JOURNAL OF COMPUTER AND SYSTEM SCIENCES **44**, 478–499 (1992)

Regular Languages in NC^1

DAVID A. MIX BARRINGTON*

Department of Computer and Information Science, The University of Massachusetts, Amherst, Massachusetts 01003

KEVIN COMPTON†

Department of Computer Science, The University of Michigan, Ann Arbor, Michigan 48109

HOWARD STRAUBING‡

Computer Science Department, Boston College, Chestnut Hill, Massachusetts 02167

AND

DENIS THÉRIEN§

School of Computer Science, McGill University, Montréal, Québec, Canada H3A 2K6

Received June 2, 1988; revised May 1, 1990

We give several characterizations, in terms of formal logic, semigroup theory, and operations on languages, of the regular languages in the circuit complexity class AC^0, thus answering a question of Chandra, Fortune, and Lipton. As a by-product, we are able to determine effectively whether a given regular language is in AC^0 and to solve in part an open problem originally posed by McNaughton. Using recent lower-bound results of Razborov and Smolensky, we obtain similar characterizations of the family of regular languages recognized by constant-depth circuit families that include unbounded fan-in mod p addition gates for a fixed prime p along with unbounded fan-in boolean gates. We also obtain logical characterizations for the class of all languages recognized by nonuniform circuit families in which mod m gates (where m is not necessarily prime) are permitted. Comparison of this characterization with our previous results provides evidence for a conjecture concerning the regular languages in this class. A proof of this conjecture would show that computing the bit sum modulo p, where p is a prime not dividing m, is not AC^0-reducible to addition mod m, and thus that *MAJORITY* is not AC^0-reducible to addition mod m. © 1992 Academic Press, Inc.

* Research supported by National Science Foundation Grant CCR-8714714.

† Research supported by National Science Foundation Grant DCR-8605358.

‡ Research supported by National Science Foundation Grants CCR-8700700 and CCR-8902369.

§ Research supported by a grant from the National Science and Engineering Research Council of Canada.

0022-0000/92 $5.00

1. Introduction

Characterization of the Regular Languages in Subclasses of NC^1

In this paper we consider the regular languages contained in various subclasses of the nonuniform circuit complexity class NC^1, which consists of the languages recognized by $O(\log n)$-depth bounded fan-in boolean circuit families. This study was motivated by a question raised in a paper of Chandra, Fortune, and Lipton [8]: Which regular languages can be recognized by constant-depth unbounded fan-in circuits of polynomial size? In other words, what are the regular languages in the nonuniform circuit complexity class AC^0? In Section 5, we provide a complete answer to this question. Indeed, we give several characterizations of this class of regular languages in terms of operations on languages, first-order logic, and computable semigroup-theoretic invariants of regular languages. As a by-product, we show that it is decidable whether a given regular language belongs to AC^0.

The same methods are used in Section 6 to characterize the regular languages that are AC^0-reducible to addition modulo p, where p is prime (that is, the regular languages in the circuit complexity class $ACC(p)$). We also identify a class of regular languages (those whose syntactic monoids contain a nonsolvable group) that are complete for NC^1 under constant-depth reductions. These constitute all the complete regular languages unless there is some integer q such that adding bits modulo q is complete.

These theorems depend on a number of lower-bound results in circuit complexity: For the characterization of the regular languages in AC^0, we require the theorem (due to Ajtai [1], and Furst, Saxe, and Sipser [11]) on circuits that add bits modulo m, and for the characterizations in Section 6 we need Smolensky's generalization of this result [21]. Our theorem on the regular languages complete for NC^1 uses a result of Barrington [2].

Nonuniform versus Uniform Circuit Complexity

The characterization of the regular languages in these circuit complexity classes may appear to address a very narrow question. We believe, however, that it provides an important clue to the structure of NC^1 and may help us find new lower bounds results. We give the detailed reasons for this belief in Section 8—here is a brief summary: Immerman [13] has shown that AC^0 consists of precisely those languages definable by first-order sentences of a certain kind. In these sentences, variables are interpreted as positions in a bit string. There is a unary predicate π, where $\pi(x)$ is interpreted to mean "the bit in position x is on," and what we call *numerical predicates* of arbitrary arity—satisfaction of such a predicate depends only on the positions associated to the variables and on the length of the string in which the predicate is interpreted, not on what bits appear in these positions. Our results in Section 5 imply that the regular languages in AC^0 are definable by sentences in which all the numerical predicates have the form

$$x \equiv 0 \quad (\text{mod } q),$$

and

$$x < y.$$

This can be viewed as a connection between nonuniform and uniform complexity, reminiscent of the results of Karp and Lipton [14] on polynomial-size circuits: The nonuniform class AC^0 contains uncountably many languages, and thus some nonrecursive ones. Nonetheless, this nonuniformity plays no role in the recognition of highly structured (in this case, regular) languages. Indeed, any regular language in AC^0 can be recognized by a highly uniform AC^0 circuit family.

There is a natural "explanation" for this phenomenon: Numerical predicates used to define a regular language can always be replaced by numerical predicates that are recognized, in a sense that can be made precise, by a finite automaton. As simple as this principle sounds, we have no qualitative proof of it. We know that it is true in the context of first-order sentences, but the only proof we can provide uses the lower-bound results of [1] and [11]. We also know that the principle holds when we permit the use of a new kind of quantifer—a "modular quantifer" of prime modulus. We are thus led to conjecture the principle in general, for both first-order sentences and those built with modular quantifiers, irrespective of the modulus. If this conjecture is true, we will obtain as a consequence a new lower-bound theorem, showing that *MAJORITY* is not in *ACC*, and new, algebraic proofs of the known lower bounds.

It is interesting to note that the question of the expressive power of first-order sentences with arbitrary numerical predicates was first raised by McNaughton, in 1960 (reported in McNaughton and Papert [16]). Our results in Section 8 prove, in part, one of McNaughton's conjectures. (More precisely, we have proved that every regular language defined by a first-order sentence with arbitrary numerical predicates is in the class that McNaughton calls FOL_C. In fact, McNaughton considers an expanded class of first-order formulas and conjectures that the regular languages definable in this class belong to a family of languages he names FOL_{UC}. Our methods do not appear to apply this larger class.)

Organization of the Article

In Section 2 we give our general terminology concerning circuit complexity classes, reductions, and first-order logic.

Everything we do in this paper requires extensive use of the theory of the *syntactic monoid* of a regular language. In Section 3, we have collected all the results of this theory that we use in the sequel. It should be noted that the results on AC^0 can be read without the material on languages recognized by wreath products and triple products.

In Section 4 we give a constant-depth reduction of multiplication in a finite monoid M to the problem of recognizing a regular language L having M as a quotient of a submonoid of its syntactic monoid. As the existence of this reduction requires a technical condition on the syntactic morphism of L, the precise statement of our reduction lemma (Lemma 2) is somewhat complicated. Lemma 2 is used in the proofs of the principal results of Sections 5, 6, and 7.

Theorem 3 of Section 5 gives the promised characterizations of the regular languages in AC^0, and Theorem 5 of Section 6 does the same for $ACC(p)$ with p prime. In Section 7 we use Lemma 2 and Barrington's results [2] on branching programs of bounded width to show that any regular language whose syntactic monoid contains a nonsolvable group is complete for NC^1 with respect to constant-depth reductions. Our conjectures of Section 8 would imply that these are the only complete regular languages.

In Section 8 we discuss, as mentioned above, the connections between nonuniform and uniform logical descriptions of the regular languages in these complexity classes. We state the conjecture suggested by this discussion and, using the algebraic machinery of Section 3, derive the consequences of these conjectures. Section 9 lists some open questions suggested by our results.

