NOTES ON AUTOMATA

1. Monoids acting on sets

We say that a monoid $\mathbf{S}$ with identity element ϵ *acts* on a set Q if $q(st) = (qs)t$ and $q\epsilon = q$. As with groups, if we set $s \cong t$ whenever $qs = qt$ for all $q \in Q$, then $\cong$ is a congruence relation and $\mathbf{S}/\cong$ acts faithfully on Q.

A *deterministic finite automaton* (DFA) is $\mathbf{M} = \langle Q, \mathbf{S}, \cdot, q_0, F\rangle$ where

(1) Q is a finite set (whose elements are called *states*),
(2) $\mathbf{S}$ is a monoid,
(3) $\cdot$ is an action of $\mathbf{S}$ on Q,
(4) $q_0 \in Q$ (called the *starting state*), and
(5) $F \subseteq Q$ (whose elements are called *final states* or *accept states*).

We define the *language* of $\mathbf{M}$ as $L_{\mathbf{S}}(\mathbf{M}) = \{s \in \mathbf{S} : q_0 s \in F\}$, and say that these elements are *recognized* or *accepted* by $\mathbf{M}$. If $h : \mathbf{T} \to \mathbf{S}$ is a homomorphism, then $\mathbf{T}$ acts on Q via $q * t = q \cdot h(t)$. Thus we obtain another DFA $\mathbf{M}' = \langle Q, \mathbf{T}, *, q_0, F\rangle$ with the corresponding language $L_{\mathbf{T}}(\mathbf{M}')$. We are particularly interested in the case when $\mathbf{T}$ is the free monoid generated by some finite alphabet A.

We now define a *nondeterministic finite automaton* (NFA) *with ϵ-moves* and show how one can be represented as a DFA with the same language. An NFA is a quadruple $\mathbf{N} = \langle Q, S_0, q_0, F\rangle$ where

(1) Q is a finite set,
(2) S_0 is a finite set of binary relations on the set Q, with a distinguished member $\epsilon \in S_0$ such that $\epsilon \supseteq \Delta$ (equality),
(3) $q_0 \in Q$, and
(4) $F \subseteq Q$.

Let $\mathcal{E}(Q)$ denote the lattice of ϵ-closed subsets of Q (subsets such that $x \in P$ and $x\epsilon y$ implies $y \in P$). Note $\mathcal{E}(Q)$ is closed under union and intersection. Denote the corresponding closure operator on Q also by $\mathcal{E}$. For the purposes of notation, define $\overline{q}_0 = \mathcal{E}(\{q_0\})$ and $\overline{F} = \mathcal{E}(F)$.

The relations of S_0 act on $\mathcal{E}(Q)$ in a natural way:

$$P * s = \mathcal{E}(\{q \in Q : \exists p \in P \text{ with } p\, s\, q\}).$$

Date: April 29, 2005.

This extends to an action of the semigroup $\mathbf{S}$ of transformations on $\mathcal{E}(Q)$ generated by S_0. (Some members of S_0 could act identically, and hence be identified in $\mathbf{S}$.)

The language of the NFA $\mathbf{N}$ with respect to $\mathbf{S}$) is defined to be $K_{\mathbf{S}}(\mathbf{N}) = \{s \in \mathbf{S} : \overline{q_0} * s \cap F \neq \emptyset\}$. Again, if $h : \mathbf{T} \to \mathbf{S}$ is a homomorphism, then an action $\times$ of $\mathbf{T}$ on $\mathcal{E}(Q)$ is induced *via* $P \times t = p * h(t)$. The language of $\mathbf{N}$ w.r.t. $\mathbf{T}$ is defined to be $K_{\mathbf{T}}(\mathbf{N}) = \{t \in \mathbf{T} : \overline{q_0} \times t \cap F \neq \emptyset\}$.

