Generalized Finite Automata Theory with an Application to a Decision Problem of Second-Order Logic

by

J. W. THATCHER

and

J. B. WRIGHT

IBM Watson Research Center

ABSTRACT

Many of the important concepts and results of conventional finite automata theory are developed for a generalization in which finite algebras take the place of finite automata. The standard closure theorems are proved for the class of sets "recognizable" by finite algebras, and a generalization of Kleene's regularity theory is presented. The theorems of the generalized theory are then applied to obtain a positive solution to a decision problem of second-order logic.

1. Introduction

Let us presuppose that the reader understands what we mean by "generalized finite automaton". With this presupposition, the results presented here are easily summarized (cf. Thatcher and Wright [19]) by saying that conventional finite automata theory goes through for the generalization—and it goes through quite neatly! Our presupposition is also simple to dispose of: A generalized finite automaton is a finite abstract algebra in the sense of Birkhoff [1]; it is a finite set A together with operations $f_1, \cdots, f_n$ on A. (The algebras considered here are not quite as general as those considered by Birkhoff [1]. We allow only a finite number of operations (the infinite case will be considered in a later paper) and each operation is a function from A^m into A for some m. Birkhoff required only that some subset of $\bigcup_i A^i$ be specified as the domain of each operation, a generalization which does not appear in his Foreword on Algebra in [2].)

The most commonly used definition (cf. Rabin and Scott [14]) describes a finite automaton with input alphabet Σ as a quadruple $\mathscr{A} = \langle A, M, a_0, A_F \rangle$, where A is a finite set, the set of *states*; M is a function from $\Sigma \times A$ into A, the *direct transition function*; $a_0 \in A$ is the *initial state* and $A_F \subseteq A$ is the set of *final states*. The direct transition function is then extended to a function

MATHEMATICAL SYSTEMS THEORY, Vol. 2, No. 1
Published by Springer-Verlag New York Inc.

from Σ^* (the set of finite strings on Σ including the empty string Λ) into A by $\overline{M}(\Lambda) = a_0$ and $\overline{M}(\sigma x) = M(\sigma, \overline{M}(x))$ for all $\sigma \in \Sigma$.[1] The *behavior* of $\mathscr{A}$, or set of strings accepted by $\mathscr{A}$, is $\{x \mid \overline{M}(x) \in A_F\}$.

The structure of an automaton $\mathscr{A}$ resembles an algebra[2] if we ignore the final states—except that the transition function is on $\Sigma \times A$ and Σ is not part of the domain of the structure. Since Σ is finite, we can obtain an algebra by considering, instead of the dyadic function M, a monadic function α_σ: $A \to A$ for each $\sigma \in \Sigma$, where $\alpha_\sigma(a) = M(\sigma, a)$.[3] Thus the structure of the automaton $\mathscr{A}$ is described as a monadic algebra, $\mathscr{A} = \langle A, \alpha_{\sigma_1}, \cdots, \alpha_{\sigma_n}, a_0 \rangle$ $(\Sigma = \{\sigma_1, \cdots, \sigma_n\})$. Then, in effect, the function $\overline{M}$ is defined the same way; $\overline{M}(\Lambda) = a_0$, $\overline{M}(\sigma x) = \alpha_\sigma(\overline{M}(x))$ for all $\sigma \in \Sigma$ and $x \in \Sigma^*$. Having left out the final states from the description of the automaton, the behavior becomes a function of the choice of final states. With appropriate notation, we write $bh_{\mathscr{A}}(A_F) = \{x \mid M(x) \in A_F\}$.

Turning now to the way the generalization arises, we let Σ be an alphabet of two symbols, $\Sigma = \{g, f\}$. Any finite automaton with input alphabet Σ is an algebra with two monadic functions and a constant (the initial state). If we view $f, g, \in \Sigma$ as monadic function symbols (syntactic objects), then we can define a set T_Σ of constant terms (using Λ as an individual symbol) as the smallest class containing the symbol Λ with the property that if $t \in T_\Sigma$ then ft and gt are both in T_Σ. Thus the constant terms are strings on the alphabet $\{f, g, \Lambda\}$ including, for example, the objects Λ, $fggf\Lambda$, $fff\Lambda$, $g\Lambda$, etc. Obviously, the set Σ^* of strings can be put in one-one correspondence with this set T_Σ of terms; Λ (the empty string) corresponds to Λ (the individual symbol) and for $w \in \Sigma^* - \{\Lambda\}$, w corresponds to the term $w\Lambda$.

So under this correspondence, we can consider inputs to an automaton to be constant terms of a term language which fits the algebraic similarity type of the automaton. And from an algebraic point of view, this set T_Σ of constant terms can be taken to be the domain of a totally free or generic algebra, $\mathscr{F}_\Sigma = \langle T_\Sigma, \overline{f}, \overline{g}, \Lambda \rangle$ where $\overline{f}(t) = ft$ and $\overline{g}(t) = gt$. Now given any automaton $\mathscr{A} = \langle A, \alpha_f, \alpha_g, a_0 \rangle$ there exists a unique homomorphism $h_{\mathscr{A}}$ of $\mathscr{F}_\Sigma$ into $\mathscr{A}$ defined by $h_{\mathscr{A}}(\Lambda) = a_0$ and $h_{\mathscr{A}}(ft) = \alpha_f(h_{\mathscr{A}}(t))$. It is, of course, easy to check that $h_{\mathscr{A}}(t)$ is precisely $\overline{M}(t)$, so that the behavior of an automaton $\mathscr{A}$

[1] To make the generalization fit more neatly, we use $\overline{M}(\sigma x)$ instead of $\overline{M}(x\sigma)$; this corresponds to an automaton working on its input tape from right to left.

[2] The resemblance to some is not clear as we find proofs in the recent literature that every automaton possesses a group of automorphisms; compare Birkhoff [1], where we find the statement: "It would be pointless to prove in detail what is already known, that every algebra has a group".

[3] This formulation for finite automata was presented in a seminar conducted at the University of Michigan by J. R. Büchi and J. B. Wright in the Fall of 1960 (cf. Büchi and Wright [5]; Büchi [6]; and Thatcher [17]). It is clear that credit is due Büchi for the algebraic interpretation of automata and the use of algebraic concepts in attacking the basic problems of the theory. The possibility of the generalization to arbitrary algebras was implicit in the work of Büchi and Wright. In some sense, this paper is an exposition which makes those ideas explicit. As an historical comment, it is interesting to note that the first formulation of finite automata as monadic algebras seems to be that of Medvedev [12]. And an aside, those referencing their work on semigroups and automata should take note of this paper as well.

with respect to a set A_F of final states is simply $h_{\mathscr{A}}^{-1}(A_F) = \{t \mid h_{\mathscr{A}}(t) \in A_F\}$.

And this is the way the generalization arises; conventional finite automata theory deals with finite *monadic* algebras; our generalization deals with finite algebras. Thus under this generalization—and paralleling the discussion above—an automaton is a finite algebra, the input symbols for the automaton are function symbols of the appropriate rank and the "input strings" are constant terms of the corresponding term language. In Section 2 we introduce the formal definitions of the concepts involved and proceed to prove the standard closure theorems for the class of sets that can be recognized by finite automata.[4]

An alternative characterization of the class of recognizable sets in the conventional theory is obtained through Kleene's concept of regular set (cf. Kleene [11] and Copi, Elgot and Wright [7]). The class of regular sets on the alphabet $\Sigma = \{f, g\}$ is the least class of subsets containing the finite sets and closed under $\cup$, $\cdot$ and $*$, where

$$\alpha \cdot \beta = \{t_1 {}^\frown t_2 \mid t_1 \in \alpha \text{ and } t_2 \in \beta\}$$

and where α^* is the smallest subset β of Σ^* containing α with the property that $\beta \cdot \beta = \beta$. These regular operators—complex product and closure—on subsets of Σ^* depend explicitly on the operation of concatenation $(t_1 {}^\frown t_2)$ on Σ^*. An attempt to generalize regularity theory as we have generalized the theory of recognizability forces us to look more closely at the nature of concatenation when the objects of Σ^* are viewed as terms, for example,

$$fgf\Lambda {}^\frown g\Lambda = fgfg\Lambda,$$

$$[fgf(x) {}^\frown g(x) = fgfg(x)].$$

This illustration may suggest to the reader—and indeed the suggestion proves fruitful—that concatenation of strings corresponds to replacement (or substitution in this case) for terms. In Section 3 we develop a theory of regularity based on a replacement relation and the important theorem "regular if and only if recognizable" is proved there.

The last section is an application of the generalized theory of finite automata. There we prove that there is a decision procedure for the weak monadic second-order theory of multiple successors.

2. The Basis for a General Theory of Recognizability

A *species* is defined to be an ordered pair $\mathscr{S} = \langle \Sigma, \sigma \rangle$, where Σ is the set of *function symbols* of $\mathscr{S}$ and σ is a map from Σ into N (the non-negative integers). For $f \in \Sigma$, $\sigma(f)$ is the *rank* of f. It is convenient to introduce notation for the set of function symbols of rank r, $\Sigma_r = \sigma^{-1}(r)$; Σ_0 is the set of

[4] This section is, for the most part, finite automata theory redone. The proofs of the generalized theorems correspond to those in the conventional theory and the reader familiar with the latter would do wisely to omit the proofs in his reading.

constant or *individual* symbols of Σ and in all cases, we assume $\Sigma_0 \neq \varnothing$. The species $\langle \Sigma, \sigma \rangle$ is *finite* if Σ is finite; we will be concerned only with finite species.

An *algebra* of species $\langle \Sigma, \sigma \rangle$, a Σ-algebra for short, is a pair $\mathscr{A} = \langle A, \alpha \rangle$, where A is a set called the *carrier* of $\mathscr{A}$ and α is a function from Σ into the class of operations on A such that $\alpha(f) = \alpha_f$ is a $\sigma(f)$-place function: $\alpha(f) \in A^{A^{\sigma(f)}}$. The 0-place functions ($\alpha(\lambda)$ for $\lambda \in \Sigma_0$) are called *constants.*

A *non-deterministic algebra* (for lack of a better term) of species $\langle \Sigma, \sigma \rangle$ is a relational system of species $\langle \Sigma, \sigma + 1 \rangle$, that is, a pair $\mathscr{R} = \langle R, \rho \rangle$ where $\rho(f) \subseteq R^{\sigma(f)+1}$. Again R is called the *carrier* of $\mathscr{R}$, and $\mathscr{R}$ is finite if R is finite.

A species $\langle \Sigma, \sigma \rangle$ uniquely determines a set T_Σ of (constant) *terms* defined to be the least subset of Σ^* (the set of all words on Σ) satisfying

(i) $\Sigma_0 \subseteq T_\Sigma$;

(ii) If $f \in \Sigma_n$ and $t_1, \cdots, t_n \in T_\Sigma$ then $ft_1 \cdots t_n \in T_\Sigma$.

One algebra of species $\langle \Sigma, \sigma \rangle$ arises immediately by using T_Σ as carrier; the *generic* (totally free) algebra $\mathscr{F}_\Sigma$ is the Σ-algebra $\langle T_\Sigma, \iota \rangle$, where

(i) For $\lambda \in \Sigma_0$, $\iota_\lambda = \lambda$;

(ii) For $f \in \Sigma_n$ and $t_1, \cdots, t_n \in T_\Sigma$, $\iota_f(t_1, \cdots, t_n) = ft_1 \cdots t_n$.

