Publ. RIMS, Kyoto Univ.
28 (1992), 913–942

The Word Problem for Groups with Regular Relations —Improvement of the Knuth-Bendix Algorithm—

By

Chuya HAYASHI*

Abstract

The Knuth-Bendix algorithm is a practical algorithm which, for a given finite presentation of a group, finds a finite confluent set of relations, if it terminates. From this confluent set we can know a solution of the word problem for the group. In this paper we introduce a concept of a regular confluent set of relations, which also gives a solution of the word problem, and represent a class of groups with such sets in terms of the Cayley graphs of the groups. Furthermore we make an algorithm to find a regular confluent set of relations as an improvement of the Knuth-Bendix algorithm. This new algorithm is a genuine improvement because there are some finite presentation, for which the Knuth-Bendix algorithm does not stop but the new one does.

§1. Introduction

In general a finitely presented group does not have a solvable word problem, that is, for a (and hence any) generating set, it does not have an *effective* algorithm (or algorithm which necessarily terminates and has the answer as its output) to determine whether or not a given word of the generators represents the identity element in the group. (Such an algorithm is called a *solution of the word problem.*) So there exists no effective algorithm that, for a given finite presentation for a group, finds a solution of the word problem. But there are such *non-effective* but *practical* algorithms, where a non-effective algorithm means an algorithm which does not necessarily terminate but has the answer as its output if it terminates.

The *Knuth-Bendix* algorithm is one of them. It finds a certain kind, called *confluent*, of finite set of relations from a given finite presentation, if it

Communicated by K. Saito, March 18, 1991. Revised March 23, 1992.
1991 Mathematics Subject Classification: 20F10, 68Q42
* Mitsubishi Electric Corporation Asic Design Engineering Center, Hyogo 664, Japan.

terminates. The finite confluent set of relations gives a solution of the word problem. When we obtain a finite confluent set, we can in an effectively algorithmic (and practical) way know whether or not given two words of the generators represent the same element in the group.

Of course some groups with finite ordered sets of generators do not have finite confluent sets of relations. Nevertheless some of them have *regular* confluent sets, where "regular" means "accepted by a finite state automaton" (see Section 4). For example it is shown in Section 5 that hyperbolic groups in Gromov's sense ([Gr]) (or negatively curved groups) have regular confluent sets. So in this paper we develop a non-effective algorithm denoted by Λ which can find a regular confluent set from a given finite presentation. When we obtain a regular confluent set, we have almost the same advantage as a finite one, namely the word problem is solvable. However it is not proved that this algorithm Λ necessarily terminates when the considered group with the finite ordered set of generators has a regular confluent set of relations.

The contents are as follows. We define a confluent set of relations in Section 2. We describe the Knuth-Bendix algorithm in Section 3 and finite state automata in Section 4. In Section 5 we define regular confluent sets, and show their properties and their characterization. We describe the algorithm Λ in Sections 6–8. We show in Section 6 that a group defined by a finite set of generators and a regular set of relations is finitely presented. Further we mention some examples in Section 10. (If one is not interested in the Knuth-Bendix algorithm nor the algorithm Λ, one may skip Sections 3 and 7–9.)

Remark. For finite groups, though the Knuth-Bendix algorithm necessarily terminates, the Todd-Coxeter coset enumeration is almost always a superior algorithm. Further there is another concept of automatic groups, of which the word problem is practicaly solvable ([CEHPT]). Incidentally it is proved that geodesic automatic groups have regular confluent sets of relations (see Theorem 5.5).

§2. Confluent Sets of Relations

Let A be a finite set, A^* stand for the free monoid (semigroup with unit) generated by A and ϵ be the identity element of A^*. An element of A^* is often called a *word*. For a subset R of $A^* \times A^*$, let $\langle R \rangle$ be the congruence generated by R (with respect to A), that is, the minimal equivalence relation in A^* which contains

$$\{(p, q) \mid p=sut,\ q=svt \text{ for some } s,\ t\in A^* \text{ and } (u, v)\in R\}.$$

Then $A^*/\langle R\rangle$ is a monoid. We say that A is a set of (semigroup) generators and R is a set of defining relations for the monoid.

For instance if G is a group presented by

$$\langle x, y, \cdots \mid s=t,\ u=v,\ \cdots\rangle,$$

then by setting $A=\{x, x^{-1}, y, y^{-1},\cdots\}$ and

$$R=\{(xx^{-1}, \epsilon), (x^{-1}x, \epsilon), (yy^{-1}, \epsilon), (y^{-1}y, \epsilon), \cdots, (s, t), (u, v),\cdots\},$$

$A^*/\langle R\rangle$ is isomorphic to G by the obvious correspondance. We do not assume that $A^*/\langle R\rangle$ is a group in Sections 2–4, but we shall assume it from Section 5 on.

We write $<$ and $\leq$ for a fixed well-ordering of A^* with the property that

(1) For all $s,\ u\in A^*$, $s\leq su$ and $s\leq us$.
(2) For all $s,\ u,\ v\in A^*$, $u<v$ implies $su<sv$ and $us<vs$.

(In fact (1) follows from (2) and the property of well-ordering.) The following two orderings have this property.

Example 2.1 (lexicographical ordering). First fix any total ordering of A. For all $u,\ v\in A^*$, we write $u<v$ if u is a shorter word than v or if u and v have the same length and u is less than v lexicographicaly.

Example 2.2. Let $A=\{x, X, y, Y, c, C\}$ and define $x<X<y<Y<c<C$. For all $u,\ v\in A^*$, we define $u<v$ if with respect to the ordering of Example 2.1 (lexicographical ordering) u° is less than v° or ($u^\circ=v^\circ$ and u is less than v), where u°, v° denote elements of A^* obtained by replacing every appearance of c or C in u, v (resp.) by ϵ.

We will deal only with the ordering of Example 2.1 (lexicographical ordering) from Section 5 on. Now we fix, A, R and $<$.

Let S be a subset of $A^*\times A^*$. (We are interested in the case when S is a subset of $\langle R\rangle$.) For all $p,\ q\in A^*$ we write $p\rightarrow_S q$ if $p=sut$, $q=svt$ for some $s,\ t\in A^*$ and $(u, v)\in S$. We introduce further relations in A^*. We define $\rightarrow_S^*$ as the reflexive transitive closure of $\rightarrow_S$, and $\leftrightarrow_S^*$ as the reflexive symmetric transitive closure of $\rightarrow_S$. We write $p \cup_S q$ when there exists a sequence of elements of A^* $p=u_0, \cdots, u_n=q$ such that $u_i<\max(p, q)$ for all $0<i<n$ and ($u_i \rightarrow_S u_{i+1}$ or $u_{i+1} \rightarrow_S u_i$) for all $0 \leq i<n$. In particular $p=q$ implies $p \cup_S q$. Notice that $\leftrightarrow_S^*$ is the same as $\langle S \rangle$. If given $S, S' \subset A^* \times A^*$ satisfy that $(p, q) \in S$ implies $p \cup_{S'} q$, then we obtain that $p \cup_S q$ implies $p \cup_{S'} q$. We use $\rightarrow_S^*$ also for two subsets P, Q of $A^* \times A^*$, that is, we define $P \rightarrow_S^* Q$ if for any $(p_1, p_2) \in P$ there exists $(q_1, q_2) \in Q$ such that $p_1 \rightarrow_S^* q_1$ and $p_2 \rightarrow_S^* q_2$. We say that S is *normalized* if for all $(u, v) \in S$ $u>v$. Then $p \rightarrow_S q$ implies $p>q$, and $p \rightarrow_S^* q$ implies $p \geq q$. We say that $u \in A^*$ is *S-irreducible* if there does not exist $t \in A^*$ such that $u \rightarrow_S t$ and $u>t$. Notice that for all $p \in A^*$ there exists an S-irreducible word q such that $p \rightarrow_S^* q$ since $<$ is a well-ordering. $\langle R \rangle$-irreducible means minimal in an equivalence class with respect to $\langle R \rangle$. From now on we assume that S is normalized. We say that S is *confluent* if for any $r \in A^*$ there exists a *unique* S-irreducible word $t \in A^*$ such that $r \rightarrow_S^* t$. We define $D_{A^*}=\{(w,w) | w \in A^*\}$. The following proposition is easily shown.

Proposition 2.3. *Let S be a normalized subset of $\langle R \rangle$. Then the following are equivalent.*

(1) *$\langle S \rangle = \langle R \rangle$ and S is confluent.*
(2) *Every equivalence class with respect to $\langle R \rangle$ has a unique S-irreducible word.*
(3) *All S-irreducible words are $\langle R \rangle$-irreducible.*
(4) *$\langle R \rangle \rightarrow_S^* D_{A^*}$.*

We say that S is *R-confluent* if it satisfies one of the above equivalent conditions. Clearly, S is S-confluent if and only if S is confluent. The set $\{(u, v) \in \langle R \rangle | u>v\}$ is R-confluent.

Assume that a *finite* normalized subset S of $A^* \times A^*$ is given. Let r be any element of A^*. We can constructively find a sequence $r=r_0 \rightarrow_S \cdots \rightarrow_S r_n$ (where r_n is S-irreducible) as follows. If r_0 is not S-irreducible, then we can find $s, t, u, v \in A^*$ such that $r_0=sut$ and $(u, v) \in S$, and we know $r_0 \rightarrow_S r_1=svt$. We can repeat this procedure until we obtain the above sequence. If S is R-confluent, r_n is $\langle R \rangle$-irreducible and $(r, r_n) \in \langle R \rangle$. In particular,

Proposition 2.4 ([Gi]). *If (A, R) has a finite R-confluent set, then $A^*/\langle R \rangle$ has a solvable word problem.*

Definition 2.5. We define
$J(R)=\{r\in A^*|r$ is not $\langle R\rangle$-irreducible, but its any proper subword is $\langle R\rangle$-irreducible$\}$ and for $S\subset A^*\times A^*$ we define $pr_1(S)=\{u\in A^*|(u, v)\in S$ for some $v\in A^*\}$.

