SYNTACTIC MONOIDS AND WORD PROBLEMS

Duncan W. Parkes
Department of Mathematics and Computer Science
University of Leicester
Leicester LE1 7RH, England
E-mail: dwp4@mcs.le.ac.uk

and

Richard M. Thomas*
Department of Mathematics and Computer Science
University of Leicester
Leicester LE1 7RH, England
E-mail: rmt@mcs.le.ac.uk

الخلاصة :

تناقش هذه الورقة بعض الارتباطات المثيرة بين نظرية الزمرة ونظرية اللغة الصورية ؛ فالموضوع الأساسي هنا هو المونوئيدات (وحيدات العملية) النحوية والمسائل التعبيرية في الزمر. ونتحدث حول ما يمكن للغات أن تميز بمونوئداتها النحوية، ومن ثم نربط نظرية المونوئدات النحوية بتلك المتعلقة بالإدراج والحذف في اللغات. وأخيراً ننتهي برسم أشكال بعض موضوعات البحث .

ABSTRACT

The purpose of this paper is to discuss some intriguing connections between group theory and formal language theory. The main topics considered here are syntactic monoids and word problems in groups. We will talk about the extent to which languages can be characterized by their syntactic monoids and relate the theory of syntactic monoids to that of insertions and deletions in languages. We finish off by drawing some of these themes together.

Key Words and Phrases: Disjunctive Subsets, Formal Languages, Insertions and Deletions, Syntactic Monoids, Word Problems.

AMS Mathematics Subject Classification:
Primary 20F10, 68Q70.
Secondary 20F05, 20M05, 20M35, 68Q45.

*To whom correspondence should be addressed.

SYNTACTIC MONOIDS AND WORD PROBLEMS

1 INTRODUCTION

The purpose of this paper is to discuss some intriguing connections between group theory and formal language theory. The main topics considered here are syntactic monoids and word problems in groups; we will explain these terms in Sections 2 and 3 respectively. We will talk about the extent to which languages can be characterized by their syntactic monoids (see Section 4 in particular) and relate the theory of syntactic monoids to that of insertions and deletions in languages (see Section 5). We finish off by drawing some of these themes together in Section 6.

This paper is intended to be reasonably self-contained; we will introduce the concepts from formal language theory we need and only assume some standard results from group theory. For further information about formal language theory, the reader is referred to [1–4], and, for group theory, to [5–8]. For some other papers surveying connections between group theory and formal language theory from a variety of perspectives, see [9–13]. Another interesting connection (which we will not explore here) between formal language theory on the one hand and groups and semigroups on the other is that of automatic groups and semigroups; see [14–21] for example.

2 PRELIMINARIES

Let Σ be a finite set or *alphabet*. The set of all finite words (or strings) over Σ (including the empty word λ) is denoted by Σ^*; to put this another way, Σ^* is the free monoid on the set Σ. The set of all non-empty strings over Σ (*i.e.* the free semigroup on Σ) is denoted by Σ^+. The subsets of Σ^* are known as *languages* over Σ. We shall denote the complement in Σ^* of the language L by L^c.

If v and w are words over an alphabet Σ then we shall use the expression $v \equiv w$ to mean that v and w are identical as strings of symbols. We shall write $|w|$ for the length of the word w.

A (*non-deterministic*) *finite automaton* M is a quintuple $(Q, \Sigma, \delta, s, F)$, where Q is a finite set of *states*, Σ is a finite set of *input symbols*, the *transition relation* δ is a subset of $Q \times (\Sigma \cup \{\lambda\}) \times Q$, the *start state* s is a special element of Q, and the set F of *accept states* is a subset of Q. We will abbreviate the expression "non-deterministic finite automaton" to NFA.

The transition relation δ may be extended inductively from a subset of $Q \times (\Sigma \cup \{\lambda\}) \times Q$ to a subset δ^* of $Q \times \Sigma^* \times Q$ in the following obvious way:

let (q, λ, q) be in δ^* for each $q \in Q$;

if $(q_1, x, q_2) \in \delta$ then let (q_1, x, q_2) be in δ^*;

if $(q_1, w, q_2) \in \delta^*$ and $(q_2, x, q_3) \in \delta$ then let (q_1, wx, q_3) be in δ^*.

An element of δ of the form (q, λ, q') is known as an *empty transition*.

We say that M *accepts* a word $w \in \Sigma^*$ if $(s, w, f) \in \delta^*$ for some $f \in F$. The set of words from Σ^* which are accepted by M is denoted by $L(M)$, and this is known as the *language accepted by* M. A language is said to be *regular* if it is accepted by an NFA. We denote the class of regular languages by $\mathcal{R}eg$.

A finite automaton is said to be *deterministic* if there are no empty transitions and if δ is a partial function from $Q \times \Sigma$ to Q (*i.e.*, for each pair $(q, x) \in Q \times \Sigma$, there is at most one state q' in Q such that $(q, x, q') \in \delta$). We write DFA for "deterministic finite automaton". We say that a DFA is *complete* if δ is a (total) function from $Q \times \Sigma$ to Q; it is well known that we may assume, without loss of generality, that our DFA is complete. It is also a standard result that any language which can be accepted by an NFA can be accepted by a DFA. In addition, given a regular language L, there is (up to isomorphism) a unique complete DFA M accepting L such

that M has the minimum number of states amongst all complete DFA's accepting L; M is known as the *minimal automaton* of L.

We can think of a finite automaton M as a device with a collection of states and an input tape which contains the word α. If we have a transition (q, a, r) in δ, then M may move from state q to state r whilst reading the symbol a on the input tape; if $a = \lambda$, then we do not read an input symbol. We start with M in the start state with the read head positioned over the leftmost cell of the input tape, and the word α is accepted if we can be in an accept state once all the input has been read.

We now extend the concept of a finite automaton by adding a "stack": the resulting machine is known as a "pushdown automaton".