With these definitions, we are, in effect, identifying a species with a particular choice of a term language for what is often called an algebraic similarity type. A different choice of symbols for the term language of a similarity type yields a different species. For example, $\Sigma_0 = \{\lambda\}$, $\Sigma_1 = \{1, 2, \cdots, k\}$ and $\Sigma = \Sigma_0 \cup \Sigma_1$ give rise to a species for "initialized k-monadic algebras". An algebra $\mathscr{A} = \langle A, \alpha \rangle$ of this species can be written $\mathscr{A} = \langle A, a_0, \alpha_1, \cdots, \alpha_k \rangle$, where $a_0 = \alpha(\lambda)$ and $\alpha_i \colon A \to A$ is $\alpha(i)$. With $\Sigma' = \{\phi, s_0, \cdots, s_{k-1}\}$, we obtain a different species for essentially the same class of algebras.

The term "similarity-type" has been used loosely in the previous paragraph; this usage could be made more precise by introducing a canonical species. A refinement of this kind is beyond the requirements of our present undertaking.

We will diverge considerably from the usual course of algebraic investigations by looking at algebras of a given species as characterizing or recognizing certain subsets of the set of terms. This divergence is emphasized by our interest in operations on species (projection is the most important example) and the closure of recognizable sets under such operations. Our development is motivated by—in fact, it is a direct generalization of—conventional finite automata theory. To facilitate comparisons with this theory, we will apply its terminology to the structures under consideration.

Given a species $\langle \Sigma, \sigma \rangle$, Σ is the set of *input* (function) *symbols* and T_Σ is the universe of possible *input terms* (strings). A [*non-deterministic*] *automaton* of species $\langle \Sigma, \sigma \rangle$ is a *finite* [non-deterministic] algebra of that species; with $\mathscr{A} = \langle A, \alpha \rangle$, A is the set of *states*, for each $\lambda \in \Sigma_0$, α_λ is an *initial state* and for $f \in \Sigma_n$, α_f is the (direct) *transition function* for the input symbol f. The terminology is similar for non-deterministic automata; ρ_λ is a set of initial states corresponding to the individual symbol λ and ρ_f is the *transition relation* for the input f.

So the inputs to a Σ-automaton $\mathscr{A} = \langle A, \alpha \rangle$ are terms from the set T_Σ and each input t produces an *output state* $h_{\mathscr{A}}(t) \in A$ defined as follows:

(i) For $\lambda \in \Sigma_0$, $h_{\mathscr{A}}(\lambda) = \alpha_\lambda$;

(ii) For $f \in \Sigma_n$ and $t_1, \cdots, t_n \in T_\Sigma$,

$$h_{\mathscr{A}}(ft_1 \cdots t_n) = \alpha_f(h_{\mathscr{A}}(t_1), \cdots, h_{\mathscr{A}}(t_n)).$$

The symbolism $h_{\mathscr{A}}(t)$ is used here because this state is nothing more than the image of t under the natural homomorphism from $\mathscr{F}_\Sigma$ into $\mathscr{A}$ induced by extending the map $\alpha|\Sigma_0$.

For a non-deterministic automaton $\mathscr{R} = \langle R, \rho \rangle$, the definition differs only slightly. Here, an input term produces a set of output states.

(i) For $\lambda \in \Sigma_0$, $h_{\mathscr{R}}(\lambda) = \rho_\lambda$.

(ii) For $f \in \Sigma_n$ and $t_1, \cdots, t_n \in T_\Sigma$,

$$h_{\mathscr{R}}(ft_1 \cdots t_n) = \{a |\ \exists a_1 \cdots \exists a_n[\rho_f(a_1, \cdots, a_n, a) \wedge \bigwedge_i a_i \in h_{\mathscr{R}}(t_i)]\}.$$

The inductive clause of the definition of $h_{\mathscr{R}}$ can be rewritten to look exactly like that for deterministic automata by using the concept of the n-place set function $\hat{\eta}$ induced by the $(n + 1)$-place relation η. $\hat{\eta}$ is defined by:

$$\hat{\eta}(\beta_1, \cdots, \beta_n) = \{b |\ \exists b_1 \cdots \exists b_n[\eta(b_1, \cdots, b_n, b) \wedge \bigwedge_i b_i \in \beta_i]\}.$$

Then, of course, (ii) can be written:

(ii′) For $f \in \Sigma_n$ and $t_1, \cdots, t_n \in T_\Sigma$,

$$h_{\mathscr{R}}(ft_1 \cdots t_n) = \hat{\rho}_f(h_{\mathscr{R}}(t_1), \cdots, h_{\mathscr{R}}(t_n)).$$

This similarity between deterministic and non-deterministic automata forecasts the equivalence theorem proved below. Before proving that theorem, however, we must define the concept of the *behavior* of, or set of terms *recognized* by, an automaton.

For a deterministic automaton $\mathscr{A} = \langle A, \alpha \rangle$ and a choice of *final states* $A_F \subseteq A$, the *behavior* of $\mathscr{A}$ with respect to A_F is given by

$$bh_{\mathscr{A}}(A_F) = \{t |\ h_{\mathscr{A}}(t) \in A_F\}.$$

Similarly, if $\mathscr{R} = \langle R, \rho \rangle$ is a non-deterministic automaton and $R_F \subseteq R$, then

$$bh_{\mathscr{R}}(R_F) = \{t |\ h_{\mathscr{R}}(t) \cap R_F \neq \varnothing\}.$$

A set $U \subseteq T_\Sigma$ is *recognizable* if there exists a Σ-automaton $\mathscr{A}$ (deterministic or non-deterministic) and set A_F of final states for $\mathscr{A}$ such that $bh_{\mathscr{A}}(A_F) = U$.

THEOREM 1. (Equivalence of non-deterministic and deterministic automata: the subset construction) *A set $U \subseteq T_\Sigma$ is recognizable by a deterministic Σ-automaton if and only if U is recognizable by a non-deterministic Σ-automaton.*

Proof. One direction of this equivalence is trivial because any deterministic automaton $\mathscr{A} = \langle A, \alpha \rangle$ corresponds to a non-deterministic automaton $\mathscr{A}' = \langle A, \alpha' \rangle$ where a'_f is the graph of α_f ($\alpha'_f(a_1, \cdots, a_n, a) \leftrightarrow \alpha_f(a_1, \cdots, a_n) = a$). For any $A_F \subseteq A$, $bh_{\mathscr{A}}(A_F) = bh_{\mathscr{A}'}(A_F)$ because $h_{\mathscr{A}'}(t) = \{h_{\mathscr{A}}(t)\}$.

For the other direction (which is also obvious if we rely on our intuition from conventional finite automata theory), we would be given a non-deterministic automaton $\mathscr{R} = \langle R, \rho \rangle$ such that $U = bh_{\mathscr{R}}(R_F)$. Then, from $\mathscr{R}$ we construct the "subset automaton" $\mathscr{A} = \langle 2^R, \hat{\rho} \rangle$.(The notation $\hat{\rho}$ is not quite accurate; we mean $\hat{\rho}(f) = \hat{\rho}_f$.) A simple inductive proof (which is left to the reader) shows that in fact $\mathscr{R}$ and $\mathscr{A}$ have the *same* output function: $h_{\mathscr{A}}(t) = h_{\mathscr{R}}(t)$. The final states of $\mathscr{A}$ are chosen as follows:

$$A_F = \{u \mid u \subseteq R \text{ and } u \cap R_F \neq \varnothing\}.$$

Thus,

$$t \in bh_{\mathscr{A}}(A_F) \leftrightarrow h_{\mathscr{A}}(t) \in A_F \leftrightarrow h_{\mathscr{R}}(t) \cap R_F \neq \varnothing \leftrightarrow t \in bh_{\mathscr{R}}(R_F).$$

Therefore, $bh_{\mathscr{A}}(A_F) = bh_{\mathscr{R}}(R_F)$ and the equivalence theorem is established.

Having demonstrated the equivalence of deterministic and non-deterministic automata, we will use the term "recognizable" without reference to the kind of automaton involved. As is familiar from conventional finite automata theory, there are purposes for which non-deterministic automata are necessary; others require the use of deterministic automata. Proof of the following closure theorem falls in the latter category; the theorems on closure of recognizable sets under projection and the regular operators depend on the use of non-deterministic automata.

THEOREM 2. (Closure of the recognizable sets under the Boolean operations) *If U and V are recognizable subsets of T_Σ, then $U \cap V$ and $T_\Sigma - U$ are recognizable.*

Proof. Let $\mathscr{A}$ and $\mathscr{B}$ be Σ-automata such that $bh_{\mathscr{A}}(A_F) = U$ and $bh_{\mathscr{B}}(B_F) = V$. Then

(i) $$bh_{\mathscr{A}}(A - A_F) = T_\Sigma - U,$$

(ii) $$bh_{\mathscr{A} \times \mathscr{B}}(A_F \times B_F) = U \cap V,$$

where $\mathscr{A} \times \mathscr{B}$ is the direct product of the algebras $\mathscr{A}$ and $\mathscr{B}$. Because of the comparison with the conventional theory, the details of the proofs of (i) and (ii) will be omitted.

The Boolean closure theorem deals with operations on subsets of terms of a single species; in contrast, the projection theorems to be considered now relate recognizable subsets of possibly distinct species. The effect of a projection is simply to change symbols (sometimes in a many-to-one fashion) in the terms. This being the case, it is not surprising to find that the class of all recognizable sets is closed under projections.

To be more precise about the concepts involved, let $\langle \Sigma, \sigma \rangle$ and $\langle \Omega, \omega \rangle$ be two species. A map $\pi: \Sigma \to \Omega$ which satisfies the condition $\sigma(f) = \omega\pi(f)$ for all $f \in \Sigma$ can be extended to a map $\overline{\pi}: T_\Sigma \to T_\Omega$ as follows:

(i) For $\lambda \in \Sigma_0$, $\overline{\pi}(\lambda) = \pi(\lambda)$;

(ii) For $f \in \Sigma_n$ and $t_1, \cdots, t_n \in T_\Sigma$, $\overline{\pi}(ft_1 \cdots t_n) = \pi(f)\overline{\pi}(t_1) \cdots \overline{\pi}(t_n)$.

Thus we are considering maps between two sets of function symbols; the extension to terms is the natural one. Any mapping so obtained is called a *projection*. The next two theorems show that recognizable sets are closed under both projections and inverse projections.

THEOREM 3. (Closure of the recognizable sets under projection) *If U is a recognizable subset of T_Σ and if $\bar{\pi}$ is a projection of T_Σ into T_Ω, then $\bar{\pi}(U)$ is a recognizable subset of T_Ω.*

Proof. Let $\mathscr{A} = \langle A, \alpha \rangle$ be a Σ-automaton with $bh_{\mathscr{A}}(A_F) = U$. We construct the non-deterministic Ω-automaton $\mathscr{R} = \langle A, \rho \rangle$ with transition relations:

(i) For $\lambda \in \Omega_0$, $\rho_\lambda = \{\alpha_\delta | \delta \in \Sigma_0 \text{ and } \pi(\delta) = \lambda\}$;

(ii) For $f \in \Omega_n$ and $a_1, \cdots, a_n, a \in A$, $\rho_f(a_1, \cdots, a_n, a) \leftrightarrow \alpha_g(a_1, \cdots, a_n) = a$ for some g with $\pi(g) = f$.

The proof of the required behavioral property, namely,

$$\text{(I)} \qquad \bar{\pi}[bh_{\mathscr{A}}(A_F)] = bh_{\mathscr{R}}(A_F),$$

depends on the following two properties:

(II) For all $t \in T_\Sigma$, $h_{\mathscr{A}}(t) \in h_{\mathscr{R}}(\pi(t))$;

(III) For all $t' \in T_\Omega$, $a \in h_{\mathscr{R}}(t') \rightarrow h_{\mathscr{A}}(t) = a$ for some t with $\pi(t) = t'$.

