Math. Proc. Camb. Phil. Soc. (2007), **143**, 271 © 2007 *Cambridge Philosophical Society*
doi:10.1017/S0305004107000606 *Printed in the United Kingdom*

The loop problem for monoids and semigroups

BY MARK KAMBITES

*School of Mathematics, University of Manchester, Manchester M*60 1*QD.*
e-mail: Mark.Kambites@manchester.ac.uk

(*Received* 12 *September* 2006*; revised* 15 *December* 2006)

Abstract

We propose a way of associating to each finitely generated monoid or semigroup a formal language, called its *loop problem*. In the case of a group, the loop problem is essentially the same as the *word problem* in the sense of combinatorial group theory. Like the word problem for groups, the loop problem is regular if and only if the monoid is finite. We also study the case in which the loop problem is *context-free*, showing that a celebrated group-theoretic result of Muller and Schupp extends to describe completely simple semigroups with context-free loop problems. We consider right cancellative monoids, establishing connections between the loop problem and the structural theory of these semigroups by showing that the syntactic monoid of the loop problem is the *inverse hull* of the monoid.

1. *Introduction*

One of the most productive and successful approaches to finitely generated groups is the study of the language of all words over a given finite generating set which represent the identity element. Many of the structural properties of a group are reflected in the language-theoretic properties of this language. For example, the language is regular, context-free, recursive or recursively enumerable exactly if the group is finite, virtually free, embeddable in a simple subgroup of a finitely presented group, or embeddable in a finitely presented group respectively [**1**, **4**, **13**, **21**].

In a group, words u and v over a symmetric generating set represent the same element exactly if uv^{-1} represents the identity. It follows that deciding membership for the language of words representing the identity is algorithmically equivalent to the *word problem* in the sense of universal algebra [**11**, section 30], that is, the problem of deciding whether two given words in the generators represent the same element of the group. In particular, the language encodes the multiplication, and hence the entire structure, of the group. For this reason, the language of words representing the identity is usually termed the *word problem* of the group.

By contrast, in a more general finitely generated monoid, there is no equivalence between the language of words representing the identity and the universal algebraic word problem. One need only consider the class of monoids obtained by adjoining identities to semigroups, to see that radically different monoids can give rise to the same language. It is perhaps for this reason that recent research in combinatorial semigroup theory has, with a few exceptions [**7**, **9**], eschewed direct interaction with formal language theory. Although automata appear with increasing frequency (as in the emerging theory of *automatic semigroups* [**5**, **16**]),

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

they are typically of a "multi-tape" form which does not admit a direct language-theoretic interpretation. However, in view of the numerous and deep interactions between combinatorial group theory and formal languages, and also between finite semigroup theory and formal languages (see, for example, [**22**]), it would be surprising if the theory of formal languages had no role to play in the combinatorial theory of finitely generated semigroups and monoids.

The purpose of this paper is to propose, and begin the study of, a new way of associating to each finitely generated monoid or semigroup a formal language, which we call the *loop problem*. Like the word problem for groups, this language encapsulates the entire structure of the semigroup, at least in the simplistic sense that it *uniquely determines* the semigroup. A less trivial question is that of whether there is a correspondence between "natural" structural properties of a semigroup or monoid, and the "natural" language-theoretic properties of its loop problem, and it is this question in which we are primarily interested.

In addition to this introduction, this paper comprises six sections. In Section 2, we briefly revise some elementary automata theory. Section 3 introduces the loop problem of a finitely generated monoid or semigroup, while Section 4 establishes some of its basic properties, including the extent to which it is invariant under change of generators.

Section 5 considers the case in which the monoid is a group, describing the precise relationship between the loop problem and the word problem. More generally, we study the important class of finitely generated completely simple semigroups, establishing a natural relationship between the loop problems of such semigroups and the loop problems (and word problems) of their maximal subgroups.

Section 6 begins the study of the interaction between natural structural properties of semigroups, and language-theoretic properties of their loop problems. We begin with a semigroup-theoretic analogue of a basic result of Anisimov [**1**] from combinatorial group theory – a monoid is finite if and only if its loop problem is regular. We consider also the class of context-free languages; we apply a result from Section 5 to show that a celebrated theorem of Muller and Schupp [**21**] from combinatorial group theory extends to characterise completely simple semigroups with context-free loop problems, and ask what can be said more generally about the class of semigroups with this property.

Finally, in Section 7, we consider the loop problem in the important special case of *right cancellative* monoids. It transpires that there is a simple and elegant relationship between the between the loop problem and the established structural theory of these monoids [**6**, **24**]. In particular, the syntactic monoid of the loop problem is the *inverse hull* of the monoid.

2. *Automata*

In this paper, a central role will be played by infinite automata over finite alphabets. It is well known, if not so well documented, that much of the classical theory of finite automata extends easily to the infinite case, provided sufficient care is taken with the definitions. In this section, we briefly introduce some fundamental definitions and results of automata theory, partly so as to make the paper accessible to the reader not familiar with classical automata theory, and partly to establish the necessary foundations in the infinite case. Proofs are not given; they can be found in the finite case in the standard textbooks on formal language theory [**14**, **22**], while the appropriate adjustments for the infinite case are left as an exercise for the reader.

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

Let X be a finite, non-empty set of symbols or *letters*, called an *alphabet*. A *word* over X is a finite sequence of letters from X; the *empty word*, with no symbols, is denoted ϵ. The collection of all words over X forms a monoid (with identity ϵ) under the operation of concatenation; it is called the *free monoid* on X and denoted X^*. The non-empty words comprise a subsemigroup of X^*, called the *free semigroup* on X and denoted X^+. It is readily verified that free monoids and semigroups satisfy the usual universal property defining free objects in the categories of monoids and semigroups respectively. A *language* over X is a collection of words over X, that is, a subset of the free monoid X^*.

An *automaton* over X consists of a directed graph with edges labelled by elements of X^*, with a distinguished *initial vertex* and a set of distinguished *terminal vertices*. The vertices and edges of an automaton are sometimes called *states* and *transitions* respectively. The labelling of edges extends naturally, via the multiplication in X^*, to a labelling of paths by words in X^*. The *language accepted* by the automaton is the set of all words in X^* which label paths from the initial vertex to a terminal vertex.

The automaton is called *finite* if the vertex and edge sets are finite; a language accepted by a finite automaton is called a *regular language*. Many textbooks on automata focus exclusively on the finite/regular case, and use the terms "automaton" and "finite automaton" interchangeably. In this article, we are equally interested in the infinite case, and an "automaton" may be infinite unless explicitly stated otherwise.

An automaton is called *deterministic* if its edges are labelled by elements of X, and for each state q and each letter $x \in X$, there is at most one edge leaving q with label x. An automaton is called *trim* if for every vertex q, there is a path from the initial state to q, and a path from q to some terminal state. Two automata are *isomorphic* if there is an isomorphism between their underlying graphs which preserves edge labels, the initial state and the set of terminal states.

Let Q be the state set of the automaton. For each word $w \in X^*$, one can define a binary relation

$$\sigma_w = \{(p, q) \mid \text{there is a path from } p \text{ to } q \text{ labelled } w\} \subseteq Q \times Q.$$

The set of all relations of the form σ_w constitutes a subsemigroup of the monoid of all binary relations on Q; the multiplication is given by $\sigma_u \sigma_v = \sigma_{uv}$. The relation σ_ϵ is easily seen to act as an identity for this semigroup (even though it may not be the identity function on Q), so in fact the semigroup is a monoid, called the *transition monoid* of the automaton. In the case that the automaton is deterministic, the relations σ_w are all partial functions, σ_ϵ is actually the identity function on Q, and the transition monoid is generated by the maps σ_x for letters $x \in X$.

The *cone* of a vertex q is the language of all words which label paths from q to some terminal vertex. Two vertices are called *equivalent* if they have the same cone.

PROPOSITION 2·1. *Let L be a language. Then there exists a unique (up to isomorphism) deterministic trim automaton $M(L)$ accepting L, with the property that no two vertices are equivalent.*

The automaton given by Proposition 2·1 is called the *minimal automaton* of the language L. It is a final object in the category of all deterministic trim automata accepting L with an appropriate notion of morphism. An easy corollary is that L is regular exactly if $M(L)$ is finite. Moreover, if L is regular then $M(L)$ has strictly fewer states than any other deterministic automaton accepting L; many authors concerned only with finite automata take

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

this property as the definition of the minimal automaton, although even in the finite case the morphism viewpoint is arguably more helpful for understanding the theory. In the infinite case, of course, the cardinality of the state set is wholly insufficient to characterize the minimal automaton.

