CONTEMPORARY MATHEMATICS

378

Groups, Languages, Algorithms

AMS-ASL Joint Special Session on
Interactions between Logic, Group Theory,
and Computer Science
January 16–19, 2003
Baltimore, Maryland

Alexandre V. Borovik
Editor

American Mathematical Society

CONTEMPORARY
MATHEMATICS

378

Groups, Languages, Algorithms

AMS-ASL Joint Special Session on
Interactions between Logic, Group Theory,
and Computer Science
January 16–19, 2003
Baltimore, Maryland

Alexandre V. Borovik
Editor

American Mathematical Society
Providence, Rhode Island

This volume is based on the AMS-ASL Joint Special Session on "Interactions between Logic, Group Theory, and Computer Science," held in Baltimore, Maryland, January 16–19, 2003.

2000 *Mathematics Subject Classification.* Primary 20B40, 20E05, 20F28, 81P68;
Secondary 68Q05, 68Q17, 68Q42, 68Q45, 68T05.

Library of Congress Cataloging-in-Publication Data

AMS-ASL Joint Special Session on Interactions between Logic, Group Theory, and Computer Science (2003 : Baltimore, MD)

Groups, Languages, Algorithms : AMS-ASL Joint Special Session on Interactions between Logic, Group Theory, and Computer Science, January 16–19, 2003, Baltimore, Maryland /Alexandre V. Borovik, editor.

p. cm. — (Contemporary mathematics, ISSN 0271-4132 ; 378)

Includes bibliographical references.

ISBN 0-8218-3618-8 (alk. paper)

1. Group theory—Congresses. 2. Finite groups—Congresses. 3. Infinite groups—Congresses. I. Borovik, Alexandre. II. Title. III. Contemporary mathematics (American Mathematical Society) ; v. 378.

QA174.A64 2003
512′.2—dc22 2005043610

♾ The paper used in this book is acid-free and falls within the guidelines
established to ensure permanence and durability.
Visit the AMS home page at `http://www.ams.org/`

10 9 8 7 6 5 4 3 2 1 10 09 08 07 06 05

Groups, Languages, Algorithms

Preface

This volume is loosely based around the major themes of the AMS/ASL Joint Special Session on "Interactions Between Logic, Group Theory and Computer Science" held in Baltimore, Maryland, in January 2003. I wish to express my thanks to the American Mathematical Society and Association for Symbolic Logic for their invitation to organize the session and for their support, which allowed this unusual interdisciplinary meeting to take place.

Since the pioneering works of Novikov and Maltsev, group theory was a testing ground for mathematical logic in its many manifestations, from the theory of algorithms to model theory. This interaction between logic and group theory led to many prominent results which enriched both disciplines. In this volume, we collect under one cover several papers devoted to the development of technique for the group theory/logic interface. They complement the previous volume, "Computational and Experimental Group Theory" (vol. 349 of *Contemporary Mathematics*), which also arose from the Baltimore Meeting but concentrated more on a similar interaction between group theory and computer science.

The first paper in the volume, by Robert Gilman, is a detailed survey of the state of art in the theory of formal languages as applied to groups. Formal languages originated as models of spoken and written languages. Subsequently they proved useful in analyzing programming languages, and more recently connections with group theory have begun to emerge. The survey concentrates on the simplest classes of languages, namely regular, context free and indexed languages; some other classes are mentioned briefly. No knowledge of formal languages is assumed on the part of the reader. The exposition emphasizes the algebraic aspects of the subject at the expense of those related to programming; in particular, the language classes are defined in terms of monoids, one for each class.

The next paper, by Myasnikov, Remeslennikov and Serbin, dramatically expands the language metaphor: here, the elements of Lyndon's free $\mathbb{Z}[t]$-group $F^{\mathbb{Z}[t]}$ are represented by *infinite* words with a regular free Lyndon length function on $F^{\mathbb{Z}[t]}$ with values in $\mathbb{Z}[t]$. This approach allows one to solve various algorithmic problems for $F^{\mathbb{Z}[t]}$ using the standard Nielsen cancellation argument for the length function $L : F^{\mathbb{Z}[t]} \to \mathbb{Z}[t]$. The concept can be generalized to A-free groups for arbitrary discrete ordered abelian group A, by considering "words", indexed by elements of A rather than integers, and defining a suitable notion of reduced word. As the next paper, by Ian Chiswell, shows, the concept of an A-free group has a nice geometric interpretation: A-free groups are exactly tree free groups, that is, groups which act by isometries on some Λ-tree freely and without inversions.

Two major papers by Kharlampovich and Myasnikov develop the machinery for the study of the elementary theory of a free group. It appears that the methods are going far beyond the free groups. They provide structural and algorithmic results for a wide class of "free-like" groups, in particular, for finitely generated fully residually free groups. The theory is highly complex and it will take time to fully assess all its aspects and implications.

