# *Complete Parameterized Presentations and Almost Convex Cayley Graphs*

*by*

*William Francis Simmonds.*

*Thesis submitted for the Degree of*
*Doctor of Philosophy*
*at the University of Warwick*
*November 1991.*

Mathematics Institute,
University of Warwick,
Gibbet Hill Road,
Coventry CV4 7AL.

*To my parents,*
*Edward and Maureen,*
*and to my sister*
*Maria*

# *Acknowledgements*

I would first like to say thank you to my supervisor Doctor Derek Holt: for suggesting research problems; for his help; for the care with which he read the proofs of my main results; also, for the kindness and patience he has shown towards me during my (lengthy) stay at Warwick – which was considerably more than I deserved.

If I may, I would like to take this opportunity (on resubmitting) to thank my Ph.D examiners Professor David Epstein (internal) and Doctor Geoff Smith (from Bath University). With hindsight, I realise that reading through my original thesis must have been pretty distressful for them. I would like to thank them for having persevered, and for their (many) suggestions for improvement (which I certainly bore in mind whilst writing this, hopefully, more readable thesis).

There are so many friends who I would like to thank at this point, but I have special reasons for saying thank you to the following people:

Marcos Bothelo,
Zac Coelho,
Matija Cencelj,
Stamatis Dostoglou,
Hermann Haaf,
Sofia Lambropoulou,
Ana Paula Santana,
and
Elaine Shiels
(for, among other things, her help with the Macintoshs (temperamental machines which I *most definitely* do not want to see *ever again!*)).

I am also pleased to acknowledge the (S)cience and (E)ngineering (R)esearch (C)ouncil for funding during the first two years of my Ph.D study.

# *Declaration*

The work of chapters 3,4 and 5 is, to the best of my knowledge, original, unless stated to the contrary.

# *Summary*

This thesis is meant as a contribution to the theory of three classes of groups, those classes being the groups defined by *complete parameterized presentations*, *automatic* groups, and groups with *almost convex* Cayley graphs.

Chapter 1 is basically definitions and terminology. Chapter 2 is a short exposition of the theory of automatic groups; we prove only one major result in this chapter (due to (CHEPT)), i.e., that the abelian groups are automatic.

In chapter 3 we study presentations of groups and monoids which are complete (with respect to certain orderings of the words in their generators). Such presentations define monoids with fast solutions to their word problems. We define a class of (possibly infinite) presentations which we call *r-parameterized*, or *of type* $P_r$; these presentations are the central theme of this thesis. With the help of the computer program described in chapter 4, we demonstrate that there are group presentations which have infinite r-parameterized completions (i.e. complete supersets), but which have no finite completion with respect to any *ShortLex* ordering. The 1-parameterized presentations are, arguably, the simplest non finite presentations we can define (at least as far as groups are concerned), but we prove that completeness of such presentations is not in general decidable.

Chapter 4 is the description of a (short) program which attempts to complete 1-parameterized group presentations by the Knuth-Bendix method. We conclude the chapter with a short report on its implementation.

In chapter 5 we study groups with almost convex Cayley graphs. Such graphs are recursive, but the property of being almost convex does tend to be hard to prove or disprove in practice. We prove that the *word length preserving* complete groups and the *least length bounded* automatic groups have almost convex Cayley graphs. We believe that these are strict subclasses because (we shall prove) the group U(3,$\mathbb{Z}$) is almost convex, but is already known not to be automatic and, we *conjecture*, it has no r-parameterized complete (ShortLex) presentation. We conclude chapter 5 with a slightly generalized, arguably simpler, algebraic proof of J.W. Cannon 's theorem that the abelian by finite groups are almost convex.

# Contents

§3 Monoids with Complete and Parameterized Presentations

§4 Programming the Knuth-Bendix Completion of (ShortLex) 1-Parameterized Group Presentations

§5 Almost Convex Cayley Graphs

# *§0*

# *Introduction*

This thesis is meant as a modest contribution to the theory of two classes of groups, those classes being the groups with *complete presentations*, with respect to a finite semigroup generating set, and the groups with *almost convex Cayley graphs*, with respect to a finite semigroup generating set. The study of these groups, primarily the complete groups, overlaps with a third class of groups, the *automatic groups*, but it is the groups defined by complete presentations which the author considers to be the focal point of this work and this is reflected in the layout of this thesis.

Chapter 1 is basically terminology, but we also quote some results on, and examples of, regular languages to which we will refer in the subsequent chapters.

Chapter 2 is meant as a short exposition of the theory of automatic groups and none of the work in this chapter is original.

A group G with finite semigroup generating set C may be automatic with respect to a finite state automaton called the *word acceptor*. The language of the word acceptor would be a subset of $C^*$, the free monoid on C, and the restriction, to this regular language, of the natural homomorphism from $C^*$ to G would be surjective. If (G,C) were automatic then (basically) there would be an interpretation, defined by the word acceptor, of the multiplication in the group G in terms of regular languages over the alphabet $C \times C$.

Automatic groups are currently the subject of much research (primarily because of their applications to certain topological problems). There is a substantial paper entitled 'Word Processing and Group Theory' (CEHPT), a collaboration of five authors closely concerned with the development of automatic groups, which details most of the current work in automatic groups. We will be referring to this paper frequently.

The emphasis of the work in chapter 2 is squarely on results which are referred to in chapters 3 and 5 and the only major result which is proved in this chapter is that the abelian groups are automatic. We believed this proof really should be included because it is a prerequisite of a proof in chapter 5 (abelian by finite groups are almost convex).

In chapters 3,4 and 5 we will be working with groups and monoids defined by complete presentations. If $\langle C \mid \mathcal{R} \rangle$ is a monoid presentation then the members of $\mathcal{R}$, taken as ordered pairs of words in $C^*$, may be thought of as *rewrite rules* for the words in $C^*$ (if a word has a subword $s$ with $(s, r)$ in $\mathcal{R}$ then that word can be *rewritten* with the subword $s$ replaced by $r$). The presentation $\langle C \mid \mathcal{R} \rangle$ would then be complete, with respect to some total ordering on the words of $C^*$, if, using the the rewrite rules of $\mathcal{R}$ repeatedly, all words can be rewritten as the least word in their $\langle \mathcal{R} \rangle$ congruence class.

So, if $\langle C \mid \mathcal{R} \rangle$ were complete, and $\mathcal{R}$ at least recursive, then the least words in the $\langle \mathcal{R} \rangle$ equivalence classes (called the *representatives*) will be a tractable normal form for the elements of the monoid $M = C^*/\langle \mathcal{R} \rangle$ in terms of the generators C. If M is a group and $\mathcal{R}$ finite, then the set of representatives is a regular subset of $C^*$, and thus a possible candidate for the language of a word acceptor of an automatic structure for (M,C). However, we give an example of a group (3.1.4) which is not automatic with the word acceptor accepting the set of representatives (although it is true that a group defined by a complete presentation where rewriting words involves no *backtracking* is automatic with word acceptor accepting the set of representatives).

Complete presentations are a small part of the more general theory of *rewriting techniques* which has a long history in theoretical computer science. So, not surprisingly, computers are apt tools for the (more recent) study of complete group and monoid presentations. In his paper 'Presentations of Groups and Monoids' (Gilman 79) R. Gilman describes an implementation of the *Knuth-Bendix procedure* for computing finite complete presentations. In his follow-up paper 'Enumerating Infinitely Many Cosets' (Gilman 84), Gilman notes that the success of this procedure is rather sensitive to the well ordering of the words and suggests that the procedure might be improved if it were to look for certain classes of infinite presentations which we will be calling *1-parameterized*.

Chapter 3 has two sections; section 3.1 is part expository, part original; section 3.2 is original. In this chapter we define *r-parameterized* presentations (which have been referred to by several authors) and look at some examples. Using the computer program, described in chapter 4, we are able to give an example of a group presentation with a 1-parameterized completion (i.e. complete superset of the defining relations), but no finite completion whatever the choice of ShortLex ordering on the words of $C^*$.

The 1-parameterized presentations are, arguably, the simplest infinite presentations which we could hope to define, but, nevertheless, there would still appear to be significant problems in the study of such presentations. Whereas completeness is decidable for finite presentations, it is not always decidable for 1-parameterized monoid presentations. We prove this by reducing the halting problem of a Turing machine to the problem of deciding the completeness of a 1-parameterized monoid presentation (although the monoid presentation in question may not be that of a group). Moreover, C.C. Squier in his paper 'Word problems and homological finiteness conditions' (Squier) cites an example of a monoid defined by a 1-parameterized presentation which suggests that the existing structure theorems for finite complete groups are perhaps unlikely to be extended to groups defined by 1-parameterized complete presentations.

In the first four sections of chapter 4 we describe a computer program (written in pseudo 'C') which attempts to complete (ShortLex) 1-parameterized group presentations. Such programs (of Knuth-Bendix completion) can be kept rather simple, but we do employ some non-standard techniques to speed up the completion process. The program appears to be reasonably successful and we used it to compute some of the presentations of chapter 3. We conclude chapter 4 with a short report on its implementation (i.e. section 4.5).

Almost convex groups, or to precise, groups with almost convex Cayley graphs are a large class of groups defined by J.W. Cannon in his preprint 'Almost Convex Groups' (81). This class of groups is of interest because their Cayley graphs are (in theory at least) recursive, in fact there is an efficient algorithm for constructing such graphs. There are several problems in this subject, notably that the property of a Cayley graph being almost convex does tend to be rather difficult to prove or disprove in practice.

We begin chapter 5 by proving that the groups defined by complete parameterized presentations with *word length preserving* orderings and the *least length bounded* automatic groups have almost convex Cayley graphs. We believe these classes to be strict inclusions. In 5.3 we prove that the (nilpotent, non abelian) group of 3 by 3 unitriangular matrices over **Z** has an almost convex Cayley graph (with respect to a certain set of generators). This group has no automatic structure, that much is known, and, we conjecture, that it has no complete, parameterized presentation with respect to a Shortlex ordering. We conclude chapter 5 with an alternative (and slightly generalized) algebraic proof of Cannon's theorem that the abelian by finite groups are almost convex.

# *§1*

# *Definitions and Terminology*

### *Words and Monoids.*

Formal word manipulation plays a large part in this thesis. If S is a set of symbols then we write $S^*$ for the the free monoid of formal words in S, i.e., the monoid consisting of the words of $S^*$ with multiplication being concatenation and the identity element being the empty word. The set of symbols S may then be referred to as the as the *alphabet* of $S^*$. The symbol $\varepsilon$ is reserved throughout the thesis for the empty word.

If $v$ and $w$ are words of $S^*$ then we write $v \equiv w$ when $v$ and $w$ are identical words. We write $|w|$ for the number of symbols occurring in the word $w$ and, if $s$ is a symbol of S, then no. $s(w)$ is the number of occurrences of the symbol $s$ in the word $w$.

If $w$ is a word of $S^*$ and $i \in \mathbb{N}$ with $1 \leq i \leq |w|$,then we write $w[\,i\,]$ for be $i^{th}$ symbol of $w$. If $i,j \in \mathbb{N}$ with $i,j \geq 1$, then we define $w(\,i,j\,)$ to be the word $w[\,i\,]\,w[\,i{+}1\,]\ldots w[\,j\,]$ with the conventions $w(\,i,j\,) \equiv \varepsilon$ if $i>j$, and $w(\,i,j\,) \equiv w(\,i,|w|\,)$ if $j>|w|$. We refer to $w(\,i,j\,)$ as being a *subword* of $w$. It is a *proper subword* of $w$ if $|w| \geq 1$ and $1<i$ or $j<|w|$. It is a *prefix* of $w$ if $i=1$, it is a *suffix* of $w$ if $j=|w|$.

Let

$$(1)\ \Lambda : S \cup \{\varepsilon\} \longrightarrow \mathbb{N}$$

be a map satisfying

$$(2)\ \Lambda(\varepsilon)=0.$$

Then we define the map $\Lambda^* : S^* \longrightarrow \mathbb{N}$ by:

$$\Lambda^*(s_1\, s_2 \ldots s_K) = \sum_{i=1}^{i=K} \Lambda(s_i)\,,$$

where every $s_i$ belongs to S. We will refer to the map $\Lambda^*$ as being a *length function* (on the words of $S^*$). When $\Lambda(s)=1$, for all $s \in S$, then $\Lambda^* = |\,|$ and, unless explicitly stated to the contrary, all future references to 'word length' are assumed to refer to this length function.

Let $\Lambda$ be a map satisfying (1) and (2), and let $>$ be an ordering of the symbol set S, then we define the (total) ordering $>_\Lambda$ on the words of $S^*$ by: $w >_\Lambda v$ if $\Lambda^*(w) > \Lambda^*(v)$, or if

$\Lambda^*(w) = \Lambda^*(v)$ and $w[i] > v[i]$ where i is the first position in which the words differ. We refer to $>_\Lambda$ as a *shortest word / lexicalgraphical* ordering (of $S^*$) (and we write $w \geq_\Lambda v$ when $w >_\Lambda v$ or $w = v$). If $\Lambda^* = |\;|$, then $>_\Lambda$ is called a *ShortLex* ordering (of $S^*$).

### *Presentations of Monoids and Groups.*

If C is a set of symbols and $\mathcal{R}$ is a subset of $C^* \times C^*$, then we write $\langle \mathcal{R} \rangle$ for the congruence generated by $\mathcal{R}$. Also, we will write left($\mathcal{R}$) = {left components of the ordered pairs in $\mathcal{R}$}, and right($\mathcal{R}$) = {right components of the ordered pairs in $\mathcal{R}$}. We refer to $\langle C \mid \mathcal{R} \rangle$ as being a *presentation* of the monoid $M = C^*/\langle \mathcal{R} \rangle$ (or any monoid which is isomorphic to M). We refer to C as a set of *generators* of M, and refer to $\mathcal{R}$ as a set of *defining relations* of M. The *natural* homomorphism, $\gamma : C^* \longrightarrow M$, is defined by $\gamma(c) = c\langle \mathcal{R} \rangle$, for all $c \in C$, where $c\langle \mathcal{R} \rangle$ denotes the $\langle \mathcal{R} \rangle$-congruence class of c. We wish to stress that, with this terminology, a group presentation, $\langle C \mid \mathcal{R} \rangle$, must be a presentation as a semigroup, i.e., whenever $c \in C$, then there is a $c^{-1} \in C$ with both $(cc^{-1}, \varepsilon)$ and $(c^{-1}c, \varepsilon)$ belonging to $\mathcal{R}$.

All groups and monoids will be finitely generated (although not necessarily finitely presented). If G is a group and C is a generating set of G, then, unless explicitly stated to the contrary, C will be assumed *inverse closed*, i.e, $c \in C \Rightarrow c^{-1} \in C$ (so that C generates G as a semigroup). If $g \in G$, then we define $\|g\|_C$, the *norm* of g, to be the minimum number of generators needed to express g as a product in the generators of C.

If $\langle C \mid \mathcal{R} \rangle$ is a presentation of the monoid M and $v\langle \mathcal{R} \rangle w$, then the *relation* $(w, v)$ of M may sometimes be written as $v =_M w$. We wish to stress the distinction between $v =_M w$, which means $v\langle \mathcal{R} \rangle w$, as opposed to (the much stronger) $v = w$, which means that $v$ and $w$ are identical words (of $C^*$).

### *Cayley Graphs.*

Let G be a group with (inverse closed) generating set C. Then $\Gamma_C = \Gamma_C(G)$ will denote the Cayley graph of G with respect to the generators C, i.e., $\Gamma_C(G)$ is the directed, labelled graph with vertex set G, and a directed edge from g to h, labelled by $c \in C$, if and only if $gc = h$. We may refer to the vertex $1_G$ of $\Gamma_C(G)$ as being the *basepoint* of the Cayley graph.

We define a metric, $d_C$, on the whole of $\Gamma_C(G)$ as follows. If g and h are two vertices of $\Gamma_C(G)$, then $d_C(g,h)$ is the minimum number of edges needed to connect g to h. We then define $d_C$ on each edge joining distinct vertices by making that edge isometric to the unit interval, and for each edge with the same endpoints, we divide the edge midway and make each half interval isometric to the half unit interval. Whence $d_C$ extends uniquely to the (standard) *path metric* on the whole $\Gamma_C(G)$. We will usually drop the subscript 'C' from $\Gamma_C(G)$, $d_C$, et al., when there is no risk of ambiguity (as now).

We wish to stress that we are thinking of the *whole* of $\Gamma_C(G)$ as a connected (path) metric space. We allow retracing of paths, and a path is said to be a *geodesic* path if it is a shortest path between its endpoints. We note the following facts. If $g\in G$, then $d(1_G,g)=\|g\|$ and, if $g,h\in G$, then $d(g,h)=\|gh^{-1}\|$. The vertices are distinguished by the fact that they are precisely those points of $\Gamma$ at integer distances from the basepoint, actually, if $n\in\mathbb{N}$, then S(n) consists precisely of those $g\in G$ with norm n. When we refer to an edge or path *staying within* a ball B(r) ($r\in\mathbb{R}$), we mean that all points of the edge or path lie in B(r). Whence, an edge stays within B(r) if and only if at least one of its endpoints is in B(r−1), a path connecting two vertices stays within B(n) ($n\in\mathbb{N}$) if and only if at least one of the end points of *every* edge that it traverses belongs to S(n−1).

If $\rho$ is a path of $\Gamma_C(G)$, then its length is denoted by $|\rho|$ and the path which traverses $\rho$ in the opposite direction is denoted by $\rho^{-1}$. We say that paths $\rho_0$ and $\rho_1$, with a common basepoint, do not *diverge by more than a distance $\Delta\in\mathbb{R}$* if (i) and (ii) hold as follows. (i) the lengths of $\rho_0$ and $\rho_1$ do not differ by more than $\Delta$. (ii) for all $0\leq r\leq$ lengths of $\rho_0$, $\rho_1$: if $p_0$ and $p_1$ are the points at distances of r along $\rho_0$ and $\rho_1$, respectively, then $d(p_0,p_1)\leq\Delta$.

We refer the interested reader to the comprehensive study of Cayley graphs (and several other subjects mentioned in this thesis) in (Lyndon,Schupp).

### *Finite State Automata.*

We shall abbreviate *finite state automata(automaton)* to *fsa*, and adopt some of the finite state automata terminology of (CHEPT).

*1.1 Definition.*

A (partial) *deterministic finite state automaton* is a quintuple $A=(C,S,s,H,\tau)$ where:

C is a finite set called the *alphabet*.

S is the finite set of *states*.

$s \in S$ is the *start state*.

H is a subset of S consisting of the *halt* (sometimes *success* ) *states*.

$\tau$ is a partial function, $S \times C \longrightarrow S$, called the *transition function*.

The partial transition function is extended to a partial function $\tau : S \times C^* \longrightarrow S$ (inductively) as follows. We define $\tau(p,\varepsilon)=p$, then, for $w \in C^*$, $c \in C$, and provided $\tau(p, w)$ is defined, we define $\tau(p, wc) = \tau( \tau(p, w) , c )$. When $p=s$, we abbreviate $\tau(s, w)$ to $\tau(w)$, we then define the *language* of A to be $\{ w \in C^* \mid \tau(w) \in H \}$, which is written as lan(A).

$\boxed{1.1}$

*1.2 Definition.*

A (partial) non-deterministic finite state automaton is a quintuple $A=(C,S,s,H,\mathcal{ARROWS})$ where:

C is a finite set called the *alphabet*.

S is the finite set of *states*.

$s \in S$ is the *start state*.

H is a subset of S consisting of the *halt* (or *success* ) *states*.

$\mathcal{ARROWS}$ is a subset of $S \times C \times S$ consisiting of the set of *arrows*.

Let $(s_0,c,s_1)$ be an arrow; then the state $s_0$ is the *source* of the arrow, the symbol c is the *label* of the arrow, and the state $s_1$ is the *target* of the arrow. A *path* of arrows is a finite, non empty, sequence of arrows $a_1,a_2\, a_3 ,\ldots,a_n$ with, for all $1 \leq i < n$, the target of the arrow $a_i$ being the source of the arrow $a_{(i+1)}$. The *source* of the path is the source of the arrow $a_1$, the *target* of the path is the target of the arrow $a_n$, and the label of the path is the word $c_1c_2c_3 \ldots c_n$ where, for all $1 \le i \le n$, $c_i$ is the label of the arrow $a_i$. We can now define the *language* of A to be $\{ w \in C^* \mid w$ is the label of a path of arrows with source s, and with target belonging to H$\}$. The language of A is written lan(A).

$\boxed{1.2}$

Deterministic and non-deterministic fsa can be (helpfully) realised as finite directed graphs called *state diagrams*. Let $A=(C,S,s,H,\tau)$ be a deterministic fsa, then A's *state diagram* is the directed, labelled graph defined as follows. The vertices of the state diagram are the states of A, and there is a directed a edge from state $s_0$ to state $s_1$, labelled by $c \in C$, if and only if $\tau(s_0,c)$ is defined and equal to $s_1$. With this, more natural realization of A, we can define the language of A as being the labels of all those paths beginning at the start state and ending at a halt state.

Let $A=(C,S,s,H,\mathcal{ARROWS})$ be a non-deterministic fsa, then A's *state diagram* is the directed, labelled graph defined as follows. The vertices of the state diagram are the states of A, and there is a directed a edge from state $s_0$ to state $s_1$, labelled by $c \in C$, if and only if $(s_0,c,s_1)$ is an arrow. We could then define the language of A as being the labels of all those paths beginning at the start state and ending at a halt state. Also, we see that the deterministic fsa are special non-deterministic fsa which have state diagrams with at most one directed edge labelled by each $c \in C$ starting from each vertex.

We do not assume familiarity with regular languages, but all of the following facts will be implicitly referred to at one point or another of the thesis. We refer the reader to (Salomaa) for proofs of the (mostly) standard results, and to (Rayward Smith) for an introduction to finite state automata.

A subset of $C^*$ is the language of a deterministic fsa (with alphabet C) if and only if it is the language of a non-deterministic fsa (with alphabet C). If $\mathcal{LAN}$ is the language of a fsa with alphabet C, then is said to be a *regular* (sometimes *recognizable*) subset (or language) of $C^*$. If $\mathcal{LAN}$ is a regular subset of $C^*$, then so is its complement, i.e., $C^*-\mathcal{LAN}$. The class of regular subsets of $C^*$ is also closed under unions, intersections and concatenation (i.e., if $\mathcal{LAN}_1$ and $\mathcal{LAN}_2$ are regular, then so is $\mathcal{LAN}_1\mathcal{LAN}_2$ consisting of all those words $\zeta_1\zeta_2$ where $\zeta_1 \in \mathcal{LAN}_1$ and $\zeta_2 \in \mathcal{LAN}_2$).

All finite subsets of $C^*$ are regular; if $\mathcal{S}$ is a regular subset of $C^*$, then the set of words in $C^*$ which do not contain words of $\mathcal{S}$ as subwords is also a regular subset. We shall refer to the following definition and example in both theorem 2.1.5 (abelian groups are automatic) and (implicitly) in theorem 5.4 (abelian by finite groups are almost convex).

*1.3 Definition.*

If $\mathcal{LAN}$ is a regular subset, then $\mathcal{LAN}$ is said to be prefix closed if, whenever $\ell \in \mathcal{LAN}$, then, also, $p \in \mathcal{LAN}$ for all prefixes $p$ of $\ell$.

$\boxed{1.3}$

*1.4 Lemma.*

Let $\{ c_1, c_2, \ldots, c_k \}$ be a subset of C, and let $n_1, n_2, \ldots, n_k \in \mathbb{N}$. Then the subset of $C^*$ consisting of all those words of the form $(c_1)^* (c_2)^* \ldots (c_k)^*$ which do not contain more than $n_i$ $c_i$'s (for $1 \leq i \leq k$) is a prefix closed regular subset.

$\boxed{1.4}$

# *§2*

# *Automatic Groups*

## *2.0*

The concept of *automatic groups*, originally suggested by W.P. Thurston, is relatively new (85), but there is already much research in the topic (primarily because of its applications to certain topological problems).

This chapter is meant as a concise introduction to the theory of automatic groups, stating the basic properties and including a short summary. The emphasis is squarely on results which will be referred to later and there is only one major theorem proved, i.e, that the abelian groups are automatic (a prerequisite of theorem 5.4.1, i.e. abelian by finite groups are almost convex). There is a comprehensive paper 'Word Processing and Group Theory', (CEHPT), a collaboration of five authors, which details most of the current work in automatic groups and to which we refer the interested reader. We will be referring to this paper frequently. This chapter includes no original work.

## *2.1*

*2.1.1 Definition (CEHPT).*

Let G be a group, C be a finite (semigroup) generating set of G, and fix a symbol, $\mathit{1}$, which is not in C. Let $\gamma : (C \cup \{ \mathit{1} \})^{*} \longrightarrow G$ be the natural homomorphism which maps $\mathit{1}$ to the identity element of G. We say that (G,C) is an *automatic group* if there are automata W and $\{M^{(c)}\}_{c \in C \cup \{ \mathit{1} \}}$ so that:

*(i)* W has alphabet C, and the restriction of the map $\gamma$ to $\mathrm{lan}(W) \longrightarrow G$ is surjective.

With $c \in C \cup \{ \mathit{1} \}$:

*(ii)* $M^{(c)}$ has alphabet $(C \cup \{ \mathit{1} \}) \times (C \cup \{ \mathit{1} \})$.

*(iii)* $\mathrm{lan}(M^{(c)}) = \Big\{ (w_1, w_2) \mid w_1$ and $w_2$ are words of the regular language $\mathrm{lan}(W)\,\mathit{1}^{*}$, and $\gamma(w_1 c) = \gamma(w_2) \Big\}$.

If (G,C) is automatic (with (i),(ii) and (iii) holding) then W is called the *word acceptor* (of (G,C)), and the $M^{(c)}$ are called the *multiplication automata* (of (G,C)).

$\boxed{2.1.1}$

There is a more accessible characterization of the automatic groups which we will be referring to, but we need first to introduce the concept of *word differences.*

*2.1.2 Definition (CEHPT).*

Let G be a group and C be a finite semigroup generating set of G, let $\gamma : C^* \longrightarrow G$ be the natural homomorphism. Let W be an automaton with alphabet C, with the restriction of the map $\gamma$ to $\text{lan}(W) \longrightarrow G$ being surjective. Then we define the set of *word differences*, of (any) $c \in C$, to be:

$$\Big\{ \gamma(w_1(1,r))^{-1}\, \gamma(w_2(1,r)) \mid \text{ for all } r \in \mathbb{N}, \text{ and all } w_1 , w_2 \in \text{lan}(W) \text{ satisfying } \gamma(w_1 c) = \gamma(w_2) \Big\}.$$

$\boxed{2.1.2}$

We then have the following, oft referred to, theorem:

*2.1.3 Theorem (CEHPT).*

Let (G,C), $\gamma$ and W be as in definition 2.1.2. Then (G,C) will be automatic, with word acceptor W, if and only if the set of word differences of each $c \in C$ is finite (see (CEHPT)).

$\boxed{2.1.3}$

We should note the (rather nice) geometrical interpretation of 2.1.3, i.e.: (G,C) is automatic, with word acceptor W, if and only if there is a number k with the property that, if $w_1 , w_2 \in$ lan(W), $c \in C$ and $w_1 c =_G w_2$, then the paths of $\Gamma_C(G)$, beginning at the basepoint, and labelled by $v$ and $w$, respectively, do not diverge by more than a distance k.

We can now prove:

*2.1.4 Corollary.*

Let (G,C) be an automatic group with word acceptor W, and take any $g \in G$. Then there is a number $\Delta$ with the property that, if $w_1, w_2 \in \mathrm{lan}(W)$, and $\gamma(w_1)g = \gamma(w_2)$, then the paths of $\Gamma_C(G)$ beginning at the basepoint and labelled by $w_1$ and $w_2$, respectively, do not diverge by more than a distance $\Delta$.

*Proof:*

Let $g = c_1 c_2 \ldots c_n$ (as a product in the generators of C). Then, with $p_1 = w_1$ and $p_{n+1} = w_2$, choose $p_i \in \mathrm{lan}$ (W) so that, for $1 \leq i \leq n$, $p_i c_i =_G p_{i+1}$. By theorem 2.1.3, the paths beginning at the basepoint and labelled by $p_i$ and $p_{i+1}$ ($1 \leq i \leq n$) do not diverge by more than a distance k. Thus the paths beginning at the basepoint and labelled $w_1$ and $w_2$ do not diverge by more than a distance $\Delta = nk$.

$\boxed{2.1.4}$

This is a summary of some of the major results on automatic groups. The reader will find proofs of all these results in (CEHPT).

The property of a group being automatic is independent of the choice of generators and all automatic groups are finitely presented with solvable word problem. Computers are apt tools for the study of (and, in particular, the construction of) automatic groups, but this is not surprising for they were defined with this in mind. There are procedures which terminate if a (finitely presented) group is automatic (the development of such procedures is, naturally, of particular concern to researchers).

The class of automatic groups is closed under direct products, extensions by finite groups, free products with finite amalgamated subgroups and HNN extensions over finite subgroups. The finite, free and abelian groups are all automatic. More recently it has been proved that the braid groups are automatic. Torsion free, non abelian nilpotent groups are not automatic.

We will prove only one major result in this chapter, i.e., that the abelian groups are automatic. This proof is included because it is a prerequisite of theorem 5.4.1 (abelian by finite groups are almost convex).

*2.1.5 Theorem (CEHPT).*

Let G be an abelian group with $C=\{c_1, c_2, \ldots, c_k\}$ a finite set of (semigroup) generators of G (i.e., if $c\in C$ then $c^{-1}\in C$). Let $>$ be an (arbitrary) ordering of C, and let $\Lambda^*$ be an (arbitrary) length function on the words of $C^*$. Let $\mathcal{LEX}$ be the subset of $C^*$ consisting of the $>_\Lambda$ least words corresponding to each group element. Then (G,C) is automatic with $\mathcal{LEX}$ being the (prefix closed) language of the word acceptor (we refer the reader to definition 1.3 of a prefix closed language).

*Proof:*

If $w$ is a word of $C^*$, we will write rep($w$) for the $>_\Lambda$ least word with $w =_G \text{rep}(w)$ (so that $\mathcal{LEX} = \{\text{rep}(w) \mid w\in C^*\}$).

We may as well assume $c_1 < c_2 < \ldots < c_k$. We will also suppose all words to be words of the regular expression:

$$\textit{(1)}\ \ (c_1)^* (c_2)^* \ldots (c_k)^*,$$

and redefine concatenation of words to be the product of those words in the free abelian monoid on the generators C expressed in the normal form of (1).

If $v$ and $w$ are words, then we say that *$v$ divides $w$*, and write $v \mid w$, if every generator which occurs in $v$ also occurs in $w$ to at least the same degree. If

$$v = (c_1)^{n_1}(c_2)^{n_2} \ldots (c_k)^{n_k}\ \ ,\ \ w = (c_1)^{\acute{n}_1}(c)^{\acute{n}_2} \ldots (c_k)^{\acute{n}_k}$$

(with $n_1, \ldots, n_k, \acute{n}_1, \ldots, \acute{n}_k \in \mathbb{N}$), and $v \mid w$, then $w/v$ is defined to be the word:

$$(c_1)^{(\acute{n}_1 - n_1)}(c_2)^{(\acute{n}_2 - n_2)} \ldots (c_k)^{(\acute{n}_k - n_k)}.$$

Note that, with these conventions, if $v \mid w$, then $w = v(\ w/v\ )$.

*(2) Claim.*

If $w\in \mathcal{LEX}$, then $s\in \mathcal{LEX}$ whenever $s$ divides $w$ (so, in particular $\mathcal{LEX}$ is prefix closed).

*Proof:*

Suppose, for a contradiction, that this claim is false. Then we may define $s$ to be the least non-empty word which divides $w$ but which is not a member of $\mathcal{LEX}$.

We will have $w = s(\, w/s)$, and so, if $\Lambda^*(\, s\,) > \Lambda^*(\mathrm{rep}(\, s\,))$, then $\Lambda^*(w) = \Lambda^*(\, s\,(\, w/s\,)\,) > \Lambda^*(\,\mathrm{rep}(s)\,(\, w/s\,)\,)$ while $w =_G \mathrm{rep}(s)\,(\, w/s\,)$ – which would contradict the fact that $w \in \mathcal{LEX}$.

So it must be that $\Lambda^*(s) = \Lambda^*(\mathrm{rep}(s))$, and thus $s[1] > \mathrm{rep}(s)[1]$ (because we took $s$ to be the least non-empty word which divides $w$ but which is not in $\mathcal{LEX}$).

Let $p$, possibly empty, be the largest prefix of $w$ with no generators in common with $s$ (we refer to the definition of prefix given on page 4, i.e., with *no reordering* of generators). Then the generator $w[\,|p|+1\,]$ must occur in $s$, and so:–

$$(3)\quad w[\,|p|+1\,] \geq s[1] > \mathrm{rep}(s)\,[\,1\,].$$

Since $p$ and $s$ both divide $w$ and have no generators in common, so $ps \mid w$, and therefore $w = ps\,(\, w/(ps)\,)$. Whence:–

$$w =_G ps\,(\, w/(ps)\,) =_G p\,\mathrm{rep}(\, s\,)\,(\, w/(ps)\,),$$

and

$$(4)\quad \Lambda^*(w) = \Lambda^*(\, ps(\, w/(ps)\,)\,) \geq \Lambda^*(\, p\,\mathrm{rep}(s)\,(\, w/(ps)\,)\,).$$

As $p$ is a prefix of $w$, so, by (3) and (4), $w >_\Lambda p\,\mathrm{rep}(s)\,(\, w/(ps)\,)$ – while $w =_G p\,\mathrm{rep}(s)\,(\, w/(ps)\,)$ – which (again) contradicts the fact that $w \in \mathcal{LEX}$.

$\boxed{2}$

We say that a relation of G, $(\, w\,,\, v\,)$, is minimal if $w >_\Lambda v$ and there is no other relation, $(\,\bar{w}, \bar{v})$, with $\bar{w} >_\Lambda \bar{v}$ and $\bar{w} \mid w$ and $\bar{v} \mid v$. Let $\mathcal{M}$ be the set of minimal relations.

*(5) Claim.*

$w \notin \mathcal{LEX}$ if and only if $w$ has a subword belonging to left( $\mathcal{M}$ ).

*Proof:*

If $w$ had a subword belonging to left( $\mathcal{M}$ ), then this subword could not belong to $\mathcal{LEX}$ and so, by (2), $w$ could not belong to $\mathcal{LEX}$. Conversely, if $w$ did not belong to $\mathcal{LEX}$, then we could define $s$ to be the least subword of $w$ not belonging to $\mathcal{LEX}$. Then every subword of $s$ would belong to $\mathcal{LEX}$, and so $(\, s\,,\, \mathrm{rep}(s)\,)$ would be a minimal relation, in particular, $s \in$ left( $\mathcal{M}$ ).

$\boxed{5}$

If n and ń are κ-tuples of non-negative integers, then we will write $n \leq ń$ if $n(j) \leq ń(j)$ for all $1 \leq j \leq \kappa$ (where n(j) is the $j^{th}$ component of n, and ń(j) is the $j^{th}$ component of ń).

*(6) Claim.*

If $n_1, n_2, n_3, \ldots$ is an infinite sequence of κ-tuples of non-negative integers, then $n_r \leq n_s$ for some $r \leq s$.

*Proof:*

By induction on κ. If $\kappa = 1$ then the problem is trivial. So suppose that, for every $i > 1$, $n_1 \not\leq n_i$. Then, for every $i \geq 1$, there is an s, depending on i, with $n_i(s) \leq n_1(s)$. So (for some s $(1 \leq s \leq \kappa)$) there must be an infinite subsequence, $n_{i_1}, n_{i_2}, n_{i_3}, \ldots$ say, with any two of these κ-tuples having equal s components. We now apply the inductive hypothesis to this subsequence to complete the proof.

$\boxed{6}$

*(7) Claim.*

The set of minimal relations, $\mathcal{M}$, is finite.

*Proof:*

If not, then $\mathcal{M}$ consists of an infinite set of relations:

$$\left( (c_1)^{n_{i,1}} (c_2)^{n_{i,2}} \ldots (c_k)^{n_{i,k}} , (c_1)^{n_{i,(k+1)}} (c_2)^{n_{i,(k+2)}} \ldots (c_k)^{n_{i,2k}} \right),$$

say, with the $n_{i,j} \in \mathbb{N}$ for all $i \geq 1$ and $1 \leq j \leq 2k$.

By (6), there would exist r and s so that:-

$$( n_{r,1} ,\ldots, n_{r,k} , n_{r,(k+1)} ,\ldots, n_{r,2k} ) \leq ( n_{s,1} ,\ldots, n_{s,k} , n_{s,(k+1)} ,\ldots, n_{s,2k} ).$$

Whence:-

$$(c_1)^{n_{r,1}} (c_2)^{n_{r,2}} \ldots (c_k)^{n_{r,k}} \mid (c_1)^{n_{s,1}} (c_2)^{n_{s,2}} \ldots (c_k)^{n_{s,k}} ,$$

and

$$(c_1)^{n_{r,(k+1)}} (c_2)^{n_{r,(k+2)}} \ldots (c_k)^{n_{r,2k}} \mid (c_1)^{n_{s,(k+1)}} (c_2)^{n_{s,(k+2)}} \ldots (c_k)^{n_{s,2k}} ,$$

and so the relation

$$( (c_1)^{n_{s,1}} (c_2)^{n_{s,2}} \dots (c_k)^{n_{s,k}} , (c_1)^{n_{s,(k+1)}} (c_2)^{n_{s,(k+2)}} \dots (c_k)^{n_{s,2k}} )$$

would not be minimal.

$\boxed{7}$

By (2) and (7), $w \in \mathcal{LEX}$ if and only if $w$ belongs to the regular language (1) and is not divisible by any word in (finite) left( $\mathcal{M}$ ). So, by lemma 1.4, $\mathcal{LEX}$ is also a (prefix closed) regular language. Thus, to complete the proof of theorem 2.1.5, we need only prove that, for any $c \in C$, the set of word differences of c,

$$(8) \left\{ \gamma( w_1(1,r))^{-1} \gamma( w_2(1,r)) \mid \text{ for all } r \in \mathbb{N}, \text{ and all } w_1 , w_2 \in \text{lan}(W) \right.$$
$$\left. \text{satisfying } \gamma(w_1 c) = \gamma(w_2) \right\},$$

is finite.

So take any $v, w \in \mathcal{LEX}$ satisfying $wc =_G v$. We may as well assume that $v$ does not contain the generator c (otherwise $w = v / c$, because both words belong to $\mathcal{LEX}$ and correspond to the same group element ). We cancel the common generators of $w$ and $v$ to derive the relation $w_1 c =_G v_1$ with $w_1 \mid w$, $v_1 \mid v$ and $w_1 c$ and $v_1$ having no common generators. It is easy to see that (8) will be finite if we can prove the following:

*(9) Claim.*

The relation ( $w_1 c$ , $v_1$ ) is one of the (finite number of) minimal relations.

*Proof:*

Suppose ( $w_2$ , $v_2$ ) is a relation with $w_2 >_\Lambda v_2$ , $w_2 \mid w_1$ c, and $v_2 \mid v_1$. The word $w_2$ is not a member of $\mathcal{LEX}$ and so it cannot be a subword of $w_1$. Thus $w_2$ must contain the generator c, and we have ( $(w_1 c) / w_2$ ) $\mid$ $w_1$. So the words $(w_1 c) / w_2$ and $v_1 / v_2$ both belong to $\mathcal{LEX}$ and correspond to the same group element, i.e., . $(w_1 c) / w_2 = v_1 / v_2$ . As $w_1$ c and $v_1$ have no generators in common, so it must be that $w_1 c \equiv w_2$ and $v_1 \equiv v_2$ , i.e., ( $w_2$ , $v_2$ ) is the relation ( $w_1 c$ , $v_1$ ).

$\boxed{\text{9 and 2.1.5}}$

# *§3*

# *Monoids with Complete and Parameterized Presentations*

## *3.0*

*Complete presentations* are a small part of the computer theoretical study of *rewriting techniques*, which is a far reaching and, potentially, powerful theory with notable applications in proof verification of algebraic theories.

We will be restricting our study to that of monoids with complete presentations (which have simple, and fast, solutions to their word problems). In this chapter we will define a class of infinite (complete) presentations, which have been referred to by several authors, but which we will be calling *r-parameterized*. As far as group presentations are concerned, the (1-)parameterized presentations are, arguably, the simplest non-finite presentations we could hope to define, but we will be proving that completeness is not, in general, decidable for parameterized monoid presentations. In subsequent chapters (4 and 5 respectively) we describe a computer program for *completing* 1-parameterized group presentations, and prove that the class of the groups defined by r-parameterized complete presentations (with *word length preserving* orderings) have *almost convex* Cayley graphs.

We recommend the comprehensive account of the history, and major theorems of, rewriting techniques in the expository paper 'History and basic features of the critical-pair/completionprocedure' (Buchberger). Also, we believe (Book), (Jantzen), and ($^2$Kapur,Narendon) may be of interest to the reader.

## *3.1*

Let C be a (fixed) finite set and let $\mathcal{D}$ be a (fixed) subset of $C^*{\times}C^*$. Let $>$ and $\geq$ denote a (fixed) well ordering of $C^*$ so that, for all $u, v, w \in C^*$:

$$uv \geq u; \quad vu \geq u; \quad w > v \Rightarrow wu > vu \text{ and } uw > uv.$$

Such orderings are sometimes referred to as Knuth-Bendix orderings, and we will adopt this terminology.

Let $\mathcal{R}$ be a subset of $C^* \times C^*$. We define the relation $\rightarrow_{\mathcal{R}}$ on the words of $C^*$ by $w \rightarrow_{\mathcal{R}} v$ if $w \equiv p b s$ and $v \equiv p a s$ for some $(b, a) \in \mathcal{R}$. We then write $\rightarrow_{\mathcal{R}}^*$ for the reflexive, transitive closure of $\rightarrow_{\mathcal{R}}$, and, if $w \rightarrow_{\mathcal{R}}^* v$, we say that $v$ is an *$\mathcal{R}$-descendant* of $w$. We write $w \bigvee_{\mathcal{R}} v$ when $w$ and $v$ have a common $\mathcal{R}$-descendant. A word is said to be *$\mathcal{R}$-irreducible* if it has no $\mathcal{R}$-descendants other than itself.