From (II) we find that if $t \in bh_{\mathscr{A}}(A_F)$ then $h_{\mathscr{A}}(t) \in h_{\mathscr{R}}(\pi(t))$ and thus $h_{\mathscr{R}}(\pi(t)) \cap A_F \neq \varnothing$ and $\pi(t) \in bh_{\mathscr{R}}(A_F)$; $\bar{\pi}[bh_{\mathscr{A}}(A_F)] \subseteq bh_{\mathscr{R}}(R_F)$. The converse inclusion comes from (III). The proofs of (II) and (III) are both inductive and quite similar; we will prove only (III).

For $t' = \lambda \in \Omega_0$, if $a \in \rho_\lambda$ then by (i) above $a = \alpha_\delta$ for some δ with $\pi(\delta) = \lambda$. Thus, in the conclusion of (III), t is taken to be δ; $h_{\mathscr{A}}(t) = \alpha_\delta$ and $\pi(t) = \lambda$. Now let f be an n-place function symbol of Ω and assume (III) is true for terms $t'_1, \cdots, t'_n$. We are looking at the term $t' = ft'_1 \cdots t'_n$. But if $a \in h_{\mathscr{R}}(t')$, then, by the definition of $h_{\mathscr{R}}$, there exist $a_1, \cdots, a_n$ with $a_i \in h_{\mathscr{R}}(t'_i)$ such that $\rho_f(a_1, \cdots, a_n, a)$. This means (by (ii) above) that there is a function symbol of $g \in \Sigma$ with $\pi(g) = f$ and $\alpha_g(a_1, \cdots, a_n) = a$. Now applying the induction hypothesis to the t'_i we know that there are Σ-terms $t_1, \cdots, t_n$ with $\pi(t_i) = t'_i$ and $h_{\mathscr{A}}(t_i) = a_i$. Thus t is taken to be $gt_1 \cdots t_n$ and we have (by the definition of $\bar{\pi}$ and of $h_{\mathscr{A}}$) $\bar{\pi}(t) = t'$ and $h_{\mathscr{A}}(t) = \alpha_g(h_{\mathscr{A}}(t_1), \cdots, h_{\mathscr{A}}(t_n)) = a$. Thus by induction we have established property (III) which, with indicated details filled in by the reader, completes the proof of Theorem 3.

THEOREM 4. (Closure of the recognizable sets under inverse projection) *If $U \subseteq T_\Omega$ is recognizable and if $\bar{\pi}$ is a projection of T_Σ into T_Ω, then $\bar{\pi}^{-1}(U)$ is a recognizable subset of T_Ω.*

Proof. Let $\mathscr{A} = \langle A, \alpha \rangle$ be a Σ-automaton such that $bh_{\mathscr{A}}(A_F) = U$. An Ω-automaton $\mathscr{B} = \langle A, \beta \rangle$ is constructed with $\beta_f = \alpha_{\pi(f)}$ for all symbols $f \in \Sigma$. By induction we will prove that

$$\text{(I)} \qquad h_{\mathscr{B}}(t) = h_{\mathscr{A}}(\pi(t)).$$

For $t = \lambda \in \Sigma_0$ this is certainly true by the definition of β: $h_{\mathscr{B}}(\lambda) = \beta_\lambda = \alpha_{\pi(\lambda)} =$

$h_{\mathscr{A}}(\pi(\lambda))$. Assuming that (I) is true for terms $t_1, \cdots, t_n$ we consider $t = ft_1 \cdots t_n$. Then,

$$h_{\mathscr{B}}(t) = \beta_f(h_{\mathscr{B}}(t_1), \cdots, h_{\mathscr{B}}(t_n)) = \alpha_{\pi(f)}(h_{\mathscr{A}}(\bar{\pi}(t_1)), \cdots, h_{\mathscr{A}}(\bar{\pi}(t_n))).$$

But the right-hand side of this equation is simply $h_{\mathscr{A}}(\pi(t))$.

Now if $t \in \bar{\pi}^{-1}(U)$, there is some t' with $h_{\mathscr{A}}(t') \in A_F$ and $\pi(t) = t'$. But by (I) $h_{\mathscr{A}}(t') = h_{\mathscr{B}}(t) \in A_F$ and thus $t \in bh_{\mathscr{B}}(A_F)$. Therefore, we have $\bar{\pi}^{-1}(U) \subseteq bh_{\mathscr{B}}(A_F)$. For the converse inclusion, assume that $t \in bh_{\mathscr{B}}(A_F)$. Then $h_{\mathscr{A}}(\bar{\pi}(t)) \in A_F$ and hence $\bar{\pi}(t) \in U$, which assures us that $t \in \bar{\pi}^{-1}(U)$. So by our construction, the automaton $\mathscr{B}$ recognizes $\bar{\pi}^{-1}(U)$, and Theorem 4 is established.

In the application of generalized recognizability to the solution of a decision problem (Section 4), we will interpret subsets of T_Σ (with appropriate choices of Σ) as representing definable relations of a restricted second-order theory. Under this interpretation, we can say that a relation is recognizable. It will be necessary to prove that certain primitive relations are recognizable and then to show that the recognizable relations are closed under the propositional connectives and quantification. This latter task is accomplished through applications of the closure theorems that we have proved so far; we could prove recognizability of the primitive relations by actually constructing automata that work. It is neater to use a theory of regularity—the theory is of interest independent of the application—and this will be developed in the next section.

The application to a decision problem would be fruitless if it were not for the fact that certain questions concerning generalized automata can be effectively handled. As is the case for the conventional theory of finite automata, we would be justified in making the gross statement that all problems can be effectively handled! However, our application requires only one proof; we will show that there is an effective procedure for answering the question "$bh_{\mathscr{A}}(A_F) = \varnothing$?"

The effective solution to the question "$bh_{\mathscr{A}}(A_F) = \varnothing$?" for conventional finite automata depends on testing all input strings of length less than or equal to the number of states of $\mathscr{A}$ for production of an output state in A_F; a longer input w must have a repetition of intermediate states and thus a "chunk" of w can be removed to obtain a shorter string with the identical output. The same can be said in the generalized case although we need to be more precise about "length" and "chunk removal". Since each Σ-term t is actually a string ($t \in \Sigma^*$), we want to maintain the concept of length, $l(t)$, as it has meaning there, namely, the number of symbols of t; instead we will talk about the *depth* of a term (which is intuitively more meaningful) and define it recursively by

(i) $$d(\lambda) = 1 \text{ for } \lambda \in \Sigma_0,$$

(ii) $$d(ft_1 \cdots t_n) = \max_i \{d(t_i)\} + 1.$$

The process of modifying an input to produce the same output state reduces, in the general case, to the familiar concept of replacement. It is convenient to introduce the idea of subterm; t' is a *subterm* of t if there exists a term t'' such that t is the result of replacing some occurrence of an individual symbol in t'' by t'. Using the notation $t \succeq t'$ for "t has t' as a subterm", a formal definition is given by:

(i) For $\lambda \in \Sigma_0$, $\lambda \succeq t'$ if and only if $\lambda = t'$;

(ii) For $f \in \Sigma_n$, $t_1, \cdots, t_n \in T_\Sigma$ and $t = ft_1 \cdots t_n$, $t \succeq t'$ if and only if $t = t'$ or $t_i \succeq t'$ for some i, $1 \leqslant i \leqslant n$.

The critical property of replacement relevant for finite automata is that if a subterm t_1 of t is replaced by a term t_2 which produces the same output state as t_1, then the result of the replacement produces the same output as t. This fact, analogous to the replacement lemmas of logic, is proved below.

LEMMA 5. (The replacement lemma) *If $\mathscr{A}$ is a Σ-automaton and t_1, t_2, t, t' are Σ-terms such that t' is obtained from t by replacing an occurrence of t_2 by t_1, and if $h_{\mathscr{A}}(t_1) = h_{\mathscr{A}}(t_2)$, then $h_{\mathscr{A}}(t) = h_{\mathscr{A}}(t')$.*

Proof. By induction on the construction of terms, if $t = \lambda$, then $t_2 = t$ and $t' = t_1$; $h_{\mathscr{A}}(t) = h_{\mathscr{A}}(t_2) = h_{\mathscr{A}}(t_1) = h_{\mathscr{A}}(t')$. For $t = ft_1' \cdots t_n'$, if the replacement occurs in one of the t_i''s, then the induction hypothesis yields $h_{\mathscr{A}}(t) = h_{\mathscr{A}}(t')$. If $t_2 = t_j'$—for convenience, say t_1'—then $t' = ft_1t_2' \cdots t_n'$ and since $h_{\mathscr{A}}(t_1) = h_{\mathscr{A}}(t_1')$, it is clear that $h_{\mathscr{A}}(t) = h_{\mathscr{A}}(t')$.

By repeated application of the replacement lemma to non-overlapping or incomparable occurrences of subterms in a term t we obtain the following corollary which will be useful in Section 3.

COROLLARY. *Given incomparable (non-overlapping) occurrences of subterms $t_1, \cdots, t_s$ of a term t, if t' results from replacing t_i by t_i' and if $h_{\mathscr{A}}(t_i) = h_{\mathscr{A}}(t_i')$ $(i = 1, \cdots, s)$, then $h_{\mathscr{A}}(t) = h_{\mathscr{A}}(t')$.*

We can now prove the property that will yield the effective solution to the emptiness problem.

LEMMA 6. *Let $\mathscr{A}$ be a Σ-automaton with n states. If there exists a term t such that $h_{\mathscr{A}}(t) = a \in A$, then there exists a term t' such that $d(t') \leqslant n$ and $h_{\mathscr{A}}(t') = a$.*

Proof. We have $h_{\mathscr{A}}(t) = a$. If $d(t) \leqslant n$, then we are done. If $d(t) > n$, then there is at least one sequence of subterms $t = t_0 \succ t_1 \succ \cdots \succ t_{d(t)} \in \Sigma_0$. ($t \succ t'$ if and only if $t \succeq t'$ and $t \neq t'$.) As is familiar from the conventional theory, the corresponding sequence of states $a_i = h_{\mathscr{A}}(t_i)$ must contain a repetition, say $a_i = a_j$ for $i < j$. Then by the replacement lemma, if t_j replaces t_i in t to obtain t', then $h_{\mathscr{A}}(t) = h_{\mathscr{A}}(t')$. Now $d(t') \leqslant l(t') < l(t)$ and thus if $d(t')$ is still greater than n, the process above can be repeated until a reduced term is obtained which has depth less than or equal to n and which produces the output state a.

THEOREM 7. (Effective solution to the emptiness problem) *Given a Σ-automaton $\mathscr{A}$, there is an effective procedure for determining whether or not $bh_{\mathscr{A}}(A_F) = \varnothing$.*

Proof. By the previous lemma it is obvious that the procedure is simply to test all terms t with $d(t) \leqslant n$, where n is the number of states of $\mathscr{A}$. If any such term has an output state in A_F, $bh_{\mathscr{A}}(A_F) \neq \varnothing$; otherwise, $bh_{\mathscr{A}}(A_F) = \varnothing$.

3. The Basis for a Generalized Theory of Regularity

In this section we will develop an algebraic characterization of the recognizable sets analogous to what might be called the Kleene theory for conventional finite automata. To this end, we begin by introducing complex product and closure operations on sets of terms. Let $\langle \Sigma, \sigma \rangle$ be a species; for each $\lambda \in \Sigma_0$ there is a product and a closure operation denoted $U \cdot_\lambda V$ and U^λ (analogous to $U \cdot V$ and U^*). For sets $U, V \subseteq T_\Sigma$, $U \cdot_\lambda V$ is the set of all terms $t \in T_\Sigma$ for which there exists $t' \in U$ such that t is obtained by replacing every occurrence of λ in t' by some element of V. Note that various occurrences of λ may be replaced by different choices from V.

The following examples of the complex product illustrate some important cases.