Let $\widetilde{F} = \{P \in \mathcal{E}(Q) : P \cap F \neq \emptyset\}$. The corresponding DFA with $L_{\mathbf{S}}(\mathbf{M}) = K_{\mathbf{S}}(\mathbf{N})$ is $\mathbf{M} = \langle \mathcal{E}(Q), \mathbf{S}, *, \overline{q_0}, \widetilde{F}\rangle$. Likewise, if $h : \mathbf{T} \to \mathbf{S}$, then for the DFA $\mathbf{M}' = \langle \mathcal{E}(Q), \mathbf{T}, \times, \overline{q_0}, \widetilde{F}\rangle$ we have $L_{\mathbf{T}}(\mathbf{M}') = K_{\mathbf{T}}(\mathbf{N})$.

If follows from the above equivalence that DFAs with a set of starting states Q_0 have the same languages as DFAs with a single starting state q_0. The same is NOT true for the final states: DFAs with F a single state have more restrictive languages than those with F an arbitrary subset. *Question:* Is there a DFA with a single final state that accepts bitstrings whose sum is not zero mod 3?

2. Regular expressions

Given a monoid $\mathbf{S}$, let

$$\mathrm{Rec}(\mathbf{S}) = \{X \subseteq \mathbf{S} : \exists \mathbf{M} \quad X = L_{\mathbf{S}}(\mathbf{M})\}.$$

That is, what subsets of a monoid are recognized by a finite automaton? *Question:* What is known about this? In this section, we determine $\mathrm{Rec}(\mathbf{T})$ for the case when $\mathbf{T} = A^*$ is the free monoid generated by a finite alphabet A. This is of course isomorphic to the semigroup of all finite strings of symbols from A, including the empty string denoted ϵ, with the operation of concatenation.

We define the *regular subsets of* A^* as follows.

(1) $\emptyset \in \mathrm{Reg}(A^*)$,
(2) $\{\epsilon\} \in \mathrm{Reg}(A^*)$,
(3) $\{a\} \in \mathrm{Reg}(A^*)$ for each $a \in A$,
(4) if X, $Y \in \mathrm{Reg}(A^*)$, then $X \cup Y \in \mathrm{Reg}(A^*)$,
(5) if X, $Y \in \mathrm{Reg}(A^*)$, then $X \cdot Y \in \mathrm{Reg}(A^*)$,
(6) if $X \in \mathrm{Reg}(A^*)$, then $X^* \in \mathrm{Reg}(A^*)$.[1]

It is useful to think of the regular subsets as being determined by *regular expressions* over A.

[1]Here, in abuse of notation, A^* denotes the free monoid generated by A, and for $X \subseteq A^*$, X^* is the submonoid generated by X, so that $X^* = \bigcup_{k\geq 0} X^k$. As a technicality, note that $X^+ = \bigcup_{k\geq 1} X^k = X \cdot X^*$ is also a regular set, which is different if $\epsilon \notin X$.

(1) $\emptyset$ is a regular expression and $L(\emptyset) = \emptyset$.
(2) ϵ is a regular expression and $L(\epsilon) = \{\epsilon\}$.
(3) a is a regular expression and $L(a) = \{a\}$ for each $a \in A$.
(4) If r and s are regular expressions, then $(r + s)$ is a regular expression and $L(r + s) = L(r) \cup L(s)$.[2]
(5) If r and s are regular expressions, then (rs) is a regular expression and $L(rs) = L(r) \cdot L(s)$.
(6) If r is a regular expression, then (r^*) is a regular expression and $L(r^*) = (L(r))^*$.

Thus $X \in \mathrm{Reg}(A^*)$ if and only if $X = L(r)$ for some regular expression r.

