ELSEVIER
Theoretical Computer Science 180 (1997) 325–339
Theoretical Computer Science

Finite semigroup varieties defined by programs

Pierre Péladeau[a], Howard Straubing[b,1], Denis Thérien[c,*]

[a] *Booz Allen & Hamilton Inc., 112 Avenue Kléber, 75116, Paris, France*
[b] *Computer Science Department, Boston College, Chestnut Hill, MA 02167, USA*
[c] *School of Computer Science, McGill University, 3480 University Street, Montréal, Québec, Canada H3A 2A7*

Received April 1992; revised June 1993
Communicated by M. Nivat

Abstract

We study the regular languages recognized by polynomial-length programs over finite semigroups belonging to product varieties $\mathbf{V} * \mathbf{LI}$, where $\mathbf{V}$ is a variety of finite monoids, and $\mathbf{LI}$ is the variety of finite locally trivial semigroups. In the case where the semigroup variety has a particular closure property with respect to programs, we are able to give precise characterizations of these regular languages. As a corollary we obtain new proofs of the results of Barrington, Compton, Straubing and Thérien characterizing the regular languages in certain circuit complexity classes.

1. Introduction

There is a growing body of research in theoretical computer science concerning the complexity theory of small-depth boolean circuits. These studies are motivated in part by the connections to parallel computing and relativized Turing machine complexity. It is also one of the few areas in which researchers have been able to prove super-polynomial lower bounds on computation resources required to solve certain specific problems (see, for example, [1, 8, 13, 14]). Still, many open questions remain. We do not even know whether every language in NP can be recognized by a polynomial-size family of bounded depth circuits in which every gate computes the sum of its inputs modulo 6.

Several new mathematical approaches have been devised to study these problems. These include the representation of circuit behavior by polynomials or similar algebraic objects (see, for example, [14, 5]), and the application of multiparty communication complexity [17, 9]. For the past several years, we and our colleagues

* Corresponding author. Research supported by NSERC grant A4546 and FCAR team grant 93-ER0642.
[1] Research supported by NSF Grant CCR-9203208.

PII S0304-3975(96)00297-6

have been studying an approach that uses finite semigroups. This has permitted us to exploit the considerable literature on the connections between semigroups and automata. Among the accomplishments of this line of investigation are the characterization of the circuit complexity class NC^1 and many of its subclasses in semigroup-theoretic terms [2, 6], new lower bounds for circuits containing only modular counting gates [5] and for bounded-width branching programs [4], and the characterization of the regular languages in various circuit complexity classes [3].

The central notion in this investigation is that of a *program over a finite semigroup*. The program inspects the bits of its input string – the order of the queries is fixed by the program, but each input bit can be queried many times – and after each query emits an element of the semigroup. Acceptance or rejection of the query is determined by the product of the emitted elements. (The precise definition will be given in Section 2.) We usually require that the number of queries made by the program is bounded by a polynomial in the number of inputs. With this notion we are able to translate many of the known and conjectured lower bounds for small-depth circuits into algebraic language. For example, the theorem of Furst et al. [8] that one cannot add modulo k in AC^0 is equivalent to the fact that one cannot use polynomial-size programs over an aperiodic semigroup (a semigroup that contains no nontrivial groups) to multiply in a non-aperiodic semigroup. This has raised the hope (not yet realized) that we might be able to give direct algebraic proofs of the translated statements, and thereby settle some of the unanswered questions in circuit complexity.

We are thus led to consider classes of finite semigroups that are, in a sense, closed with respect to polynomial-size programs. More precisely, we define a p-variety of finite semigroups to be a family $\mathbf{V}$ of semigroups that is closed under the formation of homomorphic images, subsemigroups and finite direct products, and such that if multiplication in a finite semigroup S can be performed by a family of polynomial-size programs over a member of $\mathbf{V}$, then $S \in \mathbf{V}$. Known and conjectured circuit lower bounds are then equivalent to the assertion that certain classes of finite semigroups form p-varieties. This is very closely related to the classification of the *regular languages* recognized by polynomial-size programs over certain finite semigroups: McKenzie et al. [10] show that the power of polynomial-size programs over a class of monoids is essentially captured by the regular languages recognized by such programs. Barrington et al. [3], and Straubing [16] reformulate conjectured circuit lower bounds in terms of the classification of regular languages in various circuit complexity classes, and show how these lower bounds are particular instances of a general principle (as yet unproved!) concerning the definability of regular languages in certain extensions of first-order logic.

In the present paper, we study p-varieties and prove a general theorem characterizing the regular languages contained in p-varieties. This theorem (Theorem 4.2) contains as special instances the results of [3, 16] and manages to avoid a particularly difficult point in the proof in [16]. As a result we give, in Section 5, very short proofs of the characterizations (assuming the appropriate circuit lower bounds) of the regular languages belonging to various circuit complexity classes.

We, regrettably, had to introduce a technical complication: In earlier work in this subject it was always most convenient to suppose that the semigroups in question are monoids (that is, they contain an identity element). One dealt almost exclusively with the varieties of finite solvable groups and finite solvable monoids, and with their subvarieties formed by restricting the orders of the groups that appear. For the case of the variety of finite solvable groups it was observed in [16] that we must switch from monoids to semigroups. So while it would be prettier to have a theorem characterizing the regular languages in p-varieties $\mathbf{V}$, where $\mathbf{V}$ is a variety of finite monoids, our theorem, in order to achieve the appropriate generality, characterizes the regular languages in varieties of finite semigroups that are product varieties of the form $\mathbf{V} * \mathbf{LI}$, where $\mathbf{V}$ is a variety of finite monoids. The nice closure properties of semigroup varieties of this form (Corollary 3.3 below) make their study more appropriate.

2. Definitions

2.1. Regular languages and semigroups

See Eilenberg [7] or Pin [12] for basic information on the relationship between automata and finite semigroups. Here we mention the essential points.

Let A be a finite alphabet, and A^+ (resp. A^*) the free semigroup (resp. monoid) generated by A. In this paper we consider only semigroups and A^+, but every one of the subsequent definitions can be rewritten for monoids and A^*.

The concatenation product of two subsets K and L of A^+ is the set $KL = \{uv\colon u \in K \ \& \ v \in L\}$. The $+$ of a subset K of A^+, written K^+, is the subsemigroup of A^+ generated by K.