(a) $U \cdot_\lambda \{\lambda\} = U$, $\{\lambda\} \cdot_\lambda U = U$;

(b) $U \cdot_\lambda \varnothing$ is the set of terms of U which do not involve λ;

(c) If $U_1 \subseteq U_2$ and $V_1 \subseteq V_2$, then $U_1 \cdot_\lambda V_1 \subseteq U_2 \cdot_\lambda V_2$.

The closure operation is defined as it is in the conventional theory by first describing a sequence of sets:

$$X^0 = \{\lambda\},$$

$$X^{n+1} = X^n \cup U \cdot_\lambda X^n,$$

and then

$$U^\lambda = \bigcup_{n=0}^{\infty} X^n.$$

Since $X^1 = U \cup \{\lambda\}$, we have $U \subseteq U^\lambda$ and it is easily verified that $U \cdot_\lambda U^\lambda \subseteq U^\lambda$. The following are examples of the use of the closure operators.

(d) Let $A_\Sigma = \{f\lambda_1 \cdots \lambda_n \mid \lambda_i \in \Sigma_0, f \in \Sigma_n\}$; A_Σ is the set of atomic (non-individual) terms. If $\Sigma_0 = \{\lambda\}$, then $A_\Sigma^\lambda = T_\Sigma$. If $\Sigma_0 = \{\lambda, \delta\}$, then $(A_\Sigma^\lambda)^\delta = (A_\Sigma^\delta)^\lambda = T_\Sigma$.

The last example suggests the possibility (which would be nice) that the various closure operators commute; this is dispelled by the following sequence of examples, ($\Sigma = \{g, \lambda, \delta\}$, $\Sigma_0 = \{\lambda, \delta\}$, $\Sigma_1 = \{g\}$).

(e) $$\{g\lambda\}^\delta = \{\delta, g\lambda\};$$

(f) $$\{g\lambda\}^\lambda = \{\lambda, g\lambda, gg\lambda, ggg\lambda, \cdots\};$$

(g) $$(\{g\lambda\}^\delta)^\lambda = \{\delta, g\lambda\}^\lambda = T_\Sigma;$$

(h) $$(\{g\lambda\}^\lambda)^\delta = \{\delta\} \cup \{g\lambda\}^\lambda \neq T_\Sigma.$$

Motivated by finite automata theory, these operations suggest a natural definition of regularity: The class of Σ-*regular* subsets of T_Σ is the least class of subsets of T_Σ which contains the finite subsets and is closed under $\cup$, $\cdot_\lambda$ and $^\lambda$ for all $\lambda \in \Sigma_0$. And then the desired theorem would be: Σ-regular if and only if recognizable by a Σ-automaton. Unfortunately, this does not go through as we have it now. For example, take $\Sigma_2 = \{h\}$, and define V

inductively by

(i) $\lambda \in V$;

(ii) $t \in V$, then $h\lambda t \in V$.

Then V is recognizable but not regular (assuming $\Sigma = \{h, \lambda\}$). If we add another individual symbol, δ, then

$$V = \{h\delta\lambda\}^{\lambda} \cdot_{\delta} \{\lambda\}.$$

Thus V is Σ'-regular for the symbol set Σ' which is an extension of Σ. This leads to our second definition: $U \subseteq T_\Sigma$ is *regular* if there exists a species $\langle \Sigma', \sigma' \rangle$ such that $\Sigma'_n = \Sigma_n$ for $n \geq 1$ and U is Σ'-regular. Thus, according to this definition, the set V in the example above is regular.

This example illustrates an important feature of the definition of regularity. In order to obtain all recognizable subsets of T_Σ we need to expand the symbol set Σ to contain new individual symbols; like the production rule, non-terminal $\rightarrow$ terminal, the last equation above replaces the superfluous individual symbol by the individual of Σ.

We will now proceed to prove the main theorem of this section.

THEOREM 8. (Equivalence of recognizability and regularity) *A set $U \subseteq T_\Sigma$ is recognizable if and only if it is regular.*

The proof of this equivalence theorem is in two main parts: The analysis theorem shows that each recognizable set is regular; the synthesis theorem in turn is in four parts—we must prove that all finite subsets are recognizable and that the recognizable sets are closed under $\cup$ (Theorem 2), $\cdot_\lambda$ and $^\lambda$, thereby showing that every regular set is recognizable.

THEOREM 9. (Analysis theorem) *Every recognizable subset of T_Σ is regular.*

Proof. Let $\mathscr{A}$ be an automaton recognizing a subset $U \subseteq T_\Sigma$ with states $A = \{a_1, \cdots, a_m\}$ and final states A_F. We construct a new symbol set $\Sigma' = (\Sigma - \Sigma_0) \cup \{\lambda_1, \cdots, \lambda_m\}$ and a Σ'-automaton $\mathscr{A}'$ which is exactly like $\mathscr{A}$ except that $\alpha'_{\lambda_i} = a_i$ for $i = 1, \cdots, m$. We will first analyze the behavior of $\mathscr{A}'$ and from this analysis obtain a regular set which is the bahavior of $\mathscr{A}$. The critical feature in introducing the λ_j's is indicated by the following. Let t be any Σ'-term such that $h_{\mathscr{A}'}(t) = a \in A$. If $h_{\mathscr{A}'}(t') = a_j$ then it follows from the replacement lemma that $h_{\mathscr{A}}(t'') = a$ for any t'' obtained by replacing λ_j by t' in t. Thus λ_j serves as a kind of punctuation (like a non-terminal symbol in the definition of a language by a formal grammar) and may be replaced by any term which produces the state a_j; the result of the replacement will produce the same state as the original term.

To analyze $\mathscr{A}'$, we will define a sequence of non-deterministic automata $\mathscr{A}^1, \cdots, \mathscr{A}^m = \mathscr{A}'$. In each case the set of states is A and the transitions are functional (it is very convenient to maintain functional notation), but not defined on all of A. In particular, for $f \in \Sigma_n$, $\alpha_f^k(a_1, \cdots, a_n)$ is defined if and only if $\alpha_f(a_1, \cdots, a_n) \in \{a_1, \cdots, a_k\}$ and in case it is defined, $\alpha_f^k(a_1, \cdots,$

$a_n) = \alpha_f(a_1, \cdots, a_n)$. For the individual symbols and for all k, $\alpha^k_{\lambda_i} = a_i$. The effect of the construction is simple enough; $\mathscr{A}^k$ is just like $\mathscr{A}'$ except that non-initial transition is allowed only to the set $\{a_1, \cdots, a_k\} \subseteq A$. We will consider the sequence of automata using regular sets obtained in the analysis of $\mathscr{A}^k$ to analyze the behavior of $\mathscr{A}^{k+1}$. To this end we define, for each k and i,

$$T^k_i = \{t \mid h_{\mathscr{A}^k}(t) = a_i\}.$$

Actually, $T^k_i = bh_{\mathscr{A}^k}(\{a_i\})$. Note that we should write $h_{\mathscr{A}^k}(t) = \{a_i\}$, but since all transitions are functional, it follows that all sets involved are singletons or empty, and thus we can dispense with the set notation.

Now an inductive proof on k can be given to show that T^k_i is regular for $i = 1, \cdots, m$. For $k = 1$ and $i \neq 1$, $T^k_i = \{\lambda_i\}$, which is clearly regular; for $i = 1$, we claim that

$$\text{(I)} \qquad T^1_1 = U_1^{\lambda_1},$$

where U_k in general is defined by

$$U_k = \{f\lambda_{i_1} \cdots \lambda_{i_n} \mid f \in \Sigma_n \text{ and } \alpha_f(a_{i_1}, \cdots, a_{i_n}) = a_k\}.$$

The details of the proof of (I) will be left to the reader: Essentially $T^1_1 \subseteq U_1^{\lambda_1}$ is proved by induction on the construction of terms; $U_1^{\lambda_1} \subseteq T^2_1$ is proved by induction on the sequence used in defining $U_1^{\lambda_1}$.

The major inductive hypothesis is now that T^k_i is regular for $i = 1, \cdots, m$ and we want to show that T^{k+1}_i is regular. We get T^{k+1}_i with the following definitions:

$$V = (\cdots((U_{k+1} \cdot_{\lambda_1} T^k_1) \cdot_{\lambda_2} T^k_2) \cdots) \cdot_{\lambda_k} T^k_k,$$

$$W = V^{\lambda_{k+1}}.$$

Then we want to prove that

$$\text{(II)} \qquad T^{k+1}_i = T^k_i \cdot_{\lambda_{k+1}} W, \qquad i = 1, \cdots, k+1.$$

Of course, T^{k+1}_i for $i > k+1$ is simply $\{\lambda_i\}$.

For $t = f\lambda_{i_1} \cdots \lambda_{i_n} \in U_{k+1}$, $h_{\mathscr{A}^{k+1}}(t) = a_{k+1}$ by the definition of U_{k+1}. Therefore, by the replacement lemma (if $t' \in \{t\} \cdot_{\lambda_i} T^k_i$, then $h_{\mathscr{A}'}(t') = h_{\mathscr{A}'}(t)$) we see that any term in V will also produce the output state a_{k+1}. By identical reasoning, if $t \in W$ then $h_{\mathscr{A}^{k+1}}(t) = a_{k+1}$. Finally, if $t \in T^k_i \cdot_{\lambda_{k+1}} W$ it again follows from the replacement lemma that $h_{\mathscr{A}^{k+1}}(t) = a_i$. We have $T^k_i \cdot_{\lambda_{k+1}} W \subseteq T^{k+1}_i$.

For equation (II), inclusion in the other direction is more difficult. We will attack this by proving the following statement using induction on n.

(III) If $h_{\mathscr{A}^{k+1}}(t) = a_i$ and there are exactly n occurrences of non-initial transitions to a_{k+1} in the computation of $h_{\mathscr{A}^{k+1}}(t)$, then $t \in T^k_i \cdot_{\lambda_{k+1}} X^n$, where $X^0 = \{\lambda_{k+1}\}$ and $X^{n+1} = X^n \cup V \cdot_{\lambda_{k+1}} X^n$.

If there are no non-initial transitions to a_{k+1}, then $t \in T_i^k = T_i^k \cdot_{\lambda_{k+1}} \{\lambda_{k+1}\}$. Assume that (III) is true for n and consider a term t such that $h_{\mathscr{A}^{k+1}}(t) = a_i$ and the computation of $h_{\mathscr{A}^{k+1}}(t)$ has exactly $n+1$ transitions to a_{k+1}. Under this situation there exist terms t_1, t_2 such that (1) $h_{\mathscr{A}^{k+1}}(t_1) = a_i$ and there are exactly n transitions to a_{k+1} in computing $h_{\mathscr{A}^{k+1}}(t_1)$, (2) $h_{\mathscr{A}^{k+1}}(t_2) = a_{k+1}$, and (3) t results from replacing one occurrence of λ_{k+1} in t_1 by t_2 (all others by λ_{k+1}).

From (2) it follows that $t_2 = ft_1 \cdots t_n$ and for appropriate choices of $\lambda_{i_j}, f\lambda_{i_1} \cdots \lambda_{i_n} \in U_{k+1}$. There can be no non-initial transitions to a_{k+1} in the computation of each of the $h_{\mathscr{A}^{k+1}}(t_j)$ (because there is only one in t_2) and thus each $t_j \in T_{i_j}^k$; hence $t_2 \in V$. By (1) and the induction hypothesis, $t_1 \in T_i^k \cdot_{\lambda_{k+1}} X^n$ so combined with (3) we get $t \in (T_i^k \cdot_{\lambda_{k+1}} X^n) \cdot_{\lambda_{k+1}} V$. It is easily verified that $(T_i^k \cdot_{\lambda_{k+1}} X^n) \cdot_{\lambda_{k+1}} V \subseteq T_i^k \cdot_{\lambda_{k+1}} X^{n+1}$. Thus, by induction, we have established (III). Since $\bigcup_n T_i^k \cdot_{\lambda_{k+1}} X^n = T_i^k \cdot_{\lambda_{k+1}} W$ and we have shown that $T_i^{k+1} \subseteq \bigcup_n T_i^k \cdot_{\lambda_{k+1}} X^n$ it follows that equation (II) is verified.

