comput complexity 1 (1991), 330–359 1016-3328/91/040330-30 $1.50+0.20

NC^1: THE AUTOMATA-THEORETIC VIEWPOINT

PIERRE MCKENZIE, PIERRE PÉLADEAU AND DENIS THÉRIEN

Abstract. Concepts from the algebraic theory of finite automata are carried over to the "program-over-monoid" setting which underlies Barrington's algebraic characterization of the complexity class NC^1. Sets of languages accepted by polynomial-length programs over finite monoids drawn from a given monoid variety **V** emerge as fundamental language classes: as **V** ranges over monoid varieties these classes capture and indeed refine the usual bounded-depth circuit parametrization of non-uniform NC^1 subclasses. Here it is shown that any two separable such language classes can be separated by a regular language. Basic properties of these language classes are exhibited. New conditions are given under which distinct varieties lead to equal or to distinct language classes, thus sharpening our knowledge of the internal structure of non-uniform NC^1. The paper concludes with the statement of a conjecture whose proof would refine and then resolve most open questions about this internal structure.

Key words. Automata, circuit, complexity, monoid, variety.
Subject classifications. 68Q15, 68Q70, 20M35.

1. Introduction

Over the last ten years a great deal of effort was devoted to the study of "small" complexity classes believed to be properly contained in the familiar class *PTIME*. An important such class is NC^1, defined in terms of bounded-indegree uniform Boolean circuit families of logarithmic depth [33, 15]. Interest in NC^1 and its subclasses stems from several sources: first, these classes are relevant to the study of relativized complexity [17, 1, 48, 19, 23]; second, they play a role in the study of parallel computation [15]; third, significant lower bounds can be proved concerning them [1, 48, 17, 37, 19, 41].

Partly in order to study NC^1 restrictions, Borodin *et al.* and Chandra *et al.* in 1983 introduced bounded-width branching programs [10, 14]. Three years later, Barrington proved in a far-reaching paper that width-5 polynomial-length branching programs precisely characterize non-uniform NC^1 [2]. Barrington's slick construction exploited a consequence of the algebraic property of non-solvability of the symmetric group of degree 5. The bounded-width branching program model thus gave rise to the "non-uniform deterministic finite automaton" [2, 8] or "program over a *monoid* M" or "M-program" [26, 9]. Recall that a monoid is a set equipped with an associative operation and an identity element; a monoid is *aperiodic* iff no subset of it forms a non-trivial group, and it is *solvable* iff no subset of it forms a non-solvable group. All monoids considered in this paper are finite. Deep connections between M-programs and circuit complexity were soon uncovered: when M ranges respectively over aperiodic monoids [8], solvable monoids [8] and non-solvable monoids [2], polynomial length M-programs precisely characterize the non-uniform versions of the classes AC^0, ACC^0 and NC^1 (where AC^0 and ACC^0 are defined in terms of bounded-depth unbounded-fan-in circuits with gates from $\{\vee, \wedge\}$ and from $\{\vee, \wedge, MOD_q\}$ respectively, a MOD_q gate outputting 1 iff the sum of its binary inputs is a multiple of the integer q); even more strikingly, in the cases of AC^0 and ACC^0 the hierarchies induced by taking the exact depth of the circuits correspond to natural parametrizations of aperiodic and solvable monoids [8].

Independently from the above research, the last 25 years have seen the growth of an elaborate theory elucidating connections between combinatorial properties of classes of regular languages and algebraic properties of the finite automata which recognize these languages (see [16, 34]). In particular, this theory has been very successful in the study of regular languages whose minimal automata, viewed as monoids of transformations on their state sets, exclude non-solvable groups (i.e. are solvable monoids). As an oft-cited example, the regular languages accepted by aperiodic finite automata are the star-free languages [38]. Countless further examples arise from the nilpotency class parametrization of nilpotent groups [46], from parametrizations of solvable groups and solvable monoids [42, 45], or from a wealth of natural monoid classes drawn from semigroup theory (see for instance [39, 11, 27, 16, 12]).

In the context of polynomial-length M-programs, the full power of NC^1 is attained as soon as M contains a non-solvable group [2, 3]. Hence from the M-program point of view a great deal of the internal structure of NC^1 is determined exclusively by solvable monoids. Since solvable monoids are precisely those which are well understood in the restricted setting of automata theory, it is compelling to systematically probe the connections between NC^1 subclasses and M-programs under the guidance of the algebraic theory of finite automata. This is the purpose of the present paper.

Because the uniformity issue in the definition of circuit-based complexity classes is extraneous to the connections with M-programs discussed in this paper, we focus on non-uniform NC^1 and its subclasses. Much of our work would apply to any reasonable uniform version of these classes because the languages which we use to discuss the separation of the non-uniform classes are regular languages. Membership of a regular language in a natural non-uniform complexity class should certainly be preserved under any reasonable uniformity criterion (and non-membership is preserved under any criterion). We nonetheless take care to distinguish between our usages of NC^1 and of non-uniform NC^1.

The central concept in the ensuing theory is found to be the family $\mathcal{P}(\mathbf{V})$ of languages recognized by polynomial-length M-programs when M ranges over monoids in monoid *variety* $\mathbf{V}$. Indeed the fundamental notion of a natural class of monoids is that of a variety; this concept, adapted from universal algebra, plays an important role in the theory of regular languages. Informally speaking, a variety of monoids is a class of monoids which share a set of properties. For example, the class of monoids whose elements satisfy a fixed set of equations forms a variety; a specific example is the variety of commutative monoids, defined by the equation $xy = yx$. See section 2 for technical definitions and further motivation. Monoid varieties are thus the natural units in the classification of all monoids and classes $\mathcal{P}(\mathbf{V})$ arise as the natural units in the classification of languages accepted by M-programs.

In this paper we begin by identifying basic properties of language classes $\mathcal{P}(\mathbf{V})$. We note that such classes are closed under Boolean operations, under quotients, and under a restricted version of inverse morphism which we call inverse "length-multiplying" morphisms. Then we observe that any two separable classes $\mathcal{P}(\mathbf{V})$ and $\mathcal{P}(\mathbf{W})$ can be separated by a regular language. This is of particular interest in light of the close connection emphasized herein between classes $\mathcal{P}(\mathbf{V})$ and subclasses of non-uniform NC^1. The reasoning leading to our observation points to candidate regular languages, namely "word problems" over appropriately defined monoids, for separating the various subclasses of NC^1 thought to be distinct.

Then, in the context of classes $\mathcal{P}(\mathbf{V})$, we relate different ways of extending a variety $\mathbf{V}$ to the combinatorial operation of adding a "level of counting" in the corresponding classes of circuits. From this, and from the correspondence established by Barrington, Straubing and Thérien between applying a wreath product and increasing the depth of an unbounded-fan-in circuit [8, 6], we rederive in a uniform manner and extend the connections between circuits and M-programs: as $\mathbf{V}$ ranges over well-studied monoid varieties, $\mathcal{P}(\mathbf{V})$ ranges over a very extensive list of subclasses of non-uniform NC^1 including, in fact, all such subclasses considered in the recent literature with the exception of those defined in terms of threshold functions [29] (see nonetheless [9]). Therefore, the classes $\mathcal{P}(\mathbf{V})$ not only provide a unified picture of the internal structure of non-uniform NC^1 but the details of this structure almost completely hinge on what emerges as the core question in the theory:

$$\boxed{\text{Exactly when does } \mathbf{V} \neq \mathbf{W} \text{ entail } \mathcal{P}(\mathbf{V}) \neq \mathcal{P}(\mathbf{W})?}$$

A complete answer to this question would settle most major open questions about NC^1 and non-uniform NC^1, including, for instance, the computational power of bounded-depth circuits made up solely of MOD_6 gates, and the precise relationship between ACC^0 and AC^0 or NC^1 (see [49]). We are unable to claim a complete answer to this question here. However, noting that the rich lattice of monoid varieties affords a much finer parametrization of non-uniform NC^1 than that merely obtained in terms of circuit types and circuit depth, we are able in some cases to establish new algebraic conditions, on varieties $\mathbf{V}$ and $\mathbf{W}$, under which class $\mathcal{P}(\mathbf{V})$ differs from $\mathcal{P}(\mathbf{W})$ or under which $\mathcal{P}(\mathbf{V}) = \mathcal{P}(\mathbf{W})$. To discuss these in the sequel we will say that $\mathbf{V}$ and $\mathbf{W}$ *split* if $\mathcal{P}(\mathbf{V}) \neq \mathcal{P}(\mathbf{W})$ and that $\mathbf{V}$ and $\mathbf{W}$ *merge* otherwise.

Known results about NC^1 immediately answer our core question in the case of several pairs of varieties $\mathbf{V}$ and $\mathbf{W}$. For instance, Barrington's work [2] implies that $\mathbf{V}$ and $\mathbf{W}$ merge whenever each contains a non-solvable group. On the other hand, for p a fixed prime, it follows from Smolensky's lower bounds [37, 41] that $\mathbf{V}$ and $\mathbf{W}$ split whenever exactly one of $\mathbf{V}$ and $\mathbf{W}$ has the property that the order of each group in the variety is a power of p. We give new answers to the question of the splitting or merging of varieties in the case of Abelian monoid varieties, nilpotent group varieties, J-trivial monoids, R-trivial monoids, and the variety $\mathbf{B}_2$ (see the next section for definitions). We prove that any two distinct Abelian monoid varieties split, that the variety of nilpotent groups of exponent q merges with the variety of nilpotent groups of exponent q' if and only if q and q' have the same prime divisors, and that $\mathbf{J}$, $\mathbf{R}$ and $\mathbf{B}_2$ all split from each other. Our arguments in some cases are Ramsey-theoretic and are thus weaker but substantially simpler than, say, those of Smolensky [41] or of Furst, Saxe and Sipser [17, 19].

The final contribution of this paper is the statement of a single conjecture whose validity would provide in a single blow the expected answers to just about all open questions concerning the internal structures of NC^1 and of non-uniform NC^1, with the exception of those involving bounded-depth threshold circuits. Indeed, verifying our conjecture would allow utilizing known results from automata theory to separate AC^0, CC^0, ACC^0 and NC^1 (where CC^0—also called pure ACC by Yao [49]—comprises the union over all constants q of the sets of languages recognized by families of constant-depth unbounded fan-in circuits built solely from gates computing the MOD_q function): moreover strictness of the natural subhierarchies induced by considering the exact (constant) depth of the underlying circuits would follow. Validity of our conjecture would also provide a common proof to separation results in [17] and [40]. Informally, this conjecture states that the situation with regard to the ability of solvable monoids to recursively count subwords, in the restricted setting of regular language recognition, carries over verbatim to the more general setting of classes $\mathcal{P}(\mathbf{V})$. We refer the reader to section 4 for the precise statement of this conjecture. (In passing we note that a related conjecture which however does not take the precise depth of circuits into account appears in [3]; see also [30].)

This paper is organized as follows. Section 2 contains background and definitions. Section 3 develops basic properties of classes $\mathcal{P}(\mathbf{V})$. Section 4 introduces and then treats the fundamental question of the splitting or the merging of monoid varieties. Section 5 concludes with a discussion and pointers to further work.

