FINITE SEMIGROUPS AND THE LOGICAL DESCRIPTION OF REGULAR LANGUAGES

HOWARD STRAUBING
Computer Science Department, Boston College
Chestnut Hill, Massachusetts
USA 02467
E-mail:straubin@cs.bc.edu

We survey the application of finite semigroups to the description of regular languages in first-order logic and generalized first-order logic. The emphasis is on recent results, including formulas with a bounded number of variables, and contacts with computational complexity and universal algebra.

1 Introduction

Formal languages can be defined by formulas of predicate logic and classified according to the types of languages used to define them. This idea was first applied to the languages recognized by finite automata by Büchi [3]. Büchi used formulas of second-order logic. When first-order logic and various modest generalizations thereof are used, finite semigroups become an important tool in this classification.

The present paper is an informal survey of this algebraic approach to what might be called the "descriptive theory of finite automata". I have written extensively about this subject in the monograph [10]. While the fundamentals of the theory will be presented, most of the emphasis will be on research that has appeared since the publication of [10].

2 Defining Regular Languages in Formal Logic

Let Σ be a finite alphabet. Through most of this article our examples will refer to the case where $\Sigma = \{\sigma, \tau\}$, but the results apply to general finite alphabets. Consider the following first-order sentence:

$$\forall x \forall y((x < y \wedge \forall z(z \leq x \vee y \leq z)) \rightarrow (Q_\sigma x \leftrightarrow Q_\tau y)).$$

*The author would like to acknowledge the financial support of Fundaçào Calouste Gulbenkian (FCG), Fundaçào para a Ciência e a Tecnologia (FCT), Faculdade de Ciências da Universidade de Lisboa (FCUL) and Reitoria da Universidade do Porto.

Such a sentence is meant to be interpreted in words over the alphabet Σ. The variables denote positions in the word (that is, integers between 1 and the length of the word, inclusive), and the formula $Q_\sigma x$ is interpreted to mean "the letter in position x is σ".Thus the whole sentence says, in effect, that any two consecutive positions contain different letters of Σ. Thus a word w *satisfies* the sentence if the letters σ and τ strictly alternate within w. The sentence consequently *defines* the language L consisting of all such words.

Observe that L is a regular language, and in fact is denoted by the regular expression

$$(\Lambda + \sigma)(\tau\sigma)^*(\Lambda + \tau).$$

It follows from Büchi's results, and is not difficult to prove directly, that any language defined by such a first-order sentence using the binary relation $<$ is a regular language. Let us pose the problem of determining whether a given regular language can be so defined. For instance, is there any regular language that is *not* first-order definable in this sense?

3 The McNaughton-Papert Theorem

We denote by $FO[<]$ the family of languages over Σ that are so definable. FO means "first-order", and the $<$ in brackets means that this is the only relation on positions—the only *numerical predicate*—that we use in our sentences. (Note that $x \leq y$ is equivalent to $\neg(y < x)$, so we can define both $\leq$ and $=$ in this logic.) We denote by $M(L)$ the syntactic monoid of the language L and by μ_L its syntactic morphism. The following theorem, due to McNaughton and Papert [7], is a fundamental result.

Theorem 3.1 *Let $L \subseteq \Sigma^*$. $L \in FO[<]$ if and only if $M(L)$ is finite and aperiodic.*

3.1 Using Model-theoretic Games

As a means of showing the reader the kinds of techniques that are used in this subject, we sketch the proof of Theorem 3.1. We begin by proving that every first-order definable language has an aperiodic syntactic monoid.

Let $k \geq 0$. We define the following equivalence realtion on Σ^*: $u \equiv_k v$ if and only if u and v satisfy exactly the same sentences in which the depth of nesting of the quantifiers is no more than k. Since there are only finitely many inequivalent sentences of a given depth, $\equiv_k$ is an equivalence relation on Σ^* of finite index.