Theorem 1. *If r is a regular expression over A, then there is a NFA $\mathbf{N}$ such that $K_{A^*}(\mathbf{N}) = L(r)$. Hence $\mathrm{Reg}(A^*) \subseteq \mathrm{Rec}(A^*)$.*

Proof. See the diagrams on pages 30–31 of Hopcroft and Ullman [2]. □

Theorem 2. *For any DFA $\mathbf{M}$ over A^*, $L_{A^*}(\mathbf{M}) = L(r)$ for some regular expression r. Hence $\mathrm{Rec}(A^*) \subseteq \mathrm{Reg}(A^*)$.*

Proof. The proof defines $L(q_1, q_2, X)$ the be the union of the set of languages defined by paths with source q_1, target q_2, and visiting only states in X. The first claim is that $L(q_1, q_2, X)$ is defined by a regular expression; this is proved by induction on $|X|$. The second claim is that $L(\mathbf{M}) = \bigcup_{f \in F} L(s_0, f, Q)$. See page 13 of Epstein [1] for details. □

Corollary 3. $\mathrm{Reg}(A^*) = \mathrm{Rec}(A^*)$.

As an application, the legal identifiers in a computer language are given by a regular expression. For example, an identifier in Fortran is a letter followed by at most 5 letters and/or digits (case insensitive). An automaton to recoginize these identifiers would be part of a compiler.

Theorem 4. *If L is a regular language over an alphabet A, i.e., $L \in \mathrm{Reg}(A^*)$, then the language consisting of the strings of L written in the reverse order is also regular.*

Theorem 5. **(Pumping Lemma)** *Let L be a regular language. Then there is a number $n > 0$ such that any $x \in L$ of length at least n is of the form $x = uvw$, where $|v| > 0$ and $uv^iw \in L$ for all $i \geq 0$.*

As an application, let $A = \{a, b\}$ and consider $K = \{a^k b^k : k \geq 0\}$. By the Pumping Lemma, K cannot be a regular language.

Similarly, let $B = \{a\}$ and consider $L = a^{k^2} : k \geq 0\}$. Again by the Pumping Lemma, L cannot be a regular language.

[2]In various sources, $(r + s)$ is alternately denoted $(r \vee s)$ or $(r|s)$.

Theorem 6. (Myhill-Nerode) *Given a language L over an alphabet A, consider the equivalence relation on A^* defined by*

$$w_1 \equiv_L w_2 \quad \text{if} \quad \forall u \in A^* \;\; w_1 u \in L \iff w_2 u \in L.$$

Then L is regular if and only if there are only finitely many $\equiv_L$-equivalence classes.

Proof. See page 15 of [1]. □

Let A and B be alphabets, and let $f_0 : A \to B^*$ be a map. Since A^* is a free monoid, there is a unique homomoprhism $f : A^* \to B^*$ extending f_0. As always, this induces a map from subsets of A^* (languages over A) to subsets of B^* (languages over B).

Theorem 7. *Let $f : A^* \to B^*$ be a homomorphism. If $L \subseteq A^*$ is a regular language, then so is $f(L)$. If $K \subseteq B^*$ is a regular language, then so is $f^{-1}(K)$.*

Proof. See page 17 of [1]. □

The boolean operators $\neg$, $\vee$, $\wedge$ of the predicate calculus apply naturally as set-theoretic boolean operators on languages.

Theorem 8. *If K and L are regular languages, then so are $\neg K$, $K \vee L$ and $K \wedge L$.*

Proof. To see that the complement $\neg K = A^* - K$ of a regular language is a regular language, let $\mathbf{M} = \langle Q, A^*, \cdot, q_0, F\rangle$ be a DFA with $L(\mathbf{M}) = K$. Then $\mathbf{N} = \langle Q, A^*, \cdot, q_0, Q - F\rangle$ is a DFA with $L(\mathbf{N}) = A^* - K$.

To obtain the intersection, we can either use DeMorgan's Law that $K \wedge L = \neg(\neg K \vee \neg L)$, or construct a DFA accepting $K \cap L$ directly by using direct products. □

Question: Given a regular expression, how do you find a regular expression representing its complement?