2. Background and notation

We use "$\subset$" to denote proper inclusion. By $[n]$ we mean the set $\{1, 2, \ldots, n\}$. For sets S and T we write S^T for the set of all functions $f : T \to S$. A *morphism* ϕ from a monoid M to a monoid N is simply a function from M to N mapping the identity of M to the identity of N and verifying $\phi(xy) = \phi(x)\phi(y)$ for each $x, y \in M$. We will require the following fact:

FACT 2.1. *(Ramsey, see [18]) Let s, k and c be positive integers. There exists an integer $Ramsey(s + k, k, c)$ such that for any larger integer n the following holds: $n \geq s + k$ and any assignment of one of c colours to each k-element subset of $[n]$ results in a particular $(s + k)$-element subset of $[n]$ all of whose own k-element subsets are assigned the same colour.*

2.1. From automata ....

In this subsection we outline the salient features of algebraic automata theory: more details can be found in [16, 34].

The class of regular subsets of A^* can be defined as the smallest family that contains the empty set and that is closed under Boolean operations (if L_1 and L_2 are regular, then so are $L_1 \cup L_2$ and $A^* \setminus L_1$), letter-concatenation (if L_1 and

L_2 are regular, so is L_1aL_2 for any $a \in A$) and star (if L is regular, so is L^*). A famous theorem of Kleene [22] asserts that L is regular iff it can be recognized by a finite automaton. The algebraic approach replaces automata by monoids. A language L is *morphism-recognized* or *$\mathcal{M}$-recognized* by a monoid M iff there exists a morphism $\theta : A^* \to M$ such that $L = \theta^{-1}(F)$ for some $F \subseteq M$. The equivalence between recognition by automata and morphism-recognition by monoids is readily proved, so that Kleene's result can be restated as saying that a language is regular iff it can be $\mathcal{M}$-recognized by a finite monoid. We will write $\mathcal{M}(A^*, M)$ for the collection of subsets of A^* that are $\mathcal{M}$-recognized by M and $\mathcal{M}(M)$ for the union, over all alphabets A, of the collections $\mathcal{M}(A^*, M)$.

Viewing monoids as language recognizers, it is natural to introduce an ordering on finite monoids, based on their computing power. For monoids M, N, we thus define that M *$\mathcal{M}$-divides* N, written $M \prec_{\mathcal{M}} N$, iff $\mathcal{M}(M) \subseteq \mathcal{M}(N)$, i.e., any language $\mathcal{M}$-recognized by M can also be $\mathcal{M}$-recognized by N. The relation of $\mathcal{M}$-division clearly forms a partial order[1].

The following structural characterization of $\mathcal{M}$-division is fundamental.

FACT 2.2. *$M \prec_{\mathcal{M}} N$ iff M is a morphic image of a submonoid of N.*

The last result implies that the algebraic structure of a monoid imposes combinatorial constraints on the languages $\mathcal{M}$-recognizable by this monoid: much of the research on regular languages has focussed on making explicit the relationship between combinatorial descriptions of languages and algebraic properties of their recognizers. The proper level at which to consider this relationship was shown by Eilenberg [16] to be that of varieties. A class $\mathbf{V}$ of finite monoids forms a *variety* iff it is closed under $\mathcal{M}$-division and finite direct product. Denote by $\mathcal{M}(A^*, \mathbf{V})$ the class of subsets of A^* which are $\mathcal{M}$-recognized by a monoid in $\mathbf{V}$, and by $\mathcal{M}(\mathbf{V})$ the union, over all alphabets A, of the classes $\mathcal{M}(A^*, \mathbf{V})$. If $\mathbf{V}$ is a variety, it can be shown that, for each alphabet A, $\mathcal{M}(A^*, \mathbf{V})$ is closed under Boolean operations and under left and right quotients (if $L \in \mathcal{M}(A^*, \mathbf{V})$, so is $u^{-1}L = \{x : ux \in L\}$ and $Lv^{-1} = \{x : xv \in L\}$ for any $u, v \in A^*$); moreover the class $\mathcal{M}(\mathbf{V})$ is closed under inverse morphisms (if $L \in \mathcal{M}(A^*, \mathbf{V})$ and $\phi : B^* \to A^*$ is a morphism, then $\phi^{-1}(L) \in \mathcal{M}(B^*, \mathbf{V})$). In fact, the mapping $\mathbf{V} \to \mathcal{M}(\mathbf{V})$ is a bijection between monoid varieties and classes of regular languages satisfying these closure properties.

[1] The partial order $\prec_{\mathcal{M}}$ is actually defined on the $\simeq_{\mathcal{M}}$-classes where the equivalence $\simeq_{\mathcal{M}}$ is defined by $M \simeq_{\mathcal{M}} N$ iff

$$M \prec_{\mathcal{M}} N \text{ and } N \prec_{\mathcal{M}} M.$$

In view of Fact 2.2, we know that $\simeq_{\mathcal{M}}$ is the isomorphism equivalence.

A classical example of this bijection relates star-free languages and aperiodic monoids [38]; other examples give combinatorial descriptions of the classes of languages recognized by commutative monoids [16], nilpotent groups [46], J-trivial monoids [39], solvable groups [42, 45], solvable monoids [45], the variety generated by inverse semigroups [25] and monoids whose regular J-classes are rectangular bands [36].

A slightly different point of view on the above notion of $\mathcal{M}$-recognition by monoid can be given. The operation of a monoid M induces a canonical surjective morphism $\eta_M : M^* \rightarrow M$, which is defined by evaluating the product of the elements of a given sequence. Any morphism from A^* to M can be viewed as a morphism $\phi : A^* \rightarrow M^*$ which is then composed with η_M: the action of ϕ is thus a "preprocessing" that transforms an input sequence x in A^* to a string $\phi(x)$ in M^*, the set of words over the alphabet of M, and $\phi(x)$ is then evaluated according to the "transition function" η_M. Let the set of *word problems over* M, denoted $\mathcal{W}(M)$, be the family $\{\eta_M^{-1}(m) : m \in M\}$ of subsets of M^*: obviously $\mathcal{W}(M) \subseteq \mathcal{M}(M^*, M)$ and the computing power of a monoid M is characterized in a strong sense by the languages in $\mathcal{W}(M)$.

FACT 2.3. *Let* $\mathbf{V}$ *be the variety generated by the monoid* M*: then* $\mathcal{M}(A^*, \mathbf{V})$ *is the Boolean algebra generated by the languages of the form* $\phi^{-1}(L)$*, where* $\phi : A^* \rightarrow M^*$ *is a morphism and* $L \in \mathcal{W}(M)$*.*

2.2. ... to programs. In this subsection we describe how the classical notion of recognition by a monoid, which is based on morphisms as described in the previous subsection, can be extended by allowing more general mappings between monoids: this new class of functions, which we call programs, constitutes the key to the algebraic understanding of non-uniform NC^1.

Fix a finite set A and a monoid M. An *M-program* is a sequence $\phi = (\phi_n)_{n\geq 0}$ where, for each n, ϕ_n is determined by a sequence $\nu_1 \ldots \nu_{l(n)}$ of *instructions*, each instruction having the form (i, f) for some $i \in [n]$ and $f \in M^A$. (Occasionally ϕ_n is referred to as an "n-input M-program".) Setting, for any $x = a_1 \ldots a_n \in A^n$, $(i, f)(x) = f(a_i)$, $\phi_n : A^n \rightarrow M$ is then defined as mapping x to the product in M of the elements $\nu_1(x) \ldots \nu_{l(n)}(x)$. In this way the M-program ϕ induces a map $\phi : A^* \rightarrow M$. Program ϕ has *polynomial length* if there is a constant c such that $l(n) \in O(n^c)$. A subset L of A^* is *program-recognized* or *$\mathcal{P}$-recognized* by M iff there exists a polynomial length M-program $\phi : A^* \rightarrow M$ and a sequence (F_n) of subsets of M such that $L \cap A^n = \phi^{-1}(F_n)$ for each n. The class of languages thus recognized by M will be denoted by $\mathcal{P}(A^*, M)$ and $\mathcal{P}(M)$ will stand for $\bigcup_A \mathcal{P}(A^*, M)$. For a variety $\mathbf{V}$, the important class

$$\bigcup_{M \in \mathbf{V}} \mathcal{P}(M)$$

is denoted $\mathcal{P}(\mathbf{V})$.

EXAMPLE. The (non-regular) language $L = \{w \in \{0,1\}^* \mid w \text{ is a palindrome }\}$ is $\mathcal{P}$-recognizable by a transformation monoid M on 3 points. Indeed define the transformations

$$a = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 2 & 3 \end{pmatrix} ; b = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 3 \end{pmatrix} ; c = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 3 \end{pmatrix} ; e = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$$

and define for each n the n-input M-program $\phi_n = \nu_1 \nu_2 \ldots \nu_{2\lfloor n/2 \rfloor}$ as follows, where we represent an instruction (i, f), $i \in [n]$, $f : \{0,1\} \to M$, as the triple $(i, f(0), f(1))$:

$$\begin{array}{lcr} \nu_1 & = & (1, a, e) \\ \nu_2 & = & (n, b, c) \\ \nu_3 & = & (2, a, e) \\ \nu_4 & = & (n-1, b, c) \\ \vdots & \vdots & \vdots \\ \nu_{2\lfloor n/2 \rfloor - 1} & = & (\lfloor n/2 \rfloor, a, e) \\ \nu_{2\lfloor n/2 \rfloor} & = & (\lceil n/2 \rceil + 1, b, c) \end{array}$$

Then because

$$ab = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 3 \end{pmatrix} ; ec = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 3 \end{pmatrix} ; ac = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 3 & 3 \end{pmatrix} ; eb = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 3 \end{pmatrix},$$

one sees that, for any $w \in \{0,1\}^n$, $\phi_n(w)$ fixes 1 iff w is a palindrome. Setting

$$F = \{t \in M \mid t \text{ fixes } 1\},$$

with M the transformation monoid generated by $\{a, b, c, e\}$, thus yields $L \cap \{0,1\}^n = \phi_n^{-1}(F)$ for each n. Note that in this example we do not make use of the flexibility allowing different accepting sets for different input lengths. □

Note that we do not impose conditions, apart from length, on the sequence (ϕ_n) defining a program ϕ. Such a non-uniform model of computation can thus recognize non-recursively enumerable sets: for example, any $L \subseteq \{a\}^*$ is $\mathcal{P}$-recognized by any monoid M. On the other hand, we believe that all results presented here can be made to carry over to the usual uniform settings (see for instance [15, 4] for appropriate definitions).

Observe finally that in the same way as a morphism into a monoid M can be "factored through" the evaluation morphism η_M (see the end of subsection 2.1), an M-program $\phi : A^* \to M$ can be viewed as $\phi : A^* \xrightarrow{\phi'} M^* \xrightarrow{\eta_M} M$, where $\phi' : A^* \to M^*$ is given by ϕ without doing the multiplication in M. Formally if $w \in A^*$ is a word of length n and the n^{th} program is $\phi_n = \nu_1 \ldots \nu_{l(n)}$, then $\phi'(w) = \nu_1(w) \ldots \nu_{l(n)}(w) \in M^*$.

2.3. Solvable monoids. We refer the reader to any text on group theory for the definitions of solvable groups, simple groups, and non-solvable groups. A monoid is solvable iff none of its subsets are non-solvable groups. Solvable monoids have been investigated in depth from the algebraic point of view and from the point of view of their power as $\mathcal{M}$-recognizers. In this subsection we gather known facts about solvable monoids and we introduce some of the monoid varieties which will be referred to in this paper.