2. Notation and Terminology

Circuit Complexity

For more on the circuit complexity classes and reductions defined here, see [11, 9, 2].

AC^0 denotes the family of subsets L of $\{0, 1\}^*$ for which there exists a sequence $\{C_n\}$ of unbounded fan-in boolean circuits such that C_n has n inputs and accepts the set $L \cap \{0, 1\}^n$, the depth of C_n is constant (that is, independent of n), and the number of gates of C_n is bounded by a polynomial in n.

Let $q \geqslant 1$. $ACC(q)$ denotes the family of subsets of $\{0, 1\}^*$ accepted by such circuits where we permit, along with the boolean gates, $MOD(q)$ gates, which emit 1 if the number of inputs that are on is divisible by q, and 0 otherwise. Observe that $ACC(1) = AC^0$. ACC denotes the union of the $ACC(q)$ over all q.

NC^1 denotes the family of subsets accepted by bounded fan-in families of boolean circuits with depth $O(\log n)$. It is easy to see that for any $q > 1$, $AC^0 \subseteq ACC(q) \subseteq ACC \subseteq NC^1$. As indicated in the introduction, it is known that the first of these inclusions is strict, and that the second is strict if q is prime. We believe that all three inclusions are strict for all values of q; evidence for this conjecture is discussed in Sections 8 and 9.

These circuit complexity classes as defined here are *nonuniform*; we impose no condition on our circuit families, other than that they satisfy the required size and depth bounds. As it turns out, this makes no difference to our conclusions. Indeed, our results can be interpreted as saying that any regular language recognized by a nonuniform circuit family of one of these types is also recognized by a similar family satisfying a very strong uniformity condition. (See [3].)

Reductions

Let A and B be finite alphabets, and let $K \subseteq A^*$ and $L \subseteq B^*$. As we wish to discuss recognition of these languages by boolean circuits, we assume that each letter of A is coded by a fixed-length string of bits, and similarly for B. We can build circuits that contain, in addition to the usual boolean gates, oracle gates for the language L. An n-input L-gate has $r \cdot n$ boolean inputs, where r is the number of bits used to encode an element of B; the gate emits a 1 if the encoded element of B^n is in L, and 0 otherwise. The size of circuit containing such gates is defined to be the sum, over all boolean and L-gates, of the number of input wires to each gate. K is said to be *AC^0-reducible* to L if K is recognized by a constant-depth polynomial-size family of such circuits. K is said to be *constant-depth reducible* to L if there is such a family in which no circuit contains a path from the output of one L-gate to the input of another.

Special Languages

We emply a special notation for certain languages in $\{0, 1\}^*$. If A is any finite alphabet, $w \in A^*$ and $a \in A$, then we denote by $|w|$ the length of w and by $|w|_a$ the number of a's in w. We then define:

$$LENGTH(q) = \{w \in \{0, 1\}^* \mid |w| \equiv 0 \pmod q\}$$
$$SUM(q) = \{w \in \{0, 1\}^* \mid |w|_1 \equiv 0 \pmod q\}$$
$$MAJORITY = \{w \in \{0, 1\}^* \mid |w|_1 > |w|_0\}.$$

First-Order Logic and Extensions

See [24] for a discussion of definability of regular languages in extensions of first-order logic.

We employ a logical apparatus for defining subsets of A^*, where A is a finite alphabet. Our basic language is the first-order language with a binary relation $<$ and, for each $a \in A$, a unary predicate Q_a. We interpret variables as positions in words over A, so that $x < y$ means that position x is to the left of position y, and $Q_a x$ means that the letter in position x is a. A first-order sentence then defines the set of words in A^* that satisfy the sentence. We denote by **FO** the family of all such subsets of A^*.

Let us adjoin to our language the unary predicates C_d^r, where $C_d^r x$ is interpreted to mean $x \equiv r \pmod d$. **FO[C]** denotes the family of subsets of A^* definable in this extended first-order language.

More generally, we can admit as atomic formulas relation symbols R of arbitrary arity. We interpret these so that the truth of $R(x_1, ..., x_k)$ depends only on the positions x_i and the length of the word w, and not on the letters in those positions. Such an R will be called a *numerical predicate.* We denote by **FO[R]** the class of languages definable by such formulas. Obviously $\mathbf{FO} \subseteq \mathbf{FO[C]} \subseteq \mathbf{FO[R]}$.

We shall also consider generalized quantifiers $\exists_q^r$ where $\exists_q^r x \psi(x)$ means that the number of positions satisfying ψ is congruent to r modulo q. $(\mathbf{FO} + \mathbf{MOD}(q))$ denotes the family of languages defined using both ordinary quantifiers and such modular quantifiers, with modulus q (but without the predicates C_d^r), $(\mathbf{FO} + \mathbf{MOD}(q))$ $[\mathbf{C}]$ denotes the family of languages defined by further admitting the predicates C_d^r, and $(\mathbf{FO}+\mathbf{MOD}(q))\,[\mathbf{R}]$ the family of languages defined by admitting arbitrary numerical predicates. $(\mathbf{FO}+\mathbf{MOD})$, $(\mathbf{FO}+\mathbf{MOD})\,[\mathbf{C}]$, and $(\mathbf{FO}+\mathbf{MOD})\,[\mathbf{R}]$ represent the unions of these families over all moduli q.

3. Algebraic Preliminaries

For more information on the matters discussed in this section, see the books by Eilenberg [10], Lallement [15], and Pin [17].

Syntatic Monoid and Syntactic Morphism

A *monoid* is a semigroup with an identity element. The set A^* of all words over an alphabet A is then the free monoid generated by A, with the empty word (which we denote by 1) as the identity.

Let A be a finite alphabet and let L be a subset of A^*. Let M be a monoid. We say that *M recognizes L* if there is a morphism $\phi: A^* \to M$ such that $L = X\phi^{-1}$ for some $X \subseteq M$. We also say in such an instance that the morphism ϕ recognizes L. It is not difficult to see that recognition by a *finite* monoid is equivalent to recognition by a finite automaton, so the regular languages in A^* are exactly those recognized by finite monoids.

Two words w_1 and w_2 are said to be *congruent mod L*, written $w_1 \simeq_L w_2$, if for all $u, v \in A^*$,

$$uw_1v \in L \qquad \text{if and only if} \qquad uw_2v \in L.$$

The equivalence relation $\simeq_L$ is a congruence on A^*. The quotient monoid $A^*/\simeq_L$ is denoted $M(L)$ and is called the *syntactic monoid* of L. The projection morphism $\eta_L: A^* \to M(L)$ is called the *syntactic morphism* of L. A morphism $\phi: A^* \to M$ recognizes L if and only if there is a morphism θ such that Diagram 1 is commutative. In particular, L is regular if and only if $M(L)$ is finite.

It is possible to give an equivalent definition of the syntactic monoid and syntactic morphism in terms of automata. In a complete deterministic automaton every

$$\begin{tikzcd}
& A^* \arrow[dl, "\phi"'] \arrow[dr, "\eta_L"] & \\
\mathrm{Im}(\phi) \arrow[rr, "\theta"] & & M(L)
\end{tikzcd}$$

Diagram 1

word w induces the mapping $f_w\colon q \mapsto q \cdot w$ on the set of states. These mappings form a monoid M under composition, and the function ϕ taking w to f_w is a morphism from A^* to M, provided that we compose mappings in left-to-right order. If we do this with the minimal automaton of L, then M is the syntactic monoid of L and ϕ the syntactic morphism.

