LANGUAGES ACCEPTED BY AUTOMATA WITH A COUNTER

by

JAMES LANCE ROSS

JON CORSON, COMMITTEE CHAIR
MARTYN DIXON
MARTIN EVANS
ALLEN STERN
BRUCE TRACE

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Mathematics
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2013

ABSTRACT

Automata with counter monoids, grammars and automata, context sensitive languages and word problems are examined. Included are: the non-existence of a classifying counter language for context sensitive languages, a method of forming a grammar for a language given an automaton that accepts the language, and closure properties of free products. The main result involves a form of counter that is closed under free products of word problems.

DEDICATION

I dedicate this dissertation to my wife and son. Without them I wouldn't have the support or the motivation to complete it.

ACKNOWLEDGMENTS

I would like to acknowledge the following people for their contributions to my undergraduate and graduate careers: Dr. Jon Corson for the interest in the area and for working with me. Dr. Zhijian Wu and Dr. Martin Evans for the extra time that they gave during my time as an undergraduate.

TABLE OF CONTENTS

CHAPTER 1

Introduction

This work deals with automata and the languages that they accept. The work began primarily from a paper by Corson [**1**] and a paper by Gilman [**4**]. It led to several results, which turned out to have been discovered by others in the intervening time. As I prefer a constructive proof when plausible and since it is possible to construct the necessary automaton to show theorem 4.9, I will include this new proof. With the work to that point shown to be not necessarily in vain but at least mostly unproductive, I looked for new targets.

To ensure that no more of my work would be a repeat of known results, I spent time reviewing several papers in the area. Among them were papers by Elder [**3**] and Kambites [**6**]. Over time, it became clear that word problems play a strong role in the family of languages accepted by a counter monoid. To take advantage of this fact, we develop a theory of word problems as languages, the topic of Chapter 3.

Additionally, these papers sparked an interest in context sensitive languages which led to further research in formal language theory. I used Hopcroft and Ullman[**5**] for some general information in formal languages, and set out to find an identifying counter for context sensitive languages. It would have been a great find, but Theorem 8.6 shows that no such thing exists. To prove Theorem 8.6, I needed to research grammars. This research into grammars led to the results of chapter 7.

Next it was asked, "If two word problems are in the family of languages of a common counter, when would their product be?" Some examples of special counters that this would be true for are easy to make. If $G \times G \hookrightarrow G$ then it is easy to show that the direct product is accepted. The search for a more general result lead to

the creation of monoid theory concerning *right invertible bases*, the topic of chapter 5. This was developed to prove the main theorem, Theorem 6.2, which describes a way to modify a counter so that it will be closed under free products of word problems.

In order to progress to the results, we need to establish some definitions and properties concerning monoids. This is the subject of chapter 2.

CHAPTER 2

Preliminaries: Some Monoid Theory

One can become used to thinking in terms of groups, but many of the results concerning counter automata can be generalized to monoids. At times there will be a benefit to using a monoid that is not a group. To take advantage of monoids more fully, we will need to present some general information on monoids.

2.1. Binary relations

Let X and Y be sets and let $R \subseteq X \times Y$. Such a subset R is called a *binary relation* from X to Y. For any $x \in X$, we write $R[x] = \{y \in Y \mid (x, y) \in R\}$. More generally, for any subset A of X, let $R[A] = \bigcup\{R[a] \mid a \in A\}$.

The *inverse* of a binary relation $R \subseteq X \times Y$ is the binary relation $R^{-1} \subseteq Y \times X$ given by $R^{-1} = \{(y, x) \mid (x, y) \in R\}$. The *composition* of binary relations $R \subseteq X \times Y$ and $S \subseteq Y \times Z$ is the binary relation $SR \subseteq X \times Z$ given by

$$SR = \{(x, z) \mid \text{there exists } y \in Y \text{ such that } (x, y) \in R \text{ and } (y, z) \in S\}.$$

The *diagonal* $D(X) = \{(x, x) \mid x \in X\}$ in $X \times X$ is the "equality" binary relation on X. For any relation $R \subseteq X \times Y$, note that the compositions $R\,D(X) = R$ and $D(Y)\,R = R$.

Now, let's look at some useful definitions and properties.

2.1. *Composition of binary relations is an associative operation: if $R_i \subseteq X_i \times X_{i+1}$ is a binary relation for $i = 1, 2, 3$, then*

$$(R_3R_2)R_1 = R_3(R_2R_1).$$

PROOF. Let $(x_1, x_4) \in (R_3R_2)R_1$. There exists $x_2 \in X_2$ such that $(x_1, x_2) \in R_1$ and $(x_2, x_4) \in R_3R_2$. There also exists $x_3 \in X_3$ such that $(x_2, x_3) \in R_2$ and $(x_3, x_4) \in R_3$. Note also that this means $(x_1, x_3) \in R_2R_1$. Now we have: $(x_3, x_4)(x_1, x_3) = (x_1, x_4) \in R_3(R_2R_1)$. Containment the other direction is shown similarly. $\square$

2.2. *Let $R \subseteq X \times Y$ and $S \subseteq Y \times Z$ be binary relations. Then*

(1) *$(SR)^{-1} = R^{-1}S^{-1}$ is the formal inverse;*

(2) *for any subset A of X, we write $(SR)[A] = S[R[A]]$.*

For the next observation, we will need the following definition: Let $T_i \subseteq X_i \times Y_i$ be a binary relation for $i = 1, 2$. Then, we denote by $T_1 \times T_2$ the binary relation from $X_1 \times X_2$ to $Y_1 \times Y_2$ consisting of all pairs $((x_1, x_2), (y_1, y_2))$ such that $(x_i, y_i) \in T_i$ for $i = 1, 2$.

2.3. *If $R \subseteq X_1 \times X_2$ is a binary relation, then $S = (T_1 \times T_2)[R]$ is a binary relation from Y_1 to Y_2 and*

$$S = (T_1 \times T_2)[R] = \bigcup \{T_1[x_1] \times T_2[x_2] \mid (x_1, x_2) \in R\} = T_2RT_1^{-1}$$

$$\begin{array}{ccc} X_1 & \xrightarrow{R} & X_2 \\ {\scriptstyle T_1}\downarrow & & \downarrow{\scriptstyle T_2} \\ Y_1 & \xrightarrow{S} & Y_2 \end{array}$$

Let X be a set. By a *binary relation* on X we mean a relation R from X to itself. A binary relation R on X is called an *equivalence relation* if it satisfies three properties:

(1) Reflexive: R contains the diagonal $D(X) = \{(x, x) \mid x \in X\}$.

(2) Symmetric: $R^{-1} = R$.

(3) Transitive: $R^2 \subseteq R$.

It follows that if R is an equivalence relation, then $R^2 = R$.

2.2. Congruence relations on monoids

An equivalence relation R on a monoid M is called a *congruence* if it satisfies the property: $(a_1b_1, a_2b_2) \in R$ whenever $(a_1, a_2) \in R$ and $(b_1, b_2) \in R$. In other words, a congruence on a monoid M is an equivalence relation R which when viewed as a subset of the product monoid $M \times M$, it is a submonoid.

Remark. The diagonal in $M \times M$ is a congruence on M denoted by $D(M)$, and $M \times M$ itself is a congruence on M. These are respectively the unique smallest and largest congruences on M: if R is any congruence on M, then

$$D(M) \subseteq R \subseteq M \times M.$$

2.4. *Let R be an equivalence relation on a monoid M. Then R is a congruence if and only if for all elements $a, b, c \in M$,*

$$(a, b) \in R \implies (ca, cb) \in R \text{ and } (ac, bc) \in R.$$

Proof. First assume that $(a, b) \in R \implies (ca, cb) \in R$ and $(ac, bc) \in R$ is true for all $a, b, c \in M$. Given $(a_1, a_2) \in R$ and $(b_1, b_2) \in R$, by the assumption $(a_1b_1, a_2b_1) \in R$ and $(a_2b_1, a_2b_2) \in R$. Composing these relations we get $(a_1b_1, a_2b_2) \in R$, thus R is a congruence.

Now assume that R is a congruence, and note that $(c, c) \in R$ because $(c, c) \in D(M)$. Applying the operation in the definition of congruence we have that $(ca, cb) \in R$ and $(ac, bc) \in R$. □

We wish to define a "congruence closure" operation for arbitrary binary relations on a monoid. We do that in the following way:

2.5. *Let $(R_i \mid i \in I)$ be a family of congruence relations on a monoid M. Then the intersection $\bigcap_{i\in I} R_i$ is also a congruence relation on M.*

PROOF. First, note that $\bigcap_{i\in I} R_i$ is non-empty since for any congruence R on M, $D(M) \subseteq R$. For any $(a, b) \in \bigcap_{i\in I} R_i$, we have $(ca, cb) \in R_i$ and $(ac, bc) \in R_i$ for all i since each R_i is a congruence relation on M. Thus, $(ca, cb) \in \bigcap_{i\in I} R_i$ and $(ac, bc) \in \bigcap_{i\in I} R_i$ and by 2.4 above, $\bigcap_{i\in I} R_i$ is a congruence relation on M. □

It follows that every binary relation S on M is contained in a unique smallest congruence relation—namely, the intersection of all congruence relations that contain S. We denote this congruence by $\overline{S}$ and call it the *congruence generated by* S.