Let $\mathbf{M_{sol}}$ denote the variety of solvable monoids. Conceptually, the "simplest" varieties contained in $\mathbf{M_{sol}}$ are the Abelian varieties $\mathbf{Com}_{t,q}$ defined for any $t \geq 0$ and $q \geq 1$ as follows: $\mathbf{Com}_{t,q}$ is the variety generated by the one-generated monoid $C_{t,q}$ determined by the equation $x^t = x^{t+q}$. It is known that $\mathcal{M}(\mathbf{Com}_{t,q})$ is the Boolean algebra of languages defined by counting prescribed input letters "threshold t and modulo q" [16]. Now the following language construction, based on the idea of "counting subwords in context", extends this notion of counting in order to provide a thorough analysis of the internal structure of $\mathbf{M_{sol}}$ [45]. For any $t \geq 0, q \geq 1$, let $\gamma_{t,q}$ denote the finite-index congruence on the natural integers defined by

$$i \ \gamma_{t,q} \ j \quad \text{iff} \quad (i = j) \text{ or } (\min\{i,j\} \geq t \text{ and } q | i - j).$$

Now let $L_0, \ldots, L_r \subseteq A^*$, $a_1, \ldots, a_r \in A$, $w \in A^*$ and define $|w|_{[L_0,a_1,L_1,\ldots,a_r,L_r]}$ to be the number of factorizations of w in the form $w = w_0 a_1 w_1 \ldots a_r w_r$ with $w_i \in L_i$ for $i = 0, \ldots, r$. (When $a \in A$ we write $|w|_a$ for $|w|_{A^*aA^*}$.) Finally, define the language

$$[L_0, a_1, L_1, \ldots, a_r, L_r]_{i,t,q}$$

as the set $\{w : i \ \gamma_{t,q} \ |w|_{[L_0,a_1,L_1,\ldots,a_r,L_r]}\}$. This construction induces an operation on varieties as follows: for any $\mathbf{V}$, define $\mathcal{Q}_{t,q}\mathbf{V}$ to be the smallest variety containing $\mathbf{V}$ and the syntactic monoid of any language of the form $[L_0, a_1, L_1, \ldots, a_r, L_r]_{i,t,q}$ where $L_0, \ldots, L_r$ are in $\mathcal{M}(\mathbf{V})$. There is an operation on monoids corresponding to this product on languages, hence a purely algebraic description of $\mathcal{Q}_{t,q}\mathbf{V}$ can be given [35, 47]. Let also $\mathcal{Q}\mathbf{V}$ be the join over all t and q of the $\mathcal{Q}_{t,q}\mathbf{V}$: then $\mathbf{M_{sol}}$ is the smallest variety closed under the

$\mathcal{Q}$ operator. If t is fixed to 0, the $\mathcal{Q}$-closure of the trivial variety defines the variety $\mathbf{G_{sol}}$ of solvable groups, and if q is fixed to 1 instead, the variety $\mathbf{A}$ of aperiodic or "group-free" monoids is obtained.

A natural parametrization of the variety $\mathbf{A}$ can be obtained with the operator $\mathcal{Q}_{1,1}$. Let $\mathbf{B}_1 = \mathbf{Com}_{1,1}$ and $\mathbf{B}_k = \mathcal{Q}_{1,1}B_{k-1}$ for all $k \geq 2$.

FACT 2.4. $\mathbf{A} = \bigcup_{k\geq 1} \mathbf{B}_k$.

The hierarchy given by the $\mathbf{B}_k$ corresponds to the well known dot-depth hierarchy for $k \geq 2$ (as adapted from the semigroup version given in [16]). Other well-studied aperiodic monoid varieties are $\mathbf{J}_k \subset \mathbf{J}_{k+1} \subset \mathbf{J} \subset \mathbf{R} \subset \mathbf{B}_2$, where for any k $\mathbf{J}_k$ is defined as the variety generated by the syntactic monoids of the languages $A^*a_1A^* \ldots a_kA^*$, and $\mathbf{J}$ and $\mathbf{R}$ are the varieties of J-trivial and R-trivial monoids respectively (see [34]). Now fix a finite alphabet A. Say $x = a_1a_2 \ldots a_r$ is a *k-subword* of w iff $r \leq k$ and w can be written $w_0a_1w_1a_2w_2 \ldots a_rw_r$. Define for each $k \geq 1$ the equivalence relation $\sim_k$ on A^* as follows: $u \sim_k v$ iff u and v possess the same set of k-subwords. Further, define $\equiv_k$ refining $\sim_k$ as follows: $u \equiv_k v$ iff each prefix of u is $\sim_k$-equivalent to some prefix of v and vice versa. We will require the following:

FACT 2.5. a) *[39] For any $k \geq 1$, $L \in \mathcal{M}(A^*, \mathbf{J}_k)$ iff L is a union of $\sim_k$-classes.*

b) *[16, 12] $L \in \mathcal{M}(A^*, \mathbf{R})$ iff there is a k such that L is a union of $\equiv_k$-classes.*

c) *[12] $\mathcal{M}(A^*, \mathbf{R})$ is the the set of languages which can be written as disjoint union of languages of the form*

$$A_0a_1A_1a_2A_2 \ldots a_rA_n$$

with $n \geq 0$, $a_1, \ldots, a_n \in A$, $A_i \subseteq (A \setminus \{a_{i+1}\})$ for $0 \leq i \leq n-1$ and $A_n \subseteq A$.

We now recall the definition of the wreath product, which plays a crucial role in the algebraic decomposition of solvable monoids. If M and N are monoids, the wreath product $N \circ M$ is the set $N^M \times M$ equipped with the binary operation

$$(f, m)(f', m') = (f'', mm')$$

with

$$\begin{aligned} f'' : \ & M \to N \\ & x \mapsto [f(x)][f'(xm)]. \end{aligned}$$

Then for any monoid M, denote by $\bar{M}$ the monoid obtained from the right regular representation of M by adding the constant transformations, and define $\bar{\mathbf{V}}$ as the variety generated by the monoids $\bar{M}$, $M \in \mathbf{V}$. Finally, for any two varieties $\mathbf{V}$ and $\mathbf{W}$, define $\mathbf{V} * \mathbf{W}$ as the variety generated by all wreath products $N \circ M$ with $N \in \mathbf{V}$, $M \in \mathbf{W}$ [16, 34].

FACT 2.6. **a)** *[44] For any $t \geq 0$ and $q \geq 1$, $\mathbf{Com}_{t,q} * \mathbf{V} \subseteq \mathcal{Q}_{t,q}\mathbf{V}$.*

b) *[30] For any $t \geq 0$ and $q \geq 1$, $\mathbf{Com}_{t,q} * \bar{\mathbf{V}} \subseteq \mathcal{Q}_{t,q}\mathbf{V}$.*

We say $\mathbf{V}$ is a group variety iff it only contains groups. The wreath product closure of solvable groups yields the group variety $\mathbf{G_{sol}}$ and the wreath product closure of $\mathbf{G_{sol}}$ and $\mathbf{A}$ yields $\mathbf{M_{sol}}$ [24]. Turning to $\mathbf{G_{sol}}$, let next $\mathbf{G}_p$ be the variety of p-groups, where p is a prime integer, and for any $q \geq 1$, $\mathbf{G}_{\mathbf{nil},q}$ be the smallest variety containing $\mathbf{G}_p$ for all prime divisors p of q. It is well-known that $\mathbf{G}_p$ is closed under wreath product. We will also use the following:

FACT 2.7. *[45] For any group variety $\mathbf{V}$, $\mathcal{Q}_{0,q}\mathbf{V} = \mathbf{G}_{\mathbf{nil},q} * \mathbf{V}$.*

2.4. NC^1 and its subclasses.

The complexity class non-uniform NC^1 is defined as the set of languages, over alphabet $\{0,1\}$, which are recognized by logarithmic-depth Boolean circuit families whose gates are NOTs and bounded-in-degree ANDs and ORs [33, 15]. Since we occasionally mention the uniform version, simply written NC^1, of this class, note for definiteness that we then refer to the U_{E^*} uniformity criterion adopted by Cook [15, Page 5]. Following Barrington and Thérien [8], we generalize to an arbitrary finite alphabet A by allowing input gates "$a \in B$?" which produce the Boolean value 1 iff the input symbol a belongs to the subset B of A.

We will use the following precise definitions of subclasses of non-uniform NC^1. Define for any $L \subseteq A^*$:

- $L \in AC_0^0$ iff, for each n, $L \cap A^n = A^n$ or $L \cap A^n = \emptyset$;
- for $k > 0$, $L \in AC_k^0$ iff, for each n, $L \cap A^n$ is a constant-size Boolean function of languages of the form $\mathrm{AND}(L_1, \ldots, L_r)$ where

$$L_i \in AC_{k-1}^0 \cup \{A^{j-1}aA^{n-j} : j \in [n], a \in A\}$$

and $r \in O(n^c)$.

Here $\mathrm{AND}(L_1, \ldots, L_r)$ stands for $\bigcap_{1 \leq i \leq r} L_i$: observe that, when $A = \{0, 1\}$, intersecting with $A^{j-1}1A^{n-j}$ amounts to feeding the jth input bit into the AND gate while intersecting with $A^{j-1}0A^{n-j}$ corresponds to feeding in the negation of the jth input. This definition of AND as an intersection accounts for a larger alphabet in a natural way. Note further that, by DeMorgan's laws, our circuits do not require unbounded-fan-in OR gates because constant-size Boolean combinations, including negations, are permitted at each level.

Starting with the same basis, we define the classes $CC_k^0(q)$ ($ACC_k^0(q)$) by using languages of the form $MOD_q(L_1, \ldots, L_r)$ instead of (in addition to) AND in the inductive step, where

$$MOD_q(L_1, \ldots L_r) = \{x \in A^n : |\{i : x \in L_i\}| \equiv 0 \pmod{q}\}.$$

Finally we set $AC^0 = \bigcup_k AC_k^0$, $CC^0(q) = \bigcup_k CC_k^0(q)$, $ACC^0(q) = \bigcup_k ACC_k^0(q)$, $CC^0 = \bigcup_q CC^0(q)$ and $ACC^0 = \bigcup_q ACC^0(q)$. For each k, this definition of AC_k^0 is equivalent to that used in [8] and for each $q \neq 2$, the classes $CC^0(q)$ and $ACC^0(q)$ coincide with those formulated in [6].

We now state the known algebraic characterizations of the complexity subclasses of non-uniform NC^1 which we just defined. These results are implicit in (or are easy consequences of) earlier work by Barrington, Straubing and Thérien [8, 6].

THEOREM 2.8. a) *The following holds:*

$$\begin{aligned} AC_1^0 &= \mathcal{P}(\mathbf{Com}_{1,1}), \\ CC_1^0(q) &= \mathcal{P}(\mathbf{Com}_{0,q}), \\ ACC_1^0(q) &= \mathcal{P}(\mathbf{Com}_{1,q}). \end{aligned}$$

b) *Let, for any $k \geq 1$, $\mathbf{V}_k$, $\mathbf{U}_{k,q}$ and $\mathbf{W}_{k,q}$ be varieties such that*

$$\begin{aligned} AC_k^0 &= \mathcal{P}(\mathbf{V}_k), \\ CC_k^0(q) &= \mathcal{P}(\mathbf{U}_{k,q}), \\ ACC_k^0(q) &= \mathcal{P}(\mathbf{W}_{k,q}). \end{aligned}$$

Then

$$\begin{aligned} AC_{k+1}^0 &= \mathcal{P}(\mathbf{Com}_{1,1} * \bar{\mathbf{V}}_k), \\ CC_{k+1}^0(q) &= \mathcal{P}(\mathbf{Com}_{0,q} * \mathbf{U}_{k,q}), \\ ACC_{k+1}^0(q) &= \mathcal{P}(\mathbf{Com}_{1,q} * \bar{\mathbf{W}}_{k,q}). \end{aligned}$$

3. Basic properties

In this section we investigate general properties of the classes $\mathcal{P}(\mathbf{V})$: we note some of their closure properties, and we observe that the regular languages in these classes completely characterize the classes. Here N and M are arbitrary finite monoids, and $\mathbf{V}$ is an arbitrary monoid variety.