The *left* (resp. *right*) *quotient* of a language $L \subseteq A^+$ by a set K is the set $K^{-1}L = \{v\colon$ there exists $u \in K$ such that $uv \in L\}$ (resp. $LK^{-1} = \{u\colon$ there exists $v \in K$ such that $uv \in L\}$).

A language $L \subseteq A^+$ is said to be *regular* if it can be obtained from the finite subsets of A^+ by using the union, concatenation and $+$ operations. The set of all regular languages will be denoted by *Reg*. By Kleene's theorem, the regular languages are precisely those that are recognized by finite automata.

A language $L \subseteq A^+$ is said to be *recognized* by a finite semigroup S if there is a morphism $\phi\colon A^+ \to S$ such that $L = \phi^{-1}\phi(L)$. This is equivalent to recognition of L by a finite automaton, and thus a language is regular if and only if it is recognized by a finite semigroup. We will consider another kind of recognition by finite semigroups a bit later on. Thus, to distinguish between the two, we will use the term 'm-recognize' (the 'm' stands for 'morphism') as a synonym for 'recognize'. In the next section we shall define 'p-recognize' (the 'p' stands for 'program').

A semigroup S is said to *divide* or *m-divide* another semigroup T if there is a surjective morphism from a subsemigroup of T onto S.

The smallest semigroup (under the division relation) m-recognizing a regular language $L \subseteq A^+$ is called the *syntactic semigroup* of L and is denoted $S(L)$. The syntactic monoid of a language $L \subseteq A^*$ is defined analogously and is denoted $M(L)$. There is an alternative characterization of the syntactic semigroup in terms of congruences: Let $u, v \in A^+$. Define $u \equiv_L v$ if and only if

$$\{(x, y) \in A^* \times A^* : xuy \in L\} = \{(x, y) \in A^* \times A^* : xvy \in L\}.$$

The syntactic semigroup of L is the quotient of A^+ by the congruence $\equiv_L$. The morphism $\eta_L : A^+ \to S(L)$ that maps each word to its congruence class is called the syntactic morphism of L.

A *variety* of finite semigroups is a set $\mathbf{V}$ of finite semigroups which is closed under division and finite direct product. (This is somewhat at odds with the standard terminology in universal algebra, where a variety of algebras is closed under arbitrary direct products, and thus necessarily contains infinite algebras. What we have defined is often called a *pseudovariety* of finite semigroups.) Given a variety of finite semigroups $\mathbf{V}$, we will denote by $\mathscr{M}(\mathbf{V})$ the set of languages that are m-recognized by semigroups in $\mathbf{V}$.

A finite semigroup S is *locally trivial* if for all $e, s \in S$ with e idempotent, $ese = e$. The locally trivial semigroups form a variety, which we denote $\mathbf{LI}$. An equivalent characterization of $\mathbf{LI}$ is the following: $S \in \mathbf{LI}$ if and only if there exists $d > 0$ such that for all $s_1, \ldots, s_m \in S$ with $m \geqslant d$, the product $s_1 \cdots s_m$ depends only on $s_1 \cdots s_d$ and $s_{m-d+1} \cdots s_m$. It readily follows that every element of S^d is idempotent.

If the product $s_1 \cdots s_m$ depends only on $s_{m-d+1} \cdots s_m$, then S is said to be *d-definite*. If the product depends only on $s_1 \cdots s_d$ then S is *d-reverse-definite*. Let A be a finite alphabet. The free d-definite semigroup on A has as its underlying set all the words in A^+ of length less than or equal to d. The product $u \cdot v$ of two elements is the word uv if $|uv| \leqslant d$, and is the suffix of uv of length d otherwise. The free d-reverse-definite semigroup on A is defined identically, except we replace 'suffix' by 'prefix'.

Our treatment of wreath products is essentially that of Eilenberg [7], to which the reader is referred for all the details not included in this brief summary: A transformation semigroup is a pair (Q, S) where Q is a finite set (the set of *states*) and S is a semigroup of maps from Q into itself, with left-to-right composition as the operation. The image of a state q under a map s is denoted qs or $q \cdot s$. The *wreath product* $(P, T) \circ (Q, S)$ of two transformation semigroups is the transformation semigroup

$$(P \times Q, T^Q \times S),$$

where for all $p \in P$, $q \in Q$, $F \in T^Q$, and $s \in S$ we set

$$(p, q) \cdot (F, s) = (p \cdot F(q), q \cdot s).$$

If S is a semigroup we will also use S to denote the tranformation semigroup (S^1, S). Here, S^1 denotes the set formed by adjoining an identity element to S if S is not a monoid; if S is a monoid then $S^1 = S$. Right multiplication by elements of S thus

defines a semigroup of tranformations on S^1 isomorphic to S. This is what is intended when we write wreath products $T \circ S$, where S and T are semigroups. When we say that a third semigroup U divides $S \circ T$ we mean that U divides the semigroup of transformations of $S \circ T$.

If $\mathbf{V}$ and $\mathbf{W}$ are finite semigroup varieties, then $\mathbf{V} * \mathbf{W}$ denotes the family of all finite semigroups that divide the semigroup of transformations of a transformation semigroup $(P,T) \circ (Q,S)$, where $T \in \mathbf{V}$ and $S \in \mathbf{W}$. In the case where $\mathbf{V}$ is a finite monoid variety, we require in addition that the identity of T act as the identity mapping on P.

2.2. Circuits and programs

Our definition of a boolean circuit differs a bit from standard definitions in that the inputs are elements of a finite alphabet A rather than $\{0,1\}$. For this reason, the input nodes of our circuits are labelled with functions that translate the input value in A into a boolean value.

More precisely, a *boolean circuit* with n inputs is a directed acyclic graph with one sink node. The interior nodes or gates of the circuit are labelled with boolean functions such as *AND*, *OR*, *NOT* or MOD_q. The *AND*, *OR* and *NOT* gates have their usual interpretation. A MOD_q gate outputs 1 if and only if the sum of its input bits is divisible by q. The *fan-in* of a gate is its number of input wires. The nodes with no incoming vertices are called input gates. An input gate is labelled with an integer $i \in \{1,\ldots,n\}$ and a function $f : A \to \{0,1\}$ and on an input word $w = a_1 \ldots a_n \in A^n$ it will output the value $f(a_i)$. A word $w \in A^n$ is said to be recognized by such a circuit if and only if the sink node outputs 1. A circuit with n inputs thus recognizes a subset of A^n.