Define the *quotient* R/L of languages by

$$R/L = \{x \in A^* : \exists y \in L \;\; xy \in R\}.$$

Theorem 9. *If R is regular and $\mathbf{L} \subseteq A^*$ is any language, then R/L is a regular language.*

Proof. See page 63 of [2]. □

In general, it is impossible to decide whether a regular expression yields a nonempty language. However, we do have the next two results from pages 63–64 of [2]. The first is a variation on the argument of the Pumping Lemma, and the second an application of the first to the symmetric difference of languages.

Theorem 10. *The set of words accepted by a finite automaton with n states is:*

1. *nonempty if and only if the finite automaton accepts a word of length less than n;*
2. *infinite if and only if the atomaton accepts some word of lenth ℓ with $n \leq \ell < 2n$.*

Thus there is an algorithm to determine whether a given finite automaton accepts zero, a finite number, or an infinite number of words.

Theorem 11. *There is an algorithm to determine if two finite automata are equivalent, i.e., if they accept the same language.*

3. Automatic Groups

Roughly speaking, an automatic group is a finitely generated group for which one can check, by means of a finite state automaton, whether two words in a given presentation represent the same element or not, and whether or not the elements they represent differ by right multiplication by a single generator [1].

Let $\mathbf{G}$ be a group, A an alphabet, $\pi_0 : A \to \mathbf{G}$ a map, and $\pi : A^* \to \mathbf{G}$ the unique semigroup homomorphism extending π_0. If $\pi(A^*) = \mathbf{G}$, we say that A (more properly, $\pi(A)$) *generates* $\mathbf{G}$ *as a semigroup.* We are often interested in the case where $x \in A$ implies $x^{-1} \in A$. It is convenient to denote specific pairs of formal inverses by uppercase and lowercase variants of the same letter: thus the inverse of x will be represented by X, and so on.

Example 1: free group. Let $\mathbf{F}_2$ be the free group on two (group) generators, and let $A = \{x, y, X, Y\}$. It is easy to devise an automaton $\mathbf{M}$ such that $L = L(\mathbf{M})$ is the language of all reduced strings, i.e., strings in which x and X are never adjacent and y and Y are never adjacent. Thus L is regular, and the restriction map $\pi|_L : L \to \mathbf{F}_2$ is a bijection.

Example 2: free abelian group. Let $\mathbf{A}_3$ be a free abelian group on three generators, and let $A = \{x, y, z, X, Y, Z\}$. If L is the language defined by the regular expression $(x^* \vee X^*)(y^* \vee Y^*)(z^* \vee Z^*)$, then the restriction map $\pi|_L : L \to \mathbf{A}_3$ is bijective.

Example 3: free nil-2 group Let $\mathbf{G}$ be the free nilpotent group of rank 2 generated by two elements x and y. Let $z = [x, y] = x^{-1}y^{-1}xy$, so that z commutes with x and y. Taking the same alphabet and language as in Example 2, we again obtain a bijection from L to $\mathbf{G}$.

Examples 2 and 3 show that, even when the map from the language to the group is bijective, the language may be far from determining the group.

Given an alphabet, let $\mathbf{F}(A)$ denote the free group generated by A. An element of $\mathbf{F}(A)$ is called a *word* over A. A subset $R \subseteq \mathbf{F}(A)$ is called a set of *relators*, and the pair (A, R) for a *group presentation* for the group $\mathbf{G} = \mathbf{F}(A)/\mathbf{N}$, where $\mathbf{N}$ is the smallest normal subgroup of $\mathbf{F}(A)$ containing R. We write $\mathbf{G} = \langle A|R\rangle$. We say that $\mathbf{G}$ is finitely presented if R is finite.

There are really three homomorphisms operating here: $\varphi : A^* \to \mathbf{F}(A)$, the map $\nu : \mathbf{F}(A) \to \mathbf{G}$ with kernel $\mathbf{N}$, and $\pi = \nu \circ \varphi : A^* \to \mathbf{G}$.

