( CT_s^\Sigma \) ordered by \( \preceq \).

\( \omega \)-completeness. A partially ordered set (for short, a poset) is \( \omega \)-complete if every countable chain has a least upper bound. In particular, the empty chain has a least
upper bound which is the least element of the poset. It is usually denoted \( \perp \). A poset having a least element is also called a strict poset.

\omega\text{-continuity.} Given two \( \omega\)-complete posets \( D_1 \) and \( D_2 \), a function \( f: D_1 \to D_2 \) is \( \omega\)-continuous if it preserves least upper bounds of countable chains.

The following proposition is well known.

2.1. **Proposition.** Given two \( \omega\)-complete posets \( D_1 \) and \( D_2 \), the set \([D_1 \to D_2]\) of all \( \omega\)-continuous functions from \( D_1 \) to \( D_2 \) is \( \omega\)-complete under the pointwise ordering.

We recall some properties of trees which will be used in our proofs. For every tree \( t \) in \( CT_x \), we define the truncation of order \( n \) of \( t \) denoted \( t^{(n)}\) as follows:

\[
\text{dom}(t^{(n)}) = \{ u | u \in \text{dom}(t) \text{ and } |u| \leq n \}\]

and for all \( u \) in \( \text{dom}(t^{(n)}) \), if \( |u| < n \), \( t^{(n)}(u) = t(u) \) and if \( |u| = n \) then \( t^{(n)}(u) = \perp \).

The next proposition asserts the well known fact that \( CT_x \) is an “algebraic poset” with basis \( FT_x \).

2.2. **Proposition.** Every tree \( t \) in \( CT_x \) is the least upper bound of the \( \omega\)-chain \( (t^{(n)})_{n \in \omega} \); for every \( \omega\)-chain \( (t_i)_{i \in \omega} \), for every finite tree \( t \) in \( FT_x \), if \( t \leq \bigsqcup_{i \in \omega} t_i \) then there exists \( m \) such that \( t \leq t_m \).

**Tree-composition.** The relevant operation here is that of tree-composition. We introduce for every string \( u \in S^* \) the set of variables \( X_u = \{ x^u_1, \ldots, x^u_{|u|} \} \) (with \( X_\perp = \emptyset \)). The variables \( x^u_i \) are used as markers indicating the leaves where the substitution operation takes place. Given a tree \( t \) in \( CT_{X_u} \) and an \( n\)-tuple \( (t_1, \ldots, t_n) \) of trees in \( CT_x \) with each tree \( t_i \) a tree of sort \( u_i \), the result of composing \( (t_1, \ldots, t_n) \) with \( t \) denoted \( (t_1, \ldots, t_n) \circ t \) is the tree obtained by substituting the tree \( t_i \) for each leaf labeled \( x^u_i \) in \( t \).

Formally, the composition of \( (t_1, \ldots, t_n) \) and \( t \) as above is the tree \( (t_1, \ldots, t_n) \circ t \) defined by the function whose graph is the set of pairs \( \{(w, t(w)) | w \in \text{dom}(t) \text{ and } t(w) \in X_u \} \cup \{(wz, t_i(z)) | w \in \text{dom}(t_i), t(w) = x^v_i, z \in \text{dom}(t_i) \text{ and } t_i \text{ is a tree of sort } u_i\} \).

When the set of sorts \( S \) contains a single element, we can denote this element as \( 1 \), and a string \( 11 \cdots 11 \) of \( n \) symbols is denoted as \( n \). In this case, a \( \Sigma \)-tree is just a \( \Sigma \)-tree as defined in Goguen et al. [20, 22], and tree-composition is also the standard tree-composition of [20, 22]. In addition, \( CT_x(u, v) \) is denoted as \( CT_x(m, n) \).

We now define a structure which, as we will see shortly, is the “free algebraic theory generated by the alphabet \( \Sigma \).”

**The structure \( CT_x \).** For all \( u \in S^* \) and \( v \in S^+ \) with \( u = u_1 \cdots u_n \) or \( u = \perp \) and \( v = v_1 \cdots v_p \), we define the set \( CT_x(u, v) \) as the set of all triples of the form \( (u, (t_1, \ldots, t_p), v) \) with each \( t_i \) a tree of sort \( v_i \) in \( CT_{X_{u_i}, X_{v_i}} \) (in the limit case \( u = \perp \), recall that \( X_{\perp} = \emptyset \)). In the limit case \( v = \perp \), we have \( CT_x(u, \perp) = \{(u, \perp, \perp)\} \) and we denote \( ^2 n = |u| \).
We also identify every symbol $f \in \Sigma_{u,s}$ with the element $(u, f, s)$ and every symbol $x_i^n$ with the element $(u, x_i^n, u_i)$. An element $(u, t, v)$ of $\text{CT}_z(u, v)$ is also denoted $t: u \rightarrow v$. $\text{CT}_z$ is the union of all the $\text{CT}_z(u, v)$ for $u, v \in S^*$. We define a composition operation $\circ$ as follows. Given $t: u \rightarrow v$ and $t': v \rightarrow w$, where $t = (u, (t_1, ..., t_p), v)$ and $t' = (v, (t'_1, ..., t'_q), w)$ we have $t \circ t' = (u, ((t_1, ..., t_p) \circ t'_1, ..., (t_1, ..., t_p) \circ t'_q), w)$, where $(t_1, ..., t_p) \circ t'_1$ is the tree-composition of $(t_1, ..., t_p)$ and $t'_1$. We define another operation called tupling as follows. For all $u \in S^*$ and $v \in S^+$ with $v = v_1 \cdots v_p$, for every $p$ element $\phi_i = (u, t_i, v_i): u \rightarrow v_i$, we define the element $[\phi_1, ..., \phi_p]: u \rightarrow v$ as $(u, (\phi_1, ..., \phi_p), v)$. The elements $x^n_i: u \rightarrow u_i$ are called projections, and we define the identity elements as $I_{u} = [x^n_1, ..., x^n_n]$ for each $u = u_1 \cdots u_n$ (with $I_{\lambda} = 0_{\lambda}$). The following proposition summarizes the important properties of $\text{CT}_z$.

2.3. Proposition. (1) Each $\text{CT}_z(u, v)$ is an $\omega$-complete poset.

(2) The composition $\circ$ is associative and the $I_{u}$ are identities.

(3) Composition is $\omega$-continuous on the left and on the right.

(4) For every $v = v_1 \cdots v_p$, for every $p$-tuple $(t_1, ..., t_p)$ of elements $t_i: u \rightarrow v_i$, we have $[t_1, ..., t_p] \circ x^n_i = t_i$ and for every $t: u \rightarrow v$ we have $[t \circ x^n_1, ..., t \circ x^n_p] = t$.

The construction of $\text{CT}_z$ can also be performed by restricting the trees to be total finite or partial finite and we obtain the corresponding structures $T_z$ and $\text{FT}_z$. The above construction leads us to the definition of an algebraic theory.

The notion originated with Lawvere [26]. Our presentation is closer to that of Eilenberg and Wright [11] and Thatcher et al. [36, 37, 38].

### Algebraic Theories

Let $S$ be a set called a set of sorts. An algebraic theory $T$ based on $S$, for short an $S$-theory, is a structure consisting of a family of sets $T(u, v)$ of arrows for all $u, v \in S^*$, together with a composition operation $\circ$, a tupling operation $\mid$, and for every $u \in S^+$, where $u = u_1 \cdots u_n$, of projections $x^n_i: u \rightarrow u_i$ for $1 \leq i \leq n$, such that the following conditions hold:

1. For all $u, v, w \in S^*$, the composition operation $\circ: T(u, v) \times T(v, w) \rightarrow T(u, w)$ is associative and has identities $I_{u} \in T(u, u)$ for all $u \in S^*$.

2. For every $u \in S^*$, there exists a unique arrow $0_u$ in $T(u, \lambda)$.

3. For every $v \in S^+$, where $v = v_1 \cdots v_p$, given any $p$ ($p \geq 1$) arrows $\phi_1, ..., \phi_p$ with $\phi_i$ in $T(u, v_i)$ ($u \in S^*$), $[\phi_1, ..., \phi_p]$ is an arrow in $T(u, v)$, and projections and tupling satisfy the following identities for all $\phi \in T(u, v)$, for $u \in S^*$ and $v = v_1 \cdots v_p$ in $S^+$:

$$[\phi_1, ..., \phi_p] \circ x^n_i = \phi_i \quad (1 \leq i \leq p)$$

and

$$[\phi \circ x^n_1, ..., \phi \circ x^n_p] = \phi.$$
EXAMPLE 2.2. Some algebraic theories.

(1) $T_I$, $FT_I$ and $CT_I$.

(2) Let $A$ be a nonempty set. The set of all total functions $f: A^m \to A^n$ will be denoted as $T(A)(m, n)$. $T(A)$ is an algebraic theory, composition being functional composition and tupling being defined as follows: if $f: A^m \to A^n$, then $f$ can be written uniquely as $f = (f_1, \ldots, f_n)$ with $f_i: A^m \to A$. Projections are projections in the usual sense.

(3) Let $A$ be a nonempty set. Let $A_\perp$ be the flat poset obtained by adjoining an element $\perp$ to $A$. The partial ordering on $A_\perp$ is defined as: $x \leq y$ if $x = \perp$ or $x = y$. The set $CT(A_\perp)(m, n)$ is the set of all total monotonic functions $f: A^m_\perp \to A^n_\perp$. It is easily shown that such functions are $\omega$-continuous. Then, $CT(A_\perp)$ is an algebraic theory, just as $T(A)$ is. $CT(A_\perp)$ also has an additional structure. We will be mainly interested in algebraic theories equipped with a partial ordering on the sets $T(u, v)$. Note that the structure $CT(A_\perp)$ can be substituted for continuous algebras in defining the interpretation of a scheme.

Ordered algebraic theories. An algebraic theory $T$ is ordered if each set $T(u, v)$ is partially ordered and has a least element $\perp_{u,v}$, and composition and tupling are monotonic. It is required that for all $u, v \in S^*$, $0_u \circ \perp_{u,v} = \perp_{u,v}$. This implies that for all $\phi: u \to v$, we have $\phi \circ \perp_{u,v} = \perp_{u,v}$. We say that composition is right-strict.

$\omega$-continuous algebraic theories. An ordered algebraic theory $T$ is $\omega$-continuous if each $T(u, v)$ is $\omega$-complete and composition is $\omega$-continuous.

It is easily seen that the other axioms imply the $\omega$-continuity of tupling.

There is an obvious notion of a homomorphism of algebraic theories. If $T_1$ and $T_2$ are algebraic theories, a homomorphism $h: T_1 \to T_2$ maps every arrow $\phi: u \to v$ in $T_1$ onto an arrow $h(\phi): u \to v$ in $T_2$ and preserves composition, tupling, identities and projections. In addition, for ordered algebraic theories, $h$ is monotonic on each $T(u, v)$ and preserves least elements, and for $\omega$-continuous theories, $h$ is $\omega$-continuous on each $T(u, v)$.

One of the reasons for the interest in algebraic theories comes from the fact that "free algebraic theories generated by an $S$-ranked alphabet" exist, and that they consist of trees. Furthermore, free algebraic theories are characterized by a universal extension property which proves to be a very useful tool, as we shall see in the next sections.

The free algebraic theories generated by $\Sigma$ are respectively the structures $T_{\Sigma}$, $FT_{\Sigma}$ and $CT_{\Sigma}$ previously defined.

The following theorem expresses the "freeness" of the algebraic theory $CT_{\Sigma}$ (and we have similar theorems for $FT_{\Sigma}$ and $T_{\Sigma}$).

2.4. Theorem. [21, 22, 37]. For every $\omega$-continuous algebraic theory $T$, for every function $h: \Sigma \to T$ assigning an arrow $h(f): u \to s$ to each symbol $f \in \Sigma_{u,s}$, there exists a unique homomorphism of $\omega$-continuous algebraic theories $h$ extending $h$ as in the following diagram:
Finally, we will need the fact that algebraic theories can be used to define classes of algebras. (This is actually another reason for introducing the concept; see Manes [27] for details.) Our definition is an adaptation of the definition given in Eilenberg and Wright [11].

Given an S-theory \( T \) and a family \( A = \{A_r\}_{r \in S} \) of sets indexed by \( S \), we define a \( "T\)-algebra" \( \mathcal{A} \), by assigning to every arrow \( \phi: u \rightarrow v \) in \( T \) an operation \( \phi_A: A^u \rightarrow A^v \),3 and imposing the following consistency properties:

(i) To every projection \( x_{11}^n \) (where \( u = u_1 \cdots u_n, \; n \geq 1 \)), \( (x_{11}^n)_A: A^u \rightarrow A^{u_1} \) is the projection on the \( i \)th factor of the cross-product \( A^u \).

(ii) For all \( \phi: u \rightarrow v \) and \( \psi: v \rightarrow w \), we have

\[
(\phi \circ \psi)_A = \phi_A \cdot \psi_A: A^u \rightarrow A^w.
\]

It can be shown that when \( T = T_\Sigma \), the free algebraic S-theory generated by a many-sorted ranked alphabet \( \Sigma \), the above definition is equivalent to the notion of many-sorted algebra, as defined in Goguen et al. [22]. The only difference, is that the above definition yields the same \( \Sigma \)-algebra, plus all of its “derived operators.” Finally, if \( T \) is a strict ordered algebraic S-theory \( T \), we require every \( A_r \) to be a strict poset and the operations to be monotonic. For an \( \omega \)-continuous algebraic S-theory \( T \), we require every \( A_r \) to be an \( \omega \)-continuous poset and the operations to be \( \omega \)-continuous.

3. Definition of the Class of “Tree-Like” Recursion Schemes

We start by formulating the definition of a scheme, and then give the definition of a “parameterized scheme.” Parameterized schemes can be used to define more general functionals.

3.1. Definition. Let \( \Sigma \) be a (one-sorted) ranked alphabet and let \( \Phi_N \) be the set of function symbols \( \Phi_N = \{F_1, \ldots, F_N\} \) \((N \geq 1)\), where every \( F_i \) has an arity \( m_i \geq 0 \). A recursion scheme \( \alpha \) is a function \( \alpha: \Phi_N \rightarrow T_{\Sigma, \Phi_N} \), with every \( \alpha(F_i) \) a finite tree in \( T_{\Sigma, \Phi_N}(m_i, 1) \). Each tree \( \alpha(F_i) \) is also denoted \( \alpha_i \), and is a tree labeled with symbols in \( \Phi_N \), the variables \( x_{i1}^{m_i}, \ldots, x_{in}^{m_i} \) and symbols in \( \Sigma \).

