Geometry & Topology Monographs
Volume 1: The Epstein Birthday Schrift
Pages 249–260

Automatic groups, subgroups and cosets

Derek F Holt

Abstract The history, definition and principal properties of automatic groups and their generalisations to subgroups and cosets are reviewed briefly, mainly from a computational perspective. A result about the asynchronous automaticity of an HNN extension is then proved and applied to an example that was proposed by Mark Sapir.

AMS Classification 20F32; 20F05

Keywords Automatic Groups, HNN extensions

The concept of an automatic group was introduced in 1986 by Thurston, motivated by some results of Jim Cannon on hyperbolic groups. Much of the basic theory of this important class of groups was developed by David Epstein during the following few years.

In the first section of this paper, we review briefly the history, definition and properties of automatic groups and their generalisation to subgroups and cosets, mainly from a perspective of carrying out efficient computations within such groups and their subgroups. In the second section, we prove a result about the (asynchronous) automaticity of an HNN extension, and use it, together with the results of some machine computations, to prove that a particular group, defined by Mark Sapir, is asynchronously automatic.

1 Definitions and discussion

1.1 Automatic groups

In [2], J.W. Cannon proved certain geometrical properties of the Cayley graph of cocompact discrete hyperbolic groups. Two years later, in 1986, W. Thurston noticed that some of these properties could be reformulated in terms of finite state automata (fsa; this abbreviation will be used for both the singular and plural).

In particular, the goedesic paths in the Cayley graph that start at the origin form a regular set or, equivalently, they form the language of an fsa. Furthermore, any pair of such geodesic paths that end at the same or neighbouring vertices lie within a bounded distance of each other. It can be deduced that such geodesic pairs also form the language of an fsa. This led Thurston to formulate the following general definition.

Definition 1.1 Let G be a group with finite generating set X, let $A = X \cup X^{-1}$, and let $A' = A \cup \{\$\}$, where $\$ \notin A$. Then G is said to be *automatic* (with respect to X), if there exist fsa W and M_a for each $a \in A'$, such that

(i) W has input alphabet A, and accepts at least one word in A^* mapping onto each element of G.

(ii) Each M_a has input alphabet $A' \times A'$, it accepts only padded pairs, and it accepts the padded pair (w^+, x^+) for $w, x \in A^*$ if and only if $w, x \in L(W)$ and $wa =_G x$.

Here A^* as usual denotes the set of words in A. For $w \in A^*$, $\overline{w}$ denotes the element of G onto which w maps; for $w, x \in A^*$, we also use $w =_G x$ to mean that w, x map onto the same element of G. The extra symbol \$ maps onto the identity element of G. For $w, x \in A^*$, the associated *padded pair* $(w^+, x^+) \in (A' \times A')^*$ is obtained by adjoining symbols \$ to the end of the shorter of w and x to make them have equal length. The language of the fsa W is denoted by $L(W)$. For general properties of finite state automata, the user is referred to any textbook on automata or formal language theory, such as [10].

In the definition, W is called the *word-acceptor* and the M_a the *multiplier* automata. The complete collection $\{W, M_a\}$ is known as an *automatic structure* for G. Note that the multiplier $M_\$$ recognises equality in G between words in $L(W)$. From a given automatic structure, we can always use $M_\$$ to construct another one such that W accepts a unique word mapping onto each element of G; we simply choose the lexicographically least amongst the shortest words that map onto each element as the 'normal form' representative of that element. We shall call such a W a word-acceptor with uniqueness.

The best general reference for the theory of automatic groups is the multi-author book [3]. In particular, it turns out that the automaticity of G is independent of the choice of generating set X. This immediately suggests that the definition is a sensible one, because it means that automaticity is an algebraic property of the group, rather than just a geometrical property of its Cayley graph.

All finite groups are easily seen to be automatic; in fact the class of automatic groups is invariant under finite variations, such as sub- and super-groups of finite index. It is also closed under direct and free products, and includes, for example, all word-hyperbolic groups, braid groups, Coxeter groups and Artin groups of finite and of 'large' type. All automatic groups have finite presentations.