LEMMA 3.1. a) *If $L \in \mathcal{P}(A^*, M)$ then its complement $\bar{L} \in \mathcal{P}(A^*, M)$.*

b) *If $L_1 \in \mathcal{P}(A^*, M_1)$ and $L_2 \in \mathcal{P}(A^*, M_2)$ then $L_1 \cap L_2 \in \mathcal{P}(A^*, M_1 \times M_2)$.*

c) *If $L \in \mathcal{P}(A^*, M_1 \times M_2)$ then L is a (finite) Boolean combination of languages in $\mathcal{P}(M_1)$ and $\mathcal{P}(M_2)$.*

PROOF. Part (a) is obvious. To prove (b), let ϕ_1 (ϕ_2) be an n-input M_1-program (M_2-program) recognizing $L_1 \cap A^n$ ($L_2 \cap A^n$). We replace each instruction (i, f) of ϕ_1 by $(i, \bar{f})$ where $\bar{f}(a) = (f(a), 1_{M_2})$; similarly replace each instruction (i, f) of ϕ_2 by $(i, \bar{f})$ where $\bar{f}(a) = (1_{M_1}, f(a))$. The program ψ obtained by concatenating ϕ_1 and ϕ_2 is such that $\psi(x) = (\phi_1(x), \phi_2(x))$ and thus recognizes $L_1 \cap L_2 \cap A^n$ using the obvious accepting subset of $M_1 \times M_2$. The nth term in the sequence (ψ_n) has length $|\phi_1| + |\phi_2|$, hence polynomial in n. Finally for (c), suppose a language is $\mathcal{P}$-recognized by an M-program $\phi = (\phi_n)$ with $M \in M_1 \times M_2$. Then this language can be written as a finite union of languages each of which is $\mathcal{P}$-recognized by (ϕ_n) using an accepting sequence (F_n) composed solely of the empty set and a fixed singleton subset of $M_1 \times M_2$. Pick any such language L in the finite union. Using projections we construct ϕ_1 and ϕ_2 from ϕ, and $(F_n^{(1)})$ and $(F_n^{(2)})$ from (F_n), such that $L = L_1 \cap L_2$ where L_1 is $\mathcal{P}$-recognized by M_1 via ϕ_1 with accepting sequence $(F_n^{(1)})$ and L_2 is $\mathcal{P}$-recognized by M_2 via ϕ_2 with accepting sequence $(F_n^{(2)})$. □

The proofs of the following three lemmas are similar in flavor to that of Lemma 3.1, and are omitted.

LEMMA 3.2. *If $N \prec_{\mathcal{M}} M$ then $\mathcal{P}(N) \subseteq \mathcal{P}(M)$.*

LEMMA 3.3. *$\mathcal{P}(M)$ is closed under left and right quotients.*

Note that contrary to the situation with $\mathcal{M}$-recognition, $\mathcal{P}(M)$ is not closed under quotient by a non-singleton set of words (see [30, page 58]).

LEMMA 3.4. *Let θ be a polynomial length program defining $\theta : B^* \to A^*$. If $L \in \mathcal{P}(A^*, M)$ then $\theta^{-1}(L) \in \mathcal{P}(B^*, M)$.*

We say a morphism $\theta : B^* \to A^*$ is *length-multiplying* iff for some k, $|\theta(b)| = k$ for all $b \in B$.

COROLLARY 3.5. *$\mathcal{P}(\mathbf{V})$ is closed under (finite) Boolean operations, two-sided quotients and inverse length-multiplying morphisms.*

The class $\mathcal{P}(A^*, M)$ obviously includes the set $\mathcal{M}(A^*, M)$ of all regular subsets of A^* that are recognized by M in the sense of classical automata theory: in general it includes much more. For example, all regular languages belong to $\mathcal{P}(G)$ when G is a non-Abelian simple group; moreover we have mentioned that for any M, $\mathcal{P}(M)$ includes non-recursive languages. Nevertheless, it turns out that the regular languages in $\mathcal{P}(\mathbf{V})$ completely characterize the class. *This has the striking consequence that any two separable families $\mathcal{P}(\mathbf{V})$ and $\mathcal{P}(\mathbf{W})$ can be separated by a regular language.* To see this, recall from subsection 2.1, for a monoid M, the evaluation morphism $\eta_M : M^* \to M$ and the set $\mathcal{W}(M)$ of word problems over M.

LEMMA 3.6. *If $\mathcal{W}(M) \subseteq \mathcal{P}(\mathbf{V})$ then $\mathcal{P}(M) \subseteq \mathcal{P}(\mathbf{V})$.*

PROOF. Suppose $\mathcal{W}(M) \subseteq \mathcal{P}(\mathbf{V})$. Consider a language in $\mathcal{P}(A^*, M)$. Then this language is $\mathcal{P}$-recognized by an M-program $\sigma = (\sigma_n)$. Now this language can be written as a finite union of languages each of which is $\mathcal{P}$-recognized by (σ_n) using an accepting sequence (F_n) composed solely of the empty set and a fixed singleton. Hence by closure of $\mathcal{P}(\mathbf{V})$ under finite union (Corollary 3.5), it suffices to argue that for each $m \in M$ we have $\sigma_n^{-1}(m) \in \mathcal{P}(\mathbf{V})$. Empty sets in the relevant accepting sequences can be inserted subsequently.

Now fix $m \in M$ and suppose that for some fixed $n \geq 1$ the program σ_n is determined by the sequence of instructions $\nu_1\nu_2 \ldots \nu_l$. Since $\mathcal{W}(M) \subseteq \mathcal{P}(\mathbf{V})$, there exists $N \in \mathbf{V}$ such that $\eta_M^{-1}(m)$ is $\mathcal{P}$-recognized by an N-program (ϕ_k) with an accepting sequence (H_k). Hence $M^l \cap \eta_M^{-1}(m) = \phi_l^{-1}(H_l)$. We construct an n-input N-program ψ_n from ϕ_l by replacing each instruction (i, f), $i \in [r]$, $f : M \to N$, by (j, h), where $\nu_i = (j, g)$ and $h : A \to N$ is defined by $h(a) = f(g(a))$. Thus, for any $x \in A^n$:

$$\begin{aligned} \psi_n(x) \in H_l \quad &\text{iff} \quad \phi_l(\nu_1(x)\nu_2(x)\ldots\nu_l(x)) \in H_l \\ &\text{iff} \quad \eta_M(\nu_1(x)\nu_2(x)\ldots\nu_l(x)) = m \\ &\text{iff} \quad \sigma_n(x) = m. \end{aligned}$$

Hence program ψ_n with accepting set H_l recognizes $\sigma_n^{-1}(m)$ so that a sequence $\psi = (\psi_n)$ constructed in this fashion accepts $\sigma^{-1}(m)$ and has polynomial length by virtue of the polynomial size of (σ_n) and of (ϕ_k). □

Denoting by *Reg* the set of all regular languages, we thus have:

THEOREM 3.7. $\mathcal{P}(\mathbf{V}) = \mathcal{P}(\mathbf{W})$ *iff* $\mathcal{P}(\mathbf{V}) \cap Reg = \mathcal{P}(\mathbf{W}) \cap Reg$.

PROOF. Only the "if" direction needs proof, so suppose that $\mathcal{P}(\mathbf{V}) \cap Reg = \mathcal{P}(\mathbf{W}) \cap Reg$. By symmetry it suffices to argue $\mathcal{P}(\mathbf{W}) \subseteq \mathcal{P}(\mathbf{V})$. Pick any $M \in \mathbf{W}$. Since languages in $\mathcal{W}(M)$ are regular and certainly

$$\mathcal{W}(M) \subseteq \mathcal{M}(M) \subseteq \mathcal{P}(M) \subseteq \mathcal{P}(\mathbf{W}),$$

we deduce $\mathcal{W}(M) \subseteq \mathcal{P}(\mathbf{V})$. By Lemma 3.6 this implies $\mathcal{P}(M) \subseteq \mathcal{P}(\mathbf{V})$. Hence $\mathcal{P}(\mathbf{W}) \subseteq \mathcal{P}(\mathbf{V})$. □

4. Algebraic classification of NC^1

In this section, we investigate in some detail the internal structure of non-uniform NC^1 as determined by the classes $\mathcal{P}(\mathbf{V})$. Our results divide into two groups: we present criteria under which distinct varieties merge, and then we establish a number of separation results. We are led finally to a natural conjecture whose verification would simultaneously imply most known results and settle several major open questions about NC^1 and non-uniform NC^1.

THEOREM 4.1. *[2] Any monoid variety* $\mathbf{V}$ *containing a non-Abelian simple group satisfies* $\mathcal{P}(\mathbf{V}) =$ *non-uniform* NC^1.

Theorem 4.1 is a strong "merging result" which states $\mathcal{P}(\mathbf{V}) = \mathcal{P}(\mathbf{W}) =$ non-uniform NC^1 for any two varieties $\mathbf{V}$ and $\mathbf{W}$ not contained in $\mathbf{M}_{\mathrm{sol}}$ (= the wreath product closure of aperiodic monoids and solvable groups). As already indicated, in order to study the internal structure of NC^1 we can thus restrict our attention to solvable monoids. The next theorem explains the relationship between wreath products by "Abelian counters" and the ability of monoids to "count recursively" in the context of M-programs.

THEOREM 4.2. a) *Let* $\mathbf{V}$ *be a non-trivial variety,* $t \neq 0$ *or* $q \neq 1$; *then* $\mathcal{P}(\mathcal{Q}_{t,q}\mathbf{V}) = \mathcal{P}(\mathbf{Com}_{t,q} * \bar{\mathbf{V}})$.

b) *If* $\mathbf{V}$ *is a non-trivial group variety, then* $\mathcal{P}(\mathcal{Q}_{t,q}\mathbf{V}) = \mathcal{P}(\mathbf{Com}_{t,q} * \mathbf{V})$.

PROOF. At the expense of introducing further automata theory, Theorem 4.2 could be obtained as a consequence of Theorem 2.8 by adapting an argument due to the second author [30, page 70]. However we give a direct proof here.

