Finitely Presented Algebras and the Polynomial Time Hierarchy*

Dexter Kozen

TR 77-303

March 1977

Department of Computer Science
Cornell University
Ithaca, New York 14853

*This research has been supported in part by National Science Foundation grant DCR75-09433.

Finitely Presented Algebras and the Polynomial Time Hierarchy*

Dexter Kozen
Cornell University

Abstract

Let $S_n(V_n) =$

$$\{\langle \Gamma,\ Q_1 v_1 \ldots Q_k v_k\ s \equiv j \rangle \mid$$

Γ is a finite presentation of $\mathcal{A}$, $Q_1 \ldots Q_k$ is a string of quantifiers with n alternations, the outermost an $\exists(\forall)$, $\mathcal{A} \models Q_1 v_1 \ldots Q_k v_k\ s \equiv t\}$.

It is shown that $S_n(V_n)$ is complete for $\Sigma_n^P(\Pi_n^P)$, and $\bigcup_{n=0}^{\infty} S_n \cup V_n$ is complete for PSPACE, answering a question of [1] and generalizing a result of Stockmeyer and Meyer [2].

---

*This research has been supported in part by National Science Foundation grant DCR75-09433.

0. Introduction

In [1] we discussed the importance of finitely presented algebras with respect to finite tree automata, context free derivations, and logic. We showed that several decision problems previously known to be complete for various complexity classes were in fact special cases of natural decision problems about finitely presented algebras. Here we continue on this track.

Stockmeyer and Meyer [2] proved that deciding truth of closed quantified Boolean expressions with n alternations of quantifiers, $n \geq 1$, the outermost a $\exists(\forall)$, is a problem complete for $\Sigma_n^p(\Pi_n^p)$, the n^{th} $\Sigma(\Pi)$ level of the polynomial time hierarchy. There is, however, a pathological twist to their result, namely that it fails for n=0, since Boolean trees can be evaluated in deterministic logspace. This suggests that sets of quantified Boolean expressions are perhaps not the proper canonical complete sets for Σ_n^p and Π_n^p.

In this paper we answer a question left open in [1] and generalize the result of Meyer and Stockmeyer mentioned above. Let

$S_n(V_n) = \{\langle \Gamma, Q_1 v_1 \ldots Q_k v_k \; s \equiv t \rangle \mid$
Γ is a finite presentation of $\mathscr{A}$, $Q_1 \ldots Q_k$ is a string of quantifiers with n alternations, the outermost an $\exists(\forall)$, the sentence $Q_1 v_1 \ldots Q_k v_k \; s \equiv t$ is true in $\mathscr{A}\}$.

Sets of quantified Boolean expressions are special cases of these sets, since Γ can be taken to present the two-element Boolean algebra. We show that $S_n (V_n)$ is $\le^m_{log}$-complete for $\Sigma^P_n (\Pi^P_n)$, and $\bigcup_{n=0}^{\infty} S_n \cup V_n$ is $\le^m_{log}$-complete for PSPACE. The pathology noted above disappears, since as shown in [1], $S_0 (=V_0)$ is $\le^m_{log}$-complete for P.

1. Preliminaries

We will review briefly the basic definitions. The reader should consult [1] for a more extensive treatment.

Definition 1.0

An algebra $\mathscr{A}$ is finitely presented if there is a finite set G of generator symbols, a finite set O of operator symbols of various finite arities, and a finite set Γ of axioms or defining relations of the form $x \equiv y$, where x and y are terms over G and O, such that $\mathscr{A}$ is isomorphic to the free algebra (algebra of terms) over G and O modulo the congruence induced by Γ. The triple $\langle G, O, \Gamma \rangle$ is a presentation of $\mathscr{A}$. ■

Example 1.1

The two-element Boolean algebra is presented by

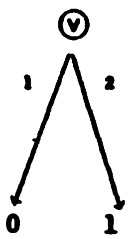
$$G = \{0, 1\}$$

$$O = \{\wedge, \vee, \neg\}$$

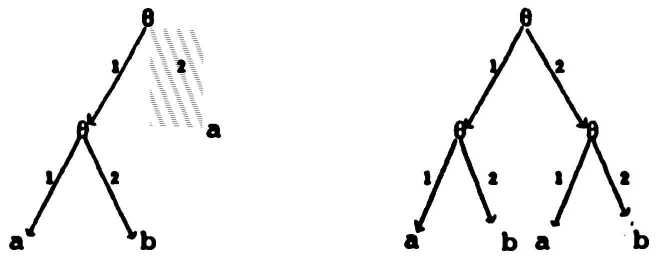
$$\Gamma = \{0\vee 0 \equiv 0,\ 0\vee 1 \equiv 1,\ 1\vee 0 \equiv 1,\ 1\vee 1 \equiv 1,\ 0\wedge 0 \equiv 0,\ 0\wedge 1 \equiv 0,\ 1\wedge 0 \equiv 0,\ 1\wedge 1 \equiv 1,\ \neg 1 \equiv 0,\ \neg 0 \equiv 1\}.$$

■

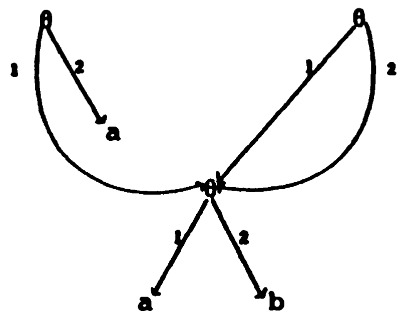
For computational purposes we shall consider terms to be represented by labeled dags, e.g. the term $0 \vee 1$ would be represented by the tree



We also allow a more efficient representation by "factoring out" common subterms, e.g. terms with tree representations



become, after factoring, the dag



A presentation will be given by a labeled dag representing the terms appearing in G and Γ, with an extra undirected edge set connecting the roots of x and y, where $x \equiv y$ is an axiom of Γ. Let $\langle G,O,\Gamma\rangle$ be a presentation. We will reuse the symbol Γ to stand for the presentation $\langle G,O,\Gamma\rangle$. We will denote the set of terms by τ, and write $x \equiv y$ when the congruence of terms x and y follows from the axioms of Γ. We will denote by $\tau/_{\Gamma}$ the algebra presented by Γ.

<u>Definition 1.2</u>

The <u>word problem</u> is the set

$WP = \{\langle\Gamma, x, y\rangle \mid x \equiv y\}$.