The authors divide their approach to elementary theories of groups and related problems into three stages. The first stage concerns equations over a given group G which is free or close to being free. In the classical terms the main problem here is to describe (effectively) the structure of solution sets of arbitrary systems of equations in finitely many variables over G. Different such descriptions are contained in the first paper. This requires the development of a fair amount of algebraic geometry over the group G and related groups (introduce algebraic sets and Zariski topology, coordinate groups and radicals, study Noetherian properties and irreducible components, to prove Nullstellensatz, etc.). On the group-theoretic level one needs to describe the algebraic structure of the coordinate groups of the irreducible varieties over G (which also appear as fully residually G groups, or limit groups (Sela), or models of the universal theory of G (Remeslennikov)). One of the ways to obtain this description is to describe effectively some canonical decomposition of such a group, so-called JSJ decomposition (introduced by Rips and Sela for finitely presented groups); this is the aim of the first paper. Algorithmically, everything is based on the so-called *elimination process*, described in the first paper; it resembles the classical elimination procedure in algebraic geometry. This process effectively relates all different techniques to each other. In the case of free groups it appears in the various forms of the Makanin-Razborov machine. In the second stage one has to introduce the main technical tool which allows one to eliminate a quantifier in a particular situation which can be described by an implicit function theorem or, in an algebro-geometric form, as lifting solutions of equations into generic points of varieties, or, in model-theoretic terms, as introducing basic Skolem's functions. Effective versions of these theorems are the target of the second paper. Implicit function theorems give the main tool to organize the *verification process* which checks whether a given formula in the group language holds in G or not. The termination mechanism, which ensures that the verification process terminates in finitely many steps, is the third stage—but these results are not included in the present volume.

The last paper of the present volume, by Esyp, Kazatchkov and Remeslennikov, studies the so-called *free partially commutative groups.* They arise naturally in many branches of mathematics and computer science, which led to a variety of names under which they are known in the literature: *semifree groups*, *graph groups*, *right-angled Artin groups.* The paper is concerned with the divisibility theory and the complexity of the word and the conjugacy problem in the partially commutative groups.

Alexandre Borovik
December 2004

Contents

Contemporary Mathematics
Volume **378**, 2005

Formal Languages and their Application to Combinatorial Group Theory

Robert H. Gilman

ABSTRACT. This article is an introduction to formal languages and their connections with combinatorial group theory.

CONTENTS

1. Introduction

Formal languages originated as models of spoken and written languages. Subsequently they proved useful in analyzing programming languages, and more recently connections with group theory have begun to emerge. These connections are the subject of this survey. We concentrate on the simplest classes of languages, namely regular, context free and indexed languages; some other classes are mentioned briefly at the end. We do not assume any knowledge of formal languages on the part of the reader.

Our exposition is somewhat novel in that the language classes are defined in terms of monoids, one for each class. This approach emphasizes the algebraic aspects of the subject at the expense of those related to programming. It is one of a number of treatments based on the observation that popping a letter off a stack

2000 *Mathematics Subject Classification.* Primary 20F32; Secondary 20F05 68Q42 68Q45.

Key words and phrases. Group, formal language.

The author expresses his thanks to Imperial College, for its hospitality while this survey was being written. He also thanks the referee for a careful reading and for suggestions which improved the quality of the paper.

is a right inverse to pushing it onto the stack; see [**8**, Chapter 4], [**17**, Volume A, Chapter X], [**18**], [**25**] and [**44**].

For other accounts of formal language theory the reader is referred to [**43, 29, 34**]. [**41**] and [**12**] are also worth consulting. The survey [**1**] includes the three language classes discussed here.

Much of the material here comes from lectures given at the City University of New York and the University of Neuchâtel, and from a previous survey [**23**]. Some of the proofs in the previous survey have been improved, indexed languages are treated more fully here than there, and references have been revised. A number of exercises are included; the words "Prove that" are usually omitted from the statements of these exercises.

2. Notation and definitions

Formal language theory's origin in linguistics is reflected in its terminology. An alphabet is a finite nonempty set, and a language over an alphabet Σ is a subset of Σ^*, the free monoid over Σ. Elements of Σ^* are called words over Σ. The length of a word $w \in \Sigma^*$ is $|w|$. The unit element of Σ^* is ϵ, the empty word. The empty set is ϕ; $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$.

Connections between languages and groups are usually by means of a choice of generators. For any monoid M a choice of generators is a surjective monoid homomorphism $\sigma : \Sigma^* \to M$. Sometimes we refer to a choice of generators $\Sigma^* \to M$ without naming the monoid homomorphism. In that case the image of a word w is denoted $\overline{w}$.