We drop the subscript ' $\mathcal{R}$' from $\rightarrow_{\mathcal{R}}$, $\rightarrow_{\mathcal{R}}^*$ and $\bigvee_{\mathcal{R}}$, and the prefix ' $\mathcal{R}$' from $\mathcal{R}$-descendant and $\mathcal{R}$-irreducible when there is no risk of ambiguity (as now).

We write (rather unimaginatively)

$$\text{left}(\mathcal{R}) = \{ \text{ left components of the ordered pairs in } \mathcal{R} \},$$

and

$$\text{right}(\mathcal{R}) = \{ \text{ right components of the ordered pairs in } \mathcal{R} \}.$$

We say that $\mathcal{R}$ is *normalized* if, whenever $(b, a) \in \mathcal{R}$, then $b > a$. If $\mathcal{R}$ is normalized, then the ordered pairs of words in $\mathcal{R}$ are called *rewrite rules*. If $(b, a) \in \mathcal{R}$ and $|b| = |a|$, then $(b, a)$ is said to be a *length preserving* rule of $\mathcal{R}$.

We say that $\langle C \mid \mathcal{R} \rangle$ is a *normalized presentation* of the monoid M if it is a presentation of M and $\mathcal{R}$ is normalized. If $\langle C \mid \mathcal{R} \rangle$ is a presentation of M, then we can always find a normalized presentation of M by discarding all those $(b, a) \in \mathcal{R}$ with $b = a$, and swapping $(b, a)$ for $(a, b)$ if $b < a$ (because this would not change the congruence $\langle \mathcal{R} \rangle$).

If $\mathcal{R}$ is normalized then $w \rightarrow v$ implies $w > v$ and $w \rightarrow^* v$ implies $w \geq v$, so every word would have at least one irreducible descendant (because > is a well ordering).

We say that $\mathcal{R}$ is *complete* if it is normalized and every word, $w$, in $C^*$ has a unique irreducible descendant (called the *$\mathcal{R}$-representative* of $w$ and denoted by $\text{rep}_{\mathcal{R}}(w)$). We say that $\mathcal{R}$ is *$\mathcal{D}$-complete* if it is complete and $\langle \mathcal{R} \rangle = \langle \mathcal{D} \rangle$. We say that $\langle C \mid \mathcal{R} \rangle$ is a *complete presentation* of the monoid M if it is a presentation of M and $\mathcal{R}$ is complete.

It is easy to prove that, if $\mathcal{R}$ is $\mathcal{D}$-complete and $w \langle \mathcal{D} \rangle v$ then $\text{rep}(w) = \text{rep}(v)$ (= least word in the $\langle \mathcal{D} \rangle$ congruence class of $v$ and $w$). So then the representatives would be normal forms for the elements of $M = C^* / \langle \mathcal{D} \rangle$ in the generators C (and, being the irreducible words, being the words with no subword in left( $\mathcal{R}$ ), constitute a regular subset of $C^*$ if and only if left( $\mathcal{R}$ ) is a regular subset of $C^*$ ).

If $\mathcal{R}$ is complete and recursive (as a subset of $C^*{\times}C^*$) then the representative of a word, $w$, is computable. We look for subwords, $s$ , of $w$ for which there is a rule ( $s$ , $r$ ) belonging to $\mathcal{R}$. There may be no such no subwords, but, if one exists, then the subword $s$ is replaced by $r$, and the process repeated until no more substitutions can be made. The resulting ($\mathcal{R}$-irreducible) word will be the representative of $w$. In particular, the word problem for (M,C) will be solvable.

Let J( $\mathcal{D}$ ) be the set of words which are not least in their $\langle$ $\mathcal{D}$ $\rangle$ congruence class, but for which all proper subwords are least in their $\langle$ $\mathcal{D}$ $\rangle$ congruence classes. Then it is reasonably easy to prove:

*3.1.1 Proposition.*

If $\langle$ C | $\mathcal{R}$ $\rangle$ is normalized then it is $\mathcal{D}$-complete if and only if left( $\mathcal{R}$ )$\supseteq$J( $\mathcal{D}$ ). (We refer the reader to (Hayashi) for a proof of this (standard) result.)

$\boxed{3.1.1}$

We say that $\mathcal{R}$ is a *minimal complete* subset (of $C^*{\times}C^*$) if it is complete and left( $\mathcal{R}$ )=J( $\mathcal{R}$ ), $\langle$ C | $\mathcal{R}$ $\rangle$ is said to be a *minimal complete presentation* of the monoid M if it is a presentation of M and $\mathcal{R}$ is minimal complete. We say that $\mathcal{R}$ is a *minimal $\mathcal{D}$-complete* subset (of $C^*{\times}C^*$) if it is $\mathcal{D}$-complete and left( $\mathcal{R}$ )=J( $\mathcal{D}$ ).

It is not difficult to prove that there is a unique minimal $\mathcal{D}$-complete subset of $C^*{\times}C^*$ with respect to the fixed well ordering > (although, clearly, it may not computable), and that, if $\mathcal{R}$ is known to be $\mathcal{D}$-complete, then we can find the minimal $\mathcal{D}$-complete subset by discarding all those ( $b$ , $a$ )$\in$ $\mathcal{R}$ where $b$ has a proper subword in left( $\mathcal{R}$ ).

It is unlikely that an arbitrary presentation will be complete, but there is a practical criterion for telling us when this is so - but to describe this we need first to define *critical pairs.*

Suppose $\mathcal{R}$ is normalized and take $((b_1, a_1), (b_2, a_2))$ to be an ordered pair of rules in $\mathcal{R}$. Then the *critical pairs* of $((b_1, a_1), (b_2, a_2))$ are all those pairs of words:

*(i)* $(a_1, pa_2s)$ with $b_1 = pb_2s$, for some words $p$ and $s$.

*(ii)* $(a_1s, pa_2)$ with $b_1s = pb_2$, for some words $\varepsilon \neq p \neq b_1$ and $\varepsilon \neq s \neq b_2$.

A *critical pair of* $\mathcal{R}$ is a critical pair of some ordered pair in $\mathcal{R}$. We say that a critical pair, $(w, v)$, of $\mathcal{R}$ is *resolved* if the words $w$ and $v$ have a common descendant. Then we have the following, well known, result.

*3.1.2 Lemma (The Knuth-Bendix lemma).*

Let $\mathcal{R}$ be a normalized subset of $C^* \times C^*$, then $\langle C \mid \mathcal{R} \rangle$ is complete if and only if all the critical pairs of $\mathcal{R}$ are resolved.

$\boxed{3.1.2}$

We refer the reader to the original 1967 paper of (Knuth,Bendix), for historical interest; to (Huet) for a, reputably, good presentation of the Knuth-Bendix procedure; but, for a proof more suited to our (restricted) study, we recommend the proof of Gilman in (Gilman79).

So, for example, checking the completeness of finite presentations is purely mechanical. By far the most commonly used Knuth-Bendix orderings are the shortest word/ lexicalgraphic orderings. We will be using these orderings almost invariably in theory and in practice. In all the examples the Knuth-Bendix ordering is the ShortLex ordering defined by the stated lexicalgraphical ordering on the generators.

We shall refer to the next two examples, 3. 1.3 and 3.1.4, of finite complete presentations.

*3.1.3 Example.*

The free abelian group of rank 2 with generators $\{a < a^{-1} < b < b^{-1}\}$ has a finite (minimal) complete presentation:

$$\langle\, a, a^{-1}, b, b^{-1} \mid (aa^{-1}, \varepsilon), (a^{-1}a, \varepsilon), (bb^{-1}, \varepsilon), (b^{-1}b, \varepsilon), (ba, ab), \\ (ba^{-1}, a^{-1}b), (b^{-1}a, ab^{-1}), (b^{-1}a^{-1}, a^{-1}b^{-1}) \,\rangle.$$

$\boxed{3.1.3}$

If $\langle C \mid \mathcal{R} \rangle$ is a finite complete presentation then the set of representatives will be a regular subset of $C^*$. So it was suggested that the groups defined by finite, complete presentations might be automatic with word acceptor accepting the set of representatives - but this is not always true, 3.1.4 is a counterexample.

*3.1.4 Lemma (CEHPT).*

Let G be the wreath product of the infinite cyclic group with the cyclic group of order 2, then G has a finite complete presentation, but is not automatic with the word acceptor accepting the set of representatives.

*Proof:*

The group G has presentation:

$$\langle\ a, b, c \mid aa = 1,\ cb = bc,\ ba = ac\ \rangle,$$

and is an automatic group (being the extension of a free abelian group of rank 2 by the cyclic group of order 2).

With semigroup generators $C = \{ a < b < b^{-1} < c < c^{-1} \}$, it is easy to confirm that G has a finite complete (semigroup) presentation, $\langle C \mid \mathcal{R} \rangle$, with:

$$\mathcal{R} = \Big\{ (aa, \varepsilon), (bb^{-1}, \varepsilon), (b^{-1}b, \varepsilon), (cc^{-1}, \varepsilon), (c^{-1}c, \varepsilon), (cb, bc), (cb^{-1}, b^{-1}c), (c^{-1}b, bc^{-1}), (c^{-1}b^{-1}, b^{-1}c^{-1}), (ba, ac), (ca, ab), (b^{-1}a, ac^{-1}), (c^{-1}a, ab^{-1}), (c^{-1}b^{-1}, b^{-1}c^{-1}) \Big\}.$$

After a few trivial reductions, we see that, for all $n \in \mathbb{N}$, $a(b)^n(c)^n$ and $(b)^n(c)^n$ are $\mathcal{R}$-irreducible and:-

$$a(b)^n(c)^n a \rightarrow^*_{\mathcal{R}} (b)^n(c)^n.$$

So, with $\gamma : C^* \longrightarrow G$ being the natural homorphism, we have:-

$$\gamma(a(b)^n(c)^n a) = \gamma((b)^n(c)^n),$$

and, for all $r \in \mathbb{N}$, the group elements $\gamma(a(b)^r)^{-1}\gamma((b)^{(r+1)})$ belong to the set of word differences of $a$ (cf. definition 2.1.2).

A few more (trivial) reductions yield:-

$$\mathrm{rep}((a(b)^r)^{-1}(b)^{(r+1)}) \equiv a(b)^{(r+1)}(c^{-1})^r,$$

whence the set of word differences of $a$ is infinite. By theorem 2.1.3, G cannot be automatic with the word acceptor accepting the set of representatives.

$\boxed{3.1.4}$

*Comment.* If $\langle C \mid \mathcal{R} \rangle$ is a finite complete group presentation then $\mathcal{R}$ is sometimes said to admit no *backtracking* if, whenever a word $cbw$ is such that $c \in C$, $bw$ is $\mathcal{R}$-irreducible and $(cb, a) \in \mathcal{R}$ for some word $a$, then $aw$ is $\mathcal{R}$-irreducible.

If the Knuth-Bendix ordering is ShortLex, then the property of $\mathcal{R}$ admitting no backtracking can be formulated in terms of $\mathcal{R}$ alone and, if $\mathcal{R}$ admits no backtracking, then it is reasonably easy to prove that the group $G = C^* / \langle \mathcal{D} \rangle$ is automatic with the word acceptor accepting the language of the $\mathcal{R}$-representatives (the norm of all the word differences being no more than $\max(|b|)_{b \in \mathrm{left}(\mathcal{R})}$).

Actually, no backtracking, as we have defined it, is stronger than is needed to ensure that G be automatic. Nevertheless, such nice group presentations are not the norm and it is an interesting open problem as to whether an arbitrary finite complete group presentation, with respect to some ShortLex ordering, necessarily defines an automatic group.

Rewriting words with no backtracking is particularly fast, but we will not bother to comment any further on this subject (cf. (Le Chenadec) where examples are cited of such group presentations (the 2-dimensional surface groups)).

We will now formalize a class of monoid presentations which have been referred to by several authors (by Le Chenadec and Gilman (84), to name but two).

Let $\mathcal{B} \in (C^*)^{(2p+1)}$ (for some $p \in \mathbb{N}$), then we write $\mathcal{B}_i$ $(1 \leq i \leq 2p+1)$ for the i th component of $\mathcal{B}$. We define $\mathcal{B}(0)$ to be the word

$$\mathcal{B}_1 \mathcal{B}_3 \mathcal{B}_5 \cdots \mathcal{B}_{(2p+1)},$$

and, if $n = (n_1, n_2, \ldots, n_p) \in \mathbb{N}^p$, we define $\mathcal{B}(n)$ to be the word

$$\mathcal{B}_1 (\mathcal{B}_2)^{n_1} \mathcal{B}_3 (\mathcal{B}_4)^{n_2} \mathcal{B}_5 \ldots \mathcal{B}_{(2p-1)} (\mathcal{B}_p)^{n_p} \mathcal{B}_{(2p+1)}.$$

Provided $p > 0$, we may refer to the words $\mathcal{B}_2, \mathcal{B}_4, \ldots, \mathcal{B}_{2p}$ as the *repeating factors* of $\mathcal{B}$.

We will adopt the convention that $\mathbb{N}^0=\{0\}$, and note that, with this convention, if $\mathcal{B}\in(C^*)^0$ and $n\in\mathbb{N}^0$, then $\mathcal{B}(n)$ is always the single word $\mathcal{B}_1$.

We say that a subset, $\mathcal{R}$, of $C^*\times C^*$ is *r-parameterized*, or *of type* $P_r$, $(r\in\mathbb{N})$ if $\mathcal{R}$ can be partitioned as a *finite* number of subsets of the form:

$$(1)\quad \{\ (\ \mathcal{B}(n), \mathcal{A}(n)\ )\mid n\in\mathbb{N}^p\ \} \text{ with } 0\le p\le r \text{ and } \mathcal{B}, \mathcal{A}\in(C^*)^{(2p+1)}.$$

(We stress that we are *not* insisting the p of (1) to be the same for the different subsets of the partition, but r, being a bound on the p's, *is* a bound on the number of repeating factors allowed for the *different* $\mathcal{A}$'s and $\mathcal{B}$'s.)

Note that, with the convention that $\mathbb{N}^0=\{0\}$, the subsets of type $P_0$ are just the finite subsets (of $C^*\times C^*$).

Infinite monoid presentations have been studied by numerous authors. We shall refer to C. Ó' Dúnlaing 's work on *infinite regular Thue systems* (Ó' Dúnlaing) and C. Hayashi's work on *semi-confluent presentations* (Hayashi) – but the (1-)parameterized presentations are, arguably, the simplest non-finite group presentations we could hope to define. Let us look at a (trivial) example of a complete group presentation of type $P_1$.

*3.1.5 Example.*

The free abelian group of rank 2 with generators $\{\ a<b<a^{-1}<b^{-1}\ \}$ has a (minimal) complete presentation of type $P_1$:

$$\langle\ a, a^{-1}, b, b^{-1}\ \mid\ (\ a^{-1}a, \varepsilon\ ), (\ b^{-1}b, \varepsilon\ ), (\ ba, ab\ ),$$
$$(\ b^{-1}a, ab^{-1}\ ), (\ a^{-1}b, ba^{-1}\ ), (\ b^{-1}a^{-1}, a^{-1}b^{-1}\ ),$$
$$(\ a(b)^n a^{-1}, (b)^n\ )\ (n\in\mathbb{N}),\ (\ b(a^{-1})^n b^{-1}, (a^{-1})^n\ )\ (n\in\mathbb{N})\ \rangle.$$

$\boxed{3.1.5}$

However, 3.1.5 is not really of much interest because, with a reordering of the generators, there is the finite complete presentation of example 3.1.3. Actually, it is not that unusual for a group presentation, $\langle\ C\mid\mathcal{R}\ \rangle$, to be such that there are no finite $\mathcal{D}$-complete subsets with respect to some ShortLex ordering, but to possess finite $\mathcal{D}$-complete subsets after some ShortLex reordering of C (the Dyck groups and surface groups (of 4.5) to name but two). We will now demonstrate (with the help of the computer program described in chapter 4) that such beneficial reorderings are not always possible (also, cf (Bauer,Otto)).

*3.1.6 Lemma.*

Let G be the group defined by the (semigroup) presentation $\langle C \mid \mathcal{D} \rangle$ where:

$$C=\{ a , a^{-1} , b , b^{-1} , c , c^{-1} \},$$

and

$$\mathcal{D}= \Big\{ ( aa^{-1} , \varepsilon ) , ( a^{-1}a , \varepsilon ) , ( bb^{-1} , \varepsilon ) , ( b^{-1}b , \varepsilon ) , \\ ( cc^{-1} , \varepsilon ) , ( c^{-1}c , \varepsilon ) , ( ba , ab ) , ( ca , bc ) \Big\}.$$

Then there are (infinite) $\mathcal{D}$-complete subsets of type $P_1$ but no finite $\mathcal{D}$-complete subsets whatever the ShortLex ordering on $C^*$ .

*Proof:*

It is clear from the definition of $\mathcal{D}$ that it suffices to prove that there is no finite $\mathcal{D}$-complete subset with a ShortLex ordering < on $C^*$ for which $a < b$.

*Case 1.* Let < be any ShortLex ordering on $C^*$ for which

$$a< b \text{ and } a< c^{-1},$$

and let $\mathcal{R}$ be any $\mathcal{D}$-complete subset with respect to <.

The subset, $\mathcal{R}_1$ , (of $C^* \times C^*$), listed below, was generated by the computer program described in chapter 4. It is a (minimal) $\mathcal{D}$-complete subset of type $P_1$ with respect to the ShortLex ordering $<_1$ defined by $a <_1 a^{-1} <_1 b <_1 b^{-1} <_1 c <_1 c^{-1}$.

$$\begin{aligned} \mathcal{R}_1 = \Big\{ & ( aa^{-1} , \varepsilon ) , ( a^{-1}a , \varepsilon ) , ( bb^{-1} , \varepsilon ) , ( b^{-1}b , \varepsilon ) , \\ & ( cc^{-1} , \varepsilon ) , ( c^{-1}c , \varepsilon ) , ( ba , ab ) , ( ba^{-1} , a^{-1}b ) , \\ & ( b^{-1}a , ab^{-1} ) , ( b^{-1}a^{-1} , a^{-1}b^{-1} ) , ( ca , bc ) , ( ca^{-1} , b^{-1}c ) , \\ & ( c^{-1} (a)^n b , ac^{-1} (a)^n ) \ (n \in \mathbb{N}) , \\ & ( c^{-1} (a^{-1})^n b , ac^{-1} (a^{-1})^n ) \ (n \in \mathbb{N}) , \\ & ( c^{-1}(a)^n b^{-1} , a^{-1}c^{-1} (a)^n ) \ (n \in \mathbb{N}) , \\ & ( c^{-1}(a^{-1})^n b^{-1} , a^{-1}c^{-1}(a^{-1})^n ) \ (n \in \mathbb{N}) \ \Big\}. \end{aligned}$$

*(1) Claim.*

$(a)^n b$ and $c^{-1}(a)^n$ are $\mathcal{R}$-irreducible words for all $n \in \mathbb{N}$.

*Proof:*

Let $w$ be the $\mathcal{R}$-representative of the word $(a)^n b$ (any $n \in \mathbb{N}$). Then

$$(2) \quad w \rightarrow_{\mathcal{R}_1}^{*} (a)^n b,$$

because $(a)^n b$ is $\mathcal{R}_1$-irreducible (and $|w| = |(a)^n b|$ ).

Whenever $c$, respectively $c^{-1}$, appears in the left component of a length preserving rewrite rule of $\mathcal{R}_1$, then $c$, respectively $c^{-1}$, is in the right component of that rewrite rule. Also, whenever $a^{-1}$ or $b^{-1}$ appears in the left component of a length preserving rewrite rule of $\mathcal{R}_1$, then $a^{-1}$ or $b^{-1}$ is in the right component of that rewrite rule.

So, by (2), only $a$'s and $b$'s could appear in $w$. As, there must be at least one $b$ in $w$ (because if $w$ consisted only of $a$'s it would be $\mathcal{R}_1$-irreducible, contradicting (2)), and $a < b$, whence $w = (a)^n b$ (is $\mathcal{R}$-irreducible).

Now let $w$ be the $\mathcal{R}$-representative of the word $c^{-1}(a)^n$ (for any $n \in \mathbb{N}$). The word $c^{-1}(a)^n$ is $\mathcal{R}_1$-irreducible, and so $w \rightarrow_{\mathcal{R}_1}^{*} c^{-1}(a)^n$. Because $\mathcal{R}_1$ has no length preserving rewrite rule in which $c^{-1}$ is the first generator of the right component, so it must be that $w[1] = c^{-1}$. Thus $w[2, |w|]$ is the $\mathcal{R}$-representative of $(a)^n$ which (we have already proved) is $\mathcal{R}$-irreducible, whence $w = c^{-1}(a)^n$ (is $\mathcal{R}$-irreducible).

$\boxed{1}$

We have, for all $n \in \mathbb{N}$, $c^{-1}(a)^n b \langle \mathcal{D} \rangle ac^{-1}(a)^n$, while $a < c^{-1}$. So, by (1) and 3.1.1, $c^{-1}(a)^n b \in J(\mathcal{D}) \subset \text{left}(\mathcal{R})$ for all $n \in \mathbb{N}$.

*Case 2.* Let $<$ to be any ShortLex ordering on $C^*$ for which

$$a < b \text{ and } c^{-1} < a,$$

let $\mathcal{R}$ be any $\mathcal{D}$-complete subset with respect to $<$.

The subset, $\mathcal{R}_2$, (of $C^* \times C^*$), listed below, was generated by the computer program described in chapter 4. It is a (minimal) $\mathcal{D}$-complete subset of type $P_1$ with respect to the ShortLex ordering $<_1$ defined by $c <_2 c^{-1} <_2 a <_2 a^{-1} <_2 b <_2 b^{-1}$.

$$\begin{aligned}
\mathcal{R}_2 = \Big\{ & (aa^{-1}, \varepsilon), (a^{-1}a, \varepsilon), (bb^{-1}, \varepsilon), (b^{-1}b, \varepsilon), \\
& (cc^{-1}, \varepsilon), (c^{-1}c, \varepsilon), (ba, ab), (ba^{-1}, a^{-1}b), \\
& (b^{-1}a, ab^{-1}), (b^{-1}a^{-1}, a^{-1}b^{-1}), (ac^{-1}, c^{-1}b), \\
& (a^{-1}c^{-1}, c^{-1}b^{-1}), (bc, ca), (b^{-1}c, ca^{-1}), \\
& (bc^{-1}(a)^{n}b, abc^{-1}(a)^{n})\ (n \in \mathbb{N}), \\
& (bc^{-1}(a^{-1})^{n}b, abc^{-1}(a^{-1})^{n})\ (n \in \mathbb{N}), \\
& (b^{-1}c^{-1}(a)^{n}b, ab^{-1}c^{-1}(a)^{n})\ (n \in \mathbb{N}), \\
& (b^{-1}c^{-1}(a^{-1})^{n}b, ab^{-1}c^{-1}(a^{-1})^{n})\ (n \in \mathbb{N}), \\
& (bc^{-1}(a)^{n}b^{-1}, a^{-1}bc^{-1}(a)^{n})\ (n \in \mathbb{N}), \\
& (bc^{-1}(a^{-1})^{n}b^{-1}, a^{-1}bc^{-1}(a^{-1})^{n})\ (n \in \mathbb{N}), \\
& (b^{-1}c^{-1}(a)^{n}b^{-1}, a^{-1}b^{-1}c^{-1}(a)^{n})\ (n \in \mathbb{N}), \\
& (b^{-1}c^{-1}(a^{-1})^{n}b^{-1}, a^{-1}b^{-1}c^{-1}(a^{-1})^{n})\ (n \in \mathbb{N}), \\
& (b^{-1}c^{-1}(a^{-1})^{n}b^{-1}, a^{-1}b^{-1}c^{-1}(a^{-1})^{n})\ (n \in \mathbb{N}), \\
& (bc^{-1}(a)^{n}aca, abc^{-1}(a)^{n}ac)\ (n \in \mathbb{N}), \\
& (bc^{-1}a^{-1}(a^{-1})^{n}ca, abc^{-1}a^{-1}(a^{-1})^{n}c)\ (n \in \mathbb{N}), \\
& (b^{-1}c^{-1}a(a)^{n}ca, ab^{-1}c^{-1}a(a)^{n}c)\ (n \in \mathbb{N}), \\
& (b^{-1}c^{-1}a^{-1}(a^{-1})^{n}ca, ab^{-1}c^{-1}a^{-1}(a^{-1})^{n}c)\ (n \in \mathbb{N}), \\
& (bc^{-1}a(a)^{n}ca^{-1}, a^{-1}bc^{-1}a(a)^{n}c)\ (n \in \mathbb{N}), \\
& (bc^{-1}a^{-1}(a^{-1})^{n}ca^{-1}, a^{-1}bc^{-1}a^{-1}(a^{-1})^{n}c)\ (n \in \mathbb{N}), \\
& (b^{-1}c^{-1}a(a)^{n}ca^{-1}, a^{-1}b^{-1}c^{-1}a(a)^{n}c)\ (n \in \mathbb{N}), \\
& (b^{-1}c^{-1}a^{-1}(a^{-1})^{n}ca^{-1}, a^{-1}b^{-1}c^{-1}a^{-1}(a^{-1})^{n}c)\ (n \in \mathbb{N}) \Big\}.
\end{aligned}$$

*(3) Claim.*

$c^{-1}(a)^{n}b$ and $bc^{-1}(a)^{n}$ are $\mathcal{R}$-irreducible for all $n \in \mathbb{N}$.

*Proof:*

Let $w$ be the $\mathcal{R}$-representative of the word $c^{-1}(a)^{n}b$ (any $n \in \mathbb{N}$). Then:-

$$(4)\quad w \rightarrow_{\mathcal{R}_2}^{*} c^{-1}(a)^{n}b.$$

because $c^{-1}(a)^{n}b$ is $\mathcal{R}_2$-irreducible.

Whenever one of the generators $a^{-1}$ or $b^{-1}$ appears in the right component of a length preserving rule of $\mathcal{R}_2$, then $a^{-1}$ or $b^{-1}$ appears in the left component of that rule. So neither $a^{-1}$ nor $b^{-1}$ appear in the $\mathcal{R}_2$-descendants of $w$. Whenever $c$ appears in the right component of a length preserving rule of $\mathcal{R}_2$, then it also appears in the left component of that rule. So $c$ cannot appear in the $\mathcal{R}_2$-descendants of $w$.

Thus only $a$, $b$ or $c^{-1}$ appear in $w$. By (3), $c^{-1}$ can appear only once in $w$ because, whenever it appears in a length preserving rule of $\mathcal{R}_2$, it appears precisely once in both components of that rule. Because $c^{-1} < a < b$ (and the words $c^{-1}(a)^{*}$ are $\mathcal{R}_2$-irreducible), whence $w \equiv c^{-1}(a)^{n}b$ (is $\mathcal{R}$-irreducible).

Now let $w$ be the $\mathcal{R}$-representative of the word $bc^{-1}(a)^{n}$ (for any $n \in \mathbb{N}$). The word $bc^{-1}(a)^{n}$ is $\mathcal{R}_2$-irreducible, and so $w \rightarrow_{\mathcal{R}_2}^{*} bc^{-1}(a)^{n}$. Because $\mathcal{R}_2$ has no length preserving rewrite rule in which $b$ is the first generator of the right component, so it must be that $w[1] \equiv b$. Thus $w[2, |w|]$ is the $\mathcal{R}$-representative of $c^{-1}(a)^{n}$, which (we have already proved) is $\mathcal{R}$-irreducible, whence $w \equiv bc^{-1}(a)^{n}$ (is $\mathcal{R}$-irreducible).

$\boxed{3}$

We have, for all $n \in \mathbb{N}$, $bc^{-1}(a)^{n}b \,\langle \mathcal{D} \rangle\, abc^{-1}(a)^{n}$, while $a < b$. So, by (3) and 3.1.1, $bc^{-1}(a)^{n}b \in J(\mathcal{D}) \subset \text{left}(\mathcal{R})$ for all $n \in \mathbb{N}$.

$\boxed{3.1.6}$

As a final example, 3.1.7 is a complete, 2-parameterized presentation of the 2-braid group (we computed this by running a Knuth-Bendix finite-completion program, guessing the necessary parameterized relations, and then confirming the completeness by hand).

We believe (but have not proved) there is no finite ShortLex completion for the 2-braid *group* (but the reader may be interested to know that there *is* a finite completion of the *monoid* presentation $\langle a, b \mid (bab, aba) \rangle$ (see ([1]Kapur,Narendon) and cf (Bauer,Otto)).

*3.1.7 Example.*

The 2-braid group with generators $\{ a < b < a^{-1} < b^{-1} \}$ has a minimal complete presentation of type $P_2$ as follows.

$$
\begin{array}{ll}
\langle\, a, a^{-1}, b, b^{-1} \mid & ( aa^{-1}, \varepsilon ), ( a^{-1}a, \varepsilon ), ( bb^{-1}, \varepsilon ), \\
& ( b^{-1}b, \varepsilon ), ( b^{-1}a^{-1}b^{-1}, a^{-1}b^{-1}a^{-1} ), \\
& ( ba^{-1}(a^{-1})^{n}b^{-1}, a^{-1}b^{-1}(b^{-1})^{n}a ) \ (n \in \mathbb{N}), \\
& ( b^{-1}a(a)^{n}b, ab(b)^{n}a^{-1} ) \ (n \in \mathbb{N}), \\
& ( baaa(a)^{n}ba, abaab(b)^{n}b ) \ (n \in \mathbb{N}), \\
& ( b^{-1}a^{-1}(a^{-1})^{n}b, ab^{-1}(b^{-1})^{n}a^{-1} ) \ (n \in \mathbb{N}), \\
& ( b^{-1}a^{-1}a^{-1}a^{-1}(a^{-1})^{n}b^{-1}a^{-1}, a^{-1}b^{-1}a^{-1}a^{-1}b^{-1}(b^{-1})^{n}b^{-1} ) \ (n \in \mathbb{N}), \\
& ( b^{-1}a^{-2}(a^{-1})^{n}(b^{-1})^{m}b^{-1}a, ab^{-1}(b^{-1})^{n}(a^{-1})^{m}a^{-2}b^{-1} ) \ (n \in \mathbb{N}, m \in \mathbb{N}), \\
& ( baa(a)^{n}(b)^{m}ba^{-1}, a^{-1}b(b)^{n}(a)^{m}aab ) \ (n \in \mathbb{N}, m \in \mathbb{N}) \ \rangle.
\end{array}
$$

### 3.1.7

There are several examples which could be cited of (sub)classes of groups which may possess parametrized, but not finite, complete presentations. Le Chenadec has described (possibly infinite) complete presentations of the Coxeter groups and has observed that, whereas some of these groups (with partial commutivity of the generators) may not possess finite complete presentations, they do have complete parameterized presentations (cf. (Le Chenadec) and the report of 4.5).

In some programs (beyond the scope of the authors work) currently being written in the research of automatic groups, the word differences are computed by attempting to find complete sets of relations, and it seems that, in practice, these relations are often r-parameterized (see (Epstein, Holt, Rees)).

The structure of groups and monoids defined by finite complete presentations has been studied by C.C. Squier (Squier), and by J.R.J. Groves and G.C. Smith in (Groves,Smith). In his paper 'Word problems and homological finiteness conditions', Squier proves that monoids defined by finite complete presentations have a certain *homological finiteness condition* (called $(FP)_3$) and cites an example of a monoid defined by a complete 1-parameterized presentation which does not have this condition (see Squier). So there are monoids which have parameterized complete presentations but which are not defined by any finite complete presentation.

In (Hayashi), C. Hayashi works with *semi-confluent presentations*(of monoids M with generators C, say). Basically, by adjoining a dummy generator to C, Hayashi was able to write a program which attempts completions of presentations considered as (possibly infinite) regular languages over C×C. If the presentation is successfully completed then, by adjoining the rule which maps the dummy generator to $\varepsilon$, it yields a complete presentation of M which Hayashi calls a *semi-confluent presentation* of M. The program is an improvement on finite completion programs, but the undecidability of completeness of such presentations is, apparently, not raised.

C. Ó' Dúnlaing has studied *infinite regular thue systems*. These are monoid (but not group) presentations, $\langle C \mid \mathcal{R} \rangle$, where left( $\mathcal{R}$ ) is a regular subset of $C^*$. Ó' Dúnlaing has proved that the completeness of such a presentation is decidable if it is *monadic* (i.e., all words in right( $\mathcal{R}$ ) have length 0 or 1), but that completeness is not necessarily decidable otherwise (see (Ó' Dúnlaing)). In the next section we will prove that completeness of the $P_1$ presentations is not necessarily decidable.

## 3.2

## *Completeness of a monoid presentation of type $P_{r>0}$ can be undecidable.*

In (Ó' Dúnlaing) C. Ó' Dúnlaing proves that there are infinite regular monoid presentations for which completeness is undecidable. In theorem 3.2.2 we focus on the $P_{r>0}$ presentations and prove that there are monoid presentations of type $P_1$ for which completeness is undecidable (which provides some excuse for omitting, from our program of chapter 4, a method for deciding the completeness of the group presentations of type $P_1$).

The proof of theorem 3.2.2 depends on the theory of Turing machines and, with prior agreement on semantics and suitable notation, is easy and concise. So, we will begin with a brief résumé of Turing machines.

### *Turing Machines.*

There is a plethora of similar, but equivalent, models of Turing machines. The model we describe is probably the simplest, and the one best suited to our needs.

Informally, a Turing machine, T, consists of a finite state control device coupled to a primitive data storage device via a scan/print head. This 'data storage device' may be thought of as a (variable) finite paper tape partitioned into a row of squares. In each of these squares we can print a single symbol from T's finite *tape alphabet*, $\mathcal{A}$, which has the reserved symbol '$\mathcal{B}$' (for blank) as a member. The paper tape will always be finite but, when necessary, it can be extended at either end by splicing on an extra square preprinted with the single symbol '$\mathcal{B}$'.

In each of its possible discrete *configurations* the machine T will be scanning, by way of its scan/print head, a single square of the paper tape, and a single state of T's finite *state set*, S, will be entered in the state control device .

Machine T's deterministic processing technique is, ostensibly, rather primitive and completely controlled by a finite set of commands called the *transitions* of T. The transitions determine the machines consecutive configurations.

The state set of T has two reserved states called the *start state* and the *halt state*. Whenever T 's configuration is such that there is a state of S - {halt state} entered in the state control device, then, depending *solely* on the current entry of the state control device and the symbol on the square currently being read by the scan/print head, the machine is commanded by a transition to, in one go:

*(1)* enter a single state of S - {start state} in the state control device,

*(2)* overprint the symbol on the scanned square by a single symbol from $\mathcal{A}$,

*(3)* shift the scan/print head one square to the left, or one square to the right.

If, during the process of (3), the scan/print head attempts to move off one end of the paper tape, then one exta square, preprinted with the single symbol ' $\mathcal{B}$ ', is spliced onto that end of the tape.

As both S and $\mathcal{A}$ are finite, we see that the transitions of T can be defined by a finite set of quintuples of the form:

( current state , symbol being scanned , new state , symbol printed , (L)eft or (R)ight motion of scan/print head ).

The machine is said to be *stable*, i.e. , there is no subsequent processing, when and only when the halt state is entered in the state control device. When we refer to $C_0$ , $C_1$ as being *consecutive configurations of T*, we mean that the machine T is not stable in the configuration $C_0$ , but changes its configuration to $C_1$ (without there being an intermediate configuration). We will, for the sake of clarity, abbreviate the phrase *consecutive configurations of T* to c.c.T.

Machine T 's *start* configurations are configurations of the machine with the start state entered in the state control device, and T 's *halt* configurations are configurations of the machine with the halt state entered in the state control device. If T is set up in a start configuration then the subsequent *computation* of the machine need not stop, but if it does stop, then it must stop in the first halt configuration of that computation.

The *halting problem* of machine T, set up in some start configuration, is the problem of being able to decide whether the subsequent computation of T stops. The traditional notion of decidability guarantees Turing machines with undecidable halting problems (cf (Kfoury,Moll,Arbib)).

With this model of T in mind, we give the formal definition of a Turing machine.

*3.2.1 Definition.*

A (deterministic) *Turing machine*, T, is defined by a quintuple

$$( S , \mathcal{A} , p_0 , p_H , \tau ),$$

where:

$$\left\{ \begin{array}{c} S \text{ is the finite } \textit{state set} . \\ \mathcal{A} \text{ is the finite } \textit{tape alphabet} \text{ , and the symbol ` } \mathcal{B} \text{ ' is in } \mathcal{A} . \\ p_0 \text{ in } S \text{ is the } \textit{start state} . \\ p_H \text{ in } S \text{ is the } \textit{halt state} . \\ \tau : S-\{p_H\} \times \mathcal{A} \longrightarrow S-\{p_0\} \times \mathcal{A} \times \{L,R\} \text{ is the } \textit{transition function} . \end{array} \right.$$

$\boxed{3.2.1}$

*3.2.2 Theorem.*

Completeness of a monoid presentation of type $P_1$ can be undecidable.

*Proof:*

We fix a Turing machine, T, with an undecidable halting problem, and suppose T to be defined by the quintuple:

$$( S , \mathcal{A} , p_0 , p_H , \tau ).$$

We may as well assume that S and $\mathcal{A}$ are disjoint.

Let ' £ ', ' \$ ', '/' and ' # ' be four (dummy) symbols not of $S \cup \mathcal{A}$, and then put:

$$\mathcal{S} = \mathcal{A} \cup S \cup \{ £ , \$ , / , \# \}.$$

We fix any shortest ShortLex ordering, $\geq$ say, of $\mathcal{S}^*$ with the proviso:

$$\# > /.$$

We will begin by confecting a subset of $\mathcal{S}^*$, $\mathcal{CONFIG}$, corresponding to the configurations of T, and a monoid presentation, $( \mathcal{S} \mid \mathcal{CHANGE} )$, so that the $\mathcal{CHANGE}$-reductions of words in $\mathcal{CONFIG}$ mimic the processing of machine T.

We define $\mathcal{CONFIG}$ to be the set of words $\Big\{$

$$\mathcal{L}/a_1/a_2\ldots/a_{(s-1)}\,\#^{2n}\,p\,a_s\,a_{(s+1)}\cdots\,a_\kappa\,\$$$

$$\mid\ 1\leq s\leq\kappa,\ \text{all the } a_j \text{ belong to } \mathcal{A},\ n\in\mathbb{N},\ \text{and } p\in S\ \Big\}.$$

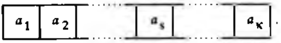
We define the correspondence

$\zeta : \mathcal{CONFIG} \longrightarrow$ set of configurations of machine T

as follows. We let

$$\zeta(\,\mathcal{L}/a_1/a_2\ldots/a_{(s-1)}\,\#^{2n}\,p\,a_s\,a_{(s+1)}\cdots\,a_\kappa\,\$\,)$$

be the configuration of machine T with the state p entered in the state control device and currently scanning the s$^{th}$ square of the paper tape



If $\mathcal{C}\in\mathcal{CONFIG}$, then we write no.#($\mathcal{C}$) for the number of ' # ' symbols which occur in $\mathcal{C}$.

We wish to mimic T 's processing by defining $\mathcal{CHANGE}$ so that the following holds.

*(1)*

$$\text{(i)}\ \ \mathcal{C}_0\in\mathcal{CONFIG}\ \text{ and }\ \mathcal{C}_0\rightarrow_{\mathcal{CHANGE}}\mathcal{C}_1\rightarrow_{\mathcal{CHANGE}}\mathcal{C}_2\cdots\rightarrow_{\mathcal{CHANGE}}\mathcal{C}_n$$

if and only if

$$\text{(ii)}\ \ \mathcal{C}_0,\mathcal{C}_1,\ldots,\mathcal{C}_n\in\mathcal{CONFIG}\ \text{ with }\ \text{no.}\#(\mathcal{C}_i)=\text{no.}\#(\mathcal{C}_{(i+1)})+2,\ \text{for } 0\leq i<n,$$

$$\text{and } \zeta(\mathcal{C}_0),\zeta(\mathcal{C}_1),\ldots,\zeta(\mathcal{C}_n) \text{ being c.c.T.}$$

Before commencing with the construction of $\mathcal{CHANGE}$, we should, informally, motivate the definition of $\mathcal{CONFIG}$ and the statement of (1).

Let us suppose the halting problem of T, set up in configuration $C_0$, say, is undecidable. The gist of (1) is as follows. Subject to (arbitrary) $\mathcal{C}_0\in\zeta^{-1}(C_0)$ being such that no.#($\mathcal{C}_0$)≥2n, there is a halt-reduction (i) if and only if there are at least n c.c.T, beginning with T set up in configuration $C_0$. Also, up to no.#($\mathcal{C}_0$), the n words of reduction (i) correspond to the first n c.c.T of the computation of $C_0$ (so, the more # 's in

$C_0$, the longer the reduction (i), the longer we mimic the computation of T).

Suppose we define $\mathcal{START}$ to be all those words of $\zeta^{-1}(C_0)$ which have at least two # 's. We will see that $\mathcal{CHANGE}$ is (trivially) complete. As each $\mathcal{CHANGE}$ reduction decreases the number of # 's by two, whence the computation of T does not stop if and only if every word of $\mathcal{START}$ has $\mathcal{CHANGE}$-representative corresponding to a non-halt, non-start configuration, and containing zero # 's. We can then append to $\mathcal{CHANGE}$, to derive (complete) $\mathcal{HALT}$, three rewrite rules so that *precisely* the latter words $\mathcal{HALT}$-reduce to the empty word. Whence (lemma 3.2.4), the computation of T does not stop if and only if every word of $\mathcal{START}$ has the empty word as $\mathcal{HALT}$-representative. We finish by appending to $\mathcal{HALT}$ the (1-parameterized) set of rules which reduce every word of $\mathcal{START}$ to the empty word. By so doing, we define a 1-parameterized presentation with undecidable completeness (lemma 3.2.5).