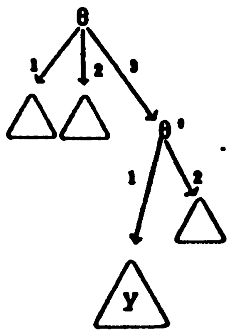
The <u>finiteness problem</u> is the set

$FIN = \{\Gamma \mid \Gamma$ presents a finite algebra$\}$. ■

As shown in [1], both WP and FIN are $\le^{m}_{\log}$-complete for P.

<u>Definition 1.3</u>

Let $D = \{\theta k \mid \theta \in O,\ 1 \le k \le \text{arity of } \theta\}$. Let $\alpha, \beta, \ldots$ represent strings in D^*. λ will represent the null string. Strings in D^* will be used to specify the position of a subterm in a term, according to the following example: if x is the term



then y occurs as a subterm of x at position $\theta 3 \theta' 1$. We will write $x \alpha y$ if y occurs as a subterm of x at position α. ■

<u>Definition 1.4</u>

$x[\alpha\backslash y]$ is the term x with the subterm at position α (if it exists) replaced by y.

$x[w\backslash y]$ is the term x with all occurrences of the subterm w in x replaced by y. ∎

<u>Definition 1.5</u>

$x \to y$ if there is an axiom $z \equiv w$ of Γ and α such that $x\alpha z$ and $y = x[\alpha\backslash w]$.

$\overset{*}{\to}$ is the reflexive transitive closure of $\to$. ∎

The relation $\overset{*}{\to}$ constitutes a convenient proof system for congruence of terms:

<u>Lemma 1.6</u>

$x \overset{*}{\to} y$ iff $x \equiv y$.

<u>Proof</u>

Easy induction on definition of $\equiv$. ∎

The following sets are of fundamental importance.

<u>Definition 1.7</u>

$R = \{$subterms of terms appearing in $\Gamma\}$.

$r = \{[x] \mid [x]$ is the congruence class of $x, x \in R\}$.

$\hat{R} = \{y \mid [y] \in r\} = \{y \mid \exists x \in R \; x \equiv y\}$. ∎

<u>Definition 1.8</u>

$x\underline{\alpha}z$ iff $\exists y \; x\alpha y$ and $y \equiv z$. ∎

The following elementary properties are easy to prove, using the techniques of [1].

<u>Lemma 1.9</u>

(i) if $s \equiv t$, $s\underline{\alpha}y$, $t\underline{\alpha\beta}z$, & $z \nmid \hat{R}$, then $y \nmid \hat{R}$ & $y\underline{\beta}z$.

(ii) if $x\underline{\alpha}y$, $y\underline{\beta}z$, & $z \nmid \hat{R}$, then $x\underline{\alpha\beta}z$.

(iii) if $x\underline{\alpha\beta}z$, $x\underline{\alpha}y$, & $z \nmid \hat{R}$, then $y \nmid \hat{R}$ & $y\underline{\beta}z$.

(iv) if $x\underline{\beta}z$ & $z \nmid \hat{R}$ then $x \nmid \hat{R}$.

(v) if $x \equiv y$, $x\underline{\alpha}z$, & $z \nmid \hat{R}$, then $y\underline{\alpha}z$.

(vi) if $s \equiv t$, $s\underline{\alpha}y$, $t\underline{\alpha}z$, & $z \nmid \hat{R}$, then $y \equiv z$.

(vii) if $s \equiv t$, $s\underline{\alpha}y$, & $y \nmid \hat{R}$, then $s[\alpha \backslash z] \equiv t[\alpha \backslash z]$ for any z.

(viii) if $y\underline{\alpha}z$, $y \equiv z$, and $y \nmid \hat{R}$, then $\alpha = \lambda$.

(ix) if $\theta x_1 \ldots x_m \equiv \theta y_1 \ldots y_m$ and $\theta x_1 \ldots x_m \nmid \hat{R}$, then $x_i \equiv y_i$, $1 \le i \le m$.

<u>Proof</u>

We will prove (i) as an illustration of the techniques; the rest are equally straightforward and are left as exercises.

Assume $s \equiv t$, $s\underline{\alpha}y$, $t\underline{\alpha\beta}z$, & $z \nmid \hat{R}$.

In a proof $t = t_1 \rightarrow t_2 \rightarrow \ldots \rightarrow t_n = s$ of $t \equiv s$, if there is a minimal i such that $t_i \rightarrow t_{i+1}$, $t_i \gamma w$, $t_{i+1} = t_i[\gamma \backslash x]$ where $w \equiv x$ is an axiom, and γ is an initial substring of $\alpha\beta$, then $t_i\underline{\alpha\beta}z$ and z is congruent to a subterm of w. But this subterm of w is in R, contradicting the assumption $z \nmid \hat{R}$. Thus no such i can exist. This says that $s\underline{\alpha\beta}z$. But $s\underline{\alpha}y$, so $\exists w$ $s\alpha w$ and $w \equiv y$, but then $w\underline{\beta}z$, and as above, $y\underline{\beta}z$.

To show $y \notin \hat{R}$, note $y\underline{\beta}z$ implies $y\beta x$ & $x \equiv z$ for some x; but if $y \in \hat{R}$ then $x \in \hat{R}$, contradicting the assumption that $z \notin \hat{R}$. ∎

<u>Definition 1.10</u>

Let Γ be given. Let $V = \{v_1, v_2, \ldots\}$ be a set of <u>variables</u>. Let $G^+ = G \cup V$, $\tau^+ = \{\text{terms over } G^+ \text{ and } 0\}$. Thus τ^+ is the set of terms with occurrences of variables. A <u>schema</u> is a formula $s \equiv t$, $s, t \in \tau^+$. A <u>closed</u> <u>formula</u> is one of the form

$$Q_1 v_1 \ldots Q_k v_k \; s \equiv t$$

where each Q_i is either $\exists$ or $\forall$, and all variables occurring in s and t are among $v_1, \ldots, v_k$. ∎

<u>Definition 1.11</u>

A closed formula ϕ is <u>true</u> in ${}^{\tau}/_{\Gamma}$ (denoted ${}^{\tau}/_{\Gamma} \models \phi$) if either

(i) ϕ is of the form $s \equiv t$ (note in this case $s, t \in \tau$ since ϕ is closed) and s is congruent to t.

(ii) ϕ is of the form $\forall v \psi$ and for all $x \in \tau$, ${}^{\tau}/_{\Gamma} \models \psi[v \backslash x]$.

