# Algorithms in Geometric Group Theory

by

Zeph Grunschlag

A.B. (Princeton University) 1992

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Mathematics

in the

GRADUATE DIVISION
of the
UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Professor John R. Stallings, Chair
Professor Andrew J. Casson
Professor Martin B. Halpern

Spring 1999

The dissertation of Zeph Grunschlag is approved:

Chair Date

Date

Date

University of California at Berkeley

Spring 1999

**Algorithms in Geometric Group Theory**

## Abstract


Algorithms in Geometric Group Theory

by

Zeph Grunschlag

Doctor of Philosophy in Mathematics

University of California at Berkeley

Professor John R. Stallings, Chair

This dissertation* explores algorithmic problems and properties occurring in geometric group theory. Special attention is paid to connections with formal language theory. Some of the results obtained are as follows:

We investigate the change in complexity of certain algorithms under passing to finite extensions. For example, it is shown that a polynomial time solution for the generalized word problem gives rise to a polynomial time solution in any finite extension. A purely language theoretic definition of word hyperbolicity is given: a group is hyperbolic if and only if its word problem is growing and terminating. An automata theoretic construction called the "balloon construction" is introduced. The balloon construction yields a method for computing the angle between quasiconvex subgroups whose join is quasiconvex, inside hyperbolic groups. We define a group theoretic property called "super local quasiconvexity" and use the balloon construction to show that virtually free groups satisfy it. It is also shown that if $S$ is a rational subset of a finitely generated virtually free group or virtually abelian group then $S$ satisfies: the complement of $S$ is rational, the membership problem of $S$ is solvable, and $S$ is unambiguously rational.

Professor John R. Stallings
Dissertation Committee Chair


*Support of NSF grant DMS-9803316 is gratefully acknowledged.

*To Leo Adiel Grunschlag, aka Joony.*

*Born May 9, 1999 at 9:34 pm.*

# Contents

# List of Figures

# List of Tables

# List of Symbols

| Symbol | Meaning | Defined |
|---|---|---|
| $M$ | A monoid | p. 1 |
| $X^*$ | The free monoid on $X$ | p. 1 |
| $\varepsilon$ | The empty string and unit of a free monoid | p. 1 |
| $\|w\|$ | The length of the word $w \in X^*$ | p. 1 |
| $\mathcal{A}, \mathcal{B}, \mathcal{C} \ldots$ | Languages, i.e. subsets of $X^*$ (often identified with automaton or grammar which generates language) | 1,9,29 |
| $G$ | A group | p. 2 |
| $Y$ | A set of group generators for $G$ | p. 2 |
| $X$ | Monoid generators for $M$ (when $M = G$, $X$ is *usually* the symmetric set $X = Y \cup Y^{-1}$). Also the terminal letters in a grammar. | 2, 28 |
| $\sigma$ | The generating map $X^* \twoheadrightarrow G$ (or $X^* \twoheadrightarrow M$) | p. 2 |
| $X^+$ | The free semigroup on $X$ ($X^+ = X^* - \{\varepsilon\}$) | p. 2 |
| $ST$ | Set product $\{ st \mid s \in S \text{ and } t \in T \}$ inside a monoid | p. 2 |
| $S^*$ | The submonoid generated by $S$ | p. 2 |
| $S^+$ | The sub-semigroup generated by $S$ | p. 2 |
| $u =_G v$ | The words $u$ and $v$ have the same image in $G$ | p. 2 |
| $R$ | A set of relators for $G$ | p. 2 |
| $\langle\langle R \rangle\rangle$ | Normal subgroup generated by $R$ | p. 2 |
| $P = \langle Y \| R \rangle$ | A presentation for $G$ | p. 2 |
| $[a, b]$ | The commutator $aba^{-1}b^{-1}$ (when inverse operation well-defined) | p. 2 |
| $S^{-1}$ | The set of inverses of elements in $S$ | p. 2 |
| $\langle S \rangle$ | The subgroup generated by $S$ | p. 2 |
| $A \vee B$ | The join of the subgroups $A$ and $B$, i.e., $\langle A \cup B \rangle$ | p. 3 |
| $H < G$ | $H$ is a subgroup of $G$ | p. 3 |
| $H \underset{n}{<} G$ | $H$ is a subgroup of $G$ with index $[G : H] = n$ | p. 3 |
| $H \underset{\text{f.i.}}{<} G$ | $H$ is a finite index subgroup of $G$ | p. 3 |
| $N \lhd G$ | $N$ is a normal subgroup of $G$ | p. 3 |

| | | |
|---|---|---|
| $N \underset{n}{\lhd} G$ | $N$ is a normal subgroup of of index $n$ in $G$ | p. 3 |
| $N \underset{\text{f.i.}}{\lhd} G$ | $N$ is a normal finite index subgroup of $G$ | p. 3 |
| $H\backslash G$ | The set of right $H$-cosets (*not* boolean difference) | p. 3 |
| $\overline{S}$ | The profinite closure of $S$ | p. 3 |
| $r$ | A relation (see §1.1.4 for many more relation conventions) | p. 3 |
| $\lceil x \rceil$ | Ceiling: smallest integer greater than or equal to $x$ | p. 6 |
| $\mathbb{N}$ | The set of numbers $\{1, 2, 3, 4, 5, \dots\}$ | p. 6 |
| $2^S$ | The power set of $S$ | p. 7 |
| $f \preceq g$ | $\exists\ a, b, c, d \in \mathbb{R}$ s.t. $f(n) \leq ag(cn + d) + b$ for all $n \in \mathbb{N}$ | p. 7 |
| $f = O(g)$ | same as: $f \preceq g$ | p. 7 |
| $f \simeq g$ | $f \preceq g$ and $g \preceq f$ | p. 7 |
| $f \prec g$ | $f \preceq g$ and $g \not\preceq f$ | p. 7 |
| $\mathsf{P} \leq f$ | $f$ is a time complexity function for problem $\mathsf{P}$ | p. 8 |
| $\mathsf{P} \preceq f$ | $f$ is equivalent to some complexity function for $\mathsf{P}$ | p. 8 |
| $\mathsf{P} \prec f$ | There is a $g$ such that $\mathsf{P} \preceq g$ and $g \prec f$ | p. 8 |
| $V(\mathcal{A})$ | Vertices of the automaton $\mathcal{A}$ | p. 9 |
| $v \xrightarrow{x} w$ | An edge from $v$ to $w$ labeled by $x$ | p. 9 |
| $\iota(\gamma)$ | Initial vertex of the path $\gamma$ | p. 10 |
| $\tau(\gamma)$ | Terminal vertex of $\gamma$ | p. 10 |
| $\lambda(\gamma)$ | Label of $\gamma$ in $X^*$ | p. 10 |
| $\mathcal{A}\mathcal{B}$ | Concatenation of $\mathcal{A}$ and $\mathcal{B}$ (set product inside free monoid) | p. 11 |
| $\mathcal{A} \bigcap \mathcal{B}$ | Pull-back of $\mathcal{A}$ and $\mathcal{B}$ (set intersection inside free monoid) | p. 11 |
| $\lvert\mathcal{A}\rvert$ | The number of edges in the automaton $\mathcal{A}$ | p. 13 |
| $\mathrm{FG}(Y)$ | The free group on the generators $Y$ | p. 13 |
| $[w]$ | A geodesic word for $w$, (in $\mathrm{FG}(Y)$, the free reduction) | 13, 89 |
| $[\alpha]$ | Geodesic representative of $\alpha$ (in $\mathrm{FG}(Y)$, the free reduction) | 13, 89 |
| $\widetilde{\mathcal{A}}$ | The free reduction of the automaton $\mathcal{A}$ | p. 15 |
| $\mathrm{Rec}(M)$ | Recognizable subsets of a monoid $M$ | p. 19 |
| $\Gamma_X(H\backslash G)$ | The Cayley graph of the right cosets of $H$ in $G$ | p. 20 |

| | | |
|---|---|---|
| $\Gamma_X(G)$ | The Cayley graph of $G$ —same as $\Gamma_X(\{1\}\backslash G)$ | p. 20 |
| $\|\|\Gamma\|\|$ | The geometric realization of the directed labeled graph $\Gamma$ | p. 21 |
| $\mathrm{Rat}(M)$ | Rational subsets of a monoid $M$ | p. 22 |
| $S \Cup T$ | Disjoint union | p. 23 |
| $S \odot T$ | Unambiguous product: same as $ST$ with condition that no product can be written in two distinct ways | p. 23 |
| $S^{\circledast}$ | Unambiguous monoid closure: same as $S^*$ with condition that $S$ is a basis of a free submonoid | p. 23 |
| $\mathcal{W}(w)$ | The word automaton for the word $w$ | p. 24 |
| $\mathfrak{B}$ | The set of all bouquet automata | p. 25 |
| $\mathcal{L}$-Reg$(M)$ | $\mathcal{L}$-regular subsets of a monoid $M$ | p. 26 |
| $u \rightrightarrows v$ | $u$ produces $v$ in the grammar "$\rightrightarrows$" | p. 28 |
| $u \rightsquigarrow^* v$ | $v$ is derived from $u$ | p. 28 |
| $\sphericalangle(A, B; C)$ | Generalized angle between subgroups $A, B$ over $C < A \cap B$ | p. 34 |
| $\sphericalangle_B^A$ | Standard angle. Equal to $\sphericalangle(A, B; A \cap B)$ | p. 34 |
| $\mathrm{Rubik}_3$ | The group of Rubik's $3 \times 3 \times 3$ cube | p. 38 |
| $\mathbb{B}$ | The Boolean set $\{0, 1\}$ | p. 39 |
| $\mathsf{P} \preceq_{\mathrm{T}} \mathsf{Q}$ | Problem $\mathsf{P}$ is Turing reducible to problem $\mathsf{Q}$ | p. 41 |
| $\mathsf{P} \approx_{\mathrm{T}} \mathsf{Q}$ | $\mathsf{P} \preceq_{\mathrm{T}} \mathsf{Q}$ and $\mathsf{Q} \preceq_{\mathrm{T}} \mathsf{P}$ | p. 41 |
| $\mathsf{P} \preceq \mathsf{Q}$ | $\mathsf{P} \preceq_{\mathrm{T}} \mathsf{Q}$ and $\mathsf{P}$ has complexity not worse than $\mathsf{Q}$ | p. 42 |
| $\mathsf{P} \eqsim \mathsf{Q}$ | $\mathsf{P} \preceq \mathsf{Q}$ and $\mathsf{Q} \preceq \mathsf{P}$ | p. 42 |
| $\overline{\mathcal{A}}$ | Profinite closure of an automaton | p. 44 |
| $\mathcal{G}$ | The language of all reduced words in $\mathrm{FG}(Y)$ | p. 50 |
| $\mathcal{T}$ | Maximal tree in the geometric realization of a graph | p. 58 |
| $\mathcal{C}$ | Set of chords: the directed edges not in $\mathcal{T}$ | p. 58 |
| $[\Gamma]_v$ | Connected component of $v$ in $\Gamma$ | p. 60 |
| $\|\gamma\|$ | Length of a path $\gamma$ inside a graph | p. 70 |
| $d_X(g, h)$ | Distance between $g, h \in G$ inside of $\Gamma_X(G)$ | p. 70 |
| $[p, q]$ | A geodesic between $p$ and $q$ in the metric space $\Gamma$ | p. 71 |

| | | |
|---|---|---|
| $B_K$ | Ball of radius $K$ about 1 in $\Gamma_X(G)$ | p. 71 |
| $\Omega_K$ | Set of all loops $\gamma$ based at $*$ s.t. $\vert\gamma\vert \leq K$ | p. 74 |
| $\Sigma$ | A 2-complex | p. 75 |
| $\Sigma_P(G)$ | The Cayley 2-complex of $G$ for the presentation $P$ | p. 75 |
| $f_\Sigma$ | The isoperimetric function of $\Sigma$ | p. 75 |
| $f_P$ | The isoperimetric function of a group with presentation $P$ | p. 76 |
| $u^v$ | Conjugation: $u^v = v^{-1}uv$ | p. 82 |
| $\mathrm{QC}(G)$ | The quasiconvex subsets of $G$ | p. 85 |
| $G_{\mathsf{Word}}$ | A finitely presented torsion free group with unsolvable word problem | p. 87 |
| $G_{\mathrm{R}}$ | Hyperbolic group with normal rank-2 subgroup $N$ s.t. $G_{\mathrm{R}}/N \approx G_{\mathsf{Word}}$ | p. 87 |
| $\gamma(w)$ | The path in $\Gamma_X(G)$ which starts at 1 and is labeled by $w$ | p. 89 |
| $\mathcal{G}_X$ | The language in $X^*$ of all geodesic words in $G$ | p. 90 |
| $B_K(\mathcal{A})$ | The $K$-thickening of the automaton $\mathcal{A}$ | p. 104 |

Table 0.1: Symbol translator

# List of Algorithmic Problems

| Problem | Input | Compute | Defined |
|---|---|---|---|
| $\mathsf{Angle}(G)$ | F.g. subgroups $A, B$ of $G$ | $\sphericalangle_B^A$ | p. 39 |
| $\mathsf{Angle}'(G)$ | F.g. subgroups $A, B, C$ of $G$ | $\sphericalangle(A, B; C)$ | p. 40 |
| $\mathsf{RatClos}(G)$ | A rational subset $A \subseteq G$ | $\overline{A}$ the profinite closure | p. 44 |
| $\mathsf{RatComp}(G)$ | A rational subset $A \subseteq G$ | $G - A$ | p. 46 |
| $\mathsf{SubgpTrans}(G, H; \mathcal{U})$ | A word $w$ in $G$ | Word $w'$ in $HU$ s.t. $w =_G w'$ | p. 58 |

Table 0.2: Computational problems

| Problem | Input | Decide if... | Defined |
|---|---|---|---|
| $\mathsf{Word}(G)$ | A word $w$ on the generators of $G$ | $w = 1$ in $G$ | p. 37 |
| $\mathsf{GenWord}(G)$ | A word $w$ and subgroup $A$ of $G$ | $w \in A$ in $G$ | p. 40 |
| $\mathsf{GenWord}_m(G)$ | A word $w$ and subgroup $A$ of $G$ with $\mathrm{rk}(A) \leq m$ | $w \in A$ in $G$ | p. 64 |
| $\mathsf{PosAngle}(G)$ | Finitely generated subgroups $A, B$ of $G$ | $\sphericalangle_B^A > 0$ | p. 41 |
| $\mathsf{RatUnity}(G)$ | A rational subset $A$ of $G$ | $1 \in A$ | p. 43 |
| $\mathsf{RatMemb}(G)$ | A rational subset $A \subseteq G$ and a word $w$ in $G$ | $w \in A$ | p. 43 |
| $\mathsf{RatInt}(G)$ | Rational subsets $A$ and $B$ of $G$ | $A \cap B = \emptyset$ | p. 43 |
| $\mathsf{RatInc}(G)$ | Rational subsets $A$ and $B$ of $G$ | $A \supseteq B$ | p. 44 |

Table 0.3: Decision problems

# Preface

## Motivation

Gersten and Stallings invented the angle between subgroups in [51]. Later, Stallings [52] investigated the special case of free groups, gave a geometric criterion for computing the angle in certain cases, and asked how one could generally compute the angle between finitely generated subgroups of a free group. Stallings used ideas of R. Gilman [23] to give a partial algorithm. In my first journal article [30] it was shown how to decide if the angle between two subgroups of a free group is zero, completing Stallings' partial algorithm by using a topological approach. My method basically boiled down to the following: Given any subgroup of a free group, there is a way of writing down a presentation for the subgroup in terms of the original generators. There are many ways of proving this, but the method that I figured out was a slight strengthening of a graph theoretic algorithm of Stallings for finding a free basis for a subgroup of a free group [50].

I then set about computing angles in a larger class of groups. I also wanted to refine Stallings' algorithm so that it could not only decide if the angle is zero, but be made to compute positive angles as well. Though I saw a way of generalizing Stallings' algorithm for surface groups, and thus could decide whether the angle was zero in such groups, I could not see a way of achieving the second goal. Over the course of a few months I began to realize that topological methods were too coarse. For example, in computing positive angles one encounters the problem of deciding membership in products of subgroups (such as double cosets): Deciding membership in a single subgroup corresponds to deciding whether a closed path lifts to a closed path in the covering space for the subgroup; unfortunately, I could not find any topological object associated with a double coset! Upon reading Gilman's paper [23] more closely, I started understanding that Gilman's work already contained the generalization of covering spaces ideal for algorithmic purposes: the automaton. In a topological graph, any edge that is traversed may be traversed in reverse. However, in an automaton edges are one-way so that no edges may be reversed. Nevertheless, one can also allow edges to have dual edges which give rise to reverse paths. In this way, the directed graph generalizes the topological graph. This approach had already been used by Serre [48], Stallings [50] and others. But all these approaches viewed directed graphs as convenient combinatorial objects for collecting information about group actions, subgroups, etc. Gilman's approach allowed the consideration of more general types of objects by dropping the insistence that edges should have dual inverses. So representing two subgroups $H$ and $K$ in a topological manner[1] one obtains a representation for the double coset $HK$ by taking the $H$-representation and attaching an "epsilon edge" from its base-point, to the base-point of the $K$-representation. This epsilon edge has unit image in the group, and is irreversible. Thus a path in the new object is composed by a path in the $H$-space, followed by the epsilon edge, followed by a path in the $K$-space, and represents a product element $hk$.

Therefore, an important tool in computing angles is the ability to deal with subsets in a group which are represented by finite state automata. These subsets are called "rational" and decision problems dealing with such sets are called "rational problems". The majority of this dissertation is concerned with solving rational problems. On the other hand, rational subsets are the images of rational languages. Thus rational problems connect one aspect of formal languages with an aspect of group theory. This dissertation also mentions another connection between these two fields. One can view the word problem as the formal language consisting of all words with trivial image in $G$. One can ask the following question: what can be said about the type of language that the word problem for $G$ is, based on the type of group that $G$ is? The most celebrated result of this nature is Muller and Schupp's theorem that a group is virtually free if and only if the word problem is context free [43]. A class of formal languages called "growing and terminating" is defined below, which characterizes the class of word hyperbolic groups.

[1] I.e., the paths in the space are identified with certain group elements.

# Outline

This dissertation is organized as follows: the front matter contains

- a table of contents,
- lists of figures, tables, symbols, and algorithmic problems,
- and this preface with acknowledgments.

Chapter 1 describes notation and concepts necessary for the results of the last two chapters. Chapter 2 explains my conventions for dealing with algorithms in finitely presented groups, and contains some results concerning the stability of recursive solutions under passing to finite extensions. Chapter 3 deals with word hyperbolic groups and describes ways of solving certain rational problems in such groups, as well as other connections with formal languages.

The bibliography and index are included at the end of this dissertation.

## Chapter 1

Most of the results described in chapter 1 are well known. The chapter starts with section 1.1 which is a collection of concepts and notations concerning groups, monoids and relations. Section 1.2 introduces complexity functions together with a partial order which induces an ordering of algorithmic problems. Finite state automata are delineated in section 1.3 and some automata theoretic operations are described. In particular, the pull-back construction is explained in 1.3.3 and Gilman's algorithm for reducing automata over free groups is delved into in 1.3.4. The languages defined by finite state automata are the regular languages, and these are looked at more closely in section 1.4. Regular expressions and operations are defined in 1.4.1 while in 1.4.2 Kleene's theorem is mentioned which gives several equivalent ways of defining regular languages. One of the formulations in Kleene's theorem is that of recognizability. This formulation is generalized in several standard ways in section 1.5 from free monoids, to all monoids. In 1.5.3 the Cayley diagram of a group relative a subgroup and a set of generators is recalled. When the Cayley diagram is finite, it defines a finite state automaton, showing that cosets of finite index subgroups are recognizable. The rational subsets generalize another formulation in Kleene's theorem from free monoids, to all monoids; these subsets are defined by rational operations and are discussed in section 1.6.

Unambiguously rational sets were introduced in [16] and are defined in 1.6.2. These are rational subsets obtained by unambiguous operations; for example, the union of two subsets is unambiguous if and only if the union is disjoint. The relationship between rationality, recognizability, and finitely generated subgroups is explained in 1.6.3 and 1.6.4. The notion of $\mathcal{L}$-regularity was introduced in [21]. It generalizes rationality and recognizability at the same time, and is explained in section 1.7. Grammars are recalled in section 1.8 and are used to define some classes of formal languages. In addition to defining context free and context sensitive languages, we introduce the growing and terminating languages in 1.8.5 and show how they generalize context free languages in 1.8.6. The chapter concludes with section 1.9 which describes the group theoretic angle and its relationship with rational subsets.

### Chapter 2

Chapter 2 begins with a description of group theoretic algorithmic problems. In particular, the word problem, the generalized word problem, the membership problem in rational subsets, and the profinite closure problem for rational subsets are reviewed in section 2.1. In 2.1.4 it is shown that (not surprisingly) the word problem as well as other rational decision problems are not solvable in sub-linear time for groups which are infinite; this is useful in showing, for example, that the complexity of the word problem is independent of the finite set of generators chosen. Conditions are given for computability of the group theoretic angle in 2.1.5. In section 2.2 we organize some old results concerning abelian monoids, free monoids and free groups. In particular it is recalled that the rational subsets of finitely generated such monoids satisfy the following three properties:

- closed under complements,
- membership is decidable,
- are unambiguous.

In section 2.3 we show that solutions to many of the problems introduced at the beginning of the chapter can be extended from a finite index subgroup to the overgroup, and we analyze complexity: Let $H$ be a finite index subgroup of an infinite finitely generated group $G$. It is well known that a solution for the word problem or generalized word problem for $H$ extends to a solution for $G$, but the analysis of complexity is new: indeed, in 2.3.1 we show that the complexity of the word problem for $H$ is the same as that for $G$, and that if $H$ admits a polynomial time solution to its generalized word problem, so does $G$. It is also demonstrated that a solution to the membership problem in rational subsets of $H$ extends to one for $G$, though I have not been able to give a good relationship for the relative complexities. In 2.3.2 we apply the results of 2.3.1 and 2.1.5 to show that angles are computable in virtually free and virtually abelian groups. The property that all rational subsets are unambiguous is shown to extend from $H$ to $G$ in 2.3.3 which implies that virtually free and virtually abelian groups possess this property by the results of section 2.2. The chapter concludes with 2.3.4 by showing that the computation of closures of rational subsets extends from $H$ to $G$ so that by results of Steinberg ([53], [54]) for abelian and free groups, the rational closure problem is solvable in virtually abelian and virtually free groups.

## Chapter 3

Word hyperbolic groups are the focus of the final chapter. In section 3.1 four classical definitions of hyperbolicity of groups are reviewed. The slim and thin triangles criteria are described in 3.1.1 and 3.1.2 and are needed in sections 3.3, 3.4, and 3.6. In 3.1.3 the solution of the word problem by means of Dehn's algorithm is described graph theoretically as the "Dehn property" while in 3.1.4 the linear isoperimetric property is reviewed. These last two notions of hyperbolicity are necessary in the proofs of the next section. In section 3.2 a new language theoretic notion is shown to be equivalent to the classical hyperbolicity criteria. More precisely, it is shown that if a group has word problem which is growing and terminating (as in 1.8.5) then the group is finitely presented with linear isoperimetric inequality, and thus is hyperbolic; on the other hand, if a group is hyperbolic, its word problem is solvable by Dehn's algorithm and this defines a growing and terminating grammar for the word problem.

In the rest of the chapter we examine the structure of certain subsets of hyperbolic groups. Given a hyperbolic group $G$, three collections of subsets are naturally definable. The first two classes —rational subsets and quasiconvex subsets— are classical. Members of the third class —geodesically regular subsets— have previously been called "rational" [21]; to avoid ambiguity I have followed [17, pp. 173-174] in reserving the term "regular" for Gersten and Short's newer concept while adhering to the older use of "rational" (see for example [15]). The notion of quasiconvexity is defined in section 3.3. Quasiconvexity is a notion depending on the set of group generators chosen; however, for hyperbolic groups the notion is independent of generators. The geodesically regular subsets are precisely the $\mathcal{L}$-regular subsets where $\mathcal{L}$ is the language of all geodesic words; these are studied in section 3.4. In section 3.5 the group theoretic angle between subgroups of hyperbolic groups is examined. An example of a hyperbolic group in which angles are incomputable is given in 3.5.2. The main positive result concerning computability of angles is stated in 3.5.3:

**Theorem 3.5.10** *Let $G$ be a word hyperbolic group, let $a_1, \ldots, a_m, b_1, \ldots, b_n$ be words in $G$, and let $A$ (resp. $B$) be the subgroup of $G$ generated by $\{a_1, \ldots, a_m\}$ (resp. $\{b_1, \ldots, b_n\}$). Suppose that $A$, $B$, and $A \vee B$ are quasiconvex subgroups of $G$. Then the angle between $A$ and $B$ is computable.*

Theorem 3.5.10 is proved in 3.5.3 modulo deciding if the angle is zero and the balloon construction. In 3.5.4 all attention is focused on the zero angle case, and it is shown how to decide if the angle is zero when the subgroups and their join are quasiconvex. The balloon construction which generalizes Gilman's algorithm 1.3.7 is introduced in 3.6. This construction is needed in showing that geodesically regular subsets of hyperbolic groups are independent of generating sets, and in the results of sections 3.5 and 3.7. Section 3.7 concludes the dissertation by showing that virtually free groups satisfy a certain property stronger than local quasiconvexity: namely that all rational subsets are geodesically regular. The reader is asked to supply other examples of super locally quasiconvex groups as I presently have no further examples.

# Technical Note

This version is a slight revision of the dissertation which was officially filed to U.C. Berkeley's Graduate Division on May 19, 1999.

I have attempted to aid the reader in several ways:

## Lists of figures, tables, symbols and algorithms

A table of contents, as well as lists of figures, tables, symbols and algorithms are included in the front matter. The list of symbols contains all the symbols used in non-conventional ways, such as "$\rightsquigarrow$", as well as letters used in a consistent manner throughout the text, such as "$G$". These are listed in order of appearance and references to pages of definition are given. I have used non-standard notation for the algorithms discussed in the dissertation; this notation is summarized in the tables of algorithmic problems.

## Index

At the end of the thesis, there is an index. I have tried to include references to pages where a concept is defined, introduced, or discussed in detail. Page numbers which are in bold type refer to a figure or table which illustrates the concept.

## Questions

There are several questions which I have not been able to resolve. Some are nit-picky problems that are likely to be trivial —with the correct perspective. Some seem to be of a deeper nature. The simplest way to find my questions is by looking up the "questions" entry in the index.

*U.C. Berkeley* Zeph Grunschlag

*June 1999*

## Acknowledgements

I would like to thank the following people for helping me bring this dissertation to life.

- Dani Wise —my first research achievement, that of deciding if the angle is zero inside free groups, was a problem of Stallings that Dani introduced me to. Dani had his own solution to the problem but generously let me claim the result when I found a different solution. Dani also guided my research during 1996 and 1997.

- Bob Gilman —through email conversations and his published work, Bob taught me the connections between formal languages and group theory. Most of the results in this dissertation directly benefitted from his insights. For example, I could never have given a language theoretic characterization of hyperbolic groups if Bob hadn't told me that hyperbolic groups have growing word problem.

- Stuart Margolis —during his visit in Berkeley in the fall of 1998 Stuart helped me understand the greater possibilities that continued research along the lines of this dissertation might lead to. This helped invigorate my research and give me renewed confidence. Though the developments that he helped fuel are not in a complete enough form to be included in this dissertation, the structure of the dissertation benefitted greatly from this research.

- Roger Alperin —conversations with Roger, especially during an August 1998 conference in Korea, lead to much improved results. Results that I could previously state only for virtually free groups, are now included as properties preserved under finite extension, thanks to his questions about computing angles in virtually abelian groups.

- John Stallings —his extraordinary patience, support, encouragement, and mastery of geometric group theory helped make my struggle through graduate school as pleasant as possible. I can't imagine completing my PhD. without John's tremendous help.

I would also like to thank John Stallings, Roger Alperin and Andrew Casson for reading the first draft of this dissertation. Final thanks to Paul Brown, whose donation made it possible to draw the figures below.

This page is needed to get the odds vs. evens margins right on double sided printings.

# Chapter 1

# Preliminaries

## 1.1 Some conventions and notation

This section delineates some of the notation used throughout this dissertation.

### 1.1.1 Monoids

For completeness let's start by defining the notion of a monoid.

**Definition 1.1.1** *A* **monoid** $M$ *is a set endowed with an associative product "·" and with a unit element "1".*

Let $X$ denote a finite set, called an **alphabet**. Then the first example of a monoid is $M = X^*$, the **free monoid** on $X$. The free monoid is the set of **strings** (or **words**) over $X$ endowed with concatenation as its product, and with the empty string $\varepsilon$ as its unit. For example, if $X = \{x, y\}$ then two elements of $X^*$ are $xyyxxy$ and $yyxxy$ and their product from left to right produces the string $xyyxxyyyxxy$. A **language** is any subset of $X^*$. Therefore, languages are just sets of strings. Languages will usually be denoted by calligraphic capitals such as $\mathcal{A}, \mathcal{B}, \mathcal{C}$ etc. Given any string $w$, the length of $w$ is denoted by $|w|$. For example:

$$|xxxyzxyxz| = 9, \quad |(xyx)^{23}| = 69 \quad \text{and} \quad |\varepsilon| = 0 \ .$$

Given a monoid $M$, certain operations called **rational operations** are defined on the power set of $M$. The rational operations are union, product (or concatenation), and monoid closure. So if $S$ and $T$ are subsets of $M$, the product $ST$ is the set $\{st \mid s \in S \text{ and } t \in T\}$, while the monoid closure $S^*$ is the set $\{s_1 s_2 \cdots s_n \mid s_1, s_2, \ldots, s_n \in S\} \cup \{1\}$. The $n$-fold product of $S$ with itself is denoted by $S^n$, and by convention $S^0 = \{1\}$. Therefore, $S^* = \bigcup_{n=0}^{\infty} S^n$. Similarly, it will be useful to consider $S^+ = \bigcup_{n=1}^{\infty} S^n$ which is the semigroup generated by $S$ in $M$. If a set $X$ has not been previously specified as a subset of a monoid then $X^*$ denotes the free monoid with generating set $X$, while $X^+ = X^* - \{\varepsilon\}$ denotes the free semigroup on $X$.

### 1.1.2 Groups

The letter $G$ will always denote a group. $X$ will denote a set of monoid generators for $G$. Formally this means that there is a surjective homomorphism

$$\sigma : X^* \twoheadrightarrow G \ .$$

In addition, we will usually assume that $X$ is **symmetric**: That is, $X = Y \cup Y^{-1}$ where $Y$ denotes a set of *group* generators for $G$. Thus for any $y \in Y$ there is a unique letter $y^{-1} \in Y^{-1}$ and by convention $(y^{-1})^{-1} = y$. The notation $u =_G v$ denotes "$u$ and $v$ are equal in $G$", that is, $u, v \in X^*$ are words such that $\sigma(u) = \sigma(v)$. A set $R \subset X^*$ is said to be a set of **relators** if $r =_G 1$ for all $r \in R$. The formula $P = \langle\ Y \mid R\ \rangle$ is called a **presentation** for $G$ if $Y$ is a set of generators, $R$ is a set of relators, and the natural homomorphism $\mathrm{FG}(Y)/\langle\langle R \rangle\rangle \twoheadrightarrow G$ is an isomorphism, where $\mathrm{FG}(Y)$ denotes the free group on $Y$ (see §1.3.4) and $\langle\langle R \rangle\rangle$ denotes the normal subgroup in $\mathrm{FG}(Y)$ generated by $R$. The notation $\langle\ y_1, y_2, \ldots, y_n \mid r_1, r_2, \ldots, r_m\ \rangle$ will be used as short-hand for $\langle\ Y \mid R\ \rangle$ where $Y = \{y_1, y_2, \ldots, y_n\}$ and $R = \{r_1, r_2, \ldots, r_m\}$. For example, denoting commutators by $[y_i, y_j] = y_i y_j y_i^{-1} y_j^{-1}$ , the free abelian group $\mathbb{Z}^n$ of rank $n$ has the presentation $\langle\ y_1, y_2, \ldots, y_n \mid [y_i, y_j] \text{ s.t. } 1 \leq i < j \leq n\ \rangle$. These concepts are the fundamental building blocks of combinatorial group theory and are much further explored and explained in Magnass, Karass and Solitar's book [41] and in Lyndon and Schupp's book [40].

If $S \subseteq G$ then $S^{-1}$ is the set of all inverses of elements appearing in $S$ and $\langle S \rangle$ is the subgroup generated by $S$; i.e., $\langle S \rangle = (S \cup S^{-1})^*$. The **join** of the subgroups $A$ and $B$ of

$G$ is the smallest subgroup containing both $A$ and $B$ and is denoted by $A \vee B$. How do you remember what a right coset and what a left coset is? By the direction on which $G$ acts on the subgroup $H$. For example $G/H = \{gH | g \in G\}$ is acted on from the left, so $gH$ is a *left* coset. Similarly $Hg$ is a right coset even though "$H$" is on the left. The notation $H \backslash G$ means the set of all right $H$-cosets $\{Hg | g \in G\}$ and *not* the boolean difference $H - G$.

As usual, $H < G$ denotes "$H$ is a subgroup of $G$". The notation $H \underset{\text{f.i.}}{<} G$ means "$H$ is a **finite index** subgroup of $G$". In other words, the set of left cosets $G/H$ is finite (or equivalently, the set of right cosets is finite). By the notation $H \underset{n}{<} G$ is meant "$H$ is a subgroup of $G$ with index $[G : H] = n$." In other words, the set of left cosets $G/H$ has $n$ elements (or equivalently, the set of right cosets has cardinality $n$). Similarly $N \lhd G$ (resp. $N \underset{\text{f.i.}}{\lhd} G$, resp. $N \underset{n}{\lhd} G$) denotes the fact that $N$ is a normal subgroup of $G$ (resp. normal finite index subgroup, resp. normal subgroup of index $n$). Two groups $G$ and $H$ are said to be **commensurable** if they contain respective finite index subgroups $A \underset{\text{f.i.}}{<} G$ and $B \underset{\text{f.i.}}{<} H$ which are isomorphic, i.e., $A \approx B$.

### 1.1.3 Profinite topology on a group

Given any group $G$ one can define a topology on $G$ by declaring a basis for the open sets to be pre-images under homorphisms to finite groups. In the language of section 1.5 below (and especially the second condition of lemma 1.5.2) a set is a basis element if and only if it is recognizable.

**Definition 1.1.2** (The Hall —or **profinite**— topology) *A subset $C \subseteq G$ is said to be* **closed** *if given any $w \notin C$ there is a finite indexed subgroup $H \underset{\text{f.i.}}{<} G$ such that $C$ and $Hw$ are disjoint.*

Given any subset $S \subseteq G$, the **profinite closure** $\overline{S}$ is the smallest set containing $S$ which is closed in the profinite topology.

### 1.1.4 Relations

Given any two sets $S$ and $T$, a **relation** $r$ from $S$ to $T$ is a subset of $S \times T$. Given a relation $r \subseteq S \times T$, $s \in S$ and $t \in T$, the notation of table 1.1 will be used.

| | | | |
|---|---|---|---|
| $srt$ | $\iff$ | $(s,t) \in r$, | i.e. $r$ relates $s$ to $t$ |
| $sr$ | $=$ | $\{t \in T \mid srt\}$, | i.e. the image of $s$ under $r$ |
| $rt$ | $=$ | $\{s \in S \mid srt\}$, | i.e. the pre-image of $t$ under $r$ |
| $r^{-1}$ | $=$ | $\{(t,s) \in T \times S \mid srt\}$, | i.e. the inverse of $r$ |

Table 1.1: Useful notation for relations.

Given sets $S$, $T$ and $U$ and relations $r_1 \subseteq S \times T$ and $r_2 \subseteq T \times U$ one defines the **composite relation**

$$r_1 \cdot r_2 \quad = \quad \{(s,u) \in S \times U \mid \exists t \in T,\ sr_1t \ \text{ and } \ tr_2u\}.$$

Often the source and target sets of the relation are the same, i.e. $S = T$. In this case, $r$ is said to be a relation *on* $S$. The identity relation is denoted by $\mathrm{id}_S = \{(s,s) \mid s \in S\}$, while the empty relation is just the empty subset $\emptyset$. One defines exponentiation inductively by $r^0 = \mathrm{id}_S$, and $r^{i+1} = r^i \cdot r$. The transitive closure of $r$ is given by $r^+ = \bigcup_{i=1}^{\infty} r^i$. The **Kleene closure** —or reflexive transitive closure— of $r$ is given by $r^* = \bigcup_{i=0}^{\infty} r^i$. Furthermore, $r$ is symmetric iff $r = r^{-1}$, reflexive iff $r \supseteq \mathrm{id}_S$, transitive iff $r \cdot r \subseteq r$, and **irreflexive** iff $r \cap \mathrm{id}_S = \emptyset$. Finally, given $s, t \in S$, $s$ is said to be $r$**-connected** to $t$ iff $sr^+t$. Notice that $s$ need not be $r$-connected to itself. The set of all relations on $S$ forms a monoid with product "$\cdot$", unit $\mathrm{id}_S$ and a zero element $\emptyset$.

## 1.2 Complexity Functions

It is often useful to try and get a handle on the difficulty of an algorithmic problem by associating a function, called a **complexity function**, with the problem. A complexity function typically measures the amount of time or the amount of space used in a computation, as a function of the length of inputs. Only time complexity functions will be considered below. Assigning a time measurement to a certain computation approximates the number of "elementary steps" required in the computation. Elementary steps are moves that require time that is independent of the input size. For example, consider a program that inputs a string of letters and changes each letter of the string to its capital form. The program is carried out by having a cursor move from left to right, checking at each cursor position if a letter is capitalized, and changing lower case letters to upper case. Each of the following moves is an elementary step because its required time doesn't depend on the length of the inputted string: read a letter, check if a character is the end of string symbol, replace a letter by its capitalization, and move the cursor position to the right by one letter. Therefore, scanning and changing a word of length $n$ requires at most $4n$ elementary steps so that $f(n) = 4n$ is a time complexity function for the capitalization program.

One may protest that some elementary steps are more complicated than others. For example, changing a letter to its capitalization typically requires subtracting a certain number from the ASCII representation of the letter; this is probably more time consuming than moving the cursor position. Imagining that capitalizing takes twice as long as moving the cursor or reading a letter, or checking if it is the end of string symbol, would give a complexity function of $f(n) = 5n$ instead of $4n$. Such considerations depend heavily on the machine used to implement the program. However, the linearity of $f(n)$ in terms of $n$ should certainly not depend on which computer, or programming language is used to implement the program. (Of course, it is possible to conceive of a poorly designed computer that insists on rereading the whole string from scratch between every move of the cursor. This would cause the complexity of the program to become quadratic. Thus it is important that only "well designed"[1] computers be considered.) To help address these concerns an equivalence relation between functions is introduced below. Under this equivalence $f(n) = 5n$ and $g(n) = n$ are equivalent so that time complexity does not depend on the machine platform, or even on the speed of time.

[1] My first computer —an `Atari 800`— was *not* "well designed". Indeed, I enjoyed composing music using the `BASIC` cartridge that came with it. To play a note, a command would be issued to have the computer speaker output at the Hz frequency of the desired note, and a second command of the form "`for i = 1 to` $n$" with empty looping body would be issued to hold the note for $n$ computer cycles. Perplexingly, $n$ needed to be changed depending on how deep into the program the looping command was issued. For example, to hold the note for 1 second at the start of the program, a command such as "`for i = 1 to 100`" would work, while to hold the command for 1 second near the end of a 200 line program, a command such as "`for i = 1 to 25`" would be necessary. Thus time *s l o w e d* as the program progressed.

### 1.2.1 A hierarchy of functions

Let's begin with some conventions regarding complexity functions. A **complexity function** is any function $f : \mathbb{N} \to \mathbb{N}$. We take the convention that $\mathbb{N}$ is the set of all *positive* integers. Here lengths of inputs and computational time are being measured discretely. Often, it is easier to write a formula down for $f$ as a real function. For example, we would like $f(x) = x^{\frac{3}{2}}$ to be a complexity function, even though $7 \cdot \sqrt{7}$ is not a whole number. We take the convention that if $f$ is given by a real valued formula $g$, then

$$\text{for } n \in \mathbb{N}, \quad f(n) = \begin{cases} 1 & \text{if } g(n) \leq 1, \\ \lceil g(n) \rceil & \text{otherwise} \end{cases} \tag{1.2.1}$$

where $\lceil x \rceil$ denotes the least integer $n$ such that $n \geq x$. We then abuse notation by blurring the distinction between $f$ and $g$. For example, we allow $f(n) = n^{\frac{3}{2}}$ to be a complexity function, by remembering that $f(7) = 19$ and not $18.52025\ldots$.

Next, an equivalence relation on the set of complexity functions will be defined. The difficulty of a particular problem should not depend on the particular coordinate system chosen for the problem. I.e., an algorithmic problem's complexity should be robust under slight perturbations in the way the problem is viewed. There is a way of making these ideas precise, especially in the context of combinatorial group theory. Let $G$ be a finitely generated group with two sets of generators $X$ and $X'$. Let $\mathsf{P}$ be a certain computational problem whose input and output involve certain finite subsets of $G$. For example, $\mathsf{P}$ might be the problem of computing the normal core of finite index subgroups in $G$: In this case, the input would consist of an arbitrary finite set of elements in $G$ which generates a finite index subgroup $H \underset{\text{f.i.}}{<} G$, and the output would consist of a finite set generating the largest normal subgroup $N \triangleleft G$ such that $N < H$. To implement this problem on a computer it will usually be necessary to consider the generators of $X$ and define the input and output to be finite sets of *words* in $G$, as opposed to group elements. Thus implementation depends on the set of generators chosen to represent $G$. It is certainly conceivable that certain generating sets will represent a "good choice of coordinates" and bring about a simple solution to a particular problem. However, it is unreasonable to expect that a group will have a "universal" set of coordinates relative which all possible algorithmic problems have the simplest possible solution. Furthermore, there is an easy way to translate $\mathsf{P}$ from a problem relative $X$ to a problem relative $X'$. Indeed, for each letter $x \in X$ choose a word $v_x \in X'^*$, and for each letter $y \in X'$ choose a word $u_y \in X^*$ such that $x =_G v_x$ for all $x \in X$ and $y =_G u_y$ for all $y \in X'$. Now consider our computational problem $\mathsf{P}$. Suppose we have an effective procedure for solving $\mathsf{P}$ relative the generators $X$. For example, suppose we have an algorithm for computing the normal core of finite index subgroups. Given a finite subset[2] $S' \in 2^{X'^*}$ translate $S'$ to a subset $S \in 2^{X^*}$ by replacing every instance of a letter $y$ by the word $u_y$. Letting $M$ be the maximal length of words $u_y$ we see that the sum of lengths of all the words in $S$ is no larger than $M$ times the sum of lengths of all the words in $S'$. In other words, translating an input of $\mathsf{P}$ relative $X'$ to an input of $\mathsf{P}$ relative $X$ results in —at worst— a linear blow up in complexity. Similarly, translating an output of $\mathsf{P}$ relative $X$ to an output of $\mathsf{P}$ relative $X'$ can also be done with worst-case linear blow up. Thus there are constants $M$ and $N$ such that if $f(x)$ is a time complexity function for $\mathsf{P}$ relative $X$, $N \cdot f(Mx)$ is a time complexity function for $\mathsf{P}$ relative $X'$. As we don't want one generating set to be inherently superior to another set of generators it is therefore necessary to declare $f(x)$ complexity equivalent with $N \cdot f(Mx)$. This is the idea behind the next definition.