The *word problem* in $\mathbf{G}$ consists in finding an algorithm that takes as its input a word w over A and answers Yes or No, depending on whether or not w represents the identity in $\mathbf{G} = \langle A|R\rangle$, i.e., whether $\pi(w) = 1$ or equivalently $\varphi(w) \in \mathbf{N}$.

Theorem 12. *Let $\mathbf{G} = \langle A|R\rangle$ be a finitely presented group. If there is a regular language L such that $\pi|_L : L \to \mathbf{G}$ is surjective, and if the inverse image $L_0 = \pi^{-1}(1) \cap L$ is also a regular language, then the word problem in $\mathbf{G}$ is solvable.*

Proof. See page 32 of [1]. □

Corollary 13. *Let $\mathbf{G} = \langle A|R\rangle$ be a finitely presented group. If there is a regular language L such that $\pi|_L : L \to \mathbf{G}$ is surjective and finite-to-one, then the word problem in $\mathbf{G}$ is solvable.*

Let $\mathbf{G}$ be a group, A a set of semigroup generators, and $\mathbf{H}$ a subgroup of $\mathbf{G}$. Let $\mathbf{H}\backslash\mathbf{G}$ denote the collection of right cosets $\mathbf{H}g$ for $g \in \mathbf{G}$. Of course, $\mathbf{G}$ acts on $\mathbf{H}\backslash\mathbf{G}$ by right multiplication. The *Cayley graph* $\Gamma(\mathbf{H}\backslash\mathbf{G}, A)$ is a directed, labelled graph with vertices $\mathbf{H}g$ for $g \in \mathbf{G}$, labels x for $x \in A$, and edges $\mathbf{H}g \to \mathbf{H}gx$ with label x. The *basepoint* of a Cayley graph is the vertex $\mathbf{H}$. When $\mathbf{H} = \{1\}$, we write just $\Gamma(\mathbf{G}, A)$.

If $\mathbf{H}$ is a normal subgroup of $\mathbf{G}$, then left and right cosets coincide, and left multiplication by an element of $\mathbf{G}$ induces a homeomorphism of $\Gamma(\mathbf{H}\backslash\mathbf{G}, A)$. Thus the Cayley graph is homogeneous: any two vertices look alike.

The Cayley graph is connected because A generates $\mathbf{G}$. We make it into a metric space by defining the distance between two vertices (cosets) to be the minimum number of edges in any path connecting the two edges (ignoring the direction of the edges). We can extend the metric to the edges by making each edge isometric to $[0, 1]$ or, for loops, a circle of length 1.

When $\mathbf{H} = \{1\}$, the metric thus obtained is called the *word metric.* For an element g, then $|g| = d(1, g)$ is the minimum length of a word over A representing g, and $d(g_1, g_2) = |g_1^{-1}g_2|$. In other words,

$d(g_1, g_2) = \min\{|u| : u \in A^* \text{ and } g_1\pi(u) = g_2\}$. Note that if $w \in A^*$ and $\pi(w) = g$, then $|w| \geq |g|$. Given a word $w \in A^*$, we can define a path $\widehat{w}$ along the edges of $\Gamma(\mathbf{G}, A)$ from the basepoint 1 to $\pi(w)$ in the obvious way.

If we read off the labels as we go around a loop in the Cayley graph $\Gamma(\mathbf{G}, A)$, the word we obtain represents the identity in $\mathbf{G}$. Given a finitely presented group $\mathbf{G} = \langle A|R\rangle$, for any element $w \in A^*$ with $\pi(w) = 1$, the corresponding free group element $\varphi(w)$ can be written in the form

$$\varphi(w) = \prod_{i=1}^{n} v_i r_i^{\pm 1} v_i^{-1} \qquad (\dagger)$$

for some n, $r_i \in R$ and $v_i \in \mathbf{F}(A)$. In the Cayley graph, this corresponds to subdividing the loop labelled by w into a number of loops labelled by the relators r_i; the v_i's translate the model relator loops, which are based at the basepoint, to the place where they fit into the loop labelled by w.