Proposition 2.6. *Let S be a normalized subset of $\langle R\rangle$. Then the following are equivalent.*

(1) *S is R-confluent.*
(2) $pr_1(S)\supset J(R)$.

Proof.

(1) implies (2) Assume that S is R-confluent. Let r be any element of $J(R)$. Since r is not $\langle R\rangle$-irreducible, r is not S-irreducible. There exist $p, q\in A^*$ and $(u, v)\in S$ such that $r=puq$. But any proper subword of r is $\langle R\rangle$-irreducible and u is not, thus $r=u$ and we obtain $(r, v)\in S$.

(2) implies (1) Assume that $pr_1(S)\supset J(R)$ and let r be not $\langle R\rangle$-irreducible element of A^*. We will show that r is not S-irreducible to show the condition (3) in Proposition 2.3. There exists a subword u of r such that $u\in J(R)$. (The reason is that there exist r', $r''\in A^*$ and $x\in A$ such that $r=r'xr''$, r' is $\langle R\rangle$-irreducible and $r'x$ is not $\langle R\rangle$-irreducible, further there exist r''', $r''''\in A^*$ and $x'\in A$ such that $r'x=r'''x'r''''$, r'''' is $\langle R\rangle$-irreducible and $x'r''''$ is not $\langle R\rangle$-irreducible, thus we can take $u=x'r''''$. Notice that a subword of an $\langle R\rangle$-irreducible word is also $\langle R\rangle$-irreducible.) Thus $u\in pr_1(S)$ and r is not S-irreducible. □

Example 2.7. Consider the free abelian group of rank 2

$$\langle x, y|[x, y]=1\rangle \quad (\text{where } [a, b]=aba^{-1}b^{-1} \text{ for all } a, b).$$

Set $A=\{x, X, y, Y\}$ (we use X, Y instead of x^{-1}, y^{-1}) and

$$R=\{(xX, \epsilon), (Xx, \epsilon), (yY, \epsilon), (Yy, \epsilon), (xyXY, \epsilon)\}.$$

2.7.1 We define $<$ as in Example 2.1 (lexicographical ordering) for $x<X<y<Y$. Then the set of all $\langle R\rangle$-irreducible words is

$$\bigcup_{n,m=0,1,2,\dots} \{x^n y^m, X^n y^m, x^n Y^m, X^n Y^m\}$$

and $J(R)=\{xX, Xx, yY, Yy, yx, yX, Yx, YX\}$. Thus one of the R-confluent sets is

$$\{(xX, \epsilon), (Xx, \epsilon), (yY, \epsilon), (Yy, \epsilon), (yx, xy), (yX, Xy), (Yx, xY), (YX, XY)\}.$$

2.7.2 We define $<$ as in Example 2.1 (lexicographical ordering) for $x<y<X<Y$. Then one of the R-confluent sets is

$$\{(xX, \epsilon), (Xx, \epsilon), (yY, \epsilon), (Yy, \epsilon), (yx, xy), (Xy, yX), (Yx, xY), (YX, XY)\} \cup \bigcup_{n=1}^{\infty} \{(xy^nX, y^n), (yX^nY, X^n)\}.$$

Example 2.8. Consider the nilpotent group

$$\langle x, y, c \mid [x, y]=c, [x, c]=1, [y, c]=1\rangle.$$

Set $A=\{x, X, y, Y, c, C\}$ and define $<$ as in Example 2.2. Set

$$R=\{(xX, \epsilon), \cdots, (Cc, \epsilon), (xyXY, c), (xcXC, \epsilon), (ycYC, \epsilon)\}.$$

Then one of the R-confluent sets is

$$\{(xX, \epsilon), \cdots, (Cc, \epsilon), (yx, xyC), (Yx, xYc), (yX, Xyc), (YX, XYC), (cx, xc), (cX, Xc), (Cx, xC), (CX, XC), (cy, yc), (cY, Yc), (Cy, yC), (CY, YC)\}.$$

§3. The Knuth-Bendix Algorithm

Definition 3.1. For any normalized subset S of $A^* \times A^*$, we define (to state the following lemma)

$$K(S)=\{(p, q) \mid \exists w \in A^* \text{ such that } (w, p) \in S \text{ and } w \to_S q\} \cup \{(p, q) \mid \exists v, v', s, t, w \in A^* \text{ such that } (st, v), (tw, v') \in S, t \neq \varepsilon, \text{ and } p=vw, q=sv' \text{ (so } stw \to_S p, q)\}.$$

If S is finite, then $K(S)$ is also finite. For any $(p, q) \in K(S)$, there exists $w \in A^*$ such that $w \to_S p, q$. In particular $K(S) \subset \langle S\rangle$.

Lemma 3.2. *Let S be a normalized subset of $A^* \times A^*$. Then the following are equivalent.*

(1) *S is confluent.*

(2) $K(S) \to_S^* D_{A^*}$. ((1)⇔(2) *is called the Knuth-Bendix lemma.*)

(3) *For all $(p, q) \in K(S)$ $p \cup_S q$.*

Proof. Clearly (1)⇒(2)⇒(3).

(3) implies (1) We assume that the condition (3) holds and show the condition (1) by induction on the well-ordering of A^*. Let r be an element of A^* and suppose that for any $r' \in A^*$ less than r there exists a unique S-irreducible word $t' \in A^*$ such that $r' \to_S^* t'$. Let t_1 and t_2 be S-irreducible words such that $r \to_S^* t_1$ and $r \to_S^* t_2$. We must show $t_1 = t_2$.

We can assume that r is not S-irreducible since if r is S-irreducible, we obtain $t_1 = t_2$ $(=r)$ clearly. Let u_1, u_2 be elements of A^* such that $r \to_S u_1 \to_S^* t_1$ and $r \to_S u_2 \to_S^* t_2$. If $u_1 = u_2$, then $t_1 = t_2$ by the induction hypothesis. So we assume that $u_1 \neq u_2$.

Claim. $u_1 \cup_S u_2$.

Proof of Claim. There exist (p_1, q_1) and $(p_2, q_2) \in S$ such that

$$r \to_{\{(p_1, q_1)\}} u_1 \quad \text{and} \quad r \to_{\{(p_2, q_2)\}} u_2.$$

We have three cases.

(a) p_1 and p_2 are disjoint subwords of r. Then $u_1 \to_{\{(p_2, q_2)\}} v$ and $u_2 \to_{\{(p_1, q_1)\}} v$ for some $v \in A^*$. Thus $u_1 \cup_S u_2$.

(b) One of p_1 and p_2 is a subword of the other in r. Without loss of generality we can assume that p_2 is a subword of p_1. Then $p_1 = sp_2t$ for some s, $t \in A^*$. Since $(p_1, q_1) \in S$ and $p_1 \to_S sq_2t$, we obtain $(q_1, sq_2t) \in K(S)$ (or rather (q_1, sq_2t) is contained in the left operand of $\cup$ in the definition of $K(S)$). By the condition (3), $q_1 \cup_S sq_2t$ and $u_1 \cup_S u_2$.

(c) p_1 and p_2 overlap. Without loss of generality we can assume that $p_1 = st$ and $p_2 = tw$ for some s, t, $w \in A^*$ such that $t \neq \epsilon$. Since $(st, q_1) \in S$ and $(tw, q_2) \in S$, we obtain $(q_1w, sq_2) \in K(S)$. Hence $q_1w \cup_S sq_2$ and so $u_1 \cup_S u_2$.

We conclude the proof of Claim and continue the proof of the lemma.

Since $u_1 \cup_S u_2$, $u_1 \to_S^* t_1$ and $u_2 \to_S^* t_2$, there exists a sequence of elements of A^* at most $\max(u_1, u_2)$ (so, less than r)

$$t_1 = z_0,\ z_0',\ z_1,\ z_1',\ \cdots,\ z_m',\ z_{m+1} = t_2$$

such that $z_i' \to_S^* z_i$ and $z_i' \to_S^* z_{i+1}$ for all $0 \le i \le m$. We can replace z_i by an S-irreducible word $z_i^\diamond$ satisfying $z_i \to_S^* z_i^\diamond$, so that each z_i is S-irreducible. By the induction hypothesis, we obtain $z_i = z_{i+1}$ for all $0 \le i \le m$. Thus $t_1 = t_2$, and we conclude the proof of the lemma. □

Let a *finite* subset R of $A^* \times A^*$ be given explicitly. Now we define (non-effective) algorithm called the *Knuth-Bendix algorithm* to produce a finite R-confluent set using the Knuth-Bendix Lemma in Lemma 3.2.

We recursively define a sequence of finite normalized subsets of $A^* \times A^*$ R_0, R_1, $\cdots$ such that $\langle R_n \rangle = \langle R \rangle$ and $K(R_{n-1}) \to_{R_n}^* D_{A^*}$ for all $n \ge 0$, where we define $K(R_{-1}) = \varnothing$. We define $R_0 = \{(u, v) | u > v$ and $((u, v)$ or $(v, u) \in R)\}$. Clearly $\langle R_0 \rangle = \langle R \rangle$. Assume that R_{n-1} and R_n are defined so that $\langle R_n \rangle = \langle R \rangle$ and $K(R_{n-1}) \to_{R_n}^* D_{A^*}$. First produce $K(R_n)$ (recall that $K(R_n) \subset \langle R_n \rangle$). Furthermore for each $(u, v) \in K(R_n) \setminus K(R_{n-1})$ find a pair (u', v') of R_n-irreducible words such that $u \to_{R_n}^* u'$ and $v \to_{R_n}^* v'$ (see the statement previous to Theorem 2.4, if necessary). Let $L_n \subset A^* \times A^*$ be the set of all such (u', v'), then $K(R_n) \setminus K(R_{n-1}) \to_{R_n}^* L_n$ and $L_n \subset \langle R_n \rangle$. If $L_n \subset D_{A^*}$, then $K(R_n) \setminus K(R_{n-1}) \to_{R_n}^* D_{A^*}$ and so $K(R_n) \to_{R_n}^* D_{A^*}$. Then we obtain a finite R-confluent set R_n and this algorithm terminates.