Wreath Product

Let M and N be monoids. The *wreath product* M**wr**N consists of all pairs (f, n) where $f \in M^N$ and $n \in N$. A multiplication in M**wr**N is defined by

$$(f_1, n_1) \cdot (f_2, n_2) = (g, n_1 n_2),$$

where for $n \in N$, $g(n) = f_1(n) f_2(nn_1)$. This product is associative, and the pair $(I, 1)$, where $I(n) = 1$ for all $n \in N$, is the identity. Thus M**wr**N is a monoid, and the projection $\pi\colon M\mathbf{wr}N \to N$ onto the right-hand coordinate is a morphism.

Relational Morphisms and the Derived Semigroup

We consider Diagram 2, where M and N are monoids and ϕ and η are morphisms, with ϕ onto. The relation $\phi^{-1}\eta\colon M \to N$, that is, the set

$$\phi^{-1}\eta = \{(m, n) \in M \times N \mid w\phi = m,\ w\eta = n \quad \text{for some} \quad w \in A^*\},$$

is called a *relational morphism*. The *derived semigroup* D of the relational morphism consists of a zero, together with the triples (n, m, n'), where $n, n' \in N$, $m \in M$, and for some $v, w \in A^*$, $v\eta = n$, $w\phi = m$, and $(vw)\eta = n'$. Multiplication is given by

$$(n_1, m_1, n_1') \cdot (n_2, m_2, n_2') = \begin{cases} (n_1, m_1 m_2, n_2'), & \text{if } n_1' = n_2; \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to check that this multiplication is well-defined and associative, and thus D is a semigroup, which is finite in case M and N are both finite. In certain instances it may be preferable to have D be a monoid, which can be accomplished by adjoining an identity element to D. (Recent research indicates that the best alternative is to use a modified version of D that is not a semigroup at all, but a

$$\begin{tikzcd} & A^* \arrow[dl, "\phi"'] \arrow[dr, "\eta"] & \\ M & & N \end{tikzcd}$$

DIAGRAM 2

$$\begin{tikzcd}
 & A^* \arrow[dl, "\phi"'] \arrow[d, "\psi"] \arrow[drr, "\eta"] & & \\
M & M' \arrow[l, "\theta"] \arrow[r, phantom, "\subseteq"] & D \,\mathrm{wr}\, N \arrow[r, "\pi"'] & N
\end{tikzcd}$$

DIAGRAM 3

category. See Tilson [27] for this approach. The somewhat cruder formalism used here is sufficient for our purposes.)

The principal property of the derived semigroup is the following: There is a commutative diagram (Diagram 3) defined by setting, for each $a \in A$, $a\psi = (F_a, a\eta)$, where for all $n \in N$, $F_a(n) = (n, a\phi, n \cdot a\eta)$.

Languages Recognized by Wreath Products

In Straubing [22] the following operation on languages is introduced. Let $L \subseteq A^*$, $a \in A$, $0 \leqslant r < d$. Then $\langle La, r, d \rangle$ consists of all $w \in A^*$ such that the number of prefixes of w in La is congruent to r modulo d. This is used in conjunction with the study of languages recognized by wreath products of the form $M_1 \mathbf{wr} M_2$, where M_1 and M_2 are finite and M_1 contains no nonsolvable groups. While the results in [22] refer only to the monoids recognizing these languages, the same arguments can be used to obtain the following sharper results, which concern recognition by morphisms:

PROPOSITION 1. *Let M_1, M_2 be finite monoids. Let $\psi: A^* \to M_1 \mathbf{wr} M_2$ be a morphism, and $\pi: M_1 \mathbf{wr} M_2 \to M_2$ the projection morphism.*

(a) *If M_1 is aperiodic (i.e., contains no nontrivial groups), then every language recognized by ψ is in the smallest family of subsets of A^* containing the letters and the languages recognized by $\psi\pi$ and closed under boolean operations and concatenation.*

(b) *If M_1 contains no nonsolvable groups, then every language recognized by ψ is in the smallest family of subsets of A^* containing the letters and the languages recognized by $\psi\pi$ and closed under boolean operations, concatenation, and the operations $L \to \langle La, r, d \rangle$, where d is a divisor of the order of a group in M_1.*

Concatenation and Triple Products

Eilenberg [10] studied a bilateral semidirect product, or *triple product* (S_1, T, S_2), where S_1, S_2, and T are finite monoids. It is shown there that there is a morphism

$$\pi: (S_1, T, S_2) \to S_1 \times S_2$$

$$\begin{tikzcd}
 & A^{*} \arrow[dl, "\eta_{L_1 L_2}"'] \arrow[dr] & \\
M(L_1 L_2) & & \mathrm{Im}(\phi) \arrow[ll, "\theta"']
\end{tikzcd}$$

DIAGRAM 4

with the following property: If G is a group in (S_1, T, S_2) then the kernel of the morphism $\pi: G \to G\pi$ is isomorphic to a group in T. In particular, if T contains no subgroup of order p, where p is prime, then π is injective when restricted to groups of order p.

It is also shown in Eilenberg's book that for regular languages $L_1, L_2 \subseteq A^*$, there is a triple product $M = (M(L_1), T, M(L_2))$, where T is the monoid of all subsets of $M(L_1) \times M(L_2)$ with union as the operation and a morphism $\phi: A^* \to M$ such that Diagram 4 is commutative, and θ is surjective. (M is the *Schützenberger product* of L_1 and L_2.) Similar remarks apply to the operation

$$(L_1, L_2) \mapsto L_1 *_q L_2,$$

where $L_1 *_q L_2$ consists of all $w \in A^*$ such that the number of factorizations $w = uv$ with $u \in L_1$, $v \in L_2$ is divisible by q. In this case the monoid T in the triple product consists of the set of all maps $f: M(L_1) \times M(L_2) \to \mathbf{Z}_q$ with pointwise multiplication as the operation—in particular T is a group of exponent q. Observe that in the first case, the morphism π is injective when restricted to groups in the triple product, and in the second case it is injective when restricted to groups of prime order p not dividing q.

Suppose we have a commutative diagram (Diagram 5), where θ, θ' are surjective, and β, β' are injective on groups of order p with p prime. Let

$$K = \{(m', m'') \in M' \times M'' \mid m'\beta = m''\theta'\}.$$

$$\begin{tikzcd}
 & & A^{*} \arrow[dll] \arrow[dl] \arrow[d] \arrow[dr] \arrow[drr] & & \\
N & M' \arrow[l, "\theta"'] \arrow[r, "\beta"'] & N' & M'' \arrow[l, "\theta'"'] \arrow[r, "\beta'"'] & M_1
\end{tikzcd}$$

DIAGRAM 5

The morphism $K \to N^1$ taking (m', m'') to $m'\theta$ is surjective, and it is not difficult to show that the morphism taking (m', m'') to $m''\beta'$ is injective on groups of order p contained in K. (See [23].) The property "injective on groups of order p" is thus closed under a kind of composition of diagrams. We use this fact in Section 8.

4. A Reduction Lemma

We are concerned in this paper with characterizing the circuit complexity of regular languages in terms of the algebraic invariants of these languages. We wish to be able to say something like: If L_1 has a "simpler" syntactic monoid than L_2, then L_1 is reducible to L_2 under some appropriate notion of reducibility that preserves membership in our complexity classes. Unfortunately this is false. (For example, let $L_1 = LENGTH(4)$ and $L_2 = SUM(2)$. Then $M(L_1)$ is the cyclic group of order 4 and $M(L_2)$ is the cyclic group of order 2. However $L_1 \in AC^0$ and $L_2 \notin AC^0$.) We do get constant-depth reducibility if the syntactic morphism of L_2 factors through the syntactic morphism of L_1. The following lemma is a stronger version of this fact.