3 For \( u = u_1 \cdots u_n, \; A^u = A^{u_1} \times \cdots \times A^{u_n} \).
A recursion scheme \( a: \Phi_N \rightarrow T_{\Sigma, \Phi_N} \) may also be represented as a system of equations:

\[
F_1(x_{i_1}^{m_1}, ..., x_{m_1}^{m_1}) = a_1 \\
... \\
F_N(x_{i_N}^{m_N}, ..., x_{m_N}^{m_N}) = a_N.
\]

In practice, we usually omit superscripts to avoid cumbersome notations.

**Example 3.1.** A recursion scheme \( a \).

Parameterized recursion schemes are defined by allowing the trees \( a_i \) to contain function symbols \( G_j \), other than the function symbols \( F_1, ..., F_N \) occurring on the left-hand side of the system of equations. In this way, we can define more general functionals, and give a structure of algebraic theory to the class of schemes. In this extension, we can also assume that the symbols in \( \Sigma \) have different sorts. For example, assume that the set of sorts is \( S = \{ \text{int}, \text{bool} \} \), that \( \Sigma \) is a two-sorted ranked alphabet where the only nonempty sets of symbols are \( \Sigma_{\text{int-\text{int}} \cdot \text{int}} = \{ f, g \} \), \( \Sigma_{\text{bool-bool}} = \{ p \} \) and \( \Sigma_{\text{bool-\text{int-\text{int}} \cdot \text{int}}} = \{ c \} \). Finally, let \( F_1 \) be a function symbol of type \( \text{int} \) and of arity \( \text{bool} \cdot \text{int} \cdot \text{int} \), \( F_2 \) and \( F_3 \) two function symbols of type \( \text{int} \) and of arity \( \text{int} \cdot \text{int} \). \( x_i \) a variable of type \( \text{bool} \) and \( x_i', x_i'' \) two variables of type \( \text{int} \). The following is a parameterized scheme:
In the general case, it is convenient to define the following sets. We assume that the set of sorts (types) is denoted $S$ and we have a many-sorted ranked alphabet $\Sigma$ with $\Sigma = (\Sigma_{u,s})_{(u,s) \in S^* \times S}$. Then, for every string $u \in S^+$, let $X_u$ be the set of variables $X_u = \{x_{i1}^u, \ldots, x_{in}^u\}$ for $u = u_1 \cdots u_n$. The set of pairs $(u, s) \in S^* \times S$ constitutes a new alphabet denoted $D(S)$. For every non-null string $\bar{u}$ of symbols in $D(S)$ with $\bar{u} = (u_1, s_1) \cdots (u_n, s_n)$, let $\Phi_{\bar{u}}$ be the set of function variables $\Phi_{\bar{u}} = \{F_{1\bar{u}}, \ldots, F_{n\bar{u}}\}$ (and $\Phi_{\lambda} = \emptyset$). Alphabets become more complex, although conceptually the idea remains the same. Now, given a (possibly null) string $\bar{u} = (u_1, s_1) \cdots (u_n, s_n)$ of symbols in $D(S)$ and a non-null string $\bar{v} = (v_1, r_1) \cdots (v_p, r_p)$ of symbols in $D(S)$, we have the two sets of function variables $\Phi_{\bar{u}} = \{F_{1\bar{u}}, \ldots, F_{n\bar{u}}\}$ and $\Phi_{\bar{v}} = \{F_{1\bar{v}}, \ldots, F_{p\bar{v}}\}$, where each function symbol $F_{i\bar{u}}$ has type $s_i$ and arity $u_i$ and each function symbol $F_{j\bar{v}}$ has type $r_j$ and arity $v_j$. A parameterized recursion scheme with a set of "input function symbols" $\Phi_{\bar{u}}$ and a set of "output function symbols" $\Phi_{\bar{v}}$, is a function $\alpha: \Phi_{\bar{u}} \to T_{\Sigma, \Phi_{\bar{v}}}$, with every $\alpha(F_{j\bar{v}})$ a tree in $T_{\Sigma, \Phi_{\bar{v}}}(v_j, r_j)$, that is, a tree of type $r_j$ labeled with variables in $X_{v_j}$ and function symbols in $\Phi_{\bar{u}}$ (with $\Phi_{\lambda} = \emptyset$ if $\bar{u} = \lambda$, the null string).

3.2. DEFINITION. Let $S$ be a set of sorts and $\Sigma = (\Sigma_{u,s})_{(u,s) \in S^* \times S}$ a many-sorted ranked alphabet. For every string $\bar{u} \in D(S)^*$ and every nonempty string $\bar{v} \in D(S)^+$, with $\bar{u} = (u_1, s_1) \cdots (u_n, s_n)$ and $\bar{v} = (v_1, r_1) \cdots (v_p, r_p)$, a parameterized recursion scheme of type $(\bar{u}, \bar{v})$ is a function $\alpha: \Phi_{\bar{u}} \to T_{\Sigma, \Phi_{\bar{v}}}$, where every $\alpha(F_{j\bar{v}})$ also denoted $\alpha_i$ is a tree of type $r_j$ in $T_{\Sigma, \Phi_{\bar{v}}}(v_j, r_j)$, labeled with variables in $X_{v_j}$ and function symbols in $\Phi_{\bar{u}}$. Such a recursion scheme is also denoted $\alpha: \bar{u} \to \bar{v}$.

When $\bar{u} = \bar{v}$, we say that the scheme is closed, and we say that it has type $\bar{u}$. The set of all recursion schemes $\alpha: \bar{u} \to \bar{v}$ is denoted $PRS_S(\bar{u}, \bar{v})$ and the set of all recursion schemes on $\Sigma$ is denoted $PRS_{\Sigma}$. We can also define sets of parameterized recursion schemes $FPRS_S$ and $CPRS_S$ by replacing $T_{\Sigma}$ successively by $FT_{\Sigma}$ and $CT_{\Sigma}$ in the definition, obtaining schemes with finite partial trees and schemes with infinite trees. However, most of the time, we restrict our attention to $PRS_{\Sigma}$. Next, we define an operation of substitution (of parameterized schemes), which confers to $PRS_{\Sigma}$ a structure of many-sorted algebraic theory over the alphabet $D(S)$. 

571/23/1-6
4. SUBSTITUTION OF SCHEMES

The operation of scheme substitution is a simultaneous substitution of trees for the occurrences of undefined function symbols occurring in another tree, performed in a homomorphic manner. It is probably best to give an example first.

EXAMPLE 4.1. Substitution of schemes.

Scheme \( \alpha \):

\[
\begin{aligned}
&F \quad G \\
&\quad \quad x < \quad x \\
&\quad \\
&h \\
\end{aligned}
\]

Scheme \( \beta \):

\[
\begin{aligned}
&f \\
&G \\
&\quad g \quad x \\
&G \\
&\quad \quad x \\
&\quad \\
\end{aligned}
\]

Scheme \( \alpha * \beta \), the result of the substitution of \( \alpha \) into \( \beta \):

\[
\begin{aligned}
&f \\
&G \\
&\quad g \quad G \\
&\quad \quad x \\
&\quad \quad \\
&h \\
&\quad G \\
&\quad \quad x \\
&\quad \quad \\
&G \\
&\quad \quad x \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \\
&\quad \quad \

For simplicity, let us first consider the substitution of a scheme $\alpha$ into a scheme $\beta$, where $\alpha$ and $\beta$ are both schemes using the same set $\Phi_N = \{F_1, \ldots, F_n\}$ of undefined function variables. The result of the substitution is denoted $\alpha \ast \beta$, and it is the scheme whose $i$th equation's right-hand side is equal to $\alpha \ast \beta_i$, where $\alpha \ast \beta_i$ is defined recursively as follows:

(i) If $\beta_i = a$ then $\alpha \ast \beta_i = a = \beta_i$.
(ii) If $\beta_i = x_i^p$ then $\alpha \ast \beta_i = x_i^p = \beta_i$.
(iii) If $\beta_i = [t'_1, \ldots, t'_k] \circ f$ then

$$\alpha \ast \beta_i = [\alpha \ast t'_1, \ldots, \alpha \ast t'_k] \circ f.$$
(iv) If $\beta_i = [t'_1, \ldots, t'_m] \circ F_j$ then

$$\alpha \ast \beta_i = [\alpha \ast t'_1, \ldots, \alpha \ast t'_m] \circ \alpha_j.$$

We note that the substitution operation $\ast$ acts as a homomorphism, and this allows us to give a more concise definition which is more convenient for proving properties about it. In fact, this equivalent definition is just as simple in the general case. The idea is to use Theorem 2.4. Let $\alpha: \overline{u} \rightarrow \overline{v}$ and $\beta: \overline{v} \rightarrow \overline{w}$ be two extended schemes. Since $\alpha$ is a function $\alpha: \Phi \rightarrow T_{\Sigma \cup \Phi}$ and $\beta$ is a function $\beta: \overline{w} \rightarrow T_{\overline{w}}$, we have the diagram:

If we extend $\alpha$ to $\Sigma \cup \Phi_{\overline{u}}$ by making $\alpha$ the identity on $\Sigma$, by Theorem 2.4, there exists a unique theory homomorphism $\overline{\alpha}: T_{\Sigma \cup \Phi_{\overline{u}}} \rightarrow T_{\Sigma \cup \Phi_{\overline{u}}}$ extending $\alpha$, and we define $\alpha \ast \beta$ as $\beta \cdot \overline{\alpha}$.

Equivalently, we have $(\alpha \ast \beta)_i = \overline{\alpha}(\beta_i)$ for all $i$, $1 \leq i \leq q$, where $|\overline{w}| = q$.

4.1. Definition. Given any two parameterized recursion schemes $\alpha: \overline{u} \rightarrow \overline{v}$ and $\beta: \overline{v} \rightarrow \overline{w}$, the result of substituting $\alpha$ into $\beta$ is denoted $\alpha \ast \beta$ and is defined by the identity $\alpha \ast \beta = \beta \cdot \overline{\alpha}$, where $\overline{\alpha}$ is the unique theory homomorphism extending $\alpha$, as explained above. Equivalently, for all $i$, $1 \leq i \leq q$, $(\alpha \ast \beta)_i = \overline{\alpha}(\beta_i)$.

We have just defined the operation of substitution $\ast$ for all schemes in $PRS\Sigma$. In the same manner, using Theorem 2.4 we can define the operation $\ast$ for all schemes in $FRPS\Sigma$ and all schemes in $CPRS\Sigma$, which is rather remarkable in the case of $CPRS\Sigma$ which contains infinite trees. Our definition pays off even more when we prove that the operation $\ast$ is associative.

4.2. Lemma. Let $\alpha: \overline{u} \rightarrow \overline{v}$, $\beta: \overline{v} \rightarrow \overline{w}$ and $\gamma: \overline{w} \rightarrow \overline{z}$ be three recursion schemes in $PRS\Sigma$. The operation $\rightarrow$ is associative, that is, $(\alpha \ast \beta) \cdot \gamma = \alpha \ast (\beta \ast \gamma)$. 
Proof. Consider the diagram:

![Diagram]

We know that \( \alpha \cdot \beta = \beta \cdot \bar{\alpha} \), \( \beta \cdot \gamma = \gamma \cdot \bar{\beta} \), \( (\alpha \cdot \beta) \cdot \gamma = \gamma \cdot (\bar{\beta} \cdot \bar{\alpha}) \) and \( \alpha \cdot (\beta \cdot \gamma) = (\gamma \cdot \bar{\beta}) \cdot \bar{\alpha} \). Therefore, it suffices to show that \( (\beta \cdot \bar{\alpha}) = \bar{\beta} \cdot \bar{\alpha} \). But \( \bar{\beta} \cdot \bar{\alpha} \) is a homomorphism extending \( \beta \cdot \bar{\alpha} \) since \( \bar{\beta} \) extends \( \beta \), and since there is a unique homomorphism with this property which is precisely \( (\beta \cdot \bar{\alpha}) \), we have \( (\beta \cdot \bar{\alpha}) = \bar{\beta} \cdot \bar{\alpha} \) as desired.

Again, the exact same proof applies when \( T_\Sigma \) is replaced by \( FT_\Sigma \) or \( CT_\Sigma \), and so, the operation \( * \) is associative for schemes in \( FPRS_\Sigma \) and in \( CPRS_\Sigma \).

For every string \( \bar{u} \in D(S)^+ \), where \( \bar{u} = (u_1, s_1) \cdots (u_n, s_n) \), we have a scheme denoted \( I_{\bar{u}} \), where \( I_{\bar{u}} : \Phi_{\bar{u}} \rightarrow T_{\Sigma \cup \Phi_{\bar{u}}} \) is the injection of \( \Phi_{\bar{u}} \) into \( T_{\Sigma \cup \Phi_{\bar{u}}} \), and called the identity scheme associated with \( \bar{u} \).

It is obvious that every \( I_{\bar{u}} \) is an identity for \( * \); that is, for all schemes \( \alpha : \bar{u} \rightarrow \bar{v} \), we have \( \alpha \cdot I_{\bar{u}} = I_{\bar{u}} \cdot \alpha = \alpha \). We also have projection schemes denoted \( \pi_i : \bar{u} \rightarrow (u_i, s_i) \) defined by

\[
F^i(u_1, s_1, \ldots, u_n) = F^i(u_i, \ldots, x_k),
\]

where \( |u_i| = k \), picking out the \( i \)th equation in the definition of a scheme. Finally, if \( \bar{u} \in D(S)^* \) and \( \bar{v} \in D(S)^+ \), where \( v = (v_1, r_1) \cdots (v_p, r_p) \), given any \( p \) recursion schemes \( \alpha_1 : \bar{u} \rightarrow (v_1, r_1), \ldots, \alpha_p : \bar{u} \rightarrow (v_p, r_p) \), we can form the scheme \( \{\alpha_1, \ldots, \alpha_p\} : \bar{u} \rightarrow \bar{v} \) by tupling. The \( i \)th component of \( \{\alpha_1, \ldots, \alpha_p\} \) is \( F^i \). In other words, we have just verified that \( PRS_\Sigma \) is an algebraic theory (for all \( \bar{u} \in D(S)^* \) we add the degenerate scheme \( 0_{\bar{u}} : \emptyset \rightarrow T_{\Sigma \cup \Phi_{\bar{u}}} \) from the empty set). \( PRS_\Sigma \) is an algebraic theory based on the alphabet \( D(S) \), called the derived alphabet of \( S \). We summarize these properties in the following.

4.3. THEOREM. Let \( S \) be a set of sorts and \( \Sigma \) a ranked alphabet indexed by \( D(S) = S^* \times S \). The set of parameterized recursion schemes \( PRS_\Sigma \) together with the operation of scheme-composition \( * \) is an algebraic \( D(S) \)-theory denoted \( PRS_\Sigma \).