When $\sigma : \Sigma^* \to G$ is a choice of generators for a group G, it is tacitly assumed that Σ is equipped with formal inverses and that σ respects these inverses. In other words there is a permutation $a \to a^{-1}$ of Σ with all cycles of length two, and $\sigma(a^{-1}) = (\sigma(a))^{-1}$. It is easy to see that in this case the condition $(wv)^{-1} = v^{-1}w^{-1}$ determines a unique extension of inverses to Σ^* and that $\sigma(w^{-1}) = (\sigma(w))^{-1}$ for all $w \in \Sigma^*$.

When Σ is equipped with formal inverses, free equivalence and free reduction are defined for Σ^* in the usual way.

The word problem of a group G with respect to a choice of generators $\Sigma^* \to G$ is the language of all words mapping to 1. A combing for G is a language $L \subset \Sigma^*$ mapping onto G. A combing with uniqueness is a combing which maps bijectively to G.

3. Regular languages

The regular languages over an alphabet Σ are the closure of the finite subsets of Σ^* under under union, product, and generation of submonoid. This definition makes sense for all monoids, not just free ones; but it seems clearer to postpone the general case.

Union, product, and generation of submonoid are called the rational operations and are denoted by $S + T$, ST and S^* respectively. It is immediate from their definition that regular languages are closed under rational operations. It is almost as immediate that with the exception of ϕ and $\{\epsilon\}$ regular languages over Σ are either singleton subsets of Σ or are constructed from singleton subsets by rational operations. Thus if we use letters of Σ to denote the corresponding singleton subsets and ϵ to denote $\{\epsilon\}$, each regular language is named by a so-called regular expression composed of letters from Σ, symbols ϕ and ϵ, and and the operators $+,^*$, and concatenation.

EXAMPLE 3.1. The regular language over $\Sigma = \{a, b\}$ consisting of all words which begin with a and end with b is named by the regular expression $a(a+b)^*b$ and also by the regular expression $ab + a(a+b)(a+b)^*b$.

In general there are many regular expressions naming a given regular language. Nevertheless we identify regular expressions with the languages they name. Thus we allow ourselves to speak of regular languages $a(a+b)^*b$ and $\Sigma + \epsilon$.

EXERCISE 3.2. Find a regular expression for the language of freely reduced words over $\Sigma = \{a, a^{-1}\}$.

EXERCISE 3.3. Give a precise definition of regular expressions.

3.1. Properties of regular languages. The first property is that regular languages are the languages accepted by finite automata.

A finite automaton $\mathcal{A}$ defined over an alphabet Σ is a finite directed graph with one vertex distinguished as the initial vertex and some others as terminal vertices. Each edge of the graph is labeled by an element from Σ. The empty word ϵ may also be an edge label, in which case the automaton is said to be defined over Σ_e.

A word in Σ^* is accepted by $\mathcal{A}$ if it is the label of a successful path, that is, a path from the initial vertex to a terminal vertex. The set of accepted words is called the language accepted by the automaton.

The label of a path is of course the product of its edge labels in the order in which the edges appear in the path. A path of length 0 has label ϵ. It is possible for an automaton to have no terminal vertices; such an automaton accepts the language ϕ.

Figure 1 shows a finite automaton $\mathcal{A}$ defined over $\{a, b\}_\epsilon$. Initial and terminal vertices are indicated by arrows with no source and no target respectively. The initial vertex of $\mathcal{A}$ is its single terminal vertex. $\mathcal{A}$ accepts $abab$ but not $baab$. The reader will recognize $\mathcal{A}$ as a Cayley diagram for S_3, the nonabelian group of order 6, with edges for the inverses of the generators a and b omitted. Adding these edges yields an automaton which accepts the word problem with respect to the present

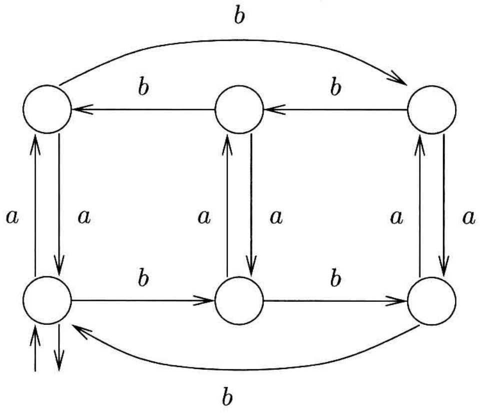

FIGURE 1. A finite automaton $\mathcal{A}$

choice of generators for S_3. It is clear from this example that the word problem for any finite group with respect to any choice of generators is accepted by a finite automaton constructed from the Cayley diagram determined by those generators.

EXERCISE 3.4. Construct automata accepting the languages ϵ, ϕ, and w where w is a nonempty word in Σ^*.