**Definition 1.2.1** *Let $f, g : \mathbb{N} \to \mathbb{N}$ be complexity functions. $f$ is said to be no more complex than $g$, which is denoted by $f \preceq g$ if there are real numbers $a, b, c, d$ such that:*

$$f(n) \leq ag(cn + d) + b \text{ for all } n \in \mathbb{N}.$$

*$f \eqsim g$ denotes "$f \preceq g$ and $g \preceq f$"; in this situation, $f$ and $g$ are said to be* **equally complex**. *Finally, $f \prec g$ denotes "$f \preceq g$ and $g \not\preceq f$" so that $g$ is strictly more complex than $f$.*

**1.2.2 Remark** *Sometimes we will also use the "big-O" notation $f = O(g)$ instead of $f \preceq g$.*

### 1.2.2 Algorithmic complexity

We next outline an intuitive notion of time complexity. Given an algorithmic problem $\mathsf{P}$, and a procedure for solving $\mathsf{P}$, it will be assumed that the "elementary steps" of the procedure are well understood (see the discussion on p. 4). We say that a complexity

[2]Given a set $A$, $2^A$ denotes the power set of $A$. I.e. $2^A = \{ B \mid B \subseteq A \}$.

function $f$ is an **actual complexity function** for $\mathsf{P}$ if there is a procedure for solving $\mathsf{P}$ such that given any input of size $n$, the procedure halts with the correct output having taken at most $f(n)$ elementary steps. This situation is denoted by $\mathsf{P} \leq f$.

As outlined at the beginning of section 1.2, just what an elementary step is depends on the machine model being used and on the speed of time. We take the hypothesis that any two "best" machine models can be translated back and forth using a linear blow-up. Under such translations, complexity functions change to functions that are complexity equivalent to each other in the sense of definition 1.2.1. The problem $\mathsf{P}$ is said to have complexity function $f$, if it has an *actual* complexity function $g$ which is complexity equivalent to $f$. In other words:

**Definition 1.2.3** *Let* $\mathsf{P}$ *be an algorithmic problem and let* $f$ *be a complexity function.* $\mathsf{P}$ *is said to have* **algorithmic time complexity** $f$*, which is denoted by*

$$\mathsf{P} \preceq f$$

*if there is a complexity function* $g \eqsim f$*, and a procedure for solving* $\mathsf{P}$ *such that given any input of size* $n$*, the procedure halts with the correct output for* $\mathsf{P}$ *using at most* $g(n)$ *elementary steps.*

**1.2.4 Remark** $\mathsf{P} \prec f$ *is used when the problem* $\mathsf{P}$ *has algorithmic time complexity strictly smaller than* $f$*. I.e., there is a complexity function* $g$ *such that* $\mathsf{P} \preceq g$ *and* $g \prec f$*.*

## 1.3 Finite State Automata

Automata are abstractions of various sorts of computing machines. Finite state automata are meant to represent classical[3] computers with bounded memory. As our universe contains only finitely many atoms, all real computers have bounded memory. Therefore, theoretically at least, finite state automata model all real classical computers. However, it is highly impractical and misleading to think of all real computers as finite state automata. Rather, the more powerful model of a Turing machine is usually called upon to model the most general types of classical computers. One of the properties that distinguishes a Turing

[3] As opposed to quantum computers.

machine from a finite state automaton is the ability to add as much memory as necessary in carrying out a computation. This means, for example, that given enough time a Turing machine can decide if an inputted number is a perfect square. On the other hand, no finite state automaton could be built which would be able to determine whether an arbitrary number is a square. Although it is apparent that our universe does not allow for computing the square root of a number containing more digits the the number of atoms in the universe, it is equally apparent that a very simple computer can be built for computing square roots of numbers (and deciding whether they are whole) for numbers that most people care about. A simple electronic calculator will easily help us decide if $1,031,213$ is a perfect square (it isn't). On the other hand, if we insisted that all computers must be as in definition 1.3.1, our calculator (especially a scientific calculator) would blow up exponentially to a machine probably containing more atoms than there are on earth!

There are many other machine models with computational power lying strictly between finite state automata and Turing machines. The standard source to learn about these models is Hopcroft and Ullman's book [34]. If the reader is interested in a very unified treatment of the various machine models and types of programming languages, Floyd and Beigel's book [19] is recommended.

### 1.3.1 Automata

An automaton is a directed graph endowed with a way of labeling (or coloring) edges, and a choice of some start and terminal vertices. Every automaton defines a certain language on the labeling alphabet. The language is defined as the set of all labels of paths which start at a start vertex and end at a terminal vertex. A language defined in this way by a *finite* automaton is called a "regular language".

**Definition 1.3.1** *An automaton $\mathcal{A}$ consists of a set of vertices $V = V(\mathcal{A})$, two subsets $S$, $T \subseteq V$, a set of labels $X$ and a set of relations $R = \{\ r_y \mid y \in X \cup \{\varepsilon\}\ \}$ on $V$. $S$ is the set of* **start** *vertices, while $T$ is the set of* **terminal** *vertices. Given $y \in X \cup \{\varepsilon\}$, and $v, w \in V$ such that $v r_y w$, $v \xrightarrow{y} w$ denotes an* **edge** *in $\mathcal{A}$ labeled by $y$. A* **path** *in $\mathcal{A}$ is denoted by*

$$\gamma = v_0 \xrightarrow{y_1} v_1 \xrightarrow{y_2} v_2 \cdots v_{n-1} \xrightarrow{y_n} v_n$$

*where each of the* $v_i \overset{y_{i+1}}{\to} v_{i+1}$ *is an edge. The path* $\gamma$ **initiates** *at* $\iota(\gamma) = v_0$ *and* **terminates** *at* $\tau(\gamma) = v_n$. *The* **label** *of* $\gamma$ *is the element of the free monoid* $X^*$ *formed by the product of edge labels* $y_1 y_2 \cdots y_n$, *and is denoted by* $\lambda(\gamma)$. *An element* $w$ *of* $X^*$ *is said to be* **accepted** *by* $\mathcal{A}$ *—denoted by* $w \in \mathcal{A}$*— if there is a path* $\gamma$ *labeled by* $w$ *which starts at a vertex inside* $S$ *and ends at a vertex inside of* $T$.

**1.3.2 Remark** *Recall that* $\varepsilon$ *denotes the identity element of the free monoid* $X^*$. *Consequently, occurrences of* $\varepsilon$ *in the label of a path are empty: For example, the label of the path*

$$\gamma = v_0 \overset{x_1}{\to} v_1 \overset{x_2}{\to} v_2 \overset{\varepsilon}{\to} v_3 \overset{\varepsilon}{\to} v_4 \overset{x_2}{\to} v_5 \overset{\varepsilon}{\to} v_6 \overset{x_3}{\to} v_7$$

*is the element* $\lambda(\gamma) = x_1 x_2 x_2 x_3$ *of* $X^*$.

*For each color* $x$ *there is a unique* **colored relation** $r_x$ *which is the set of all directed edges labeled by* $x$. *Furthermore, there is also a unique* **colorless relation** $r_\varepsilon$ *which is the set of all epsilon edges. As defined, automata can have at most one edge of a given label between any two vertices. Removing such redundant edges does not change the language accepted by an automaton, so that this convention does not hinder the descriptive power of automata.*

*The automaton* $\mathcal{A}$ *is identified with the language of all words which it accepts, so we abuse notation by viewing* $\mathcal{A} \subseteq X^*$.

*Figure 1.1 hilights the way that start and terminal vertices will be denoted. A vertex with a quivered arrow pointing into it represents a start state, while a vertex with a double arrow pointing out of it is an accept state.*

### 1.3.2 Concatenating automata

Given automata $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ with respective labeling sets $X_1, X_2, \ldots, X_n$ one can construct the $n$**-fold concatenation** $\mathcal{P} = \mathcal{A}_1 \mathcal{A}_2 \cdots \mathcal{A}_n$ with label set $X = X_1 \cup X_2 \cup \cdots \cup X_n$ by defining the vertex set to be $V(\mathcal{P}) = \bigcup_{i=1}^n V(\mathcal{A}_i)$, the start vertices to be $S(\mathcal{P}) = S(\mathcal{A}_1)$, and the terminal vertices to be $T(\mathcal{P}) = T(\mathcal{A}_n)$. The colored relations are defined by $r_x(\mathcal{P}) = \bigcup_{i=1}^n r_x(\mathcal{A}_i)$, for each $x \in X$; that is, the colored edges of $\mathcal{P}$ are the

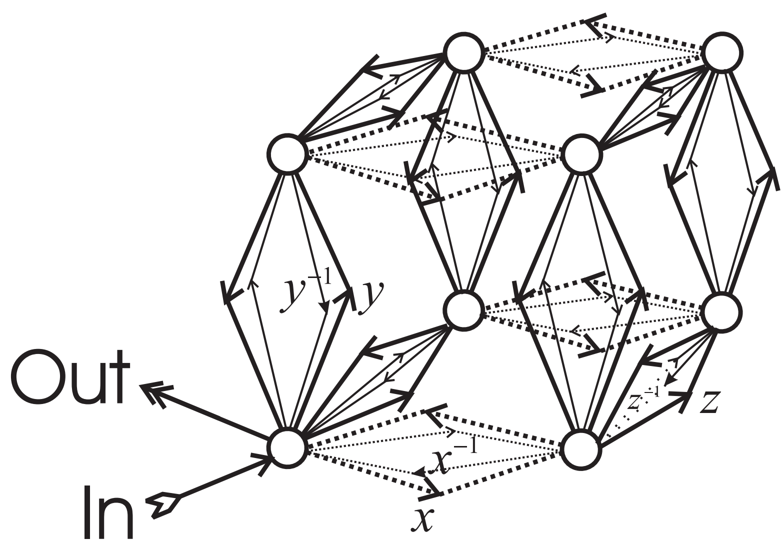



Figure 1.1: An automaton accepting all words representing the identity of $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$.

colored edges of all the $\mathcal{A}_i$. The $\varepsilon$-relation is given by

$$r_\varepsilon(\mathcal{P}) = \left[\bigcup_{i=1}^{n} r_\varepsilon(\mathcal{A}_i)\right] \bigcup \left[\bigcup_{i=1}^{n-1} T(\mathcal{A}_i) \times S(\mathcal{A}_{i+1})\right].$$

In other words, the $\varepsilon$-edges of $\mathcal{P}$ are all the $\varepsilon$-edges of the $\mathcal{A}_i$ with additional $\varepsilon$-edges connecting the terminal states of $\mathcal{A}_i$ to the start states of $\mathcal{A}_{i+1}$. Notice that the notation used for the $n$-fold concatenation is in the spirit of the previous remark. Indeed, the language accepted by $\mathcal{A}_1\mathcal{A}_2\cdots\mathcal{A}_n$ in $X^*$ is precisely the $n$-fold product of the languages accepted by $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$, which is given by

$$\mathcal{A}_1\mathcal{A}_2\cdots\mathcal{A}_n = \{\ u_1u_2\cdots u_n \mid u_i \in \mathcal{A}_i \text{ for } i = 1, 2, \ldots, n\}\ .$$

### 1.3.3 Intersecting automata

There is also a pull-back construction which gives the $n$-**fold intersection** of automata. For simplicity we assume that $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ are $\varepsilon$-free automata, i.e. $r_\varepsilon(\mathcal{A}_i) = \emptyset$ for all $i$. The $n$-fold intersection will only be used on $\varepsilon$-free automata, though such a construction is definable (but more cumbersome) for general automata. One constructs the $n$-fold intersection $\mathcal{I} = \bigcap_{i=1}^{n} \mathcal{A}_i$ by defining the vertex set to be $V(\mathcal{I}) = \times_{i=1}^{n} V(\mathcal{A}_i)$, the

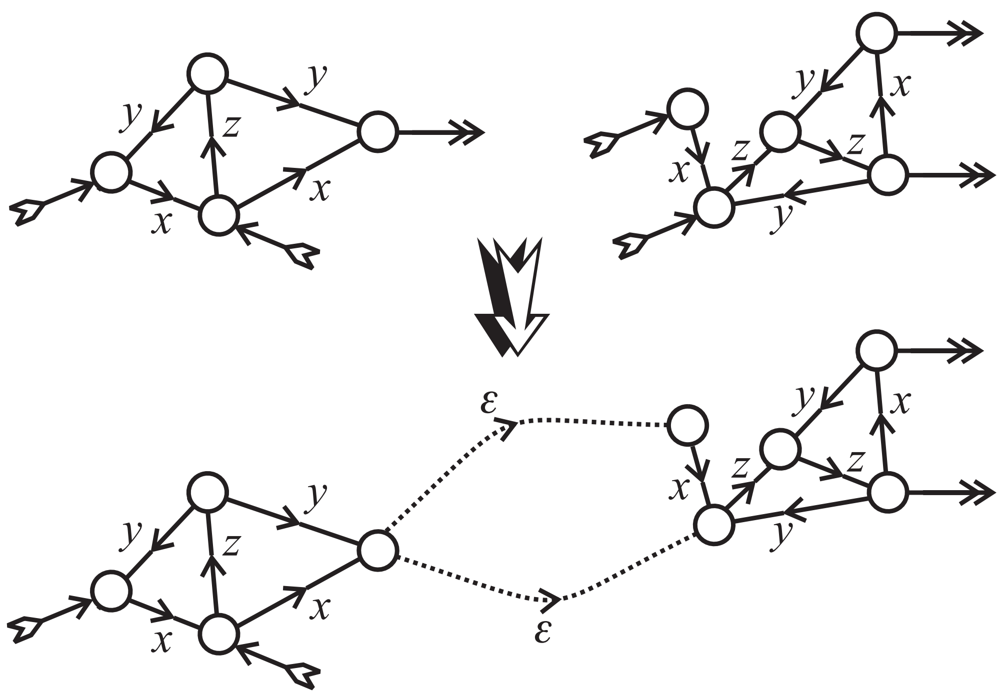



Figure 1.2: Concatenation of automata.

start states to be $S(\mathcal{I}) = \times_{i=1}^{n} S(\mathcal{A}_i)$, and the terminal states to be $T(\mathcal{I}) = \times_{i=1}^{n} T(\mathcal{A}_i)$. The colored relations are defined by

$$r_x(\mathcal{I}) =$$

$$\Big\{ \big( (s_1, \dots, s_n) , (t_1, \dots, t_n) \big) \in V(\mathcal{I}) \times V(\mathcal{I}) \quad \Big| \quad (s_i, t_i) \in r_x(\mathcal{A}_i) \text{ for all } i = 1, 2, \dots n \Big\} .$$

In other words, $\mathcal{I}$ has as its vertices (resp. start vertices, resp. accept vertices) the Cartesian product of the vertices (resp. start vertices, resp. accept vertices) of all the $\mathcal{A}_i$; therefore, vertices of $\mathcal{I}$ can be thought of as vectors whose $i$-th coordinate is a vertex of $\mathcal{A}_i$. An $x$-colored edge exists between two such vectors in $\mathcal{I}$ if and only if $x$-colored edges exist in all coordinates. Again, the notation does not contradict our terminology as the language accepted by $\bigcap_{i=1}^{n} \mathcal{A}_i$ is the set theoretic intersection of the languages accepted by $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n$ inside of $X^*$.

Notice that the pull-back construction is linear in the complexity of each automaton: Define the complexity of an automaton $\mathcal{A}$ to be the number of edges in the automaton, and denote this number by $|\mathcal{A}|$. At first one might protest that the complexity should also depend on the number of vertices. Nevertheless, any automaton can be trimmed to an equivalent automaton containing at most one isolated vertex, justifying the counting of edges only. By inspection we have:

**Lemma 1.3.3**

$$\left|\bigcap_{i=1}^{n} \mathcal{A}_i\right| \leq \prod_{i=1}^{n} |\mathcal{A}_i| \quad .$$

### 1.3.4 Automata and free groups

In this subsection we regard automata as representing certain subsets of free groups. $X = Y \cup Y^{-1}$ is a symmetric set of generators. A **freely reduced word** $w \in X^*$ is a word which has no subwords of the form $xx^{-1}$. Given any word $w$ there is a unique reduced word $[w]$ which is obtained from $w$ by removing subwords of the form $xx^{-1}$ until no such subwords can be found. One defines the **free group** on $Y$, $\mathrm{FG}(Y)$, as the set of all reduced words in $X^*$; furthermore, the product in $\mathrm{FG}(Y)$ is given by the rule:

$$u \circ v = [uv].$$

A **freely reduced path** in an automaton is a path with no subpaths of the form

$$a \xrightarrow{\varepsilon} b \xrightarrow{\varepsilon} c, \quad \text{or} \quad a \xrightarrow{x} b \xrightarrow{x^{-1}} c, \quad \text{or} \quad a \xrightarrow{x} b \xrightarrow{\varepsilon} c \xrightarrow{x^{-1}} d \quad \text{or} \quad a \xrightarrow{\varepsilon} a. \tag{1.3.1}$$

A freely reduced automaton $\mathcal{A}$ is an automaton with the property that shortcuts can be found around paths given by formulas (1.3.1). Formally:

**Definition 1.3.4** *An automaton $\mathcal{A}$ is said to be* **freely reduced** *iff*

- $r_\varepsilon \cup \mathrm{id}_V$ *is transitive,*
- *and for all $x \in X, r_x \cdot (r_\varepsilon \cup \mathrm{id}_V) \cdot r_{x^{-1}} \subseteq r_\varepsilon \cup \mathrm{id}_V$.*

Equivalently, we could have defined an automaton to be freely reduced if every path with label $w$ admits a shortcut path between the same vertices whose label is the reduction $[w]$:

**Lemma 1.3.5 (Second definition of freely reduced automaton)** *An automaton $\mathcal{A}$ is freely reduced if and only if given any path $\alpha$ from $a$ to $b$ with label $w$, there is a freely reduced path $[\alpha]$ from $a$ to $b$ with label $[w]$.*

*Proof.* 1.3.4⇒1.3.5: Prove by contradiction. So find a shortest path $\alpha$ in $\mathcal{A}$ with no shortcut to a freely reduced path. By assumption $\alpha$ is not freely reduced, and no shorter path $\alpha'$ can be found between the same endpoints of $\alpha$ such that $\lambda(\alpha')$ is a reduction of $\lambda(\alpha)$. As $\alpha$ is not freely reduced there is a subpath as in formulas (1.3.1). In the first case of these formulas, the transitivity of $r_\varepsilon \cup \mathrm{id}_V$ implies that a shortcut to $a \xrightarrow{\varepsilon} b \xrightarrow{\varepsilon} c$ of the form $a \xrightarrow{\varepsilon} c$ can be found (or in the case $a = c$ the shortcut is just the constant path $a$). In the second and third cases, $r_x \cdot (r_\varepsilon \cup \mathrm{id}_V) \cdot r_{x^{-1}} \subseteq r_\varepsilon \cup \mathrm{id}_V$ implies that a shortcut of the form $a \xrightarrow{\varepsilon} c$ (or $a \xrightarrow{\varepsilon} d$ or just $a$) can be found. In the last case, replace $a \xrightarrow{\varepsilon} a$ by $a$. Thus in all cases a shorter path $\alpha'$ whose label reduces the label of $\alpha$ has been found, which contradicts our assumption.

1.3.5⇒1.3.4: Suppose that all paths admit free reductions. In particular, any path of the form $a \xrightarrow{\varepsilon} b \xrightarrow{\varepsilon} c$ must have a reduction of the form $a \xrightarrow{\varepsilon} c$ (or just $a$) showing that $r_\varepsilon \cup \mathrm{id}_V$ is transitive. Similarly paths of the second and third kind of formulas (1.3.1) admit shortcuts of the form $a \xrightarrow{\varepsilon} c$ (or $a \xrightarrow{\varepsilon} d$ or just $a$) showing that the second condition of definition 1.3.4 is satisfied. ⧫

**1.3.6 Remark** *Given any path $\alpha$ in a reduced automaton, $[\alpha]$ will denote a reduced path as in the previous lemma.*

In the sequel, automata will be constructed to better understand subsets of $G$ which are the images of regular languages (these are called "rational subsets", cf. section 1.6). Thus, given an automaton $\mathcal{A} \subseteq X^*$ we will be more interested in its image $\sigma(\mathcal{A})$ in $G$. The advantage of a freely reduced automaton $\mathcal{A}$ is that for any word $w$ accepted by $\mathcal{A}$, so is the reduction $[w]$. In other words, letting $G = \mathrm{FG}(Y)$ be the free group on $Y$ and $\sigma : X^* \twoheadrightarrow G$ be the reduction map $\sigma = [-]$, a freely reduced automaton has the property that every element of $\sigma(\mathcal{A})$ is represented by some reduced path accepted by $\mathcal{A}$. This is useful for solving various decision problems. Thus given an automaton $\mathcal{A}$ we aim to construct a freely reduced automaton $\widetilde{\mathcal{A}}$ such that $\sigma(\mathcal{A}) = \sigma(\widetilde{\mathcal{A}})$. The following algorithm of Gilman [23] achieves this goal:[4]

[4] Independently, Benois [5] gave an analogous algorithm, but for regular expressions (definition 1.4.1) as

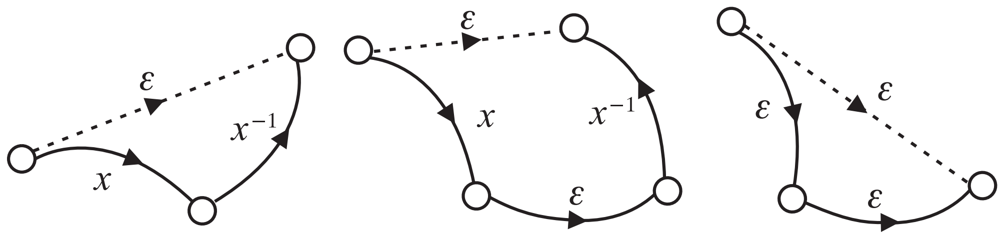



Figure 1.3: The $\varepsilon$-edges added in Gilman's reduction algorithm with $x \in Y \cup Y^{-1}$.

**Algorithm 1.3.7 (free reduction)** *The free reduction $\widetilde{\mathcal{A}}$ is built from $\mathcal{A}$ as follows:*

*While there are two vertices $a \neq b$ and a path of the form $a \xrightarrow{\varepsilon} c \xrightarrow{\varepsilon} b$, or of the form $a \xrightarrow{x} c \xrightarrow{x^{-1}} b$ or $a \xrightarrow{x} c \xrightarrow{\varepsilon} d \xrightarrow{x^{-1}} b$ with $x \in X = Y \cup Y^{-1}$, but there is no edge $a \xrightarrow{\varepsilon} b$, add this missing edge.*

*The initial and accept states of $\widetilde{\mathcal{A}}$ are inherited from $\mathcal{A}$.*

*$\widetilde{\mathcal{A}}$ satisfies:*

1. *$\widetilde{\mathcal{A}}$ is freely reduced.*
2. *If $\mathcal{A}$ is freely reduced then $\widetilde{\mathcal{A}} = \mathcal{A}$.*
3. *Given any vertices $a$ and $b$, if there is a path $\gamma$ from $a$ to $b$ in $\widetilde{\mathcal{A}}$ with label $u$ then there is a path $\gamma'$ from $a$ to $b$ in $\mathcal{A}$ whose label $u'$ has the property that $[u'] = [u]$.*
4. *For $a \neq b$, the edge $a \xrightarrow{\varepsilon} b$ exists in $\widetilde{\mathcal{A}}$ if and only if there is a path $\gamma$ in $\mathcal{A}$ starting at $a$, ending at $b$ and with label $w$ such that $\sigma(w) = 1$.*
5. *$\sigma(\mathcal{A}) = \sigma(\widetilde{\mathcal{A}})$.*
6. *If $w \in X^*$ is a reduced word and $\sigma(w) \in \sigma(\mathcal{A})$ then $w \in \widetilde{\mathcal{A}}$.*

*Proof.* 1. $\widetilde{\mathcal{A}}$ is freely reduced: Consider a path $\alpha$ in $\widetilde{\mathcal{A}}$. Let $\alpha'$ be a shortest path with the same endpoints as $\alpha$ and with the same image in $\mathrm{FG}(Y)$. By virtue of the minimality of $\alpha'$ and the construction of $\widetilde{\mathcal{A}}$, $\alpha'$ contains no sub-paths of the form $a \xrightarrow{\varepsilon} c \xrightarrow{\varepsilon} b$, or of the form $a \xrightarrow{x} c \xrightarrow{x^{-1}} b$ or $a \xrightarrow{x} c \xrightarrow{\varepsilon} d \xrightarrow{x^{-1}} b$. Therefore, $\alpha'$ is a reduced path showing that $\widetilde{\mathcal{A}}$ is reduced.

opposed to automata. Gilman's algorithm also works for monadic groups (see footnote 7, p. 95).

2. If $\mathcal{A}$ is freely reduced then $\widetilde{\mathcal{A}} = \mathcal{A}$: By definition 1.3.4 since $\mathcal{A}$ is reduced, algorithm 1.3.7 produces no new edges so that $\widetilde{\mathcal{A}} = \mathcal{A}$.

3. There is a sequence of automata

$$\mathcal{A} = \mathcal{A}_0 \subset \mathcal{A}_1 \subset \cdots \subset \mathcal{A}_{n-1} \subset \mathcal{A}_n = \widetilde{\mathcal{A}}$$

such that $\mathcal{A}_{i+1}$ is obtained from $\mathcal{A}_i$ by adding a single edge $a \xrightarrow{\varepsilon} b$ to shortcut $a \xrightarrow{\varepsilon} c \xrightarrow{\varepsilon} b$, or $a \xrightarrow{x} c \xrightarrow{x^{-1}} b$ or $a \xrightarrow{x} c \xrightarrow{\varepsilon} d \xrightarrow{x^{-1}} b$. Consider a path $\gamma$ in $\mathcal{A}_{i+1}$ with label $u$. Construct a path $\gamma'$ in $\mathcal{A}_i$ from $\gamma$ by replacing each appearance of the new edge $a \xrightarrow{\varepsilon} b$ by the path $\alpha$ for which this edge was a shortcut to. Notice that $\alpha$ has trivial label in $\mathrm{FG}(Y)$. Therefore, letting $u$ (resp. $u'$) be the label of $\gamma$ (resp. $\gamma'$) we have $[u] = [u']$. Finally, using induction property number 3 is obtained.

4. This is obtained from the previous by letting $\gamma = a \xrightarrow{\varepsilon} b$ so that $u = \varepsilon$ and $[u] = \varepsilon$, and letting $w = u'$ so $\sigma(w) = \sigma(u) = 1$.

5. $\sigma(\mathcal{A}) = \sigma(\widetilde{\mathcal{A}})$: As $\widetilde{\mathcal{A}}$ contains the edges, initial states and accepts states of $\mathcal{A}$, it follows that $\sigma(\mathcal{A}) \subseteq \sigma(\widetilde{\mathcal{A}})$. On the other hand, let $g \in \sigma(\widetilde{\mathcal{A}})$. Then $g = \sigma(u)$ where $u$ is the label of a certain path $\gamma$ in $\widetilde{\mathcal{A}}$ from $a \in S$ to $b \in T$. By property 3 there is path $\gamma'$ in $\mathcal{A}$ whose label $u'$ has the same free reduction as the label of $\gamma$. In other words, $\sigma(u') = \sigma(u) = g$, with $u' \in \mathcal{A}$. Therefore, $g \in \sigma(\mathcal{A})$.

6. Suppose $w$ is a reduced word such that $\sigma(w) \in \sigma(\mathcal{A})$. By number 5, $\sigma(w) \in \sigma(\widetilde{\mathcal{A}})$. Therefore, there is a word $w' \in \widetilde{\mathcal{A}}$ such that $\sigma(w') = \sigma(w)$. In other words, $[w'] = [w] = w$, since $w$ is reduced. Since $\widetilde{\mathcal{A}}$ is freely reduced and $w$ is the free reduction of $w'$ lemma 1.3.5 implies that $w \in \widetilde{\mathcal{A}}$. ♦

## 1.4 Regular Languages

The regular languages will play a fundamental role in the rest of this dissertation. Facts about regular languages will be useful in analyzing algorithmic properties of groups, and will also provide motivation for closely related concepts in the theory of discrete groups.

What does it mean for a subset of a monoid to be regular? If the monoid is free, then one can just use the usual notions of regularity defined in this section. But when talking about more general monoids, or groups, it becomes less clear what regularity really is. In sections 1.5, 1.6, 1.7 and 3.4 reinterpretations of the notions of regularity for monoids, groups and word hyperbolic groups will be given.

Proofs of statements in this section and the following sections concerning regular, recognizable and rational subsets are either omitted or are given in a very sketchy manner. These proofs can be found in [15], [17, chapter 1], [19], [34], [35], [47] and many other books on automata and formal languages. Furthermore, what I term a "definition" may be the result of some argument which can be found in these sources. A recommended reference for Kleene's theorem and syntactic monoids is [35]. Recognizable and rational subsets are treated in [6] and [15] while the relationship between infinite groups and rational subsets is expounded upon in [25].

### 1.4.1 Regular expressions

Let $M$ be a finitely generated free monoid, i.e. $M = X^*$ for a finite alphabet $X$. Regularity is a property of subsets of $M$, i.e., languages. Kleene's theorem (definition 1.4.2) shows that regularity is a very stable notion as it may be defined in several ways. Another way to define regularity is in terms of grammars (see subsection 1.8.7). Before giving the definition of a regular language we need the notion of a regular expression. Regular expressions are built up from words by applying formal rational operations.

**Definition 1.4.1** *Let $X$ be an alphabet. A* **regular expression** *over $X$ is a string with letters in $X \bigcup \{(,), \vee, {}^*\}$ which is obtainable using the following inductive process:*

- *there is an empty expression ( ),*
- *each word $u \in X^*$ is a regular expression,*
- *if $u$ and $v$ are regular expressions, so is the disjunction $(u) \vee (v)$,*
- *if $u$ and $v$ are regular expressions, so is the concatenation $(u)(v)$,*
- *if $u$ is a regular expression, so is the Kleene star $(u)^*$.*

*A regular expression $w$ generates a language $\mathcal{L}(w) \subseteq X^*$ using the inductive process:*

- *if $w = (\ )$ then $\mathcal{L}(w) = \emptyset$, i.e., the empty set,*

- *if* $w \in X^*$ *then* $\mathcal{L}(w) = \{w\}$, *i.e., the singleton containing* $w$,
- *if* $w = (u) \vee (v)$ *then* $\mathcal{L}(w) = \mathcal{L}(u) \cup \mathcal{L}(v)$, *i.e., the union,*
- *if* $w = (u)(v)$ *then* $\mathcal{L}(w) = \mathcal{L}(u) \cdot \mathcal{L}(v)$, *i.e., the product,*
- *if* $w = (u)^*$ *then* $\mathcal{L}(w) = \mathcal{L}(u)^*$, *i.e., the monoid closure.*

Figure 1.5 on p. 23 shows graphically how the rational operations act on automata.

### 1.4.2 Kleene's theorem

Languages that are generated by regular expressions are termed regular. Kleene's theorem states that regular languages are also characterized by finite state automata acceptance (see definition 1.3.1), and as the homomorphic pre-images of finite subsets.

**Definition 1.4.2** (Kleene's theorem [39]) *Let* $S \subseteq X^*$. *Then* $S$ *is* **regular** *if one of the following equivalent conditions is satisfied:*

1. $S$ *is generated by a regular expression.*
2. $S$ *is accepted by a finite automaton whose edges are labeled by* $X \cup \{\varepsilon\}$.
3. *There is a* finite *monoid* $N$. $\rho : M \to N$ *and a subset* $T \subseteq N$ *s.t.* $S = \rho^{-1}(T)$.

That 1 and 2 are equivalent is well known. A sketch that 1 implies 2 is described in remark 1.6.5; on the other hand, the proof of corollary 2.2.5 below can be modified slightly to show that 2 implies 1. For a complete proof of the equivalence of 1 and 2 the reader is referred to theorem 1.2.7 of [17], though many other sources explain this fact. That 2 and 3 are equivalent follows from the fact that one can think of the image monoid $\rho(M)$ as an automaton accepting $S$; on the other hand, $S$ is the pre-image of a subset of the syntactic monoid corresponding to a finite state automaton. The reader is referred to [25, §3], or [35] for a proof of the equivalence of 2 and 3.

For an arbitrary monoid $M$, the equivalence of the conditions in 1.4.2 breaks down. Conditions 1 and 3 split apart, while condition 2 becomes ambiguous and can be reinterpreted in several ways. Condition 1 gives rise to the notion of "rationality" which is discussed in section 1.6. Condition 3 is "recognizability" and is described in section 1.5.

Each of these sections give a way of defining "acceptance" so as to make condition 2 the same as rationality or recognizability. A third re-interpretation of condition 2 in section 1.7 gives rise to the notion of "$\mathcal{L}$-regularity". In the context of word hyperbolic groups, one can specialize $\mathcal{L}$-regularity to "geodesic regularity" (section 3.4).

## 1.5 Recognizable Subsets

A reinterpretation of condition 2 of definition 1.4.2 which makes it equivalent with condition 3 gives rise to the notion of recognizability. Facts about recognizable subsets can be found in [6] and [15].

### 1.5.1 $M$-automata

A subset $S \subseteq M$ is recognizable if it is the set accepted by some finite $M$-automaton:

**Definition 1.5.1** *A finite $M$-**automaton**, is a triple $A = (Q, q_0, F)$ with $M$ acting on $Q$. That is, given $q \in Q$ and $m, n \in M$, there is a well defined element $qm \in Q$ such that $q1 = q$ and $(qm)n = q(mn)$. The state $q_0 \in Q$ is called the **start** state, and $F$ is the set of **final** states. The set **recognized** by $A$ is $\{m \in M \mid q_0 m \in F\}$ which is the set of all elements carrying the initial state to a final state. The $M$-automaton $A$ is identified with the subset of $M$ which it recognizes. This set is called **recognizable**, and the collection of all recognizable subsets of $M$ is denoted by* $\mathrm{Rec}(M)$.

### 1.5.2 Other formulations of $\mathrm{Rec}(M)$

Assume that $M$ is generated by $X$. That is, there is a surjective homomorphism $\sigma : X^* \twoheadrightarrow M$.

**Lemma 1.5.2** *Let $S \subseteq M$. Then $S$ is recognizable if one of the following equivalent conditions is satisfied:*

1. *$S$ is recognized by some $M$-automaton.*
2. *There is a* finite *monoid $N$, a homomorphism $\rho : M \to N$ and a subset $T \subseteq N$ for which $S$ is the pre-image of $T$ under $\rho$.*

3. *The pre-image $\sigma^{-1}(S)$ of $S$ in $X^*$ is regular.*

*Sketch of proof.* $1 \Rightarrow 2$ since an $M$-automaton gives rise to a homomorphism into the finite transformation monoid $N = Q^Q$ (the monoid of all functions on the set $Q$) and we can let $T = \{f : Q \to Q \mid f(q_0) \in F\}$. $2 \Rightarrow 3$ because the pre-image of $S = \rho^{-1}(T)$ under $\sigma$ is just the pre-image of $T$ in $X^*$ so we can use condition 3 of definition 1.4.2. $3 \Rightarrow 1$ because if the pre-image of $S$ is regular, its syntactic monoid is finite and admits a quotient to a monoid acted on by $M$; furthermore, the quotient monoid can be given an $M$-automaton structure which recognizes $S$. ⋄

### 1.5.3 Cayley graphs for a group $G$

Let $H < G$, and let $X$ be a symmetric set of generators for $G$. The **Cayley graph** or **coset diagram** of the right cosets of $H$ relative $X$ is denoted by $\Gamma_X(H\backslash G)$. It is a directed $X$-labeled graph defined as follows:

1. The vertex set $V(\Gamma_X(H\backslash G))$ is the set of right cosets $H\backslash G$.

2. For every vertex $Hg \in H\backslash G$ and every generator $x \in X$ there is an edge $(Hg, Hgx)$ in $E(\Gamma_X(H\backslash G))$ which is labeled by $x$. This $x$-labeled edge is denoted by $Hg \xrightarrow{x} Hgx$.

**1.5.3 Remark** *When $H = \{1\}$ then $\Gamma_X(G) = \Gamma_X(\{1\}\backslash G)$ denotes the standard Cayley graph of G relative $X$.*

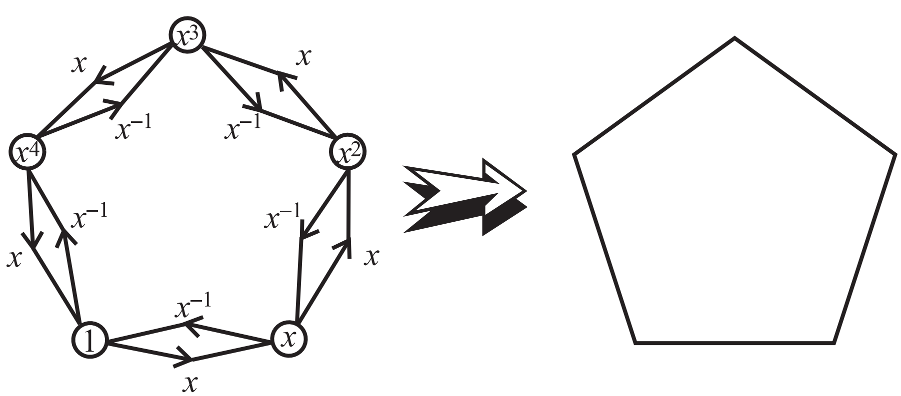



Figure 1.4: The Cayley graph $\Gamma_X(G)$ and its geometric realization $||\Gamma_X(G)||$ for the cyclic group $G = \langle\, x \mid x^5 = 1 \,\rangle$.

It is often useful to have a topological formulation of the Cayley graph. Then the Cayley graph is viewed as a topological space and metric methods may be applied. We follow Serre's convention [48, p. 14] in defining the geometric realization:

**Definition 1.5.4** *Let $\Gamma$ be a directed graph labeled symmetrically by a symmetric set $X$. I.e., for any edge $u \stackrel{x}{\to} v$ suppose that there is a unique inverse edge $v \stackrel{x^{-1}}{\to} u$. The* **geometric realization** $||\Gamma||$ *is an undirected graph with the same vertices as $\Gamma$ and with an edge arising from mutually inverse pairs. In other words, an edge of $||\Gamma||$ is given by:*

$$e = \{u \stackrel{x}{\to} v, v \stackrel{x^{-1}}{\to} u\} \ .$$

*The graph $||\Gamma||$ is endowed with a topology by viewing it as a 1-complex.*

When $H \underset{\text{f.i.}}{<} G$, $\Gamma_X(H\backslash G)$ is a finite graph. Therefore, by defining certain start states $S \subseteq H\backslash G$ and terminal states $T \subseteq H\backslash G$ one obtains a finite $G$-automaton. In fact:

**Lemma 1.5.5** *Let $S = \{H\}$ and $T = \{Hg\}$ for some $g \in G$. Let $\mathcal{L}$ be the language accepted by $\Gamma_X(H\backslash G)$ with $S$ as the set of start states, and $T$ as the set of terminal states. Then*

$$\mathcal{L} = \sigma^{-1}(Hg) \quad . \tag{1.5.1}$$

*Therefore, cosets of finite indexed subgroups are recognizable.*

## 1.6 Rational Subsets

This section describes the rational subsets. These subsets arise by re-interpreting condition 2 of definition 1.4.2 in a manner that makes it equivalent with condition 1 for general monoids $M$. The statements which go unproven in this section come from Gilman's recommended survey article [25]. For a thorough treatment of rational subsets Eilenberg's treatise [15] is recommended.

### 1.6.1 Rat($M$)

The rational subsets of $M$ are the subsets which are generated by regular expressions:

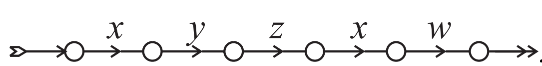
**Definition 1.6.1** *The collection of rational subsets of a monoid $M$, denoted by* $\mathrm{Rat}(M)$, *is the smallest collection of subsets which contains all finite subsets, and is closed under monoid closure, product, and finite union.*

**1.6.2 Remark** *Notice that if $S$ is a rational subset, so is $S^+ = SS^*$. Conversely, $S^* = S^+ \cup \{1\}$ so one could have defined rational subsets using semigroup closure instead of monoid closure.*

**1.6.3 Example** *In a group $G$, all finitely generated subgroups are rational. In fact, suppose $H = \langle h_1, h_2, \dots, h_n \rangle$ is the subgroup generated by $h_1, h_2, \dots, h_n \in G$. Then $H = \{h_1, h_2, \dots, h_n, h_1^{-1}, h_2^{-1}, \dots, h_n^{-1}\}^*$ so $H$ is the monoid closure of a finite set. Conversely, it can be shown that if a subgroup $H$ is rational, then $H$ is finitely generated [24, corollary 1].*

**1.6.4 Example** *Consider two finitely generated subgroups $H, K$ of a group $G$. By the previous example $H$ and $K$ are rational. Therefore, given $g \in G$, the double coset $H\{g\}K$ is also rational. Thus a solution to the membership problem for rational subsets of $G$ yields a solution to the membership problem of $H\{g\}K$.*

**1.6.5 Remark** *Rational subsets are connected to finite state automata via Kleene's theorem. If $M$ happens to be a free monoid, the rational subsets are precisely those subsets which are accepted by finite state automata. In other words, the rational subsets of a free monoid are the regular languages. To see this notice, for example, that the singleton $\{xyzxw\}$ is accepted by the automaton*




*Any rational subset can be built up from singletons by rational operations, and the rational operations are well defined on automata. Union, concatenation and monoid closure of nondeterministic automata are illustrated in figure 1.5. In the figure, initial and final vertices of $S$ and $T$ are represented by regions respectively shaded light and dark.*

In general, the rational subsets of $M$ are precisely the images of rational languages:

**Lemma 1.6.6** (cf. [24, §3]) *Let $M$ be a monoid and $X$ a set of generators. That is, let $\sigma : X^* \twoheadrightarrow M$ be a surjection from the free monoid on $X$ to $M$. Then* $\mathrm{Rat}(M) = \sigma(\mathrm{Rat}(X^*))$.

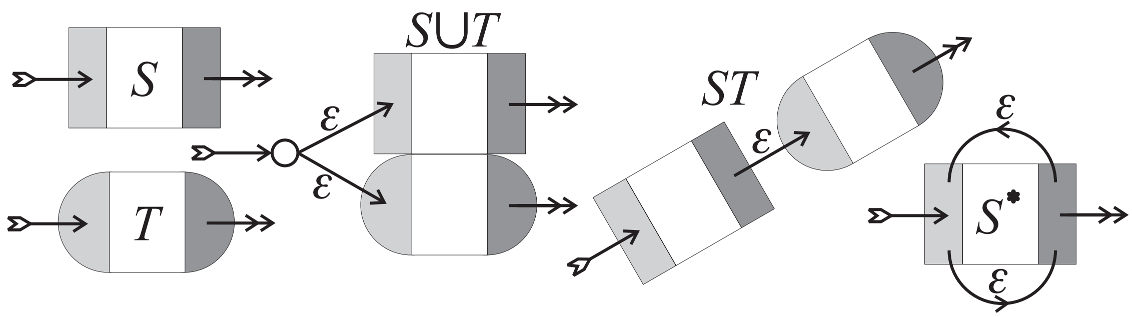



Figure 1.5: The effect of the rational operations on automata.

