JOURNAL OF COMPUTER AND SYSTEM SCIENCES 18, 228–242 (1979)

Regular Trees and the Free Iterative Theory

SUSANNA GINALI*

Department of Computer Science, Illinois Institute of Technology, Chicago, Illinois 60616

Received January 15, 1978; revised September 22, 1978

This paper establishes the relationship between regular trees, equationally defined trees, and Elgot's iterative algebraic theories; in particular, it provides a regular tree characterization of the free iterative theory and an equational tree characterization of the free iterative theory. These tree characterizations have implications for decision problems involving flowchart schemes interpreted in iterative theories and they also help to explicate the connections between order theoretic and non-order theoretic fixed point models for iteration.

1. INTRODUCTION

Because it provides a framework in which syntax and semantics can be defined precisely, algebra has been increasingly applied to the formulation and analysis of various familiar computer science concepts and constructions. An early utilization of algebraic technique may be found in Eilenberg and Wright [4] where an elegant unified basis is given for the different manifestations of automata and formal language theory. More recently, an algebraic formulation and extension of the Chomsky hierarchy appeared in Wand [16]. The application of algebraic techniques to the study of syntax and semantics of programming languages is developed by Goguen, Thatcher, Wagner, and Wright (henceforth referred to as ADJ) in [10] and a completely algebraic (and implementable) specification language is described in Burstall and Goguen [3]. Algebra has also been used to specify and provide correctness proofs for abstract data types, most notably by Guttag [12], Zilles [18], and ADJ [11]; Goguen [8] uses algebra to describe error messages for abstract data types.

All of the applications described above share, either explicitly or implicitly, the common underlying structure of Lawvere's "algebraic theories" [13]. The structure of an algebraic theory is such as to allow the representation of a system of n equations in n variables and p parameters as an expression in the algebraic theory. Fixed point solutions to such systems have been used by Scott [14], among others, to model iteration or "looping." The basic algebraic theory definition can be enriched so as to guarantee that appropriate systems of equations in an algebraic theory can be meaningfully solved for a fixed point. One possible enrichment method is developed in ADJ [10] and involves the addition of order theoretic properties. An alternate method, due to Elgot [5], replaces order theoretic properties by the single requirement that fixed point solutions exist uniquely. An algebraic theory with such unique fixed point solutions is called an "iterative algebraic theory" or, more simply, an "iterative theory."

* Present address: Bell Laboratories, Naperville, Ill. 60540.

0022-0000/79/030228–15$02.00/0

This paper is concerned with iterative theories and, in particular, with the problem of determining the semantic equivalence of two flowschemes which are to be interpreted or implemented via the unique map to the "semantic theory" or "implementation theory" that is consistent with the interpretation or implementation of the primitive instruction symbols in the schemes. (See ADJ [11] for further details concerning this use of the term implementation.) This problem is approached through the examination of concrete constructions of the "free iterative theory" whose existence was first shown by Bloom and Elgot in [1].

Free algebraic theories and free iterative theories are of interest because they form "Herbrand universes" for the interpretation of expressions involving primitive instruction symbols, variables, parameters, and the basic theory operations. Thus two expressions have the same meaning when interpreted in every (iterative) algebraic theory if and only if they have the same meaning when interpreted in the free (iterative) algebraic theory. There is a well known construction of the free algebraic theory as the set of forests of (suitably labeled) finite trees along with the operations of "tupling" and "leaf substitution." This paper presents several concrete "tree constructions" of the free iterative theory and explores the implications of these constructions for the decidability of the question of semantic equivalence of expressions and for the problem of constructing a semantically equivalent "minimal" expression from a given expression. The constructions also explicate the relationship between Elgot's unique fixed point model and ADJ's and Wand's order theoretic models, thus answering questions posed in ADJ [10] and, moreover, they provide an analog for trees to the main theorem in Eilenberg and Wright [4], which states that in a free algebraic theory, the "algebraic" and recognizable (by generalized finite automata) sets coincide. However, unlike the theorems of Eilenberg and Wright and of others using order theoretic structures, our versions are proved without recourse to continuity arguments and rely instead on simple computational notions, such as the unwinding of a loop in a flowscheme and the construction of a minimal semantically equivalent flowscheme by "folding up" redundant loops and removing inaccessible pieces.

This flowscheme metaphor is both instructive and legitimate because an expression in an iterative theory is, in fact, a flowscheme represented in a one dimensional notation, much as a regular expression is a one dimensional representation of a finite state automaton. Various algebraic manipulations on expressions correspond exactly to natural flowscheme operations (this is developed further in Ginali [7] and Elgot [6]) and computing the meaning of an expression in an iterative theory is equivalent to interpreting the corresponding flowscheme in an iterative theory, now viewed as a semantic domain. The algebraic operations used in the construction of the free iterative theory correspond to flowscheme operations which preserve semantic equivalence.

2. Preliminaries

An *algebraic theory* (the term "algebraic" will often be dropped) may be informally defined as a collection of sets $T(n, m)$ of *morphisms* where the indices n and m are non-negative integers. The operations on these sets consist of an associative *compositon* which assigns to morphisms f in $T(n, m)$ and g in $T(m, p)$ a morphism fg in $T(n, p)$ and a *tupling* operation which assigns to morphisms f_i in $T(1, p)$ for $i = 1,..., n$, a morphism $(f_1,..., f_n)$ in $T(n, p)$. In addition, we require that the set of variables $\{x_1,..., x_p\}$ be a subset of each $T(1, p)$; that $T(n, p)$ be isomorphic to $(T(1, p))^n$ under tupling; and that for each x_i in $\{1,..., n\}$ and $f = (f_1,..., f_n)$ in $T(n, p)$, $x_i f = f_i$.