Let L be a language over an alphabet X. We say that two words $u, v \in X^*$ are *syntactically equivalent* (with respect to L), and write $u \equiv_L v$, if for every pair of words $x, y \in X^*$, we have $xuy \in L$ if and only if $xvy \in L$. Thus, two words are syntactically equivalent if one can substitute one for the other in any word, without affecting membership of the language L. We write $[u]_L$ for the syntactic equivalence class of a word u. It is readily verified that the relation of syntactic equivalence is a congruence on the free monoid X^*, so that the syntactic equivalence classes form a monoid with multiplication well-defined by $[u]_L[v]_L = [uv]_L$. This is called the *syntactic monoid* of L and denoted $S(L)$. The following proposition relates syntactic monoids to minimal automata.

PROPOSITION 2·2. *Let L be a language. Then the syntactic monoid $S(L)$ is isomorphic to the transition monoid of the minimal automaton $M(L)$ via the map $[u]_L \mapsto \sigma_u$.*

We shall also need the notion of a *rational transduction*. We provide here a brief definition; for a detailed introduction, see [**3**]. Let X and Y be finite alphabets. A *finite transducer* from X to Y is a finite directed graph with edges labelled by elements of the direct product $X^* \times Y^*$, and with a distinguished initial vertex and a set of distinguished terminal vertices. Just as with automata, the labelling of edges extends, via the multiplication in the direct product monoid $X^* \times Y^*$, to a labelling of paths. The *relation accepted* by the automaton is the set of all pairs in $X^* \times Y^*$ which label paths from the initial vertex to a terminal vertex. A relation between free monoids X and Y recognised by a finite transducer is called a *rational transduction* from X to Y.

If $\rho \subseteq X^* \times Y^*$ is a relation and $L \subseteq X^*$ is a language, then the *image* of L under ρ is the language

$$L\rho = \{v \mid (u, v) \in \rho \text{ for some } u \in L\} \subseteq Y^*.$$

If ρ is a rational transduction then we say that $L\rho$ is *a rational transduction of* L (via ρ). Two languages are called *rationally equivalent* if each is a rational transduction of the other; rational equivalence is, as the name implies, an equivalence relation on the class of all languages. The *inverse* of ρ is the relation

$$\rho^{-1} = \{(u, v) \mid (v, u) \in \rho\} \subseteq Y^* \times X^*.$$

It is easily seen that if ρ is a rational transduction then so is ρ^{-1}.

We shall also need another operation on languages. If $L \subseteq X^*$ is a language then the *Kleene closure* of L is the submonoid of X^* generated by L, that is, the set of all words of the form $w_1 \cdots w_n$ for some integer $n \geqslant 0$ and words $w_1, \ldots, w_n \in L$. The Kleene closure of L is usually denoted L^*; note that if $L = X$ is in fact the alphabet viewed as a subset of the free monoid it generates, then the Kleene closure of L is the entire free monoid on X, so the notation X^* is unambiguous.

3. *The loop problem*

In this section, we introduce the languages which are the main object of study in this paper. We assume familiarity with the basic terminology and notation of semigroup theory

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

[**15**] and combinatorial group theory [**19**, **20**]. We define the loop problem first for monoids, and then more generally for semigroups.

Let M be a monoid. By a *choice of (monoid) generators* for M we mean a surjective morphism $\sigma\colon X^* \to M$ from a free monoid onto M. The choice of generators is called *finite* if X is finite. A monoid which admits a finite choice of generators is called *finitely generated*. Since our aim is to connect semigroup theory with the theory of formal languages (over finite alphabets) we are primarily interested in finitely generated monoids, although many of our results will also hold for infinite choices of generators.

Recall that the (*right*) *Cayley graph* $\Gamma_\sigma(M)$ of M with respect to σ is a directed graph, possibly with multiple edges and loops, with edges labelled by elements of X. Its vertices are the elements of M, and it has an edge from $a \in M$ to $b \in M$ labelled $x \in X$ exactly if $a(x\sigma) = b$ in the monoid M.

Now let $\overline{X} = \{\overline{x} \mid x \in X\}$ be a set of formal inverses for the generators in X, and let $\hat{X} = X \cup \overline{X}$. We extend the map $x \to \overline{x}$ to an involution on $\hat{X}^*$ by defining $\overline{\overline{x}} = x$ for all $x \in X$, and $\overline{x_1 \cdots x_n} = \overline{x_n} \cdots \overline{x_1}$ for all $x_1, \ldots, x_n \in \hat{X}$. (In fact, $\hat{X}^*$ together with the unary operation $x \to \overline{x}$ is the *free monoid with involution* on X [**18**, section 6·1].)

The (*right*) *loop automaton* $\hat{\Gamma}_\sigma(M)$ of M with respect to X is obtained from the Cayley graph $\Gamma_\sigma(M)$ by adding for each edge labelled x an inverse edge, in the opposite direction, labelled $\overline{x}$. Notice that for each path π in the loop automaton labelled $w \in \hat{X}^*$, there is a corresponding path in the opposite direction labelled $\overline{w}$; we denote this path $\overline{\pi}$. We view the loop automaton as a (typically infinite) automaton over $\hat{X}$, with initial state and terminal state the identity of M. The (*right*) *loop problem* of M with respect to σ is the language $L_\sigma(M) \subseteq \hat{X}^*$ of words recognised by the loop automaton $\hat{\Gamma}_\sigma(M)$.

Words in X^* and $\overline{X}^*$ are called *positive* and *negative* words respectively; words in $\hat{X}^*$ which are neither positive nor negative are called *mixed*. Similarly, an edge or path in $\hat{\Gamma}_\sigma(M)$ is called *positive* [*negative*, *mixed*] if it has a positive [respectively negative, mixed] label.

We now introduce the corresponding definitions for more general semigroups, possibly without identity. Suppose S is a semigroup. By a *choice of* (*semigroup*) *generators* for S we mean a surjective morphism $\sigma\colon X^+ \to S$ from a free semigroup onto S; again, σ is called *finite* if X is finite. Let S^1 be the monoid obtained from S by adjoining an identity element, denoted 1, even if there already is one. Then σ extends uniquely to a monoid choice of generators $\sigma^1\colon X^* \to S^1$. We define the *loop automaton* $\hat{\Gamma}_\sigma(S)$ of S with respect to σ to be the loop automaton $\hat{\Gamma}_{\sigma^1}(S^1)$.

Note that if M is a monoid, then it admits choices of generators, and hence Cayley graphs and loop problems, both as a semigroup and as a monoid. We shall see below (Proposition 4·7) that there is a close relationship between the loop problems of M as a monoid and as a semigroup.

For brevity, we introduce a few notational conventions. If X is a semigroup [respectively, monoid] generating *subset* for a semigroup [monoid] S, then the inclusion map $X \to S$ extends naturally to a formal choice of generators $\sigma\colon X^+ \to S$ [$\sigma\colon X^* \to S$]. In this case, we write $\Gamma_X(S)$, $\hat{\Gamma}_X(S)$ and $L_X(S)$ for $\Gamma_\sigma(S)$, $\hat{\Gamma}_\sigma(S)$ and $L_\sigma(S)$ respectively. Moreover, where the choice of generators is clear, we shall omit the subscript entirely, writing simply $\Gamma(S)$, $\hat{\Gamma}(S)$ and $L(S)$. Similarly, if only one monoid or semigroup is under consideration, we simply write Γ_σ, $\hat{\Gamma}_\sigma$ and L_σ, or even just Γ, $\hat{\Gamma}$ and L.

There is of course a dual notion of the *left Cayley graph* of a monoid or semigroup with respect to a given generating set, obtained by considering multiplication by generators on the

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

left. Consequently, one obtains also the *left loop automaton* (which reads input words from right to left) and the *left loop problem*. In this paper, we shall restrict attention to right loop problems. The loss of generality through this restriction is limited by the fact that the left loop problem of M is simply the reverse of the right loop problem of the opposite monoid of M.

In fact the loop problem is one of two closely related languages which can naturally be associated to a finitely generated monoid or semigroup. The other, which we might call the *meeting problem*, is simply the set of all words of the form $u\overline{v}$ such that u and v are words over the generators representing the same element of the monoid or semigroup. The meeting problem is clearly the intersection of the loop problem with the regular language $X^*\overline{X}^*$; in contrast, the loop problem cannot be described so easily in terms of the meeting problem. Of the two, then, the loop problem is the more complex; it is more naturally analogous to the word problem of a group, and better describes the geometry of the Cayley graph of the monoid; the meeting problem is typically easier to work with and more closely related to the word problem in the sense of universal algebra, but is less geometric and does not directly generalise the usual notion of a word problem (as a language) for groups. One might reasonably expect that a language-theoretic restriction on the loop problem will place much stronger conditions on a semigroup than the same restriction applied to the meeting problem.