By Fact 2.6, $\mathbf{Com}_{t,q}{*}\mathbf{V}$ and $\mathbf{Com}_{t,q}{*}\bar{\mathbf{V}}$ are both contained in $\mathcal{Q}_{t,q}\mathbf{V}$, therefore it suffices to show that the language $L = [L_0, a_1, L_1, \ldots, a_r, L_r]_{i,t,q}$ belongs to $\mathcal{P}(A^*, \mathbf{Com}_{t,q}{*}\mathbf{V})$ when $L_0, \ldots L_r$ are in $\mathcal{M}(\mathbf{V})$. Let $C_{t,q}$ be the one-generated monoid generating the variety $\mathbf{Com}_{t,q}$, with its generator denoted c. Let M_i be the syntactic monoid of L_i with $\theta_i : A^* \to M_i$ being the syntactic morphism. Let S be a non-trivial monoid in $\mathbf{V}$ with $s \in S$, $s \neq 1_S$. Consider a word $x \in A^n$, a sequence $\sigma = (i_1, \ldots, i_r)$ with $1 \leq i_1 < \ldots < i_r \leq n$ and write $x = w_0 b_1 w_1 \ldots b_r w_r$ for the induced factorization of x: let $M = M_0 \times S \times M_1 \times \ldots \times S \times M_r$ and $\lambda : M \to C_{t,q}$ be defined by $\lambda(m) = 1_{C_{t,q}}$ for all $m \in M$. Construct the $(C_{t,q} \circ M)$-program $\phi_\sigma = (1, f_1) \ldots (n, f_n) = \phi_{0,\sigma}(i_1, f_{i_1})\phi_{1,\sigma} \ldots (i_r, f_{i_r})\phi_{r,\sigma}$ such that $\phi_{j,\sigma}(x) = (\lambda, (m_0, 1_S, m_1, \ldots, 1_S, m_r))$ with

$$m_k = \begin{cases} \theta_k(w_k) & \text{if } k = j, \\ 1_{M_k} & \text{otherwise,} \end{cases}$$

and $(i_j, f_{i_j})(x) = (\lambda, (1_{M_0}, s_1, 1_{M_1}, \ldots, s_r, 1_{M_r}))$ with

$$s_k = \begin{cases} s & \text{if } k = j \text{ and } b_k = a_k, \\ 1_S & \text{otherwise.} \end{cases}$$

Thus $\phi_\sigma(x) = (\lambda, (\theta_0(w_0), s_1, \theta_1(w_1), \ldots, s_r, \theta_r(w_r)))$, where

$$s_k = \begin{cases} s & \text{if } b_k = a_k, \\ 1_S & \text{otherwise.} \end{cases}$$

Case 1: $\mathbf{V}$ is a group-variety. Let $\psi_\sigma = \phi_\sigma(1, f)\phi_\sigma^{|M|-1}$, where $f : A \to C_{t,q} \circ M$ is defined for all $a \in A$ by $f(a) = (\mu, (1_{M_0}, 1_S, 1_{M_1}, \ldots, 1_S, 1_{M_r}))$ and $\mu : M \to C_{t,q}$ is such that

$$\mu((m_0, s_1, m_1, \ldots, s_r, m_r)) = \begin{cases} c & \text{if } m_k \in \theta_k(L_k) \text{ and } s_k = s \text{ for all } k, \\ 1_{C_{t,q}} & \text{otherwise.} \end{cases}$$

Thus $\psi_\sigma(x) = (\rho, (1_{M_0}, 1_S, 1_{M_1}, \ldots, 1_S, 1_{M_r}))$, where $\rho : M \to C_{t,q}$ has the property that

$$\rho(1_M) = \begin{cases} c & \text{if } w_k \in L_k \text{ and } b_k = a_k \text{ for all } k, \\ 1_{C_{t,q}} & \text{otherwise.} \end{cases}$$

Concatenating the programs ψ_σ for all sequences σ we get $\psi : A^n \to C_{t,q} \circ M$ such that $\psi(x) = (\rho, 1_M)$ where $\rho(1_M) = c^i$ if $|x|_{[L_0, a_1, L_1, \ldots, a_r, L_r]} \gamma_{t,q}\, i$. Hence L is $(C_{t,q} \circ M)$-recognizable.

Case 2: $\mathbf{V}$ contains non-groups. The argument here is very similar. The only difference comes in the way in which the "M-component" in the wreath product $C_{t,q} \circ M$ is reset in the forming of ψ_σ. Instead of tacking on $\psi_\sigma^{|M|-1}$ (which in the group case cancels the effect, on the M-component, of the leftmost ϕ_σ by raising the occurring element to the $|M|$th power), one tacks on the constant map which sends all elements of M to 1_M (these constant maps are available by definition of $\bar{\mathbf{V}}$). □

Here are some consequences of Theorem 4.2.

COROLLARY 4.3. *[8] For all $k \geq 2$, $AC_k^0 = \mathcal{P}(\mathbf{B}_k)$.*

PROOF. Appeal to Fact 2.4 and Theorem 2.8. □

Now define $\mathbf{G}_{1,q}$ as $\mathbf{Com}_{0,q}$ and, for each $k \geq 1$, $\mathbf{G}_{k+1,q}$ as $\mathbf{G}_{\mathrm{nil},q} * \mathbf{G}_{k,q}$.

COROLLARY 4.4. *For any q, q' having the same set of prime divisors and any $k \geq 2$, $CC_k^0(q) = \mathcal{P}(\mathbf{G}_{k,q}) = \mathcal{P}(\mathbf{G}_{k,q'}) = CC_k^0(q')$.*

PROOF. We know that $\mathbf{G}_{\mathrm{nil},q} * \mathbf{G}_{k,q} = \mathcal{Q}_{0,q}\mathbf{G}_{k,q}$ (see Fact 2.7) and that $\mathcal{Q}_{0,p}\mathbf{V} = \mathcal{Q}_{0,p^c}\mathbf{V}$ for any variety $\mathbf{V}$, p prime and $c \geq 2$ [30, page 140]. Furthermore, for any relatively prime integers r and s, $\mathcal{Q}_{0,r \cdot s}\mathbf{V} = \mathcal{Q}_{0,r}\mathbf{V} \times \mathcal{Q}_{0,s}\mathbf{V}$ [44]. Thus, for $k \geq 2$, $\mathbf{G}_{k,q} = \mathbf{G}_{k,q'}$ if q and q' have the same set of prime divisors. □

From Smolensky's result [41] and Theorem 4.2 we may deduce the following.

PROPOSITION 4.5. a) *For any non-trivial variety $\mathbf{V}$ and any prime p:*

$$\mathcal{P}(\mathbf{Com}_{0,p} * \mathbf{Com}_{0,p} * \bar{\mathbf{V}}) = \mathcal{P}(\mathcal{Q}_{0,p}(\mathcal{Q}_{0,p}\mathbf{V})) = \mathcal{P}(\mathcal{Q}_{0,p}\mathbf{V}) = \mathcal{P}(\mathbf{Com}_{0,p} * \bar{\mathbf{V}}).$$

b) *For any non-trivial group variety $\mathbf{V}$ and any prime p:*

$$\mathcal{P}(\mathbf{Com}_{0,p} * \mathbf{Com}_{0,p} * \mathbf{V}) = \mathcal{P}(\mathcal{Q}_{0,p}(\mathcal{Q}_{0,p}\mathbf{V})) = \mathcal{P}(\mathcal{Q}_{0,p}\mathbf{V}) = \mathcal{P}(\mathbf{Com}_{0,p} * \mathbf{V}).$$

PROOF. We only need to show the middle equality, the others being special cases of Theorem 4.2.

From Theorem 4.2 and Corollary 4.4, we see that the operation $\mathcal{Q}_{0,p}$ applied to any variety $\mathbf{V}$ has the effect of adding a level of MOD_p gates plus some constant Boolean operations to the circuits corresponding to $\mathbf{V}$.

Smolensky's technique of using polynomials to recognize languages [41] yields that any constant depth circuit using only MOD_p gates (of polynomial size fan-in) and constant fan-in Boolean gates is equivalent to a circuit having only one MOD_p gate (of polynomial size fan-in) with NC^0 circuits as entries. □

Notice that this is in fact a result on varieties, namely that $\mathcal{Q}_{0,p}(\mathcal{Q}_{0,p}\mathbf{V}) = \mathcal{Q}_{0,p}\mathbf{V}$ for any variety $\mathbf{V}$ (see [30, chap. 6] and [31]). As a special case we have the following.

COROLLARY 4.6. $CC^0(p) = CC^0(p^\alpha) = \mathcal{P}(\mathbf{G}_p) = \mathcal{P}(\mathbf{Com}_{0,p}*\mathbf{Com}_{0,p}) = CC^0_2(p)$ *for any prime* p *and any* $\alpha \geq 1$.

Our last merging result involves the reversal operator. For any $\mathbf{V}$, let $\mathbf{V}^\mathrm{R}$ denote the variety generated by the syntactic monoids of the languages L^R, where L belongs to $\mathcal{M}(\mathbf{V})$. In general $\mathbf{V} \neq \mathbf{V}^\mathrm{R}$, yet because programs can "scan their inputs" in arbitrary order, it is straightforward to show:

THEOREM 4.7. $\mathcal{P}(\mathbf{V} \vee \mathbf{V}^\mathrm{R}) = \mathcal{P}(\mathbf{V})$.

4.2. Splitting varieties. In this subsection we discuss known cases in which distinct varieties $\mathbf{V}$ and $\mathbf{W}$ lead to distinct $\mathcal{P}(\mathbf{V})$ and $\mathcal{P}(\mathbf{W})$, and we establish a number of new such results. We begin with the conceptually simple Abelian monoid varieties.

THEOREM 4.8. *If* $\mathbf{V}$ *and* $\mathbf{W}$ *are distinct commutative varieties then* $\mathcal{P}(\mathbf{V}) \neq \mathcal{P}(\mathbf{W})$.

PROOF. An Abelian variety is determined by the monoids $C_{t,1}$ and $C_{0,q}$ it contains.

Case 1: $C_{t,1} \in \mathbf{V} \setminus \mathbf{W}$. Let $L = \{x : |x|_a \geq t\} \subseteq \{a,b\}^*$. Clearly $L \in \mathcal{P}(\mathbf{V})$. Let $M \in \mathbf{W}$: we can suppose $M = C_{t_1,1} \times \ldots C_{t_r,1} \times C_{0,q_1} \times \ldots C_{0,q_s}$, with $\max t_i < t$. Any M-program can be written in the form $\phi = (1, f_1) \ldots (n, f_n)$. Let $\bar{t} = t - 1$ and $\bar{q} = \mathrm{lcm} q_i$. We choose n large enough so that $\bar{t} + \bar{q} + t$ instructions make use of the same function f. Among the set of corresponding positions, choose any two subsets of cardinality $\bar{t}$ and $\bar{t} + \bar{q}$ respectively: let x and y be the two input strings defined by setting the corresponding positions to a and all others to b: then, for some m, $\phi(x) = m + \bar{t} \times f(a) + (\bar{q} + t) \times f(b)$ and $\phi(y) = m + (\bar{t} + \bar{q}) \times f(a) + t \times f(b)$. Thus $\phi(x) = \phi(y)$ while $x \notin L$ and $y \in L$.

Case 2: $C_{0,q} \in \mathbf{V} \setminus \mathbf{W}$. Let $L = \{x : |x|_a \equiv 0 \pmod{q}\} \subseteq \{a,b\}^*$. Clearly $L \in \mathcal{P}(\mathbf{V})$. Let $M \in \mathbf{W}$: we can suppose $M = C_{t_1,1} \times \ldots C_{t_r,1} \times C_{0,q_1} \times \ldots C_{0,q_s}$, with q not dividing $\mathrm{lcm} q_i$. Let $\bar{t} = \max t_i$, $\bar{q} = \mathrm{lcm} q_i$ and c be the least non-negative integer such that $q \mid \bar{t} + \bar{q} + c$. We choose n large enough so that $2\bar{t} + 2\bar{q} + c$ instructions are using the same f. As above we define x and y by setting $\bar{t}+\bar{q}+c$ and $\bar{t}+2\bar{q}+c$ positions respectively to a: then, for some m, $\phi(x) = m+(\bar{t}+\bar{q}+c)\times f(a)+(\bar{t}+\bar{q})\times f(b)$ and $\phi(y) = m+(\bar{t}+2\bar{q}+c)\times f(a)+\bar{t}\times f(b)$. Hence $\phi(x) = \phi(y)$ while $x \in L$ and $y \notin L$. □

In the case of aperiodic monoids, Corollary 4.3 together with Sipser's proof that the AC^0_k hierarchy is infinite imply that the $\mathcal{P}(\mathbf{B}_k)$ hierarchy is infinite (this was also noted in [8]); moreover if $\mathbf{V} \supset \mathbf{A}$, then $\mathbf{V}$ includes a non-trivial cyclic group and $\mathcal{P}(\mathbf{V})$ contains the language MOD_q for some q, which is outside AC^0 by [17], hence:

THEOREM 4.9. a) *[17]* $\mathcal{P}(\mathbf{V}) \subseteq \mathcal{P}(\mathbf{A})$ *iff* $\mathbf{V} \subseteq \mathbf{A}$.

b) *[40]* $\mathcal{P}(\mathbf{B}_k) \subset \mathcal{P}(\mathbf{B}_{k+1})$ *for any* $k \geq 1$.