Let us now consider a computation of T. We have already mentioned that T 's transitions can be defined by a subset of

$$(S-\{p_H\}) \times \mathcal{A} \times (S-\{p_0\}) \times \mathcal{A} \times \{L,R\}.$$

So let us suppose, during this computation, T 's current configuration is:

$$\zeta(\pounds / a_1 / a_2 \ldots / a_{(s-1)} \#^{(2n+2)}\, p\, a_s\, a_{(s+1)} \cdots a_k \$ ),$$

with s>1, and that there is a transition:

$$(p, a_s, \not{p}, o, L).$$

Then $\not{p}$ would be entered in the state control device, the symbol in the $s^{th}$ square, $a_s$, would be overprinted by the symbol $o$, and the scan/print head would shift one square to the left. So the machines configuration changes to:-

$$\zeta(\pounds / a_1 / a_2 \ldots / a_{(s-2)} \#^{2n}\, \not{p}\, a_{(s-1)}\, o\, a_{(s+1)} \cdots a_k \$ ).$$

To mimic this transition, $\mathcal{CHANGE}$ must have the rewrite rule:-

$$(2)\ \ (/ a_{(s-1)} \#^{(2n+2)}\, p\, a_s\, ,\ \#^{2n}\, \not{p}\, a_{(s-1)}\, o).$$

Supposing T 's current configuration is:

$$\zeta(\pounds \#^{(2n+2)}\, p\, a_1\, a_2 \ldots a_k \$ ),$$

and there is a transition:

$$(p, a_1, \not{p}, o, L).$$

Then $\not{p}$ would be entered in the state control device, the symbol in the first square, $a_1$, would be overprinted by symbol $o$, and the scan/print head begins to scan an extra square, preprinted with the single symbol '$\mathcal{B}$', spliced onto the left end of the tape. So the machines configuration changes to:-

$$\zeta(\pounds\, \#^{2n}\, \not{p}\, \mathcal{B}\, o\, a_2 \ldots\ a_\kappa\, \$\,).$$

To mimic this transition, *CHANGE* must have the rewrite rule:-

$$\mathit{(3)}\ \ (\,\pounds\, \#^{(2n+2)}\, p\, a_1\, ,\, \pounds\, \#^{2n}\, \not{p}\, \mathcal{B}\, o\,).$$

We define *MOVELEFT* to be the set of all the rewrite rules (2) and (3) for all n∈ℕ.

Suppose T 's current configuration is:

$$\zeta(\,\pounds/a_1/a_2 \ldots /a_{(s-1)}\, \#^{(2n+2)}\, p\, a_s\, a_{(s+1)} \cdots\ a_\kappa\, \$\,),$$

with s<κ, and that there is a transition:

$$(\,p\, ,\, a_s\, ,\, \not{p}\, ,\, o\, ,\, R\,).$$

Then $\not{p}$ would be entered in the state control device, the symbol in the s$^{th}$ square, $a_s$, would be overprinted by the symbol $o$, and the scan/print head would shift one square to the right. So the machines configuration changes to:-

$$\zeta(\,\pounds/a_1/a_2 \ldots /a_{(s-1)}/o\, \#^{2n}\, \not{p}\, a_{(s+1)} \cdots\ a_\kappa\, \$\,).$$

To mimic this transition, *CHANGE* must have the rewrite rule:-

$$\mathit{(4)}\ \ (\,\#^{(2n+2)}\, p\, a_s\, a_{(s+1)}\, ,\, /o\, \#^{2n}\, \not{p}\, a_{(s+1)}\,).$$

Supposing T 's current configuration is:

$$\zeta(\,\pounds/a_1/a_2 \ldots /a_{(s-1)}\, \#^{(2n+2)}\, p\, a_\kappa\, \$\,),$$

and there is a transition:

$$(\,p\, ,\, a_\kappa\, ,\, \not{p}\, ,\, o\, ,\, R\,).$$

Then $\not{p}$ would be entered in the state control device, the symbol in the κ$^{th}$ square, $a_\kappa$, is overprinted by symbol $o$, and the scan/print head begins to scan an extra square, preprinted with the single symbol '$\mathcal{B}$', spliced onto the right end of the tape . So the machines configuration changes to:-

$$\zeta(\,\pounds/a_1/a_2 \ldots /a_{(s-1)}/o\, \#^{2n}\, \not{p}\, \mathcal{B}\, \$\,).$$

To mimic this transition, *CHANGE* must have the rewrite rule:-

$$(5)\ ( \#^{(2n+2)}\, p\, a_{\kappa}\, \$\, , / o\, \#^{2n}\, \acute{p}\, \mathcal{B}\, \$ )$$

(recall the proviso that # >/).

We define $\mathcal{MOVERIGHT}$ to be the set of all the rewrite rules (4) and (5) for all n∈ℕ.

We have, $\mathcal{MOVELEFT} = \Big\{$

$$( / a\, \#^{(2n+2)}\, p\, s\, ,\, \#^{2n}\, \acute{p}\, a\, o )\, ,\, ( \pounds\, \#^{(2n+2)}\, p\, s\, ,\, \pounds\, \#^{2n}\, \dot{p}\, \mathcal{B}\, o )$$

$\mid$ ( p , $s$ , $\acute{p}$ , $o$ , L ) is a transition of T, $a \in \mathcal{A}$ and n∈ℕ $\Big\}$.

Also, $\mathcal{MOVERIGHT} = \Big\{$

$$( \#^{(2n+2)}\, p\, s\, a\, , / o\, \#^{2n}\, \acute{p}\, a )\, ,\, ( \#^{(2n+2)}\, p\, s\, \$\, , / o\, \#^{2n}\, \acute{p}\, \mathcal{B}\, \$ )$$

$\mid$ ( p , $s$ , $\dot{p}$ , $o$ , R ) is a transition of T, $a \in \mathcal{A}$ and n∈ℕ $\Big\}$.

Whence, with:

$$\mathcal{CHANGE} = \mathcal{MOVELEFT} \cup \mathcal{MOVERIGHT}\, ,$$

i.e., $\mathcal{CHANGE}$ consisting of the rewrite rules (2)-(5) for all n∈ℕ, we have the following.

*(6)*

*(i)* $C_0 \in \mathcal{CONFIG}$ and $C_0 \rightarrow_{\mathcal{CHANGE}} C_1$

if and only if

*(ii)* $C_0 , C_1 \in \mathcal{CONFIG}$ with no.#($C_0$)=no.#($C_1$)+2,

and $\zeta(C_0)$ , $\zeta(C_1)$ being c.c.T.

We iterate (6) to derive:-

*(1)*

*(i)* $C_0 \in \mathcal{CONFIG}$ and $C_0 \rightarrow_{\mathcal{CHANGE}} C_1 \rightarrow_{\mathcal{CHANGE}} C_2 \cdots \rightarrow_{\mathcal{CHANGE}} C_n$

if and only if

*(ii)* $C_0 , C_1 , \ldots , C_n \in \mathcal{CONFIG}$ with no.#($C_i$)=no.#($C_{(i+1)}$)+2, for 0≤i<n,

and $\zeta(C_0)$ , $\zeta(C_1)$ ,..., $\zeta(C_n)$ being c.c.T.

We now put

$\mathcal{HALT}$ =

$$\mathcal{CHANGE} \cup \left\{ (/a\,p\,,\,p\,)\,,\,(\,\pounds\,p\,a\,,\,\pounds\,p\,)\,,\,(\,\pounds\,p\,\$\;,\,\varepsilon\,) \mid p \in S - \{\,p_0\,,\,p_H\,\} \text{ and } a \in \mathcal{A} \right\}.$$

Note that $\langle\, \mathcal{S} \mid \mathcal{HALT} \,\rangle$ is of type $P_1$ and, trivially, is complete (because for $\mathcal{HALT}$ to conceivably have a critical pair, it would be necessary for T to have transitions $(\,p\,,\,s\,,\,\not{p}\,,\,o\,,\,L\,)$ and $(\,p\,,\,s\,,\,\bar{p}\,,\,o\,,\,R\,)$ – which is not possible because the transition function of T would not be well defined).

We now fix a start configuration, $C_0$ say, so that the halting problem of machine T, set up in configuration $C_0$, is undecidable. Let us suppose:

$$\zeta^{-1}(C_0) = \left\{\, \pounds/a_1/a_2 \ldots /a_{(s-1)}\,\#^{2n}\,p_0\,a_s\,a_{(s+1)} \cdots\, a_\kappa\,\$ \;\mid\; n \in \mathbb{N} \right\}$$

(for some $a_j$, $1 \leq s \leq \kappa$, belonging to $\mathcal{A}$).

We then put:

$$(7)\quad \mathcal{START} = \left\{\, \pounds/a_1/a_2 \ldots /a_{(s-1)}\,\#^{(2n+2)}\,p_0\,a_s\,a_{(s+1)} \cdots\, a_\kappa\,\$ \;\mid\; n \in \mathbb{N} \right\}.$$

*3.2.3 Lemma.*

The (start state) symbol $p_0$ cannot appear in proper $\mathcal{HALT}$-descendants of words belonging to $\mathcal{START}$.

*Proof:*

By the definition of the transition function of T, there are no transitions of the form $(\,p\,,\,s\,,\,p_0\,,\,o\,,\,L\,)$ or $(\,p\,,\,s\,,\,p_0\,,\,o\,,\,R\,)$. Thus, we can see, from the definitions of $\mathcal{CHANGE}$ and $\mathcal{HALT}$, that the symbol $p_0$ does not appear in any word of right( $\mathcal{HALT}$). The result is then a trivial consequence of the formats of $\mathcal{START}$ and left( $\mathcal{HALT}$ ).

$\boxed{3.2.3}$

*3.2.4 Lemma.*

Let machine T be set up in configuration $C_0$, then the subsequent computation does not stop if and only if the $\mathcal{HALT}$-representative of every word in $\mathcal{START}$ is the empty word.

*Proof:*

($\Rightarrow$) If the subsequent computation did not stop, then there would be an infinite sequence of c.c.T, say $C_0, C_1, C_2, \ldots$.

Then, with arbitrary

$$\mathcal{C}_0 \equiv \pounds/a_1/a_2\ldots/a_{(s-1)}\#^{(2n+2)}\,p_0\,a_s\,a_{(s+1)}\cdots a_k\,\$$$

belonging to $\mathcal{START}$, we define $\{\mathcal{C}_j\}_{1\le j\le n+1} \subseteq \mathcal{CONFIG}$ by:

$$\text{no.}\#(\mathcal{C}_j) = 2(n+1-j) \text{ and } \mathcal{C}_j \in \zeta^{-1}(C_j).$$

By (5), we would then have:-

$$\mathcal{C}_0 \rightarrow_{\mathcal{CHANGE}} \mathcal{C}_1 \rightarrow_{\mathcal{CHANGE}} \mathcal{C}_2 \cdots \rightarrow_{\mathcal{CHANGE}} \mathcal{C}_{(n+1)}.$$

As no.$\#(\mathcal{C}_{(n+1)})=0$ and $\zeta(\mathcal{C}_{(n+1)})$ cannot be a halt configuration, so it must be that:-

$$\mathcal{C}_{(n+1)} \equiv \pounds/\acute{a}_1/\acute{a}_2\ldots/\acute{a}_{(\iota-1)}\,p\,\acute{a}_\iota\,\acute{a}_{(\iota+1)}\cdots\acute{a}_\mu\,\$$$

for some $\acute{a}_j$, $1\le j\le\mu$, belonging to $\mathcal{A}$ and $p\in S-\{p_0, p_H\}$. Whence:-

$$\mathcal{C}_0 \rightarrow_{\mathcal{HALT}}^{*} \pounds/\acute{a}_1/\acute{a}_2\ldots/\acute{a}_{(\iota-1)}\,p\,\acute{a}_\iota\,\acute{a}_{\iota+1}\cdots\acute{a}_\mu\,\$$$

$$\rightarrow_{\mathcal{HALT}}^{*} \pounds\,p\,\acute{a}_\iota\,\acute{a}_{(\iota+1)}\cdots\acute{a}_\mu\,\$,$$

because $(/a\,p\,,\,p\,)\in\mathcal{HALT}$ for all $p\in S-\{p_0, p_H\}$ and all $a\in\mathcal{A}$,

$$\rightarrow_{\mathcal{HALT}}^{*} \pounds\,p\,\$,$$

because $(\,\pounds\,p\,a\,,\,\pounds\,p\,)\in\mathcal{HALT}$ for all $p\in S-\{p_0, p_H\}$ and all $a\in\mathcal{A}$,

$$\rightarrow_{\mathcal{HALT}}^{*} \varepsilon,$$

because $(\,\pounds\,p\,\$\,,\,\varepsilon\,)\in\mathcal{HALT}$ for all $p\in S-\{p_0, p_H\}$ and all $a\in\mathcal{A}$.

Whence $\text{rep}_{\mathcal{HALT}}(\mathcal{C}_0)\equiv\varepsilon$, as required.

($\Leftarrow$) Conversely, if T 's subsequent computation does stop, then there must be a finite sequence of c.c.T, $C_0, C_1, C_2, \ldots, C_h$ say, with $C_h$ being a halt configuration. We could then define $\{\mathcal{C}_j\}_{1\le j\le h} \subseteq \mathcal{CONFIG}$ by:

$$\text{no.}\#(\mathcal{C}_j) = 2(h+1-j) \text{ and } \mathcal{C}_j \in \zeta^{-1}(C_j),$$

so that, by (5 ):-

$$C_0 \to_{\mathcal{CHANGE}} C_1 \to_{\mathcal{CHANGE}} C_2 \cdots \to_{\mathcal{CHANGE}} C_h .$$

As no.$\#(C_h)=2$ and $\zeta(C_h)$ is a halt configuration, so it must be that:-

$$C_{(n+1)} \equiv \pounds / \acute{a}_1 / \acute{a}_2 \ldots / \acute{a}_{(t-1)} \#^2 p_H \, \acute{a}_t \, \acute{a}_{(t+1)} \cdots \acute{a}_\mu \, \$$$

for some $\acute{a}_j$, $1 \leq j \leq \mu$, belonging to $\mathcal{A}$. We can see, by the definition of $\mathcal{HALT}$, that this word is $\mathcal{HALT}$-irreducible, so $C_0 \in \mathcal{START}$, while $\mathrm{rep}_{\mathcal{HALT}}(C_0) \not\equiv \varepsilon$.

$\boxed{3.2.4}$

We can now define a 1-parameterized presentation, $\mathcal{P} = \langle\, S \mid \mathcal{U} \,\rangle$, of which completeness is undecidable.

Recall that $\mathcal{MOVELEFT} = \Big\{$

$$(/a\,\#^{(2n+2)}\, p\, s\, ,\ \#^{2n}\, \acute{p}\, a\, o)\, ,\ (\pounds\, \#^{(2n+2)}\, p\, s\, ,\ \pounds\, \#^{2n}\, \acute{p}\, \mathcal{B}\, o)$$

$$\mid (p, s, \acute{p}, o, L) \text{ is a transition of T, } a \in \mathcal{A} \text{ and } n \in \mathbb{N} \Big\}.$$

Also, $\mathcal{MOVERIGHT} = \Big\{$

$$(\#^{(2n+2)}\, p\, s\, a\, ,/o\, \#^{2n}\, \acute{p}\, a)\, ,\ (\#^{(2n+2)}\, p\, s\, \$\, ,/o\, \#^{2n}\, \acute{p}\, \mathcal{B}\, \$)$$

$$\mid (p, s, \acute{p}, o, R) \text{ is a transition of T, } a \in \mathcal{A} \text{ and } n \in \mathbb{N} \Big\}.$$

Then we defined $\mathcal{HALT} =$

$$\mathcal{MOVELEFT} \cup \mathcal{MOVERIGHT} \cup \Big\{ (/a p\, , p)\, , (\pounds p\, a\, , \pounds p)\, , (\pounds p\, \$\ \ , \varepsilon) \mid$$

$$p \in S - \{ p_0 , p_H \} \text{ and } a \in \mathcal{A} \Big\}.$$

We put:-

$$(8)\quad \mathcal{U} = \{ (C, \varepsilon) \mid C \in \mathcal{START} \} \cup \mathcal{HALT},$$

where, we remind the reader,

$$(7)\quad \mathcal{START} = \Big\{ \ (\pounds / a_1 / a_2 \ldots / a_{(s-1)}\, \#^{(2n+2)}\, p_0\, a_s\, a_{(s+1)} \cdots a_k\, \$ \ ) \mid n \in \mathbb{N} \Big\}.$$

Then $\mathcal{P} = \langle\, S \mid \mathcal{U} \,\rangle$ will be a normalized monoid presentation of type $P_1$, with respect to any shortlex ordering, $\geq$, with the the proviso that $\# > /$. Also, it is easy to prove that the completeness of $\mathcal{P}$ is undecidable.

*3.2.5 Lemma.*

The completeness of $\mathcal{P}$ is undecidable.

*Proof:*

Let us suppose, by way of example, that ( $p_0$ , $a_s$ , $\acute{p}$ , $o$, L ) is a transition of T. Then, for all $n \in \mathbb{N}$, the rewrite rule

$$( / a_{(s-1)} \#^{(2n+2)} p_0 \, a_s \, , \#^{2n} \text{þ} \, a_{(s-1)} \, o )$$

belongs to $\mathcal{MOVELEFT}$. We then note that $\mathcal{P}$ will be complete if and only if, for all $n \in \mathbb{N}$, the criticalpairs

$$( \pounds / a_1 / a_2 \ldots / a_{(s-2)} \#^{2n} \acute{p} \, a_{(s-1)} \, o \, a_{(s+1)} \cdots a_k \, \$ \, , \varepsilon )$$

are resolved, i.e, if and only, for all $n \in \mathbb{N}$,

$$(9) \; \pounds / a_1 / a_2 \ldots / a_{(s-1)} \#^{(2n+2)} p_0 \, a_s \, a_{(s+1)} \cdots a_k \, \$ \rightarrow_{\mathcal{MOVELEFT}}$$
$$\pounds \;\; / a_1 / a_2 \ldots / a_{(s-2)} \#^{2n} \acute{p} \, a_{(s-1)} \, o \, a_{(s+1)} \cdots a_k \, \$ \rightarrow_{u}^{*} \varepsilon .$$

We know by, 3.2.3, that the (start state) symbol, $p_0$ , cannot appear in proper $\mathcal{HALT}$-descendants of words in $\mathcal{START}$. Thus, by (7), the rules of $\{ ( C , \varepsilon ) \mid C \in \mathcal{START} \}$ play no part in the reduction (9). It follows, from (8), that $\mathcal{P}$ will be complete if and only if, for all $n \in \mathbb{N}$,

$$\pounds / a_1 / a_2 \ldots / a_{(s-1)} \#^{(2n+2)} p_0 \, a_s \, a_{(s+1)} \cdots a_k \, \$ \rightarrow_{\mathcal{HALT}}^{*} \varepsilon ,$$

i.e., if and only if the $\mathcal{HALT}$-representative of each word in $\mathcal{START}$ is the empty word. So, by 3.2.4, $\mathcal{P}$ will be complete if and only if the computation of T, set up in configuration $C_0$ , does not stop (which is undecidable).

$\boxed{\text{3.2.5 and 3.2.1}}$

The 1-parameterized presentations, such as $\mathcal{P}$ of 3.2.1, are the simplest examples of r-parameterized presentations with r>1 (r−1 repeating factors being the empty word). Although the author would not necessarily agree, it may still seem, perfectly reasonably, to some readers to be 'cheating' to claim that 3.2.1 demonstrates the undecidability of completeness of a general r-parameterized presentation. However, not surprisingly, it is easy to redefine $\mathcal{MOVELEFT}$, $\mathcal{MOVERIGHT}$, $\mathcal{HALT}$, et al. of theorem 3.2.1, so as to exhibit a more 'realistic' r-parameterized presentation, with $r>1$ and no empty repeating factors, of which completeness is undecidable. We will not bother, however, to pursue this topic any further.

# §4

# Programming the Knuth–Bendix Completion of (ShortLex) 1-Parameterized Group Presentations

## 4.0

Suppose $\langle C \mid \mathcal{D} \rangle$ is a normalized monoid presentation. Then there is a corollary of the Knuth–Bendix lemma (probably familiar to the reader), called the *Knuth-Bendix completion procedure*, which attempts to *complete* $\mathcal{D}$, i.e., find a $\mathcal{D}$-complete subset of $C^*\times C^*$.

In his paper 'Presentations of Groups and Monoids' (Gilman 79), R. Gilman (comprehensively) describes the Knuth–Bendix procedure for attempting to compute finite $\mathcal{D}$-complete subsets of group presentations. In 'Enumererating Infinitely Many Cosets' (Gilman 84) he notes that the success of this procedure is susceptible to small changes in the Knuth–Bendix ordering (in fact he cites examples 3.1.3 and 3.1.5). Gilman then suggests that the program might be improved if it were to attempt to compute 1-parameterized, rather than just finite, $\mathcal{D}$-complete subsets. This is the subject of this chapter.

In the first four sections of this chapter we will be describing a computer program (written in pseudo 'C') which attempts to compute 1-parameterized $\mathcal{D}$-complete group presentations. We do not claim that this short program is particularly sophisticated (the Knuth–Bendix procedure in this restricted setting is anyway not complicated), but it is reasonably successful, and we conclude the chapter with a brief report on its implementation (i.e. section 4.5). It would be flippant to infer from this that writing a program to complete the more general r-parameterized group presentations would be easy. Nevertheless, there would probably not be many theoretical difficulties involved in such a project, and we believe it to be worth consideration.

## 4.1

## *Preliminaries*

Throughout this chapter $\langle C \mid \mathcal{D} \rangle$ will be a (semigroup) presentation of a group, G, with $\mathcal{D}$ being a normalized subset (of $C^* \times C^*$) of type $P_1$ with respect to a (fixed) ShortLex ordering >.

Recall that, if $\mathcal{T} \in (C^*)^\kappa$ ($\kappa=1$ or 3), then $\mathcal{T}_i$ ($1 \leq i \leq \kappa$) is the i th component of $\mathcal{T}$. If $n \in \mathbb{N}$, then it will be convenient to write $\mathcal{T}(n)$ for the word $\mathcal{T}_1$, if $\kappa=1$, or the word $\mathcal{T}_1(\mathcal{T}_2)^n \mathcal{T}_3$, if $\kappa=3$.

If

$$(1)\ \ \mathcal{R} \text{ is a subset of } (C^* \times C^*) \cup ((C^*)^3 \times (C^*)^3),$$

then we write $\mathcal{R}^{(\mathbb{N})}$ for the set $\{ (\mathcal{B}(n), \mathcal{A}(n)) \mid (\mathcal{B}, \mathcal{A}) \in \mathcal{R},\ n \in \mathbb{N} \}$. Then $\mathcal{D}$ being of type $P_1$ just means that $\mathcal{D} = \mathcal{R}^{(\mathbb{N})}$ for some subset, $\mathcal{R}$, of $(C^* \times C^*) \cup ((C^*)^3 \times (C^*)^3)$. We will assume $\mathcal{R}$ to be variable throughout this chapter, but it will always have the form (1).

Recall that the critical pairs of $\mathcal{R}^{(\mathbb{N})}$ are all those pairs of words:

$$(2)\ \ (a_1, pa_2s) \text{ with } b_1 = pb_2s \text{, for some words } p \text{ and } s,$$

$$(3)\ \ (a_1s, pa_2) \text{ with } b_1s = pb_2 \text{, for some words } \varepsilon \neq p \neq b_1 \text{ and } \varepsilon \neq s \neq b_2,$$

where $(b_1, a_1)$ and $(b_2, a_2)$ are any two rules of $\mathcal{R}^{(\mathbb{N})}$.

A critical pair is resolved if the words in that pair have a common $\mathcal{R}^{(\mathbb{N})}$-descendant, then the Knuth-Bendix lemma states that:

$$(4)\ \ \langle C \mid \mathcal{R}^{(\mathbb{N})} \rangle$$

is complete if and only if all the critical pairs of $\mathcal{R}^{(\mathbb{N})}$ are resolved.

So, whilst $\mathcal{R}^{(\mathbb{N})}$ is not complete there will be critical pairs of $\mathcal{R}^{(\mathbb{N})}$ with distinct $\mathcal{R}^{(\mathbb{N})}$-irreducible descendants. Suppose $(w, v)$ is one such pair with $a$ and $b$, respectively, being distinct $\mathcal{R}^{(\mathbb{N})}$-irreducible descendants of $w$ and $v$. Then a new rule (for example; $(a, b)$, if $a > b$; or $(b, a)$, if $b > a$) may be adjoined to $\mathcal{R}^{(\mathbb{N})}$ so that $(w, v)$ is resolved (in the augmented $\mathcal{R}^{(\mathbb{N})}$). By doing this we *resolve* the critical pair $(w, v)$, and the

*Knuth-Bendix procedure*, which we program, is the resolving of all the critical pairs of (variable) $\mathcal{R}^{(\mathbb{N})}$ whilst $\mathcal{R}^{(\mathbb{N})}$ is not complete.

There are problems, foremost is that the completeness of (4) may not be decidable. We have only proved this for monoid presentations of type $P_1$, which does not mean that the completeness of group presentations of type $P_1$ is not always decidable – but this is not probable (we did try to prove that completeness was decidable for a few restricted classes of group presentations of type $P_1$ but without success). Nevertheless, in practice, with non-contrived presentations, a reasonable attempt can be made to resolve the 1-parameterized critical pairs.

The second problem is that there may be critical pairs which can only be described by 2-parameterized sets, but we are only attempting to find a complete 1-parameterized set. So these (unavoidable) 2-parameterized critical pairs are supposed disjoint from $\mathcal{R}^{(\mathbb{N})}$, stored appropriately, and thereafter ignored until the (probable) completion of the 1-parameterized sets stops (if ever). Then, to prove that the resultant $\mathcal{R}^{(\mathbb{N})}$ is complete, we must go back and prove that all the 2-parameterized critical pairs are resolved, i.e., that the words in each pair have common $\mathcal{R}^{(\mathbb{N})}$-descendants.

We will be referring to the members of $(C^{*})^3$ as *triples*. If $\mathcal{W}$ and $\mathcal{V}$ are triples, we will write $\mathcal{W} \equiv \mathcal{V}$ whenever $\mathcal{W}_1\mathcal{W}_3 \equiv \mathcal{V}_1\mathcal{V}_3$ and $\mathcal{W}_1\mathcal{W}_2\mathcal{W}_3 \equiv \mathcal{V}_1\mathcal{V}_2\mathcal{V}_3$. We then have:

*4.1.1 Lemma.*

If $\mathcal{W}$ and $\mathcal{V}$ are triples, then $\mathcal{W} \equiv \mathcal{V}$ if and only if $\mathcal{W}_1(\mathcal{W}_2)^n\mathcal{W}_3 \equiv \mathcal{V}_1(\mathcal{V}_2)^n\mathcal{V}_3$ for all $n \in \mathbb{N}$. (The proof is trivial string manipulation and we omit it.)

$\boxed{4.1.1}$

*4.1.2 Definition.*

We say that a triple $\mathcal{T}$ is *right sided* if $\mathcal{T}_1$ is a suffix of $\mathcal{T}_1\mathcal{T}_2$. $\boxed{4.1.2}$

Let us motivate this terminology. If $\mathcal{T}$ is right sided, then we may write $\mathcal{T}_1\mathcal{T}_2 \equiv p\mathcal{T}_1$ for some word $p$. Whence $\mathcal{T}_1(\mathcal{T}_2)^n\mathcal{T}_3 \equiv (p)^n\mathcal{T}_1\mathcal{T}_3$ for all $n \in \mathbb{N}$, i.e., $\mathcal{T} \equiv (\varepsilon, p, \mathcal{T}_1\mathcal{T}_3)$.

We have:

*4.1.3 Lemma.*

Let $\mathcal{T}$ be a triple, then the following are equivalent. *(i)* $\mathcal{T}$ is right sided. *(ii)* $\mathcal{T}_1\mathcal{T}_2$ is a *proper* suffix of a word of the form $\mathcal{T}_1(\mathcal{T}_2)^* \mathcal{T}_2(1,r)$ $(0 \le r \le |\mathcal{T}_2|)$.(The proof is reasonably simple string manipulation, we will omit it.)

$\boxed{4.1.3}$

*4.1.4 Definition.*

We say that a triple $\mathcal{T}$ is *left sided* if $\mathcal{T}_3$ is a prefix of $\mathcal{T}_2\mathcal{T}_3$. $\boxed{4.1.4}$

If $\mathcal{T}$ is left sided, then $\mathcal{T}_2\mathcal{T}_3 \equiv \mathcal{T}_3\, s$ for some word $s$, whence $\mathcal{T}_1(\mathcal{T}_2)^n\, \mathcal{T}_3 \equiv \mathcal{T}_1\mathcal{T}_3\,(s)^n$ for all $n \in \mathbb{N}$, i.e., $\mathcal{T} \equiv (\mathcal{T}_1\mathcal{T}_3, s, \varepsilon)$.

*4.1.5 Lemma.*

Let $\mathcal{B}$ be a triple which is not right sided. Then, without altering the set of words $\{\mathcal{B}(n) \mid n \in \mathbb{N}\}$, we may redefine $\mathcal{B}$ so that $\mathcal{B}_1[\,|\mathcal{B}_1|\,] \not\equiv \mathcal{B}_2[\,|\mathcal{B}_2|\,]$.

*Proof:*

Recall that $\mathcal{B}$ is right sided if $\mathcal{B}_1$ is a suffix of $\mathcal{B}_1\mathcal{B}_2$, so neither $\mathcal{B}_1$ nor $\mathcal{B}_2$ is the empty word. So let $\mathcal{B} = (a\mathrm{c}, b\mathrm{c}, d)$ for some words $a, b, d$ and $\mathrm{c} \in \mathrm{C}$. Then, by lemma 4.1.1, we may redefine $\mathcal{B}$ to be $(a, \mathrm{c}b, \mathrm{c}d)$. Note that $(a, \mathrm{c}b, \mathrm{c}d)$ could not be right sided (without $a$ being a suffix of $a\mathrm{c}b \Rightarrow a\mathrm{c}$ is a suffix of $a\mathrm{c}b\mathrm{c}$ ,i.e., $\mathcal{B}$ being right sided – not so). Whence, we may repeat if necessary.

$\boxed{4.1.5}$

Let us illustrate lemma 4.1.5 with a simple example (which may well occur in practice as the left component of a rule in a completion of the surface group of a torus with 2 holes).

Let $\mathrm{C}=(a, b, c, d, a^{-1}, b^{-1}, c^{-1}, d^{-1}\}$ and put $\mathcal{B}=(b^{-1}a^{-1}ba, cb^{-1}a^{-1}a^{-1}ba, d^{-1})$.We, note that $\mathcal{B}$ is not right sided. Also:–

$$\mathcal{B}(0) \equiv (b^{-1}a^{-1}ba)(d^{-1}) \equiv (b^{-1})(a^{-1}ba\,d^{-1}),$$

and:–

$$\mathcal{B}(1) \equiv (b^{-1}a^{-1}ba)(cb^{-1}a^{-1}a^{-1}ba)(d^{-1}) \equiv (b^{-1})(a^{-1}ba\,cb^{-1}a^{-1})(a^{-1}ba\,d^{-1}).$$

Whence, by lemma 4.1.1, we may redefine $\mathcal{B}$ to be the triple:

$$\mathcal{B}=(\, b^{-1} ,\, a^{-1}bacb^{-1}a^{-1} ,\, a^{-1}bad^{-1} \,),$$

without altering $\{\, \mathcal{B}(n) \mid n \in \mathbb{N} \,\}$, so that $\mathcal{B}_1[\, |\mathcal{B}_1| \,] \not\equiv \mathcal{B}_2[\, |\mathcal{B}_2| \,]$.

Let $(\, \mathcal{B}, \mathcal{A}) \in \mathcal{R}$ with $\mathcal{B}$ and $\mathcal{A}$ being triples. In corollaries 4.3.4 and 4.3.6, we shall see that $\mathcal{B}$ need never, indeed, should never, be left or right sided. By 4.1.5, this fact allows us to propose the following (useful property of $\mathcal{R}$ which we rely on in the next section).

*4.1.6 Proposition.*

Whenever $(\, \mathcal{B}, \mathcal{A}) \in \mathcal{R}$ with $\mathcal{B}$ and $\mathcal{A}$ being triples, then $\mathcal{B}$ is stored so that $\mathcal{B}_1[\, |\mathcal{B}_1| \,] \not\equiv \mathcal{B}_2[\, |\mathcal{B}_2| \,]$. $\boxed{4.1.6}$

There are numerous practical methods by which the Knuth-Bendix completion of $\mathcal{R}^{(\mathbb{N})}$ may be speeded up. We will not bother to discuss many of these methods but, instead, refer the interested reader to the papers (Bachmair, Dershowitz), (Book, Ó' Dunlaing), and (Winkler,Buchberger).

We should just mention the criterion of *prime* critical pairs. A critical pair (2) is *prime* if $b_2$ is $\mathcal{R}^{(\mathbb{N})}$-irreducible, a critical pair (3) is *prime* if $b_1(\, |p|+1, |\, b_1| \,)$ is $\mathcal{R}^{(\mathbb{N})}$-irreducible. When attempting *finite* completions, non-prime critical pairs need not be resolved. This criterion is probably common knowledge (cf. (Gilman79)), but, interestingly, by attempting infinite completions, we cannot use it so freely. Let us defer further discussion on this topic until a more appropriate point (i.e. page 76).

The pseudo 'C' listings in the subsequent sections of this chapter comprise a short, and relatively simple, program for attempting a $P_1$ completion of $\mathcal{R}^{(\mathbb{N})}$. We hope to justify our believe that, without difficulty, the Knuth-Bendix completion procedure may be applied to a program of $P_1$ completions, and that (by the report of 4.5) such a program is worthwhile. We would like to recommend the book 'The C programming language' by (Kernighan, Ritchie) to the reader (unfamiliar with C) who wishes to implement this program.

We believe it would be helpful to the reader to conclude this section with the pseudo code for the main body of the completion program.

```
Complete( C, $\mathcal{R}$, < )
{
    //     Attempts to complete the 1-parameterized presentation $\langle C \mid \mathcal{R}^{(\mathbb{N})} \rangle$ (with respect to the
    // ShortLex ordering <) by the Knuth-Bendix method. We assume that, if $(\mathcal{B},\mathcal{A}) \in \mathcal{R}$, then $\mathcal{B}$ is not
    // left or right sided (cf [1]page 75), and that $\mathcal{B}_1[\,|\mathcal{B}_1|\,] \neq \mathcal{B}_2[\,|\mathcal{B}_2|\,]$ (cf 4.1.5).
    $S=\{0\}\cup C$; $s=0$; $\tau(p,c)=c$ for all $p\in S$ and $c\in C$; $f(p)=\varnothing$ for all $p\in S$;
    modify$(((C,S,s,\varnothing,\tau),f),\mathcal{B})$ as each $(\mathcal{B},\mathcal{A})$ of $\mathcal{R}$ is input;
    // ( $((C,S,s,\varnothing,\tau),f)$ is a *reductor* of $\mathcal{R}^{(\mathbb{N})}$ (cf pp 49-53).)
    for ($i=1$; $i\leq|\mathcal{R}|$; $i=i+1$)
        for ($j=i$; $j\leq|\mathcal{R}|$; $j=j+1$)
        {
            $(\mathcal{B}_1,\mathcal{A}_1)=i^{th}$ member of $\mathcal{R}$; $(\mathcal{B}_2,\mathcal{A}_2)=j^{th}$ member of $\mathcal{R}$;
            if ($\mathcal{B}_1$, $\mathcal{A}_1$, $\mathcal{B}_2$ and $\mathcal{A}_2$ are triples)
            {
                CritPair$_1((\mathcal{B}_1,\mathcal{A}_1),(\mathcal{B}_2,\mathcal{A}_2))$; CritPair$_1((\mathcal{B}_2,\mathcal{A}_2),(\mathcal{B}_1,\mathcal{A}_1))$;
                // (are described on page 88.)
                CritPair$_2((\mathcal{B}_1,\mathcal{A}_1),(\mathcal{B}_2,\mathcal{A}_2))$; CritPair$_2((\mathcal{B}_2,\mathcal{A}_2),(\mathcal{B}_1,\mathcal{A}_1))$;
                // (are described on page 92, and may store 2-parameterized critical pairs disjoint form $\mathcal{R}$.)
            }
            else {
                CritPair$((\mathcal{B}_1,\mathcal{A}_1),(\mathcal{B}_2,\mathcal{A}_2))$; CritPair$((\mathcal{B}_2,\mathcal{A}_2),(\mathcal{B}_1,\mathcal{A}_1))$;
                // (are described on pp 80-82.)
            }
            //     The procedures CritPair, CritPair$_1$ and CritPair$_2$, collectively described in section 4.4,
            // compute and resolve the critical pairs by calling the following predefined procedures of
            // sections 4.2 and 4.3. reduce(word) (page 55); reduce(triple) (page 56); secure(word,word)
            // (page 60); resolve(word,word) (page 63); modify(reductor,word) (page 51);
            // resolve(triple,triple) (page 69); and modify(reductor,triple) (page 52).
     for (each set, $\{((\mathcal{B}(n),\mathcal{A}(n)) \mid n\in\mathbb{N}^2\}$, of 2-parameterized critical pairs (found by CritPair$_2$))
         if ( resolved?$(\mathcal{B},\mathcal{A})$==0 )
             writeln('cannot prove',{ ( (',$\mathcal{B}$,'(n),',$\mathcal{A}$,'(n)) | $n\in\mathbb{N}^2$ } are resolved');
    //resolved?$(\mathcal{B},\mathcal{A})$, described on page 77, attempts to prove that the critical pairs of $\{((\mathcal{B}(n),\mathcal{A}(n)) \mid n\in\mathbb{N}^2\}$
    // are resolved. If resolved?$(\mathcal{B},\mathcal{A})$==1 for all the quintuples $(\mathcal{B},\mathcal{A})$, then $\langle C \mid \mathcal{R}^{(\mathbb{N})} \rangle$ is complete.
}
```

## 4.2

## *Computing the functions reduce(word) and reduce(triple)*

Whilst $\mathcal{R}^{(\mathbb{N})}$ is not complete it is possible for a word to have distinct $\mathcal{R}^{(\mathbb{N})}$-irreducible descendants. So we should agree on one method of computing an (invariant) $\mathcal{R}^{(\mathbb{N})}$-irreducible descendant, reduce( $w$ ), for each word $w$ (as a by-product, reduce() is, of course, the function rep() provided $\mathcal{R}^{(\mathbb{N})}$ is successfully completed).

We define *reduce( word $w$ )* as follows. Search for the first subword, $s$, of $w$ for which there is some ( $\mathcal{B}$ , $\mathcal{A}$ ) in $\mathcal{R}$ and n$\in\mathbb{N}$ with $s \equiv \mathcal{B}$(n). There may be no such pair but, if it exists, then replace the subword $s$ by $\mathcal{A}$(n). Repeat this process until no more substitutions can be made. The resulting ($\mathcal{R}^{(\mathbb{N})}$-irreducible) word is defined to be reduce( $w$ ).

We will need an analog of reduce(word) which, for a triple $\mathcal{T}$, computes a triple, reduce( $\mathcal{T}$ ), so that the words { $\mathcal{T}$(n) | n$\in\mathbb{N}$ } are simultaneously $\mathcal{R}^{(\mathbb{N})}$-reduced to the words { (reduce( $\mathcal{T}$ ))(n) | n$\in\mathbb{N}$ }. This is a problem because there is no reason why we should be able to define reduce( $\mathcal{T}$ ) so that every word in { (reduce( $\mathcal{T}$ ))(n) | n$\in\mathbb{N}$ } is $\mathcal{R}^{(\mathbb{N})}$-irreducible. We really cannot hope to define *reduce( triple $\mathcal{T}$ )* much better than as follows.

If $\mathcal{T} \equiv$ ( $a$ , $b$ , $c$ ) with $a$ (respectively $b$ , $c$ ) being $\mathcal{R}^{(\mathbb{N})}$-reducible, then redefine $\mathcal{T}$ to be ( reduce($a$) , $b$ , $c$ ) (respectively ( $a$ , reduce($b$) , $c$ ), ( $a$ , $b$ , reduce($c$) ) ).

Suppose there is a pair of triples ( $\mathcal{B}$ , $\mathcal{A}$ )$\in\mathcal{R}$, words $p$ , $s$ and $\kappa$,r$\in\mathbb{N}$ such that $\mathcal{T}(0) \equiv p\mathcal{B}(\kappa)s$ and $\mathcal{T}(1) \equiv p\mathcal{B}(\kappa+r)s$. Then, by lemma 4.1.1, $\mathcal{T}(n) \equiv p\mathcal{B}(\kappa+nr)s$ for all n$\in\mathbb{N}$, and we redefine $\mathcal{T}$ to be ( $p\mathcal{A}_1(\mathcal{A}_2)^{\kappa}$ , $(\mathcal{A}_2)^{r}$ , $\mathcal{A}_3 s$ ).

Repeat while such ( $a$ , $b$ , $c$ )'s or ( $\mathcal{B}$ , $\mathcal{A}$ )'s exist.

The program frequently calls a procedure which we have named *secure( word, word )*. We may only call secure( $w$ , $v$ ) when $w > v$, then the reduction $w \rightarrow^{*} v$ would be secured (by allocating at most one new rewrite rule to $\mathcal{R}^{(\mathbb{N})}$).

We can suggest a way of considerably speeding up the search for possible subwords belonging to left( $\mathcal{R}^{(\mathbb{N})}$ ). The method is for the program to maintain a *reductor of* $\mathcal{R}^{(\mathbb{N})}$).

*4.2.1 Definition.*

We say that the pair ((C,S,s,∅,τ),f) is a *reductor* (of $\mathcal{R}^{(\mathbb{N})}$) if (i) and (ii) hold as follows. (i) (C,S,s,∅,τ) is a deterministic fsa called the *reduction automaton* of the reductor (we refer the reader to definition 1.1 of a deterministic fsa). (ii) f is a map, $f : S \longrightarrow$ power set of $|\mathcal{R}|$, such that, if $p$ is any word, then $r \in f(\tau(p))$ if and only if $p$ has a suffix $\mathcal{S}(n)$, where $\mathcal{S}$ is the left component of the $r^{th}$ member of $\mathcal{R}$, and $n \in \mathbb{N}$.