If $L_n \not\subset D_{A^*}$, we define

$$R_{n+1} = R_n \cup \{(p, q) | p > q \quad \text{and} \quad (p, q) \text{ or } (q, p) \in L_n\}.$$

Then $K(R_n) \setminus K(R_{n-1}) \to_{R_{n+1}}^* D_{A^*}$ and so $K(R_n) \to_{R_{n+1}}^* D_{A^*}$, and $\langle R_{n+1} \rangle = \langle R_n \rangle = \langle R \rangle$. We conclude the definition of this algorithm.

By (1)$\Leftrightarrow$(3) in Lemma 3.2, it is easily shown that we are allowed to change the definition of R_{n+1} (and R_{n+1} itself) to

$$R_{n+1} = \hat{C}(R_n) \cup \{(p, q) | p > q \quad \text{and} \quad (p, q) \text{ or } (q, p) \in L_n\}$$

where

$$\hat{C}(S) \overset{\text{def}}{=} S \setminus \{(r, p) \in S \,|\, \text{some proper subword of } r \text{ is contained in } pr_1(S)\}.$$

In this case we have to assume that for all $(u, v) \in K(R_{n-1})$ $u \cup_{R_n} v$ instead of $K(R_{n-1}) \rightarrow^*_{R_n} D_{A^*}$. It is almost always better to use this definition in terms of computer time and memory because each R_n contains fewer elements.

This algorithm succeeds for Example 2.7.1 and Example 2.8. More generally it is shown that if $J(R)$ is finite, this algorithm terminates ([Gi]). But in the case of Example 2.7.2 it does not terminate, since $J(R)$ is infinite and so there exists no finite R-confluent set. One of our main purpose is to introduce an algorithm to find an R-confluent set also in such a case. The algorithm Λ in Section 8 is to produce such an R-confluent set with a certain regular property.

§4. Finite State Automata and Regular Languages

In this section we follow [CEHPT]. We define a finite state automaton and a regular language. In addition we show some of their properties.

Let B be a finite set. We call a subset of B^* a *language* over B.

Definition 4.1 (non-deterministic finite state automaton). A *non-deterministic finite state automaton* over B is a finite labelled directed graph (not necessarily connected), with two set S_0 and Y of distinguished vertices, where a label is an element of B or the identity element ϵ of B^*. A vertex of the graph is called a *state*. A vertex in S_0 is called an *initial state*, and a vertex in Y is called an *accept state*. A directed edge is called an *arrow*.

Let M be a non-deterministic finite state automaton over B. We say that an element of B^* is *accepted* by M if the element is represented by some *path of arrows* from an initial state to an accept state (where a path of arrows means a finite sequence of arrows such that the target of each arrow of the sequence except the last one is the source of the next arrow). The language over B consisting all elements accepted by M, is called the language *accepted* by M and denoted by $L(M)$. We say that a state of M is *dead* (resp. *inaccessible*) if there does not exist a path of arrows from the state to an accept state (resp. from an initial state to the state). We can make a new non-deterministic finite state automaton accepting $L(M)$ by removing from M all dead states, all inaccessible states and all arrows with such states as their sources or their targets.

Definition 4.2 (finite state automaton). We say that M is a *finite* *state automaton* over B if M is connected non-deterministic finite state automaton with the just one initial state and without the label ϵ.

Notice that this definition is exceptional. It is usual that a finite state automaton means a deterministic finite state automaton defined below. Anyway a finite state automaton is a special case of a non-deterministic finite state automaton.

Definition 4.3 (deterministic finite state automaton). We say that M is a *deterministic finite state automaton* if M is a finite state automaton with the property that any pair of distinct arrows with the same state as their source does not have the same label.

The following theorem means that languages accepted by the above three kind of automata are equivalent (the proof is omitted).

Theorem 4.4 ([RS]). *If M_1 is a non-deterministic finite state automaton, then there exists a deterministic finite state automaton M_2 such that $L(M_2)=L(M_1)$, and there exists an explicit construction of M_2 from M_1.*

We say that a language over B is *regular* (over B) if the language is accepted by a (non-deterministic, deterministic) finite state automaton over B. We identify $(B\times B)^*$ with a subset of $B^*\times B^*$. For example D_{B^*} is a regular language over $B\times B$. The proof of the following theorem is also omitted.

Theorem 4.5. *Let B be a finite set. Let L_1 and L_2 be regular languages over B. Then the following languages are regular over B.*

(1) $L_1L_2 \stackrel{\text{def}}{=} \{l_1l_2 \mid l_1 \in L_1,\ l_2 \in L_2\}$

(2) $L_1 \cap L_2$

(3) $L_1 \cup L_2$

(4) $L_1 \setminus L_2$.

The following is a regular language over $B\times B$.

(5) $(L_1 \times L_2) \cap (B\times B)^*$.

Further, if L is a regular language over $B\times B$, then

(6) $pr_1(L)$

is a regular language over B.

If we are given finite state automata accepting L_1, L_2 and L, we can construct finite state automata accepting the above regular languages (1)–(6).

The following lemma is necessary to define the algorithm A.

Lemma 4.6. *Let S be a regular language over $B\times B$ such that $(\epsilon,\epsilon)\notin S$. Then the following are regular languages over $B\times B$ and we can construct finite state automata accepting them from a finite state automaton accepting S.*

(1) $K(S)=\{(p, q)\mid \exists w\in B^* \text{ such that } (w, p)\in S \text{ and } w\rightarrow_S q\}$

$\cup\{(p, q)\mid \exists v, v', s, t, w\in B^* \text{ such that } (st, v), (tw, v')\in S,\ t\neq\epsilon, \text{ and } p=vw,\ q=sv' \text{ (so } stw\rightarrow_S p, q)\}$

(2) $\hat{C}(S)=S\setminus\{(r, p)\in S\mid \text{some proper subword of } r \text{ is contained in } pr_1(S)\}$.

Proof. (1) Notice that

$$K(S)\subset K' \overset{\text{def}}{=} \{(p, q)\mid \exists\, w\in B^* \text{ such that } (w, p)\in S\, D_{B^*} \text{ and } (w, q)\in D_{B^*}S\, D_{B^*}\}.$$

Let M be a finite state automaton over $B\times B$ accepting S. Since M does not accept (ϵ, ϵ), the initial state of M is not an accept state. We can assume that no arrow of M has an accept state as its source by adding one new accept state to M (the old accept states are no longer accept states) and adding to M an arrow with a label (x, y) from a state s to the new state if there exists an arrow labelled (x, y) from s to an old accept state. Further we can similarly assume that M has no arrow with the initial state as its target. We can construct a finite state automaton M_1 over $B\times B$ accepting $S\, D_{B^*}$ by adding an arrow labelled (x, x) from the accept state to itself for each $x\in B$. Similarly construct a finite state automaton M_2 accepting $D_{B^*}S\, D_{B^*}$.

We construct a finite state automaton T over $B\times B$ accepting K' as follows. A state of T is a pair of a state of M_1 and a state of M_2. The initial (resp. accept) state of T is the pair of the initial (accept) state of M_1 and the initial (accept) state of M_2. T has an arrow labelled (x, y) from a state (s_1, s_2) to a state (s_1', s_2') if and only if for some $z\in B$ M_1 has the arrow labelled (z, x) from the state s_1 to the state s_1' and M_2 has the arrow labelled (z, y) from the state s_2 to the state s_2'.

Finally we obtain the required finite state automaton over $B\times B$ accepting $K(S)$ by removing from T every arrow with the accept state of T as its source (so its target) and removing the state the pair of the accept state of M_1

and the initial state of M_2 (and every arrow with the state as its source or its target).

(2) Notice that

$$\{(r, p)\in S \mid \text{some proper subword of } r \text{ is contained in } pr_1(S)\}$$

is the union of the two language

$$\{(r, p)\in S \mid \exists q\in B^* \text{ such that } (r, q)\in (D_{B^*}\setminus\{(\epsilon, \epsilon)\})S\ D_{B^*}\}$$

and

$$\{(r, p)\in S \mid \exists q\in B^* \text{ such that } (r, q)\in S(D_{B^*}\setminus\{(\epsilon, \epsilon)\})\}.$$

Similarly as the proof of (1), we can construct two finite state automata T_1, T_2 over $B\times B$ accepting these two languages. From the above theorem we can construct the required finite state automaton accepting $S\setminus(L(T_1)\cup L(T_2))$. □

§5. Groups with Regular Confluent Sets of Relations

We can assume that R is normalized by throwing away $(u, v)\in R$ if $u=v$ and interchanging u and v if $u<v$. From now on we assume that $A^*/\langle R\rangle$ is a group. We define $\tilde{A}=A\cup\{e\}$ where e is a new element. (In the result the element e works similarly as the padded string \$. Nevertheless we do not use \$ but e because \$ has the ristriction that it cannot appear before any element of A.) We fix $<$ as in Example 2.1 (lexicographical ordering) assuming that $e<x$ for all $x\in A$. Let Lex be the subset of $(\tilde{A}\times\tilde{A})^*$ consisting of every element (p, q) such that $p>q$. Then Lex is a regular language over $\tilde{A}\times\tilde{A}$. We identify $A^*\times A^*$ with a subset of $\tilde{A}^*\times\tilde{A}^*$. We define

$$\tilde{R}=R\cup\{(e, \epsilon)\}.$$

Clearly $A^*/\langle R\rangle$ is isomorphic to $\tilde{A}^*/\langle\tilde{R}\rangle$ by the obvious correspondance. ($\langle R\rangle$ and $J(R)$ are still defined over A, not $\tilde{A}$. So $\langle R\rangle\subset A^*\times A^*$ and $J(R)\subset A^*$.) Further an element r of A^* is $\langle R\rangle$-irreducible if and only if r is $\langle\tilde{R}\rangle$-irreducible. Any element of $\tilde{A}^*\setminus A^*$ is not $\langle\tilde{R}\rangle$-irreducible. In fact $\langle\tilde{R}\rangle$ consists of all $(u, v)\in\tilde{A}^*\times\tilde{A}^*$ such that $(u^\circ, v^\circ)\in\langle R\rangle$, where u°, $v^\circ$are the elements obtained by replacing every appearance of e in u, v (resp.) by ϵ. It is easily shown that $J(\tilde{R})=J(R)\cup\{e\}$. So instead of looking for R-confluent sets (over A), we shall look for $\tilde{R}$-confluent sets (over $\tilde{A}$). We define

$$\tilde{R}'=\{(p,\ q)\in(A\times\tilde{A})^*\mid q=e^n q' \text{ and } (p,\ q')\in R \text{ for some } q'\in A^*,\ n\geq 0\}$$

$$\cup\bigcup_{x\in A}\{(xe,\ ex)\}$$

$$\subset\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*.$$

The following two lemmas are necessary to define (and show correctness of) the algorithm Λ in Section 8.