Alternatively, for all $x, y \in M$, write $x \to y$ whenever $x = cad$ and $y = cbd$ for some elements $a, b, c, d \in M$ such that $(a, b) \in S \cup S^{-1}$. Then $\to$ is a symmetric relation on M with the following property:

2.6. *The monoid congruence $\overline{S}$ generated by S is the reflexive and transitive closure of the relation $\to$. Thus, for all $a, b \in M$, $(a, b) \in \overline{S}$ if and only if there is a finite sequence $a_0, a_1, \ldots, a_n$ in M such that*

$$a = a_0 \to a_1 \to \cdots \to a_n = b.$$

When $M = G$ is a group, congruence relations correspond to normal subgroups of G and the product of two congruences is again a congruence. This is not necessarily true in general for monoids.

2.7. *Let R_1 and R_2 be congruence relations on a monoid M. Then R_1R_2 is a congruence relation if and only if $R_1R_2 = R_2R_1$. In this case, $R_1R_2 = \overline{R_1 \cup R_2} = R_2R_1$.*

PROOF. Assume that $R_1R_2 = R_2R_1$.

First we need to show that R_1R_2 is an equivalence relation.

(1) R_1R_2 is reflexive since $D(M) \in R_i$ implies $D(M) \in R_1R_2$.

(2) R_1R_2 is symmetric as follows:
$(R_1R_2)^{-1} = (R_2)^{-1}(R_1)^{-1}$. Since R_1 and R_2 are reflexive we have $(R_2)^{-1}(R_1)^{-1} = R_2R_1 = R_1R_2$.

(3) R_1R_2 is transitive by: $(R_1R_2)^2 = R_1(R_2R_1)R_2 = R_1(R_1R_2)R_2 = R_1^2R_2^2 = R_1R_2$.

Thus R_1R_2 is an equivalence relation.

We show R_1R_2 is a congruence by 2.4. Let $(a, b) \in R_1R_2$ and $c \in M$. There exists $y \in M$ such that $(a, y) \in R_2$ and $(y, b) \in R_1$. Note that $(c, c) \in R_i$, therefore $(ca, cy) \in R_2$ and $(cy, cb) \in R_1$ thus $(ca, cb) \in R_1R_2$. Similarly, $(ac, yc) \in R_2$ and $(yc, bc) \in R_1$ thus $(ac, bc) \in R_1R_2$. R_1R_2 is a congruence.

Now, assume R_1R_2 is a congruence. Since R_1R_2 is a congruence it is symmetric, and $R_1R_2 = (R_1R_2)^{-1} = R_2^{-1}R_1^{-1} = R_2R_1$. □

Note that if R is a congruence on a monoid M, then the equivalence class of the identity $R[1]$ is a submonoid of M. However, not every submonoid of M arises this way. Those that do are characterized by the following result:

2.8. *Let M be a monoid and let L be a subset of M. Then $L = R[1]$ for some congruence R on M if and only if L satisfies the two properties:*

(1) $1 \in L$;

(2) *for all $a, b \in M$ and all $c \in L$, $ab \in L$ if and only if $acb \in L$.*

Proof. Suppose L satisfies the two properties, and let R be the congruence generated by the binary relation $L \times \{1\}$. By 2.6, R is the reflexive and transitive closure of the relation on M given by $ab \to acb$ and $acb \to ab$ for all $a, b \in M$ and all

$c \in L$. In particular, $R[1]$ consists of all $x \in M$ such that

$$1 \to x_1 \to \cdots \to x_n = x$$

for some $n \geq 0$ and $x_i \in M$. It follows easily by induction on n that $R[1] \subseteq L$. Thus, $L = R[1]$ as $L \times \{1\} \subseteq R$.

The converse follows immediately from the definition of a congruence on M. $\square$

The reader should note here that when $M = G$ is a group, the two conditions of 2.8 are equivalent to L being a normal subgroup of G.

2.3. Quotients of monoids and presentations

Since every monoid congruence corresponds to a quotient monoid we have the following:

2.9. *Let R be an equivalence relation on a monoid M. Then M/R is a monoid and the quotient map $M \to M/R$ is a homomorphism if and only if R is a congruence.*

Next we consider mappings of congruences given by homomorphisms of monoids. Under inverse images, this is very simple.

2.10. *Let $\alpha\colon M \to M'$ be a homomorphism of monoids and let R' be a congruence on M'. Then $(\alpha \times \alpha)^{-1}[R'] = \alpha^{-1}R'\alpha$ is a congruence on M.*

On the other hand, images $(\alpha \times \alpha)[R]$ of congruences are not always congruences even when $\alpha\colon M \to M'$ is a surjective homomorphism. The problem is actually more fundamental: if R is an equivalence relation on M, then $(\alpha \times \alpha)[R]$ may not be an equivalence relation on M'.

2.11 (Correspondence Theorem). *Let $\alpha\colon M \to M'$ be a surjective homomorphism of monoids and let $K = \alpha^{-1}\alpha$. Then there is a one-to-one correspondence given by*

$$R \mapsto (\alpha \times \alpha)[R] = \alpha R \alpha^{-1}$$

between the set of all congruences R on M such that $K \subseteq R$ and the set of all congruences on M'. Furthermore, if R is a congruence on M with $K \subseteq R$ and $R' = \alpha R \alpha^{-1}$ is the corresponding congruence on M', then there is a unique isomorphism of monoids $\overline{\alpha}\colon M/R \to M'/R'$ such that the diagram

$$\begin{array}{ccc} M & \xrightarrow{\alpha} & M' \\ \downarrow & & \downarrow \\ M/R & \xrightarrow{\overline{\alpha}} & M'/R' \end{array}$$

commutes.

A *presentation* of a monoid consists of a set A of generators and a binary relation R on the free monoid A^*. The monoid with this presentation is the quotient monoid $A^*/\overline{R}$, where $\overline{R}$ is the congruence relation on A^* generated by R.

2.4. Rational relations on monoids

Let M and M' be monoids. A binary relation $R \subseteq M \times M'$ is called a *rational relation* if it is a rational subset of the monoid $M \times M'$. A *choice of generators* for a monoid M is a set A and surjective monoid homomorphism $\alpha\colon A^* \to M$. The following lemmas will be used in proofs in chapter 4:

Lemma 2.12. *Let $\alpha\colon A^* \to M$ and $\beta\colon B^* \to M'$ be choices of generators for monoids M and M'. Then for every rational relation $S \subseteq A^* \times B^*$, the composition $T = \beta S \alpha^{-1} \subseteq M \times M'$ is also a rational relation; and every rational relation from M*

to M' arises in this way.

$$\begin{array}{ccc} A^* & \xrightarrow{S} & B^* \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\beta} \\ M & \xrightarrow{T} & M' \end{array}$$

PROOF. For any $S \subseteq A^* \times B^*$, note that $T = \beta S \alpha^{-1} = (\alpha \times \beta)[S]$ is the image of S under the epimorphism $\alpha \times \beta$. The result follows since a homomorphism of monoids maps rational subsets to rational subsets, and every rational subset in its range is the image of some rational subset. □

LEMMA 2.13 (Gilman [**4**]). *If $R_1 \subseteq M \times \Sigma^*$ and $R_2 \subseteq \Sigma^* \times M'$ are rational relations, then $R_2 \circ R_1 \subseteq M \times M'$ is also rational.*

With this background on monoids, we can now move on to word problems.

CHAPTER 3

Word Problems for Groups and Monoids

If the word problem of G is accepted by an automaton with a counter monoid M then the entire family of languages of G is accepted by M-automata; we will show a proof of this in chapter 4. Because the word problem is so important, we need to examine word problems as they relate to monoids and groups, particularly the structure of the word problem as a language. We will also show that the choice of generating set is not significant.