Some of the most important and useful applications of this theory only involve an explicit knowledge of a word acceptor with uniqueness, particularly in the frequently occurring case when the accepted words are all geodesics in the Cayley graph. From such a word-acceptor, one can quickly enumerate unique representatives of all words up to a given length. This can serve as an invaluable time-saving device in certain computer graphics applications, such as drawing tessellations of hyperbolic space on which these groups act freely. One can also use W to compute the growth function for the group (see [5]).

Another important application of automatic structures for groups G is their use for the efficient (quadratic time) solution of the word problem in G. More precisely, the multiplier automata can be used to reduce an arbitrary word in A^* in quadratic time to the G–equivalent word in $L(W)$.

With these applications in mind, a collection of programs was written at Warwick in the late 1980's for computing automatic structures. These programs take a finite presentation of the group G as input. Currently, they only work for so-called *shortlex* structures, which are those in which $L(W)$ consists of the lexicographically least amongst the shortest words that map onto each group element. (So W depends upon the order of A as well as on A itself.) Many, but not all, of the known classes of automatic groups are known to possess shortlex structures. The programs are described in some detail in [4] and [8], and in a much more general setting in [3]. The latest version is part of a package called `kbmag` and is available by anonymous `ftp` from `ftp.maths.warwick.ac.uk` in the directory `people/dfh/kbmag2`.

From an algorithmic point of view, there is a close connection between automatic groups and rewriting systems for groups, and the programs used make use of the Knuth–Bendix completion process in groups. However, typically, this process alone would not terminate and in fact automatic groups normally have infinite regular rather than finite complete rewriting systems. When the automatic structure is successfully computed it is, in some sense, enabling this infinite regular system to be used to solve the word problem in a manner that is typically at least as efficient as could be done with a finite rewriting system. The idea of trying to use infinite regular rewriting systems for this purpose was first proposed by Gilman in [7].

Given a word-acceptor automaton for a group, it turns out that the existence and properties of the multiplier automata are equivalent to the so-called (synchronous) fellow-traveller property, which was one of the geometrical properties of hyperbolic groups observed originally by J. W. Cannon, and is defined as follows.

For a word $w \in A^*$ we denote the length of w by $l(w)$ and, for $g \in G$, $l(g)$ (or more precisely $l_A(g)$) denotes the length of the shortest word $w \in A^*$ with $\overline{w} = g$. For $t \geq 0$, $w(t)$ denotes the prefix of w of length t when $t \leq l(w)$, and $w(t) = w$ for $t \geq l(w)$. The fellow-traveller property asserts that there exists a constant k such that, for all $w, x \in L(W)$ and $a \in A$ such that $wa =_G x$, and all $t \geq 0$, we have $l_A(\overline{w(t)}^{-1}\overline{x(t)}) \leq k$. In other words, two travellers proceeding at the same speed along the words w and x from the base point in the Cayley graph of G would always remain a bounded distance away from each other.

The fellow-traveller property enables the multiplier automata M_a to be defined in a uniform manner (see Definition 2.3.3 of [3]). Their state set is the set of triples (s_1, s_2, g), where s_1, s_2 are states of W, and $g \in G$ with $l(g) \leq k$. The start state is $(s_0, s_0, 1)$, where s_0 is the start state of W. For $(a_1, a_2) \in A \times A$, there is a transition from (s_1, s_2, g) to (t_1, t_2, h) with label (a_1, a_2) if and only if there are transitions $s_1 \to t_1$ and $s_2 \to t_2$ in W with labels a_1 and a_2, respectively, and if $a_1^{-1} g a_2 =_G h$. The state (s_1, s_2, g) is a success state of M_a if and only if s_1 and s_2 are success states of W, and $g =_G a$. Thus the M_a differ only in their accept states. (We have omitted a technicality from this definition. To deal with the padding symbol, we have to add an extra state to W which is reached when W is in an accept state and the padding symbol is read.) It is clear that the M_a behave precisely according to Condition (ii) of Definition 1.1. This method is used to construct the M_a in the programs mentioned above.

Note also that it follows from the fellow-traveller property that if g is any fixed element of G and $w, x \in L(W)$ with $wg =_G x$, then w and x fellow-travel with constant at most $kl_A(g)$.