(iii) ϕ is of the form $\exists v \psi$ and there is an $x \in \tau$ such that ${}^{\tau}/_{\Gamma} \models \psi[v \backslash x]$. ∎

Let $v_1, \ldots, v_k$ be the variables occurring in s and t. We will write $\bar{x}$ for a k-tuple of terms in τ, and $s(\bar{x})$ for $s[v_1 \backslash x_1] \ldots [v_k \backslash x_k]$. By the definition of truth above,

$$\tau/_{\Gamma} \models Q_1 v_1 \ldots Q_k v_k \; s \equiv t \qquad (*)$$

iff

$$Q_1 x_1 \ldots Q_k x_k \; s(\bar{x}) \equiv t(\bar{x}). \qquad (**)$$

Note that (*) is an assertion about the truth of a sentence in some language interpreted over the domain $\tau/_{\Gamma}$ (here the Q_i are symbols in the language), while (**) is a metastatement about congruence of terms in τ (here the Q_i represent the English "for all" and "there is" in Definition 1.11 (ii) and (iii)). This is a subtle distinction which should be recognized, but which we will find convenient to ignore in general.

2. Main Results

Definition 2.0

Let $S_n(V_n) = \{\langle\Gamma, Q_1v_1\ldots Q_kv_k\ s\equiv t\rangle \mid$
$Q_1\ldots Q_k$ is a string of quantifiers with n alternations, the outermost a $\exists(\forall)$, and
$\tau/_\Gamma \models Q_1v_1\ldots Q_kv_k\ s\equiv t\}$. ∎

In the following, let Γ and $Q_1v_1\ldots Q_kv_k\ s\equiv t$ be fixed.

Definition 2.1

$R^+ = \{$subterms of s and t$\}$.

$R_1 = R\cup(R^+ \cap \tau)$;

$r_1 = \{[x] \mid x\in R_1\}$;

$\hat{R}_1 = \{y \mid [y]\in r_1\} = \{y \mid \exists x\in R_1\ x\equiv y\}$. ∎

We wish to develop a means of describing the syntactic interdependence of terms in R^+.

Definition 2.2

Let $\sim$ be defined on $(R^+)^2$ as the smallest equivalence relation satisfying

(i) $s\sim t$

(ii) $$\frac{\theta x_1\ldots x_n \sim \theta y_1\ldots y_n}{x_1\sim y_1,\ldots,x_n\sim y_n}.$$ ∎

Note that (i) and (ii) plus the axioms

(iii) $x \sim x$

(iv) $\dfrac{x \sim y}{y \sim x}$

(v) $\dfrac{x \sim y,\ y \sim z}{x \sim z}$

constitute a complete proof system for $\sim$.

The purpose of $\sim$ is to determine which subterms of s & t are forced by syntax to be congruent under most interpretations of the variables.

__Lemma 2.3__

If $s(\bar{x}) \equiv t(\bar{x})$, $u \sim w$, and $u(\bar{x}) \nmid \hat{R}$, then $u(\bar{x}) \equiv w(\bar{x})$.

__Proof__

Easy induction on proof of $u \sim w$, using Lemma 1.9. ■

The axioms and rules (i) - (v) above for $\sim$ allow us to decide whether $u \sim w$ in polynomial time. More importantly,

__Lemma 2.4__

There is a polynomial time algorithm to construct $\sim$ on R^+.

__Proof__

Construct edges between subtrees representing terms in R^+ as in the proof of Theorem 1 of [1]. ■

The following technical lemma establishes the relationship between $\sim$ and the ternary relation of Definition 1.3.

<u>Lemma 2.5</u>

If $x \sim x'$, $x\beta y$, and $x'\beta y'$, then $y \sim y'$.

<u>Proof</u>

Induction on length of β. ■

At this point we introduce two essential concepts. Let v_i be a variable.

<u>Definition 2.6</u>

v_i is <u>immune</u> if any of the following hold:

(i) $\exists u$ $v_i \sim u$ and $u \in R_1$ (i.e. u contains no variables);

(ii) $\exists u$ $v_i \sim u$ and v_i is a proper subterm of u;

(iii) $\exists u,w$ such that v_i is a proper subterm of u, $u \sim w$, and the labels at the roots of u and w differ. ■

<u>Definition 2.7</u>

v_i is <u>principal</u> if there is no proper term u (one not in G or V) with $v_i \sim u$. ■

Intuitively, immunity is a sufficient condition that a variable be forced syntactically to assume a value in r_1 in any assignment to the variables satisfying the schema $s \equiv t$.

<u>Lemma 2.8</u>

If v_i is immune and $s(\bar{x}) \equiv t(\bar{x})$, then $[x_i] \in r_1$.

<u>Proof</u>

We consider the 3 cases of immunity separately. Suppose $s(\overline{x}) \equiv t(\overline{x})$.

(i) $\exists u\ v_i \sim u\ \&\ u \in R_1$.

If $x_i \nmid \hat{R}_1$, then by Lemma 2.3, $x_i \equiv u(\overline{x}) \equiv u$, thus $[x_i] = [u] \in r_1$.

(ii) $\exists u\ v_i \sim u\ \&\ v_i$ is a proper subterm of u.

If $x_i \nmid \hat{R}_1$, then by Lemma 2.3, $x_i \equiv u(x)$ and $u(\overline{x})\alpha x_i$ for some $\alpha \neq \lambda$. But this contradicts Lemma 1.9 (viii).

(iii) $\exists u, w, \alpha \neq \lambda$ such that $u\alpha v_i$, $u \sim w$, and labels at the roots of u & w differ. If $x_i \nmid \hat{R}_1$, then $u(\overline{x}) \nmid \hat{R}$ by Lemma 1.9 (iv), hence $u(\overline{x}) \equiv w(\overline{x})$ by Lemma 2.3. But by Lemma 1.9 (v), $w(\overline{x})\underline{\alpha} x_i$, thus the root of w is the same as the root of u, contradicting the assumption. ■

Using the algorithm of Lemma 2.4, it is very easy to decide immunity and principality for a variable v_i. In fact,

<u>Lemma 2.9</u>

There is a polynomial time algorithm to

(i) decide whether v_i is immune;

(ii) decide whether v_i is principal;

(iii) if neither (i) nor (ii), produce a proper tree u such that $v_i \sim u$ and u contains occurrences of variables, but no occurrence of v_i.