A language $L \subseteq A^+$ is said to be recognized by a family $(C_n)_{n \geqslant 1}$ of boolean circuits (one for each input length) if C_n recognizes $L \cap A^n$.

Families of circuits, and the languages they recognize, are classified according to their size, depth, fan-in and type of gates. The class NC^i is made up of families of circuits of polynomial size, $\mathrm{O}((\log n)^i)$ depth and constant fan-in using *AND*, *OR* and *NOT* gates. We will follow the practice of using NC^i (and the other classes we introduce here) to refer to both a class of circuit families and to the class of languages accepted by such families. It will always be clear from the context which meaning is intended. We will only consider the classes NC^0 and NC^1.

The class $\mathrm{ACC}^0(q)$ is comprised of families of circuits of polynomial size, constant depth and arbitrary fan-in using *AND*, *OR*, *NOT* and MOD_q gates for a fixed $q > 1$. ACC^0 is the union over all q of $\mathrm{ACC}^0(q)$. The class AC^0 is the same as ACC^0 without the MOD_q gates. The class $\mathrm{CC}^0(q)$ is the same as $\mathrm{ACC}^0(q)$ without *AND*, *OR*, and *NOT* gates. CC^0 is the union of the $\mathrm{CC}^0(q)$ over all $q > 1$. A $CONG_{c,t,q}$ gate outputs 1 if and only if the sum of the inputs is congruent to c mod q and threshold t (this last phrase means that the sum of the inputs is less than t if and only if $c < t$). Observe that a $CONG_{1,1,1}$ gate is an *OR* gate, and a $CONG_{0,0,q}$ gate is a MOD_q gate.

Given a class of circuits $\mathscr{C}$, we will denote by $\widehat{\mathscr{C}}$ the set of circuits obtained from $\mathscr{C}$ by replacing the input gates by NC^0 circuits.

An *instruction* over a semigroup S is a pair (i,f) where i is an integer and f is a function from A to S. Given a word $w=a_1\dots a_n\in A^n$ with $i\leqslant n$, the instruction (i,f) will produce the element $f(a_i)$ of S. A *program* over a semigroup S is a sequence of instructions over S. Let $\Psi_n=(i_1,f_1)(i_2,f_2)\dots(i_l,f_l)$ be a program in which the indices i_j $(1\leqslant j\leqslant l)$ range from 1 to n. Then Ψ_n defines a function from A^n to S in the following way: for $w=a_1\dots a_n\in A^n$ $\Psi_n(w)=f_1(a_{i_1})*f_2(a_{i_2})*\cdots*f_l(a_{i_l})$, with $*$ denoting the product in S. The *size* of the program is the length of the sequence of instructions. If the program has size n and the index in the ith instruction is i, then the program is said to be a *single-scan program.* A language $L\subseteq A^+$ is said to be *p-recognized* by a semigroup S if there is a sequence $(\Psi_n)_{n\geqslant 1}$ of polynomial-size programs over S such that $L\cap A^n=\Psi_n^{-1}\Psi_n(L\cap A^n)$ for each $n\geqslant 1$. Equivalently, for each $n>0$ there is a set $X_n\subseteq S$ of *accepting values* such that $L\cap A^n=\Psi^{-1}(X_n)$.

A fundamental result of Barrington [2] states that a language belongs to NC^1 if and only if it is p-recognized by a finite semigroup.

Given a variety of finite semigroups $\mathbf{V}$, we will denote by $\mathscr{P}(\mathbf{V})$ the set of all languages p-recognized by the semigroups in $\mathbf{V}$.

A language $K\subseteq A^+$ is p-reducible to $L\subseteq B^+$ if there exists a polynomial-size sequence $(\Psi_n)_{n\geqslant 1}$ of programs over B^+ such that every instruction of Ψ_n emits a single letter of B, and for each $n\geqslant 1$, $K\cap A^n=\Psi_n^{-1}(L)$.

Let S be a semigroup and $\eta : S^+\to S$ be the unique semigroup morphism that extends the identity mapping on S. A semigroup S is said to *p-divide* a semigroup T if for each $Q\subseteq S$ the language $\eta^{-1}(Q)$ is p-recognized by T. The set of languages $\eta^{-1}(Q)$ as Q ranges over the subsets of S is called the set of *word problems* of S. A *p-variety* of semigroups is a class of semigroups which is closed under p-division and finite direct products. We define p-varieties of monoids similarly.

Let $L\subseteq A^*$ be a language. A letter $e\in A$ is a *neutral letter* for L if for all $u,v\in A^*$, $uv\in L$ if and only if $uev\in L$.

A *k-instruction* over a finite semigroup S is a pair $(\boldsymbol{i},f)$ where $\boldsymbol{i}=(i_1,\dots,i_k)$ is a vector of k integers and f is a function from A^k to S. On an input word w the instruction emits $f(a_{i_1},\dots,a_{i_k})\in S$. A *k-program* is a sequence of k-instructions. We will say that a language $L\subseteq A^+$ is *$\widehat{p}$-recognized* by a semigroup S if for some $k>0$ there is a sequence of k-programs over S of polynomial length recognizing L. $\widehat{\mathscr{P}}(\mathbf{V})$ will denote the set of all languages $\widehat{\mathrm{p}}$-recognized by semigroups in $\mathbf{V}$.

3. Closure properties and equivalences

Languages recognized by members of varieties of monoids have nice closure properties [10].

Proposition 3.1. *If $\mathbf{V}$ is a variety of monoids, then $\mathscr{P}(\mathbf{V})$ is closed under finite boolean operations, left and right quotients by a finite set of words, and p-reductions.*

We are not able to prove closure under boolean operations when we switch from monoids to arbitrary finite semigroups. Thus, it is usually easier to work with monoids. However, for purposes of determining the regular languages in certain complexity classes, it turns out to be more convenient to work with semigroup varieties of the form $\mathbf{V} * \mathbf{LI}$, where $\mathbf{V}$ is a variety of finite monoids.