$\boxed{4.2.1}$

Let us describe the part a reductor plays in the program. If $w$ is any word, then we will know that $w$ is $\mathcal{R}^{(\mathbb{N})}$-reducible if and only if $w$ has a prefix $p$ with $f(\tau(p)) \neq \varnothing$. Also, if $p$ were such a prefix and $r \in f(\tau(p))$, then, by the definition of τ, we know that $p$ has a suffix, $\mathcal{S}(n)$, with $\mathcal{S}$ being the left component of the $r^{th}$ member of $\mathcal{R}$, but, as yet, we do not know the value of n. This is the subject of the next lemma.

*4.2.2 Lemma.*

*(i)* If $\mathcal{S}$ is word, then, by convention, n=0.

*(ii)* If $\mathcal{S} = ( \mathcal{a}c_0 , \mathcal{b}c_1 , \mathcal{c} )$ is a triple (with $c_0,c_1 \in C$), then n is the least integer such that:

$$p[\, |p| - |\mathcal{c}| - n(|\mathcal{b}|+1)\, ] = c_0 .$$

*Proof of (ii):*

Let $pos(r) = |p| - |\mathcal{c}| - r(|\mathcal{b}|+1)$, then, provided $r \geq 0$ and $|p| - |\mathcal{c}| - r(|\mathcal{b}|+1) \geq 0$, we note that $p[pos(r)]$ is the generator in position $|\mathcal{c}| + r(|\mathcal{b}|+1) + 1$ of $p$ from the *right*. As the word $\mathcal{a}c_0(\mathcal{b}c_1)^n \mathcal{c}$ is known to be a prefix of $p$, it is not difficult to see that, for each r in the range $0 \leq r < n$: $p[pos(r)] = c_0$; and, for r=n: $p[pos(r)] = c_1$. As $c_0 \neq c_1$ (cf proposition 4.1.6), the result follows.

$\boxed{4.2.2}$

Although the reductor is a powerful tool, allowing us to describe fast procedures for reducing words and triples, it does have a drawback, i.e., it must be modified whenever new rules are adjoined to $\mathcal{R}^{(N)}$. We must deal with this problem by suggesting procedures for the modification of the reductor. Although short and fast, these procedures, theoretically, may require substantial memory with which to store the transition function (as a |S| by |C| array, where S is the state set); but, in practice, memory is rarely a problem and we will not bore the reader with theoretical memory estimates (which are, anyway, difficult).

The formal proofs of correctness of the procedures we are about to describe require numerous inductive arguments, and are not trivial. We have decided to forgo all formal proofs of correctness, as we believe they would simply belie the simplicity of this program without adding anything to the theory.

We begin by describing a subprocedure, suffix, which is called by both of the modification procedures.

```
suffix( $((C,S,s,\varnothing,\tau),f), b, P^{(0)}$ )
{
    // $P^{(0)}$ is a subset of S. Let suff be the set of words with the following property. If $w$ is any word,
    // then $\tau(w) \in P^{(0)}$ if and only if $w$ has a suffix belonging to suff.
    //     We will redefine S,$\tau$ and f as follows. $((C,S,s,\varnothing,\tau),f)$ will remain a reductor for $\mathcal{R}^{(N)}$, but S
    // will have a subset, $P^{(1)}$, with the following property. If $w$ is any word, then $\tau(w) \in P^{(1)}$ if and only
    // if $w$ has a suffix of the form $sb$ where $s$ belongs to suff.
    //     We note that, if $P^{(0)}=S$, then, as $\tau(\varepsilon)=s$ (i.e. the start state), and s belongs S, so $\varepsilon$ has a suffix
    // belonging to suff, i.e., $\varepsilon \in$ suff. As all words have the empty word as a suffix, so $P^{(1)}$ will be
    //  such that $\tau(w) \in P^{(1)}$ if and only if $w$ has $b$ as a suffix.
    for ( $i=1$; $i \le |b|$; $i=i+1$ )
    {
        $P^{(1)}=\varnothing$;
        $c_i = b[i]$;
        partition $P^{(0)}$ as $pre_1 \cup pre_2 \cup \dots \cup pre_n$ so that $\tau$ is constant on each $pre_j \times \{c_i\}$;
```

the result follows.

$\boxed{4.2.2}$

Although the reductor is a powerful tool, allowing us to describe fast procedures for reducing words and triples, it does have a drawback, i.e., it must be modified whenever new rules are adjoined to $\mathcal{R}^{(N)}$. We must deal with this problem by suggesting procedures for the modification of the reductor. Although short and fast, these procedures, theoretically, may require substantial memory with which to store the transition function (as a |S| by |C| array, where S is the state set); but, in practice, memory is rarely a problem and we will not bore the reader with theoretical memory estimates (which are, anyway, difficult).

The formal proofs of correctness of the procedures we are about to describe require numerous inductive arguments, and are not trivial. We have decided to forgo all formal proofs of correctness, as we believe they would simply belie the simplicity of this program without adding anything to the theory.

We begin by describing a subprocedure, suffix, which is called by both of the modification procedures.

```
suffix( $((C,S,s,\varnothing,\tau),f), б, P^{(0)}$ )
{
    // $P^{(0)}$ is a subset of S. Let suff be the set of words with the following property. If $w$ is any word,
    // then $\tau(w) \in P^{(0)}$ if and only if $w$ has a suffix belonging to suff.
    //     We will redefine S,$\tau$ and f as follows. $((C,S,s,\varnothing,\tau),f)$ will remain a reductor for $\mathcal{R}^{(N)}$, but S
    // will have a subset, $P^{(1)}$, with the following property. If $w$ is any word, then $\tau(w) \in P^{(1)}$ if and only
    // if $w$ has a suffix of the form $sб$ where $s$ belongs to suff.
    //     We note that, if $P^{(0)}=S$, then, as $\tau(\varepsilon)=s$ (i.e. the start state), and s belongs S, so $\varepsilon$ has a suffix
    // belonging to suff, i.e., $\varepsilon \in$ suff. As all words have the empty word as a suffix, so $P^{(1)}$ will be
    //  such that $\tau(w) \in P^{(1)}$ if and only if $w$ has $б$ as a suffix.
    for ( i=1;  i≤|$б$| ;  i=i+1 )
    {
        $P^{(1)}=\varnothing$;
        $c_i= б[i]$;
        partition $P^{(0)}$ as $pre_1 \cup pre_2 \cup \dots \cup pre_n$ so that $\tau$ is constant on each $pre_j \times \{c_i\}$;
        for ( j=1;  j<n;  j=j+1 )
        {
            im=$\tau(p,c_i)$ where p is any state of $pre_j$ ;
            if ( im==$\tau(p,c)$ for some $p\in P^{(0)}$ or $c\in C-\{c_i\}$ )
            {
                let ns be a state not already in S;
                f(ns)=f(im);
                S=S$\cup$\{ns\};
                $P^{(1)}=P^{(1)}\cup\{ns\}$;
                $\tau(p,c_i)$=ns for all $p\in pre_j$ ;
                $\tau(ns,c)=\tau(im,c)$ for all $c\in C$;
            }
            else
                $P^{(1)}=P^{(1)}\cup\{im\}$;
        }
        for ( j=1; j<n; j=j+1 )
        {
            let $p\in pre_j$ ;
            $\tau(\tau(p,c_i),c_i)=\tau(im,c_i)$ for all $p\in pre_j$ ;
        }
        $P^{(0)}=P^{(1)}$;
    }
    // We will need to return $P^{(1)}$.
    return $P^{(1)}$;
}
```

suffix( )

```
modify( ((C,S,s,$\varnothing$,$\tau$) ,f), word $\beta$ )
{
    //     The parameters are as follows: ((C,S,s,$\varnothing$,$\tau$),f) is a reductor for $\mathfrak{R}^{(N)}$ and $\beta$ is a non-empty
    // word. If ( $\beta$ , $\alpha$ ) is a rule, then we modify ((C,S,s,$\varnothing$,$\tau$),f) to that of a reductor for $\mathfrak{R}^{(N)}\cup\{(\beta,\alpha)\}$.
    //     We first put:-
    $P^{(1)}$=suffix( (C,S,s,$\varnothing$,$\tau$) , S, $\beta$ );
    // Then, with reference to the description of procedure suffix, $P^{(1)}\subseteq S$ will be such that
    // $((C,S,s,\varnothing,\tau),f)$ is a reductor for $\mathcal{R}^{(\mathbb{N})}$ with the additional property that $\tau(w)\in P^{(1)}$ if and only if $w$
    // has $b$ as a suffix. Whence, cf definition 4.2.1, we need only redefine
    for (all $p\in P^{(1)}$)
        $f(p)=f(p)\cup\{|\mathcal{R}|+1\}$;
}
```

modify( reductor , word )

```
*modify( $((C,S,s,\varnothing,\tau)\,f)$, triple $\mathcal{B}$ )*
{
    //     The parameters are as follows: $((C,S,s,\varnothing,\tau),f)$ is a reductor for $\mathcal{R}^{(\mathbb{N})}$ and $\mathcal{B}$ is a triple which is
    // neither left nor right sided. If $\mathcal{A}$ is a triple, then we modify $((C,S,s,\varnothing,\tau),f)$ to that of a reductor for
    // $\mathcal{R}^{(\mathbb{N})}\cup\{\ (\ \mathcal{B}(n)\ ,\ \mathcal{A}(n)\ )\mid n\in\mathbb{N}\ \}$.
    //     Note that, by definition 4.1.2 and 4.1.4, none of the components $\mathcal{B}_1$, $\mathcal{B}_2$ or $\mathcal{B}_3$ of $\mathcal{B}$ are empty.
    // We begin by putting:-
    $P^{(1)}$=suffix( $(C,S,s,\varnothing,\tau)$, $\mathcal{B}_1\mathcal{B}_2$ , S );
    // so that $P^{(1)}\subseteq S$ will be such that $((C,S,s,\varnothing,\tau),f)$ remains a reductor for $\mathcal{R}^{(\mathbb{N})}$, but with the
    // additional property that $\tau(w)\in P^{(1)}$ if and only if $w$ has $\mathcal{B}_1\mathcal{B}_2$ as a suffix.
    //     The following while-loop further redefines $\tau$ and S so that $((C,S,s,\varnothing,\tau),f)$ remains a reductor
    // for $\mathcal{R}^{(\mathbb{N})}$, but so that there is a subset $P^{(2)}$ of S with the property that $\tau(w)\in P^{(2)}$ if and only if $w$
    // has a suffix of the form $\mathcal{B}_1\mathcal{B}_2(\mathcal{B}_2)^*$.
    $\sigma$=identity map on S;
    i=0;
    $P_0=P^{(1)}$;
    while ( $P_i=\varnothing$ )
    {
        $P_{(i+1)}=\varnothing$;
        $c_{(i+1)}=\mathcal{B}_2[\,(i \bmod |\mathcal{B}_2|\,)+1]$;
        partition $P_{(i+1)}$ as $pre_1\cup pre_2\cup\ldots\cup pre_n$ so that $\tau$ is constant on each $pre_j\times\{c_i\}$;
        for ( j=1;  j<n;  j=j+1 )
        {
            im=$\tau(p,c_i)$ where p is any state of $pre_j$ ;
            if ( $\sigma(im)==\sigma(\bar{p})$ for some $\bar{p}\in P_\kappa$ with $0\le\kappa<i$ and $i+1=\kappa \bmod |\mathcal{B}_2|$ )
                $\tau(p,c_{(i+1)})=\bar{p}$ for all $p\in pre_j$ ;
        else
        {
            let ns be a state not already in S;
            f(ns)=f(im);
            $\sigma$(ns)=$\sigma$(im);
            S=S$\cup$\{ns\};
            $P_{(i+1)}=P_{(i+1)}\cup\{ns\}$;
            $\tau(ns,c)=\sigma(\tau(im,c))$ for all $c\in C$;
            $\tau(p,c_{(i+1)})=ns$ for all $p\in pre_j$ ;
        }
    }
    if ( $P_{(i+1)}\neq\varnothing$ )
        i=i+1;
}
$P^{(2)}=\bigcup_{i \bmod |\mathcal{B}_2|=0} P_i$ ;
//    We now put:-
$P^{(3)}$=suffix( ((C,S,s,$\varnothing$,$\tau$),f) , $\mathcal{B}_3$, $P^{(2)}$ );
// Then ((C,S,s,$\varnothing$,$\tau$),f) remains a reductor for $\mathcal{R}^{(\mathbb{N})}$, but $P^{(3)}$ will be a subset of S with the property
// that $\tau(w)\in P^{(3)}$ if and only if $w$ has a suffix of the form $\mathcal{B}_1\mathcal{B}_2(\mathcal{B}_2)^*\mathcal{B}_3$. Whence, cf definition
// 4.2.1, by redefining:
for (all $p\in P^{(3)}$)
    $f(p)=f(p)\cup\{|\mathcal{R}|+1\}$;
// ((C,S,s,$\varnothing$,$\tau$),f) will be a reductor for $\mathcal{R}^{(\mathbb{N})}\cup\{ ( \mathcal{B}(n) , \mathcal{A}(n) ) \mid n\geq 1 \}$. We want a reductor for
// $\mathcal{R}^{(\mathbb{N})}\cup\{ ( \mathcal{B}(n) , \mathcal{A}(n) ) \mid n\in \mathbb{N} \}$, so, to finish, we need only call:
modify( ((C,S,s,$\varnothing$,$\tau$),f) , $\mathcal{B}_1\mathcal{B}_3$ );
}
```

modify( reductor , triple )

We are almost ready to describe the pseudo code for the functions *reduce(word)*, *reduce(triple)* and for the procedure *secure(word, word)*. Let us begin, however, with two short, but labour saving, procedures, and with the procedures for adjoining rules to $\mathcal{R}^{(\mathbb{N})}$.

```
shiftleft ( triple $\mathcal{T}$ )
{
    //   Without altering the set of words { $\mathcal{T}(n) \mid n \in \mathbb{N}$ }, we redefine $\mathcal{T}$ so that $|\mathcal{T}_3|$ is maximal. By
    // lemma 4.1.1, we may achieve this by the following while-loop.
    while ( $|\mathcal{T}_1|$ ) > 0 and $\mathcal{T}_1[\,|\mathcal{T}_1|\,] = \mathcal{T}_2[\,|\mathcal{T}_2|\,]$ )
    {
        chr = $\mathcal{T}_1[\,|\mathcal{T}_1|\,]$;
        $\mathcal{T}_1 = \mathcal{T}_1\ (1,\ |\mathcal{T}_1| - 1)$;
        $\mathcal{T}_2 =$ chr ( $\mathcal{T}_2\ (1,\ |\mathcal{T}_2| - 1)$);
        $\mathcal{T}_3 =$ chr $\mathcal{T}_3$ ;
    }
}
```

shiftleft( triple $\mathcal{T}$ )

```
shiftright ( triple $\mathcal{T}$ )
{
    //   Without altering the set of words { $\mathcal{T}(n) \mid n \in \mathbb{N}$ }, we redefine $\mathcal{T}$ so that $|\mathcal{T}_1|$ is maximal. By
    // lemma 4.1.1, we may achieve this by the following while-loop.
    while ( $|\mathcal{T}_3|$ ) > 0 and $\mathcal{T}_3[1] = \mathcal{T}_2[1]$ )
    {
        chr = $\mathcal{T}_3[1]$;
        $\mathcal{T}_1 = \mathcal{T}_1$chr
        $\mathcal{T}_2 = \mathcal{T}_2\ (2, |\mathcal{T}_2|)$ chr;
        $\mathcal{T}_3 = \mathcal{T}_3\ (2, |\mathcal{T}_3|)$ ;
    }
}
```

shifright( triple $\mathcal{T}$ )

```
adjoin( rule (($b$ ),($a$ )) )
{
    //   Adjoins the rule (($b$ ),($a$ )) to $\mathcal{R}$ where $b$ and $a$ are words.
    modify( ((C,S,s,∅,t) ,f), $b$ );
$\mathcal{R}=\mathcal{R}\cup\{((b),(a))\};$
}
$\boxed{\text{adjoin( )}}$

adjoin( rule $((\mathcal{B}),(\mathcal{A}))$ )
{
    //   Adjoins the rule $((\mathcal{B}),(\mathcal{A}))$ to $\mathcal{R}$ where $\mathcal{B}$ and $\mathcal{A}$ are triples. Recall that, by proposition 4.1.6, we
    // want $\mathcal{B}$ to be such that $\mathcal{B}_1[\,|\mathcal{B}_1|\,]\neq\mathcal{B}_2[\,|\mathcal{B}_2|\,]$ (so that we may apply lemma 4.2.2), but we can achieve
    // this by calling shiftleft.
    shiftleft( $\mathcal{B}$ );
    modify( $((C,S,s,\varnothing,\tau),f)$, $\mathcal{B}$ );
    $\mathcal{R}=\mathcal{R}\cup\{((\mathcal{B}),(\mathcal{A}))\};$
}
$\boxed{\text{adjoin( )}}$
```

We now describe the pseudo code for the functions *reduce( word )*, *reduce( triple )*, and for the procedure *secure( word, word )* (all of which refer to lemma 4.2.2)). These procedures are actually faster than those we used in practice, this is because, in practice, we referred to a less powerful tool (but still akin) to the reductor.

```
reduce ( word $w$ )
{
    // We define $s_{(0)}$ to be the start state, s, of the reduction automaton.
    $s_{(0)}=s;$
    $\kappa=0;$
    // By definition, $s_{(0)}=\tau(w(1,0))$;  we proceed to compute $s_{(\kappa)}=\tau(w(1,\kappa))$, for all $1<\kappa\leq|\,w\,|$.
    while ( $\kappa<|\,w\,|$ )
    {
        $\kappa=\kappa+1;$
        // By definition 1.1, $\tau(w(1,\kappa))=\tau(w(1,\kappa-1)),\ w[\kappa])=\tau(\,s_{(\kappa-1)},\ w[\kappa])$, so we put:
        $s_{(\kappa)}=\tau(\,s_{(\kappa-1)},\ w[\kappa]);$
        if ( $f(s_{(\kappa)})\neq\varnothing$ )
        {
            // then, by definition 4.2.1, if:
            $r \in f(s_{(\kappa)})$;
            // and:
            $p = u(1,\kappa)$;
            // we know that $p$ will have a suffix $s = \mathcal{B}(n)$ where:
            $(\mathcal{B}, \mathcal{A}) = r^{th}$ pair of $\mathcal{R}$;
            // and $n \in \mathbb{N}$.
            if ( $\mathcal{B}$ and $\mathcal{A}$ are words )
            // then, by convention:-
                $n=0$;
            else
                // $\mathcal{B}$ and $\mathcal{A}$ are triples and we must calculate n by the method described in 4.2.2, i.e.,
                for ( $n=0$; $p[\kappa - |\mathcal{B}_3| - n|\mathcal{B}_2|] \neq \mathcal{B}_1[|\mathcal{B}_1|]$ ); $n=n+1$);
            // We now know that $s = \mathcal{B}(n)$, and therefore $w = p(1, \kappa - |\mathcal{B}(n)|)\ \mathcal{B}(n)\ u(\kappa+1, |w|)$. We put:-
            $w = p(1, \kappa - |\mathcal{B}(n)|)\ \mathcal{A}(n)\ u(\kappa+1, |w|)$;
            // and note that, for all $1<i\leq\kappa - |\mathcal{B}(n)|$, $\tau(u(1,i))=s_{(\kappa)}$. Thus we need only calculate $s_{(i)}$ anew
            // beginning at $i=\kappa - |\mathcal{B}(n)|$, so we put:-
            $\kappa=\kappa - |\mathcal{B}(0)|$;
        }
    }
    // We now have $\kappa=|w|$ and, for all $1<i\leq\kappa=|w|$, $f(s_{(\kappa)})\neq\varnothing$. Thus, by the definition of $\tau$ and f, we
    // know that $w$ will be $\mathcal{R}^{(\mathbb{N})}$-reduced, and so we may return $w$.
    return $w$;
}
```

reduce( word $w$ )

```
reduce ( triple $\mathcal{T}$ )
{
    do
    {
        // Store a copy of $\mathcal{T}$ (for future comparison).
        $\mathcal{T}copy = \mathcal{T}$;
```

```
//    We begin by searching for triples ( $a$, $b$, $c$ ) where $\mathcal{T}$ = ( $a$, $b$, $c$ ), and $a$, respectively $b$ or $c$,
// is $\mathcal{R}^{(\mathbb{N})}$-reducible. For each such triple, we redefine $\mathcal{T}$ to be ( reduce($a$), $b$ , $c$ ), respectively
// ( $a$, reduce($b$) , $c$ ) or ( $a$ , $b$ , reduce($c$) ), and continue if possible.
//    We first need to redefine $\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{T}_3$ , without altering { $\mathcal{T}_1( \mathcal{T}_2 )^n \mathcal{T}_3$ | n∈ ℕ }, so that $\mathcal{T}_1$
// is of minimal length. We do this calling shiftleft( $\mathcal{T}$ ).
shiftleft( $\mathcal{T}$ );
$a$= $(\mathcal{T})_1$ ; $b$= $(\mathcal{T})_2$ ; $c$= $(\mathcal{T})_3$ ;
//    In each pass of the following do-loop, we will attempt one of (i) or (ii) as follows. (i), if
// possible, strictly reduce at least one of $\mathcal{T}$'s components, then further redefine $\mathcal{T}$ so that the $\mathcal{T}_1$
// component is of minimal length. (ii), provided (i) was not possible, then, if possible, redefine
// $\mathcal{T}$ so as to increase | $\mathcal{T}_1$ | by 1 and decrease | $\mathcal{T}_3$ | by 1. If neither (i) nor (ii) was possible, then
// we exit the do-loop. It is not difficult to see that the do-loop must stop, and that, when it
// stops, $\mathcal{T}$ could not be further redefined (without altering { $\mathcal{T}_1( \mathcal{T}_2 )^n \mathcal{T}_3$ | n∈ ℕ }) so that one of
// its components is reducible (as required).
do
{
    continue=0;
    $\mathcal{T}$ = ( $a$ , $b$ , $c$ );
    $a$ =reduce($a$);
    $b$ =reduce($b$);
    $c$ =reduce($c$);
    if ( $(\mathcal{T})_1 > a$ or $(\mathcal{T})_2 > b$ or $(\mathcal{T})_3 > c$ )
    {
        // We may be able to achieve (i) by redefining $a$, $b$ and $c$, without altering
        // { $a ( b )^n c$ | n∈ ℕ }, so that $a$ is of minimal length. By lemma 4.1.1, we may do this
        // by the following while-loop, i.e., shiftleft( ( $a$ , $b$ , $c$ ) )
        while (| $a$ | > 0 and $a$[ | $a$ | ] = $b$[ | $b$ | ] )
        {
            chr = $a$[ | $a$ | ] ;
            $a$ = $a$ (1, | $a$ | -1);
            $b$ = chr ( $b$(1, | $b$ | -1));
            $c$ = chr $c$ ;
            // We have achieved (i), and so will need to make at least one more pass of the
            // do-loop, i.e.,
                continue=1;
            }
            $\mathcal{T} = (a, b, c)$;
        }
        if (continue=0 and $|c| > 0$ and $c[1] = b[1]$ )
        {
            //     We may still be able to achieve (ii) by redefining $a$, $b$ and $c$, without altering
            // $\{ a(b)^n c \mid n \in \mathbb{N} \}$, so as to increase $|a|$ by 1 and decrease $|c|$ by 1. By lemma 4.1.1,
            // we may do this as follows.
            chr = $c[1]$ ;
            $a = a$chr ;
            $b = b(2, |b|)$chr ;
            $c = c(2, |c|)$;
            continue=1;
        }
    }
    while (continue==1);
    //     We now search for possible triples ( $\mathcal{B}, \mathcal{A}$ ) of $\mathcal{R}$ with $\mathcal{T}_1(\mathcal{T}_2)^n\mathcal{T}_3 = p\,\mathcal{B}_1(\mathcal{B}_2)^{(nr+\kappa)}\mathcal{B}_1 s$
    // for some words $p$ , $s$ and $\kappa, r \in \mathbb{N}$, and, for each such ( $\mathcal{B}, \mathcal{A}$ ), replace $\mathcal{T}$ by
    // ( $p\,\mathcal{A}_1(\mathcal{A}_2)^\kappa, (\mathcal{A}_2)^r, \mathcal{A}_3$ ).
    //     By lemma 4.1.1, we know that ( $\mathcal{B}, \mathcal{A}$ ) is such a triple if and only if
    //                         (1) $\mathcal{T}(0) = p\mathcal{B}(\kappa)s$ and $\mathcal{T}(1) = p\mathcal{B}(\kappa+r)s$ .
    // Note that r must be divisible by $|\mathcal{T}_2|$ , actually $r = |\mathcal{T}_2| / |\mathcal{B}_2|$. So, let us begin by putting:
    $t^0 = \mathcal{T}(0)$;
    $t^1 = \mathcal{T}(1)$;
    // and then setting $s_{(0)}$ equal to the start state of the reduction automaton, i.e.,
    $s_{(0)} = s$;
    // We proceed to compute all $s_{(i)} = \tau(t^0(1,i))$, for $1 < i \leq |t^0|$ .
    i=0;
    while ( $i < |t^0|$ )
    {
        i=i+1;
        // By definition 1.1, $\tau(t^0(1,i)) = \tau(\tau(t^0(1,i-1)), t^0[i])$, so we put: -
        $s_{(i)} = \tau(s_{(i-1)}, t^0[i])$;
    if ( $f(s_{(i)})\neq\varnothing$ )
        // then, by definition 4.2.1, we know that:
        for( all $j\in f(s_{(i)})$ )
            if ( ( $\mathcal{B}$ , $\mathcal{A}$ )= $j^{th}$ pair of $\mathcal{R}$ with $\mathcal{B}$ and $\mathcal{A}$ being triples and $|\mathcal{T}_2|$ divisible by $|\mathcal{B}_2|$ ) );
            {
                // then $\iota^0(1,i)$ has a suffix $s=\mathcal{B}(\kappa)$ for some $\kappa\in\mathbb{N}$. Actually, we need to know
                // the words $p$ and $s$ and $\kappa\in\mathbb{N}$ where $\iota^0(1,i)=p\mathcal{B}(\kappa)$ and $\iota^0=p\mathcal{B}(\kappa)s$ . We begin
                // by calculating $\kappa$ by the method described in 4.2.2, i.e.,
                for ( $\kappa=0$; $(\iota^0(1,i))(i-|\mathcal{B}_3|-n|\mathcal{B}_2||\neq\mathcal{B}_1(|\mathcal{B}_1||$ ), $\kappa=\kappa+1$);
                // We now know that $\iota^0(1,i)$ has suffix $\mathcal{B}(\kappa)$, and we need to define $p$ and $s$ so
                // that:
                // $\mathcal{T}(0)=p\,\mathcal{B}(\kappa)s$ , i.e., we need:-
                $p=\iota^0(1,i-|\mathcal{B}(\kappa)|)$;
                $s=\iota^0(i+1,|\iota^0|)$;
                // We then define:-
                $r=|\mathcal{T}_2|/|\mathcal{B}_2|$ .
                // and check
                if ( $\mathcal{T}(1)=p\,\mathcal{B}(\kappa+r)s$)
                {
                    // Then (1) does hold, so we may redefine:-
                    $\mathcal{T}=(\,p\,\mathcal{A}_1(\mathcal{A}_2)^{\kappa},\,(\mathcal{A}_2)^{r},\,\mathcal{A}_3\,)$;
                    // and reset:-
                    i=0;
                    $\iota^0=\mathcal{T}(0)$;
                    $\iota^1=\mathcal{T}(1)$;

                    // We now break the for-loop.
                    break;
                }
            }
}
//    As a last refinement, we wish to search for all possible suffixes, $s$ of $\mathcal{T}_1$ so that:
//                          (2)  $s\,\mathcal{T}_2\rightarrow_{\mathcal{R}(\mathbb{N})}\mathcal{T}_2\,s$ ,
// and all possible prefixes, $p$ of $\mathcal{T}_3$ so that:
        //                          (3)  $\mathcal{T}_2\, p \rightarrow_{\mathcal{R}^{(N)}} p\, \mathcal{T}_2$.
        // With (2), we could define $p$ by $\mathcal{T}_1 = p\, s$, and so replace $\mathcal{T}$ by ( $p$, $\mathcal{T}_2$, $s\, \mathcal{T}_3$ ). With (3),we could
        // define $s$ by $\mathcal{T}_3 = p\, s$, and so replace $\mathcal{T}$ by ( $\mathcal{T}_1\, p$, $\mathcal{T}_2$, $s$ ). In practice, though, neither is really
        // feasible. It is feasible, however, to check whether $\mathcal{T}_1$ has a suffix $s$, satisfying (2), and, in
        // addition, with $s^{-1}$ being a prefix of $\mathcal{T}_2$. Also, we could feasibly check whether $\mathcal{T}_3$ has a prefix
        // $p$, satisfying (3), and with $p^{1}$ being a suffix of $\mathcal{T}_1$. We have found that both situations occur,
        // not infrequently, in practice, so it would be worthwhile implementing both checks.
    }
    while ( $\mathcal{T}_{copy} \neq \mathcal{T}$ );
    return $\mathcal{T}$;
}
```

reduce( triple $\mathcal{T}$ )

```
secure ( word $w$ , word $u$ )
{
    //    The parameters, $w$ and $u$ , must be such that $w > u$, and, by adjoining at most a single rewrite
    // rule to $\mathcal{R}^{(N)}$, we aim to secure the reduction  $w \rightarrow_{\mathcal{R}^{(N)}}{}^{*} u$. We begin by defining $s_{(0)}$ to be the
    // start state, $s$, of the reduction automaton.
    $s_{(0)} = s$;
    $\kappa = 0$;
    //    By definition, $s_{(0)} = \tau(x(1,0))$ and we proceed to compute, in the following while-loop, all
    // $s_{(\kappa)} = \tau(w(1,\kappa))$, $1 \le \kappa \le |\, w\,|$. We may strip common prefixes of $w$ and $u$, and replace $w(1,\kappa)$ by
    // words $v$ with  $w(1,\kappa) \rightarrow_{\mathcal{R}^{(N)}} v$, but  the (inductive) hypotheses will always be (i) and (ii) as
    // follows. (i), that $w$ and $u$ do not have a non-trivial  common prefix or suffix, and, (ii), we do not
    // have $w(1,\kappa-1) \rightarrow_{\mathcal{R}^{(N)}} v$ with $v\, w(\kappa \rfloor w|\,) > u$.
    //    Note that, when $\kappa = |\, w\,|$, then $w$ and $u$ will have no common suffixes or prefixes, and we will
    // not have $w \rightarrow \mathcal{R}^{(N)}) v$ with $v \ge u$. We will then check to see whether $w$ can be written as $p\, s$
    // with $p > u\, s^{-1}$ (respectively $s \ge p^{-1} u$ ). If so, then we will replace $w$ with $p$ , and $u$ with $u\, s^{-1}$
    // (respectively $w$ with $s$ , and  $u$ with $p^{-1} u$ ). We will then be in the agreeable situation of being
    // absolutely sure that the reduction  $w \rightarrow_{\mathcal{R}^{(N)}}{}^{*} u$ could not be secured more cheaply than by
    // adjoining the rule ($w$ ,$u$) to $\mathcal{R}^{(N)}$.
    while ( $\kappa < |w|$ )
```

```
{
  $\kappa=\kappa+1$;
  // By definition 1.1, $\tau(w(1,\kappa))=\tau(w(1,\kappa-1)), w[\kappa])=\tau(s_{(\kappa-1)}, w[\kappa])$, so we put
  $s_{(\kappa)}=\tau(s_{(\kappa-1)}, w[\kappa])$;
  if ( $f(s_{(\kappa)})\neq\varnothing$ )
  {



    $p = w(1,\kappa)$;
    for ( all $r\in f(s_{(\kappa)})$ );
    {
      // we know that $p$ will have a suffix $s = \mathcal{B}(n)$ where:
      $(\mathcal{B},\mathcal{A})=r^{th}$ pair of $\mathcal{R}$;
      // and $n\in\mathbb{N}$.
      if ( $\mathcal{B}$ and $\mathcal{A}$ are words )
      // then, by convention:-
        $n=0$;
      else
        // $\mathcal{B}$ and $\mathcal{A}$ are triples and we calculate n by the method described in 4.2.1, i.e.,
        for ( $n=0$; $p[\kappa-|\mathcal{B}_3|-n|\mathcal{B}_2|]\neq\mathcal{B}_1[|\mathcal{B}_1|]$ ); $n=n+1$ );
      // We now know that $s = \mathcal{B}(n)$, and thus $w = p(1, \kappa-|\mathcal{B}(n)|)\,\mathcal{B}(n)\,w(\kappa+1, |w|)$. So,
      with:-
      $v = p(1, \kappa-|\mathcal{B}(n)|)\,\mathcal{A}(n)\,u(\kappa+1, |w|)$;
      // we will have $w \rightarrow_{\mathcal{R}}(\mathbb{N})\ v$.
      if ( $v \geq u$ )
      {
        // we put:-
        $w = v$;
        // noting that, for all $1<i\leq\kappa-|\mathcal{B}(n)|$, $\tau(u(1,i))=s_{(\kappa)}$. Whence, if we now:-
        strip $w$ and $u$ of their largest common prefix and largest common suffix;
        // then, with:-
        $\kappa=\kappa-|\mathcal{B}(n)|$;
        // are inductive hypothesis, trivially, still holds, and we break the for-loop.
        break;
```

```
                }
            }
        }
    }
    if ( $w \equiv p\, s$ with $p \geq u\, s^{-1}$ )
    {
        $w \equiv p$;
        $u \equiv u\, s^{-1}$;
    }
    if ( $w \equiv p\, s$ with $s \geq p^{-1} u$ )
    {
        $w \equiv s$;
        $u \equiv p^{-1} u$;
    }
    if ( $w > u$ )
        adjoin( (($w$), ($u$)) );
}
```

secure( word $w$, word $v$)

## 4.3

## *Resolving the Critical Pairs*

In this section we describe the procedures for resolving the critical pairs of $\mathcal{R}^{(\mathbb{N})}$. We note that there is no problem in deciding, for a single critical relation ( $w$ , $v$ ), whether or not the words $w$ and $v$ have a common $\mathcal{R}^{(\mathbb{N})}$-descendant; also there is a more or less standard way of resolving these single critical pairs. However, resolving a 1-parameterized set of critical pairs, $\{ ( \mathcal{W}(n) , \mathcal{V}(n) ) \mid n \in \mathbb{N} \}$, say, is not so straightforward because we do not have a method of deciding whether or not, for all $n \in \mathbb{N}$, the words $\mathcal{W}(n)$ and $\mathcal{V}(n)$ have common $\mathcal{R}^{(\mathbb{N})}$-descendants. Actually, we believe that this may be an undecidable problem in general (cf. theorem 3.2.2). Nevertheless, we can still describe a procedure, admittedly composed of basic techniques, which makes a sensible attempt at resolving these critical pairs, and which seems to work reasonably well in practice. Before this though, we will describe, so as to provide a complete pseudo program, the (probably) familiar code for resolving a single critical pair ( $w$ , $v$ ).

```
resolve( word $w$ , word $v$ )
{
    //   We resolve the critical pair ( $w$, $v$ ).
    reduce($w$);
    reduce($v$);
    if ( $v = w$ )
        return;
    if ( $v < w$ )
        swap $w$ and $v$ ;
    strip $w$ and $v$ of their largest common prefix and largest common suffix;
    let $p$= largest prefix of $w$ so that $w( |p| + 1 , |w| ) > p^{-1}v$ ;
    if ( $p \neq \varepsilon$ )
    {
        $w = w( |p| + 1 , |w| )$;
        $v = p^{-1}v$;
    }
    let $s$=largest suffix of $w$ so that $w(1, |w|-|s|) > v s^{-1}$;
    if ( $s \neq \varepsilon$ )
    {
        $w = w(1, |w|-|s|)$;
        $v = v s^{-1}$;
    }
    adjoin( (($w$),($v$)) );
    return;
}
```

```
resolve( word $w$, word $v$)
```

So, for the remainder of this section we are free to study the problem of resolving the 1-parameterized critical pairs

$$\{ ( \mathcal{W}(n) , \mathcal{V}(n) ) \mid n \in \mathbb{N} \}$$

for (variable) triples $\mathcal{W}, \mathcal{V} \in (C^*)^3$.

We will frequently refer to the terminology and results of the *preliminary* section(4.1), and to the procedure *secure( word $w$, word $v$)* (specified in section 4.2). Recall that the arguments, $w$ and $v$, must be such that $w \geq v$, and by calling secure ( $w$, $v$), we would secure the reduction $w \rightarrow^* v$ (by adjoining at most one new rewrite rule to $\mathcal{R}^{(\mathbb{N})}$).

Now, for many triples, $\mathcal{W}$ and $\mathcal{V}$, the critical pairs $\{ ( \mathcal{W}(n) , \mathcal{V}(n) ) \mid n \in \mathbb{N} \}$ can be resolved by adjoining just a finite number of new relations to $\mathcal{R}^{(\mathbb{N})}$. Such methods are the subjects of the next three (trivial) lemmas and their corollaries.

*4.3.1 Lemma.*

*(i)* Suppose there is a $m \in \mathbb{N}$ such that:

$$(1) \ \ \mathcal{W}_1(\mathcal{W}_2)^m \rightarrow^* \mathcal{V}_1(\mathcal{V}_2)^m \mathcal{V}_1^{-1}\mathcal{W}_1$$

and

$$(2) \ \ (\mathcal{V}_2)^n\mathcal{V}_3 \vee \mathcal{V}_1^{-1}\mathcal{W}_1(\mathcal{W}_2)^n\mathcal{W}_3 \text{, for all } n<m.$$

Then $\mathcal{W}(n) \vee \mathcal{V}(n)$ for all $n \in \mathbb{N}$.

*(ii)* Suppose there is a $m \in \mathbb{N}$ such that:

$$(\mathcal{W}_2)^m\mathcal{W}_3 \to^* \mathcal{W}_3\mathcal{V}_3^{-1}(\mathcal{V}_2)^m\mathcal{V}_3$$

and

$$\mathcal{W}_1(\mathcal{W}_2)^n\,\mathcal{W}_3\mathcal{V}_3^{-1} \vee \mathcal{V}_1(\mathcal{V}_2)^n \text{, for all } n<m.$$

Then $\mathcal{W}(n) \vee \mathcal{V}(n)$ for all $n\in\mathbb{N}$.

*Proof of (i):*

We should first note that the relations:

$$\mathcal{W}_1\mathcal{W}_3 =_G \mathcal{V}_1\mathcal{V}_3 \text{ and } \mathcal{W}_1(\mathcal{W}_2)^m\mathcal{W}_3 =_G \mathcal{V}_1(\mathcal{V}_2)^m\mathcal{V}_3$$

yield:-

$$\mathcal{W}_1(\mathcal{W}_2)^m =_G \mathcal{V}_1(\mathcal{V}_2)^m(\mathcal{V}_3\mathcal{W}_3^{-1}) =_G \mathcal{V}_1(\mathcal{V}_2)^m(\mathcal{V}_1^{-1}\mathcal{W}_1).$$

So, both relation:

$$\mathcal{W}_1(\mathcal{W}_2)^m =_G \mathcal{V}_1(\mathcal{V}_2)^m(\mathcal{V}_1^{-1}\mathcal{W}_1),$$

andrelation:

$$\mathcal{W}_1(\mathcal{W}_2)^n\,\mathcal{W}_3\mathcal{V}_3^{-1} =_G \mathcal{V}_1(\mathcal{V}_2)^n,\ n<m,$$

are consequences of the relations $\mathcal{W}(n) =_G \mathcal{V}(n)$, for all $n\in\mathbb{N}$.

We would have, for all $r\in\mathbb{N}$ and $n<m$,

$$\begin{aligned}
\mathcal{W}_1(\mathcal{W}_2)^{(rm+n)}\mathcal{W}_3 &\to^* (\mathcal{V}_1(\mathcal{V}_2)^m\mathcal{V}_1^{-1}\mathcal{W}_1)(\mathcal{W}_2)^{((r-1)m+n)}\mathcal{W}_3\\
&\to^* (\mathcal{V}_1(\mathcal{V}_2)^m\mathcal{V}_1^{-1})(\mathcal{V}_1(\mathcal{V}_2)^m\mathcal{V}_1^{-1}\mathcal{W}_1)(\mathcal{W}_2)^{((r-2)m+n)}\mathcal{W}_3\\
&\to^* \cdots\\
&\cdots \to^* (\mathcal{V}_1(\mathcal{V}_2)^m\mathcal{V}_1^{-1})^r\mathcal{W}_1(\mathcal{W}_2)^n\mathcal{W}_3 \text{, applying (1) r times,}\\
&\cdots \to^* \mathcal{V}_1(\mathcal{V}_2)^{mr}\mathcal{V}_1^{-1}\mathcal{W}_1(\mathcal{W}_2)^n\mathcal{W}_3,\\
&\vee\ \mathcal{V}_1(\mathcal{V}_2)^{mr}(\mathcal{V}_2)^n\mathcal{V}_3 \text{, by (2).}
\end{aligned}$$

$\boxed{4.3.1(i)}$

The proof of 4.3.1(ii) is similar.