4. *Basic properties*

In this section we describe some basic properties of the loop problem. The broad objective is to show that finitely generated monoids and semigroups which are "similar" (that is, which enjoy similar structure) give rise to languages which are "similar" (that is, belong to the same classes of languages, from amongst the classes most frequently arising in formal language theory). The technique is to show that when one has two finitely generated monoids or semigroups which are closely related, one can often pass between the corresponding languages using simple language-theoretic operations such as morphisms, inverse morphisms, rational transductions and Kleene closure.

The following lemma, which follows easily from the definitions, describes the relationship between paths in the loop automaton $L(M)$ and multiplication in the monoid M.

LEMMA 4·1 (The Zig Zag Lemma). *Let $\sigma\colon X^* \to M$ be a choice of generators for a monoid M, and let $x, y \in M$. Let $n \geqslant 1$ and suppose $u_0, \ldots, u_{n-1}, v_1, \ldots, v_n \in X^*$. Then the following are equivalent*:

(i) *$\hat{\Gamma}_\sigma(M)$ has a path from x to y labelled $u_0\overline{v_1}u_1\overline{v_2}\ldots u_{n-1}\overline{v_n}$;*
(ii) *there exist $p_0, \ldots, p_n \in M$ such that $x = p_0$, $y = p_n$, and $p_i(u_i\sigma) = p_{i+1}(v_{i+1}\sigma)$ for $0 \leqslant i < n$.*

Proof. Suppose first that (i) holds, and let π be a path from x to y labelled

$$u_0\overline{v_1}u_1\overline{v_2}\cdots u_{n-1}\overline{v_n}.$$

Let $p_0 = x$, and for $1 \leqslant i \leqslant n$, define p_i to be the vertex reached while tracing the path π from x, after reading the prefix $u_0\overline{v_1}\ldots u_{i-1}\overline{v_i}$. That these vertices have the required property follows easily from the definitions.

Conversely, if (ii) holds then, since the loop automaton contains the Cayley graph, there is a path π_i from each p_i to $p_i(u_i\sigma)$ labelled u_i and a path ϕ_{i+1} from each p_{i+1} to

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

$p_{i+1}(v_{i+1}\sigma) = p_i(u_i\sigma)$ labelled v_{i+1}. In the loop automaton, each path ϕ_{i+1} has an inverse path $\overline{\phi_{i+1}}$ which runs from $p_i(u_i\sigma) = p_{i+1}(v_{i+1}\sigma)$ to p_{i+1} and has label $\overline{v_i}$. Now the concatenation

$$\pi_0\overline{\phi_1}\pi_1 \ldots \pi_{n-1}\overline{\phi_n}$$

is clearly a path with the properties required to show that (i) holds.

Since every word in $\hat{X}^*$ can be factored in the form $u_0\overline{v_1}u_1\overline{v_2}\cdots u_{n-1}\overline{v_n}$ for some words $u_0, \ldots, u_{n-1}, v_1, \ldots, v_n \in X^*$, the Zig Zag Lemma gives a complete algebraic description of all the path labels between any two vertices. The special case of the Lemma in which $n = 1$ and $x = y = 1$ shows that the loop problem of a monoid encodes the word problem in the sense of universal algebra [**11**, section 30]. Substituting S^1 for M in the statement gives a corresponding result for semigroups.

COROLLARY 4·2. *Let $\sigma\colon X^+ \to S$ [respectively, $\sigma\colon X^* \to M$] be a choice of generators for a semigroup [monoid] M, and let $u, v \in X^+$ [$u, v \in X^*$]. Then $u\sigma = v\sigma$ if and only if $u\overline{v} \in L_\sigma$.*

From this, we can deduce that, unlike the sets of words representing the identity, the loop problem is sufficient to determine up to isomorphism the monoid or semigroup, together with its generating system.

COROLLARY 4·3. *Let X be a finite alphabet, and M_1 and M_2 be monoids with choices of generators $\sigma_1\colon X^* \to M_1$ and $\sigma_2\colon X^* \to M_2$. If $L_{\sigma_1}(M_1) = L_{\sigma_2}(M_2)$ then there is a monoid isomorphism $\rho\colon M_1 \to M_2$ such that $\sigma_1\rho = \sigma_2$. The corresponding statement for semigroups also holds.*

Proof. Suppose $L_{\sigma_1}(M_1) = L_{\sigma_2}(M_2)$. Define $\rho\colon M_1 \to M_2$ by $(w\sigma_1)\rho = w\sigma_2$. This map is well-defined since if $u\sigma_1 = v\sigma_1$ then by Corollary 4·2 we have $u\overline{v} \in L_{\sigma_1}(M_1)$; but then $u\overline{v} \in L_{\sigma_2}(M_2)$ so by Corollary 4·2 again we have $u\sigma_2 = v\sigma_2$. An entirely similar argument establishes that ρ is injective.

To see that ρ is a monoid homomorphism, we observe that

$$[(u\sigma_1)(v\sigma_1)]\rho = ((uv)\sigma_1)\rho = (uv)\sigma_2 = (u\sigma_2)(v\sigma_2) = [(u\sigma_1)\rho][(v\sigma_1)\rho].$$

Surjectivity of ρ is immediate from surjectivity of σ_2. Finally, that $\sigma_1\rho = \sigma_2$ follows straight from the definition, completing the result for monoids.

The statement for semigroups follows directly from the definition and the result for monoids, with S^1 substituted for M.

LEMMA 4·4. *Let $\sigma\colon X^* \to M$ be a choice of generators for a monoid M, and suppose $w \in L_\sigma(M)$. Then w labels a loop at every vertex in $\hat{\Gamma}_\sigma(M)$. The corresponding statement for semigroups also holds.*

Proof. Let $x \in M$. Write $w = u_0\overline{v_1}u_1\overline{v_2}\cdots u_{n-1}\overline{v_n}$ with each $u_i, v_i \in X^*$. That $w \in L_\sigma(M)$ means exactly that $w \in \hat{X}^*$ labels a loop at 1 in $\hat{\Gamma}_\sigma(M)$. By the Zig Zag Lemma (Lemma 4·1), there exist $p_0, \ldots, p_n \in M$ such that $p_0 = p_n = 1$ and $p_i(u_i\sigma) = p_{i+1}(v_{i+1}\sigma)$ for $0 \leqslant i < n$. For each i, let $q_i = xp_i$. Then we have $p_0 = p_1 = x$ and for each i, $q_i(u_i\sigma) = xp_i(u_i\sigma) = xp_{i+1}(v_{i+1}\sigma) = q_{i+1}(v_{i+1}\sigma)$. Now applying the Zig Zag Lemma again, we obtain the required loop at x labelled w.

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

The corresponding result for semigroups is once again obtained by substituting S^1 for M, and applying the definition of the loop automaton of a semigroup.

We apply Lemma 4·4 to prove a simple language-theoretic property of the loop problem. Recall that a language L is *insertion-closed* if whenever x, y, w are (possibly empty) words such that $xy \in L$ and $w \in L$ we have $xwy \in L$.

PROPOSITION 4·5. *Every loop problem is insertion-closed.*

Proof. By definition, every loop problem of a semigroup S is also a loop problem of the monoid S^1 with respect to a choice of monoid generators, so it suffices to consider the loop problem $L_\sigma(M)$ of a monoid M with respect to some choice of monoid generators $\sigma\colon X^* \to M$. Suppose $xy \in L_\sigma(M)$ and $w \in L_\sigma(M)$. Then $\hat{\Gamma}_\sigma(M)$ has a vertex v such that there is a path from 1 to v labelled x and a path from v to 1 labelled y. Now by Lemma 4·4, there is a loop at v labelled w. Hence, there is a path from 1 to 1 labelled xwy so $xwy \in L_\sigma(M)$. Thus, $L_\sigma(M)$ is insertion-closed, as required.

It is well known that the word problems of a group with respect to different finite choices of generators can be obtained from each other as inverse images under morphisms of free monoids. Many important classes of languages are closed under inverse morphism, and it follows that these classes correspond to invariant properties of groups which do not depend upon the finite choice of generators. It transpires that the loop problems of monoids and semigroups enjoy a similar property.

PROPOSITION 4·6. *Let M be a monoid and $\sigma\colon X^* \to M$ and $\tau\colon Y^* \to M$ be finite choices of generators. Then $L_\sigma(M)$ is an inverse morphic image of $L_\tau(M)$. The corresponding statement for semigroups also holds.*

Proof. For each generator $y \in Y$, let w_y be a word in X^* such that $w_y\sigma = y\tau$. Define a morphism $\rho\colon \hat{Y}^* \to \hat{X}^*$ by $y\rho = w_y$ and $\overline{y}\rho = \overline{w_y}$ for all $y \in Y$. Note that $u\rho\sigma = u\tau$ for every $u \in Y^*$, and $\overline{v}\rho = \overline{v\rho}$ for every $v \in \hat{Y}^*$. We claim that $L_\tau(M) = L_\sigma(M)\rho^{-1}$.