We will show in the next section that \( FPRS_\Sigma \) is an ordered algebraic \( D(S) \)-theory and that \( CPRS_\Sigma \) is an \( \omega \)-continuous algebraic\( D(S) \)-theory. We now turn to the definition of a generalized interpretation.
5. INTERPRETATIONS AND FUNCTIONALS

Let \( S \) be a set of sorts and let \( \Sigma \) be a ranked alphabet indexed by \( D(S) = S^* \times S \).

5.1. DEFINITION. A \textit{generalized interpretation} is a pair \((I, T)\), where \( T \) is an \( \omega \)-continuous algebraic \( S \)-theory, and \( I: \Sigma \to T \) is a function such that, for every symbol \( f \in \Sigma_{u,s} \), \( I(f) \) is an arrow \( I(f): u \to s \) in the \( S \)-theory \( T \).

When there is only one type, and we choose the \( \omega \)-continuous algebraic theory \( CT(A) \) defined in Section 2 we have the standard notion of interpretation.

When the \( S \)-theory \( T \) is given and is assumed to remain fixed, we usually refer to an interpretation \((I, T)\) as an interpretation \( I \). Intuitively, every \( I(f): u \to s \) represents a "function" of type \( s \) and of arity \( u \).

Given a parameterized recursion scheme \( \alpha: \vec{u} \to \vec{v} \) and a generalized interpretation \( I: \Sigma \to T \), the pair \((\alpha, I)\) defines a functional as we now explain. Let \( \vec{u} = (u_1, s_1) \cdots (u_n, s_n) \) and \( \vec{v} = (v_1, r_1) \cdots (v_p, r_p) \). Then, \((\alpha, I)\) defines a functional denoted \( \alpha_I \), where \( \alpha_I: T(u_1, s_1) \times \cdots \times T(u_n, s_n) \to T(v_1, r_1) \times \cdots \times T(v_p, r_p) \), and if \( \vec{u} = \lambda, \vec{v} \neq \lambda \), \( \alpha_I \) is a \( p \)-tuple of elements in \( T(v_1, r_1) \times \cdots \times T(v_p, r_p) \), if \( \vec{v} = \lambda \), \( \alpha_I \) is the constant function with target \{\lambda\}. To define the functional \( \alpha_I \), we show how to define its value \( \alpha_I(a_1, \ldots, a_n) \) for every \( n \)-tuple of elements in \( T(u_1, s_1) \times \cdots \times T(u_n, s_n) \). For this, we note that any \( n \)-tuple of elements \((a_1, \ldots, a_n)\) in \( T(u_1, s_1) \times \cdots \times T(u_n, s_n) \) corresponds to a unique function \( a: \Phi_{\vec{u}} \to T(u_1, s_1) \times \cdots \times T(u_n, s_n) \), the function such that \( a(\vec{u}) = a_i \), and conversely, any such function determines a unique \( n \)-tuple \((a_1, \ldots, a_n)\).

To simplify the notations, let \( T^\vec{u} = T(u_1, s_1) \times \cdots \times T(u_n, s_n) \), \( T^\vec{v} = T(v_1, r_1) \times \cdots \times T(v_p, r_p) \), with the convention that when \( \vec{u} = \lambda \) or \( \vec{v} = \lambda \), \( T^\lambda = \{\lambda\} \). Then, since \( I \) is a function \( I: \Sigma \to T \), and any \( n \)-tuple \( a \in T^\vec{u} \) corresponds uniquely to a function \( a: \Phi_{\vec{u}} \to T^\vec{u} \), \( I \) and \( a \) together define a function denoted \( \alpha_I \), where \( \alpha_I: \Sigma \cup \Phi_{\vec{u}} \to T \). Since the scheme \( \alpha: \vec{u} \to \vec{v} \) is also defined by a function \( \alpha: \Phi_{\vec{u}} \to T_{\Sigma \cup \Phi_{\vec{u}}} \), we have the diagram:

By Theorem 2.4 there is a unique theory homomorphism \( \bar{\alpha}_I: T_{\Sigma \cup \Phi_{\vec{u}}} \to T \) extending \( \alpha_I \), and we define the value \( \alpha_I(a) \) of the functional \( \alpha_I \) at \( a \), as \( \alpha_I(a) = \alpha \cdot \bar{\alpha}_I \). Actually, \( \alpha_I(a) \) is a function from \( \Phi_{\vec{u}} \) to \( T^\vec{u} \), but as we said above, there is a bijection between the set of functions \( b: \Phi_{\vec{u}} \to T^\vec{u} \) and \( T^\vec{u} \) itself. Equivalently, we can define \( \alpha_I(a) \) as \((\bar{\alpha}_I(a_1), \ldots, \bar{\alpha}_I(a_p))\), where \( a_j = \alpha(\bar{I}(a_j)) \). In the limit case \( \vec{u} = \lambda \), \( \Phi_{\vec{u}} = \emptyset \), and in this case we replace \( \alpha_I(a) \) by \( I \), obtaining the homomorphism \( \bar{I}: T_{\Sigma} \to T \). Then, we define the functional \( \alpha_I: \{\lambda\} \to T^\vec{u} \) as the \( p \)-tuple \((\bar{I}(a_1), \ldots, \bar{I}(a_p)) = \bar{I}(a) \). This leads us to the following definition.
5.2. DEFINITION. Let $\alpha : \bar{u} \rightarrow \bar{v}$ be a parameterized scheme and $I : \Sigma \rightarrow T$ be an extended interpretation. The pair $(\alpha, I)$ defines the functional $\alpha_I : T^{\bar{u}} \rightarrow T^{\bar{v}}$, where for all $a \in T^{\bar{u}}$,

$$
\alpha_I(a) = (\bar{a}_I(a_1), \ldots, \bar{a}_I(a_p)), \quad a_I = a(F_i^{\bar{u}})
$$

and $\bar{a}_I$ is the unique theory homomorphism extending $a_I$ as explained above. Equivalently, $\alpha_I(a) = a \cdot \bar{a}_I$, with the convention that $a_I = I$ when $\bar{u} = \lambda$.

The same definition applies without any change to the schemes in $\text{FPRS}_\Sigma$ and the schemes in $\text{CPRS}_\Sigma$.

Given two schemes $\alpha : \bar{u} \rightarrow \bar{v}$ and $\beta : \bar{v} \rightarrow \bar{w}$, the functional $(\alpha * \beta)_I$ associated with the substitution of $\alpha$ into $\beta$ is precisely equal to $\alpha_I \cdot \beta_I$, the result of composing the functionals associated, respectively, with $\alpha$ and $\beta$. In other words, substitution of schemes corresponds to composition of functionals as expected.

5.3. THEOREM. Let $\alpha : \bar{u} \rightarrow \bar{v}$ and $\beta : \bar{v} \rightarrow \bar{w}$ be two extended schemes, and let $I : \Sigma \rightarrow T$ be an interpretation. We have the identity, $(\alpha * \beta)_I = \alpha_I \cdot \beta_I$.

Proof: The proof is identical to the proof of associativity given in Lemma 4.2 and results from the fact that $E_{\beta} = \alpha_{\beta}$, as indicated by the following diagram:

As before, we note that Theorem 5.3 also holds for all schemes in $\text{FPRS}_\Sigma$ and all schemes in $\text{CPRS}_\Sigma$, using Theorem 2.4.

We also notice that the functional $(x^\bar{u})_I$ associated with the projection scheme $x^\bar{u}$ is the projection function

$$(x^\bar{u})_I : T(u_1, s_1) \times \cdots \times T(u_n, s_n) \rightarrow T(u_i, s_i)$$

sending $T^{\bar{u}}$ on its $i$th factor $T(u_i, s_i)$. Combining this observation with Theorems 4.3 and 5.3, we obtain the fact that the set of functionals form a $\text{PRS}_\Sigma$-algebra, as defined at the end of Section 2. To be more explicit, for every $(u, s) \in D(\Sigma)$, we have a carrier $T(u, s)$, for every

$$
\bar{u} = (u_1, s_1) \cdots (u_n, s_n),
$$

and $\bar{T}$ is the function which assigns to every $\alpha : \bar{u} \rightarrow \bar{v}$ the function $\bar{I}(\alpha) : T^{\bar{u}} \rightarrow T^{\bar{v}}$ with $\bar{I}(\alpha) = \alpha_I$. Now, since $\bar{I}(\alpha * \beta) = \bar{I}(\alpha) \cdot \bar{I}(\beta)$ and $\bar{I}(x^\bar{u}) : T^{\bar{u}} \rightarrow T(u_i, s_i)$ is the $i$th projection, it is clear that the pair $((T(u, s))_{(u,s) \in D(\Sigma)}, \bar{I})$ constitutes a $\text{PRS}_\Sigma$-algebra denoted $\text{RFA}_I$ (Recursive Functional Algebra). The set of functions of the form $\alpha_I$:
$T^\delta \to T^\delta$ is also an algebraic theory denoted $\text{RFA}_T$, and its definition is analogous to the algebraic theory $CT(A_\lambda)$ described in Example 2.2. Every $\text{RFA}_T(\bar{u}, \bar{v})$ consists of all functionals of the form $\alpha : T^\delta \to T^\delta$, where $\alpha : \bar{u} \to \bar{v}$ is any recursion scheme in $\text{PRS}_T(\bar{u}, \bar{v})$, and composition is simply functional composition. Also from Theorem 5.3 and the above remarks, $\bar{I}$ is a homomorphism between the $D(S)$-theories $\text{PRS}_T$ and $\text{RFA}_T$. This can be rephrased by saying that the interpretation $I : \Sigma \to T$ extends to a theory homomorphism $\bar{I} : \text{PRS}_T \to \text{RFA}_T$. Summarizing the above facts, we state for the record:

5.4. THEOREM. For every interpretation $I : \Sigma \to T$, the set of all functionals of the form $\alpha_I : T^\delta \to T^\delta$, where $\alpha : \bar{u} \to \bar{v}$ is any recursion scheme in $\text{PRS}_T(\bar{u}, \bar{v})$, forms an algebraic $D(S)$-theory $\text{RFA}_T$ under functional composition, the pair $(\{ T(u,s) \}((u,s) \in D(S)), \bar{I})$ is a $\text{PRS}_T$-algebra $\text{RFA}_T$, and $\bar{I} : \text{PRS}_T \to \text{RFA}_T$ is a homomorphism of $D(S)$-theories.

Intuitively, saying that $\text{RFA}_T$ is a $D(S)$-theory means that the set of functionals of the form $\alpha_I : T^\delta \to T^\delta$ is closed under composition and under tupling. Theorem 5.4 also holds for the functionals defined by schemes in $\text{FRPS}_T$ and in $\text{CPRS}_T$, and we obtain algebraic theories $\text{FRFA}_T$ and $\text{CRFA}_T$ (and also a $\text{FPRS}_T$-algebra $\text{FRFA}_T$ and $\text{CPRS}_T$-algebra $\text{CRFA}_T$).

We now prove that, if the theory $T$ in an interpretation $I : \Sigma \to T$ is an ordered $S$-theory, then the functionals $\alpha_I$ are monotonic, and if $T$ is an $\omega$-continuous $S$-theory, the functionals $\alpha_I$ are $\omega$-continuous. This will allow us to define the meaning of a program $(\alpha, I)$ defined by a closed scheme $\alpha : \bar{u} \to \bar{v}$, as the least fixpoint of the functional $\alpha_I : T^\delta \to T^\delta$. Let us first assume that $T$ is an ordered $S$-theory.

5.5. THEOREM. Let $(T, I)$ be an interpretation where $T$ is an ordered algebraic $S$-theory. For every scheme $\alpha : \bar{u} \to \bar{v}$ in $\text{PRS}_T$, the functional $\alpha_I : T^\delta \to T^\delta$ is monotonic, and similarly for schemes in $\text{FRPS}_T$ and in $\text{CPRS}_T$.

Proof. We first note that for every scheme $\alpha : \bar{u} \to \bar{v}$, where $\bar{v} = (v_1, r_1) \cdots (v_p, r_p)$ and $\alpha = (\alpha(F'_i))$, the functional $\alpha_I$ is equal to the $p$-tuple $((\alpha_I)_1, \ldots, (\alpha_I)_p)$. Therefore, we can assume without loss of generality that $\alpha$ has a unique component, that is, $\bar{v} = (v, r) \in D(S)$. Then, the proof proceeds by induction on the structure of the tree $\alpha$. Let $a, b \in T^\delta$ such that $a \leq b$. (The ordering on $T^\delta$ is the ordering componentwise, that is, $(a_1, \ldots, a_n) \leq (b_1, \ldots, b_n)$, if and only if for all $i, 1 \leq i \leq n, a_i \leq b_i$ in $T(u_i, s_i)$.) We have to show that $\alpha_I(a) \leq \alpha_I(b)$, that is, using Definition 5.2, we have $\bar{a}_i(a) \leq \bar{b}_i(b)$. The case $\bar{u} = \lambda$ is obvious, since in this case, $\alpha_I$ is simply the element $\bar{I}(\lambda) \in T(v, r)$, where $\bar{I}$ is the unique theory homomorphism extending $I$. If $\alpha = a, \alpha_I$ is the constant functional with value $I(a)$, which is monotonic. If $\alpha = x_i y_i$ (assuming $u_i = u_{i1} \cdots u_{in}$), $I(x_i y_i)_{\bar{I}}$ is the constant functional with value the projection arrow $x_i y_i : u_i \to u_{ij}$ in $T(u_i, u_{ij})$, which is monotonic. If $\alpha = [t_1, \ldots, t_k] \circ f,$

$$\alpha_I(a) = \bar{a}_i(a) \quad \text{(since $\bar{a}_i$ is a theory homomorphism)}$$

$$= [\bar{a}_i(t_1), \ldots, \bar{a}_i(t_k)] \circ I(f)$$
Similarly, \( a_i(b) = [\bar{b}_i(t_1), ..., \bar{b}_i(t_k)] \circ I(f) \). By the inductive hypothesis, for \( i, 1 \leq i \leq k \), we have \( \bar{a}_i(t_i) \leq \bar{b}_i(t_i) \), and since \( T \) is an ordered algebraic theory, tupling and composition are monotonic, and so, we have

\[
[\bar{a}_i(t_1), ..., \bar{a}_i(t_k)] \circ I(f) \leq [\bar{b}_i(t_1), ..., \bar{b}_i(t_k)] \circ I(f),
\]

that is, \( a_i(a) \leq a_i(b) \). Finally, if \( \alpha = [t_1, ..., t_k] \circ F^\alpha \), we have \( a_i(\alpha) = \bar{a}_i(\alpha) = [\bar{a}_i(t_1), ..., \bar{a}_i(t_k)] \circ \bar{a}_i(F^\alpha) \). But \( \bar{a}_i \) extends \( a \), so \( \bar{a}_i(F^\alpha) = a(F^\alpha) = a_j \) (by definition of \( a \)). Since by hypothesis we have \( a \leq b \), we have \( a_j \leq b_j \), and the rest of the proof is the same as in the previous case, using the inductive hypothesis \( \bar{a}_i(t_i) \leq \bar{b}_i(t_i) \) and the monotonicity of composition and tupling. Therefore, we have shown that the functional \( a_i \) is monotonic for finite schemes in \( \text{PRS}_1 \).