Theorem 3.2. *For any nontrivial variety of finite monoids* $\mathbf{V}$: $\mathscr{P}(\mathbf{V} * \mathbf{LI}) = \widehat{\mathscr{P}}(\mathbf{V})$.

Proof. First, suppose $L \in \mathscr{P}(\mathbf{V} * \mathbf{LI})$. There is thus a polynomial-size sequence of programs over a finite semigroup $S \in \mathbf{V} * \mathbf{LI}$ such that to every input sequence

$$w = a_1 \cdots a_n,$$

the program associates the sequence

$$a_{i_1} \cdots a_{i_r}$$

in the order in which they are consulted by the program, and emits the string

$$(s_{i_1}, \ldots, s_{i_r})$$

of elements of S. We use the characterization of $\mathbf{V} * \mathbf{LI}$ given by Straubing [15]: There exists $k > 0$ such that the value $s_{i_1} \cdots s_{i_r}$ is determined by

(i) The sequence $(s_{i_1}, \ldots, s_{i_k})$.
(ii) The sequence $(s_{i_{r-k+1}}, \ldots, s_{i_r})$.
(iii) The sequence

$$(s_{i_1}, \ldots, s_{i_k}), (s_{i_2}, \ldots, s_{i_{k+1}}), \ldots, (s_{i_{r-k+1}}, \ldots, s_{i_r}),$$

modulo a congruence $\cong$ on $(S^k)^*$ whose quotient is in $\mathbf{V}$.

Now consider the sequence of k-tuples

$$(i_1, \ldots, i_k), (i_2, \ldots, i_{k+1}), \ldots, (i_{r-k+1}, \ldots, i_r).$$

This will be the sequence of vectors in the k-program. The function associated with the vector $(i_m, \ldots, i_{m+k-1})$ maps

$$(a_{i_m}, \ldots, a_{i_{m+k-1}})$$

to

$$(s_{i_m}, \ldots, s_{i_{m+k-1}}) \bmod \cong .$$

This will allow us to check condition (iii) with a k-program over $(S^k)^*/\cong$. To check condition (i), choose $M \in \mathbf{V}$ such that $|M| \geqslant |S|^k$. (This can be done because $\mathbf{V}$ contains a nontrivial monoid and is closed under direct product.) There is thus a subset of M in one-to-one correspondence with S^k. To the first vector in the sequence of vectors given above we associate the function that maps $(a_{i_1}, \ldots, a_{i_k})$ to the element of M corresponding to $(s_{i_1}, \ldots, s_{i_k})$, and to the remaining vectors we associate the function

that maps every element of A^k to the identity of M. Condition (ii) is handled similarly. We thus have a family of k-programs over $((S^k)^*/\cong)\times M\times M\in\mathbf{V}$ that checks whether $w\in L$. Thus $L\in\widehat{\mathscr{P}}(\mathbf{V})$.

For the converse, suppose that L is recognized by a polynomial-size family of k-programs over a monoid $M\in\mathbf{V}$. Let T be the free $(k-1)$-definite semigroup over A. We show that L is recognized by a polynomial-size program over the wreath product $M\circ T$. The $(ik+j)$th instruction of the program $(i\geqslant 0,\ 1\leqslant j\leqslant k)$ consults the tth input letter, where t is the jth component of the $(i+1)$th vector in the k-program. If $j<k$ then the instruction emits

$$(I,a_t),$$

where I maps every element of $T\cup\{1\}$ to $1\in M$. If $j=k$, then the program emits

$$(f,a_t),$$

where $f(a_{i_1},\ldots,a_{i_{k-1}})$ is the value of the $(i+1)$th function of the program on the k-tuple $(a_{i_1},\ldots,a_{i_{k-1}},a_t)$. The set of accepting values of the program is

$$\{(F,v)\colon F(1)\in X\},$$

where X is the set of accepting values of the k-program. It is now easy to see that the resulting program over $M\circ T$ accepts the same language as the original k-program over M. □

Corollary 3.3. *For any variety of monoids* $\mathbf{V}$, $\mathscr{P}(\mathbf{V}*\mathbf{LI})$ *is closed under finite boolean operations, left and right quotients by a finite set of words, and p-reductions.*

Proof. For p-reductions, quotients and complements, the proof is the same as that of Theorem 3.1 given in [10]. It suffices to prove closure under intersection. For this, we take two languages $L_1,L_2\in\mathscr{P}(\mathbf{V}*\mathbf{LI})$ and use Theorem 3.2 to recognize these by k-programs Ψ_1,Ψ_2 over monoids $M_1,M_2\in\mathbf{V}$. (Observe that we can use the same k for both programs, since if $k<k'$, any language recognized by a k-program over M is recognized by a k'-program over M having the same length.) We now consider the k-program over $M_1\times M_2$ obtained by concatenating the two k-programs – when Ψ_1 emits a value in M_1, the second component of the instruction emits the identity in M_2, and when Ψ_2 emits a value in M_2, the first component of the instruction emits the identity of M_1. An input is accepted if and only if the value of the program on the input is (m_1,m_2), where m_1 is an accepting value for Ψ_1, and m_2 is an accepting value of Ψ_2. □

4. Regular languages and programs

Theorem 4.1. *Let* $\mathbf{V}$ *be a p-variety of monoids and* $L\subseteq A^*$ *be a language such that there is a neutral letter* $e\in A$ *for* L. *Then* $L\in\mathscr{P}(\mathbf{V})\cap Reg$ *if and only if* $M(L)\in\mathbf{V}$.

Proof. Every language L is m-recognized by $M(L)$, and thus p-recognized by $M(L)$. In particular, if $M(L) \in \mathbf{V}$ then $M(L)$ is finite, which implies that L is regular, and $L \in \mathscr{P}(\mathbf{V})$.

For the converse, suppose L is a regular language in $\mathscr{P}(\mathbf{V})$. Let $\eta : M(L)^* \to M(L)$ be the unique morphism extending the identity. It suffices to show that for each $Q \subseteq M(L)$, $\eta^{-1}(Q) \in \mathscr{P}(\mathbf{V})$, for in this case $M(L)$ p-divides a member of $\mathbf{V}$.