This completes the major inductive proof that T_i^k is regular. Now we want to relate these results to the behavior of the original automaton $\mathscr{A}$.

We have shown that each of the sets T_i^m is regular and that T_i^m is the set of all Σ'-terms which yield the state a_i in the automaton $\mathscr{A}'$. If we were only worried about $\mathscr{A}'$, then

$$bh_{\mathscr{A}'}(A_F) = \bigcup_{a_i \in A_F} T_i^m.$$

In effect, we have the behavior of $\mathscr{A}$ except for initial states. If we make the definition

$$\Lambda_j = \alpha^{-1}(a_j),$$

then

$$bh_{\mathscr{A}}(A_F) = bh_{\mathscr{A}'}(A_F) \cdot_{\lambda_1} \Lambda_1 \cdot_{\lambda_2} \Lambda_2 \cdots \cdot_{\lambda_m} \Lambda_m .$$

This works because (1) if a_j is not an initial state, then $\Lambda_j = \varnothing$ and all terms involving λ_j are eliminated by the product, and (2) if a_j is an initial state, then Λ_j is the set of individual symbols for which it is such, and in the product λ_j may be replaced by any of the individuals in Λ_j. We have shown that T_i^m is regular. Thus, $bh_{\mathscr{A}'}(A_F)$ is regular, and finally the last equation gives us the desired result; $bh_{\mathscr{A}}(A_F)$ is regular.

As was already mentioned, the synthesis theorem is in four parts, the first of which we will state without proof.

LEMMA 10. *All finite subsets of T_Σ are recognizable.*

The simplest proof of this lemma inductively constructs automata to recognize the singleton sets, and, with Theorem 2, all finite sets are obtained.

We have already proved closure of the recognizable sets under set union; the next two lemmas give us closure under the complex product and closure operations.

LEMMA 11. (Closure of the recognizable sets under $\cdot_\delta$) *If U and V are recognizable subsets of T_Σ, then $U \cdot_\delta V$ is recognizable for any $\delta \in \Sigma_0$.*

Proof. Given automata $\mathscr{A}$ and $\mathscr{B}$ (final states A_F, B_F) which recognize U and V, respectively, we construct a non-deterministic automaton $\mathscr{R} = \langle R, \rho\rangle$ with $R = A \cup B$ and the transition relations defined as follows.

(i) For $\lambda \in \Sigma_0$,

$$\rho_\lambda = \begin{cases} \{\alpha_\delta, \beta_\lambda, \alpha_\lambda\} & \text{if } \lambda \neq \delta \text{ and } \beta_\lambda \in B_F, \\ \{\alpha_\lambda, \beta_\lambda\} & \text{if } \lambda \neq \delta \text{ and } \beta_\lambda \notin B_F, \\ \{\alpha_\delta, \beta_\delta\} & \text{if } \lambda = \delta \text{ and } \beta_\delta \in B_F, \\ \{\beta_\delta\} & \text{if } \lambda = \delta \text{ and } \beta_\delta \notin B_F. \end{cases}$$

(ii) For $f \in \Sigma_n$, $a_i \in A$, $b_i \in B$ and $X \in R$,

$$\rho_f(a_1, \cdots, a_n, X) \leftrightarrow \alpha_f(a_1, \cdots, a_n) = X,$$

$$\rho_f(b_1, \cdots, b_n, X) \leftrightarrow \beta(b_1, \cdots, b_n) = X, \text{ or}$$

$$\beta_f(b_1, \cdots, b_n) \in B_F \text{ and } X = \alpha_\delta.$$

We claim that $bh_{\mathscr{R}}(A_F) = U \cdot_\delta V$. This assertion is a consequence of the following two propositions:

$$\text{(I)} \qquad a \in h_{\mathscr{R}}(t) \leftrightarrow \exists t'[t \in \{t'\} \cdot_\delta V \text{ and } h_{\mathscr{A}}(t') = a],$$

$$\text{(II)} \qquad b \in h_{\mathscr{R}}(t) \leftrightarrow h_{\mathscr{B}}(t) = b.$$

If $t \in bh_{\mathscr{R}}(A_F)$, then $h_{\mathscr{R}}(t) \cap A_F \neq \varnothing$. Thus there is some $a \in h_{\mathscr{R}}(t)$ which is also a final state of $\mathscr{A}$. But (I) says that there is a $t' \in U$ such that $t \in \{t'\} \cdot_\delta V$ and thus $t \in U \cdot_\delta V$. The converse, i.e., $U \cdot_\delta V \subseteq bh_{\mathscr{R}}(A_F)$, also follows immediately from (I). In effect, we need (II) to carry out the inductive proof of (I) which we now begin.

For $t \in \Sigma_0$, consider first the case $t = \lambda \neq \delta$. If $a \in h_{\mathscr{R}}(\lambda)$ then from (i) either (a) $a = \alpha_\lambda$ or (b) $a = \alpha_\delta$ and $\beta_\lambda \in B_F$. In case (a), take t' to be λ. Then $\lambda \in \{\lambda\} \cdot_\delta V$ and $h_{\mathscr{A}}(\lambda) = \alpha_\lambda$. In case (b), take $t' = \delta$. Since $\lambda \in V$ we have $\lambda \in \{\delta\} \cdot_\delta V$ and also $h_{\mathscr{A}}(\delta) = \alpha_\delta$. Now if $t = \delta$ then $a \in h_{\mathscr{R}}(t)$ if and only if $a = \alpha_\delta$ and $\beta_\delta \in B_F$. But under this circumstance $\delta \in V$ and thus with $t' = \delta$, $t \in \{\delta\} \cdot_\delta V$ and $h_{\mathscr{A}}(\delta) = \alpha_\delta$. All this just takes care of the basis proof ($t \in \Sigma_0$) of one direction of (I), call it (I $\rightarrow$). We leave (I $\leftarrow$) to the reader as the proof is similarly accomplished by cases. For the basis case of (II), we observe that (i) gives us $b \in h_{\mathscr{R}}(\lambda) \leftrightarrow b = \beta_\lambda$ (whether or not $\lambda = \delta$) and $h_{\mathscr{B}}(\lambda) = \beta_\lambda$, so (II) is easily verified.

For the inductive part of the proof, assume (I) and (II) are true for terms $t_1, \cdots, t_n$ and let f be an n-place function symbol with $t = ft_1 \cdots t_n$. Attacking (I $\rightarrow$), if $a \in h_{\mathscr{R}}(t)$ there are, by (ii), two cases to consider. Either (a) there exist $a_i \in h_{\mathscr{R}}(t_i)$ with $\alpha_f(a_1, \cdots, a_n) = a$ or (b) there exist $b_i \in h_{\mathscr{R}}(t_i)$ with $\beta_f(b_1, \cdots, b_n) \in B_F$ and $a = \alpha_\delta$. Looking at the latter case, the inductive hypothesis (II) gives us $h_{\mathscr{B}}(t_i) = b_i$ and therefore $h_{\mathscr{B}}(t) \in B_F$, $t \in V$. So for (b) we can take $t' = \delta$; $t \in \{\delta\} \cdot_\delta V$ and $h_{\mathscr{A}}(\delta) = \alpha_\delta = a$. For case (a), the inductive hypothesis (I) assures the existence of terms t_i' with $t_i \in t_i' \cdot_\delta V$ and $h_{\mathscr{A}}(t') = a_i$. Thus we can take $t' = ft_i' \cdots t_n'$; $t \in t' \cdot_\delta V$ and $h_{\mathscr{A}}(t') = \alpha_f(h_{\mathscr{A}}(t_i'), \cdots, h_{\mathscr{A}}(t_n')) = \alpha_f(a_1, \cdots, a_n) = a$. We have established the inductive step for (I $\rightarrow$) and again will leave (I $\leftarrow$) to the reader. For (II), as in the basis case, we see that $b \in h_{\mathscr{R}}(t)$ if and only if there exist $b_i \in h_{\mathscr{R}}(t_i)$ and $\beta_f(b_1, \cdots, b_n) = b$. But by the induction hypothesis we know this is equivalent to $h_{\mathscr{B}}(t) = b$

We have established (I) and (II) by induction; by the arguments above relating (I) and (II) to our desired result, this completes the proof of Lemma 11.

LEMMA 12. (Closure of the recognizable sets under $^\delta$) *If $U \subseteq T_\Sigma$ is recognizable, then U^δ is recognizable.*

Proof. Let $\mathscr{A}$ be an automaton with $bh_{\mathscr{A}}(A_F) = U$. A non-deterministic automaton $\mathscr{R} = \langle A, \rho \rangle$ is constructed with transition relations:

(i) For $\lambda \in \Sigma_0$, $\rho_\lambda = \{\alpha_\lambda\}$.

(ii) For $f \in \Sigma_n$ and $a_i, x \in A$,

$$\rho_f(a_1, \cdots, a_n, x) \leftrightarrow \alpha_f(a_1, \cdots, a_n) = x \text{ or}$$

$$\alpha_f(a_1, \cdots, a_n) \in A_F \text{ and } x = \alpha_\delta.$$

The proof that this construction works, i.e., that $bh_{\mathscr{R}}(A_F) = U^\delta$, is similar to the proof of Lemma 11 and the details will be omitted.

Lemmas 10, 11 and 12, together with Theorem 2 of Section 2, give us the synthesis theorem, i.e., every regular set is recognizable. With the analysis theorem, these results complete the proof of the main statement of this section, the equivalence of recognizability and regularity (Theorem 9).

We will conclude this section on generalized regularity by introducing one more operation on sets of terms which preserves regularity: a kind of inverse to the δ-complex product.

$$U/_\delta V = \{t \mid (\{t\} \cdot_\delta V) \cap U \neq \varnothing\}.$$

This operation, sometimes called truncation or right division, can be described as an operation on U; the process is to take each term $t \in U$ and put into $U/_\delta V$ every term t' that can be obtained by simultaneously removing (replacing by δ) non-overlapping subterms of t which belong to V—with the restriction that, although subterms not containing δ may or may not be replaced, every occurrence of δ must occur in some subterm that is replaced.

LEMMA 13. (Closure of the recognizable sets under truncation) *If $U \subseteq T_\Sigma$ is recognizable and V is an arbitrary subset of T_Σ, then $U/_\delta V$ is recognizable.*

Proof. Assume that the automaton $\mathscr{A} = \langle A, \alpha \rangle$ recognizes U; $bh_{\mathscr{A}}(A_F) = U$. Let $A_V = \{h_{\mathscr{A}}(t) \mid t \in V\}$. We construct a non-deterministic automaton $\mathscr{R} = \langle A, \rho \rangle$ as follows:

(i) For $\lambda \in \Sigma_0$ and $\lambda \neq \delta$, $\rho_\lambda = \{\alpha_\lambda\}$ and $\rho_\delta = A_V$;

(ii) For $f \in \Sigma_n$, ρ_f is the graph of α_f.