A *choice of generators* (or *generating set*) for a monoid M consists of a set A and a surjective homomorphism $\alpha\colon A^* \to M$. The *word problem* for M with respect to this choice of generators is the language of all words $w \in A^*$ that evaluate to 1 in M. We denote it by

$$W(M, A) = \{w \in A^* \mid \alpha(w) = 1\}.$$

For two choices of generators the following arises naturally:

LEMMA 3.1. *Let $\alpha\colon A^* \to M$ and $\beta\colon B^* \to M$ be two choices of generators for a monoid M. Then there is a homomorphism $f\colon A^* \to B^*$ such that $\beta f = \alpha$ and hence $W(M, A) = f^{-1}(W(M, B))$.*

When $M = G$ is a group, we often prefer to use a *symmetric* choice of generators. By this, we mean a choice of generators $\alpha\colon A^* \to G$ together with a fixed point free involution $a \mapsto a^{-1}$ on A such that $\alpha(a^{-1}) = \alpha(a)^{-1}$ for all $a \in A$. We extend this involution to A^* in the usual way: $(a_1 \cdots a_n)^{-1} = a_n^{-1} \cdots a_1^{-1}$. In this situation, the word problem $W(G, A)$ corresponds to the problem of deciding whether two words u

and v in A^* evaluate to the same group element in G, as this is equivalent to deciding the membership of $u^{-1}v$ in $W(G, A)$.

The languages $L \subseteq A^*$ that arise as word problems are characterized by the following result.

PROPOSITION 3.2. *Let L be a language in A^*. Then L is the word problem of some monoid M with respect to some choice of generators $\alpha\colon A^* \to M$ if and only if L satisfies the two conditions:*

(1) $\epsilon \in L$;

(2) *for all $u, v \in A^*$ and all $w \in L$, $uwv \in L$ if and only if $uv \in L$.*

PROOF. If L is a word problem, then certainly it satisfies the two conditions. For the converse, let M be the monoid given by the presentation with generators A and defining binary relation $L \times \{\epsilon\}$. Note that the conditions meet the form set in 2.8 and L is a submonoid equal to the equivalence class of the identity, the word problem in this context. □

When dealing with word problems for groups, the following characterization is particularly interesting:

LEMMA 3.3. *Let L be a language in A^*. Then L is the word problem for some group G with respect to some choice of generators $\alpha\colon A^* \to G$ if and only if L satisfies the two conditions of the previous result and the condition:*

(3) *each $a \in A$ is the first letter of some word in L; in other words, for each $a \in A$ there exists a word u_a in A^* such that $au_a \in L$.*

PROOF. Let L be a language in A^* that satisfies the three conditions (1)–(3). Let R be the congruence generated by $L \times \{\epsilon\}$. By the previous result, A^*/R is a monoid whose word problem with respect to the natural choice of generators $A^* \to A^*/R$ is $R[\epsilon] = L$. We claim that condition (3) implies that every element of $G = A^*/R$ has

a right inverse and hence G is a group. For if $w = a_1 \cdots a_n$ is a word in A^*, write $w^{-1} = u_{a_n} \cdots u_{a_1}$. Then

$$ww^{-1} \to a_1 \cdots a_{a_{n-1}} u_{a_{n-1}} \cdots u_{a_1} \to \cdots \to \epsilon$$

as each $a_i u_{a_i} \in L$. It follows that $R[w^{-1}]$ is a right inverse of $R[w]$ in G. $\square$

CHAPTER 4

Languages Accepted by G-automata

With the background on monoids and word problems in place, we are ready to address the main topic of this dissertation: which languages are accepted by machines with chosen counters. Now, we begin to look at the role the counter plays in determining the languages accepted by G-automata. Note that the major role of the word problem in this context is shown in this chapter.

In [**4**], Gilman associates a family of languages to each pair (G, X), where G is a monoid and X is a nonempty subset of G. This family $\mathcal{L}(G, X)$ consists of all languages obtained as follows. Let Σ be a finite set and $L \subseteq \Sigma^*$, then L is a language over the alphabet Σ. The language L is in $\mathcal{L}(G, X)$ if and only if there exists a rational relation $R \subseteq G \times \Sigma^*$ such that $L = R[X] = \{w \in \Sigma^* \mid (x, w) \in R \text{ for some } x \in X\}$.

According to the following observation, we may always replace G by a free monoid.

PROPOSITION 4.1. *Let $\alpha\colon A^* \to G$ be a choice of generators for a monoid G and let $X \subseteq G$ be a nonempty subset. Then*

$$\mathcal{L}(G, X) = \mathcal{L}(A^*, \alpha^{-1}[X])$$

PROOF. Let L be a language over a finite alphabet Σ. Suppose first that $L \in \mathcal{L}(G, X)$. Then there exists a rational relation $R \subseteq G \times \Sigma^*$ such that $R[X] = L$. By Lemma 2.12, we can choose a rational relation $S \subseteq A^* \times \Sigma^*$ such that $R = \mathrm{id}_{\Sigma^*} S\alpha^{-1} = S\alpha^{-1}$. Then $S[\alpha^{-1}[X]] = R[X] = L$, and thus $L \in \mathcal{L}(A^*, \alpha^{-1}[X])$.

Conversely, assume that $L \in \mathcal{L}(A^*, \alpha^{-1}[X])$ and choose a rational relation $S \subseteq A^* \times \Sigma^*$ such that $S[\alpha^{-1}[X]] = L$. By Lemma 2.12, $R = S\alpha^{-1} = \mathrm{id}_{\Sigma^*} S\alpha^{-1} \subseteq G \times \Sigma^*$ is a rational relation and $R[X] = (S\alpha^{-1})[X] = L$. Thus $L \in \mathcal{L}(G, X)$. □

We will be primarily concerned only with the case when $X = \{1\}$. In this case we write simply $\mathcal{L}(G) = \mathcal{L}(G, \{1\})$.

COROLLARY 4.2. *If $\alpha\colon A^* \to G$ is a choice of generators for G, then*

$$\mathcal{L}(G) = \mathcal{L}(A^*, W)$$

where $W = W(G, A) = \alpha^{-1}[1]$ is the word problem for G relative to the generating set A.

LEMMA 4.3 (Corson [**1**]). *Let H and K be monoids. If K is finitely generated and $W(K) \in \mathcal{L}(H)$, then $\mathcal{L}(K) \subseteq \mathcal{L}(H)$.*

PROOF. Let L be a language over an alphabet Σ such that $L \in \mathcal{L}(K)$. Choose a finite generating set $\alpha\colon A^* \to K$ for K. By hypothesis, there exists a rational relation $R \subseteq H \times A^*$ such that $R[1] = W$, where $W = W(K, A)$. Then by Corollary 4.2, there exists a rational relation $S \subseteq A^* \times \Sigma^*$ such that $S[W] = L$. Now $S \circ R \subseteq H \times \Sigma^*$ is a rational relation by Lemma 2.13, and $(S \circ R)[1] = S[R[1]] = S[W] = L$. Thus $L \in \mathcal{L}(H)$. □

EXAMPLES 4.4. (1) Regular languages. If G is a finite monoid, then $\mathcal{L}(G)$ is the family of regular languages.

(2) Context-free languages. In [**4**] a monoid denoted M_{cf} is defined such that $\mathcal{L}(M_{cf})$ is the family of all context-free languages. Based on that observation, Corson shows in [**1**] that $\mathcal{L}(F_2)$ is also the family of context-free languages, where F_2 is the free group of rank 2. The reader should note that in the next chapter, we introduce the notion of a monoid with right invertible basis and show that for the monoid CF_2

with right invertible basis of rank 2, the family $\mathcal{L}(CF_2)$ is also that of all context-free languages.

(3) Context-sensitive languages. There is no monoid G such that $\mathcal{L}(G)$ is the family of all context-sensitive languages. This is shown in Chapter 8.

(4) Recursively enumerable languages. In [**7**, Theorem 10], it is shown that $\mathcal{L}(F_2 \times F_2)$ is the family of recursively enumerable languages.

(5) Blind counter languages. The languages in the family $\mathcal{L}(Z^n)$ are called *blind n-counter languages*.

A machine called an extended finite automaton with register monoid G (or G-automaton, for short) in the sense of [**2**] can be used to recognize a language in the family $\mathcal{L}(G)$. Using Corollary 4.2, we modify the definition of a G-automaton slightly to better suit our purposes. However, first let us set up some notation and recall the basic definition of a finite automaton over a monoid.

Recall that a *directed graph* Γ consists of two sets $V(\Gamma)$, $E(\Gamma)$ and two functions $s, t\colon E(\Gamma) \to V(\Gamma)$. The members of $V(\Gamma)$ are called *vertices* or *states*; the members of $E(\Gamma)$ are called (directed) *edges*. If $e \in E(\Gamma)$, then $s(e)$ is called the *source* of e and $t(e)$ is called the *target* of e.