### 1.6.2 UR($M$)

The class of unambiguously rational subsets of a monoid was introduced by Eilenberg and Schützenberger [16]. It is analogous to the class of rational subsets except that all operations must be unambiguous. I.e., unions are only allowed between disjoint sets, products only between pairs of subsets in which no product can factor in two distinct ways, and monoid closures are only allowed on sets which are bases of free monoids:

**Definition 1.6.7** *Let $M$ be a monoid with two subsets $S, T$. The* **unambiguous rational operations** *are defined as follows. If $S \cap T = \emptyset$ then $S \Cup T = S \cup T$ is the* unambiguous union. *If for all $s, s' \in S$ and $t, t' \in T$ we have $st = s't'$ iff $s = s'$ and $t = t'$ then the* unambiguous product *is defined by $S \odot T = ST$. If for all $s_i, s'_i \in S$ we have $s_1 s_2 \cdots s_n = s'_1 s'_2 \cdots s'_m$ iff $n = m$ and $s_i = s'_i$ for all $i$ then the* unambiguous monoid closure *is defined by $S^{\circledast} = S^*$.*

Just as the rational sets were defined in terms of the rational operations, so are the unambiguous rational sets:

**Definition 1.6.8** *The collection of* **unambiguously rational subsets** *of a monoid $M$, denoted by* UR($M$)*, is the smallest collection of subsets which contains all finite subsets, and is closed under unambiguous rational operations.*

### 1.6.3 The relationship between Rec($M$) and Rat($M$)

An immediate corollary of lemma 1.6.6 is:

**Lemma 1.6.9** *If $M$ is finitely generated then* $\mathrm{Rec}(M) \subset \mathrm{Rat}(M)$.

The inclusion in proposition 1.6.9 is usually *proper*.

**1.6.10 Example** *Let $M = \mathbb{N} \times \mathbb{N}$. Then $\mathrm{Rat}(M) \not\subset \mathrm{Rec}(M)$. In fact, the set $S = \{(m,m) \mid m \in \mathbb{N}\}$ is rational, but not regular. $S$ is rational as $S = \{(1,1)\}^*$. Let $\Sigma = \{a,b\}$ with the surjection $\Sigma^* \to M$ defined by $a \mapsto (1,0), \quad b \mapsto (0,1)$. The pre-image of $S$ in $\Sigma^*$ is the set of words $w \in \Sigma^*$ such that the the number of $a$'s appearing in $w$ is the same as the number of $b$'s. Suppose this set were regular. Then intersecting with the regular language $\{a\}^*\{b\}^*$ we would conclude that the intersection $\{a^n b^n | n \in \mathbb{N}\}$ is regular, which a simple application of the pumping lemma (cf. [34]) shows to be false. Therefore, the pre-image of $S$ in $\Sigma^*$ is* not *regular, so that $S$ is not recognizable.*

Another useful fact is the following:

**Lemma 1.6.11** *If $A \in \mathrm{Rat}(M)$ and $B \in \mathrm{Rec}(M)$ then $A \cap B \in \mathrm{Rat}(M)$. In fact, supposing that $\sigma : X^* \twoheadrightarrow M$ is a surjective map with $\mathcal{A}$ and $\mathcal{B}$ automata such that $\sigma(\mathcal{A}) = A$ and $\mathcal{B} = \sigma^{-1}(B)$, then*

$$A \cap B = \sigma(\mathcal{A} \cap \mathcal{B}) \tag{1.6.1}$$

*which gives an algorithm linear in $|\mathcal{A}| \cdot |\mathcal{B}|$ for representing the intersection of a recognizable subset with a rational subset.*

Lemma 1.6.11 will be a specialization of theorem 1.7.3. See the proof of this theorem along with remark 1.7.4. This lemma also appears in [6, proposition III.2.6].

### 1.6.4 Bouquet automata

Let $X$ be a symmetric set of generators for a group $G$. Let $w$ be a string in $X^*$. The **word automaton** $\mathcal{W}(w)$ for $w = a_1 a_2 \cdots a_n$ is an automaton for the rational subset $\{\sigma(w)\}$ and which includes inverses of all edges. $\mathcal{W}(w)$ looks like:

$$\rightarrowtail v_0 \underset{a_1^{-1}}{\overset{a_1}{\rightleftharpoons}} v_1 \underset{a_2^{-1}}{\overset{a_2}{\rightleftharpoons}} v_2 \cdots v_{n-1} \underset{a_n^{-1}}{\overset{a_n}{\rightleftharpoons}} v_n \twoheadrightarrow \quad .$$

Consider a finite set of such words $T = \{w_1, w_2, \ldots, w_m\}$. The **bouquet** on $T$ is an automaton $\mathcal{B}(T)$ and is built as follows:

1. Construct word automata $\mathcal{W}(w_i)$ for all $i$.

2. Wedge all the $\mathcal{W}(w_i)$ together by identifying all the initial and accept states to a single state $*$; denote this union by $\mathcal{B}(T) = \bigvee_{i=1}^{m} \mathcal{W}(w_i)$.

3. Declare $*$ to be the unique initial and accept state of $\mathcal{B}(T)$.

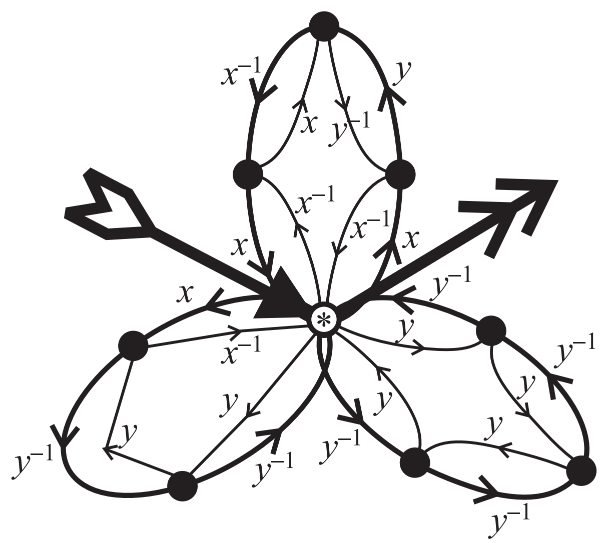



Figure 1.6: The bouquet $\mathcal{B}(T)$ on $T = \{xyx^{-1}x, xy^{-2}, y^{-4}\}$.

The collection of all bouquet automata is denoted by $\mathfrak{B}$. A language is called a **bouquet language** if it is accepted by a bouquet automaton. If $\mathcal{B}(T)$ is a bouquet automaton labeled by $X$, where $X$ is a set of generators for the group $G$, then the image of the language accepted by $\mathcal{B}(T)$ is the subgroup generated by $T$. Conversely, every finitely generated subgroup is the image of the language accepted by a bouquet automaton. Therefore:

**Lemma 1.6.12** *Let $G$ be a group. The collection of all homomorphic images in $G$ of bouquet languages is precisely the collection of finitely generated subgroups.*

## 1.7 $\mathcal{L}$-Regular Subsets

This section describes a re-interpretation of condition 2 of definition 1.4.2 in a way that generalizes conditions 2 and 3 of this definition.

**Definition 1.7.1** *Let $\sigma : X^* \twoheadrightarrow M$ and $\mathcal{L} \subseteq X^*$ be a regular language. $S \subseteq G$ is $\mathcal{L}$-**regular** iff $\sigma^{-1}(S) \cap \mathcal{L}$ is a regular language. In other words, a set is $\mathcal{L}$-regular when its pre-image in $\mathcal{L}$ is regular. The collection of all $\mathcal{L}$-regular subsets of $M$ is denoted by $\mathcal{L}\text{-Reg}(M)$.*

**1.7.2 Remark** *By the third condition of lemma 1.5.2, $\text{Rec}(M) = X^*\text{-Reg}(M)$. I.e., a set is recognizable if and only if its pre-image in $X^*$ is a regular language. On the other hand, $\sigma(\mathcal{L})$ is $\mathcal{L}$-regular for any regular language $\mathcal{L} \subseteq X^*$. Thus for any rational subset $S \in \text{Rat}(M)$ there is a regular language $\mathcal{L} \subseteq X^*$ such that $S \subset \sigma(\mathcal{L})$ and $S$ is $\mathcal{L}$-regular.*

The concept of $\mathcal{L}$-regularity (assuming that $\sigma(\mathcal{L}) = M$) was introduced by Gersten and Short [21].

The main observation of this section is the following:

**Theorem 1.7.3** *If $A \in \mathcal{L}\text{-Reg}(M)$ and $B \in \mathcal{L}'\text{-Reg}(M)$ then $A \cap B \in (\mathcal{L} \cap \mathcal{L}')\text{-Reg}(M)$. In fact, supposing that $\sigma : X^* \twoheadrightarrow M$ is onto and $\mathcal{A}$ and $\mathcal{B}$ are automata such that $\mathcal{A} = \mathcal{L} \cap \sigma^{-1}(A)$ and $\mathcal{B} = \mathcal{L}' \cap \sigma^{-1}(B)$, then*

$$\mathcal{A} \cap \mathcal{B} = (\mathcal{L} \cap \mathcal{L}') \cap \sigma^{-1}(A \cap B) \ . \tag{1.7.1}$$

*If in addition $\mathcal{L} \subset \mathcal{L}'$, and $A = \sigma(\mathcal{A})$ then*

$$\sigma(\mathcal{A} \cap \mathcal{B}) = A \cap B \tag{1.7.2}$$

*which gives an algorithm linear in $|\mathcal{A}| \cdot |\mathcal{B}|$ for representing the intersection of $A$ and $B$.*

*Proof.* Using the definitions of $\mathcal{A}$ and $\mathcal{B}$ above, we get

$$\mathcal{A} \cap \mathcal{B} = (\mathcal{L} \cap \sigma^{-1}(A)) \cap (\mathcal{L}' \cap \sigma^{-1}(B)) = (\mathcal{L} \cap \mathcal{L}') \cap \sigma^{-1}(A \cap B)$$

which shows that $A \cap B$ is $(\mathcal{L} \cap \mathcal{L}')$-regular and verifies equation (1.7.1). Furthermore, applying $\sigma$ to this equation shows that $\sigma(\mathcal{A} \cap \mathcal{B}) \subseteq A \cap B$. Finally suppose that $\mathcal{L} \subseteq \mathcal{L}'$. Let $c \in A \cap B$. Because $\mathcal{A} = \mathcal{L} \cap \sigma^{-1}(A)$ and $\sigma(\mathcal{A}) = A$, a word $w \in \mathcal{L}$ can be found such that

$\sigma(w) = c$. As $\mathcal{L} \subseteq \mathcal{L}'$, $c \in B$ and $\mathcal{B} = \mathcal{L}' \cap \sigma^{-1}(B)$, it follows that $w \in \mathcal{A} \cap \mathcal{B}$. Therefore, $c \in \sigma(\mathcal{A} \cap \mathcal{B})$ which completes equation (1.7.2). ♦

**1.7.4 Remark** *Theorem 1.7.3 specializes to give lemma 1.6.11. Indeed, in the statement of lemma 1.6.11 we have $A$ rational, $\mathcal{A}$ a regular language such that $A = \sigma(\mathcal{A})$, and $B$ recognizable with total pre-image $\sigma^{-1}(B) = \mathcal{B}$. Setting $\mathcal{L} = \mathcal{A}$ and $\mathcal{L}' = X^*$ we have that $A$ is $\mathcal{L}$-regular, $B$ is $\mathcal{L}'$-regular, $\sigma(\mathcal{L}) = A$ and $\mathcal{L} \subseteq \mathcal{L}'$ which satisfy the hypotheses necessary for equation* (1.7.2)*. This equation gives $\sigma(\mathcal{A} \cap \mathcal{B}) = A \cap B$ which validates lemma 1.6.11.*

## 1.8 Formal Languages

Regular languages were touched upon in sections 1.3 and 1.4. Definition 1.4.2 was given which gave three possible ways to define regularity inside free monoids. There are many other ways to define this notion, and below we shall see a way of defining regular languages by means of a grammar. Other language classes that we'll visit upon are context free, context sensitive, and recursively enumerable languages. Although most of the language classes considered may be defined as languages accepted by certain types of machines, and in other ways, below I sketch only the grammar theoretic definitions.

The material in this section is a variation on the theme of Gilman's [25, §9]. In addition to defining the notion of a grammar, [25, §9] contains an elegant proof of the fact that context free languages are precisely the languages accepted by pushdown automata. The standard reference on formal languages and automata theory is Hopcroft and Ullman's book [34]. Other recommended books and surveys are [6], [13], [15], [19], [27], [31], [35] and [47].

### 1.8.1 Defining languages by means of a grammar

A grammar[5] consists of a set of syntactic rules by which a language is generated. The word "grammar" is chosen because of its usual sense in the realm of natural languages. Consider the following over-simplified example. A set of nouns for our language

[5]By "grammar" I mean "phrase structure grammar" in the sense of [34, p. 230]. Other types of grammars exist, such as indexed grammars [34, §14.3], which do not adhere to the conventions of phrase structure grammars.

is $\{\text{Tim}, \text{John}, \text{I}, \text{We}, \text{People}\}$ and a set of verbs is $\{\text{\_go.}, \text{\_goes.}, \text{\_eat.}, \text{\_eats.}\}$. We add a symbol "a" which can be replaced by either of $\{\text{Tim}, \text{John}\}$ and a symbol "b" that can be replaced by $\{\text{I}, \text{We}, \text{People}\}$. Also add a symbol "c" which is replaceable by $\{\text{\_goes.}, \text{\_eats.}\}$ and a symbol "d" for which $\{\text{\_go.}, \text{\_eat.}\}$ may be substituted. Let "s" be a symbol standing for "sentence" or "start" which may be replaced either by ac or by bd. This whole situation is encapsulated by specifying the relation "$\rightrightarrows$" with the following **production rules**:

$$
\begin{array}{lcl}
\mathsf{s} & \rightrightarrows & \{\mathsf{ac}, \mathsf{bd}\} \\
\mathsf{a} & \rightrightarrows & \{\text{Tim}, \text{John}\} \\
\mathsf{b} & \rightrightarrows & \{\text{I}, \text{We}, \text{People}\} \\
\mathsf{c} & \rightrightarrows & \{\text{\_goes.}, \text{\_eats.}\} \\
\mathsf{d} & \rightrightarrows & \{\text{\_go.}, \text{\_eat.}\}
\end{array}
$$

Figure 1.7: A context free grammar for a simple language of English sentences.

The production system "$\rightrightarrows$" is a finite relation on the strings formed by using letters in the Roman alphabet, union the Sans Serif alphabet, union punctuation marks and the underscore symbol (the underscore is used to make spaces visible). In fact, letting $A$ be the alphabet

$$A = \{\text{I}, \text{J}, \text{T}, \text{W}, \text{P}, \text{i}, \text{m}, \text{o}, \text{h}, \text{n}, \text{e}, \text{p}, \text{l}, \text{g}, \text{s}, \text{a}, \text{t}, \mathsf{s}, \mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \ . \ , \_\}$$

the production system $\rightrightarrows$ is a finite relation on $A^+$. Furthermore, one can expand $\rightrightarrows$ by multiplying it from the left and right:

$$\rightsquigarrow \quad = \quad \{(u,v) \in A^+ \times A^+ \mid \exists\ \alpha, \beta, u', v' \in A^* \text{ s.t. } u = \alpha u' \beta, v = \alpha v' \beta, \text{ and } u' \rightrightarrows v'\} \tag{1.8.1}$$

which is called the set of one-step **derivations**. The set of $n$-step derivations is given by $\rightsquigarrow^n$ while the reflexive transitive closure

$$\rightsquigarrow^* \quad = \quad \bigcup_{n=0}^{\infty} \rightsquigarrow^n$$

is called the set of full derivations. The language determined by a grammar consists of all the words in a certain sub-alphabet, called the **terminal alphabet**, which are derivable from the start letter $\mathsf{s}$. In our case, the terminal alphabet is the set of all Roman letters appearing in $A$ along with the period and underscore:

$$X = \{\mathrm{I}, \mathrm{J}, \mathrm{T}, \mathrm{W}, \mathrm{P}, \mathrm{i}, \mathrm{m}, \mathrm{o}, \mathrm{h}, \mathrm{n}, \mathrm{e}, \mathrm{p}, \mathrm{l}, \mathrm{g}, \mathrm{s}, \mathrm{a}, \mathrm{t}, \ . \ , \_\} \ .$$

The accepted language $\mathcal{L} = (\mathsf{s} \rightsquigarrow^*) \bigcap X^+$ consists of all the syntactically correct 2 word sentences which have one of the above nouns as the first word, and one of the above verbs as the second word, e.g. "I_eat." or "Tim_goes.", etc.

Summarizing:

**Definition 1.8.1** *A grammar is defined by specifying an alphabet $A$, a terminal alphabet $X \subseteq A$, a* **variable alphabet** *$V = A - X$, a start letter $\mathsf{s} \in V$ and a finite production relation $\rightrightarrows \subseteq A^+ \times A^+$. The one-step derivation relation $\rightsquigarrow \subseteq A^+ \times A^+$ is defined by equation* (1.8.1)*, while the full derivation relation is the reflexive transitive closure $\rightsquigarrow^*$. The language defined by the grammar is $(\mathsf{s} \rightsquigarrow^*) \bigcap X^+$, i.e., the set of all words derivable from $\mathsf{s}$ which are purely composed of terminal letters. The grammar, as well as its language, are denoted by $\mathcal{L}$.*

**1.8.2 Remark** *Notice that only productions between non-empty words are allowed. Consequently, all languages generated by grammars must be $\varepsilon$-free. However, usually it will be necessary to consider languages containing $\varepsilon$. Therefore, a language $\mathcal{L}$ is considered to be generated by a certain grammar if and only if $\mathcal{L} - \{\varepsilon\}$ is generated by that grammar. This has the negative consequence that the language generated by a grammar is ambiguous. A slight variant of the above definition would permit one special production of the form $\mathsf{s} \rightrightarrows \varepsilon$ thus allowing for generation of the empty word $\varepsilon$. I am opting to disallow this production because I want to quote the definition of Buntrock and Loryś for growing grammars without any modifications (cf. §1.8.4 below).*

*Letters in the variable alphabet $V$ will be in the* Sans Serif *font, while letters in the terminal alphabet $X$ will be in Roman. The letter $X$ was chosen to denote a terminal alphabet because often the terminal alphabet will also be the alphabet where words for a certain group reside. I.e., $X$ will usually be a set of monoid generators for a group $G$.*

Now let's define a sequence of restrictions on a grammar which give rise to the recursively enumerable, context sensitive, growing, growing and terminating, context free, and regular classes of languages.

### 1.8.2 Recursively enumerable languages

Grammars as defined in 1.8.1 are wholly "unrestricted" or of "type-0". Any grammar gives rise to a recursively enumerable language[6], as it is possible to list all derived words and check for membership in $X^*$. On the other hand, type-0 grammars are powerful enough to simulate all Turing machines. Indeed this model of type-0 grammar is less restrictive than that presented in Salomaa's book [47, p. 15] and Salomaa's type-0 grammar is already strong enough to generate all recursively enumerable languages. Therefore, the languages obtained from unrestricted grammars are precisely the recursively enumerable languages.

### 1.8.3 Context sensitive languages

Context sensitive grammars, also called "type-1" satisfy:

$$|\alpha| \leq |\beta| \quad \text{for all productions } \alpha \rightrightarrows \beta \ . \tag{1.8.2}$$

Then $\mathcal{L}$ is called a **context sensitive** language. So the context sensitive grammars are grammars with productions which do not shrink from left to right.

### 1.8.4 Growing languages

Next we give a definition of Buntrock and Loryś for a **growing** grammar [8, definition 1]. A grammar is growing if it is context sensitive, the start letter $\mathsf{s}$ never appears on the right hand side of any production and all productions which don't have $\mathsf{s}$ on the left hand side are strictly length increasing:

$$\alpha \rightrightarrows \beta \text{ and } \alpha \neq \mathsf{s} \quad \Longrightarrow \quad |\alpha| < |\beta|. \tag{1.8.3}$$

Then $\mathcal{L}$ is called a growing language. As growing languages are essentially context sensitive with strict inequalities, one can think of such grammars as "type-$(1+\epsilon)$".

[6] A **recursively enumerable** language is a language for which an algorithm exists which lists only words in the language, and —given enough time— will eventually list any particular word. Often, one defines a language to be recursively enumerable if there is a Turing machine which halts exactly when a word in the language is given to the machine as input. More on recursively enumerable languages can be found in [34, §7.3].

### 1.8.5 Growing and terminating languages

The next definition is non-standard. Admittedly, the definition has been created in order to give a language theoretic characterization of word hyperbolic groups. Nevertheless, the definition is a natural restriction of growing grammars using a context free like condition of termination. In fact, we will see in the next section that all context free languages may be generated by a terminating context free grammar (proposition 1.8.4). Thus one may think of growing and terminating grammars as "type-$1\frac{1}{2}$" existing in between context sensitive and context free grammars. The terminating condition is independent of the growing condition so we state the terminating condition separately:

**Definition 1.8.3** *A grammar as in definition 1.8.1 is said to be* **terminating** *if given any variable letter $\mathsf{v} \in V$, there is a production of the form $\mathsf{v} \rightrightarrows \beta$ with $\beta \in X^*$ a terminal word.*

So as may be expected, a language is **growing and terminating** if it is generated by a growing grammar which is also terminating.

### 1.8.6 Context free languages

A grammar is **context free**, or "type-2" if all productions have variable letters as their left hand sides:

$$\alpha \rightrightarrows \beta \quad \Longrightarrow \quad \alpha \in V \, . \tag{1.8.4}$$

A language is context free if it is generated by a context free grammar. Context free grammars are so named because the derivations do not depend on the context around a particular variable. As by assumption grammars do not have $\varepsilon$ on the right hand side of a production, context free grammars are necessarily context sensitive. In fact:

**Proposition 1.8.4** *Suppose $\mathcal{L}$ is a context free language. Then $\mathcal{L}$ is generated by a context free grammar which is also growing and terminating.*

*Proof.* Consider the grammar $\rightrightarrows$. Without loss of generality we may assume that $\rightrightarrows$ is an acyclic relation[7] when restricted to the alphabet $X$: We will modify $\rightrightarrows$ so as to make

[7] A relation $r$ on $A$ is **acyclic** if for all $a, b \in A$, if $a$ is $r$-connected to $b$, then $b$ is *not* $r$-connected to $a$, i.e., $ar^+b$ implies that $\neg(br^+a)$. In other words, $r^+$ is irreflexive.

it acyclic on $X$. Restricting the productions to the alphabet $A$, we obtain a partial order on the strongly connected components[8] of $\rightrightarrows|_A$. Notice that from any strongly connected component, a single representative variable may be chosen while dropping all other variables in the component, without changing the generated language. For example, suppose that $\mathsf{u} \rightrightarrows^* \mathsf{v}$ and $\mathsf{v} \rightrightarrows^* \mathsf{u}$. Then we may eliminate $\mathsf{v}$ by simply replacing it by $\mathsf{u}$ wherever it appears. Finally, all productions of the form $\mathsf{v} \rightrightarrows \mathsf{v}$ may be deleted resulting in a grammar which is acyclic on $A$.

Thus we may view $\rightrightarrows$ as defining a directed forest on $A$ —i.e. an acyclic graph. Now we aim to get rid of as many productions as possible between single letters. We do this by going backwards along the forest starting from its leaves. A leaf of the forest is a letter which does not produce another letter. Show that all letters which are not roots[9] —i.e. are produced by other letters— can be made to be leaves, without changing the generated language. By induction, it is enough to show that any letter which is not a root but which only produces letters which are leaves can be turned into a leaf. Let $\mathsf{v}$ be such a letter. Suppose that $\mathsf{v} \rightrightarrows \mathsf{w}$ and $\mathsf{u} \rightrightarrows \mathsf{v}$. Replace $\mathsf{v} \rightrightarrows \mathsf{w}$ by $\mathsf{u} \rightrightarrows \mathsf{w}$. Furthermore, for any $\mathsf{a} \in V$ and any production of the form $\mathsf{a} \rightrightarrows \alpha\mathsf{v}\beta$ add a production $\mathsf{a} \rightrightarrows \alpha\mathsf{w}\beta$. Now repeat the last step until no new such productions can be added. This guarantees that any derivation that involved a production of the form $\mathsf{v} \rightrightarrows \mathsf{w}$ can be achieved without this production: Since the grammar is context free, the appearance of $\mathsf{v}$ must have occurred from some production of the form $\mathsf{a} \rightrightarrows \alpha\mathsf{v}\beta$ and the changing of $\mathsf{v}$ to $\mathsf{w}$ could have been achieved by changing $\mathsf{a} \rightrightarrows \alpha\mathsf{v}\beta$ to $\mathsf{a} \rightrightarrows \alpha\mathsf{w}\beta$. Thus $\mathsf{v}$ has been turned into a leaf and we may assume that $\rightrightarrows|_A$ is a forest consisting only of roots and leaves.

Next we show that all roots other than $\mathsf{s}$ may be turned into leaves. Consider a root $\mathsf{v} \neq \mathsf{s}$ which is not a leaf. Since $\mathsf{v}$ is not a leaf, $\mathsf{s}$ is not $\rightrightarrows$-connected to $\mathsf{v}$. Therefore, any terminal words derived by the grammar will not pass through $\mathsf{v}$ when starting at $\mathsf{s}$. Consequently, there is no harm in dropping all productions of the form $\mathsf{v} \rightrightarrows a$ with $a \in A$ as long as for all productions of the form $\mathsf{u} \rightrightarrows \alpha\mathsf{v}\beta$ a production of the form $\mathsf{u} \rightrightarrows \alpha a\beta$ is added, and this process is repeated until no such new productions can be added. Thus we may

[8] A subset $S \subseteq A$ is an **$r$-strongly connected component** if for all $s, t \in S$ we have $sr^*t$ and $S$ is maximal with respect to this property.

[9] A **leaf** is a letter $\mathsf{v} \in A$ such that there is no $\mathsf{w} \in A$ with $\mathsf{v} \rightrightarrows \mathsf{w}$. A **root** $\mathsf{v} \in A$ is a letter such that there is no letter $\mathsf{u} \in A$ with $\mathsf{u} \rightrightarrows \mathsf{v}$.

assume that the only possible non-leaf in $\rightrightarrows|_A$ is $\mathsf{s}$. But this means that all productions which do not have $\mathsf{s}$ for their left hand sides are strictly length increasing. This is almost a growing grammar, except that $\mathsf{s}$ may appear on right hand sides of productions.

Next, we may assume that if $\mathsf{s} \rightrightarrows a$, then $a$ is a terminal. For if $\mathsf{s} \rightrightarrows \mathsf{v}$ where $\mathsf{v}$ is a variable, we drop this production while adding a production $\mathsf{s} \rightrightarrows \alpha$ for any production $\mathsf{v} \rightrightarrows \alpha$. As $\mathsf{v}$ must be a leaf, $\alpha$ is not a single letter so that no new edges have been introduced to $\rightrightarrows|_A$.

To get rid of $\mathsf{s}$ on right hand sides of productions, introduce a new variable $\mathsf{s}'$. Change any appearance on the right hand side of a production of $\mathsf{s}$ to $\mathsf{s}'$. Next, for any production of the form $\mathsf{v} \rightrightarrows \alpha \mathsf{s}' \beta$ and any production $\mathsf{s} \rightrightarrows x$ add a production $\mathsf{v} \rightrightarrows \alpha x \beta$. Repeat this process until no new productions of this form may be added. Furthermore, for any production of the form $\mathsf{s} \rightrightarrows \alpha$ with $\alpha \notin A$ add a production $\mathsf{s}' \rightrightarrows \alpha$. Finally, add the production $\mathsf{s} \rightrightarrows \mathsf{s}'$. This grammar is a growing context free grammar.

Finally, we must arrange our grammar to be terminating. Consider a variable $\mathsf{v}$. Suppose that it is not the case that $\mathsf{v} \rightsquigarrow^* X^+$. As the grammar is context free, there is no way to ever derive a terminal word from a word in which $\mathsf{v}$ appears. Consequently, removing such a $\mathsf{v}$ from our alphabet $A$ does not change the language generated by the grammar. So without loss of generality, for all variables $\mathsf{v}$ there is a string $\alpha_{\mathsf{v}} \in X^+$ such that $\mathsf{v} \rightsquigarrow^* \alpha_{\mathsf{v}}$. The language generated by our grammar will not change if we simply add the productions $\mathsf{v} \rightrightarrows \alpha_{\mathsf{v}}$ to it, so that we may assume that the grammar is terminating. Finally, for $\mathsf{v} \neq \mathsf{s}$ we can assume that $\alpha_v \notin X$, as $\mathsf{v}$ is a leaf, showing that adding the terminating productions does not disturb the growing property of the grammar. This completes the proof. ⧫

### 1.8.7 Grammars for regular languages

For completeness we provide a grammar theoretic description of regular languages. A context free grammar is termed "right linear" (or "type-3") if all productions are of the form $\mathsf{u} \rightrightarrows \alpha \mathsf{v}$ or $\mathsf{u} \rightrightarrows \beta$ where $\alpha \in X^*$ and $\beta \in X^+$ are terminal words. The grammar is also called "regular" because the languages generated by such grammars are precisely the regular languages. A regular grammar defines an automaton labeled in $X^*$ by taking the vertices to be the variables plus a single accept vertex $t$, taking the start vertex to be $\mathsf{s}$ and defining labeled edges of the form $\mathsf{u} \xrightarrow{\alpha} \mathsf{v}$ for every production $\mathsf{u} \rightrightarrows \alpha \mathsf{v}$, and $\mathsf{u} \xrightarrow{\beta} t$ for every production $\mathsf{u} \rightrightarrows \beta$. This automaton accepts the language generated by the grammar. On the other hand, any automaton can be changed to one without $\varepsilon$-edges, with a unique start state with no edges pointing towards it, and with a unique accept state which does not have any outward edges; this automaton can be turned into a regular grammar in a manner analogous to the above. Thus the class of regular languages coincides with the class generated by regular grammars.

## 1.9 The Group Theoretic Angle

The Gersten-Stallings angle first appeared in [51] and was used to give a non-positive curvature condition on a triangle of groups which, for example, guarantees that the vertex groups inject in the colimit of the triangle.

### 1.9.1 A definition of Gersten and Stallings

Let $G$ be a group with subgroups $A$, $B$ and $C \leq A \cap B$.

**Definition 1.9.1** *An* **alternating word** *in $A$ and $B$ over $C$ is either the empty word, an element of $C$ or a word of the form*

$$w = g_1 g_2 \dots g_n \tag{1.9.1}$$

*where $g_i \in A \cup B - C$ alternate between $A$ and $B$. If $w$ is the empty word or an element of $C$, the* **alternating length** *of $w$ is 0. If $w$ is a word given by equation 1.9.1 then its alternating length is $n$.*

The group theoretic angle is defined by considering all alternating words in $A$ and $B$ over $C$ which represent the identity in $G$.

**Definition 1.9.2** *Let $W$ be the set of all alternating words $w$ of positive alternating length which are trivial in $G$. The* **generalized angle** *between $A$ and $B$ over $C$ is defined by*

$$\sphericalangle(A, B; C) = \begin{cases} 0 & \text{if } W = \emptyset. \text{ Otherwise,} \\ \frac{2\pi}{n} & \text{where } n \text{ is the minimum alternating length of words in } W. \end{cases} \tag{1.9.2}$$

*The* **standard angle** *between $A$ and $B$ is the generalized angle with $C = A \cap B$:*

$$\sphericalangle_B^A = \sphericalangle(A, B; A \cap B). \tag{1.9.3}$$

**Lemma 1.9.3** *If $C \lneq A \cap B$ then $\sphericalangle(A, B; C) = \pi$. The only positive angles possible are integral fractions of $\pi$. If $C = A \cap B$ then the smallest possible positive angle is $\frac{\pi}{2}$.*

*Proof.* Let $W$ be the set of all trivial alternating words of positive length as in definition 1.9.2.

Suppose that $C \lneq A \cap B$. Find $a \in A \cap B - C$. Let $b = a$. Then according to definition 1.9.1 $a \cdot b^{-1}$ is an alternating word of length 2 which is equal to 1 in $G$. The set $W$ contains $a \cdot b^{-1}$ so that the angle is non-zero. By definition, no alternating words of unit length can be trivial implying that $a \cdot b^{-1}$ is of minimal alternating length in $W$ and $\sphericalangle(A, B; C) = \frac{2\pi}{2} = \pi$.

Next, show that the only possible positive angles are integral fractions of $\pi$; i.e., the minimal alternating length of elements in $W$ is an even number. By the previous paragraph, we may assume that $C = A \cap B$ and that any element of $W$ has alternating length greater than 2. Let $w \in W$ have minimal alternating length $n$. Suppose that $n$ is odd. Without loss of generality

$$w = g_1 g_2 \cdots g_n$$

with both $g_1$ and $g_n$ elements of $A - A \cap B$. Conjugating by $g_2$ produces the word

$$w' = (g_n g_1) g_2 \cdots g_{n-1} \ .$$

$w'$ is equal to 1 in $G$. Furthermore, if $g_n g_1 \notin A \cap B$ then $w'$ is alternating with length $n - 1$, contradicting the minimality of $n$. Otherwise, $g_n g_1 \in A \cap B$ and we set

$$w'' = (g_n g_1 g_2) g_3 \cdots g_{n-1}.$$

$w''$ is then alternating with length $n - 2$ and trivial in $G$, which contradicts the minimality of $n$.

To show that the smallest possible angle when $C = A \cap B$ is $\frac{\pi}{2}$ it suffices to show that $\pi$ does not occur as an angle. Suppose it did. Then there are elements $a \in A - A \cap B$ and $b \in B - A \cap B$ such that $ab^{-1} = 1$. This is impossible. ♦

**Lemma 1.9.4** *Let $C = A \cap B$. Let $\phi : A *_{A\cap B} B \to G$ be the natural map given by the inclusions $A < G$ and $B < G$. Then $\sphericalangle(A, B; C) = 0$ if and only if $\phi$ is injective. In particular, if $A < B$ then $\sphericalangle(A, B; A) = 0$.*

*Proof.* By the reduced form theorem for amalgamated products (see for example [9, p. 32]) the non-trivial elements of $\ker\phi$ give non-empty alternating words representing 1 in $G$, and vice-versa. The last statement is a consequence of the fact that $A *_A B = B$. ♦

### 1.9.2 The relationship between angles and Rat$(G)$

The Gilman–Stallings observation [52] is that angles can also be defined as follows:

**Lemma 1.9.5** *Let $G$ be a group with subgroups $A$ and $B$.*

$$\sphericalangle_B^A = \begin{cases} 0 & \text{if } 1 \notin \bigcup_{n=1}^{\infty}[(A-B)(B-A)]^n. \text{ Otherwise,} \\ \frac{\pi}{n} & \text{where } n \text{ is the smallest positive number s.t. } 1 \in [(A-B)(B-A)]^n \end{cases}.$$

*Proof.* Throughout the proof we implicitly use the fact that $A - B = A - A \cap B$ and $B - A = B - A \cap B$.

Let $N$ be the minimum alternating length of elements of $\ker(A *_{A\cap B} B \to G)$ which are non-trivial. If $N$ is well defined then $N$ is even and by conjugating we can find a kernel element $w = g_1 \cdots g_N$ with the $g_i \in A - B$ for $i$ odd, $g_i \in B - A$ for $i$ even, and $N \geq 2$ minimal. The existence of $w$ shows that $1 \in [(A-B)(B-A)]^{\frac{N}{2}}$. Furthermore, the minimality of $N$ verifies the second case of $\sphericalangle_B^A$. On the other hand, if the minimum $N$ is undefined, $\sphericalangle_B^A = 0$ and also $1 \notin \bigcup_{n=1}^{\infty}[(A-B)(B-A)]^n$, which gives the first case of $\sphericalangle_B^A$. ♦

**Proposition 1.9.6** *Suppose that* Rat$(G)$ *is closed under complements. Let $A$ and $B$ be finitely generated subgroups of $G$. Then $[(A-B)(B-A)]^n$ and $\bigcup_{i=1}^{\infty}[(A-B)(B-A)]^i$ are rational subsets for all $n \in \mathbb{N} \cup \{0\}$.*

*Proof.* The fact that $A - B = \neg((\neg A) \cup B)$ and the closure of Rat$(G)$ under unions, complements and products show that $[(A-B)(B-A)]^n$ is regular. The last part follows from the formula

$$\bigcup_{i=1}^{\infty}[(A-B)(B-A)]^i = [(A-B)(B-A)]^+$$

and the fact that semigroup closure is a rational operation since $S^+ = S^* \cdot S$. ♦

# Chapter 2

# Algorithms for Finitely Presented Groups

## 2.1 Group Theoretic Decision Problems

### 2.1.1 Conventions for group theoretic algorithmic problems

Algorithmic problems will be named using the Sans Serif font and will follow certain uniform conventions described below by example. The **word problem** for a group $G$ is the problem of deciding whether a word in the generators of $G$ represents the identity. It was first formulated by Max Dehn in [11]. To properly define the word problem one needs to specify a presentation for $G$. Nevertheless, for $G$ finitely presented, the existence of a recursive[1] solution to the word problem does not depend on the finite presentation chosen. Indeed, given two finite presentations there is a linear time algorithm that translates any word from one set of generators to the other, and vice-versa. So if the word problem has optimal complexity function $f(x)$ relative the first presentation and optimal complexity function $g(x)$ relative the second presentation, it must be the case that $f$ and $g$ are linearly equivalent (see definition 1.2.1). Therefore, the complexity class of the word problem is invariant under finite presentation chosen, and the notation

$$\mathsf{Word}(G)$$

[1]A **recursive** algorithm is a procedure which can be programmed (at least theoretically) on a computer. That is, there is a computer program which given any input for the problem, halts and spits out the correct solution. The terms **recursively solvable**, **computable**, **solvable**, etc. are all synonyms of "recursive".

is well defined. The same convention will be used in problems whose complexity is independent of presentation.

The above justifies talking about the word problem "in $G$"; nevertheless, in making the precise definition of $\mathsf{Word}(G)$ we must make use of a specific presentation. Let's see how $\mathsf{Word}(G)$ is formally defined:

**Decision problem 1** ($\mathsf{Word}(G)$)
*GIVEN: A finite presentation $P = \langle\, Y \mid R \,\rangle$ for the group $G$.*
*INPUT: A word $w \in (Y \cup Y^{-1})^*$.*
*DECIDE: Is $w$ equal to 1 in $G$?*

This illustrates the format in which a decision problem will be given. The GIVEN denotes information that a programmer solving the problem can make use of in building the program, before any input is considered. For example, suppose that in the solution to $\mathsf{Word}(G)$ a special automaton needs to be created that can read words and decide if they are trivial. It may be that building such an automaton requires an enormous amount of time and space. However, since such an automaton need only be built once, it does not affect complexity considerations. Furthermore, we may assume that special information —such as the availability of a solution to the problem by means of an automaton— has been made use of. For example, even though there is no general algorithm for deciding whether or not a finite presentation defines a hyperbolic group, when $G$ is hyperbolic it will be assumed that this information is available.

**2.1.1 Example** *Consider the group of the $3\times 3\times 3$ Rubik's cube, denoted by* $\mathrm{Rubik}_3$. *This group is the set of all possible symmetries obtainable by legal moves of the cube, modulo rigid motions of the whole cube. That is, letting $a, b, c, d, e$ and $f$ be the clockwise twists*[2] *of each of the cubes six faces,* $\mathrm{Rubik}_3$ *is identifiable —as a set– with the set of all possible configurations of the cube which are obtainable by sequences of twists in $\{a, b, c, d, e, f\}^*$. Inverses of generators have also been included in figure 2.1, though they are unnecessary in the description of this group, as for example $a^3 = a^{-1}$ in* $\mathrm{Rubik}_3$. *The group is huge. Indeed,*

$$|\mathrm{Rubik}_3| = 8! \cdot 3^7 \cdot 12! \cdot 2^{10} = 43,252,003,233,529,856,000\ .$$

[2]For example, in figure 2.1 $f$ is the clockwise twist of the front face (shaded lightest).

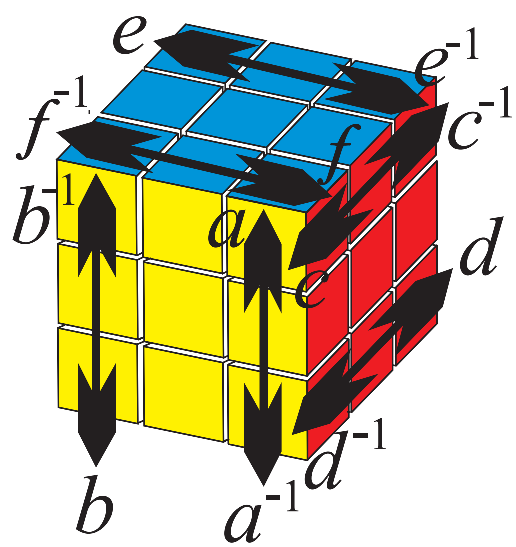



Figure 2.1: A clever data structure for solving $\mathsf{Word}(\mathrm{Rubik}_3)$.

*Yet its word problem is easily solved: Given a word $w$ on the generators of* $\mathrm{Rubik}_3$*, starting with a pristine cube (as in figure 2.1) simply view $w$ as a sequence of twists. After twisting the cube as instructed by $w$ check to see if the cube is returned to its pristine form. This is the case, if and only if $w = 1$ in* $\mathrm{Rubik}_3$*. To learn more about the group theoretic properties of Rubik's cube [44, §19] is recommended.*

The INPUT specifies a set of data on which the problem is defined. The time and space complexity of an algorithm is defined in terms of the input size. For example, if given a word $w$ our automaton is guaranteed to stop after $4|w|^2$ elementary steps then $f(n) = 4n^2$ is a time complexity function for $\mathsf{Word}(G)$. Decision problems will always have the same boolean output set $\mathbb{B} = \{$ "No", "Yes" $\} = \{0, 1\}$, and the predicate being determined is given after "DECIDE". More general computations will have other output types. For example, in computing the group theoretic angle the output set is the set of all possible angles $\{0, \pi, \frac{\pi}{2}, \frac{\pi}{3}, \ldots\}$:

**Computational problem 1** ($\mathsf{Angle}(G)$)

*GIVEN: A finite presentation $P = \langle\ Y \mid R\ \rangle$ for the group $G$.*

*INPUT: A sequence of words $a_1, a_2, \ldots, a_l, b_1, b_2, \ldots b_m \in (Y \cup Y^{-1})^*$ which generate subgroups $A = \langle a_1, a_2, \ldots, a_l\rangle$ and $B = \langle b_1, b_2, \ldots b_m\rangle$ of $G$.*

*COMPUTE: The standard angle* $\sphericalangle_B^A$.

Often, an algorithmic problem has a number of variants. In particular, it may be useful to consider a generalized version of a particular problem. For example, in the **generalized word problem** one is asked to determine whether a given word in the generators of $G$ is spanned by other given words. This generalizes the usual word problem which asks if a given word is spanned by the trivial word. The prefix "Gen" is used to denote the generalization of a particular problem.