Tupling may be extended in an obvious manner to produce a morphism (f, g) in $T(n + m, p)$ from morphisms f in $T(n, p)$ and g in $T(m, p)$. A more formal treatment of algebraic theories can be found in ADJ [9], Lawvere [13], and MacLane [14].

We will write $f: n \to p$ for f in $T(n, p)$ and, following Elgot, we will call morphisms $x_i: 1 \to p$, and tuples of such morphisms, *base* morphisms. It is convenient to write x_i simply as i, to let 0_n denote the unique base morphism in $T(0, n)$, and to let 1_n denote the base morphism $(1,..., n): n \to n$. For $f: n \to p$, we will often represent $if: 1 \to p$ by f_i, calling it the *i'th component of f*.

DEFINITION 2.1. An *ideal algebraic theory* is an algebraic theory which satisfies the property that whenever $f: 1 \to n$ is not base, then neither is the composition $fg: 1 \to p$ for any $g: n \to p$. An *ideal morphism* in an ideal theory is a morphism with no base components.

A *theory morphism* $F: T \to T'$ maps each $T(n, m)$ to $T'(n, m)$ in such a manner as to preserve composition, tupling, and variables. If T and T' are ideal theories and if F preserves ideal morphisms, then F is an *ideal theory morphism.*

Roughly speaking, each component of a morphism $f: n \to n + p$ may be viewed as a term $f_i(x_1,..., x_n, x_{n+1},..., x_{n+p})$ containing n variables $x_1,..., x_n$ and p parameters $x_{n+1},..., x_{n+p}$ and the morphism f itself may be viewed as the system of n fixed point equations

$$\begin{aligned} x_1 &= f_1(x_1,..., x_n, x_{n+1},..., x_{n+p}) \\ &\vdots \\ x_n &= f_n(x_1,..., x_n, x_{n+1},..., x_{n+p}) \end{aligned}$$

Composition is really a generalized substitution operation, i.e., the composition fg, where $f: n \to m$ and $g: m \to p$, is the tuple obtained by replacing each occurrence of j in f_i by g_j for i in $\{1,..., n\}$ and j in $\{1,..., m\}$.

Thus, if $f: n \to n + p$ and $f^*: n \to p$ satisfies $f^* = f(f^*, 1_p)$ then f^* is a fixed point solution to the system of equations defined by f. An iterative theory is an ideal algebraic theory with a (vector) iteration operation which assigns to each ideal $f: n \to n + p$, the (unique) morphism $f^*: n \to p$ which satisfies $f^* = f(f^*, 1_p)$.

DEFINITION 2.2. An *iterative* (*algebraic*) *theory* is an ideal theory in which, for all ideal morphisms $f: n \to n + p$, there exists a unique morphism $f^*: n \to p$ such that $f^* = f(f^*, 1_p)$.

An apparently simpler structure is manifested by "scalar iterative theories." A *scalar iterative theory* is an algebraic theory with a (scalar) iteration operation which assigns to each nonbase $f: 1 \to 1 + p$ the unique morphism $f^*: 1 \to p$ satisfying $f^* = f(f^*, 1_p)$.

The vector iteration operation obviously requires that unique $f^*: n \to p$ exist satisfying $f^* = f(f^*, 1_p)$ for all ideal $f: n \to n + p$, while scalar iteration requires such unique f^* only for nonbase $f: 1 \to 1 + p$. Surprisingly, the existence of unique $f^*: 1 \to p$ for nonbase $f: 1 \to 1 + p$ implies the existence of unique $f^*: n \to p$ for ideal $f: n \to n + p$ as shown in Bloom, Ginali, Rutledge [2]. Therefore, we may replace all occurrences of n in Definition 2.2 by 1.

At this point, it seems appropriate to give some indication of the relationship between flowschemes and expressions interpreted in an iterative theory. The flowschemes considered here have multiple entrances and exits (as in Elgot [5]) where entrances are labeled consecutively from left to right starting with one and exits may be labeled by arbitrary positive integers. Several exits can share the same label. We call an n entrance flowscheme whose largest exit label is at most m an (n, m) flowscheme and we call a flowscheme with at most one instruction symbol a primitive flowscheme. The first top flowscheme in Fig. 2.1 is primitive with the single instruction symbol a. The set of all (finite) flowschemes can be constructed recursively by including all primitive flowschemes at the base step and then, at successive steps, by forming new flowschemes from existing ones by parallel and sequential connections and by loop application.

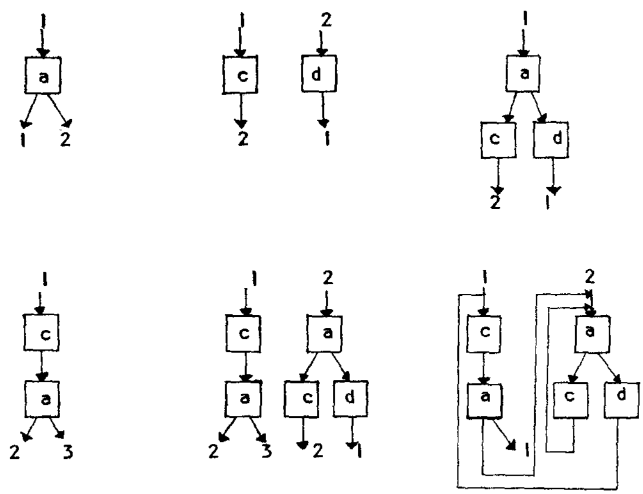

FIG. 2.1. The flowschemes below correspond, left to right, top to bottom, to the expressions $a(1, 2): 1 \to 2$, $(c(2), d(1)): 2 \to 3$, $a(c(2), d(1)): 1 \to 3$, $ca(2, 3): 1 \to 3$, $(ca(2, 3), a(c(2), d(1))): 2 \to 3$, and $(ca(2, 3), a(c(2), d(1)))^*: 2 \to 1$.