Thus $\mathscr{R}$ is non-deterministic only in its multiplicity of initial states for the individual δ; otherwise, it is exactly like $\mathscr{A}$. We will prove that $bh_{\mathscr{R}}(A_F) = U/_\delta V$ by establishing the following two statements:

(I) $$t' \in t \cdot_\delta V \rightarrow h_{\mathscr{A}}(t') \in h_{\mathscr{R}}(t),$$

(II) $$a \in h_{\mathscr{R}}(t) \rightarrow \exists t'[h_{\mathscr{A}}(t') = a \text{ and } t' \in t \cdot_\delta V].$$

Statement (I) gives us the inclusion $U/_\delta V \subseteq bh_{\mathscr{R}}(A_F)$ as follows. If $t \in U/_\delta V$, then by definition of the truncation operation, there is some $t' \in U \cap t \cdot_\delta V$. Thus $h_{\mathscr{A}}(t') \in A_F$ and by (I), $h_{\mathscr{A}}(t') \in h_{\mathscr{R}}(t)$. Therefore, $h_{\mathscr{R}}(t) \cap A_F \neq \varnothing$ and $t \in bh_{\mathscr{R}}(A_F)$. From (II) we get the converse inclusion, $bh_{\mathscr{R}}(A_F) \subseteq U/_\delta V$: if $t \in bh_{\mathscr{R}}(A_F)$ then there is some $a \in A_F \cap h_{\mathscr{R}}(t)$. By (II) there exists a t' with $h_{\mathscr{A}}(t') = a \in A_F$ and $t' \in t \cdot_\delta V$. Thus, $t \cdot_\delta V \cap U \neq \varnothing$ and, by definition of $/_\delta$, $t \in U/_\delta V$. Now we continue with the proofs of (I) and (II), both by induction on the definition of terms.

Proof of (I). Induction is on t. If $t = \lambda \neq \delta$, then t' must also be λ and $h_{\mathscr{A}}(t') = \alpha_\lambda \in h_{\mathscr{R}}(t) = \{\alpha_\lambda\}$. If $t = \delta$ then t' can be any element of V and $h_{\mathscr{A}}(t') \in A_V = h_{\mathscr{R}}(t)$. Assuming that (I) is true for terms $t_1, \cdots, t_n$, consider the term $t = ft_1 \cdots t_n$. For $t' \in t \cdot_\delta V$ there must be terms t_i' $(i = 1, \cdots, n)$ with $t' = ft_1' \cdots t_n'$ and $t_i' \in t_i \cdot_\delta V$. By the induction hypothesis $h_{\mathscr{A}}(t_i') \in h_{\mathscr{R}}(t_i)$ and so by the definition of $h_{\mathscr{R}}(t') = \hat{\rho}_f(h_{\mathscr{R}}(t_1'), \cdots, h_{\mathscr{R}}(t_n'))$ we have $h_{\mathscr{A}}(t) = \alpha_f(h_{\mathscr{A}}(t_1), \cdots, h_{\mathscr{A}}(t_n)) \in h_{\mathscr{R}}(t')$.

Proof of (II). For $t = \lambda \neq \delta$, $a \in h_{\mathscr{R}}(t)$ means $a = \alpha_\lambda$, and taking $t' = \lambda$ we have $h_{\mathscr{A}}(t') = a$ and $t' \in t \cdot_\delta V$. If $t = \delta$ then $a \in h_{\mathscr{R}}(t)$ implies $h_{\mathscr{A}}(t') = a$ for some $t' \in V$. But also $t' \in t \cdot_\delta V$ so this takes care of the individual symbols in the inductive proof. Now let $t = ft_1 \cdots t_n$ and assume that property (II) holds for t_i $(i = 1, \cdots, n)$. $a \in h_{\mathscr{R}}(t)$ means that there exists $a_i \in h_{\mathscr{R}}(t_i)$ and $\rho_f(a_1, \cdots, a_n, a)$. By induction there exist t_i' with $h_{\mathscr{A}}(t_i') = a_i$ and $t_i' \in t_i \cdot_\delta V$. Clearly, $t' = ft_1' \cdots t_n'$ satisfies the conclusion of property (II).

4. An Application of Generalized Automata Theory to the Decision Problem of a Restricted Interpreted Second-Order Theory

We will use a standard alphabet $A_k = \{1, 2, \cdots, k\}$ for the set N_k of strings on k-symbols; N_k contains the empty string Λ. Theories of multiple successor arithmetics are based on the set N_k (for some k) together with the k (right-) successor functions, $r_1, \cdots, r_k$ where $r_\sigma(w) = w\sigma(\Lambda\sigma = \sigma)$ for all $w \in N_k$, and on relations and functions which can be defined recursively from the successor functions. Here we will be concerned with the weak monadic second-order theory of N_k together with the k successor functions. The associated language, $\mathscr{L}_k$, is an applied monadic second-order language consisting of the following:

(i) individual variables, $x, y, z, x_1, \cdots$, ranging over N_k;

(ii) set variables, $\alpha, \beta, \alpha_1, \cdots$, ranging over *finite* subsets of $N_k(p_\omega N_k)$;

(iii) constants $=, \in$ with their usual interpretation;

(iv) constant binary predicate symbols $R_\sigma(\sigma \in A_k)$ interpreted as the graphs of the k successor functions,

$$R_\sigma(u, v) \leftrightarrow r_\sigma(u) = v;$$

(v) propositional connectives, individual and set quantifiers, punctuation and parentheses as required for the applied language.

The formation rules for formulas in $\mathscr{L}_k$ are the usual; atomic formulas are those expressions of the form $x = y$, $x \in \alpha$, or $R_\sigma(x, y)$ and if F and G are formulas, then $F \wedge G$, $\neg F$, $\exists\, x\, F$ and $\exists\, \alpha F$ (x any individual variable, α any set variable) are all formulas. The sentences of $\mathscr{L}_k$ are those formulas in which there are no free variables.

The theory of interest—called the weak monadic second-order theory of multiple successors—consists of all those sentences in $\mathscr{L}_k$ which are true under the interpretation indicated. A formal definition of the language and its interpretation are omitted on the basis of the belief that the nature of the language should be clear from the description given and that further formality at this point would only complicate matters.

The expressive power of the elementary language of multiple successors (the language described above *without* set variables or quantifiers) is quite limited; the only definable sets are the definite sets and, for example, the prefix relation, $u \preceq v \leftrightarrow \exists w\; uw = v$, is not definable [18]. Adding the finite set variables and quantifiers increases the expressive power; J. B. Wright has shown that all regular subsets of N_k are definable and, to give an example of a definition, the following formula defines the prefix relation in $\mathscr{L}_2$.

$$\forall \alpha[y \in \alpha \wedge \forall z\, \forall w[z \in \alpha \wedge (R_1(w, z) \vee R_2(w, z)) \rightarrow w \in \alpha] \rightarrow x \in \alpha].$$

This formula, call it $F(x, y)$, defines the prefix relation in the sense that $F(u, v)$ is true if and only if u is a prefix of v for all $u, v \in N_2$. $F(x, y)$ requires that every finite set containing y and closed under predecessor (if $w\sigma \in \alpha$ then $w \in \alpha$) must also contain x.

The weak monadic second-order theory of one successor (the case $k = 1$) is decidable. The history of this result, as summarized by J. E. Doner [8], is as follows: a proof was first found by A. Ehrenfeucht, but was not published by him (see R. M. Robinson [16] and S. Feferman and R. L. Vaught [10]. Subsequent proofs were found independently by J. R. Büchi and C. C. Elgot [3] and published by them (J. R. Büchi [4], C. C. Elgot [9]).

The question of the decidability of the weak monadic second-order theory of multiple successors was first posed by J. R. Büchi [4]. J. E. Doner [8] obtained the positive result using concepts of generalized finite automata which were similar to those presented here and were developed independently. The method of applying these concepts parallels that developed by Büchi and Elgot for the one-successor case.

THEOREM 14. (Doner) *The weak monadic second-order theory of multiple successors is decidable.*

From this point on, we will speak only of the case $k = 2$; the generalization for more than two successor functions offers no difficulty. The procedure will be as follows.

(I) Find a language $\mathscr{L}_2'$ equivalent to $\mathscr{L}_2$ which involves no individual variables or quantifiers.

(II) Encode n-tuples of finite subsets of N_2 as terms in the species Σ^n where $\Sigma_0^n = \{\lambda\}$ and $\Sigma_2^n = \{\underline{0}, \underline{1}\}^n$; $\Sigma^n = \Sigma_0^n \cup \Sigma_2^n$.

(III) Show that every definable relation of $\mathscr{L}_2'$ is recognizable when interpreted under the encoding of (II).

In the process of accomplishing (I–III) we must convince the reader that there is an effective procedure for going from a formula in $\mathscr{L}_2$ to a formula in $\mathscr{L}_2'$, then to an automaton which recognizes the corresponding (under (II)) set of terms. Being so convinced, the decision procedure for $\mathscr{L}_2$ would be as follows: given a sentence S in $\mathscr{L}_2$, find the equivalent sentence S^* in $\mathscr{L}_2'$; S^* is either in the form (a) $\exists\, \alpha\, T(\alpha)$ or (b) $\forall\, \alpha\, T(\alpha)$. $T(\alpha)$ defines a set of finite subsets of N_2; find the automaton which recognizes the corresponding set of terms, say $bh_{\mathscr{A}}(A_F)$ is that set of terms. Then if S^* was in the form (a), S is true if and only if $bh_{\mathscr{A}}(A_F) \neq \varnothing$ and if it was in form (b), then S is true if and only if $bh_{\mathscr{A}}(A - A_F) = \varnothing$. Since the emptiness question can be effectively settled for generalized finite automata (Theorem 7), we have outlined an effective procedure for deciding truth of sentences of $\mathscr{L}_2$. This outline will be clarified as we continue now to describe the equivalent language $\mathscr{L}_2'$ and the encoding of n-tuples of finite sets.

The individual variables of $\mathscr{L}_2$ can be eliminated by simply replacing them with set variables which, in the translation from $\mathscr{L}_2$ to $\mathscr{L}_2'$, are restricted to be singleton sets. In the description of the language $\mathscr{L}_2'$ to follow, we introduce set variables in one-one correspondence with the individual variables; this is not necessary but facilitates the translation.

(i) Set variables $\alpha_x, \alpha_y, \alpha_z, \alpha_{x_1}, \cdots$ } ranging over finite subsets of N_2;
(ii) set variables $\alpha, \beta, \alpha_1, \cdots$ }
(iii) constant $\subseteq$ with its usual interpretation;
(iv) constant binary predicate symbols $\overline{R}_\sigma (\sigma \in A_k)$ interpreted as the graphs of the successor functions on singleton subsets: $\overline{R}_\sigma(\alpha, \beta) \leftrightarrow \alpha, \beta$ singletons and $\hat{r}_\sigma(\alpha) = \beta$ ($\hat{r}_\sigma$ is the set function induced by r_σ);
(v) propositional connectives, set quantifiers, etc.

The formation rules for $\mathscr{L}_2'$ are like those of $\mathscr{L}_2$ except that individual variables are not involved. Now we define a translation * from $\mathscr{L}_2$ to $\mathscr{L}_2'$ with the properties:

(A) If $\langle v, U\rangle$ satisfies $F(x, \alpha)$, then $\langle \{v\}, U\rangle$ satisfies $F^*(\alpha_x, \alpha)$;

(B) If $\langle V, U\rangle$ satisfies $F^*(\alpha_x, \alpha)$, then $V = \{v\}$ for some v and $\langle v, U\rangle$ satisfies $F(x, \alpha)$.