We first note that if e is a neutral letter for L, then $\eta_L(e)$ is the identity of $M(L)$. Second, the $\equiv_L$-class of a word is easily seen to be a finite boolean combination of languages of the form $u^{-1}Lv^{-1}$, where $u, v \in L$. Thus, $\eta_L^{-1}(Q)$ is a finite union of $\equiv_L$-classes, and hence a finite boolean combination of languages of the form described. Since $\mathbf{V}$ is a p-variety, by Theorem 3.1, $\eta_L^{-1}(Q) \in \mathscr{P}(\mathbf{V})$. For each $m \in M$, there exists $v_m \in A^*$ such that $\eta_L(v_m) = m$. Because of the presence of a neutral letter, and the fact that the neutral letter maps to the identity of $M(L)$, we can assume that all the v_m have the same length. Let $t \geqslant 1$ be the length of the v_m. Define a morphism $\phi : M(L)^* \to A^*$ by setting $\phi(m) = v_m$ for each $m \in M(L)$. Clearly, $\eta_L \circ \phi = \eta$ (here $\circ$ denotes the usual composition of morphisms). Thus $\eta^{-1}(Q) = \phi^{-1}\eta_L^{-1}(Q)$. Because of the condition on the lengths of v_w, this constitutes a p-reduction of $\eta^{-1}(Q)$ to $\eta_L^{-1}(Q)$: The program for inputs of length n has tn instructions; the $(it + j)$th instruction $(0 \leqslant i < n,\ 1 \leqslant j \leqslant t)$ reads the $(i + 1)$th input letter m and emits the jth letter of v_m. By Theorem 3.1, $\eta^{-1}(Q) \in \mathscr{P}(\mathbf{V})$. □

Theorem 4.2. *Let* $\mathbf{V}$ *be a monoid variety such that* $\mathbf{V} * \mathbf{LI}$ *is a p-variety, and let* $L \subseteq A^+$ *be a language. Then the following are equivalent*:

(a) $L \in \mathscr{P}(\mathbf{V} * \mathbf{LI}) \cap Reg$.

(b) $S(L)$ *is finite, and for all* $t \geqslant 1$, *every semigroup in* $\eta_L(A^t)$ *belongs to* $\mathbf{V} * \mathbf{LI}$.

(c) *There is some* $q \geqslant 1$ *such that* L *is recognized by a morphism* $\phi : A^+ \to S \circ \mathbb{Z}_q$, *where* $S \in \mathbf{V} * \mathbf{LI}$ *and for all* $a \in A$, *the projection of* $\phi(a)$ *onto* $\mathbb{Z}_q$ *is* 1 (*the generator of* $\mathbb{Z}_q$).

(d) L *is regular and is recognized by a single-scan program over some semigroup* $S \in \mathbf{V} * \mathbf{LI}$.

Proof. (a) *implies* (b): This is identical to the proof of the preceding theorem. In that proof we used the neutral letter to find a set of words in A^* all of the same length that maps onto $M(L)$. In the present instance we are given a set of words of length t that maps onto the subsemigroup S of $S(L)$.

(b) *implies* (c): The sets $\eta_L(A^k) = \eta_L(A)^k$, $k > 0$ form a semigroup under the usual product of subsets of a semigroup. Since this semigroup is finite, it contains an idempotent, and thus there is some $k > 0$ such that the set

$$S = \eta_L(A^k) = \eta_L(A^{2k}) = \cdots$$

is a subsemigroup. By assumption $S \in \mathbf{V} * \mathbf{LI}$, and thus S divides a wreath product $U = M \circ T$, where $M \in \mathbf{V}$ and $T \in \mathbf{LI}$.

We note some properties of T and U: As noted in Section 2, there exists $d > 0$ such that every element of T^d is idempotent. Thus, every element of U^d has the form (F, e), where $e \in T$ is idempotent. Now consider $(I, e) \in U$, where for all $t \in T \cup \{1\}$, $I(t) = 1$, the identity of M. We have

$$(I, e) \cdot (I, e) = (J, e),$$

where

$$J(t) = I(t) \cdot I(te) = 1 \cdot 1 = 1,$$

so $J = I$ and (I, e) is idempotent. Further,

$$(F, e) \cdot (I, e) = (G, e),$$

where

$$G(t) = F(t) \cdot I(te) = F(t),$$

so $G = F$. Thus, *every element of U^d is stabilized on the right by an idempotent of U.*

Before proceeding to the details of the proof, we shall try to provide a more intuitive notion of what is going on. We would like to show (but are not quite able to do so) that η_L factors through a morphism $\phi : A^+ \to U \circ K \circ \mathbb{Z}_q$, for some $q > 0$, where $K \in \mathbf{LI}$, and where for each $a \in A$, the projection of $\phi(a)$ in $\mathbb{Z}_q$ is 1. (We are writing the product in $\mathbb{Z}_q$ additively, so that 1 denotes the generator of $\mathbb{Z}_q$, and not the identity.) We choose $q \geqslant d$ to be a multiple of k, so that $\eta_L(A^q) = S$, and so that every element of U^q is stabilized on the right by an idempotent. We shall use the $\mathbb{Z}_q$ factor in the wreath product to count, modulo q, the number of letters that have been read, and we shall use the factor K to remember the last q letters that have been read. Every time the counter reaches 0, the input to the leftmost factor will be an element of U that maps to $\eta_L(v) \in S$, where v is the word consisting of the last q letters that have been read. When the counter says something different from 0, the input to the leftmost factor will be an idempotent that stabilizes the current state. The problem with this scheme is that while the first q letters of the input are being read there may be no idempotent that stabilizes the state of U. We solve this problem by keeping a separate copy of U for each prefix of length q, and by adjoining to K a factor that remembers the first q letters of the input, so that we are able to determine which of the copies will hold the correct value. We will also need to have K remember the last $2q$ letters that have been read, and not just the last q, in order to determine correctly a stabilizing idempotent.