$\boxed{4.3.1}$

*4.3.2 Corollary.*

If $|\mathcal{W}_2| > |\mathcal{V}_2|$, then the critical pairs $\{ ( \mathcal{W}(n) , \mathcal{V}(n) ) \mid n \in \mathbb{N} \}$ may be resolved by the following procedure.

```
{
    for( m=0; $\mathcal{W}_1(\mathcal{W}_2)^m < \mathcal{V}_1(\mathcal{V}_2)^m \mathcal{V}_1^{-1}\mathcal{W}_1$ and $(\mathcal{W}_2)^m \mathcal{W}_3 < \mathcal{W}_3\mathcal{V}_3^{-1}(\mathcal{V}_2)^m \mathcal{V}_3$ ; m=m+1 );
    if ( $\mathcal{W}_1(\mathcal{W}_2)^m \geq \mathcal{V}_1(\mathcal{V}_2)^m \mathcal{V}_1^{-1}\mathcal{W}_1$ )
    {
        secure( $\mathcal{W}_1(\mathcal{W}_2)^m , \mathcal{V}_1(\mathcal{V}_2)^m \mathcal{V}_1^{-1}\mathcal{W}_1$ );
        for ( n=1; n<m; n=n+1 )
        resolve( $\mathcal{V}_1^{-1}\mathcal{W}_1(\mathcal{W}_2)^n\mathcal{W}_3 , (\mathcal{V}_2)^n\mathcal{V}_3$ );
    }
    else
    {
        secure( $(\mathcal{W}_2)^m \mathcal{W}_3 , \mathcal{W}_3\mathcal{V}_3^{-1}(\mathcal{V}_2)^m \mathcal{V}_3$ );
        for ( n=1; n<m; n=n+1 )
        resolve( $\mathcal{W}_1(\mathcal{W}_2)^n\mathcal{W}_3\mathcal{V}_3^{-1} , \mathcal{V}_1(\mathcal{V}_2)^n$ );
    }
    return;
}
```

$\boxed{4.3.2}$

*4.3.3 Lemma.*

Suppose $\mathcal{W}$ is right sided so, cf. definition 4.1.2, we may define the word $p$ by:

$$(1)\ \mathcal{W}_1\mathcal{W}_2 \equiv p\mathcal{W}_1.$$

If:

$$(2)\ \mathcal{W}_1\mathcal{W}_3 \rightarrow^{*} \mathcal{V}_1\mathcal{V}_3,$$

as well as:

$$(3)\ p\mathcal{V}_1 \rightarrow^{*} \mathcal{V}_1\mathcal{V}_2$$

or

$$(4)\ p\mathcal{V}_1 \,{}^{*}\!\leftarrow \mathcal{V}_1\mathcal{V}_2,$$

then, for all $n \in \mathbb{N}$, $\mathcal{W}(n) \vee \mathcal{V}(n)$.

*Proof:*

We note that (1) yields, for all $n \in \mathbb{N}$,

$$(5)\ \ \mathcal{W}(n) \equiv (p)^{n} \mathcal{W}_1 \mathcal{W}_3 ,$$

and so:-

$$p\mathcal{V}_1\mathcal{V}_3 =_G p\mathcal{W}_1\mathcal{W}_3 \equiv \mathcal{W}(1) =_G \mathcal{V}(1) \equiv \mathcal{V}_1\mathcal{V}_2\mathcal{V}_3 .$$

Thus, the relation:

$$p\mathcal{V}_1 =_G \mathcal{V}_1\mathcal{V}_2$$

is a consequence of the identities (5) and the relation $\mathcal{W}(1) =_G \mathcal{V}(1)$.

So, supposing (2) and (3), we would have, for all $n \in \mathbb{N}$:-

$$\begin{aligned}
\mathcal{W}_1(\mathcal{W}_2)^{n}\mathcal{W}_3 &\equiv (p)^{n}\mathcal{W}_1\mathcal{W}_3 , \text{ by (5)},\\
&\rightarrow^{*} (p)^{n}\mathcal{V}_1\mathcal{V}_3 , \text{ by (2)},\\
&\rightarrow^{*} (p)^{(n-1)}\mathcal{V}_1\mathcal{V}_2\mathcal{V}_3\\
&\rightarrow^{*} \cdots\\
&\cdots \rightarrow^{*} p\,\mathcal{V}_1(\mathcal{V}_2)^{(n-1)}\mathcal{V}_3\\
&\rightarrow^{*} \mathcal{V}_1(\mathcal{V}_2)^{n}\mathcal{V}_3 , \text{ applying (3) n times.}
\end{aligned}$$

Supposing (2) and (4), we would have, for all $n \in \mathbb{N}$:-

$$\begin{aligned}
\mathcal{W}_1(\mathcal{W}_2)^{n}\mathcal{W}_3 &\equiv (p)^{n}\mathcal{W}_1\mathcal{W}_3 , \text{ by (5)},\\
&\rightarrow^{*} (p)^{n}\mathcal{V}_1\mathcal{V}_3 , \text{ by (2)},\\
&{}^{*}\!\leftarrow (p)^{(n-1)}\mathcal{V}_1\mathcal{V}_2\mathcal{V}_3\\
&\cdots {}^{*}\!\leftarrow\\
&\cdots {}^{*}\!\leftarrow p\mathcal{V}_1(\mathcal{V}_2)^{(n-1)}\mathcal{V}_3\\
&{}^{*}\!\leftarrow \mathcal{V}_1(\mathcal{V}_2)^{n}\mathcal{V}_3 , \text{ applying (4) n times.}
\end{aligned}$$

$\boxed{4.3.3}$

*4.3.4 Corollary.*

If $\mathcal{W}$ is right sided and $\mathcal{W}_1\mathcal{W}_3 \geq \mathcal{V}_1\mathcal{V}_3$, then the critical pairs $\{ ( \mathcal{W}(n) , \mathcal{V}(n) ) \mid n \in \mathbb{N} \}$ may be resolved by the following procedure.

```
{
    $p=(\mathcal{W}_1\mathcal{W}_2)(1, |\mathcal{W}_2|)$;
    secure( $\mathcal{W}_1\mathcal{W}_3$ , $\mathcal{V}_1\mathcal{V}_3$ );
    if ( $p\mathcal{V}_1 > \mathcal{V}_1\mathcal{V}_2$ )
        secure( $p\mathcal{V}_1$ , $\mathcal{V}_1\mathcal{V}_2$ );
    else
        secure( $\mathcal{V}_1\mathcal{V}_2$ , $p\mathcal{V}_1$ );
    return;
}
```

$\boxed{4.3.4}$

*4.3.5 Lemma.*

Suppose $\mathcal{W}$ is left sided so, cf. definition 4.1.4, we may define $s$ by $\mathcal{W}_2\mathcal{W}_3 = \mathcal{W}_3 s$. If:

$$\mathcal{W}_1\mathcal{W}_3 \rightarrow^* \mathcal{V}_1\mathcal{V}_3$$

as well as either:

$$\mathcal{V}_3 s \rightarrow^* \mathcal{V}_2\mathcal{V}_3 \text{ or } \mathcal{V}_3 s \ {}^*\!\leftarrow \mathcal{V}_2\mathcal{V}_3,$$

then, for all $n \in \mathbb{N}$, $\mathcal{W}(n) \vee \mathcal{V}(n)$.

$\boxed{4.3.5}$

*4.3.6 Corollary.*

If $\mathcal{W}$ is left sided and $\mathcal{W}_1\mathcal{W}_3 \geq \mathcal{V}_1\mathcal{V}_3$, then the critical pairs $\{ ( \mathcal{W}(n) , \mathcal{V}(n) ) \mid n \in \mathbb{N} \}$ may be resolved by the following procedure.

```
{
    $s=(\mathcal{W}_2\mathcal{W}_3)( |\mathcal{W}_3|+1, |\mathcal{W}_2\mathcal{W}_3| )$;
    secure( $\mathcal{W}_1\mathcal{W}_3$ , $\mathcal{V}_1\mathcal{V}_3$ );
    if ( $\mathcal{V}_3 s > \mathcal{V}_2\mathcal{V}_3$ )
        secure( $\mathcal{V}_3 s > \mathcal{V}_2\mathcal{V}_3$ );
    else
        secure( $\mathcal{V}_2\mathcal{V}_3$ , $\mathcal{V}_3 s$ );
    return;
}
```

$\boxed{4.3.6}$

*4.3.7 Lemma.*

Suppose $|\mathcal{W}_1\mathcal{W}_3| = |\mathcal{V}_1\mathcal{V}_3|$ , $|\mathcal{W}_2| = |\mathcal{V}_2| \neq 0$, and define the integer m as follows. If $|\mathcal{W}_1| \geq |\mathcal{V}_1|$, then let m be least so that $|\mathcal{W}_1| \leq |\mathcal{V}_1| + m|\mathcal{V}_2|$; if $|\mathcal{W}_1| \leq |\mathcal{V}_1|$, then let m be least so that $|\mathcal{V}_1| \leq |\mathcal{W}_1| + m|\mathcal{W}_2|$. It would follow that, if $\mathcal{V}(n) > \mathcal{W}(n)$ for some n, then $\mathcal{V}(n) > \mathcal{W}(n)$ for some $n \leq m+1$.

The proof is trivial string manipulation, and we omit it.

$\boxed{4.3.7}$

We can now describe the procedure for resolving the critical pairs:

$$\{ ( \mathcal{W}(n) , \mathcal{V}(n) ) \mid n \in \mathbb{N} \}$$

for (variable) triples $\mathcal{W}, \mathcal{V} \in (C^*)^3$.

```
resolve( triple $\mathcal{W}$ , triple $\mathcal{V}$ )
{
    // We shall resolve the critical pairs { ( $\mathcal{W}(n)$ , $\mathcal{V}(n)$ ) | $n \in \mathbb{N}$ }.
    if ( $\mathcal{W}(0)=\mathcal{V}(0)$ and $\mathcal{W}(1)=\mathcal{V}(1)$ )
        return;
    $\mathcal{W}$=reduce($\mathcal{W}$);
    $\mathcal{V}$=reduce($\mathcal{V}$);
    if ( $\mathcal{W}(0)=\mathcal{V}(0)$ and $\mathcal{W}(1)=\mathcal{V}(1)$ )
        return;
    if ( $\mathcal{W}_2=\mathcal{V}_2=\varepsilon$ )
    {
```

```
    resolve( $\mathcal{W}(0),\mathcal{V}(0)$ );
    return;
}
shiftleft($\mathcal{V}$);
shiftleft($\mathcal{W}$);
//    Recall that shiftleft( triple $\mathcal{T}$ ) redefines the components of $\mathcal{T}$, without altering { $\mathcal{T}(n) \mid n \in \mathbb{N}$ },
// so that $\mathcal{T}_3$ is maximal.
strip $\mathcal{W}_3$ and $\mathcal{V}_3$ of their largest common suffix;
shiftright($\mathcal{V}$);
shiftright($\mathcal{W}$);
//    Recall that shiftright( triple $\mathcal{T}$ ) redefines the components of $\mathcal{T}$, without altering
// { $\mathcal{T}(n) \mid n \in \mathbb{N}$ }, so that $\mathcal{T}_1$ is maximal.
strip $\mathcal{W}_1$ and $\mathcal{V}_1$ of their largest common prefix;
if ( $|\mathcal{W}_2| \neq |\mathcal{V}_2|$ )
{
    if ( $|\mathcal{W}_2| < |\mathcal{V}_2|$ )
    swap $\mathcal{W}$ and $\mathcal{V}$;
    // We now have $|\mathcal{W}_2| > |\mathcal{V}_2| > 0$, so we may apply the method of corollary 4 3 2 to
    // resolve all the critical pairs { ( $\mathcal{W}(n)$ , $\mathcal{V}(n)$ ) | $n \in \mathbb{N}$ }.
    for( m=0; $\mathcal{W}_1(\mathcal{W}_2)^m < \mathcal{V}_1(\mathcal{V}_2)^m\mathcal{V}_1^{-1}\mathcal{W}_1$ and $(\mathcal{W}_2)^m\mathcal{W}_3 < \mathcal{W}_3\mathcal{V}_3^{-1}(\mathcal{V}_2)^m\mathcal{V}_3$; m=m+1 );
    if ( $\mathcal{W}_1(\mathcal{W}_2)^m \geq \mathcal{V}_1(\mathcal{V}_2)^m\mathcal{V}_1^{-1}\mathcal{W}_1$ )
    {
        secure( $\mathcal{W}_1(\mathcal{W}_2)^m$ , $\mathcal{V}_1(\mathcal{V}_2)^m\mathcal{V}_1^{-1}\mathcal{W}_1$ );
        for ( n=1; n<m; n=n+1 )
            resolve( $\mathcal{V}_1^{-1}\mathcal{W}_1(\mathcal{W}_2)^n\mathcal{W}_3$ , $(\mathcal{V}_2)^n\mathcal{V}_3$ );
    }
    else
    {
        for ( n=1; n<m; n=n+1 )
            resolve( $\mathcal{W}_1(\mathcal{W}_2)^n\mathcal{W}_3\mathcal{V}_3^{-1}$ , $\mathcal{V}_1(\mathcal{V}_2)^n$ );
    }
    return;
}
```

```
//    We now have $|\mathcal{W}_2| = |\mathcal{V}_2| > 0$.
shiftleft($\mathcal{V}$);
shiftleft($\mathcal{W}$);
let m be maximal so that $\mathcal{V}_3$ has $(\mathcal{V}_2)^m$ as prefix, and $\mathcal{W}_3$ has $(\mathcal{W}_2)^m$ as prefix.
if m>0
{
    $S_{\mathcal{V}_3} = \mathcal{V}_3(m|\mathcal{V}_2|+1, |\mathcal{V}_3|)$;
    $S_{\mathcal{W}_3} = \mathcal{W}_3(m|\mathcal{W}_2|+1, |\mathcal{W}_3|)$;
    if ( reduce($\mathcal{V}_1 S_{\mathcal{V}_3}$)=reduce($\mathcal{W}_1 S_{\mathcal{W}_3}$) )
    {
        // Then a simple (inductive) argument yields $\mathcal{V}_1(\mathcal{V}_2)^n S_{\mathcal{V}_3} \langle \mathbb{R}^{(\mathbb{N})} \rangle \mathcal{W}_1(\mathcal{W}_2)^n S_{\mathcal{V}_3}$ for all n.
        // Also, as $\mathcal{V}_3 = (\mathcal{V}_2)^m S_{\mathcal{V}_3}$, so { $\mathcal{V}(n)$ | $n \in \mathbb{N}$ }$\subsetneq${ $\mathcal{V}_1(\mathcal{V}_2)^n S_{\mathcal{V}_3}$ | $n \in \mathbb{N}$ } and as $\mathcal{W}_3 = (\mathcal{W}_2)^m S_{\mathcal{W}_3}$
        // so { $\mathcal{W}(n)$ | $n \in \mathbb{N}$ }$\subsetneq${ $\mathcal{W}_1(\mathcal{W}_2)^n S_{\mathcal{W}_3}$ | $n \in \mathbb{N}$ }. Thus, we may as well put:
        $\mathcal{V}_3 = S_{\mathcal{V}_3}$;
        $\mathcal{W}_3 = S_{\mathcal{W}_3}$;
    }
}
shiftright($\mathcal{V}$);
shiftright($\mathcal{W}$);
let m be maximal so that $\mathcal{V}_1$ has $(\mathcal{V}_2)^m$ as suffix, and $\mathcal{W}_1$ has $(\mathcal{W}_2)^m$ as suffix.
if m>0
{
    $\mathcal{P}_{\mathcal{V}_1} = \mathcal{V}_3(1, |\mathcal{V}_3| - m|\mathcal{V}_2|)$;
    $\mathcal{P}_{\mathcal{W}_1} = \mathcal{W}_3(1, |\mathcal{W}_3| - m|\mathcal{W}_2|)$;
    if ( reduce($\mathcal{P}_{\mathcal{V}_1} \mathcal{V}_3$)=reduce($\mathcal{P}_{\mathcal{W}_1} \mathcal{W}_3$) )
    {
        // A simple (inductive) argument yields $\mathcal{P}_{\mathcal{V}_1}(\mathcal{V}_2)^n \mathcal{V}_3 \langle \mathbb{R}^{(\mathbb{N})} \rangle \mathcal{P}_{\mathcal{W}_1}(\mathcal{W}_2)^n \mathcal{W}_3$ for all n. Also,
        // as $\mathcal{V}_1 = \mathcal{P}_{\mathcal{V}_1}(\mathcal{V}_2)^m$, so { $\mathcal{V}(n)$ | $n \in \mathbb{N}$ }$\subsetneq${ $\mathcal{P}_{\mathcal{V}_1}(\mathcal{V}_2)^n \mathcal{V}_3$ | $n \in \mathbb{N}$ } and as $\mathcal{W}_1 = \mathcal{P}_{\mathcal{W}_1}(\mathcal{W}_2)^n$, so
        // { $\mathcal{W}(n)$ | $n \in \mathbb{N}$ }$\subsetneq${ $\mathcal{P}_{\mathcal{W}_1}(\mathcal{W}_2)^n \mathcal{W}_3$ | $n \in \mathbb{N}$ }. Thus, we may as well put:
        $\mathcal{V}_1 = S_{\mathcal{V}_1}$;
```

```
            // and
            $\mathcal{W}_1 = s_{\mathcal{W}_1}$;
        }
}
if $\mathcal{W}(0) < \mathcal{V}(0)$
        swap $\mathcal{W}$ and $\mathcal{V}$;
//    We now have $|\mathcal{W}_2| = |\mathcal{V}_2| > 0$, $\mathcal{W}(0) > \mathcal{V}(0)$ and $\mathcal{W}(0)$ and $\mathcal{V}(0)$ have no non-empty common
// prefixes and no non-empty common suffixes.
shiftright($\mathcal{W}$);
if ( $\mathcal{W}_1$ can be written as $ps$ so that the triple ( $s$ , $\mathcal{W}_2$ , $\mathcal{W}_3$ ) is left or right sided and $s\mathcal{W}_3 \geq p^{-1}\mathcal{V}_1\mathcal{V}_3$)
{
        $\mathcal{W}_1 = s$;
        $\mathcal{V}_1 = p^{-1}\mathcal{V}_1$;
}
else
{
        shiftleft($\mathcal{W}$);
        if ( $\mathcal{W}_3$ can be written as $ps$ so that ( $\mathcal{W}_1$ , $\mathcal{W}_2$ , $p$ ) is left or right sided and $\mathcal{W}_1 p \geq \mathcal{V}_1\mathcal{V}_3 s^{-1}$)
        {
            $\mathcal{W}_3 = p$;
            $\mathcal{V}_3 = \mathcal{V}_3 s^{-1}$;
        }
}
if ( $\mathcal{W}$ is not already left or right sided )
{
        shiftright($\mathcal{W}$);
        let $s$ be of minimal length so that $\mathcal{W}_1$ can be written as $ps$ with $s\mathcal{W}_3 \geq p^{-1}\mathcal{V}_1\mathcal{V}_3$;
        {
            $\mathcal{W}_1 = s$;
            $\mathcal{V}_1 = p^{-1}\mathcal{V}_1$;
        }
```

```
    shiftleft($\mathcal{W}$);
    let $p$ be of minimal length so that $\mathcal{W}'_3$ can be written as $ps$ with $\mathcal{W}'_1 p > \mathcal{V}'_1 \mathcal{V}'_3 s^{-1}$)
    {
        $\mathcal{W}'_3 = p$;
        $\mathcal{V}_3 = \mathcal{V}_3 s^{-1}$;
    }
    //    Note that we still have $\mathcal{W}(0) > \mathcal{V}(0)$.
    $p$=reduce($\mathcal{W}'_1 \mathcal{W}'_2 \mathcal{W}'^{-1}_1$);
    if ( $\mathcal{W}'_1 \mathcal{W}'_2 \geq p\mathcal{W}'_1$ )
    {
        secure( $\mathcal{W}'_1 \mathcal{W}'_2$ , $p\mathcal{W}'_1$ );
        // We now have $\mathcal{W}'_1 \mathcal{W}'_2 \to_{\mathcal{R}(\mathbb{N})}^{*} p\mathcal{W}'_1$, whence $\mathcal{W}'_1 (\mathcal{W}'_2)^n \mathcal{W}'_3 \to_{\mathcal{R}(\mathbb{N})}^{*} (p)^n \mathcal{W}'_1 \mathcal{W}'_3$ , for
        // all $n \in \mathbb{N}$. Whence $(p)^n \mathcal{W}'_1 \mathcal{W}'_3 \vee \mathcal{V}(n)$ (for all $n \in \mathbb{N}$) $\Rightarrow \mathcal{W}(n) \vee \mathcal{V}(n)$ (for all $n \in \mathbb{N}$). Thus,
        // we may put:-
        $\mathcal{W}$=( $\varepsilon$ , $p$ , $\mathcal{W}'_1 \mathcal{W}'_3$ );
        // so that $\mathcal{W}$ is now right sided and $\mathcal{W}'_1 \mathcal{W}'_3 \geq \mathcal{V}'_1 \mathcal{V}_3$ (cf corollary 4.3 4).
    }
    else
    {
        s=reduce($\mathcal{W}'^{-1}_3 \mathcal{W}'_2 \mathcal{W}'_3$);
        if ( $\mathcal{W}'_2 \mathcal{W}'_3 \geq \mathcal{W}'_3 s$ )
        {
            secure( $\mathcal{W}'_2 \mathcal{W}'_3$ , $\mathcal{W}'_3 s$ );
            // We now have $\mathcal{W}'_2 \mathcal{W}'_3 \to_{\mathcal{R}(\mathbb{N})}^{*} \mathcal{W}'_3 s$, whence $\mathcal{W}'_1 (\mathcal{W}'_2)^n \mathcal{W}'_3 \to_{\mathcal{R}(\mathbb{N})}^{*} \mathcal{W}'_1 \mathcal{W}'_3 (s)^n$ , for
            // all $n \in \mathbb{N}$. Whence $\mathcal{W}'_1 \mathcal{W}'_3 (s)^n \vee \mathcal{V}(n)$ (for all $n \in \mathbb{N}$) $\Rightarrow \mathcal{W}(n) \vee \mathcal{V}(n)$ (for all $n \in \mathbb{N}$).
            //Thus, we may put:-
            $\mathcal{W}$=( $\mathcal{W}'_1 \mathcal{W}'_3$ , $s$ , $\varepsilon$ );
            // so that $\mathcal{W}$ is now left sided and $\mathcal{W}'_1 \mathcal{W}'_3 \geq \mathcal{V}'_1 \mathcal{V}'_3$ (cf corollary 4.3.6).
        }
    }
}
```

```
//    Note that we still have $\mathcal{W}(0)>\mathcal{V}(0)$ and that $\mathcal{W}(0)$ and $\mathcal{V}(0)$ have no non-empty common
// prefixes and no non-empty common suffixes.
if ( $\mathcal{W}'$ is left sided )
{
    // We may apply the method of corollary 4.3.6 to resolve all the critical pairs
    // { ( $\mathcal{W}(n)$ , $\mathcal{V}(n)$ ) | $n\in\mathbb{N}$ }.
    $s=(\mathcal{W}'_2\mathcal{W}'_3)(|\mathcal{W}'_3|+1, |\mathcal{W}'_2\mathcal{W}'_3|)$;
    secure( $\mathcal{W}'_1\mathcal{W}'_3$ , $\mathcal{V}'_1\mathcal{V}'_3$ );
    if ( $\mathcal{V}'_3 s > \mathcal{V}'_2\mathcal{V}'_3$ )
        secure( $\mathcal{V}'_3 s > \mathcal{V}'_2\mathcal{V}'_3$ );
    else
        secure( $\mathcal{V}'_2\mathcal{V}'_3$ , $\mathcal{V}'_3 s$ );
    return;
}
if ( $\mathcal{W}$ is right sided )
{
    // We may apply the method of corollary 4.3.4 to resolve all the critical pairs
    // { ( $\mathcal{W}(n)$ , $\mathcal{V}(n)$ ) | $n\in\mathbb{N}$ }.
    $p=(\mathcal{W}'_1\mathcal{W}'_2)(1, |\mathcal{W}'_2|)$;
    secure( $\mathcal{W}'_1\mathcal{W}'_3$ , $\mathcal{V}'_1\mathcal{V}'_3$ );
    if ( $p\mathcal{V}'_1 > \mathcal{V}'_1\mathcal{V}'_2$ )
        secure( $p\mathcal{V}'_1$ , $\mathcal{V}'_1\mathcal{V}'_2$ );
    else
        secure( $\mathcal{V}'_1\mathcal{V}'_2$ , $p\mathcal{V}'_1$ );
    return;
}
//    We know that $\mathcal{W}(0)>\mathcal{V}(0)$, $\mathcal{W}(0)$ and $\mathcal{V}(0)$ have no non-empty common prefixes and no
// non-empty common suffixes, and that $\mathcal{W}$ is not left or right sided. We now need to know
// whether $\mathcal{W}(n)<\mathcal{V}(n)$ for some $n\in\mathbb{N}$. By lemma 4.3.7, we know that $\mathcal{V}(n)>\mathcal{W}(n)$ for some n if and
// only if $\mathcal{V}(n)>\mathcal{W}(n)$ for some $n\leq m+1$ where m is defined as follows. If $|\mathcal{W}_1|\geq|\mathcal{V}_1|$, then m is least
// so that $|\mathcal{W}_1|\leq|\mathcal{V}_1|+m|\mathcal{V}'_2|$; if $|\mathcal{V}_1|\geq|\mathcal{W}_1|$, then m is least so that $|\mathcal{V}_1|\leq|\mathcal{W}'_1|+m|\mathcal{W}'_2|$.
if ( $|\mathcal{W}_1|\geq|\mathcal{V}_1|$ )
```

```
        for(m=0; $|\mathcal{W}'_1| > |\mathcal{V}'_1| + m|\mathcal{V}'_2|$; m=m+1);
    else
        for(m=0; $|\mathcal{V}'_1| > |\mathcal{W}'_1| + m|\mathcal{W}'_2|$; m=m+1);
    m=m+1;
    for(n=1; n≤m; n=n+1)
    if ( $\mathcal{V}(n) > \mathcal{W}(n)$ )
    {
        //As $\mathcal{V}(0) \le \mathcal{W}(0)$ and $\mathcal{W}'_1$ is not empty (because $\mathcal{W}'$is not right sided), so it has to be that $\mathcal{V}'_1 = \varepsilon$.
         for(i=0; i<n; i=i+1)
            resolve( $\mathcal{W}(i)$ , $\mathcal{V}(i)$ );
        // We are now left with the problem of resolving all { ( $\mathcal{W}(i)$ , $\mathcal{V}(i)$ ) | i>n},i.e, with:
        $\mathcal{V}=(\varepsilon, \mathcal{V}_2, (\mathcal{V}_2)^n\mathcal{V}_3)$;
        $\mathcal{W}'=(\mathcal{W}'_1, \mathcal{W}_2, (\mathcal{W}_2)^n\mathcal{W}_3)$;
        // we want to resolve all { ( $\mathcal{W}(i)$ , $\mathcal{V}(i)$ ) | n∈ℕ }  However, we now have $\mathcal{V}$ right sided and
        // with $\mathcal{V}(0)=(\mathcal{V}_2)^n\mathcal{V}_3 > \mathcal{W}_1(\mathcal{W}_2)^n\mathcal{W}_3=\mathcal{W}(0)$. So, with reference to corollary 4 3 4, we may
        // resolve the remaining critical pairs, { ( $\mathcal{W}(i)$ , $\mathcal{V}(i)$ ) | i∈ℕ }, as follows.
        secure( $(\mathcal{V}_2)^n\mathcal{V}_3$ , $\mathcal{W}_1(\mathcal{W}_2)^n\mathcal{W}_3$ );
        if ( $\mathcal{V}_2\mathcal{W}_1 > \mathcal{W}_1\mathcal{W}_2$ )
            secure( $\mathcal{V}_2\mathcal{W}_1$ , $\mathcal{W}_1\mathcal{W}_2$ );
        else
            secure( $\mathcal{W}_1\mathcal{W}_2$ , $\mathcal{V}_2\mathcal{W}_1$ );
        return;
    }
    //    We now know that $\mathcal{W}(n) > \mathcal{V}(n)$ and (amongst other things) that $\mathcal{W}$is neither left or right sided,
    // so we resolve the critical pairs { ( $\mathcal{W}(i)$ , $\mathcal{V}(i)$ ) | i∈ℕ } simply by putting:-
    adjoin(($\mathcal{W}$),($\mathcal{V}$);
}
```

resolve( triple $\mathcal{W}$, triple $\mathcal{V}$)

[1]We note that, a pair ( $\mathcal{W}$, $\mathcal{V}$) ($\mathcal{W}$ and $\mathcal{V}$ being triples) is adjoined to $\mathcal{R}$ only if $\mathcal{W}$ is not left or right sided. Also, suppose we wish to input a set rules { ( $\mathcal{W}(n)$ , $\mathcal{V}(n$ ) ) | n∈ℕ } as data to the program. Let us suppose $\mathcal{W}$ is right sided, then, by corollary 4.3.4, we should

do as follows. Define $p$ by $\mathcal{U}'_1\mathcal{U}'_2 = p\mathcal{U}'_1$, then input $(\mathcal{U}'_1\mathcal{U}'_3 , \mathcal{V}'_1\mathcal{V}'_3)$ together with $(p\mathcal{V}'_1 , \mathcal{V}'_1\mathcal{V}'_2)$, if $p\mathcal{V}'_1 \geq \mathcal{V}'_1\mathcal{V}'_2$, or with $(\mathcal{V}'_1\mathcal{V}'_2 , p\mathcal{V}'_1)$, if $\mathcal{V}'_1\mathcal{V}'_2 \geq p\mathcal{V}'_1$. Also, by corollary 4.3.6, if $\mathcal{U}'$ were left sided, then we should define $s$ by $\mathcal{U}'_2\mathcal{U}'_3 = \mathcal{U}'_3 s$, and input $(\mathcal{U}'_1\mathcal{U}'_3 , \mathcal{V}'_1\mathcal{V}'_3)$ together with $(\mathcal{V}'_3 s , \mathcal{V}'_2\mathcal{V}'_3)$, if $\mathcal{V}'_3 s \geq \mathcal{V}'_2\mathcal{V}'_3$, or with $(\mathcal{V}'_2\mathcal{V}'_3 , \mathcal{V}'_3 s)$, if $\mathcal{V}'_2\mathcal{V}'_3 \geq \mathcal{V}'_3 s$. In short, we have justified proposition 4.1.6, i.e., that left( $\mathcal{R}$) should never contain left or right sided triples.

*Comment.* Whenever the procedure *secure( word , word )* is called it might adjoin to $\mathcal{R}^{(\mathbb{N})}$ a rewrite rule, $( b , a )$, say, with $b$ being $\mathcal{R}^{(\mathbb{N})}$-reducible. Although such rules do not belong to the minimal $\mathcal{R}^{(\mathbb{N})}$-complete presentation (because, cf. theorem 3.1.1, $b \notin J(\mathcal{R}^{(\mathbb{N})})$ it must still be checked against other rules for possible critical superpositions. This is certainly a hindrance, but preferable to adjoining an infinite 1-parameterized set of rules to $\mathcal{R}^{(\mathbb{N})}$ when a finite number of rules would suffice.

We hoped that this setback would be somewhat offset by testing for prime critical pairs before resolution (we refer the reader to the definition of a prime critical pair on page 46). The fact that non-prime critical pairs may be ignored is a commonplace labour saving test in *finite* completion programs, and the fact that it works for such programs is easy to prove. We point out that the test works because, when attempting finite completions, whenever a rule, $( b , a )$, say, is adjoined $\mathcal{R}^{(\mathbb{N})}$, it is adjoined with the *aim* of guaranteeing that $b$ and $a$ have a common $\mathcal{R}^{(\mathbb{N})}$-descendant, namely $a$. In our program, however, the procedure *secure* might adjoin a rule, $( b , a )$, to $\mathcal{R}^{(\mathbb{N})}$ with the aim of guaranteeing that $b \rightarrow^*_{\mathcal{R}^{(\mathbb{N})}} a$ (which is a stronger requirement than that $a$ and $b$ simply have a common descendant). It is because of this phenomena that we cannot, so freely, rely on the prime critical pair test. Nevertheless, it is not difficult to prove that, by marking the rules which are adjoined by secure, we may ignore the non-prime critical pairs of two *unmarked* sets of rules (cf. (Kapur,Musser,Narendon)).

Recall that the program makes no attempt to resolve the 2-parameterized critical pairs. The (unavoidable) 2-parameterized pairs computed during the completion process are supposed disjoint from $\mathcal{R}^{(\mathbb{N})}$, stored appropriately, and thereafter ignored until, if ever, the (probable) completion of the (1-parameterized) set $\mathcal{R}^{(\mathbb{N})}$ stops (cf pages 44 and 92). When this happens we must, to prove that $\mathcal{R}^{(\mathbb{N})}$ is complete, confirm that all the 2-parameterized critical pairs are resolved, i.e., have common $\mathcal{R}^{(\mathbb{N})}$-descendants. This will most probably be true, indeed for most of the examples cited in section 4.5, it was easily confirmed by inspection. We infer from this that attempts to prove that a 2-parameterized set of critical pairs, $\{ ( \mathcal{W}(n) , \mathcal{V}(n) ) \mid n \in \mathbb{N}^2 \}$, say, are resolved need not be very sophisticated in order to be reasonably successful. Basically, by calling reduce( $\mathcal{W}'_1 , \mathcal{W}'_2 , \mathcal{W}'_3$ ) and reduce( $\mathcal{W}'_3 , \mathcal{W}'_4 , \mathcal{W}'_5$ ), respectively, we reduce the words of $\{ \mathcal{W}'_1(\mathcal{W}'_2)^n\mathcal{W}'_3 \mid n \in \mathbb{N} \}$ and of $\{ \mathcal{W}'_3(\mathcal{W}'_4)^n\mathcal{W}'_5 \mid n \in \mathbb{N} \}$ as far as possible. We repeat the process with the quintuple $\mathcal{V}$, and then compare the resulting sets of strings.

```
*reduce( quintuple Q )*
{
    do
    {
        $Temp$=( $Q_1 , Q_2 , Q_3 , Q_4 , Q_5$ )
        shiftleft( ( $Q_1 , Q_2 , Q_3$ ) );
        ( $Q_3 , Q_4 , Q_5$ )=reduce(( $Q_3 , Q_4 , Q_5$ )) );
        shiftright( ( $Q_3 , Q_4 , Q_5$ ) );
        ( $Q_1 , Q_2 , Q_3$ )=reduce(( $Q_1 , Q_2 , Q_3$ ) );
    }
    while ( $Q(0,0) \neq Temp(0,0)$ or $Q(0,1) \neq Temp(0,1)$ or $Q(1,0) \neq Temp(0,1)$ or $Q(1,1) \neq Temp(1,1)$ )
    // i.e., while $\{ Q(n) \mid n \in \mathbb{N}^2 \} \neq \{ Temp(n) \mid n \in \mathbb{N}^2 \}$.
    return $Q$;
}
```

reduce( quintuple )

```
*resolved?( quintuple $\mathcal{W}$ , quintuples $\mathcal{V}$ )*
{
    // We make an attempt to prove that the the critical pairs $\{ ( \mathcal{W}(n) , \mathcal{V}(n) ) \mid n \in \mathbb{N}^2 \}$ are resolved.
    $\mathcal{V}$=reduce( $\mathcal{V}$);
    $\mathcal{W}$=reduce( $\mathcal{W}$);
    if ( $\mathcal{M}(0,0) \neq \mathcal{N}(0,0)$ or $\mathcal{M}(0,1) \neq \mathcal{N}(0,1)$ or $\mathcal{M}(1,0) \neq \mathcal{N}(0,1)$ or $\mathcal{M}(1,1) \neq \mathcal{N}(1,1)$ )
    // i.e., while { $\mathcal{N}(n) \mid n \in \mathbb{N}^2$ } $\neq$ { $\mathcal{M}(n) \mid n \in \mathbb{N}^2$ }.
    // we have not confirmed that the critical pairs are resolved, so:
    return 0;
    else
        // the critical pairs are resolved, and we:
    return 1;
}
```

resolved?( quintuple,quintuple )

## *4.4*

## *Computing the Critical Pairs*

In this section we will describe the highest level of the completion program, i.e., the procedure which computes and then, by calling the predefined procedures of sections 4.2 and 4.3, resolves all the critical pairs of $\mathcal{R}^{(\mathbb{N})}$.

Recall that the critical pairs of ( $b_1$ , $a_1$ ) and ( $b_2$ , $a_2$ ), in $\mathcal{R}^{(\mathbb{N})}$, are all those pairs of words:

*(1)* ( $a_1$ , $pa_2s$ ) with $b_1 \equiv pb_2s$, for some words $p$ and $s$.

*(2)* ( $a_1s$ , $pa_2$ ) with $b_1s \equiv pb_2$ , for some words $\varepsilon \not\equiv p \not\equiv b_1$ and $\varepsilon \not\equiv s \not\equiv b_2$.

So the program has to compute all the pairs of the form (1) and (2) for all ( $b_1$ , $a_1$ ) and ( $b_2$ , $a_2$ ) in $\mathcal{R}^{(\mathbb{N})}$. When $\mathcal{R}^{(\mathbb{N})}$ is finite this poses no problems, but with $\mathcal{R}^{(\mathbb{N})}$ possibly containing (infinite) 1-parameterized sets, it is not a completely trivial matter to prove that all the pairs (1) and (2) are resolved. Anyway, it is natural to split this procedure into four parts as follows.

(i) We resolve all the pairs (1) and (2) for a (fixed) ordered pair, ( $b_1$ , $a_1$ ) and ( $b_2$ , $a_2$ ) in $\mathcal{R}^{(\mathbb{N})}$. In so doing we might adjoin 1-parameterized rules to $\mathcal{R}^{(\mathbb{N})}$, not that this is necessary (to resolve a finite number of critical pairs), but because we are attempting to predict an infinite sequence of rules which would normally be computed and which could be 1-parameterized. The method was suggested by practising on some of the examples of section 4.5 (the Dyck and surface groups) and is really quite a natural method of predicting the necessary 1-parameterized rules. We will not comment any further on the method other than to say that, although it may not be foolproof, it appears to work just as well as the method suggested by Gilman in (Gilman84), but without wasting much time in testing for reasonable candidates for the 1-parameterized rules.

(ii) We resolve all the pairs (1) and (2) for (fixed) ( $b_1$ , $a_1$ ) in $\mathcal{R}^{(\mathbb{N})}$ and all ( $b_2$ , $a_2$ )$\in$ { ( $\mathcal{B}$(n) , $\mathcal{A}$(n) ) | n$\in \mathbb{N}$ } where ( $\mathcal{B}$ , $\mathcal{A}$ ) are (fixed) triples of $\mathcal{R}$.

(iii) We resolve all the pairs (1) and (2) for all ( $b_1$ , $a_1$ )$\in$ { ( $\mathcal{B}$(n) , $\mathcal{A}$(n) ) | n$\in \mathbb{N}$ }, where ( $\mathcal{B}$ , $\mathcal{A}$ ) are (fixed) triples of $\mathcal{R}$, and (fixed) ( $b_2$ , $a_2$ ) in $\mathcal{R}^{(\mathbb{N})}$.

(iv) We resolve all the pairs (1) and (2) for all the ( $b_1$ , $a_1$ ) and ( $b_2$ , $a_2$ ) belonging to (fixed) 1-parameterized sets. This is the only procedure whose correctness we shall bother to prove (i.e. corollary 4.4.3).