**Decision problem 2** ($\mathsf{GenWord}(G)$)
*GIVEN: A finite presentation* $P = \langle\ Y \mid R\ \rangle$ *for the group* $G$.
*INPUT: A word* $w \in (Y \cup Y^{-1})^*$ *and a sequence of words* $a_1, a_2, \dots, a_n \in (Y \cup Y^{-1})^*$ *which generate a subgroup* $A = \langle a_1, a_2, \dots, a_n\rangle$ *of* $G$.
*DECIDE: Does* $w$ *represent an element of* $A$ *in* $G$*?*

**2.1.2 Remark** *On the other hand, the "*Gen*" prefix is used in a very specific way. For example, computing the generalized angle will* not *be denoted by* GenAngle*. Instead,* Angle′ *will be used:*

**Computational problem 2** ($\mathsf{Angle}'(G)$)
*GIVEN: A finite presentation* $P = \langle\ Y \mid R\ \rangle$ *for the group* $G$.
*INPUT: A sequence of words* $a_1, a_2, \dots, a_l, b_1, b_2, \dots b_m, c_1, c_2, \dots, c_n \in (Y \cup Y^{-1})^*$ *which generate subgroups* $A = \langle a_1, a_2, \dots, a_l\rangle, B = \langle b_1, b_2, \dots b_m\rangle$ *and* $C = \langle c_1, c_2, \dots, c_n\rangle$ *of* $G$ *such that* $C < A \cap B$.
*COMPUTE: The generalized angle* $\sphericalangle(A, B; C)$.

The use of the "Gen" prefix is excluded in this case because $\mathsf{Angle}(G)$ is not a specific case of $\mathsf{Angle}'(G)$. In particular, in $\mathsf{Angle}'(G)$ the input consists of a *finite* number of generators for $A, B$, and $C$; on the other hand, $\mathsf{Angle}(G)$ takes as its input generators for $A$ and $B$. To apply an algorithm for $\mathsf{Angle}'(G)$ to solve $\mathsf{Angle}(G)$ —as defined— would require a method of finding a finite set of generators for $C = A \cap B$ starting only from generators of $A$ and $B$. In particular, $G$ would have to satisfy the Howson property[3] which is certainly not the case in general, even for hyperbolic groups (cf. [46]).

[3] A group $G$ is said to have the **Howson property** if given any finitely generated subgroups $A, B < G$, $A \cap B$ is also finitely generated.

It is often useful to modify the input or output of a particular problem in order to obtain a new problem. This is denoted haphazardly by the restriction notation. For example, restricting to the case that $A = \{1\}$ the generalized word problem becomes the usual word problem and this is denoted by

$$\mathsf{GenWord}(G)|_{A=\{1\}} = \mathsf{Word}(G) \ .$$

By restricting the output, a computational problem can turn into a decision problem. For example the "positive angle problem" requires deciding whether or not the angle between two finitely generated subgroups is positive. The positive angle problem is obtained from the angle problem by "restricting" the original output set $\{0, \pi, \frac{\pi}{2}, \frac{\pi}{3}, \dots\}$ to the set $\mathbb{B}$ by keeping the 0 output unchanged, while changing any positive output to the boolean truth value 1. Therefore,

$$\mathsf{PosAngle}(G) = \mathsf{Angle}(G)|_{\mathbb{B}} \ .$$

For completeness, a formal definition of the positive angle problem is given:

**Decision problem 3** ($\mathsf{PosAngle}(G)$)
*GIVEN:* *A finite presentation* $P = \langle\ Y \mid R\ \rangle$ *for the group* $G$.
*INPUT:* *A sequence of words* $a_1, a_2, \dots, a_m, b_1, b_2, \cdots, b_n \in (Y \cup Y^{-1})^*$ *which generate subgroups* $A = \langle a_1, a_2, \dots, a_n\rangle$ *and* $B = \langle b_1, b_2, \dots, b_n\rangle$ *of* $G$.
*DECIDE:* *Is* $\sphericalangle^A_B > 0$?

### 2.1.2 The relative difficulty of algorithmic problems

The restriction of an algorithmic problem results in an easier problem. Two formal notions for what "easier" means will be used. The first notion is that of a **Turing reduction**. Problem $\mathsf{P}$ is Turing reducible to problem $\mathsf{Q}$ if there is an algorithm solving problem $\mathsf{P}$ depending on a black box solution to problem $\mathsf{Q}$. For example, the fact that $\mathsf{PosAngle}(G)$ Turing reduces to $\mathsf{Angle}(G)$ is denoted by

$$\mathsf{PosAngle}(G) \preceq_{\mathrm{T}} \mathsf{Angle}(G).$$

Certainly, deciding if the angle is positive is easier than computing the angle. On the other hand, it may be misleading to describe problem $\mathsf{P}$ as easier than problem $\mathsf{Q}$ just because a

Turing reduction holds. Indeed, integer multiplication Turing reduces to integer addition, though certainly multiplication isn't easier than addition. If problem $\mathsf{P}$ Turing reduces to problem $\mathsf{Q}$ and vice-versa, the two problems are said to be **Turing equivalent** and the notation "$\approx_{\mathrm{T}}$" will be used.

The second notion of complexity reduction is stronger than Turing reduction and is more closely related to the intuitive idea of relative difficulty.

**Definition 2.1.3** *Let* $\mathsf{P}$ *and* $\mathsf{Q}$ *be algorithmic problems. Problem* $\mathsf{P}$ *is said to* **complexity reduce** *to problem* $\mathsf{Q}$ *which is denoted by*

$$\mathsf{P} \preceq \mathsf{Q}$$

*if* $\mathsf{P}$ *Turing reduces to* $\mathsf{Q}$ *and every complexity function for* $\mathsf{Q}$ *is a complexity function for* $\mathsf{P}$*. I.e.,*

$$\mathsf{P} \preceq_T \mathsf{Q} \quad \textit{and} \quad \forall f, \mathsf{Q} \preceq f \Rightarrow \mathsf{P} \preceq f \ .$$

*The problems* $\mathsf{P}$ *and* $\mathsf{Q}$ *are said to be* **complexity equivalent***, denoted by* $\mathsf{P} \simeq \mathsf{Q}$*, if* $\mathsf{P} \preceq \mathsf{Q}$ *and* $\mathsf{Q} \preceq \mathsf{P}$*.*

For example, if there is a procedure of linear complexity for translating every instance of problem $\mathsf{P}$ to an instance of problem $\mathsf{Q}$ (and $\mathsf{P}$ and $\mathsf{Q}$ have optimal complexity which isn't sublinear) then problem $\mathsf{P}$ is complexity reducible to problem $\mathsf{Q}$. Restrictions of problems, as defined above, only involve trivial modifications of input or output and are typically achievable in linear time.

### 2.1.3 Rational problems

Rational problems are algorithmic problems which are definable by means of rational subsets. For example, the word problem is rational as it involves deciding if a single word —which can be thought of as a singleton rational subset— represents 1 in the group. The generalized word problem is also rational as it involves deciding whether a singleton is contained inside a finitely generated subgroup —which is rational by example 1.6.3. More generally, we may want to know when two arbitrary rational subsets intersect trivially, or when a singleton is contained in a rational subset, etc.

The “rational unity problem” concerns deciding whether a given rational subset of a group $G$ contains 1:

**Decision problem 4** ($\mathsf{RatUnity}(G)$)

*GIVEN: A finite presentation* $P = \langle\, Y \mid R \,\rangle$ *for the group* $G$ *with the natural homomorphism* $\sigma : (Y \cup Y^{-1})^* \twoheadrightarrow G$.

*INPUT: An automaton* $\mathcal{A}$ *whose edges are labeled by elements in* $Y \cup Y^{-1} \cup \{\varepsilon\}$*. Set* $A = \sigma(\mathcal{A})$.

*DECIDE: Does the rational subset* $A$ *contain* 1*?*

There are two rational decision problems that are seemingly more general. The first is “rational membership”: decide whether an arbitrary element $w$ is contained in an arbitrary rational subset $A$ (denoted by $\mathsf{RatMemb}(G)$). The second is “rational intersection”: decide whether arbitrary rational subsets $A$ and $B$ have non-empty intersection (denoted by $\mathsf{RatInt}(G)$). We have:

$$\mathsf{RatInt}(G)|_{B=\{w\}} = \mathsf{RatMemb}(G) \text{ and } \mathsf{RatMemb}(G)|_{w=1} = \mathsf{RatUnity}(G) \quad .$$

Therefore, $\mathsf{RatUnity}(G) \preceq \mathsf{RatMemb}(G) \preceq \mathsf{RatInt}(G)$. On the other hand, let’s show that $\mathsf{RatInt}(G) \preceq \mathsf{RatUnity}(G)$ which implies that

$$\mathsf{RatUnity}(G) \simeq \mathsf{RatMemb}(G) \simeq \mathsf{RatInt}(G) \ . \tag{2.1.1}$$

**Lemma 2.1.4** $\mathsf{RatInt}(G) \preceq \mathsf{RatUnity}(G)$ *.*

*Proof.* Let $\mathcal{A}$ and $\mathcal{B}$ be automata labeled by $Y \cup Y^{-1} \cup \{\varepsilon\}$ and set $A = \sigma(\mathcal{A}), B = \sigma(\mathcal{B})$. We are interested in determining whether $A \cap B = \emptyset$. Let $\mathcal{B}^{-1}$ be the automaton obtained from $\mathcal{B}$ by keeping the same vertex set, changing the start states to terminal states and vice-versa, and replacing any edge $u \xrightarrow{x} v$ by an edge $u \xleftarrow{x^{-1}} v$ (and taking the convention that $\varepsilon^{-1} = \varepsilon$). $\mathcal{B}^{-1}$ is constructed from $\mathcal{B}$ in linear time, and $\sigma(\mathcal{B}^{-1}) = B^{-1}$. Concatenate $\mathcal{A}$ and $\mathcal{B}^{-1}$ —in linear time— to obtain an automaton $\mathcal{C} = \mathcal{A} \cdot \mathcal{B}^{-1}$ such that $\sigma(\mathcal{C}) = AB^{-1}$. Using our supposed algorithm for $\mathsf{RatUnity}(G)$ we can decide if

$$1 \in AB^{-1} \iff \exists\ a, b \text{ with } a \in A, b \in B \text{ s.t. } ab^{-1} = 1 \iff A \cap B \neq \emptyset \quad .$$

This completes the proof. ♦

The generalized word problem is a restriction of the rational membership problem. This just follows from the fact that finitely generated subgroups are rational subsets which are the images of bouquet automata (lemma 1.6.12). Therefore:

$$\mathsf{RatMemb}(G)|_{\mathfrak{B}} = \mathsf{GenWord}(G) \quad . \tag{2.1.2}$$

The "rational inclusion" problem is:

**Decision problem 5** ($\mathsf{RatInc}(G)$)

*GIVEN: A finite presentation* $P = \langle\, Y \mid R\,\rangle$ *for the group* $G$ *with the natural homomorphism* $\sigma : (Y \cup Y^{-1})^* \twoheadrightarrow G$.

*INPUT: Automata* $\mathcal{A}$ *and* $\mathcal{B}$ *whose edges are labeled by elements in* $Y \cup Y^{-1} \cup \{\varepsilon\}$. *Set* $A = \sigma(\mathcal{A})$ *and* $B = \sigma(\mathcal{B})$.

*DECIDE: Does* $A$ *contain* $B$*?*

We have $\mathsf{RatInc}(G)|_{B=\{1\}} = \mathsf{RatUnity}(G)$ so that:

$$\mathsf{RatUnity}(G) \preceq \mathsf{RatInc}(G) \quad .$$

The "rational closure problem" consists of constructing the profinite closure of a given rational set (see subsection 1.1.3):

**Computational problem 3** ($\mathsf{RatClos}(G)$)

*GIVEN: A finite presentation* $P = \langle\, Y \mid R\,\rangle$ *for the group* $G$ *with the natural homomorphism* $\sigma : (Y \cup Y^{-1})^* \twoheadrightarrow G$. $G$ *has the property that the profinite closure of a rational subset is rational.*

*INPUT: An automaton* $\mathcal{A}$ *whose edges are labeled by elements in* $Y \cup Y^{-1} \cup \{\varepsilon\}$. *Set* $A = \sigma(\mathcal{A})$.

*COMPUTE: An automaton* $\overline{\mathcal{A}}$ *such that* $\sigma(\overline{\mathcal{A}}) = \overline{A}$ *is the profinite closure of* $A$.

### 2.1.4 Lower bounds for rational problems

It is not difficult to show that $\mathsf{Word}(G)$ cannot have sublinear time complexity when $|G| > 1$. All that is necessary is the standard assumption that such an algorithm could be carried out using a deterministic Turing machine. The reader is referred to [34] for further details on Turing machines and their relationship with computational complexity.

**Proposition 2.1.5** *Let $G$ be a group generated by $X$ as a monoid. Consider the decision problem* $\mathsf{Word}(G)$ *with respect to $X$. Then*

$$G = \{1\} \iff \mathsf{Word}(G) \preceq 1 \iff \mathsf{Word}(G) \prec n \ .$$

*In particular, if $G$ is non-trivial then its word problem is not sublinear.*[4]

*Proof.* If $G$ is the trivial group, then all words are trivial so that words need not be read to verify triviality, giving a constant time algorithm.

On the other hand, suppose the word problem for $G$ is solvable in sublinear time. Show that $G = \{1\}$: If $\mathsf{Word}(G)$ is solvable in sublinear time then there is a deterministic Turing machine $\mathcal{M}$ which halts precisely on trivial words in sublinear time. In other words, there is a sublinear function $f(n)$ such that given any word $w \in X^*$ with trivial image in $G$ and of length $|w| \leq n$, the Turing machine halts in at most $f(n)$ moves. As $f(n)$ is sublinear and monotonic, there is a number $N$ such that $n > N \Longrightarrow f(n) < n$. In particular, for all accepted words $w$ of length greater than $N$, the machine $\mathcal{M}$ cannot read all of $w$. Let $w$ be one such word. Suppose for contradiction that $G$ is not the trivial group. Thus there is a letter $x \in X$ such that $wx \neq_G 1$. Now $\mathcal{M}$ halts on $w$ because $w =_G 1$ and this happens before $\mathcal{M}$ scans all of $w$. Therefore, $\mathcal{M}$ must also halt on $wx$ because $\mathcal{M}$ is deterministic; moreover, as $\mathcal{M}$ halted before reading the last letter of $w$ it halts on $wx$ at exactly the same point that it did for $w$. As $\mathcal{M}$ halts on $wx$ conclude that $wx =_G 1$ which is a contradiction. Therefore we must have $G = \{1\}$ which completes the proof. ♦

Proposition 2.1.5 implies:

**Corollary 2.1.6** *If $G$ is a finitely generated non-trivial group then the following problems have time complexity which is invariant under choice of finite generators:*

- $\mathsf{Word}(G)$,
- $\mathsf{GenWord}(G)$,
- $\mathsf{RatUnity}(G)$.

*Proof.* Since changing generators induces a linear time translation of all of the above problems, it is enough to show that each of the above problems is not sublinear. By

[4] The proof grew out of a fruitful discussion with with Saul Schleimer.

proposition 2.1.5 $\mathsf{Word}(G)$ is not sublinear. Furthermore:

$$\mathsf{RatUnity}(G) \circeq \mathsf{RatMemb}(G), \mathsf{RatMemb}(G)|_{\mathfrak{B}} = \mathsf{GenWord}(G), \mathsf{GenWord}(G)|_{A=\{1\}} = \mathsf{Word}(G)$$

imply that

$$\mathsf{Word}(G) \preceq \mathsf{GenWord}(G) \preceq \mathsf{RatUnity}(G) \quad .$$

Therefore, all of the above are not sublinear. ♦

### 2.1.5 Computing angles with rational algorithms

To compute angles using rational sets it is useful to compute the complements of rational subsets. This motivates the following algorithmic problem:

**Computational problem 4** ($\mathsf{RatComp}(G)$)
*GIVEN: A finite presentation $P = \langle\, Y \mid R \,\rangle$ for the group $G$ with canonical map $\sigma : (Y \cup Y^{-1})^* \twoheadrightarrow G$. $G$ is a group such that* $\mathrm{Rat}(G)$ *is closed under taking complements.*
*INPUT: An automaton $\mathcal{A}$ labeled by $Y \cup Y^{-1} \cup \{\varepsilon\}$ which represents the rational subset $\sigma(\mathcal{A})$.*
*COMPUTE: An automaton $\neg\mathcal{A}$ labeled by $Y \cup Y^{-1} \cup \{\varepsilon\}$ with $\sigma(\neg\mathcal{A}) = G - \sigma(\mathcal{A})$.*

**Proposition 2.1.7** *Let $G$ be a group with* $\mathrm{Rat}(G)$ *closed under complements and suppose that* $\mathsf{RatComp}(G)$ *is computable. If* $\mathsf{RatUnity}(G)$ *is recursively solvable then* $\mathsf{Angle}(G)$ *and* $\mathsf{Angle}'(G)$ *are recursively solvable.*

*Proof.* $\sphericalangle_B^A$: By proposition 3.5.1, the subsets $\bigcup_{n=1}^{\infty}[(A-B)(B-A)]^n$ and $[(A-B)(B-A)]^n$ are rational. We ask if $1 \in \bigcup_{n=1}^{\infty}[(A-B)(B-A)]^n$. This is decidable by assumption. If the answer is 'no' then by lemma 1.9.5 $\sphericalangle_B^A = 0$. Otherwise, $\sphericalangle_B^A$ is just $\pi$ divided by the first positive $n$ s.t. $1 \in [(A-B)(B-A)]^n$. By assumption, we can find this $n$ by asking if $1 \in [(A-B)(B-A)]^i$ for $i = 1, 2, 3, 4, \ldots$.

$\sphericalangle(A, B; C)$: When $C = A \cap B$ we have $\sphericalangle(A, B; C) = \sphericalangle_B^A$. Otherwise, $C \lneq A \cap B$ and $\sphericalangle(A, B; C) = \pi$. Thus by the first paragraph it is enough to be able to decide if $C \neq A \cap B$. We have

$$\exists g \text{ s.t. } g \in A \cap B - C \iff \exists g \text{ s.t. } g \in A - C \text{ and } g \in B - C \iff (A - C) \cap (B - C) \neq \emptyset \; .$$

Therefore, to decide if $C \neq A \cap B$ it is enough to decide if $A - C$ and $B - C$ intersect. By assumption $A-C$ and $B-C$ are rational and computable. Furthermore, by equation (2.1.1) $\mathsf{RatUnity}(G) \Leftrightarrow \mathsf{RatInt}(G)$ so that we can decide if $(A - C) \cap (B - C) \neq \emptyset$. ♦

In section 2.2 it will be shown that finitely generated free groups and free abelian groups satisfy the hypotheses of corollary 2.1.7. In section 2.3 we will see that these hypotheses are invariant under passing to finite extensions. Consequently, we will obtain corollary 2.3.12 which states that angles are computable in virtually[5] free groups and virtually abelian groups.

## 2.2 Three Properties Shared by Free and Free Abelian Groups

Let $G$ be a finitely generated group which is either free or free abelian. In this section we show that $G$ satisfies the following three properties involving their rational subsets:

1. The complement of a rational subset is rational, and this complement is computable.
2. Membership in rational subsets is decidable.
3. All rational subsets are unambiguously rational.

Below we prove that finitely generated free groups satisfy these three properties and that finitely generated free abelian groups satisfy the second. The first and third properties for abelian groups are quoted from [16].

### 2.2.1 Free monoids

We start by explaining why finitely generated free monoids satisfy the three properties. This will imply that the same holds true for finitely generated free groups (see subsection 2.2.2). The main idea is that of determinism. It is a remarkable fact that all rational subsets are accepted by automata which are *deterministic*. Determinism can be used to easily compute complements of languages, and to represent rational subsets unam-

[5]Let $p$ be a property associated with groups. For example $p$ could be finiteness, freeness, being abelian etc., so some groups may have property $p$ while others may not. A group $G$ is said to be "**virtually** $p$" if it contains a subgroup $H$ of finite index such that $H$ has the property $p$. For example, $G$ is virtually free if there is a free subgroup $H \underset{\text{f.i.}}{<} G$.

biguously. An automaton is deterministic if it has only one start state and any label defines a unique path from any vertex. More precisely:

**Definition 2.2.1** *Recall the notation of definition 1.3.1 (p. 9) in defining an automaton $\mathcal{A}$. Suppose that $|S| = 1$ so there is a unique start state, $r_x$ is a* function *for all $x \in X$, and that $r_\varepsilon = \emptyset$ so that there are no $\varepsilon$-edges. Then $\mathcal{A}$ is said to be a* **deterministic automaton**.

Proofs of the following fundamental result can be found in [17, §1.2] and [34, §2].

**Theorem 2.2.2** *Suppose that $\mathcal{L}$ is the language accepted by a finite state automaton $\mathcal{A}$. Then there is a constructible deterministic finite state automaton $\mathcal{A}'$ which accepts the language $\mathcal{L}$.*

As immediate corollaries we have:

**Corollary 2.2.3** $\mathrm{Rat}(X^*)$ *is closed under complements and* $\mathsf{RatComp}(X^*)$ *is recursively solvable.*

*Proof.* Consider any rational subset $S \in \mathrm{Rat}(X^*)$. By Kleene's theorem (definition 1.4.2) we can find an automaton $\mathcal{A}$ accepting the language $S$. By theorem 2.2.2 we may assume that $\mathcal{A}$ is deterministic. Construct an automaton $\neg\mathcal{A}$ accepting $X^* - S$ by switching all non-terminal states of $\mathcal{A}$ to terminal states and vice-versa. In other words, $\neg\mathcal{A}$ has the same edges, vertices, and start state as $\mathcal{A}$, but the terminal states are given by $T(\neg\mathcal{A}) = V(\mathcal{A}) - T(\mathcal{A})$. As $\mathcal{A}$ is deterministic, every $w \in X^*$ is the label of a unique path starting at the start vertex $s$. Consequently, $w$ is accepted by $\neg\mathcal{A}$ iff $w$ is not accepted by $\mathcal{A}$. Therefore, the language accepted by $\neg\mathcal{A}$ is $X^* - S$, which demonstrates algorithmically that complements of rational subsets are rational. ♦

**Corollary 2.2.4** $\mathsf{RatMemb}(X^*)$ *is recursively solvable.*

*Proof.* Given a rational subset $\mathcal{A} \in \mathrm{Rat}(X^*)$ we may assume as in the proof of corollary 2.2.3 that $\mathcal{A}$ is the accepted language of a deterministic automaton. To decide if $w \in \mathcal{A}$ simply follow the unique path determined by $w$ which starts at the start state $s$ and check if the path ends in $T$. ♦

**Corollary 2.2.5 (**Corollary VII.8.3 in [15]**)** $\mathrm{UR}(X^*) = \mathrm{Rat}(X^*)$.

*Proof.* By definition, $\mathrm{UR}(X^*) \subseteq \mathrm{Rat}(X^*)$.

On the other hand, consider $\mathcal{A} \in \mathrm{Rat}(X^*)$. As in the previous two proofs, we may assume that $\mathcal{A}$ is a deterministic automaton (identified with the language which it accepts). Prove by induction on the number of vertices in $\mathcal{A}$ that deterministic automata accept unambiguous languages. If there are no vertices, the language is empty, so finite, so unambiguous. If there is one vertex, the language is either empty (if $T = \emptyset$) or is $X^*$ (if $T = V$), and in both cases the language is unambiguous.

So suppose $\mathcal{A}$ has $n$ vertices with $n > 1$ and that all deterministic automata with less than $n$ vertices accept unambiguous languages. Let $v_1, v_2, \ldots, v_n$ be the vertices of $\mathcal{A}$ with $v_1$ the start state, and terminal states $T = \{v_{i_1}, v_{i_2}, \ldots, v_{i_m}\}$. First notice that we may assume that $T$ is a singleton. For letting $\mathcal{A}(i,j)$ be the language accepted by setting $S = \{v_i\}$ and $T = \{v_j\}$ in $\mathcal{A}$ we have:

$$\mathcal{A} = \mathcal{A}(1, i_1) \uplus \mathcal{A}(1, i_2) \uplus \cdots \uplus \mathcal{A}(1, i_m) \tag{2.2.1}$$

which expresses $\mathcal{A}$ as the unambiguous union of languages accepted by deterministic automata with the same number of vertices as $\mathcal{A}$ but with a unique accept state.

So assume that $\mathcal{A}$ has a unique accept state $v_{i_*}$. Given any set of vertices $U \subset V$ define

$$\mathcal{A}(i, j, U) = \{w \in X^* \mid w \text{ labels a path from } v_i \text{ to } v_j \text{ with all intermediate vertices in } U\}\ .$$

We show that $\mathcal{A}(i,j,U)$ is an unambiguous rational language by induction on $|U|$. Setting $U = V$ we have $\mathcal{A}(1, i_*, U) = \mathcal{A}$ which will prove that $\mathcal{A}$ is unambiguous. If $|U| = 0$ then $\mathcal{A}(i, j, \emptyset)$ consists of finitely many labels of edges starting at $v_i$ and ending at $v_j$ so is an unambiguous rational language (in fact finite). If $|U| > 0$, pick a vertex $v_k \in U$. Then we claim that

$$\mathcal{A}(i,j,U) = \mathcal{A}(i,j,U - \{v_k\}) \uplus \left(\mathcal{A}(i,k,U - \{v_k\}) \odot \mathcal{A}(k,k,U - \{v_k\})^{\circledast} \odot \mathcal{A}(k,j,U - \{v_k\})\right) \tag{2.2.2}$$

is an unambiguous expression for $\mathcal{A}(i,j,U)$. By induction, the components on the right side of equation (2.2.2) are unambiguous, so that $\mathcal{A}(i,j,U)$ is unambiguous. So why is equation (2.2.2) unambiguous? First consider the union. $\mathcal{A}(i,j,U - \{v_k\})$ is the set of labels of paths which start at $v_i$ end at $v_j$ and *don't* go through $v_k$ while

$$\mathcal{A}(i,k,U-\{v_k\})\cdot\mathcal{A}(k,k,U-\{v_k\})^*\cdot\mathcal{A}(k,j,U-\{v_k\})$$

labels paths with the same endpoints which *do* go through $v_k$. So these sets of paths are completely disjoint while having the same start and end states. Thus by determinism, the set of labels of these two sets is disjoint, so the union is disjoint. Next, consider

$$\mathcal{A}(i,k,U-\{v_k\})\odot\mathcal{A}(k,k,U-\{v_k\})^*\odot\mathcal{A}(k,j,U-\{v_k\}\ .$$

Why are the concatenations unambiguous? Suppose $w_1, w_1' \in \mathcal{A}(i,k,U-\{v_k\})$, $w_2, w_2' \in \mathcal{A}(k,k,U-\{v_k\})^*$ and $w_3, w_3' \in \mathcal{A}(k,j,U-\{v_k\})$ are such that $w_1w_2w_3 = w_1'w_2'w_3'$. Find unique successful paths $\gamma_1, \gamma_1', \gamma_2, \gamma_2', \gamma_3, \gamma_3'$ respectively labeled by $w_1, w_1', w_2, w_2', w_3, w_3'$. By determinism we may assume that $\gamma_1$ is a prefix path of $\gamma_1'$. But as $\gamma_1$ and $\gamma_1'$ both must end at $v_k$, and must not have traversed $v_k$ in mid-path, conclude that $\gamma_1 = \gamma_1'$. Similarly, $\gamma_3 = \gamma_3'$. Consequently, it follows that $w_1 = w_1'$, $w_3 = w_3'$ so by cancellation in free monoids, conclude that $w_2 = w_2'$ so that the product is unambiguous. Finally, consider $\mathcal{A}(k,k,U-\{v_k\})^{\circledast}$. Why is the monoid closure unambiguous? Suppose $w_i, w_j' \in \mathcal{A}(k,k,U-\{v_k\})$ are such that $w_1w_2\cdots w_l = w_1'w_2'\cdots w_m'$. Find corresponding paths $\gamma_i, \gamma_i'$ never traversing through $v_k$ in midpath. By the same argument as before conclude that $\gamma_1 = \gamma_1'$ so by induction $l = m$ and $w_i = w_i'$ for all $i$, showing that the monoid generation is unambiguous and completing the proof. ♦

### 2.2.2 Free groups

We now turn our attention to free groups. We want to prove the same theorems as we did in the previous subsection, but for free groups as opposed to free monoids. The work of Gilman [23] or independently of Benois [5] enables a translation of the three properties of the beginning of section 2.2 from free groups to free monoids. In fact, letting $\mathcal{G}$ be the set of all reduced words, $\mathcal{G}$ is a regular language that makes it possible to push facts about free monoids to facts about free groups. One can see directly that $\mathcal{G}$ is a regular language —as in the following figure, or one can rely on an observation due to Gromov (see theorem 3.4.1 below which restates theorem 3.4.5 of [17]) that the collection of all labels of geodesic paths in a hyperbolic group is a regular languages.

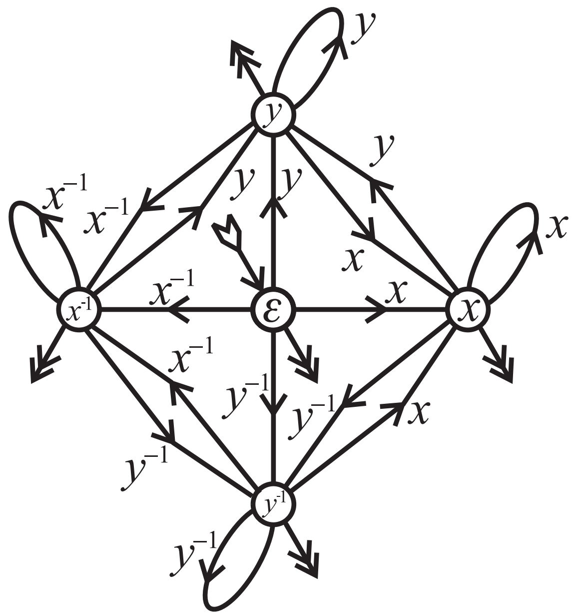


Figure 2.2: An automaton accepting the language $\mathcal{G}$ of all reduced words in $\mathrm{FG}(x, y)$.

The letter "$\mathcal{G}$" is used for the set of all reduced words since reduced words are precisely the "geodesics" in the free group. One translates problems in the free group to problems in the free monoid by using the fact that in the free group all rational sets are $\mathcal{G}$-regular:

**Theorem 2.2.6 (Gilman, Benois)** *Consider the free group $G = \mathrm{FG}(Y)$, and let $X = Y \cup Y^{-1}$ with canonical homomorphism $\sigma : X^* \twoheadrightarrow G$. Let $\mathcal{G} \subset X^*$ be the set of all reduced words. Then given any automaton $\mathcal{A} \in \mathrm{Rat}(X^*)$, there is a constructible automaton $\mathcal{A}' \in \mathrm{Rat}(X^*)$ with the property that $\sigma^{-1}(\sigma(\mathcal{A})) \cap \mathcal{G} = \mathcal{A}'$. Therefore, all rational subsets of $G$ are $\mathcal{G}$-regular so that* $\mathrm{Rat}(G) = \mathcal{G}\text{-}\mathrm{Reg}(G)$.

*Proof.* Consider Gilman's algorithm 1.3.7. Starting with $\mathcal{A}$ one obtains a reduced automaton $\widetilde{\mathcal{A}}$ with the property that $\sigma(\mathcal{A}) = \sigma(\widetilde{\mathcal{A}})$ (this is property 5 of the algorithm). Property 6 of the algorithm states that any reduced word $w$ such that $\sigma(w) \in \sigma(\widetilde{\mathcal{A}})$ will itself be accepted by $\widetilde{\mathcal{A}}$. In other words $\sigma(\widetilde{\mathcal{A}}) = \sigma(\mathcal{G} \cap \mathcal{A})$. Considering the pull-back $\mathcal{A}' = \mathcal{G} \cap \mathcal{A}$ we see that $\mathcal{A}'$ has the claimed properties. ⧫

We have the following corollaries:

**Corollary 2.2.7** *Let $G$ be a free group as in theorem 2.2.6. Then* $\mathrm{Rat}(G)$ *is closed under complements and* $\mathsf{RatComp}(G)$ *is recursively solvable.*

*Proof.* Consider any rational subset $S \in \mathrm{Rat}(G)$ so the image of an automaton $\mathcal{A} \subseteq X^*$. By theorem 2.2.6, we may assume that $\mathcal{A} \subseteq \mathcal{G}$. Therefore, by corollary 2.2.3 $\neg\mathcal{A}$ is constructible in $X^*$. Letting $\mathcal{B} = \neg\mathcal{A} \cap \mathcal{G}$ and $T = \sigma(\mathcal{B})$ we have $T = G - S$ which constructs the complement of $S$ in $G$ and completes the proof. ⧫

**Corollary 2.2.8** *Let $G$ be a free group as in theorem 2.2.6. Then* $\mathsf{RatMemb}(G)$ *is recursively solvable.*

*Proof.* Given a rational subset $S \in \mathrm{Rat}(G)$ we may assume that $S = \sigma(\mathcal{A})$ with $\mathcal{A} \subseteq \mathcal{G}$ constructible. Given any word $w$ find the reduced representative $[w]$ by extracting all subwords of the form $xx^{-1}$ repeatedly. It follows that $\sigma(w) \in S$ iff $[w] \in \mathcal{A}$ (which also follows using property 6 of algorithm 1.3.7). By corollary 2.2.4 this problem is decidable. ⧫

**Corollary 2.2.9** *Let $G$ be a free group as in theorem 2.2.6. Then* $\mathrm{UR}(G) = \mathrm{Rat}(G)$.

*Proof.* Given any subset $S \in \mathrm{Rat}(G)$, as before we represent it by a rational sublanguage $\mathcal{A} \subseteq \mathcal{G}$. By corollary 2.2.5 it follows that $\mathcal{A}$ is unambiguous in $X^*$. Finally, since $\sigma$ restricts to a one-to-one map on $\mathcal{G}$ it preserves unambiguity of subsets contained in $\mathcal{G}$. In particular, $S = \sigma(\mathcal{A})$ is unambiguous. ⧫

It is possible to distill the techniques the above proofs to obtain the following generalization:

**Theorem 2.2.10** *Let $M$ be a monoid finitely generated by $X$, and with canonical homomorphism $\sigma : X^* \twoheadrightarrow M$. Let $\mathcal{L} \subseteq X^*$ be a* **rational cross-section** *for $M$; i.e., suppose that $\mathcal{L}$ is a regular language such that $\sigma|_{\mathcal{L}}$ is a bijection onto $M$. If*

$$\mathcal{L}\text{-Reg}(M) = \mathrm{Rat}(M) \tag{2.2.3}$$

*then* $\mathrm{Rat}(M) = \mathrm{UR}(M)$*; furthermore,* $\mathrm{Rat}(M)$ *is closed under complements. In addition, if equation 2.2.3 is effective so that given any rational subset $S$ an automaton $\mathcal{A} \subseteq \mathcal{L}$ can be constructed such that $\sigma(\mathcal{A}) = S$, then* $\mathsf{RatMemb}(M)$ *and* $\mathsf{RatComp}(M)$ *are recursively solvable.*

**Question 1** *What groups satisfy the hypotheses of theorem 2.2.10? Does the fundamental group of the 2-holed torus (the group with presentation $\langle\, a, b, c, d \mid [a, b][c, d]\, \rangle$ ) satisfy these hypotheses?*

### 2.2.3 Free abelian groups

Finally we turn our attention to free abelian groups. Eilenberg and Schützenberger showed that the rational subsets of a finitely generated commutative monoid $M$ are unambiguous [16, theorem IV] and are closed under complements [16, theorem III]. Furthermore, the methods of proof in [16] are constructive (albeit highly impractical) so that given a regular expression representing $S \in \mathrm{Rat}(M)$, an expression can be constructed to represent $M - S$. In other words, finitely generated commutative monoids satisfy the properties 1 and 3 described at the beginning of this section:

**Theorem 2.2.11 (Eilenberg and Schützenberger)** *Suppose that $M$ is a finitely generated commutative monoid. Then $\mathrm{Rat}(M)$ is closed under complements, $\mathsf{RatComp}(X^*)$ is computable, and $\mathrm{UR}(M) = \mathrm{Rat}(M)$.*

Finally, a solution to the membership problem in rational subsets of free abelian groups is obtained from standard integer linear programming techniques:

**Proposition 2.2.12** $\mathsf{RatMemb}(\mathbb{Z}^n)$ *is recursively solvable.*

*Proof.* Let $S$ be a rational subset represented by an automaton $\mathcal{A}$ labeled by $X$ where $X = Y \cup Y^{-1}$, $Y = \{y_1, \ldots, y_n\}$ and $\mathbb{Z}^n$ has presentation $\left\langle\, Y \;\middle|\; \{[y_i, y_j] \mid 1 \le i < j \le n\} \,\right\rangle$. By Kleene's theorem we may assume that $\mathcal{A}$ is given by a regular expression $\mathcal{B}$. We have $\sigma(\mathcal{B}) = S$. The claim is that $\mathcal{B}$ can be suitably modified (without changing its image in $\mathbb{Z}^n$) to be of the form:

$$\mathcal{B} = \bigcup_{i=1}^{N} S_i^* s_i \quad \text{with} \quad S_i \subset X^* \ \text{ finite and } \ s_i \in X^* \tag{2.2.4}$$

so that every rational subset is the finite union of translates of finitely generated submonoids. This is proved by induction on the length of the expression $\mathcal{B}$. If the length is 0 or 1, $S$ is finite, and the claim is immediate. Otherwise, there are regular expressions $\mathcal{C}$ and $\mathcal{D}$ such that one of the following holds: $\mathcal{B} = \mathcal{C} \cup \mathcal{D}$, or $\mathcal{B} = \mathcal{C}\mathcal{D}$ or $\mathcal{B} = \mathcal{C}^*$. By induction, $\mathcal{C}$ and $\mathcal{D}$ may be assumed to be of the form described by equation (2.2.4). If $\mathcal{B} = \mathcal{C} \cup \mathcal{D}$ then the form of equation (2.2.4) is preserved.

Next, suppose $\mathcal{B} = \mathcal{C}\mathcal{D}$. Assume by induction that $\mathcal{C}$ and $\mathcal{D}$ are given as by equation (2.2.4) so

$$\mathcal{C} = \bigcup_{i=1}^{L} T_i^* t_i \quad \text{and} \quad \mathcal{D} = \bigcup_{j=1}^{M} U_j^* u_j \qquad \text{with } T_i, U_i \subset X^* \text{ finite and } t_i, u_i \in X^*. \tag{2.2.5}$$

Therefore,

$$\sigma(\mathcal{C}\mathcal{D}) = \sigma\left(\left[\bigcup_{i=1}^{L} T_i^* t_i\right]\left[\bigcup_{j=1}^{M} U_j^* u_j\right]\right) = \bigcup_{(i,j)=(1,1)}^{(L,M)} \sigma(T_i^* U_j^*)\sigma(t_i u_j) \tag{2.2.6}$$

where we have used the fact that $\sigma$ is a homomorphism as well as the commutativity of $\mathbb{Z}^n$. If $A$ and $B$ are submonoids in a commutative monoid, so is $AB$. Therefore conclude that $T_i^* U_j^* = (T_i \cup U_j)^*$ so equation (2.2.6) gives

$$\sigma(\mathcal{B}) = \sigma\left(\bigcup_{(i,j)=(1,1)}^{(L,M)} (T_i \cup U_j)^* t_i u_j\right)$$

which implies that $S$ is the image of a rational subset of the form as in equation (2.2.4).

Finally suppose that $\mathcal{B} = \mathcal{C}^*$ with $\mathcal{C}$ as in equation (2.2.5). Consider any two subsets $A$, $B$ and any two elements $a$, $b$ of a commutative monoid. Notice that

$$(A^*a \cup B^*b)^* = (A \cup B \cup \{a,b\})^* ab \bigcup (A \cup \{a,b\})^* a \bigcup (B \cup \{a,b\})^* b \bigcup \{a,b\}^* \ . \tag{2.2.7}$$

Why is this true? For example, consider an element of $(A \cup B \cup \{a,b\})^* ab$. By commutativity, such an element is of the form $\alpha\beta a^m b^n$ with $\alpha \in A^*, \beta \in B^*$ and $m, n \geq 1$. We can express this element in the form

$$\alpha\beta a^m b^n = \alpha a \cdot \beta b \cdot a^{m-1} \cdot b^{n-1} \in A^* a \cdot B^* b \cdot a^* \cdot b^* \subseteq (A^* a B^* b)^*$$

which proves that $(A \cup B \cup \{a,b\})^* ab \subseteq (A^*a \cup B^*b)^*$. Similar arguments can be used to show that the left hand side of equation (2.2.7) contains the right hand side. On the other hand, consider an arbitrary element of the left hand side. For example, it may be of the form $\alpha_1 a \beta_1 b \cdots \alpha_n a \beta_n b$. By commutativity this equals $(\alpha_1 \cdots \alpha_n) \cdot (\beta_1 \cdots \beta_n) \cdot a^{n-1} \cdot b^{n-1} \cdot ab$

which is in $(A \cup B \cup \{a,b\})^* ab$ so is an element of the right hand side. Similar arguments can be used to show that the left hand side is contained in the right hand side. More generally, letting $\tau = \{t_1, t_2, \dots, t_L\}$ with the $t_i$ as in equation (2.2.5) if we define $\mathcal{E}$ according to the following formula

$$\mathcal{E} = \bigcup_{S \subseteq \{1,2,\dots,L\}} \left[ \left( \tau \cup \bigcup_{i \in S} T_i \right)^* \prod_{i \in S} t_i \right] \tag{2.2.8}$$

then it follows that $\sigma(\mathcal{E}) = \sigma\left(\bigcup_{i=1}^{L} T_i^* t_i\right)^* = \sigma(\mathcal{C}^*)$ which shows that we may replace $\mathcal{B}$ by $\mathcal{E}$ to obtain the claimed form of equation (2.2.4) for rational subsets in commutative monoids.

Thus to solve $\mathsf{RatMemb}(\mathbb{Z}^n)$ it suffices to solve it for rational subsets of the form given in equation (2.2.4). Now if membership is decidable for components of a union, it is decidable for the whole union. Therefore, it suffices to decide membership inside the translates of a finitely generated submonoid of $\mathbb{Z}^n$. We can reduce this problem to membership in finitely generated submonoids of $\mathbb{Z}^n$ because (using additive notation):

$$u \in S + s \quad \Longleftrightarrow \quad u - s \in S \ .$$

Thus we have reduced $\mathsf{RatMemb}(\mathbb{Z}^n)$ to membership in finitely generated submonoids. I.e. given integers $a_i, b_i^j \in \mathbb{Z}$ (here $j$ is an index and *not* the power that $b_i$ is being raised to) such that $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, m$ we would like to decide if the following matrix equation has a solution in non-negative integers $x_j \in \mathbb{N} \cup \{0\}$:

$$\begin{pmatrix} b_1^1 & b_1^2 & \cdots & b_1^m \\ b_2^1 & b_2^2 & \cdots & b_2^m \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ b_n^1 & b_n^2 & \cdots & b_n^m \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ \vdots \\ a_n \end{pmatrix} \ . \tag{2.2.9}$$

This reduces $\mathsf{RatMemb}(\mathbb{Z}^n)$ to an integer linear programming problem, and integer linear programming is well known to be solvable[6]. ♦

[6] Although integer linear programming also happens to be NP-complete! See [20, p. 245] for further details.