A *pushdown automaton* (PDA) M is a septuple $(Q, \Sigma, \Gamma, \delta, s, \triangleright, F)$, where Q is a finite set of *states*, Σ is a finite set of *input symbols*, Γ is a finite set of *stack symbols*, the *transition relation* δ is a subset of

$$Q \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\}) \times Q \times (\Gamma \cup \{\lambda\}),$$

the *start state* s is a special element of Q, the *start symbol* $\triangleright$ is a special element of Γ, and $F \subseteq Q$ is the set of *accept states*.

As with a finite automaton, we have an input tape, but we now also have a stack, which is a tape with a leftmost cell but which is of unbounded length to the right. Initially the stack contains the special symbol $\triangleright$ on the leftmost cell, with the head positioned over that cell, and with the rest of the stack blank. The idea here is that the head is positioned over the rightmost non-blank cell of the stack at any stage. We may delete the contents of this cell and move the head left (provided that we are not scanning the leftmost cell of the stack); alternatively, we may move the head one cell to the right and write a new symbol in that cell; or we can combine these two operations together (so that we delete the contents of the current cell and replace it with a new symbol). If we have a quintuple (q, a, g, r, h) in δ, then, if we are in state q reading a symbol a on the input tape with a symbol g on the rightmost cell of the stack, we read a, we delete g from the stack, we write h on the stack, and we move to state r. We do not read a symbol if $a = \lambda$, we do not delete a symbol from the stack if $g = \lambda$, and we do not write a symbol to the stack if $h = \lambda$.

As with a finite automaton, a word α is accepted by a PDA M if we can be in an accept state once all the input has been read (we do not specify what the contents of the stack should be at the end of the computation). The set of words accepted by M is denoted by $L(M)$, and a language is said to be *context-free* if it is accepted by some PDA. We shall denote the class of context-free languages by $\mathcal{CF}$.

If M is a PDA such that, for any configuration of M, there is at most one possible transition that can be executed (in particular, δ must be a partial function from $Q \times (\Sigma \cup \{\lambda\}) \times (\Gamma \cup \{\lambda\})$ to $Q \times (\Gamma \cup \{\lambda\})$, although we need more than this), then M is said to be a *deterministic pushdown automaton* (DPDA), and $L(M)$ is then said to be a *deterministic context-free* language. The class of deterministic context-free languages will be denoted by $\mathcal{DCF}$. It is well known that there are context-free languages that are not deterministic context-free and there are deterministic context-free languages that are not regular, so that we have $\mathcal{Reg} \subset \mathcal{DCF} \subset \mathcal{CF}$.

We now come to the notion of a "Turing machine"; we will consider the deterministic model, although (as with finite automata) one does not increase the range of languages accepted if one allows non-determinism. As with the other types of automaton defined above, there are several variations in the definition of a Turing machine, and the following is one of the many that can be found in the literature.

A *deterministic Turing machine* (DTM) M is a sextuple $(Q, \Sigma, \Gamma, \delta, s, h)$ where Q is a finite set of *states*, Σ is a finite set of *input symbols*, Γ is a finite set of *work tape symbols* (including $\triangleright$ and Δ, where Δ is the "blank symbol"), the transition function δ is a partial function

$$Q \times (\Sigma \cup \{\triangleright, \triangleleft\}) \times \Gamma \to Q \times \{L, R, N\} \times \Gamma \times \{L, R, N\},$$

and the *start state* s and *halt state* h are two special elements of Q.

Our Turing machines have a read-only input tape which, for an input of length n, consists of $n+2$ tape cells, the first of which contains the "left-end-of-tape" marker $\triangleright$, the last of which contains the "right-end-of-tape" marker $\triangleleft$ and such that the cells in-between hold the input word. They also have a work tape on which we perform the computation. The work tape initially contains $\triangleright$ in its leftmost cell, is blank everywhere else and is unbounded in length to the right. We may move freely over the work tape and read from, and write to, the cells as we wish. Initially, the work head is positioned over the leftmost cell of the work tape (the cell initially holding the symbol $\triangleright$) and the input head is positioned over the cell adjacent to the leftmost cell of the input tape (the cell holding the first symbol of the input string, if the input string is non-empty, or over the cell holding the symbol $\triangleleft$ otherwise).

If $\delta(q,x,g) = (r,d_1,g',d_2)$, we imagine that, when M is in state q reading a symbol x on the input tape and g on the work tape, then M erases g and writes g' in its place, changes state to r, moves the input head in the direction indicated by d_1 (left if $d_1 = L$, right if $d_1 = R$, and not at all if $d_1 = N$), and moves the head on the work tape in the direction indicated by d_2. If ever M attempts to move left on either tape when reading the leftmost cell then M crashes, as it does if ever it attempts to move the input head right when reading the rightmost cell of the input tape; there is no transition defined from the halt state. If M is set up in the start state s with input word α, then α is said to be *accepted* if M reaches the halt state h and *rejected* otherwise (*i.e.* if the machine either crashes or else runs indefinitely without entering the halt state). The *language* $L(M)$ of M is the set of all words accepted by M; we say that a language L is *recursively enumerable* if $L = L(M)$ for some DTM M. We let $\mathcal{RE}$ denote the class of recursively enumerable languages.

However, with this notion, we may never know if the input word lies outside L as M may run indefinitely on some inputs. We can modify our definition of a Turing machine so as to have two halt states h_y and h_n, and then insist that such a machine halts if it enters either of these states. We also insist that, for any input α, the machine necessarily reaches one of these two states. We define the *yes-language* $Y(M)$ of such a Turing machine M to be the set of all input words such that M reaches h_y (and the *no-language* $N(M) = \Sigma^* - Y(M)$ of M to be the set of all input words such that M reaches h_n) and we call such a machine a *decision-making DTM.* We say that a language L is *recursive* if $L = Y(M)$ for some decision-making DTM M. It is a standard result that a recursive language is necessarily recursively enumerable, but not conversely. We let $\mathcal{R}ec$ denote the class of recursive languages.