Let $u, v \in \Sigma^*$. We define the *k-round Ehrenfeucht-Fraïßé game in* (u, v) as follows. Each player is equipped at the outset with k pebbles, labeled $p_1, \ldots p_k$. In the i^{th} round, Player I places his pebble labeled p_i on a position in one of the two words; Player II responds by placing her p_i on a position of the other word. Player II wins the game if, after k rounds, the following conditions are satisfied. First, whenever pebble p_i is to the left of pebble p_j in one word, p_i is to the left of p_j in the other word. Second, the letter in the position pebbled by p_i in u is the same as the letter in the position pebbled by p_i in v. Player I wins if Player II does not win.

As an example, let $u = \sigma\tau\sigma$ and $v = \tau\sigma\tau\sigma$. Player II has a winning strategy in the 1-round game in these two words, since the two words contain the same set of letters. But Player I has a winning strategy in the 2-round game: he can place p_1 on the initial τ in v. Player II is forced to respond on the second position of u. Player I then responds by playing p_2 on the first position of u, and Player II now has no safe move in v.

Here is the principal result about these games:

Theorem 3.2 *Let $u, v \in \Sigma^*$, $k \geq 0$. $u \equiv_k v$ if and only if Player II has a winning strategy in the k-round game in (u, v).*

As an illustration of this result, consider our previous example. Since Player I has a winning strategy in the 2-round game in u and v, there must be a sentence of quantifier depth 2 satisfied by one of these words and not the other. Such a sentence says "the first letter is τ". We can write it as

$$\exists x(Q_\tau x \wedge \forall y(y \geq x)).$$

Given this theorem, it is not hard to prove the following facts: First: If $u_1 \equiv_k v_1$, and $u_2 \equiv_k v_2$, then $u_1u_2 \equiv_k v_1v_2$. The proof is just the observation that winning strategies for Player II in the k-round games in (u_1, v_1) and (u_2, v_2) can be combined to make a winning strategy for the game in (u_1u_2, v_1v_2). Thus $\equiv_k$ is a congruence of finite index on Σ^*.

Second: For all $v \in \Sigma^*$, $k \geq 1$, $v^{2^k} \equiv_k v^{2^k-1}$. The winning strategy for Player II in these two words is constructed by induction on k. The statement is obvious for $k = 1$. In the inductive step, Player I's first move induces a factorization of the word he plays in as $w\sigma w'$, where $\sigma \in \Sigma$. If $|w| < |w'|$ then Player II factors the other word as $w\sigma w''$, and plays on the position containing σ. In subsequent rounds, if Player I plays in the prefix $w\sigma$ of either word, Player II plays on the corresponding position of the other word. If Player I plays in the factor w' or w'', the Player II responds according to her winning strategy in the $(k-1)$-round game in (w', w''), which exists by the

inductive hypothesis. The symmetric argument is made in the case where $|w'| < |w|$.

If L is defined by a sentence of quantifier depth k, then it is a union of $\equiv_k$-classes, and thus $M(L)$ is a homomorphic image of the quotient monoid $\Sigma^*/\equiv_k$. The two facts above imply that this quotient monoid is finite and aperiodic, and thus $M(L)$ is aperiodic. This gives us one direction of the McNaughton-Papert theorem.

3.2 Using the Krohn-Rhodes Theorem

The converse direction of Theorem 3.1 is proved using the following consequence of the Krohn-Rhodes Theorem: If M is aperiodic, then M divides an iterated wreath product

$$U \circ \cdots \circ U,$$

where U is the transformation monoid generated by the automaton over $\Sigma = \{\sigma, \tau\}$, with state set $Q = \{q_1, q_2\}$, such that $Q\sigma = q_1$, $Q\tau = q_2$.

Let L_1, L_2 be languages over Σ. Let $< L_1, L_2 >$ denote the set of all words uv such that $u \in L_1$, and every prefix uv' of uv is in L_2. It is not hard to show that if X is a transformation monoid, then every language recognized by the wreath product $U \circ X$ is a boolean combination of languages $< L_1, L_2 >$, where L_1 and L_2 are recognized by X. Secondly, if one has defining first-order sentences for L_1 and L_2, one can easily construct such a sentence for $< L_1, L_2 >$. It follows that if L is recognized by an aperiodic monoid, then it is first-order definable.