The second top flowscheme in Fig. 2.1 is the result of the parallel connection of two primitive flowschemes and the second bottom flowscheme is the result of connecting the first bottom and last top flowschemes in parallel. The last top flowscheme is, in turn, the result of the sequential connection of the first two top flowschemes. Note that an (n, m) and an (m', p) flowscheme can be connected in sequence if and only if m and m' are identical. In this case, each exit labeled by j in the first is "pasted over" entrance j of the second and the label j is removed. The last bottom flowscheme was obtained by applying loops to the second bottom flowscheme. In general, loops are applied to an $(n, n+p)$ flowscheme by pasting each exit labeled by j, for j not greater than n, over entrance j and then subtracting n from the remaining exit labels.

It was noted in the introduction that an expression in an iterative theory could be viewed as a one dimensional representation for a finite flowscheme. Such a representation for primitive flowscheme is straightforward and illustrated in Fig. 2.1. In the expressions, a indicates the theory morphism which interprets the instruction symbol a in the flowscheme and the variables interpret the correspondingly labeled exits. If f and g are expressions representing two flowschemes, then the composition fg represents the sequential connection of the flowschemes and the tupling (f, g) represents their parallel connection. Finally, if f represents a flowscheme, then the expression f^* represents the result of applying loops to that flowscheme. Thus, an expression representing a flowscheme can be obtained by induction on the construction of the flowscheme.

An algebraic theory of particular interest is the theory whose morphisms are forests of (possibly infinite) Σ-labeled trees (where Σ is a ranked set or operator domain) and whose composition is tree substitution.

DEFINITION 2.3. An *operator domain* or *ranked set* Σ is a sequence of sets Σ_0, $\Sigma_1, \ldots, \Sigma_n, \ldots$. We let Σ also denote $\bigcup_i \Sigma_i$. For σ in Σ_i, say arity$(\sigma) = i$.

DEFINITION 2.4. Let $N = \{1, 2, \ldots\}$. For i in N, say arity$(i) = 0$. A *Σ-tree in p variables* ($p = 0, 1, \ldots$) is a partial function $t\colon N^* \rightharpoonup \Sigma \cup \{1, \ldots, p\}$ whose domain of definition is denoted Dom t, such that for each w in N^* and i in N, wi in Dom t implies w in Dom t and wi in Dom t iff $i \leqslant$ arity$(t(w))$.

For t a Σ-tree in p variables, write $t\colon 1 \to p$. The set Dom t is the set of *vertices* of t and for w in Dom t, $t(w)$ is the *label* of the vertex w in t. Observe that arity$(t(w)) = 0$ if and only if w is a leaf in t. The set of vertices in t labeled by ρ in $\Sigma \cup \{1, \ldots, p\}$ is defined by

$$t^{-1}(\rho) = \{w \in N^* / t(w) = \rho\}$$

and t contains an *occurrence* of ρ iff $t^{-1}(\rho)$ is nonempty. In case Dom t is finite, t is said to be a *finite* tree. For i in $\{1, \ldots, p\}$, the tree $i\colon 1 \to p$ is the tree whose single vertex has the label i.

An *n-forest* in p variables is an n-tuple $s = (s_1, \ldots, s_n)$ of trees in p variables. Write $s\colon n \to p$. If $t\colon 1 \to n$ and $s\colon n \to p$, then composition $ts\colon 1 \to p$ is defined to be the tree t except with each leaf labeled by i in $\{1, \ldots, n\}$ replaced by the tree s_i.

DEFINITION 2.5. For $t\colon 1 \to n$, the *degree* of t, denoted $\deg t$, is the number of vertices in t not labeled by i in $\{1, \ldots, n\}$. For $s\colon n \to p$, $\deg s = \deg s_1 + \cdots + \deg s_n$.

DEFINITION 2.6. If $f\colon n \to p$ and $g\colon m \to q$ are forests in p and q variables respectively, then $f + g\colon n + m \to p + q$ is the forest in $p + q$ variables whose first n components are the n components of f and whose last m components are the m components of g except with each leaf labeled k where $k \in \{1, \ldots, q\}$ in g relabeled by $p + k$.

DEFINITION 2.7. Let Σ be an operator domain. The *free theory generated by* Σ, denoted T_Σ, is an algebraic theory along with a map u which assigns to each element σ in Σ, an ideal morphism $u(\sigma)\colon 1 \to n$ (where $n = \text{arity}(\sigma)$) in T_Σ such that for every theory T and every map v which assigns to each element σ in Σ, an ideal morphism $v(\sigma)\colon 1 \to n$ in T, there exists a unique theory morphism $G\colon T_\Sigma \to T$ such that $uG = v$. The *free iterative theory generated by* Σ, J_Σ, is similarly defined except that J_Σ replaces T_Σ and T is required to be iterative.

THEOREM 2.8. *The forests of finite Σ-trees along with tree composition form the free theory generated by Σ.*

Proof. This is well known. See for example, ADJ [9]. We will denote this theory by T_Σ.