In view of Theorem 2.8 and Corollary 4.3, we see that, for any variety $\mathbf{V}$ the question of whether $\mathcal{P}(\mathbf{V}) = \mathcal{P}(\mathbf{B}_k)$ has been answered except for those varieties which are not contained in $\mathbf{B}_k$ and which do not contain $\mathbf{V}_{k+1}$ (as defined inductively in the statement of Theorem 2.8). We provide some partial answers pertaining to the internal structure of $\mathbf{B}_2$.

THEOREM 4.10. $\mathcal{P}(\mathbf{J}_k) \subset \mathcal{P}(\mathbf{J}_{k+1}) \subset \mathcal{P}(\mathbf{J}) \subset \mathcal{P}(\mathbf{R}) \subset \mathcal{P}(\mathbf{B}_2)$.

PROOF. The two leftmost strict inclusions in the chain can also be found in [28]. Fix $k \geq 1$. To show $\mathcal{P}(\mathbf{J}_k) \subset \mathcal{P}(\mathbf{J}_{k+1})$, we will prove that the language $Y = (\{0,1\}^*1)^{k+1}\{0,1\}^*$ does not belong to $\mathcal{P}(\mathbf{J}_k)$. ($Y \in \mathcal{M}(\mathbf{J}_{k+1})$ by Fact 2.5, hence $Y \in \mathcal{P}(\mathbf{J}_{k+1})$.) Suppose to the contrary that an M-program ϕ with $M \in \mathbf{J}_k$ accepts Y. Then, viewing program ϕ as $\phi : \{0,1\}^* \to M^*$ it follows from Fact 2.5 that, for each n, $\phi(Y \cap \{0,1\}^n)$ is a union of $\sim_k$-classes. We will show this to be impossible.

Let s be the number of words of length at most k over the alphabet M; for any $z \in M^*$, $[z]_{\sim_k}$ is determined by the set $\{u : |u| \leq k, u \text{ is a subword of } z\}$, hence there can be at most 2^s $\sim_k$-classes. For any $I \subseteq [n]$ let x_I be the binary string having 1 in position i iff $i \in I$. For any k-subset I of $[n]$ define $\chi(I)$ to be the set of subwords of length $\leq k$ appearing in $\phi_n(x_I)$; there are thus $\leq 2^s$ possible colors and, by Fact 2.1, the integer $n = Ramsey(s+k+1, k, 2^s)$ has the property that we can find a subset $I \subseteq [n]$ of cardinality $s+k+1$ such that all k-subsets contained in I have the same color, i.e. $\phi_n(x_J) = \phi_n(x_K)$ whenever J and K are k-subsets of I.

Suppose $I = \{i_1, \ldots, i_{s+k+1}\}$ and consider x_J where $J = \{i_1, \ldots, i_k\}$: then x_J has only k 1's in it and thus $x_J \notin Y$. We can find a subword z of length at most s in $\phi_n(x_J)$ such that $z \sim_k \phi_n(x_J)$. There is thus a set S of $\leq s$ positions in x_J such that every subword of length $\leq k$ of $\phi_n(x_J)$ has an occurrence which is induced by positions in S. The set $I \setminus S$ has cardinality $\geq k+1$; define y by setting $y_i = 1$ if $i \in I \setminus S$ and $y_i = (x_J)_i$ if $i \notin I \setminus S$: thus $y \in Y$ and y agrees with x_J in the positions of S. Hence, if u is a subword of length $\leq k$ in $\phi_n(x_J)$, there is an occurrence of u induced by positions of S and u must appear as a subword of y. Conversely if u appears as a subword of $\phi_n(y)$ let T be a set of $\leq k$ positions in y inducing an occurrence of u in $\phi_n(y)$. Then $|T \cap I| \leq k$ and consider any k-set $K \subseteq I$ such that $T \cap I \subseteq K$. Thus u appears as a subword of $\phi_n(x_K)$ and also as a subword of $\phi_n(x_J)$ since J and K have the same color. This shows that $\phi_n(x_J) \sim_k \phi_n(y)$, whereas $x_J \notin Y$, $y \in Y$: thus ϕ does not recognize Y. This concludes the proof that $\mathcal{P}(\mathbf{J}_k) \subset \mathcal{P}(\mathbf{J}_{k+1})$.

This also proves that $\mathcal{P}(\mathbf{J}_k) \subset \mathcal{P}(\mathbf{J})$ for each k. Indeed if some fixed $\mathcal{P}(\mathbf{J}_k)$ were equal to $\mathcal{P}(\mathbf{J})$ then $\mathcal{P}(\mathbf{J}_{k+1}) \subseteq \mathcal{P}(\mathbf{J}_k)$ would follow.

To prove that $\mathcal{P}(\mathbf{J}) \subset \mathcal{P}(\mathbf{R})$, we modify the above Ramsey argument slightly in order to show that the language $Y = c^*bA^*$ with $A = \{a, b, c\}$ cannot be accepted by an M-program with $M \in \mathbf{J}_k$ for any k. This suffices since Fact 2.5 implies $Y \in \mathcal{M}(\mathbf{R})$, hence $Y \in \mathcal{P}(\mathbf{R})$, and further if Y belonged to $\mathbf{J}$ then it would belong to $\mathbf{J}_k$ for some fixed k.

The Ramsey argument is adapted as follows. Let s be as before and set $n = Ramsey(s+3k+2, 2k, 2^s)$. Then pick $\{i_1, i_2, \ldots i_{s+3k+2}\} \subseteq [n]$ such that $\phi(x_1) \sim_k \phi(x_2)$ for any two inputs x_1 and x_2 of length n having the subword $(ab)^k$ occurring at $2k$ positions chosen from $\{i_1, i_2, \ldots i_{s+3k+2}\}$ (and letter c everywhere else). Then define x of length n having $(ab)^k$ at positions $i_{s+k+3}, i_{s+k+4}, \ldots, i_{s+3k+1}, i_{s+3k+2}$ and letter c everywhere else. Clearly $x \notin Y$. However the k-subwords in $\phi(x)$ are collectively determined by at most s of the positions $\{i_1, i_2, \ldots i_{s+3k+2}\}$ (and some positions outside $\{i_1, i_2, \ldots i_{s+3k+2}\}$) in x, so that there remain at least $k+2$ "free" positions to the left of i_{s+k+3} within $\{i_1, i_2, \ldots i_{s+3k+2}\}$: we set position i_{s+k+2} in x to b and call the resulting input word y. Then $y \in Y$. However the same argument as before shows that $\phi(x) \sim_k \phi(y)$: by having kept $k+1$ "free" positions to the left of the b inserted to obtain y we maintain the ability to insert an a to the left of this b when necessary to complete any k-subword of y (including a k-subword of y involving as many as k of the $k+1$ free positions) into a length-n word having $(ab)^k$ at $2k$ positions in $\{i_1, i_2, \ldots i_{s+3k+2}\}$ (and c everywhere else). This concludes the proof that $\mathcal{P}(\mathbf{J}) \subset \mathcal{P}(\mathbf{R})$.

Finally we prove $\mathcal{P}(\mathbf{R}) \subset \mathcal{P}(\mathbf{B}_2)$. Let $A = \{a, b, c\}$ and let $L = (c^*ac^*bc^*)^*$; $L \in Rat(\mathbf{B}_2)$ and we claim that $L \notin \mathcal{P}(\mathbf{R})$. Suppose to the contrary that $L \in \mathcal{P}(\mathbf{R})$. Then, for some $M \in \mathbf{R}$, an M-program (ϕ_n) with accepting sequence (F_n) accepts L. Now fix n and view $\phi_n : A^n \to M^*$. Consider the regular language $K = \eta_M^{-1}(F_n)$. Since $K \in \mathcal{M}(M^*, M)$ and $M \in \mathbf{R}$, it is known [34, page 112] that K can be written as a disjoint union of products of the form $B_0^* b_1 B_1^* \dots b_s B_s^*$ where $b_i \in M$, $B_i \subseteq M \setminus \{b_{i+1}\}$, and $B_s \subseteq M$; we say that such a product has degree s and we observe that it is unambiguous i.e. if a word x belongs to the product there is a unique factorization of x as $x_0 b_1 x_1 \dots b_s x_s$ with $x_i \in B_i$.

Let $x \in L \cap A^n$, then $\phi_n(x) = w_0 b_1 w_1 \dots b_s w_s$ with $w_i \in B_i^*$ for some $K' = B_0^* b_1 B_1^* \dots b_s B_s^*$ in the disjoint union expressing K. Let $\phi_n = \psi_0 \nu_1 \psi_1 \dots \nu_s \psi_s$ be the corresponding factorization of ϕ_n: suppose another word $y \in L \cap A^n$ is such that $\phi_n(y) = z_0 b_1 z_1 \dots b_s z_s$ is also in K' and induces the same factorization of ϕ_n. Write $x = udv$ and $y = uev'$ with d and e distinct letters: we claim that $\phi_n(uev)$ is in K' and induces the same factorization: indeed $\nu(uev) = \nu(x)$ or $\nu(uev) = \nu(y)$ for each instruction of ϕ_n, so $\psi_i(x) \in B_i^*$ iff $\psi_i(uev) \in B_i^*$ and $\nu_i(x) = b_i$ iff $\nu_i(uev) = b_i$. But it is easily verified that when $udv \in L$ and $d \neq e$ then uev cannot be in L. Thus any set of s instructions of ϕ_n can give rise to $\phi_n(x) = w_0 b_1 w_1 \dots b_s w_s$, $w_i \in B_i^*$, for a unique x. If $K = K_1 \cup \dots K_t$ and r is the maximum degree of the K_i, we would have at most $\begin{pmatrix} l \\ rt \end{pmatrix}$ words of L in $\phi_n^{-1}(K)$, where l is the length of ϕ_n. Since $|L \cap A^n|$ is exponential in n we get a contradiction. □

Consider now solvable groups:

THEOREM 4.11. a) *$\mathcal{P}(\mathbf{V}) \subseteq \mathcal{P}(\mathbf{G}_p)$ iff $\mathbf{V} \subseteq \mathbf{G}_p$.*

b) *$\mathcal{P}(\mathbf{G}_{\mathbf{nil},q}) = \mathcal{P}(\mathbf{G}_{\mathbf{nil},q'})$ iff q and q' have the same prime divisors.*

PROOF. If p is prime, it can be shown that a p-group cannot recognize the language AND [7] nor the language MOD_q if q is not a power of p [43]. Now if $\mathbf{V} \not\subseteq \mathbf{G}_p$ then either $\mathbf{V}$ contains a non-trivial aperiodic monoid (which can compute AND) or $\mathbf{V}$ contains a cyclic group of order q unequal to any power of p (which can compute MOD_q). This proves part a).