3.3 Non-expressibility and Decidability

An immediate corollary of the McNaughton-Papert Theorem is that any language whose syntactic monoid contains a nontrivial group is not in $FO[<]$. The simplest example is the language consisting of all words of even length.

More generally, we can effectively decide whether a given regular language is in $FO[<]$, since we can effectively compute its syntactic monoid and determine whether the monoid is aperiodic. Further, given a finite aperiodic monoid, we can effectively compute a Krohn-Rhodes decomposition and thus effectively construct a defining first-order sentence for the language.

4 A Gallery of Complementary Results

We give here a summary (not intended to be exhaustive) of results similar to the McNaughton-Papert Theorem. These concern variants of $FO[<]$ obtained

by changing the defining sentences along several different dimensions: The kinds of quantifiers used, the numerical predicates allowed, and the number of variables used.

4.1 First-order Sentences with Successor

We denote by $FO[+1]$ the family of languages definable by first-order sentences in which the successor relation $y = x+1$ is permitted, but the ordering relation $x < y$ is not. The following fundamental result is due to Beauquier and Pin [2].

Theorem 4.1 *Let $L \in \Sigma^*$ be a regular language. $L \in FO[+1]$ if and only if $M(L)$ is aperiodic, and*

$$esfs'es''f = es''fs'esf$$

for all $e, f, s, s', s'' \in \mu_L(\Sigma^+)$ with e, f idempotent.

Example. Let us return to our original example $L = (\Lambda+\sigma)(\tau\sigma)^*(\Lambda+\tau)$. We have already seen that L is first-order definable, and thus, by Theorem 3.1, $M(L)$ is aperiodic. In fact $M(L) = \{1, 0 = \sigma^2 = \tau^2, \sigma, \tau, \sigma\tau, \tau\sigma\}$, with $\sigma\tau$ and $\tau\sigma$ both idempotent. It follows that $\mu_L(\Sigma^+)$ (that is, the non-identity elements of $M(L)$) satisfy the identity in the statement of Theorem 4.1, and consequently $L \in FO[+1]$. In fact, a defining sentence is

$$\forall x \forall y(y = x + 1 \rightarrow (Q_\sigma x \leftrightarrow Q_\tau y)).$$

Example. Theorem 4.1 allows us to give an algebraic proof of the purely model-theoretic fact that $<$ is not first-order definable in terms of successor. Let $\Sigma = \{\rho, \sigma, \tau\}$, and consider the language $\rho^*\sigma\rho^*\tau\rho^*$. One shows easily, either by exhibiting a defining sentence or by computing the syntactic monoid, that $L \in FO[<]$. Let $e = f = \mu_L(\rho)$. (This is the identity of $M(L)$.) Let $s = \mu_L(\sigma)$, $s' = \mu_L(\rho)$, and $s'' = \mu_L(\tau)$. Then $esfs'es''f = \mu_L(\sigma\tau)$ and $es''fs'esf = \mu_L(\tau\sigma)$. The two elements are unequal, since $\sigma\tau \in L$ and $\tau\sigma \notin L$. Thus $L \notin FO[+1]$.

4.2 Modular Quantifiers

We introduce a new kind of quantifier, which we call a *modular quantifier*: Let $0 \leq r < n$. We interpret

$$\exists^{(r \bmod n)} x \phi(x)$$

to mean "the number of positions x for which $\phi(x)$ holds is congruent to r modulo n". For example,

$$\exists^{(0 \bmod 2)} x Q_\sigma x$$

defines the set of strings over $\{\sigma, \tau\}$ that contain an even number of occurrences of σ.