The same proof extends immediately to the schemes in \( \text{FRPS}_1 \), with the only difference that for every \( \alpha = \perp_{u_i} \), we have a constant functional equal to the least element \( \perp_{u_i} \) of \( T(u_i, u_i) \). To extend the result to infinite schemes, we use the fact that every infinite tree \( t \) is the least upper bound of the \( \omega \)-chain of truncations \( \{t^n\} \) and the fact that from Theorem 2.4, the homomorphism \( \bar{a}_i \) is also \( \omega \)-continuous. Then, if \( a \leq b \), \( a_i(\alpha) = \bar{a}_i(\alpha) = \bar{a}_i(\alpha) \cup \alpha^{(n)} = (\text{by } \omega \text{-continuity}) \cup \bar{a}_i(\alpha^{(n)}) \), and similarly, \( a_i(b) = \bar{b}_i(\alpha) = \bar{b}_i(\alpha^{(n)}) \). Since the schemes \( \alpha^{(n)} \) are finite, by the above result we know that \( \bar{a}_i(\alpha^{(n)}) \leq \bar{b}_i(\alpha^{(n)}) \), and therefore, we conclude that \( \cup \bar{a}_i(\alpha^{(n)}) \leq \cup \bar{b}_i(\alpha^{(n)}) \), that is, \( a_i(a) \leq a_i(b) \) and the proof is complete.

We now show that, if \( T \) is an \( \omega \)-continuous algebraic \( S \)-theory, the functionals \( a_i \) are \( \omega \)-continuous.

5.6. Theorem. Let \( (T, I) \) be an interpretation where \( T \) is an \( \omega \)-continuous algebraic \( S \)-theory. For every scheme \( \alpha: u \rightarrow \bar{v} \) in \( \text{PRS}_1 \), the functional \( a_i: \bar{v} \rightarrow \bar{v} \) is \( \omega \)-continuous, and similarly for all schemes in \( \text{FRPS}_1 \) and in \( \text{CPRS}_1 \).

Proof. Without loss of generality, we can assume that \( \alpha \) consists of a single component. Let \( \{a_i\}_{i \in \omega} \) be an \( \omega \)-chain in \( \bar{v} \). We want to prove that \( a_i(\cup \{a_i\}_{i \in \omega}) = \cup \{a_i(a_i)\}_{i \in \omega} \). Since by the previous theorem \( a_i \) is monotonic, \( \{a_i(a_i)\}_{i \in \omega} \) is an \( \omega \)-chain and so, \( \{(a_i)_i(a)\}_{i \in \omega} \) is an \( \omega \)-chain. Let \( h \) be the function \( h: \text{CT}_{\Sigma \cup \Phi^\bar{v}} \rightarrow T \) defined for all \( \alpha \in \text{CT}_{\Sigma \cup \Phi^\bar{v}} \) by \( h(\alpha) = \cup \{(a_i)_i(a)\}_{i \in \omega} \), where \( (a_i)_i \) is the unique theory homomorphism extending \( a_i \) and \( I \), as in the diagram:

We claim that \( h \) is an \( \omega \)-continuous theory homomorphism extending \( (\cup \{a_i\}_{i \in \omega})_I \). Since such a homomorphism is unique, we have \( h = (\cup \{a_i\}_{i \in \omega})_I \), and since \( a_i(\cup \{a_i\}_{i \in \omega}) = (\cup \{a_i\}_{i \in \omega})_I(\alpha) = h(\alpha) = (\text{by definition of } h) \cup \{(a_i)_i(a)\}_{i \in \omega} = \cup \{a_i(a_i)\}_{i \in \omega} \), we have just shown that \( a_i \) is \( \omega \)-continuous. It remains to show that \( h \) has the properties mentioned above. It is obvious that \( h \) extends \( (\cup \{a_i\}_{i \in \omega})_I \), since
RECURSION-CLOSED ALGEBRAIC THEORIES

\[ \bigcup \{ \{ a_i \} \} \subset M = \text{(since } \{ a_i \} \subset M \text{ by } \omega\text{-continuity of } \{ a_i \} \text{)} \bigcup \{ a'_i \} \subset \omega, \]  

\[ h \text{ is } \omega\text{-continuous because for any } \omega\text{-chain } \{ \beta_i \} \subset \omega, \]  

\[ \bigcup \{ \{ a_i \} \} = \bigcup \{ \{ a'_i \} \} \text{ by } \omega\text{-continuity of } \{ a_i \}. \]  

\[ \text{Finally, } h(\alpha \circ \beta) = \bigcup \{ a_i \} (\alpha \circ \beta = (\text{since } \{ a_i \} \text{ is a homomorphism}) \]  

\[ \bigcup \{ \{ a_i \} \} (\alpha \circ \beta) = (\text{by } \omega\text{-continuity of composition } \circ) (\bigcup \{ a_i \}) \circ (\bigcup \{ \{ a_i \} \} (\beta)) = h(\alpha) \circ h(\beta). \]  

Therefore, \( h \) is a theory homomorphism, and this completes the proof.

The ordering on \( \mathcal{C}_T \) induces an ordering on the schemes in \( \mathbf{CPRS}_\Sigma \) (and similarly for \( \mathbf{FPRS}_\Sigma \)). Given two schemes \( \alpha: \mathcal{A} \rightarrow \mathcal{B} \) and \( \beta: \mathcal{V} \rightarrow \mathcal{W} \), where \( \alpha = (\alpha_1, \ldots, \alpha_p) \) and \( \beta = (\beta_1, \ldots, \beta_p) \), we define \( \alpha \leq \beta \) between schemes if and only if for all \( i, 1 \leq i \leq p, \alpha_i \leq \beta_i \) as trees in \( \mathcal{CT}_{\xi} (v_i, r_i) \). Then, it is obvious that every \( \mathbf{CPRS}_\Sigma (\mathcal{U}, \mathcal{V}) \) is an \( \omega\)-complete poset, with least element \( \bot_{\mathcal{U}, \mathcal{V}} = (\bot, \ldots, \bot) \). It only remains to prove that the operation of substitution * is \( \omega\)-continuous in both arguments to show that \( \mathbf{CPRS}_\Sigma \) is an \( \omega\)-continuous algebraic \( D(S) \)-theory since all the other conditions are met. To prove \( \omega\)-continuity on the right, we go back to Definition 4.1. Given two schemes \( \alpha: \mathcal{A} \rightarrow \mathcal{B} \) and \( \beta: \mathcal{V} \rightarrow \mathcal{W} \), \( \alpha * \beta = (\alpha \circ \beta) \) (where \( \mathcal{U} \subset \mathcal{V} \)). Assuming without loss of generality that \( q = 1 \), for any \( \omega\)-chain \( \{ \beta_i \} \subset \omega \) of schemes \( \beta_i: \mathcal{B} \rightarrow \mathcal{W} \), we have, for every \( p\)-tuple \( (\beta_1, \ldots, \beta_p) \) of trees \( \beta_i \) in \( \mathcal{CT}_{\Sigma} (v_i, r_i) \) (this time \( \beta: \mathcal{B} \rightarrow \mathcal{W} \) and \( \alpha: \mathcal{A} \rightarrow \mathcal{B} \)), we have \( \alpha_i (\beta) = \beta_i (\alpha) \), which is precisely \( \beta * \alpha \). In other words, \( \alpha_i \) performs substitution on the left in \( \alpha \). By Theorem 5.6 the functional \( \alpha_i \) is \( \omega\)-continuous, and therefore, * is \( \omega\)-continuous on the left. Consequently, the substitution operation * is \( \omega\)-continuous in both arguments, and so, \( \mathbf{FPRS}_\Sigma \) is an ordered algebraic \( D(S) \)-theory and \( \mathbf{CPRS}_\Sigma \) is an \( \omega\)-continuous algebraic \( D(S) \)-theory. Since \( \mathbf{CPRS}_\Sigma \) is an \( \omega\)-continuous algebraic \( D(S) \)-theory and the functions \( \alpha_i: T^\mathcal{B} \rightarrow T^\mathcal{A} \) are \( \omega\)-continuous, \( \mathbf{RFRA}_\Sigma \) is an \( \omega\)-continuous \( \mathbf{CPRS}_\Sigma \)-algebra, and \( \mathbf{FRFA}_\Sigma \) is a strict ordered \( \mathbf{FPRS}_\Sigma \)-algebra. \( \mathbf{FRFA}_\Sigma \) and \( \mathbf{CRFA}_\Sigma \) are also ordered \( D(S) \)-theories, but we are unable to show that every \( \mathbf{CRFA}_\Sigma (\mathcal{U}, \mathcal{V}) \) is \( \omega\)-complete, and in fact, we conjecture that this is false in general. The above results are summarized in the following theorem.

5.7. Theorem. Let \( S \) be a set of sorts and let \( \Sigma = \{ \Sigma_{u,v} \} \subset D(S) \) be a ranked alphabet indexed by \( D(S) = S^* \times S \). Then, the following properties hold:

(1) The set \( \mathbf{PRS}_\Sigma \) of finite total recursion schemes is an algebraic \( D(S) \)-theory; the set \( \mathbf{FPRS}_\Sigma \) of finite partial recursion schemes is an ordered algebraic \( D(S) \)-theory; the set \( \mathbf{CPRS}_\Sigma \) of all finite and infinite recursion schemes is an \( \omega\)-continuous algebraic \( D(S) \)-theory.

(2) For every extended interpretation \( I: \Sigma \rightarrow T \), the following holds:

(i) The set \( \mathbf{RFRA}_\Sigma \) of functionals associated with schemes in \( \mathbf{PRS}_\Sigma \) is a \( \mathbf{PRS}_\Sigma \)-algebra. In addition, \( \mathbf{RFA}_\Sigma \) is an algebraic \( D(S) \)-theory.
(ii) If $T$ is an ordered $S$-theory, the set $\text{FRFA}_T$ of functionals associated with schemes in $\text{FPRS}_T$ is an ordered $\text{FPRS}_T$-algebra.

(iii) If $T$ is an $\omega$-continuous algebraic $S$-theory, the set $\text{CRFA}_T$ of functionals associated with schemes in $\text{CPRS}_T$ is an $\omega$-continuous $\text{CPRS}_T$-algebra. In addition, both $\text{FRFA}_T$ and $\text{CRFA}_T$ are ordered algebraic $D(S)$-theories.

We now turn to the study of fixpoint solutions of functionals defined by recursion schemes.

6. Fixpoints Solutions and "Recursion-Closed" Algebraic Theories

This section is divided into three subsections. In Section 6.1 we review some definitions and results about rational algebraic theories. We define recursion-closed algebraic theories in Section 6.2 and prove that every recursion-closed theory is rational. In Section 6.3, we give the construction of the structure $\text{RCT}_T$ and prove that it is the free recursion-closed algebraic theory generated by $\Sigma$.

6.1. Rational Theories

In the previous section, we have shown that for any $\omega$-continuous interpretation $I: \Sigma \rightarrow T$ and for any scheme $\alpha: \bar{u} \rightarrow \bar{v}$, the functional $\alpha_I: T^{\bar{u}} \rightarrow T^{\bar{v}}$ is $\omega$-continuous. Consequently, if $\alpha$ is a closed scheme (that is, $\bar{u} = \bar{v}$), $\alpha_I$ has a least fixpoint $(\alpha_I)^- = \bigcup_{I \in \omega} \alpha_I^-(\bot)$. For our purposes, it will be necessary to consider "fixpoints" of "functionals with parameters." To simplify the discussion, assume that we have a functional $F: A^{m+n} \rightarrow A^n$, where $A$ is an $\omega$-complete poset. Holding the first $m$ arguments $(a_1, \ldots, a_m)$ fixed, we obtain a "functional with parameters" $F(a_1, \ldots, a_m): A^n \rightarrow A^n$, and we can solve for the least fixpoint of this new functional with parameters obtaining a functional $F^+: A^m \rightarrow A^n$. This process corresponds to solving for the least fixpoint of the following system of equations where $(a_1, \ldots, a_m)$ is held constant and we solve with respect to the unknown $(x_1, \ldots, x_n)$:

**System 1:**

\[
x_1 = F_1(a_1, \ldots, a_m, x_1, \ldots, x_n) \\
\vdots \\
x_n = F_n(a_1, \ldots, a_m, x_1, \ldots, x_n).
\]

This is equivalent to solving for the system of $n + m$ equations:

**System 2:**

\[
x_1 = a_1 \\
\vdots \\
x_m = a_m \\
x_{m+1} = F_1(x_1, \ldots, x_m, x_{m+1}, \ldots, x_{m+n}) \\
\vdots \\
x_{m+n} = F_n(x_1, \ldots, x_m, x_{m+1}, \ldots, x_{m+n}).
\]
The first \(m\) solutions of this system are \((a_1, \ldots, a_m)\), and the last \(n\) solutions are equal to the \(n\) solutions of the previous system. This process can be adapted to algebraic theories as we now explain.

First, we define an operation which simplifies the presentation. Let \(T\) be an \(S\)-theory. Given any two arrows \(\phi: u \to v\) and \(\psi: u \to w\), where \(v, w \in S^+\), with \(v = v_1 \cdots v_p, w = w_1 \cdots, w_q\), we define \([\phi, \psi]: u \to vw\) as the arrow \([\phi_1, \ldots, \phi_p, \psi_1, \ldots, \psi_q]\) obtained by tupling of the \(p\) components of \(\phi\) and the \(q\) components of \(\psi\). Therefore, this operation is really an abbreviation for tupling of vectors (instead of scalars). We also have projections denoted \(x^{u, w}_1, x^{u, w}_2\), such that \(\phi \circ x^{u, w}_1 = \phi\) and \(\phi \circ x^{u, w}_2 = \psi\) (in fact, \(x^{u, w}_1 = [x^{w}_1, \ldots, x^{w}_n]\) and \(x^{u, w}_2 = [x^{u, w}_1, \ldots, x^{u, w}_p]\)).