<u>Proof</u>

Exercise. ■

Finally we introduce a ternary relation on $V\times D^{*}\times V$ defined inductively as follows:

<u>Definition 2.10</u>

$v_i\underline{\underline{\alpha}}v_k$ iff (i) $\exists u$ $v_i \sim u$ and $u\alpha v_k$; or

(ii) $\alpha=\beta\gamma$ and $\exists v_j$ $v_i\underline{\underline{\beta}}v_j\underline{\underline{\gamma}}v_k$. ■

This relation is meant to represent the syntactic interdependence of the variables. Its essential properties are outlined in the following 3 lemmas:

<u>Lemma 2.11</u>

If $v_i\underline{\underline{\alpha}}v_k$ and $\alpha\neq\lambda$, then v_i is not principal.

<u>Proof</u>

By Definition 2.10 there must be a v_j, an initial substring β of α, $\beta\neq\lambda$, and $v_{i_1}, u_{i_1}, \ldots, v_{i_m}, u_{i_m}$ such that

$v_i \sim u_{i_1}\lambda v_{i_1} \sim u_{i_2}\lambda v_{i_2} \sim \ldots \sim u_{i_m}\beta v_j$, i.e.

$v_i \sim v_{i_1} \sim \ldots \sim v_{i_m} \sim u_{i_m}\beta v_j$, hence $v_i \sim u_{i_m}\beta v_j$. ■

<u>Lemma 2.12</u>

If $v_i\underline{\underline{\alpha}}v_k$, $s(\bar{x})\equiv t(\bar{x})$, and $x_k\notin\hat{R}$, then $x_i\underline{\alpha}x_k$.

<u>Proof</u>

Induction on definition of $v_i\underline{\underline{\alpha}}v_k$:

<u>Basis</u>

By Lemma 2.3, $x_i\equiv u(\bar{x})\alpha x_k$, and the conclusion follows from Lemma 1.9 (v).

<u>Induction step</u> $v_i \underline{\underline{\beta}} v_j \underline{\underline{\gamma}} v_k$ and $\alpha = \beta\gamma$.

By the induction hypothesis, $x_j \underline{\gamma} x_k$, and by Lemma 1.9 (iv), $x_j \nmid \hat{R}$. Again by induction hypothesis, $x_i \underline{\beta} x_j$. The conclusion follows from Lemma 1.9 (ii). ∎

<u>Lemma 2.13</u>

If $v_i \underline{\underline{\alpha\beta}} v_k$ and $v_i \underline{\underline{\alpha}} v_j$ then $v_j \underline{\underline{\beta}} v_k$.

<u>Proof</u>

Induction on definition of $v_i \underline{\underline{\alpha\beta}} v_k$, using Lemma 2.5. ∎

The following lemma is key to the proof of the main theorem. It asserts the "independence" of principal variables from other variables.

<u>Lemma 2.14</u>

Let v_1 be principal and let $x_1, y_1 \nmid \hat{R}_1$. Let $\bigwedge_{j=1}^{\ell} w_j \equiv u_j$ be a conjunction of schemata with each $w_j \in R_1$ (i.e. w_j contains no variables) and $u_j \in R^+$. Then

$$\tau/_\Gamma \models Q_2 v_2 \ldots Q_k v_k \; s[v_1 \backslash x_1] \equiv t[v_1 \backslash x_1] \;\&\; \bigwedge_{j=1}^{\ell} w_j \equiv u_j [v_1 \backslash x_1]$$

iff

$$\tau/_\Gamma \models Q_2 v_2 \ldots Q_k v_k \; s[v_1 \backslash y_1] \equiv t[v_1 \backslash y_1] \;\&\; \bigwedge_{j=1}^{\ell} w_j \equiv u_j [v_1 \backslash y_1].$$

<u>Proof</u>

Let $x_1, y_1 \nmid \hat{R}_1$ and assume $x_1 \neq y_1$, otherwise the result is immediate. Let $x_2, \ldots, x_k \in \tau$ be arbitrary.

For $2 \le i \le k$, let

$$y_i = x_i [\alpha_1 \backslash y_1][\alpha_2 \backslash y_1] \ldots [\alpha_m \backslash y_1] \tag{2.15}$$

where the α_j are all strings α such that $v_i\underline{\alpha}v_1$ and $x_i\underline{\alpha}x_1$. Note that no α_j is an initial substring of another α_k, since distinct terms congruent to x_1 must be incomparable with respect to the subterm relation, by Lemma 1.9 (viii); thus y_i is well-defined.

y_i is then just x_i with some subterms congruent to x_1 replaced by y_1. The subterms replaced are determined by the dependence of v_i on v_1.

We now claim that

$$s(\bar{x})\equiv t(\bar{x}) \rightarrow s(\bar{y})\equiv t(\bar{y}). \tag{2.16}$$

Note $s(\bar{y})$ is just $s(\bar{x})$ with some subterms congruent to x_1 replaced by y_1, and similarly for $t(\bar{y})$. Since $s(\bar{x})\underline{\alpha}x_1$ iff $t(\bar{x})\underline{\alpha}x_1$ by Lemma 1.9 (v), it suffices to show that

(a) if $s(\bar{x})\underline{\alpha}x_1$ and $s(\bar{y})\underline{\alpha}y_1$ then $t(\bar{y})\underline{\alpha}y_1$, and

(b) if $t(\bar{x})\underline{\alpha}x_1$ and $t(\bar{y})\underline{\alpha}y_1$ then $s(\bar{y})\underline{\alpha}y_1$,

i.e. if a subterm of $s(\bar{x})$ is replaced by y_1 in $s(\bar{y})$, then the subterm in the corresponding position of $t(\bar{x})$ is replaced by y_1 in $t(\bar{y})$, and vice versa; for, it follows from several applications of Lemma 1.9 (vii) that $s(\bar{y})\equiv t(\bar{y})$. We will show (a); (b) is analogous.

Let $s(\bar{x})\underline{\alpha}x_1$ and $s(\bar{y})\alpha y_1$. There must v_i,β,γ with $\alpha = \beta\gamma$, $s\beta v_i$, $x_i\underline{\gamma}x_1$, and $v_i\underline{\gamma}v_1$. Since $t(\bar{x})\underline{\alpha}x_1$, one of the following three cases must hold:

(a) $\exists v_j\ \exists\xi\ \xi$ is a substring of α and $t\xi v_j$,

(b) $\exists v_j\ \exists \delta \neq \lambda\ t\alpha\delta v_j$, or

(c) $\exists u \epsilon R_1\ tau$.