## 2.3 Algorithmic Stability Under Finite Index

The main goal of this section is to show that certain algorithms can be pushed through finite extensions and examine the impact that such extensions have on algorithmic complexity. We start in subsection 2.3.1 with the rational decision problems Word, GenWord and RatUnity[7]. The invariance of recursiveness for Word and GenWord under finite extensions is well known; e.g., this follows from Farb's analysis of generalized isoperimetric functions [18]. Theorem 2.3.3 summarizes another point of view. In particular we obtain corollary 2.3.4 which implies that if a polynomial time algorithm for the generalized word problem exists for a finite index subgroup, then it also exists for the overgroup. I believe that the analysis of complexity, and the solvability of RatUnity under finite extensions is new. Then we examine angle computation in subsection 2.3.2, and show that closure of rational subsets under complements is preserved when passing to finite extensions; this shows that the first two properties mentioned at the beginning of section 2.2 are preserved under finite extension and thus angles are computable in virtually free and virtually abelian groups (corollary 2.3.12). In subsection 2.3.3 we show that the third property of section 2.2 is preserved as well (theorem 2.3.13); therefore, the unambiguity of rational subsets is inherited by finite extensions so that virtually free and virtually abelian groups satisfy UR = Rat (corollary 2.3.15). We conclude by looking at the computational problem RatClos$(G)$ and show that its solution can also be pushed up through finite extensions. By work of Steinberg [53], [54] this implies that closures of rational subsets are computable in virtually free and virtually abelian groups (corollary 2.3.17).

### 2.3.1 Stability of rational decision problems

We examine the relationship between solutions to rational problems and subgroups. First, we examine the easier direction: If there is an algorithm for solving a rational problem in a certain group, then the algorithm should certainly apply to any subgroup. More precisely:

**Proposition 2.3.1** *Let $H$ be a finitely generated subgroup of the group $G$. Then*

- Word$(H) \preceq$ Word$(G)$,

[7] Thus also the problems RatMemb and RatInt which are equivalent to RatUnity (see p. 43).

- $\mathsf{GenWord}(H) \preceq \mathsf{GenWord}(G)$,
- *and* $\mathsf{RatUnity}(H) \preceq \mathsf{RatUnity}(G)$.

*Proof.* Each of the above decision problem for $H$ is already a decision problem for $G$ except that each generator in $H$ appearing in the problem needs to be translated to a generator in $G$. As $H$ is finitely generated, this translation process is of linear complexity. ♦

Clearly, proposition 2.3.1 cannot have a full converse. For example, a group with undecidable word problem contains many subgroups (including cyclic groups) whose word problem is solvable in linear time. Therefore $H < G$ cannot imply $\mathsf{Word}(G) \preceq \mathsf{Word}(H)$. However, when restricting to finite index subgroups, theorem 2.3.3 shows that the converse does hold —as far as Turing reducibility is concerned.

**2.3.2 Remark** *Note that this is not the case for all group theoretic decision problems. Indeed, there are finitely presented groups $A, B, C, D$ such that $A \underset{2}{<} B, C \underset{2}{<} D$ with $A$ and $D$ having decidable conjugacy problem, but with $B$ and $C$ having* undecidable *conjugacy problem (cf. [10]).*

The condition in theorem 2.3.3 that the groups be infinite is necessary as all decision problems are trivial for $\{1\}$.

**Theorem 2.3.3** *Let $G$ and $H$ be finitely generated infinite groups which are commensurable. Then*

- $\mathsf{Word}(G) \precsim\succsim \mathsf{Word}(H)$.
- *If* $\mathsf{GenWord}(H) \preceq f(x)$ *then* $\mathsf{GenWord}(G) \preceq f(x^2)$.
- $\mathsf{RatUnity}(G) \approx_{\mathrm{T}} \mathsf{RatUnity}(H)$.

An immediate corollary is:

**Corollary 2.3.4** *Let $G$ and $H$ be finitely generated infinite groups which are commensurable. Suppose that* $\mathsf{GenWord}(G)$ *has polynomial time complexity. Then* $\mathsf{GenWord}(H)$ *has polynomial time complexity.*

Before giving the proof of theorem 2.3.3, let's build some machinery. We'll need a procedure for translating words in a group to a normal form with respect to the subgroup. For example, consider $G = \mathbb{Z}^2 = \langle\ x, y \mid [x, y]\ \rangle$. Let $H$ be the subgroup generated by $x$. Consider $H\backslash G$. The set $\mathcal{U} = \{\ldots, y^{-2}, y^{-1}, \varepsilon, y, y^2, \ldots\} \subset \{x, x^{-1}, y, y^{-1}\}^*$ in the free monoid is called a **right transversal** for $H\backslash G$ because each element of $\mathcal{U}$ represents a unique right $H$-coset, and all right cosets are represented in this way. Our normal forms will consist of the language $\{x, x^{-1}\}^* \cdot \mathcal{U}$. We are interested in taking an arbitrary word in $G$ and translating it to a word of this special form. For example, if $w = xyyyx^{-1}y^{-1}xxyy$ then a good translation of $w$ is $w' = x^2 \cdot y^4$, as $w =_G w'$, and $w'$ is of the required form. In particular, we can immediately tell in which $H$-coset $w$ belongs to by glancing at the "$y$" portion of $w'$. The next proposition will consider the case of *finite index* $H$, so that $\mathcal{U}$ is finite. However, it is worthwhile to state the "subgroup translation" problem in greater generality to allow for more flexibility (e.g. the case of $\mathbb{Z}^2$ just considered).

**Computational problem 5** ($\mathsf{SubgpTrans}(G, H; \mathcal{U})$)

*GIVEN: Groups $H < G$ with respective finite symmetric generating sets $Z \subseteq X$, and $\sigma : X^* \twoheadrightarrow G$. A recursive right transversal $\mathcal{U} \subset X^*$ for $H\backslash G$.*

*INPUT: A word $w \in X^*$.*

*COMPUTE: A word $w' \in Z^*\mathcal{U}$ such that $\sigma(w) = \sigma(w')$.*

Consider the case that $H$ is a subgroup of index $n$ in $G$. Choose a right transversal $\mathcal{U} = \{u_1, \ldots, u_n\}$ and set $u_1 = \varepsilon$ (the representative of $H$):

**Proposition 2.3.5** *If $H \underset{n}{<} G$ then* $\mathsf{SubgpTrans}(G, H; \mathcal{U})$ *is solvable in linear time.*

*Proof.* Consider the coset diagram $\Gamma_X(H\backslash G)$. In addition to viewing $\Gamma_X(H\backslash G)$ as an automaton, it is convenient to realize $\Gamma_X(H\backslash G)$ as a topological graph $||\Gamma_X(H\backslash G)||$ (see p. 21); furthermore, $||\Gamma_X(H\backslash G)||$ is a based topological space if we set the basepoint to be $* = H$. Choose a *maximal tree* $\mathcal{T}$ in $||\Gamma_X(H\backslash G)||$. The set of **chords** $\mathcal{C}$ is the set of all labeled edges (in the directed graph $\Gamma_X(H\backslash G)$) which are not in $\mathcal{T}$. For any vertex $a$ there is a unique reduced path in $\mathcal{T}$ connecting $*$ to $a$. Denote this path by $t(a)$. Thus any chord $c \in \mathcal{C}$ defines a unique loop $l(c)$ based at $*$ of the form

$$l(c) = t(\iota(c)) \cdot c \cdot t(\tau(c))^{-1}\ . \tag{2.3.1}$$

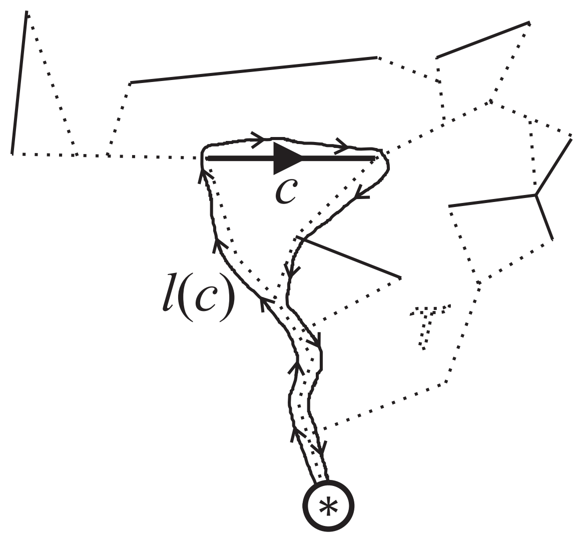



Figure 2.3: The loop at $*$ defined by a maximal tree $\mathcal{T}$ (dotted lines) and a chord $c$.

Denote the label of the loop $l(c)$ by $\lambda(c)$.[8]

Starting from the start state $H$, $w$ defines a path in $\Gamma_X(H\backslash G)$ which ends at $H\sigma(w)$. For some $i$ we have $H\sigma(w) = H\sigma(u_i)$. Notice that $\Gamma_X(H\backslash G)$ is deterministic because for each label $x \in X$ and each vertex $v$ there is a unique edge starting at $v$ which is labeled by $x$. Therefore, finding the state $H\sigma(u_i)$ and consequently $u_i$ takes time linear in $|w|$. Consider the word $v = wu_i^{-1}$. Since $w$ and $u_i$ represent the same coset, it follows that $\sigma(v) \in H$. Suppose that there is a linear time algorithm for translating $v \in X^*$ to a word $v' \in Z^*$ such that $\sigma(v) = \sigma(v')$. Then $w' = v'u_i$ is an element of $Z^* \cdot \mathcal{U}$ such that $\sigma(w) = \sigma(w')$. This gives a reduction to the case that $\sigma(w) \in H$.

So suppose that $\sigma(w) \in H$. Therefore, the path labeled by $w$ and starting at $H$ in $\Gamma_X(H\backslash G)$ is a loop. Let $(c_1, c_2, \ldots, c_s)$ be the sequence of chords traversed by $w$. Certainly $s \leq |w|$. Furthermore,

$$\sigma(w) = \sigma(\lambda(c_1)\lambda(c_2)\cdots\lambda(c_s)) \ .$$

Thus to effect a linear time translation of $w$ to a word in $Z^*$ it suffices to give a constant time algorithm for translating each of the chord loops $\lambda(c_i)$. But there are only finitely many such chord loops so we can keep a translation table in the background giving a constant

[8]In the notation of definition 1.3.1, this label is actually $\lambda(l(c))$, which is abbreviated here to $\lambda(c)$.

time translation of chord loops. ⧫

*Proof of theorem 2.3.3.* Recall the definition of commensurability: $H$ and $G$ are commensurable if there is a group $K$ which embeds as a finite index subgroup in both $H$ and $G$. Proposition 2.3.1 implies that each decision problem considered is no more difficult for a subgroup than it is for the overgroup. Thus it is enough to study what happens to each decision problem when passing to an overgroup of finite index; therefore, assume $H$ is a finite index subgroup of $G$ and let $n = [G : H]$ be the index.

Choose a symmetric set of generators $X$ (resp. $Z$) for $G$ (resp. $H$). Since the complexity of the decision problems considered here does not depend on the presentation chosen (corollary 2.1.6), we may assume that $Z \subseteq X$. As usual, let $\sigma : X^* \twoheadrightarrow G$ be the canonical map. The linear time algorithm of 2.3.5 for $\mathsf{SubgpTrans}(G, H; \mathcal{U})$ is applicable.

$\mathsf{Word}$: Suppose $\mathsf{Word}(H) \preceq f$. Translate $w \in X^*$ in linear time to a word $w' = vu_i$ for some $i$ and some $v \in Z^*$ with $v$ linearly bounded by $w$. So there is a $K \geq 1$ such that $|v| \leq K|w|$ for all such $v$. If $u_i \neq \varepsilon$ then $\sigma(w) \notin H$; in particular, $w \neq_G 1$. On the other hand, if $u_i = \varepsilon$ then we can decide if $v \in Z^*$ represents 1 in $H$ in time $f(|v|) \leq f(K|w|)$. As $f(Kn) \approx f(n)$ it follows that $\mathsf{Word}(G) \preceq f$ so that $\mathsf{Word}(G) \preceq \mathsf{Word}(H)$.

$\mathsf{GenWord}$: Suppose $\mathsf{GenWord}(H) \preceq f(x)$. Consider the input $w_0, w_1, \ldots, w_m \in X^*$. The complexity of the input is $\kappa = \sum_{i=0}^{m} |w_i|$. We would like to decide if $\sigma(w_0) \in \langle \sigma(w_1), \ldots, \sigma(w_m) \rangle = A$. This is done as follows: Start with the coset diagram $\mathcal{B} = \Gamma_X(H \backslash G)$. For each vertex $Hu_i$ and each word $w_j$ add a topological edge $(Hu_i, Hu_i w_j)$ whose positive orientation is labeled by $w_j$ (and negative orientation by $w_j^{-1}$). Such an edge is constructible in time $|w_j|$ by following the path labeled by $w_j$ in $\mathcal{B}$ starting from $Hu_i$. Constructing all such edges from a single vertex can thus be achieved in time $\sum_{i=1}^{m} |w_j| \leq \kappa$. As $\mathcal{B}$ has $n$ vertices, the automaton $\mathcal{B}'$ consisting of the same vertices as $\mathcal{B}$ but only of the $w_i^{\pm 1}$-labeled edges is constructible in time $n\kappa$. Let $[\mathcal{B}']_*$ denote the component connected to $* = H$. Now we are ready to read $w_0$ in $\mathcal{B}$. Let $b$ be the terminal vertex of the path labeled by $w_0$ and starting at $*$; the vertex $b$ is retrievable in time $|w_0|$.

First suppose that $b \notin [\mathcal{B}']_*$. Then $\sigma(w_0) \notin A$: Otherwise, there are words $v_j \in \{w_1, w_1^{-1}, w_2, w_2^{-1}, \ldots, w_m, w_m^{-1}\}$ such that $\sigma(w_0) = \sigma(v_1 v_2 v_3 \cdots v_k)$, and therefore

$$H\sigma(w_0) = H\sigma(v_1)\sigma(v_2)\sigma(v_3) \cdots \sigma(v_k) \ .$$

The right hand side defines a vertex in $[\mathcal{B}']_*$ which contradicts the fact that the vertex $b$ is the left hand side.

Next suppose that $b = *$. Therefore, $\sigma(w_0) \in H$ and we may translate $w_0$ to a word $w_0' \in Z^*$ such that $\sigma(w_0) = \sigma(w_0')$. Since $\sigma(w_0) \in H$ we have

$$\sigma(w_0) \in A \Longleftrightarrow \sigma(w_0') \in H \cap A \ .$$

Our aim is to use the solution for the generalized word problem in $H$ so we need a method of obtaining generators for $H \cap A$ in terms of $Z$. Let $\mathcal{T}'$ be a maximal tree for $[\mathcal{B}']_*$ with complementary chord set $\mathcal{C}'$. Given a chord $c \in \mathcal{C}'$ a unique loop $l'(c)$ in $\mathcal{B}'$ based at $*$ is defined, as in equation (2.3.1). The label of $l'(c)$ is denoted by $\lambda'(c)$ and is an element of $\{w_1, w_1^{-1}, w_2, w_2^{-1}, \dots, w_m, w_m^{-1}\}^*$ as each edge is labeled by some $w_i$ or its inverse. Therefore, we see that $\sigma(\lambda'(c)) \in H \cap A$. On the other hand, consider any element of $g \in H \cap A$. Since $g \in A$, there is a word $u \in \{w_1, w_1^{-1}, w_2, w_2^{-1}, \dots, w_m, w_m^{-1}\}^*$ such that $g = \sigma(u)$. Thus there is a path in $\mathcal{B}'$ which is labeled by $u$ and starts at $*$. As $g \in H$, this path must also end at $*$. Letting $(c_1, c_2, \dots, c_s)$ be the sequence of chords in $\mathcal{C}'$ traversed by $w$ we have $g = \sigma(\lambda'(c_1)\lambda'(c_2)\cdots\lambda'(c_s))$ so that $g$ is represented by a product of chord loop labels. Summarizing,

$$A \cap H = \langle \lambda'(c) \rangle_{c \in \mathcal{C}'} \text{ is generated by labels of chord loops in } [\mathcal{B}']_*.$$

The number of topological chords is no greater than the number of topological edges in $\mathcal{B}'$ so that $|\mathcal{C}'| \leq nm$. As $[\mathcal{B}']_*$ contains at most $n$ vertices, the tree $\mathcal{T}'$ contains at most $n-1$ edges. Equation (2.3.1) implies that each chord loop contains at most $(n-1)+1+(n-1) = 2n-1$ edges, so that $|\lambda'(c)| \leq (2n-1)\kappa$. Translating each chord loop to a word in $Z^*$ results in a word of length linearly bounded by some factor $K$. Putting this all together, we get a generating set $\{w_1', w_2', \cdots, w_t'\}$ for $H \cap A$ with $w_i' \in Z^*$, $t \leq nm$, and $|w_i'| \leq K(2n-1)\kappa$. Therefore, the complexity of $\{w_0', w_1', w_2', \cdots, w_t'\}$ is

$$\sum_{i=0}^{t} |w_i'| \quad \leq \quad K(2n-1)\kappa nm \quad \preceq \quad \kappa^2 \tag{2.3.2}$$

where the fact that $m \leq \kappa$ is used. Thus an instance of $\mathsf{Word}(G)$ with complexity $\kappa$, has been translated to an instance of $\mathsf{Word}(H)$ with complexity of order $\kappa^2$ —in the case that $b = *$.

Finally, we need to consider the case that $b \in [\mathcal{B}']_*$, but $b \neq *$. Let $u$ be the label of the path $t'(b)$ connecting $*$ to $b$ in $\mathcal{T}'$. Certainly $\sigma(u) \in A$. Furthermore, $H\sigma(w_0) = H\sigma(u)$ so that $w_0 u^{-1}$ defines a loop based at $*$ in $\mathcal{B}$. By the previous case considered, we can decide in time of order $f(\kappa^2)$ whether or not $\sigma(w_0 u^{-1}) \in A$. Since $\sigma(u) \in A$, this is equivalent to deciding if $\sigma(w_0) \in A$. Thus for any $w_0, w_1, \ldots, w_m \in X^*$ of total length $\kappa$, one can decide whether or not $w_0 \in \langle w_1, \ldots, w_m \rangle$ in time of order $f(\kappa^2)$ —when the corresponding question for $H$ is decidable in time $f(\kappa)$.

RatUnity: We are given an automaton $\mathcal{A}$ and would like to decide if $1 \in \sigma(\mathcal{A})$. As we're only after Turing reducibility, complexity considerations will not be kept track of (the algorithm described involves translating an automaton to a regular expression, so is at best exponential). Viewing the Cayley graph $\mathcal{B}$ above as an automaton, and therefore as a language, we have that $\mathcal{B} = \sigma^{-1}(H)$ (see lemma 1.5.5); in particular, we have $\mathcal{B} \supset \sigma^{-1}(1)$. Therefore, $1 \in \sigma(\mathcal{A})$ if and only if there is a $w \in \mathcal{A} \cap \mathcal{B}$ such that $\sigma(w) = 1$. Using the pull-back construction produces an automaton $\mathcal{C} = \mathcal{A} \cap \mathcal{B}$ whose language is $\mathcal{A} \cap \mathcal{B}$. The pull-back relies on $\varepsilon$-free automata, so $\mathcal{A}$ needs to be made $\varepsilon$-free before applying the construction. One way to achieve this is by replacing every edge $a \xrightarrow{\varepsilon} c$ by $a \xrightarrow{x} b \xrightarrow{x^{-1}} c$ where $b$ is a new vertex. This changes the language $\mathcal{A}$ but not its image $\sigma(\mathcal{A}) \subseteq G$.

Thus we have reduced the problem of deciding if $1 \in \sigma(\mathcal{A})$ to deciding if $1 \in \sigma(\mathcal{C})$ where $\sigma(\mathcal{C}) \in \mathrm{Rat}(G)$ also happens to be a subset of $H$. Seemingly, we are done, as we assume a solution to RatUnity$(H)$ and $\mathcal{C}$ represents a rational subset which is contained in $H$. Unfortunately, we are not yet done as $\mathcal{C}$ is an automaton labeled by $X$ and *not* the generating set for $H$. Therefore, $\mathcal{C}$ is invalid input for RatUnity$(H)$. However corollary 2.3.8 below gives a method for translating $\mathcal{C}$ into an automaton on a generating set for $H$.

**Proposition 2.3.6** (proof on p. 65) *Let $N \lhd G$. Let $X$ with subset $Z$ respectively generate $G$ and $N$. Let $\mathcal{U} \subset X^*$ be a right transversal for $N\backslash G$. Suppose that $\mathcal{C}$ is a regular expression on $X$ such that $\sigma(\mathcal{C}) \subseteq N\sigma(u)$ for some $u \in \mathcal{U}$. Then there is a regular expression $\mathcal{C}'$ on $Z$ such that $\sigma(\mathcal{C}')\sigma(u) = \sigma(\mathcal{C})$. In particular, if $S \in \mathrm{Rat}(G)$ and $S \subset N$ then $S \in \mathrm{Rat}(N)$.*

*Furthermore, suppose that $\mathcal{U}$ is recursive and that* SubgpTrans$(G, N; \mathcal{U})$ *is recursively solvable. Then $\mathcal{C}'$ is computable from $\mathcal{C}$.*

Propositions 2.3.5 and 2.3.6 immediately imply:

**Corollary 2.3.7** *Let $N \underset{\text{f.i.}}{\lhd} G$. Let $X$ with subset $Z$ respectively generate $G$ and $N$. Let $\mathcal{U} \subset X^*$ be a right transversal for $N\backslash G$. Suppose that $\mathcal{C}$ is a regular expression on $X$ such that $\sigma(\mathcal{C}) \subseteq N\sigma(u)$ for some $u \in \mathcal{U}$. Then there is a computable regular expression $\mathcal{C}'$ on $Z$ such that $\sigma(\mathcal{C}')\sigma(u) = \sigma(\mathcal{C})$.*

In the case of finite index, it is also possible to generalize to subgroups which are not necessarily normal:

**Corollary 2.3.8** *Let $H \underset{n}{<} G$. Let $X$ with subset $Y$ respectively generate $G$ and $H$. Let $\mathcal{U} \subset X^*$ be a right transversal for $H\backslash G$. Suppose that $\mathcal{C}$ is a regular expression on $X$ such that $\sigma(\mathcal{C}) \subseteq H\sigma(u)$ for some $u \in \mathcal{U}$. Then there is a computable regular expression $\mathcal{C}'$ on $Y$ such that $\sigma(\mathcal{C}')\sigma(u) = \sigma(\mathcal{C})$.*

*Proof of corollary 2.3.8.* In this proof the distinction between regular expressions and automata will be blurred. As we are not considering complexity issues in this case, all that matters is that there is a procedure for translating automata to regular expressions and vice-versa.

We may assume that $H$ is a *normal* subgroup of $G$: For consider the automaton obtained as the pullback of the Cayley graphs of the conjugates of $H$ (see section 1.3.3):

$$\mathcal{N} = \bigcap_{g \in \sigma(\mathcal{U})} \Gamma_X(g^{-1}Hg\backslash G)$$

where $\mathcal{U}$ is a right transversal set for $H\backslash G$. Let $N = \sigma(\mathcal{N})$. The subgroup $N$ is the normal core of $H$ —i.e., the intersection of all the conjugates of $H$ in $G$, so the biggest normal subgroup of $G$ which is contained in $H$. The finite intersection of finite index subgroups is of finite index, so $N \underset{\text{f.i.}}{<} G$. Set $\mathcal{B} = [\mathcal{N}]_* = \Gamma_X(N\backslash G)$ where $* = (Hg)_{g\in\mathcal{U}}$ is the Cartesian product of all the basepoints of the coset diagrams $\Gamma_X(g^{-1}Hg\backslash G)$. Let $Z$ be a finite generating set for $N$. Without loss of generality, $Z \subseteq Y \subseteq X$ and all these sets are symmetric. Let $\mathcal{V} \subset Y^*$ be a right transversal for $N\backslash H$. Then $\mathcal{V}\mathcal{U}$ is a right transversal for $N\backslash G$.

Consider a regular expression $\mathcal{C}$ in $X^*$ which is contained in the right coset $H\sigma(u)$. For each $v \in \mathcal{V}$ there is an automaton $\mathcal{B}(vu)$ obtained by choosing the unique accept state of $\mathcal{B}$ to be the vertex $N\sigma(vu)$ and keeping the unique start state as $* = N$. Setting

$\mathcal{C}(v) = \mathcal{C} \cap \mathcal{B}(vu)$ and using lemma 1.5.5 implies that:

$$\sigma(\mathcal{C}) = \bigcup_{v \in \mathcal{V}} \sigma(\mathcal{C}(v)) \quad \text{with} \quad \sigma(\mathcal{C}(v)) \subseteq N\sigma(vu) \text{ for all } v \in \mathcal{V} \ . \tag{2.3.3}$$

But corollary 2.3.7 implies that each of the $\mathcal{C}(v)$ give rise to a computable regular expression $\mathcal{C}'(v)$ on $Z$ with $\sigma(\mathcal{C}'(v))\sigma(vu) = \sigma(\mathcal{C}(v))$. Letting

$$\mathcal{C}' = \bigcup_{v \in \mathcal{V}} \mathcal{C}'(v) \cdot v$$

we see that $\mathcal{C}'$ is a computable regular expression on $Y$ with the property that $\sigma(\mathcal{C}')\sigma(u) = \sigma(\mathcal{C})$. This is exactly the regular expression that we were after. ◇2.3.8

To complete the $\mathsf{RatUnity}$ portion of the proof of theorem 2.3.3 we translate our automaton $\mathcal{A}$ on $X$ to a regular expression on $X$, apply corollary 2.3.8 to translate this expression to the generators of $H$, and then convert this regular expression to an automaton on the generators of $H$, which is valid input for $\mathsf{RatUnity}(H)$. ⧫2.3.3

**2.3.9 Remark** *I am unable to show that the generalized word problem is completely invariant under finite index. However, a close examination of the proof of theorem 2.3.3 —especially equation (2.3.2)— reveals that an input of* $\mathsf{Word}(G)$ *consisting of* $m$ *words of total length* $\kappa$ *is translated to an input of* $\mathsf{Word}(H)$ *consisting of less than* $nm$ *words and of total length linearly bounded in* $m\kappa$*, where* $n = [G : H]$*. This implies that:*

$$\mathsf{GenWord}_m(G) \preceq \mathsf{GenWord}_{[G:H]m}(H)$$

*where* $\mathsf{GenWord}_m$ *is the restriction of the generalized word problem to subgroups of rank no greater than* $m$*:*

**Decision problem 6** ($\mathsf{GenWord}_m(G)$)
*GIVEN: A finite presentation* $P = \langle\ Y \mid R\ \rangle$ *for the group* $G$*.*
*INPUT: A word* $w \in (Y \cup Y^{-1})^*$ *and a sequence of words* $a_1, a_2, \ldots, a_m \in (Y \cup Y^{-1})^*$ *which generate a subgroup* $A = \langle a_1, a_2, \ldots, a_m \rangle$ *of* $G$*.*
*DECIDE: Does* $w$ *represent an element of* $A$ *in* $G$*?*

*The hurdle in an attempt to show that* $\mathsf{GenWord}(H) \Bumpeq \mathsf{GenWord}(G)$ *is that although usually* $m \ll \kappa$*, there is no way of forcing this in the inequalities, and particular examples are constructible where* $m$ *is relatively large with respect to* $\kappa$*. I suspect that on average the two problems are equivalent, which brings up:*

**Question 2** *Is it true that* $\mathsf{GenWord}(H) \Bumpeq \mathsf{GenWord}(G)$ *on average? Furthermore, is it true that* $\mathsf{GenWord}(H) \Bumpeq \mathsf{GenWord}(G)$*?*

I also suspect that there is a better automata theoretic proof of proposition 2.3.5. This viewpoint would probably give rise to a polynomial time translation from $\mathsf{RatUnity}(H)$ to $\mathsf{RatUnity}(G)$:

**Question 3** *Is there a polynomial time algorithm which implements the algorithm of corollary 2.3.7 for automata instead of regular expressions? Is there a constant* $r \geq 1$ *such that if* $\mathsf{RatUnity}(H) \preceq f(x)$ *then* $\mathsf{RatUnity}(G) \preceq f(x^r)$*?*

The last question has to do with generalizing proposition 2.3.6. Perhaps an answer exists in the literature but I have not yet run across it. Proposition 2.3.6 shows that if $S \in \mathrm{Rat}(G)$ and $S \subseteq N \lhd G$ then $S \in \mathrm{Rat}(N)$. What if the conditions on $N$ and $G$ are relaxed?

**Question 4** *Let* $M_1$ *be a submonoid of the monoid* $M_2$*. Let* $S \in \mathrm{Rat}(M_2)$ *be a subset of* $M_1$*. Is* $S \in \mathrm{Rat}(M_1)$*? If this is not the case, how about when* $M_1$ *or* $M_2$ *is a group, or when both are groups but* $M_1$ *is not necessarily normal?*

We conclude this subsection by giving the delayed proof of proposition 2.3.6.

*Proof of proposition 2.3.6.* We give a proof for the case that $\mathcal{U}$ and $\mathsf{SubgpTrans}(G, N; \mathcal{U})$ are recursive. For the more general case, use the axiom choice instead of recursiveness whenever recursiveness is called upon.

Prove by induction on the length of the regular expression $\mathcal{C}$. If the length is 0 or 1, then $\mathcal{C}$ is finite. The given algorithm for $\mathsf{SubgpTrans}(G, N; \mathcal{U})$ produces the required representation of $\sigma(\mathcal{C})$.

Therefore, suppose that the length of $\mathcal{C}$ is at least 2. Let $S = \sigma(\mathcal{C})$. We are assuming that $S \subseteq N\sigma(u)$. There are three cases:

1. $\mathcal{C} = \mathcal{C}_1^*$
2. $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$
3. $\mathcal{C} = \mathcal{C}_1\mathcal{C}_2$

1. Certainly $\varepsilon \in \mathcal{C}$ so $1 \in S$. As $S$ is contained in a single $N$-coset, it follows that $S \subseteq N$ and therefore $\sigma(\mathcal{C}_1) \subseteq N$. By induction, we can find a regular expression $\mathcal{C}_1'$ on $Z$ such that $\sigma(\mathcal{C}_1') = \sigma(\mathcal{C}_1)$. Therefore, by letting $\mathcal{C}' = (\mathcal{C}_1')^*$ we obtain the desired representation for $\sigma(\mathcal{C})$.

2. We must have $\sigma(\mathcal{C}_i) \subseteq N\sigma(u)$ for $i = 1, 2$. By induction we can find regular expressions $\mathcal{C}_i'$ on $Z$ for which $\sigma(\mathcal{C}_i')\sigma(u) = \sigma(\mathcal{C}_i)$. Letting $\mathcal{C}' = \mathcal{C}_1' \cup \mathcal{C}_2'$ we have

$$\sigma(\mathcal{C}')\sigma(u) = [\sigma(\mathcal{C}_1' \cup \mathcal{C}_2')]\sigma(u) = \sigma(\mathcal{C}_1')\sigma(u) \cup \sigma(\mathcal{C}_2')\sigma(u) = \sigma(\mathcal{C}_1) \cup \sigma(\mathcal{C}_2) = \sigma(\mathcal{C})$$

which completes the induction step for case 2.

3. Let $S_i = \sigma(\mathcal{C}_i)$ for $i = 1, 2$. Notice that there are unique $u_1, u_2 \in \mathcal{U}$ such that

$$S_i \subseteq N\sigma(u_i) \text{ for } i = 1, 2, \text{ and } N\sigma(u_1)\sigma(u_2) = N\sigma(u) \ . \tag{2.3.4}$$

Otherwise, by symmetry (since $N$ is normal!) we can assume that there are $u_1 \neq u_1'$ in $\mathcal{U}$ and $h, h' \in N$ such that $h\sigma(u_1), h'\sigma(u_1') \in S_1$. Choosing any $s \in S_2$ implies that $h\sigma(u_1)s$, and $h'\sigma(u_1')s$ are in the same $N$-coset. Therefore $h\sigma(u_1)$ and $h'\sigma(u_1')$ are in the same coset, contradicting the fact that $u_1$ and $u_1'$ are different elements of the right transversal $\mathcal{U}$. This establishes equation (2.3.4). Notice that the $u_i$ in (2.3.4) are computable. For example, choose any word $w$ generated by the regular expression $\mathcal{C}_1$. The algorithm for $\mathsf{SubgTrans}(G, H; \mathcal{U})$ translates $w$ to some word of the form $w'u_1$ with $u_1$ exactly what we're after. Therefore, we have reduced the problem to finding regular expressions $\mathcal{C}_i'$ such that $\sigma(\mathcal{C}_i')\sigma(u_i) = \sigma(\mathcal{C}_i)$ for $i = 1, 2$. By induction, we can solve this problem. ♦2.3.6

### 2.3.2 Computing angles

Proposition 2.1.7 implies that if $G$ satisfies

1. for $S \in \mathrm{Rat}(G)$, $G - S$ is rational and computable,

2. and $\mathsf{RatUnity}(G)$ is solvable,

then $\mathsf{Angle}(G)$ and $\mathsf{Angle}'(G)$ are computable. Theorem 2.3.3 implies that the second property is preserved under passing to a finite index overgroup. Let's show that the first property is preserved as well:

**Theorem 2.3.10** *Suppose that $H$ is finitely generated and that $H \underset{n}{<} G$. Suppose that* $\mathrm{Rat}(H)$ *is closed under taking complements, and that* $\mathsf{RatComp}(H)$ *is solvable. Then the same holds for $G$.*

*Proof.* Let $\mathcal{A}$ be an automaton labeled by $X$, a symmetric set of generators for $G$, and let $A = \sigma(\mathcal{A})$. Let $Z$ be a symmetric set of generators for $H$ and let $\mathcal{U} = \{u_1, u_2, \ldots, u_n\} \subset X^*$ be a right transversal for $H\backslash G$ with $g_i = \sigma(u_i)$ and $u_1 = \varepsilon$. We have

$$G - A = \bigcup_{i=1}^{n} (Hg_i - A \cap Hg_i) = \bigcup_{i=1}^{n} (H - Ag_i^{-1} \cap H)g_i \ .$$

Because $H$ is of finite index, for all $i$, $Ag_i^{-1} \cap H \in \mathrm{Rat}(G)$ and $Ag_i^{-1} \cap H \subseteq H$. Furthermore, there are computable automata $\mathcal{A}_i$ obtained by taking the pull-back of $\mathcal{A}u_i^{-1}$ with $\Gamma_X(H\backslash G)$ and such that $\sigma(\mathcal{A}_i) = Ag_i^{-1} \cap H$. Corollary 2.3.8 implies that $\mathcal{A}_i$ may be translated to automata $\mathcal{A}_i'$ labeled by $Z$ such that $\sigma(\mathcal{A}') = \sigma(\mathcal{A})$. By assumption, the $\mathcal{A}_i'$ may be complemented to obtain automata $\neg\mathcal{A}_i'$ such that $\sigma(\neg\mathcal{A}_i') = H - \sigma(\mathcal{A}_i') = H - Ag_i^{-1} \cap H$. Letting $\neg\mathcal{A} = \bigcup_{i=1}^{n} \neg\mathcal{A}_i' \cdot u_i$ we see that $\neg\mathcal{A}$ is an automaton whose image is the complement of $A$ in $G$. ♦

**2.3.11 Remark** *In section 2.2 we saw that if $H$ is a finitely generated free group or free abelian group, then $H$ satisfies the conditions of theorem 2.3.10. Therefore, proposition 2.1.7 implies that angles are computable in groups which are either virtually free abelian or virtually free. Here finite generation is unnecessary (if there is a way of dealing with elements of the group in a computable fashion) as computing angles can always be carried out in a finitely generated restriction and any finitely generated subgroup of a virtually free (resp. virtually abelian) group is itself virtually free (resp. virtually abelian). Finally, the structure theorem for abelian groups implies that any finitely generated abelian group is virtually free abelian. Thus any virtually abelian group is virtually free abelian. Summarizing:*

**Corollary 2.3.12** *Suppose that $G$ is virtually free or virtually abelian. Then* $\mathsf{Angle}(G)$ *and* $\mathsf{Angle}'(G)$ *are recursively solvable.*

It would be nice to have a larger class of groups in which angles are computable. As above, a good answer to the following question would lead to more examples of such groups:

**Question 5** *What conditions on $G$ guarantee that* $\mathsf{RatUnity}(G)$ *is solvable and that* $\mathrm{Rat}(G)$ *is closed under complements?*

### 2.3.3 Unambiguity

Next we show that property 3 of section 2.2 is preserved under finite extensions:

**Theorem 2.3.13** *Suppose that $H \underset{\text{f.i.}}{<} G$ and that* $\mathrm{UR}(H) = \mathrm{Rat}(H)$*. Then* $\mathrm{UR}(G) = \mathrm{Rat}(G)$ *so that unambiguity of all rational subsets is preserved under finite extensions.*

*Proof.* Let $S \in \mathrm{Rat}(G)$. Let $\mathcal{A}$ be an automaton representing $S$, i.e. $\sigma(\mathcal{A}) = S$. Then letting $\mathcal{A}_i = \mathcal{A} \cap \sigma^{-1}(Hg_i)$ we have

$$\mathcal{A} = \bigcup_{i=1}^{n} \mathcal{A}_i \quad \text{and} \quad S = \sigma(\mathcal{A}_1) \uplus \sigma(\mathcal{A}_2) \uplus \cdots \uplus \sigma(\mathcal{A}_n) \ . \tag{2.3.5}$$

Corollary 2.3.8 implies that there are automata $\mathcal{A}_i' \subseteq Z^*$ with the property that $\sigma(\mathcal{A}_i')g_i = \sigma(\mathcal{A}_i)$. Letting $S_i = \sigma(\mathcal{A}_i')$ we have $S_i \in \mathrm{Rat}(H)$ and therefore, as all rational subsets of $H$ are unambiguous, $S_i \in \mathrm{UR}(H) \subset \mathrm{UR}(G)$. Therefore, by equation (2.3.5) surmise that

$$S = (S_1 \odot g_1) \uplus (S_2 \odot g_2) \uplus \cdots \uplus (S_n \odot g_n)$$

is unambiguous, because the union and products are unambiguous. ♦

**2.3.14 Remark** *The proof of theorem 2.3.13 is constructive in the following sense: if there is an algorithm for transforming regular expressions to unambiguous regular expressions in* $\mathrm{Rat}(H)$*, such an algorithm extends to* $\mathrm{Rat}(G)$*.*

**Corollary 2.3.15** *Suppose that $G$ is virtually free or virtually abelian. Then* $\mathrm{UR}(G) = \mathrm{Rat}(G)$ *so all rational subsets of $G$ are unambiguous.*

*Proof.* Let $S \in \mathrm{Rat}(G)$. As only finitely many generators of $G$ appear in a representation of $S$, it follows that $S$ is contained in a finitely generated subgroup $G_1 < G$. As being virtually free (resp. virtually abelian) is inherited by subgroups, without loss of generality, $G$ is finitely generated. Thus $G$ contains a subgroup $H$ of finite index which is free (resp. free abelian) so has the property $\mathrm{UR}(H) = \mathrm{Rat}(H)$ by results of §2.2. Apply theorem 2.3.13 to conclude that $\mathrm{UR}(G) = \mathrm{Rat}(G)$. ♦

### 2.3.4 Profinite closure

We conclude the chapter by examining algorithms for closures of rational subsets.

**Theorem 2.3.16** *Suppose that $H$ is finitely generated and that $H \underset{n}{<} G$. Suppose that* $\mathsf{RatClos}(H)$ *is recursively computable. Then* $\mathsf{RatClos}(G)$ *is recursively solvable.*

*Proof.* Let $\mathcal{A}$ be an automaton labeled by $X$, a symmetric set of generators for $G$, and let $A = \sigma(\mathcal{A})$. Let $Z$ be a symmetric set of generators for $H$, let $\mathcal{U} = \{u_1, u_2, \dots, u_n\} \subset X^*$ be a right transversal for $H\backslash G$ with $g_i = \sigma(u_i)$ and $u_1 = \varepsilon$, and let $\mathcal{B} = \Gamma_X(H\backslash G)$ —an automaton accepting the language $\sigma^{-1}(H)$. We have

$$A = \bigcup_{i=1}^{n} A \cap Hg_i = \bigcup_{i=1}^{n} (Ag_i^{-1} \cap H)g_i \ . \tag{2.3.6}$$

Furthermore, as right multiplication induces a homeomorphism of $G$, we have $\overline{(Ag_i^{-1} \cap H)g_i} = \left(\overline{Ag_i^{-1} \cap H}\right)g_i$. Now $H$ is closed as it is of finite index; therefore, the closure of $Ag_i^{-1} \cap H$ in $H$ is the same as its closure in $G$. Notice that $Ag_i^{-1} \cap H \in \mathrm{Rat}(G)$ and that an automaton in $X^*$ representing this set is constructible. Indeed, letting $\mathcal{A}'_i = \mathcal{A}u_i^{-1} \cap \mathcal{B}$, because $\mathcal{B}$ accepts the full pre-image of $H$ in $X^*$ it follows that $\mathcal{A}'_i$ is an automaton whose accepted language has the image $\sigma(\mathcal{A}'_i) = Ag_i^{-1} \cap H$. Therefore, equation (2.3.6) and the distributivity of closure over finite unions implies that $\overline{A}$ is the union of translates of the closures of $Ag_i^{-1} \cap H$ inside of $H$. By assumption $\mathsf{RatClos}(H)$ is solvable so it suffices to find automata $\mathcal{A}''_i$ over the $Z$ for which $\sigma(\mathcal{A}''_i) = Ag_i^{-1} \cap H$. This can be done by corollary 2.3.8. ♦

Steinberg showed that $\mathsf{RatClos}(G)$ is computable when $G$ is free [54] or free abelian [53]. Therefore:

**Corollary 2.3.17** *Suppose that $G$ is virtually free or virtually abelian, and is finitely generated. Then* $\mathsf{RatClos}(G)$ *is recursively solvable.*

# Chapter 3

# Algorithms for Hyperbolic Groups

Word hyperbolic groups were introduced by Gromov [28]. Recommended sources for learning about word hyperbolic groups are [3] and [22]. The next two sections are based on the [3], though most of the proofs which exist in this reference have been omitted below. Sections 3.3 through 3.6 are based on [29], with some additions and improvements. Section 3.7 contains the result promised [29]: That virtually free groups are "super locally quasiconvex" (see definition 3.5.3).

## 3.1 Four Classical Notions of Hyperbolicity

Recall the definition of a Cayley graph of a group $G$ with respect to a generating set $X$ (§1.5.3). The geometric realization $||\Gamma_X(G)||$ is a path connected topological space. It is possible to endow $||\Gamma_X(G)||$ with a metric $d_X$ by declaring all edges to have length 1: Then any combinatorial[1] path $\gamma$ has **length** denoted by $|\gamma|$ which is just the number of edges traversed by $\gamma$. Distance is defined between any two *vertices* by taking the infimum of lengths of all paths between the vertices. It is not hard to extend this distance to points on the interiors of edges. In fact, the metric is **geodesic**: i.e., between any two points $p, q \in ||\Gamma_X(G)||$ there is a path $\gamma$ —called a geodesic— whose length is exactly the distance $d_X(p, q)$. This is true even when $X$ is infinite so that $||\Gamma_X(G)||$ is not locally finite. A recommended source containing a proof of this fact —actually a generalization to simplicial

[1] A **combinatorial path** is a path that starts and ends at vertices and progresses at unit speed without reversing direction in mid-edge. See also definition 1.3.1.

complexes— is Bridson's treatise [7, cf. theorem 1.1].