THEOREM 2.9. *The forests of (possibly infinite) Σ-trees, along with tree composition, form an iterative theory (which we shall denote by Ct_Σ).*

Proof. It is easily checked that Ct_Σ is an algebraic theory and, given the substitution like nature of tree composition, Ct_Σ is obviously ideal. If $f\colon 1 \to 1 + p$ is a nonbase morphism, then $f^*\colon 1 \to p$ is obtained as the limit of the process of substituting f for each occurrence of 1 in f, substituting f for each occurrence of 1 in the result of the previous substitution, etc. Uniqueness of f^* obtained in this fashion is ascertained using inductive arguments on the depth of a vertex w in f^*.

Thus T_Σ is a subtheory of Ct_Σ and in Section 3 it will be demonstrated that J_Σ is also a subtheory of Ct_Σ and, in particular, that J_Σ is the subtheory determined by the "algebraic" trees in Ct_Σ. This result was first conjectured by ADJ in [10]. Section 4 will give another tree characterization of J_Σ, this time as the subtheory of Ct_Σ determined by the "regular" trees. This will be a consequence of the results of Section 3 combined with a proof that the regular and algebraic trees coincide.

3. ALGEBRAIC TREES

DEFINITION 3.1. Let $t\colon 1 \to p$ be ideal in Ct_Σ. Then t is *algebraic* iff there exists n and $d\colon n \to n + p$ ideal in T_Σ such that $t = (1_1 + 0_{n-1})\, d^*$.

Thus, let $J_\Sigma(1, p) = \{1,\ldots, p\} \cup \{t \in Ct_\Sigma(1, p) / t \text{ is algebraic}\}$ and let $J_\Sigma(n, p) = (J_\Sigma(1, p))^n$. We show that the sets $J_\Sigma(n, p)$ with tree composition form an iterative theory J_Σ, and then that J_Σ is the free iterative theory generated by Σ.

THEOREM 3.2. *The morphism sets $J_\Sigma(n, p)$ determine an iterative subtheory J_Σ of Ct_Σ.*

Proof. It suffices to prove that J_Σ is closed under composition and scalar iteration. Let

$$s = (1_1 + 0_{n-1})\, d^*\colon 1 \to p$$
$$t = (1_1 + 0_{m-1})\, e^*\colon 1 \to r$$

where $d: n \to n + p$ and $e: m \to m + r$ are ideal in T_Σ. Suppose $z: (p - 1) \to r$ is base. Because every composition of algebraic trees can be written as a sequence of compositions of morphisms whose first components are either base or algebraic and whose other components are base, to verify closure under composition, it suffices to show that expressions of the form $s(t, z)$ denote algebraic trees where s, t, and z are as above. But observe that

$$s(t, z) = (1_1 + 0_{n+m-1})\, h^*: 1 \to r$$

where

$$h = (d(1_{n+1} + 0_{m-1} + z), e(0_n + 1_{m+r})): n + m \to n + m + r$$

is ideal in T_Σ.

Consider $s = (1_1 + 0_{n-1})\, d^*: 1 \to 1 + p$ where $d: n \to n + 1 + p$ is ideal in T_Σ. To verify closure under (scalar) iteration, it suffices to show that $s^*: 1 \to p$ is algebraic. But observe that

$$s^* = (1_1 + 0_{n-1})\, h^*: 1 \to p$$

where

$$h = d(1_n + 0_p\,, 1_1 + 0_{n+p-1}\,, 0_n + 1_p): n \to n + p$$

is ideal in T_Σ.

The essence of the proof that J_Σ is the free iterative theory generated by Σ is the verification that for any iterative theory J and any ideal theory morphism $F: T_\Sigma \to J$, $G: J_\Sigma \to J$ defined for $i: 1 \to p$ base by

$$G(i) = i$$

and for $d: n \to n + p$ ideal in T_Σ by

$$G((1_1 + 0_{n-1})\, d^*) = (1_1 + 0_{n-1})\, Fd^*$$

determines a well defined theory morphism. We will prove first that G is well defined, based on the verification that for ideal $d, e: n \to n + p$ in T_Σ, $d^* = e^*$ in J_Σ implies that $Fd^* = Fe^*$ in J.

Let $t: n \to p$ be ideal in J_Σ. Define

$$R(t) = \{d: n \to n + p \text{ ideal in } T_\Sigma / d^* = t\}.$$

We show that if all components of t are distinct and if $R(t)$ is nonempty, then it contains a unique element c of minimal degree. We then inductively define "partial unwindings" of c, show that every element of $R(t)$ is a partial unwinding of c, and finally prove that if d is a partial unwinding of c, then $(Fc)^* = (Fd)^*$. (Note that theory morphism application takes precedence over the iteration operation and therefore parentheses in expressions such as $(Fc)^*$ can and often will be omitted.) Ultimately, it is shown that even if the components of t are not distinct, if d and e are in $R(t)$, then $Fd^* = Fe^*$ in J. We begin with the construction of the minimal element c in $R(t)$.

LEMMA 3.3. *Let $t\colon n \to p$ in J_Σ be such that all components of t are distinct and $R(t)$ is nonempty. Then $R(t)$ contains a unique element $c\colon n \to n + p$ of minimal degree and, moreover, for all d in $R(t)$ and ρ in $\Sigma \cup \{n + 1,\ldots, n + p\}$*

$$c_i^{-1}(\rho) \subseteq d_i^{-1}(\rho) \qquad \text{for} \quad i = 1,\ldots, n$$