We denote by $MOD[<]$ the family of languages over Σ definable by sentences in which $<$ is the only numerical predicate and only modular quantifiers are used. $(FO + MOD)[<]$ denotes the family of languages definable by sentences in which ordinary quantifiers as well as modular quantifiers are allowed. The following theorem is due to Straubing, Thérien and Thomas [13]:

Theorem 4.2 *Let $L \subseteq \Sigma^*$ be a regular language. $L \in MOD[<]$ if and only if $M(L)$ is a solvable group. $L \in (FO + MOD)[<]$ if and only if every group in $M(L)$ is solvable.*

4.3 Regular Numerical Predicates

Let us admit into our defining formulas both the ordering relation and the predicates

$$x \equiv r \pmod{n},$$

where $0 \leq r < n$. We call the numerical predicates that are definable in terms of these *regular numerical predicates,* since any numerical predicate outside this class can be used to define non-regular languages. We denote by $FO[Reg]$ the family of languages over Σ definable by first-order sentences over this base of predicates. (See [10].) The following is due to Barrington, Compton, Straubing and Thérien [1]:

Theorem 4.3 *Let $L \subseteq \Sigma^*$ be a regular language. $L \in FO[Reg]$ if and only if for all $k > 0$, $\mu_L(\Sigma^k)$ contains no nontrivial groups.*

Example. The set of strings over Σ of even length is defined by the sentence

$$\forall x(\forall y(y \leq x) \rightarrow (x \equiv 0 \pmod{2})),$$

and is thus in $FO[Reg]$. Observe that while $M(L)$ is the group of order 2, $\mu_L(\Sigma^k)$ consists of a single element for each k. In contrast, the language consisting of all strings over $\{\sigma, \tau\}$ with an even number of occurrences of σ has the same syntactic monoid. But in this case, for every $k > 0$, $\mu_L(\Sigma^k)$ contains both elements of $M(L)$, and so, by the theorem, is not in $FO[Reg]$.

In [1] it is shown that any regular language definable by a first-order sentence with arbitrary non-regular numerical predicates is in $FO[Reg]$. This theorem depends upon (and is, in fact, a reformulation of) a deep result from circuit complexity.

4.4 Formulas with a Bounded Number of Variables

The language $\Sigma^*\sigma\sigma\Sigma^*\tau\Sigma^*$ is defined by the sentence

$$\exists x \exists y \exists z (Q_\sigma x \wedge Q_\sigma y \wedge Q_\tau z \wedge x < y \wedge y < z \wedge \forall w (w \leq x \vee y \leq w)).$$

Now observe that we can replace the variable w by z without changing the meaning of the formula—the new occurrences of z are bound by the innermost quantification and have nothing to do with the original use of z. Thus, by reusing variables, we have reduced the total number of variables in the sentence to three.. Remarkably, this can always be done: every language in $FO[<]$ is definable by a sentence in which only three variables occur. (Immerman and Kozen [6].)

What happens if we allow only two variables? We denote by **DA** the family of finite aperiodic monoids in which every regular $\mathcal{J}$-class is a subsemigroup. (Schützenberger [9].) We denote by $FO^k[<]$ the family of languages over Σ definable by k-variable first-order sentences over $<$. We use the notations $MOD^k[<]$ and $(FO + MOD)^k[<]$ analogously. The following theorem is a combination of results of Thérien and Wilke [14] and Pin and Weil [8]:

Theorem 4.4 *Let $L \subseteq \Sigma^*$ be a regular language. The following are equivalent:*
(a) L is definable by a first-order sentence over $<$ with only two variables.
(b) Both L and its complement are definable by Σ_2 sentences over $<$.
*(c) $M(L) \in$ **DA**.*

What happens when we add modular quantifiers to the mix? We extend to this case the notation used above, in which we denote by a superscript the number of variables allowed in our sentences. The following is due to Straubing and Thérien [12]:

Theorem 4.5 *(a)*

$$(FO + MOD)[<] = (FO + MOD)^3[<]$$

(b)

$$MOD[<] = MOD^2[<]$$

*(c) Let $L \subseteq \Sigma^*$ be a regular language. Then $L \in (FO+MOD)^2[<]$ if and only if $M(L)$ divides a wreath product $M \circ G$, where G is a finite solvable group and $M \in$ **DA**.*
(d) $L \in (FO + MOD)^2[<]$ if and only if both L and its complement are definable by Σ_2-sentences over $MOD[<]$. (That is, sentences whose atomic formulas are formulas quantified with modular quantifiers.)