The notation $[p, q]$ represents a choice of some geodesic $\gamma$ from $p$ to $q$. The **ball** of radius $K$ about 1 is denoted by

$$B_K = \{x \in \Gamma_X(G) \mid d_X(1, x) \leq K\} .$$

A **triangle** is a union of three geodesics drawn between three points in the space.

### 3.1.1 Slim triangles criterion

$G$ is termed a **hyperbolic group** if there is a finite generating set $X$ such that $\Gamma_X(G)$ is a hyperbolic metric space where —unless explicitly told otherwise— the Rips **slim triangles** hyperbolicity criterion is used:

**Definition 3.1.1** *A geodesic metric space* $(\Gamma, d)$ *is* **hyperbolic** *if there is a universal* **hyperbolicity constant** $\delta$ *such that given any geodesic triangle in* $\Gamma$*, each side of the triangle is contained inside the* $\delta$*-neighborhood of the other two sides. The group* $G$ *is said to be* $\delta$*-hyperbolic with respect to the generating set* $X$*, if* $||\Gamma_X(G)||$ *is a hyperbolic space with universal constant* $\delta$*.*

### 3.1.2 Thin triangles criterion

It is possible to put a stricter requirement —that of **thin triangles**— on a metric space. Each pair of sides is required to $\delta$-fellow travel (see definition 3.6.1 below) for a prescribed distance. More precisely:

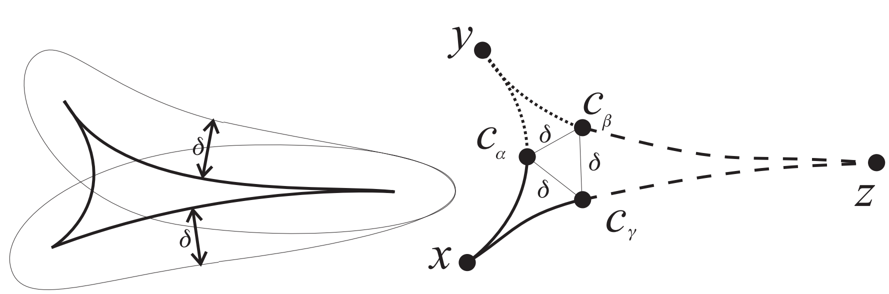



Figure 3.1: $\delta$-slim vs. $\delta$-thin.

**Definition 3.1.2** ([3, p. 10–11]) *Let* $(\Gamma, d)$ *be a geodesic metric space, and* $x, y, z \in \Gamma$. *Consider the triangle* $T$ *formed by the geodesics* $\alpha = [x, y], \beta = [y, z]$ *and* $\gamma = [x, z]$. *There are unique points* $c_\alpha \in \alpha$, $c_\beta \in \beta$ *and* $c_\gamma \in \gamma$ *such that* $d(x, c_\alpha) = d(x, c_\gamma)$, $d(y, c_\alpha) = d(y, c_\beta)$, *and* $d(z, c_\beta) = d(z, c_\gamma)$. *The vertex* $x$ *is called* $\delta$*-thin if for all* $u \in [x, c_\alpha], u' \in [x, c_\gamma]$ *such that* $d(x, u) = d(x, u')$ *then* $d(u, u') \leq \delta$. *Similar definitions are given for the other two vertices* $y$ *and* $z$. *The triangle* $T$ *is called* $\delta$*-thin if all its vertices are* $\delta$*-thin.*

**Lemma 3.1.3 ([3, p. 17])** *Let* $(\Gamma, d)$ *be a* $\delta$*-slim hyperbolic metric space (definition 3.1.1). Then all geodesic triangles of* $\Gamma$ *are* $6\delta$*-thin. Any* $\delta$*-thin triangle is also* $\delta$*-slim.*

By definition, hyperbolic groups only need be endowed with one hyperbolic Cayley graph. It turns out, however, that if one such Cayley graph exists then any finite generating set induces a hyperbolic Cayley graph for the group [22, p. 89], although the hyperbolicity constant is likely to change.

### 3.1.3 The Dehn property of a graph

The next formulation of hyperbolicity, is motivated by an algorithm for solving the word problem. The "obvious" algorithm for solving $\mathsf{Word}(G)$ would go as follows: Starting with a presentation $P = \langle\ Y \mid R\ \rangle$ for $G$, consider any word $w \in X^*$ where $X = Y \cup Y^{-1}$. Now search $w$ for subwords which happen to be more than half of some relator $r \in R$. If such a subword is found, replace the subword by its complement[2] in $r$, resulting in a shorter word $w'$, and apply the same procedure to $w'$. Repeating this for as long as possible, we will either obtain the empty word $\varepsilon$, or at some point we'll get stuck with a word $w''$ containing no subword which is more than half of some relator. Such a word $w''$ is declared to be non-trivial. Furthermore, as replacing a sub-relator by its complement doesn't change the image of the word in $G$, we have $w =_G w''$ so that $w$ is declared to be non-trivial. On the other hand, if we obtained $\varepsilon$ at the very end of the process, then $w$ is declared trivial (and this declaration is always correct). This "obvious" algorithm is called **Dehn's algorithm** for the word problem, in honor of Max Dehn who first used it to solve the word problem for surface groups [12]. Though the algorithm is obvious, it is not obvious at all that the algorithm is correct. First of all, the algorithm seems highly dependent on the presentation

[2] Given a relator of the form $r = uvw$, the **complement** of $v$ in $r$ is the word $u^{-1}w^{-1}$.

given. Furthermore, there are groups whose word problem is undecidable, so surely Dehn's algorithm cannot possibly work for such groups! Amazingly, the groups for which Dehn's algorithm works (for some presentation) are precisely the word hyperbolic groups.

The algorithmic property above will be abstracted slightly so as to become a graph theoretic property, as opposed to a presentation-dependent property. The Dehn property is a strengthening of the following homogeneity criterion:

**Definition 3.1.4** *Let $(\Gamma, d)$ be a metric space. Furthermore, let $\Phi$ be a group of isometries acting on $\Gamma$ from the left, and let $K$ be a non-negative number. The isometry group $\Phi$ is a $K$-***homogeneous action** *if given any two points $x, y \in \Gamma$, there is an isometry $\phi \in \Phi$ such that $d(\phi(x), y) \leq K$. That is, any point can be carried within $K$ of any other point by some isometry in $\Phi$. If $\Gamma$ admits a $K$-homogeneous action for some $K$, then we say that $\Gamma$ is a* **homogeneous space**. *If the group of isometries on $\Gamma$ is 0-homogeneous, then $\Gamma$ is* **totally homogeneous**.

Left multiplication in $G$ induces graph automorphisms on $\Gamma_X(G)$ which can take any vertex to any other vertex. Furthermore, graph automorphisms are isometries relative the metric induced from setting the lengths of all edges to 1. Finally, any point in the interior of an edge is within distance $\frac{1}{2}$ of some vertex. Therefore:

**Lemma 3.1.5** *$G$ is a $\frac{1}{2}$-homogeneous action on $||\Gamma_X(G)||$.*

A **loop** is a path $\gamma$ which starts and ends at the same point. It is often useful to forget about the loop's basepoint. Thus a path $\alpha$ may be a subpath of a loop $\gamma$ even though it is not a subpath of the *path* $\gamma$. For example consider the loop

$$\gamma = v_0 \to v_1 \to v_2 \to v_3 \to v_4 \to v_0 \ .$$

Then $\alpha = v_3 \to v_4 \to v_0 \to v_1$ is a valid subpath of the *loop* $\gamma$, but not of the path. In a topological graph (so each edge has an inverse) the **complement** in a loop $\gamma$ of a path $\alpha$ is the unique path $\alpha'$ such that $\alpha \cdot (\alpha')^{-1}$ is a loop which can be cyclically permuted to obtain $\gamma$. In our case the complement of $\alpha$ is given by $\alpha' = v_3 \to v_2 \to v_1$ (see figure 3.2).

With the notion of a loop and loop complements under our belts, we can define the Dehn property:

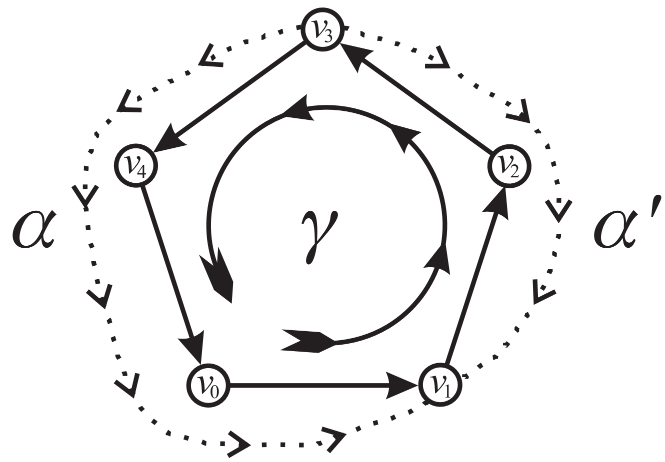



Figure 3.2: A loop $\gamma$, a subpath $\alpha$ and its complement $\alpha'$.

**Definition 3.1.6** *Let $\Gamma$ be a graph with a vertex $*$ and metric $d$, $K$ a positive number, and let $\Omega_K$ be the set of all combinatorial loops of length no larger than $K$ which are based at $*$. Let $\Phi$ be a group of graph automorphisms acting from the left on $\Gamma$. The pair $(\Omega_K, \Phi)$ is said to be a* **Dehn atlas** *for $\Gamma$ if given any loop $\gamma$ in $\Gamma$, there is a loop $\beta$ in $\Omega_K$ and a combinatorial subpath $\alpha$ of $\beta$ such that:*

- $|\alpha| > \frac{1}{2}|\beta|$, *i.e., $\alpha$ is more than half of the loop $\beta$,*
- *and there is a graph automorphism $\phi \in \Phi$ such that $\phi(\alpha)$ is a subpath of $\gamma$.*

*$\Gamma$ is said to have the* **Dehn property** *if it admits a Dehn atlas. A group $G$ is said to have the Dehn property if its topological Cayley graph $||\Gamma_X(G)||$ has the Dehn property with respect to $\Phi = G$.*

The following is a reformulation of [3, theorem 2.12]:

**Theorem 3.1.7** *Let $G$ be a hyperbolic group in the sense that all triangles of $||\Gamma_X(G)||$ are $\delta$-thin. Then $(\Omega_{8\delta}, G)$ is a Dehn atlas for $||\Gamma_X(G)||$. Therefore, hyperbolic groups have the Dehn property.*

In the next subsection we will see that the converse is true as well so that hyperbolicity can be defined in terms of the Dehn property.

### 3.1.4 Linear isoperimetric inequality

For the last definition of hyperbolicity, the notion of isoperimetric functions is required. Classically, the isoperimetric problem consists of bounding the most area with the least amount of fencing. This is the idea behind the formal definition. Given a metric space where a notion of area enclosed by loops is well defined, an isoperimetric function is a measure of the maximal area that is enclosed by loops of a given length. We need a way of measuring areas inside graphs. One way to handle this problem is by adding 2-cells to the graph and creating a simply connected 2-complex. Then the area of a loop is exactly the minimal number of cells required to homotope the loop down to a constant path. The Greek letter "$\Sigma$" is used to denote 2-complexes because of the letter "S" in "surface". The 2-complex $\Sigma$ will mostly not be a surface in the usual sense of the word as the links of vertices can be much more complicated than intervals and circles.

**Definition 3.1.8** *Let $\Sigma$ be a simply connected locally finite 2-complex. Let $\gamma$ be a loop in the 1-skeleton $\Sigma^{(1)}$. The area of a homotopy $H : [0,1] \times [0,1] \to \Sigma$ between $\gamma$ and a constant path is the number of interiors of 2-cells intersected by the image of $H$. The* **area** *of $\gamma$ is the minimum area over all possible null-homotopies for $\gamma$. I.e.,*

$$\operatorname{area}(\gamma) = \min\{\operatorname{area}(H) \mid H \textit{ is a null-homotopy map for } \gamma\} \ . \tag{3.1.1}$$

*Consider the formula*

$$f_\Sigma(n) = \max\{\operatorname{area}(\gamma) \mid \gamma \textit{ is a loop in } \Sigma \textit{ s.t. } |\gamma| \leq n\} \ . \tag{3.1.2}$$

*If $f_\Sigma$ is well defined so that for every length there is a maximal area for loops of that given length, then $f_\Sigma$ is called the* **isoperimetric function** *for $\Sigma$. The 2-complex $\Sigma$ is said to have* **linear isoperimetric inequality** *if there is a constant $K$ such that*

$$f_\Sigma(n) \leq Kn \textit{ for all } n \in \mathbb{N} \ . \tag{3.1.3}$$

Now consider any presentation $P = \langle\ Y \mid R\ \rangle$ for a group $G$ with $Y$ finite. As usual, set $X = Y \cup Y^{-1}$. It is possible to expand the topological Cayley graph $||\Gamma_X(G)||$ to a **Cayley 2-complex** $\Sigma_P(G)$ by gluing a 2-cell labelled by $r$ for all relators $r \in R$ and all vertices $g \in G$: i.e., glue the boundary of the 2-cell starting from the vertex $g$ to a loop whose label in $\Gamma_X(G)$ spells out the word $r$. This construction is well defined since relations are the labels of loops in the Cayley graph. As any loop in $||\Gamma_X(G)||$ defines a trivial word in $G$, and any trivial word is the product of conjugates of relators in $\mathrm{FG}(Y)$, it follows that $\Sigma_P(G)$ is simply connected. Another way to see that the $\Sigma_P(G)$ is simply connected is to observe that $\Sigma_P(G)$ is the universal covering of the standard 2-complex for the presentation $P$ obtained by taking a bouquet on $Y$ and gluing 2-cells according to the relators in $R$. Furthermore, $\Sigma_P(G)$ is homogeneous so that for any loop $\gamma$ of length $n$, the area of $\gamma$ is determined by the area of a corresponding loop based at $*$ (of which there are only finitely many, since $Y$ is finite). Thus up to isometry, there are only finitely many loops in $\Sigma_P(G)$ of a given length so that given any $n$ the maximum area over all loops of length $n$ is well defined. Summarizing, given any presentation $P$ (even with infinitely many relations) of a finitely generated group $G$, the isoperimetric function $f_P = f_{\Sigma_P(G)}$ is well defined. Furthermore, if $G$ is hyperbolic, then $f_P$ is bounded by a linear function:

**Theorem 3.1.9 (**[3, theorem 2.16]**)** *Let $G$ be a group such that $||\Gamma_X(G)||$ has the Dehn property with atlas $(\Omega_K, G)$. Let $R = \lambda(\Omega_K)$ be the set of labels of the loops in $\Omega_K$. Then $P = \langle\ Y \mid R\ \rangle$ is a presentation for $G$ with the property that*

$$f_P(n) \leq n\ . \tag{3.1.4}$$

*Therefore, groups with the Dehn property have linear isoperimetric inequality.*

**3.1.10 Remark** *The presentation $P = \langle\ Y \mid R\ \rangle$ with $R$ constructed as in theorem 3.1.9 is called a* **Dehn presentation**. *Thus the theorem states that a Dehn presentation has an isoperimetric inequality which is bounded by the identity function* $\mathrm{id}_{\mathbb{N}}$.

*Proof.* Let $\gamma$ be a loop in $\Gamma_X(G)$. Prove by induction on $n$ that $\gamma$ is null-homotopic in $\Sigma_P(G)$ via a homotopy of area no greater than $n$. The base case is $n = 0$ and there is nothing to prove in this case. Suppose the statement has been proved for all $m < n$. Since $\Gamma_X(G)$ has the Dehn property with constant $K$ there is a loop $\beta$ in $\Omega_K$ and a *combinatorial* subpath $\alpha$ of $\beta$ such that $|\alpha| > \frac{1}{2}|\beta|$ and left multiplication sending $\alpha$ to a subpath $g\alpha$ of $\gamma$. Let $\alpha'$ be the complement of $\alpha$ in $\beta$. We have that $|\alpha'| < |\alpha|$. Furthermore, let $\gamma'$ be the loop obtained by replacing $g\alpha$ by $g\alpha'$. We have that $|\gamma'| < |\gamma| = n$. Thus by induction, $\mathrm{area}(\gamma') \leq n-1$. By definition, there is a 2-cell $E$ in $\Sigma_P(G)$ whose boundary is $\beta$.

Consequently, $gE$ has $g\beta$ for its boundary. Therefore, any null-homotopy for $\gamma'$ in $\Sigma_P(G)$ can be extended to a null-homotopy of $\gamma$ by gluing the homotopy for $\gamma'$ with a homotopy from $g\alpha'$ to $g\alpha$ across $gE$. The area of this homotopy for $\gamma$ is one greater than the area of the homotopy for $\gamma'$ showing that $\text{area}(\gamma) \leq \text{area}(\gamma') + 1 \leq n$. This completes the proof. ♦

Theorems 3.1.7 and 3.1.9 give the following corollary:

**Corollary 3.1.11** *Let $G$ be a hyperbolic group in the sense that all triangles of $\Gamma_X(G)$ are $\delta$-thin. Then, $G$ has linear isoperimetric inequality.*

Finally we quote without proof the fact that linear isoperimetric inequality implies the $\delta$-slim property. Therefore, the above theorems show that $\delta$-slimness, $\delta$-thinness, the Dehn property, and linear isoperimetric inequality all characterize hyperbolic groups.

**Theorem 3.1.12 (**[3, theorem 2.5]**)** *If $G$ is a finitely presented group satisfying a linear isoperimetric inequality, then there is a constant $\delta$ such that all geodesic triangles in $\Gamma_X(G)$ are $\delta$-slim.*

## 3.2 A Language Theoretic Characterization

In this section we continue investigating the connections between formal languages and groups.

### 3.2.1 The word problem viewed as a language

There is a standard way of converting any decision problem to the membership problem in some formal language. First of all, the inputs are encoded as words in a finite alphabet $X$. It is assumed that the language $\mathcal{I} \subseteq X^*$ of all valid inputs —i.e. words which correspond to inputs of the decision problem— is recursive so that it can be ascertained whether or not an arbitrary string in $X^*$ corresponds to an input; furthermore, consider

$$\mathcal{I}_1 = \{w \in \mathcal{I} \mid \text{ if } w \text{ is inputted, then the output is 1 —i.e., “Yes” } \}$$

which is the language of all words representing inputs for which the decision on the input is “Yes”. Deciding whether or not a particular word $w \in \mathcal{I}$ is in $\mathcal{I}_1$ is equivalent to solving the original decision problem. For example, converting $\mathsf{Word}(G)$ (or any of the other decision problems considered above) from a decision problem to a language membership problem is immediate since $\mathsf{Word}(G)$ is already couched in terms of languages. Indeed, $\mathsf{Word}(G)$ is the problem of deciding whether or not a word $w \in X^*$ represents the identity, which is exactly the problem of deciding whether or not $w \in \sigma^{-1}(1)$, where $\sigma : X^* \twoheadrightarrow G$ is the canonical generating homomorphism.

Thus it makes sense to abuse notation and refer to "the word problem" as a language, specifically the language $\sigma^{-1}(1)$, and not only as a decision problem. As a language, the word problem is highly dependent on the choice of generators used. For example, the word problem of the trivial group $\{1\}$ is $X^*$ for any choice of generators $X$. As free monoids of different rank are not isomorphic, the word problem of $\{1\}$ varies according to the cardinality of $X$. Nevertheless, if we consider certain classes of languages, such as some of the classes mentioned in section 1.8, then the type of language that the word problem resides in will not depend on the set of finite generators chosen. For example, any finitely generated and recursively related group has recursively enumerable word problem since it is possible to list any trivial word by listing all products of conjugates of relations and all words freely equivalent with such words; therefore, the recursive enumerability of the word problem does not depend on the set of generators chosen. Similarly, as the existence of a solution to the word problem does not depend on the choice of finite generators, having recursive word problem is independent of generators. In all the language classes mentioned below, the type of word problem will be independent of the choice of generators.

### 3.2.2 Some previous classifications of word problems

Anisimov [4] started investigating the connection between classes of groups and classes of languages; in Anisimov's article it was established that a finitely generated group has regular word problem if and only if the group is finite. Muller and Schupp [43] continued along these lines by investigating groups with context free word problem. They showed that a finitely generated accessible group has context free word problem if and only if the group is virtually free. Later, Dunwoody [14] showed that all finitely presented groups are accessible, which eliminated the need for the accessibility condition. Herbst [32] specialized Muller and

Schupp's result and showed that a finitely generated group has a one-counter[3] word problem if and only if the group is virtually cyclic. Recently Gilman and Shapiro [26] showed that a finitely generated accessible group has indexed[4] word problem if and only if the group is virtually free. As it is expected that the accessibility condition is unnecessary, Gilman and Shapiro's work will probably show that no new groups arise when passing from context free languages to the larger class of indexed languages. Theorem 3.2.1 below continues this line of research by establishing that a finitely generated group has growing and terminating word problem if and only if the group is word hyperbolic. Table 3.1 summarizes the current state of knowledge.

| The word problem of $G$ is... $\iff$ | $G$ is... | Reference |
|---|---|---|
| regular | finite | Anisimov [4] |
| one counter | virtually cyclic | Herbst [32] |
| context free | virtually free | Muller & Schupp [43] |
| indexed (with $G$ accessible) | virtually free | Gilman & Shapiro [26] |
| growing & terminating | hyperbolic | theorem 3.2.1 |

Table 3.1: Relationship between language classes and group classes.

Applying theorem 3.2.1 implies that finitely generated groups with growing and terminating word problem are finitely presented[5] and therefore accessible, by Dunwoody's result [14]. Proposition 1.8.4 showed that context free languages are growing and terminating. Therefore, a corollary of theorem 3.2.1 is that groups with context free word problem are accessible. Similarly, if it is the case that indexed languages are also growing and terminating then the accessibility condition in [26] can be dropped. There is some hope that this might be the case as indexed languages are context sensitive (cf. [1]). This leads to:

**Question 6** *Suppose that $\mathcal{L}$ is an indexed language. Is there a growing and terminating grammar which generates $\mathcal{L}$?*

[3] A language is one-counter if it is accepted by a pushdown automaton using only one stack symbol. See [33] for more details.

[4] Indexed languages were introduced by Aho in [1] and [2]. A grammar-like definition of indexed languages can be found in [34, §14.3], and an elegant automata theoretic definition exists in [26].

[5] Actually, this follows from equation 3.2.1 which follows immediately from the definition of growing and terminating languages. The terminating condition is of main import for showing finite presentedness. The additional growing condition implies the linear isoperimetric inequality.

### 3.2.3 The word problem for hyperbolic groups

Next we prove that it is possible to characterize word hyperbolic groups in a language theoretic manner. Namely, a group is word hyperbolic if and only if its word problem is growing and terminating. I would like to express my gratitude to Bob Gilman for pointing out to me —in an email correspondence— what is basically one half of my proof: Hyperbolic groups have growing word problem. My contribution was the simple observation that the word problem is terminating and that growing and terminating suffices to imply hyperbolicity.

Notice that in the following theorem, it is not necessary to assume that $X$ is a symmetric set of monoid generators for $G$.

**Theorem 3.2.1** *Let $G$ be a group with finite monoid generating set $X$ such that $\sigma : X^* \twoheadrightarrow G$. Then $G$ is word hyperbolic if and only if $\sigma^{-1}(1)$ is a growing and terminating language.*

*Proof.* If $\sigma^{-1}(1)$ is growing and terminating, then show that $G$ has linear isoperimetric inequality: Suppose that $\sigma^{-1}(1)$ is the language generated by a growing and terminating grammar $\rightrightarrows$ with variables $V$ and terminals $X$ (see §1.8.5). First assume that $X = Y \cup Y^{-1}$ is symmetric. Without loss of generality, we may assume that the left hand side of each production in the grammar appears in some word which is derivable from $\mathsf{s}$. Otherwise the production is extraneous and may be dropped. As the grammar is terminating, for each $\mathsf{a} \in V - \{\mathsf{s}\}$ there is a word $w_{\mathsf{a}} \in W^+$ for which the production $\mathsf{a} \rightrightarrows w_{\mathsf{a}}$ exists; furthermore, set $w_{\mathsf{s}} = \varepsilon$ and for each terminal $x \in X$ set $w_x = x$ so that $w_a$ is defined for all letters in the alphabet $V \cup X$. Therefore, a unique homomorphism is obtained:

$$\eta : (V \cup X)^* \to X^* \text{ s.t. } \eta(a) = w_a \text{ for all } a \in V \cup X \ .$$

In fact, $G$ has the presentation

$$P = \langle\, Y \mid R \,\rangle \quad \text{with} \quad R = \{\, \eta(\alpha)^{-1}\eta(\beta) \mid \alpha \rightrightarrows \beta \text{ or } \beta \rightrightarrows \alpha \,\} \ . \tag{3.2.1}$$

Notice that $R$ is symmetric, i.e., $R^{-1} = R$. To see that $\eta(\alpha)^{-1}\eta(\beta)$ is a relator for each production $\alpha \rightrightarrows \beta$, find a word $u$ derivable from $\mathsf{s}$ and such that $u = v_1\alpha v_2$. Let $u' = v_1\beta v_2$. Applying $\eta$ to each of these words results in a terminal word derivable from $\mathsf{s}$. In other words,

$$\eta(v_1)\eta(\alpha)\eta(v_2), \eta(v_1)\eta(\beta)\eta(v_2) \in \sigma^{-1}(1) \ .$$

Therefore, $\eta(v_1)\eta(\alpha)\eta(v_2) =_G \eta(v_1)\eta(\beta)\eta(v_2) =_G 1$ and after cancellation $\eta(\alpha) =_G \eta(\beta)$ so that $\eta(\alpha)^{-1}\eta(\beta)$ is a relator for $G$.

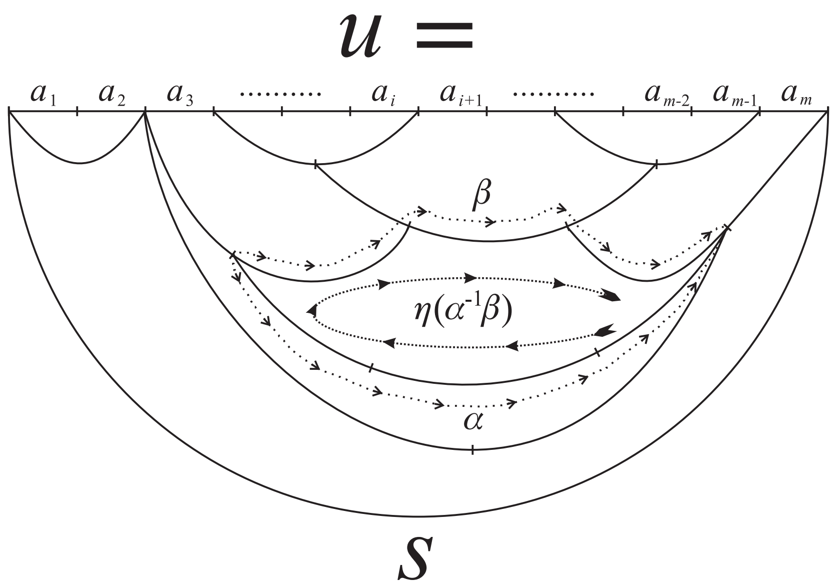



Figure 3.3: Growing and terminating word problem implies hyperbolic.

The figure shows how a derivation $s \rightsquigarrow^* u = a_1 a_2 \cdots a_m$ gives rise to a Dehn diagram. At a certain stage the production $\alpha \rightrightarrows \beta$ has been applied, and this gives rise to the disc $\eta(\alpha^{-1}\beta)$ in the Dehn diagram.

In fact, notice that the isoperimetric function for the presentation $P$ satisfies

$$f_P(n) \leq n \ .$$

This implies that $G$ is word hyperbolic since finitely presented groups with linear isoperimetric inequality are hyperbolic by theorem 3.1.12. Specifically we prove that if $u \in (V \cup X)^*$ is derivable from $\mathsf{s}$ in $n$ steps, then $\eta(u)$ is expressible in $\mathrm{FG}(Y)$ as a product of at most $n$ conjugates in $R$. As the grammar is growing, any non-empty word $w \in \sigma^{-1}(1)$ is derivable from $\mathsf{s}$ in at most $|w|$ steps; moreover, as $\eta(w) = w$ (since $w$ is terminal) it will follow that $w$ is expressible as a product of at most $|w|$ conjugates of relators proving that $f_P(n) \leq n$. So let's prove that $\eta(u)$ is the product no more than $n$ conjugates in $R$. Consider a derivation

$$\mathsf{s} \rightsquigarrow u_1 \rightsquigarrow u_2 \rightsquigarrow \cdots \rightsquigarrow u_{n-1} \rightsquigarrow u_n = u \ . \tag{3.2.2}$$

If $n = 1$ then $\eta(s)^{-1}\eta(u_1) = \varepsilon \cdot \eta(u) = \eta(u)$ is a relator. Now suppose that the claim has been proved up to $n-1$. Let $u_{n-1} = v_1\alpha v_2$ and $u_n = v_1\beta v_2$ with $\alpha \rightrightarrows \beta$. Therefore, in $\mathrm{FG}(Y)$ we have:

$$\eta(u_n) = \eta(v_1)\eta(\beta)\eta(v_2) = \eta(v_1)\eta(\alpha)\eta(v_2) \cdot (\eta(\alpha)^{-1}\eta(\beta))^{\eta(v_2)} \ . \tag{3.2.3}$$

Here we are using the exponentiation short-hand for conjugation; i.e. $u^v = v^{-1}uv$. By induction, $\eta(v_1)\eta(\alpha)\eta(v_2)$ is expressible as the product of at most $n-1$ conjugates of relators, and equation (3.2.3) implies that $\eta(v_1)\eta(\alpha)\eta(v_2) = \eta(u)$ is the product of one additional conjugate of a relator. This completes the proof that groups with growing and terminating word problem are hyperbolic, assuming that $X$ is symmetric.

If $X$ is a monoid generating set of $G$ which is not necessarily symmetric, the argument above shows that any *positive* word $w \in \sigma^{-1}(1)$ is expressible as the product of at most $|w|$ conjugates of relators in the set $R$ of equation (3.2.1). Consider the *disjoint* union $X' = X \cup X^{-1}$. Since $X$ generates $G$ as a monoid, for each $x \in X$ one can find a word $v_x \in X^*$ such that $x^{-1} =_G v_x$. Let $R' = \{\ x^{-1}v_x^{-1}, v_x x \mid x \in X\ \}$ and let $K = \max\{\ |v_x| \mid x \in X\ \}$. Notice that $G$ has the presentation $P' = \langle\ X \mid R \cup R'\ \rangle$. In fact, one can translate any word $w \in X'^*$ to a word of the form $w_1w_2$ where $w_1$ is a product of at most $|w|$ conjugates in $R'$ and $w_2 \in X^*$ is a positive word of length at most $K|w|$. For example, supposing $X = \{x, y, z\}$ and $w = xyx^{-1}z$ we have in the free group $\mathrm{FG}(X)$:

$$w = xyx^{-1}z = xyx^{-1}v_x^{-1}v_xz = (x^{-1}v_x^{-1})^{y^{-1}x^{-1}} \cdot xyv_xz = w_1 \cdot w_2$$

where $w_1 = (x^{-1}v_x^{-1})^{y^{-1}x^{-1}}$ and $w_2 = xyv_xz$. In this manner it follows straightforwardly that any $w \in X'^*$ with trivial $G$ image is freely equivalent to a product of at most $(K+1)|w|$ conjugates of relators in $R \cup R'$ proving that $f_{P'}$ is bounded by a linear function so that $G$ is hyperbolic. This completes the first half of the proof of theorem 3.2.1.

If $G$ is word hyperbolic, show that Dehn's algorithm defines a growing and terminating grammar for $\sigma^{-1}(1)$: Suppose the generating set for $G$ is given by $X = \{x_1, x_2, \ldots, x_n\}$. First assume that $X$ is symmetric. As usual, $X$ will be the terminal alphabet. For variables we take Sans Serif copies of the letters in $X$ together with the start variable $\mathsf{s}$ and an additional variable $\mathsf{t}$:

$$V = \{\mathsf{s}, \mathsf{t}\} \cup \mathsf{X} \text{ where } \mathsf{X} = \{\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_n\} \ .$$

The first productions are

$$\mathsf{s} \rightrightarrows \mathsf{t} \tag{3.2.4}$$

and

$$\mathsf{t} \rightrightarrows \mathsf{tt} \ . \tag{3.2.5}$$

Choosing some trivial representative $r =_G 1$ with $|r| \geq 1$ add a production

$$\mathsf{s} \rightrightarrows r \tag{3.2.6}$$

in order to terminate $\mathsf{s}$. For each variable $\mathsf{x}_i$ add productions

$$\mathsf{x}_i \rightrightarrows \mathsf{x}_i\mathsf{t} \quad , \quad \mathsf{x}_i \rightrightarrows \mathsf{t}\mathsf{x}_i \quad , \quad \mathsf{x}_i \rightrightarrows x_i\mathsf{t} \quad \text{and} \quad \mathsf{x}_i \rightrightarrows \mathsf{t}x_i \ . \tag{3.2.7}$$

Let $R$ be a set of relators such that $\langle\ X \mid R\ \rangle$ is a Dehn presentation for $G$. Without loss of generality $R$ is closed under inverses and cyclic permutations. Consider the folding homomorphisms

$$\psi : (\mathsf{X} \cup X)^* \twoheadrightarrow X^* \text{ s.t. } \psi(x_i) = \psi(\mathsf{x}_i) = x_i \text{ for all } i$$

and

$$\phi : (\mathsf{X} \cup X)^* \twoheadrightarrow \mathsf{X}^* \text{ s.t. } \phi(x_i) = \phi(\mathsf{x}_i) = \mathsf{x}_i \text{ for all } i \ .$$

For each $r \in R$ with $|r| > 1$ add $2^{|r|}$ productions of the form

$$\mathsf{t} \rightrightarrows \psi^{-1}(r) \ . \tag{3.2.8}$$

Furthermore, for each non-empty prefix $p$ of $r = ps$ such that $|p| < |s|$ add $2^{|s|}$ productions

$$\phi(p^{-1}) \rightrightarrows \psi^{-1}(s) \ . \tag{3.2.9}$$

Finally, for each relator $r \in R$ of length 1 and each $\mathsf{x}_i \in \mathsf{X}$ add productions

$$\mathsf{s} \rightrightarrows \{r, \phi(r)\} \quad \text{and} \quad \mathsf{x}_i \rightrightarrows \{\mathsf{x}_i r, r\mathsf{x}_i, \mathsf{x}_i\phi(r), \phi(r)\mathsf{x}_i, x_i r, r x_i, x_i\phi(r), \phi(r)x_i\} \ . \tag{3.2.10}$$

We claim that every non-empty word in $\sigma^{-1}(1)$ is derivable from $\mathsf{s}$ using the above productions. In fact, something stronger is the case: Let $\rho : (\{\mathsf{t}\} \cup \mathsf{X} \cup X)^* \twoheadrightarrow X^*$ be a homomorphism which erases $\mathsf{t}$ (i.e. $\rho(\mathsf{t}) = \varepsilon$) and which restricts to $\psi$ on $(\mathsf{X} \cup X)^*$. We claim that if $w \in \rho^{-1}(\sigma^{-1}(1))$ then $w$ is derivable from $\mathsf{s}$. Otherwise, let $w \neq \mathsf{t}$ be a shortest non-empty counterexample. First of all, $\mathsf{t}$ cannot appear in $w$ as then a shorter counterexample could be gotten by using the reverse of a production appearing in equations (3.2.5) or (3.2.7). Similarly, equations (3.2.10) show that if $r$ is a relator of length 1, then neither $r$ nor $\phi(r)$ may appear in $w$. Thus $w$ must be an element of $(\mathsf{X} \cup X)^*$ of length at least 2 which represents 1 in $G$, and in which 1-letter relators do not appear. Since $P$ is a Dehn presentation for $G$, it must be the case that $\psi(w)$ contains a subword $s$ which is more than half of some relator, i.e. there is a word $p \in X^*$ such that $ps \in R$ and $|p| < |s|$ (here we use the fact that $R$ is invariant under cyclic permutations); furthermore, there is a word $s' \in \psi^{-1}(s)$ (so of the same length as $s$) which is a subword of $w$. If $p = \varepsilon$ we can reverse a production of the form $\mathsf{t} \rightrightarrows s'$ as in equation (3.2.8) to obtain a shorter word (since in this case $|s'| \geq 2$). So we may assume that $|p| \geq 1$. In this case we use the production $\phi(p^{-1}) \rightrightarrows s'$ arising from equation (3.2.9). Therefore, by replacing $s'$ by $\phi(p^{-1})$ in $w$ a shorter counterexample has been produced, which contradicts the assumption that $\mathsf{s}$ does not derive some element of $\rho^{-1}(\sigma^{-1}(1))$. Restricting to terminal words, conclude that the grammar defined by equations (3.2.4) through (3.2.10) generates all words in $\sigma^{-1}(1)$.

We must still show that the grammar above generates a language contained in $\sigma^{-1}(1)$. Let $\tau : (V \cup X)^* \twoheadrightarrow X^*$ be defined by extending $\rho$ via $\tau(\mathsf{s}) = \varepsilon$. Notice that if $w$ is derivable from $\mathsf{s}$ then $\sigma(\tau(w)) = 1$. This follows from the fact that $\sigma$ and $\rho$ are homomorphisms and that for any production $\alpha \rightrightarrows \beta$ we have $\sigma(\rho(\alpha)) = \sigma(\rho(\beta))$. Finally, notice that equations (3.2.6), (3.2.8), (3.2.9) and (3.2.10) include productions which terminate all variables in the grammar. This concludes the proof that a hyperbolic group has a growing and terminating word problem with respect to a symmetric set of generators.

What if $X$ is not symmetric? Consider the disjoint union $X' = X \cup X^{-1}$ and let $\sigma' : X'^* \twoheadrightarrow G$ be the generating homomorphism. The above shows that ${\sigma'}^{-1}(1)$ is growing and terminating, as hyperbolicity is independent of generating set. Letting $\sigma$ be the generating homomorphism relative the generators $X$, we have that $\sigma^{-1}(1) = {\sigma'}^{-1}(1) \cap X^*$. In general, growing and terminating languages are *not* closed under intersection with regular languages. However, we show that intersecting with $X^*$ preserves the growing and terminating property in our case. Consider the growing and terminating grammar $\mathcal{L}$ for ${\sigma'}^{-1}(1)$. Let $V$ be the set of variables. Set $U = V \cup X^{-1}$ as a new set of variables and $X$ as a new set of terminals for a grammar $\mathcal{M}$ with the same productions as $\mathcal{L}$. Then $\mathcal{M}$ is a growing grammar for ${\sigma'}^{-1} \cap X^*$. Furthermore, since $X$ is a set of monoid generators for $G$, one can find $v_x \in X^*$ for each $x \in X$ with $x^{-1} =_G v_x$. One can also choose the length $|v_x| > 1$. Adding the productions $x^{-1} \rightrightarrows v_x$ to $\mathcal{M}$ does not change the language generated by $\mathcal{M}$ and turns it into a growing and terminating grammar. ♦

## 3.3 Quasiconvex Subsets

Given a group $G$, $\mathrm{Rat}(G)$ depends only on $G$. On the other hand, the definitions of quasiconvex subsets –$\mathrm{QC}(G)$– and geodesically regular subsets –$\mathrm{Reg}(G)$– depend on the set of generators chosen for $G$. However, if $G$ is word hyperbolic, then $\mathrm{QC}(G)$ and $\mathrm{Reg}(G)$ are independent of generating set (lemmas 3.3.4 and 3.4.4).

### 3.3.1 Quasiconvexity in hyperbolic groups

Quasiconvex subsets of a hyperbolic group are subsets which are quasiconvex inside the Cayley graph of the group. A quasiconvex subset of a geodesic metric space is a set whose geodesic chords never stray too far from the set:

**Definition 3.3.1** *Let $\Gamma$ be a geodesic space. Let $S \subseteq \Gamma$. $S$ is called* **quasiconvex** *if there is a constant $K$ such that if $\gamma$ is any geodesic in $\Gamma$ which starts and ends in $S$, then $\gamma$ is contained in the $K$-neighborhood of $S$. If $G$ is a hyperbolic group, a subset $S \subseteq G$ is quasiconvex if it is quasiconvex in the Cayley graph $\Gamma_X(G)$. The collection of all quasiconvex subsets of $G$ is denoted by* $\mathrm{QC}(G)$.

**3.3.2 Remark** *When $S$ is a quasiconvex subset, with $K$ as in the above definition, $S$ is termed $K$-quasiconvex and the number $K$ is called a quasiconvexity constant for $S$.*

**3.3.3 Example** *Cyclic subgroups of hyperbolic groups are quasiconvex [3, p. 39].*

Apparently, quasiconvexity depends on $\Gamma_X(G)$ and therefore on $X$. Furthermore, there is no obvious reason why hyperbolicity was required. For hyperbolic groups, however, we have:

**Lemma 3.3.4** *Let $G$ be a hyperbolic group. Let $X$ and $X'$ be finite symmetric generating sets for $G$. Suppose that $S \subseteq G$ is quasiconvex inside of $\Gamma_X(G)$. Then $S$ is also quasiconvex inside of $\Gamma_{X'}(G)$. Therefore, quasiconvexity is independent of finite generating set.*

*Sketch of proof.* In this proof we assume that the reader is familiar with standard techniques in hyperbolic groups involving quasi-isometries. For each letter $x$ find a word $w_x \in X'^*$ such that $x =_G w_x$. This defines a mapping $\rho : ||\Gamma_X(G)|| \to ||\Gamma_{X'}(G)||$. By sending each topological edge corresponding to a directed edge $g \xrightarrow{x} gx$ to the unique path in $\Gamma_{X'}(G)$ which starts at $g$ and is labelled by $w_x$. This map $\rho$ is a **quasi-isometry**. I.e., there are constants $\lambda \geq 1$, $C \geq 0$ such that

$$\frac{1}{\lambda} d_X(x_1, x_2) - C \leq d_{X'}(\rho(x_1), \rho(x_2)) \leq \lambda d_X(x_1, x_2) + C \quad \text{for all } x_1, x_2 \in ||\Gamma_X(G)|| \ . \tag{3.3.1}$$

(The constants $\lambda$ and $C$ are expressible in terms of the maximal lengths of the words $w_x$ and the analogous words obtained in defining a reverse map $\rho' : \Gamma_{X'}(G) \to \Gamma_X(G)$.) Under $\rho$, geodesics are sent to quasigeodesics[6] whose quasigeodesic constants depend only on $\rho$. Consider a geodesic chord $\gamma$ of $S$ inside of $\Gamma_X(G)$. Because $\rho$ is a quasi-isometry and $\gamma$ is in a $K$-neighborhood of $S$ (by the quasiconvexity of $S$ in $\Gamma_X(G)$), it follows that $\rho(\gamma)$ is in a $K'$-neighborhood of $\rho(S) = S$ for $K'$ depending only on $K$ and $\rho$. Furthermore, $\rho(\gamma)$ is a quasigeodesic chord of $S$, so there is a geodesic $[\rho(\gamma)]$ with the same endpoints as $\rho(\gamma)$ and within the $L$-neighborhood of $\rho(\gamma)$, where $L$ depends only on $\Gamma_{X'}(G)$ and the quasigeodesic constants of $\rho(\gamma)$ (cf. [22, p. 82]). Therefore, $[\rho(\gamma)]$ is within a $(K' + L)$-neighborhood of $S$ with $K' + L$ a universal constant, proving that $S$ is quasiconvex in $\Gamma_{X'}(G)$. ⋄

**3.3.5 Remark** *If $G$ is not hyperbolic, there is no guaranteeing that a subset quasiconvex relative one set of generators will be remain so relative other generators. For example, in $\mathbb{Z} \times \mathbb{Z}$ the cyclic subgroup $\langle(1,1)\rangle$ is quasiconvex with respect to the group generators $\{(1,0),(1,1)\}$ and not quasiconvex with respect to $\{(1,0),(0,1)\}$. For this reason we restrict attention to word hyperbolic groups.*

Often, the following notion of quasiconvexity is required. A subset of a hyperbolic group is quasiconvex if and only if all geodesic rays starting at the origin of the Cayley

[6] A combinatorial path $u_0 \to u_1 \to \cdots \to u_n$ is a $(\lambda, C)$-**quasigeodesic** if for all $i, j$ between 0 and $n$ we have $|i - j| \leq \lambda d(u_i, u_j) + C$

graph and terminating at the subset never stray too far from the subset:

**Lemma 3.3.6** *Let $G$ be a hyperbolic group with $X$ a finite symmetric set of generators. Let $S \subseteq G$. Then $S$ is quasiconvex if and only if there is a constant $K'$ such that if $\gamma$ is a geodesic in $\Gamma_X(G)$ starting at 1 and ending in $S$, then $\gamma$ is inside the $K'$-neighborhood of $S$.*

The proof of this lemma is a straightforward application of definitions 3.1.1 and 3.3.1. It is left as an exercise for the reader.