Finally, we must mention the weaker concept of an asynchronously automatic group, because it will arise in the next section. The definition is the same as before, except that the multiplier automata are allowed to read their two input strings at different rates. More precisely, rather than reading one symbol from each of the two input words at each transition, they read a symbol from one of the two words only, where the choice of which word to read is a function of the state of M_a. Of course, when the end of one of the words is reached, the other word must be selected. See Chapter 7 of [3] for the formal definition. Again there is a corresponding fellow-traveller property, in which the imaginary travellers are allowed to move at different speeds. See [3] or Section 7, Part II of [1] for details.

The word problem is still solvable for asynchronously automatic groups, but it is unknown whether this can be done in polynomial time. There are examples known, such as the Baumslag–Solitar groups $\langle x, y \mid y^{-1}x^p y = x^q \rangle$ with $p \neq q$, which are asynchronously automatic but not automatic.

There is a more detailed treatment, with references to the literature, of the synchronous and asynchronous fellow-traveller properties in groups in the article [16] in these proceedings.

1.2 Subgroups

Let $L = L(W)$ be the language of the word-acceptor in an automatic structure of a group G. A subgroup H of G is called L–rational if $L \cap H$ is a regular language (ie the language of an fsa). Such subgroups were studied in [6], where it is proved that L–rational is equivalent to L–quasiconvex. This means that any prefix of a word in $L \cap H$ lies within a bounded distance of H in the Cayley graph of G. Such subgroups are always finitely generated.

An algorithm for constructing an fsa W_H with language $L \cap H$, which takes as input an automatic structure for G and a set of generators for an L–rational subgroup H of G, is described in [12]. A practical and efficient version is described in [11], and an implementation is available in kbmag.

The fsa W_H can be used together with the automatic structure to determine whether a given word in A^* lies in H; that is, to solve the generalised word problem for H in G. First use the the multiplier automata to reduce the word to one in L, and then use W_H to test whether it lies in H. Given W_H and W_K for two subgroups H and K of G, it is easy to intersect their languages to obtain a fsa $W_{H \cap K}$ for their intersection, which can then be used to construct a finite generating set for $H \cap K$.

1.3 Cosets

It is possible to generalise the concept of an automatic group from a notion about the elements of the group to one about the cosets of a given subgroup H of G. This has been carried out by two doctoral students of the author (see [15] and [11]). The definition is as follows.

Definition 1.2 Let G be a group with finite generating set X, let $A = X \cup X^{-1}$, $A' = A \cup \{\$\}$, and let H be a subgroup of G. Then G is said to be *coset automatic* with respect to H, if there exist fsa W, and M_a for each $a \in A'$, such that:

(i) W has input alphabet A, and accepts at least one word in each right coset of H in G;

(ii) Each M_a has input alphabet $A' \times A'$, it accepts only padded pairs, and it accepts the padded pair (w^+, x^+) for $w, x \in A^*$ if and only if $w, x \in L(W)$ and $H\overline{wa} = H\bar{x}$.

Here W is called the *coset word-acceptor* and the M_a the *coset multiplier* automata. The complete collection $\{W, M_a\}$ is known as an *automatic coset system* for the pair (G, H). Again the existence of such a system turns out to be independent of the generating set X of G, and we can, if we wish, always find a new system in which W accepts a unique word in each right coset.

It is proved in [15] that if L is the language of the shortlex automatic structure of a word-hyperbolic group G (or even the set of all geodesics in the Cayley graph of G), and if the subgroup H is L–quasiconvex, then G is coset automatic with respect to H. In [11] the converse is proved for word-hyperbolic groups, although we shall see from the example in the next section that the converse does not hold in general.

An interesting application to the drawing of limit sets of Kleinian groups is described in [14]. As in the graphical applications of ordinary automatic structures, this involves only the use of W to enumerate unique shortest words in each coset.