One possible restriction on our model is where we limit the work tape to having the same number of cells as the input tape. Here, if we have an input of length n, so that the input tape has $n+2$ cells, then the work tape also consists of $n+2$ cells, the first of which contains $\triangleright$ and the last of which contains $\triangleleft$ and such that the cells between them are initially blank. Such a machine is called a *linear bounded automaton* (LBA), and a language accepted by an LBA is said to be *context-sensitive.* We denote the class of context-sensitive languages by $\mathcal{CS}$, and we have that

$$\mathcal{R}eg \subset \mathcal{DCF} \subset \mathcal{CF} \subset \mathcal{CS} \subset \mathcal{R}ec \subset \mathcal{RE}.$$

The chain $\mathcal{R}eg \subset \mathcal{CF} \subset \mathcal{CS} \subset \mathcal{RE}$ is known as the *Chomsky Hierarchy.*

Let $\mathcal{F}$ be a family of languages; then $\mathcal{F}$ is said to be *closed under inverse homomorphism* if

$$L \subseteq \Omega^*,\ L \in \mathcal{F},\ \phi : \Sigma^* \to \Omega^* \text{ a monoid homomorphism } \Rightarrow\ L\phi^{-1} \in \mathcal{F}.$$

We say that $\mathcal{F}$ is *closed under intersection with regular languages* if

$$L \subseteq \Sigma^*,\ L' \subseteq \Sigma^*,\ L \in \mathcal{F},\ L' \in \mathcal{R}eg \ \Rightarrow\ L \cap L' \in \mathcal{F}.$$

There is a useful table showing which of the classes of languages we have mentioned here are closed under various operations at [3, pages 280–281].

The *syntactic congruence* $\sim_L$ of a language $L \subseteq \Sigma^*$ is the coarsest congruence on Σ^* such that L is a union of congruence classes; we shall denote the congruence class of a word w under the syntactic congruence by $[w]$.

The *syntactic monoid* M_L of L is the quotient of Σ^* by $\sim_L$, and the *syntactic morphism* η_L is the natural homomorphism from Σ^* onto M_L, *i.e.* η_L maps w to $[w]$. We will summarize some properties of the syntactic congruence; proofs may be found, for example, in [4].

The following is a well known alternative characterization of the syntactic congruence (which is sometimes taken as the definition):

Proposition 2.1 *Let L be a language over Σ; the syntactic congruence $\sim_L$ is given by*

$$(w_1 \sim_L w_2) \iff \forall u, v \in \Sigma^*(uw_1v \in L \Leftrightarrow uw_2v \in L).$$

Since any congruence on Σ^* which has L as a union of congruence classes must also have L^c as a union of congruence classes, the following observation is clear:

Proposition 2.2 *The syntactic monoid of a language $L \subseteq \Sigma^*$ is equal to the syntactic monoid of its complement L^c.*

The syntactic monoid M_L is, in a sense, the smallest monoid M onto which there is a homomorphism such that the images of L and L^c are disjoint.

We say that a monoid M *recognizes* a language $L \subseteq \Sigma^*$ if there is a homomorphism $\phi : \Sigma^* \to M$ such that $L = A\phi^{-1}$ for some subset A of M. It is clear that a language is recognized by its syntactic monoid, since $L = (L\eta_L)\eta_L^{-1}$.

In fact we can say something more here. We first need another definition. If M_1 and M_2 are monoids then M_1 is said to *divide* M_2 if M_1 is a homomorphic image of some submonoid of M_2. We then have the following result:

Proposition 2.3 *Let $L \subseteq \Sigma^*$ be a language and M be a monoid; then M recognizes L if and only if M_L divides M.*

The minimal complete DFA accepting a regular language L is closely related to M_L; see [4] for details. One important point in all this is the following result:

Theorem 2.4 *A language is regular if and only if it has finite syntactic monoid.*

Another way of stating this result is to say that a language L is regular if and only if the syntactic congruence η_L has finitely many congruence classes. In fact, we have the following generalization of this:

Proposition 2.5 *If $L \subseteq \Sigma^*$ then L is regular if and only if there is a congruence $\sim$ on Σ^* such that $\sim$ has finitely many congruence classes on Σ^* and L is a union of congruence classes.*

A lot of very interesting work has been done on classifying various subclasses of the regular languages by means of their syntactic monoids (including Schützenberger's beautiful result in [22] that the "star-free" languages are precisely those that have finite syntactic monoids with no non-trivial subgroups), but we shall not look at this here. The reader is referred to [4, 23, 24], for example.

If M is a monoid then a subset A of M is said to be *disjunctive* (or *syntactic*) if there is no nontrivial congruence on M such that A is a union of congruence classes; in particular, the image of a language in its syntactic monoid is disjunctive. We then have:

Proposition 2.6 *If L is a language over Σ, M is a monoid, A is a disjunctive subset of M, and $\phi : \Sigma^* \to M$ is a surjective homomorphism such that $L = A\phi^{-1}$, then M is isomorphic to the syntactic monoid of L.*

3 WORD PROBLEMS

A set X, where each $x \in X$ represents an element of a group G, is said to be a *monoid generating set* for G if every element of G is represented by a word from X^*. Let X^{-1} be a new set of symbols $\{x^{-1} : x \in X\}$, where x^{-1} represents the inverse of the element represented by x (we tend to identify the symbol x with the element of G it represents); then X is said to be a *group generating set* for G if $X \cup X^{-1}$ is a monoid generating set for G. Given a word w over $X \cup X^{-1}$, we define w^{-1} in the obvious way.

Given a monoid generating set X for a group G, the *word problem* $W_X^m(G)$ of G with respect to X is the set of all words from X^* which are equal to the identity in G. The word problem $W_X^g(G)$ of G with respect to a group generating set X is then $W_{X \cup X^{-1}}^m(G)$.

It is more traditional to think of the word problem of a group G as being the question as to whether or not two words w_1 and w_2 over $X \cup X^{-1}$ represent the same element of G, or (equivalently) whether or not the word $w = w_1 w_2^{-1}$ represents the identity. If we define the word problem W of G to be the set of words that do represent the identity, the question reduces to that of determining whether or not the word w lies in W. This approach (considering the word problem to be a set of words) is more natural if we want to connect word problems in groups with classes of formal languages.