Proof. The tree c_i for $i = 1,\ldots, n$ is defined by taking as the set of vertices of c_i

$$\operatorname{Dom} c_i = \bigcap_{d \in R(t)} \operatorname{Dom} d_i$$

A simple inductive argument based on the fact that $\operatorname{Dom} d_i \subseteq \operatorname{Dom} t_i$ for all d in $R(t)$ shows that if $wj \in \operatorname{Dom} c_i$, then $w \in \operatorname{Dom} c_i$ and for $w \in \operatorname{Dom} c_i$ and $d, d' \in R(t)$, either $d_i(w) = d_i'(w)$ or $d_i(w) = k \in \{1,\ldots, n\}$ and $d_i'(w) = t_k(\lambda)$. Thus c_i is well defined if for ρ in $\Sigma \cup \{n + 1,\ldots, n + p\}$

$$c_i^{-1}(\rho) = \bigcap_{d \in R(t)} d_i^{-1}(\rho)$$

and for k in $\{1,\ldots, n\}$

$$c_i^{-1}(k) = \Big\{w \in \bigcup_{d \in R(t)} d_i^{-1}(k) \text{ and } w \in \operatorname{Dom} c_i\Big\}^1$$

That c is in $R(t)$ is immediate as is the uniqueness and minimality of c.

DEFINITION 3.4. Let $c\colon n \to n + p$ be ideal in T_Σ. A *partial unwinding* of c in T_Σ is recursively defined by

(i) c is a partial unwinding of c.

(ii) Suppose that d is c except that there are possibly some vertices w in d such that the subtree of d at w is the k'th component of a partial unwinding of c and the label of c at w is k. Then d is a partial unwinding of c.

The term "partial unwinding" has not been chosen arbitrarily here. If d is a partial unwinding of c, then the flowscheme corresponding to the expression d^* is obtained from the flowscheme corresponding to the expression c^* by unwinding some loops in that flowscheme a finite number of times. Unwinding all loops an infinite number of times yields the infinite tree c^*. (See Ginali [7]).

LEMMA 3.5. *Let $c\colon n \to n + p$ and $R(t)$ be as in the statement of Lemma 3.3. If $d\colon n \to n + p$ is in $R(t)$, then d is a partial unwinding of c.*

Proof. By induction on $\deg d_i$, $i = 1,..., n$. Base step: $\deg d_i = \deg c_i$ implies $d_i = c_i$. Inductive step: Suppose w is a leaf in c_i labeled by k in $\{1,..., n\}$. Then, either $d_i(w) = k$ or the subtree of d_i at w is the k'th component of a morphism e in $R(t)$ whose other components can be assumed to be the corresponding components of c. By the inductive hypothesis, e is a partial unwinding of c.

LEMMA 3.6. *Let J be an iterative theory and let $F: T_\Sigma \to J$ be an ideal theory morphism. Let $c: n \to n + p$ be ideal in T_Σ and suppose that $d: n \to n + p$ is a partial unwinding of c. Then $Fc^* = Fd^*$ in J.*

Proof. We show by induction on $\deg d_i$ that

$$Fd_i(Fc^*, 1_p) = Fc_i^*$$

Base step: Trivial because $\deg d_i = \deg c_i$ implies $d_i = c_i$. Inductive step: Suppose $w_1,..., w_m$ are the leaves in c_i such that w_j is labeled by b_j in $\{1,..., n\}$ and the subtree of d_i at w_j is the b_jth component of a partial unwinding of c. Let $g: 1 \to m + n + p$ be c_i except with leaf w_j labeled by j and any other leaves labeled by k in $\{1,..., n + p\}$ labeled instead by $m + k$. Let e_j denote the subtree of d_i at w_j. Then $e = (e_1,..., e_m)$: $m \to n + p$ is ideal and for b_j as above, $b = (b_1,..., b_m): m \to n$ is base. Moreover

$$g(b + 0_p, 1_{n+p}) = c_i \tag{3.1}$$

$$g(e, 1_{n+p}) = d_i \tag{3.2}$$

By the inductive hypothesis,

$$Fe(Fc^*, 1_p) = bFc^*$$

By (3.1) and (3.2),

$$Fd_i(Fc^*, 1_p) = Fc_i^*$$

Lemmas 3.3, 3.5, 3.6 prove:

PROPOSITION 3.7. *Suppose $d, e: n \to n + p$ are ideal in T_Σ such that $d^* = e^*$ in J_Σ and $d^* = e^*$ has distinct components. Then if J is an iterative theory and $F: T_\Sigma \to J$ is an ideal theory morphism, $Fd^* = Fe^*$ in J.*

The proposition holds even if the components of $d^* = e^*$ are not distinct.

PROPOSITION 3.8. *Let $d, e: m \to m + p$ be ideal in T_Σ such that $d^* = e^*$ in J_Σ. Then if J is an iterative theory and $F: T_\Sigma \to J$ is an ideal theory morphism, $Fd^* = Fe^*$ in J.*

Proof. Suppose that $d^* = e^*$ has n distinct components $t_1,..., t_n$ and let $z_i = \min\{k/d_k^* = t_i\}$. $z = (z_1,..., z_n): n \to m$ defines a base morphism and $t = zd^*$. It is easily verified that $R(t)$ is nonempty and thus let $c: n \to n + p$ be the minimal element in $R(t)$ as constructed in Lemma 3.3. Let $b = (b_1, ..., b_m): m \to n$ be such that d_j^* is the b_jth component of t for $j = 1,..., m$. Now pick j in $\{1,..., m\}$ and define $a = (a_1, ..., a_n): n \to m$ such that $a_k = j$ if $b_j = k$ and $a_k = z_k$ otherwise. Then

$$b_j a = j: 1 \to m \tag{3.3}$$

$$(ad(b + 0_p, 0_n + 1_p))^* = c^*: n \to p \tag{3.4}$$

By (3.4) and Proposition 3.7,

$$aFd(bFc^*, 1_p) = Fc^*$$

Composing each side of the left with b_j, by (3.3),

$$jFd(bFc^*, 1_p) = b_jFc^*$$

Thus $Fd^* = bFc^*$ and obviously also $Fe^* = bFc^*$, so $Fd^* = Fe^*$.