Suppose $w \in \hat{Y}^*$. Write

$$w = u_0\overline{v_1}u_1\overline{v_2}\cdots u_{n-1}\overline{v_n}$$

where $u_0, \ldots u_{n-1}, v_0, \ldots, v_n \in Y^*$. Then $w \in L_\tau(M)$ if and only if the loop automaton $\hat{\Gamma}_\tau(M)$ has a path from 1 to 1 labelled w. By the Zig Zag Lemma (Lemma 4·1), this is true exactly if there exist elements $p_0, \ldots, p_n \in M$ such that $p_0 = p_n = 1$ and $p_i(u_i\tau) = p_{i+1}(v_{i+1}\tau)$ for $0 \leqslant i < n$. But this holds exactly if there exist elements $p_0, \ldots, p_n \in M$ such that $p_0 = p_n = 1$ and $p_i(u_i\rho\sigma) = p_{i+1}(v_{i+1}\rho\sigma)$ for $0 \leqslant i < n$. Now by the Zig Zag Lemma again, this is true if and only if

$$\begin{aligned} w\rho &= (u_0\rho)(\overline{v_1}\rho)(u_1\rho)(\overline{v_2}\rho)\cdots(u_{n-1}\rho)(\overline{v_n}\rho) \\ &= (u_0\rho)(\overline{v_1\rho})(u_1\rho)(\overline{v_2\rho})\cdots(u_{n-1}\rho)(\overline{v_n\rho}) \end{aligned}$$

labels a loop at 1 in $\hat{\Gamma}_\sigma(M)$, that is, lies in $L_\sigma(M)$.

For the semigroup case, if $\sigma\colon X^+ \to S$ and $\tau\colon Y^+ \to S$ are semigroup choices of generators, then $\sigma^1\colon X^* \to S^1$ and $\tau^1\colon Y^* \to S^1$ are monoid choices of generators for the monoid S^1,

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

so arguing as above, we obtain a morphism $\rho\colon Y^* \to X^*$ such that

$$L_\tau(S) = L_{\tau^1}(S^1) = L_{\sigma^1}(S^1)\rho^{-1} = L_\sigma(S)$$

as required.

Proposition 4·6 tells us that, for example, the classes of regular and context-free languages can be expected to correspond to invariant properties of finitely generated monoids. It is not, however, immediately obvious that these properties will be natural ones from a semigroup-theoretic perspective. In Section 6 below, we shall attempt to isolate the properties in question.

Suppose M is a monoid. We remarked in Section 4 that M admits loop problems both as a semigroup and a monoid. Proposition 4·6 ensures that the different loop problems for M as a monoid are closely related, as are those for M as a semigroup. It does not, however, guarantee a relationship between the loop problems with respect to monoid generating sets, and those with respect to semigroup generating sets. The following proposition shows that there is nonetheless a close relationship.

PROPOSITION 4·7. *Let M be a monoid, let $\sigma\colon X^+ \to M$ be a semigroup choice of generators for M, and let $\tau\colon X^* \to M$ be the unique extension of σ to a monoid choice of generators for M. Then*

$$L_\sigma(M) = L_\tau(M) \cap (X\hat{X}^*\overline{X} \cup \{\epsilon\}),$$

and there exists a word $w \in X^+$ such that

$$L_\tau(M) = w^{-1}L_\sigma(M)\overline{w}^{-1} = \{u \in \hat{X}^* \mid wu\overline{w} \in L_\sigma(M)\}.$$

Proof. Since by definition $L_\sigma(M) = L_{\sigma^1}(M^1)$, we need to consider both the identity of M and the extra identity adjoined in M^1; we write e for the identity of M, and 1 for the identity of M^1. Observe that $\hat{\Gamma}_\sigma(M) = \hat{\Gamma}_{\sigma^1}(M^1)$ is isomorphic to $\hat{\Gamma}_\tau(M)$ with an extra vertex 1 adjoined. The edges incident with 1 are as follows; for each generator $x \in X$, there is an edge from 1 to $x\sigma$ labelled x, and of course a corresponding inverse edge from $x\sigma$ to 1 labelled $\overline{x}$. In particular, whenever there is an edge from 1 to a vertex v [respectively, v to 1], there is also an edge in both $\hat{\Gamma}_\sigma(M)$ and $\hat{\Gamma}_\tau(M)$ from e to v [v to e] with the same label.

We prove first that

$$L_\sigma(M) = L_\tau(M) \cap (X\hat{X}^*\overline{X} \cup \{\epsilon\}).$$

It is immediate from our description above of the edges incident with 1, that every non-empty label of a loop at 1 in $\hat{\Gamma}_\sigma(M)$ must begin with a positive generator and end with a negative generator, so that $L_\sigma(M) \subseteq X\hat{X}^*\overline{X} \cup \{\epsilon\}$. Moreover, from our observation that every edge incident with 1 corresponds to an edge incident with e in $\hat{\Gamma}_\tau(M)$, it follows easily that every word accepted by $\hat{\Gamma}_\sigma(M)$ is accepted by $\hat{\Gamma}_\tau(M)$, so that $L_\sigma(M) \subseteq L_\tau(M)$.

Conversely, suppose $u \in L_\tau(M) \cap (X\hat{X}^*\overline{X} \cup \{\epsilon\})$. If $u = \epsilon$ then certainly $u \in L_\sigma(M)$, so we may assume that $u \in X\hat{X}^*\overline{X}$, that is, $u = xv\overline{y}$ for some $x, y \in X$ and $v \in \hat{X}^*$. Let π be a loop at e labelled u in the loop automaton $\hat{\Gamma}_\tau(M)$. Then π begins with an edge leaving e with label x; from the definition of the loop automaton, it follows easily that the target of this edge is the vertex $x\sigma$. Similarly, π ends with an edge from $y\sigma$ to e labelled $\overline{y}$. In between is a path from $x\sigma$ to $y\sigma$ labelled v. But $\hat{\Gamma}_\sigma(M)$ has an edge from 1 to $x\sigma$ labelled x, and an edge from $y\sigma$ to 1 labelled $\overline{y}$. Moreover, since $\hat{\Gamma}_\sigma(M)$ contains $\hat{\Gamma}_\tau(M)$, it also has a path

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

from $x\sigma$ to $y\sigma$ labelled v. Thus, we obtain a path in $\hat{\Gamma}_\sigma(M)$ from 1 to 1 with label u, so that $u \in L_\sigma(M)$. This completes the proof of the first claim.

Now let $w \in X^+$ be any non-empty word such that $w\sigma = e$. We claim that

$$L_\tau(M) = w^{-1}L_\sigma(M)\overline{w}^{-1} = \{u \in \hat{X}^* \mid wu\overline{w} \in L_\sigma(M)\}.$$

To show one inclusion, suppose $u \in L_\tau(M)$, so that $u \in \hat{X}^*$ labels a loop at e in $\hat{\Gamma}_\tau(M)$. Clearly $\hat{\Gamma}_\sigma(M)$ has a path from 1 to e labelled w and hence a path from e to 1 labelled $\overline{w}$. Since $\hat{\Gamma}_\sigma(M)$ contains $\hat{\Gamma}_\tau(M)$, it also has a loop at e labelled u. Thus, $\hat{\Gamma}_\sigma(M)$ has a loop at 1 labelled $wu\overline{w}$, so that $u \in w^{-1}L_\sigma(M)\overline{w}^{-1}$.

Conversely, suppose that $u \in w^{-1}L_\sigma(M)\overline{w}^{-1}$, that is, that $wu\overline{w} \in L_\sigma(M)$, so that $\hat{\Gamma}_\sigma(M)$ has a loop at 1 labelled $wu\overline{w}$. It follows easily from the definition of the loop automaton that there is a unique path beginning at 1 with label w, and that this path ends at e. Dually, there is a unique path ending at 1 with label $\overline{w}$, and this path begins at e. Hence, $\hat{\Gamma}_\sigma(M)$ must have a loop at e labelled u. Moreover, because of our observation at the start of the proof that every edge incident with 1 has a corresponding edge incident with e, we can find such a loop which does not visit the vertex 1. It follows that $\hat{\Gamma}_\tau(M)$ also has a loop at e labelled u, so that $u \in L_\tau(M)$, as required to complete the proof of the second claim.

Since taking an intersection with a regular language, left and right division, and taking an inverse image under a morphism are all operations which can be performed by rational transductions [**3**], Propositions 4·6 and 4·7 yield the following result, which says that the loop problem of a semigroup or monoid is, modulo rational equivalence, invariant under choice of generators.