An algorithm for computing automatic coset systems in the shortlex case was first described in [15], and was implemented by him as a standalone program. It has the disadvantage that it is not usually possible to prove conclusively that the system computed is correct. A different approach is described in [11]. This does enable the output to be proved correct, but it requires an additional hypothesis, to be described below, for it to work at all. It has the further advantage that it has an optional extension to compute a finite presentation for the subgroup H of G after the automatic coset system has been found. This second algorithm, together with the subgroup presentation facility, has been implemented and is available in kbmag. The theory, implementation details and performance statistics can also be found in [9].

These algorithms provide an alternative method to that described in the previous subsection for solving the generalised word problem for H in G. The given word in $w \in A^*$ is reduced (in quadratic time, using the coset multiplier automata) to the unique word w' in the language of the coset word-acceptor for which $H\overline{w} = H\overline{w'}$. Then $w \in H$ if and only if w' is the empty word. The two methods of solving the generalised word problem are to some extent complementary to each other, since there can exist L–quasiconvex subgroups that are not coset automatic and vice versa, although the two concepts are equivalent in word-hyperbolic groups.

The additional hypothesis required for the algorithm developed by Hurt is the following generalisation of the fellow-traveller condition. Let $\{W, M_a\}$ be the shortlex automatic coset system for (G, H) that we are trying to compute. Then, if $(w^+, x^+) \in L(M_a)$ for some $a \in A$, there exists $h \in H$ such that $wa =_G hx$. The hypothesis is that there exists a constant $k \geq 0$ such that for all such w, x, a and h, and all $t \geq 0$, we have $l_A(\overline{w(t)}^{-1} h\overline{x(t)}) \leq k$. In particular, taking $t = 0$, we get $l_A(h) \leq k$, and so in all such equations, only a finite number of elements h occur.

One step in the algorithm is to define the states of the M_a as triples (s_1, s_2, g), as in the automatic group case, but now the initial states are (s_0, s_0, h), where s_0 is the initial state of W, and h is one of the elements of H occurring in the above equations. So the M_a are in fact constructed initially as non-deterministic automata with multiple initial states,

If the hypothesis holds, then we shall say that G is *strongly coset automatic* with respect to H, and call $\{W, M_a\}$ a *strong automatic coset system* for (G, H). It is proved in [11] that word-hyperbolic groups are always strongly coset automatic with respect to their quasiconvex subgroups. It is easy to construct examples in which the hypothesis does not hold, by choosing H to be normal in G, in which case G coset automatic with respect to H is equivalent to G/H automatic, but we do not know of any example in which $\mathrm{Core}_G(H) = 1$.

2 HNN extensions and an example

For the application to be described in this section, we need to strengthen the hypothesis defined at the end of the preceding section for strong automatic coset systems.

Definition 2.1 Let $\{W, M_a\}$ be a strong automatic coset system for (G, H) with respect to the generating set X of G. Let Y be a finite set of generators of H, and let $B = Y \cup Y^{-1}$. Then Y is said to be efficient with respect to

$\{W, M_a\}$ if, for any $w, x \in L(W)$ and any $b \in B, h \in H$ such that $wb =_G hx$, we have either $h = 1$ or $h \in B$.

We are not currently aware of any particular situations under which an efficient generating set could be shown to exist; it would be interesting to investigate this question. In specific examples of automatic coset systems that we have calculated with the programs, it is often possible to observe directly from the calculation that a particular Y is efficient. The concept is useful to us here, because it enables us to prove the following result about HNN extensions, which can then be applied to a specific example. Note that a rather different condition under which an HNN extension of an automatic group is asynchronously automatic has been proved by Shapiro in [17], and results of a similar nature for amalgamated free products are proved in [1].

Theorem 2.2 *Let $\{W, M_a\}$ be a strong automatic coset system for (G, H), let $G = \langle X \mid R \rangle$ be a finite presentation of G, and suppose that H has the efficient generating set Y. Suppose also that H is automatic, and let α be an automorphism of H such that $\alpha(Y) = Y$.*

Then the HNN extension

$$K = \langle X, z \mid R,\, z^{-1}yz = \alpha(y)\ (y \in Y) \rangle$$

is asynchronously automatic.