6.1.1. DEFINITION. For every arrow \(\alpha: uv \to v\), we define the sequence of elements \(\alpha^{(i)}: u \to uv\) in the following way:

\[
\alpha^{(i)} = [x^u_1, \ldots, x^u_\nu, \downarrow_{u, v_1} \cdots, \downarrow_{u, v_\nu}] = [I_u, \downarrow_{u, v}]
\]

(where \(u = \lambda\) or \(u = u_1 \cdots u_\nu\) and \(v \in S^+\) with \(v = v_1 \cdots v_\nu\)) and

\[
\alpha^{(i+1)} = \alpha^{(i)} \circ [x^u_{1:v}, \ldots, x^u_{n:v}, \alpha_1, \ldots, \alpha_n] = \alpha^{(i)} \circ [x^u_{1:v}, \alpha],
\]

Intuitively, an arrow \(\alpha: uv \to v\) represents a function with \(|u|\) parameters and \(|v|\) variables. The definition of the \(\alpha^{(i)}\) is suggested by the method for solving a system of equations such as System 2. The proof of the following lemma is easily done by induction and is left to the reader.

6.1.2. LEMMA. Let \(T\) be an \(\omega\)-continuous algebraic \(S\)-theory. For any arrow \(\alpha: uv \to v\), we have the following properties:

(i) For all \(i \geq 0\), \(\alpha^{(i)} \leq \alpha^{(i+1)}\).

(ii) For all \(i \geq 0\), \(\alpha^{(i+1)} = [I_u, \alpha^{(i)} \circ \alpha]\).

Since \((\alpha^{(i)})_{i \in \omega}\) is an \(\omega\)-chain in \(T(u, uv)\) and \(T\) is \(\omega\)-complete, \((\alpha^{(i)})_{i \in \omega}\) has a least upper bound, and we define \(\alpha^v\) as \(\sqcup \alpha^{(i)}\). Picking out the last \(n\) components of \(\alpha^v\) by projection, we define \(\alpha^+\) as \(\alpha^+ = [x^{uv}_1, \ldots, x^{uv}_{m+n}] = \alpha^v \circ x^{uv}_2\). Then, as expected, \(\alpha^+\) is the least fixpoint of the equation \(\eta = [I_u, \eta] \circ \alpha\), analogous to the system of equations System 1. The proof of the following lemma can be found in [21].

6.1.3. LEMMA. Let \(T\) be an \(\omega\)-continuous algebraic \(S\)-theory. For any arrow \(\alpha: uv \to v\), \(\alpha^+ : u \to v\) is the least solution of the equation \(\eta = [I_u, \eta] \circ [I_u, \eta] \circ \alpha\).

We note that in the case where \(u = \lambda\), \(\alpha^v\) and \(\alpha^+\) are identical and they are both "constant" arrows in \(T(\lambda, v)\). In terms of the system of equations System 2, this means that the solutions are constants and not functions, since there are no parameters.

Given two \(\omega\)-continuous algebraic \(S\)-theories \(T_1, T_2\) and an \(\omega\)-continuous
homomorphism $h$ between them, by a straightforward proof by induction, it is easily shown that $h$ preserves the operations $\lor$ and $\land$. This result is also shown by Goguen et al. [21]. We note for the record:

6.1.4. **Lemma.** Let $h$ be an $\omega$-continuous theory homomorphism between two $\omega$-continuous algebraic $S$-theories $T_1$ and $T_2$. Then, $h$ preserves the operations $\lor$ and $\land$, that is, $h(a^\lor) = h(a)^\lor$ and $h(a^\land) = h(a)^\land$.

The preservation of the operations $\lor$ and $\land$ is a "Mezei, Wright" type of result [29]. This result can be paraphrased by saying that it is equivalent to first solve in $T_1$ for the least solution of the equation associated with $a$ and give it an interpretation in $T_2$ using $h$, or to first give an interpretation in $T_2$ to the equation associated with $a$ using $h$, and then to solve for its least solution in $T_2$.

Given a scheme $\alpha: \tilde{u} \cdot \tilde{v} \rightarrow \tilde{v}$ in CPRS and an $\omega$-continuous interpretation $I: \Sigma \rightarrow T$, the "Mezei, Wright" Theorem holds, that is, we have the identity $\tilde{I}(\alpha^+) = \tilde{I}(a)^+$. To prove this fact, we only have to show that $\tilde{I}$ is an $\omega$-continuous theory homomorphism. We already know from Theorem 5.4 that $\tilde{I}$ is a homomorphism of algebraic theories, with $\tilde{I}: \text{CPRS}_T \rightarrow \text{CRFA}_T$. First of all, we have to take care of a minor technicality. Indeed, the set CRFA, of functionals of the form $\alpha: T_{\tilde{u}} \rightarrow T_{\tilde{v}}$ is not necessarily $\omega$-complete, but this is not a problem, because CRFA, is a subtheory of the algebraic theory $\text{CF}(T)$ of all $\omega$-continuous functions of the form $f: T_{\tilde{u}} \rightarrow T_{\tilde{v}}$, which is $\omega$-complete by Proposition 2.1. Another useful fact which is easily verified is the following: let $(f_\iota)_{\iota \in \omega}$ be an $\omega$-chain of $\omega$-continuous functions $f_\iota: T_{\tilde{u}} \rightarrow T_{\tilde{v}}$. Then, the least upper bound $\tilde{f}$ of the chain $(f_\iota)_{\iota \in \omega}$ is the function defined pointwise by the identity $\tilde{f}(\alpha) = \cup_{\iota \in \omega} f_\iota(\alpha)$. For any $\omega$-chain $(\alpha_\iota)_{\iota \in \omega}$ of schemes $\alpha_\iota: \tilde{u} \cdot \tilde{v} \rightarrow \tilde{v}$ and for any $\alpha \in T^{\tilde{u} \cdot \tilde{v}}$, we have $\tilde{I}(\bigcup \alpha_\iota)(\alpha) = \tilde{\alpha}(\bigcup \alpha_\iota) = (\text{by } \omega\text{-continuity of } \tilde{\alpha}) \cup \tilde{\alpha}(\alpha_\iota) = \cup \tilde{\alpha}(\alpha_\iota)(\alpha)$. By the above remark, the least upper bound $\bigcup \tilde{I}(\alpha_\iota)$ of the $\omega$-chain $(\tilde{I}(\alpha_\iota))_{\iota \in \omega}$ is the function such that for all $\alpha$, $(\bigcup \tilde{I}(\alpha_\iota))(\alpha) = \bigcup \tilde{I}(\alpha_\iota)(\alpha)$, and so, $\tilde{I}(\bigcup \alpha_\iota) = \bigcup \tilde{I}(\alpha_\iota)$, establishing the $\omega$-continuity of $\tilde{I}$. Therefore, $\tilde{I}$ is an $\omega$-continuous homomorphism between the $\omega$-continuous algebraic theories CPRS and $\text{CF}(T)$ and we can apply Lemma 6.1.4 to obtain the "Mezei, Wright" Theorem.

6.1.5. **Theorem.** The function $\tilde{I}: \text{CPRS}_T \rightarrow \text{CF}(T)$ is an $\omega$-continuous homomorphism of $\omega$-continuous algebraic $D(S)$-theories. For any scheme $\alpha: \tilde{u} \cdot \tilde{v} \rightarrow \tilde{v}$ we have the "Mezei, Wright" property: $\alpha^+ \equiv (\alpha_I)^+$. When $\alpha$ is a closed scheme, that is, $\tilde{u} = \lambda$, $\alpha_I$ is, in general, an $n$-tuple $(\alpha^{+}_1, \ldots, \alpha^{+}_n)$ of finite "constant" trees, where each $\alpha^{+}_i$ is a tree without undefined function variables in $\text{CT}_n(u_i, s_i)$. $\alpha^+$ is the "unfoldment" of the recursion scheme $\alpha$.

Given a closed scheme $\alpha: \tilde{u} \rightarrow \tilde{u}$ and an interpretation $I$, we define the meaning of the program $(\alpha, I)$ as the least fixpoint $(\alpha_I)^+$ of the functional $\alpha_I$ associated with $(\alpha, I)$. Equivalently, we can define the meaning of the program $(\alpha, I)$ as the interpretation $(\alpha^+_I)_I$ under $I$, of the unfoldment $\alpha^+$ of $\alpha$. We say that two schemes $\alpha, \beta: \tilde{u} \rightarrow \tilde{u}$ are strongly equivalent, if for all interpretations $I$, we have $(\alpha_I)^+ = (\beta_I)^+$. From the previous theorem, $(\alpha_I)^+ = (\beta_I)^+$ if and only if $(\alpha^+_I)_I = (\beta^+_I)_I$. But in the inter-
pretation \( I: \Sigma \to CT \), with \( I \) the inclusion of \( \Sigma \) into \( CT \), \( I \) is the identity and therefore, if \( \alpha \) and \( \beta \) are strongly equivalent, they are equivalent in the interpretation \( CT \), which shows that \( \alpha^+ = \beta^+ \). Conversely, by the above remark, if \( \alpha^+ = \beta^+ \), for all interpretations we have \( (\alpha^r)^+ = (\beta^r)^+ \), and \( \alpha \) and \( \beta \) are strongly equivalent. Therefore, two closed schemes are strongly equivalent, if and only if their unfoldments \( \alpha^+ \) and \( \beta^+ \) are equal. The interpretation \( CT \) is called a "Herbrand" interpretation by Courcelle and Nivat [10] and is also known under the name of "free interpretation."

Checking carefully the proof in [21] of Lemma 6.1.3, where it is shown that \( \alpha^+ \) is the least solution of the equation \( \eta = [I_\alpha, \eta] \circ \alpha \) we note that we only need the fact that the chain \( (\alpha^{(n)})_{n \in \omega} \) has a least upper bound, and a left continuity condition, namely, that for any arrow \( \beta: uv \to w \), we have \( (\bigcup \alpha^{(n)}) \circ \beta = \bigcup \alpha^{(n)} \circ \beta \). This fact among others, suggests the definition of a more "economical" concept than the concept of an \( \omega \)-complete algebraic theory, provided that all fixpoint calculations remain feasible. The concept of a "rational" algebraic theory, defined and studied by Goguen et al. [21] is a very attractive candidate to fulfill the above goal. "Rational" algebraic theories have also the very nice property of being closed under quotient by a certain kind of congruence, a property which fails for \( \omega \)-continuous algebraic theories. Furthermore, there exists a "free rational algebraic theory generated by a ranked alphabet \( \Sigma \)," a fact which has some interesting consequences. For example, it is shown in [21, 36, 38] that (monadic) flowchart programs can be translated into "regular" recursion schemes (a regular recursion scheme is a scheme in which all "undefined function symbols" \( F_1, ..., F_N \) have arity zero, and so they only appear as leaves). It should be noted that this translation is different from the translation of a flowchart program into a "linear recursion scheme" (see [15, 24]).

Unfortunately, as soon as polyadic function symbols are allowed, it is necessary to introduce "undefined functions symbols" of non null arity, and the above technique is inapplicable. Furthermore, if we are interested in programs defined by unrestricted recursion schemes (not necessarily regular), "rational" algebraic theories are insufficient for another unescapable reason. The reason is that "rational" theories may fail to contain fixpoints of functionals defined by non-regular recursion schemes. Indeed, if we take the “free rational algebraic theory” \( RT \) generated by a ranked alphabet \( \Sigma \) as an interpretation, functionals defined by non-regular recursion schemes over \( \Sigma \) may fail to have a least fixpoint in \( RT \). The reason for this is that the (infinite) trees which constitute the free rational algebraic theory \( RT \) have the property that their set of branches can be encoded by a regular language, as shown by Ginali [16]. However, from Courcelle [8] and Gallier [13] it is known that the least fixpoints of arbitrary recursion schemes are trees whose set of branches can be encoded by deterministic context free languages which are generally non-regular. Hence the rational theory \( RT \) is not closed under fixpoints of non-regular recursion schemes.

The solution that we are proposing to solve this problem, consists essentially in shifting the rationality requirement to the level of functionals. More precisely, given any interpretation \( I: \Sigma \to T \), \( FRFA_I \) denotes the set of "recursive functionals" over \( T \), and we require all such sets of functionals to be rationally closed. This condition is
stronger than requiring $T$ to be rationally closed and it is satisfied by any $\omega$-continuous algebraic theory. We shall call these algebraic theories, "recursion-closed" algebraic theories. We now proceed with the formal definitions. We begin with the definition of a "rational" algebraic theory (Goguen et al. [21]).

6.1.6. Definition. An ordered algebraic $S$-theory $T$ is rational if the following conditions hold:

1. (Completeness.) For all $\alpha: uv \to v$ in $T$, the $\omega$-chain $(\alpha^{(n)})$ has a least upper bound $\alpha^{v} = \bigcup \alpha^{(n)}$.
2. (Right continuity.) For all $\alpha: uv \to v$ and $\beta: w \to u$, $\beta \circ \alpha^{v} = \bigcup \beta \circ \alpha^{(n)}$.
3. (Left continuity.) For all $\alpha: uv \to v$ and $\beta: uv \to w$, $\alpha^{v} \circ \beta = \bigcup \alpha^{(n)} \circ \beta$.

A homomorphism of rational theories is a homomorphism of ordered theories preserving the operator $V$. Lemmas 6.1.3 and 6.1.4 are immediately shown to hold in rational theories, and so, we can solve for fixpoints and apply the "Mezei, Wright" Theorem. One of the main results about rational theories is the existence of the "rational closure" of a strict ordered theory which is a subtheory of another rational theory. This is Theorem 7 of Goguen et al. [21] that we now describe. First of all, let us observe that every $\omega$-continuous algebraic theory is obviously a rational theory. Now, let $T$ be a given rational theory, and let $F$ be an ordered subtheory of $T$. Define $R(u, v)$ to be the set of all arrows of the form $\beta^{v} \circ \alpha: u \to v$, for all $\alpha: uw \to v$ and $\beta: uw \to w$ in $F$ and $u, v, w \in S^*$. Since $\beta^{v}: u \to uw$, we verify that $\beta^{v} \circ \alpha$ is in $T(u, v)$. The remarkable fact is that $R$ is the smallest rational subtheory of $T$ containing $F$. It is called the rational closure of $F$.

6.1.7. Theorem (Goguen et al. [21]). Let $T$ be a given rational theory and let $F$ be an ordered subtheory of $T$. There exists a rational algebraic theory $R$ containing $F$ which is the smallest such theory, and its elements can be described as the set of all arrows of the form $\beta^{v} \circ \alpha: u \to v$, for all $\alpha: uw \to v$ and $\beta: uw \to w$ in $F$ ($u, v, w \in S^*$).

By applying this construction to the algebraic theory of finite partial trees $FT_{\Sigma}$ it is shown in [21] that the rational closure $RT_{\Sigma}$ of $FT_{\Sigma}$ is the free rational theory generated by the ranked alphabet $\Sigma$.

We now come to the definition of a "recursion-closed" algebraic theory.