Lemma 5.1. *Let S be a normalized confluent subset of $(\tilde{A}\times\tilde{A})^*$ such that $\langle\tilde{R}'\rangle\subset\langle S\rangle\subset\langle\tilde{R}\rangle$. Suppose that $(u,\ v)\in(\tilde{A}\times\tilde{A})^*$ and $(eu,\ ev)\in S$ implies $(u,\ v)\in S$. Then S is $\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*$-confluent.*

Proof. Notice that $\langle\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*\rangle=\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*$. By the hypothesis it holds that $\langle S\rangle\subset\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*$. We must show that $\langle S\rangle\supset\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*$. Let $(p,\ q)$ be an element of $\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*$. First we show that there exists $n\geq 0$ such that $(e^n p,\ e^n q)\in\langle\tilde{R}'\rangle\subset\langle S\rangle$. If $(z,w)\in\tilde{A}^*\times\tilde{A}^*$ and $z\rightarrow_{\{(e,\epsilon)\}} w$, then $z\rightarrow^*_{\tilde{R}'} ew$ since $\tilde{R}'$ contains $(xe,\ ex)$ for each $x\in A$. So $z\rightarrow_{\tilde{R}} w$ implies $z\rightarrow^*_{\tilde{R}'} e^{n'}w$ for some $n'\geq 0$. Thus we obtain $(e^n p,\ e^n q)\in\langle\tilde{R}'\rangle$for some $n\geq 0$. Let p', q' be S-irreducible elements such that $p\rightarrow^*_S p'$, $q\rightarrow^*_S q'$. Because $(u,\ v)\in(\tilde{A}\times\tilde{A})^*$ and $(eu,\ ev)\in S$ implies $(u,\ v)\in S$, $e^n p'$ and $e^n q'$ are S-irreducible. We obtain $e^n p'=e^n q'$ and $p'=q'$ since S is (S)-confluent, $e^n p\rightarrow^*_S e^n p'$ and $e^n q\rightarrow^*_S e^n q'$. So $(p,\ q)\in\langle S\rangle$ and we obtain $\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*\subset\langle S\rangle$. Thus $\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*=\langle S\rangle$. Since S is confluent, S is $\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*$-confluent. □

Lemma 5.2. *Let S be a normalized $\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*$-confluent subset of $(\tilde{A}\times\tilde{A})^*$. Then $S\cup\{(e,\ \varepsilon)\}$ is $\tilde{R}$-confluent.*

Proof. For $r\in A^*$, r is $\langle R\rangle$-irreducible if and only if r is $\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*$-irreducible. Furthermore $pr_1\ (S)\supset J(\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*)\supset J(R)$.
So $pr_1(S\cup\{(e,\ \varepsilon)\})\supset J(\tilde{R})$. Since $S\cup\{(e,\ \varepsilon)\}\subset\langle\tilde{R}\rangle$, $S\cup\{(e,\ \varepsilon)\}$ is $\tilde{R}$-confluent. □

Definition 5.3. We say that a normalized subset S of $(\tilde{A}\times\tilde{A})^*$ is $\tilde{R}$*-semiconfluent* if $S\cup\{(e,\epsilon)\}$ is $\tilde{R}$-confluent.

Clearly $pr_1(S)\supset J(R)$ if and only if S is $\tilde{R}$-semiconfluent. If S is a regular $\tilde{R}$-semiconfluent set over $\tilde{A}\times\tilde{A}$, the set consisting of all $\langle R\rangle$-irreducible words $A^*\setminus(A^*pr_1(S)A^*)$ is regular over $\tilde{A}$ (so A). Now we concentrate on finding a regular $\tilde{R}$-semiconfluent set from R.

Definition 5.4. We call the following set the *minimal* $\tilde{R}$*-semiconfluent* set

$$\{(p,q)\in\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*\mid p\in J(R)\text{ and }q\text{ is }\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*\text{-irreducible}\}.$$

Notice that $q\in\tilde{A}^*$ is $\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*$-irreducible if and only if $q=e^nq'$ for some nonnegative integer n and for some $\langle R\rangle$-irreducible word $q'\in A^*$.

For each word $r=x_1\cdots x_n$ ($x_i\in\tilde{A}$), we define

$$r(i)=\begin{cases}x_1\cdots x_i & \text{for } 0\le i\le n\\ r & \text{for } i\ge n\end{cases}$$
$$\text{length}(r)=n.$$

Let G be the group $\tilde{A}^*/\langle\tilde{R}\rangle$. For each $w\in\tilde{A}^*$ let $\bar{w}$ be the element of G represented by w. From now on in this section except Theorem 5.9, we assume that A is closed under inversion. Let $\Gamma_{\tilde{A}}(G)$ stand for the *Cayley graph* of the group G with respect to the generating set $\tilde{A}$ (or rather the directed labelled graph such that the set of vertices is G and for each $x\in\tilde{A}$ and for each $g\in G$ there is a directed edge labelled x from the vertex g to the vertex $g\bar{x}$). Let d be the distance function in $\Gamma_{\tilde{A}}(G)$ where length of each edge is 1. $\Gamma_A(G)$ is considered as a subgraph of $\Gamma_{\tilde{A}}(G)$. We often consider an element w of $\tilde{A}^*$ as a path from the identity element to $\bar{w}$ in $\Gamma_{\tilde{A}}(G)$.

For a positive number k and a point q in $\Gamma_A(G)$ we call the following subset of $\Gamma_A(G)$ *the hausdorff k-neighbourhood of q*.

$$\{g\in\Gamma_A(G)\mid\exists i\ge 0\ d(g,\overline{q(i)})<k\}.$$

Theorem 5.5. *The following are equivalent.* (*Recall that A is closed under inversion.*)

(1) *The minimal $\tilde{R}$-semiconfluent set is regular over $\tilde{A}\times\tilde{A}$.*

(2) *There exists a constant $k>0$ with the following property. For any*

$(p,q)\in\langle R\rangle$ such that $p\in J(R)$ and q is $\langle R\rangle$-irreducible, p is in the hausdorff k-neighbourhood of q in the Cayley graph $\Gamma_A(G)$.

Proof.

(1) implies (2) Let M be a finite state automaton without a dead state, accepting the minimal $\tilde{R}$-semiconfluent set. For each state s we fix a path of arrows from the state s to an accept state, and let $(u_s, v_s)\in(\tilde{A}\times\tilde{A})^*$ be represented by this path. We take k' greater than the length of (u_s, v_s) in any state s. Let (p, q) be an element of $\langle R\rangle$ such that $p\in J(R)$ and q is $\langle R\rangle$-irreducible. By setting $n=\text{length}(p)-\text{length}(q)$ and $q'=e^n q$, we obtain $(p, q')\in L(M)$. We show that for all $0\le i\le\text{length}(p)$ $d(\overline{p(i)}, \overline{q'(i)})<2k'$. Since $(p, q')\in L(M)$, $(p(i), q'(i))$ is represented by a path of arrows from the initial state to a state s. Thus

$$\big(p(i)u_s,\ q'(i)v_s\big)\in L(M),\ \overline{p(i)u_s}=\overline{q'(i)v_s}\quad\text{and}\quad \overline{p(i)}^{\,-1}\,\overline{q'(i)}=\overline{u_s}\ \overline{v_s}^{\,-1}.$$

We obtain $d(\overline{p(i)}, \overline{q'(i)})=d(\overline{u_s},\overline{v_s})<2k'$. Hence p is in the hausdorff $2k'$-neighbourhood of q' in the Cayley graph $\Gamma_{\tilde{A}}(G)$ and so p is in the hausdorff $2k'$-neighbourhood of q in $\Gamma_A(G)$.

(2) implies (1) Let T be the minimal $\tilde{R}$-semiconfluent set. For each $x\in A$ we fix $z_x\in A$ such that $\overline{z_x x}=1$. We will show that for any $(p, q')\in T$

$$d(\overline{p(i)}, \overline{q'(i)})\le 2k+2\quad\text{for all } 0\le i\le\text{length}(p).$$

Since q' is $\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*$-irreducible, there exist $n\ge 0$ and $\langle R\rangle$-irreducible word $q\in A^*$ such that $q'=e^n q$. Then $(p, q)\in\langle R\rangle$. By the hypothesis p is in the hausdorff k-neighbourhood of q in $\Gamma_A(G)$. First we show $n\le 2$. Let $p=xw$ $(x\in A,\ w\in A^*)$. Then $(w, z_x q)\in\langle R\rangle$. Since $p\in J(R)$, w is $\langle R\rangle$-irreducible. Thus $w\le z_x q$ holds.

$$\begin{aligned} n&=\text{length}(p)-\text{length}(q)\\ &\le\text{length}(w)+1-\text{length}(q)+\big(1-\text{length}(z_x)\big)\\ &=\text{length}(w)-\text{length}(z_x q)+2\\ &\le 2. \end{aligned}$$

So we obtain $n\le 2$.