But (c) is impossible, since $x_1 \nmid \hat{R}_1$, and (b) is not possible, since if it were true, and u were such that $t\beta u\gamma\delta v_j$, then $v_i \sim u$ by Lemma 2.5, hence $v_i \underline{\gamma\delta} v_j$; but $v_i \underline{\gamma} v_1$, so by Lemma 2.13, $v_1 \underline{\delta} v_j$, which by Lemma 2.11 contradicts the principality of v_1.

This leaves (a). Then $\exists \eta\ \alpha = \xi\eta$ and $x_j \underline{\eta} x_1$.

We have

$\alpha = \beta\gamma$, $s\beta v_i$, $v_i \underline{\gamma} v_1$,

$\alpha = \xi\eta$, $t\xi v_j$, $x_j \underline{\eta} x_1$.

If β is a substring of ξ, let $\xi = \beta\delta$. Then $\gamma = \delta\eta$, and $t\beta\delta v_j$ and $s\beta v_i$, so $v_i \underline{\delta} v_j$ by Lemma 2.5, thus $v_j \underline{\eta} v_1$ by Lemma 2.13.

If ξ is a substring of β, let $\beta = \xi\delta$. Then $\eta = \delta\gamma$, and $s\xi\delta v_i$ and $t\xi v_j$, so $v_j \underline{\delta} v_i$ by Lemma 2.5, thus $v_j \underline{\eta} v_1$.

Thus in either case we have $y_j \eta y_1$, hence $t(\bar{y})\alpha y_1$, as was to be shown, and claim (2.16) is verified.

Note also that

$$w_j \equiv u_j(\bar{x}) \rightarrow w_j \equiv u_j(\bar{y}),$$

since if $u_j(\bar{x}) \neq u_j(\bar{y})$ then both contain subterms not in $\hat{R}_1$, hence both $w_j \equiv u_j(\bar{x})$ and $w_j \equiv u_j(\bar{y})$ are false, by Lemma 1.9 (iv) & (v).

Let ϕ be the formula

$$s \equiv t \ \& \ \bigwedge_{j=1}^{k} w_j \equiv u_j.$$

Proceeding by induction on quantifiers, assume that for all terms $x_2, \ldots, x_\ell$,

$$Q_{\ell+1}x_{\ell+1}\cdots Q_k x_k\ \phi(x_1,x_2,\ldots,x_\ell,x_{\ell+1},\ldots,x_k)$$
$$\rightarrow Q_{\ell+1}x_{\ell+1}\cdots Q_k x_k\ \phi(y_1,y_2,\ldots,y_\ell,x_{\ell+1},\ldots,x_k),$$

where $y_2,\ldots,y_\ell$ are defined by (2.15). For any fixed $x_2,\ldots,x_{\ell-1}$, the existence of some x_ℓ satisfying

$$Q_{\ell+1}x_{\ell+1}\cdots Q_k x_k\ \phi(x_1,x_2,\ldots,x_\ell,\ldots,x_k)$$

implies the existence of a z satisfying

$$Q_{\ell+1}x_{\ell+1}\cdots Q_k x_k\ \phi(y_1,y_2,\ldots,y_{\ell-1},z,x_{\ell+1},\ldots,x_k),$$

namely $z = y_\ell$. Therefore

$$\exists x_\ell Q_{\ell+1}x_{\ell+1}\cdots Q_k x_k\ \phi(x_1,\ldots,x_k)$$
$$\rightarrow \exists x_\ell Q_{\ell+1}x_{\ell+1}\cdots Q_k x_k\ \phi(y_1,\ldots,y_{\ell-1},x_\ell,\ldots,x_k).$$

For any fixed $x_2,\ldots,x_{\ell-1}$, if $\forall x_\ell Q_{\ell+1}x_{\ell+1}\cdots Q_k x_k\ \phi(\bar{x})$, then since $x_1 \notin \hat{R}_1$, $\tau/_{\Gamma}$ is infinite (see Lemma 9 of [1]), and since for any α, an x_ℓ can be constructed such that not $x_\ell \underline{\alpha} x_1$, by Lemma 2.12 it follows that not $v_\ell \underline{\underline{\alpha}} v_1$. Thus $y_\ell = x_\ell$. But then

$$\forall x_\ell Q_{\ell+1}x_{\ell+1}\cdots Q_k x_k\ \phi(x_1,x_2,\ldots,x_\ell,\ldots,x_k)$$
$$\rightarrow \forall x_\ell Q_{\ell+1}x_{\ell+1}\cdots Q_k x_k\ \phi(y_1,y_2,\ldots,y_{\ell-1},y_\ell,x_{\ell+1},\ldots,x_k)$$
$$\rightarrow \forall x_\ell Q_{\ell+1}x_{\ell+1}\cdots Q_k x_k\ \phi(y_1,\ldots,y_{\ell-1},x_\ell,x_{\ell+1},\ldots,x_k).$$

We have shown by induction that

$$Q_2x_2\cdots Q_k x_k\ \phi(x_1,x_2,\ldots,x_k)$$
$$\rightarrow Q_2x_2\cdots Q_k x_k\ \phi(y_1,x_2,\ldots x_k),$$

and the converse follows from symmetry. ■

Parallel algorithms were introduced in [3,5]. It was shown in [3,5] that parallel polynomial time computations accept exactly the sets in PSPACE, and in [3] that such computations restricted to n alternations of $\wedge$-and $\vee$-branches, starting with an $\vee$-branch ($\wedge$-branch), accept exactly the sets in Σ_n^P (Π_n^P), where Σ_n^P (Π_n^P) is the nth Σ (Π) level of the polynomial time hierarchy [2]. Here we apply such a computation to the closed formula

$$Q_1v_1\ldots Q_kv_k \; s\equiv t,$$

using $\wedge$-branching to eliminate $\forall$ quantifiers, and $\vee$-branching to eliminate $\exists$ quantifiers.

<u>Theorem 2.17</u>

There is a parallel polynomial time algorithm to decide, given Γ and $Q_1v_1\ldots Q_kv_k \; s\equiv t$, whether

$$\tau/_\Gamma \models Q_1v_1\ldots Q_kv_k \; s\equiv t.$$

<u>Proof</u>

Let $w_j\equiv y_j$, $1\le j\le \ell$, be a set of schemata, $w_j\in R_1$, $y_j\in R^+$. The schemata $w_j\equiv y_j$ may be represented by an extra undirected edge set on the dag representing $s\equiv t$ and Γ. We show how to decide truth of formulas of the form

$$Q_1x_1\ldots Q_kx_k \; [s(\bar{x})\equiv t(\bar{x}) \;\&\; \bigwedge_{j=1}^{\ell} w_j\equiv y_j(\bar{x})]. \qquad (2.18)$$

On input Γ and (2.18), run the word problem algorithm of Theorem 1 of [1] to determine all congruent pairs of terms in R_1, and run the finiteness algorithm of Theorem 12 of [1] to determine if $\tau/_\Gamma$ is finite. Use the algorithm of Lemma 2.9 to determine immunity and principality of variables.