6.2. Recursion-Closed Algebraic Theories

6.2.1. Definition. Given any ordered algebraic theory $T$, we define the $S$-ranked alphabet $T \Omega = \{ T \Omega_{u,s} \} \cup \{ T \Omega_{u,s} \} \in S \times S$, where $T \Omega_{u,s}$ is the set of symbols $\{ \phi \} \phi \in T(u, s) \}$ in one to one correspondence with the set of arrows in $T(u, s)$. Every symbol $\phi$ is a name for the arrow $\phi: u \to s$ in $T(u, s)$. We also define the interpretation $TI: T \Omega \to T$ such that $TI(\hat{\phi}) = \phi$, that is, $TI$ is the function assigning to each name the arrow it represents.
RECURSION-CLOSED ALGEBRAIC THEORIES

The reason for defining $T\Omega$ and $\Omega I$ is the following. For any arbitrary $S$-ranked alphabet $\Sigma$, any arbitrary finite recursion scheme $a$ over $\Sigma$ and any arbitrary interpretation $I: \Sigma \to T$, there is a recursion scheme over $T\Omega$ denoted $Ta$ such that the functionals $a_I$ and $(Ta)_{\Omega I}$ are identical. Indeed, the scheme $Ta$ is the scheme obtained by renaming every symbol $f \in \Sigma_{n,s}$ with the symbol $I(f)$ corresponding to the arrow $I(f)$ assigned to $f$ by $I$. This property yields immediately the following lemma.

6.2.2. Lemma. Given any algebraic theory $T$, for every finite closed recursion scheme $a$ (over an arbitrary $S$-ranked alphabet $\Sigma$) and for every interpretation $I: \Sigma \to T$, the functional $a_I$ has a least fixpoint in $T$ if and only if for every finite closed recursion scheme $\beta$ over $T\Omega$, the functional $(\beta)_{\Omega I}$ (under the fixed interpretation $I\Omega$) has a least fixpoint.

Noticing that for $\omega$-continuous interpretations $(I, T)$, the least fixpoint of a functional of the form $a_I$ (where $a$ is a finite closed scheme) is given by the identity $(a_I)^\omega \cup a_I^\omega(\perp)$ and the fact that in any ordered algebraic theory the identity $a_I^\omega(\perp) = (a^{(n)}_I)_{n<\omega}$ holds (only monotonicity is needed), we see that the $\omega$-chain $((a^{(n)}_I)_{n<\omega})_{n<\omega}$ has a least upper bound. We will require that in a recursion-closed algebraic theory, for every finite closed scheme $a$ over $T\Omega$, the $\omega$-chain $((a^{(n)}_I)_{n<\omega})_{n<\omega}$ has a least upper bound $a$. We actually need the following slightly stronger conditions in order to prove the existence of free recursion-closed algebraic theories.

6.2.3. Definition. An ordered algebraic $S$-theory $T$ is recursion-closed if the following conditions hold.

(1) (Completeness.) For all finite schemes $a$ and $\beta$ over $T\Omega$, with $\beta: \bar{u} \to \bar{v}$ a closed scheme of type $\bar{u} = (w_1, s_1) \cdots (w_n, s_n)$ and $a: \bar{u} \to \bar{v}$ (not necessarily closed) a scheme of type $(\bar{u}, \bar{v})$, where $\bar{v}$ is of the special form $(u, v_1) \cdots (u, v_p)$, the $\omega$-chain $((\beta(a))_{\Omega I})_{n<\omega}$ has a least upper bound in $T(u, v)$ denoted $(\beta(a))_{\Omega I}$ (with $v = v_1 \cdots v_p$).

(2) (Right continuity.) For all $a$ and $\beta$ as in (1), for all $\phi: w \to u$ in $T$, we have $\phi \circ (\cup (\beta(a))_{\Omega I}) = \cup (\phi \circ (\beta(a))_{\Omega I})$.

(3) (Left continuity.) For all $a$ and $\beta$ as in (1), for all $\phi: v \to w$, we have $(\cup (\beta(a))_{\Omega I}) \circ \phi = \cup ((\cup (\beta(a))_{\Omega I}) \circ \phi)$.

It should be noted that for all $i \in \omega$, $(\beta(a))_{\Omega I}$ is an element of $T(u, v)$, because the special form of $\bar{v}$ implies that $T^v = T(u, v)$, and therefore the above compositions are meaningful.

As a consequence of this definition, we can show immediately that a recursion-closed theory is a rational theory, which is the least we could hope for. Let $\alpha: uv \to v$ be any arrow in $T(uv, v)$, with $\alpha = (\alpha_1, \ldots, \alpha_p)$, $u = u_1 \cdots u_n$ and $v = v_1 \cdots v_p$. Let $\bar{w} = (u, u_1) \cdots (u, u_n)(u, v_1) \cdots (u, v_p)$. We associate with $\alpha$ the scheme $A: \bar{w} \to \bar{\bar{w}}$ defined as follows:
Scheme A:

\[
F_1^u(x_1^u, \ldots, x_n^u) \Leftarrow x_1^u \\
\ldots \\
F_n^u(x_1^u, \ldots, x_n^u) \Leftarrow x_n^u \\
F_{n+1}^u(x_1^u, \ldots, x_n^u) \Leftarrow \alpha_1(x_1^u, \ldots, x_n^u, F_{n+1}^u(x_1^u, \ldots, x_n^u), \ldots, F_{n+1}^u(x_1^u, \ldots, x_n^u)) \\
\ldots \\
F_{n+p}^u(x_1^u, \ldots, x_n^u) \Leftarrow \alpha_p(x_1^u, \ldots, x_n^u, F_{n+1}^u(x_1^u, \ldots, x_n^u), \ldots, F_{n+p}^u(x_1^u, \ldots, x_n^u)).
\]

It is readily verified that for all \(\beta: u \to u, v\), with \(\beta = (\beta_1, \ldots, \beta_n, \beta_{n+1}, \ldots, \beta_{n+p})\), we have

\[
A_T(\beta_1, \ldots, \beta_n, \beta_{n+1}, \ldots, \beta_{n+p}) = \left[ x_1^u, \ldots, x_n^u, [x_1^u, \ldots, x_n^u, \beta_{n+1}, \ldots, \beta_{n+p}] \circ \alpha \right],
\]

which can also be written as

\[
A_T(\beta) = [I_u, I_u, \beta \circ x_2^u(v) \circ \alpha].
\]

Since \(\alpha^{(0)} = [I_u, I_u, v], \alpha^{(n+1)} = \alpha^{(n)} \circ [x_1^u, v, \alpha]\) and \(\alpha^{(n+1)} = [I_u, \alpha^{(n)} \circ \alpha]\), we have

\[
A_T(\bot_{u,uv}) = [I_u, I_u, I_u, \bot_{u,uv}] \circ \alpha = [I_u, \alpha^{(0)} \circ \alpha] = \alpha^{(1)}.
\]

(Note also that \((A^{(0)})_{TI} = \bot_{u,uv}\), so we do not have to worry about \((A^{(0)})_{TI}\). Also, with our choice of \(\tilde{\omega}\), we have \(T^\omega = T(u, uv)\)). Assume by induction that \((A^{(n)})_{TI} = \alpha^{(n)}\). Then, \((A^{(n+1)})_{TI} = (A^{(n)} \times A)_{TI} = (\text{Theorem 5.3}) (A^{(n)})_{TI} = (\text{since \((A^{(n)})_{TI}\) is a constant functional}) A_{TI}((A^{(n)})_{TI}) = (\text{by inductive hypothesis}) A_{TI}((A^{(n)})_{TI}) = A_{TI}([I_u, \alpha^{(n-1)} \circ \alpha]) = [I_u, [I_u, \alpha^{(n-1)} \circ \alpha] \circ \alpha] = [I_u, \alpha^{(n)} \circ \alpha] = \alpha^{(n+1)}). Therefore, the induction is established and this proves that \(\cup \alpha^{(n)} = \cup (A^{(n)})_{TI}\), which shows that \(\alpha^\omega\) exists. Left and right continuity are then easily verified, we leave the details to the reader.

6.2.4. Lemma. Every recursion-closed algebraic theory is a rational algebraic theory.

From Theorems 5.6 and 5.7, since all functionals of the form \(\alpha_T: T^\omega \rightarrow T^\omega\) for any (even infinite) recursion scheme \(\alpha: \tilde{u} \rightarrow \tilde{v}\) in \(\text{CPRS}_{T\Omega}\) are \(\omega\)-continuous, every \(\omega\)-continuous algebraic theory is recursion-closed.

6.2.5. Definition. A homomorphism \(h: T_1 \rightarrow T_2\) between two recursion-closed algebraic theories \(T_1\) and \(T_2\) is a homomorphism of ordered theories such that for all pairs of schemes \(\alpha\) and \(\beta\) as in Definition 6.2.3 we have the identity, \(h(\cup (\beta^{(l)} \ast \alpha)_{TI}) = \cup h((\beta^{(l)} \ast \alpha)_{TI})\).

Definition 6.2.5 implies immediately that a homomorphism of recursion-closed
algebraic theories preserves the operation $V$, that is, it is a homomorphism of rational theories.

The following lemma gives an equivalent definition of a recursion-closed algebraic theory. As mentioned in Section 6.1, this definition shows that the concept of a recursion-closed algebraic theory is obtained by lifting the rationality requirement to the level of "recursive functionals."

6.2.6. **Lemma.** An ordered algebraic theory $T$ is recursion-closed if and only if $(TI: T\Omega \to T$ being the interpretation of Definition 6.2.1) every functional in the ordered algebraic theory $\text{FRFA}_T$ of recursive functionals over $T$ has a least fixpoint.

**Proof.** The proof is straightforward using Lemma 6.2.2 and the fact that least upper bounds of functionals are defined pointwise in terms of least upper bounds of chains in $T$. We leave the details to the reader. □

Recursion-closed algebraic theories fulfill our goal, namely, to find a class of interpretations in which all finite programs defined by recursion schemes and interpretations can be given a meaning by fixpoint semantics. To show this, let $\Sigma = \{(u, u) \mid (u, s) \in D(\Sigma)\}$ be a ranked alphabet indexed by $D(\Sigma)$, let $T$ be a recursion-closed algebraic theory, and let $I: \Sigma \to T$ be an interpretation. We can define a program with main procedure as a pair $((\alpha, \beta), I)$, where $I$ is an interpretation, and $(\alpha, \beta)$ is a pair of schemes, with $\alpha: \bar{u} \to (u, r)$ a finite scheme in $\text{FPRS}_\Sigma$ consisting of a unique component and called the "main program," and $\beta: \bar{u} \to \bar{u}$ a finite closed scheme in $\text{FPRS}_\Sigma$, called a "procedure declaration," with $\bar{u} = (w_1, s_1) \cdots (w_n, s_n)$. Therefore, the pair $(\alpha, \beta)$ represents a main program $\alpha$ of type $r$ and with set of program variables $X_u$, and a set $\beta$ of $n$ procedure declarations, one for each procedure name $F^\bar{u}$ occurring in the main program $\alpha$. Each procedure $\beta_i$ is of type $s_i$ and has a set of program variables $X_{w_i}$. The procedures $\beta_i$ are mutually recursive, and the main program may call any of these procedures, but the main program cannot call itself.

Using Lemma 6.2.2 there exist two schemes $\alpha'$ and $\beta'$ isomorphic to $\alpha$ and $\beta$ by renaming, such that, for all $n \geq 0$, $(\beta^{(n)} \ast \alpha)_t = (\beta^{(n)} \ast \alpha')_t$ and since $T$ is recursion-closed, $\bigcup (\beta^{(n)} \ast \alpha)_t = \bigcup (\beta^{(n)} \ast \alpha')_t$ exists in $T(u, v)$, and so, we can take the meaning of the program $((\alpha, \beta), I)$ as this least upper bound $\bigcup (\beta^{(n)} \ast \alpha)_t$ in $T(u, v)$.

The above discussion shows us that the meaning of the unfoldment of a recursion scheme can be defined as the least upper bound of the $\omega$-chain $(\beta^{(n)} \ast \alpha)_t$. This idea can be exploited to construct the "free recursion-closed algebraic theory" generated by a ranked alphabet $\Sigma$.

6.3. **Free Recursion-Closed Algebraic Theories**

The free recursion-closed algebraic theory $\text{RCT}_\Sigma$ generated by a (many-sorted) ranked alphabet $\Sigma$ is a proper extension of the free rational theory generated by $\Sigma$, and consists of the set of all $p$-tuples of trees in $\text{CT}_\Sigma$, which are of the form $\beta^* \ast \alpha$, where $\alpha$ and $\beta$ are two finite recursion schemes of the form,

$$\alpha: (w_1, s_1) \cdots (w_n, s_n) \to (u, v_1) \cdots (u, v_p)$$
and

\[ \beta: (w_1, s_1) \cdots (w_n, s_n) \rightarrow (w_1, s_1) \cdots (w_n, s_n). \]

The scheme \( \beta \) is always a closed scheme, and so, its least fixpoint \( \beta^\nu \) consists of an \( n \)-tuple of trees, where \( \beta^\nu \) is a tree of type \( s_i \) which may only be labeled with variables in \( X_w = \{ x_1^w, \ldots, x_k^w \} \) and symbols in \( \Sigma \) (no undefined function variables). By substituting the \( n \)-tuple \( \beta^\nu \) in \( \alpha \), all undefined function symbols disappear, and we obtain a \( p \)-tuple of trees, where each tree \( \beta^\nu \ast \alpha_i \) is a tree of type \( v_i \) which may only be labeled with variables in \( X_w \) and symbols in \( \Sigma \). Intuitively speaking when \( \alpha \) is a single tree, we can think of it as the "main program," and we think of \( \beta \) as a "procedure declaration" for the procedure names occurring in \( \alpha \). In fact, this is really what is going on, but we are also interested in the unfoldments of these schemes.

In order to prove that the set of trees defined above, form a recursion-closed algebraic theory under tree-composition, we need a property about fixpoint solutions. We first explain informally the content of this lemma in the one-sorted case. The idea is that, if we have a closed system of \( N \) equations defining a recursion scheme \( \alpha \), we can split \( \alpha \) into two subsystems \( \beta \) and \( \gamma \), \( \beta \) having \( m \) equations and \( \gamma \) having \( n \) equations (\( m, n \geq 1 \), \( N = m + n \)).