For $0\le i<\text{length}(p)$ there exists $0\le j\le\text{length}(q)$ such that $d(\overline{p(i)}, \overline{q(j)})\le k$. Since $p(i)$ and $q(j)$ are $\langle R\rangle$-irreducible, they are geodesics in $\Gamma_A(G)$ and so $|i-j|\le k$.

$$\begin{aligned} d(\overline{p(i)}, \overline{q(i)}) &\leq d(\overline{p(i)}, \overline{q(j)}) + d(\overline{q(j)}, \overline{q(i)}) \\ &\leq k+k \\ &= 2k \\ d(\overline{p(i)}, \overline{q'(i)}) &\leq d(\overline{p(i)}, \overline{q(i)}) + d(\overline{q(i)}, \overline{q'(i)}) \\ &\leq 2k+n \\ &\leq 2k+2. \end{aligned}$$

For $i=\text{length}(p)$, $d(\overline{p(i)}, \overline{q'(i)})=0$.

Now we construct a finite state automaton M over $\tilde{A} \times \tilde{A}$ accepting at least all elements of T. Let N be the subset of G consisting of elements with distance at most $2k+2$ from the identity element in $\Gamma_{\tilde{A}}(G)$. The state set of M is N. The initial and accept state is the identity element in G. M has an arrow labelled $(x, y) \in \tilde{A} \times \tilde{A}$ from a state $n_s \in N$ to a state $n_t \in N$ if and only if $\bar{x}^{-1} n_s \bar{y} = n_t$. Clearly $T \subset L(M) \subset \langle \tilde{R} \rangle$. $L(M) \cap Lex$ is regular. It contains T and so it is $\tilde{R}$-semiconfluent. Define

$$S = \hat{C}(L(M) \cap Lex \setminus (\tilde{A}^*\{e\}\tilde{A}^* \times \tilde{A}^*)).$$

Then S is regular and contains T. Further $pr_1(S) = J(R)$. Let $H = J(R) \cup \bigcup_{x \in A} \{xe\}$, then the required T is equal to $S \setminus (\tilde{A}^* \times \tilde{A}^* H \tilde{A}^*)$, and it is regular. □

Corollary 5.6. *Suppose that A and the set of all $\langle R \rangle$-irreducible words are a part of automatic structure ([CEHPT]) or in particular suppose that $G = A^*/\langle R \rangle$ is a hyperbolic group ([Gr]) (or negatively curved group). Then the minimal $\tilde{R}$-semiconfluent set with respect to any ordering of A is regular.*

Similarly we can show the following two theorems.

Theorem 5.7. *The following are equivalent.*

(1) *There exists a regular $\tilde{R}$-semiconfluent set.*
(2) *There exists a constant $k > 0$ with the following property. For any $p \in J(R)$ there exists $q \in A^*$ such that $(p, q) \in \langle R \rangle$, $p > q$ and p is in the hausdorff k-neighbourhood of q in the Cayley graph $\Gamma_A(G)$.*

Theorem 5.8. *Suppose that there exists a constant $k > 0$ with the following property. For any $p \in A^*$ such that in $\Gamma_A(G)$ p is not a geodesic but any proper* *subword of* p *is a geodesic, there exists* $q \in A^*$ *such that* $(p, q) \in \langle R \rangle$, $\text{length}(p) > \text{length}(q)$ *and* p *is in the hausdorff* k*-neighbourhood of* q *in the Cayley graph* $\Gamma_A(G)$. *Then the set of all geodesic words in* A^* *is a regular language over* A.

Theorem 5.9. *If* (A,R) *has a regular* $\tilde{R}$*-semiconfluent set, then* $G = A^*/\langle R \rangle$ *has a solvable word problem.*

Proof. This is clear since the set of $\langle R \rangle$-irreducible words is regular. But directly for a given word $r \in \tilde{A}^*$ we can find $\langle R \rangle$-irreducible word r' such that $(r, r') \in \langle \tilde{R} \rangle$ by using (e, ϵ) and elements of the regular $\tilde{R}$-semiconfluent set with length at most length(r). (See the statement previous to Theorem 2.4, if necessary.) □

Theorem 5.10. *If the set of all* $\langle R \rangle$*-irreducible words is regular, or in particular if* (A, R) *has a regular* $\tilde{R}$*-semiconfluent set, then the growth function of* $G = A^*/\langle R \rangle$ *with respect to the generating set* A *is a rational function.*

Sketch of Proof (*K. Saito et al.*). Let M be the finite state automaton accepting all the $\langle R \rangle$-irreducible words. For any nonnegative integer n let a_n be the number of elements in $L(M)$ with length n. Then the growth function in z

$$\sum_{n=0}^{\infty} a_n z^n = \sum_{n=0}^{\infty} ({}^t e \cdot H^n \cdot f) \cdot z^n = {}^t e \cdot (I - zH)^{-1} \cdot f$$

is a rational function in z, where H is the transition matrix of M, e is the vector of its accept states, f is the vector of its initial states and I is the unit matrix. □

§6. Amalgamation of States with the Same Word Difference

When we start with a finite set R and try to find an infinite regular $\tilde{R}$-semiconfluent set, how can we produce a finite state automaton T over $\tilde{A} \times \tilde{A}$ such that $L(T) \subset \langle \tilde{R} \rangle$ and $L(T)$ is infinite? If we are given a candidate for a finite state automaton accepting an $\tilde{R}$-semiconfluent set, we can check its correctness at least in easy cases such as Example 2.7.2. But if we simply continue the Knuth-Bendix algorithm, we cannot obtain even a candidate in finite time. In this section we describe a way, *amalgamation of states*, for producing an infinite (and regular) set of relations from a given finite one. Furthermore we show that finitely generated and regularly related groups are finitely related.

Definition 6.1. Let M be a finite state automaton over $\tilde{A}\times\tilde{A}$ and let s_1 and s_2 be states of M. *Amalgamation* of the states s_1 and s_2 is topological identification of s_1 and s_2 *in* M. More strictly, the result of amalgamation of s_1 and s_2 is the finite state automaton obtained by adding one new state s_0 to M, changing the source of every arrow with s_1 or s_2 as its source to the state s_0, changing targets similarly and removing s_1 and s_2 from M. (s_0 is an accept (resp. initial) state if and only if either s_1 or s_2 is an accept (initial) state.)

Notice that all words accepted by the original finite state automaton are accepted by the result of amalgamation.

Definition 6.2. Let M be a finite state automaton over $\tilde{A}\times\tilde{A}$. Let S be a subset of $\tilde{A}^*\times\tilde{A}^*$ such that $G=\tilde{A}^*/\langle S\rangle$ is a group. We say that M is a word difference automaton (with respect to S) if there exists a function f from the state set of M to the group G, with the following property. The images of the initial state and all accept states are the identity element in G, and existence of an arrow labelled $(x, y)\in\tilde{A}\times\tilde{A}$ with a source state s_1 and a target state s_2 implies $f(s_2)=\bar{x}^{-1}f(s_1)\bar{y}$ (where for all $z\in\tilde{A}^*$, $\bar{z}$ denotes the element of G represented by z).

From now on we deal only with finite state automata which have no dead state and no inaccessible state.

Lemma 6.3. *Let S be a subset of $\tilde{A}^*\times\tilde{A}^*$ such that $G=\tilde{A}^*/\langle S\rangle$ is a group. Let M be a finite state automaton over $\tilde{A}\times\tilde{A}$. Then the following are equivalent.*

(1) $L(M)\subset\langle S\rangle$.

(2) *M is a word difference automaton.*

Proof. Clearly (2) implies (1).

(1) implies (2) We define f in Definition 6.1. First fix a state s of M. Choose a path of arrows from the initial state to the state s. Let

(u, v) be the element of $(\tilde{A} \times \tilde{A})^*$ represented by this path and define $f(s) = \bar{u}^{-1}\bar{v}$. We show that $f(s)$ is well-defined (for this fixed s) to conclude the proof. We fix a path of arrows from the state s to an accept state. Let (u', v') be the element of $(\tilde{A} \times \tilde{A})^*$ represented by this path. Then $(uu', vv') \in L(M) \subset \langle S \rangle$. Thus $\overline{uu'} = \overline{vv'}$ and so $\bar{u}^{-1}\bar{v} = \bar{u}' \bar{v}'^{-1}$. □

Definition 6.4. When M is a word difference automaton over $\tilde{A} \times \tilde{A}$, the function f in Definition 6.2 is unique and so we call the image of a state by f the *word difference* of the state.

The following two lemmas are easily shown.

Lemma 6.5. *Let M be a word difference automaton. Then the result of amalgamation of states of M with the same word difference is also a word difference automaton.*

Lemma 6.6. *Let M and M' be word difference automata with respect to S (which may be the same). Suppose that M accepts $(s_1w_1t_1, s_2w_2t_2)$ and M' accepts (s_1t_1, s_2t_2) for some (s_1, s_2), (w_1, w_2), $(t_1, t_2) \in (\tilde{A} \times \tilde{A})^*$. Then the target state of the path of arrows in M representing (s_1, s_2) and the target state of the path in M representing (s_1w_1, s_2w_2) have the same word difference.*

Way to find a regular set of relations from finite relations. Lemmas 6.5 and 6.6 give the way to make a finite state automaton T over $\tilde{A} \times \tilde{A}$ such that $L(T) \subset \langle \tilde{R} \rangle$ and $L(T)$ is infinite, from finite sets of relations $L(M)$ and $L(M')$. That is, we may be able to make such a word difference automaton by finding two states of M with the same word difference using M', and amalgamating these two states.

Theorem 6.7. *Let T be a finite state automaton (without a dead state nor an inaccessible state) over $\tilde{A} \times \tilde{A}$ and N be a finite subset of $\tilde{A}^* \times \tilde{A}^*$ which contains (e, ϵ). If $G = \tilde{A}^*/\langle L(T) \cup N \rangle$ is a group, then G is finitely presented.*

Proof. Without loss of generality we can assume that no arrow of T has the initial state as its target. We fix $z_x \in \tilde{A}^*$ such that $(xz_x, \epsilon) \in \langle L(T) \cup N \rangle$ (so $(z_x x, \epsilon) \in \langle L(T) \cup N \rangle$). We can assume that (xz_x, ϵ), $(z_x x, \epsilon) \in N$ for all $x \in A$ by adding to N these elements (xz_x, ϵ), $(z_x x, \epsilon)$, if necessary (of course this addition does not affect $\langle L(T)\cup N\rangle$. Then for all $L\subset\tilde{A}^*\times\tilde{A}^*$ $\tilde{A}^*/\langle L\cup N\rangle$ is a group.