We can also talk about $W_X^m(M)$ where M is a monoid which is not a group. It should be noted that, in this case, knowing how to decide whether or not a word is in $W_X^m(M)$ does not necessarily give a solution to the full word problem for M. An extreme case of this is when we have a semigroup S generated (as a semigroup) by a set Y (*i.e.* every element of S is represented by a word in Y^+), and we add an identity element to S to form a monoid M. The only word of Y^* that represents the identity of M is λ, and this clearly tells us nothing whatsoever about the difficulty of the general word problem in M.

Another related way of thinking of the word problem is the following. Let G be a group and X be a finite alphabet, and then let $\phi : X^* \to G$ be a surjective homomorphism; then the image of X in G is a finite monoid generating set for G (as above, we shall generally just call this generating set X, identifying the set of formal symbols with their images), and the word problem of G with respect to X is just the kernel of ϕ.

Let $\mathcal{F}$ be a class of languages which is closed under inverse homomorphism, and let M be a finitely generated monoid. A subset A of M is said to be an *$\mathcal{F}$-subset* if, for any alphabet X and surjective homomorphism $\phi : X^* \to M$, we have that $A\phi^{-1} \in \mathcal{F}$. The independence of this concept with respect to choice of generating set and surjective homomorphism is provided by the following result from [25, 26] (see also [10]):

Lemma 3.1 *Let M be a finitely generated monoid, Σ and Ω finite alphabets, $\phi : \Sigma^* \to M$ a homomorphism, and $\psi : \Omega^* \to M$ a surjective homomorphism; then there is a homomorphism $\chi : \Sigma^* \to \Omega^*$ such that $\chi\psi = \phi$.*

Let A be an $\mathcal{F}$-subset of a monoid M, so that there is a monoid generating set X and a surjective homomorphism $\psi : X^* \to M$, with $A\psi^{-1} \in \mathcal{F}$. If Y is another alphabet and $\phi : Y^* \to M$ is another surjective homomorphism then, by Lemma 3.1, there is a homomorphism $\chi : Y^* \to X^*$ such that $\chi\psi = \phi$. We then have $A\phi^{-1} = A\psi^{-1}\chi^{-1}$, so that $A\phi^{-1}$ is an inverse image of $A\psi^{-1}$, and thus $A\phi^{-1} \in \mathcal{F}$ by the closure of $\mathcal{F}$ under inverse homomorphism.

If $\mathcal{F}$ is closed under inverse homomorphism and the word problem of G with respect to a finite monoid generating set X is in $\mathcal{F}$ then $\{1\}$ is an $\mathcal{F}$-subset of G; thus $W_Y^m(G) \in \mathcal{F}$ for any finite monoid generating set Y. In other words, we have:

Proposition 3.2 *Let X and Y be finite monoid generating sets for a group G and let $\mathcal{F}$ be a class of languages which is closed under inverse homomorphism; if $W_X^m(G) \in \mathcal{F}$ then $W_Y^m(G) \in \mathcal{F}$.*

In the light of this result, if $\mathcal{F}$ is a class of languages which is closed under inverse homomorphism, and the word problem of a group G with respect to some particular finite monoid generating set X lies in $\mathcal{F}$, then we may simply say that the word problem of G is in $\mathcal{F}$ (and write $W(G) \in \mathcal{F}$) without reference to any particular generating set, and we say that G is an $\mathcal{F}$-group.

4 CHARACTERIZATIONS OF LANGUAGES

As has been noted in [27], it is not possible to give a characterization of the context-free languages solely in terms of their syntactic monoids (in a similar way as was done for the regular languages in Theorem 2.4) since languages which are very different in terms of their position in the Chomsky Hierarchy can have the same syntactic monoid. For example, the context-free languages are not closed under complementation, and, by Proposition 2.2, a language always has the same syntactic monoid as its complement; therefore there are monoids which are the syntactic monoids of languages which are context-free and of languages further up the hierarchy. One can say rather more than this; the following is typical of the sort of result one can prove here:

Theorem 4.1 *Let G be a finitely generated group which contains an element of infinite order. Let $\mathcal{F}$ be a family of languages which is closed under inverse homomorphism and intersection with regular languages such that there exists a language $K \subseteq \{a\}^*$ with $K \notin \mathcal{F}$; then $G = M_L$ for some $L \notin \mathcal{F}$.*

Proof. Let a be an element of G of infinite order. Let X be a group generating set for G containing a, let $\Sigma = X \cup X^{-1}$, and then let $\phi : \Sigma^* \to G$ be the natural homomorphism.

Let K be a subset of $\{a\}^*$ such that $K \notin \mathcal{F}$, and let $I = \{i : a^i \in K\}$. If $\lambda \notin K$ and $K \cup \{\lambda\} \in \mathcal{F}$, then $(K \cup \{\lambda\}) \cap \{a\}^+ = K \in \mathcal{F}$, a contradiction; so, replacing K by $K \cup \{\lambda\}$ if necessary, we may assume that $\lambda \in K$, and thus that $0 \in I$. Let $S = \{a^i : i \in I\} \subseteq G$; note that, since $0 \in I$, we must have $1 \in S$. Let $L = S\phi^{-1}$. If $L \in \mathcal{F}$, then $K = L \cap \{a\}^* \in \mathcal{F}$, a contradiction; so $L \notin \mathcal{F}$.

In order to prove that $G = M_L$ we show that there is no non-trivial congruence on G such that S is a union of congruence classes. Let $\sim$ be a non-trivial congruence on G. Suppose that a^i and a^j are in S, with $i > j$, and that $a^j \sim a^i$; then $a^{j-i} \sim 1 \in S$, and so $a^{j-i} \in S$ with $j - i < 0$, a contradiction. □

In the light of the problems with using syntactic monoids to classify languages above the regular languages in the Chomsky Hierarchy, Sakarovitch suggests in [28] the framework of *syntactic pointed monoids*; the idea here is that languages should be classified by the structure of their syntactic monoid and by the image of the language in that monoid. Sakarovitch showed that, if two languages have the same syntactic monoid M and the same image in M, then each is the image of the other via an inverse homomorphism. It would therefore be useful to have methods of finding out whether or not a particular subset of a monoid is disjunctive. We will be particularly interested here in the case where our monoid is a group.