Proof Let T be a right transversal for H in G. Then by the normal form theorem for HNN extensions (see, for example, Theorem 2.1 (II), page 182 of [13]), each element of $g \in K$ has a unique expression of the form

$$k = ht_1 z^{n_1} t_2 z^{n_2} \dots t_r z^{n_r},$$

where $h \in H$, $t_i \in T$, $n_i \in \mathbb{Z}$, $t_i \notin H$ for $i > 1$ and $n_i \neq 0$ for $i < r$.

We use this normal form in the natural manner to construct a regular language L_K for K on the alphabet $A \cup B \cup \{z^{\pm 1}\}$ where, as before, $A = X \cup X^{-1}$ and $B = Y \cup Y^{-1}$. We are assuming that H is automatic, so we can use the language L_H of the word-acceptor from an associated automatic structure with alphabet B to obtain a word $w_h \in L_H$ for the element $h \in H$ in the normal form. For T we choose the image in G of $L(W)$, and to represent t_i, we choose the unique word $w_i \in L(W)$ with $\overline{w_i} = t_i$. This clearly yields a regular language L_K mapping bijectively onto K.

We now have to show how to construct the asynchronous multiplier automata M_c for $c \in A \cup B \cup \{z^{\pm 1}\}$. Since this is fairly routine, we describe the construction in outline only. Suppose that $u, v \in L_K$ and $uc =_K v$, and let the

HNN normal form of $k = \overline{u}$ be $ht_1z^{n_1}t_2z^{n_2}\ldots t_rz^{n_r}$, as above. If $c = z$ or z^{-1}, then the HNN normal form for kc in K is just $ht_1z^{n_1}\ldots t_rz^{n_r\pm 1}$, and it is easy to construct M_c. So suppose $c \in A \cup B$. We shall suppose that $n_r \neq 0$ and omit the details of the case $n_r = 0$, which are similar. There exist words $c_1 \in B^*$ and $c_2 \in L(W)$ such that $c =_G c_1c_2$. Let $l_B(c_1) = k$. Then, from the assumptions that the generating set Y of H is efficient and that $\alpha(Y) = Y$, it follows that the HNN normal form in K for kc is

$$kc = h't'_1z^{n_1}t'_2z^{n_2}\ldots t'_rz^{n_r}\overline{c_2},$$

where there are elements $x_i, y_i \in H$ $(1 \le i \le r)$, all having B–length at most k, such that $z^{n_r}\overline{c_1} = y_rz^{n_r}$, $t_iy_i = x_it'_i$ for $1 \le i \le r$, $z^{n_i}x_{i+1} = y_iz^{n_i}$ for $1 \le i < r$, and $hx_1 = h'$. Thus we have $u = w_hw_1z^{n_1}\ldots w_rz^{n_r}$ and $v = w_{h'}w'_1z^{n_1}\ldots w'_rz^{n_r}c_2$, where $w_h, w_{h'} \in L_H$ map onto $h, h' \in H$, and $w_i, w'_i \in L(W)$ map onto $t_i, t'_i \in T$ for $1 \le i \le r$.

The multiplier M_c proceeds by reading the words w_h and $w_{h'}$ in parallel at the same rate, then the z^{n_1} together, then t_1 and t'_1 together, and so on. If either of w_h or $w_{h'}$ is longer than the other, then it will wait at the end of the shorter one until the longer word has been read, and similarly for t_i and t'_i. (This explains why M_c needs to be asynchronous. Although $|l(w_h) - l(w_{h'})|$ and $|l(t_i) - l(t'_i)|$ are all bounded, there is no bound on r, and so one of the two tapes of the input of M_c may conceivably get indefinitely ahead of the other; indeed, we have verified that this really can happen in the example below.)

Of course, if either of the two words input to M_c is not in L_K, or if they do not both have the same pattern with respect to the occurrences of z, then they are rejected. Otherwise, if after t transitions, M_c has read $\phi(t)$ symbols from u and $\psi(t)$ from v, then the element $g(t) = \overline{u(\phi(t))}^{-1}\overline{v(\psi(t))}$ of K is remembered as a function of the state of M_c. As in the synchronous case, it is sufficient to show that $l(g(t))$ is bounded.

There are four essentially different situations that occur as the words u, v are read.