COROLLARY 4·8. *Let S be a semigroup or monoid. Then any two loop problems for S (as a semigroup or as a monoid if appropriate) are rationally equivalent.*

In view of Corollary 4·8, we shall permit ourselves to speak simply of "the" loop problem of a monoid or semigroup where, as is often the case, our interest is in language-theoretic properties which are invariant under rational transductions.

5. *Groups and completely simple semigroups*

In this section, we consider the loop problems of groups, and then more generally of completely simple semigroups.

Let $\sigma\colon X^* \to G$ be a choice of monoid generators for a group G. Recall that the *word problem* for G with respect to σ is the language of all words $w \in X^*$ such that $w\sigma = 1$ in G. Equivalently, it is the language accepted by the Cayley graph $\Gamma_\sigma(G)$ when viewed as an automaton with initial and terminal state the identity of G. An obvious question concerns the relationship between the loop problem for a group (with respect to a particular finite choice of generators) and the word problem. The following proposition says that they are almost the same thing.

PROPOSITION 5·1. *Let $\sigma\colon X^* \to G$ be a choice of monoid generators for a group G. Then there exists a choice of monoid generators $\tau\colon \hat{X}^* \to G$ such that the loop problem $L_\sigma(G)$ of G with respect to σ is equal to the word problem of G with respect to τ.*

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

Proof. Define $\tau\colon \hat{X}^* \to G$ by $x\tau = x\sigma$ and $\overline{x}\tau = (x\sigma)^{-1}$ for all $x \in X$. It is readily verified that $\hat{\Gamma}_\sigma(G)$ is exactly the same as $\Gamma_\tau(G)$, so that $L_\sigma(G)$ is equal to the word problem of G with respect to τ, as required.

Since we are chiefly interested in intrinsic structural properties of groups (and semigroup and monoids) which are not dependent upon a particular choice of generators, this correspondence is quite sufficient for our purposes.

Next, we consider the broader class of *completely simple* semigroups. Recall that a *primitive idempotent* in a semigroup S is an idempotent e with the property that for any non-zero idempotent f such that $ef = fe = f$, we have $e = f$. A semigroup is called *completely simple* if it has a primitive idempotent and no proper ideals. For a detailed introduction to the theory of completely simple semigroups, including a number of equivalent definitions, see [**6**, chapter 2] or [**15**, chapter 3]. A *subgroup* of a semigroup S is a subsemigroup of S which forms a group with the multiplication inherited from S; a subgroup of S is *maximal* if it is not properly contained in any other subgroup. A construction of Rees [**23**] gives a simple combinatorial description of completely simple semigroups, in terms of their maximal subgroups. Let G be a group and I and J be sets, and let P be a $J \times I$ matrix with entries drawn from G. The *Rees matrix semigroup* $M(G; I, J; P)$ is the semigroup with set of elements $I \times G \times J$ and multiplication given by $(i, g, j)(k, h, l) = (i, gP_{jk}h, l)$ for all $i, k \in I$, $g, h \in G$ and $j, l \in J$. The following theorem is usually attributed to Rees, although the essential idea was given by Suschkewitz [**26**].

THEOREM 5·2. (Suschkewitz [**26**], Rees [**23**]). *Let G be a group, I and J sets and P a $J \times I$ matrix over G. Then the Rees matrix semigroup $M(G; I, J; P)$ is completely simple with maximal subgroups all isomorphic to G. Conversely, every completely simple semigroup is isomorphic to one of this form.*

Both completely simple semigroups and Rees matrix constructions are of central importance in the structural theory of semigroups. We shall need the following elementary property of completely simple semigroups.

LEMMA 5·3. *Let H be a maximal subgroup of a completely simple semigroup S. If $a \in S^1$ and $x, y \in H$ are such that $ax \in H$, then there exists $b \in H$ such that $bx = ax$ and $by = ay$.*

Proof. To prove the claim when $a = 1$ we simply take b to be the identity of H. Otherwise, by the Rees theorem (Theorem 5·2), we may assume that $S = M(G; I, J; P)$ for some group G, sets I and J and $J \times I$ sandwich matrix P. It is well known, and can easily be deduced from the Rees theorem, that there exist $i \in I$ and $j \in J$ such that the maximal subgroup H is the set of all elements of the form (i, g, j) for different $g \in G$.

Thus, we may suppose that $a = (i_a, g_a, j_a)$, $x = (i, g_x, j)$ and $y = (i, g_y, j)$. Set $b = (i, g_a P_{j_a i} P_{ji}^{-1}, j) \in H$. Now we have $b \in H$ and

$$ax = (i_a, g_a, j_a)(i, g_x, j) = (i_a, g_a P_{j_a i} g_x, j).$$

On the other hand,

$$bx = \big(i, g_a P_{j_a i} P_{ji}^{-1}, j\big)(i, g_x, j) = \big(i, g_a P_{j_a i} P_{ji}^{-1} P_{ji} g_x, j\big) = \big(i, g_a P_{j_a i} g_x, j\big).$$

Moreover, since we know that $ax \in H$, we must have $i_a = i$, from which it follows that $ax = bx$ as required. An similar argument (but using the fact that we already know that

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

$i_a = i$ in order to avoid the need to presuppose that $ay \in H$) shows that $ay = by$, completing the proof.

The Rees theorem often allows results about groups to be extended in some form to completely simple semigroups. The following two theorems say that the loop problem of a finitely generated completely simple semigroup is closely related to the loop problem (or equivalently the word problem) of its maximal subgroups.

THEOREM 5·4. *Let G be a maximal subgroup of a completely simple semigroup S. Let $\sigma: X^+ \to G$ be a choice of semigroup generators for G, and $\tau: Y^+ \to S$ a choice of generators for S, such that $X \subseteq Y$ and σ is the restriction of τ to X^+. Then $L_\sigma(G) = L_\tau(S) \cap \hat{X}^*$.*

Proof. The loop automaton $\hat{\Gamma}_\sigma(G)$ of G is clearly embedded in a natural way into the loop automaton $\hat{\Gamma}_\tau(S)$. It follows easily that any word accepted by the former is also accepted by the latter, so that $L_\sigma(G) \subseteq L_\tau(S) \cap \hat{X}^*$.

Conversely, suppose w is a word in $L_\tau(S) \cap \hat{X}^*$. Then $\hat{\Gamma}_\tau(S) = \hat{\Gamma}_{\tau^1}(S^1)$ has a loop at 1 labelled w. Write $w = u_0\overline{v_1}u_1\overline{v_2}\cdots u_{n-1}\overline{v_n}$ with each $u_i, v_i \in X^*$. By the Zig Zag Lemma (Lemma 4·1), there exist $p_0, \ldots, p_n \in S^1$ such that $p_0 = p_n = 1$ and $p_i(u_i\tau) = p_{i+1}(v_{i+1}\tau)$ for $0 \leqslant i < n$. We claim that the elements $p_0, \ldots, p_n$ can all be chosen to lie in G^1. Indeed, suppose we are given $p_0, \ldots, p_n$ satisfying the above equations and not all lying in G^1, and let k be minimal such that p_k does not lie in G^1. Certainly, $1 \leqslant k \leqslant n-1$. Now we have $u_{k-1}\tau \in G$ and $p_{k-1} \in G^1$, so that $p_{k-1}(u_{k-1}\tau) = p_k(v_k\tau)$ lies in G. Since $v_k\tau$ and $u_k\tau$ also lie in G, it follows by Lemma 5·3 that there exists an element $q_k \in G$ with $q_k(v_k\tau) = p_k(v_k\tau) = p_{k-1}(u_{k-1}\tau)$ and $q_k(u_k\tau) = p_k(u_k\tau) = p_{k+1}(v_{k+1}\tau)$. Hence, we can replace p_k with q_k to obtain a sequence $p_0, \ldots, p_n$ with strictly fewer elements outside G^1. Continuing in the same vein, we eventually obtain $p_0, \ldots, p_n \in G^1$ with the desired properties.

Applying the Zig Zag Lemma again, we now see that $\hat{\Gamma}_\sigma(G) = \hat{\Gamma}_{\sigma^1}(G^1)$ has a loop at 1 labelled w, so that $w \in L_\tau(G)$, as required.

THEOREM 5·5. *Let S be a finitely generated completely simple semigroup with maximal subgroups isomorphic to a group G. Then the loop problem for S is the Kleene closure of a rational transduction of the word problem for G.*

Proof. By the Rees theorem (Theorem 5·2), we may assume that $S = M(G; I, J; P)$ where by the main theorem of [**2**], I and J are finite. Let $\sigma: X^* \to G$ be a finite choice of monoid generators for G, and $\tau: Y^+ \to S$ a finite choice of semigroup generators for S. For each $y \in Y$, suppose $y\tau = (i_y, g_y, j_y)$ and let $w_y, w'_y \in X^*$ be words representing $g_y, g_y^{-1} \in G$ respectively. For each $i \in I$ and $j \in J$, let $w_{ji}, w'_{ji} \in X^*$ be words representing $P_{ji}, P_{ji}^{-1} \in G$ respectively.