Lemma 2. *Let A and B be finite alphabets, and let $L_1 \subseteq A^*$, $L_2 \subseteq B^*$ be regular languages. Suppose that L_2 is recognized by a morphism $\delta: B^* \to N$, where N is a quotient, via a morphism α, of a submonoid M' of $M(L_1)$. Suppose further that there is a positive integer t such that for every $b \in B$ there exists $w \in A^t$ such that $w\eta_{L_1}\alpha = b\delta$. Then L_2 is constant-depth reducible to L_1.*

Proof. Let $u, v \in A^*$. We define, as in [10],

$$u^{-1}L_1v^{-1} = \{w \in A^* \mid uwv \in L_1\}.$$

It follows from the finiteness of $M(L_1)$ that there are only finitely many different languages $u^{-1}L_1v^{-1}$. Choose $u_1, v_1, ..., u_r, v_r$ so as to obtain the complete set of such languages. Two words w_1, w_2 have different images in $M(L)$ if and only if there exists $i \in \{1, ..., r\}$ such that $u_i w_1 v_i \in L$ and $u_i w_2 v_i \notin L$, or vice versa.

We now show how to construct a constant-depth, $O(n)$-size circuit with L_1-gates that accepts the strings of length n in L_2.

Let $b_1, ..., b_n \in B$ be the inputs. For each $b \in B$ there is a word $w_b \in A^*$ of length t such that $w_b\phi\eta_L = b\delta$. The inputs $b_2, ..., b_{n-1}$ are fed into fixed-size boolean circuits that compute the encoding $b \mapsto w_b$. The input b_1 is fed into r fixed-size circuits that produce the outputs $u_1 w_{b_1}, ..., u_r w_{b_1}$, and b_n is fed into r fixed-size circuits that produce $w_{b_n}v_1, ..., w_{b_n}v_r$. The results are fed into r separate circuit elements for testing membership in L_1—the ith element tests inputs of length $tn + |u_i v_i|$ and is used to determine if $u_i w_{b_1} \cdots w_{b_n} v_i$ is in L_1. The result in $\{0, 1\}^r$ completely determines $(w_{b_1} \cdots w_{b_n})\phi$ (and hence $(b_1 \cdots b_n)\delta$). We can now feed this result into a fixed-size circuit with r inputs and one output, to determine if $b_1 \cdots b_n$ is in L_2. ∎

5. Regular Languages in AC^0

The *star-free* languages in A^* are those subsets obtained, beginning with the letters of A, by repeated application of boolean operations and concatenation. Evidently, these languages are all regular. There are two important characterizations of the star-free languages in the literature. The first, due to Schützenberger [19], is in terms of the syntactic monoid

$$L \subseteq A^* \quad \text{is star-free if and only if} \quad M(L) \quad \text{is aperiodic;}$$

that is, $M(L)$ contains no nontrivial groups. The second, due to McNaughton [16], is in terms of first-order definability:

$$L \subseteq A^* \quad \text{is star-free if and only if} \quad L \in \mathbf{FO}.$$

It is easy to show that AC^0 is closed under boolean operations and concatenation and therefore contains all the star-free subsets of $\{0, 1\}^*$. AC^0 also contains some regular languages that are not star-free, for example all the languages $LENGTH(q)$, where $q > 1$. (We know these are not star-free because the syntactic monoid of $LENGTH(q)$ is $\mathbf{Z}_q$, a nontrivial group.) It is shown in [1, 11] that $SUM(q)$ is not in AC^0. Since $SUM(q)$ and $LENGTH(q)$ have isomorphic syntactic monoids, we will not be able to obtain a characterization of the regular languages in AC^0 in terms of the syntactic monoid alone. We do, however, obtain a characterization in terms of the syntactic morphism, and another in terms of first-order definability.

Theorem 3. *Let $L \subseteq \{0, 1\}^*$. The following are equivalent:*

(a) *L is regular and $L \in AC^0$,*

(b) *$L \in \mathbf{FO}[\mathbf{C}]$.*

(c) *L belongs to the smallest family of subsets of $\{0, 1\}^*$ that contains $\{0\}$, $\{1\}$ and the languages $LENGTH(q)$ for $q > 1$, and that is closed under boolean operations and concatenation.*

(d) *L is recognized by a morphism $\psi \colon \{0, 1\}^* \to M \mathbf{wr} \mathbf{Z}_r$, where M is a finite aperiodic monoid, and where the composition $\psi\pi \colon \{0, 1\}^* \to \mathbf{Z}_r$ takes both 0 and 1 to the generator 1 of $\mathbf{Z}_r$.*

(e) *$M(L)$ is finite, and for each $t \geqslant 0$, the image of $\{0, 1\}^t$ under the syntactic morphism η_L contains no nontrivial group.*

Proof. (a) $\Rightarrow$ (e). Since L is regular, $M(L)$ is finite. Suppose that for some $t \geqslant 0$, the image of $\{0, 1\}^t$ under η_L contains a nontrivial group G. Then G has an element m of order $k > 0$. Let $M' = \{m, m^2, ..., m^k\}$ be the subgroup generated by m. We see that m^k is the identity element of M'. Choose $u, v \in \{0, 1\}^t$ so that $u\eta_L = m^k$,

$$\begin{tikzcd} & \{0,1\}^* \arrow[dl, "\eta_L"'] \arrow[dr, "\zeta"] & \\ M(L) & & \mathbf{Z}_r \end{tikzcd}$$

DIAGRAM 6

$v\eta_L = m$. Define a morphism $\delta\colon \{0, 1\}^* \to M'$ by $0\delta = m^k$, $1\delta = m$. Then $u\eta_L = 0\delta$ and $v\eta_L = 1\delta$, so by Lemma 2, $SUM(k) = 0\delta^{-1}$ is reducible to L and is thus in AC^0, a contradiction.

(e) $\Rightarrow$ (d). For each nontrivial group G contained in $M = M(L)$ pick a nonempty word v_G such that $v_G\eta_L$ is the identity of G. Let r be a common multiple of the lengths of all these v_G. Consider the relational morphism defined by Diagram 6, where $0\zeta = 1\zeta = 1 \in \mathbf{Z}_r$. We show that the derived semigroup D of this relational morphism is aperiodic. By the remarks in Section 3 concerning relational morphisms and the derived semigroup, L is recognized by a morphism $\psi\colon \{0, 1\}^* \to D\mathbf{wr}\mathbf{Z}_r$. Suppose D contains a nontrivial group. This group cannot contain the zero of D, so by the definition of multiplication in D, there is an $n \in \mathbf{Z}_r$ such that all the elements in the group are of the form (n, m, n). Let (n, m, n) be one such element and suppose it has order $k > 1$. Then $G = \{m, m^2, ..., m^k\}$ is a nontrivial group in M. Choose $v \in \{0, 1\}^*$ such that $v\eta_L = m$. It follows from the definition of the derived semigroup that $v\zeta = 0$, and thus $|v|$ is a multiple of $|v_G|$. There is thus some power w of v_G such that v and w have the same length. Let $t = k|v|$. For $1 \leqslant j \leqslant k$, the string $v^j w^{k-j}$ has length t. But $(v^j w^{k-j})\eta_L = m^j$, so the image of $\{0, 1\}^t$ under η_L contains G, a contradiction.

(d) $\Rightarrow$ (c). The languages recognized by the morphism $\zeta\colon \{0, 1\}^* \to \mathbf{Z}_r$ that takes both 0 and 1 to $1 \in \mathbf{Z}_r$ are unions of languages of the form $\{w \mid |w| \equiv s \pmod{r}\} = LENGTH(r) \cdot \{0, 1\}^s$. The conclusion now follows at once from Proposition 1.