Of course, (A) and (B) should be stated in general for a formula

$F(x_1, \cdots, x_n, \alpha_1, \cdots, \alpha_m)$, but the point of the process can easily be seen without introducing this excess notation. We introduce the predicate $Sing(\alpha)$ which is the set of singleton subsets; $Sing$ is definable in $\mathscr{L}_2'$:

$$Sing(\alpha) \leftrightarrow_{\text{def}} \forall \beta[\beta \subseteq \alpha \rightarrow (\alpha \subseteq \beta \vee \forall \beta_1\, \beta \subseteq \beta_1)] \wedge \exists \beta_1 \neg \alpha \subseteq \beta_1.$$

Then the translation is as follows:
(1) For atomic formulas

$$(x = y)^* = \alpha_x \subseteq \alpha_y \wedge Sing(\alpha_x) \wedge Sing(\alpha_y)$$
$$(x \in \alpha)^* = \alpha_x \subseteq \alpha \wedge Sing(\alpha_x)$$
$$(R_\sigma(x, y))^* = \bar{R}_\sigma(\alpha_x, \alpha_y).$$

(2) * is extended to all formulas by

$$(F \wedge G)^* = F^* \wedge G^*$$
$$(\neg F)^* = \neg F^*$$
$$(\exists\, x\, F)^* = \exists\, \alpha_x[Sing(\alpha_x) \wedge F^*]$$
$$(\exists\, \alpha\, F)^* = \exists\, \alpha\, F^*.$$

Because the translation is so simple, we will omit the details of the proof that it works in the sense that properties (A) and (B) hold. It should also be clear that if S is any sentence in $\mathscr{L}_2$ then S is true if and only if S^* is true.

The encoding of finite subsets of N_2 is based on the close correspondence between the structure of any term in a dyadic species (where non-individual symbols have rank 2) and a finite prefix closed subset of N_2. Indeed, instead of defining terms as strings (thus, functions from finite segments of N into Σ) we might have chosen to define terms as functions from finite segments (finite prefix closed subsets) of N_2 into Σ. By example, the term t shown graphically in Figure 1 corresponds to the function from P into Σ defined by the corresponding position of the graphs, e.g., $t(\Lambda) = f$, $t(11) = \lambda$, $t(21) = g$.

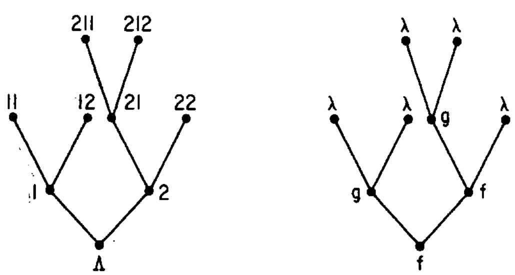

Figure 1

Now if we turn this correspondence around and look at the inverse image of some function symbol, we are able to associate a finite subset of N_2 with a term; in effect (the notation will be refined below) $t^{-1}(g) = \{1, 21\}$ (cf. Figure 1) and $t^{-1}(f) = \{\Lambda, 2\}$.

Switching now to the species with which we will be working, a term in T_{Σ^1} can be viewed, in the sense described above, as a characteristic function of a finite subset of N_2; the set associated with such a term t is, in the intuitive notation used above, $t^{-1}(\underline{1})$. Instead of going through the formal process of identifying terms with functions on closed subsets of N_2, we will give a straightforward inductive definition of the map c which assigns to each $t \in T_{\Sigma^1}$ a finite subset $c(t) \subseteq N_2$.

(i) $$c(\lambda) = \varnothing;$$

(ii) $$c(\underline{0}t_1t_2) = \hat{l}_1c(t_1) \cup \hat{l}_2c(t_2)$$

$$c(\underline{1}t_1t_2) = \hat{l}_1c(t_1) \cup \hat{l}_2c(t_2) \cup \{\Lambda\}$$

(l_σ is the left successor by the symbol σ; $l_\sigma(w) = \sigma w$).

The function c is onto $p_\omega N_2$. This would be proved inductively with the hypothesis that if α is of length $\leqslant n$ (length of α is the length of the longest member) then α is in the range of c. Then any set of length $n + 1$ can be decomposed in the form $\alpha = \hat{l}_1\alpha_1 \cup \hat{l}_2\alpha_2$ or $\alpha = \hat{l}_1\alpha_1 \cup \hat{l}_2\alpha_2 \cup \{\Lambda\}$ where the length $\alpha_1 \leqslant n$ and the same for α_2. The inductive hypothesis assures the existence of t_1, t_2 with $c(t_i) = \alpha_i$ and under the two alternatives above $\alpha = c(\underline{0}t_1t_2)$ or $\alpha = c(\underline{1}t_1t_2)$ respectively.

c is certainly not one-one as the two examples in Figure 2 indicate. It should be clear that if $c(t) = \alpha$ and $t' \in \{t\} \cdot_\lambda \{\underline{0}\lambda\lambda\}^\lambda$, then $c(t') = \alpha$ as well. Put another way, $c(t) = c(t')$ if and only if they differ only by subterms in the function symbol $\underline{0}$.

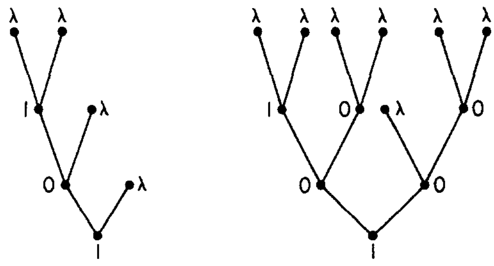

Figure 2

Thus, to get at an encoding of finite sets, we must choose a representative of $c^{-1}(\alpha)$, call it $e(\alpha)$ (for encoding). The natural choice is the smallest term in $c^{-1}(\alpha)$ and besides being intuitively clear, we can formalize "smallest" in several equivalent ways. As indicated above, terms can be treated as functions on prefix closed subsets of N_2. Under this interpretation, $e(\alpha) = \bigcap c^{-1}(\alpha)$. Alternatively, we can define the partial order relation $\precsim$ generalizing the prefix relation for strings by

$$t_1 \precsim t_2 \leftrightarrow t_2 \in t_1 \cdot_\lambda T_\Sigma.$$

Then $e(\alpha)$ is the greatest lower bound of $c^{-1}(\alpha)$ with respect to this generalized prefix relation:

$$e(\alpha) = t_1 \leftrightarrow c(t_1) = \alpha \wedge \forall t_2[c(t_2) = \alpha \rightarrow t_1 \preceq t_2].$$

Still another description of $e(\alpha)$ is that term in $c^{-1}(\alpha)$ in which $\underline{0}\,\lambda\lambda$ is *not* a subterm. From Figure 2 we see that $e\{\Lambda, \underline{1}\underline{1}\} = t_1$ and two further examples of the encoding e are given in Figure 3.

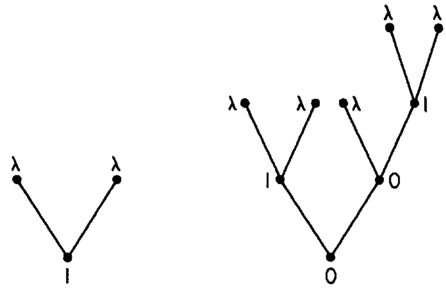

Figure 3

We have labored over describing the encoding of finite sets in the hope that the reader will obtain a reasonably good feeling for the technique. This is necessary because we will rely on this understanding to justify several of the claims to be made in the sequel, omitting detailed and cumbersome proofs where possible.

Before proceeding to extend the mapping e to an encoding of n-tuples of finite sets, the following lemma, proved in detail, is exemplary of the type of result we need.

LEMMA 15. *The set of encodings of finite subsets of* N_2, $U_1 = \hat{e}(p_\omega N_2)$, *is recognizable.*

Proof. We will actually prove that U_1 is regular, so that, by Theorem 3, U_1 is recognizable. To this end, define

$$U_1'' = \{\underline{0}\lambda\delta,\ \underline{0}\delta\lambda,\ \underline{0}\delta\delta,\ \underline{1}\lambda\delta,\ \underline{1}\delta\lambda,\ \underline{1}\lambda\lambda,\ \underline{1}\delta\delta\}$$

$$U_1' = (U_1'')^{\delta}.$$

Now we claim

$$U_1 = U_1' \cdot_\delta \varnothing \cup \{\lambda\}.$$

As was observed when the complex product, $\cdot_\delta$, was introduced, $U_1' \cdot_\delta \varnothing$ is simply that set of terms in U_1' which do not contain the individual δ. Thus if $t \in U_1' \cdot_\delta \varnothing$, then $t \in U_1'$ and δ does not occur in t. But by the definition of U_1', the atomic subterms of t must all be $\underline{1}\lambda\lambda$; $\underline{0}$ is not the function symbol of any atomic subterm of t so $t \in U_1$ (cf. the third alternative description of $e(\alpha)$). For the converse, we will prove the following statement by induction: If $t \in U_1$ with $1 \leqslant$ depth $(t) \leqslant n$, then $t \in X^n$, where $X^0 = \{\delta\}, X^{n+1} = X^n \cup U_1'' \cdot_\delta X^n$. Since $U_1' = \cup X^n$, this immediately gives us $U_1 \subseteq U_1'$ (except for λ), but because δ does not occur in any term of U_1, we also know that $U_1 \subseteq U_1' \cdot_\delta \varnothing \cup \{\lambda\}$. For $n = 1$, $t = \underline{1}\lambda\lambda$ is the only possible term and $\underline{1}\lambda\lambda \in X^1 = \{\delta\} \cup U_1''$. If $t \in U_1$ is of depth $n+1$ then t must be in one of the following forms: $t = t_1t_2$, $t = f\lambda t_2$, or $t = ft_1\lambda$, where $f = \underline{0}$ or $f = \underline{1}$, $1 \leqslant$ depth $(t_i) \leqslant n$ and $t_i \in U_1$. By the induction hypothesis, $t_i \in X^n$ and with $X^{n+1} = U_1'' \cdot_\delta X^n \cup X^n$ it follows that $t \in X^{n+1}$.

Continuing now with the encoding of $(p_\omega N_2)^n$, we will be working with the species $\Sigma^n = \{\lambda\} \cup \{\underline{0}, \underline{1}\}^n$. Define the map π_i: $\{\underline{0}, \underline{1}\}^n \rightarrow \{\underline{0}, \underline{1}\}$ $(i = 1, \cdots, n)$ by $\pi_i\langle\delta_1, \cdots, \delta_n\rangle = \delta_i$. Then, as in Section 2, we extend π_i to a projection $\bar{\pi}_i$: $T_{\Sigma^n} \rightarrow T_{\Sigma^1}$. Following the development of the encoding of finite sets, we first define a map c_n from T_{Σ^n} onto $(p_\omega N_2)^n$ by

$$c_nt = \langle c\bar{\pi}_1t, \cdots, c\bar{\pi}_nt\rangle.$$

A graphical example is given in Figure 4.

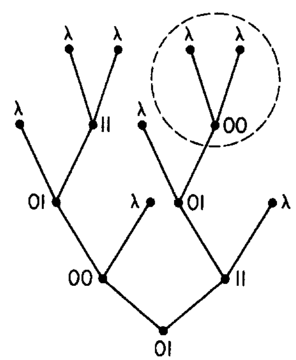

Figure 4

Again, the encoding for an n-tuple $\bar{\alpha} = \langle\alpha_1, \cdots, \alpha_n\rangle$ is chosen to be the smallest term in $c_n^{-1}(\bar{\alpha})$:

$$e(\bar{\alpha}) = t \leftrightarrow c_n(t) = \bar{\alpha} \wedge \forall t'[c_n(t') = \bar{\alpha} \rightarrow t \preceq t'].$$

The alternative characterization of $e(\alpha)$ is again straightforward: $e(\bar{\alpha})$ is that term in $c_n^{-1}(\bar{\alpha})$ in which $0 \cdots 0\lambda\lambda$ is *not* a subterm. In Figure 4, for example, if the circled subterm $00\lambda\lambda$ is replaced by λ, then the resulting term is $e\langle 1, 2, 112\}, \{\Lambda, 2, 11, 112, 21\}\rangle$. The generalization of Lemma 15 goes through with an almost identical proof.