Example. Let us look once again at our original example $L = (\Lambda + \sigma)(\tau\sigma)^*(\Lambda + \tau)$. $\mu_L(\sigma)$ belongs to the unique nontrivial regular $\mathcal{J}$-class J of $M(L)$, but $\sigma^2 = 0 \notin J$. Thus $M(L) \notin \mathbf{DA}$, and so L is not definable by a first-order sentence with two variables. But L is two-variable definable if we allow modular quantifiers. The sentence

$$\forall x(Q_\sigma x \leftrightarrow \exists^{0 \bmod 2} y(y \leq x)).$$

defines the set of strings $(\sigma\tau)^*(\sigma + \Lambda)$. The disjunction of this with the analogous sentence with σ replaced by τ defines L. The role played by the modular quantifiers in this sentence is rather remarkable. There are no groups in $M(L)$, so we do not need modular counting at all to define the language. Nonetheless, by including them, we are able to define the language more economically than would otherwise be possible.

Up until this point, all of our algebraic characterizations of languages defined by first-order and generalized first-order sentences have been effective in two senses: We have always had an algorithm to determine whether a given regular language is in the class of languages under consideration, and we have always had an algorithm to construct a defining sentence of the required type. However, we know of no algorithm for determining if a given finite monoid divides a wreath product of a monoid in **DA** and a solvable group. To see where the difficulty lies, we give an alternative characterization of this family of finite monoids. Let M be a finite monoid, and let J be a regular $\mathcal{J}$-class of M. If M divides a wreath product of a finite group and a monoid in **DA**, then it is possible to partition the set of $\mathcal{L}$-classes of J in such a manner that whenever s, t belong to the same block of the partition, and $m \in M$, then sm and tm are either both in J or both outside of J, and if they are both in J then they belong to the same block. We thus have a well-defined partial action of M on the set of blocks of this partition. Furthermore, this action is one-to-one. Now it turns out that M divides a wreath product of a solvable group with a monoid in **DA** if and only if the partial one-to-one action on the blocks of each J-class can be extended to a solvable permutation group. In fact, the decidability of membership in this class of finite monoids is equivalent to the decidability of the following question: Given a set $\mathcal{F}$ of partial one-to-one maps on a finite set Q, can $\mathcal{F}$ be extended to a solvable group of permutations on a superset of Q? This question remains open.

5 Connections with Computational Complexity

We earlier mentioned some connections with circuit complexity. These are discussed at length in [10]. Here we briefly discuss another contact with computational complexity.

In computational complexity, we usually take the underlying alphabet Σ to be $\{0,1\}$. Let $\mathcal{C}$ be a class of languages over this alphabet. We define

$L \in \exists \cdot \mathcal{C}$ if and only if there exists a polynomial p, and a language $K \in \mathcal{C}$, such that

$$x \in L \Leftrightarrow \exists y \in \Sigma^{p(|x|)}(xy \in K).$$

We define classes $\forall \cdot \mathcal{C}$ and $\oplus \cdot \mathcal{C}$ analogously, replacing $\exists$ in the definition by, respectively, $\forall$ and $\exists^{0 \bmod 2}$.

For example, if $\mathcal{P}$ denotes the class of polynomial-time languages, then $\exists \cdot \mathcal{P}$ is the class $\mathcal{NP}$, $\forall \cdot \mathcal{P}$ is the class co-$\mathcal{NP}$, and the closure of $\mathcal{P}$ under these two operators is the polynomial-time hierarchy $\mathcal{PH}$.