Here are the details. There is a subsemigroup U' of U such that S is a quotient of U'; let $\theta : U' \to S$ denote the morphism onto S. For each $w \in A^q$ choose $\psi(w) \in U'$ such that $\theta(\psi(w)) = \eta_L(w)$. Let $\mathscr{U}$ denote the direct product of $|A^q|$ copies of the transformation semigroup U. If γ is a state or a transformation of $\mathscr{U}$, then the components of γ are denoted γ_w, where $w \in A^q$. Let K_1 denote the free $2q$-definite semigroup on A, and let K_2 denote the free q-reverse-definite semigroup on A. Let K be the direct product $K_1 \times K_2$. We claim that η_L factors through $\mathscr{U} \circ K \circ \mathbb{Z}_q$.

To show this we will produce a map Ξ from a subset of the set of states of the wreath product onto S^1, and for each $a \in A$ an element $\phi(a)$ of the wreath product, such that for all states p in the domain of Ξ,

$$\Xi(p\phi(a)) = \Xi(p)\eta_L(a).$$

The map ϕ extends to a morphism from A^* into the underlying semigroup of the wreath product, and the above equation readily implies that η_L factors through ϕ. Furthermore, each $\phi(a)$ will have the form

$$(F, f, 1),$$

where

$$f : \mathbb{Z}_q \to K$$

and

$$F : (K \cup \{1\}) \times \mathbb{Z}_q \to (U \times \cdots \times U)$$

are maps. This implies the condition on the projection of $\phi(a)$. $K \in \mathbf{LI}$, the product of semigroup varieties is associative, and $\mathbf{LI} * \mathbf{LI} = \mathbf{LI}$ (see [7]); thus the underlying semigroup of $\mathscr{U} \circ K$ is in $\mathbf{V} * \mathbf{LI}$.

It remains to define $\phi(a)$ and Ξ. $\phi(a) = (F, f, 1)$. The map f is constant; $f(r) = (a, a)$ for all $r \in \mathbb{Z}_q$. To define F, we consider three cases: If $|v| < q$ then we define

$$F(1, 0) = F((v, v), |v| - 1) = \gamma,$$

where for each $w \in A^q$, γ_w is an idempotent in U that stabilizes $\psi(w)$. If $|v| = q$, $m \neq q - 1$, and $|u| \geqslant m + q$, then

$$F((v, u), m) = \gamma,$$

where every component of γ is the idempotent e obtained as follows: $u = a_r \cdots a_0$. Let $z = a_{m+q-1} \cdots a_m$, and set e to be an idempotent that stabilizes $\psi(z)$ on the right. If $|v| = q$ and $m = q - 1$ then

$$F((v, u), m) = \gamma,$$

where every component of γ is $\psi(a_{q-1} \cdots a_0 a)$. The value of F can be set arbitrarily in all other cases. The domain of Ξ is the set of all triples (q_1, q_2, q_3) where $(q_2, q_3) \in (K \cup \{1\}) \times \mathbb{Z}_q$ has one of the forms discussed above. We set $\Xi(\alpha, 1, 0) = 1$. If $|v| < q$ then $\Xi(\alpha, (v, v), |v| - 1) = \eta_L(v)$. If $|v| = q$, and $u = a_r \cdots a_0$, then

$$\Xi(\alpha, (v, u), m) = \eta_L(v)\alpha_v\eta_L(a_{m-1} \cdots a_0).$$

It is straightforward, if a bit tedious, to verify case by case that if p is in the domain of Ξ, then so is $p \cdot \phi(a)$, and that $\Xi(p \cdot \phi(a)) = \Xi(p)\eta_L(a)$.

(c) *implies* (d): Suppose there is a morphism $\phi : A^+ \to S \circ \mathbb{Z}_q$ with $S \in \mathbf{V} * \mathbf{LI}$ and such that the projection onto $\mathbb{Z}_q$ of each letter goes to $1 \in \mathbb{Z}_q$.

We will note the projections from $S \circ \mathbb{Z}_q$ onto $S^{\mathbb{Z}_q}$ and $\mathbb{Z}_q$ by Π_1 and Π_2, respectively.

Given $(f,c) \in S \circ \mathbb{Z}_q$ (i.e. $f : \mathbb{Z}_q \to S$ and $c \in \mathbb{Z}_q$), we will show how to recognize $\phi^{-1}(f,c)$ with a single-scan program over S^q.

We construct a program Ψ_n for words of length n as follows: for $1 \leqslant i \leqslant n$ the ith instruction of Ψ_n is (i, α_i) where $\alpha_i : A \to S^q$ is given by

$$(\alpha_i(a))_j = (\Pi_1(\phi(a)))(i - 1 + j + c \bmod q).$$

Notice that the product in $S \circ \mathbb{Z}_q$ is given by $(f_1,c_1) \cdot (f_2,c_2) = (f, c_1 + c_2 \bmod q)$, where $f : \mathbb{Z}_q \to S$ is given by $f(j) = f_1(j) + f_2(c_1 + j)$. Thus, $w \in \phi^{-1}(f,c) \cap A^n$ if and only if $(\Psi_n(w))_j = f(n - 1 + j + c)$ for each $0 \leqslant j \leqslant q - 1$ and $c = n \bmod q$.

(d) *implies* (a): Immediate. □

5. Regular languages in circuit classes

Proposition 5.1. *Let $\mathscr{C}$ be a circuit class and* **V** *be a variety of finite monoids. Then $\mathscr{P}(\mathbf{V}) = \mathscr{C}$ implies $\mathscr{P}(\mathbf{V} * \mathbf{LI}) = \widehat{\mathscr{C}}$.*

Proof. By Theorem 3.2 programs over $\mathbf{V} * \mathbf{LI}$ are equivalent to k-programs over **V**. A single k-instruction can be computed by an NC^0 circuit and vice-versa. □

Theorem 4.2 gives us new proofs of all the known characterizations of the regular languages in the circuit classes AC^0, $CC^0(p)$ (for $p > 2$ prime) and $ACC^0(p)$ (for $p > 1$ prime).

Let **A** be the variety of aperiodic monoids. Then $\mathbf{A} * \mathbf{LI} = \mathbf{A_S}$, the variety of aperiodic semigroups.

Corollary 5.2 (Barrington). *The following are equivalent:*

(a) $L \in AC^0 \cap Reg$.

(b) *$S(L)$ is finite, and for all $t \geqslant 1$, every semigroup in $\eta_L(A^t)$ belongs to $\mathbf{A_S}$.*

(c) *There is some $q \geqslant 1$ such that L is recognized by a morphism $\phi : A^+ \to S * \mathbb{Z}_q$ where $S \in \mathbf{A_S}$ and for all $a \in A$, the projection onto $\mathbb{Z}_q$ of $\phi(a)$ is* 1.