In order to solve the word problem for $\langle A|R\rangle$, it suffices to find recursive bounds (in terms of $|w|$) on the parameters n and $|v_i|$ in ($\dagger$). The next lemma gives a bound on the lengths of the v_i's that must be considered. First, however, we must make some adjustments in the terms of R.

The presentation R is said to be *cyclically reduced* if each $r \in R$ is reduced, and no $r \in R$ is of the form $r = xtx^{-1}$ with $x \in A$. This amounts to saying that the terms of R are chosen to have minimal length, for replacing $r = xtx^{-1}$ by t would yield an equivalent presentation using a shorter relator.

Lemma 14. *Let $\langle A|R\rangle$ be a finitely presented group, with R cyclically reduced. If w is a reduced word expressed in the form ($\dagger$), then we can rewrite the product so that each v_i has length at most $(|w| + 2k)2^n$, where $k = \max |r_i|$.*

Proof. A handwritten proof will be distributed. The proof sketched on pages 40–43 of [1] is incomplete. See also Madlener and Otto [3]. □

Suppose there is a recursive function η such that, whenever an element $w \in A^*$ has $\pi(w) = 1$, then $\phi(w)$ can be written in the form ($\dagger$) with $n \leq \eta(|w|)$. Then the word problem for the group $\langle A|R\rangle$ is solvable. (The converse is also true.)

The minimum value of n for a word w with $\pi(w) = 1$ is called its *area*, and the expression $area(w) \leq \eta(|w|)$ is called an *isoperimetric inequality.* In the 1950's, Novikov and Boone independently proved that not all finitely presented groups have a solvable word problem. We will show that automatic groups do.

4. Automatic groups

By an *automaton over* (A, A) we mean an automaton over $A' \times A'$ where $A' = A \cup \{\$\}$ and \$ is an end-of-string symbol. Thus a typical element would be $(abba, ba\$\$)$, with the \$-signs used to give the strings equal length.

An *automatic structure* on a group $\mathbf{G}$ consists of a set A of semigroup generators, a finite state automaton W over A, and finite state automata M_x over (A, A) for $x \in A \cup \{\epsilon\}$, satisfying the following conditions.

(1) $\pi(L(W)) = \mathbf{G}$, i.e., $\pi|_{L(W)}$ is surjective.
(2) $(w_1, w_2) \in L(M_x)$ if and only if w_1, $w_2 \in L(W)$ and $\pi(w_1 x) = \pi(w_2)$.

We call W the *word acceptor*, M_ε the *equality recognizer*, and each M_x for $x \in A$ a *multiplier automaton.* An *automatic group* is one that admits an automatic structure.

Example 1: finitely generated free groups.

Example 2: finitely generated free abelian groups.

Example 3: finite groups.

Non-example 4: $\langle x, y \,|\, yx = xy^2\rangle$ (by Theorem 16). See pages 154–155 of [1] for the Cayley graph of this group.

Non-example 5: An infinite non-abelian nilpotent group is not automatic.

For a word w and integers $t \geq 0$, let $w(t)$ denote the prefix (initial segment) of w of length t, or w itself if $t \geq |w|$. The following lemma shows that automatic groups have a sort of Lipschitz property in the word metric.