(c) $\Rightarrow$ (b). The language $LENGTH(d)$ is defined by the sentence $\exists x(\forall y(y \leqslant x) \wedge C_d^0 x)$, and is consequently in $\mathbf{FO}[\mathbf{C}]$. It is easy to show (see, for example, [24]) that $\mathbf{FO}[\mathbf{C}]$ is closed under boolean operations and concatenation.

(b) $\Rightarrow$ (a). The predicates C_d^r can be expressed in terms of the modular quantifiers $\exists_d^r$, so every language in $\mathbf{FO}[\mathbf{C}]$ is in $(\mathbf{FO} + \mathbf{MOD})$. It is shown in [24] that every language in $(\mathbf{FO} + \mathbf{MOD})$ is regular. To show that $L \in AC^0$ we take a first-order sentence that defines L and rewrite it as a circuit that accepts $L \cap \{0, 1\}^n$. Replace each occurrence of $\exists x\, \phi(x)$ by $\bigvee_{i=1}^n \phi(i)$ and each occurrence of $\forall x\, \phi(x)$ by $\bigwedge_{i=1}^n \phi(i)$. (We need to introduce a distinct index i for each quantifier that we replace.) We now replace the atomic formulas $i < j$ by the boolean constant 1 if $i < j$

and by 0 otherwise. $Q_0 i$ is replaced by $\neg x_i$ and $Q_1 i$ by x_i. $C_d^r x$ is replaced by 1 if $n \equiv r \pmod{d}$ and by 0 otherwise. The result is an unbounded fan-in boolean expression in variables $x_1, ..., x_n$—that is, a circuit—that defines $L \cap \{0, 1\}^n$. The depth of the circuit is equal to the depth k of nesting of quantifiers in the original sentence (and is therefore independent of n). The size of the circuit is $O(n^k)$. Thus $L \in AC^0$. ∎

Example. Consider the language $LENGTH(2)$. It obviously satisfies condition (c) of the theorem. It satisfies condition (b), as it is defined by the sentence

$$\forall x((\forall y(y \leqslant x)) \rightarrow x \equiv 0 \pmod{2}).$$

While the syntactic monoid of this language is a nontrivial group, we never have two words of the same length mapped under the syntactic morphism to different elements of the syntactic monoid. Thus condition (e) is satisfied. In contrast, $SUM(2)$, which has an isomorphic syntactic monoid, does not satisfy condition (e), since words containing an even number of 1s are mapped to the identity of the group and words containing an odd number of 1s are mapped to the generator of the group.

Our characterization implies that any regular language in nonuniform AC^0 is accepted by a very uniform constant-depth circuit family. Thus our characterization remains true even if a stringent uniformity condition (e.g., the DLOGTIME uniformity of [3]) is imposed.

Theorem 3 enables us to determine in many instances from the multiplication table of $M(L)$ whether a regular language L is in AC^0. (For instance, one easy consequence of Theorem 3 is that every group in $M(L)$ must be cyclic.) In fact, the general problem is decidable: Our proof of Theorem 3 shows that a regular language L is in AC^0 if and only if the relational morphism $\eta_L^{-1}\zeta$, where ζ: $\{0, 1\}^* \rightarrow \mathbf{Z}_r$ is as defined in the first part of the proof, has an aperiodic derived semigroup. We can compute η_L, the multiplication table of $M(L)$, and an appropriate value for r from the minimal automaton of L. We will be able to write down the multiplication table for D, and hence determine whether or not it is aperiodic, provided we can calculate the relation $\eta_L^{-1}\zeta$. But this is just the image of the morphism $(\eta_L, \zeta): \{0, 1\}^* \rightarrow M(L) \times \mathbf{Z}_r$, and every element is the image of a word of length no more than $r \cdot |M|$. Thus the relation can be effectively computed. We have proved:

Theorem 4. *It is decidable whether a given regular language belongs to* AC^0.

The class **FO[C]** is considered, in another context, in the book by McNaughton and Papert [16]. They also show, using a different argument, that membership in this class is decidable.

6. Regular Languages Accepted by Circuits with Gates That Count Modulo a Prime

Smolensky [21] studied the behavior of constant-depth families of circuits that contain both boolean gates and gates that compute the sum of the inputs modulo a fixed prime p. He showed that such a family of circuits requires exponential size to recognize the language $SUM(q)$, where q is relatively prime to p. (An earlier version of this result in the case $p = 2$ is due to Razborov [18].) We use this theorem, as we used the Furst–Saxe–Sipser result in the preceding section, to characterize the regular languages recognized by such circuits. (We state our results in terms of circuit complexity classes defined with polynomial size bounds. The restriction to polynomial size, rather than the more generous lower bound implied by Smolensky's result, is relatively unimportant here; however, it is crucial to our arguments in Section 8.)

Theorem 5. *Let p be prime, and let $L \subseteq \{0, 1\}^*$. The following are equivalent.*

(a) *L is regular and $L \in ACC(p)$.*

(b) *$L \in (\mathbf{FO} + \mathbf{MOD}(p))$ [C].*

(c) *L belongs to the smallest family of subsets of $\{0, 1\}^*$ that contains $\{0\}$, $\{1\}$ and the languages $LENGTH(q)$ for $q > 1$ and that is closed under boolean operations, concatenation, and the operation*

$$T \rightarrow \langle Ta, r, p \rangle,$$

where $0 \leqslant r < p$.

(d) *L is recognized by a morphism $\psi: \{0, 1\}^* \rightarrow M\mathbf{wr}\mathbf{Z}_r$, where M is a finite monoid in which every group is a p-group, and where the composition $\psi\pi: \{0, 1\}^* \rightarrow \mathbf{Z}_r$ takes both 0 and 1 to the generator 1 of $\mathbf{Z}_r$.*

(e) *$M(L)$ is finite, and for every $t \geqslant 0$, every group contained in the image of $\{0, 1\}^t$ under the syntactic morphism η_L is a p-group.*

Proof. The proof is very close to that of Theorem 3. We indicate here the few points of difference.

(a) ⇒ (e). We suppose that the image of $\{0, 1\}^t$ under η_L contains a q-group, where q is a prime not equal to p. By the same argument as in Theorem 3, $SUM(q)$ is in $ACC(p)$, contradicting Smolensky's result.

(e) ⇒ (*d*). Let r and ζ be as in the proof of Theorem 3. It follows by the same argument that every group in the derived semigroup D is a p-group and that L is recognized by a morphism $\psi: \{0, 1\}^* \rightarrow D\mathbf{wr}\mathbf{Z}_r$.

(d) ⇒ (c). This is immediate from Proposition 1, as in the proof of Theorem 3.

(c) $\Rightarrow$ (b). It follows from the techniques used in [24] that $(\mathbf{FO}+\mathbf{MOD}(p))$ [C] is closed under all the cited operations.

(b) $\Rightarrow$ (a). We use the same argument as in Theorem 3, but we must show how modular quantifiers are treated when we transform quantified formulas into circuits. Observe that if $0 \leqslant r < p$ then $\exists_p^r \phi(x)$ is equivalent to

$$\exists y\,(\exists_p^0 x(x \leqslant y \wedge \phi(x)) \wedge \exists^{=r} x(x > y \wedge \phi(x))),$$

where $\exists^{=r}$ means that exactly r positions satisfy the condition in the scope of the quantifier. Since this can easily be defined in terms of ordinary existential quantifiers, it suffices to consider modular quantifiers of the form $\exists_p^0$. We rewrite $\exists_p^0 x\phi(x)$ as $\mathrm{MOD}(p)_{i=1}^n \phi(i)$ and as a result show that $(\mathbf{FO}+\mathbf{MOD}(p))$ [C] $\subseteq ACC(p)$. ∎

We can effectively determine from the multiplication table of a finite monoid whether the only groups it contains are p-groups. Thus we can use the argument of Theorem 4 to prove another decidability result:

THEOREM 6. *Let p be prime. It is decidable whether a given regular language is in $ACC(p)$.*

7. REGULAR LANGUAGES COMPLETE FOR NC^1

Let G be a finite group. Let $A_G = \{a_g \mid g \in G\}$ be a finite alphabet in one-to-one correspondence with G. The map $a_g \mapsto g$ extends to a morphism $\phi: A_G^* \to G$. Then for any $g \in G$, $L_g = g\phi^{-1}$ is a regular language whose syntactic monoid is G.