For part b), if q and q' have the same prime divisors then $\mathbf{G}_{\mathbf{nil},q} = \mathbf{G}_{\mathbf{nil},q'}$, proving one direction. The converse is proved using [7, Theorem 6] and an adaptation of a subsequent lemma [7, page 121], to wit: Let G be a nilpotent group of exponent q and nilpotency class m and suppose that $L \subseteq \{0,1\}^*$ belongs to $\mathcal{P}(G)$ by means of the sequence (ϕ_n) of n-input G-programs. Then for n sufficiently large, any input $w \in \{0,1\}^n$ is such that among the n input bits there exists a subset S of $q^{1+\lceil \log_2(m!) \rceil}$ identical bits such that $\phi_n(w) = \phi_n(w')$, for w' the input obtained from w by complementing all bits in S.

It follows that if prime p does not divide q then $MOD_p \notin \mathcal{P}(\mathbf{G}_{\mathbf{nil},q})$. On the other hand clearly $MOD_p \in \mathcal{P}(C_{0,q'}) \subseteq \mathcal{P}(\mathbf{G}_{\mathbf{nil},q'})$ for any q' which is a multiple of p, completing the proof. Note that the exponent $1 + \lceil \log_2(m!) \rceil$ of q was chosen because adapting the lemma in [7] requires that $\begin{pmatrix} q^{1+\lceil \log_2(m!) \rceil} \\ i \end{pmatrix}$ be divisible by q for $1 \leq i \leq m$; this is clearly the case since $\gcd(q^{1+\lceil \log_2(m!) \rceil}, i!) \leq q^{\lceil \log_2(m!) \rceil}$ for each such i. □

We also mention the following consequence of the work of Smolensky and Razborov. Let $\mathbf{M}_p$ be the largest variety such that the only groups belonging to the variety are p-groups.

THEOREM 4.12. *[41, 37]* $\mathcal{P}(\mathbf{V}) \subseteq \mathcal{P}(\mathbf{M}_p)$ *iff* $\mathbf{V} \subseteq \mathbf{M}_p$.

4.3. The main conjecture.

Theorem 2.8 asserts that the natural subclasses of ACC^0 can each be characterized in the form $\mathcal{P}(\mathbf{Com}_{t,q} * \mathbf{V})$ for some appropriate variety $\mathbf{V}$ consisting of solvable monoids only. It is thus crucial to understand what can be done by a polynomial-length program over a monoid $C_{t,q} \circ T$ with T solvable. Because of Theorem 3.7, we need only concentrate on the regular languages that can be recognized in this way.

The computing power of a morphism over $C_{t,q} \circ T$ is well-understood. Essentially the morphism can only count, with respect to $\gamma_{t,q}$, the number of times that an input $x \in A^*$ can be factorized as $x = x_0 a x_1$, with $a \in A$, $x_0 \in L_0$, where L_0 is $\mathcal{M}$-recognized by T. More intuitively, given input $x = a_1 \ldots a_n$, the morphism "looks" at the n factorizations $x = (a_1 \ldots a_{i-1}) a_i (a_{i+1} \ldots a_n)$ and counts how many times $a_i = a$ and $a_1 \ldots a_{i-1} \in L_0$.

The proof of Theorem 4.2 shows that a program of length $O(n^c)$ over $C_{t,q} \circ T^{2c+1}$ can be used to count, with respect to $\gamma_{t,q}$, the number of times that an input x of length n can be factorized as $x = x_0 a x_1 \ldots a_c x_c$, where each $a_i \in A$, each $x_i \in L_i$ for some L_i $\mathcal{M}$-recognized by T. Intuitively, the $O(n^c)$ intructions are used to "look" at the $O(n^c)$ c-tuples of positions in the input, and the program counts how many times the induced factorization belongs to $L_0 a_1 L_1 \ldots a_c L_c$.

We conjecture that this feature characterizes the computing power of polynomial length programs over monoids in $\mathbf{Com}_{t,q} * \mathbf{V}$ when $\mathbf{V}$ contains only solvable monoids.

CONJECTURE. Let $\mathbf{V}$ be contained in $\mathbf{M}_{\mathbf{sol}}$, $t > 0$ or $q > 1$, then $\mathcal{P}(\mathbf{W}) \subseteq \mathcal{P}(\mathcal{Q}_{t,q}\mathbf{V})$ implies $\mathbf{W} \subseteq \mathcal{Q}_{t,q}\mathbf{V}$.

Verifying this conjecture would yield the following corollaries:

COROLLARY 4.13. *If the above conjecture is true then:*

a) $AC^0 \subset ACC^0(q) \subset$ *non-uniform* NC^1 *for all* $q > 1$;

b) $CC^0(q) \subset ACC^0(q)$ *for all* $q > 1$;

c) $ACC^0 \subset$ *non-uniform* NC^1;

d) $AC^0_k \subset AC^0_{k+1}$ *[40] for all* $k \geq 1$;

e) $ACC^0_k(q) \subset ACC^0_{k+1}(q)$ *for all* $k \geq 1$ *and* $q > 1$;

f) $CC^0_k(q) \subset CC^0_{k+1}(q)$ *for all* $k \geq 1$ *and* $q \neq p^l$ *for some prime* p *and some* $l \geq 1$;

g) AC^0_k *is incomparable to any class* $CC^0_k(q)$, *to* $CC^0(q)$ *or to* CC^0;

h) $CC^0_k(q)$ *and* $CC^0_k(q')$ *are incomparable when* q, q' *do not have the same prime divisors.*

PROOF. From Theorems 2.8 and 4.2, we may characterize each of the circuit classes in this corollary (except possibly non-uniform NC^1 which equals $\mathcal{P}(\mathbf{V})$ for any $\mathbf{V}$ containing a non-solvable group) by programs over varieties built up from $\mathbf{Com}_{t,q}$ using repeated applications of the operation $\mathcal{Q}_{t,q}$ for appropriate choices of $t \geq 0$ and $q \geq 1$. The statements in the corollary then follow from the conjecture using the following separation results on varieties:

a) Let $\mathbf{M_{sol}}(q)$ be the closure of $\mathbf{Com}_{1,q}$ by the operation $\mathcal{Q}_{t,q}$, and $\mathbf{M}$ be the variety of all finite monoids; then $\mathbf{A} \subset \mathbf{M_{sol}}(q) \subset \mathbf{M}$ for all $q > 1$;

b) Let $\mathbf{G_{sol}}(q)$ be the closure of $\mathbf{Com}_{0,q}$ by the operation $\mathcal{Q}_{0,q}$; then $\mathbf{G_{sol}}(q) \subset \mathbf{M_{sol}}(q)$ for all $q > 1$;

c) $\mathbf{M_{sol}} \subset \mathbf{M}$;

d) $\mathbf{J} \subset \mathbf{B}_k \subset \mathbf{B}_{k+1}$ for all $k \geq 2$;

e) Let $\mathbf{M_{sol,1}}(q) = \mathbf{Com}_{1,q}$ and $\mathbf{M_{sol,\mathit{k}}}(q) = \mathcal{Q}_{1,q}\mathbf{M_{sol,\mathit{k}-1}}(q)$ for $k \geq 1$; then $\mathbf{M_{sol,\mathit{k}}}(q) \subset \mathbf{M_{sol,\mathit{k}+1}}(q)$ for all $k \geq 1$ and all $q > 1$;

f) Let $\mathbf{G_{sol,1}}(q) = \mathbf{Com}_{0,q}$ and $\mathbf{G_{sol,\mathit{k}}}(q) = \mathcal{Q}_{0,q}\mathbf{G_{sol,\mathit{k}-1}}(q)$ for $k \geq 1$; then $\mathbf{G_{sol,\mathit{k}}}(q) \subset \mathbf{G_{sol,\mathit{k}+1}}(q)$ for all $k \geq 1$ and all $q > 1$ not a power of a prime;

g) $\mathbf{A}$ is incomparable to any group variety;

h) $\mathbf{G}_{\mathbf{sol}}(q)$ and $\mathbf{G}_{\mathbf{sol}}(q')$ are incomparable when q and q' do not have the same set of prime divisors.

For d) see [13]; for f) see for example [44]; all the other strict inclusions are more or less part of the folklore of semigroup theory.

Let us argue that the conjecture implies $ACC^0(q) \subset$ non-uniform NC^1 for all $q \geq 1$. Suppose for a contradiction that $ACC^0(q) =$ non-uniform NC^1 for some $q \geq 1$. Then $\mathcal{P}(\mathbf{M}) =$ non-uniform $NC^1 = ACC^0(q) = \mathcal{P}(\mathbf{M}_{\mathbf{sol}}(q)) \subseteq \mathcal{P}(\mathcal{Q}_{0,q}\mathbf{M}_{\mathbf{sol}}(q))$. By the conjecture, this implies $\mathbf{M} \subseteq \mathcal{Q}_{0,q}\mathbf{M}_{\mathbf{sol}}(q) = \mathbf{M}_{\mathbf{sol}}(q)$, contradicting a) above.

As another example, suppose to the contrary that $AC^0_k = AC^0_{k+1}$ for some $k \geq 1$. Then $\mathcal{P}(\mathbf{B}_{k+1}) = AC^0_{k+1} = AC^0_k = \mathcal{P}(\mathbf{B}_k) \subseteq \mathcal{P}(\mathcal{Q}_{1,1}\mathbf{B}_{k-1})$, with $\mathbf{B}_0$ the trivial variety and where the last step in the case $k = 1$ follows from $\mathbf{Com}_{1,1} \subseteq \mathcal{Q}_{1,1}\mathbf{B}_0 = \mathbf{J}$. Applying the conjecture, $\mathbf{B}_{k+1} \subseteq \mathcal{Q}_{1,1}\mathbf{B}_{k-1}$. When $k \geq 2$, $\mathcal{Q}_{1,1}\mathbf{B}_{k-1} = \mathbf{B}_k$, contradicting d) above. When $k = 1$, $\mathbf{J} \subset \mathbf{B}_2$ provides the contradiction.

All other consequences of the conjecture follow in a similar way. □

Our conjecture is closely related to other conjectures—formulated in terms of formal logic and which would characterize the regular languages in $ACC^0(q)$ (see [3]) and $CC^0(q)$ (see [43], and also [32])—in the sense that it would yield many of the same separation results. Notice though that the conjecture in this paper is much more fine grain (i.e., Corollary 4.13 d), e) and f)). However a finer grain conjecture in the logical framework may also be given (see [30]). Also, a proof of the logical conjectures would yield a proof of our conjecture and vice-versa.

5. Conclusion

When the classical notion of recognition by morphisms is extended to that of recognition by polynomial-length programs, the lattice of finite monoid varieties very naturally provides a detailed and elegant parametrization of non-uniform NC^1 and its subclasses. The difficult combinatorics underlying open questions like the status of ACC^0 relative to $ACC^0(q)$ or to non-uniform NC^1 do not vanish just because these classes suddenly fit in a global algebraic picture. Indeed the sceptic might be tempted to brush off the algebraic viewpoint on the sweeping grounds that the difficulty in answering open questions about NC^1 lies in the intricacies of the programs rather than in the algebraic properties of the monoids over which these programs are defined. But such an extreme position is untenable not only because we have listed here a myriad of (old and new) pairs of splitting varieties (see Section 4.2) but because for the "touchy" varieties involved in our Conjecture (see Section 4.3) the answers are simply not known. Barrington and Straubing have in fact shown recently that the power of programs of length $n \log \log n$ is completely determined by the algebraic properties of the underlying monoids and that such programs behave in a strong sense like morphisms [5].