We define a finite transducer from X to Y with:

(i) vertex set $(I \times J) \cup \{A, Z\}$ where A and Z are new symbols;
(ii) initial state A;
(iii) terminal state Z;
(vi) for each generator $y \in Y$, an edge from A to (i_y, j_y) labelled (w_y, y);
(v) for each generator $y \in Y$, an edge from (i_y, j_y) to Z labelled $(w'_y, \overline{y})$;
(vi) for each generator $y \in Y$, each $k \in J$ and each $i \in I$, an edge from (i, k) to (i, j_y) labelled $(w_{ki_y}w_y, y)$; and

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

(vii) for each generator $y \in Y$, each $j \in J$ and each $i \in I$, an edge from (i, j_y) to (i, j) labelled $(w'_y w'_{ji_y}, \overline{y})$.

Now let $g \in G$, $i \in I$, $j \in J$ and $v \in \hat{Y}^*$ and suppose n is a positive integer. We say that a path in the loop automaton $\hat{\Gamma}_\tau(S)$ which originates at 1 is *non-returning* if it does not visit the vertex 1 at any point other than the start and (possibly) the end. As a first step towards the proof, we claim that the following conditions are equivalent:

(i) the loop automaton $\hat{\Gamma}_\tau(S)$ has a non-returning path of length n from 1 to (i, g, j) labelled v;

(ii) the transducer has a path of length n from A to (i, j) labelled (u, v) for some $u \in X^*$ which represents g.

The proof of equivalence proceeds by induction on the path length n. That the equivalence holds when $n = 1$ follows immediately from the definition of the edges in the transducer. Now let $n > 1$ and suppose true for smaller n.

Suppose first that (i) holds, and let π be the path given by the hypothesis. Let e be the last edge the path π, and let π' be the path π with the last edge removed, so that $\pi = \pi' e$. Let v' be the label of π'. Since $n > 1$ and π is non-returning, the path π' must end at a vertex of the form (i', g', k). It follows easily from the definition of the multiplication in a Rees matrix semigroup that the vertices in the loop automaton corresponding to elements with first coordinate i are connected to the rest of the automaton only via the vertex 1. Hence, since the path π is non-returning, we must have $i = i'$, so that π actually ends at (i, g', k). Now π is a path of length $n - 1$, so by the inductive hypothesis, the transducer has a path of length $n - 1$ from A to (i, k) labelled (u', v') for some word $u' \in X^*$ which represents $g' \in G$.

We now treat separately the case where e is a positive edge, and that where e is a negative edge. Suppose first that e is a positive edge, with label $y \in Y$. Then clearly $j = j_y$ and from the definition, the transducer has an edge from (i, k) to (i, j) with label $(w_{ki_y} w_y, y)$. Hence, the transducer has a path of length n from 1 to (i, j) with label

$$(u', v')\big(w_{ki_y} w_y, y\big) = \big(u' w_{ki_y} w_y, v' y\big) = \big(u' w_{ki_y} w_y, v\big).$$

Now from the definition of the loop automaton, we must have

$$(i, g, j) = (i, g', k)(y\tau) = (i, g', k)(i_y, g_y, j_y) = \big(i, g' P_{ki_y} g_y, j_y\big).$$

Equating second coordinates, we see that $g = g' P_{ki_y} g_y$. But it follows that g is represented by the word $u' w_{ki_y} w_y \in X^*$, so setting $u = u' w_{ki_y} w_y \in X^*$, we see that (ii) holds as required.

On the other hand, suppose e is a negative edge, with label $\overline{y}$ for some $y \in Y$. In this case $k = j_y$ and the transducer has an edge from (i, k) to (i, j) with label $(w'_y w'_{ji_y}, \overline{y})$. Hence, there is a path of length n from 1 to (i, j) with label

$$(u', v')\big(w'_y w'_{ji_y}, \overline{y}\big) = \big(u' w'_y w'_{ji_y}, v' \overline{y}\big) = \big(u' w'_y w'_{ji_y}, v\big).$$

Now from the definition of the loop automaton, we must have

$$(i, g', k) = (i, g, j)(y\tau) = (i, g, j)(i_y, g_y, j_y) = \big(i, g P_{ji_y} g_y, j_y\big).$$

Again equating second coordinates, we see this time that $g P_{ji_y} g_y = g'$, so that $g = g' g_y^{-1} P_{ji_y}^{-1}$, and g is represented by the word $u' w'_y w'_{ji_y}$. Now setting $u = u' w'_y w'_{ji_y}$, we again see that (ii) holds as required. This completes the proof that (i) implies (ii).

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

Conversely, suppose (ii) holds, and this time let π be a path of length n in the transducer from A to (i, j) labelled (u, v) for some $u \in X^*$ which represents g. Much as before, we let e be the last edge of π and π' be the path π with the final edge removed. Then π' is a path of length $n-1$ from A to some vertex (i', k) with label of the form (u', v'). Moreover, it follows easily from the definition of the transducer that $i = i'$, so that π' ends at (i, k). Let $g' \in G$ be the element represented by u'. Then by the inductive hypothesis, there exists a path of length $n-1$ in the loop automaton from 1 to (i, g', k) with label v'. Now e is an edge from (i, k) to (i, j). From the definition of the edges in the transducer, we see that there exists $y \in Y$ such that either $j_y = j$ and e has label $(w_{ki_y} w_y, y)$, or $j_y = k$ and e has label $(w'_y w'_{ji_y}, \overline{y})$. As before, we treat these two cases separately. In the former case, observe that we have $u = u' w_{ki_y} w_y$ from which we deduce that $g = g' P_{ki_y} g_y$. But now

$$(i, g', k)(y\tau) = \left(i, g' P_{ki_y} g_y, j_y\right) = (i, g, j)$$

so we see that the loop automaton has an edge from (i, g', k) to (i, g, j) labelled y. Combining this with the path whose existence we deduced using the inductive hypothesis, we conclude that the loop automaton has a path from 1 to (i, g, j) labelled $v = v'y$, so that (i) holds as required. An entirely similar argument suffices to show that (i) also holds in the case that e has label of the form $(w'_y w'_{ji_y}, \overline{y})$, thus completing the proof of the equivalence of conditions (i) and (ii) above.

Now let K be the language of all words which label non-returning loops at the identity in $\hat{\Gamma}_\tau(S)$. We claim now that K is exactly the image of the word problem of G under the transduction defined by our transducer. Clearly the Kleene closure K^* is exactly the loop problem $L_\tau(S)$, so this will suffice to complete the proof of the theorem.

Suppose first that $v \in K$. Then by definition the loop automaton $\hat{\Gamma}_\tau(S)$ has a non-returning loop π at 1 labelled v. Note that all edges in $\hat{\Gamma}_\tau(S)$ which end at 1 run from vertices corresponding to a generator $y \in Y$ and have label $\overline{y}$, so we may assume that the last edge of the path runs from $y\tau$ to 1, and has label $\overline{y}$. Let π' be the path π without this last edge, so that π' runs from 1 to $y\tau$, and let $v' \in \hat{Y}^*$ be the label of π', so that $v = v'\overline{y}$. Then by the equivalence above, the transducer has a path from A to (i_y, j_y) labelled (u, v') for some $u \in X^*$ representing g_y. But directly from the definition, the transducer also has an edge from (i_y, j_y) to Z labelled $(w'_y, \overline{y})$. Hence, we deduce that (uw'_y, v) is accepted by the transducer, where uw'_y represents $g_y g_y^{-1} = 1$ in G, so that v lies in the image under the transduction of the word problem of G.

Conversely, suppose u is a word representing 1 in G, such that the transducer accepts (u, v). Then the transducer has a path π from A to Z labelled (u, v). Again, we proceed by letting π' be the path obtained from π by deleting the last edge. Then there must exist a generator y such that π' ends at (i_y, j_y). Moreover, π' must be labelled (u', v') where $u = u'w'_y$ and $v = v'\overline{y}$. Now since u represents 1 and w'_y represents g_y^{-1}, we deduce that u' represents g_y. By the equivalence above, it follows that the loop automaton $\hat{\Gamma}_\tau(S)$ has a non-returning path from 1 to (i_y, g_y, j_y) labelled v'. Now it certainly also has an edge from (i_y, g_y, j_y) to 1 labelled $\overline{y}$, so we deduce that $v = v'\overline{y} \in K$, as required. This completes the proof.

We remark that the transducer constructed in the proof of Theorem 5·5 can be construed as a *G-automaton* [**17**] accepting part of the loop problem for S.

Combining Theorems 5·4 and 5·5 with Propositions 4·7 and 5·1, we obtain the following.