Let $L \in NC^1$. Barrington [2] showed that if G is a nonsolvable group and $g \in G$, then L is constant-depth reducible to L_g. Since NC^1 contains L_g (relative to any encoding of the alphabet A_G by fixed-length bit strings), this shows that L_g is complete for NC^1 with respect to constant-depth reductions. Here we extend this result.

THEOREM 7. *Let A be a finite alphabet, and let $L \subseteq A^*$ be a regular language such that $M(L)$ contains a nonsolvable finite group. Then L is complete for NC^1 with respect to constant-depth reductions.*

Proof. $M(L)$ contains a group G' that maps onto a simple nonabelian group G via a morphism α. The morphism $\phi_G: A_G^* \to G$ recognizes L_g. We show that there is an integer t and for each $a_g \in A_G$ a word $w_g \in A^*$ of length t, such that $w_g \eta_L \alpha = a_g \phi_G$. Lemma 2 then implies that L_g is constant-depth reducible to L, and the result follows from the completeness of L_g.

Let $z \in A^*$ be a nonempty word such that $z\eta_L \in G'$. Let $k = |z| \cdot |G|$, and let

$$H = \{h \in G \mid \text{there exists } w \in A^* \text{ such that } w\eta_L \alpha = h \text{ and } k \mid |w|\}.$$

Clearly H is closed under product, so H is a subgroup of G. If $g \in G$, $h \in H$, $v\eta_L\alpha = g$, $w\eta_L\alpha = h$, then $g^{-1}hg = (v^{k-1}hv)\eta_L\alpha$, so H is a normal subgroup of G. Let g_1, g_2 be two noncommuting elements of G, and let $v_1\eta_L\alpha = g_1$, $v_2\eta_L\alpha = g_2$. Then

$$(v_1v_2)^k\eta_L\alpha \neq [v_2v_1(v_1v_2)^{k-1}]\,\eta_L\alpha,$$

so H contains at least two distinct elements. Thus $H = G$, by the simplicity of G. We now choose for each $g \in G$ a word v_g of length divisible by k such that $v_g\eta_L\alpha = g$. Observe that $z^{|G|}$ is of length k and maps to the identity of G, so by concatenating each v_g with enough copies of $z^{|G|}$ we obtain words w_g, all of the same length, such that $w_g\eta_L\alpha = g$, as required. ∎

The circuit complexity class ACC is closed under boolean operations, concatenation, and the operation

$$L \to \langle La, r, d \rangle.$$

It follows from the result of [22] that ACC contains the family of regular languages $\mathscr{L}_{\text{sol}}$ consisting of all $L \subseteq \{0, 1\}^*$ for which $M(L)$ contains no nonsolvable groups. ACC is also closed under constant-depth reductions. It is not yet known whether $ACC = NC^1$. If the two classes are equal, then ACC contains all regular languages. If (as we suspect) the two classes are different, then Theorem 7 implies that the regular languages in ACC are precisely those in the family $\mathscr{L}_{\text{sol}}$. The same remarks apply to any complexity class lying between ACC and NC^1. One such class that has been the subject of some study is TC^0, which consists of those languages in $\{0, 1\}^*$ that are AC^0-reducible to $MAJORITY$. (See [12].) We summarize this as follows:

THEOREM 8. *Let $L \subseteq \{0, 1\}^*$ be regular.*

(a) *If $ACC \neq NC^1$ then $L \in ACC$ if and only if $L \in \mathscr{L}_{\text{sol}}$.*

(b) *If $TC^0 \neq NC^1$ then $L \in TC^0$ if and only if $L \in \mathscr{L}_{\text{sol}}$.*

8. Definability of Regular Languages in Extensions of First-Order Logic

Our results in the preceding sections show that one cannot exploit the nonuniformity of a circuit family to recognize unusual regular languages. We now examine this "forced uniformity" phenomenon in light of some recent work of Barrington and Thérien [4], which characterizes the complexity classes AC^0 and $ACC(q)$ in terms of certain sorts of programs over finite automata. We are thus able to isolate the property that we believe is responsible for the forced uniformity. A direct proof of our conjectured property would answer many of the unsolved problems concerning the structure of constant-depth reducibilities within NC^1.

Let A be a finite alphabet and let $L \subseteq A^*$. A program over L is a sequence Φ of triples

$$(i_0, a_0, b_0), ..., (i_{r-1}, a_{r-1}, b_{r-1}),$$

where each i_j is in $\{0, ..., n-1\}$. Given a bit string

$$\mathbf{u} = u_0 \cdots u_{n-1} \in \{0, 1\}^n,$$

the program emits the word

$$\Phi(\mathbf{u}) = c_0 \cdots c_{r-1} \in A^*,$$

where $c_k = a_k$ if $u_{i_k} = 0$, and $c_k = b_k$ if $u_{i_k} = 1$. The program accepts $\mathbf{u}$ if $\Phi(\mathbf{u}) \in L$. In this manner a family $\{\Phi_n\}_{n \geqslant 0}$ of programs over L recognizes a subset of $\{0, 1\}^*$. Barrington and Thérien's main result is:

(i) $T \in AC^0$ if and only if T is recognized by a polynomial-size family of programs over a star-free regular language L.

(ii) $T \in ACC(q)$ if and only if T is recognized by a polynomial-size family of programs over a regular language L such that $M(L)$ contains no groups of order relatively prime to q.

If $L \subseteq A^*$ is a regular language we define a new regular language L' by introducing a new letter t and setting $L' = L\gamma^{-1}$, where $\gamma: (A \cup \{t\})^* \to A^*$ is the morphism defined by $a\gamma = a$ for $a \in A$ and $t\gamma = 1$. Now, given a polynomial-size family $\{\Phi_n\}$ of programs over L we obtain a polynomial-size family $\{\Phi'_n\}$ of programs over L' that recognizes the same subset of $\{0, 1\}^*$ by adding polynomially many triples of the form (i, t, t) in an arbitrary fashion. In particular, we can add new triples in such a fashion that the program $\{\Phi'_n\}$ has length n^k and makes n^{k-1} complete passes over the input—that is, the jth triple has the form (i, a, b), where $i \equiv j \pmod{n}$. It is easy to see that L' is recognized by $M(L)$. Thus the results of Barrington and Thérien cited above remain true under the assumption, which we shall henceforth make, that the programs are in this normal form.

Our next theorem characterizes the nonuniform complexity classes AC^0 and ACC in terms of first-order logic and its extensions. Part (a) appears, in a somewhat different form, in Immerman [13].

THEOREM 9. (a) $AC^0 = \mathbf{FO}[\mathbf{R}]$.

(b) *For all* q, $ACC(q) = (\mathbf{FO} + \mathbf{MOD}(q))\,[\mathbf{R}]$.

(c) $ACC = (\mathbf{FO} + \mathbf{MOD})\,[\mathbf{R}]$.