We prove the theorem inductively with respect to the number

(1) $\sum_s\{$(the number of arrows with the state s as their targets)$-1\}$

where the sum is taken over all states s of T except the initial state. Notice that this number (1) is at least 0. When it is 0, there exists just one arrow with each state as its target except for the initial state. This implies that $L(T)$ is finite.

We assume that the theorem is true when the number (1) is $n(\geq 0)$. Suppose that the number (1) with respect to T is $n+1$. We must show that $\tilde{A}^*/\langle L(T)\cup N\rangle$ is finitely presented. Since the number (1) is greater than 0, more than one arows have one state as their targets. Choose one such arrow a_1, let s_0 be its source state and let s_1 be its target. There exists a sequence of arrows from the state s_1 to an accept state. Let a_2, $\cdots$, a_m be such a sequence.

We construct a new finite state automaton T' as follows. Remove the arrow a_1 from T. Add m new states s'_1, $\cdots$, s'_m. Finally add m arrows a'_i $(1\leq i\leq m)$ with the same label as a_i from the state s'_{i-1} to the state s'_i where s'_0 denotes s_0 and s'_m is an accept state (s'_1, $\cdots$, s'_{m-1} is not accept states). Notice that the number (1) with respect to T' is n.

$\tilde{A}^*/\langle L(T')\cup N\rangle$ is a group. T' is a word difference automaton with respect to $L(T')\cup N$ by Lemma 6.3. Further s_m and s'_m have the same word difference 1. Since a_m and a'_m have the same label, s_{m-1} and s'_{m-1} have the same word difference. Similarly s_i and s'_i have the same word difference for all $m\geq i\geq 1$. We obtain (a copy of) T by amalgamating s_i and s'_i for all $1\leq i\leq m$. This means that T is a word difference automaton with respect to $L(T')\cup N$. Thus

$$L(T)\subset\langle L(T')\cup N\rangle \quad\text{and}\quad L(T)\cup N\subset\langle L(T')\cup N\rangle.$$

Clearly $L(T)\supset L(T')$. Hence $\langle L(T)\cup N\rangle=\langle L(T')\cup N\rangle$. By the induction hypothesis $G=\tilde{A}^*/\langle L(T)\cup N\rangle=\tilde{A}^*/\langle L(T')\cup N\rangle$ is finitely presented. □

Corollary 6.8. *If (A, R) has a regular $\tilde{R}$-semiconfluent set over $\tilde{A}\times\tilde{A}$ and $A^*/\langle R\rangle$ is a group, then $A^*/\langle R\rangle$ $(=\tilde{A}^*/\langle\tilde{R}\rangle)$ is a finitely presented group.*

Corollary 6.9. *All (asynchronous) automatic groups are finitely presented.*

§7. To Reduce All Elements Accepted by a Finite State Automaton

We want to apply the Knuth-Bendix algorithm to the case when R_i's in the definition of this algorithm are regular over $\tilde{A} \times \tilde{A}$. Then the remaining problem is how from given finite state automata S and T over $\tilde{A} \times \tilde{A}$ such that $L(S)$, $L(T) \subset \langle \tilde{R} \rangle$ and $L(T)$ is normalized, we should find a finite state automaton M over $\tilde{A} \times \tilde{A}$ such that $L(S) \to^*_{L(T)} L(M)$, $L(M) \subset \langle \tilde{R} \rangle$ and all elements of $L(M)$ are pairs of $L(T)$-irreducible elements. But as recognized later the author gave up finding M which accepts only pairs of $L(T)$-irreducible elements (see Remark below).

Definition 7.3 below gives a way to find a finite state automaton M as above. To do this more efficiently, we decompose S by path of arrows (Definition 7.1) previously. These ways can be done constructively.

Definition 7.1 (decomposition of a finite state automaton by path). Let S be a finite state automaton. Let AR be the set consisting of all arrows of S. Let AR_{seq} be the set consisting of all finite sequences of arrows from an initial state to an accept state. Let AR_{im} be the image of the obvious projection from AR_{seq} to 2^{AR} (the set consisting of all subset of AR). For all $P \in AR_{\mathrm{im}}$, we define S_P as the finite state automaton obtained by removing from S all arrows not in P (and all dead states). Then we obtain the *decomposition* $L(S) = \bigcup_{P \in AR_{\mathrm{im}}} L(S_P)$.

Definition 7.2. Let T be a finite state automaton over $\tilde{A} \times \tilde{A}$. We say that T is *of reducing type* if its initial state is the only accept state, and $L(T)$ is normalized.

Definition 7.3 (complete reduction of a finite state automaton). Let $S_0 = S$ be a finite state automaton over $\tilde{A} \times \tilde{A}$ (we may think of it as an above S_P). Let T be a finite state automaton of reducing type. Let T' stand for the finite state automaton obtained by adding to T an arrow labelled (x, x) from the initial (and accept) state to itself for each $x \in \tilde{A}$. Then

$$L(T') = D_{\tilde{A}^*} \cup D_{\tilde{A}^*}\ L(T) D_{\tilde{A}^*} \cup D_{\tilde{A}^*}\ L(T) D_{\tilde{A}^*}\ L(T) D_{\tilde{A}^*} \cup \cdots.$$

For $i = 1$ or 2, we say that a function h is an *i-1-homomorphism* from S to

T' if h maps states of S to states of T', the initial state and all accept states of S to the initial (and accept) state of T', each arrow labelled (x_1, x_2) to an arrow labelled (x_i, y) for some $y \in \tilde{A}$ so that the image of its source state is the source state of its image and the image of its target is the target of its image.

We say that h is an *i-1-reduction homomorphism* if h is an i-1-homomorphism and the label of the image of some arrow of S is the form (x_i, y), where $x_i > y$.

If there exists an i-1-reduction homomorphism for some i, let S_1 be the finite state automaton obtained by replacing x_i of each label of S_0 by y of the label of T' in the definition of i-1-homomorphism. Then $L(S_0) \to^*_{L(T)} L(S_1)$ and $L(S_1) \subset \langle L(S_0) \cup L(T) \rangle$. Then we say that S_1 is a result of *(i-)reduction* of S_0 by T. By repeating this procedure, we obtain a sequence of finite state automata $S = S_0, \cdots, S_m$ such that $L(S_0) \to^*_{L(T)} \cdots \to^*_{L(T)} L(S_m)$, S_m cannot be reduced (or there does not exist i-1-reduction homomorphism from S_m to T') and $L(S_m) \subset \langle L(S_0) \cup L(T) \rangle$.

Claim. This process necessarily terminates.

We say that S_m is a result of *complete reduction* of S by T.

Proof of Claim. We can assume that there exists a path γ of arrows in S from the initial state to an accept state including all arrows in S, by considering the decomposition of S. It is sufficient to show that for fixed i there does not exist an infinite sequence of finite state automata $S = S_0$, $S_1, \cdots$ such that S_{j+1} is a result of i-reduction of S_j for all $j \geq 0$. Assume that there exists such a sequence. For nonnegative integer k let $(w_{1,k}, w_{2,k})$ be the element of $(\tilde{A} \times \tilde{A})^*$ represented by the path of arrows in S_k corresponding to the path γ in S. Since S_{j+1} is a result of i-reduction of S_j, $w_{i,j+1} < w_{i,j}$ holds. This means that $w_{i,0} > w_{i,1} > \cdots$. Since $<$ is a well-ordering, this cannot happen. Thus we conclude the proof. □

Remark. By the following example it seems to be impossible to make a finite state automaton M consisting of only $L(T)$-irreducible elements.

Set $L(T) = \{(zz, ee), (zz, yy), (yy, wx), (xw, wx)\}$ and $L(S) = \{(z^{2n}, y^{2n}) | n \geq 1\}$. Then for example $L(M)$ should be $\{(e^{2n}, w^n x^n) | n \geq 1\}$ because of $z^{2n} = (zz)^n \to^*_{L(T)} (ee)^n = e^{2n}$ and $y^{2n} = (yy)^n \to^*_{L(T)} (wx)^n \to^*_{L(T)} w^n x^n$. But this set is not regular.

Definition 7.4. Let H be a finite state automaton. We say that H is

NAIT if H has no arrow with the initial state as its target and if its initial state is not an accept state. We say that H is *NAAS* if H has no arrow with an accept state as its source.

If H and H' are finite state automata which are NAIT and if H'' is the result of amalgamation of the initial states of H and H', clearly $L(H'')=L(H)\cup L(H')$ holds.

The following definition gives a way to find M in the head of this section from S and T using decomposition and complete reduction.

Definition 7.5 (decomposition complete reduction). Let S be a finite state automaton over $\tilde{A}\times\tilde{A}$ which is NAIT. Let T be a finite state automaton of reducing type. Let $L(S)=\bigcup_P L(S_P)$ be the decomposition of S. Since S is NAIT, S_P has just one arrow with the initial state as its source for all P. For each P let S'_P be the result of complete reduction of S_P by T. S'_P has just one arrow with the initial state as its source. Let (x_1, x_2) be the label of this arrow. We can assume that $x_1\geq x_2$ by interchanging y_1 and y_2 for all label (y_1, y_2) in S_P, if necessary. We call the result of amalgamation of the initial states of all S_P's *the result of decomposition complete reduction* of S by T.