*Comment*. We do not test, in any of the pseudo code of this section, for prime critical pairs (cf pages 46 and 76), although it is advisable that some such test be made.

```
CritPair( words ( $b_1$ , $a_1$ ) , words ( $b_2$ , $a_2$ ) )
{
    //    Computes and resolves the critical pairs of ( ( $b_1$ , $a_1$ ) , ( $b_2$ , $a_2$ ) ), where $b_1$,$a_1$,$b_2$ and $a_2$ are
    // words.
    for ( all words $p$ , $s$ such that $\varepsilon < p < b_1$, $\varepsilon < s < b_2$ and $pb_2 = b_1 s$ )
    {
        $\kappa = |b_1| - |p|$;
        if (
                                        $|b_2| - |a_2|$,
                                 $a_2(|s|+1, |a_2|) = b_2(1,\kappa)$,
                                  $p$ is not a suffix of $p\, a_2(1,|s|)$,
                  $p(a_2(1,|s|))^n b_2(1,\kappa)$ and $a_1(s)^n$ are $\mathcal{R}^{(\mathbb{N})}$-reduced for all $n \in \mathbb{N}$.
            )
            // Then we note that, with n=1, $p(a_2(1,|s|))^n b_2(1,\kappa) = pa_2(1,|s|) b_2(1,\kappa)$
            //                                                    $= pa_2(1,|s|) a_2(|s|+1, |a_2|) = pa_2$.
            // Also, it is not difficult to prove, by induction on n, that, if $p(a_2(1,|s|))^n b_2(1,\kappa)$ and $b_2$
            // belong to $I(\mathcal{R}^{(\mathbb{N})})$, and $a_1(s)^n$ and $a_2$ are least words in their $\langle\mathcal{R}^{(\mathbb{N})}\rangle$-congruence class, then
            // ( $p(a_2(1,|s|))^{(n+1)} b_2(1,\kappa)$ , $a_1(s)^{(n+1)}$ ) will be a critical pair of an $\mathcal{R}^{(\mathbb{N})}$-complete
            // presentation (cf section 3.1, page 19). We also know that the triple ( $p$ , $a_2(1,|s|)$ , $b_2(1,\kappa)$ )
            // will not be left or right sided, that the words $p$ and $a_1 s$ have no common prefixes, that the
            // words $b_2(1,\kappa)$ and $a_1 s$ have no common suffixes, etc. In short, we prejudge the natural
            // completion process and resolve the (probable) critical pairs ( $p(a_2(1,|s|))^n b_2(1,\kappa)$ , $a_1(s)^n$ )
            // by putting:
            adjoin( ( ( p , $a_2(1,|s|)$ , $b_2(1,\kappa)$ ) , ( $a_1$ , $s$ , $\varepsilon$ ) ) );
        else if (
                                        $|b_1| - |a_1|$,
                                     $a_1(1,\kappa) = b_2(1,\kappa)$,
                               $s$ is not a prefix of $s\, a_1(\kappa+1,|a_1|)$,
                    $|b_2| > |a_2|$ and, if $|b_2| = |a_2|+2$, then $a_1[1]^{-1} > a_1[2]$
                    $a_1(1,\kappa)(a_1(\kappa+1,|a_1|))^n s$ and $(p)^n a_2$ are $\mathcal{R}^{(\mathbb{N})}$-reduced for all $n \in \mathbb{N}$.
            )
                // Then we note that, with n=1, $a_1(1,\kappa)(a_1(\kappa+1,|a_1|))^n s = a_1 s$. Also, it is not difficult to
                // prove, by induction on n, that, if $a_1(1,\kappa)(a_1(\kappa+1,|a_1|))^n s$ and $b_1$ belong to $I(\mathcal{R}^{(\mathbb{N})})$,
                // and $(p)^n a_2$ and $a_1$ are least words in their $\langle\mathcal{R}^{(\mathbb{N})}\rangle$-congruence class, then
                // $( a_1(1,\kappa)(a_1(\kappa+1,|a_1|))^{(n+1)} s , (p)^{(n+1)} a_2 )$ will be a critical pair of an $\mathcal{R}^{(\mathbb{N})}$-complete
                // presentation. We will also know that the triple $( a_1(1,\kappa) , a_1(\kappa+1,|a_1|) , s )$ is not left
                // or right sided, that the words $a_1(1,\kappa)$ and $pa_2$ have no common prefixes, that the
                // words $s$ and $pa_2$ have no common suffixes, etc. So, we prejudge the natural
                // completion process and resolve the (probable) critical pairs
                // $( a_1(1,\kappa)(a_1(\kappa+1,|a_1|))^n s , (p)^n a_2 )$ by putting:
                adjoin( ( ( $a_1(1,\kappa)$ , $a_1(\kappa+1,|a_1|)$ , $s$ ) , ( $\varepsilon$ , $p$ , $a_2$ ) ) );
        else
            // we simply:
            resolve( $a_1 s$ , $pa_2$ );
    for ( all words $p$ , $s$ such that $b_1 = pb_2$ )
        resolve( $a_1$ , $pa_2 s$ );
}
```

CritPair( words , words )

```
CritPair( words ( $b$ , $a$ ) , triples ( $\mathcal{B}$ , $\mathcal{A}$ ) )
{
    //    Computes and resolves the critical pairs of ( ( $b$ , $a$ ) , ( $\mathcal{B}$(n), $\mathcal{A}$(n) ) ) for all $n \in \mathbb{N}$, where $b$ and $a$
    // are words and $\mathcal{B}$ and $\mathcal{A}$ are triples.
    for( all words $p$ , $s$ and all $r \in \mathbb{N}$ such that $p\mathcal{B}(r)s = b$ )
        resolve( $p\mathcal{A}(r)s$ , $a$ );
    for ( all words $p$ , $s$ and all $r \in \mathbb{N}$ such that $\varepsilon < p < b$ , $\varepsilon < s < \mathcal{B}_3$ and $p\mathcal{B}(r) = bs$ )
        resolve( $p\mathcal{A}(r)$ , $as$ );
    for ( all words $p$ , $s$ and all $r \in \mathbb{N}$ such that $\varepsilon < p < b$ , $\varepsilon < s < \mathcal{B}_2$ and $p\mathcal{B}(r) = bs\mathcal{B}_3$ )
        resolve( ( $p\mathcal{A}(r+n)$ , $as(\mathcal{B}_2)^n\mathcal{B}_3$ ) for all $n \in \mathbb{N}$ );
    for ( all words $p$ and $s$ such that $\varepsilon < p < b$ , $\varepsilon < s < \mathcal{B}_1$ and $p\mathcal{B}_1 = bs$ )
        resolve( ( $p\mathcal{A}(n)$ , $as(\mathcal{B}_2)^n\mathcal{B}_3$ ) for all $n \in \mathbb{N}$ );
}
```

CritPair( words , triples )

```
CritPair( triples ( $\mathcal{B}$ , $\mathcal{A}$ ) , words ( $b$ , $a$ ) )
{
    //   Computes and resolves the critical pairs of ( ( $\mathcal{B}$(n), $\mathcal{A}$(n) ) , ( $b$ , $a$ ) ) for all n$\in\mathbb{N}$, where $\mathcal{B}$ and $\mathcal{A}$
    // are triples and $b$ and $a$ are words.
    for( all words $p$ , $s$ such that $pbs = \mathcal{B}_1$ )
        resolve( ( $pas\,(\mathcal{B}_2)^n\mathcal{B}_3$ , $\mathcal{A}$(n) ) for all n$\in\mathbb{N}$ );
    for( all words $p$ , $s$ and r$\in\mathbb{N}$ such that $p < \mathcal{B}_1$ , $s < \mathcal{B}_2$ and $pbs = \mathcal{B}_1(\mathcal{B}_2)^r$ )
        resolve( ( $pas\,(\mathcal{B}_2)^n\mathcal{B}_3$ , $\mathcal{A}$(r+n) ) for all n$\in\mathbb{N}$ );
    for( all words $p$ , $s$ and r$\in\mathbb{N}$ such that $p$ , $s < \mathcal{B}_2$ and $pbs = (\mathcal{B}_2)^r$ )
        resolve( ( $\mathcal{B}_1(\mathcal{B}_2)^{n_1}\,(pas)^n\,(\mathcal{B}_2)^{n_2}\,\mathcal{B}_3$ , $\mathcal{A}(n_1+n_2+nr)$ ) for all n$\in\mathbb{N}$ and $n_1,n_2<r$ );
    for( all words $p$ , $s$ and r$\in\mathbb{N}$ such that $p < \mathcal{B}_1$ , $s < \mathcal{B}_3$ and $pbs = \mathcal{B}$(r) )
        resolve( $pas$ , $\mathcal{A}$(r) );
    for( all words $p$ , $s$ and r$\in\mathbb{N}$ such that $p < \mathcal{B}_2$ , $s < \mathcal{B}_3$ and $pbs = (\mathcal{B}_2)^r\mathcal{B}_3$ )
        resolve( ( $\mathcal{B}_1(\mathcal{B}_2)^n\,pas$ , $\mathcal{A}$(r+n) ) for all n$\in\mathbb{N}$ );
    for( all words $p$ and $s$ such that $p < \mathcal{B}_3$ , $s < \mathcal{B}_3$ and $pbs = \mathcal{B}_3$ )
        resolve( ( $\mathcal{B}_1(\mathcal{B}_2)^n\,pas$ , $\mathcal{A}$(n) ) for all n$\in\mathbb{N}$ );
    for( all words $p$ , $s$ and r$\in\mathbb{N}$ such that $\varepsilon < p < \mathcal{B}_1$ , $\varepsilon < s < b$ and $pb = \mathcal{B}(r)s$ )
        resolve( $pa$ , $\mathcal{A}(r)s$ );
    for( all words $p$ , $s$ and r$\in\mathbb{N}$ such that $\varepsilon < p < \mathcal{B}_2$ , $\varepsilon < s < b$ and $pb = (\mathcal{B}_2)^r\mathcal{B}_3 s$ )
        resolve( ( $\mathcal{B}_1(\mathcal{B}_2)^n\,pa$ , $\mathcal{A}(r+n)s$ ) for all n$\in\mathbb{N}$ );
    for( all words $p$ , $s$ and r$\in\mathbb{N}$ such that $\varepsilon < p < \mathcal{B}_3$ , $\varepsilon < s < b$ and $pb = \mathcal{B}_3 s$ )
        resolve( ( $\mathcal{B}_1(\mathcal{B}_2)^n\,pa$ , $\mathcal{A}(n)s$ ) for all n$\in\mathbb{N}$ );
}
```

CritPair( triples , words )

Let ( $\mathcal{B}_1$ , $\mathcal{A}_1$ ) and ( $\mathcal{B}_2$ , $\mathcal{A}_2$ ) be (fixed) triples of $\mathcal{R}$. We are left with the problem of computing all those (critical) pairs:

*(3)* ( $\mathcal{A}_1(r)$ , $p\mathcal{A}_2(s)s$ ), whenever:

$$\mathcal{B}_1(r) = p\mathcal{B}_2(s)s.$$

*(4)* ( $\mathcal{A}_1(r)s$ , $p\mathcal{A}_2(s)$ ), whenever:

$$\mathcal{B}_1(r)s \equiv p\mathcal{B}_2(s)\ ,\ 0<|p|<|\mathcal{B}_1(r)| \text{ and } 0<|s|<|\mathcal{B}_2(s)|\ .$$

We will write $b_{ij}$ for $(\mathcal{B}_i)_j$ (for i=1,2 and j=1,2,3). We recall (proposition 4.1.6) that neither $\mathcal{B}_1$ nor $\mathcal{B}_2$ would be left or right sided. In particular, $|b_{12}|,|b_{22}| > 0$, and so we may define:

$$L=\mathrm{lcm}(\ |b_{12}|\ ,\ |b_{22}|\ ),$$

and then put:

$$L_1=L/|b_{12}| \text{ and } L_2=L/|b_{22}|\ .$$

Since (3) and (4) cannot occur simultaneously, it is natural to describe separate procedures for resolving the critical pairs (3) and (4) respectively. We will only be proving the correctness of the former, which is the more difficult, but still a straightforward corollary of the next lemma.

*4.4.1 Lemma.*

Suppose $\mathcal{B}_1(r) \equiv p\mathcal{B}_2(s)s$, and define:

$$f(p\ ,s\ )=\max(0,|p|+|b_{21}|-|b_{11}|\ )-\min(0,|b_{13}|-|b_{23}|-|s|\ ).$$

*(i)* Suppose:

$$b_{11}(b_{12})^{L_1} \text{ is a prefix of a word of the form } pb_{21}(b_{22})^{L_2}(b_{22})^*,$$

or

$$pb_{21}(b_{22})^{L_2} \text{ is a prefix of a word of the form } b_{11}(b_{11})^{L_1}(b_{12})^*.$$

Then, provided $p\in\mathbb{N}$ is such that $r-pL_1 \geq f(p\ ,s)/|b_{12}|$ ,

$$\mathcal{B}_1(r-pL_1+nL_1) \equiv p\mathcal{B}_2(s-pL_2+nL_2)s\ ,\ \text{for all } n\in\mathbb{N}.$$

*(ii)* $r \geq f(p\ ,s)/|b_{12}| + L_1 \Rightarrow$ the supposition of (i).

*(iii)* Suppose $p \equiv b_{11}(b_{12})^{\alpha}\overline{p}$ for some $\alpha\leq r$, then:-

$$\mathcal{B}_1(r-\alpha+m) \equiv b_{11}(b_{12})^{m}\overline{p}\,\mathcal{B}_2(s)s \text{ for all } m\in\mathbb{N},$$

and

$$r-\alpha < (\ L+|\tilde{p}|+|b_{21}|-\min(0,|b_{13}|-|b_{23}|-|s|)\ )/|b_{12}|\ .$$

*(iv)* Suppose $s \equiv \tilde{s}\,(b_{12})^{\gamma}\,b_{13}$ for some $\gamma \leq r$, then:-

$$\mathcal{B}_1(r-\gamma+n) \equiv p\mathcal{B}_2(s)\tilde{s}\,(b_{12})^n b_{13} \text{ for all } n \in \mathbb{N},$$

and

$$r-\gamma < (\ L+\max(0,|p|+|b_{21}|-|b_{11}|)+|b_{23}|-|\tilde{s}|)\ )/|b_{12}|\ .$$

*(v)* Supposing both (i) and (ii) hold, then, for all $m,n \in \mathbb{N}$,

$$\mathcal{B}_1(r-\alpha-\gamma+m+n) \equiv b_{11}(b_{12})^m\,\tilde{p}\,\mathcal{B}_2(s)\tilde{s}\,(b_{12})^n b_{13}\ ,$$

and

$$r-\alpha-\gamma < (\ L+|\tilde{p}|+|b_{21}|+|b_{23}|+|\tilde{s}|\ )/|b_{12}|\ .$$

The proofs of (i) and (ii) are a little tricky but, nevertheless, are no more than basic string manipulation. They would be of little or no interest and so we omit them.

*Proof of (iii):*

Substituting $p \equiv b_{11}(b_{12})^{\alpha}\tilde{p}$ in the identity:

$$b_{11}(b_{12})^r b_{13} \equiv p\mathcal{B}_2(s)s\ ,$$

we derive:-

$$b_{11}(b_{12})^r b_{13} \equiv b_{11}(b_{12})^{\alpha}\,\tilde{p}\,\mathcal{B}_2(s)s\ .$$

Then, because $\alpha \leq r$, we may cancel the common prefix $b_{11}(b_{12})^{\alpha}$ to derive:-

$$(b_{12})^{(r-\alpha)}b_{13} \equiv \tilde{p}\,\mathcal{B}_2(s)s\ .$$

Whence, for all $m \in \mathbb{N}$,

$$(b_{11}(b_{12})^m)(b_{12})^{(r-\alpha)}b_{13} \equiv (b_{11}(b_{12})^m)\,\tilde{p}\,\mathcal{B}_2(s)s\ ,$$

i.e.,

$$(5)\quad \mathcal{B}_1(r-\alpha+m) \equiv b_{11}(b_{12})^m\,\tilde{p}\,\mathcal{B}_2(s)s$$

(which is the first part of (iii)).

We can easily complete the proof of (iii) by proving the following:

*(6) Claim.*

If $r \geq f(p,s)/|b_{12}| + L_1$, *in addition to (iii)*, then the triple $\mathcal{B}_2$ is right sided.

*Proof:*

If $r \geq f(p,s)/|b_{12}| + L_1$, then (ii) would imply (i), i.e.,

$$\textit{(7)}\ \text{provided } r - \mathrm{p}L_1 \geq f(p,s)/|b_{12}|, \text{ then}$$

$$\mathcal{B}_1(r - \mathrm{p}L_1 + nL_1) \equiv p\mathcal{B}_2(s - \mathrm{p}L_2 + nL_2)s, \text{ for all } n \in \mathbb{N}.$$

So we may substitute $m = \alpha + L_1$ in (5), and $\mathrm{p} = 0$, $n = 1$ in (7), to derive two expressions for the word $\mathcal{B}_1(r + L_1)$, namely:-

$$b_{11}(b_{12})^{(\alpha+L_1)}\bar{p}\,\mathcal{B}_2(s)s \equiv pb_{21}(b_{22})^{(s+L_2)}\,b_{23}s\,.$$

We can then cancel the common suffix $(b_{22})^s b_{23}s$ to obtain :-

$$b_{11}(b_{12})^{(\alpha+L_1)}\bar{p}\,b_{21} \equiv pb_{21}(b_{22})^{L_2}\,.$$

Whence, by comparing the suffixes of the latter identity, $b_{21}$ is a suffix of $b_{21}(b_{22})^{L_2}$, which, by lemma 4.1.3, is a criterion of the triple $\mathcal{B}_2$ being right sided.

$\boxed{6}$

Now, we are assuming (iii), and so, by (6) and the proviso that neither of the triples $\mathcal{B}_1$ nor $\mathcal{B}_2$ was left or right sided, we must have:-

$$\textit{(8)}\ r < f(p,s)/|b_{12}| + L_1\,.$$

We are assuming $p \equiv b_{11}(b_{12})^{\alpha}\bar{p}$, therefore:-

$$f(p,s) = \max(0, |p| + |b_{21}| - |b_{11}|) - \min(0, |b_{13}| - |b_{23}| - |s|)$$

$$= \alpha|b_{12}| + |\bar{p}| + |b_{21}| - \min(0, |b_{13}| - |b_{23}| - |s|).$$

Substituting for $f(p,s)$ in (8):-

$$r < (\alpha|b_{12}| + |\bar{p}| + |b_{21}| - \min(0, |b_{13}| - |b_{23}| - |s|))/|b_{12}| + L_1\,,$$

but $L_1 = L/|b_{12}|$, whence:-

$$\textit{(9)}\ r - \alpha < (L + |\bar{p}| + |b_{21}| - \min(0, |b_{13}| - |b_{23}| - |s|))/|b_{12}|$$

(which is the second part of (iii)).

$\boxed{4.4.1(\mathrm{iii})}$

*Proof of (v):*

By (iii), we have:-

$$(5)\quad \mathcal{B}_1(r-\alpha+m) \equiv b_{11}(b_{12})^m \tilde{p}\, \mathcal{B}_2(s)s \text{, for all } m \in \mathbb{N},$$

and, by (iv),

$$(10)\quad \mathcal{B}_1(r-\gamma+n) \equiv p\mathcal{B}_2(s)\, \tilde{s}\, (b_{12})^n\, b_{13} \text{, for all } n \in \mathbb{N}.$$

Applying (5) and (10) concurrently we obtain, for all m, n∈ $\mathbb{N}$,

$$\mathcal{B}_1(r-\alpha-\gamma+m+n) \equiv b_{11}(b_{12})^m\, \tilde{p}\, \mathcal{B}_2(s)\, \tilde{s}\, (b_{12})^n b_{13}$$

(which is the first part of (v)).

By (iii), we have (9). Also, as we are assuming $s \equiv \tilde{s}\, (b_{12})^\gamma\, b_{13}$, therefore:-

$$-\min(0,|b_{13}|-|b_{23}|-|s|) = \gamma|b_{12}|+|\tilde{s}|+|b_{23}|\,.$$

Substituting the latter expression in (9) and rearranging we obtain:-

$$r-\alpha-\gamma < (\, L+|\tilde{p}|+|b_{21}|+|\tilde{s}|+|b_{23}|\,)/|b_{12}|$$

(which is the second part of (v)).

$\boxed{\text{4.4.1(v) and 4.4.1}}$

*4.4.2 Corollary.*

Suppose $\mathcal{B}_1(r) \equiv p\mathcal{B}_2(s)s$ , $p \equiv b_{11}\tilde{p}$ , where $\tilde{p}$ is a proper prefix of $b_{12}$ or of $b_{13}$ , and $s \equiv \tilde{s}\, b_{13}$ , where $\tilde{s}$ is a proper suffix of $b_{12}$ or of $b_{11}$. Then, for all m,n∈ $\mathbb{N}$,

$$(11)\quad (\, \mathcal{A}_1(r+m+n)\, ,\, b_{11}(b_{12})^m\, \tilde{p}\, \mathcal{A}_2(s)\, \tilde{s}\, (b_{12})^n b_{13}\, )$$

are critical pairs which can be resolved (without resorting to 2-parameterized rules) by the following procedure.

```
{
    if ( $b_{12}\, \tilde{p}\, \mathcal{A}_2(s)\tilde{s} > \tilde{p}\, \mathcal{A}_2(s)\tilde{s}\, b_{12}$ )
    {
        secure( $b_{12}\, \tilde{p}\, \mathcal{A}_2(s)\tilde{s}\, ,\, \tilde{p}\, \mathcal{A}_2(s)\tilde{s}\, b_{12}$ );
        resolve( ( $\mathcal{A}_1(r+\kappa)\, ,\, b_{11}\, \tilde{p}\, \mathcal{A}_2(s)\tilde{s}\, (b_{12})^\kappa\, b_{13}$ ) for all $\kappa \in \mathbb{N}$ )
    }
    else
    {
        secure( $\bar{p}\,\mathcal{A}_2(s)\bar{s}\,b_{12}$ , $b_{12}\,\bar{p}\,\mathcal{A}_2(s)\bar{s}$ );
        resolve( ( $\mathcal{A}_1(r+\kappa)$ , $b_{11}(b_{12})^{\kappa}\,\bar{p}\,\mathcal{A}_2(s)\,\bar{s}\,b_{13}$ ) for all $\kappa\in\mathbb{N}$ )
    }
}
```

*Proof:*

By lemma 4.4.1(v), we know that, for all $m,n\in\mathbb{N}$,

$$(12)\quad \mathcal{B}_1(r+m+n)\equiv b_{11}(b_{12})^{m}\bar{p}\,\mathcal{B}_2(s)\bar{s}\,(b_{12})^{n}b_{13}\ .$$

so the pairs of (11) will be critical pairs (as claimed). Also, substituting first $n=1$, $m=0$ in (11), and then $n=0$, $m=1$ in (11), yields two expressions for the word $\mathcal{B}_1(r+1)$, namely:-

$$b_{11}b_{12}\,\bar{p}\,\mathcal{B}_2(s)\bar{s}\,b_{13}\equiv b_{11}\bar{p}\,\mathcal{B}_2(s)\bar{s}\,b_{12}b_{13}\ .$$

By cancelling the common prefix, $b_{11}$, and the common suffix, $b_{13}$, we have:-

$$b_{12}\bar{p}\,\mathcal{B}_2(s)\bar{s}\equiv\bar{p}\,\mathcal{B}_2(s)\bar{s}\,b_{12}\ ,$$

and so the relation:

$$b_{12}\bar{p}\,\mathcal{A}_2(s)\bar{s}=_G\bar{p}\,\mathcal{A}_2(s)\bar{s}\,b_{12}\ .$$

is a consequence of the rules (12).

So, supposing:

$$(13)\quad b_{12}\bar{p}\,\mathcal{A}_2(s)\bar{s}\ \to^{*}\ \bar{p}\,\mathcal{A}_2(s)\bar{s}\,b_{12}\ ,$$

and

$$(14)\quad b_{11}\bar{p}\,\mathcal{A}_2(s)\bar{s}\,(b_{12})^{\kappa}b_{13}\vee\mathcal{A}_1(r+\kappa),\ \text{for all }\ \kappa\in\mathbb{N}.$$

We would then have, for all $m,n\in\mathbb{N}$:-

$$\begin{aligned} b_{11}(b_{12})^{m}\bar{p}\,\mathcal{A}_2(s)\bar{s}\,(b_{12})^{n}b_{13}\ &\to^{*}\ b_{11}(b_{12})^{(m-1)}\bar{p}\,\mathcal{A}_2(s)\bar{s}\,(b_{12})^{(n+1)}b_{13}\\ &\to^{*}\cdots\\ &\cdots\to^{*}\ b_{11}\bar{p}\,\mathcal{A}_2(s)\bar{s}\,(b_{12})^{(m+n)}b_{13}\,,\\ &\qquad\text{applying (13) m times,}\\ &\vee\ \mathcal{A}_1(r+m+n),\\ &\qquad\text{by (14).}\end{aligned}$$

Supposing:

$$(15)\ \tilde{p}\,\mathcal{A}_2(s)\tilde{s}\,b_{12} \rightarrow^{*} b_{12}\,\tilde{p}\,\mathcal{A}_2(s)\tilde{s}\,,$$

and

$$(16)\ b_{11}(b_{12})^{\kappa}\,\tilde{p}\,\mathcal{A}_2(s)\,\tilde{s}\,b_{13} \vee \mathcal{A}_1(r+\kappa),\ \text{for all}\ \ \kappa\in\mathbb{N}.$$

We would then have, for all $m,n\in\mathbb{N}$:-

$$\begin{aligned} b_{11}(b_{12})^{m}\tilde{p}\,\mathcal{A}_2(s)\,\tilde{s}\,(b_{12})^{n}\,b_{13} &\rightarrow^{*} b_{11}(b_{12})^{(m+1)}\tilde{p}\,\mathcal{A}_2(s)\tilde{s}\,(b_{12})^{(n-1)}b_{13} \\ &\rightarrow^{*} \cdots \\ &\cdots \rightarrow^{*} b_{11}(b_{12})^{(m+n)}\,\tilde{p}\,\mathcal{A}_2(s)\,\tilde{s}\,b_{13}\,, \\ &\qquad \text{applying (15) n times,} \\ &\vee\ \mathcal{A}_1(r+m+n), \\ &\qquad \text{by (16).} \end{aligned}$$

$\boxed{4.4.2}$

We can now describe the procedure for resolving the critical pairs:

$$(3)\ \ (\ \mathcal{A}_1(r)\ ,\ p\mathcal{A}_2(s)s\ ),\ \text{whenever:}$$

$$\mathcal{B}_1(r)=p\mathcal{B}_2(s)s\,.$$

```
CritPair$_1$( triples ( $\mathcal{B}_1$ , $\mathcal{A}_1$ ) , triples ( $\mathcal{B}_2$ , $\mathcal{A}_2$ ) )
{

    //    Computes and resolves the critical pair(s)
    //                                    (3)  ( $\mathcal{A}_1(r)$ , $p\mathcal{A}_2(s)s$ ),  whenever:
    //                                               $\mathcal{B}_1(r)=p\mathcal{B}_2(s)s$ .

    $b_{ij}=(\mathcal{B}_i)_j$  for i=1,2 and j=1,2,3;
    L=lcm( $|b_{12}|$ , $|b_{22}|$ );
    L$_1$=L/$|b_{12}|$;
```

```
$L_2 = L/|b_{22}|$;
# define $f(p, s) = \max(0, |pb_{21}| - |b_{11}|) - \min(0, |b_{13}| - |b_{23}s|)$;
for( all proper prefixes, $p$, of $b_{12}$ or $b_{13}$ )
    for( all proper suffixes, $s$, of $b_{13}$ )
        for( $r=0$; $r < (L + |pb_{21}| - \min(0, |b_{13}| - |b_{23}s|))/|b_{12}|$; $r=r+1$ )
            if ( $b_{11}(b_{12})^r b_{13} = b_{11}\, p\, b_{21}(b_{22})^S b_{23}s$ )
                resolve( ( $\mathcal{A}_1(r+m)$ , $b_{11}(b_{12})^m p\mathcal{A}_2(s)s$ ) for all $m \in \mathbb{N}$ );
for( all proper prefixes, $p$, of $b_{11}$ )
    for( all proper suffixes, $s$, of $b_{12}$ or $b_{11}$ )
        for( $r=0$; $r < (L + \max(0, |pb_{21}| - |b_{11}|) + |b_{23}s|)/|b_{12}|$; $r=r+1$ )
            if ( $b_{11}(b_{12})^r b_{13} = p\, b_{21}(b_{22})^S b_{23}s b_{13}$ )
resolve( ( $\mathcal{A}_1(r+n)$ , $p\,\mathcal{A}_2(s)s(b_{12})^n b_{13}$ ) for all $m \in \mathbb{N}$ );
    for( all proper prefixes, $p$, of $b_{12}$ or $b_{13}$ )
        for( all proper suffixes, $s$, of $b_{12}$ or $b_{11}$ )
            for( $r=0$; $r < (L + |pb_{21}| + |b_{23}s|)/|b_{12}|$; $r=r+1$ )
                if ( $b_{11}(b_{12})^r b_{13} = b_{11}\, p\, b_{21}(b_{22})^S b_{23}s b_{13}$ )
                {
                    if ( $b_{12}p\mathcal{A}_2(s)s > p\mathcal{A}_2(s)sb_{12}$ )
                    {
                        secure( $b_{12}p\mathcal{A}_2(s)s$ , $p\mathcal{A}_2(s)sb_{12}$ );
                        resolve( ( $\mathcal{A}_1(r+n)$ , $b_{11}p\mathcal{A}_2(s)s\,(b_{12})^n b_{13}$ ) for all $n \in \mathbb{N}$ );
                    }
                    else
                    {
                        secure( $p\mathcal{A}_2(s)sb_{12}$ , $b_{12}p\mathcal{A}_2(s)s$ );
                        resolve( ( $\mathcal{A}_1(r+n)$ , $b_{11}(b_{12})^n\, p\mathcal{A}_2(s)s\, b_{13}$ ) for all $n \in \mathbb{N}$ );
                    }
                }
for( all proper prefixes, $p$, of $b_{11}$ )
    for( all proper suffixes, $s$, of $b_{13}$)
        for( $r=0$; $r < f(p, s)/|b_{12}| + L_1$; $r=r+1$ )
            if ( $b_{11}(b_{12})^r b_{13} = p\, b_{21}(b_{22})^s b_{23}s$ )
            {
                if (
                        $r > f(p, s)/|b_{12}|$ ,
                        and either:
              $b_{11}(b_{12})^{L_1}$ is a prefix of a word of the form  $pb_{21}(b_{22})^{L_2}(b_{22})^*$
                        or
              $pb_{21}(b_{22})^{L_2}$ is a prefix of a word of the form  $b_{11}(b_{12})^{L_1}(b_{12})^*$
          )
              resolve( ($\mathcal{A}_1(r+nL_1)$ , $p\mathcal{A}_2(s+nL_2)\,s$ )  for all $n \in \mathbb{N}$ ) ;
          else
              resolve( $\mathcal{A}_1(r)$ , $p\mathcal{A}_2(s)s$ );
      }
}
```

CritPair$_1$( triples , triples )

*4.4.3 Corollary.*

The proof of correctness of CritPair$_1$( ), i.e., the (critical) pair

$$(3)\quad ( \mathcal{A}_1(r) , p\mathcal{A}_2(s)s )$$

is resolved if and only if

$$(17)\quad \mathcal{B}_1(r) = p\mathcal{B}_2(s)s \, .$$

*Proof:*

The proof that we resolve only the critical pairs is a special case of lemma 4.4.1 (i), (iii), (iv), or of corollary 4.4.2 – none of which we bother to restate.

Conversely, we suppose (17) holds. Then, the proof that the critical pair (3) will be resolved is basically a restatement of lemma 4.4.1. We suppose (for example):

$$|s| < |b_{13}| \text{ and } |p| \geq |b_{11}| \, .$$

Then $s$ would be a proper suffix of $b_{13}$ and we could write $p = b_{11}(b_{12})^{\alpha}\overline{p}$ where $\alpha \leq r$ and $\overline{p}$ is proper prefix of $b_{12}$ or of $b_{13}$ .

By lemma 4.4.1 (iii), we would have:-

$$\mathcal{B}_1(r-\alpha+m) = b_{11}(b_{12})^{m}\,\overline{p}\,\mathcal{B}_2(s)s \, , \text{ for all } m \in \mathbb{N},$$

and

$$r-\alpha < (\ L+|\tilde{p}|+|b_{21}|-\min(0,|b_{13}|-|b_{23}|-|s|)\ )/|b_{12}|\ .$$

With $\mathcal{B}_1(r-\alpha)=b_{11}\,\tilde{p}\,\mathcal{B}_2(s)s$ , and $s$ being a proper suffix of $b_{13}$ , so the pairs:

$$(\ \mathcal{A}_1(r-\alpha+m)\ ,\ b_{11}(b_{12})^m\tilde{p}\,\mathcal{A}_2(s)\ )$$

would be resolved for all $m\in\mathbb{N}$ (which includes (3)).

Similarly, by lemma 4.4.1 (iv), we could confirm that the pair (3) is resolved when:

$$|s|\geq|b_{13}| \text{ and } |p|<|b_{11}|\ ,$$

and, by lemma 4.4.1(v), when:

$$|s|\geq|b_{13}| \text{ and } |p|\geq|b_1|\ .$$

So, we may now assume:

$$|s|<|b_{13}| \text{ and } |p|<|b_{11}|\ ,$$

i.e., $s$ will be a proper suffix of $b_{13}$ and $p$ will be a proper prefix of $b_{11}$.

If, in addition:

$$r< f(p\,,s)/|b_{12}|+L_1,$$

then the pair (3) is easily seen to be resolved. So we will also assume:

$$r\geq f(p\,,s)/|b_{12}|+L_1.$$

Then, by lemma 4.4.1 (ii) and (i),

$$b_{11}(b_{12})^{L_1} \text{ is a prefix of a word of the form } p b_{21}(b_{22})^{L_2}(b_{22})^*\ ,$$

or

$$p\, b_{21}(b_{22})^{L_2} \text{ is a prefix of a word of the form } b_{11}(b_{11})^{L_1}(b_{12})^*\ .$$

and, provided $p\in\mathbb{N}$ is such that $r-pL_1\geq f(p\,,s)/|b_{12}|$ , then:-

$$\mathcal{B}_1(r-pL_1+nL_1) = p\mathcal{B}_2(s-pL_2+nL_2)s\ , \text{ for all } n\in\mathbb{N}.$$

So, by choosing $p\in\mathbb{N}$ maximal so that:

$$r-pL_1\geq f(p\,,s)/|b_{12}|\ ,$$

we will have:-

$$f(p\,,s)/|b_{12}|+L_1 > r-pL_1\geq f(p\,,s)/|b_{12}|\ .$$

Then, with $\mathcal{B}_1(r-pL_1)= p(s-pL_2)s$, $p$ being a proper prefix of $b_{11}$ and $s$ being a proper suffix of $b_{13}$, so the pairs:

$$( \mathcal{B}_1(r-pL_1+nL_1)\ ,\ p\mathcal{B}_2(s-pL_2+nL_2)s\ )$$

would be resolved for all $n \in \mathbb{N}$ (which includes (3)).

$\boxed{4.4.3}$

We conclude this section by describing the procedure which resolves the pairs

$$(4)\ \ ( \mathcal{A}_1(r)s\ ,\ p\mathcal{A}_2(s)\ ),\ \text{whenever:}$$

$$\mathcal{B}_1(r)s \equiv p\mathcal{B}_2(s)\ \ \text{with}\ \ 0<|p|<|\mathcal{B}_1(r)|\ \ \text{and}\ \ 0<|s|<|\mathcal{B}_2(s)|\,.$$

We omit the proof of correctness (being a restatement of the proof of corollary 4.4.3). Note that it may be necessary for this procedure to store 2-parameterized critical pairs which we must (later) prove to be resolved for $\mathcal{R}^{(\mathbb{N})}$ to be complete (cf procedure resolved?() defined on page 77). Such pairs will be computed if there are instances of (4) with both $p$ and $s$ are relatively large, i.e., $|p|>|b_{11}|$ and $|s|>|b_{13}|$.

```
CritPair$_2$( triples ( $\mathcal{B}_1$ , $\mathcal{A}_1$ ) , triples ( $\mathcal{B}_2$ , $\mathcal{A}_2$ ) )
{

//    Computes and resolves the critical pair(s)
//                              (4)  ( $\mathcal{A}_1(r)s$ , $p\mathcal{A}_2(s)$ ),  whenever:
//                                         $\mathcal{B}_1(r)s \equiv p\mathcal{B}_2(s)$ .

    $b_{ij}=(\mathcal{B}_i)_j$  for i=1,2 and j=1,2,3;
    L=lcm( $|b_{12}|$ , $|b_{22}|$ );
    $L_1=L/|b_{12}|$ ;
    $L_2=L/|b_{22}|$ ;
    # define $f(p,s)=\max(0, |pb_{21}|-|b_{11}|)-\min(0, |b_{13}s|-|b_{23}|)$;
    for( all proper prefixes, $p$, of $b_{12}$ or $b_{13}$ )
        for( all proper suffixes, $s$, of $b_{23}$ )
            for( r=0; $r<(L+|pb_{21}|-\min(0, |b_{13}s|-|b_{23}|))/|b_{12}|$ ; r=r+1 )
                if ( $b_{11}(b_{12})^r b_{13}s \equiv b_{11}\, p\, b_{21}(b_{22})^s b_{23}$ )
                    resolve ( ( $\mathcal{A}_1(r+m)s$ , $b_{11}(b_{12})^m p\mathcal{A}_2(s)$ ) for all $m \in \mathbb{N}$ );
for( all proper prefixes, $p$, of $b_{11}$ )
    for( all proper suffixes, $s$, of $b_{22}$ or $b_{21}$ )
        for( r=0; r< ( L+max( 0 , $|pb_{21}|$ - $|b_{11}|$ ) )/$|b_{12}|$ ; r=r+1 )
            if ( $b_{11}(b_{12})^r b_{13} s b_{23} = p\, b_{21}(b_{22})^s b_{23}$ )
                resolve ( ( $\mathcal{A}_1(r) s (b_{22})^n\, b_{23}$ , $p\, \mathcal{A}_2(s+n)$ ) for all m$\in \mathbb{N}$ );
for( all proper prefixes, $p$, of $b_{12}$ or $b_{13}$ )
    for( all proper suffixes, $s$, of $b_{22}$ or $b_{21}$ )
        for(r=0; r< ( L+$|p|$+$|b_{21}|$ )/$|b_{12}|$ ; r=r+1 )
            if ( $b_{11}(b_{12})^r b_{13} s b_{23} = b_{11}\, p\, b_{21}(b_{22})^s b_{23}$ )
                store( ( $\mathcal{A}_1(r+m) s\, (b_{22})^n\, b_{13}$ , $b_{11}(b_{12})^m\, p\, \mathcal{A}_2(s+n)$ ) for all n$\in \mathbb{N}$ );
                    // We store these 2-parameterized critical pairs  disjoint from $\mathcal{R}^{(\mathbb{N})}$. When, if
                    // ever, the completion procedure stops, then, to prove that $\mathcal{R}^{(\mathbb{N})}$ is complete,
                    // we must confirm that all the words of these 2-parameterized critical pairs
                    // have common $\mathcal{R}^{(\mathbb{N})}$-descendants (cf. procedure resolved?() (page 77)).
for( all proper prefixes, $p$, of $b_{11}$ )
    for( all proper suffixes, $s$, of $b_{23}$ )
        for( r=0; r< f($p$, $s$)/$|b_{12}|$ + $L_1$ ; r=r+1 )
            if ( $b_{11}(b_{12})^r b_{13} s = p b_{21}(b_{22})^s b_{23}$ )
            {
                if (
                                        r > f($p$, $s$)/$|b_{12}|$
                                        and either:
                        $b_{11}(b_{12})^{L_1}$ is a prefix of a word of the form  $p b_{21}(b_{22})^{L_2}(b_{22})^*$
                                        or
                        $p b_{21}(b_{22})^{L_2}$ is a prefix of a word of the form  $b_{11}(b_{12})^{L_1}(b_{12})^*$
                )
                    resolve( ( $\mathcal{A}_1(r+nL_1) s$ , $p \mathcal{A}_2(s+nL_2)$ )  for all n$\in \mathbb{N}$ ) ;
                else
                    resolve( $\mathcal{A}_1(r) s$ , $p \mathcal{V}_2(s)$ );
            }
}
```

CritPair$_2$( triples , triples )

## 4.5

## Program Report

This is a brief report on the implementation of the program described in the first four sections of this chapter. The program was written in C++ on a Sun 1/130.

The complete presentation of the free abelian group of rank 2 of example 3.1.5 took about a second to complete.

Recall the abelian by cyclic group, of lemma 3.1.6, with (semigroup) presentation:

$$\langle\, a, a^{-1}, b, b^{-1}, c, c^{-1} \mid (aa^{-1}, \varepsilon), (a^{-1}a, \varepsilon), (bb^{-1}, \varepsilon), (b^{-1}b, \varepsilon),$$
$$(cc^{-1}, \varepsilon), (c^{-1}c, \varepsilon), (ba, ab), (ca, bc) \,\rangle.$$

With the SortLex ordering $<_1$ defined by $a <_1 a^{-1} <_1 b <_1 b^{-1} <_1 c <_1 c^{-1}$, the completion, listed on page 24, took 2 seconds. With the SortLex ordering $<_2$ defined by $c <_2 c^{-1} <_2 a <_2 a^{-1} <_2 b <_2 b^{-1}$, the completion, listed on page 26, took about 70 seconds and the resolution of some of the 2-parameterized critical pairs had to be confirmed by hand.

The surface group of a torus with p holes has group presentation:

$$\langle\, a_1, a_2, \ldots, a_{2p} \mid (a_{2p}a_{(2p-1)} \cdots a_1, a_1a_2 \cdots a_{2p}) \,\rangle.$$

With the ShortLex ordering defined by $a_1 < a_1^{-1} < a_2 < a_2^{-1} < \ldots < a_{2p} < a_{2p}^{-1}$, we computed minimal complete $P_1$ presentations for p=2,3 and 4. The completion of the p=2 presentation took 6 seconds; the completion of the p=3 presentation took 33 seconds; and the completion of the p=4 presentation took 118 seconds.

It would not be difficult to work out the complete $P_1$ presentation for general p, but Le Chenadec has already catalogued finite complete presentations with respect to the ShortLex ordering defined by:

$$a_{(2p-1)}^{-1} < a_{(2p-1)} < \ldots < a_3^{-1} < a_3 < a_1^{-1} < a_1 < a_2^{-1} < a_2 < \ldots < a_{(2p-2)}^{-1} < a_{(2p-2)} < a_{2p}^{-1} < a_{2p}$$

(see (Le Chenadec)).

The Coxeter group with presentation:

$$(\, a, b, c, d \mid (aa, \varepsilon), (bb, \varepsilon), (cc, \varepsilon), (dd, \varepsilon), (dada, adad),$$
$$(dbdb, bdbd), (dcd, cdc) \,),$$

took 31 seconds to complete. We believe, but have not proved, that it has no finite, complete (ShortLex) presentation.

In (Le Chenadec) Le Chenadec describes (not necessarily finite) complete presentations for the Coxeter groups with no two generators commuting - there are difficulties with partial commutivity of the generators. We were able, however, to compute a complete $P_1$ presentation for the coxeter group:

$$\langle\, a, b, c, d \mid (aa, \varepsilon), (bb, \varepsilon), (cc, \varepsilon), (dd, \varepsilon), (ca, ac), (cb, bc),$$
$$(dad, ada), (dbd, bdb), (dcd, cdc) \,\rangle,$$

in 15 seconds. We believe, but have not proved, that there is no finite complete (ShortLex) presentation.

With the input of both these Coxeter group presentations the program was unable to compute all the necessary 1-parameterized rules. In both cases we had to run the program for a about 30 seconds, guess some of the 1-parameterized rules which the program was unable to predetermine, and then include these rules as additional data in a (successful) rerun of the program.

The completion of the (2,3,8) group, $\langle\, a, b \mid ((a)^8, \varepsilon), ((b)^3, \varepsilon), (abab, \varepsilon) \,\rangle$, suggested by Gilman(84) took 8 seconds.

The Dyck groups have presentations:

$$D(n_1, n_2, \ldots, n_k) = \langle\, a_1, a_2, \ldots, a_k \mid ((a_1)^{n_1}, \varepsilon), \ldots, ((a_k)^{n_k}, \varepsilon), (a_1 a_2 \ldots a_k, \varepsilon) \,\rangle.$$

We tried numerous Dyck group presentations and ShortLex orderings, all the presentations had either finite or (infinite) $P_1$ presentations. With the ShortLex ordering defined by $a_1 < a_1^{-1} < a_2 < a_2^{-1} < \ldots < a_k < a_k^{-1}$, there was a $P_1$ completion of D(6,5,5,5) which took 69 seconds.

It is interesting to note that with respect to the ShortLex ordering defined by:

$$a_1 < a_2 < \ldots < a_p < a_1^{-1} < a_2^{-1} < \ldots < a_p^{-1} <$$
$$a_{(p+1)} < a_{(p+2)} < \ldots < a_{2p} < a_{(p+1)}^{-1} < a_{(p+2)}^{-1} < \ldots < a_{2p}^{-1},$$

all our examples of Dyck groups on a even number of generators (i.e. k=2p) had finite complete presentations. We infer from this that there might well be a better ordering than that suggested by Le Chenadec where only *confluence* is proved (Le Chenadec).