Lemma 15. *If $\mathbf{G}$ has an automatic structure, then there is a constant k such that, whenever (w_1, w_2) is accepted by M_x for some $x \in A \cup \{\varepsilon\}$, then in the word metric $d(\pi(w_1(t)), \pi(w_2(t))) \leq k$ for all $t \geq 0$.*

Proof. See page 46 of [1]. □

Using this lemma, one can show that if $\mathbf{G}$ has an automatic structure, then one can construct a *standard automatic structure* for $\mathbf{G}$. That is, given the word acceptor W and sufficiently large neighborhood of the identity (say the set of elements of $\mathbf{G}$ with word length at most k), one can construct the multiplier automata M_x in a canonical way. This construction is described on page 47 of [1]. It means that we can talk about "(A, W)" or even "(A, L)" as an automatic structure for $\mathbf{G}$, the rest being obtained in the standard way. The construction also yields an important characterization.

Theorem 16. *Let* $\mathbf{G}$ *be a group and* A *a finite set of semigroup generators for* $\mathbf{G}$*. Let* W *be a finite state automaton over* A *such that* $\pi : L(W) \to \mathbf{G}$ *is surjective. Then* A *and* W *are part of an automatic structure on* $\mathbf{G}$ *if and only if there is a number* k *such that whenever* $w_1, w_2 \in L(W)$ *and* $\pi(w_1 x) = \pi(w_2)$ *for some* $x \in A \cup \{\varepsilon\}$*, then in the word metric* $d(\pi(w_1(t)), \pi(w_2(t))) \leq k$ *for all* $t \geq 0$*.*

Corollary 17. *If* (A, L) *is an automatic structure for* $\mathbf{G}$*, and* $L' \subseteq L$ *is a regular language over* A *that maps onto* $\mathbf{G}$*, then* (A, L') *is also an automatic structure for* $\mathbf{G}$*.*

Lemma 18. *Let* $\mathbf{G}$ *be an automatic group and let* (A, L) *be an automatic structure for* $\mathbf{G}$*. There is a constant* N *such that if* $w \in L$ *and* $g \in \mathbf{G}$ *and* $d(g, \pi(w)) \leq 1$ *in the Cayley graph of* $\mathbf{G}$ *(i.e.,* $g = \pi(w)x$ *for some* $x \in A \cup \{\epsilon\}$*), then*

(1) *there exists* $u \in L$ *with* $g = \pi(u)$ *and* $|u| \leq |w| + N$*, and*
(2) *if* $g = \pi(v)$ *for some* $v \in L$ *with* $|v| > |w| + N$*, then* $g = \pi(t)$ *for infinitely many words* $t \in L$*.*

It follows from (1) *that for any word* $w \in A$*, there exists* $u \in L$ *with* $\pi(u) = \pi(w)$ *and* $|u| \leq N|w| + n_0$*, where* n_0 *is the length of an accepted representative of* 1*.*

This is turn yields the big theorem. (We are following [1], pages 47–51 closely here.)

Theorem 19. *Let* $\mathbf{G}$ *be an automatic group with automatic structure* (A, L)*. For any word* $w \in A$*, we can find a word* $u \in L$ *with* $\pi(u) = \pi(w)$ *in time proportional to* $|w|^2$*.*

This shows that the word problem for $\mathbf{G}$ is solvable, for we can then take some word $e \in L$ with $\pi(e) = 1$ and feed the pair (u, e) to the equality recognizer automaton M_ϵ.

Theorem 20. *Every automatic group* $\mathbf{G}$ *is finitely presented and satisfies a quadratic isoperimetric inequality, that is, there is a finite presentation* $\mathbf{G} = \langle A|R \rangle$ *and for any such presentation, if* $\pi(w) = 1$ *then in* (†) *we can take* $n = \mathcal{O}(|w|^2)$*.*

References

[1] D. Epstein *et al.*, Word Processing in Groups, Jones and Bartlett Pub.

[2] J. Hopcroft and J. Ullman, Introduction to Automata Theory, Languages and Computation, Addison-Wesley, 1979.

[3] K. Madlener and F. Otto, *Pseudo-natural algorithms for the word problem for finitely presented monoids and groups*, Journal of Symbolic Computation **1** (1985), 383–418.