Lemma 7.6. *Let S be a finite state automaton over $\tilde{A}\times\tilde{A}$ which is NAIT. Let T be a finite state automaton of reducing type. If S' be the result of decomposition complete reduction of S by T. Then $L(S)\to^*_{L(T)}L(S')$ and $L(S')\subset\langle L(S)\cup L(T)\rangle$. Further every element of $L(S')$ has the form (x_1w_1, x_2w_2) where $x_1, x_2\in\tilde{A}$, $w_1, w_2\in\tilde{A}^*$ and $x_1\geq x_2$.*

Let S be a finite state automaton over $\tilde{A}\times\tilde{A}$ and let T be a finite state automaton of reducing type. Assume that $L(S)$, $L(T)\in\langle\tilde{R}\rangle$. Now we go into details to find a finite state automaton M such that $L(M)\subset\langle\tilde{R}\rangle$ and

(1) $$\text{for all } (p, q)\in L(S) \quad p\cup_{L(T)\cup L(M)} q.$$

Notice that (1) is a weaker condition than $L(S)\to^*_{L(T)}L(M)$. (Of course defining $L(M)=L(S)$ satisfies (1), but it will not be practical in terms of computer time and memory, and further whether the algorithm stops or not.) It may seem to be sufficient that we define M as the result of decomposition complete reduction. Nevertheless such M may accept (eu, ev) for some $u, v\in\tilde{A}^*$. So such M is not good (see Lemma 5.1). Next we consider removing such elements (eu, ev).

For all $U \subset \tilde{A}^* \times \tilde{A}^*$, we define $\lceil U \rceil = \{(p, q) \in \tilde{A}^* \times \tilde{A}^* | p \cup_U q\}$. Then (1) is the same as $L(S) \subset \lceil L(T) \cup L(M) \rceil$. Notice that if $U_1 \subset \lceil U_2 \rceil$, then $\lceil U_1 \rceil \subset \lceil U_2 \rceil$.

Definition 7.7 (removing arrows with diagonal labels). Let S be a word difference automaton over $\tilde{A} \times \tilde{A}$ with respect to $\tilde{R}$. Define S' as the result of amalgamation of the initial state of S, its all accept states and its states connected with the initial state or accept states by paths of non-oriented arrows with labels of the form $(x, x) \in \tilde{A} \times \tilde{A}$. (Notice that $L(S) \subset L(S') \subset \langle \tilde{R} \rangle$.) Let S'' be the finite state automaton obtained by removing from S' every arrow labelled (x, x) for some $x \in \tilde{A}$ with the initial state as its source (so its target). (Then S'' does not accept (xu, xv) or (ux, vx) for any $x \in \tilde{A}$, $(u, v) \in (\tilde{A} \times \tilde{A})^*$. Notice that $L(S'') \subset L(S')$ and $L(S') \subset \lceil L(S'') \rceil$.)Let S''' be the finite state automaton obtained by adding to S'' a new initial state (the old initial state is no longer an initial state) and changing the source of every arrow with the old initial state as its source to the new initial state. (Then $L(S''') \subset L(S'')$ and $L(S'') \subset \lceil L(S''') \rceil$. Notice that the result of amalgamation of the initial state of S''' and its accept state is the same as S''.) We say that S''' is the *result of removing arrows with diagonal labels.*

Lemma 7.8. *Let S be a word difference automaton with respect to $\tilde{R}$. Let S''' be the result of removing arrows with diagonal labels. Then S''' does not accept (xu, xv) or (ux, vx) for any $x \in \tilde{A}$, $(u, v) \in (\tilde{A} \times \tilde{A})^*$. In particular S''' does not accept (eu, ev) for any $(u, v) \in (\tilde{A} \times \tilde{A})^*$. Further $L(S) \subset \lceil L(S''') \rceil \subset \langle \tilde{R} \rangle$. S''' is NAIT and NAAS.*

The following definition gives a way to make M itself in the head of this section from S and T.

Definition 7.9 (RAD-DCR*). Let $S = S_0$ and T be word difference automata over $\tilde{A} \times \tilde{A}$ with respect to $\tilde{R}$. Let T be of reducing type. Define S'_0 as the result of removing arrows with diagonal labels of S_0, and S_1 as the result of decomposition complete reduction of S'_0. (Then Lemma 7.8 tells that $L(S_0) \subset \lceil L(S'_0) \rceil \subset \langle \tilde{R} \rangle$. Lemma 7.6 tells that $L(S'_0) \to^*_{L(T)} L(S_1)$ and $L(S_1) \subset \langle L(S'_0) \cup L(T) \rangle$. These mean that $L(S_0) \subset \lceil L(S'_0) \rceil \subset \lceil L(S_1) \cup L(T) \rceil \subset \langle \tilde{R} \rangle$.) Similarly we can define S'_1, S_2, S'_2, $\cdots$, S'_m, S_{m+1}, where

S'_m cannot be reduced (or more strictly, any element S'_{mP} of decomposition of S'_m cannot be reduced).

Claim. This process necessarily terminates.

Then we call S_{m+1} the *result of RAD-DCR** of S by T.

Sketch of the proof of Claim: For each $n \geq 0$ we define a_n as follows. Let $L(S'_n) = \bigcup_P L(S'_{nP})$ be the decomposition of S'_n. We define

$$a_n = \max_P \{\text{the number of arrows in } S'_{nP}\},$$

where the maximum is taken over P for which S'_{nP} can be reduced. Then a_n is strictly decreasing sequence. So this sequence is not infinite. □

Since any element S'_{mP} of decomposition of S'_m cannot be reduced,

$$L(S_{m+1}) = \{(w_1, w_2) \mid (w_1, w_2) \text{ or } (w_2, w_1) \in S'_m, \text{ and } w_1(1) > w_2(1)\},$$

where $w(1)$ means the first letter of w. (Notice that S'_m does not accept (xu, xv) for any $x \in \tilde{A}$, $(u, v) \in (\tilde{A} \times \tilde{A})^*$ from Lemma 7.8). Further S_{m+1} is NAIT and NAAS since so is S'_m. From $L(S_0) \subset \lceil L(S_1) \cup L(T) \rceil \subset \langle \tilde{R} \rangle$, similarly it holds that

$$L(S_0) \subset \lceil L(S_1) \cup L(T) \rceil \subset \lceil L(S_2) \cup L(T) \rceil \subset \cdots \subset \lceil L(S_{m+1}) \cup L(T) \rceil \subset \langle \tilde{R} \rangle.$$

Thus we obtain the following lemma.

Lemma 7.10. *Let S and T be word difference automata over $\tilde{A} \times \tilde{A}$ with respect to $\tilde{R}$. Let T be of reducing type. Define M as the result of RAD-DCR* of S by T. Then $L(S_0) \subset \lceil L(M) \cup L(T) \rceil \subset \langle \tilde{R} \rangle$. M is NAIT and NAAS. Further M accepts only elements of the form $(w_1, w_2) \in (\tilde{A} \times \tilde{A})^*$ where $w_1(1) > w_2(1)$.*

§8. An Improvement of the Knuth-Bendix Algorithm

Let a finite set A and a normalized finite subset R of $A^* \times A^*$ be given explicitly. Let a total ordering of A be given. Then a well-ordering $<$ of A^* is determined as in Example 2.1 (lexicographical ordering). We have prepared to exhibit our (non-effective) algorithm A to produce a regular $\tilde{R}$-semiconfluent set using Lemmas 3.2, 5.1 and 5.2. This algorithm is very similar to the Knuth-Bendix algorithm except for using finite state automata.

Now we describe the algorithm. First construct the finite state automaton S_0 over $\tilde{A} \times \tilde{A}$ accepting $\tilde{R}'$. (Since we no longer refer to R, finiteness of R is not essential as long as $\tilde{R}'$ is regular.) Let T_0 be the result of amalgamation of the initial state of S_0 and all its accept states. We define recursively finite state automata S_n and T_n for $n \geq 0$, with the property that for all $n \geq 0$ (in fact, T_n is the result of amalgamation of the initial state of S_n and all its accept states. See Remark below about the reason why T_n is necessary.)

(1) S_n is NAIT and NAAS,
(2) $L(S_n)$ and $L(T_n)$ are normalized,
(3) T_n is of reducing type,
(4) $\langle \tilde{R}' \rangle \subset \langle L(S_n) \rangle = \langle L(T_n) \rangle \subset \langle \tilde{R} \rangle$,
(5) S_n does not accept (eu, ev) for any $(u, v) \in (\tilde{A} \times \tilde{A})^*$,
(6) for all $(p, q) \in T_n$ $p \cup_{L(S_n)} q$,
(7) for all $(p, q) \in K(S_{n-1})$ $p \cup_{L(T_n)} q$ and so $p \cup_{L(S_n)} q$, where we define $L(S_{-1}) = \emptyset$.

Let n be a nonnegative integer and assume that S_{n-1}, S_n and T_n are defined, satisfying the above (1)–(7). We must define S_{n+1} and T_{n+1}. First construct a finite state automaton K_n accepting $K(L(S_n)) \backslash K(L(S_{n-1}))$. Next let M_n be the result of RAD-DCR* of K_n by T_n. Then

a) for all $(p, q) \in K(L(S_n)) \backslash \mathrm{K}(\mathrm{L}(S_{n-1}))$, $p \cup_{L(M_n) \cup L(T_n)} q$,
b) $L(M_n) \subset \langle \tilde{R} \rangle$,
c) $L(M_n)$ is normalized,
d) M_n does not accept (eu, ev) for any $(u, v) \in (\tilde{A} \times \tilde{A})^*$,
e) M_n is NAIT and NAAS.

If $L(M_n) \subset L(T_n)$, then for all $(p, q) \in K(L(S_n)) \backslash K(L(S_{n-1}))$, $p \cup_{L(T_n)} q$. From (7) it holds that for all $(p, q) \in K(L(S_n))$, $p \cup_{L(T_n)} q$ and so $p \cup_{L(S_n)} q$. From (2), Lemma 3.2 tells that S_n is confluent. From (2), (4) and (5), Lemmas 5.1 and 5.2 tell that $L(S_n)$ is $\tilde{R}$-semiconfluent and this algorithm terminates.