System \( \beta \):

\[
\begin{align*}
F_1(x_1, \ldots, x_{k_1}) &\leq \alpha_1(F_1, \ldots, F_m, F_{m+1}, \ldots, F_{m+n}) \\
F_m(x_1, \ldots, x_{k_m}) &\leq \alpha_m(F_1, \ldots, F_m, F_{m+1}, \ldots, F_{m+n}).
\end{align*}
\]

System \( \gamma \):

\[
\begin{align*}
F_{m+1}(x_1, \ldots, x_{k_{m+1}}) &\leq \alpha_{m+1}(F_1, \ldots, F_m, F_{m+1}, \ldots, F_{m+n}) \\
F_{m+n}(x_1, \ldots, x_{k_{m+n}}) &\leq \alpha_{m+n}(F_1, \ldots, F_m, F_{m+1}, \ldots, F_{m+n}).
\end{align*}
\]

The lemma says that to compute the least fixpoint of \( \alpha \), we can first solve for the least fixpoint of \( \gamma \) with respect to \( F_{m+1}, \ldots, F_{m+n} \) holding \( F_1, \ldots, F_m \) as parameters obtaining \( \gamma^\nu \), then substitute \( \gamma^\nu \) for \( (F_{m+1}, \ldots, F_{m+n}) \) in \( \beta \) obtaining \( \beta^\nu \ast \gamma \), and finally solve for the least fixpoint of the substituted system \( \gamma^\nu \ast \beta \). This fact can be expressed concisely as the identity:

\[ \alpha^+ = [\beta, \gamma]^+ = [\gamma^\nu \ast \beta]^+, \quad (\gamma^\nu \ast \beta)^+ \circ \gamma^+ \]

This identity actually holds in arbitrary rational theories, as we shall now prove. This is one of the identities given in Goguen et al. [21] without proof. We first need a technical lemma.

**6.3.1. Lemma.** Let \( T \) be a rational algebraic theory, and let \( \alpha: uv \rightarrow v \).

1. For all \( \beta: u \rightarrow u \), \( \beta \circ \alpha^+ \) is the least solution of the equation \( \eta = [\beta, \eta] \circ \alpha \).
2. For all \( \beta: u \rightarrow u \) and \( \eta: u \rightarrow v \), if \( [\beta, \eta] \circ \alpha \leq \eta \), then we have \( \beta \circ \alpha^+ \leq \eta \) (least fixpoint property).
3. If we split \( \alpha \) into two parts \( \beta: uv \rightarrow v_1 \) and \( \gamma: uv \rightarrow v_2 \) such that \( \alpha = [\beta, \gamma] \) \((v = v_1, v_2, v_1 \neq \lambda \text{ and } v_2 \neq \lambda) \), if \( \alpha^+ = [a_1, \ldots, a_m, a_{m+1}, \ldots, a_{m+n}] \) is the least solution of \( \eta = [I_u, \eta] \circ \alpha \) \((m = |v_1|, n = |v_2|) \), then \( [a_1, \ldots, a_m] \) is the least solution of the
equation \[\eta_1, ..., \eta_m = [I_u, \eta_1, ..., \eta_m, a_{m+1}, ..., a_{m+n}] \circ \beta \] and \([a_{m+1}, ..., a_{m+n}] \circ \gamma\) is the least solution of the equation \([\eta_1, ..., \eta_n = [I_u, a_1, ..., a_m, \eta_1, ..., \eta_n] \circ \gamma.\]

**Proof.** Since \(\alpha^+\) is the least solution of \(\eta = [I_u, \eta] \circ \alpha\), composing with \(\beta\) on the left, we have \(\beta \circ [I_u, \alpha^+] \circ \alpha = [\beta, \beta \circ \alpha^+] \circ \alpha = \beta \circ \alpha^+\). Therefore, \(\beta \circ \alpha^+\) is a solution of \(\eta = [\beta, \eta] \circ \alpha\). Assume \([\beta, \gamma] \circ \alpha \leq \gamma\). Since \(\bot_u \leq \gamma\), we have \([\beta, \bot_u] \leq [\beta, \gamma]\), implying \(\beta \circ [I_u, \bot_u] = [\beta, \bot_u] \leq [\beta, \gamma]\) and also \(\beta \circ [I_u, \bot_u] \circ \alpha \leq [\beta, \gamma] \circ \alpha \leq \gamma\). Therefore, we have \(\beta \circ \alpha^+ \circ \alpha \leq \gamma\). Assume inductively that \(\beta \circ \alpha^{(n)} \circ \alpha \leq \gamma\) holds. Then, we have \(\beta \circ \alpha^{(n+1)} = \beta \circ [I_u, \alpha^{(n)}] = [\beta, \beta \circ \alpha^{(n)}] \leq (by \text{ inductive hypothesis}) [\beta, \gamma], \) and we obtain \(\beta \circ \alpha^{(n+1)} \circ \alpha \leq [\beta, \gamma] \circ \alpha \leq \gamma\) (by assumption) \(\gamma\), which establishes the induction step. Therefore, \(\beta \circ \alpha^+ = \bigcup \beta \circ \alpha^{(n)} \circ \alpha \leq \gamma\) as desired. Finally, the proof of (3) is obvious using (2) and is left to the reader. \(\blacksquare\)

We now prove the lemma about the "iteration" of fixpoint solutions.

6.3.2. **Lemma** (Iteration of fixpoint solutions). *Let \(T\) be a rational algebraic theory, and let \(\alpha: VW \rightarrow VW\) be split into two parts \(\beta: VW \rightarrow v\) and \(\gamma: VW \rightarrow w,\) with \(|v| = m\) and \(|w| = n,\) \(m, n \geq 1.\) We have the identity, \(\alpha^+ = [\beta, \gamma]^+ = ((\gamma^+ \circ \beta)^+, \gamma^+ \circ (\beta^+)\circ \gamma^+).\)

**Proof.** Our main technique is to use the "least fixpoint property," that is, part (2) of Lemma 6.3.1. Let \(\alpha = [a_1, ..., a_m, a_{m+1}, ..., a_{m+n}]\) be the least solution of \(\eta = \eta \circ \alpha = \eta \circ [\beta, \gamma].\) Let \(\gamma^+ = [I_v, \gamma^+],\) with \(\gamma^+ = [c_1, ..., c_n]\) the least solution of \(\eta = [I_v, \eta] \circ \gamma.\) Finally, let \((\gamma^+ \circ \beta)^+ = [b_1, ..., b_m]\) be the least solution of \(\eta = \eta \circ \gamma^+ \circ \beta.\) We want to show the following equalities: \(a_i = b_i\) for \(1 \leq i \leq m\) and \(a_{m+j} = [b_1, ..., b_m] \circ c_j\) for \(1 \leq j \leq n.\) We first establish:

**Claim 1.** \(b_i \leq a_i\) for \(1 \leq i \leq m,\) and \(a_{m+j} \circ c_j \leq a_{m+j}\) for \(1 \leq j \leq n.\)

Since \(\alpha^+\) is the least solution of \(\eta = \eta \circ \alpha = [\eta \circ \beta, \eta \circ \gamma],\) we have \([a_{m+1}, ..., a_{m+n}] = [a_1, ..., a_m, a_{m+1}, ..., a_{m+n}] \circ \gamma.\) Therefore, \([a_{m+1}, ..., a_{m+n}]\) is a solution of the equation \(\eta = [a_1, ..., a_m, \eta] \circ \gamma.\) Since \([c_1, ..., c_n]\) is the least solution of \(\eta = [I_v, \eta] \circ \gamma,\) by Lemma 6.3.1(1), we must have \([a_1, ..., a_m] \circ c_j \leq a_{m+j}\) for \(1 \leq j \leq n,\) since \([a_1, ..., a_m] \circ [c_1, ..., c_n]\) is the least fixpoint of \(\eta = [a_1, ..., a_m, \eta] \circ \gamma.\) Then, we have the inequality, \([a_1, ..., a_m, [a_1, ..., a_m] \circ c_j, ..., [a_1, ..., a_m] \circ c_n] \circ \alpha \leq [a_1, ..., a_m],\) that is, \([a_1, ..., a_m] \circ \gamma^+ \circ \beta \leq [a_1, ..., a_m].\) Since \([b_1, ..., b_m]\) is the least fixpoint of \(\eta = \eta \circ \gamma^+ \circ \beta,\) this implies that \(b_i \leq a_i\) for \(1 \leq i \leq m,\) and Claim 1 is established. \(\blacksquare\)

**Claim 2.** \(a_i \leq b_i\) for \(1 \leq i \leq m,\) and \(a_{m+j} \leq [b_1, ..., b_m] \circ c_j\) for \(1 \leq j \leq n.\)

We know that \([b_1, ..., b_m]\) is the least solution of \(\eta = \eta \circ \gamma^+ \circ \beta,\) so we have \([b_1, ..., b_m] = [b_1, ..., b_m] \circ [1_v, c_1, ..., c_n] \circ \beta = [b_1, ..., b_m, [b_1, ..., b_m] \circ c_1, ..., [b_1, ..., b_m] \circ c_n] \circ \beta.\)
We also know that \([c_1, \ldots, c_n]\) is the least solution of \(\eta = [I_v, \eta] \circ \gamma\), so we have
\([c_1, \ldots, c_n] = [I_v, c_1, \ldots, c_n] \circ \gamma\), which implies, by composing on the left with \([b_1, \ldots, b_m]\),
\([b_1, \ldots, b_m] \circ c_1, \ldots, [b_1, \ldots, b_m] \circ c_n = [b_1, \ldots, b_m, [b_1, \ldots, b_m] \circ c_1, \ldots, [b_1, \ldots, b_m] \circ c_n] \circ \gamma\). By putting these two equations together, we obtain
\([b_1, \ldots, b_m] \circ c_1, \ldots, [b_1, \ldots, b_m] \circ c_n = [b_1, \ldots, b_m, [b_1, \ldots, b_m] \circ c_1, \ldots, [b_1, \ldots, b_m] \circ c_n] \circ \alpha\).

Since \(\alpha^+ = [a_1, \ldots, a_m, a_{m+1}, \ldots, a_{m+n}]\) is the least solution of \(\eta = \eta \circ \alpha\), we obtain immediately the desired inequalities, \(a_i \leq b_i\) for \(1 \leq i \leq m\) and
\(a_{m+j} \leq [b_1, \ldots, b_m] \circ c_j\) for \(1 \leq j \leq n\). Combining Claims 1 and 2, we obtain the desired equalities
\[
\begin{align*}
a_i &= b_i \\
a_{m+j} &= [b_1, \ldots, b_m] \circ c_j
\end{align*}
\]
for \(1 \leq i \leq m\), \(1 \leq j \leq n\).

Lemma 6.3.2 is very useful. It says that to solve for the least fixpoint of the equation \(\eta = \eta \circ [\beta, \gamma]\), we can first solve for the least solution \([c_1, \ldots, c_n]\) of the equation with parameters \(\eta = [I_v, \eta] \circ \gamma\), then substitute \(\gamma^\nu = [I_v, c_1, \ldots, c_n] \circ \beta\) to form the equation \(\eta = \eta \circ \gamma^\nu \circ \beta\), and finally solve for the least solution \([b_1, \ldots, b_m]\) of this last equation. Then, the least solution of the original equation \(\eta = \eta \circ \alpha\) is given by the equalities
\[
\begin{align*}
a_i &= b_i \\
a_{m+j} &= [b_1, \ldots, b_m] \circ c_j
\end{align*}
\]
for \(1 \leq i \leq m\), \(1 \leq j \leq n\).

The lemma even holds if \(\alpha\) has parameters, that is, if \(\alpha\) is of the form \(\alpha: uvw \rightarrow vw\).

We now give the construction announced earlier, of the free recursion-closed algebraic theory generated by a ranked alphabet \(\Sigma\). We begin with the definition of the set of trees which constitute this algebraic theory.

6.3.3. DEFINITION. Let \(S\) be a set of sorts and let \(\Sigma = \{\Sigma_{(u,s)}(u,s) \in D(S)\}\) be a ranked alphabet indexed by \(D(S)\). We define the subset \(RCT_\Sigma\) of \(CT_\Sigma\) as the set of all \(p\)-tuples of trees of the form \(\nu^\alpha\), where \(\alpha: (w_1, s_1) \cdots (w_n, s_n) \rightarrow (u, v_1) \cdots (u, v_p)\) and \(\beta: (w_1, s_1) \cdots (w_n, s_n) \rightarrow (w_1, s_1) \cdots (w_n, s_n)\) are finite recursion schemes in \(\text{FPRS}_\Sigma\). \(\beta\) is always a closed scheme, and \(\alpha\) is always a \(p\)-tuple of trees where all the trees \(a_i\) are built from the same set of individual variables \(X_u = \{x_1^u, \ldots, x_m^u\}\) (with \(u = u_1 \cdots u_m\)).

We can say that the trees in \(RCT_\Sigma\) are "context-free" (or "algebraic"). Indeed, the results of Gallier [13] can be easily adapted to show that for each tree, the set of tree addresses labeled with a given function symbol is accepted by a deterministic pushdown automaton.
The set of finite partial trees $\mathbf{FT}_\Sigma$ is a subset of $\mathbf{RCT}_\Sigma$, and this is shown by noticing that there exists a recursion scheme $\alpha$ such that $\alpha^0 = I_{\bar{u}}$, the identity recursion scheme for $\bar{u} = (w_1, s_1) \cdots (w_n, s_n)$. Recall that for any $\alpha: \bar{u} \cdot \bar{v} \rightarrow \bar{v}, \alpha^{(0)} = [I_{\bar{u}}, I_{\bar{v}}, \alpha]$ and $\alpha^{(i+1)} = \alpha^{(i)} \circ [x_1^\bar{u}, \bar{v}, \alpha]$. For $\bar{v} = \lambda$, we have the (unique) degenerate element $0_{\bar{u}}: \bar{u} \rightarrow \lambda$, and in this case, we define $[I_{\bar{u}}, 0_{\bar{v}}] = [I_{\bar{u}}, I_{\bar{v}}, \alpha]$ as $I_{\bar{u}}$, which yields $0_{\bar{u}}^\lambda = I_{\bar{u}}$ (recall that $0_{\bar{u}}$ is the only function from $\Phi_{\bar{u}}$ to the empty set). Alternatively, if we want to avoid the scheme $0_{\bar{u}}$, we can include $\mathbf{FT}_\Sigma$ in the set $\mathbf{RCT}_\Sigma$ by definition.

To prove that $\mathbf{RCT}_\Sigma$ is an algebraic theory, we have to show that it is closed under tree-composition and under tupling, since $\mathbf{RCT}_\Sigma$ contains $\mathbf{FT}_\Sigma$ and identities and projections are already in $\mathbf{FT}_\Sigma$. We also prove that $\mathbf{RCT}_\Sigma$ is recursion-closed, which makes $\mathbf{RCT}_\Sigma$ a recursion-closed algebraic $\Sigma$-theory.

6.3.4. Theorem. The set of trees $\mathbf{RCT}_\Sigma$ is a recursion-closed algebraic subtheory of $\mathbf{CT}_\Sigma$ denoted $\mathbf{RCT}_\Sigma$ with tree-composition as the composition operation.

Proof. First of all, we want to emphasize the fact that the composition operation is tree-composition (substitution only at leaves) and not scheme-substitution (which can happen inside of a tree). We prove the closure operations in the one-sorted case, to avoid complicated subscripting, the extension to the many-sorted case being an exercise in keeping the notation clear.