Using Proposition 3.8, we show that if $d: n \to n + p$ and $e: m \to m + p$ are ideal in T_Σ and $(1_1 + 0_{n-1}) d^* = (1_1 + 0_{m-1}) e^*$, then $(1_1 + 0_{n-1}) Fd^* = (1_1 + 0_{m-1}) Fe^*$ in J for J any iterative theory and $F: T_\Sigma \to J$ any ideal theory morphism. The next two lemmas demonstrate that it suffices to prove this for d, e where each component is "reachable" from the first and where the root of each component (except possibly the first) is "the head of a loop" in the "computation" of d^* and e^* in J_Σ.

DEFINITION 3.9. Let $f: n \to n + p$ be ideal in T_Σ. The j'th component of f is said to be *reachable* from the i'th component of f iff there exists $m \geqslant 1$ such that the i'th component of $f^m = f(f^{m-1}, 0_n + 1_p)$ (where $f^0 = 1_n + 0_p$) contains an occurrence of j.

Note that reachability is transitive but neither reflexive nor symmetric. A component j is reachable from a component i iff in the flowscheme corresponding to the expression f^*, there is a path from the entrance of f_i to the entrance of f_j.

LEMMA 3.10. *Let $f: n \to n + p$ be ideal in T_Σ. Then there exists $m \leqslant n$ and ideal $g: m \to m + p$ (not necessarily unique) such that each component of g (except possibly the first) is reachable from the first component of g and, moreover, if J is an iterative theory and $F: T_\Sigma \to J$ is an ideal theory morphism, then $(1_1 + 0_{n-1}) Ff^* = (1_1 + 0_{m-1}) Fg^*$.*

Proof. Let $z_1, ..., z_m$ be an enumeration of the distinct indices of f such that $z_1 = 1$ and $z_2, ..., z_m$ are the indices of components reachable from the first component of f. $z = (z_1, ..., z_m): m \to n$ defines a base morphism. If j in $\{1,..., n\}$ occurs in $zf: m \to n + p$, then $j = z_k$ for some k in $\{1,..., m\}$. Let g be the result of replacing z_k in zf by k and replacing $n + k$ in zf by $m + k$ for k in $\{1,..., p\}$. Then

$$zf = g(z + 1_p)$$

and thus by a simple computation

$$zFf^* = Fg^*$$

Because $z_1 = 1$,

$$(1_1 + 0_{n-1})Ff^* = (1_1 + 0_{m-1})Fg^*$$

Obviously, $m \leqslant n$.

LEMMA 3.11. *Let $f\colon n \to n + p$ be ideal in T_Σ such that some component i of f does not contain an occurrence of its own index i. Then there exists $m < n$ and ideal $g\colon m \to m + p$ (not necessarily unique) in T_Σ such that if J is an iterative theory and $F\colon T_\Sigma \to J$ is an ideal theory morphism, then $(1_1 + 0_{n-1})Ff^* = (1_1 + 0_{m-1})Fg^*$.*

Proof. Suppose that the i'th component of f does not contain an occurrence of i ($i \neq 1$). Let $h\colon n \to n + p$ be f with each occurrence of i replaced by f_i. Then $f^* = h^*$ in J_Σ and since h contains no occurrences of i, the i'th component of h is not reachable from the first component of h. Apply Lemma 3.10 to h to obtain m and g and observe that $m < n$.

Note that f_i contains an occurrence of its own index iff in the flowscheme corresponding to f^*, there is a loop involving only boxes in f_i.

LEMMA 3.12. *Let $f\colon n \to n + p$ be ideal in T_Σ and suppose that each component of f (except possibly the first) is reachable from the first and contains an occurrence of its own index. Then there exists ideal $h\colon n \to n + p$ in T_Σ such that the first component of h contains an occurrence of each i in $\{2,..., n\}$ and $f^* = h^*$ in J_Σ.*

Proof. For i in $\{2,..., n\}$, there exists m_i such that the first component of f^{m_i} contains an occurrence of i. If $k \geqslant m_i$, then f^k contains an occurrence of i in the first component. Take h to be f^m where $m = \max\{m_2, ..., m_n\}$.

THEOREM 3.13. *J_Σ is the free iterative theory generated by Σ.*

Proof. It suffices to show that if $I\colon T_\Sigma \to J_\Sigma$ is the inclusion theory morphism, then for every iterative theory J and every ideal theory morphism $F\colon T_\Sigma \to J$, there exists a unique ideal theory morphism $G\colon J_\Sigma \to J$ such that $F = IG$. Define G for i in $\{1,..., p\}$ by

$$G(i) = i$$

for $t = (1_1 + 0_{n-1})\, d^*$ by

$$G(t) = (1_1 + 0_{n-1})Fd^*$$

and for $t = (t_1, ..., t_n)$ by

$$G(t) = (G(t_1), ..., G(t_n)).$$

We must show that for $d\colon n \to n + p$ and $e\colon m \to m + p$ ideal in T_Σ, if $(1_1 + 0_{n-1})\, d^* =$