Suppose $Q_1 = \forall$. Then

$$\forall x_1 Q_2 x_2 \cdots Q_k x_k \; [s(\bar{x}) \equiv t(\bar{x}) \;\&\; \bigwedge_{j=1}^{\ell} w_j \equiv y_j(\bar{x})]$$

iff

$$\forall x_1 [x_1] \epsilon r_1 \rightarrow Q_2 x_2 \cdots Q_k x_k \; [s(\bar{x}) \equiv t(\bar{x}) \;\&\; \bigwedge_{j=1}^{\ell} w_j \equiv y_j(\bar{x})]$$
$$\& \; \forall x_1 [x_1] \notin r_1 \rightarrow Q_2 x_2 \cdots Q_k x_k \; [s(\bar{x}) \equiv t(\bar{x}) \;\&\; \bigwedge_{j=1}^{\ell} w_j \equiv y_j(\bar{x})]$$

iff

$$\bigwedge_{x_1 \epsilon R_1} Q_2 x_2 \cdots Q_k x_k \; [s(\bar{x}) \equiv t(\bar{x}) \;\&\; \bigwedge_{j=1}^{\ell} w_j \equiv y_j(\bar{x})] \tag{2.19}$$
$$\& \; \forall x_1 \notin \hat{R}_1 \; Q_2 x_2 \cdots Q_k x_k \; [s(\bar{x}) \equiv t(\bar{x}) \;\&\; \bigwedge_{j=1}^{\ell} w_j \equiv y_j(\bar{x})]. \tag{2.20}$$

At this point in the algorithm a series of $\wedge$-branches is made, spawning n+1 independent parallel processes, where n is the cardinality of R_1. The first n processes will each pick a different $x_1 \epsilon R_1$ and verify one clause of (2.19), by calling the algorithm recursively with input

$$\langle \Gamma, \; Q_2 v_2 \cdots Q_k v_k \; [s[v_1 \backslash x_1] \equiv t[v_1 \backslash x_1] \;\&\; \bigwedge_{j=1}^{\ell} w_j \equiv y_j[v_1 \backslash x_1]] \rangle.$$

The size of the problem is not increased, since consolidation of common subterms is allowed; occurrences of v_1 in the representation of s, t, and y_j may be replaced with pointers to x_1.

The remaining process will verify (2.20). If $\tau/_\Gamma$ is finite, then the process may immediately accept, since by Theorem 9 of [1], $r_1 = \tau/_\Gamma$, hence (2.20) holds vacuously. Suppose then that $\tau/_\Gamma$ is infinite. If v_1 is immune, then the process may immediately reject, since by Lemma 2.8, there is no satisfying assignment with $[x_1] \notin r_1$.

If v_1 appears in any of the y_j, then the process may immediately reject, by Lemma 1.9 (iv) and (v). If v_1 is not immune and does not appear in any y_j, but v_1 is principal, then by Lemma 2.14, (2.20) holds iff

$$Q_2x_2\ldots Q_kx_k\ [s(x^*,x_2,\ldots,x_k)\equiv t(x^*,x_2,\ldots,x_k)\ \&\ \bigwedge_{j=1}^{\ell} w_j\equiv y_j(\bar{x})] \quad (2.21)$$

where x^* is any term not in $\hat{R}_1$. In polynomial time, the process may find such an x^* by locating an m-ary θ and terms $u_1,\ldots,u_m \in R$ such that $\theta u_1\ldots u_m \notin \hat{R}$. The algorithm is exactly the one used to decide finiteness (see [1], Theorem 12). The word problem algorithm may be run to determine if $\theta u_1\ldots u_m \equiv x$ for some $x \in R_1$; if not, take $x^* = \theta u_1\ldots u_m$; if so, let u be the largest term in R_1 containing x as a subterm and take $x^* = \theta u\ldots u$. It is easily verified that $x^* \notin \hat{R}_1$, but all proper subterms of x^* are in R_1.

Now the process determines whether (2.21) holds, by calling the algorithm recursively with input

$$\langle \Gamma,\ Q_2v_2\ldots Q_kv_k\ [s[v_1\backslash x^*]\equiv t[v_1\backslash x^*]\ \&\ \bigwedge_{j=1}^{\ell} w_j\equiv y_j]\rangle.$$

As before, the size of the problem is not increased; we may replace each occurrence of v_1 in the dag representing s and t with the label θ and edges to each of the immediate subterms of x^*, which all lie in R_1.

Finally, if v_1 is neither immune nor principal, then Lemma 2.9 guarantees us a $u \in R^+$ such that $v_1 \sim u$, u is a proper tree containing occurrences of variables, and v_1 does not occur in u. Then by Lemma 2.3, (2.20) is equivalent to

$$\forall x_1 \nmid \hat{R}_1\ Q_2 x_2 \ldots Q_k x_k\ [s(\bar{x}) \equiv t(\bar{x})\ \&\ \bigwedge_{j=1}^{\ell} w_j \equiv y_j(\bar{x})\ \&\ x_1 \equiv u(\bar{x})], \qquad (2.22)$$

and since $x_1 \equiv u(\bar{x})$ implies $s[v_1 \backslash u](\bar{x}) \equiv s(\bar{x})$ and $t[v_1 \backslash u](\bar{x}) \equiv t(\bar{x})$, (2.22) is equivalent to

$$\forall x_1 \nmid \hat{R}_1\ Q_2 x_2 \ldots Q_k x_k\ [s[v_1 \backslash u](\bar{x}) \equiv t[v_1 \backslash u](\bar{x})\ \&\ \bigwedge_{j=1}^{\ell} w_j \equiv y_j(\bar{x})$$
$$\&\ x_1 \equiv u(\bar{x})]. \qquad (2.23)$$

Note that schema $s[v_1 \backslash u] \equiv t[v_1 \backslash u]$ may be represented as concisely as $s \equiv t$, by replacing occurrences of v_1 with pointers to u. Since u contains no occurrences of v_1, the graph remains a dag.