### 3.3.2 Quasiconvexity versus rationality

In general, the collections $\mathrm{Rat}(G)$ and $\mathrm{QC}(G)$ are incomparable:

**3.3.7 Example (Rips [46])** *Given any finitely presented group $H$, there is a hyperbolic group $G$ with a subgroup $N$ which is generated by two elements and is normal, and such that $H \approx G/N$. Taking $H$ to be a torsion free group whose word problem is unsolvable —which we denote by $H = G_{\mathsf{Word}}$— implies that $G$ is a group for which membership in the kernel $N$ is undecidable. Denote this hyperbolic group by $G = G_{\mathrm{R}}$ ("R" is for "Rips"). Summarizing, the hyperbolic group $G_{\mathrm{R}}$ has a normal subgroup $N$ of rank 2 such that $G_{\mathrm{R}}/N \approx G_{\mathsf{Word}}$. The subgroup $N$ is finitely generated so is rational by example 1.6.3. $N$ is thus rational but not quasiconvex, since quasiconvex subgroups have a solvable membership problem. The last fact will follow as a consequence of proposition 3.5.8 below.*

The previous example shows that in general $\mathrm{Rat}(G) \not\subset \mathrm{QC}(G)$. On the other hand, the following shows that usually $\mathrm{QC}(G) \not\subset \mathrm{Rat}(G)$:

**3.3.8 Example** *Consider the infinite cyclic hyperbolic group $\mathbb{Z}$. The decimal expansion of $\pi = 3.1415926535\ldots$ generates the set $S = \{3, 11, 24, 31, 45, 59, 62, 76, 85, 93, 105, \ldots\}$ where the last digit of each number in the sequence corresponds to the digit sequence of $\pi$. The set $S$ is quasiconvex. Indeed, one can take $K = 19$ as the quasiconvexity constant. The fact that $S$ is not rational follows from the complete description of the rational subsets of $\mathbb{N}$ as eventually periodic subsets (see for example [15]).*

Alternatively, the following counting argument proves that $\mathrm{QC}(G) \not\subset \mathrm{Rat}(G)$ when $G$ is infinite:

**Proposition 3.3.9** *If $G$ is a countable monoid then $\mathrm{Rat}(G)$ is countable. If $G$ is an infinite hyperbolic group then $\mathrm{QC}(G)$ is uncountable.*

*Proof.* Let $X$ be a countable set generating $G$ as a monoid. A rational subset is described by a finite expression in the symbols $X$ and the symbols $\cup$, $\{$, $\}$, $($, $)$, and $^*$. There can only be countably many such expressions so $\mathrm{Rat}(G)$ is countable.

If $G$ is infinite hyperbolic then there is some element $g \in G$ of infinite order. This is a consequence of the fact that hyperbolic groups are Markov —i.e. admit a rational cross-section— (cf. [22, p. 174]) and the fact that infinite groups with rational cross-section must possess an element of infinite order [24, p. 179]. By example 3.3.3, $\langle g \rangle$ is quasiconvex. Using this fact, it follows that given any infinite dyadic sequence $[a_i]_{i\in\mathbb{N}} \in \{0,1\}^\omega$, the set

$$S([a_i]_{i\in\mathbb{N}}) = \langle g^2 \rangle \cup \{g^{2i+a_i}\}_{i\in\mathbb{N}}$$

is quasiconvex. Distinct sequences $[a_i]_{i\in\mathbb{N}}$ give rise to distinct sets $S([a_i]_{i\in\mathbb{N}})$ showing that $QC(G)$ is uncountable. ♦

$\mathrm{Rat}(G)$ is by definition closed under finite union, product and monoid closure. Often, $\mathrm{QC}(G)$ is not closed under monoid closure. For consider example 3.3.7. The non-quasiconvex subgroup $N$ is the monoid closure of the two generators of $N$ and their inverses. On the other hand the following does hold:

**Proposition 3.3.10** *Let $G$ be a hyperbolic group. Then $\mathrm{QC}(G)$ is closed under finite union and product.*

*Proof.* Use the second definition of quasiconvexity (lemma 3.3.6). Let $S, T \in \mathrm{QC}(G)$ have quasiconvexity constants $K_S, K_T$.

First consider $S \cup T$. Any geodesic from 1 to an element $u$ of $S \cup T$ is either within $K_S$ of $S$ if $u \in S$, or within $K_T$ of $T$ if $u \in T$. Therefore, $S \cup T$ is quasiconvex with constant $K_{S\cup T} = \max(K_S, K_T)$. Now consider $ST$. Let $u = st$ with $s \in S$ and $t \in T$. Find geodesics $\alpha, \beta, \gamma$ starting at 1 and respectively ending at $s$, $t$, and $u$. Translating $\beta$ by $s$ produces a geodesic triangle with sides $\alpha, s\beta$ and $\gamma$, and vertices 1, $s$ and $u$. By assumption $\beta$ is within

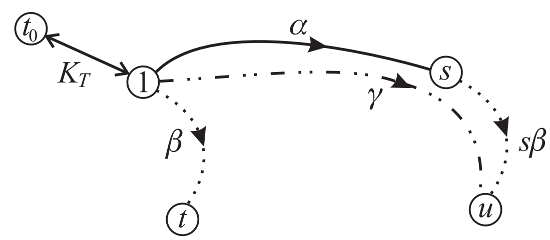



Figure 3.4: The product of quasiconvex subsets is quasiconvex.

$K_T$ of $T$ so that $s\beta$ is within $K_T$ of $sT \subseteq ST$. On the other hand, 1 is within $K_T$ of some $t_0 \in T$ which implies that $S$ is within $K_T$ of $St_0 \subseteq ST$. Therefore, as $\alpha$ is within $K_S$ of $S$, it is within $K_S + K_T$ of $ST$. Finally, $\gamma$ is within $\delta$ of $\alpha \cup s\beta$ so deduce that $\gamma$ is within $K_{ST} = K_S + K_T + \delta$ of $ST$. ♦

**3.3.11 Example** *Quasiconvex subsets are not closed under intersection as the integer subsets* $S = \{n \in \mathbb{Z} \mid n = 2k$ *or* $n = k^2$ *for some* $k \in \mathbb{N}\}$ *and* $T = \{n \in \mathbb{Z} \mid n = 2k+1$ *or* $n = k^2$ *for some* $k \in \mathbb{N}\}$ *demonstrate. On the other hand, the finite intersection of quasiconvex* subgroups *is quasiconvex [49, proposition 3].*

## 3.4 Geodesically Regular Subsets

We continue to analyze a hyperbolic group $G$ with a finite symmetric set of monoid generators $X$ and with canonical homomorphism $\sigma : X^* \twoheadrightarrow G$. We will consider certain automata labelled by $X \cup \{\varepsilon\}$. Recall that automata are denoted by calligraphic capitals such as $\mathcal{A}, \mathcal{B}, \mathcal{C} \dots$ and are identified with the regular language of $X^*$ that they accept (see definition 1.3.1). Given $w \in X^*$, $\gamma(w)$ denotes the path in the Cayley graph $\Gamma_X(G)$ which starts at 1, is labelled by $w$, and therefore ends at $\sigma(w)$. Given any path $\gamma$ in $\Gamma_X(G)$ let $[\gamma]$ denote $[\iota(\gamma), \tau(\gamma)]$ which is a choice of some geodesic between the endpoints of $\gamma$. Similarly, given any word $w \in X^*$ on the monoid generators of $G$, one can define

$$[w] = \lambda([\gamma(w)])$$

to be a shortest word such that $[w] =_G w$. This bracket notation generalizes its former use as the free reduction in $\mathrm{FG}(Y)$ (see p. 13). The language of all geodesics is denoted by $\mathcal{G}_X$.

Formally we have:

$$\mathcal{G}_X = \{\ w \in X^* \mid \gamma(w) \text{ is a shortest path between 1 and } \sigma(w) \text{ in } \Gamma_X(G)\}$$

which generalizes the notation used on p. 50 for the reduced words in the free group.

Gromov observed the following remarkable fact:

**Theorem 3.4.1** ([17, theorem 3.4.5]) *If $G$ is hyperbolic, and $X$ is a finite symmetric set of generators for $G$, then $\mathcal{G}_X$ is a regular language.*

This theorem implies that the geodesics in a hyperbolic group are completely described by their local structure. It also gives a (rather slow) solution to the word problem: Given $w \in X^*$ one can recursively enumerate $\sigma^{-1}\sigma(w)$ from a given finite presentation of $G$. Eventually one obtains a geodesic representative $u$ of $\sigma(w)$ which is detectable as a geodesic because $\mathcal{G}_X$ is regular. Finally, $w =_G 1$ if and only if $u = \varepsilon$ –the empty word. Though this algorithm is vastly inferior to Dehn's algorithm (see §3.1.3) for the hyperbolic group $G$, it hints at the algorithmic implications of the regularity of $\mathcal{G}_X$. The class of geodesically regular subsets exploits the nice properties that regular languages have in order to solve problems involving the subsets.

### 3.4.1 Reg($G$)

Given a hyperbolic group $G$, a subset $S$ is geodesically regular if there is an automaton accepting exactly those geodesic words in $G$ which represent elements of $S$. More precisely:

**Definition 3.4.2** *Let $G$ be a hyperbolic group. A subset $S \subseteq G$ is termed* **geodesically regular**, *if the inverse image of $S$ in $\mathcal{G}_X$ is a regular language. In other words, $S$ is geodesically regular iff it is $\mathcal{G}_X$-regular as in definition 1.7.1. The collection of all geodesically regular subsets of $G$ is denoted by* Reg($G$).

In the sequel "geodesically regular" will often be shortened to "regular".

**3.4.3 Remark** *In the terminology of definition 1.7.1 we have*

$$\mathrm{Reg}(G) = \mathcal{G}_X\text{-}\mathrm{Reg}(G)\ .$$

*We use "*Reg*" instead of "*$\mathcal{G}_X$-Reg*" because as lemma 3.4.4 shows, in hyperbolic groups geodesic regularity does not depend on the finite symmetric set of generators $X$.*

*Since we will deal with algorithmic questions, it usually will not suffice to verify that a set $S$ is geodesically regular. Rather, it will be necessary to construct the automaton $\mathcal{A}$ accepting the regular language $\sigma^{-1}(S) \cap \mathcal{G}_X$ explicitly.*

Regularity is independent of generating set:

**Lemma 3.4.4** *Let $G$ be a hyperbolic group with a subset $S \subseteq G$. Let $X$ and $X'$ be finite symmetric generating sets for $G$ with homomorphisms $\sigma : X^* \twoheadrightarrow G$ and $\sigma' : X'^* \twoheadrightarrow G$. If $\sigma^{-1}(S) \cap \mathcal{G}_X$ is regular then ${\sigma'}^{-1}(S) \cap \mathcal{G}_{X'}$ is regular. Therefore, geodesic regularity is independent of the finite generating set chosen for $G$.*

A proof of lemma 3.4.4 will require the balloon construction. It is included starting on p. 109.

### 3.4.2 Reg($G$), QC($G$), Rat($G$) and a theorem of Gersten and Short

As an immediate corollary to definition 3.4.2 (and also remark 1.7.2) we have:

**Lemma 3.4.5** *If $G$ is a hyperbolic group, then* $\text{Reg}(G) \subseteq \text{Rat}(G)$ *so that all geodesically regular subsets are rational.*

*Proof.* Suppose $S$ is geodesically regular. Then $\sigma^{-1}(S) \cap \mathcal{G}_X$ is a regular language. In particular, $S = \sigma(\sigma^{-1}(S) \cap \mathcal{G}_X)$ is the image of a regular language so by lemma 1.6.6, $S$ is rational. ♦

A theorem due to Gersten and Short implies that for subgroups, geodesic regularity and quasiconvexity are the same:

**Theorem 3.4.6 ([21])** *Let $G$ be a hyperbolic group and let $H$ be a subgroup of $G$. $H$ is geodesically regular if and only if $H$ is quasiconvex.*

Thus by example 3.3.3, all cyclic subgroups of hyperbolic groups are regular.

**3.4.7 Example** *A hyperbolic group $G$ is termed* **locally quasiconvex** *if all finitely generated subgroups of $G$ are quasiconvex. Examples of locally quasiconvex groups include*

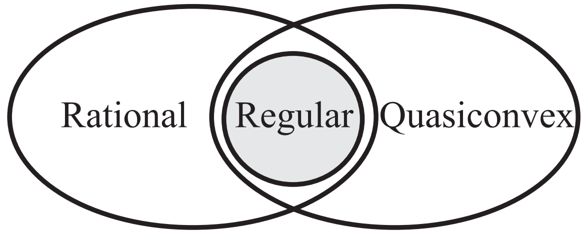



Figure 3.5: Geodesically regular subsets are both rational and quasiconvex.

*free groups, surface groups, and groups investigated by McCammond and Wise [42] and by I. Kapovich [37]. By theorem 3.4.6, all finitely generated subgroups of locally quasiconvex groups are regular. Finite generation is equivalent to rationality for subgroups (example 1.6.3); therefore, theorem 3.4.6 implies that* "QC = Reg" *for a subgroup of a hyperbolic group, while local quasiconvexity of $G$ is the property that* "Rat = Reg" *for all subgroups of $G$.*

Since "QC = Reg" for subgroups, and quasiconvex subgroups are necessarily rational it is tempting to generalize Gersten and Short's theorem, but for sub*sets* as opposed to sub*groups*. I.e., does "QC = Reg" for rational subsets?

**Question 7** *Given a hyperbolic group $G$ and a subset $S \subseteq G$. Is it the case that $S$ is geodesically regular if and only if $S$ is both quasiconvex and rational?*

At present I am unable to answer this question. However, the following shows that the "only if" portion is correct:

**Proposition 3.4.8** *If $G$ is a hyperbolic group, then* $\mathrm{Reg}(G) \subseteq \mathrm{Rat}(G) \cap \mathrm{QC}(G)$; *i.e., geodesically regular subsets are rational and quasiconvex.*

*Proof.* Let $S \in \mathrm{Reg}(G)$ and let $\mathcal{A}$ be a finite state automaton accepting the language $\sigma^{-1}(S) \cap \mathcal{G}_X$. By lemma 3.4.5, regular subsets are rational so it is enough to verify that $S$ is quasiconvex. Consider any geodesic $\gamma$ starting at 1 and ending in $S$. $\gamma$ is accepted by the automaton $\mathcal{A}$. I.e., there is a path $\tilde{\gamma}$ in $\mathcal{A}$ with the same label as $\gamma$, and which starts at an initial vertex and ends at a terminal vertex $v$. From any point on $\tilde{\gamma}$, $v$ is reachable by a path of length no larger than $|V(\mathcal{A})|$ —the number of states of the automaton. This shows that $\gamma$ is never farther than $|V(\mathcal{A})|$ from $S$ inside of $\Gamma_X(G)$ so that $S$ is quasiconvex. ♦

Were the "if" direction of question 7 correct, it would follow immediately that $\mathrm{Reg}(G)$ is closed under product: for given two regular subsets $S$ and $T$, their quasiconvexity and rationality (by the "only if" direction) and closure of $\mathrm{QC}(G)$ and $\mathrm{Rat}(G)$ under product (proposition 3.3.10) would imply that $ST$ is quasiconvex and rational. Thus the "if" direction would yield that $ST$ is regular. Nevertheless, it is possible to prove that regular subsets are closed under product even without the "if" part. In fact:

**Theorem 3.4.9** *Let $G$ be a hyperbolic group. Then* $\mathrm{Reg}(G)$ *is closed under boolean operations and product. In other words, supposing that $S$ and $T$ are geodesically regular subsets of $G$ then all of $S \cup T$, $S \cap T$, $G - S$ and $ST$ are geodesically regular.*

*Proof.* Let $X$ be a symmetric finite set of generators for $G$ and $\sigma : X^* \twoheadrightarrow G$. By assumption $\sigma^{-1}(S) \cap \mathcal{G}_X$ and $\sigma^{-1}(T) \cap \mathcal{G}_X$ are regular languages. Therefore,

- $\sigma^{-1}(S \cup T) \cap \mathcal{G}_X = (\sigma^{-1}(S) \cap \mathcal{G}_X) \cup (\sigma^{-1}(T) \cap \mathcal{G}_X)$,
- $\sigma^{-1}(S \cap T) \cap \mathcal{G}_X = (\sigma^{-1}(S) \cap \mathcal{G}_X) \cap (\sigma^{-1}(T) \cap \mathcal{G}_X)$,
- and $\sigma^{-1}(G - S) \cap \mathcal{G}_X = \mathcal{G}_X - (\sigma^{-1}(S) \cap \mathcal{G}_X)$

are regular languages, by closure of the regular languages under boolean operations and the fact that the language of geodesics $\mathcal{G}_X$ is regular (theorem 3.4.1).

Closure of $\mathrm{Reg}(G)$ under product depends on the balloon construction and will be proved in section 3.6 (see corollary 3.6.7). ♦

The above theorem is already strong enough to yield angle computation when the angle is known to be positive.

## 3.5 Angles in Hyperbolic Groups

In this section we explore the problem Angle of computing angles between finitely generated subgroups of hyperbolic groups. We show that under certain restrictions on the subgroups, Angle is decidable, but that in its full generality the problem is unsolvable, even when $G$ is hyperbolic.

### 3.5.1 Angles between quasiconvex subgroups

Suppose that $A$ and $B$ are quasiconvex subgroups of the hyperbolic group $G$. By theorem 3.4.6 $A$ and $B$ are geodesically regular. As $\mathrm{Reg}(G)$ is closed under boolean operations and products (theorem 3.4.9) and $\mathrm{Rat}(G)$ is closed under monoid closure, the following holds:

**Proposition 3.5.1** *Let $G$ be a hyperbolic group with quasiconvex subgroups $A$ and $B$. Then $[(A-B)(B-A)]^n$ is geodesically regular for all $n \in \mathbb{N}$. In addition, $\bigcup_{i=1}^{\infty}[(A-B)(B-A)]^i$ is rational.*

*Proof.* $A - B = \neg((\neg A) \cup B)$ and the closure of $\mathrm{Reg}(G)$ under unions, complements and products shows that $[(A-B)(B-A)]^n$ is regular. The last part follows from the formula

$$\bigcup_{i=1}^{\infty}[(A-B)(B-A)]^i = [(A-B)(B-A)]^+$$

and the fact that regular subsets are rational. ♦

**Corollary 3.5.2** *Let $G$ be a hyperbolic group with quasiconvex subgroups $A$ and $B$. If* $\mathsf{RatMemb}(G)$ *is decidable, then $\sphericalangle_B^A$ is computable.*

*Proof.* By proposition 3.5.1, the subsets $\bigcup_{i=1}^{\infty}[(A-B)(B-A)]^i$ and $[(A-B)(B-A)]^n$ are rational. We ask if $1 \in \bigcup_{i=1}^{\infty}[(A-B)(B-A)]^i$. This is decidable by assumption. If the answer is 'no' then by lemma 1.9.5 $\sphericalangle_B^A = 0$. Otherwise, $\sphericalangle_B^A$ is just $\pi$ divided by the first positive $n$ s.t. $1 \in [(A-B)(B-A)]^n$. By assumption, we can find this $n$ by asking if $1 \in [(A-B)(B-A)]^i$ for $i = 1, 2, 3, 4, \ldots$. ♦

There is a class of groups for which corollary 3.5.2 immediately yields an algorithm for computing angles:

**Definition 3.5.3** *Let $G$ be a hyperbolic group. $G$ is termed* **super locally quasiconvex** *if* $\mathrm{Reg}(G) = \mathrm{Rat}(G)$.

**3.5.4 Remark** *Since* $\mathrm{Reg}(G) \subseteq \mathrm{Rat}(G)$ *(proposition 3.4.5), a hyperbolic group is super locally quasiconvex if all rational subsets are regular. Since all finitely generated subgroups are rational (example 1.6.3) super local quasiconvexity is a property stronger than local* *quasiconvexity (example 3.4.7). Gilman's results [23] can be used to prove that all monadic groups*[7] *are super locally quasiconvex. Monadic groups are necessarily virtually free, and below (section 3.7) it is demonstrated that finitely generated virtually free groups are also super locally quasiconvex, and that there is an algorithm for computing an automaton accepting the geodesic pre-image of any rational set. At the moment I know of no other examples of super locally quasiconvex groups (see the question on page 113).*

The previous remark leads to:

**Theorem 3.5.5 (Gilman, private communication)** *If $G$ is virtually free then* $\mathsf{RatMemb}(G)$ *is decidable. Therefore,* $\mathsf{Angle}(G)$ *is computable.*

*Proof.* By the local quasiconvexity of free groups and corollary 3.5.2, to show that angles are computable it suffices to show that the membership problem for rational subsets of $G$ is decidable. We give three different proofs of this fact.

1. The first proof is essentially Gilman's communication. By equation 2.1.1, $\mathsf{RatUnity}(G)$ is equivalent to $\mathsf{RatMemb}(G)$. Let $\sigma : X^* \twoheadrightarrow G$ be the canonical homomorphism. Given an automaton $\mathcal{A}$ labeled by $X$, we would like to decide if $1 \in \sigma(\mathcal{A})$. Let $K = \sigma^{-1}(1)$ be the set of all trivial words. Since $G$ is virtually free, $K$ is a context free language and is accepted by a constructible pushdown automaton [43]. Notice that $1 \in \sigma(\mathcal{A})$ if and only if $K \cap \mathcal{A}$ is nonempty. The intersection of a context free language with a regular language is context free, and the pushdown automaton is constructible. Finally, the pumping lemma for pushdown automata implies that the emptiness of $K \cap \mathcal{A}$ is decidable, proving that it is decidable if $\sigma(\mathcal{A})$ contains 1.

2. The second method quotes corollary 2.3.12. Let's recap the proof of this corollary: $\mathsf{RatUnity}$ is solvable for free groups using Gilman's algorithm 1.3.7. Theorem 2.3.3 bootstraps a solution for a group to a finite index overgroup. Therefore, the solution for free groups extends to one for virtually free groups.

3. The final method follows from remark 3.5.4. Given any rational language $\mathcal{A}$, applying the results of section 3.7 will give an algorithm for constructing an automaton for $\sigma^{-1}\sigma(\mathcal{A}) \cap \mathcal{G}_X$. Given any word $w$, find a geodesic representative $[w]$ for $w$ (for example this can be done

[7] A group is **monadic** if it admits a confluent rewriting system where the right hand sides of reductions are words of length less than or equal to 1. For more details see [23].

by finding a successful path in the constructible automaton for $\sigma^{-1}\sigma(\{w\}) \cap \mathcal{G}_X$ ) and ask if $[w] \in \sigma^{-1}\sigma(\mathcal{A}) \cap \mathcal{G}_X$. The answer is 'yes' if and only if $\sigma(w) \in \sigma(\mathcal{A})$. ♦

### 3.5.2 Undecidable angles

Let's construct a word hyperbolic group $G$ in which the group theoretic angle cannot be recursively computed. Consider the hyperbolic group $G_{\mathrm{R}}$ obtained by Rips' construction in example 3.3.7. $G_{\mathrm{R}}$ contains a two generator subgroup $N$ such that the following sequence is exact:

$$1 \longrightarrow N \longrightarrow G_{\mathrm{R}} \longrightarrow G_{\mathsf{Word}} \longrightarrow 1.$$

By virtue of its definition, $G_{\mathrm{R}}$ is a hyperbolic group for which membership in the finitely generated subgroup $N$ is undecidable. Furthermore, we shall see below that it is impossible to compute angles between $N$ and cyclic subgroups of $G_{\mathrm{R}}$.

As $G_{\mathrm{R}}/N \approx G_{\mathsf{Word}}$ is torsion free, it follows that for any element $g \in G_{\mathrm{R}} - N$, the cyclic subgroup $\langle g \rangle$ is infinite and that $\langle g \rangle \cap N = \{1\}$:

$$g \notin N \Longrightarrow \langle g \rangle \cap N = \{1\}. \tag{3.5.1}$$

Consider the generalized group theoretic angle between $N$ and $\langle g \rangle$ over the trivial subgroup as well as the standard group theoretic angle between $N$ and $\langle g \rangle$ (over the intersection).

**Claim 3.5.6**

$$\sphericalangle(N, \langle g \rangle; \{1\}) = \begin{cases} 0 & \textit{if } g = 1, \\ \pi & \textit{if } g \in N - \{1\}, \\ \frac{\pi}{2} & \textit{if } g \notin N. \end{cases} \tag{3.5.2}$$

$$\sphericalangle^{N}_{\langle g \rangle} = \begin{cases} 0 & \textit{if } g \in N, \\ \frac{\pi}{2} & \textit{if } g \notin N. \end{cases} \tag{3.5.3}$$

*In both cases, the angle is $\frac{\pi}{2}$ if and only if $g \notin N$.*

*Proof.* When $A \subseteq B$, lemma 1.9.4 implies that $\sphericalangle(B, A; A) = \sphericalangle^B_A = 0$. In particular $\sphericalangle(N, \{1\}; \{1\}) = 0$, and $\sphericalangle^N_{\langle g \rangle} = \sphericalangle(N, \langle g \rangle; \langle g \rangle) = 0$ when $g \in N$. This establishes the first case of equations (3.5.2) and (3.5.3).

Suppose that $g \in N - \{1\}$. Then $\langle g \rangle \cap N = \langle g \rangle$ contains $\{1\}$ as a proper subgroup so that lemma 1.9.3 is applicable. Therefore, $\sphericalangle(N, \langle g \rangle; \{1\}) = \pi$ which is the second case of (3.5.2).

Finally, suppose that $g \notin N$. Equation (3.5.1) implies that $\langle g \rangle \cap N = \{1\}$. Therefore, the generalized angle between $N$ and $\langle g \rangle$ over $\{1\}$ coincides with the standard angle $\sphericalangle^N_{\langle g \rangle}$. Let $n \in N - \{1\}$. As $N$ is normal, there is an element $n' \in N - \{1\}$ such that

$$w = n' g^{-1} n g = 1.$$

The word $w$ is a non-empty reduced alternating word representing the identity which is of minimal possible alternating length. Therefore, $\sphericalangle(N, \langle g \rangle; \{1\}) = \sphericalangle^N_{\langle g \rangle} = \frac{\pi}{2}$ which establishes the last case of (3.5.2) and (3.5.3). ♦

**Corollary 3.5.7** *There is no recursive algorithm for computing either the generalized angle or the standard angle between finitely generated subgroups of a word hyperbolic group. In fact, letting $G_{\mathrm{R}}$ and $N \lhd G_{\mathrm{R}}$ be as above, an algorithm for either of the following problems*

- *given $g \in G_{\mathrm{R}}$, compute $\sphericalangle(N, \langle g \rangle; \{1\})$*
- *given $g \in G_{\mathrm{R}}$, compute $\sphericalangle^N_{\langle g \rangle}$*

*would yield a solution to the word problem in $G_{\mathsf{Word}}$.*

*Proof.* Suppose there is an algorithm for computing either the generalized angle or standard angle. By the previous claim, the angle is $\frac{\pi}{2}$ exactly when $g \notin N$. Therefore, a procedure is given for deciding whether or not $g \in N$. In particular, a solution for the word problem in $G_{\mathrm{R}}/N \approx G_{\mathsf{Word}}$ is obtained. This contradicts the undecidability of $\mathsf{Word}(G_{\mathsf{Word}})$. ♦

The above shows that one cannot even compute angles between rank-1 and rank-2 subgroups. On the other hand, theorem 3.5.9 will imply that it is possible to compute angles between cyclic subgroups, if the angle is known to be positive. Remark 3.5.13 below implies that to decide if the angle is zero it suffices to compute a presentation for $A \vee B$. This leads to:

**Question 8** *Let $g, h \in G$ with $G$ hyperbolic. Is it possible to decide if $\sphericalangle^{\langle g \rangle}_{\langle h \rangle} = 0$? Can one give conditions guaranteeing that $\langle g, h \rangle$ is finitely presented and that the presentation* *is computable?*

### 3.5.3 Computing positive angles

Most of the group theoretic algorithms above have involved automata. Usually it has been easy enough to construct the necessary automata. Nevertheless, a more careful analysis is in order. For example, to make proposition 3.5.1 effective, it is necessary to take quasiconvex subgroups and construct automata for them. I.e., we need to make Gersten and Short's theorem constructive. Gersten and Short's proof is constructive in the sense that if one knows a subgroup to be $K$-quasiconvex there is an automaton which accepts the geodesic pre-image of the subgroup and whose size can be predicted. But this is not good enough for our purposes. Indeed, the description of $\mathsf{Angle}(G)$ says nothing about inputting a quasiconvexity constant along with the subgroup generators. Thus we would like to give an algorithm assuming as little about the subgroups as possible. Described below is a collection of previous results which enable us to compute angles armed only with the additional knowledge that $G$ is hyperbolic and that $A$, $B$, and $A \vee B$ are quasiconvex. So we don't assume that the quasiconvexity constants of $A$, $B$ and $A \vee B$ are known, or that the automaton for $\mathcal{G}_X$ has been given, or that we even have a hyperbolicity constant for $G$ at our disposal.

Given a finite presentation $\langle\ Y \mid R\ \rangle$ for a hyperbolic group $G$, and generators $Z = \{h_1, h_2, \dots, h_n\}$ of a quasiconvex subgroup $H$, we would like to construct automata accepting $\mathcal{G}_X$ and $\sigma^{-1}(H) \cap \mathcal{G}_X$. First, a ShortLex automatic structure ShortLex$(G)$ is computed (see chapter 6 of [17]) and a hyperbolicity constant $\delta$ is found for $G$ using Papasoglu's method (see [45]). Next, constructions of I. Kapovich [36] are applied to compute a quasiconvexity constant $K$ of $H$, and an automaton $\mathcal{H}$ accepting the pre-image of $H$ inside of ShortLex$(G)$. $\mathcal{H}$ gives a solution to the membership problem in $H$. Next, using our knowledge of $\delta$ and the ShortLex structure we can build as large a finite portion of $\Gamma_X(G)$ as we desire. With the ShortLex structure and $\Gamma_X(G)$ apply the constructive proof of [17, theorem 3.2.2] to build $\mathcal{G}_X$. Let $M$ be the 2-complex for the presentation $\langle\ Y \mid R\ \rangle$ and let $\widetilde{M}_H$ be the covering space corresponding to the subgroup $H$. Since $H$ is $K$-quasiconvex, any image in $\widetilde{M}_H$ of a geodesic word ending in $H$ must stay in the $K$-neighborhood of the origin in the 1-skeleton $\widetilde{M}_H^{(1)}$. So let $\mathcal{A}'$ be the $K$-neighborhood of the origin in $\widetilde{M}_H^{(1)}$. Make $\mathcal{A}'$ an automaton by setting the origin as its unique initial and accept state. By the above, all geodesic representatives of $H$ are accepted in $\mathcal{A}'$ so that the regular language $\mathcal{A} = \mathcal{A}' \cap \mathcal{G}_X$ is equal to $\sigma^{-1}(H) \cap \mathcal{G}_X$. Thus to construct an automaton for the geodesic pre-image of $H$, it suffices to construct $\mathcal{A}'$, and this is possible since the membership problem for $H$ is solvable with the Kapovich automaton $\mathcal{H}$.

In section 3.5.4 it will also be necessary to have a presentation of $H$ at our fingertips. By work of Gersten and Short [21, theorem 3.1] $H$ is itself a hyperbolic group with triangle thinness constant $\delta'$ depending only on $\delta$ and $K$, which have already been computed above. Then theorem 3.1.7 implies that to construct a presentation of $H$ it suffices to construct a $4\delta'$-neighborhood in $\Gamma_S(H)$ where $S = Z \cup Z^{-1}$ is the symmetric generating set for $H$. This neighborhood is constructible because the word problem is solvable for $G$ and therefore for its subgroup $H$. With our ability to construct $\Gamma_X(G)$ we inherit the ability to write any element of $H$ as a product in the original generators $\{h_1, h_2, \cdots, h_n\}$. Therefore, we may apply Tietze transformations to obtain a presentation for $H$ of the form $\langle h_1, h_2, \ldots, h_n \mid r_1, r_2, \cdots, r_m \rangle$ with the $r_i$ words in the $h_i$. Putting all of the above together we get:

**Proposition 3.5.8** *Let $\langle\ Y \mid R\ \rangle$ be a finite presentation for a hyperbolic group $G$. Let $X = Y \cup Y^{-1}$, and let $\sigma : X^* \twoheadrightarrow G$ be the canonical homomorphism. Suppose that $H$ is a quasiconvex subgroup of $G$ given by a set of generators $\{h_1, h_2, \ldots, h_n\} \subset X^*$. Then automata accepting the language of geodesics $\mathcal{G}_X$ and the geodesic pre-image $\sigma^{-1}(H) \cap \mathcal{G}_X$ are constructible. Furthermore, a presentation $\langle\ h_1, h_2, \ldots, h_n \mid r_1, r_2, \ldots, r_m\ \rangle$ for the subgroup $H$ is computable. Finally, the membership problem for $H$ is decidable.*

Proposition 3.5.8 implies that the angle between two quasiconvex subgroups is computable, if it is known to be positive:

**Theorem 3.5.9** *Let $G$ be a word hyperbolic group, let $a_1, \ldots, a_m, b_1, \ldots, b_n$ be words in $G$, and let $A$ (resp. $B$) be the subgroup of $G$ generated by $\{a_1, \ldots, a_m\}$ (resp. $\{b_1, \ldots, b_n\}$). Suppose that $A$ and $B$ are quasiconvex subgroups of $G$ and that $\sphericalangle_B^A > 0$. Then the angle between $A$ and $B$ is computable.*

*Proof.* By proposition 3.5.8 we may assume that $A$ and $B$ come equipped with automata $\mathcal{A}$ and $\mathcal{B}$ for their geodesic pre-images. Therefore, by theorem 3.4.9 geodesic automata are computable for $A \cap B$, $A - B$, $B - A$ and $[(A-B)(B-A)]^n$, and hence membership in these subsets is decidable. The second definition of angles (lemma 1.9.5) and $\sphericalangle_B^A > 0$ imply that $\sphericalangle_B^A$ is $\pi$ divided by the first positive $n$ such that $1 \in [(A-B)(B-A)]^n$. By assumption, we can find this $n$ by asking if $1 \in [(A-B)(B-A)]^i$ for $i = 1, 2, 3, 4, \ldots$. ♦

So if we can decide whether or not the angle is zero, we have a complete algorithm for computing the angle. When $A, B$ and $A \vee B$ are quasiconvex, this question is decidable, although the proof of this fact will be postponed to the next subsection. As a consequence we have the following result:

**Theorem 3.5.10** *Let $G$ be a word hyperbolic group, let $a_1, \ldots, a_m, b_1, \ldots, b_n$ be words in $G$, and let $A$ (resp. $B$) be the subgroup of $G$ generated by $\{a_1, \ldots, a_m\}$ (resp. $\{b_1, \ldots, b_n\}$). Suppose that $A$, $B$, and $A \vee B$ are quasiconvex subgroups of $G$. Then the angle between $A$ and $B$ is computable.*

**Corollary 3.5.11** *Angles are computable between any finitely generated subgroups of a locally quasiconvex hyperbolic group $G$.*

*Proof.* In a locally quasiconvex group finitely generated subgroups are quasiconvex. Therefore the hypotheses of theorem 3.5.10 are automatically satisfied. ♦

**Corollary 3.5.12** *Let $G$ be a hyperbolic group with quasiconvex infinite subgroups $A$ and $B$. Suppose that $A \cap B$ has finite index in both $A$ and $B$. Then the angle between $A$ and $B$ is computable.*

*Proof.* Apply the generalized Greenberg's theorem [38, theorem 1] which states that with $A$ and $B$ as in the hypotheses above, $A \cap B$ has finite index in $A \vee B$. Since $A$ and $B$ are quasiconvex they are regular so that their intersection is regular and therefore quasiconvex. As $A \cap B$ is quasiconvex and of finite index in $A \vee B$, it follows that $A \vee B$ is quasiconvex. This is because any geodesic chord of $A \vee B$ is close to some geodesic chord in $A \cap B$ which is itself close to $A \cap B \underset{\text{f.i.}}{<} A \vee B$. Thus the hypotheses of theorem 3.5.10 are satisfied. ♦

### 3.5.4 Deciding if the angle is zero

In this subsection we complete the proof of theorem 3.5.10 by showing that there is an algorithm for deciding whether the Gersten-Stallings angle is zero when $A, B$ and $A \vee B$ are quasiconvex.

**3.5.13 Remark** *The only piece missing from the proof of theorem 3.5.10 is an algorithm for deciding whether the angle is zero. Here, use is made of the assumption that $A \vee B$ is quasiconvex. Really, all that is needed is that a finite presentation of $A \vee B$ be computable. That this presentation is computable is a consequence of the fact that $A \vee B$ is quasiconvex so is hyperbolic, and presentations of hyperbolic groups are computable.*

**Theorem 3.5.14** *Let $G$ be a word hyperbolic group, let $a_1, \dots, a_m, b_1, \dots, b_n$ be words in $G$, and let $A$ (resp. $B$) be the subgroup of $G$ generated by $\{a_1, \dots, a_m\}$ (resp. $\{b_1, \dots, b_n\}$). Suppose that $A$, $B$, and $A \vee B$ are quasiconvex subgroups of $G$. Then it is decidable if the angle between $A$ and $B$ is zero.*

*Proof.* It is assumed that $A$, $B$, and $A \vee B$ are quasiconvex. Therefore, the intersection of the regular subsets $A \cap B$ is quasiconvex. By proposition 3.5.8 we may assume that $A$, $B$ and $A \cap B$ come equipped with respective automata $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ accepting the geodesic pre-images. Also by this proposition, we can find the relations for the presentation $\langle a_1, a_2, \dots, a_m, b_1, b_2, \dots, b_n \mid r_1, r_2, \dots, r_l \rangle$ of $A \vee B$. Furthermore, a construction of Gilman [24, corollary 1.9] implies that any group generated by a rational set $S$ is finitely generated, and that the generating set is computable from an automaton representing $S$. Therefore, we can find a set of generators $\{c_1, c_2, \dots, c_l\}$ for $A \cap B = \langle \sigma(\mathcal{C}) \rangle$. Finally, let $X = \{a_1, a_2, \dots, a_m, b_1, b_2, \dots, b_n\}$. Since the word problem for $A \vee B = \langle X \rangle$ is decidable we can construct a ball of the Cayley graph $\Gamma_X(A \vee B)$ of arbitrarily large diameter.

Deciding if $\sphericalangle_B^A = 0$ is the same as deciding if the natural homomorphism $A *_{A \cap B} B \twoheadrightarrow A \vee B$ is injective (lemma 1.9.4). This can be decided as follows: for each relation $r_i$ in the presentation of $A \vee B$, ask if the word $r_i$ is trivial in $A *_{A\cap B} B$. If for some $r_i$ the answer is 'no' then $r_i$ is a witness to the fact that $A *_{A\cap B} B \twoheadrightarrow A \vee B$ is not injective. On the other hand, suppose all $r_i$ are already trivial in $A *_{A\cap B} B$, then $A *_{A\cap B} B \twoheadrightarrow A \vee B$ is injective. For consider any presentation for $A*_{A\cap B}B$ of the form $\langle a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_n \mid q_1, q_2, \ldots\rangle$.[8] Since $r_i$ are trivial in $A*_{A\cap B}B$, they are products of conjugates of the $q_j$ and their inverses. On the other hand, since $A*_{A\cap B}B \twoheadrightarrow A\vee B$ is onto, $A\vee B$ has the presentation obtained by adding the relations $r_i$ to the presentation of $A*_{A\cap B}B$. But each $r_i$ is the product of conjugates of the $q_i$ so can be deleted in the presentation. Therefore $A\vee B$ has the same presentation as $A*_{A\cap B}B$ relative the same generators so that $A*_{A\cap B}B \twoheadrightarrow A\vee B$ is an isomorphism. Thus, deciding if the angle is zero has boiled down to deciding if a relation $r_i$ of $A\vee B$ is trivial in $A*_{A\cap B}B$. This is decidable because $A*_{A\cap B}B$ has solvable word problem as will be shown in the next paragraph.

Let $w = g_1 g_2 \cdots g_n$ represent a word in $A*_{A\cap B}B$ with the $g_i$ alternating between words in $A$ and words in $B$. Prove by induction on $n$ that it is decidable if $w$ represents $1 \in A*_{A\cap B}B$. If $n = 0$, $w$ is the trivial word. If $n = 1$, decide whether $w$ is trivial by using the given solution to the word problem in $G \supseteq A, B$. So assume $n > 1$ and that the word problem is decidable for words of alternating length smaller than $n$. First consider the situation where the $g_i$ alternate between $A - A\cap B$ and $B - A\cap B$. The reduced form theorem for amalgamated products implies then that $w$ represents a non-trivial element. Since the membership problem is solvable for $A$ and $B$ using the automata $\mathcal{A}$ and $\mathcal{B}$, it is also solvable for $A\cap B, A - A\cap B$ and $B - A\cap B$. Therefore, the fact that $w$ is non-trivial in this case is detectable. If the first case does not hold, there is some $j$ such that $g_j \in A\cap B$. Suppose without loss of generality that $g_j$ is a word in the $a_i$. Find the vertex $\sigma(g_j)$ in $\Gamma_X(A\vee B)$ by following the path starting at 1 whose label is word $g_j$. Since $\sigma(g_j) \in A\cap B$, there is also a word $g'_j$ in the letters $b_i$ which is the label of a path from 1 to $\sigma(g_j)$. By constructing a large enough portion of the Cayley graph, $g'_j$ will eventually be found. (In fact, since $A\cap B$ is quasiconvex, it is undistorted, so that $|g'_j|$ will be bounded linearly in $|g_j|$ [21, proposition 2.6] so we can predict how large a portion of the Cayley graph is necessary to find $g'_j$.) Now concatenate $g'_j$ to $g_{j-1}$ and/or $g_{j+1}$ —words in the $b_i$. This reduces to the case of alternating length $n-2$ or $n-1$ and completes the proof. ♦

[8] This presentation need not be constructed though it is possible to do so.