$(1_1 + 0_{m-1})\, e^*$ in J_Σ, then $(1_1 + 0_{n-1})\, Fd^* = (1_1 + 0_{m-1})\, Fe^*$ in J. By Proposition 3.8 and Lemmas 3.10, 3.11, and especially 3.12, it suffices to prove this for d and e satisfying the conditions on h in Lemma 3.12. Thus pick $w_2, \dots, w_n$ in Dom d_1 such that the label of d_1 at w_i is i. Also pick k greater than $\max\{\text{length}(w)/w \in \text{Dom } d_1\}$. Then the subtree r_i at w_i in $e_1{}^k$ is ideal and $r = (e_1, r_2, \dots, r_n)\colon n \to m + p$ is ideal in T_Σ. In a similar manner we can pick $s = (d_1, s_2, \dots, s_m)\colon m \to n + p$ ideal in T_Σ and

$$(r(s, 0_n + 1_p))^* = d^*$$
$$(s(r, 0_m + 1_p))^* = e^*$$

By Proposition 3.8

$$(F(r(s, 0_{n+p})))^* = Fd^*$$
$$(F(s(r, 0_m + 1_p)))^* = Fe^*$$

Thus $Fs(Fd^*, 1_p) = Fe^*$ and, because $s_1 = d_1$, $(1_1 + 0_{n-1})\, Fd^* = (1_1 + 0_{m-1})\, Fe^*$. Therefore, G is well defined.

The proof that G preserves composition is similar to the proof that J_Σ is closed under composition. G is clearly ideal. To see that $F = IG$, suppose $f\colon 1 \to p$ in T_Σ. Then $f = (f(0_1 + 1_p))^*$ and

$$\begin{aligned} IG(f) &= G((f(0_1 + 1_p))^*) \\ &= (F(f(0_1 + 1_p)))^* \\ &= Ff \end{aligned}$$

G is unique because ideal theory morphisms preserve iteration and because the requirement that $F = IG$ forces the definition $G((1_1 + 0_{n-1})\, d^*) = (1_1 + 0_{n-1})\, Fd^*$.

4. Regular Trees

We now show that the algebraic trees and the ideal regular trees coincide. This coincidence, along with the results of Section 3, implies that our definition of semantic equivalence, i.e., two expressions are semantically equivalent iff they have the same interpretation in every iterative theory, is decidable. This is because it is decidable whether two expressions are semantically equivalent iff they represent the same regular tree. The characterization of the free iterative theory as the forests of algebraic trees and its characterization as the forests of regular trees provides an analog to the main theorem in Eilenberg and Wright [4] and also explicates the relationship between iterative theories and μ-clones (Wand [15]). Wand showed that the free μ-clone generated by Σ consists of the partially defined regular Σ trees (partially defined means that not all leaves have labels). Thus, the free iterative theory generated by Σ is properly contained in the free μ-clone generated by Σ.

DEFINITION 4.1. Let $t: 1 \to p$ be in Ct_Σ. Then t is *regular* iff

(i) $t^{-1}(\sigma)$ is empty for all but a finite number of $\sigma \in \Sigma$.

(ii) For each $\rho \in \Sigma \cup \{1,..., p\}$, $t^{-1}(\rho)$ is a regular set (i.e., a set recognized by a conventional finite state automaton).

PROPOSITION 4.2. *If $t: 1 \to p$ is algebraic, then it is regular.*

Proof. Let $t = (1_1 + 0_{n-1})\, d^*$. The only elements of Σ which appear as labels in t are those which appear in d and since d is finite, only a finite number of elements of Σ appear as labels in t. For $\sigma \in \Sigma$ or $k \in \{1,..., p\}$, we can convert $d: n \to n + p$ into a finite state automaton recognizing $t^{-1}(\sigma)$ or $t^{-1}(k)$. The states of the automaton are the vertices in d not labeled by elements of $\{1,..., n\}$, and the arrows are the (directed) edges of d where the arrows from a vertex labeled by σ are labeled left to right from 1 to arity(σ). Any edge to a leaf labeled $i \in \{1,..., n\}$ in d is, in the automaton, redirected to the state corresponding to the root of d_i. The initial state is the state corresponding to the root of d_1 and the final states are the states corresponding to the vertices labeled by σ if the automaton is to recognize $t^{-1}(\sigma)$ or the states corresponding to vertices labeled by $n + k$ if the automaton is to recognize $t^{-1}(k)$.

PROPOSITION 4.3. *If $t: 1 \to p$ is an ideal regular tree, then it is algebraic.*

Proof. Let t be an ideal regular tree and let $m = \max\{\text{arity}(\sigma)/t^{-1}(\sigma) \text{ is nonempty}\}$. Define $\equiv$ on $\{1,..., m\}^*$ by $w \equiv w'$ iff

(i) $w, w' \in \text{Dom}\, t$ or

(ii) the subtree of t at w is the subtree of t at w'.

It is easily shown that $\equiv$ is a finite right invariant congruence on $\{1,..., m\}^*$. Let $w_1 ,..., w_n$ be an enumeration of the roots of the distinct ideal subtrees of t with $w_1 = \lambda$. We construct ideal $d: n \to n + p$ such that $t = (1_1 + 0_{n-1})\, d^*$ in J_Σ where each d_i has degree 1. The label of the root of d_i is the label of w_i in t. For j in $\{1,..., \text{arity}(t(w_i))\}$, the j'th leaf of d_i is labeled by

$$\begin{array}{lll} \text{(i)} & k & \text{if } w_i j \equiv w_k \quad \text{or} \\ \text{(ii)} & n+k & \text{if } t(w_i j) = k \in \{1,..., p\}. \end{array}$$

An inductive argument on the length of w in $\{1,..., m\}^*$ shows that $(1_1 + 0_{n-1})\, d^* = t$.

COROLLARY 4.4. *The algebraic and the ideal regular trees coincide.*

COROLLARY 4.5. *The free iterative theory generated by Σ consists of the forests of regular Σ-labeled trees.*

Proof. Apply Corollary 4.4 and observe that the nonideal regular trees are simply the base trees.