Now let M be a monoid. A *finite automaton* over M is a finite directed graph Γ with

(i) an edge labeling function $\lambda\colon E(\Gamma) \to M$;

(ii) a start state $s(\Gamma) \in V(\Gamma)$;

(iii) a set of final states $F(\Gamma) \subseteq V(\Gamma)$.

Extend the labeling function λ to a homomorphism from the free monoid $E(\Gamma)^*$ to M. Then, in particular, λ is defined on paths in Γ. The set of elements of M that are *accepted* by Γ are precisely all the labels of successful paths in Γ, where by a *successful path* we mean a path from the start state to a final state.

Now let G be a monoid. It is easy to see that a finite automaton with register G over the alphabet Σ, as defined in [**2**], is the same thing as a finite automaton Γ over the monoid $G \times \Sigma^*$; see [**1**]. When interpreting Γ as a G-automaton, the language $L(\Gamma) \subseteq \Sigma^*$ accepted by Γ consists of all words $w \in \Sigma^*$ such that there is a successful path p in Γ with label $\lambda(p) = (1_G, w)$. Notice that $L(\Gamma) = R[1]$, where R is the rational relation in $G \times \Sigma^*$ accepted by Γ when viewed as a finite automaton over $G \times \Sigma^*$.

We are now ready to make our definition.

DEFINITION 4.5. *Let G be a monoid, let $\alpha\colon A^* \to G$ be a choice of generators for G, and let Σ be an alphabet. A (G, A)-automaton over Σ is a finite automaton Γ over $A^* \times \Sigma^*$ such that each edge label is an element of $(A \cup \{\epsilon\}) \times (\Sigma \cup \{\epsilon\})$. The language accepted by Γ is the language $L(\Gamma)$ in Σ^* consisting of all words $w \in \Sigma^*$ such that there is a successful path p in Γ with label $\lambda(p) = (v, w)$ for some $v \in W(G, A)$.*

Similar to Corson [**1**, Proposition 2.2], we have the following:

PROPOSITION 4.6. *Let $\alpha : A^* \to G$ be a choice of generators for a monoid G. A language L over an alphabet Σ is in $\mathcal{L}(G)$ if and only if it is the language accepted by some (G, A)-automaton over Σ.*

We state the following observation for future reference.

LEMMA 4.7. *If G is a monoid and $L \in \mathcal{L}(G)$, then $L \in \mathcal{L}(K)$ for some finitely generated submonoid K of G.*

PROOF. Let L be a language over an alphabet Σ such that $L \in \mathcal{L}(G)$. Then there exists a G-automaton Γ such that $L(\Gamma) = L$. Since Γ has only finitely many edge labels in G, it can be viewed as a K-automaton for some finitely generated submonoid K of G. Hence $L \in \mathcal{L}(K)$. □

As noted in [**7**, Theorem 2], it follows immediately from this lemma that for any monoid G, the family of languages $\mathcal{L}(G)$ is equal to the union $\bigcup \mathcal{L}(K)$ as K runs through all finitely generated submonoids of G.

THEOREM 4.8. *If $W(H_i, A_i) \in \mathcal{L}(G_i)$, for $i = 1, 2$, then $W(H_1 \times H_2, A_1 \cup A_2) \in \mathcal{L}(G_1 \times G_2)$.*

PROOF. We know there exist G_i-automata over A_i, Γ_i, such that Γ_i accepts $W(H_i, A_i)$, for $i = 1, 2$. To show the result, we construct Γ in the following way:

$V(\Gamma) = V(\Gamma_1) \times V(\Gamma_2)$

$E(\Gamma) = [E(\Gamma_1) \times V(\Gamma_2)] \cup [V(\Gamma_1) \times E(\Gamma_2)]$

Let $e = (e_1, e_2) \in E(\Gamma)$. These edges are chosen so that one can only move in one of the base graphs at a time. To keep track of the movement in each graph in terms of the counter, we define a labeling function as follows:

Given $\lambda_i \colon E(\Gamma_i) \to G_i$ as the projection onto the counter coordinate for Γ_i. We define $\lambda \colon E(\Gamma) \to G_1 \times G_2$ as

$$\lambda(e) = \begin{cases} (\lambda_1(e_1), 1) & : e \in E(\Gamma_1) \times V(\Gamma_2) \\ (1, \lambda_2(e_2)) & : e \in V(\Gamma_1) \times E(\Gamma_2) \end{cases}$$

In this way the product $G_1 \times G_2$ is treated as two counters and each counter increments only when there is movement in the respective base graph.

To clarify how paths are formed we define source and target functions. The source function is defined by:

$$s(e) = \begin{cases} (s_1(e_1), e_2) & : e \in E(\Gamma_1) \times V(\Gamma_2) \\ (e_1, s_2(e_2)) & : e \in V(\Gamma_1) \times E(\Gamma_2) \end{cases}$$

The target function is defined similarly:

$$t(e) = \begin{cases} (t_1(e_1), e_2) & : e \in E(\Gamma_1) \times V(\Gamma_2) \\ (e_1, t_2(e_2)) & : e \in V(\Gamma_1) \times E(\Gamma_2) \end{cases}$$

Next we define $S_\Gamma = (S_{\Gamma_1}, S_{\Gamma_2})$ and $F_\Gamma = F_{\Gamma_1} \times F_{\Gamma_2}$ as the start and final states of Γ.

The successful paths in Γ are those made of a weaving of paths from the two base graphs. Weaving here means that the words are of the form $a_1b_1a_2b_2 \dots a_nb_n$ where $a_i \in A_1 \cup \{\epsilon\}$ and $b_i \in A_2 \cup \{\epsilon\}$. This weaving allows Γ to accept all versions of a word that allows the elements from A_1 to commute with those of A_2. In this way, Γ is a $G_1 \times G_2$-automaton that accepts $W(H_1 \times H_2, A_1 \cup A_2)$, □

The following is an alternate proof of lemma 4.3. In this version of the proof we construct an automaton to show the result.

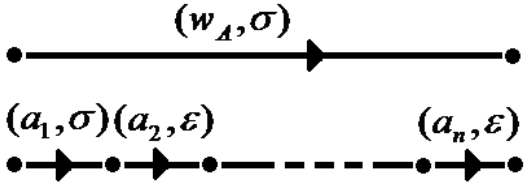
PROPOSITION 4.9. *If* $W(K) \in \mathcal{L}(H)$ *then* $\mathcal{L}(K) \subseteq \mathcal{L}(H)$.

PROOF. Let $L \in \mathcal{L}(K)$ be a language over Σ and let $W(K) \in \mathcal{L}(H)$ with A a generating set for $W(K)$. There exists a K-automaton over Σ, Γ, that accepts L, and an H-automaton over A, Δ, that accepts $W(K)$. The edges labels for Γ consist of a pair (k, σ), where $\sigma \in \Sigma' = \Sigma \cup \{\epsilon\}, k \in K$. We now write each edge label k in the alphabet A. The edge labels of Γ are now of the form (w_A, σ), where $w_A = a_1a_2 \dots a_n$ is a finite word in the alphabet A. We need the edge labels to be in A thus we subdivide the edges of Γ in the following way:

We also add loops at all vertices of Γ and Δ with empty labels. These empty loops allow one to move in a graph while waiting in the other graph when correlating words of different lengths. In this way we construct new graphs Γ' and Δ'.

Note that the edge labels of Δ' are of the form (h, a), $a \in A' = A \cup \{\epsilon\}$, $h \in H$ and the edge labels of Γ' are of the form (a, σ), $a \in A \cup \{\epsilon\}$, $\sigma \in \Sigma'$.

FIGURE 4.1. Method in which edges are subdivided.



Define $\alpha : E(\Gamma') \to A \times \Sigma'$ and define $\beta : E(\Delta') \to H \times A$. We now construct an H-automaton over Σ, M, via a pullback method such that:

$$\begin{array}{ccc} M & \longrightarrow & \Delta' \\ \downarrow & & \downarrow \beta_2 \\ \Gamma' & \xrightarrow{\alpha_1} & A' \end{array}$$

commutes, where α_1 and β_2 are the projections of α, β onto the respective coordinate.

Then $V(M) = V(\Gamma') \times V(\Delta')$ and $E(M) = \{(e_1, e_2) \in E(\Gamma') \times E(\Delta') | \alpha_1(e_1) = \beta_2(e_2)\}$. Each edge is defined by $s(e_1, e_2) = (s(e_1), s(e_2))$, and $t(e_1, e_2) = (t(e_1), t(e_2))$. The label of (e_1, e_2) is given by: $\lambda(e_1, e_2) = (\beta_1(e_2), \alpha_2(e_1)) \in H \times \Sigma'$

For M we define $S_M = (S_{\Gamma'}, S_\Delta)$ to be the start state, and $F(M) = F(\Gamma') \times F(\Delta)$ to be the final states. M is an H-automaton over Σ that accepts L since the labels on successful paths are those that would be accepted by Γ. Thus, $W(K) \in \mathcal{L}(H)$ implies $\mathcal{L}(K) \subseteq \mathcal{L}(H)$. □

Now, we've shown the importance of the word problem in G-automata and we've shown one simple closure property. In the next chapter we develop the monoid theory needed for the main theorem which gives us a stronger closure property.