For every congruence $\sim$ on a group G, there is a normal subgroup N of G such that $x \sim y$ if and only if $Nx = Ny$; conversely, if we have a normal subgroup N of G, then defining $\sim$ in this way yields a congruence. A subset A of a group G is disjunctive if and only if there is no non-injective homomorphism ϕ from G onto a group K such that A is the full inverse image of a subset B of K. As a consequence, we have the following result:

Proposition 4.2 *Let G be a group, and A be a subset of G; then A is a disjunctive subset of G if and only if A is not the union of cosets of a non-trivial normal subgroup of G.*

Proof. Assume A is not a disjunctive subset of G, so that there is a non-trivial congruence $\sim$ on G with A a union of $\sim$-classes. If N is the associated normal subgroup of G, then N consists of all those elements n such that $n \sim 1$; since $\sim$ is a non-trivial congruence, we have that $N \neq \{1\}$. Recall that $x \sim y$ if and only if x and y lie in the same coset of N; since A is a union of $\sim$-classes, A is a union of cosets of N.

Conversely, assume that A is a union of cosets of a non-trivial normal subgroup N of G. Define the non-trivial congruence $\sim$ on G by $x \sim y$ if and only if $Nx = Ny$. Since A is the union of cosets of N, A is a union of $\sim$-classes, and so is not disjunctive. □

In the case where our subset A is a subgroup, we get the following immediate consequence:

Corollary 4.3 *Let G be a group, and H be a subgroup of G; then H is a disjunctive subset of G if and only if it contains no non-trivial normal subgroup of G.*

An immediate consequence of Corollary 4.3 is the following well-known observation:

Proposition 4.4 *A group is the syntactic monoid of its word problem.*

Proof. Let G be a group and ϕ be a homomorphism from Σ^* onto G. Let $L = \{1\}\phi^{-1}$, so that L is the word problem of G. Now $\{1\}$ is a subgroup of G which obviously contains no non-trivial normal subgroup of G; by Corollary 4.3, $\{1\}$ is a disjunctive subset of G, and so G is the syntactic monoid of L by Proposition 2.6. □

This gives us another proof of the following result from [29]:

Corollary 4.5 *The groups with regular word problems are exactly the finite groups.*

Proof. A finite group G is the syntactic monoid of its word problem which must therefore be regular by Theorem 2.4. Conversely, if G has regular word problem then it is the syntactic monoid of a regular language and hence is finite, again by Theorem 2.4. □

We can say rather more here. Suppose that G is a group and that A is any finite subset of G such that $A \in \mathcal{R}eg(G)$, say $\phi : X^* \to G$ is a surjective homomorphism such that $L = A\phi^{-1}$ is regular. If A is disjunctive then G is the syntactic monoid of L by Proposition 2.6 and so G is finite by Theorem 2.4. If A is not disjunctive, then A is the union of cosets of a non-trivial normal subgroup N of G by Proposition 4.2. Choose N to be a maximal such normal subgroup; since A is finite, N is also finite. We have a homomorphism $\theta : G \to G/N$; if B is the finite set $A\theta$, then B is disjunctive in G/N. We have a surjective homomorphism $\chi = \phi\theta : X^* \to G/N$ and $L = B\chi^{-1}$; so G/N is the syntactic monoid of the regular language L and therefore G/N is finite by Theorem 2.4, giving that G is finite.

If G is a finite group, A is any subset of G and $\phi : X^* \to G$ is a surjective homomorphism, then we have a congruence $\sim$ on X^* defined by $\alpha \sim \beta$ if and only if $\alpha\phi = \beta\phi$. Since $\sim$ has finitely many congruence classes and $A\phi^{-1}$ is a union of classes, $A\phi^{-1}$ is regular by Proposition 2.5, and so $A \in \mathcal{R}eg(G)$. So we have proved:

Theorem 4.6 *Let G be a finitely generated group, and let A be a finite non-empty subset of G such that $A \in \mathcal{R}eg(G)$; then every finite subset of G is regular.*

Some important contributions by Herbst to the theory of $\mathcal{F}$-subsets may be found in [30] and [31]. In particular, he proved the following result:

Theorem 4.7 *Let G be a finitely generated group and let A be a finite non-empty subset of G such that $A \in \mathcal{CF}(G)$; then every finite subset of G is deterministic context-free.*

So we have the analog of Theorem 4.6 for context-free (and for deterministic context-free) languages. In the case where G is residually finite, this result had previously been established by Sakarovitch in [28]. It is particularly interesting that, if the word problem of a group is context-free, then it is deterministic context-free; see [32–34] for more details. Moreover, Herbst also proved the following two results:

Theorem 4.8 *Let G be a finitely generated group, and let A be a finite non-empty disjunctive subset of G such that $A \in \mathcal{CS}(G)$; then every finite subset of G is context-sensitive.*

Theorem 4.9 *Let G be a finitely generated group, and let A be a finite non-empty disjunctive subset of G such that $A \in \mathcal{RE}(G)$; then every finite subset of G is recursively enumerable.*

It is not clear whether the hypothesis that A is disjunctive can be dropped in Theorems 4.8 and 4.9.

5 INSERTIONS AND DELETIONS

Given a language $L \subseteq X^*$, the word problem of the syntactic monoid M_L of L (with respect to the the generating set X) is the set of words in X^* which are equal to the identity in M_L, *i.e.* the congruence class $[\lambda]$ of the empty word under the syntactic congruence η_L.

Let $\mathcal{F}$ be a family of languages that is closed under inverse homomorphism. We are interested in exploring which groups can be syntactic monoids of languages in $\mathcal{F}$. The following observation essentially allows us to assume that our alphabet contains inverses:

Lemma 5.1 *Let $\mathcal{F}$ be a class of languages which is closed under inverse homomorphism, and let $L \subseteq X^*$ be a language in $\mathcal{F}$. If the syntactic monoid M_L of L is a group G, then G is also the syntactic monoid of a language K over the generating set $\Sigma = X \cup X^{-1}$ with $K \in \mathcal{F}$.*

Proof. Let S be the image of L in G under the syntactic morphism η_L (recall that $L = S\eta_L{}^{-1}$). Let ϕ be the monoid homomorphism from Σ^* to G which maps each $x \in X$ to $x\eta_L$, and each $x^{-1} \in X^{-1}$ to $(x\eta_L)^{-1}$. Since η_L is surjective, ϕ must also be surjective.