At this point the process may reject immediately if there is more than one operator symbol in θ; otherwise there would certainly be an $x_1 \nmid \hat{R}_1$ with its root symbol differing from that of u, thus $x_1 \not\equiv u(\bar{x})$ for any $\bar{x}$ by Lemma 1.9 (v). Hence assume θ is the only operator symbol, and let θ be m-ary.

The process may reject immediately if $u\alpha b$ and $b \epsilon G$, since if y is any term not in $\hat{R}$ and $x_1 \alpha y$, then $x_1 \nmid \hat{R}$ and $x_1 \not\equiv u(\bar{x})$ by Lemma 1.9 (iv), (v). Therefore we may assume all leaves of u are variables.

Finally, the process may reject immediately if $u\alpha v_i$, $u\beta v_i$, and $\alpha \neq \beta$, since if $y_1, y_2 \nmid \hat{R}$, $y_1 \not\equiv y_2$, $x_1 \alpha y_1$, $x_1 \beta y_2$, then $x_1 \nmid \hat{R}$ and $x_1 \not\equiv u(\bar{x})$, as above. y_1 and y_2 exist, since we have assumed $\tau/_\Gamma$ infinite. Thus we may assume that each leaf of u is labeled with a different variable name. This says that no two subterms of u are identical, hence the physical representation of the term u is a tree, not just a dag. This observation will be instrumental in achieving a good time analysis below.

Now, since u is a proper tree, say $u = \theta u_1 \ldots u_m$, and since every element of G is in R_1, (2.23) is equivalent to

$$\forall x_{11}\forall x_{12}\ldots\forall x_{1m}\ \theta x_{11}\ldots x_{1m}\notin\hat{R}_1 \rightarrow Q_2x_2\ldots Q_kx_k\ [s[v_1\backslash u](\bar{x})\equiv t[v_1\backslash u](\bar{x})\ \&\ \bigwedge_{j=1}^{\ell} w_j\equiv y_j(\bar{x})\ \&\ \theta x_{11}\ldots x_{1m}\equiv\theta u_1(\bar{x})\ldots u_m(\bar{x})],$$

which by Lemma 1.9 (ix) is equivalent to

$$\forall x_{11}\forall x_{12}\ldots\forall x_{1m}\ \theta x_{11}\ldots x_{1m}\notin\hat{R}_1 \rightarrow Q_2x_2\ldots Q_kx_k\ [s[v_1\backslash u](\bar{x})\equiv t[v_1\backslash u](\bar{x})\ \&\ \bigwedge_{j=1}^{\ell} w_j\equiv y_j(\bar{x})\ \&\ \bigwedge_{j=1}^{m} x_{1j}\equiv u_j(\bar{x})], \tag{2.24}$$

which in turn is equivalent to

$$\bigwedge_{x_{11}\in\hat{R}_1}(\forall x_{12}\ldots\forall x_{1m}\ \theta x_{11}\ldots x_{1m}\notin\hat{R}_1 \rightarrow Q_2x_2\ldots Q_kx_k\ [s[v_1\backslash u](\bar{x})\equiv t[v_1\backslash u](\bar{x})\ \&\ \bigwedge_{j=1}^{\ell} w_j\equiv y_j(\bar{x})\ \&\ \bigwedge_{j=1}^{m} x_{1j}\equiv u_j(\bar{x})]) \tag{2.25}$$

$$\&\ \forall x_{11}\notin\hat{R}_1\ \forall x_{12}\ldots\forall x_{1m}\ \theta x_{11}\ldots x_{1m}\notin\hat{R}_1 \rightarrow Q_2x_2\ldots Q_kx_k\ [s[v_1\backslash u](\bar{x})\equiv t[v_1\backslash u](\bar{x})\ \&\ \bigwedge_{j=1}^{\ell} w_j\equiv y_j(\bar{x})\ \&\ \bigwedge_{j=1}^{m} x_{1j}\equiv u_j(\bar{x})] \tag{2.26}$$

and as above, the process $\wedge$-branches into n+1 processes, the first n of which each verify a clause of (2.25) for some $x_{11}\in R_1$, and the remaining process verifies (2.26). Since $x_{11}\notin\hat{R}_1 \rightarrow \theta x_{11}\ldots x_{1m}\notin\hat{R}_1$, (2.26) is equivalent to

$$\forall x_{11}\notin\hat{R}_1\ \forall x_{12}\ldots\forall x_{1m}\ Q_2x_2\ldots Q_kx_k\ [s[v_1\backslash u](\bar{x})\equiv t[v_1\backslash u](\bar{x})\ \&\ \bigwedge_{j=1}^{\ell} w_j\equiv y_j(\bar{x})\ \&\ \bigwedge_{j=2}^{m} x_{1j}\equiv u_j(\bar{x})\ \&\ x_{11}\equiv u_1(\bar{x})]. \tag{2.27}$$

Now each process attempting to verify a clause of (2.25) for some $x_{11}\in R_1$ may take $w_{\ell+1} = x_{11}$, $y_{\ell+1} = u_1$, and verify that

$$\forall x_{12} \ldots \forall x_{1m}\ \theta x_{11} \ldots x_{1m} \nmid \hat{R}_1 \rightarrow$$
$$Q_2 x_2 \ldots Q_k x_k\ [s[v_1 \backslash u](\bar{x}) \equiv t[v_1 \backslash u](\bar{x})$$
$$\& \bigwedge_{j=1}^{\ell+1} w_j \equiv y_j(\bar{x})\ \& \bigwedge_{j=2}^{m} x_{1j} \equiv u_j(\bar{x})]$$

by attacking x_{12} in the same way x_{11} was attacked in (2.24). Eventually, there are some processes attempting to verify

$$\theta x_{11} \ldots x_{1m} \nmid \hat{R}_1 \rightarrow$$
$$Q_2 x_2 \ldots Q_k x_k\ [s[v_1 \backslash u](\bar{x}) \equiv t[v_1 \backslash u](\bar{x})$$
$$\& \bigwedge_{j=1}^{\ell} w_j \equiv y_j(\bar{x})\ \& \bigwedge_{j=1}^{m} x_{1j} \equiv u_j(\bar{x})],$$

with $x_{11}, \ldots, x_{1m} \in \hat{R}_1$. The process may use the word problem algorithm to dispose of the antecedent.