CHAPTER 5

Monoids with Right Invertible Bases and Context-free Languages

In Gilman [**4**], a monoid M_{cf} is given such that $\mathcal{L}(M_{cf})$ is the family of all context-free languages. Corson shows in [**1**] that the free group F_2 also has this property. In this section we introduce a "free-er" monoid for this purpose. Roughly speaking, we will consider the free-est monoid generated by a given set X such that each element of X has a right inverse. A theory for this monoid is developed similar to that for free groups. To show the main theorem, it is convenient to use this monoid with a right invertible basis. This is made precise as follows.

DEFINITION 5.1. Let P be a monoid. A subset X of P is a *right invertible basis* for P if there exists a subset $\overline{X}$ of P, disjoint from X, and a bijection $x \mapsto \overline{x}$ of X onto $\overline{X}$ such that $x\overline{x} = 1$ for all $x \in X$ and the following universal property holds: if α is any function from the set $X \cup \overline{X}$ to a monoid M such that $\alpha(x)\alpha(\overline{x}) = 1$ for all $x \in X$, then there exists a unique extension of α to a homomorphism from P to M.

We will see that for any set X, there is an essentially unique monoid with X as a right invertible basis. The construction is very similar to that of free groups. Uniqueness is a consequence of the next proposition. It is also a more general result.

PROPOSITION 5.2. *Let P_1, P_2 be monoids with right invertible bases X_1, X_2. Then P_1 and P_2 are isomorphic if and only if X_1 and X_2 have the same cardinality.*

PROOF. Assume that $P_1 \cong P_2$ and let F_2 be the free group on X_2. Then there is a homomorphism from P_2 onto F_2 determined by $x \mapsto x$ and $\overline{x} \mapsto x^{-1}$ for all $x \in X_2$.

Taking the composition, we get a homomorphism of P_1 onto F_2 and the image of X_1 under such a map generates F_2. Thus $|X_1| \geq |X_2|$ as the free rank of F_2 is $|X_2|$. Similarly $|X_2| \geq |X_1|$, and hence $|X_1| = |X_2|$.

Conversely, assume that $|X_1| = |X_2|$. Then there exists a bijection $f\colon X_1 \cup \overline{X}_1 \to X_2 \cup \overline{X}_2$ such that $f(\overline{x}) = \overline{f(x)}$ for all $x \in X_1$. Since X_1 and X_2 are right invertible bases, there exist extensions of f to a homomorphism from P_1 to P_2 and of f^{-1} to a homomorphism from P_2 to P_1. Now the composition $f^{-1}f$ is a homomorphism from P_1 to itself that agrees with the identity map on $X_1 \cup \overline{X}_1$. Hence, by uniqueness of the extension of a map on a right invertible basis, it follows that $f^{-1}f$ must be the identity homomorphism on P_1. Likewise, ff^{-1} is the identity homomorphism on P_2 and thus $P_1 \cong P_2$. □

An immediate result is the following corollary:

Corollary 5.3. *If P is a monoid with right invertible basis, then all right invertible bases for P have the same cardinality, called the "rank" of P.*

Next we turn to the existence of monoids with right invertible bases. Let X be a set. For each $x \in X$, choose a new symbol $\overline{x}$ so that the set $\overline{X}$ of these new symbols is disjoint from X and the mapping $x \mapsto \overline{x}$ is a bijection $X \to \overline{X}$. Let $CF(X)$ be the monoid given by the presentation with generating set $X \cup \overline{X}$ and defining relations $x\overline{x} = 1$ for all $x \in X$. More precisely, form the binary relation $R = \{(x\overline{x}, 1) \mid x \in X\}$ on $(X \cup \overline{X})^*$. Then $CF(X) = (X \cup \overline{X})^*/\overline{R}$, where $\overline{R}$ is the congruence on $(X \cup \overline{X})^*$ generated by R.

Proposition 5.4. *The canonical mapping $X \cup \overline{X} \to CF(X)$ is injective and forms a right invertible basis for $CF(X)$.*

Proof. This is obvious from the construction of $CF(X)$ except possibly for the fact that $X \cup \overline{X}$ embeds in $CF(X)$. For this, note that the homomorphism of the

free monoid $(X \cup \overline{X})^*$ onto the free group $F(X)$, given by $x \mapsto x$ and $\overline{x} \mapsto x^{-1}$ for all $x \in X$, factors through $CF(X)$. Since this map embeds $X \cup \overline{X}$ in $F(X)$, our result follows. $\square$

In our construction, the elements of $CF(X)$ are the equivalence classes $[w] = \overline{R}[w]$ of words w in $(X \cup \overline{X})^*$. Analogous to the situation for free groups, we say that a word w is *reduced* if it contains no subword of the form $x\overline{x}$, $x \in X$. Clearly every word is equivalent to a reduced word, and it turns out that this reduced word is unique yielding the following normal form theorem.

PROPOSITION 5.5 (Normal forms). *Each equivalence class* $[w]$ *of words in* $(X \cup \overline{X})^*$ *contains exactly one reduced word.*

PROOF. The van der Waerden method can be adapted to this situation as follows. Let W be the set of reduced words in $(X \cup \overline{X})^*$. For each $x \in X$, define a mapping $w \mapsto x \cdot w$ from W to itself by $x \cdot w = xw$ if xw is reduced and $x \cdot w = u$ if $w = \overline{x}u$. Also define the mapping $w \mapsto \overline{x} \cdot w$ from W to itself by $\overline{x} \cdot w = \overline{x}w$. Since $x \cdot (\overline{x} \cdot w) = x \cdot \overline{x}w = w$ for all $x \in X$ and all $w \in W$, our mapping from $X \cup \overline{X}$ to the monoid $M(W)$ of self mappings of the set W extends to a homomorphism $CF(X) \to M(W)$, i.e., a monoid action of $CF(X)$ on W through self functions. This map has the property that if u is a reduced word in $(X \cup \overline{X})^*$, then $[u] \cdot \epsilon = u$. Thus, if u_1 and u_2 are reduced words in the same $\overline{R}$-equivalence class, then $u_1 = [u_1] \cdot \epsilon = [u_2] \cdot \epsilon = u_2$. The result follows. $\square$

In light of the previous proposition, the monoid $CF(X)$ can be identified with the set of all reduced words in $(X \cup \overline{X})^*$. Taking this point of view, if u and v are two reduced words, then their product in $CF(X)$ is the reduced word obtained by forming the concatenation uv and reducing it.

As another easy consequence of Proposition 5.5, note that if X is partitioned by a family of subsets $(X_i)_{i\in I}$, then the canonical homomorphism $CF(X_i) \to CF(X)$ is injective for each $i \in I$ and these maps determine an isomorphism from the free product of monoids $*_{i\in I}CF(X_i)$ to $CF(X)$.

Monoids with right invertible bases of countable rank will be most important for our purposes. We denote the unique (up to isomorphism) monoid with a right invertible basis of finite rank n by CF_n, and that of countably infinite rank by CF_∞.

LEMMA 5.6. *$\mathcal{L}(CF_\infty)$ is the family of all context-free languages.*

PROOF. It suffices to show that $\mathcal{L}(CF_\infty) = \mathcal{L}(M_{cf})$, where M_{cf} is the monoid defined by Gilman [**4**]. Let $X = \{x_1, x_2, \ldots\}$ be a right invertible basis for CF_∞. Gilman's monoid M_{cf} is generated by elements P_i, Q_i, $i = 1, 2, \ldots$, and each $P_iQ_i = 1$. Thus, there exists a surjective homomorphism $f\colon CF_\infty \to M_{cf}$ such that $f(x_i) = P_i$ and $f(\overline{x}_i) = Q_i$. Let $A = X \cup \overline{X}$ and take as choices of generators the canonical map $\alpha\colon A^* \to CF_\infty$ and the map $\beta = f\alpha\colon A^* \to M_{cf}$. Given a finite subset A_0 of A, consider the finitely generated submonoids $H = \alpha(A_0^*)$ and $K = \beta(A_0^*)$ of CF_∞ and M_{cf}, respectively.

Claim: $W(H, A_0) = W(K, A_0)$.