# §5

# *Almost Convex Cayley Graphs*

## *5.0*

The concept of *almost convex groups* or, more precisely, groups with almost convex Cayley graphs, is due to J.W. Cannon and is first defined in Cannon's 1984 preprint of that title. The class of almost convex groups is large, and of interest because the (geometrical) property of a Cayley graph being almost convex means that it is (in theory at least) recursive, in fact there is an efficient method for constructing such graphs (theorem 5.1.3). In Cannon's preprint the interested reader will find proofs that the following classes of groups are almost convex: the groups satisfying the small cancellation hypothesis, HNN extensions of finite groups, free products with amalgamation of two finite groups, and discrete groups of Euclidean isometries (the latter groups being free abelian by finite).

There are, however, numerous problems in the subject of almost convexity, notably that the property of a Cayley graph being almost convex does tend to be difficult to prove or disprove in practice. Also (unlike the automatic groups), it is not known whether a group being almost convex is independent of the choice of (inverse closed) generators (although we can prove a partial result on the independence of generators, i.e., proposition 5.1.4).

We will begin this chapter with a summary of Cannon's (defining) work which appeared in his preprint, notably what is meant by a Cayley graph being almost convex and a description of the procedure which constructs almost convex Cayley graphs. Then, in section 5.2 we will prove that the class of *word length preserving*, complete, $P_r$ groups are almost convex (i.e. proposition 5.2.1) and, the analog for automatic groups (i.e. theorem 5.2.10), that the class of *least length bounded* automatic groups are almost convex (the latter is due to (CEHPT)). We believe these subclasses are strict: in section 5.3 we prove that the matrix group U(3,$\mathbb{Z}$) is almost convex, but this group is known not to be automatic, and, we *conjecture*, it has no ShortLex, complete, parameterized presentation.

We conclude the chapter with an alternative, generalized, proof of Cannon's theorem that the free abelian by finite groups are almost convex (theorem 5.4.1). The proof, by reference to 5.2.10, is algebraic, as opposed to Cannon's geometric proof, and is, arguably, the simpler proof.

## *5.1*

At this point we recommend that the subsection *Cayley Graphs* (page 5), of (the definition and terminology) chapter 1 be reread, but we make no apologies for repeating the following fundamental definitions and facts. We think of the *whole* of $\Gamma=\Gamma_C(G)$ as a connected (path) metric space with metric $d=d_C$; we allow retracing of paths; a *geodesic* path is a shortest path between its endpoints. If $g\in G$, then $d(1_G,g)=\|g\|$ and, if $g,h\in G$, then $d(g,h)=\|gh^{-1}\|$. We then defined (for all $r\in\mathbb{R}$) the r-ball of $\Gamma$ to be $\{\, p\in\Gamma \mid d(1_G,p)\leq r \,\}$, and the r-shell to be $\{\, p\in\Gamma \mid d(1_G,p)=r \,\}$. The vertices of $\Gamma$ are distinguished by the fact that they are precisely those points of $\Gamma$ at integer distances from the basepoint; if $n\in\mathbb{N}$, then $S(n)$ consists precisely of those $g\in G$ with norm n. When we refer to an edge or path *staying within* a ball $B(r)$ ($r\in\mathbb{R}$), we mean that all points of that edge or path lie in $B(r)$. Whence, an edge stays within $B(r)$ if and only if at least one of its endpoints is in $B(r-1)$, a path connecting two vertices stays within $B(n)$ ($n\in\mathbb{N}$) if and only if at least one of the end points of *every* edge that it traverses belongs to $S(n-1)$.

We now define, for all $\kappa\in\mathbb{N}$, the relation *join(κ)*, on the points of $\Gamma_C(G)$, by: $p \,{}_{\text{join}(\kappa)}\, \bar{p}$ if and only if p is joined to $\bar{p}$ by at least one path which stays within $B(\|p\|)$ and has length no more than κ. These relations were introduced as abbreviations and are used frequently throughout this chapter (note that, apart from the trivial cases, join(κ) is neither symmetric nor transitive).

We can now define the geometrical property of *almostconvexity*.

### *5.1.1 Definition (Cannon).*

Let G be a group generated by C, then:

*(i)* If κ is a positive number then $\Gamma_C$ (G) is said to be *almost convex( κ )*,which may be written as *ac(κ)*, if there is an integer b(κ) with the property that, whenever $g,\bar{g}\in G$ are such that $\|g\|=\|\bar{g}\|$ with $d(g,\bar{g})\leq\kappa$, then $g \,{}_{\text{join}(b(\kappa))}\, \bar{g}$. *(ii)* The Cayley graph $\Gamma_C$ (G) is said to be *almost convex*,which may be written as ac, if it is almost convex(κ) for all $\kappa\in\mathbb{N}$.

$\boxed{5.1.1}$

*5.1.2 Theorem (Cannon).*

If G is generated by C and $\Gamma_C$ (G) is almost convex(2), then $\Gamma_C$ (G) is almost convex.

*Proof:*

We take arbitrary $\kappa$ and prove that $\Gamma_C$ (G) is almost convex($\kappa$). So let us suppose that, for some $n \in \mathbb{N}$, $g, \bar{g} \in S(n)$ with $d(g, \bar{g}) \leq \kappa$. Let $\rho$ be a geodesic path between g and $\bar{g}$, and then define $m \in \mathbb{N}$ to be maximal so that $\rho$ does not stay completely within B(m). We note that:–

$$(1)\ |\rho| \leq \kappa,$$

and so we may as well assume that $\rho$ does not stay within B(m). Whence, $m \leq n+|\rho|$, and we have:–

$$(2)\ n-1 \leq m \leq n+|\rho| .$$

Now, whilst $m \geq n$ there will be subpaths of $\rho$ with just their endpoints belonging to S(m), we argue that all such subpaths have length at most 2. This is because no edge of such a subpath could have both its endpoints outside of B(m) (without all intermediate points on the edge lying outside of B(m) (which would contradict the choice of m)). Whence, such subpaths must be a single edge joining the same vertex of S(m), or else a path of length 2 which joins two vertices in S(m). Whichever, we see that the subpaths of $\rho$, with just their endpoints belonging to S(m), will have length at most 2. Whence, by the fact that $\Gamma_C$ (G) is ac(2), we may replace each of these paths by paths which stay within B(m) and have length bounded by b(2). As there are at most $|\rho|$ such subpaths of $\rho$, so, at the cost of increasing the length of $\rho$ by a factor of at most b(2), we may push $\rho$ inside the ball B(m). By (2), we see that this process need be repeated at most $|\rho|+1$ times before $\rho$ is pushed completely within B(n). Thus g and $\bar{g}$ are joined by a path lying within B(n) and of length at most $|\rho|\, b(2)^{(|\rho|+1)}$, which, by (1), $\leq |\kappa|\, b(2)^{(|\kappa|+1)}$.

$\boxed{5.1.2}$

*5.1.3 Theorem (Cannon).*

If $\Gamma_C$ (G) is almost convex, then $\Gamma_C$ (G) is recursive, i.e., there is a finite procedure for constructing B(n) for all $n \in \mathbb{N}$.

*Proof:*

We will describe the procedure without proof of correctness (which may be found in (Cannon)).

Let R be the set of all those relators of G which have length at most b(2) (where b(2) is as in definition 5.1.1. Now remove all the trivial relators from R, also all the relators of R which have other relators of R as subwords; we continue to call the resultant set R.

We note that B(0) is just the singleton $\{1_G\}$. Let us now suppose that B(n−1) has been constructed, we then proceed, to construct B(n) from B(n−1), as follows:-

*Step 1.* For each vertex, g, of B(n−1), and for each $c \in C$, if there is not already a directed edge labelled by c between g and gc in B(n−1), then add one.

*Step 2.* If ρ is a path labelled by a relator of R then identify its endpoints.

*Step 3.* For each pair of vertices g and $\bar{g}$, and for each label c, identify all the edges between g and $\bar{g}$ which are labelled by c.

The resultant graph is B(n).

$\boxed{5.1.3}$

*Comment.* As the set R is finite, so it may be included as a finite set of data in the method of constructing $\Gamma_C(G)$. However, Cannon mentions, in his preprint, that it is an open problem as to whether, knowing that $\Gamma_C(G)$ is almost convex, the finite set R can be computed (although it is difficult to believe that this would not, in practice, always be possible).

It is another, more interesting, but perhaps more difficult, open problem as to whether or not the property of almost convexity is dependant of the generating set. We do, however, have the following partial result.

*5.1.4 Proposition*

Let $C_1$ and $C_2$ be generating sets of G and suppose that, for all $g \in G$, the difference between $\|g\|_{C_1}$ and $\|g\|_{C_2}$ is bounded. Then, if $\Gamma_{C_1}(G)$ is almost convex, so also is $\Gamma_{C_2}(G)$.

*Proof:*

Let us suppose that:

$$\textit{(1)}\ \big|\, \|g\|_{C_1} - \|g\|_{C_2} \big| \text{ is bounded by } \kappa \text{ for all } g \in G.$$

By theorem 5.1.2, we need only prove that $\Gamma_{C_2}(G)$ is almost convex(2). So suppose $g_0$ and $g_1$ both lie in some n shell of $\Gamma_{C_1}(G)$ with:

$$\textit{(2)}\ d_{C_2}(g_0, g_1) \le 2.$$

There will be no loss of generality in assuming $n \ge 4\kappa+2$.

We now choose $\bar{g}_0 \in G$ so as to lie on a geodesic path of $\Gamma_{C_2}(G)$ between $g_0$ and the basepoint and so that:

$$\textit{(3)}\ d_{C_2}(g_0 , \bar{g}_0) = 2\kappa+2,$$

then we will also have:-

$$\textit{(4)}\ d_{C_2}(1_G , \bar{g}_0) = n-(2\kappa+2).$$

*5.1.5 Lemma.*

If $g \in G$ with $\|g\|_{C_1} = \|\bar{g}_0\|_{C_1}$, then $n-(4\kappa+2) \le d_{C_2}(1_G, g)$.

*Proof:*

If $\|g\|_{C_1} = \|\bar{g}_0\|_{C_1}$, we would have:-

$$\big|\, \|\bar{g}_0\|_{C_2} - \|g\|_{C_2} \big| \le \big|\, \|\bar{g}_0\|_{C_2} - |\tilde{g}_0\|_{C_1} \big| + \big|\, \|g\|_{C_1} - \|g\|_{C_2} \big|,$$

which, by (1),

$$\le 2\kappa.$$

Whence:-

$$d_{C_2}(1_G, g) = \|g\|_{C_2} \ge \|\bar{g}_0\|_{C_2} - 2\kappa$$
$$= d_{C_2}(1_G , \bar{g}_0) - 2\kappa,$$

which, by (4),

$$= n-(2\kappa+2)-2\kappa$$

i.e.,

$$d_{C_2}(1_G, g) \ge n-(4\kappa+2).$$

$\boxed{5.1.5}$

By lemma 5.1.5, if g is any element of G satisfying $\|g\|_{C_1} = \|\bar{g}_0\|_{C_1}$, then we must also have $d_{C_2}(1_G,g) \geq n-(4\kappa+2)$. As we are, anyway, assuming $n\geq(4\kappa+2)$, so it is possible to choose $\bar{g}_1$ so that $\|\bar{g}_1\|_{C_1}=\|\tilde{g}_0\|_{C_1}$, and with $\bar{g}_1$ lying at a distance $\leq 4\kappa+2$ from $g_1$ along a geodesic path of $\Gamma_{C_2}(G)$ between $g_1$ and the basepoint. We would have:-

$$\textit{(6)}\ \ d_{C_2}(g_1\,,\bar{g}_1\,)\leq 4\kappa+2,$$

$$\textit{(7)}\ \ \|\bar{g}_1\|_{C_1} = \|\tilde{g}_0\|_{C_1}.$$

We now have:-

$$d_{C_2}(\bar{g}_0,\bar{g}_1) \leq d_{C_2}(\tilde{g}_0,g_0)+d_{C_2}(g_0,g_1)+d_{C_2}(g_1,\tilde{g}_1),$$

which, by (3), (2) and (6),

$$\leq (2\kappa+2)+2+(4\kappa+2),$$

Whence:-

$$d_{C_1}(\tilde{g}_0,\tilde{g}_1) = \|\,\tilde{g}_0\tilde{g}_1{}^{-1}\|_{C_1},$$

which, by (1),

$$\leq \|\,\tilde{g}_0\tilde{g}_1{}^{-1}\|_{C_2}+\kappa$$
$$= d_{C_2}(\tilde{g}_0,\tilde{g}_1)+\kappa$$
$$\leq (6\kappa+6)+\kappa.$$

We are assuming $\Gamma_{C_1}(G)$ to be almost convex, so, with (7) and $d_{C_1}(\bar{g}_0,\bar{g}_1) \leq 7\kappa+6$, we see that $\bar{g}_0$ and $\bar{g}_1$ are joined by a path, ρ say, (of $\Gamma_{C_1}(G)$) of bounded length which stays within the $\|\bar{g}_0\|_{C_1}$-ball of $\Gamma_{C_1}(G)$. As each vertex on ρ belongs to the $(\|\bar{g}_0\|_{C_2}+\kappa)$-ball of $\Gamma_{C_2}(G)$, so each pair of adjacent vertices on ρ may be joined by a path (of $\Gamma_{C_2}(G)$), of length no more than $\kappa+1$, which must stay within the $(\|\bar{g}_0\|_{C_2}+2\kappa+1)$-ball of $\Gamma_{C_2}(G)$. By (4), $\|\bar{g}_0\|_{C_2}+2\kappa+1=n-1$ , whence, we have found a path of $\Gamma_{C_2}(G)$, which stays within the n ball of $\Gamma_{C_2}(G)$, has length no more than $|\rho|(\kappa+1)$ (which is bounded), and which joins $\bar{g}_0$ to $\bar{g}_1$. By the choice of $\bar{g}_0$ and $\bar{g}_1$, we have shown that there is a (composite) path joining $g_0$ to $g_1$, which stays within the n-ball of $\Gamma_{C_2}(G)$, and has length at most $(2\kappa+2)+|\rho|(\kappa+1)+(4\kappa+2)$.

$\boxed{5.1.4}$

## *5.2*

## *wlp-complete groups of type $P_r$ and llb-automatic groups are almost convex.*

Let C be any alphabet, then we say that a total ordering, <, of $C^*$ is a *word length preserving* ordering (or *wlp* ordering) if, for all $w, v \in C^*$, $|w|>|v| \Rightarrow w>v$. If $\langle C \mid \mathcal{R} \rangle$ is a complete presentation with respect to some wlp ordering, then it said to be a *wlp-complete* presentation.

The most familiar wlp orderings must surely be the ShortLex orderings, but there are others. As an example, suppose < is any ShortLex ordering of C and $c \in C$. We then define the ordering $\tilde{<}$ by: $v \,\tilde{<}\, w$ if $|v|<|w|$, or $|v| = |w|$ and $\tilde{v} < \tilde{w}$ where $\tilde{v}$ and $\tilde{w}$ are,respectively, the words $v$ and $w$ after all occurrences of c have been replaced by $\varepsilon$. Then $\tilde{<}$ is a (non-ShortLex) wlp Knuth-Bendix ordering which (despite appearances) is not just of theoretical interest (similar orderings are used in the computer program of (Hayashi)).

Let (G,C) be an automatic group with word acceptor W, and let $\gamma : C^* \longrightarrow G$ be the natural homomorphism. Then we say that (G,C) is *least length bounded automatic* (or *llb-automatic*) with respect to W, if $|\ell|-\|\gamma(\ell)\|_C$ is bounded (independently of $\ell$) for all $\ell \in \mathrm{lan}(W)$. We stress that the property of a group being *llb-automatic is* usually *dependant on the generating set* (while the property of a group being automatic is an invariant of the generating set).

In this section we will prove that the groups defined by parameterized wlp-complete presentations are almost convex (i.e. proposition 5.2.1), and that the llb-automatic groups are almost convex (i.e. theorem 5.2.9). Theorem 5.2.9 is due to (CHEPT) and is included, not just because it is the analog of 5.2.1 for automatic groups, but also because it is pivotal to our proof of theorem 5.4.1, i.e., that abelian by finite groups are almost convex.

*5.2.1 Proposition.*

Let $\mathcal{P} = \langle C \mid \mathcal{R} \rangle$ be a wlp-complete presentation of type $P_r$ of the group G (with respect to the wlp-ordering <), and let $\gamma : C^* \longrightarrow G$ be the natural homomorphism. Then $\Gamma_{\gamma(C)}(G)$ is almost convex.

*Proof:*

Let us begin by restating some of the definitions and terminology of pp 17-20 and, hopefully, in so doing, restate the suppositions in a more manageable form.

We defined the relation $\rightarrow_{\mathcal{R}}$ on the words of $C^*$ by $w \rightarrow_{\mathcal{R}} v$ if $w \equiv p\,b\,s$ and $v \equiv p\,a\,s$ for some $(b, a) \in \mathcal{R}$. Then we write $\rightarrow_{\mathcal{R}}^*$ for the reflexive, transitive closure of $\rightarrow_{\mathcal{R}}$. If $w \rightarrow_{\mathcal{R}}^* v$, then $v$ is an $\mathcal{R}$-descendant of $w$. A word is said to be $\mathcal{R}$-irreducible if it has no $\mathcal{R}$-descendants other than itself.

The set of rules, $\mathcal{R}$, will, by definition, be normalized, i.e., whenever $(b, a) \in \mathcal{R}$, then $b > a$. Whence $w \rightarrow_{\mathcal{R}}^* v \Rightarrow w \geq v$, also:-

$$(1) \quad (b, a) \in \mathcal{R} \Rightarrow |b| \geq |a|,$$

because $(b, a) \in \mathcal{R} \Rightarrow b > a$ and thus, as $<$ is wlp, we could not have $|a| > |b|$.

By $\mathcal{R}$ being complete we mean that each word, $w$, of $C^*$ has a unique irreducible descendant, called its $\mathcal{R}$-representative and denoted by rep($w$), which is the $<$ least word in its ( $\mathcal{R}$ ) congruence class. Thus, any irreducible word, $w$, will be a word of minimal length in its ( $\mathcal{R}$ ) congruence class (because if there were a strictly shorter word, then, as $<$ is wlp, this word would be $< w$ (which would contradict $\mathcal{R}$being complete)). We may interpret this geometrically as follows: any path of $\Gamma_{\gamma(C)}(G)$ labelled by an irreducible word will be a geodesic path. This, together with (1), is the crux of the supposition that $<$ is wlp.

If $\mathcal{B} \in (C^*)^{(2p+1)}$ (for some $p \in \mathbb{N}$), then $\mathcal{B}_i$ $(1 \leq i \leq 2p+1)$ is the i th component of $\mathcal{B}$. We defined $\mathcal{B}(0)$ to be the word

$$\mathcal{B}_1\, \mathcal{B}_3\, \mathcal{B}_5 \ldots \mathcal{B}_{(2p+1)},$$

and, if $n = (n_1, n_2, \ldots, n_p) \in \mathbb{N}^p$, $\mathcal{B}(n)$ is the word

$$\mathcal{B}_1 (\mathcal{B}_2)^{n_1}\, \mathcal{B}_3 (\mathcal{B}_4)^{n_2}\, \mathcal{B}_5 \ldots \mathcal{B}_{(2p-1)} (\mathcal{B}_p)^{n_p}\, \mathcal{B}_{(2p+1)}.$$

If $p > 0$, then we refer to the words $\mathcal{B}_2, \mathcal{B}_4, \ldots, \mathcal{B}_{2p}$ as the repeating factors of $\mathcal{B}$. We adopted the convention that $\mathbb{N}^0 = \{0\}$ so, if $\mathcal{B} \in (C^*)^0$ and $n \in \mathbb{N}^0$, then $\mathcal{B}(n)$ is *always* the single word $\mathcal{B}_1$.

By $\mathcal{R}$ being of type $P_r$ we mean that $\mathcal{R}$ can be partitioned as a *finite* number of subsets, of the form:

$$(2) \quad \{ (\mathcal{B}(n), \mathcal{A}(n)) \mid n \in \mathbb{N}^p \} \text{ with } 0 \leq p \leq r \text{ and } \mathcal{B}, \mathcal{A} \in (C^*)^{(2p+1)}.$$

We stressed that the p of (1) need *not* be the same for the different subsets of the partition, but r, being a bound on the p's, *is* a bound on the number of repeating factors allowed for the *different* $\mathcal{A}$'s and $\mathcal{B}$'s. Also, note that the subsets of type $P_0$ will just be the finite subsets (of $C^{\bullet}\times C^{\bullet}$).

Now let us suppose that some some $\mathcal{B}$ of (2) has a repeating factor, $\mathcal{B}_{2\kappa}$ say, which is the empty word. Then $\mathcal{A}_{2\kappa}$ must also be the empty word (otherwise $|\mathcal{A}_{2\kappa}|>0 \Rightarrow$

$$|\mathcal{B}_1\mathcal{B}_3\mathcal{B}_5 \cdots \mathcal{B}_{(2\kappa-1)}(\mathcal{B}_\kappa)^n \mathcal{B}_{(2\kappa+1)}\mathcal{B}_{(2\kappa+3)} \cdots \mathcal{B}_{(2p+1)}| <$$
$$|\mathcal{A}_1\mathcal{A}_3\mathcal{A}_5 \cdots \mathcal{A}_{(2\kappa-1)}(\mathcal{A}_\kappa)^n \mathcal{A}_{(2\kappa+1)}\mathcal{A}_{(2\kappa+3)} \cdots \mathcal{A}_{(2p+1)}|,$$

for some sufficiently large n – contradicting (1)). So, we could replace $\mathcal{B}$ by $(\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{(2\kappa-1)}, \mathcal{B}_{(2\kappa+1)}, \ldots, \mathcal{B}_{(2p+1)})$, $\mathcal{A}$ by $(\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{(2\kappa-1)}, \mathcal{A}_{(2\kappa+1)}, \ldots, \mathcal{A}_{(2p+1)})$ and p by p−1 (without changing the rules of (2)). Thus, repeating, if necessary, we may as well assume that $\mathcal{B}$ has no empty repeating factors. Henceforth we shall assume the $\mathcal{A}$'s and $\mathcal{B}$'s of (2) to be fixed.

We hope that the theory summarized hitherto will now be familiar to the reader, it will be referred to (mostly implicitly) throughout the (short) proof proper.

Now let $\{ (\mathcal{B}(n), \mathcal{A}(n)) \mid n \in \mathbb{N}^p \}$ be as in (2). If $p>0$ then, as $\mathcal{B}$ has no empty repeating factors, we may define $n_{(\mathcal{B})}$ to be the least integer such that $n_{(\mathcal{B})}\min(|\mathcal{B}_{2i}|)_{1\le i\le p}>2|\mathcal{B}(0)|$. We would then put $L_{(\mathcal{B})}=n_{(\mathcal{B})}\max(|\mathcal{B}_{2i}|)_{1<i<p}+|\mathcal{B}(0)|$, and $m_{(\mathcal{B})}=\max(6|\mathcal{B}(0)|, 2L)$. If $p=0$, then we would put $m_{(\mathcal{B})}=6|\mathcal{B}(0)|$. Finally, we define $M=\max(m_{(\mathcal{B})})_{\mathcal{B}\text{ as in }(2)}$.

By theorem 5.1.2, we know that $\Gamma_{\gamma(C)}(G)$ will be almost convex if we can prove that it is ac(2) (we refer the reader to definition 5.1.1). We can be precise, we shall prove that, for any $g\in G$ and $c_0, c_1 \in C$:

*(i)* whenever $\|g\| = \|g\gamma(c_0)\|$, then $g \;{}_{\text{join}(2|C|M)}\; g\gamma(c_0)$.

and

*(ii)* whenever $\|g\| = \|g\gamma(c_0c_1)\|$, then $g \;{}_{\text{join}(2|C|M)}\; g\gamma(c_0c_1)$.

The body of the proof is the following lemma.

*5.2.2 Lemma.*

Let $g\in G$ and $c\in C\cup\{\varepsilon\}$ be such that $g\gamma(c)\neq 1$, so that $\bar{g}=\bar{g}(g,c)\in G$ can be defined by $\text{rep}(\bar{g})\bar{c}=\text{rep}(g\gamma(c))$ for some $\tilde{c}\in C$, then $g \;{}_{\text{join}(|C|M)}\; \tilde{g}$.

Before proving 5.2.2, though, let us demonstrate how 5.2.2 implies (i) and (ii).

*5.2.3 Corollary.*

Let $g \in G$, $c_0, c_1 \in C \cup \{\varepsilon\}$ and suppose $\|g\| = \|g\gamma(c_0c_1)\|$, then $g \;{}_{\mathrm{join}(2|C|M)}\; g\gamma(c_0c_1)$. (Note that (i) would follow by taking $c_1 \equiv \varepsilon$, and so $\Gamma_{\gamma(C)}(G)$ would be ac(2).)

*Proof:*

We can certainly assume that $\|g\gamma(c_0)\| \geq 1$ (otherwise $g\gamma(c_0) = 1$ and, trivially, $g \;{}_{\mathrm{join}(2)}\; g\gamma(c_0c_1)$). Thus, we may define $\bar{g} \in G$ by:

$$\mathrm{rep}(\bar{g})\bar{c} \equiv \mathrm{rep}(g\gamma(c_0)) \equiv \mathrm{rep}((g\gamma(c_0c_1))\gamma(c_1^{-1})) \;(>\varepsilon)$$

for some $\bar{c} \in C$, so that, by 5.2.2, $g \;{}_{\mathrm{join}(|C|M)}\; \bar{g}$ and $g\gamma(c_0c_1) \;{}_{\mathrm{join}(|C|M)}\; \bar{g}$. As $\|g\| = \|g\gamma(c_0c_1\|$, so $g \;{}_{\mathrm{join}(2|C|M)}\; g\gamma(c\bar{c})$.

$\boxed{5.2.3}$

*Proof of 5.2.2:*

We will construct, inductively, a finite sequence

$$(g_i, c_i)$$

where $(g_0, c_0) = (g, c)$, and, if $i \geq 1$:

$$\left\{\begin{array}{c} (g_i, c_i) \in G \times C, \\ \mathrm{rep}(g_{(i-1)})c_{(i-1)} \overset{*}{\rightarrow} \mathrm{rep}(g_i)c_i \text{ and } \mathrm{rep}(g_{(i-1)})c_{(i-1)} > \mathrm{rep}(g_i)c_i, \\ g \;{}_{\mathrm{join}(iM)}\; g_i. \end{array}\right.$$

So we assume $(g_j, c_j)$ has been defined for $0 \leq j \leq i$. If $\mathrm{rep}(g_i)c_i$ is irreducible, we stop. If $\mathrm{rep}(g_i)c_i$ is not irreducible, then as $\mathrm{rep}(g_i)$ is irreducible, $c_i \not\equiv \varepsilon$ and we may choose $(\mathcal{B}(n), \mathcal{A}(n)) \in \mathcal{R}$, for some $\mathcal{B}, \mathcal{A} \in (C^*)^{(2p+1)}$ and $n \in \mathbb{N}^p$ as in (2), so that:

$$\textit{(3)}\;\; \mathrm{rep}(g_i)c_i \equiv p\mathcal{B}(n)$$

for some proper prefix $p$ of $\mathrm{rep}(g_i)$. We then define $c_{(i+1)} \in C$ by:

$$\textit{(4)}\;\; p\mathcal{A}(n) \equiv \bar{p}\, c_{(i+1)}$$

(with $\tilde{p}$ being the largest proper prefix of $p\mathcal{A}(n)$), and put $g_{(i+1)}=\gamma(\tilde{p})$. Whence:-

$$(5)\quad \mathrm{rep}(g_i)c_i \equiv p\mathcal{B}(n) \rightarrow_{\mathcal{R}} p\mathcal{A}(n) \equiv \tilde{p}\,c_{(i+1)} \rightarrow_{\mathcal{R}}^{*} \mathrm{rep}(g_{(i+1)})\,c_{(i+1)},$$

and so, as $\mathcal{B}(n)>\mathcal{A}(n)$,

$$\mathrm{rep}(g_i)c_i \equiv p\mathcal{B}(n) > p\mathcal{A}(n) \equiv \tilde{p}\,c_{(i+1)} \geq \mathrm{rep}(g_{(i+1)})\,c_{(i+1)}.$$

So we now have to prove that $g \;{}_{\mathrm{join}((i+1)M)}\; g_{(i+1)}$, but, by the inductive hypothesis, we already have $g \;{}_{\mathrm{join}(iM)}\; g_i$, thus it suffices to prove that $g_i \;{}_{\mathrm{join}(M)}\; g_{(i+1)}$. We will, for the most part, demonstrate this pictorially.

Let us first suppose that $\mathcal{A}(n)\equiv\varepsilon$. Then, by (5),

$$g_i\gamma(c_i)=\gamma(p),$$

whence:-

$$\|g_i\gamma(c_i)\| \leq |p|$$

Thus, as $p$ is a proper prefix of $\mathrm{rep}(g_i)$,

$$\|g_i\gamma(c_i)\| \leq |p| < \|g_i\|,$$

but, also by (5), we have $g_i\gamma(c_i)=g_{(i+1)}\gamma(c_{(i+1)})$, so, trivially, $g_i \;{}_{\mathrm{join}(2)}\; g_{(i+1)}$.

We may now assume $\mathcal{A}(n)\not\equiv\varepsilon$ so that, by (3) and (4) respectively, $c_i$ is the last character of the word $\mathcal{B}(n)$, and $c_{(i+1)}$ is the last character of the word $\mathcal{A}(n)$. Let us illustrate the reductions (5) as part of the Cayley graph $\Gamma_{\gamma(C)}(G)$.

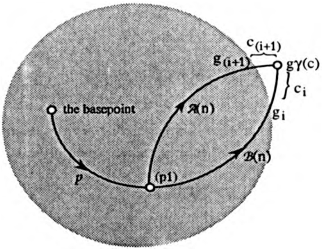


figure 1.

*5.2.4 Lemma.*

With reference to figure 1, all points within the shaded area can be supposed to lie within the ball $B(\|g_i\|)$.

*Proof:*

By (3) we have $\mathrm{rep}(g_i)c_i \equiv p\mathcal{B}(n)$, thus the path between the basepoint, (p1) and $g_i$, labelled by the irreducible word $\mathrm{rep}(g_i) \equiv p\,(\mathcal{B}(n)(1,|\mathcal{B}(n)|-1))$, is a geodesic path and so (most certainly) stays within $B(\|g_i\|)$. Also, by (4), $p\mathcal{B}(n) > p\mathcal{A}(n)$, thus $|p\mathcal{B}(n)| \geq |p\mathcal{A}(n)|$, and so $\|g_i\| = |p\mathcal{B}(n)| - 1 \geq |p\mathcal{A}(n)| - 1$. Whence the path between the basepoint, (p1) and $g_{(i+1)}$, labelled by $p(\mathcal{A}(n)(1,|\mathcal{A}(n)|-1))$ (and of length $|p\mathcal{A}(n)|-1$), could not possibly go outside the ball $B(\|g_i\|)$.

$\boxed{5.2.4}$

*5.2.5 Corollary.*

If $|\mathcal{B}(n)| \leq 3|\mathcal{B}(0)$, then $g_i\ {}_{\mathrm{join}(M)}\ g_{(i+1)}$.

*Proof:*

By 5.2.4, the path between $g_i$, (p1) and $g_{(i+1)}$, labelled by

$$(\mathcal{B}(n)(1,|\mathcal{B}(n)|-1))\,(\mathcal{A}(n)(1,|\mathcal{A}(n)|-1))^{-1}$$

stays within $B(\|g_i\|)$. Also, this path has length $< |\mathcal{B}(n)|+|\mathcal{A}(n)|$, but $|\mathcal{A}(n)| \leq |\mathcal{B}(n)|$, so

$$|\mathcal{B}(n)|+|\mathcal{A}(n)| \leq 2|\mathcal{B}(n)| \leq 6|\mathcal{B}(0)<M,$$

whence $g_i\ {}_{\mathrm{join}(M)}\ g_{(i+1)}$ (as required).

$\boxed{5.2.5}$

We can now assume that $|\mathcal{B}(n)| > 3|\mathcal{B}(0)|$, and so, with $\mathcal{B}, \mathcal{A} \in (C^*)^{(2p+1)}$ and $n \in \mathbb{N}^p$ as in (2), it must be that $p>1$ (because $p=0 \Rightarrow n=0$ by convention). Let $n=(n_1,n_2,\ldots,n_p)$, then:

$$\mathcal{B}(n) \equiv \mathcal{B}_1(\mathcal{B}_2)^{n_1}\,\mathcal{B}_3(\mathcal{B}_4)^{n_2}\,\mathcal{B}_5 \ldots \mathcal{B}_{(2p-1)}(\mathcal{B}_p)^{n_p}\,\mathcal{B}_{(2p+1)},$$

and

$$\mathcal{B}(0) \equiv \mathcal{B}_1\mathcal{B}_3\mathcal{B}_5 \ldots \mathcal{B}_{(2p-1)}\mathcal{B}_{(2p+1)}.$$

As $|\mathcal{B}(n)| > 3|\mathcal{B}(0)|$, we see that $|(\mathcal{B}_2)^{n_1}\,(\mathcal{B}_4)^{n_2} \ldots (\mathcal{B}_p)^{n_p}| > 2|\mathcal{B}(0)|$, whence we may choose $1 \leq s \leq p$ and then $0 \leq n_s$ so that:

$$(\mathcal{B}_{2s})^{\tilde{n}_s}\,\mathcal{B}_{(2s+1)}(\mathcal{B}_{(2(s+1))})^{n_{(s+1)}}\,\mathcal{B}_{(2s+3)}\cdots\mathcal{B}_{(2p-1)}(\mathcal{B}_p)^{n_p}\,\mathcal{B}_{(2p+1)}$$

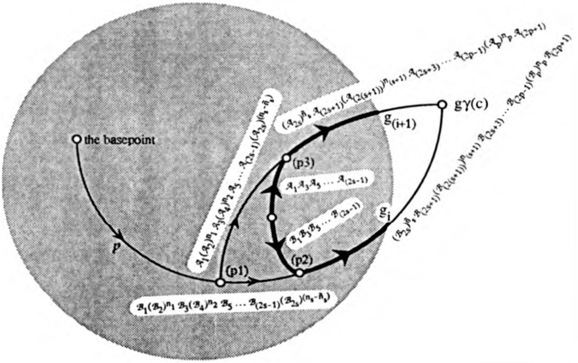
is the shortest such suffix of $\mathcal{B}(n)$ to have length $> 2|\mathcal{B}(0)|$. Thus, because $|\mathcal{A}(n)| \leq |\mathcal{B}(n)|$, we will have:–

$$(6)\ \ |(\mathcal{B}_{2s})^{\tilde{n}_s}\,\mathcal{B}_{(2s+1)}(\mathcal{B}_{(2s+2)})^{n_{(s+1)}}\,\mathcal{B}_{(2s+3)}\cdots\mathcal{B}_{(2p-1)}(\mathcal{B}_p)^{n_p}\,\mathcal{B}_{(2p+1)}| > |\mathcal{B}(0)| + |\mathcal{A}(0)|.$$

Also, (with the abbreviation $0^{(s-1)}$ for $s-1$ 0's), the rule

$$(7)\ \ (\ \mathcal{B}(0^{(s-1)}, \bar{n}_s, n_{(s+1)}, n_{(s+2)},\ldots,n_p)\ ,\ \mathcal{A}(0^{(s-1)}, \tilde{n}_s, n_{(s+1)}, n_{(s+2)},\ldots,n_p)\ )$$

belongs to $\mathcal{R}$ (because all rules of $\{\ (\ \mathcal{B}(n), \mathcal{A}(n)\ )\ |\ n\in\mathbb{N}^p\ \}$ belong to $\mathcal{R}$). Let us see how rule (7) fits into our picture (figure 1) of $\Gamma_{\gamma(C)}(G)$.



the basepoint
$p$
(p1)
(p2)
(p3)
$g_{(i+1)}$
$g_i$
$g\gamma(c)$
$\mathcal{A}_1(\mathcal{A}_2)^{n_1}\mathcal{A}_3(\mathcal{A}_4)^{n_2}\mathcal{A}_5\ldots\mathcal{A}_{(2s-1)}(\mathcal{A}_{2s})^{(n_s-\tilde{n}_s)}$
$\mathcal{B}_1(\mathcal{B}_2)^{n_1}\mathcal{B}_3(\mathcal{B}_4)^{n_2}\mathcal{B}_5\ldots\mathcal{B}_{(2s-1)}(\mathcal{B}_{2s})^{(n_s-\tilde{n}_s)}$
$\mathcal{A}_1\mathcal{A}_3\mathcal{A}_5\ldots\mathcal{A}_{(2s-1)}$
$\mathcal{B}_1\mathcal{B}_3\mathcal{B}_5\ldots\mathcal{B}_{(2s-1)}$
$(\mathcal{A}_{2s})^{\tilde{n}_s}\mathcal{A}_{(2s+1)}(\mathcal{A}_{(2(s+1))})^{n_{(s+1)}}\mathcal{A}_{(2s+3)}\ldots\mathcal{A}_{(2p-1)}(\mathcal{A}_p)^{n_p}\mathcal{A}_{(2p+1)}$
$(\mathcal{B}_{2s})^{\tilde{n}_s}\mathcal{B}_{(2s+1)}(\mathcal{B}_{(2(s+1))})^{n_{(s+1)}}\mathcal{B}_{(2s+3)}\ldots\mathcal{B}_{(2p-1)}(\mathcal{B}_p)^{n_p}\mathcal{B}_{(2p+1)}$

figure 2.

*5.2.6 Lemma.*

With reference to figure 2, all points within the shaded area can be supposed to lie within the ball $B(\|g_i\|)$.

*Proof:*

By lemma 5.2.4, we know that all points within the shaded area except, conceivably, those of the path between (p2) and (p3) labelled by

$$(8)\ \ (\mathcal{B}_1\mathcal{B}_3\mathcal{B}_5 \ldots \mathcal{B}_{(2s-1)})(\mathcal{A}_1\mathcal{A}_3\mathcal{A}_5 \ldots \mathcal{A}_{(2s-1)})^{-1}$$

belong to $B(\|g_i\|)$), but this path (trivially) stays within $B(\|g_i\|)$ (as we now demonstrate).

As, the path between the basepoint, (p1), (p2) and $g_i$ labelled by $rep(g_i)$= $p(\mathcal{B}(n)(1,|\mathcal{B}(n)|-1))$ ), is a geodesic path, so the point (p2) is at a distance no more than

$$\|g_i\| - |(\mathcal{B}_{2s})^{\bar{n}_s}\, \mathcal{B}_{(2s+1)}(\mathcal{B}_{(2(s+1))})^{n_{(s+1)}}\, \mathcal{B}_{(2s+3)} \cdots \mathcal{B}_{(2p-1)}(\mathcal{B}_p)^{n_p}\, \mathcal{B}_{(2p+1)}|+1$$

from the basepoint. Thus, by (6), we see that the point (p2) lies at a distance of at most $\|g_i\| - |\mathcal{B}(0)| - |\mathcal{A}(0)|$ from the basepoint. Whence the path beginning at (p2) and labelled by (8) could not possibly go outside the ball $B(\|g_i\|)$ (as required).

$\boxed{5.2.6}$

*5.2.7 Corollary.*

If $|\mathcal{B}(n)| > 3|\mathcal{B}(0)$, then $g_i\ {}_{join(M)}\ g_{(i+1)}$.

*Proof:*

With reference to figure 2 and lemma 5.2.6, we see that the bold path between $g_i$ and $g_{(i+1)}$ stays within $B(\|g_i\|)$ and has length <

$$|\mathcal{B}(0(s-1), \bar{n}_s, n_{(s+1)}, n_{(s+2)}, \ldots, n_p)| + |\mathcal{A}(0(s-1), \bar{n}_s, n_{(s+1)}, n_{(s+2)}, \ldots, n_p)|$$

(which, by (7))

$$\leq 2|\mathcal{B}(0(s-1), \bar{n}_s, n_{(s+1)}, n_{(s+2)}, \ldots, n_p)|\,.$$

Also, we remind the reader that we chose s and $\bar{n}_s$ so that:

$$(\mathcal{B}_{2s})^{\tilde{n}_s}\, \mathcal{B}_{(2s+1)}(\mathcal{B}_{(2s+2)})^{n_{(s+1)}}\, \mathcal{B}_{(2s+3)} \cdots \mathcal{B}_{(2p-1)}(\mathcal{B}_p)^{n_p}\, \mathcal{B}_{(2p+1)}$$

was the shortest such suffix of $\mathcal{B}(n)$ to have length $> 2|\mathcal{B}(0)|$. As $n_{(\mathcal{B})}$ was defined to be least such that $n_{(\mathcal{B})}\min(|\mathcal{B}_{2i}|)_{1<i\leq p} > 2|\mathcal{B}(0)|$, we must have:-

$$\tilde{n}_s + n_{(s+1)} + n_{(s+1)} + \ldots + n_p < n_{(\mathcal{B})}\,.$$

Whence:-

$$2|\mathcal{B}(0(s-1), \bar{n}_s, n_{(s+1)}, n_{(s+2)}, \ldots, n_p)| \leq 2(B(0) + n_{(\mathcal{B})}\max(|\mathcal{B}_{2i}|)_{1<i<p})$$

(which, by the definition of $m_{(\mathcal{B})}$ and M)

$$= m_{(\mathcal{B})} \leq M,$$

and so we have shown that $g_i\ {}_{join(M)}\ g_{(i+1)}$ (as required).