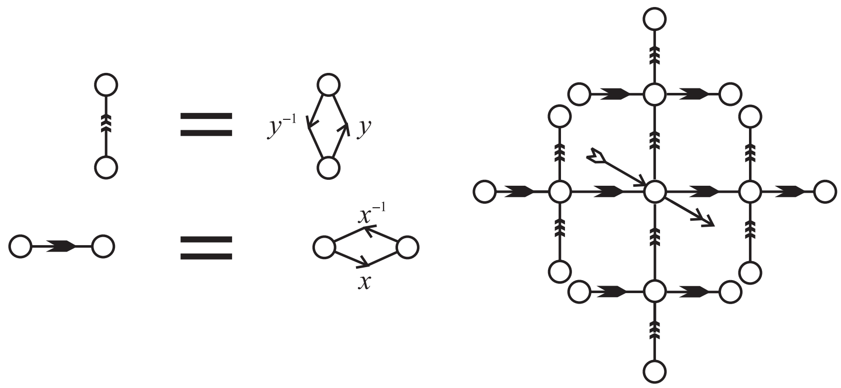



Figure 3.6: The balloon $B_2$ of the free group of rank 2.

## 3.6 The Balloon Construction

The balloon construction is a generalization of Gilman's algorithm 1.3.7. We use the balloon construction to show that in a word hyperbolic group, regular subsets are closed under product and are independent of the chosen generating set. In the next section the balloon construction will be used to show that virtually free groups are super locally quasiconvex.

### 3.6.1 The $K$-thickening of $\mathcal{A}$, $B_K(\mathcal{A})$

Gilman's algorithm 1.3.7 is based on the idea of free reduction of words. Free reduction is the process of cutting out all sub-words of the form $u^{-1}u$. Letting $v = u$ we have that $u$ and $v$ are 0-fellow travelers in the sense defined below. Therefore, Gilman's reduction algorithm gives a way of building shortcuts around all paths of the form $u^{-1}v$ where $u$ and $v$ are 0-fellow travelers. The balloon construction generalizes Gilman's algorithm in that shortcuts are built around all paths of the form $u^{-1}v$ where $u$ and $v$ are $K$-fellow travelers.

**Definition 3.6.1** *Let $(\Gamma, d)$ be a metric space and consider two discrete paths $u, v : \mathbb{N} \to \Gamma$. If for all $i \in \mathbb{N}$ we have $d(u(i), v(i)) \leq K$ then $u$ and $v$ are called $K$-***fellow travelers***.*

**3.6.2 Remark** *By convention, a word $u = u_1u_2 \cdots u_n$ with $u_i \in X$ defines a path in the Cayley graph $u : \mathbb{N} \to \Gamma_X(G)$ such that $u(0) = 1$, $u(i) = \sigma(u_1u_2 \cdots u_i)$ for $i$ between 1 and $n$, and $u(i) = \sigma(u)$ for all $i \geq n$.*

The balloon construction consists of gluing neighborhoods of the Cayley graph (balloons) to each vertex of an automaton, and then freely reducing the automaton.

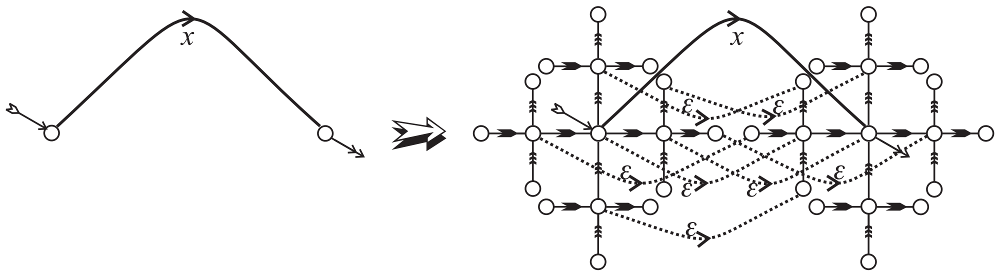



Figure 3.7: An automaton $\mathcal{A}$ over the free group of rank 2, and its 2-thickening $B_2(\mathcal{A})$.

**Algorithm 3.6.3** *Let $G$ be a group generated by $Y$ and $\mathcal{A}$ be an automaton as in definition 1.3.1 with $X = Y \cup Y^{-1}$. Let $B_K$ be an explicitly given ball of radius $K$ around the origin in the Cayley graph $\Gamma_X(G)$. View $B_K$ as a directed graph labelled by $X = Y \cup Y^{-1}$. The* $K$**-thickening** *of $\mathcal{A}$, or $B_K(\mathcal{A})$, is defined by the following* **balloon construction***:*

1. *For each vertex $v$ of $\mathcal{A}$, take a copy of $B_K$ and glue it to $\mathcal{A}$ by identifying the vertex $1$ of $B_K$ with $v$. Call the resulting automaton $\mathcal{A}'$.*

2. *Freely reduce $\mathcal{A}'$ to obtain $B_K(\mathcal{A}) = \widetilde{\mathcal{A}'}$.*

3. *The initial and accept states of $B_K(\mathcal{A})$ are inherited from $\mathcal{A}$.*

*$B_K(\mathcal{A})$ has the following properties:*

- *$\sigma(\mathcal{A}) = \sigma(B_K(\mathcal{A}))$.*

- *Suppose that the words $u, v \in X^*$ represent $K$-fellow travelers in the Cayley graph. Furthermore, suppose that the word $u^{-1}v$ appears as a label of some path between the vertices $a$ and $b$ of $\mathcal{A}$. Then, a geodesic representative $[u^{-1}v]$ appears as a label of some path between $a$ and $b$ in $B_{2K}(\mathcal{A})$.*

- *If $u \in \mathcal{A}$ is in the $K$-neighborhood of $[u]$, then $[u] \in B_{3K}(\mathcal{A})$.*

- *Suppose that $G$ is a $\delta$-hyperbolic group and $u \in \mathcal{A}$ is a $(\lambda, C)$-quasigeodesic segment.*[9] *Then there is a $K$ depending only on $\delta$, $\lambda$ and $C$ such that all geodesic representatives $[u]$ of $u$ are accepted by $B_K(\mathcal{A})$.*

**3.6.4 Remark** *Notice that $B_0(\mathcal{A})$ is just the free reduction of $\mathcal{A}$. $B_K(\mathcal{A})$ has $|V(B_K)| \cdot |V(\mathcal{A})|$ vertices. Therefore, for groups with exponential growth this construction is exponential in $K$. Finally, if $K$ is not a whole number, by convention $B_K(\mathcal{A})$ denotes $B_{\lceil K \rceil}(\mathcal{A})$.*

*Also notice that only the last property of algorithm 3.6.3 makes use of hyperbolicity.*

*Proof of 3.6.3.* $\sigma(\mathcal{A}) = \sigma(B_K(\mathcal{A}))$: Since $B_K(\mathcal{A})$ is the reduction of $\mathcal{A}'$ and by algorithm 1.3.7 free reduction does not change the image of the accepted language, it is enough to show that $\sigma(\mathcal{A}') = \sigma(\mathcal{A})$. Certainly $\sigma(\mathcal{A}') \supseteq \sigma(\mathcal{A})$ since $\mathcal{A}$ is a sub-automaton of $\mathcal{A}'$. On the other hand, consider a path $\gamma$ in $\mathcal{A}'$ from an initial state to an accept state. Now $\gamma$ is the product of paths in $\mathcal{A}$ interspersed with excursions into various balloons. There is only one way to get in or out of any particular balloon and the label of an excursion from its entrance to its exit inside a balloon represents a closed path in $\Gamma_X(G)$ —i.e. has trivial image in $G$. Therefore, the excursions all have trivial $G$ images so that the image of any path in $\mathcal{A}'$ is the same as the image of the path in $\mathcal{A}$ obtained by removing all the balloon excursions.

$K$-fellow travelers: Suppose that the words $u, v \in X^*$ are $K$-fellow travelers in $\Gamma_X(G)$. If $K = 0$ then $u$ and $v$ are identical paths in $\Gamma_X(G)$ and it is enough to show that there is a path labelled by 1 from $a$ to $b$ in $B_{2K}(\mathcal{A}) = B_0(\mathcal{A}) = \widetilde{A}$. This is indeed the case, by property 4 of algorithm 1.3.7. Thus we may assume that $K \geq 1$ and write $u$ and $v$ as products of words

$$u = u_1 u_2 \cdots u_n,\ v = v_1 v_2 \cdots v_n,\ |u_i| \leq K \text{ and } |v_i| \leq K \tag{3.6.1}$$

and such that there are geodesic words $r_1, \ldots, r_n$ of length bounded by $K$ for which the following equations hold in $G$

$$u_1 u_2 \cdots u_i \cdot r_i = v_1 v_2 \cdots v_i \quad \text{for} \quad i = 1, 2, \ldots, n.$$

[9]See footnote 6 on p. 86 for the definition of quasigeodesics.

Let $r_0 = 1$. Then for $i = 0, 1, \dots, n-1$ we have in $G$:

$$\begin{aligned} u_{i+1}^{-1} \cdot r_i \cdot v_{i+1} \cdot r_{i+1}^{-1} &= u_{i+1}^{-1} u_i^{-1} \cdots u_1^{-1} \cdot u_1 \cdots u_i r_i \cdot v_{i+1} \cdot r_{i+1}^{-1} \\ &= u_{i+1}^{-1} u_i^{-1} \cdots u_1^{-1} \cdot v_1 \cdots v_i v_{i+1} r_{i+1}^{-1} = u_{i+1}^{-1} u_i^{-1} \cdots u_1^{-1} \cdot u_1 \cdots u_i u_{i+1} = 1. \end{aligned} \quad (3.6.2)$$

So by conjugating and taking the inverse we see that $v_{i+1}^{-1} r_i^{-1} u_{i+1} r_{i+1}$ represents a closed loop of length no greater than $4K$ and thus inside of $B_{2K}$. By assumption, there is a path in $\mathcal{A}$ starting at $a$ and ending at $b$ with label $u^{-1}v$. Adjoining balloons of radius $2K$ implies that the following path is in $\mathcal{A}'$:

$$u_n^{-1} u_{n-1}^{-1} \cdots u_1^{-1} \cdot v_1 \cdot v_1^{-1} u_1 r_1 \cdot v_2 \cdots \cdot v_{n-1} \cdot v_{n-1}^{-1} r_{n-2}^{-1} u_{n-1} r_{n-1} \cdot v_n \cdot v_n^{-1} r_{n-1}^{-1} u_n r_n.$$

This path freely reduces to $r_n$ so that a path labelled by $r_n$ (after extraction of $\varepsilon$'s) exists in the free reduction $B_{2K}(\mathcal{A}) = \widetilde{\mathcal{A}'}$. Therefore the geodesic representative $r_n = [u^{-1}v]$ is the label of a word starting at $a$ and ending at $b$ which is what we wanted to prove.

$K$-neighborhood geodesics are captured: Show that if $u \in \mathcal{A}$ is in the $K$-neighborhood of a geodesic representative $[u]$, then $[u] \in B_{3K}(\mathcal{A})$. The method of proof is similar to that of the previous paragraph. First, if $K = 0$ then $[u]$ is the free reduction of $u$ and therefore $[u] \in \widetilde{\mathcal{A}} = B_0(\mathcal{A}) = B_{3K}(\mathcal{A})$ by the first property of algorithm 1.3.7. Thus we may assume that $K \geq 1$ and write $u = u_1 u_2 \cdots u_n$ as a product of words $u_i$ of length bounded by $K$. Let $v = [u]$. By assumption, there are geodesic words $r_i$ of length bounded by $K$ which connect the vertex $\sigma(u_1 u_2 \cdots u_i)$ to a vertex on the path $v$. As $u =_G v$ take $r_n = r_0 = 1$. Define $v_i$ inductively as follows: $v_1$ is the sub-geodesic of $v$ from 1 to $\sigma(u_1 r_1)$. Then $v_{i+1}$ is the sub-geodesic of $v$ (or its inverse) from $\sigma(u_1 u_2 \cdots u_i r_i)$ to $\sigma(u_1 u_2 \cdots u_i u_{i+1} r_{i+1})$. A calculation similar to that of equation 3.6.2 reveals that in $G$ we have:

$$v_i = r_{i-1}^{-1} u_i r_i. \quad (3.6.3)$$

As $v_i$ is a sub-geodesic, $|v_i| \leq |r_{i-1}^{-1} u_i r_i| \leq 3K$. Therefore, $r_{i-1} v_i r_i^{-1} u_i^{-1}$ is a trivial loop in $G$ of length bounded by $6K$ and so is contained in $B_{3K}$. By assumption, there is a path in $\mathcal{A}$ labelled by $u$, starting from an initial vertex and ending at an accept state. Adjoining balloons of radius $3K$ implies that the following path is in $\mathcal{A}'$:

$$v_1 r_1^{-1} u_1^{-1} \cdot u_1 \cdot r_1 v_2 r_2^{-1} u_2^{-1} \cdot u_2 \cdots u_{n-1} \cdot r_{n-1} v_n u_n^{-1} \cdot u_n \ . \quad (3.6.4)$$

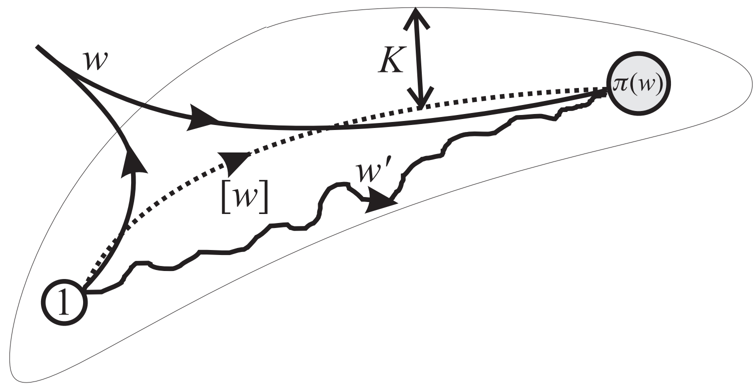



Figure 3.8: The quasigeodesic property of a language.

This path freely reduces to $v$ so that a path labelled by $v$ is accepted in the free reduction $B_{3K}(\mathcal{A}) = \widetilde{\mathcal{A}'}$.

Finally, suppose that $G$ is a $\delta$-hyperbolic group and $u \in \mathcal{A}$ is a $(\lambda, C)$-quasigeodesic segment. Since quasigeodesics are close to geodesics in a hyperbolic space [22, p. 82] and all geodesics between the same two points are $\delta$-close to each other, there is a number $L$ depending only on $\delta$, $\gamma$, and $C$ such that $u$ is in the $L$-neighborhood of any geodesic representative $[u]$. Letting $K = 3L$ and using the result of the previous paragraph implies that the geodesics $[u]$ are accepted by $B_K(\mathcal{A})$. ♦

The fourth property of the balloon algorithm motivates the following useful definition:

**Definition 3.6.5** *Let $G$ be a group with finite symmetric set of generators $X$. Let $\mathcal{L} \subseteq X^*$. The language $\mathcal{L}$ is called a $K$-***quasigeodesic language** *if for all $w \in \mathcal{L}$ and for all geodesic representatives $[w]$ there is a word $w' \in \mathcal{L}$ such that $\sigma(w) = \sigma(w')$ and such that $w'$ is within a $K$-neighborhood of $[w]$ in the Cayley graph $\Gamma_X(G)$. If $K = 0$ then $\mathcal{L}$ is called a* **geodesic language**.

**3.6.6 Exercise** *Show that $S \subseteq G$ is a geodesically regular subset, if and only if it is the image of a geodesic regular language $\mathcal{A} \subseteq X^*$. In fact, $\mathcal{A}$ is a geodesic regular language if and only if $\widetilde{\mathcal{A}} \cap \mathcal{G}_X = \sigma^{-1}\sigma(\mathcal{A}) \cap \mathcal{G}_X$. Therefore, if $w \in \mathcal{A}$ then for any geodesic representative $[w]$ we have $[w] \in \widetilde{\mathcal{A}}$.*

Algorithm 3.6.3 yields:

**Corollary 3.6.7** *Suppose that $G$ is a group with symmetric generating set $X$, and that $\mathcal{A}$ is an automaton labelled by $X$. If $\mathcal{A}$ is a $K$-quasigeodesic language, then $\sigma(\mathcal{A})$ is a geodesically regular subset of $G$. In fact, $\sigma^{-1}\sigma(\mathcal{A}) \cap \mathcal{G}_X = B_{3K}(\mathcal{A}) \cap \mathcal{G}_X$. Finally, suppose that $G$ is also $\delta$-hyperbolic and that $\mathcal{A}$ and $\mathcal{B}$ are geodesic regular languages. Then $B_{12\delta}(\mathcal{AB})$ is $9\delta$-quasigeodesic. Therefore,* $\mathrm{Reg}(G)$ *is closed under product.*

*Proof.* Suppose $\mathcal{A}$ is $K$-quasigeodesic. So given any geodesic representative $[w]$ of a word in $\mathcal{A}$, there is a representative $w' \in \mathcal{A}$ within $K$ of $[w]$. By the third property of the balloon construction 3.6.3, this means that $[w] \in B_{3K}(\mathcal{A})$. Thus $\mathcal{G}_X \cap \sigma^{-1}\sigma(\mathcal{A}) \subseteq B_{3K}(\mathcal{A})$. On the other hand, the first property of the balloon construction implies that $\sigma^{-1}\sigma(\mathcal{A}) \supseteq B_{3K}(\mathcal{A})$ so that $\sigma^{-1}\sigma(\mathcal{A}) \cap \mathcal{G}_X = B_{3K}(\mathcal{A}) \cap \mathcal{G}_X$.

Now suppose that $\mathcal{A}$ and $\mathcal{B}$ are geodesic regular languages and let $\mathcal{C} = \mathcal{AB}$. Consider $w \in B_{12\delta}(\mathcal{C})$ and a geodesic representative $[w]$. We would like to show that $B_{12\delta}(\mathcal{C})$ accepts a word $w'$ which is close to $[w]$ and such that $\sigma(w') = \sigma([w])$. Without loss of generality $w = uv$ with $u \in \mathcal{A}$ and $v \in \mathcal{B}$ because $\sigma(B_{12\delta}(\mathcal{C})) = \sigma(\mathcal{A})\sigma(\mathcal{B})$. Notice that by construction $B_{12\delta}(\mathcal{C}) = B_{12\delta}(\widetilde{\mathcal{C}})$ as automata. Since $\mathcal{A}$ and $\mathcal{B}$ are geodesic languages exercise 3.6.6 implies that $[u] \in \widetilde{\mathcal{A}}, [v] \in \widetilde{\mathcal{B}}$ and therefore $[u][v] \in \widetilde{\mathcal{A}}\widetilde{\mathcal{B}} \subseteq \widetilde{\mathcal{C}}$. So without loss of generality $u$ and $v$ are geodesics; form a geodesic triangle in $\Gamma_X(G)$ with sides labelled by $u$, $v$ and $[w]$. (It may help to think of $w$ in figure 3.8 as the product $uv$ and the corner on $w$ as the point $\sigma(u)$.) Since $G$ is $\delta$-hyperbolic, lemma 3.1.3 implies that this triangle is $6\delta$-thin. Let $c_u \in u$, $c_v \in v$ and $c_w \in [w]$ be the unique points arising from definition 3.1.2 (see figure 3.1 on p. 71). We have that

$$d(c_u, c_v),\ d(c_v, c_w),\ d(c_w, c_u) \ \leq\ 6\delta.$$

Let $\alpha = [\sigma(u), c_u]$ be the sub-geodesic of $u^{-1}$, starting at $\sigma(u)$ and ending at $c_u$. On the other hand, let $\beta = [\sigma(u), c_v]$ be the sub-geodesic of $v$ starting from $\sigma(u)$ and ending at $c_v$. By definition 3.1.2, $\alpha$ and $\beta$ are $6\delta$-fellow travelers. Letting $r = [\alpha^{-1}\beta]$ it then follows by the second property of algorithm 3.6.3 that the path $[1, c_u] \cdot r \cdot [c_v, \sigma(w)]$ is accepted by $B_{12\delta}(\widetilde{\mathcal{C}})$. By $6\delta$-thinness we have that $[1, c_u]$ is within $6\delta$ of $[1, c_w]$ and that $[c_v, \sigma(w)]$ is within $6\delta$ of $[c_w, \sigma(w)]$. Finally, as $r$ is the geodesic between $c_u$ and $c_v$, its length is bounded by $6\delta$. Therefore any point on $r$ is within $3\delta$ of $c_u$ or $c_v$, and consequently $r$ is in the $9\delta$-neighborhood of $[w]$. This proves that $B_{12\delta}(\mathcal{A})$ is a $9\delta$-quasigeodesic language so that $\sigma(B_{12\delta}(\mathcal{C})) = \sigma(\mathcal{C}) = \sigma(\mathcal{A}) \cdot \sigma(\mathcal{B})$ is a regular subset of $G$. Finally, given any regular subsets $S,\ T \in \mathrm{Reg}(G)$, by definition $\sigma^{-1}(S) \cap \mathcal{G}_X = \mathcal{A}$ and $\sigma^{-1}(T) \cap \mathcal{G}_X = \mathcal{B}$ are geodesic languages. Therefore, $ST = \sigma(\mathcal{AB})$ is a regular subset of $G$ by the preceding arguments. ♦

### 3.6.2 Proof that $\mathrm{Reg}(G)$ is independent of generators

The section is concluded with a proof of lemma 3.4.4:

*Proof.* Let $G$ be a hyperbolic group with finite symmetric generating sets $X, X'$ and natural homomorphisms $\sigma : X^* \twoheadrightarrow G$ and $\sigma' : X'^* \twoheadrightarrow G$. Suppose that $S \subseteq G$ is geodesically regular relative $X$. We would like to show that $S$ is geodesically regular relative $X'$.

For each $x \in X$ choose a word $w_x \in X'^*$ such that $x$ and $w_x$ are equal in $G$, i.e. $\sigma(x) = \sigma'(w_x)$. These choices define a monoid homomorphism $\rho : X^* \to X'^*$, with the property that $\sigma = \sigma' \circ \rho$. The homomorphism $\rho$ may also be viewed as a quasi-isometry $\Gamma_X(G) \to \Gamma_{X'}(G)$ induced by replacing each edge in the Cayley graph by a corresponding path. Furthermore, $\rho$ fixes the vertices of the Cayley graph. Now we are supposing that $S$ is regular which means that the language $\mathcal{A} = \sigma^{-1}(S) \cap \mathcal{G}_X$ is regular. As monoid homomorphisms preserve regularity, $\rho(\mathcal{A})$ is a regular language in $X'^*$. Furthermore, since $\rho$ is a quasi-isometry, the language $\rho(\mathcal{A})$ consists of quasigeodesics with constants depending only on the quasi-isometry constants of $\rho$ and hyperbolicity constants of the $\Gamma_X(G)$ and $\Gamma_{X'}(G)$. Since in a hyperbolic group geodesics are close to quasigeodesics it follows that $\rho(\mathcal{A})$ is a quasigeodesic language in the sense of definition 3.6.5. Therefore, by corollary 3.6.7 $\sigma'(\rho(\mathcal{A})) = \sigma(\mathcal{A}) = S$ is a geodesically regular subset —relative to $\mathcal{G}_{X'}$. ♦

## 3.7 Super Local Quasiconvexity

Recall definition 3.5.3 of super local quasiconvexity of a group. The following theorem shows that finitely generated virtually free groups are super locally quasiconvex.

**Theorem 3.7.1** *Suppose that $G$ is a virtually free group and that $X$ is a finite symmetric set of generators with canonical map $\sigma : X^* \twoheadrightarrow G$. Then given any automaton $\mathcal{A}$ labeled by $X$ there is a constructible geodesic automaton $\mathcal{A}'$ with the property that $\mathcal{A}' = \sigma^{-1}\sigma(\mathcal{A}) \cap \mathcal{G}_X$. In particular, $G$ is super locally quasiconvex.*

*Proof.* Starting with a presentation $P = \langle\ Y \mid R\ \rangle$ for $G$ (with $X = Y \cup Y^{-1}$) construct a presentation $P' = \langle\ Y' \mid R'\ \rangle$ relative which the theorem holds. By the constructive nature of the proof of lemma 3.4.4 it follows that the theorem holds relative the monoid generating set $X$ as well. As $G$ is virtually free, it contains a free subgroup $F$ of finite index. Supposing that $F = \mathrm{FG}(\{y_1, y_2, \ldots, y_m\})$ and that $F \underset{n}{<} G$, then $G$ (theoretically speaking) admits a presentation of the following form:

$$P' = \left\langle \begin{array}{l} y_1, y_2, \ldots, y_m, \\ z_2, z_3, \ldots, z_n \end{array} \left| \begin{array}{l} z_i^{-1} = t(i) z_{\lambda(i)}, \\ z_i y_j = u(i,j) z_{\mu(i,j)}, \\ z_i y_j^{-1} = v(i,j) z_{\nu(i,j)}, \\ z_i z_k = w(i,k) z_{\omega(i,k)} \end{array} \right. \quad \text{s.t.} \quad \begin{array}{l} 2 \leq i,k \leq n \\ 1 \leq j \leq m \end{array} \right\rangle \tag{3.7.1}$$

where $t(i), u(i,j), v(i,j), w(i,k) \in \{y_1, y_1^{-1}, \ldots y_m, y_m^{-1}\}^*$ and $\lambda(i), \mu(i,j), \nu(i,j)$ and $\omega(i,k)$ take values between 1 and $n$. Finally, we take the convention that $z_1 = \varepsilon$ is the empty word, so for example, if $\omega(2,4) = 1$ we have the relation $z_2 z_4 = w(2,4)$.

Why is there a presentation like $P'$ for $G$? The $y_j$ are free generators for $F$, and the $z_i$ are $F\backslash G$ coset representatives. Thus for any $z_i, y_j$ and $z_k$ we have in $G$ that

$$z_i^{-1}, z_i y_j, z_i y_j^{-1}, z_i z_k \in F \cdot \{1, z_2, z_3, \ldots, z_n\}\ .$$

In other words, we can find the relations claimed. On the other hand, we need to show that a choice of such relations suffices to give a presentation for $G$. Given any word $s$ on the generators $\{y_1, y_1^{-1} \ldots, y_m, y_m^{-1}, z_2, z_2^{-1}, \ldots, z_n, z_n^{-1}\}$ we can use the relations above (together with free cancellations like $y_1 y_1^{-1} = 1$, etc.) to find a reduced word $s'$ such that

$s =_G s'$ but $s'$ has the form:

$$s' \in \{y_1, y_1^{-1}, \ldots, y_m, y_m^{-1}\}^* \cdot \{\varepsilon, z_2, \ldots, z_n\} \ . \tag{3.7.2}$$

This is possible because the relations of equation (3.7.1) allow us to push all the $z_i$'s from left to right, and get rid of any $z_i$ streak. In particular, suppose that $s =_G 1$ so that $s' =_G 1$. But $s' = wz_i$ for some word $w$ in the $y_j$'s and some $z_i$ (or possibly for $z_1 = \varepsilon$). Therefore, $wz_i =_G 1$ which can only happen if $w = z_i = \varepsilon$ since $w$ is reduced. Thus we have reduced $s$ to $\varepsilon$ using only the relations in equation (3.7.1) which proves that $P'$ is a presentation for $G$.

In order to make use of the presentation $P'$ in our algorithms, we need to actually construct it. This is done by enumerating all finitely generated subgroups of $G$ and beginning a coset enumeration process on each of these. For those subgroups which are of finite index, the coset enumeration terminates, so that finite index subgroups are detectable. Furthermore, the resulting coset table can be used to construct a presentation for any finite index subgroup, using the presentation $P$. Applying sequences of Tietze transformations to all such presentations, we will eventually transform one of the presentations to one having no relations, i.e. the standard presentation of a free group $F$, since $G$ was assumed to be virtually free. We can then reverse this successful Tietze transformation to find words $\alpha_j$ in the generators of $P$ which represent a basis of a free finite index subgroup. Using the coset diagram it is also possible to find words $\beta_i$ which are right coset representatives of $F\backslash G$. Adding generators $y_j$ and $z_i$ and relations $y_j = \alpha_j$ and $z_i = \beta_i$ to $P$ defines a presentation $P''$ which is transformable to the presentation $P'$ that we are after.

Thus, we may assume that the presentation $P'$ of equation (3.7.1) is at our disposal. Let $\mathcal{V}$ consist of all the words $t(i), u(i,j), v(i,j), w(i,k)$ which appear in equation (3.7.1) and let

$$K = \max\{ \ |v| \ \big| \ v \in \mathcal{V} \ \} \ + \ 4. \tag{3.7.3}$$

Let $X = \{y_1, y_1^{-1} \ldots, y_m, y_m^{-1}, z_2, z_2^{-1}, \ldots, z_n, z_n^{-1}\}$ and consider any automaton $\mathcal{A}$ labeled by $X$. We claim that given any $s \in \mathcal{A}$ there is a word $s' \in B_{\frac{K}{2}}(\mathcal{A})$ such that $s =_G s'$, $s'$ is reduced, and $s'$ is a word such that no inverse of any letter $z_k$ appears in it. Indeed, suppose $s = s_1 s_2 \cdots s_q$. For each $s_l$ let

$$r_l = \begin{cases} t(i) z_{\mu(i)} z_i & \text{if } s_l = z_i \text{ for some } i, \\ \varepsilon & \text{otherwise.} \end{cases} \tag{3.7.4}$$

The $r_l$'s are relators in $G$ and $|r_l| \leq K - 2$. Therefore, the word

$$s'' = r_1 s_1 r_2 s_2 \cdots r_q s_q$$

is accepted by $B_{\frac{K}{2}}(\mathcal{A})$. The word $s''$ freely reduces to a word $s'$ because each occurrence of $z_i^{-1}$ is replaced by $t(i) z_{\mu(i)}$ and freely reducing this word does not produce new inverses of $z_i$'s.

Next we show that given any $s \in \mathcal{A}$, there is a word $s' \in B_{\frac{K}{2}}(B_{\frac{K}{2}}(\mathcal{A}))$ such that $s' =_G s$, $s'$ is reduced and $s'$ conforms to the form of equation (3.7.2). By the previous paragraph, we may assume that $s$ is a word with only positive appearances of the $z_i$, (though we must allow it to occur in $B_{\frac{K}{2}}(\mathcal{A})$ instead of $\mathcal{A}$). Supposing again that $s = s_1 s_2 \cdots s_q$, we inductively define the words $m_l$ and $r_l$ by setting $m_1 = s_1$, $r_1 = \varepsilon$ and

$$r_{l+1} \quad = \quad \begin{cases} u(i,j) z_{\mu(i,j)} y_j^{-1} z_i^{-1} & \text{if } m_l = z_i \text{ and } s_{l+1} = y_j \ , \\ v(i,j) z_{\nu(i,j)} y_j z_i^{-1} & \text{if } m_l = z_i \text{ and } s_{l+1} = y_j^{-1} \ , \\ w(i,k) z_{\omega(i,k)} z_k^{-1} z_i^{-1} & \text{if } m_l = z_i \text{ and } s_{l+1} = z_k \ , \\ \varepsilon & \text{otherwise .} \end{cases} \tag{3.7.5}$$

$$m_{l+1} \quad = \quad \begin{cases} z_{\mu(i,j)} & \text{if } m_l = z_i \text{ and } s_{l+1} = y_j \ , \\ z_{\nu(i,j)} & \text{if } m_l = z_i \text{ and } s_{l+1} = y_j^{-1} \ , \\ z_{\omega(i,k)} & \text{if } m_l = z_i \text{ and } s_{l+1} = z_k \ , \\ s_{l+1} & \text{otherwise .} \end{cases} \tag{3.7.6}$$

Again, the $r_l$ are relators implying that $r_{l+1}^{m_l} =_G 1$. But as $|m_l^{-1} r_{l+1} m_l| \leq K$ it follows that the word

$$w = r_2 s_1 s_2 r_3^{m_2} s_3 r_4^{m_3} \cdots r_q^{m_{q-1}} s_q \tag{3.7.7}$$

is accepted by $B_{\frac{K}{2}}(B_{\frac{K}{2}}(\mathcal{A}))$. The word $w$ freely reduces to a word $s'$ in the form of equation (3.7.2). This is because the $r_l$ and $m_l$ have been defined in a way that implies that in the free group on our generators we have that $r_{l+1}m_l s_{l+1}m_{l+1}^{-1}$ is purely in terms of the $y_j^{\pm 1}$'s. But equation (3.7.7) is the same as

$$w = r_2 m_1 s_2 m_2^{-1} \cdot r_3 m_2 s_3 m_4^{-1} \cdots r_{q-1} m_{q-2} s_{q-1} m_{q-1}^{-1} \cdot r_q m_{q-1} s_q m_q^{-1} \cdot m_q$$

in the free group; thus everything except possibly the last letter $m_q$ freely reduces to a word in the $y_j^{\pm 1}$'s showing that $w$ freely reduces to the claimed $s'$ as in equation (3.7.2). Finally, we may assume that $s'$ is reduced as the balloon construction results in a freely reduced automaton.

Notice that $s'$ is a quasigeodesic in $\Gamma_X(G)$. For $s'$ is the product of a reduced word in $F$ with a single letter like $z_i$. The reduced word portion of $s'$ is a geodesic in the Cayley graph of $F$. As $F$ has finite index in $G$ it embeds quasi-isometrically and therefore sends geodesics to quasigeodesics. Thus the reduced word portion of $s'$ is a quasigeodesic in $G$. Multiplying any quasigeodesic by a single letter, does not change the fact that it is a quasigeodesic, though it does perturb the quasigeodesic constant slightly. Finally, the quasigeodesic constant is computable from the quasi-isometry constants of $F \hookrightarrow G$ which are themselves computable from the presentation $P'$.

The above implies that $B_{\frac{K}{2}}(B_{\frac{K}{2}}(\mathcal{A}))$ is an $M$-quasigeodesic language in the sense of definition 3.6.5, and with $M$ some computable number. Thus we may apply corollary 3.6.7 to conclude that

$$\sigma^{-1}\sigma(\mathcal{A}) \cap \mathcal{G}_X = B_{3M}(B_{\frac{K}{2}}(B_{\frac{K}{2}}(\mathcal{A}))) \cap \mathcal{G}_X$$

which establishes constructively that virtually free groups are super locally quasiconvex. ♦

**Question 9** *What hyperbolic groups are super locally quasiconvex? Is the fundamental group of the 2-holed torus super locally quasiconvex?*

# Bibliography

[1] A. V. Aho. Indexed grammars—an extension to context free grammars. *J. Assoc. for Computing Machinery*, 15(4):647–671, 1968.

[2] A. V. Aho. Nested stack automata. *J. Assoc. for Computing Machinery*, 16(3):383–406, 1969.

[3] J. M. Alonso, T. Brady, D. Cooper, V. Ferlini, M. Lustig, M. Mihalik, M. Shapiro, and H. Short. Notes on word hyperbolic groups (ed. H. Short). In E. Ghys, A. Haefliger, and A. Verjovsky, editors, *Group Theory from a Geometrical Viewpoint*, pages 3–63. World Scientific, 1991.

[4] A. V. Anisimov. Group languages. *Kybernetika*, 4:18–24, 1971.

[5] M. Benois. Partie rationelles du groups libre. *C. R. Acad. Sci. Paris, Sér. A*, 269:1188–1190, 1969.

[6] J. Berstel. *Transductions and Context-Free Languages.* Teubner, Stuttgart, 1979.

[7] M. R. Bridson. Geodesics and curvature in metric simplicial complexes. In E. Ghys, A. Haefliger, and A. Verjovsky, editors, *Group Theory from a Geometrical Viewpoint*, pages 373–463. World Scientific, 1991.

[8] Gerhard Buntrock and Krzysztof Loryś. On growing context-sensitive languages. In *Automata, languages and programming (Vienna, 1992)*, pages 77–88. Springer, Berlin, 1992.

[9] D. E. Cohen. *Combinatorial Group Theory: A Topological Approach.* Number 14 in London Math. Soc. Student Texts. Cambridge University Press, 1989.

[10] Donald J. Collins and Charles F. Miller III. The conjugacy problem and subgroups of finite index. *Proc. London Math. Soc. (3)*, 34(3):535–556, 1977.

[11] M. Dehn. Über die Topologie des dreidimensionalen Raumes. *Math. Ann.*, 69:137–168, 1910.

[12] M. Dehn. Über unendliche diskontinuierliche Gruppen. *Math. Ann.*, 71:116–144, 1911.

[13] P. J. Denning, J. B. Dennis, and J. E. Qualitz. *Machines, Languages and Computation.* Prentice–Hall, 1978.

[14] M. J. Dunwoody. The accessibility of finitely presented groups. *Invent. Math.*, 85:449–457, 1985.

[15] S. Eilenberg. *Automata, Languages, and Machines*, volume A. Academic Press, 1974.

[16] S. Eilenberg and M. Schützenberger. Rational sets in commutative monoids. *J. Algebra*, 13:173–191, 1969.

[17] D. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson, and W. P. Thurston. *Word Processing in Groups.* Jones and Bartlett, 1992.

[18] B. Farb. The extrinsic geometry of subgroups and the generalized word problem. *Proc. London Math. Soc.*, 68(3):577–593, 1994.

[19] R. Floyd and R. Beigel. *The Language of Machines.* Computer Science Press, New York, 1994.

[20] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, 1979.

[21] S. M. Gersten and H. Short. Rational subgroups of biautomatic groups. *Annals Math.*, 134:125–158, 1991.

[22] E. Ghys and P. de la Harpe editors. *Sur les groupes hyperboliques d'après Mikhael Gromov.* Number 83 in Progress in Math. Birkhäuser, 1990.

[23] R. Gilman. Computations with rational subsets of confluent groups. In *EUROSAM '84*, number 174 in Lecture Notes in Comp. Sci., pages 331–340. Springer–Verlag, 1984.

[24] R. Gilman. Groups with a rational cross-section. In S. M. Gersten and J. R. Stallings, editors, *Combinatorial Group Theory and Topology*, number 111 in Annals Math. Studies, pages 175–183. Princeton University Press, 1987.

[25] R. Gilman. Formal languages and infinite groups. In G. Baumslag, D. Epstein, R. Gilman, H. Short, and C. Sims, editors, *Geometric and Computational Perspectives on Infinite Groups*, volume 25 of *DIMACS*, pages 27–51. A.M.S., 1996.

[26] R. Gilman and M. Shapiro. On groups whose word problem is solved by a nested stack automaton. Preprint. Available at: `http://attila.stevens-tech.edu/~rgilman`.

[27] J. Goldstine. Formal languages and their relation to automata: what Hopcroft and Ullman didn't tell us. In R. Book, editor, *Formal Language Theory: Perspectives and Open Problems*, pages 109–140. Academic Press, 1980.

[28] M. Gromov. Hyperbolic groups. In S. M. Gersten, editor, *Essays in Group Theory*, number 8 in M.S.R.I. Publications, pages 75–263. Springer–Verlag, 1987.

[29] Z. Grunschlag. Computing angles in hyperbolic groups. In Robert Gilman, editor, *Geometric Group Theory and Computer Science*, Contemp. Math. A.M.S. to appear.

[30] Z. Grunschlag. Deciding if the angle is zero inside free groups. To appear in *Top. and its Appl.*

[31] M. A. Harrison. *Introduction to Formal Language Theory.* Addison–Wesley, 1978.

[32] T. Herbst. On a subclass of context-free groups. *Informatique théorique et Applications*, 25(3):255–272, 1991.

[33] T. Herbst and R. M. Thomas. Group presentations, formal languages and characterizations of one-counter groups. *Theoret. Comput. Sci.*, 112:187–213, 1993.

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

[35] J. M. Howie. *Automata and Languages.* Oxford, 1991.

[36] I. Kapovich. Detecting quasiconvexity: algorithmic aspects. In G. Baumslag, D. Epstein, R. Gilman, H. Short, and C. Sims, editors, *Geometric and Computational Perspectives on Infinite Groups*, volume 25 of *DIMACS*, pages 91–99. A.M.S., 1996.

[37] I. Kapovich. Quasiconvexity and amalgams. *Internat. J. Algebra Comput.*, 7:771–811, 1997.

[38] I. Kapovich and H. Short. Greenberg's theorem for quasiconvex subgroups of word hyperbolic groups. *Can. J. Math.*, 48:1224–1244, 1996.

[39] S. C. Kleene. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, number 34 in Annals Math. Studies, pages 3–42. Princeton, 1956.

[40] R. C. Lyndon and P. E. Schupp. *Combinatorial Group Theory.* Number 89 in Ergebnisse der Mathematik und ihrer Grenzgebiete. Springer–Verlag, 1977.

[41] W. Magnus, A. Karrass, and D. Solitar. *Combinatorial Group Theory: Presentations of Groups in Terms of Generators and Relations.* Dover, 2nd edition, 1976.

[42] J. P. McCammond and D. T. Wise. Coherent groups and the perimeter of two-complexes. Preprint.

[43] D. Muller and P. Schupp. Groups, the theory of ends, and context-free languages. *J. Computer and System Sciences*, 26:295–310, 1983.

[44] Peter M. Neumann, Gabrielle A. Stoy, and Edward C. Thompson. *Groups and geometry.* The Clarendon Press, Oxford University Press, New York, 1994.

[45] P. Papasoglu. An algorithm detecting hyperbolicity. In G. Baumslag, D. Epstein, R. Gilman, H. Short, and C. Sims, editors, *Geometric and Computational Perspectives on Infinite Groups*, volume 25 of *DIMACS*, pages 193–200. A.M.S., 1996.

[46] E. Rips. Subgroups of small cancellation groups. *Bull. London Math. Soc.*, 14:45–47, 1982.

[47] A. Salomaa. *Formal Languages.* Academic Press, 1973.

[48] J. P. Serre. *Trees*. Springer–Verlag, 1980. (Translated by J. Stillwell).

[49] H. Short. Quasiconvexity and a theorem of Howson's. In E. Ghys, A. Haefliger, and A. Verjovsky, editors, *Group Theory from a Geometrical Viewpoint*, pages 168–176. World Scientific, 1991.

[50] J. R. Stallings. Topology of finite graphs. *Invent. Math.*, 71:551–565, 1983.

[51] J. R. Stallings. Non-positively curved triangles of groups. In E. Ghys, A. Haefliger, and A. Verjovsky, editors, *Group Theory from a Geometrical Viewpoint*, pages 491–503. World Scientific, 1991.

[52] J. R. Stallings. Geometric understanding of the angle between subgroups. In *Groups — Korea '94*, pages 295–301. de Gruyter, 1995.

[53] B. Steinberg. Monoid kernels and profinite topologies on the free abelian group. Preprint.

[54] B. Steinberg. Finite state automata: A geometric approach. Tech. Rep. Univ. of Porto, 1999.

# Index