To see this, let w be a word in A_0^*. If $w \in W(H, A_0)$, then $\alpha(w) = 1$ and hence $\beta(w) = f(\alpha(w)) = 1$; whence $w \in W(K, A_0)$. On the other hand, assuming that $w \in W(K, A_0)$, then $\beta(w) = 1$. So by [**4**, Lemma 7.1], either $w = \epsilon \in W(K, A_0)$ or $w = ux_i\overline{x}_iv$ for some i and $u, v \in A^*$ (i.e., w contains a part that maps to P_iQ_i). In the latter case, $\beta(uv) = \beta(u)P_iQ_i\beta(v) = \beta(w) = 1$ so that $uv \in W(H, A_0)$. Now by induction on the length of the word, we conclude that $uv \in W(K, A_0)$. Then it follows that $w = ux_i\overline{x}_iv$ is also in $W(K, A_0)$. The claim is established.

By the claim and Lemma 4.3, we see that $\mathcal{L}(H) \subseteq \mathcal{L}(M_{cf})$ for each submonoid $H \leq CF_\infty$ of the form $H = \alpha(A_0^*)$, where A_0 is a finite subset of A. Since every finitely generated submonoid of CF_∞ is contained in some submonoid H of this form, it follows from Lemma 4.7 that $\mathcal{L}(CF_\infty) \subseteq \mathcal{L}(M_{cf})$. Similarly, $\mathcal{L}(M_{cf}) \subseteq \mathcal{L}(CF_\infty)$ and we are done. □

We next improve on this lemma by showing that CF_∞ can be replaced by CF_2. For this purpose, we use the following lemma.

LEMMA 5.7. *If n is a positive integer or $n = \infty$, then CF_2 contains a submonoid isomorphic to CF_n.*

PROOF. Let $X = \{x_1, x_2, \ldots\}$ and $\{x, y\}$ be right invertible bases for CF_n and CF_2. Define a function $\alpha\colon X \cup \overline{X} \to CF_2$ by

$$\alpha(x_i) = x^i y^i x^i \quad \text{and} \quad \alpha(\overline{x}_i) = \overline{x}^i \overline{y}^i \overline{x}^i$$

for all i. Then $\alpha(x_i)\alpha(\overline{x}_i) = 1$, so α extends to a homomorphism from CF_n to CF_2. We complete the proof by showing that α is injective.

Corresponding to each reduced word $w = a_1 \cdots a_n$ in $(X \cup \overline{X})^*$ is a finite sequence $(i_1, \ldots, i_n)$ of nonzero integers determined by $i_j = k$ if $a_j = x_k$ and $i_j = -k$ if $a_j = \overline{x}_k$. Since w is reduced, this sequence has the property that whenever some term is positive, the next term is not its negative. Furthermore, it is easy to see that for this word w, the reduce representation of $\alpha(w)$ is of the form

$$\alpha(w) = u_0 y^{i_1} u_1 y^{i_2} u_2 \cdots u_{n-1} y^{i_n} u_n$$

where each u_i is a nontrivial word in $\{x, \overline{x}\}^*$ and y^j is interpreted as $\overline{y}^{-j}$ if $j < 0$. It follows that for distinct reduced words u and v, $\alpha(u) \neq \alpha(v)$. □

Our theorem about CF_2 is an easy consequence of the above lemmas.

THEOREM 5.8. *$\mathcal{L}(CF_2)$ is the family of all context-free languages.*

PROOF. $CF_2 \hookrightarrow CF_\infty$ naturally, and by Lemma 5.7, $CF_\infty \hookrightarrow CF_2$. Thus $\mathcal{L}(CF_2) = \mathcal{L}(CF_\infty)$, and hence the result follows from Lemma 5.6. □

CHAPTER 6

The Word Problem of a Free Product

The main theorem of this work deals with closure properties of word problems under free products. To begin, we must clarify what a free product of word problems is as a language. Then, we will use that classification of a language as a word problem in the proof of the main theorem.

Recall that the free product of two monoids M_1 and M_2 is the monoid $M = M_1 * M_2$ that can be defined as follows: Let R be the binary relation on the free monoid $(M_1 \cup M_2)^*$ consisting of the pairs $(1_{M_1}, \epsilon)$, $(1_{M_2}, \epsilon)$, and each pair of the form (ab, c), where $a, b, c \in M_i$ and $ab = c$ in M_i for some $i = 1, 2$. Then $M_1 * M_2 = (M_1 \cup M_2)^*/\overline{R}$.

The canonical homomorphisms $M_i \to M_1 * M_2$, $i = 1, 2$, are injective. Moreover, the free product of monoids M_1 and M_2 is characterized by the following universal mapping property: if H is a monoid and $\alpha_i \colon M_i \to H$ is a homomorphism for $i = 1, 2$, then there exists a unique homomorphism α from $M_1 * M_2$ to H such that $\alpha|_{M_i} = \alpha_i$ for each $i = 1, 2$.

Let $\alpha_i \colon A_i \to M_i$ be a choice of generators for the monoid M_i. Then $\alpha \colon A_1 \cup A_2 \to M_1 * M_2$ defined by $\alpha|_{A_i} = \alpha_i$ is a choice of generators for $M_1 * M_2$. For these choices of generators, the word problem for $M_1 * M_2$ can be characterized as described by the next lemma.

LEMMA 6.1. *Let $L \subseteq (A_1 \cup A_2)^*$. Then $L = W(M_1 * M_2, A_1 \cup A_2)$ if and only if it satisfies the three conditions:*

(1) *$W(M_i, A_i) \subseteq L$ for $i = 1, 2$;*

(2) *if* $w \in L$ *and* $w \neq \epsilon$, *then there exists* $u \in W(M_i, A_i) \setminus \{\epsilon\}$ *for some* $i = 1, 2$ *such that* $w = w_1 u w_2$ *and* $w_1 w_2 \in L$;

(3) *if* $w_1, w_2 \in (A_1 \cup A_2)^*$ *and* $w_1 w_2 \in L$, *then* $w_1 L w_2 \subseteq L$.

PROOF. Write $W = W(M_1 * M_2, A_1 \cup A_2)$ and note that (1)–(3) hold for $L = W$; (1) and (3) follow immediately and (2) is a consequence of the normal form theorem. Now let $L \subseteq (A_1 \cup A_2)^*$ be any language that satisfies (1)–(3). It remains to show that $L = W$.

Let $w \in L$. If $w = \epsilon$, then $w \in W$. Next, we assume that $w \neq \epsilon$. Then $w = w_1 u w_2$, where $\epsilon \neq u \in W(M_i, A_i)$ and $w_1 w_2 \in L$, by (2). So $|w_1 w_2| < |w|$, and thus $w_1 w_2 \in W$ by induction on length. Since $u \in W(M_i, A_i) \subseteq W$, it follows that $w \in W$. Hence $L \subseteq W$.

On the other hand, let $w \in W$. If $w = \epsilon$, then $w \in L$ by (1), so we may assume that $w \neq \epsilon$. Since W satisfies (2), there exist $i \in \{1, 2\}$ and $\epsilon \neq u \in W(M_i, A_i)$ such that $w = w_1 u w_2$ and $w_1 w_2 \in W$. However $|w_1 w_2| < |w|$, so $w_1 w_2 \in L$ by induction on length. Now, we also have $u \in L$ by (1). Thus, it follows by (3) that $w \in L$, and hence $W \subseteq L$. □

We now state and prove the main theorem. Let G denote a finitely generated monoid.

THEOREM 6.2 (Main Theorem). *Let* M_1, M_2 *be finitely generated monoids. If each* $W(M_i)$ *is in* $\mathcal{L}(G)$, *then* $W(M_1 * M_2)$ *is in* $\mathcal{L}(G * CF_2)$.

Our proof relies on the following lemma. For this and the proof of the theorem, fix a finite generating set B and $\gamma\colon B^* \to G$ for G.

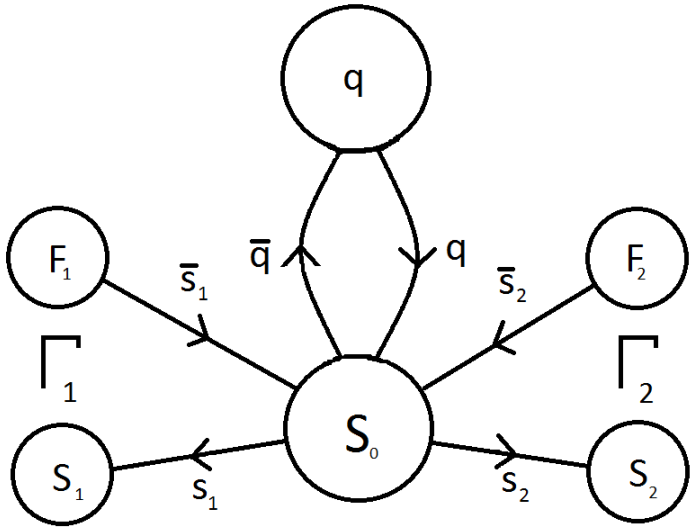
LEMMA 6.3. *Let* X *be a finite set and form the finite set of generators* $C = B \cup X \cup \overline{X}$ *for* $G * CF(X)$. *For any word* w *in* C^*, *if* w *is in* $W(G * CF(X), C)$ *then either*

(1) $w \in W(G, B)$, *or*

(2) $w = w_1 x u \overline{x} w_2$, *where* $w_i \in C^*$, $x \in X$, *and* $u \in W(G, B)$.