Let K be the inverse image of S under ϕ. Since S is a disjunctive subset of G, we know that G is the syntactic monoid of K. By Lemma 3.1, we must have that $K \in \mathcal{F}$. □

Let L be a language over an alphabet Σ. Then $\text{INS}(L)$ is defined to be the set of words which, when inserted at any point into a word from L, always result in another word from L, *i.e.*

$$\text{INS}(L) = \{w \in \Sigma^* : uv \in L \Rightarrow uwv \in L\}.$$

Let $\text{SUB}(L)$ be the set of all subwords of words of L, so that

$$\text{SUB}(L) = \{w \in \Sigma^* : uwv \in L \text{ for some } u, v \in \Sigma^*\}.$$

Then $\text{DEL}(L)$ is defined to be the set of words in $\text{SUB}(L)$ which, when they are deleted from any word in L, the resulting word always lies in L, *i.e.*

$$\text{DEL}(L) = \{w \in \text{SUB}(L) : uwv \in L \Rightarrow uv \in L\}.$$

The subsets $\text{INS}(L)$ and $\text{DEL}(L)$ are defined and studied in [35].

It is interesting to note the following characterization of the word problem of the syntactic monoid of a language:

Lemma 5.2 *Let L be a language over an alphabet X. Then the word problem W of the syntactic monoid of L is $\text{INS}(L) \cap \text{DEL}(L)$.*

Proof. We observe that $W = [\lambda]$, and so, by Proposition 2.1,

$$\begin{aligned} u \in W &\iff (w_1w_2 \in L \Leftrightarrow w_1uw_2 \in L) \\ &\iff (w_1w_2 \in L \Rightarrow w_1uw_2 \in L) \text{ and } (w_1uw_2 \in L \Rightarrow w_1w_2 \in L) \\ &\iff u \in \text{INS}(L) \text{ and } u \in \text{DEL}(L), \end{aligned}$$

which is exactly what we wanted. □

For two languages L_1 and L_2 over the alphabet Σ, the *dipolar deletion* $L_1 \rightleftharpoons L_2$ is defined by the equation:

$$\begin{aligned} L_1 \rightleftharpoons L_2 \;=\; \{x \in \Sigma^* :\; &\text{there exists } u \in L_1 \text{ and } v \in L_2 \\ &\text{such that } u \equiv \alpha x \beta \text{ and } v \equiv \alpha\beta\}. \end{aligned}$$

The following result from [35] gives the relationships between $\text{INS}(L)$, $\text{DEL}(L)$ and $L_1 \rightleftharpoons L_2$:

Proposition 5.3 *Let L be a language over Σ. Then*

i. $\text{INS}(L) = (L^c \rightleftharpoons L)^c$;

ii. $\text{DEL}(L) = (L \rightleftharpoons L^c)^c \cap \text{SUB}(L)$.

Suppose that $\Sigma = X \cup X^{-1}$ and define $^{-1} : \Sigma^* \to \Sigma^*$ in the obvious way. If A and B are subsets of Σ^*, then

$$\begin{aligned} (A \rightleftharpoons B)^{-1} &= \{x \in \Sigma^* : \exists \alpha, \beta \in \Sigma^* (u \equiv \alpha x \beta \in A \text{ and } v \equiv \alpha\beta \in B)\}^{-1} \\ &= \{x^{-1} \in \Sigma^* : \exists \alpha, \beta \in \Sigma^* (u \equiv \alpha x \beta \in A \text{ and } v \equiv \alpha\beta \in B)\} \\ &= \{x^{-1} \in \Sigma^* : \exists \alpha^{-1}, \beta^{-1} \in \Sigma^* (u^{-1} \equiv \beta^{-1}x^{-1}\alpha^{-1} \in A^{-1} \\ &\qquad \text{and } v^{-1} \equiv \beta^{-1}\alpha^{-1} \in B^{-1})\} \\ &= \{y \in \Sigma^* : \exists \gamma, \delta \in \Sigma^* (\gamma y \delta \in A^{-1} \text{ and } \gamma\delta \in B^{-1})\} \\ &= A^{-1} \rightleftharpoons B^{-1} \end{aligned}$$

and

$$\begin{aligned}(A^c)^{-1} &= \{x \in \Sigma^* : x \notin A\}^{-1} \\ &= \{x^{-1} \in \Sigma^* : x \notin A\} \\ &= \{x^{-1} \in \Sigma^* : x^{-1} \notin A^{-1}\} \\ &= \{y \in \Sigma^* : y \notin A^{-1}\} \\ &= (A^{-1})^c,\end{aligned}$$

and so we have the following result:

Lemma 5.4 *Let A and B be subsets of Σ^*. Then*

i. $(A \rightleftharpoons B)^{-1} = A^{-1} \rightleftharpoons B^{-1}$;

ii. $(A^c)^{-1} = (A^{-1})^c$.

We can now show the following:

Proposition 5.5 *Let L be a language over the alphabet $\Sigma = X \cup X^{-1}$ such that M_L is a group G (where X is a group generating set for G). Let $I = \text{INS}(L)$ and $D = \text{DEL}(L)$; then $D = I^{-1} = \text{INS}(L^{-1})$.*

Proof. We must have $\alpha\alpha^{-1} \in W_\Sigma^m(G)$ for any $\alpha \in \Sigma^*$, so that $\alpha\alpha^{-1} \in I \cap D$ for any $\alpha \in \Sigma^*$ by Lemma 5.2.

Let $\alpha \in I$; then $u\alpha^{-1}v \in L$ implies that $u\alpha\alpha^{-1}v \in L$ (since $\alpha \in I$), and thus $uv \in L$ (since $\alpha\alpha^{-1} \in D$), and we see that $\alpha^{-1} \in D$.