Suppose that $L(M_n) \not\subset L(T_n)$. Find pairs of states of M_n with the same word difference, if any, using T_n except for its initial state and its accept states (see Lemma 6.6). Amalgamate each of the pairs and let M_n' be the result. We define S_{n+1} as the result of amalgamation of the initial state of S_n and the initial state of M_n' (so $L(S_{n+1}) = L(S_n) \cup L(M_n')$). Define T_{n+1} as the result of amalgamation of the initial (and accept) state of T_n, the initial state of M'_n and all its accept states.

Remark. Similarly to the case of the Knuth-Bendix algorithm, we can define S_{n+1} as $L(S_{n+1})=\hat{C}(L(S_n))\cup L(M'_n)$. In this case notice that T_n is not necessarily the result of amalgamation of the initial state of S_n and all its accept states. This change of definition almost always makes this algorithm more efficient. Further the author thinks that we should require only the minimal $\tilde{R}$-semiconfluent set to find a semiconfluent set more quickly. Actually in our computer program some other tricks are used. But we shall not go into details.

This algorithm Λ is implemented as a computer program written by C language. The program consists of about 2900 lines. In Section 10 we give some examples to which we apply this program.

§9. Orderings of Generators

When we apply the algorithm Λ to a given finite set A and a given subset R of $A^*\times A^*$, generally we do not know which total ordering of A is the best. In this section we describe a principle to know, while executing this algorithm for an ordering of A, that the ordering is likely to be bad.

From now on we assume that A is closed under inversion in $A^*/\langle R\rangle$ so in $\tilde{A}^*/\langle\tilde{R}\rangle$. (But this restriction is not essential.) Let x and y be elements of A and let u and v be elements of A^* satisfying $(xuy, v)\in\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*$. Further let x' and y' be elements of A such that $xx'=yy'=1$ in $\tilde{A}^*/\langle\tilde{R}\rangle$. Then $(x'v,eeuy)$, $(vy',eexu)\in\langle\tilde{R}\rangle\cap(\tilde{A}\times\tilde{A})^*$. Suppose that $x<v(1)$ (the first letter of v). Then $xuy<v$ by the lexicographical ordering. So these two relations $(x'v, eeuy)$, $(vy', eexu)$ are unnecessary to produce a $\tilde{R}$-semiconfluent set since v is not $\langle R\rangle$-irreducible and so $x'v$, $vy'\notin J(R)$. Thus

Principle 9.1. A given total ordering of A is perhaps bad for the algorithm Λ if, while executing Λ for the ordering, we get a pair of relations of the form $(x'v, eeuv)$, $(vy', eexu)$ accepted by an S_n in the definition of Λ where x, y, x', $y'\in A$ and $u,v\in A^*$ such that $x>v(1)$, v is $L(S_n)$-irreducible and $xx'=yy'=1$ in $A^*/\langle R\rangle$. Thus if we get such two relations, it is perhaps better to interchange the ordering of x and $v(1)$. (The ordering of the other elements of A may also change.)

Of course these two relations may be unnecessary even if we do not change the ordering. In addition since this change of the ordering affects the other relations, in some cases this principle will work wrong. In the next section some examples will be given where it works well.

§10. Some Examples

In this section we give some examples to which we apply the algorithm A, implemented as a computer program. In the case of Examples 10.1–10.3 the cpu time to find the semiconfluent set is less than 10 seconds for small l, m and n.

Example 10.1. We consider the free abelian group of rank 2 as in Example 2.7

$$\langle x, y \mid [x, y]=1 \rangle.$$

Set $A=\{x, X, y, Y\}$ and $R=\{(xX, \epsilon), (Xx, \epsilon), (yY, \epsilon), (Yy, \epsilon), (xyXY, \epsilon)\}$. Then $\tilde{A}=\{e, x, X, y, Y\}$ and $\tilde{R}'=\{(xX, ee), (Xx, ee), (yY, ee), (Yy, ee), (xyXY, eeee)\}$.

10.1.1 We define $<$ as in Example 2.1 (lexicographical ordering) assuming that $e<x<X<y<Y$. Then the algorithm A terminates with the following finite $\tilde{R}$-semiconfluent set (or a finite state automaton accepting it).

$$\{(xX, ee), (Xx, ee), (yY, ee), (Yy, ee), (yx, xy), (yX, Xy), (Yx, xY), (YX, XY)\}.$$

10.1.2 We define $<$ assuming that $e<x<y<X<Y$. The algorithm A still terminates with the $\tilde{R}$-semiconfluent set

$$\{(xX, ee),\cdots,(Yy, ee), (yx, xy), (yX, Xy), (Yx, xY), (YX, XY)\} \cup \bigcup_{n=1}^{\infty} \{(xy^nX, eey^n), (yX^nY, eeX^n)\}.$$

Example 10.2. We consider the triangle groups

$$\Delta(l, m, n)=\langle x, y, z \mid x^2=y^2=z^2=(xy)^l=(xz)^m=(yz)^n=1\rangle.$$

Set $A=\{x, y, z\}$ $(x<y<z)$ and

$$R=\{(xx, \epsilon), (yy, \epsilon), (zz, \epsilon), ((xy)^l, \epsilon), ((xz)^m, \epsilon), ((yz)^n, \epsilon)\}.$$

The algorithm Λ is likely to terminate with a finite $\tilde{R}$-semiconfluent set for any l, m, $n\geq 1$.

Example 10.3. Consider the von Dyck groups

$$D(l, m, n)=\langle u, v \mid u^l=v^m=(uv)^n=1\rangle.$$

Set $A=\{u, U, v, V\}$ $(u<U<v<V)$ and

$$R=\{(uU, \epsilon),\cdots,(Vv, \epsilon), (u^l, \epsilon), (v^m, \epsilon), ((uv)^n, \epsilon)\}.$$

Then the algorithm Λ seems to terminate for all l, m, $n\geq 1$. The produced $\tilde{R}$-semiconfluent set is often finite, but in the cases $l=3$, $n=2$ it appears to be infinite. However also in these cases the algorithm perhaps produces a finite $\tilde{R}$-semiconfluent set, if we newly define the ordering $<$ as satisfying $e<u<v<U<V$.

Example 10.4. Consider the groups

$$\langle x_1, y_1, x_2, y_2, z \mid [x_1, y_1]\,[x_2, y_2]z^l=1\rangle.$$

Define $A=\{x_1, X_1, y_1, Y_1, x_2, X_2, y_2, Y_2, z, Z\}$ and R as usual.

10.4.1 Define the ordering $<$ as satisfying $e<x_1<$(the ordering in the definition of A). For $l=0$, 1 and 2 the algorithm Λ terminates. For $l=2$ the produced $\tilde{R}$-semiconfluent set is finite and for $l=0,1$ it is infinite.

10.4.2 By Principle 9.1 we know that the following ordering may be better.

$$e<x_1<X_1<x_2<X_2<y_1<Y_1<y_2<Y_2<z<Z.$$

So we apply the algorithm Λ for this ordering. Then for $l=0$, 1, 2, 3 this algorithm terminates. The produced set is finite for $l=0$, 2 and infinite for $l=1$, 3. (In the case $l=1$, for any ordering of A the $\tilde{R}$-semiconfluent set is perhaps infinite.)

The following example is not the result of the computer program (but it is noticed by the above example).

Example 10.5. Consider the compact 2-dimensional surface groups

$$\langle x_1, y_1, \cdots, x_g, y_g \mid [x_1, y_1]\cdots[x_g, y_g]=1\rangle.$$

Set $A=\{x_1, X_1, y_1, Y_1, \cdots, x_g, X_g, y_g, Y_g\}$ and R as usual. Define $<$ as in Example 2.1 (lexicographical ordering) assuming that $x_1 < X_1 < \cdots < x_g < X_g < y_1 < Y_1 < \cdots < y_g < Y_g$. By manipulating we can easily show that the union of the following two sets is R-confluent.

$$\{(x_1X_1, \varepsilon), (X_1x_1, \varepsilon), \cdots, (y_gY_g, \varepsilon), \cdots, (Y_gy_g, \varepsilon)\}$$

$$\{(p, q)\in(A\times A)^* \mid p^{-1}q \text{ or } q^{-1}p \text{ is a subword of } (x_1y_1X_1Y_1\cdots x_gy_gX_gY_g)^2 \text{ with length } 4g \text{ and } p(1) \text{ (the first letter of } p\text{) is } y_i \text{ or } Y_i \text{ for some } i\}$$

where r^{-1} is the formal inverse of r for all $r\in A^*$, for example $(x_1y_1X_2)^{-1}=x_2Y_1X_1$.

Acknowledgment

The author wishes to thank K. Saito and the referee for their valuable comments and I. Naruki for his encouragement.

Bibliography

[AU] Aho, A.V. and Ullman, J.D., *The Theory of Parsing Translation, and Compiling*, Vol. I, Prentice-Hall, Englewood Cliffs, N.J., 1972.

[BGSS] Baumslag, G., Gersten, S.M., Shapiro, M. and Short, H., Automatic groups and amalgams, preliminary version, (*MSRI preprint* 09123–90).

[CEHPT] Cannon, J.W., Epstein, D.B.A., Holt, D.F., Patterson, M.S. and Thurston, W.P., Word processing and group theory, to appear. Preprint available from Epstein or Holt.

[ER] Epstein, D.B.A. and Rees, S., *Aut programs for automatic structures*, computer tape from Warwick University.

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

[Gr] Gromov, M., Hyperbolic groups, in Essays in Group Theory, *M.S.R.I. series* **8**, Gersten, S.M., editor, Springer Verlag (1987), 75–263.

[HU] Hopcroft, J.E. and Ullman, J.D., *Formal languages and their relation to automata*, Addison-Wesley, 1969.

[RS] Rabin, M.O. and Scott, D., Finite automata and their decision problems, *IBM Jour. of Research and development*, **3** (1959), 63–91, in Sequential machines: selected papers, edited by Moore, E.F., Addison-Wesley.

[Sa] Saito, K., The Limit Element in the Configuration Algebra for a Discrete Group: A précis, *Proc. Internat. Congress Math.*, Kyoto (1990), 931–942.