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

THEOREM 5·6. *Let $\mathcal{F}$ be a family of languages closed under rational transduction and Kleene closure, and let S be a finitely generated completely simple semigroup. Then the following are equivalent*:

(i) *the loop problem for S belongs to $\mathcal{F}$;*
(ii) *the loop problem for each maximal subgroup of S belongs to $\mathcal{F}$;*
(iii) *the word problem for each maximal subgroup of S belongs to $\mathcal{F}$.*

6. *Structural-linguistic correspondences*

In this section, we exhibit some correspondences between the structural properties of a monoid, and the linguistic properties of its loop problem. As a first step, we obtain a generalisation of a foundational result of Anisimov [**1**] from combinatorial group theory.

THEOREM 6·1. *Let $\sigma\colon X^* \to M$ be a finite choice of generators for a monoid M. Then M is finite if and only if $L_\sigma(M)$ is regular. The corresponding statement for semigroups also holds.*

Proof. If M is finite then it follows immediately from the definitions that the loop automaton $\hat{\Gamma}_\sigma(M)$ is finite. Hence, $L_\sigma(M)$ is recognised by a finite automaton, and so is regular.

For the converse, recall that a *(right) cone* of a language $L \subseteq Y^*$ is a language of the form

$$w^{-1}L = \{x \in Y^* \mid wx \in L\}$$

for some $w \in Y^*$. The Myhill–Nerode Theorem [**14**, theorem 3·9] states that a language is regular if and only if it has finitely many distinct right cones. Now let $u, v \in X^*$ be words in the generators of M. Then by Corollary 4·2, we have $u\overline{v} \in L$ if and only if u and v represent the same element of M. It follows that $\overline{v} \in u^{-1}L(M)$ if and only if u and v represent the same element of M, and hence that $L(M)$ has a distinct cone for each element of M. So if M is infinite then $L(M)$ has infinitely many cones, and so is not regular.

The corresponding statement for semigroups follows from the result for monoids together with the fact that a semigroup S is finite if and only if S^1 is finite.

We now turn our attention to the class of monoids whose loop problem is a *context-free* language (see [**3**]). In the group case, a well-known theorem of Muller and Schupp [**21**], augmented by a subsequent result of Dunwoody [**8**], says that a finitely generated group has context-free word problem if and only if it is *virtually free*, that is, has a free subgroup of finite index. Despite the straightforward combinatorial nature of this statement, the only known proof depends essentially on deep geometric results about group Cayley graphs, and specifically on Stallings' theory of *ends* [**25**]. We pose the following question.

Question 6·2. Can one characterize or even classify the monoids and semigroups with context-free loop problem?

We suspect this question (in general) to be difficult, and believe that a satisfactory answer and the techniques used to obtain it may be a significant development with implications reaching well beyond combinatorial semigroup theory. Firstly, such a result may lead to a purely combinatorial understanding of the Muller-Schupp Theorem, which has been much sought after by combinatorial group theorists. Secondly, it may require the development of new tools for showing that certain languages are *not* context-free; since there are a number of notable languages conjectured, but not proven, to be non-context-free, such techniques are likely to be of considerable interest to formal language theorists

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

We hope that seeking answers in different classes of "well-behaved" semigroups may give an indication of how one could proceed in the general case. In semigroup theory, of course, "well-behaved" typically means one of "combinatorially straightforward", "in some way group-like" or perhaps "combinatorially straightforward modulo group theory". One might expect that the question can be answered in the first case by elementary combinatorial means, and in the last by application of the Muller–Schupp theorem. As one example, recalling that the class of context-free languages is closed under rational transduction [**3**, corollary 4·2] and Kleene closure [**3**, theorem 2·1], we can combine the Muller-Schupp Theorem with Theorem 5·6 to obtain a complete description of completely simple semigroups with context-free loop problem.

THEOREM 6·3. *Let S be a finitely generated completely simple semigroup. Then S has context-free loop problem if and only if its maximal subgroups are virtually free.*

To conclude this section, we discuss briefly the relationship between semigroups with context-free loop problem, and recent attempts to define *word hyperbolic* semigroups. A *choice of representatives* for a semigroup S consists of a choice of generators $\sigma\colon X^+ \to S$ together with a subset $R \subseteq X^+$ such that $R\sigma = S$. The choice of representatives is called *regular* if X is finite and R is a regular language. The *multiplication table* for S with respect to σ and R is the language $\{u\#v\#w^R \mid (uv)\sigma = w\sigma\}$ where # is a new symbol not in X, and w^R denotes the word w written backwards.

An interesting recent result of Gilman [**10**] is that a finitely generated group is word hyperbolic (in the sense of Gromov [**12**]) if and only if it admits a regular choice of representatives with respect to which the multiplication table is a context-free language. Duncan and Gilman [**7**] proposed that the latter condition might form a suitable basis for a theory of *word hyperbolic semigroups*. Virtually free groups – that is, groups with context-free word problem – form an elementary class of word hyperbolic groups. It transpires that a more general relationship holds between semigroups with context-free loop problem and word hyperbolic semigroups in the sense of Duncan and Gilman.

PROPOSITION 6·4. *Let S be a semigroup with context-free loop problem. Let $\sigma\colon X^+ \to S$ be any choice of generators for S. Then the multiplication table of S with respect to σ and X^+ is context-free. In particular, S is word hyperbolic in the sense of Duncan and Gilman* [**7**].

Proof. Consider first the language

$$K_1 = \{u\#v\#w \mid u, v, w \in \hat{X}^*, uvw \in L_\sigma(S)\}.$$

It is an easy exercise to verify that K_1 is a rational transduction of $L_\sigma(S)$, and hence is context-free. Now by intersecting with a regular set we see that the language

$$K_2 = K_1 \cap (X^*\#X^*\#\overline{X}^*)$$

is also context-free. Finally, applying the substitution $\overline{x} \to x$ for all $x \in X$ gives the multiplication table we require, so the latter is also context-free.

7. *Right cancellative monoids*

In this section, we study the loop problem in the case of right cancellative monoids. We begin with a proposition which tells us that, in this case, the loop automaton is even more closely related to the loop problem than in general.

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

PROPOSITION 7·1. *Let $\sigma\colon X^* \to M$ be a choice of generators for a right cancellative monoid M. Then the loop automaton $\hat{\Gamma}_\sigma(M)$ is the minimal automaton for the loop problem $L_\sigma(M)$.*

Proof. By definition, $\hat{\Gamma}_\sigma(M)$ is an automaton accepting the loop problem of M. Clearly there is a path from the identity vertex to every vertex, and a path from each vertex to the identity vertex, so that the automaton is trim.

For determinism, suppose $x \in X$ and $p, q, r \in M$ are such that there are edges from p to q and from p to r both labelled x. Then we have $q = p(x\sigma) = r$. On the other hand, suppose there are edges from p to q and from p to r both labelled $\overline{x}$. Then $q(x\sigma) = p = r(x\sigma)$ so by right cancellativity we deduce again that $q = r$. Thus, the automaton is deterministic.

For minimality, it suffices to show that no two states in the loop automaton are equivalent, that is, have the same cone. To this end, let p and q be vertices in the loop automaton, that is, elements of M, and choose a word $u \in X^*$ such that $u\sigma = p$. Then there is a path from 1 to p labelled u, but no path from 1 to q labelled u. But now there is an inverse path from p to 1 labelled $\overline{u}$, but no path from q to 1 labelled $\overline{u}$. Hence, $\overline{u}$ lies in the cone of p but not in the cone of q. We deduce that all vertices have distinct cones, so that the automaton is minimal.

Combining Proposition 7·1 with Proposition 2·2, we immediately obtain the following description of the syntactic monoid of the loop problem of a right cancellative monoid.

COROLLARY 7·2. *Let $\sigma\colon X^* \to M$ be a choice of generators for a right cancellative monoid M. Then the transition monoid of the loop automaton $\hat{\Gamma}_\sigma(M)$ is the syntactic monoid of the loop problem $L_\sigma(M)$.*

We now describe an interaction between the loop problem and the classical structural theory of right cancellative monoids. Let M be a right cancellative monoid. For each element $m \in M$, define the *right translation map* $\rho_m\colon M \to M$ by $x\rho = xm$ for all $x \in M$. Since the monoid is right cancellative, each map ρ_m is injective. Its *(relational) inverse* is the map $\rho_m^{-1}\colon M\, m \to M$ well-defined by $(xm)\rho = x$ for all $x \in M$. Viewed as partial bijections on M, the maps of the form ρ_m and ρ_m^{-1} generate an inverse monoid, which is called the *inverse hull* of M. Inverse hulls, were introduced by Rees [**24**], who used them to study embeddings of cancellative semigroups into groups. A detailed study can be found in [**6**, sections I·9 and I·10].