EXERCISE 3.5. Construct an automaton accepting the language of freely reduced words over $\Sigma = \{a, b, a^{-1}, b^{-1}\}$.

The following definition and lemma are preparation for Theorem 3.9, which characterizes regular languages in a number of ways.

DEFINITION 3.6. A finite automaton is deterministic if it is defined over an alphabet Σ and if no vertex has two outedges with the same label.

Deterministic finite automata are sometimes defined differently than in Definition 3.6. From now on we say automaton instead of finite automaton.

EXERCISE 3.7. If a language is accepted by a deterministic automaton over Σ, it is accepted by one such that each vertex has exactly one outedge with label a for each $a \in \Sigma$.
Hint: Add a vertex v' for which all outedges are loops back to v'.

LEMMA 3.8. *Every nonempty language accepted by an automaton is accepted by an automaton satisfying the following conditions.*

(1) *The initial vertex has no inedges.*
(2) *There is one terminal vertex, and it has no outedges.*
(3) *Each edge and each vertex is on a successful path.*
(4) *Distinct edges with the same source and target vertices have distinct labels.*

PROOF. If the initial vertex p_0 has inedges, add a new vertex p_0' and edge $p_0' \xrightarrow{\epsilon} p_0$. Make p_0' the initial vertex. Add a new terminal vertex p_t, and for each previously existing terminal vertex p, add an edge $p \xrightarrow{\epsilon} p_t$. Make p_t the unique terminal vertex. These alterations do not change the accepted language. In addition deleting edges and vertices not on successful paths and identifying multiple edges from with the same source, target and label do not change the accepted language either. Since the language is nonempy, the initial vertex and terminal vertex will survive these deletions. □

THEOREM 3.9. *Let L be a language over Σ. The following are equivalent.*

(1) *L is a regular language.*
(2) *L is the language accepted by a finite automaton over Σ_ϵ.*
(3) *L is the language accepted by a deterministic finite automaton.*
(4) *$L = f^{-1}(X)$ for some homomorphism $f : \Sigma^* \to M$ from Σ^* to a finite monoid M and some subset $X \subset M$.*

PROOF. To prove (1) implies (2) we argue by induction on regular languages. The languages ϵ, ϕ, and $w \neq \epsilon$ are accepted by automata constructed in Exercise 3.4. Every other regular language may be decomposed as $L = L_1 + L_2$, L_1L_2 or L_1^*, and by the induction hypothesis each language L_i is accepted by an automaton $\mathcal{A}_i$. If any $L_i = \phi$ for some i, there is nothing to prove. Thus we may assume each $\mathcal{A}_i$ satisfies the conditions of Lemma 3.8. If $L = L_1 + L_2$, identify the initial vertices of A_1 and A_2 to obtain an automaton accepting L. If $L = L_1L_2$, identify the initial vertex of $\mathcal{A}_2$ with the terminal vertex of A_1 and change the terminal vertex of A_1 into a nonterminal vertex. Finally if $L = L_1^*$, identify the terminal vertex of A_1 with its initial vertex.

Next we prove (2) implies (4). L be the language accepted by a finite automaton $\mathcal{A}$ over Σ_ϵ, and let M be the monoid of binary relations on the vertices of $\mathcal{A}$. The monoid operation is composition of relations. For each $w \in \Sigma^*$ define a binary relation $\sim_w$ on the vertices of $\mathcal{A}$ by $p \sim_w q$ if and only if there is a path from p to q with label w. We claim $\sim_{wv}$ equals the composite $\sim_w \circ \sim_v$. Indeed there is a path from p to q with label wv if and only if for some vertex r there is a path from p to r with label w and a path from r to q with label v. Thus the map $w \to\sim_w$ is a homomorphism from Σ^* to the finite monoid M. L is the inverse image of the set X of all relations $\sim$ such that $p_0 \sim p$ for the initial vertex p_0 and some terminal vertex p.

To show (4) implies (3) suppose that $\sigma : \Sigma^* \to M$ is a homomorphism to a finite monoid M, and $L = \sigma^{-1}(X)$ for some $X \subset M$. We may assume σ is onto. In other words σ is a choice of generators for M. Let Γ be the corresponding Cayley diagram of M, and make Γ into an automaton with initial vertex 1 and terminal vertices consisting of all elements of X. As a path from 1 with label w ends at $\sigma(w)$, we see that Γ accepts L.

Finally (3) implies (1) by induction on the number of edges of the accepting automaton $\mathcal{A}$. If there are no edges, the accepted language is either ϵ or ϕ depending on whether or not the initial vertex is terminal. Otherwise pick an edge e from vertex p to q with label $a \in \Sigma_\epsilon$ and consider four automata $\mathcal{A}_0, \dots, \mathcal{A}_3$ constructed by removing e from $\mathcal{A}$. $\mathcal{A}_0$ has the same initial and terminal vertices as $\mathcal{A}$. $\mathcal{A}_1$ has the same initial vertex as $\mathcal{A}$ and p as a terminal vertex. $\mathcal{A}_2$ has initial vertex q and terminal vertex p. $\mathcal{A}_3$ has initial vertex q and the same terminal vertices as $\mathcal{A}$. By induction these automata accept regular languages $R_0, \dots, R_3$.