Let $\alpha^{-1} \in D$; then $uv \in L$ implies that $u\alpha\alpha^{-1}v \in L$ (since $\alpha\alpha^{-1} \in I$), and thus $u\alpha v \in L$ (since $\alpha^{-1} \in D$), and we have $\alpha \in I$.

We now have that $D = \{\alpha^{-1} \in \Sigma^* : \alpha \in I\} = I^{-1}$, and thus,

$$D = I^{-1} = [(L^c \rightleftharpoons L)^c]^{-1} = [(L^{-1})^c \rightleftharpoons (L^{-1})]^c = \text{INS}(L^{-1})$$

by Lemma 5.4 and Proposition 5.3. □

In particular, given Lemma 5.2, we have:

Corollary 5.6 *If $L = L^{-1}$ is a language over the alphabet $\Sigma = X \cup X^{-1}$ and the syntactic monoid of L is a group G (where X is a group generating set for G), then $W_\Sigma^m(G) = \text{INS}(L) = \text{DEL}(L)$.*

6 WORD PROBLEMS OF SYNTACTIC MONOIDS

Let $\mathcal{F}$ be a family of languages which is closed under inverse homomorphism. One general question which we shall look at is the following:

Question 6.1 *Given a language $L \in \mathcal{F}$ such that the syntactic monoid of L is a group, is the word problem of the syntactic monoid of L in $\mathcal{F}$?*

More generally, we have:

Question 6.2 *Given a language L from $\mathcal{F}$, what can we say about the word problem of M_L?*

Of course, the answers to these questions will depend on precisely which family $\mathcal{F}$ of languages we are considering. As Question 6.1 implies, we will be interested in the case where the syntactic monoid is a group. Lemma 5.1 tells us that, when trying to answer Question 6.1 in this situation, we may assume (without loss of generality) that the alphabet Σ contains inverses.

In the case where $\mathcal{F}$ is the class of regular langages, the answer to Question 6.1 is clear. If L is a regular language with syntactic monoid M, then $\{1\}\eta_L^{-1}$ is recognized by the finite monoid M, and so is regular. Theorem 4.7 gives a partial positive answer to Question 6.1 in the case of context-free languages:

Proposition 6.3 *Let L be a context-free language with syntactic monoid a group G. If the image of L in G is finite then G has a context-free word problem.*

Proof. Assume that $L \neq \emptyset$; then the image of L in G is a finite non-empty context-free set, and, by Theorem 4.7, all finite subsets of G are context-free; in particular $\{1\} \in \mathcal{CF}(G)$. If L is empty then its syntactic monoid is the trivial group, which obviously has context-free word problem. □

In general, however, the word problem of a group which is the syntactic monoid of a context-free language need not be context-free; a particular example from [36] is the following:

Example 6.4 *Let $\Sigma = \{a, b, c, d\}$, and let*

$$L = \{w \in \Sigma^* : |w|_a = |w|_b \text{ or } |w|_c = |w|_d\}.$$

Then L is a context-free language with syntactic monoid M_L isomorphic to the free abelian group $C_\infty \times C_\infty$ of rank 2, but the word problem of M_L is not context-free.

In the case of deterministic context-free languages, the answer to Question 6.1 appears to be unknown. We have the following question from [30]:

Question 6.5 *Let $L \subseteq \Sigma^*$ be a deterministic context-free language with syntactic monoid a group G; is the word problem of G necessarily context-free?*

Question 6.5 reduces to Question 6.1 for deterministic context-free languages, since, as we noted in Section 4, if the word problem of a group is context-free, then it is always deterministic context-free. In general, we say that a monoid M is *deterministic* if every context-free language whose syntactic monoid is isomorphic to M is deterministic context-free, and there is at least one such language. Sakarovitch conjectured in [25, 37] that the thin syntactic monoids are exactly the deterministic monoids (a monoid is said to be *thin* if it is the union of subsets of the form uv^*w with $u, v, w \in M$). It is noted in [30] that, if the answer to Question 6.5 is positive, then this would lead to a proof of Sakarovitch's conjecture in the special case of groups.

ACKNOWLEDGEMENT

The second author would like to thank Hilary Craig for all her help and encouragement.

References

[1] S. Eilenberg, *Automata, Languages and Machines, Volumes A and B.* New York: Academic Press, 1974.

[2] M.A. Harrison. *Introduction to Formal Language Theory.* New York: Addison Wesley, 1978.

[3] J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation.* New York: Addison Wesley, 1979.

[4] J.M. Howie, *Automata and Languages.* Oxford University Press, 1991.

[5] D.L. Johnson, "Presentations of Groups", *London Mathematical Society Student Texts*, **15**. Cambridge University Press (1997).

[6] R.C. Lyndon and P.E. Schupp, "Combinatorial Group Theory", *Ergebnisse der Mathematik und ihrer Grenzgebiete*, **89**. Berlin: Springer-Verlag (1977).

[7] W. Magnus, A. Karass, and D. Solitar, *Combinatorial Group Theory.* New York: Dover Publications, 1976.

[8] J.J. Rotman, *Introduction to the Theory of Groups.* Berlin: Springer-Verlag, 1995.

[9] R.H. Gilman, "Formal Languages and Infinite Groups", in *Geometric and Computational Perspectives on Infinite Groups.* ed. G. Baumslag, D.B.A. Epstein, R.H. Gilman, H. Short, and C.C. Sims. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, **25**. American Mathematical Society, 1996, pp. 27–51.

[10] T. Herbst and R.M. Thomas, "Group Presentations, Formal Languages and Characterizations of One-Counter Groups", *Theoret. Comput. Sci.*, **112** (1993), pp. 187–213.

[11] K. Madlener and F. Otto, "Groups Presented by Certain Classes of Finite Length-Reducing String Rewriting Systems", in *Rewriting Theory and Applications (Bordeaux, 1987).* ed. P. Lescanne. *Lecture Notes in Computer Science*, **256**. Berlin: Springer-Verlag, 1987, pp. 133–144.