We also introduce a new computational model for language recognition: Let p be a polynomial, let $f : \Sigma^* \to M$ be a polynomial-time computable function, where M is a finite monoid, and let $X \subseteq M$. We define L to be the set of all strings w such that

$$\prod_{|z|=p(|w|)} f(wz) \in X,$$

where the product in M is taken in lexicographic order of the words z of length $p(|w|)$. We say that L is *polynomially recognized* by the monoid M.

Observe that $L \in \mathcal{NP}$ if and only if L is polynomially recognized by the monoid $\{0,1\}$ with $X = \{0\}$ as the set of accepting values, and that similarly, L is in co-$\mathcal{NP}$ if and only if it is so recognized with $\{1\}$ as the set of accepting values. It is possible to prove that L is recognized in this sense by a finite monoid if and only if L is in $PSPACE$, and that L is recognized by a finite aperiodic monoid if and only if L is in the polynomial-time hierarchy. (See, for example, Hertrampf, *et. al.* [5])

The following theorem follows from work of Toda [15]:

Theorem 5.1

$$\mathcal{PH} \subseteq \exists \cdot \forall \cdot \oplus\mathcal{P} \cap \forall \cdot \exists \cdot \oplus\mathcal{P}.$$

Observe the remarkable similarity between this result and our discussion of formulas with two variables and modular quantifiers. Once again, we can use modular counting to more efficiently express or recognize languages that do not in any intrinsic way require modular counting. The form of the result

even suggests our alternative characterization, in terms of Σ_2 formulas, of the two-variable definable languages. We suspect that the underlying algebra is the same; indeed, we conjecture that every language in the polynomial time hierarchy is polynomially recognized by a wreath product of a monoid in **DA** and the cyclic group of order 2.

6 Why Semigroups?

All the classes of regular languages considered in this article (and there are other examples as well) were defined in terms of the kinds of logical formulas used to express the languages, but were characterized in terms of the syntactic monoids of the languages. Why are the answers to these *logical* questions always *algebraic*? Here we outline a general explanation of the phenomenon, based on a generalization of Eilenberg's theorem connecting pseudovarieties of finite semigroups and monoids with varieties of regular languages. ([4].)

Let $\mathcal{C}$ be a class of homomorphisms between finitely generated free monoids such that *(a)* $\mathcal{C}$ is closed under composition, and *(b)* for each finite alphabet Σ, the identity homomorphism on Σ^* is in $\mathcal{C}$. $\mathcal{C}$ is consequently the class of morphisms of a *category* whose objects are the finitely generated free monoids. Examples include: $\mathcal{C}_{all}$, the class of all homomorphisms between finitely generated free monoids, $\mathcal{C}_{ne}$, the class of non-erasing homomorphisms (that is, homomorphisms $\phi : \Sigma^* \to \Gamma^*$ such that $\phi(\Sigma^+) \subseteq \Gamma^+$), and $\mathcal{C}_{lm}$, the class of length-multiplying homomorphisms–those for which there exists $k > 0$ such that $\phi(\Sigma) \subseteq \Gamma^k$.

Given such a class $\mathcal{C}$, we define a $\mathcal{C}$-pseudovariety of homomorphisms to be a family $\mathbf{V}$ of surjective homomorphisms $\phi : \Sigma^* \to M$, where M is a finite monoid, with the following properties:
(a) Let $\phi : \Sigma^* \to M$ be in $\mathbf{V}$, $f : \Gamma^* \to \Sigma^*$ in $\mathcal{C}$, and suppose there is a homomorphism α from $Im(\phi \circ f)$ onto a finite monoid N. Then $\alpha \circ \phi \circ f : \Gamma^* \to N$ is in $\mathbf{V}$.
(b) If $\phi : \Sigma^* \to M$ and $\psi : \Sigma^* \to N$ belong to $\mathbf{V}$, then so does $\phi \times \psi : \Sigma^* \to Im(\phi \times \psi) \subseteq M \times N$.