Every successful path in $\mathcal{A}$ not containing e is a path in $\mathcal{A}_0$. Successful paths γ in $\mathcal{A}$ which do contain e factor into products $\gamma = \gamma_1 e \gamma_2 e \cdots \gamma_n$ where γ_1 is a successful path in $\mathcal{A}_1$, γ_n is a successful path in $\mathcal{A}_3$, and for $1 < i < n$, γ_i is a successful path in $\mathcal{A}_2$. Conversely all paths of these types are successful paths in $\mathcal{A}$. It follows that the language accepted by $\mathcal{A}$ can be described as $R_0 + R_1 a(R_2 a)^* R_3$. Since each of the R_i's is regular, $R_0 + R_1 a(R_2 a)^* R_3$ is regular too. □

COROLLARY 3.10 (Pumping Lemma). *For each regular language L there is an integer n with the following property. Every $w \in L$ of length at least n can be written as a product xyz with $y \neq \epsilon$ in such a way that $xy^*z \subset L$. Further if u is any subword of w of length n, we may take y to be a subword of u.*

PROOF. Let L be accepted by an automaton over Σ with n vertices. If w is the label of a successful path, then u must have a nontrivial subword which labels a cycle in that path. □

We already know that regular languages are closed under the rational operations. The next corollary gives some more closure properties. Closure under homomorphism means that the image of a regular language under a homomorphism from one finitely generated free monoid to another is regular in the target monoid, and likewise for closure under inverse homomorphism.

COROLLARY 3.11. *Regular languages are closed under intersection, complement, homomorphism, and inverse homomorphism.*

PROOF. The first two assertions together with the last one follow easily from Theorem 3.9(4). The remaining assertion may be proved in a straightforward way by induction on regular languages. Alternatively if $f : \Sigma^* \to \Delta^*$ is a homomorphism and L is the language accepted by an automaton $\mathcal{A}$ over Σ, replace edge labels of $\mathcal{A}$ by their images under f. Subdivide edges as necessary to get an automaton over Δ accepting $f(L)$. □

EXERCISE 3.12. The language $\{a^n b^n\}$ is not regular.

EXERCISE 3.13. Use the Pumping Lemma to show that word problems of infinite groups are not regular languages. Conclude that the word problem of a group is regular if and only if the group is finite.
Hint: Consider words ww^{-1} where w is the label of a geodsic path in the Cayley diagram.

EXERCISE 3.14. The language of all words in $(a + b + a^{-1} + b^{-1})^*$ freely equal to ϵ is not regular.

DEFINITION 3.15. For any language $L \subset \Sigma^*$ the syntactic congruence of L is defined by $w \sim v$ if $xwy \in L$ if and only if $xvy \in L$ for all $x, y \in \Sigma^*$. The quotient $\Sigma^*/\sim$ is the syntactic monoid of L.

EXERCISE 3.16. The syntactic congruence is a congruence.

EXERCISE 3.17. The syntactic monoid of L is finite if and only if L is regular.

EXERCISE 3.18. The syntactic monoid of a language L over an alphabet with formal inverses is a group if and only if L is closed under free equivalence.

3.2. Regular languages and groups. We continue with connections between regular languages and groups.

DEFINITION 3.19. A group G has a rational structure if some choice of generators $\Sigma^* \to G$ admits a regular combing with uniqueness. In other words a rational structure for G is a choice of generators together with a regular language which maps bijectively to G.

All automatic groups have rational structures.

EXERCISE 3.20. If G has a rational structure with respect to one choice of generators, then it does for all choices.
Hint: Let $\sigma : \Sigma^* \to G$ and $\tau : \Delta^* \to G$ be two choices of generators, and consider a homomorphism $f : \Sigma^* \to \Delta^*$ with $\tau \circ f = \sigma$.

EXERCISE 3.21. If N is normal in G, and both N and G/N have rational structures, then G has one too.

EXERCISE 3.22. The wreath product $Z \wr Z$ has a rational structure.

EXERCISE 3.23. An infinite torsion group does not have a rational structure.

Suppose $\Sigma^* \to G$ is a choice of generators and H is a finitely generated subgroup of G. We can pick words $w_1, \dots, w_n$ in Σ^* whose images generate H. It follows that H is the image of a regular language; the next theorem shows that the converse is true.