5. Conclusion

Because two expressions have the same meaning when interpreted in every iterative theory iff they have the same meaning when interpreted in the free iterative theory; and, because the morphisms in the free iterative theory are regular trees, semantic equivalence is decidable. This particular notion of semantic equivalence is a very strong one indeed and it should be observed that its decidability does not imply that it is decidable whether two expressions have the same meaning when interpreted in some specific iterative theory.

Any iterative theory has a "presentation" in terms of the free iterative theory J_Σ and a set of identities (involving trees in J_Σ) which determine a congruence on the morphism sets of J_Σ. That is, any iterative theory is isomorphic to a quotient theory $J_\Sigma/\equiv$ where $\equiv$ is the smallest theory congruence on J_Σ consistent with the identities. A denotational semantics for an expression to be interpreted in $J_\Sigma/\equiv$ may be defined as the image of the corresponding tree in J_Σ under the quotient theory morphism $J_\Sigma \rightarrow J_\Sigma/\equiv$.

Any identity or set of identities may determine a theory congruence and if the congruence determined by an identity has the property that it is decidable whether or not two trees fall into the same one of its congruence classes, we will (informally) call the identity decidable. Two expressions are equivalent in an iterative theory $J_\Sigma/\equiv$ iff the trees in J_Σ corresponding to the expressions are in the same congruence class of $\equiv$. The proportion of undecidable to decidable identities in the identity set determining $\equiv$ might be viewed as a measure of the "extent of undecidability" in $J_\Sigma/\equiv$.

According to the results in this paper, the equivalence of two expressions in $J_\Sigma/\equiv$ can be verified in the following "interactive" manner: An algorithmic preprocessing step is performed to determine whether the expressions are, in fact, equivalent in every iterative theory. If so, there is no need to proceed and, if not, equivalence must be somehow hand verified. The power of the algorithmic step can be greatly enhanced by the isolation of any decidable identities which contribute to $\equiv$ because this step may then be expanded to include tests to determine whether the trees corresponding to the expressions are in the same class of the congruence determined by the decidable identities.

Any general characterization of decidable identities (in terms of form, distribution of variables, etc.) would provide a valuable tool for the construction of interactive proofs of equivalence or correctness of flowschemes and should also provide insight into the nature and severity of undecidability within a particular semantic domain or as related to a particular definition of semantic equivalence.

Acknowledgment

The author wishes to thank J. A. Goguen for his excellent suggestions and constructive criticisms during the preparation of this manuscript.

References

1. S. L. Bloom and C. C. Elgot, The existence and construction of free iterative theories, *J. Comput. System Sci.* **12** (1976), 305–318.
2. S. L. Bloom, S. Ginali, and J. D. Rutledge, Scalar and vector iteration, *J. Comput. System Sci.* **14** (1977), 251–256.
3. R. Burstall and J. A. Goguen, Putting theories together to make specifications, *in* "Proc. of the Int. Joint Conf. on Artificial Intelligence, 1977."
4. S. Eilenberg and J. B. Wright, Automata in general algebras, *Inform. Contr.* **11** (1967), 452–470.
5. C. C. Elgot, Monadic computation and iterative algebraic theories, "Logic Colloquium, 1973," H. E. Rose and J. C. Shepherdson, Eds., Vol. 80, Studies in Logic, North-Holland, Amsterdam, 1975.
6. C. C. Elgot, Structured programming with and without GO TO statements, *IEEE Trans. Softw. Eng.* **SE-2**, No. 1 (1976).
7. S. Ginali, "Iterative Algebraic Theories, Infinite Trees, and Program Schemata," Ph.D. dissertation, University of Chicago, June, 1976.
8. J. A. Goguen, "Abstract Errors and Abstract Data Types," Semantics and Theory of Computation Report No. 6, UCLA Computer Science Department, February, 1977.
9. J. A. Goguen, J. Thatcher, E. Wagner, and J. Wright, "A Junction between Computer Science and Category Theory; I. Basic Definitions and Examples," Part 2, IBM Research Report RC 5908, March, 1976.
10. J. A. Goguen, J. Thatcher, E. Wagner, J. Wright, Initial algebra semantics and continuous algebras, *J. Assoc. Comput. Mach.* **24** (1977), 68–95.
11. J. A. Coguen, J. Thatcher, E. Wagner, and J. Wright, "An Initial Algebra Approach to the Specification, Correctness, and Implementation of Abstract Data Types," IBM Research Report RC 6487, 1977.
12. J. V. Guttag, "The Specification and Application to Programming of Abstract Data Types," Computer Systems Research Technical Report CSEG-59, University of Toronto, 1975.
13. F. W. Lawvere, Functional semantics of algebraic theories, *Proc. Nat. Acad. Sci. U.S.A.* **50** (1963), 869–872.
14. S. MacLane, "Categories for the Working Mathematician," Graduate Texts in Mathematics, Springer-Verlag, New York/Heidelberg/Berlin, 1971.
15. D. Scott, Lattice of flow diagrams, *in* "Symposium on Semantics of Algorithmic Languages," Vol. 188, Lecture Notes in Mathematics, Springer-Verlag, New York/Heidelberg/Berlin, 1971.
16. M. Wand, "Mathematical Foundations of Language Theory," Ph. D. dissertation, Project MAC, M.I.T., Cambridge, Mass., 1973.
17. M. Wand, "Algebraic Theories and Three Rewriting Systems," Technical Report No. 66, Indiana University Computer Science Department, 1977.
18. S. H. Zilles, "Algebraic Specification of Data Types," Project MAC Progress Report 11, M.I.T., Cambridge, Mass., 1974.