(d) *L is regular, and is recognized by a single-scan program over some semigroup $S \in \mathbf{A_S}$.*

Proof. From [6], $AC^0 = \mathscr{P}(\mathbf{A})$, and from Proposition 5.1, $\mathscr{P}(\mathbf{A_S}) = \widehat{AC^0} = AC^0$. It follows from the result of Furst et al. [8] and Ajtai [1] that none of the word problems for $\mathbb{Z}_q$ is in AC^0. Thus, no nontrivial group has a word problem in AC^0, so $\mathbf{A_S}$ is a p-variety. The result follows from Theorem 4.2. □

If p is prime, then $\mathbf{G}_p$ denotes the variety of all finite p-groups.

Corollary 5.3 (Straubing et al. [16]). *Let $p > 2$ be a prime number, then the following are equivalent*:

(a) $L \in \mathrm{CC}^0(p) \cap \mathit{Reg}$.

(b) $S(L)$ *is finite, and for all* $t \geqslant 1$, *every semigroup in* $\eta_L(A^t)$ *belongs to* $\mathbf{G}_p * \mathbf{LI}$.

(c) *There is some* $q \geqslant 1$ *such that* L *is recognized by a morphism* $\phi : A^+ \to S \circ \mathbb{Z}_q$ *where* $S \in \mathbf{G}_p * \mathbf{LI}$ *and for all* $a \in A$, *the projection onto* $\mathbb{Z}_q$ *of* $\phi(a)$ *is* 1.

(d) L *is regular and is recognized by a single-scan program over some semigroup* $S \in \mathbf{G}_p * \mathbf{LI}$.

Proof. From [16], $\mathrm{CC}^0(p) = \mathscr{P}(\mathbf{G}_p)$, and from Proposition 5.1, $\mathscr{P}(\mathbf{G}_p * \mathbf{LI}) = \widehat{\mathrm{CC}^0}(p) = \mathrm{CC}^0(p)$. Neither an *AND* gate [5] nor a MOD_q gate for q not a power of p [16] can be simulated in $\mathrm{CC}^0(p)$. A semigroup belongs to $\mathbf{G}_p * \mathbf{LI}$ if and only if it contains no copy of the monoid $U_1 = \{0, 1\}$, nor any of the monoids $\mathbb{Z}_q$. Thus, no semigroup outside of $\mathbf{G} * \mathbf{LI}$ has its word problem p-recognized by a semigroup in this variety. Hence, $\mathbf{G}_p * \mathbf{LI}$ is a p-variety. The result follows from Theorem 4.2. When $p = 2$, languages in $\mathrm{CC}^0(2)$ are those p-recognized by $\mathbf{Z}_2$, and thus $\mathrm{CC}^0(2)$ is strictly contained in $\widehat{\mathrm{CC}^0}(2)$. □

If p is prime, then $\mathbf{M}_p$ denotes the variety of finite monoids in which every group is a p-group. Then $\mathbf{M}_p * \mathbf{LI} = \mathbf{S}_p$ the variety of all semigroups whose groups are p-groups.

Corollary 5.4 (Barrington et al. [3]). *Let p be prime. Then the following are equivalent*:

(a) $L \in \mathrm{ACC}^0(p) \cap \mathit{Reg}$.

(b) $S(L)$ *is finite, and for all* $t \geqslant 1$, *every semigroup in* $\eta_L(A^t)$ *belongs to* $\mathbf{S}_p$.

(c) *There is some* $q \geqslant 1$ *such that* L *is recognized by a morphism* $\phi : A^+ \to S \circ \mathbb{Z}_q$ *where* $S \in \mathbf{S}_p$ *and for all* $a \in A$, *the projection onto* $\mathbb{Z}_q$ *of* $\phi(a)$ *is* 1.

(d) L *is regular and is recognized by a single-scan program over some semigroup* $S \in \mathbf{S}_p$.

Proof. From [6], and the fact that every p-group is solvable, $\mathrm{ACC}^0(p) = \mathscr{P}(\mathbf{M}_p)$. From Proposition 5.1

$$\mathscr{P}(\mathbf{S}_p) = \widehat{\mathrm{ACC}^0}(p) = \mathrm{ACC}^0(p).$$

From Razborov [13] and Smolensky [14], a MOD_q gate cannot be simulated in $\mathrm{ACC}^0(p)$ if q is not a power of p. Thus no semigroup outside of $\mathbf{S}_p$ has its word problem p-recognized by a member of $\mathbf{S}_p$. Hence $\mathbf{S}_p$ is a p-variety. The result follows from Theorem 4.2. □

If our conjectures about the structure of CC^0 and ACC^0 are true then we can characterize the regular languages in these circuit complexity classes as well. Given $q > 1$, let $\mathbf{G}_{\mathrm{sol},q}$ be the variety of all solvable groups whose order divides a power of q.

Corollary 5.5 (Straubing [16]). *Let $q > 2$. If $AND \notin \mathrm{CC}^0(q)$ and for all q' relatively prime to q, $MOD_{q'} \notin \mathrm{CC}^0(q)$, then the following are equivalent.*

(a) $L \in \mathrm{CC}^0(q) \cap \mathit{Reg}$.

(b) *$S(L)$ is finite, and for all $t \geqslant 1$, every semigroup in $\eta_L(A^t)$ belongs to* $\mathbf{G}_{\mathrm{sol},q} * \mathbf{LI}$.

(c) *There is some $q'' \geqslant 1$ such that L is recognized by a morphism $\phi : A^+ \to S \circ \mathbb{Z}_{q''}$ where $S \in \mathbf{G}_{\mathrm{sol},q} * \mathbf{LI}$ and for all $a \in A$, the projection onto $\mathbb{Z}_{q''}$ of $\phi(a)$ is* 1.