THEOREM 3.24. *Let $\sigma : \Sigma^* \to G$ be a choice of generators. A subgroup $H \subset G$ is finitely generated if and only if it is the image of a regular language over Σ.*

PROOF. It suffices to show that if $H = \sigma(R)$ for some regular language $R \subset \Sigma^*$, then H is finitely generated. Clearly we may assume R is nonempty. Hence R is accepted by an automaton $\mathcal{A}$ satisfying the conditions of Lemma 3.8.

Pick a spanning subtree $\mathcal{A}_0$ of $\mathcal{A}$ with root p_0, the initial vertex of $\mathcal{A}$, and with all edges directed away from the root. The label of any successful path in $\mathcal{A}$ is a product $w = w_0 v_1 w_1 v_2 w_2 \cdots w_{m-1} v_m w_m$ in which each w_i is the label of a path (possibly of length 0) in $\mathcal{A}_0$, and each v_i is the label of an edge e_i not in $\mathcal{A}_0$.

Let x_i and y_i be the labels of the paths in $\mathcal{A}_0$ from the root to the source and target vertex of e_i respectively, and let z be the label of the path in $\mathcal{A}_0$ from p_0 to the single terminal vertex p_t. Since there is at most one path in $\mathcal{A}_0$ between any two vertices, $x_1 = w_0$. Likewise $x_{i+1} = y_i w_i$ for $1 < i < m$, and $z = y_m w_m$. Thus w and $x_1 v_1 y_1^{-1} x_2 v_2 y_2^{-1} \cdots x_m v_m y_m^{-1} z$ have the same image in G. It follows that $\sigma(R)$ lies in the subgroup of G generated by $\sigma(z)$ together with one generator $\sigma(xvy^{-1})$ for each edge e not in $\mathcal{A}_0$. Here v is the label of e, and x and y are the labels of the paths in $\mathcal{A}_0$ to the source and target of e respectively.

To complete the proof it suffices to show that $\sigma(z)$ and each $\sigma(xvy^{-1})$ mentioned above lie in H. As z is the label of a successful path, $z \in R$ whence $\sigma(z) \in H$. Consider $\sigma(xvy^{-1})$. Let u be the label of a path in $\mathcal{A}$ from the target vertex of e to p_t. It follows from $xvu, yu \in R$ that $\sigma(xvy^{-1}) = \sigma(xvu)\sigma(yu)^{-1} \in H$. □

EXERCISE 3.25. If $\Sigma^* \to G$ is a choice of generators and R is regular over Σ, then the subgroup generated by $\overline{R}$ (the image of R in G) is finitely generated.

THEOREM 3.26. *Let $R \subset \Sigma^*$ be a language and L the language obtained by freely reducing all words in R. If R is regular, so is L.*

PROOF. Let $\mathcal{A}$ be an automaton accepting R. If for any $a \in \Sigma$ there is a path with label aa^{-1} (some edges of this path may have label ϵ) from vertex p to q, add a new edge with label ϵ from p to q if such an edge does not already exist. Continue until no more additions are possible. Observe that if $waa^{-1}v$ is the label of a successful path, then so is wv. Consequently the modified automaton accepts the language R' consisting of R and all words which may be derived from R by free reduction. As L is the intersection of R' with the regular language of freely reduced words (see Problem 3.2), it is regular. □

THEOREM 3.27. *The intersection of two finitely generated subgroups of a free group is finitely generated.*

PROOF. Let G be a finitely generated free group and $\Sigma \to G$ a choice of free generators. By Theorem 3.26 we may assume that for $i = 1, 2$ the finitely generated subgroup H_i is the image of the rational language L_i over Σ and L_i contains all the freely reduced words which represent elements of H_i. Hence $L_1 \cap L_2$ contains all freely reduced words projecting to $H_1 \cap H_2$. Consequently $L_1 \cap L_2$ projects onto $H_1 \cap H_2$, and $H_1 \cap H_2$ is finitely generated by Theorem 3.24. □

3.3. Notes. Regular languages play an important role in the theory of automatic groups [**15**] and make an appearance in the theory of word hyperbolic groups [**28**].

Problem 3.20 and the problems following it are from [**22**].

When the technique used in the proof of Theorem 3.24 is applied to finitely generated subgroups of free groups, it yields a set of free generators and hence a proof that finitely generated subgroups of free groups are free. For this and other applications of automata to free groups we refer the reader to [**35**], [**46**], and [**47**].

We have not touched on the well developed connections between regular languages and finite semigroups. See for example [**12, 17, 20, 36**].

4. Rational sets

As we observed previously, the definition of regular language in Section 3 makes sense for arbitrary monoids, not just finitely generated free ones. Since only subsets of finitely generated free monoids are languages, we use the term rational subset in this more general situation.

DEFINITION 4.1. The rational subsets of a monoid are the closure of its finite subsets under the rational operations.