(i) $u(\phi(t))$ and $v(\psi(t))$ are prefixes of w_h and ${}_{h'}$, where $|\phi(t) - \psi(t)|$ is bounded. Then the the boundedness of $l(g(t))$ from the automaticity of H, and the fact that $hx_1 = h'$ with $l(x_1) \le k$.

(ii) $u(\phi(t)) = w_hw_1z^{n_1}\ldots w_i(s_1)$ for some i and some prefix $w_i(s_1)$ of w_i, and $v(\psi(t)) = w_{h'}w'_1z^{n_1}\ldots w'_i(s_2)$, where $|s_1 - s_2|$ is bounded. Then $g(t) = \overline{w_i(s_1)}^{-1}x_i\overline{w'_i(s_2)}$, and its boundedness follows from the assumptions that $l_B(x_i) \le k$ and that $\{W, M_a\}$ is a strong automatic coset system for (G, H).

(iii) $u(\phi(t)) = w_h w_1 z^{n_1} \dots w_i z^{m_1}$ for some i and some $m_1 \le n_i$, and $v(\psi(t)) = w_{h'} w'_1 z^{n_1} \dots w'_i z^{m_2}$, where $|m_1 - m_2| \le 1$. Then $g(t) z^{-m_1} y_i z^{m_2}$, and its boundedness follows from $l_B(y_i) \le k$ and the assumption that $\alpha(Y) = Y$.

(iv) $\phi(t) > l(u)$ and $\psi(t) \ge l(v) - l(c_2)$. Then $l(g(t)) \le l(c_2)$ which is clearly bounded.

This completes the proof of the theorem. □

As an application, we shall use this theorem together with the results of some machine computations that were done with kbmag, to prove that the group defined by the presentation

$$\langle\, a, b, r, t, x, z \mid \\ xaxa = t, bxbx = t, bbtaa = t, a^{-1}br = ra^{-1}b, zt = tz, btaz = zbta \,\rangle$$

is asynchronously automatic.

This group, which we shall denote by K, was originally proposed by Mark Sapir as a possible building block in his attempts to construct groups with given Dehn functions. However, he later found a different approach to his problem, and so the example is no longer relevant from that viewpoint. He had hoped that it could be proven automatic, but the methods we have been discussing in this paper only appear to be sufficient to prove it asynchronously automatic.

The computer programs could make no progress with the presentation as given above, but matters improved after manipulating it a little. Eliminating $t = bxbx$, we get

$$\langle\, a, b, r, x, z \mid xaxa = bxbx, bbxbxaa = xbx, \\ a^{-1}br = ra^{-1}b, zbxbx = bxbxz, bbxbxaz = zbbxbxa \,\rangle.$$

Now, putting $u = xa$ and $v = bx$, and eliminating $a = x^{-1}u = v^{-1}bu$ and $x = b^{-1}v$, we get

$$\langle\, u, v, b, r, z \mid u^2 = v^2, bvbuv^{-1}bu = b^{-1}v^2, \\ u^{-1}b^{-1}vbr = ru^{-1}b^{-1}vb, zv^2 = v^2 z, bvbuz = zbvbu \,\rangle.$$

Finally, using $u^2 = v^2$ to simplify the second relation , we get

$$\langle\, u, v, b, r, z \mid u^2 = v^2, bvbu = b^{-1}ub^{-1}v, \\ u^{-1}b^{-1}vbr = ru^{-1}b^{-1}vb, zu^2 = u^2 z, zbvbu = bvbuz \,\rangle,$$

This is now visibly an HNN extension of the group

$$G = \langle\, u, v, b, r \mid u^2 = v^2, bvbu = b^{-1}ub^{-1}v, u^{-1}b^{-1}vbr = ru^{-1}b^{-1}vb \,\rangle.$$

with respect to the subgroup $H = \langle u^2, bvbu \rangle$, where H is centralised by the new generator z. (In fact G is itself an HNN extension with extra generator r, but we shall not make use of that fact.)