Proof. Let $T \in AC^0$. Then T is recognized by a family of programs over a star-free language $L \subseteq A^*$. By the theorem of McNaughton cited in Section 4, L is defined by a sentence $\psi \in \mathbf{FO}$. We assume that the programs have the normal form described above, and that the length of the nth program is n^k. We now rewrite the sentence ψ to obtain an sentence in **FO[R]** defining the strings of length n in T. The value of the nth program on a bit string $\mathbf{u} \in \{0, 1\}^n$ is a string of length n^k in A^*. We are thus interpreting the variables $x_1, ..., x_t$ of ψ in the universe $\{0, ..., n^k - 1\}$. Each $m \in \{0, ..., n^k - 1\}$ has a unique decomposition in the form $m = \sum_{i=0}^{k-1} m_i n^i$, where $0 \leqslant m_i < n$. We thus rewrite ψ replacing each of the variables x_i by the k variables $x_i^{(0)}, ..., x_i^{(k-1)}$, where these are interpreted in $\{0, ..., n-1\}$: Each existential quantifier $\exists x_i$ is replaced by

$$\exists x_i^{(0)} \cdots \exists x_i^{(k-1)},$$

and similarly, each universal quantifier is replaced by the corresponding block of k universal quantifiers. We rewrite $x_i < x_j$ as

$$\bigvee_{r=1}^{k} \left((x_i^{(k-r)} < x_j^{(k-r)}) \wedge \bigwedge_{s=1}^{r-1} (x_i^{(k-r+s)} = x_j^{(k-r+s)}) \right),$$

which is equivalent if we interpret the $x_i^{(r)}$ as the components of x_i. A term of the form $Q_a x_i$ will be true if either the x_i^0 bit of the input is off and the second component of the x_ith instruction of the program is a or if the x_i^0 bit of the input is on and the third component of the x_ith instruction of the program is a. We introduce $2|A|$ k-ary relation symbols $R_a, S_a (a \in A)$, where $R_a(m_0, ..., m_{k-1})$ is interpreted as "the second component of the mth triple of the program is a," where $m = \sum_{i=0}^{k-1} m_i n^i$. We interpret the relation symbols S_a similarly, in terms of the third components of the program triples. These relations depend only on the length n of the program and the values of the arguments, and are thus instances of the numerical predicates defined in Section 2. We are thus able to translate $Q_a x_i$ in terms of $Q_0 x_i^{(0)}$, $Q_1 x_i^{(0)}$, and the new relation symbols.

We have shown that every $T \in AC^0$ is in **FO[R]**. Conversely, if a subset of $\{0, 1\}^*$ is in **FO[R]**, it is in AC^0, for we can unravel a defining sentence into a circuit, as was done in the proof of Theorem 3. This completes the proof of part(a).

We treat part (b) similarly, using the fact that $T \in ACC(q)$ if and only if it is recognized by a polynomial-size family of programs over a language $L \in (\mathbf{FO} + \mathbf{MOD}(q))$. (This follows from the results of Barrington and Thérien cited above and is proved in [24].) We only need to show how to rewrite the modular quantifiers in the sentence defining L in terms of modular quantification over positions in the program output. The quantifier $\exists_q^r x \phi$ means there are exactly r (mod q) k-tuples $(x_0, ..., x_{k-1})$ such that $\phi(x_0, ..., x_{k-1})$. This is equivalent to

$$\bigvee_{f} \bigwedge_{j=0}^{q-1} \exists_q^{f(j)} x_0 \exists_q^j x' \phi,$$

where the inner quantification is over the $(k-1)$-tuples $x' = (x_1, ..., x_{k-1})$, and where the disjunction runs through all functions f from $\mathbf{Z}_q$ to itself such that

$$\sum_{i=0}^{q-1} i \cdot f(i) \equiv r \pmod{q}.$$

We obtain the desired rewriting by induction. This shows that $ACC(q)$ is contained in $(\mathbf{FO}+\mathbf{MOD}(q))\,[\mathbf{R}]$. The opposite inclusion follows from an argument identical to the one given in the last part of the proof of Theorem 5. Part (c) follows immediately from (b). ∎

Theorem 9, coupled with our results on regular languages, enables us to answer a number of questions concerning definability of regular languages in extensions of first-order logic. For example it has been asked (originally by Büchi [7]) what regular languages can be defined by adjoining a binary predicate "$y=2x$" to our first-order formulas. We now know that although we can define some nonregular languages this way, and some regular languages (e.g., $LENGTH(2^k)$) that are not in **FO**, we cannot escape from $\mathbf{FO}[\mathbf{C}]$ in this way. More generally, no such device, adjoined to any of our logical languages, will suffice to define regular languages outside of $\mathscr{L}_{\text{sol}}$. The general principle is the following corollary, which follows at once from Theorems 3, 5, and 9.

COROLLARY 10. *Let* **Reg** *denote the family of regular languages in* $\{0,1\}^*$. *Then*

(a)

$$\mathbf{Reg} \cap \mathbf{FO}[\mathbf{R}] = \mathbf{FO}[\mathbf{C}].$$

(b) *If p is prime, then*

$$\mathbf{Reg} \cap (\mathbf{FO}+\mathbf{MOD}(p))\,[\mathbf{R}] = (\mathbf{FO}+\mathbf{MOD}(p))\,[\mathbf{C}].$$

There appears to be a general principle whereby regularity of the language defined can be used to replace arbitrary numerical predicates by periodic ones. Our guess is that this is related to an inherent periodicity of regular languages; in particular, we think it unlikely that part (b) of the corollary has anything to do with the primality of p. We thus conjecture:

Conjecture. For all q,

$$\mathbf{Reg} \cap (\mathbf{FO}+\mathbf{MOD}(q))\,[\mathbf{R}] = (\mathbf{FO}+\mathbf{MOD}(q))\,[\mathbf{C}].$$

The conjecture implies the results of [11, 21] (for polynomial-size circuits), and, if proved, would settle a number of open questions about the structure of NC^1:

THEOREM 11. *Suppose the conjecture is true. Then*

(a) *If q is relatively prime to r then $SUM(q) \notin ACC(r)$.*

(b) *$ACC(q) = ACC(r)$ if and only if q and r have the same set of prime factors.*

(c) *$MAJORITY \notin ACC$. In particular $ACC \neq NC^1$.*

Proof. (a) Since $SUM(u)$ is AC^0-reducible to $SUM(v)$ whenever $u \mid v$, we may assume without loss of generality that q is prime. The techniques of [24] can

$$\begin{tikzcd}
 & \{0,1\}^* \arrow[dl, "\eta_L"'] \arrow[d, "\phi"] \arrow[dr, "\zeta"] & \\
M(L) & M \arrow[l, "\theta"] \arrow[r, "\psi"'] & Z_r
\end{tikzcd}$$

Diagram 7

be used to show that every language in $(\mathbf{FO}+\mathbf{MOD}(r))\,(\mathbf{C}]$ is in the closure of the letters and the languages $LENGTH(m)$ under boolean operations and the operations

$$(L_1, L_2) \mapsto L_1 L_2$$

and

$$(L_1, L_2) \mapsto L_1 *_r L_2.$$

If $L \in ACC(r)$ is regular, then by the conjecture, $L \in (\mathbf{FO}+\mathbf{MOD}(r))\,[\mathbf{C}]$. It follows from the remarks in Section 3 concerning triple products that there is a commutative diagram where θ is surjective, $w\zeta$ depends only on the length of w, and ψ is injective on groups of order q. (We are also using the easily proved fact that for each letter $a \in \{0, 1\}$, $M(\{a\})$ contains no nontrivial groups, and hence any projection morphism $M \times M(\{a\}) \to M$ is injective when restricted to groups.