Many properties of regular languages persist for rational subsets, and the proofs are often the same. Just as in the case of regular languages rational subsets are closed under rational operations. Likewise for any set of generators of a monoid M, the rational subsets of M consist of ϕ, $\{1\}$ (where 1 is the unit element of M) and the closure of the singleton subsets of the set of generators under the rational operations union, product, and generation of submonoid.

The next lemma is proved by induction on rational sets.

LEMMA 4.2. *If R is a rational subset of M, then R lies in a finitely generated submonoid of M.*

PROOF. Each finite subset of M lies in a finitely generated submonoid; and if the subsets R and S are contained in finitely generated submonoids, so are $R+S$, RS, and R^*. □

EXERCISE 4.3. If $f : M_1 \longrightarrow M_2$ is a homomorphism of monoids and R is a rational subset of M_1, then $f(R)$ is a rational subset of M_2.

Hint: Use induction on rational sets.

EXERCISE 4.4. If $f : M_1 \longrightarrow M_2$ and S is a rational subset of M_2 lying in $f(M_1)$, then $S = f(R)$ for some rational subset of M_1.

EXERCISE 4.5. Rational sets are not closed under inverse homomorphism.

Hint: Consider the word problem of an infinite group.

EXERCISE 4.6. Formulate a pumping lemma for rational sets.

EXERCISE 4.7. The intersection of the rational subsets

$$(a,c)^*(b,1)^* \text{ and } (a,1)^*(b,c)^*$$

of the monoid $M = \Sigma^* \times \Sigma^*$ is not rational.

Hint: Use the result of Exercise 4.6.

Observe that the monoid M in Exercise 4.7 is a submonoid of the direct product of two free groups. Thus rational subsets of groups are not closed under intersection. Since they are closed under union, they cannot be closed under complement either. On the other hand rational subsets of finitely generated free groups are closed under Boolean operations.

EXERCISE 4.8. Rational subsets of free groups are closed under intersection and complement.

Hint: Use Theorem 3.26 to show that every rational subset lifts to a regular language of freely reduced words.

4.1. Finite automata over monoids.

Finite automata over a monoid M are defined in just the same way as finite automata over an alphabet Σ except that since M need not have a preferred generating set, we allow the edge labels of automata to be arbitrary elements of M. When all the edge labels lie in a subset $X \subset M$, we say that the automaton is defined over X.

Automata over Σ^* in this sense are more general than the automata over Σ and Σ_ϵ defined in Section 3, but by Theorem 4.9 they accept the same languages.

THEOREM 4.9. *Let R be a subset of a monoid M. The following are equivalent.*

(1) *R is a rational subset of M.*
(2) *For any generating set $X \subset M$, R is accepted by a finite automaton over X.*

PROOF. The proof rests on the following observation. Let $f : M_1 \to M_2$ be a homomorphism of monoids, and suppose the automaton $\mathcal{A}_1$ over M_1 accepts $R \subset M_1$. If $\mathcal{A}_2$ is constructed by replacing the edge labels of $\mathcal{A}_1$ with their images in M_2, then $\mathcal{A}_2$ accepts $f(R)$.

Suppose R is a rational subset of M. By Theorem 4.2 R is in the submonoid generated by a finite subset $X_0 \subset X$. Choose an alphabet Σ in one to one correspondence with X_0, and let $f : \Sigma^* \to M$ be the induced homomorphism. By Exercise 4.4 $R = f(L)$ for some regular language $L \subset \Sigma^*$. Theorem 3.9 guarantees that L is accepted by an automaton with edge labels from Σ. Replacing these edge labels by their images yields an automaton over X_0 accepting R.

For the converse suppose R is accepted by an automaton $\mathcal{A}$ over X. Let X_0 be the finite set of elements of X occurring as edge labels in $\mathcal{A}$. Choose an alphabet Σ in one to one correspondence with X_0, and let $f : \Sigma^* \to M$ be the induced homomorphism. Define $\tilde{\mathcal{A}}$ to be $\mathcal{A}$ with its edge labels replaced by corresponding elements of Σ. $\tilde{\mathcal{A}}$ accepts a rational language L such that $f(L) = R$. Thus R is rational by Exercise 4.3. □

4.2. Rational Relations.

We will use rational relations in the definitions of context free and indexed languages. A rational relation is a rational subset of the direct product of two monoids. We write $\rho : M_1 \to M_2$ as well as $\rho \subset M_1 \times M_2$. The image of $S \subset M_1$ under ρ is $\rho(S) = \{m \in M_2 \mid \exists s \in S, (s, m) \in \rho\}$. Singleton sets are identified with elements; that is, if $S = \{m\}$, we write $\rho(m)$ instead of $\rho(\{m\})$; and likewise $\rho(S) = m'$ if $\rho(S) = \{m'\}$. The domain of ρ is set of elements $m \in M$ such that $\rho(m)$ is nonempty, and the image is $\rho(M)$.