PROOF. Suppose $w \in W(G * CF(X), C)$. Let $\theta\colon C^* \to (X \cup \overline{X})^*$ be the projection determined by mapping B to ϵ and mapping $X \cup \overline{X}$ to itself. Then $\theta(w)$ is in $W(CF(X), X \cup \overline{X})$. If $\theta(w) = \epsilon$, then $w \in B^*$ and it follows that $w \in W(G, B)$ since $G \hookrightarrow G * CF(X)$. So assume that $\theta(w) \neq \epsilon$. Then by proposition 5.5, there are words $v_1, v_2 \in (X \cup \overline{X})^*$ and $x \in X$ such that $\theta(w) = v_1 x \overline{x} v_2$. Thus $w = w_1 x u \overline{x} w_2$, where $\theta(w_i) = v_i$ and $u \in B^*$. Furthermore, u must be in $W(G, B)$ for otherwise, $xu\overline{x}$ would be a nontrivial part of the normal form of w and hence w would not evaluate to 1 in $G * CF(X)$. □

FIGURE 6.1. Automaton used in proof of the Main Theorem. Edge labels are of the form (x, ϵ), where x is shown in the figure.



PROOF OF THE MAIN THEOREM. Let Γ_i be a (G, B)-automaton over A_i that accepts the word problem $W(M_i, A_i)$. We assume for convenience, as we may by [**4**, Lemma 2.5], that the start state s_i of Γ_i has no inedges, and that Γ_i has a single final

state f_i which has no outedges. Let C be the set of generators for $G * CF(X)$ as in Lemma 6.3, where $X = V(\Gamma_1) \cup V(\Gamma_2)$. Construct a $(G * CF(X), C)$-automaton Γ over $A_1 \cup A_2$ consisting of copies of the labeled graphs Γ_i, one new state s_0, and some new edges E' joining s_0 to vertices of the Γ_i as follows:

The vertex set $V(\Gamma) = V(\Gamma_1) \cup V(\Gamma_2) \cup \{s_0\}$, where s_0 is a new state. The edge set $E(\Gamma) = E(\Gamma_1) \cup E(\Gamma_2) \cup E'$, where E' consists of a directed edge $[s_0, s_i]$ from s_0 to s_i with label (s_i, ϵ), a directed edge $[f_i, s_0]$ from f_i to s_0 with label $(\overline{s_i}, \epsilon)$, and for each state q in $V(\Gamma) \setminus \{s_0, s_1, s_2, f_1, f_2\}$ an edge $[q, s_0]$ from q to s_0 with label (q, ϵ) and an edge $[s_0, q]$ from s_0 to q with label $(\overline{q}, \epsilon)$; see Figure 6.1. The start state is s_0 which is also the sole final state of Γ. We claim that $L(\Gamma) = W(M_1 * M_2, A_1 \cup A_2)$.

To establish our claim, it suffices to verify that the three conditions of Lemma 6.1 hold for the language $L(\Gamma)$. (1) this follows immediately from the way Γ_i is embedded in Γ and since $L(\Gamma_i) = W(M_i, A_i)$.

(2) Let $w \in L(\Gamma)$ with $w \neq \epsilon$. Choose a successful path p in Γ of minimal length with label of the form $\lambda(\gamma) = (v, w)$, where v is in $W(G * CF(X), C)$. Since p is a nontrivial path starting at s_0, we see that v is not in B^*. So, by Lemma 6.3, $v = v_1 x z \overline{x} v_2$ for some words $v_i \in C^*$, $x \in X$, and $z \in W(G, B)$. By the nature of Γ, it follows that the path p must be of the form $p = p_1 e p' e' p_2$, where $\lambda(p_i) = (v_i, w_i)$, $\lambda(p') = (z, u)$, $e, e' \in E'$ with $\lambda(e) = (x, \epsilon)$, $\lambda(e') = (\overline{x}, \epsilon)$, and $w = w_1 u w_2$. Notice that if $e = [q, s_0]$ and $e' = [s_0, q]$ for some q in $V(\Gamma_1) \cup V(\Gamma_2)$, then p' would have to be the trivial path at s_0 as otherwise u could not be a word in B^*. Hence, if this were the case, then the path $p_1 p_2$ would be a successful path in Γ with label $\lambda(p_1 p_2) = (v_1 v_2, w_1 w_2) = (v_1 v_2, w)$ and $v_1 v_2 \in W(G * CF(X), C)$. However, the length of $p_1 p_2$ is two less than that of p contradicting the choice of p. Consequently, it must be that $e = [s_0, s_i]$ and $e' = [f_i, s_0]$ for some $i = 1, 2$. So p' is a path in $\Gamma_i \subseteq \Gamma$ from s_i to f_i with label (z, u) and $z \in W(G, B)$; whence $u \in L(\Gamma_i) = W(M_i, A_i)$. Furthermore, removing the loop $ep'e'$ from p, we get the path $p_1 p_2$ from s_0 to itself

with label $\lambda(p_1p_2) = (v_1v_2, w_1w_2)$ and v_1v_2 is in $W(G * CF(X), C)$. Thus we also have that $w_1w_2 \in L(\Gamma)$, and we see that the condition holds.

(3) Let $w_1, w_2, u \in (A_1 \cup A_2)^*$ such that w_1w_2 and u are in $L(\Gamma)$. Choose successful paths p, p' in Γ with labels $\lambda(p) = (v, w_1w_2)$, $\lambda(p') = (z, u)$ where v, z are in $W(G * CF(X), C)$. Then $p = p_1p_2$, where $\lambda(p_i) = (v_i, w_i)$ and $v = v_1v_2$. Let $q = t(p_1) = s(p_2)$ and form the path $p'' = p_1[q, s_0]p'[s_0, q]p_2$. So $\lambda(p'') = (v_1qz\overline{q}v_2, w_1uw_2)$ and $v_1qz\overline{q}v_2$ is in $W(G * CF(X), C)$. Thus $w_1uw_2 \in L(\Gamma)$, as required.

Now applying Lemma 6.1, we see that $L(\Gamma) = W(M_1 * M_2, A_1 \cup A_2)$ as claimed. Thus $W(M_1 * M_2, A_1 \cup A_2)$ is in $\mathcal{L}(G * CF(X))$. However, it follows from Lemma 5.7 that $G * CF(X)$ embeds in $G * CF_2$. Hence $\mathcal{L}(G * CF(X) \subseteq \mathcal{L}(G * CF_2)$, completing the proof. □

COROLLARY 6.4. *The family of all those finitely generated monoids with word problem in $\mathcal{L}(G * CF_2)$ is closed under free products of its members.*

PROOF. Let M_1, M_2 be finitely generated monoids with word problem in $\mathcal{L}(G * CF_2)$. Then by the theorem, $W(M_1 * M_2) \in \mathcal{L}(G * CF_2 * CF_2)$. However, $CF_2 * CF_2 \cong CF_4$ and there is an embedding $CF_4 \hookrightarrow CF_2$. Thus, there is an embedding $G * CF_2 * CF_2 \hookrightarrow G * CF_2$ and hence, $\mathcal{L}(G * CF_2 * CF_2) \subseteq \mathcal{L}(G * CF_2)$. □

Since $\mathcal{L}(CF_2)$ is the family of context free languages, this can be used to show that the family of finitely generated monoids with context free word problems is closed under taking free products. This is well known for groups.

CHAPTER 7

Grammar Associated to an (M, A)-automaton

The automata approach to dealing with languages identifies a language by an accepting machine. To prove some theorems in chapters 7 and 8, we need to use a different point of view, that of grammars.

A *grammar* G is defined by $G = (N, T, P, S)$, where N, T, P, S are defined as follows: N is a finite set of non-terminal symbols, T a finite set of terminal symbols, and S is the start symbol, which is a special symbol in N. The set of productions, P, is a finite set of relations on $(N \cup T)^* \times (N \cup T)^*$. The language represented by G is formed by beginning with S and applying productions from P to subwords. Any word formed in this manner that consists of only elements of T is in the language generated by G.