One of the major problems in our algebraic classification lies in the fact that two different monoids or two different varieties of monoids may actually $\mathcal{P}$-recognize the same class of languages. We would thus like to determine what properties allow a monoid to have a stronger $\mathcal{P}$-recognizing power than another monoid. Obviously "being $\mathcal{M}$-divided by" is *sufficient*, as was noted in Lemma 3.2. However it is not *necessary*. Indeed we may have $\mathcal{P}(M) \subseteq \mathcal{P}(N)$ without having $M \prec_{\mathcal{M}} N$. A closer look at $\mathcal{M}$-division reveals the following property (see [30, page 54]): let $\eta : M^* \to M$ be the canonical morphism, then $M \prec_{\mathcal{M}} N$ iff for each $P \subseteq N$, N $\mathcal{M}$-recognizes $\eta^{-1}(P)$. This suggests the following definition: M *$\mathcal{P}$-divides* N iff for each $P \subseteq M$, N $\mathcal{P}$-recognizes $\eta^{-1}(P)$. This turns out to be the right definition of division at the level of programs. For example, we get that $M \prec_{\mathcal{P}} N$ iff $\mathcal{P}(M) \subseteq \mathcal{P}(N)$ [30, page 57]. This basic idea is used by the second author to develop a new theory of varieties of monoids and varieties of languages which is adapted to recognition by programs [30, chap. 3].

The most obvious question left open in this paper is to prove (or disprove) the far-reaching conjecture discussed in Section 4.3. Another open question is why the non-uniform NC^1 subclass TC^0 (defined in terms of bounded-depth circuits of MAJORITY gates [29]) is left out of the monoid-theoretic framework. No obvious monoid variety $\mathbf{V}$ seems to have the property that $TC^0 = \mathcal{P}(\mathbf{V})$ (unless, say, TC^0 in fact equals non-uniform NC^1). Yet Bédard, Lemieux and McKenzie prove that TC^0, as well as classes apparently larger than NC^1, can be captured by replacing monoids with (non associative) groupoids [9]. The generalization from "computation over monoids" to "computation over groupoids" leads to several further questions, including that of developing a theory of finite groupoids, perhaps along the lines of the well-developed theory of monoids which forms the basis of the work reported here.

Acknowledgements

Support from the Natural Sciences and Engineering Research Council of Canada through Grant A9979 (McKenzie), Grant A4546 (Thérien), and a Post-doctoral Fellowship (Péladeau), and support from the *Fonds pour la Formation de Chercheurs et l'Aide à la Recherche du Québec* through Grant 89-EQ-

2933 (McKenzie, Thérien) are gratefully acknowledged. This paper originates with the extended abstract: P. McKenzie and D. Thérien, Automata Theory Meets Complexity Theory, *Proc. of the 16th International Colloquium on Automata, Languages and Programming*, Springer Lecture Notes in Comp. Sci. 372 (1989), pp. 589-602. The first author acknowledges useful discussions with John Mullins.

References

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

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

[3] D. A. M. Barrington, K. Compton, H. Straubing, and D. Thérien, Regular languages in NC^1, Boston College Technical Report TR-BCCS-88-02, 1988, to appear in *J. Computer and Systems Science.*

[4] D. A. M. Barrington, N. Immerman, and H. Straubing, On uniformity within NC^1, *J. Computer and Systems Science* **41** (1990), 274-306.

[5] D. A. M. Barrington and H. Straubing, Linear-size bounded-width branching programs, Boston College Tech. Rep. BCCS 90-11, October 1990.

[6] D. A. M. Barrington, H. Straubing, and D. Thérien, unpublished manuscript, 1988.

[7] D. A. M. Barrington, H. Straubing, and D. Thérien, Non-uniform automata over groups, *Inform. and Computation* **89, 2** (1990), 109-132.

[8] D. A. M. Barrington and D. Thérien, Finite Monoids and the Fine Structure of NC^1, *J. Assoc Comput. Mach.* **35** (1988), 941-952.

[9] F. Bédard, F. Lemieux, and P. McKenzie, Extensions to Barrington's M-program model, *Proc. of the 5th Annual Structure in Complexity*

Theory Conf., IEEE Computer Society Press, 1990, 200-209, to appear in *Theoret. Comput. Science.*

[10] A. BORODIN, D. DOLEV, F. FICH, AND W. PAUL, Bounds for width two branching programs, *SIAM J. Comput.* **15** (1986), 549-560.

[11] J. A. BRZOZOWSKI AND I. SIMON, Characterizations of locally testable events, *Discrete Mathematics* **4** (1973), 243-271.

[12] J. A. BRZOZOWSKI AND F. FICH, Languages of R-trivial monoids, *J. Computer and Systems Science* **20** (1980), 32-49.

[13] J. A. BRZOZOWSKI AND R. KNAST, The dot-depth hierarchy of star-free languages is infinite, *J. Computer and Systems Science* **16** (1978), 37-55.

[14] A. CHANDRA, M. FURST, AND R. LIPTON, Multi-party protocol, *Proc. 15th Ann. ACM Symp. Theory of Computing*, 1983, 94-99.

[15] S.A. COOK, A taxonomy of problems with fast parallel solutions, *Inform. and Computation* **64** (1985), 2-22.

[16] S. EILENBERG, *Automata, Languages, and Machines,* Academic Press, Vol. A (1974), Vol. B (1976).

[17] M. L. FURST, J. B. SAXE, AND M. SIPSER, Parity, circuits, and the polynomial-time hierarchy, *Math. Systems Theory* **18** (1984), 13-27.

[18] R. GRAHAM, B. ROTHSCHILD, AND J. SPENCER, *Ramsey Theory,* Wiley, New York, 1980.

[19] J. T. HÅSTAD, *Computational Limitations for Small-Depth Circuits,* Ph. D. Thesis, M.I.T., ACM Doctoral Dissertation Awards, MIT Press, 1987.

[20] J. E. HOPCROFT AND J. D. ULLMAN, *Introduction to Automata Theory, Languages, and Computation,* Addison-Wesley, 1979.

[21] D. S. JOHNSON, The *NP*-completeness column: an ongoing guide, *J. Algorithms* **7** (1986), 289-305.

[22] S. C. KLEENE, Representation of events in nerve nets and finite automata, *Automata Studies*, (Shannon and McCarthy, eds), Princeton University Press, Princeton, 1954, pp 3-41.

[23] K. Ko, Relativized polynomial time hierarchies having exactly K levels, *SIAM J. Comput.* **18** (1989), 392-408.

[24] K. Krohn and J. L. Rhodes, Algebraic theory of machines, I. Prime decomposition theorem for finite semigroups and machines, *Trans. Amer. Math. Soc.* **116** (1965), 450-464.

[25] S. W. Margolis and J.-E. Pin, Inverse semigroups and varieties of finite semigroups, *J. Algebra* **110** (1987), 306-323.

[26] P. McKenzie and D. Thérien, Automata theory meets circuit complexity, Proc. 16th Intern. Colloquium on Automata, Languages and Programming, *Springer Lecture Notes in Comp. Sci.* **372**, 1989, 589-602.

[27] R. McNaughton, Algebraic decision procedures for local testability, *Math. Systems Theory* **8** (1974), 60-76.

[28] J. Mullins, *Programmes sur des petites variétés de monoïdes apériodiques*, Mémoire de maîtrise, Dép. I.R.O., Univ. de Montréal, 1988.

[29] I. Parberry and G. Schnitger, Parallel computation with threshold functions, *J. Computer and Systems Science* **36** (1988), 278-302.

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

[31] P. Péladeau, Sur le produit avec compteur modulo un nombre premier, to appear in *Revue Automatique Informatique et Recherche Opérationnelle – Informatique Théorique.*

[32] P. Péladeau, Formulas, regular languages and Boolean circuits, to appear in *Theoret. Comput. Science A.*

[33] N. Pippenger, On simultaneous resource bounds, *Proc. 20th IEEE Ann. Symp. Foundations of Computer Science*, 1979, 307-311.

[34] J.-E. Pin, *Variétés de langages formels,* Masson (1984). English version: *Varieties of formal languages*, Plenum, New York, 1986.

[35] J.-E. Pin, Finite group topology and p-adic topology for free monoids, Proc. 12th Intern. Colloquium on Automata, Languages and Programming, *Springer Lecture Notes in Comp. Sci.* **194**, 1985, 445-455.

[36] J.-E. Pin, H. Straubing, and D. Thérien, Locally trivial categories and unambiguous concatenation, *J. Pure Applied Algebra* **52** (1988), 297-311.

[37] A. A. Razborov, Lower bounds for the size of circuits of bounded depth with basis $\{\&, \oplus\}$, *Mathematicheskie Zametki* **41:4** (April 1987), 598-607 (in Russian). English translation *Mathematical Notes of the Academy of Sciences of the USSR* **41** (1987), 333-338.

[38] M. P. Schützenberger, On finite monoids having only trivial subgroups, *Inform. and Control* **8** (1965), 190-194.

[39] I. Simon, *Hierarchies of events of dot-depth one*, Ph. D. Thesis, University of Waterloo, 1972.

[40] M. Sipser, Borel sets and circuit complexity, *Proc. 15th Ann. ACM Symp. Theory of Computing*, 1983, 61-69.

[41] R. Smolensky, Algebraic methods in the theory of lower bounds for Boolean circuit complexity, *Proc. 19th Ann. ACM Symp. Theory of Computing*, 1987, 77-82.

[42] H. Straubing, *Varieties of recognizable sets whose syntactic monoids contain solvable groups*, Ph. D. Thesis, University of California at Berkeley, 1978.

[43] H. Straubing, Constant-depth periodic circuits, *Int. J. of Algebra and Computation,* **1** (1991), 49-88.

[44] D. Thérien, *Classification of regular languages by congruences*, Ph. D. Thesis, University of Waterloo, 1980.

[45] D. Thérien, Classification of finite monoids: the language approach, *Theoret. Comput. Science* **14** (1981), 195-208.

[46] D. Thérien, Subword counting and nilpotent groups, in *Combinatorics on Words: Progress and Perspectives* (L.J. Cummings ed.), Academic Press, 1983, 297-305.

[47] P. Weil, Product of languages with counter, to appear in *Theoret. Comput. Science*.

[48] A. C. Yao, Separating the polynomial-time hierarchy by oracles, *Proc. 26th IEEE Ann. Symp. Foundations of Computer Science*, 1985, 1-10.

[49] A. C. YAO, On ACC and threshold circuits, *Proc. 31st IEEE Ann. Symp. Foundations of Computer Science*, 1990, 619-627.

Manuscript received 10 May 1991

PIERRE MCKENZIE
Département d'informatique et
de recherche opérationnelle
Université de Montréal
C.P. 6128, succursale A
Montréal (Québec)
H3C 3J7 Canada
`mckenzie@iro.umontreal.ca`

PIERRE PÉLADEAU
School of Computer Science
McGill University
3480 University Street
Montreal (Quebec)
H3A 2A7 Canada
`pierre@opus.cs.mcgill.ca`

DENIS THÉRIEN
School of Computer Science
McGill University
3480 University Street
Montreal (Quebec)
H3A 2A7 Canada
`denis@cs.mcgill.ca`