Running the automatic coset system program from kbmag on the subgroup H of G verifies that G is strongly coset automatic with respect to H. (The coset word acceptor has 302 states, and the coset multiplers about 1400 states.) The presentation of H computed by the program proves that H is free of rank 2, and so it is certainly automatic. The programs can also be used to verify that the set $Y = \{u^2, bvbu^{-1}\}$ is an efficient generating set for H. (Briefly, this is done by constructing the multiple initial state multiplier automata for the elements u^2 and $bvbu^{-1}$. The elements of H corresponding to the initial states of these automata can then be inspected from the output, and it turns out that these are just the identity and elements of $B = Y \cup Y^{-1}$.) We can now deduce from the theorem that Sapir's group K is asynchronously automatic.

As a final remark about this example, it turns out (again using calculations carried out by kbmag) that the subgroup H is not L–quasiconvex, where L is the language of the word-acceptor of the shortlex automatic structure of G. The element $(bub^{-1}v^{-1})^n(b^{-1}vbu^{-1})^n$ of L lies in H for all $n \geq 0$, but the coset representative of $(bub^{-1}v^{-1})^n$ in the language of the coset word acceptor is b^{2n}.

References

[1] **G Baumslag**, **S M Gersten**, **M Shapiro**, **H Short**, *Automatic groups and amalgams*, J. of Pure and Applied Algebra 76 (1991) 229–316

[2] **J W Cannon**, *The combinatorial structure of cocompact discrete hyperbolic groups*, Geom. Dedicata 16 (1984) 123–148

[3] **David B A Epstein**, **J W Cannon**, **D F Holt**, **S Levy**, **M S Patterson**, **W Thurston**, *Word Processing in Groups*, Jones and Bartlett (1992)

[4] **D B A Epstein**, **D F Holt**, **S E Rees**, *The use of Knuth–Bendix methods to solve the word problem in automatic groups*, J. Symbolic Computation 12 (1991) 397–414

[5] **D B A Epstein**, **A R Fletcher**, **U Zwick**, *Growth functions and automatic groups*, Experimental Mathematics 5 (1996) 297–315

[6] **S M Gersten**, **H B Short**, *Rational Subgroups Of Biautomatic Groups*, Annals of Math. 134 (1991) 125–158

[7] **R H Gilman**, '*Presentations of groups and monoids*, J. Algebra 57 (1979) 544–554

[8] **Derek F Holt**, *The Warwick automatic groups software*, from: "Geometrical and Computational Perspectives on Infinite Groups", Gilbert Baumslag et al (editors), DIMACS Series in Discrete Mathematics and Theoretical Computer Science 25 (1995) 69–82

[9] **Derek F Holt**, **Darren F Hurt**, *Computing automatic coset systems and subgroup presentations*, submitted to J. Symbolic Computation

[10] **John E Hopcroft**, **Jeffrey D Ullman**, *Introduction to automata theory, languages and computation*, Addison–Wesley (1979)

[11] **Darren F Hurt**, *The Use of Knuth–Bendix Methods and Automatic Coset Systems for Solving the Generalized Word Problem and Finding Subgroup Presentations*, PhD Thesis, University of Warwick (1996)

[12] **Ilya Kapovich**, *Detecting quasiconvexity: algorithmic aspects*, from: "Geometric and Computational Perspectives on Infinite Groups", Gilbert Baumslag et al (editors), DIMACS Series in Discrete Mathematics and Theoretical Computer Science 25 (1995) 91–99

[13] **R C Lyndon**, **P E Shupp**, *Combinatorial Group Theory*, Springer–Verlag, Berlin–Heidelberg–New York (1977)

[14] **G McShane**, **J Parker**, **I Redfern**, *Drawing Limit Sets of Kleinian Groups Using Finite State Automata*, Experimental Mathematics 3 (1994) 153–172

[15] **I D Redfern**, *Automatic Coset Systems*, PhD Thesis, University of Warwick (1993)

[16] **Sarah Rees**, *Hairdressing in groups: a survey of combings and formal languages*, these proceedings

[17] **M Shapiro**, *Graph products of automatic groups*, unpublished

Mathematics Institute, University of Warwick
Coventry, CV4 7AL, UK.

Email: `dfh@maths.warwick.ac.uk`

Received: 26 August 1997