1) Closure under composition. Let

$$\alpha_1: (u_1, 1) \cdots (u_n, 1) \rightarrow (m_1) \cdots (m_p),$$
$$\beta_1: (u_1, 1) \cdots (u_n, 1) \rightarrow (u_1, 1) \cdots (u_n, 1),$$
$$\alpha_2: (v_1, 1) \cdots (v_r, 1) \rightarrow (p_1) \cdots (p_q),$$
$$\beta_2: (v_1, 1) \cdots (v_r, 1) \rightarrow (v_1, 1) \cdots (v_r, 1).$$

Then, $\beta_2^p \circ \alpha_1$ is a $p$-tuple of trees in $\mathbf{CT}_\Sigma(m, p)$, and $\beta_2^q \circ \alpha_2$ is a $q$-tuple of trees in $\mathbf{CT}_\Sigma(p, q)$, so they are composable.

We claim that $(\beta_2^p \circ \alpha_1) \circ (\beta_2^q \circ \alpha_2) = \gamma^\Sigma \circ \delta$, for the finite schemes $\gamma$ and $\delta$ given below:

\textit{Scheme} $\gamma$:

For all $i, 1 \leq i \leq q$,

$$F_\gamma(x_1, ..., x_m) = [H_{u_1}, ..., H_{v_r}] \cdot \left([G_1^m, ..., G_p^m] \circ (\alpha_2)_i\right).$$

For all $i, 1 \leq i \leq p$,

$$G_i^\gamma(x_1, ..., x_m) = [K_{u_1}, ..., K_{u_n}] \cdot (\alpha_1)_i.$$
For all $i$, $1 \leq i \leq n$,

$$K_u(x_1, \ldots, x_n) \equiv [K_{u_1}, \ldots, K_{u_n}] \ast (\beta_1)_i.$$ 

For all $i$, $1 \leq i \leq r$,

$$H_v(x_1, \ldots, x_n) \equiv [H_{v_1}, \ldots, H_{v_r}] \ast (\beta_2)_i.$$ 

$\delta$ is the projection scheme which picks out the first $q$ components corresponding to $F_1^q, \ldots, F_n^q$.

Using the "iteration lemma" for solving fixpoints, we see that $K_u$ computes $(\beta_1^U)_i$, $H_v$ computes $(\beta_2^V)_i$, $G_i^m$ computes $(\beta_i^V \ast (\alpha_i)_i)$ and $F_i^q$ computes $(\beta_i^V \ast (\alpha_i)_i)$, which is equal to $(\beta_i^V \ast (\alpha_i)_i) \circ (\alpha_i)_i$, because $(\beta_i^V \ast (\alpha_i)_i)$ is a constant tree. (This is easy to show by going back to the definition.)

Therefore, we have closure under composition. $\Box$

(2) Closure under tupling. The proof is analogous and is left to the reader.

Therefore, we conclude that $\text{RCT}_\Sigma$ is an ordered algebraic theory. It only remains to show that $\text{RCT}_\Sigma$ is recursion-closed.

6.3.5. **Lemma.** The ordered algebraic theory $\text{RCT}_\Sigma$ is recursion-closed.

**Proof.** Let $\Omega$ be the alphabet $\text{RCT}_\Sigma \Omega$ obtained from $\text{RCT}_\Sigma$ as explained in Definition 6.2.1, where every symbol $f$ in $\Omega_{u,s}$ is the name of a unique (possibly infinite) tree of the form $(\beta_f^U \ast (\alpha_f)_f)$ in $\text{RCT}_\Sigma(u, s)$, for some finite recursion schemes $\alpha_f$ and $\beta_f$. For simplicity, we denote the interpretation $\text{RCT}_\Sigma I$ as $J$. We shall use the observation that every tree $f$ in $\text{RCT}_\Sigma$, where $f$ is a symbol in $\Sigma$, is also represented as $f$ in $\Omega$, and so, every finite tree in $\text{FT}_\Sigma$ exists as the same tree in $\text{FT}_\Omega$ (and also as its name in $\Omega$). More precisely, if $t$ is a finite tree in $\text{FT}_\Sigma$, there exists a tree $t'$ in $\text{FT}_\Omega$ isomorphic to $t$ and formed of the symbols $f$ corresponding to the elementary trees $f$ and such that $(t'_J) = t$ under the interpretation $J$. However, we will make this identification to simplify the proof. We note in passing, that there are usually more than one tree $t'$ in $\text{FT}_\Omega$ such that $(t'_J) = t$ for a given tree $t$ in $\text{FT}_\Sigma$ corresponding to the fact that $t$ may be obtained in several ways by substitution of other trees.

Now, we have to prove that for any pair of schemes $(\alpha, \beta)$ in $\text{FPRS}_\Omega$ with

$$\alpha : (u_1, 1) \cdots (u_n, 1) \to (m, 1) \cdots (m, 1)$$

and

$$\beta : (u_1, 1) \cdots (u_n, 1) \to (u_1, 1) \cdots (u_n, 1),$$

the $\omega$-chain $((\beta^i \ast (\alpha)_i))_{i \in \omega}$ has a least upper bound in $\text{RCT}_\Sigma$. $\alpha$ and $\beta$ are finite schemes built from symbols standing for trees in $\text{RCT}_\Sigma$, and we have to show that there exist two finite schemes $\gamma$ and $\delta$ over $\Sigma$, such that $\bigcup \{(\beta^i \ast (\alpha)_i)_{i \in \omega} = \gamma^V \ast \delta$, or
equivalently, $(\beta^\gamma \ast \alpha)_j = \gamma^\gamma \ast \delta$. First of all, we can assume without loss of generality that $\alpha$ is a single tree, that is, $p = 1$. For every symbol $f \in \Omega$ occurring in $\alpha$ or in $\beta$, let $F_f$ be a new undefined function symbol having the same arity $m_f$ as $f$. Let $\alpha'$ be the tree obtained from $\alpha$ by substituting the symbol $F_f$ for every symbol $f \in \alpha$, and $\beta'$ the tree obtained from $\beta$ in the same way. Assume that every symbol $f \in \Omega$ is the name of the tree $(\beta^\gamma \ast \alpha_f)$, where $\alpha_f$ and $\beta_f$ are the schemes in $\text{FPRS}_\Sigma$ given by

$$
\alpha_f: (u_1', 1) \cdots (u_{m_f}', 1) \to (m_f, 1)
$$

and

$$
\beta_f: (u_1', 1) \cdots (u_{m_f}', 1) \to (u_1', 1) \cdots (u_{m_f}', 1).
$$

For every such $f$ we construct the following scheme $S_f$ computing the tree $(f)_j = (\beta^\gamma \ast \alpha_f)$:

**Scheme $S_f$:**

$$F_f(x_1, \ldots, x_{m_f}) \iff [G_{u_1}', \ldots, G_{u_{m_f}}'] \ast \alpha_f$$

and for all $j$, $1 \leq j \leq n_f$,

$$G_{u_j}'(x_1, \ldots, x_{u_j}) \iff [G_{u_1}', \ldots, G_{u_{m_f}}'] \ast \beta_j.$$

(If $m_f = 0$, the same equations apply, discarding the variables $x_1, \ldots, x_{m_f}$.)

The scheme $\gamma$ is the following set of equations:

**Scheme $\gamma$:**

$$F(x_1, \ldots, x_m) \iff [F_{u_1}, \ldots, F_{u_n}] \ast \alpha',$$

and for all $i$, $1 \leq i \leq n$, we have

$$F_{u_i}(x_1, \ldots, x_{u_i}) \iff [F_{u_1}, \ldots, F_{u_n}] \ast (\beta')_i,$$

and for all $f$ occurring in $\alpha$ and $\beta$, we have the union $\bigcup S_f$ of the sets of equations $S_f$.

$\delta$ is the projection scheme which picks out the first component of $\gamma$ corresponding to $F$.

By the iteration lemma, since every $F_f$ computes the tree $(\beta^\gamma \ast \alpha_f)$, $F_{u_i}$ computes the tree $((\beta^\gamma)_i)_j$, and $F$ computes the tree $(\beta^\gamma)_j \ast (\alpha)_j = (\beta^\gamma \ast \alpha)_j$, since $J$ is a homomorphism. Therefore, $(\beta^\gamma \ast \alpha)_j = \gamma^\gamma \ast \delta$, and the proof is complete.

Consequently, $\text{RCT}_\Sigma$ is a recursion-closed algebraic theory, and in fact, we now prove that it is the free recursion-closed algebraic theory generated by $\Sigma$.

**6.3.6. Theorem.** $\text{RCT}_\Sigma$ is the free recursion-closed algebraic theory generated by $\Sigma$. More precisely, for every recursion-closed algebraic theory $T$, for every interpretation $I: \Sigma \to T$, there exists a unique homomorphism of recursion-closed algebraic theories $\bar{I}: \text{RCT}_\Sigma \to T$ extending $I$, as in the diagram:
Furthermore, for all finite recursion schemes \( \alpha \) and \( \beta \) in \( \text{FPRS}_\Sigma \) with \( \alpha; (u_1, s_1) \cdots (u_n, s_n) \rightarrow (u, v, \cdots) (u, v_p) \) and \( \beta; (u_1, s_1) \cdots (u_n, s_n) \rightarrow (u, s_1) \cdots (u_n, s_n) \), \( \tilde{I} \) is given by, 
\[
\tilde{I}(\beta^N * \alpha) = \bigcup \tilde{I}(\beta^{(i)} * \alpha)_i.
\]

**Proof.** By the discussion after Definition 6.2.5, if such an \( h \) extending \( I \) is to exist, we must have \( h(\bigcup (\beta^{(i)} * \alpha)) = \bigcup h(\beta^{(i)} * \alpha) \). But \( h \) being a homomorphism extending \( I \), we must have \( h(\beta^{(i)} * \alpha) = (\beta^{(i)} * \alpha)_i \). Therefore, \( h \) is uniquely determined. It remains to show that it is well defined, that it is a homomorphism, and that it has the property of Definition 6.2.5.

1. \( \tilde{I} \) is well defined. The techniques of Theorem 10 of [21] can be used. We leave the details to the reader.

2. \( \tilde{I} \) is a homomorphism. This comes from the fact that in (1) of Theorem 6.3.4, we can show by induction that, for all \( i \geq 0 \), we have
\[
(\beta_1^{(i)} * \alpha_1) \circ (\beta_2^{(i)} * \alpha_2) = (\gamma^{(i)} * \delta).
\]

We leave the details to the reader.

3. For any pair of schemes \( (\alpha, \beta) \) in \( \text{FPRS}_\Sigma \), we have \( \tilde{I}(\bigcup (\beta^{(i)} * \alpha)_i) = \bigcup \tilde{I}(\beta^{(i)} * \alpha)_i \).

Let \( \gamma_m \) be the scheme obtained from scheme \( \gamma \) of Lemma 6.3.5, by replacing every \( \beta_i \) by \( \beta_i^{(m)} \). Using the iteration lemma, we have \( (\bigcup \beta^{(m)} * \alpha)_i = \bigcup_{m \in \omega} \gamma_m^{(n)} * \delta \). Since by Lemma 6.3.5 we have \( \bigcup (\beta^{(m)} * \alpha)_i = \gamma^N * \delta \), this implies \( \tilde{I}((\bigcup \beta^{(m)} * \alpha)_i) = \tilde{I}(\gamma^N * \delta) = \tilde{I}(\bigcup \gamma_m^{(m)} * \delta) \). But observe that for all \( m \geq 0 \), we have \( \gamma_m^{(m)} * \delta = \gamma_m^{(m)} * \delta \), and for all \( m, n \geq 0 \), letting \( N = \max(m, n) \), we have \( \gamma_m^{(m)} * \delta \leq \gamma_N^{(m)} * \delta = \gamma^{(N)} * \delta \). Therefore, we can conclude that \( \bigcup (\gamma_m^{(m)} * \delta)_i = \bigcup (\gamma_m^{(m)} * \delta)_i \). Then, we obtain \( \tilde{I}((\bigcup \beta^{(m)} * \alpha)_i) = \tilde{I}(\bigcup \gamma_m^{(m)} * \delta) = \tilde{I}((\bigcup \gamma_m^{(m)} * \delta)_i) \). Therefore, \( \tilde{I}(\bigcup (\beta^{(m)} * \alpha)_i) = \bigcup \tilde{I}(\beta^{(m)} * \alpha)_i \) as desired.

Consequently, \( \text{RCT}_\Sigma \) is the free recursion-closed algebraic theory generated by \( \Sigma \). Following the terminology of Goguen et al. [21], we can say that \( \text{RCT}_\Sigma \) consists of the "behaviors" of recursion schemes defined by pairs \( (\alpha, \beta) \), as in the definition of \( \text{RCT}_\Sigma \). Indeed, given any program \( ((\alpha, \beta), I) \), where \( I \) is an interpretation \( I: \Sigma \to T \) with \( T \) a recursion-closed algebraic theory, the unique homomorphism \( \tilde{I}: \text{RCT}_\Sigma \to T \) extending \( I \), gives a fixpoint semantics \( (\alpha, \beta)_I = \bigcup (\beta^{(i)} * \alpha)_I \) to the program \( ((\alpha, \beta), I) \).

We also have the fact that, two pairs of schemes \( (\alpha_1, \beta_1) \) and \( (\alpha_2, \beta_2) \) are equivalent in all recursion-closed interpretations, if and only if they are "tree equivalent," that is, \( (\beta_1^{(i)} * \alpha_1) = (\beta_2^{(i)} * \alpha_2) \). In other words, \( \text{RCT}_\Sigma \) is a "Herbrand" interpretation in the class of all recursion-closed interpretations.

Our study of recursion-closed algebraic theories is far from being complete, and we
feel that their properties deserve to be investigated more thoroughly. One of these properties relates to the question: Are recursion-closed algebraic theories closed under a "natural" quotient operation? We believe that this is true. This would allow the development of a theory of "presentations" of recursion-closed theories by generators and equations.

The fact that the set $\text{CPRS}_\mathcal{F}$ of finite and infinite parameterized recursion schemes is an $\omega$-continuous algebraic $D(S)$-theory under scheme-substitution has some interesting consequences. One of these applications is that we can define parameterized recursion schemes of "higher types" by taking $\text{CPRS}_\mathcal{F}$ itself as an extended interpretation. This topic also deserves further investigation.

ACKNOWLEDGMENTS

I would like to thank Professor R. V. Book for many helpful suggestions. It is also a pleasure to thank Professor J. A. Goguen for many illuminating discussions and Drs. J. W. Thatcher and E. G. Wagner for their encouragement. Finally, I thank the referee for his pertinent remarks and his most helpful suggestions.

REFERENCES

14. J. H. GALLIER, Recursion schemes and generalized interpretations, in "Automata, Languages and