For example if M is a finitely generated monoid, then a monoid homomorphism $f : M \to M'$ (or more precisely its graph) is a rational relation. Indeed if $X \subset M$ is a finite set of generators, then $f = (\sum_X (x, f(x)))^*$ is a rational relation. The comparator automata involved in the definition of asynchronous automatic groups are finite automata accepting certain rational relations. The comparator automata for synchronous automatic groups accept regular languages which encode rational relations.

THEOREM 4.10. *If $\rho : M \to \Sigma^*$ and $\rho' : \Sigma^* \to M'$ are rational relations, so is the composition $\rho' \circ \rho$.*

PROOF. By Theorem 4.9 ρ is accepted by an automaton $\mathcal{A}$ with edge labels from $M \times \Sigma_\epsilon$, and ρ' is accepted by $\mathcal{A}'$ with edge labels from $\Sigma_\epsilon \times M'$. Add edges as necessary so that each vertex of $\mathcal{A}$ has a loop, i.e., an edge from the vertex to itself, with label $(1, \epsilon)$ and each vertex of $\mathcal{A}'$ has a loop with label $(\epsilon, 1)$. Clearly the sets accepted by the automata are unchanged.

The point of these alterations is the following. Suppose (m, v) is the label of a path γ from p to q in $\mathcal{A}$. Then the sequence of edge labels in the path will be $(m_1, a_1), \ldots, (m_k, a_k)$ for some $m_i \in M$ and $a_i \in \Sigma_\epsilon$ with $m_1 \cdots m_k = m$ and $a_1 \cdots a_k = v$. The loops we have added allow us to interpolate ϵ's in the sequence $a_1 \cdots a_k = v$ without changing p, q or the path label (m, v). A similar condition holds for $\mathcal{A}'$. Thus if there is a path γ from p to q in $\mathcal{A}$ with label (m, v) and a path γ' from p' to q' in $\mathcal{A}'$ with label (v, m'), we may take the edge labels of these paths to be $(m_1, a_1), \ldots, (m_k, a_k)$ and $(a_1, m'_1), \ldots, (a_k, m'_k)$, so that v is expressed as a product of elements of Σ_ϵ in exactly the same way along both paths.

We claim that the automaton $\mathcal{B}$ over $M \times M'$ defined as follows accepts $\rho \circ \rho'$.

(1) The vertices of $\mathcal{B}$ are pairs (p, p') where p is a vertex of $\mathcal{A}$ and p' is a vertex of $\mathcal{A}'$.
(2) There is an edge $(p, p') \xrightarrow{(m,m')} (q, q')$ if and only if for some $a \in \Sigma_\epsilon$ there are edges $p \xrightarrow{(m,a)} p'$ and $q \xrightarrow{(a,m')} q'$.
(3) The initial vertex is (p_0, p'_0) where p_0 and p'_0 are initial vertices.
(4) (p, p') is a terminal vertex if p and p' are.

It is straightforward to show by induction on path length that there is a path in $\mathcal{B}$ from (p, p') to (q, q') with label (m, m') if and only if the following conditions hold.

(1) For some word $v \in \Sigma^*$ there is a path in $\mathcal{A}$ from p to q with label (m, v) and a path in $\mathcal{A}'$ from p' to q' with label (v, m').
(2) Along both paths v is expressed as a product of elements of Σ_ϵ in exactly the same way.

It follows from these conditions that if $\mathcal{B}$ accepts (u, v), then $(u, v) \in \rho \circ \rho'$. Similarly the converse follows from the conditions together with the remarks above about the addition of extra edges. □

COROLLARY 4.11. *The image of a regular language $L \subset \Sigma^*$ under a rational relation $\rho : \Sigma^* \to M$ is a rational subset of M.*

PROOF. Define $\rho_L : \Sigma^* \to \Sigma^*$ to be the restriction of the identity map to L. If $\mathcal{A}$ is an automaton over Σ^* accepting L, then replacing each edge label a by (a, a) yields an automaton accepting ρ_L. Thus ρ_L is a rational relation. As the projection of $\rho_L \circ \rho$ from $\Sigma^* \times M$ to M is $\rho(L)$, it follows from Lemma 4.4 and Theorem 4.10 that $\rho(L)$ is rational. □

Rational binary relations between finitely generated free monoids have a special name; they are called rational transductions. An automaton accepting a rational transduction is a transducer.

EXERCISE 4.12. Homomorphisms and inverse homomorphisms are rational transductions. (An inverse homomorphism $\rho : \Sigma^* \to \Delta^*$ is defined by $\rho(x) = g^{-1}(x)$ for some homomorphism $g : \Delta^* \to \Sigma^*$.)