THEOREM 7.1. *Given a G-automaton that accepts a language L and a generating grammar for the word problem of G, one can construct a generating grammar for L.*

PROOF. Let Γ be the graph associated to a G-automaton that accepts a language L, and let Σ be the alphabet of L. Define labeling functions on the edges of Γ, $E(\Gamma)$ by:

$$\lambda \colon E(\Gamma) \to \Sigma \cup \{\epsilon\}$$

$$\mu \colon E(\Gamma) \to G$$

We form the grammar (N, T, P, S) as follows: First, we extend μ to a generating set T_0 and $\mu \colon T_0^* \to G$, $T_0 \subseteq E(\Gamma)$. Take a generative grammar (N_0, T_0, P_0, S_0) for the word problem of G with generating set T_0. Then, $N = N_0 \cup T_0 \cup N_0' \cup T_0' \cup T_0'' \cup \{S\}$

where $x \mapsto x'$ is a bijection from $T_0 \cup N_0 \to T_0' \cup N_0'$ and $x \mapsto x''$ is a bijection from $T_0 \to T_0''$, $T = \Sigma$, S a new symbol, and P is formed by:

- Include all productions in P_0,
- Add the production $S \to \epsilon$ if $\epsilon \in L$,
- Add the production $S \to a_1' a_2 \dots a_n$ for each $S_0 \to a_1 a_2 \dots a_n$ in P_0,
- Add the production $a_1' a_2 \dots a_n \to b_1' b_2 \dots b_n$ for each $a_1 a_2 \dots a_n \to b_1 b_2 \dots b_n$ in P_0.

Note that, at this point, the grammar *sub-generates* the word problem for G, but with a primed first letter. The introduction of the new start symbol S ensures that the first letter and only the first letter is primed. "Sub-generates" has a dual meaning. First, these are not terminal symbols, thus these words are not generated by the grammar. They are however, the terminals of the original grammar and are the edges. Secondly, these are not all of the words formed by these productions. All other words can never be transformed to terminals, and can be ignored since they cannot be in the formation chain of a generated word.

The next set of new productions will ensure only those words on paths from the start state to an accept state are generated.

- Add the production $e' \to e''$, if $s(e) = S(\Gamma)$,
- Add the production $e_1'' e_2 \to \lambda(e_1) e_2''$, if $t(e_1) = s(e_2)$,
- Add the production $e'' \to \lambda(e)$, if $t(e) \in F(\Gamma)$.

The single prime can only be removed when the first edge starts at the start state. The double prime can only move along the word if the edges are sequential. Lastly, the double prime can only be removed if the last edge leads to a final state. Thus the entire word can only be converted to terminals if it represents a path in the edges of Γ from the start state to the final state. In this way, we have formed a grammar that generates the language accepted by the given automaton. $\square$

CHAPTER 8

Results Concerning Context Sensitive Languages

Context sensitive languages have several different properties when compared to the other categories of languages. For example: they are not closed under homomorphism, they have a non-contracting grammar, and they do not have an identifying counter monoid. Some of these differences are exploited or exhibited in this chapter.

Definition 8.1. A grammar represents a *context sensitive language* if each production is of the form $\alpha A\beta \to \alpha\gamma\beta$, where A is a single non-terminal and γ is non-empty.

This definition shows the descriptive nature of the name context sensitive, since the change is only allowed when the conditions around the non-terminal are set. While it is descriptive, this is not the most convenient definition for our uses. Below is an equivalent definition if we allow the following conditions in the event ϵ is in L: we do not allow the start symbol, S, to appear on the right hand side of any productions and we allow $S \to \epsilon$.

Definition 8.2. A language is *context sensitive* if it has a non-contracting grammar.

Here *non-contracting grammar* means that the number of symbols on the right of each production is greater than or equal to the number of symbols on the left.

Corollary 8.3. *If a G-automaton, M, has a counter with a context sensitive word problem and no empty labeled loops, then M accepts a context sensitive language.*

PROOF. Given such an automaton, apply theorem 7.1. Note that all the new productions created in the proof of theorem 7.1 are non-contracting since each edge label is non-empty. The new grammar therefore generates a context sensitive language. Thus M accepts a context sensitive language. □

As seen in Examples 4.4, there are counter monoids such that the family of languages they accept are each of the Chomsky categories of languages except in the case of the context sensitive languages. We now show that no such identifying counter exists.

LEMMA 8.4. *Every recursively enumerable language is a homomorphic image of a context sensitive language.*

PROOF. Given $G = (N, T, P, S)$ and $L = L(G)$. Construct $G' = (N, T \cup \{z\}, P', S)$, where P' consists of:

- $\alpha \to \beta \in P$, if $|\beta| \geq |\alpha|$
- $\alpha \to \beta z^n$, where $n = |\alpha| - |\beta|$and $|\alpha| > |\beta|$
- $zx \to xz$ for all $x \in N \cup T$.

Note that, $L(G') \subseteq (T \cup \{z\})^*$, and define $f\colon (T \cup \{z\})^* \to T^*$ by

$$f(x) = \begin{cases} x & : x \in T \\ \epsilon & : x = z \end{cases}$$

We now have a context sensitive language $L(G')$ and a homomorphism f, such that $f(L(G')) = L(G)$, proving the result. □

LEMMA 8.5. *The family of languages accepted by M-automata is closed under homomorphism.*

PROOF. Let g be a homomorphism from $A^* \to B^*$. Let L be a language over A with $L \in \mathcal{L}(M)$, where $\mathcal{L}(M)$ is the family of languages accepted by automata

with counter M. There exists an M-automaton, Γ, that accepts the language L. The set of successful paths in Γ is given by $S = \{e_1, e_2, \ldots, e_n\}$, where the labels on $e_i = (m_i, a_i)$, $s(e_1)$ is a start state, $t(e_n)$ is an accept state, and $m_1 m_2 \ldots m_n = 1$ in M.

Now consider the M-automaton over B, Γ', formed by applying g to the second edge labels of all edges in Γ. The set of successful paths in Γ' is given by $S' = \{e_1, e_2, \ldots, e_n\}$, where the labels on $e_i = (m_i, g(a_i))$, $s(e_1)$ is a start state, $t(e_n)$ is an accept state, and $m_1 m_2 \ldots m_n = 1$ in M.

Note that the paths in S are the same paths in S' since the second edge label is not used in determining a successful path. The words accepted by Γ are $\{a_1 a_2 \ldots a_n\}$ and the words accepted by Γ' are $\{g(a_i)g(a_2)\ldots g(a_n)\}$. The language accepted by $\Gamma' = g(L)$, and $g(L) \in \mathcal{L}(M)$.Thus, $\mathcal{L}(M)$ is closed under homomorphism. □

THEOREM 8.6. *There does not exist a monoid, M, such that $\mathcal{L}(M)$ is the context sensitive languages.*

PROOF. Assume that M is such a monoid. Let l be a recursively enumerable language that is not context sensitive, and note that $l \notin \mathcal{L}(M)$. By lemma 8.4 there exists a context sensitive language, L, and l is the image of L under a homomorphism. Since L is context sensitve, $L \in \mathcal{L}(M)$. Lemma 8.5 tells us that $L \in \mathcal{L}(M)$ implies that $l \in \mathcal{L}(M)$. This is a contradiction, and thus M cannot exist. □

We've shown that there is no counter that identifies the context sensitive languages. We've also shown several closure properties of automata with counters, including our main theorem. To do this, we developed several theories here that could possibly be used for other applications.

References

[1] J. M. Corson. Extended finite automata and word problems. *Internat. J. Algebra Comput.*, 15(3):455-466, 2005.

[2] J. Dassow and V. Mitrana. Finite automata over free groups. *Internat. J. Algebra Comput.*, 10(6):725-737, 2000.

[3] M. Elder. G-automata, counter languages and the Chomsky hierarchy. In C. Campbell, M. Quick, E. Robertson, and G. Smith, editors, *Proceedings of Groups St Andrews 2005*, volume 339 of *London Mathematical Society Lecture Note Series*, pages 409-415, Cambridge, 2007. Cambridge University Press.

[4] R. H. Gilman. Formal languages and infinite groups. In *Geometric and Computational Perspectives on Infinite Groups*, pages 27-51, Providence, RI, 1996. American Mathematical Society.

[5] J. E. Hopcroft and J. D. Ullman. *Formal Languages and Thier Relation to Automata*. Addison-Wesly, Reading, Massachusetts, 1969.

[6] M. Kambites. Word problem recognisable by deterministic blind monoid automata. *Theoret. Comput. Sci.*, 362:232-2337, 2006.

[7] V. Mitrana and R. Stiebe. Extended finite automata over groups. *Disctete Appl. Math.*, 108(3)287-300, 2001.