Now let $\tau\colon X^* \to M$ be a choice of generators for M. It is readily verified from the definitions that for each $x \in X$, the right translation map $\rho_{x\tau}$ is exactly the same as the transition map σ_x in the loop automaton $\hat{\Gamma}_\tau(M)$. Similarly, its relational inverse $\rho_{x\tau}^{-1}$ is the transition map $\sigma_{\overline{x}}$ in the loop automaton. Thus, we see that the transition monoid of the loop automaton $\hat{\Gamma}_\tau(M)$ is exactly the inverse hull of the right cancellative monoid M. Combining with Corollary 7·2, we obtain the following relationship between the loop problem and the inverse hull of a right cancellative monoid.

THEOREM 7·3. *Let $\tau\colon X^* \to M$ be a choice of generators for a right cancellative monoid M. Then the syntactic monoid of the loop problem $L_\tau(M)$ is the inverse hull of M. (Moreover, its action by partial maps on the minimal automaton of $L_\tau(M)$ coincides with its action by partial bijections on M.)*

Note that the inverse hull of a group is easily seen to be isomorphic to the group itself. Hence, in this case, we recover the "folklore" fact that a group is the syntactic monoid of its own word problem.

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

In the case that M is right cancellative, we can describe another nice language-theoretic property of the loop problem. Recall that a language L is called *deletion-closed* if whenever x, y, w are (possibly empty) words such that $w \in L$ and $xwy \in L$ we have $xy \in L$. Note that a non-empty deletion-closed language will always contain the empty word.

PROPOSITION 7·4. *Any loop problem of a right cancellative monoid is deletion-closed.*

Proof. Let $\sigma\colon X^* \to M$ be a choice of generators for a right cancellative monoid M, and suppose xwy, $w \in L_\sigma(M)$. Then the loop automaton has vertices p and q such that there is a path π_1 from 1 to p labelled x, a path π_2 from p to q labelled w and a path π_3 from q to 1 labelled y. There is also a loop at 1 labelled w, so by Lemma 4·4 there is a loop π_4 at p labelled w. Now the paths π_2 and π_4 both start at p and have label w, and by Proposition 7·1 the loop automaton is deterministic, so π_2 and π_4 must end at the same vertex, that is, $p = q$. But now $\pi_1\pi_3$ is a loop at 1 labelled xy, so $xy \in L_\sigma(M)$ as required.

Combining Propositions 4·5 and 7·4 we obtain the following corollary.

COROLLARY 7·5. *Every inverse hull of a right cancellative monoid is the syntactic monoid of an insertion-closed, deletion-closed language.*

PROPOSITION 7·6. *Let $L \subseteq X^*$ be a non-empty language. Then the following are equivalent*:

(i) *L is insertion-closed and deletion-closed*;
(ii) *L is the language of words representing the identity in its syntactic monoid*;
(iii) *L is the language of words representing the identity in some monoid.*

Proof. Suppose (i) holds. Observe that the map $w \mapsto [w]_L$ is a choice of generators for the syntactic monoid $S(L)$. Now the fact that L is insertion-closed and deletion-closed means precisely that for any $w \in L$ and any x, $y \in X^*$, we have $xwy \in L$ if and only if $xy \in L$. But this means that every $w \in L$ is syntactically equivalent to the empty word, that is, $[w]_L$ is the identity element in $S(L)$. Conversely, if $[w]_L$ is the identity in $S(L)$ then $w \equiv_L \epsilon$. Since L is non-empty and deletion-closed we have $\epsilon \in L$, and so $w \in L$. Hence, (ii) holds.

That (ii) implies (iii) is immediate, so suppose now that (iii) holds, that is, that L is the language of words representing the identity in some monoid M generated by X. Clearly if xy and w both represent the identity in M then xwy also represents the identity, so L is insertion-closed. Conversely, if xwy and w represent the identity then xy represents the identity, so L is deletion-closed. Thus, (i) holds.

Combining Theorem 7·3 with Propositions 7·4 and 7·6 we obtain the following additional description of the loop problem in a right cancellative monoid.

COROLLARY 7·7. *Let M be a right cancellative monoid. Then the loop problem of M is the language of words representing the identity in the inverse hull of M.*

An application of the connections developed in this section to the study of a class of inverse monoids will form part of a forthcoming paper of J. B. Fountain and the author.

Acknowledgements. This research was started while the author was at Carleton University supported by the Leverhulme Trust, continued at Universität Kassel with the support of a Marie Curie Intra-European Fellowship within the 6th European Community Framework Programme, and completed at the University of Manchester supported by an RCUK

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press

Academic Fellowship. The author would like to thank John Fountain for some helpful discussions and Kirsty for all her support and encouragement.

REFERENCES

[**1**] A. V. ANĪSĪMOV. The group languages. *Kibernetika* (*Kiev*), (4), (1971), 18–24.
[**2**] H. AYIK and N. RUŠKUC. Generators and relations of Rees matrix semigroups. *Proc. Edinburgh Math. Soc.* **42** (1999), 481–495.
[**3**] J. BERSTEL. *Transductions and Context-Free Language* (Informatik, Teubner, 1979).
[**4**] W. W. BOONE and G. HIGMAN. An algebraic characterization of groups with soluble word problem. *J. Austral. Math. Soc.* **18** (1974), 41–53. Collection of articles dedicated to the memory of Hanna Neumann, IX.
[**5**] C. M. CAMPBELL, E. F. ROBERTSON, N. RUŠKUC, and R. M. THOMAS. Automatic semigroups. *Theoret. Comput. Sci.* **250** (2001), 365–391.
[**6**] A. H. CLIFFORD and G. B. PRESTON. *The Algebraic Theory of Semigroups* (*Volume I*). (Amer. Math. Soc., 1961).
[**7**] A. DUNCAN and R. H. GILMAN. Word hyperbolic semigroups. *Math. Proc. Camb. Phil. Soc.* **136**(3), (2004), 513–524.
[**8**] M. J. DUNWOODY. The accessibility of finitely presented groups. *Invent. Math.* **81**(3), (1985), 449–457.
[**9**] J. B. FOUNTAIN and M. KAMBITES. Hyperbolic groups and completely simple semigroups. In *Semigroups and Languages* (World Science Publishing, 2004), pp. 106–132.
[**10**] R. H. GILMAN. On the definition of word hyperbolic groups. *Math. Z.* **242**(3), (2002), 529–541.
[**11**] G. GRÄTZER. *Universal Algebra* (D. Van Nostrand Co., Inc., 1968).
[**12**] M. GROMOV. Hyperbolic groups. In *Essays in Group Theory*, volume 8 of *Math. Sci. Res. Inst. Publ.* (Springer, 1987), pp. 75–263.
[**13**] G. HIGMAN. Subgroups of finitely presented groups. *Proc. Roy. Soc. Ser. A* **262** (1961), 455–475.
[**14**] J. E. HOPCROFT and J. D. ULLMAN. *Formal Languages and their Relation to Automata* (Addison-Wesley, 1969).
[**15**] J. M. HOWIE. *Fundamentals of Semigroup Theory* (Clarendon Press, 1995).
[**16**] J. F. P. HUDSON. Regular rewrite systems and automatic structures. In J. Almeida, G. Gomes, and P. V. Silva, editors, *Semigroups, Automata and Languages* (World Scientific, 1996), pp. 145–152.
[**17**] M. KAMBITES. Formal languages and groups as memory. `arXiv:math.RA/0601061`, 2006.
[**18**] M. V. LAWSON. *Inverse Semigroups: The Theory of Partial Symmetries* (World Scientific, 1998).
[**19**] R. C. LYNDON and P. E. SCHUPP. *Combinatorial Group Theory* (Springer-Verlag, 1977).
[**20**] W. MAGNUS, A. KARRASS, and D. SOLITAR. *Combinatorial Group Theory* (Dover Publications Inc., revised edition, 1976).
[**21**] D. E. MULLER and P. E. SCHUPP. Groups, the theory of ends, and context-free languages. *J. Comput. System Sci.* **26**(3), (1983), 295–310.
[**22**] J.-E. PIN. *Varieties of Formal Languages* (North Oxford Academic, 1986).
[**23**] D. REES. On semi-groups. *Proc. Cam. Phi. Soc.* **36** (1940), 387–400.
[**24**] D. REES. On the group of a set of partial transformations. *J. London Math. Soc.* **22** (1948), 281–284.
[**25**] J. R. STALLINGS. On torsion-free groups with infinitely many ends. *Ann. of Math.* (2) **88** (1968), 312–334.
[**26**] A. K. SUSCHKEWITZ. Über die endlichen Gruppen ohne das Gesetz der eindeutigen Umkehrbarkeit. *Math. Ann.* **99** (1928), 30–50.

https://doi.org/10.1017/S0305004107000606 Published online by Cambridge University Press