. However, if $L = SUM(q)$ then such a diagram is impossible. Indeed, we can argue as in the proofs of Theorems 3 and 5 that the derived semigroup D of $\phi^{-1}\zeta$ cannot contain a group with an element of order q. On the other hand, the derived semigroup D' of $\eta_L^{-1}\zeta$ is a quotient of D via the morphism $(n, m, n') \mapsto (n, m\theta, n')$ and the element $(0, (10^{n-1})\eta_L, 0) \in D'$ generates a group of order q.

(b) Observe that $SUM(m)$ is AC^0-reducible to $SUM(n)$ if $m \mid n$. The result now follows easily from (a) (only if direction) and the fact that a gate that counts mod r^k can be built from boolean gates and gates that count mod r.

(c) If $MAJORITY \in ACC$ then $MAJORITY \in ACC(q)$ for some q. Pick a prime p that does not divide q. Since $SUM(p)$ is constant-depth reducible to $MAJORITY$, the result follows from (a). ∎

9. Open Questions

We would like to find a direct proof of the conjecture of Section 8—that nonuniform quantified sentences defining regular languages can be rewritten as equivalent sentences in $(\mathbf{FO}+\mathbf{MOD})$ [C]. Even a proof of this fact for ordinary quantifiers only, which would yield a new proof of the fundamental result of Furst, Saxe, and Sipser [11], would be of considerable interest. It seems to be the case that natural number-theoretic properties of regular languages can be expressed in terms of the predicates $x<y$ and $x \equiv r \pmod{d}$—for example, the predicate $R(x, y) \equiv$ "there exists $w \in L$ with 0 in position x and 1 in position y" can be expressed in this fashion. Our conjecture can be viewed as an extension of this principle.

Our transformation from first-order formulas defining star-free regular languages to nonuniform formulas defining languages in AC^0 preserves quantifier depth—that is $\sum_k$ formulas are transformed into $\sum_k$ formulas. It is also known from the work of Barrington and Thérien [4] and Thomas [26] that quantifier depth of the original first-order formula corresponds to circuit depth of the language defined by the nonuniform formula. We would like to prove, using techniques like those in Section 4 and the hierarchy results of Sipser [20] that the regular languages in depth-k AC^0 are precisely those defined by boolean combinations of sentences in depth-k **FO**[**C**]. One obstacle to doing this is formulating the appropriate definition of this latter class. We suspect that the correct formulation is the following: Admit as "atomic" formulas any formula in which predicates of the form Q_a do not appear in the scope of a quantifier, then take boolean combinations of Σ_k-formulas over this basis. Of course we are most interested in a direct proof of this characterization of the regular languages in depth-k AC^0. Such an extension to our conjecture, coupled with techniques used to prove hierarchy theorems concerning regular languages (as, for example, in Brzozowski and Knast [6]) would yield a new proof of the main theorem of [20].

If we restrict our attention to nonuniform sentences constructed using only modular quantifiers we get a logical characterization of the languages recognized by polynomial-size families of programs over finite solvable groups. It has been conjectured [5] that such a program cannot compute the AND function. Can the Ramsey-theoretic techniques of [5], together with this new characterization, be used to prove this conjecture? A detailed treatment of this class of languages, along the lines of Section 9 above, is given in [25].

Finally, let us admit a new "majority quantifier" U, where $Ux_1 \cdots x_k \psi$ is interpreted as "more than half of the k-tuples of positions of w satisfy ψ." We can define language classes **MAJ**, **MAJ**[**C**], and **MAJ**[**R**]. Then **MAJ** = **MAJ**[**C**], and $TC^0 =$ **MAJ**[**R**]. We would like to know if the analogue of our conjecture holds for this kind of quantifier. If so, it would, coupled with Theorem 8(b), show that $TC^0 = NC^1$ if and only if every regular language is in **MAJ**.

References

1. M. Ajtai, $\sum_1^1$ formulae on finite structure, *Ann. Pure Appl. Logic* **24** (1983), 1–48.
2. D. Barrington, Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1, *J. Comput. System Sci.* **38** (1989), 150–164.

3. D. Barrington, N. Immerman, and H. Straubing, On uniformity in NC^1, *J. Comput. System Sci.* **4** (1990), 274–306.
4. D. Barrington and D. Thérien, Finite monoids and the fine structure of NC^1, *J. Assoc. Comput. Mach.* **35** (1988), 941–952.
5. D. Barrington, H. Straubing, and D. Thérien, Non-uniform automata over groups, *Inform. Comput.* **89** (1990), 109–132.
6. J. A. Brzozowski and R. Knast, The dot-depth hierarchy of star-free events is infinite, *J. Comput. System Sci.* **16** (1978), 37–55.
7. J. Büchi, Weak second-order arithmetic and finite automata, *Z. Math. Logik Grundlagen Math.* **6** (1960), 66–92.
8. A. Chandra, S. Fortune, and R. Lipton, Unbounded fan-in circuits and associative functions, *J. Comput. System Sci.* **30** (1985), 222–234.
9. A. Chandra, L. Stockmeyer, and U. Vishkin, Constant depth reducibility, *SIAM J. Comput.* **13** (1984), 423–439.
10. S. Eilenberg, "Automata, Languages and Machines," Vol. B, Academic Press, New York, 1976.
11. M. Furst, J. Saxe, and M. Sipser, Parity, circuits, and the polynomial time hierarchy, *Math Systems Theory* **17** (1984), 13–27.
12. A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, and G. Turán, Threshold circuits of bounded depth, *in* "Proceedings, 28th IEEE FOCS, 1987," pp. 99–110.
13. N. Immerman, Languages that capture complexity classes, *SIAM J. Comput.* **16** (1987), 760–778.
14. R. Karp and R. Lipton, Some connections between non-uniform and uniform complexity classes, *in* "Proceedings, 12th ACM STOC, 1980," pp. 302–309.
15. G. Lallement, "Semigroups and Combinatorial Applications," Wiley, New York, 1979.
16. R. McNaughton and S. Papert, "Counter-free Automata," MIT Press, Cambridge, MA, 1971.
17. J. E. Pin, "Varieties of Formal Languages," Plenum, London, 1986.
18. A. A. Razborov, Lower bounds for the size of circuits of bounded depth with basis $\{\&, \oplus\}$, *Mat. Zametki* **41** (1987), 598–607. [English translation in *Math. Notes* **41**, 333–338.]
19. M. P. Schützenberger, On finite monoids having only trivial subgroups, *Inform. Control* **8** (1965), 190–194.
20. M. Sipser, Borel sets and circuit complexity, *in* "Proceedings, 15th ACM STOC, 1983," pp. 61–69.
21. R. Smolensky, Algebraic methods in the theory of lower bounds for boolean circuit complexity, *in* "Proceedings, 19th ACM STOC, 1987," pp. 77–82.
22. H. Straubing, Families of recognizable sets corresponding to certain varieties of finite monoids, *J. Pure Appl. Algebra* **15** (1979) 305–318.
23. H. Straubing, Aperiodic homomorphisms and the concatenation product of recognizable sets, *J. Pure Appl. Algebra* **15** (1979), 319–327.
24. H. Straubing, D. Thérien and W. Thomas, Regular languages defined with generalized quantifiers, *in* "Proceedings, 15th ICALP," Springer Lecture Notes in Computer Science, pp. 561–575, 1988.
25. H. Straubing, "Constant-Depth Periodic Circuits," *Internat. J. Algebra Comput.* **1** (1991), 49–87.
26. W. Thomas, Classifying regular events in symbolic logic, *J. Comput. System Sci.* **25** (1982), 360–376.
27. B. Tilson, Categories as algebra, *J. Pure Appl. Algebra* **48** (1987), 83–198.