(d) *L is regular and is recognized by a single-scan program over some semigroup $S \in \mathbf{G}_{\mathrm{sol},q} * \mathbf{LI}$.*

Proof. From [16], $\mathrm{CC}^0(q) = \mathscr{P}(\mathbf{G}_{\mathrm{sol},q})$, and from Proposition 5.1, $\mathscr{P}(\mathbf{G}_{\mathrm{sol},q} * \mathbf{LI}) = \widehat{\mathrm{CC}}^0(q) = \mathrm{CC}^0(q)$. A semigroup is in $\mathbf{G} * \mathbf{LI}$ if and only if it contains no copy of U_1, no cyclic group of order q', where q' does not divide a power of q, and no non-solvable group. Thus if a semigroup S outside of this variety has its word problems p-recognized by a member of the variety, we would either be able to simulate an *AND* gate or a $MOD_{q'}$ gate (contrary to assumption), or compute products in a finite non-solvable group. In this latter case, the theorem of Barrington [2] implies $\mathrm{NC}^1 \subseteq \mathrm{CC}^0(q)$, and in particular we would be able to simulate an *AND* gate. Thus, $\mathbf{G}_{\mathrm{sol},q} * \mathbf{LI}$ is a p-variety. The result follows from Theorem 4.2. □

Let $q > 0$, and let $\mathbf{M}_{\mathrm{sol},q}$ denote the variety consisting of all finite monoids in which every group belongs to $G_{\mathrm{sol},q}$. The variety of finite semigroups with the same property is denoted $\mathbf{S}_{\mathrm{sol},q}$. Barrington and Thérien proved [6] that $\mathscr{P}(\mathbf{M}_{\mathrm{sol},q}) = \mathrm{ACC}^0(q)$. As a consequence we have the following corollary, whose proof follows the same pattern as the two preceding ones.

Corollary 5.6 (Barrington et al. [3]). *If for all q' relatively prime to q, $MOD_{q'} \notin \mathrm{ACC}^0(q)$, then the following are equivalent.*

(a) $L \in \mathrm{ACC}^0(q) \cap \mathit{Reg}$.

(b) *$S(L)$ is finite, and for all $t \geqslant 1$, every semigroup $\eta_L(A^t)$ belongs to* $\mathbf{S}_{\mathrm{sol},q}$.

(c) *There is some $q'' \geqslant 1$ such that L is recognized by a morphism $\phi : A^+ \to S \circ \mathbb{Z}_{q''}$ where $S \in \mathbf{S}_{\mathrm{sol},q}$ and for all $a \in A$, the projection onto $\mathbb{Z}_{q''}$ of $\phi(a)$ is* 1.

(d) *L is regular and is recognized by a single-scan program over some semigroup $S \in \mathbf{S}_{\mathrm{sol},q}$.*

6. Conclusion

We have extended the notion of program from monoids to semigroups in varieties of the form $\mathbf{V} * \mathbf{LI}$, where $\mathbf{V}$ is a variety of monoids, showing that such varieties preserve natural closure properties such as closure under boolean operations.

This extension has allowed us to characterize exactly the regular languages recognized by programs over such semigroup varieties that are closed under p-division. We obtained as a consequence characterizations of the regular languages recognized by certain circuit classes which were previously given by Barrington et al. [3] and Straubing [16].

Although the methods employed in the proof of Theorem 4.2 allowed us to give a single proof of these characterizations, we are still unable to give direct algebraic proofs of separation results for these circuit classes. This remains one of the principal concerns of this field of study.

References

[1] M. Ajtai, Σ_1^1 formulae on finite structures, *Ann. Pure Appl. Logic* **24** (1983) 1–48.

[2] D.A.M. Barrington, Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1, *J. Comput. Systems Sci.* **38** (1989) 150–164.

[3] D.A.M. Barrington, K. Compton, H. Straubing and D. Thérien, Regular languages in NC^1, *J. Comput. Systems Sci.* **44** (1992) 478–499.

[4] D.A.M. Barrington and H. Straubing, Superlinear lower bounds for bounded-width branching programs, in: *Proc. 6th IEEE Conf. on Structure in Complexity Theory* (1991) 305–314.

[5] D.A.M. Barrington, H. Straubing and D. Thérien, Non-uniform automata over groups, *Inform. Comput.* **89** (1990) 109–132.

[6] D.A.M. Barrington and D. Thérien, Finite monoids and the fine structure of NC^1, *J. Assoc. Comput. Mach.* **35** (1988) 941–952.

[7] S. Eilenberg, *Automata, Languages and Machines* (Academic Press, New York, 1974 (vol. A), 1976 (vol. B)).

[8] M. Furst, J. Saxe and M. Sipser, Parity, circuits and the polynomial-time hierarchy, *Math. Systems Theory* **17** (1984) 13–27.

[9] V. Grolmusz, Separating the communication complexity of mod p and mod m circuits, in: *Proc. 33rd IEEE Ann. Symp. Foundations of Computer Science* (1992) 278–287.

[10] P. McKenzie, P. Péladeau and D. Thérien, NC^1: the automata-theoretic viewpoint, *Comput. Complexity* **1** (1991) 330–359.

[11] P. Péladeau, Classes de circuits booléens et variétés de monoïdes, Ph. D. Thesis, Université Paris VI, 1990.

[12] J.E. Pin, *Varieties of Formal Languages* (Plenum, London, 1986).

[13] A.A. Razborov, Lower bounds for the size of circuits of bounded depth with basis $\{\&, \oplus\}$, *Math. Zametki* **41** (1987) 598–607 (in Russian). English translation *Math. Notes Academy Sci. USSR* **41** (1987) 333–338.

[14] R. Smolensky, Algebraic methods in the theory of lower bounds for boolean circuit complexity, in: *Proc. 19th Ann. ACM Symp. Theory of Computing* (1987) 77–82.

[15] H. Straubing, Finite semigroup varieties of the form $\mathbf{V} * \mathbf{D}$, *J. Pure Appl. Algebra* **36** (1985) 53–94.

[16] H. Straubing, Constant-depth periodic circuits, *Internat. J. Algebra Comput.* **1** (1991) 49–88.

[17] M. Szegedy, Functions with bounded symmetric communication complexity and circuits with MOD m gates, in: *Proc. 22nd Ann. ACM Symp. Theory of Computing* (1990) 278–286.

[18] A.C. Yao, On ACC and threshold circuits, in: *Proc. 31st IEEE Ann. Symp. Foundations of Computer Science* (1990) 619–627.