As for the process verifying (2.27), if u_1 is a proper tree, x_{11} and u_1 are broken up in the same way x_1 and u were in (2.23). This procedure continues until all subterms of u have been broken up. Since variables reside at the leaves of u, we are left with formulas of the form

$$\forall z_{i_1} \nmid \hat{R}_1 \ldots \forall z_{i_r} \nmid \hat{R}_1\ Q_2 x_2 \ldots Q_k x_k\ [s[v_1 \backslash u](\bar{x}) \equiv t[v_1 \backslash u](\bar{x})$$
$$\& \bigwedge_{j=1}^{\ell} w_j \equiv y_j(\bar{x})\ \& \bigwedge_{j=1}^{r} z_{i_j} \equiv x_{i_j}]. \tag{2.28}$$

It follows from the fact that the size of u was no bigger than the size of its physical representation that the above was a polynomial time $\wedge$-computation. Moreover, since no two subterms of u were identical, each x_i occurs among the x_{i_j} at most once. If some x_{i_j} is universally quantified, then the process can immediately reject, since (2.28) would imply $\forall z_{i_j}\ \forall x_{i_j}\ z_{i_j} \equiv x_{i_j}$, which is false in any infinite (or for that matter, any nontrivial) structure.

Let $\bar{x}'$ be $\bar{x}$ with each x_{i_j} replaced by z_{i_j}. Since $z_{i_j} \equiv x_{i_j}$ is in conjunction with the rest of the formula, (2.28) is equivalent to

$$\forall z_{i_1} \nmid \hat{R}_1 \ldots \forall z_{i_r} \nmid \hat{R}_1 \; Q_2 x_2 \ldots Q_k x_k \quad [s[v_1 \backslash u](\bar{x}') \equiv t[v_1 \backslash u](\bar{x}') \;\&\; \bigwedge_{j=1}^{\ell} w_j \equiv y_j(\bar{x}') \;\&\; \bigwedge_{j=1}^{r} z_{i_j} \equiv x_{i_j}]. \tag{2.29}$$

But now, since each x_{i_j} occurs only in the atomic formula $z_{i_j} \equiv x_{i_j}$ and each x_{i_j} is existentially quantified, (2.29) is equivalent to

$$\forall z_{i_1} \nmid \hat{R}_1 \ldots \forall z_{i_r} \nmid \hat{R}_1 \; Q_2' x_2' \ldots Q_{k'}' x_{k'}' \; [s[v_1 \backslash u](\bar{x}') \equiv t[v_1 \backslash u](\bar{x}') \;\&\; \bigwedge_{j=1}^{\ell} w_j \equiv y_j(\bar{x}')] \tag{2.30}$$

where $Q_2' x_2' \ldots Q_{k'}' x_{k'}'$ is $Q_2 x_2 \ldots Q_k x_k$ with all the $\exists x_{i_j}$ removed.

(2.30) is similar to (2.18), with the exception that some quantifiers are bounded (this constitutes no problem, as the reader may easily verify). The new string of quantifiers has one less quantifier than the old string, and has no more alternations than that of (2.18); and if the new string begins with an $\exists$, it has fewer. Moreover, the physical representation of (2.30) is no bigger than that of (2.18). Thus the algorithm may be reapplied to (2.30).

Eliminating a leading $\exists$ is similar in most respects, and we leave this case as an exercise.

Finally, when all quantifiers have been eliminated, we are left with a formula

$$\bigwedge_{j=1}^{\ell} w_j \equiv u_j, \quad w_j, u_j \in \tau,$$

which can be solved by the word problem algorithm of Theorem 1 of [1]. ∎

Theorem 2.31

$S_n (V_n)$ is $\le_{\log}^{m}$-complete for $\Sigma_n^P (\Pi_n^P)$, and $\bigcup_{n=0}^{\infty} S_n \cup V_n$ is $\le_{\log}^{m}$-complete for PSPACE.

Proof

That $S_n (V_n)$ is in $\Sigma_n^P (\Pi_n^P)$ follows the observation that executing the algorithm of Theorem 2.17 on an input in $S_n (V_n)$ results in a $\Sigma_n^P (\Pi_n^P)$ computation [3]; that $\bigcup_{n=0}^{\infty} S_n \cup V_n$ is in PSPACE follows from the fact that //PTIME = PSPACE [3], and $\bigcup_{n=0}^{\infty} S_n \cup V_n$ is in //PTIME by the algorithm of Theorem 2.17.

To show these problems are hard for their respective classes, there are trivial reductions from the corresponding sets of quantified Boolean expressions, shown in [2] to be complete, to S_n and V_n, by taking Γ to present the two-element Boolean algebra, as in Example 1.1. ∎

3. Directions for Further Research

The proofs in the preceding section are far from elegant. It would perhaps be instructive to exploit the relationship between finitely presented algebras and the finite tree automata of Thatcher and Wright [4] and others, to smooth things out a bit. It is conjectured that, given schema $s \equiv t$ and Γ, the set

$$\{\langle x_1 \ldots x_k\rangle \mid s(x_1 \ldots x_k) \equiv t(x_1 \ldots x_k)\}$$

is a regular set of k-tuples of trees.

If the above is the case, what can be said if we limit our attention to monadic operators? Trees become strings in this case, and it is known that the membership problem for regular sets is complete for deterministic logspace and the emptiness problem is complete for nondeterministic logspace. Do these results generalize to an analog of Theorem 2.31 for the logspace hierarchy in the monadic case?

Finally, what is the import of Theorem 2.31 with respect to first order logic with equality? Can Theorem 2.17 be extended easily to include atomic formulas $s \equiv t$ connected by the Boolean connectives $\wedge, \vee$? The conjecture is yes, based on the fact that first order predicate logic with equality but without negation is NP-complete [6].

References

[1] Kozen, D., "Complexity of Finitely Presented Algebras," Cornell University Department of Computer Science TR76-294, December 1976.

[2] Stockmeyer, L.J., and A.R. Meyer, "Word Problems Requiring Exponential Time," Proc. 5th ACM Symposium on Theory of Computing, April 1973, pp. 1-9.

[3] Kozen, D.,"On Parallelism in Turing Machines," Proc. 17th IEEE Symposium on Foundations of Computer Science, October 1976, pp. 89-97.

[4] Thatcher, J.W., and J.B. Wright, "Generalized Finite Automata Theory with an Application to a Decision Problem of Second Order Logic," Math. Syst. Th. 2, 1968, pp. 57-81.

[5] Chandra, A.K., and L.J. Stockmeyer, "Alternation", Proc. 17th IEEE Symposium on Foundations of Computer Science, October 1976, pp. 98-108.

[6] Kozen, D., "First Order Predicate Logic Without Negation is NP-Complete," in preparation.