[12] S.E. Rees, "A Language Theoretic Analysis of Combings", in *Groups, Languages and Geometry.* ed. R.H. Gilman. *Contemporary Mathematics*, **250**. American Mathematical Society, 1999, pp. 117–136.

[13] I.A. Stewart and R.M. Thomas, "Formal Languages and the Word Problem for Groups", in *Groups St Andrews 1997 in Bath, Volume 2.* ed. C.M. Campbell, E.F. Robertson, N. Ruškuc, and G.C. Smith. *London Mathematical Society Lecture Note Series*, **261**. Cambridge University Press, 1999, pp. 689–700.

[14] G. Baumslag, S.M. Gersten, M. Shapiro, and H. Short, "Automatic Groups and Amalgams", *J. Pure Appl. Algebra*, **76** (1991), pp. 229–316.

[15] C.M. Campbell, E.F. Robertson, N. Ruškuc, and R.M. Thomas, "Direct Products of Automatic Semigroups", *J. Austral. Math. Soc.*, **69** (2000), pp. 19–24.

[16] C.M. Campbell, E.F. Robertson, N. Ruškuc, and R.M. Thomas, "Automatic Semigroups", *Theoret. Comput. Sci.*, **250** (2001), pp. 365–391.

[17] A.J. Duncan, E.F. Robertson, and N. Ruškuc, "Automatic Monoids and Change of Generators", *Math. Proc. Cambridge Philos. Soc.*, **127** (1999), pp. 403–409.

[18] D.B.A. Epstein, J.W. Cannon, D.F. Holt, S.V.F. Levy, M.S. Paterson, and W.P. Thurston, *Word Processing in Groups.* London: Jones and Bartlett (1992).

[19] F. Otto, "On s-Regular Prefix-Rewriting Systems and Automatic Structures", in *Computing and Combinatorics, Proceedings COCOON'99.* ed. T. Asano, H. Imai, D.T. Lee, S. Nakano, and T. Tokuyama. *Lecture Notes in Computer Science*, **1627**. Berlin: Springer-Verlag, 1999, pp. 422-431.

[20] F. Otto, "On Dehn Functions of Finitely Presented Biautomatic Monoids", *J. Autom. Lang. Comb.*, **5** (2000), pp. 405–420.

[21] F. Otto, A. Sattler-Klein, and K. Madlener, "Automatic Monoids versus Monoids with Finite Convergent Presentations", in *Rewriting Techniques and Applications - Proceedings RTA '98.* ed. T. Nipkow. *Lecture Notes in Computer Science*, **1379**. Berlin: Springer-Verlag, 1998, pp. 32–46.

[22] M. Schützenberger, "On Finite Monoids Having only Trivial Subgroups", *Information and Control*, **8** (1965), pp. 190–194.

[23] J.-E. Pin, "Finite Semigroups and Recognizable Languages: an Introduction", in *Semigroups, Formal Languages and Groups.* ed. J. Fountain. *NATO ASI Series C*, **466**. Dordrecht: Kluwer, 1995, pp. 1–32.

[24] J.-E. Pin, "Syntactic Semigroups", in *Handbook of Formal Languages, Volume 1.* ed. G. Rozenberg and A. Salomaa. Berlin: Springer-Verlag, 1997, pp. 679–746.

[25] J. Sakarovitch. "Monoïdes syntactiques et langages algébriques", *Thèse 3ème cycle math., Université Paris-VII*, Paris, 1976.

[26] J. Sakarovitch, "Syntaxe des langages de Chomsky", *Thèse Sc. Math. Université Paris-VII*, Paris, 1979.

[27] J.-F. Perrot and J. Sakarovitch, "A Theory of Syntactic Monoids for Context-Free Languages", in *Information Processing 77 (Proc. IFIP Congr., Toronto, Ont., 1977).* ed. B. Gilchrist. *IFIP Congr. Ser.*, **7**. Amsterdam: North Holland, 1977, pp. 69–72.

[28] J. Sakarovitch, "An Algebraic Framework for the Study of the Syntactic Monoids: Application to the Group Languages", in *Mathematical Foundations of Computer Science (Gdansk, 1976).* ed. A. Mazurkiewicz. *Lecture Notes in Computer Science*, **45**. Berlin: Springer-Verlag, 1977, pp. 510–516.

[29] A.V. Anisimov, "Some Algorithmic Problems for Groups and Context-Free Languages", *Kibernetica*, **2** (1972), pp. 4–11.

[30] T. Herbst, "On a Subclass of Context-Free Groups", *RAIRO Inform. Théor. Appl.*, **25** (1991), pp. 255–272.

[31] T. Herbst, "Some Remarks on a Theorem of Sakarovitch", *J. Comput. System Sci.*, **44** (1992), pp. 160–165.

[32] J.M. Autebert, L. Boasson, and G. Sénizergues, "Groups and NTS Languages", *J. Comput. System Sci.*, **35** (1987), pp. 243–267.

[33] D.E. Muller and P.E. Schupp, Groups, "The Theory of Ends, and Context-Free Languages", *J. Comput. System Sci.*, **26** (1983), pp. 295–310.

[34] D.E. Muller and P.E. Schupp, "The Theory of Ends, Pushdown Automata, and Second-Order Logic", *Theoret. Comput. Sci.*, **37** (1985), pp. 51–75.

[35] M. Ito, L. Kari, and G. Thierrin, "Insertion and Deletion Closure of Languages", *Theoret. Comput. Sci.*, **183** (1997), pp. 3–19.

[36] J. Sakarovitch, "Sur les groupes infinis, consideres comme monoïdes syntaxiques de langages formels", in *Séminaire d'Algèbre Paul Dubreil, 29ème année.* ed. M.P. Malliavin. *Lecture Notes in Mathematics*, **586**. Berlin: Springer-Verlag (1977), pp. 168–179.

[37] J. Sakarovitch, "Sur une propriété d'itération des langages algébriques déterministes", *Math. Systems Theory*, **14** (1981), pp. 247–288.

Invited Paper Received 5 December 2000.