LEMMA 16. *The set of encodings of n-tuples of finite subsets of N_2, $U_n = \hat{e}(p_\omega N_2)^n$, is recognizable.*

Now we are in a position to define what we mean for a relation on finite subsets of $p_\omega N_2$ to be recognizable: $R \subseteq (p_\omega N_2)^n$ is *recognizable* if and only if $\hat{e}R$ is recognizable. And the theorem we need to obtain the decision procedure outlined in the beginning of this section is

THEOREM 17. *If R is a relation definable in $\mathcal{L}_2'$, then R is recognizable.*

As is to be expected, we will prove Theorem 17 in two parts, showing first that the relations defined by atomic formulas are recognizable and then showing that the recognizable sets are closed under the propositional operations ($\wedge, \neg$) and quantification ($\exists\alpha$).

LEMMA 18. *The relations on $p_\omega N_2$ defined by atomic formulas in $\mathscr{L}_2'$ are recognizable.*

Proof. We must show that $\subseteq$ and $\bar{R}_\sigma (\sigma \in \Sigma_2)$ are recognizable. Consider the following regular sets:

(A) $$S_\subseteq' = \{\underline{11}\lambda\lambda,\ \underline{01}\lambda\lambda,\ \underline{00}\lambda\lambda\}^\lambda,\ S_\subseteq = S_\subseteq' \cap U_2$$

(B) $$V = \{\underline{00}\lambda\delta,\ \underline{00}\delta\lambda\}^\delta,\ S_1 = V \cdot_\delta \{\underline{10}\ \underline{01}\lambda\lambda\lambda\},\ S_2 = V \cdot_\delta \{\underline{10}\lambda\ \underline{01}\lambda\lambda\}.$$

We claim that $\hat{e} \subseteq = S_\subseteq$ and $\hat{e}\bar{R}_\sigma = S_\sigma$. This gives the required result because $S_\subseteq'$, S_1 and S_2 are all regular and thus Theorem 8 recognizable; $S_\subseteq$ is also recognizable (Lemma 16, Theorem 2).

(A) From an informal point of view, what exactly is $\hat{e} \subseteq$? It should not be hard for the reader to convince himself that $\hat{e} \subseteq$ is precisely the set of terms in U_2 which do *not* contain the function symbol $\underline{10}$. But $S_\subseteq'$ is the set of all terms not involving $\underline{10}$ so $S_\subseteq = S_\subseteq' \cap U_2 = \hat{e} \subseteq$.

(B) We will first prove $\hat{c}_2 S_1 \subseteq \bar{R}_1$ by induction on the construction of S_1. It is easy to verify (the complex operations are completely distributive) that

$$S_1 = \bigcup (Y_n \cdot_\delta \{\underline{10}\ \underline{01}\lambda\lambda\lambda\})$$

when $Y_0 = \{\delta\}$ and $Y_{n+1} = \{\underline{00}\lambda\delta,\ \underline{00}\delta\lambda\} \cdot_\delta Y_n$. For $n = 0$, $t \in Y_0 \cdot_\delta \{\underline{10}\ \underline{01}\lambda\lambda\lambda\}$ means $t = \underline{10}\ \underline{01}\lambda\lambda\lambda$ and $c_2 t = \langle\{\Lambda\}, \{1\}\rangle \in \bar{R}_1$. Now if $t \in Y_{n+1} \cdot_\delta \{\underline{10}\ \underline{01}\lambda\lambda\lambda\}$, then

$$t \in \{\underline{00}\lambda\delta,\ \underline{00}\delta\lambda\} \cdot_\delta (Y_n \cdot_\delta \{\underline{10}\ \underline{01}\lambda\lambda\lambda)$$

(note the complex operations are associative, $(U \cdot_\delta V) \cdot_\delta T = U \cdot_\delta (V \cdot_\delta T)$, but for example $\{t\} = (\{\delta\} \cdot_\lambda \{\lambda\}) \cdot_\delta \{t\} \neq \{\delta\} \cdot_\lambda (\{\lambda\} \cdot_\delta \{t\}) = \{\delta\}$). Hence, there exists a $t' \in Y_n \cdot_\delta \{\underline{10}\ \underline{01}\lambda\lambda\lambda\}$ with $t = \underline{00}\lambda t'$ or $t = \underline{00}t'\lambda$ and by induction $c_2 t' = \langle\{w\}, \{w1\}\rangle \in \bar{R}_1$. But by the definition of c_2, $c_2 t = \langle\{2w\}, \{2w1\}\rangle \in \bar{R}_1$ or $c_2 t = \langle\{1w\}, \{1w1\}\rangle \in \bar{R}_1$ in the two cases respectively. By induction we obtain $\hat{c}_2 S_1 \subseteq \bar{R}$. It is easy to see that $S_1 \subset U_1$, so that we have shown that $S_1 \subseteq \hat{e}\bar{R}_1$. The converse, which is notationally just as bad, is left to the reader.

The last property we need is the closure of the recognizable relations under the logical operations. Our argument follows very closely Ritchie's proof [15] for regular relations. We will exhibit the constructions and omit the details.

LEMMA 19. (Closure of the recognizable relations under logical operations) *If $R(\alpha_1, \cdots, \alpha_m, \beta_1, \cdots, \beta_n)$ and $S(\beta_1, \cdots, \beta_n, \gamma_1, \cdots, \gamma_p)$ are recognizable relations, then*

(1) *$R(\bar{\alpha}, \bar{\beta}) \wedge S(\bar{\beta}, \bar{\gamma})$ is recognizable;*

(2) *$\neg R(\bar{\alpha}, \bar{\beta})$ is recognizable;*

(3) *$\exists\, \alpha_1\, R(\alpha_1, \alpha_2, \cdots, \alpha_m, \bar{\beta})$ is recognizable.*

Proof. Without loss of generality, we will consider the case $m = n = p = 1$. The extension to other values of the indices should be clear.

(1) First we observe that the relations $R'(\alpha, \beta, \gamma) \leftrightarrow R(\alpha, \beta)$ and $S'(\alpha, \beta, \gamma) \leftrightarrow S(\beta, \gamma)$ are recognizable. Consider the map $\pi\colon \Sigma^3 \to \Sigma^2$ with $\pi\langle\sigma_1, \sigma_2, \sigma_3\rangle = \langle\sigma_1, \sigma_2\rangle$. Then it should be clear that $\bar{\pi}^{-1}\hat{e}R \cap U_3 = \hat{e}R'$. Now by assumption $\hat{e}R$ is recognizable and Theorems 4 and 2 give us the result that $\hat{e}R'$ is recognizable; thus R' is recognizable. Similarly, S' is recognizable. Now $\hat{e}(R \wedge S)$ is simply $\hat{e}R' \cap \hat{e}S'$ and thus $R \wedge S$ is recognizable.

(2) $T_{\Sigma^2} - \hat{e}R$ is recognizable by assumption and by Theorem 2. But this is not quite $\hat{e}(\neg R)$ because it contains terms not in the range of e. This is simply remedied; $\hat{e}(\neg R) = (T_{\Sigma^2} - \hat{e}R) \cap U_2$, and $\neg R$ is recognizable.

(3) Given that $R(\alpha, \beta)$ is recognizable. We again come close to the required result by considering the map $\pi\colon \Sigma^2 \to \Sigma^1$ defined by $\pi\langle\sigma_1, \sigma_2\rangle = \sigma_2$. Then, $U' = \bar{\pi}\hat{e}R$ is almost $\hat{e}(\exists\, \alpha\, R)$ except we are not in the range of $\hat{e}$. In effect, the terms of U' must be trimmed down, removing the subterms in 0. This is accomplished with the truncation operation of Section 3.

$$\hat{e}(\exists\ \alpha\ R) = (\bar{\pi}\ \hat{e}\ R/_{\lambda}\{0\lambda\lambda\}^{\lambda}) \cap U_1.$$

Lemmas 18 and 19 complete the proof of Theorem 17. Except in outline in the beginning of this section, we have mentioned nothing about effectiveness. There we stated that the reader must be convinced that one could effectively construct an automaton which recognized the set of terms corresponding to any given formula. The process of construction is indeed effective as could easily be checked because each of the closure lemmas of Section 2 describe procedures for constructing resultant automata from given automata.

REFERENCES

[1] G. BIRKHOFF, On the structure of abstract algebras. *Proc. Cambridge Phil. Soc.* **31** (1938), 433–454.

[2] G. BIRKHOFF, *Lattice Theory*, Amer. Math. Soc. Colloq. Publ., Vol. 25, New York, 1948.

[3] J. R. BÜCHI and C. C. ELGOT, Decision problems of weak second-order arithmetics and finite automata. Abstract 553-112, *Notices Amer. Math. Soc.* **5** (1958), 834.

[4] J. R. BÜCHI, "Weak second-order arithmetic and finite automata". University of Michigan, Logic of Computers Group Technical Report, September 1959; *Z. Math. Logik Grundlagen Math.* **6** (1960), 66–92.

[5] J. R. BÜCHI and J. B. WRIGHT, "Mathematical Theory of Automata". Notes on material presented by J. R. Büchi and J. B. Wright, Communication Sciences 403, Fall 1960, The University of Michigan.

[6] J. R. BÜCHI, Mathematische Theorie des Verhaltens endlicher Automaten. *Z. Angewandte Math. und Mech.* **42** (1962), 9–16.

[7] I. M. COPI, C. C. ELGOT and J. B. WRIGHT, Realization of events by logical nets. *J. Assoc. Comp. Mach.* **5** (1958) 181–196. (Also in Moore [13].)

[8] J. E. DONER, Decidability of the weak second-order theory of two successors. Abstract 65T-468, *Notices Amer. Math. Soc.* **12** (1965), 819; erratum, *ibid.* **13** (1966), 513.

[9] C. C. ELGOT, "Decision problems of finite automaton design and related arithmetics". University of Michigan, Department of Mathematics and Logic of Computers Group Technical Report, June 1959; *Trans. Amer. Math. Soc.* **98** (1961), 21–51.

[10] S. FEFERMAN and R. L. VAUGHT, The first-order properties of products of algebraic systems. *Fund. Math.* **47** (1959), 57–103.

[11] S. C. KLEENE, Representation of events in nerve nets and finite automata. *Automata Studies*, pp. 3–42, Annals of Math. Studies, No. 34, Princeton University Press, Princeton, N. J., 1956.

[12] I. T. MEDVEDEV, On a class of events representable in a finite automaton. Supplement to the Russian translation of *Automata Studies* (C. Shannon and J. McCarthy, eds.), published by the Publishing Agency for Foreign Literature, Moscow, 1956. Translated by Jacques J. Schorr-Kon for Lincoln Laboratory Report 34–73, June 1958. (Also in Moore [13].)

[13] E. F. MOORE, *Sequential Machines, Selected Papers.* Addison-Wesley, Reading, Mass., 1964.

[14] M. O. RABIN and DANA SCOTT, Finite automata and their decision problems. *IBM J. Res. Develop.* **3** (1959), 114–125. (Also in Moore [13].)

[15] R. W. RITCHIE, Classes of predictably computable functions. *Trans. Amer. Math. Soc.* **106** (1963), 139–173.

[16] RAPHAEL M. ROBINSON, Restricted set-theoretic definitions in arithmetic. *Proc. Amer. Math. Soc.* **9** (1958), 238–242.

[17] J. W. THATCHER, "Notes on Mathematical Automata Theory". University of Michigan Technical Note, December, 1963.

[18] J. W. THATCHER, "Decision Problems and Definability for Generalized Arithmetic". Doctoral Dissertation, The University of Michigan; IBM Research Report RC 1316, November, 1964.

[19] J. W. THATCHER and J. B. WRIGHT, Generalized finite automata. Abstract 65T-469, *Notices Amer. Math. Soc.* **12** (1965), 820.

(Received 1 August 1967)