In this formalism, the $\mathcal{C}_{all}$ pseudovarieties are in essence identical to pseudovarieties of finite monoids, and the $\mathcal{C}_{ne}$-pseudovarieties to the pseudovarieties of finite semigroups. Given such a $\mathcal{C}$-pseudovariety $\mathbf{V}$, we define the corresponding $\mathcal{C}$-variety of languages, which associates to each finite alphabet Σ the family of regular languages L such that the syntactic morphism of L is in $\mathbf{V}$. As in the original theory of Eilenberg, the correspondence between $\mathcal{C}$-pseudovarieties and the associated varieties of languages is one-to-one.

The following result is due to the author [11]: Let $\mathcal{Q}$ be a class of quanti-

fiers, either FO, MOD, or $FO+MOD$. Let $k, d > 0$, and let $\mathcal{N}$ be one of the following classes of numerical predicates: equality, equality with successor, ordering, ordering with successor, or all regular numerical predicates. Let

$$\mathcal{Q}[k, d, \mathcal{N}]$$

associate to each finite alphabet Σ the family of languages over Σ^* defined by sentences using the given class of quantifiers, with quantifier depth d, no more than k variables, and numerical predicates in $\mathcal{N}$. Then

Theorem 6.1 *$\mathcal{Q}[k, d, \mathcal{N}]$ is a $\mathcal{C}$-variety of languages, where $\mathcal{C} = \mathcal{C}_{all}$ if $\mathcal{N}$ is equality or ordering, $\mathcal{C} = \mathcal{C}_{ne}$ if $\mathcal{N}$ is one of the classes containing successor, and $\mathcal{C} = \mathcal{C}_{lm}$ if $\mathcal{N}$ is the class of regular numerical predicates.*

It follows that membership in each of these logically defined classes depends only on the syntactic morphism of the language.

References

[1] D. Mix Barrington, K. Compton, H. Straubing, and D. Thérien, "Regular Languages in NC^1", *J. Comp. Syst. Sci.* **44** (1992) 478–499.

[2] D. Beauquier and J. E. Pin, "Factors of Words", *Proc. 16th ICALP,* Springer Lecture Notes in Computer Science **372** (1989) 63–79.

[3] J. R. Büchi,"Weak second-order arithmetic and finite automata", *Zeit. Math. Logik. Grund. Math.* **6** (1960) 66-92.

[4] S. Eilenberg, *Automata, Languages and Machines,* vol. B, Academic Press, New York, 1976.

[5] U. Hertrampf, C. Lautemann, T. Schwentick, H. Vollmer, K. Wagner, "On the Power of Polynomial-Time Bit Reductions", *Proc. 8th IEEE Conference on Structure in Complexity Theory* (1993) 200-207.

[6] N. Immerman and D. Kozen, "Definability with a Bounded Number of Bound Variables", *Information and Computation,* **83**, 121-139 (1989).

[7] R. McNaughton and S. Papert, *Counter-Free Automata,* MIT Press, Cambridge, Massachusetts, 1971.

[8] J.-E. Pin et P. Weil, "Polynomial closure and unambiguous product", *Theory Comput. Systems* **30**, 1-39, (1997).

[9] M. P. Schützenberger, "Sur le Produit de Concatenation Non-ambigu", *Semigroup Forum* **13** (1976), 47-76.

[10] H. Straubing, *Finite Automata, Formal Logic and Circuit Complexity,* Birkhäuser, Boston, 1994.

[11] H. Straubing, "On the logical characterization of regular languages" Under review.

[12] H. Straubing and D. Thérien, "Regular languages defined by generalized first-order formulas with a bounded number of bound variables", *Proc. 2001 STACS*.

[13] H. Straubing, D. Thérien, and W. Thomas, "Regular Languages Defined by Generalized Quantifiers", *Information and Computation* **118** 289-301 (1995).

[14] D. Thérien and T. Wilke, "Over Words, Two Variables are as Powerful as One Quantifier Alternation", *Proc. 30th ACM Symposium on Theory of Computing*, 256-263 (1988).

[15] S. Toda, "PP is as Hard as the Polynomial-Time Hierarchy", *SIAM J. Computing* **20** (1991) 865-877.