$\boxed{5.2.7}$

So we have constructed a sequence

$$(g_i, c_i)$$

where $(g_0,c_0)=(g,c)$, and, if $i\geq1$:

$$\begin{cases} (g_i,c_i)\in G\times C, \\ \mathrm{rep}(g_{(i-1)})c_{(i-1)} \xrightarrow{*} \mathrm{rep}(g_i)c_i \text{ and } \mathrm{rep}(g_{(i-1)})c_{(i-1)} > \mathrm{rep}(g_i)c_i\,, \\ g \;{}_{\mathrm{join}(iM)}\; g_i\,. \end{cases}$$

As $\mathrm{rep}(g_1)c_1 > \mathrm{rep}(g_2)c_2 > \ \ldots$ , so the sequence is finite. Actually, we stopped when $\mathrm{rep}(g_r)c_r$ was found to be irreducible, i.e., when $g_r = \bar{g}(g,c)$ (we refer the reader to the statement of lemma 5.2.2). We require $g\ {}_{\mathrm{join}(|C|M)}\ \bar{g}$, which is a trivial corollary of the next lemma together with the fact that $g\ {}_{\mathrm{join}(rM)}\ g_r$ (i.e. $g\ {}_{\mathrm{join}(rM)}\ \bar{g}$).

*5.2.8 Lemma.*

$r<|C|$.

*Proof.*

Suppose not, then there would be some i,j with $0\leq i<j\leq r$ and $c_i\equiv c_j$ . As $\mathrm{rep}(g_i)c_i \rightarrow_{\mathcal{R}}^{*} \mathrm{rep}(g_j)\,c_j$ , so $g_i\,\gamma(c_i)= g_j\,\gamma(c_j)$, and thus $g_i= g_j$ (because $\gamma(c_i)=\gamma(c_j)$). So we would have $\mathrm{rep}(g_i)\equiv\mathrm{rep}(g_j)$, whence $\mathrm{rep}(g_i)c_i\equiv\mathrm{rep}(g_j)c_j$ (because $c_i\equiv c_j$), but, anyway, we have $\mathrm{rep}(g_i)c_i > \mathrm{rep}(g_{i+1})c_{i+1} > \ldots > \mathrm{rep}(g_j)c_j$ (which is clearly a contradiction).
$\boxed{\text{5.2.8, 5.2.2 and 5.2.1}}$

We will now prove the analog of proposition 5.2.1 for the automatic groups.

*5.2.9 Theorem ( CEHPT ).*

If (G,C) is llb-automatic with word acceptor W, then $\Gamma_C(G)$ is almost convex.

*Proof:*

Let W accept the language $\mathcal{LW}$, then, as (G,C) is llb-automatic, there will be a b such that, for all $\ell \in \mathcal{LAN}$, $|\ell| - \|\gamma(\ell)\|_C \leq b$. Also, by corollary 2.1.4, we know that there is a $\Delta$ with the property that: whenever $\ell_0, \ell_1 \in \mathcal{LAN}$ with $\|\gamma(\ell_0\ell_1^{-1})\|_C \leq 2b+2$, then any two paths of $\Gamma_C(G)$, beginning at the same point, and labelled by $\ell_0$ and $\ell_1$, respectively, do not diverge by more than a distance $\Delta$.

By theorem 5.1.2, we need only prove that $\Gamma_C(G)$ is ac(2). So let us take any $n \in N$ and suppose $g_0, g_1 \in S(n)$ to be such that $d(g,h) \leq 2$. Now choose $\bar{g}_0 \in G$ so that $\bar{g}_0$ is at a distance b along some geodesic path between $g_0$ and the basepoint. Then, trivially:-

$$(1)\quad g_0 \;_{\text{join(b)}}\; \bar{g}_0 \text{ and } \bar{g}_0 \in S(n-b).$$

Also, we choose $\bar{g}_1 \in G$ with $\bar{g}_1$ being at a distance b along some geodesic path between $g_1$ and the basepoint, so that:-

$$(2)\quad g_1 \;_{\text{join(b)}}\; \tilde{g}_1 \text{ and } \bar{g}_1 \in S(n-b).$$

We shall prove that $\bar{g}_0 \;_{\text{join(b+4}\Delta)}\; \tilde{g}_1$ (because $g_0 \;_{\text{join(b)}}\; \bar{g}_0$, $\bar{g}_0 \;_{\text{join(b+4}\Delta)}\; \bar{g}_1$, $g_1 \;_{\text{join(b)}}\; \bar{g}_1$ and $\| g_0\| = \| g_1\| \Rightarrow g_0 \;_{\text{join(3b+4}\Delta)}\; g_1$, so $\Gamma_C(G)$ would be ac(2)). Note that we lose no generality by assuming $n \geq 2b+\Delta$.

Let $\ell_0, \ell_1 \in \mathcal{LAN}$ with $\gamma(\ell_0) = \tilde{g}_0$ and $\gamma(\ell_1) = \bar{g}_1$. Because $d(g,h) \leq 2$ and $g_0$ and $g_1$ were chosen with $d(g_0, \bar{g}_0) = b$ and $d(g_1, \bar{g}_1) = b$, so we have $d(\bar{g}_0, \bar{g}_1) \leq 2b+2$, i.e., $\|\bar{g}_0\bar{g}_1^{-1}\| \leq 2b+2$, or $\|\gamma(\ell_0)\gamma(\ell_0^{-1})\| \leq 2b+2$. Thus, with $\rho_0$ and $\rho_1$ being, respectively, the paths beginning at the basepoint and labelled by $\ell_0$ and $\ell_1$, we know that $\rho_0$ and $\rho_1$ do not diverge by more than a distance $\Delta$. Also, because $\ell_0 \in \mathcal{LAN}$ and $\gamma(\ell_0) = \bar{g}_0$, we have $|\ell_0| - \|\bar{g}_0\| \leq b$, i.e. $|\ell_0| \leq b + \|\bar{g}_0\|$, and so, by(1):-

$$(3)\quad |\ell_0| \leq n.$$

Thus the path $\rho_0$, which begins at the basepoint and has length $|\ell_0|$, does not go outside the ball B(n). Similarly, $\rho_1$ has length $\leq n$ and so does not go outside B(n).

*5.2.10 Lemma.*

If p is the point on $\rho_0$ at a distance r along $\rho_0^{-1}$ from $\bar{g}_0$ (respectively, if p is the point on $\rho_1$ at a distance r along $\rho_1^{-1}$ from $\bar{g}_1$), then $n-b-r \leq d(1_G, p) \leq n-r$.

***Proof:***

We chose $\bar{g}_0$ so that $d(1_G, \bar{g}_0) = n-b$, but p is at a distance r from $\bar{g}_0$ along a geodesic path between $\bar{g}_0$ and the basepoint, so $d(1_G, p) \geq d(1_G, \bar{g}_0) - r = (n-b) - r$.

Also, we traverse a distance r along $\rho_0^{-1}$ (respectively $\rho_1^{-1}$) to get to the point p. If p was still more than a distance n−r from the basepoint, then the length of $\rho_0$, i.e $|\zeta_0|$, (respectively the length of $\rho_1$) would certainly need to be > r+(n−r), i.e, >n; but $|\zeta_0|$>n would contradict (3). So p could not be more than a distance n−r from the basepoint, i.e. $d(1_G , p) \leq n-r$.

$\boxed{5.2.10}$

Now, we wish to prove that $\bar{g}_0$ is joined to $\bar{g}_1$ by a path of length at most b+4Δ which stays within the ball B(n). We begin at $\bar{g}_0$ and traverse $\rho_0^{-1}$ (within B(n)) for a distance Δ to arrive at the point $p_0$, say. Then, by lemma 5.2.10,

$$(4)\ \ n-b-\Delta \leq d(1_G , p_0) \leq n-\Delta.$$

As $\rho_0$ and $\rho_1$ do not diverge by more than a distance Δ, we may now a traverse a path, of length no more than Δ, to arrive at some point, $p_1$, say, on $\rho_1$. By (4), we see that this path will not go outside B(n), and that:-

$$(5)\ \ n-b-2\Delta \leq d(1_G , p_1) .$$

So far we have traversed, within B(n), a path of length at most 2Δ to arrive at the point $p_1$ on the path $\rho_1$. We now traverse $\rho_1^{-1}$ (within B(n)) between $p_1$ and $\bar{g}_1$, noting that this subpath could not have length > b+2Δ (without, by 5.2.10, $d(1_G , p_1) < n-(b+2\Delta)$, which would have contradicted (5)).

$\boxed{5.2.9}$

## *5.3*

## *The Group U(3,ℤ) and Almost Convexity*

In section 5.2 we proved that the parameterized complete groups (with word length preserving orderings) and the (least length bounded) automatic groups were subclasses of the groups possessing almost convex Cayley graphs. We believe these are strict subclasses, and this the subject of this section.

The group $U(3,\mathbb{Z})$ is the group of 3 by 3 (lower) unitriangular matrices over $\mathbb{Z}$. We will be working with the group G, isomorphic to $U(3,\mathbb{Z})$, which we define as the group $\mathbb{Z}^3$ with multiplication

$$\textit{(1)}\ \ (a,c,b)(\acute{a},\acute{c},\acute{b}) = (\,a+\acute{a}\,,\,c+b\acute{a}+\acute{c}\,,\,b+\acute{b}\,).$$

(A trivial calculation will confirm that the map between the groups G and $U(3,\mathbb{Z})$ defined by $(a,c,b) \mapsto \begin{bmatrix} 1\,0\,0 \\ a\,1\,0 \\ c\,b\,1 \end{bmatrix}$ is an isomorphism.)

We now put $x=(1,0,0)$, $y=(0,0,1)$ and $C=\{\,x\,,x^{-1}\,,y\,,y^{-1}\,\}$. Then, with G being isomorphic to $U(3,\mathbb{Z})$, the following facts will probably be familiar to the reader. The set C is a (minimal) generating set of G, which is a non-abelian, torsion free, nilpotent group of class 2.

By Theorem 18.1 of (CHEPT), we know that non-abelian, torsion free, nilpotent groups are not automatic, and, we *conjecture*, G has no complete presentation of type $P_r$ with respect to ShortLex orderings. We will, for the remainder of this section, be wholly concerned with the proof of the following theorem.

*5.3.1 Theorem.*

$\Gamma_C(G)$ is almost convex.

We begin the proof with some terminology. We put $z=(0,1,0)$, and note that z generates the centre of G. If $g\in G$, then $(g)_i$ will denote the $i^{th}$ component of the triple g. If p is a product of generators in C and $c\in C$, then $occ(c,p)$ will denote the number of occurrences of c in p. Recall that, if $g \in G$, then $\| g \|_C$ is the norm of g with respect to C, i.e., $\| g \|_C$ is the minimal number of generators needed to express g as a product of generators in C. Henceforth we will write $\| g \|$ for $\| g \|_C$ , the latter will only be needed towards the end of this section (in corollary 5.3.19).

Pivotal to our proof of theorem 5.3.1 is a method of calculating $\| g \|$ for arbitrary $g \in G$. The method we use is easily derived from lemmas 5.3.7 and 5.3.11 and is described immediately after the statement of lemma 5.3.11. To prove these lemmas, however, we will need to make a number of calculations, concerning products in the generators C, and prove some preliminary lemmas.

After proving lemmas 5.3.7 and 5.3.11, the proof of theorem 5.3.1 is purely mechanical. Basically, in propositions 5.3.16 and 5.3.18, we will exhibit a number of paths in $\Gamma_C(G)$ and then, by referring to these lemmas, confirm that these paths stay within specific n-balls. We begin with some simple calculations.

*5.3.2 Lemma.*

Let $p = y^{b_1} x^{a_1} y^{b_2} x^{a_2} \ldots y^{b_n} x^{a_n}$ , with all the $a_i , b_i \in \mathbb{Z}$, then:

$$p = ( a_1+a_2+\ldots+a_n \, , \sum_{i=1}^{i=n} a_i ( b_1+b_2+\ldots+b_i ) \, , b_1+b_2+\ldots+b_n ).$$

The proof is a trivial inductive argument which refers to (1) and the definition of x and y.

$\boxed{5.3.2}$

*5.3.3 Corollary.*

Suppose p is given as a product of the generators in C, then:

*(i)* $(p)_1 = occ(x,p) - occ(x^{-1},p)$,

*(ii)* $(p)_3 = occ(y,p) - occ(y^{-1},p)$,

*Proof:*

Suppose p to be as in lemma 5.3.2 and note that:

$$a_1+a_2+\ldots+a_n = occ(x,p)-occ(x^{-1},p)$$

and

$$b_1+b_2+\ldots+b_n = occ(y,p)-occ(y^{-1},p).$$

Then the result is an immediate corollary of lemma 5.3.2.

$\boxed{5.3.3}$

*5.3.4 Corollary.*

If $g \in G$, then the difference between $\| g \|$ and $(g)_1+(g)_3$ is even.

*Proof:*

By expressing g as any product, p, of generators in C, we would have:-

$$\begin{aligned} occ(x,p)-occ(x^{-1},p) &= (p)_1 \text{ ,by 5.3.3(i),} \\ &= (g)_1 \text{ ,} \end{aligned}$$

and

$$\begin{aligned} occ(y,p)-occ(y^{-1},p) &= (p)_3 \text{ , by 5.3.3(ii),} \\ &= (g)_3 \text{ .} \end{aligned}$$

Whence:-

$$occ(x,p)=occ(x^{-1},p)+(g)_1 \text{ ,}$$

and

$$occ(y,p)=occ(y^{-1},p)+(g)_3 \text{ .}$$

Now, we may choose p so as to contain precisely $\| g \|$ generators of C, i.e.,

$$occ(x,p)+occ(x^{-1},p)+occ(y,p)+occ(y^{-1},p) = \| g \| \text{ .}$$

Substituting for occ(x,p) and occ(y,p) in the latter expression yields:-

$$2\,occ(x^{-1},p) + (g)_1 + 2\,occ(y^{-1},p) + (g)_3 = \| g \|$$

(as required).

$\boxed{5.3.4}$

*5.3.5 Corollary.*

If $p \in G$ is a product involving just the generators x and y, then:

$$\| p \| = occ(x,p) + occ(y,p).$$

*Proof:*

By expressing p as any product, $\acute{p}$, of generators in C, we would have:-

$$\begin{aligned} occ(x,\acute{p})-occ(x^{-1},\acute{p}) &= (\acute{p})_1 \text{, by 5.3.3(i),} \\ &= (p)_1 \\ &= occ(x,p)-occ(x^{-1},p) \text{, by 5.3.3(i),} \\ &= occ(x,p). \\ occ(y,\acute{p})-occ(y^{-1},\acute{p}) &= (\acute{p})_3 \text{, by 5.3.3(ii),} \\ &= (p)_3 \\ &= occ(y,p)-occ(y^{-1},p) \text{, by 5.3.3(ii),} \\ &= occ(y,p). \end{aligned}$$

Whence:-

$$occ(x,\acute{p}) \geq occ(x,p) \text{ and } occ(y,\acute{p}) \geq occ(y,p).$$

So, whenever p is expressed as a product of generators in C, the product in question must contain a minimum of $occ(x,p)+occ(y,p)$ generators. As p is a product containing precisely $occ(x,p)+occ(y,p)$ generators, so $occ(x,p)+occ(y,p)$ is the minimal number of generators needed to express p as a product of generators in C, i.e.,
$\| p \| = occ(x,p)+occ(y,p)$.

$\boxed{5.3.5}$

*5.3.6 Lemma.*

Suppose we are given $p \in G$ as a product of generators in C. If $(p)_1, (p)_3 \geq 0$, then

$$(p)_2 \leq occ(x,p)\, occ(y,p).$$

*Proof:*

We take p to be the product:

$$p = y^{b_1} x^{a_1} y^{b_2} x^{a_2} \dots y^{b_n} x^{a_n}\text{, with all the } a_i\text{ , } b_i \in \mathbb{Z}.$$

Then we have:-

$$occ(x,p) = \sum_{\text{all } a_i \geq 0} a_i\ ,$$

$$occ(x^{-1},p) = -\sum_{\text{all } a_i \leq 0} a_i\ ,$$

$$occ(y,p) = \sum_{\text{all } b_i \geq 0} b_i\ ,$$

$$occ(y^{-1},p) = -\sum_{\text{all } b_i \leq 0} b_i\ ,$$

and, by lemma 5.3.2,

$$(p)_2 = \sum_{i=1}^{i=n} a_i\,(\,b_1+b_2+\dots+b_i\,).$$

Thus, we have to prove:

$$\sum_{i=1}^{i=n} a_i\,(\,b_1+b_2+\dots+b_i\,) \leq \left(\sum_{\text{all } a_i \geq 0} a_i\right)\left(\sum_{\text{all } b_i \geq 0} b_i\right).$$

We start by defining:

$$I+ = \{\ i \mid 1 \leq i \leq n,\ 0 < a_i \text{ and } 0 < b_1+b_2+\dots+b_i\ \},$$

$$I- = \{\ i \mid 1 \leq i \leq n,\ a_i < 0 \text{ and } b_1+b_2+\dots+b_i < 0\ \},$$

$$b_{max} = \max(b_1+b_2+\dots+b_i)_{i \in I+}\ ,$$

and

$$b_{min} = \min(b_1+b_2+\dots+b_i)_{i \in I-}\ .$$

By 5.3.3(ii),

$$(p)_3 = occ(y,p) - occ(y^{-1},p),$$

but we are assuming $0 \leq (p)_3$ , whence:-

$$\mathrm{occ}(y^{-1},p)\le\mathrm{occ}(y^{-1},p),$$

i.e.,

$$(2)\quad -\sum_{\text{all } b_i\le 0} b_i \le \sum_{\text{all } b_i\ge 0} b_i .$$

By 5.3.3(i),

$$(p)_1 = \mathrm{occ}(x,p)-\mathrm{occ}(x^{-1},p),$$

but we are assuming $0\le(p)_1$ , whence:-

$$\mathrm{occ}(x^{-1},p)\le\mathrm{occ}(x^{-1},p),$$

i.e.,

$$-\sum_{\text{all } a_i<0} a_i \le \sum_{\text{all } a_i>0} a_i ,$$

and from which we derive:-

$$(3)\quad b_{\min}\left(\sum_{\text{all } a_i\le 0} a_i\right) \le -b_{\min}\left(\sum_{\text{all } a_i>0} a_i\right).$$

Now, we have:-

$$\sum_{i=1}^{i=n} a_i(\,b_1+b_2+\ldots+b_i\,) \le \sum_{i\in I+} a_i(\,b_1+b_2+\ldots+b_i\,) + \sum_{i\in I-} a_i(\,b_1+b_2+\ldots+b_i\,),$$

by the definition of I+ and I-,

$$\le b_{\max}\left(\sum_{i\in I+} a_i\right) + b_{\min}\left(\sum_{i\in I-} a_i\right),$$

by the definition of $b_{\max}$ and $b_{\min}$ ,

$$\le b_{\max}\left(\sum_{\text{all } a_i\ge 0} a_i\right) + b_{\min}\left(\sum_{\text{all } a_i\le 0} a_i\right),$$

because $a_i\ge 0$ if $i\in I+$, and $a_i\le 0$ if $i\in I-$.

$$\leq b_{max}\left(\sum_{\text{all } a_i \geq 0} a_i\right) - b_{min}\left(\sum_{\text{all } a_i > 0} a_i\right)$$

by (3). Whence:-

$$(3)\ \sum_{i=1}^{i=n} a_i\,(b_1+b_2+\ldots+b_i) \leq (b_{max}-b_{min})\left(\sum_{\text{all } a_i \geq 0} a_i\right).$$

We need to prove the inequality:

$$\sum_{i=1}^{i=n} a_i\,(b_1+b_2+\ldots+b_i) \leq \left(\sum_{\text{all } a_i > 0} a_i\right)\left(\sum_{\text{all } b_i > 0} b_i\right).$$

and so it will now suffice to prove:

$$b_{max}-b_{min} \leq \left(\sum_{\text{all } b_i \geq 0} b_i\right).$$

By the definition of $b_{max}$ and $b_{min}$, we have:-

$$b_{max}=b_1+b_2+\ldots+b_r\text{, for some } r\in I+,$$

and

$$b_{min}=b_1+b_2+\ldots+b_s\text{, for some } s\in I-.$$

There are two possibilities.

*$r \geq s$:* then:-

$$b_{max}-b_{min} = b_{(s+1)}+\ldots+b_r \leq \left(\sum_{\text{all } b_i > 0} b_i\right)$$

(as required).

*$r \leq s$:* then:-

$$b_{max}-b_{min} = -(b_{(r+1)}+\ldots+b_s).$$

However:

$$b_{(r+1)}+\ldots+b_s \geq \sum_{\text{all } b_i<0} b_i \,,$$

thus:-

$$-(b_{(r+1)}+\ldots+b_s) \leq -\sum_{\text{all } b_i\leq 0} b_i \,.$$

Then, as (2) states that:

$$-\sum_{\text{all } b_i<0} b_i \leq \sum_{\text{all } b_i>0} b_i \,,$$

so we have:-

$$b_{max}-b_{min} = -(b_{(r+1)}+\ldots+b_s) \leq \sum_{\text{all } b_i\geq 0} b_i$$

(as required).

$\boxed{5.3.6}$

With a,b∈ ℕ, we now define six products.

$p_1(b) = y^{b}.$

$p_2(a,b,r,s) = x^{(a-r-1)}y^{s}xy^{(b-s)}x^{r},$

for all 0≤r,s such that r<a and s≤b.

$p_3(a,b,r,s) = x^{-(r+1)}y^{s}xy^{(b-s)}x^{(a+r)},$

for all 0≤r and 0<s such that a+r≤b and s≤b.

$p_4(a,b,r,s) = y^{(b+r)}x^{(a-s)}yx^{s}y^{-(r+1)},$

for all 0≤r and 0<s such that b+r≤a and s≤a.

$p_5(a,b,r,s) = x^{-(b+r-a)}y^{s}xy^{(b+r-s)}x^{(b+r-1)}y^{-r},$

for all 0<r,s such that a<b+r and s≤b+r.

$p_6(a,b,r,s) = x^{-(b+r+1-a)}y^{s}xy^{(b+r-s)}x^{(b+r)}y^{-r},$

for all 0<r,s such that a≤b+r and s≤b+r.

We first prove that $\| p_i(a,b,r,s)\|$ = number of generators occurring in $p_i(a,b,r,s)$, or, to be more formal:

*5.3.7 Lemma.*

*(i)* $\| p_1(b)\| = b$.

*(ii)* $\| p_2(a,b,r,s)\| = a+b$.

*(iii)* $\| p_3(a,b,r,s)\| = 2(r+1)+a+b$.

*(iv)* $\| p_4(a,b,r,s)\| = 2(r+1)+a+b$.

*(v)* $\| p_5(a,b,r,s)\| = 4r+3b-a$.

*(vi)* $\| p_6(a,b,r,s)\| = 4r+2+3b-a$.

*Proof:*

The proofs of (i) and (ii) are trivial corollaries of lemma 5.3.5 (i.e., if a product, p, involves just the generators x and y, then the number of these generators involved in p is the norm of p).

*Proof of (iii):*

We defined

$$p_3(a,b,r,s) = x^{-(r+1)}y^s x y^{(b-s)} x^{(a+r)},$$

for all $0 \leq r$ and $0 < s$ such that $a+r \leq b$ and $s \leq b$. By lemma 5.3.2, we calculate that

$$\textit{(1)}\ \ p_3(a,b,r,s) = (a\ ,\ s+b(a+r)\ ,\ b).$$

We aim to prove $\| p_3(a,b,r,s)\| = 2(r+1)+a+b$. Note, however, that $p_3(a,b,r,s)$ is already defined as a product of $2(r+1)+a+b$ generators, thus $\| p_3(a,b,r,s)\| \leq 2(r+1)+a+b$.

So, we may assume, for a contradiction, that $\| p_3(a,b,r,s)\| < 2(r+1)+a+b$. It would follow that there is a product p, say, (in the generators of C) such that $p=p_3(a,b,r,s)$, but containing strictly fewer than $2(r+1)+a+b$ generators. Whence:-

$$\textit{(2)}\ \ occ(x,p) + occ(x^{-1},p) + occ(y,p) + occ(y^{-1},p) < 2(r+1)+a+b.$$

Also, by (1) and lemma 5.3.3(i), we would have:-

$$(p)_1 = a = occ(x,p)-occ(x^{-1},p),$$

and, by (1) and lemma 5.3.3(ii), we have:-

$$(p)_3 = b = occ(y,p) - occ(y^{-1},p).$$

So, we may substitute

$$(3)\ \ occ(x,p) = occ(x^{-1},p) + a$$

and

$$(4)\ \ occ(y,p) = occ(y^{-1},p) + b$$

in (2) to derive:-

$$2occ(x^{-1},p) + 2occ(y^{-1},p) + a + b < 2(r+1) + a + b,$$

i.e.,

$$(5)\ \ occ(x^{-1},p) + occ(y^{-1},p) \leq r.$$

As (5) includes $occ(x^{-1},p) < r$, and we are assuming $0 \leq r \leq b-a$, so:-

$$(6)\ \ occ(x^{-1},p) + a \leq b.$$

By lemma 5.3.6, we know that, provided $(p)_3$ , $(p)_1 \geq 0$, then $(p)_2 \leq occ(y,p)\, occ(x,p)$. We have $(p)_3 = b \geq 0$ and $(p)_1 = a \geq 0$, therefore:-

$$(p)_2 = occ(y,p)\, occ(x,p),$$

by 5.3.6,

$$= (occ(y^{-1},p)+b)(occ(x^{-1},p)+a),$$

by (3) and (4),

$$= occ(y^{-1},p)(occ(x^{-1},p)+a) + b(occ(x^{-1},p)+a)$$

$$\leq occ(y^{-1},p)b + b(occ(x^{-1},p)+a),$$

by (6),

$$= b(\ occ(y^{-1},p)+occ(x^{-1},p)+a\ ).$$

So, by (5), we have $(p)_2 \leq b(a+r)$. This is the required contradiction because we were assuming $p = p_3(a,b,r,s)$, while, by (1), $(p_3(a,b,r,s))_2 = s+b(a+r)$ with $s>0$.

$\boxed{5.3.7(iii)}$

The proof of (iv), being similar, is omitted.

*Proof of (v):*

Recall that we defined

$$p_5(a,b,r,s) = x^{-(b+r-a)}y^s x y^{(b+r-s)} x^{(b+r-1)} y^{-r},$$

for all $0<r,s$ such that $a<b+r$ and $s\leq b+r$. By lemma 5.3.2, we can calculate

$$\textit{(7)}\ \ p_5(a,b,r,s) = (a\ ,\ s+(b+r)(b+r-1)\ \ ,\ b).$$

We aim to prove $\| p_5(a,b,r,s)\| = 4r+3b-a$. We note, however, that $p_5(a,b,r,s)$ is already defined as a product of $4r+3b-a$ generators, thus $\| p_5(a,b,r,s)\| \leq 4r+3b-a$.

So, we assume, for a contradiction, that $\| p_5(a,b,r,s)\| < 4r+3b-a$. It follows that there is a product p, say, (in the generators of C) such that $p=p_5(a,b,r,s)$, but containing strictly fewer than $4r+3b-a$ generators. Whence:-

$$\textit{(8)}\ \ occ(x,p) + occ(x^{-1},p) + occ(y,p) + occ(y^{-1},p) < 4r+3b-a.$$

Also, by (7) and lemma 5.3.3(i), we would have:-

$$(p)_1 = a = occ(x,p)-occ(x^{-1},p),$$

and, by (1) and lemma 5.3.3(ii), we have:-

$$(p)_3 = b = occ(y,p)-occ(y^{-1},p).$$

So we may substitute

$$occ(x^{-1},p) = occ(x,p\ )-a$$

and

$$occ(y^{-1},p) = occ(.p\ )-b$$

in (8) to derive:-

$$2occ(x,p) + 2occ(y,p) - a - b < 4r+3b-a,$$

and thus:-

$$0 \leq occ(x,p) + occ(y,p) < 2r+2b$$

By a simple calculation, the latter inequality yields:-

$$occ(x,p)occ(y,p) \leq (b+r)(b+r-1).$$

We have $(p)_3= b \geq 0$ and $(p)_1= a \geq 0$, so, by lemma 5.3.6,

$$(p)_2 \leq occ(y,p)\ occ(x,p),$$

whence

$$(p)_2 \leq (b+r)(b+r-1).$$

This is the contradiction because we were assuming $p = p_5(a,b,r,s)$, while, by (7),

$(p_5(a,b,r,s))_2 = s+(b+r)(b+r-1)$ with $s>0$.

The proof of (vi), being similar, is omitted.

$\boxed{5.3.7\text{(v) and } 5.3.7}$

We believe the reader may find the following table helpful. In the first column we have calculated the products $p_i(a,b,r,s)$ ($a,b \in \mathbb{N}$ and $2 \le i \le 6$), in the second column are the restrictions on r and s (of these products), and in the third column are the norms (of these products).

*5.3.8 Table*

| | | restrictions on r and s | norm |
|---|---|---|---|
| $p_2(a,b,r,s)=$ | $(a, s+br, b)$ | $0 \le r,s;\ r<a;\ s \le b$ | $a+b$ |
| $p_3(a,b,r,s)=$ | $(a, s+b(a+r), b)$ | $0 \le r;\ 0<s;\ a+r \le b;\ s \le b$ | $2(r+1)+a+b$ |
| $p_4(a,b,r,s)=$ | $(a, s+a(b+r), b)$ | $0 \le r;\ 0<s;\ b+r \le a;\ s \le a$ | $2(r+1)+a+b$ |
| $p_5(a,b,r,s)=$ | $(a, s+(b+r)(b+r-1), b)$ | $0<r,s;\ a<b+r;\ s \le b+r$ | $4r+3b-a$ |
| $p_6(a,b,r,s)=$ | $(a, s+(b+r)^2, b)$ | $0<r,s;\ a \le b+r;\ s \le b+r$ | $4r+2+3b-a$ |

We now introduce the maps $\sigma_j$ ($1 \le j \le 8$) in the context of:

*5.3.9 Lemma.*

The following maps, $\sigma_j : C \longrightarrow C$ ($1 \le j \le 8$), extend to automorphisms of G which preserve the norms of the elements of G, i.e., for all $g \in G$, $\| g \| = \| \sigma_j(g) \|$.

$\sigma_1$ being the identity map on G; $\sigma_2 : x \mapsto x^{-1}, y \mapsto y$;
$\sigma_3 : x \mapsto x, y \mapsto y^{-1}$; $\sigma_4 : x \mapsto x^{-1}, y \mapsto y^{-1}$;
$\sigma_5 : x \mapsto y, y \mapsto x$; $\sigma_6 : x \mapsto y^{-1}, y \mapsto x$;
$\sigma_7 : x \mapsto y, y \mapsto x^{-1}$; $\sigma_8 : x \mapsto y^{-1}, y \mapsto x^{-1}$.

$\boxed{5.3.9}$

Recall that the multiplication in G is defined by

$$(a,c,b)\,(\acute{a},\acute{c},\breve{b}) = (\, a+\acute{a}\,,\, c+b\acute{a}+\acute{c}\,,\, b+\breve{b}\,).$$

Also, we defined $x=(1,0,0)$, $y=(0,0,1)$ and $z=(0,1,0)$.

We do not assume that the reader is adept at calculating products in the $x^{\pm1}$'s, $y^{\pm1}$'s and $z^{\pm1}$'s, but hope that the next lemma will be found adequate to confirm any relation we may state during the course of the proof.

*5.3.10 Lemma.*

*(i)* If $p = y^{b_1}x^{a_1}y^{b_2}x^{a_2}\ldots y^{b_n}x^{a_n}$, with all the $a_i, b_i \in \mathbb{Z}$, then

$$p=(\, a_1+a_2+\ldots+a_n\,,\, \sum_{i=1}^{i=n} a_i(\, b_1+b_2+\ldots+b_i\,)\,,\, b_1+b_2+\ldots+b_n\,).$$

(This is lemma 5.3.1, it may seem rather daunting but, in practice, we will only need to calculate such products for n up to 4.)

*(ii)* Let p be a product in terms of $x^{\pm1}$,$y^{\pm1}$ and $z^{\pm1}$. If $occ(x,p)=occ(x^{-1},p)$ then p commutes with y, if $occ(y,p)=occ(y^{-1},p)$ then p commutes with x.

*(iii)* z generates the centre of G and $z=yxy^{-1}x^{-1}=xy^{-1}x^{-1}y=y^{-1}x^{-1}yx=x^{-1}yxy^{-1}$.

*(iv)* With $a,c,b,d \in \mathbb{Z}$:

$$(a,c,b)=x^ay^bz^c=y^bx^az^{(c-ab)}.$$

$$(a,c,b)x^d=(a+d,c+bd,b).$$

$$(a,c,b)y^d=(a,c,b+d).$$

$$(a,c,b)z^d=(a,c+d,b).$$

$$\sigma_2((a,c,b)) = (-a,-c,b).$$

$$\sigma_3((a,b,c)) = (a,-c,-b).$$

$$\sigma_4((a,b,c)) = (-a,c,-b).$$

$$\sigma_5((a,b,c)) = (b,-c+ab,a).$$

$$\sigma_6((a,b,c)) = (-b,c-ab,a).$$

$$\sigma_7((a,b,c)) = (b,c-ab,-a).$$

$$\sigma_8((a,b,c)) = (-b,-c+ab,-a).$$

$\boxed{5.3.10}$

We justify the introduction of the maps $\sigma_j$ $(1\leq j\leq 8)$ by the next lemma.

*5.3.11 Lemma.*

*(i)* If $(a,c,b)\in G$ with $a,c,b\in \mathbb{N}$, then g can be expressed as $p_1(b)$, or as $p_i(a,b,r,s)$ $(2\leq i\leq 6)$ for some $r,s\in \mathbb{N}$.
*(ii)* If $g\in G$, then g can be expressed as $\sigma_j(p_1(b))$, or as $\sigma_j(p_i(a,b,r,s))$ $(1\leq j\leq 8$ and $2\leq i\leq 6)$ for some $a,b,r,s\in \mathbb{N}$.

This lemma will provide us with the following method of calculating $\| g \|$, for arbitrary $g\in G$. By (ii) we can express g in the form $\sigma_j(p_1(b))$ or $\sigma_j(p_i(a,b,r,s))$. Then, by 5.3.9, we will have $\| g \| = \| p_1(b)\|$ or $\| p_i(a,b,r,s)\|$, respectively, which can then be read off table 5.3.8.

*Proof of (i):*

We choose (arbitrary) $a,c,b\in \mathbb{N}$ and exhibit $(a,c,b)$ as a product $p_1(b)$, or as a product $p_2(a,b,r,s)$ (depending on one of six possible situations).

*$0\leq c\leq ab$ and $0=a$.*

Then $c=0$, and we may put $(0,0,b)=p_1(b)=y^b$.

*$0\leq c\leq ab$ and $0<a$.*

Then $c=s+br$ for some $0\leq s\leq b$ and $0\leq r<a$. So (cf. 5.3.8), we may put:-

$$(a,c,b)=p_2(a,b,r,s).$$

*$a\leq b$ and $ab<c\leq b(b+1)$.*

Let n be maximum so that $nb<c$. Then $a\leq n\leq b$, and so we can write $c=s+bn$ for some $0<s\leq b$. We then define r by $n=a+r$, so that $c=s+b(a+r)$. Note that $0\leq r$, $0<s\leq b$ and $a+r=n\leq b$, so (cf. 5.3.8) we may put:-

$$(a,c,b)=p_3(a,b,r,s).$$

*$a\leq b$ and $b(b+1)<c$.*

Let n (necessarily $>b$) be maximum so that $1+n(n-1)\leq c$. Note that (by the choice of n)

$c<1+n(n+1)$, so we may assume either $1+n(n-1)\le c<1+n^2$, or $1+n^2\le c<1+n(n+1)$.

Supposing $1+n(n-1)\le c<1+n^2$. We would then have $c=s+n(n-1)$ for some $1\le s\le n$. By defining r so that $b+r=n$, we have: $c=s+(b+r)(b+r-1)$ with $1\le s\le b+r$, $0<r$ (because $n>b$) and $a<b+r$ (because $a\le b$). So (cf. 5.3.8), we may put:-

$$(a,c,b)=p_5(a,b,r,s).$$

Supposing $1+n^2\le c<1+n(n+1)$. We would then have $c=s+n^2$ for some $1\le s\le n$. By defining r so that $b+r=n$, we have: $c=s+(b+r)^2$ with $1\le s\le b+r$, $0<r$ (because $n>b$) and $a\le b+r$ (because $a\le b$). So (cf. 5.3.8), we may put:-

$$(a,c,b)=p_6(a,b,r,s).$$

*$b\le a$ and $ab<c\le a(a+1)$.*

Let n be maximum so that $na<c$. Then $b\le n\le a$, and we may write $c=s+an$ for some $0<s\le a$. We then define r by $n=b+r$, so that $c=s+a(b+r)$. Note that $0\le r$, $0<s\le a$ and $b+r=n\le a$, so (cf. 5.3.8) we may put:-

$$(a,c,b)=p_4(a,b,r,s).$$

*$b\le a$ and $a(a+1)<c$.*

Let n (necessarily $>a$) be maximum so that $1+n(n-1)\le c$. Note that (by the choice of n) $c<1+n(n+1)$, so we may assume either $1+n(n-1)\le c<1+n^2$, or $1+n^2\le c<1+n(n+1)$.

Supposing $1+n(n-1)\le c<1+n^2$. We would then have $c=s+n(n-1)$ for some $1\le s\le n$. By defining r so that $b+r=n$, we have: $c=s+(b+r)(b+r-1)$ with $1\le s\le b+r$, $0<r$ (because $n>a\ge b$) and $a<b+r$ (because $b+r=n>a$). So (cf. 5.3.8), we may put:-

$$(a,c,b)=p_5(a,b,r,s).$$

Supposing $1+n^2\le c<1+n(n+1)$. We would then have $c=s+n^2$ for some $1\le s\le n$. By defining r so that $b+r=n$, we have: $c=s+(b+r)^2$ with $1\le s\le b+r$, $0<r$ (because $n>a\ge b$) and $a\le b+r$ (because $b+r=n>a$). So (cf. 5.3.8), we may put:-

$$(a,c,b)=p_6(a,b,r,s).$$

$\boxed{5.3.11(i)}$

*Proof of (ii):*

Let $a,c,b\in\mathbb{N}$, then, by (i), we need only exhibit $(\pm a,\pm c,\pm b)$ as $\sigma_j((\acute{a},\acute{c},\acute{b}))$ for some $1\le j\le 8$ and $\acute{a},\acute{c},\acute{b}\in\mathbb{N}$. The required expressions are easily derived provided we bear in mind the definitions of the maps $\sigma_j$ (defined on page 124), and the relations of lemma 5.3.10.

$(a,c,b) = \sigma_1((a,c,b))$.

$$(-a,-c,b) = x^{-a}y^{b}z^{-c}$$

$$= x^{-a}y^{b}(y^{-1}xyx^{-1})^{c} = \sigma_2(x^{a}y^{b}(y^{-1}x^{-1}yx)^{c})$$

$$= \sigma_2(x^{a}y^{b}z^{c}) = \sigma_2((a,c,b)).$$

$$(a,-c,-b) = x^{a}y^{-b}z^{-c}$$

$$= x^{a}y^{-b}(y^{-1}xyx^{-1})^{c} = \sigma_3(x^{a}y^{b}(yxy^{-1}x^{-1})^{c})$$

$$= \sigma_3(x^{a}y^{b}z^{c}) = \sigma_3((a,c,b)).$$

$$(-a,c,-b) = x^{-a}y^{-b}z^{c}$$

$$= x^{-a}y^{-b}(yxy^{-1}x^{-1})^{c} = \sigma_4(x^{a}y^{b}(y^{-1}x^{-1}yx)^{c})$$

$$= \sigma_4(x^{a}y^{b}z^{c}) = \sigma_4((a,c,b)).$$

$$(a,-c,b) = x^{a}y^{b}z^{-c} = y^{b}x^{a}z^{-(c+ab)}$$

$$= y^{b}x^{a}(xyx^{-1}y^{-1})^{(c+ab)} = \sigma_5(x^{b}y^{a}(yxy^{-1}x^{-1})^{(c+ab)})$$

$$= \sigma_5(x^{b}y^{a}z^{(c+ab)}) = \sigma_5((b,c+ab,a)).$$

$$(a,c,-b) = x^{a}y^{-b}z^{c} = y^{-b}x^{a}z^{(c+ab)}$$

$$= y^{-b}x^{a}(xy^{-1}x^{-1}y)^{(c+ab)} = \sigma_6(x^{b}y^{a}(yxy^{-1}x^{-1})^{(c+ab)})$$

$$= \sigma_6(x^{b}y^{a}z^{(c+ab)}) = \sigma_6((b,c+ab,a)).$$

$$(-a,c,b) = x^{-a}y^{b}z^{c} = y^{b}x^{-a}z^{(c+ab)}$$

$$= y^{b}x^{-a}(xy^{-1}x^{-1}y)^{(c+ab)} = \sigma_7(x^{b}y^{a}(yxy^{-1}x^{-1})^{(c+ab)})$$

$$= \sigma_7(x^{b}y^{a}z^{(c+ab)}) = \sigma_7((b,c+ab,a)).$$

$$(-a,-c,-b) = x^{-a}y^{-b}z^{-c} = y^{-b}x^{-a}z^{-(c+ab)}$$

$$= y^{-b}x^{-a}(x^{-1}y^{-1}xy)^{(c+ab)} = \sigma_8(x^{b}y^{a}(yxy^{-1}x^{-1})^{(c+ab)})$$

$$= \sigma_8(x^{b}y^{a}z^{(c+ab)}) = \sigma_8((b,c+ab,a)).$$

**5.3.11(ii) and 5.3.11**

Before going any further with the proof of 5.3.1, it is worthwhile restating some terminology and elementary facts concerning the paths of the Cayley graph $\Gamma=\Gamma_C(G)$.

If $r\in\mathbb{R}$, then $B(r)=\{\ p\in\Gamma \mid d(1,p)\le r\ \}$ is the r ball of $\Gamma$, and $S(r)=\{\ p\in\Gamma \mid d(1,p)=r\ \}$ is the r shell of $\Gamma$. We remind the reader that d is a metric on the *whole of $\Gamma$*, so an edge stays within $B(n)$ $(n\in\mathbb{N})$ if and only if at least one of its end points lies in $B(n-1)$.

In all the subsequent proofs we will be referring to 'paths' strictly within the context of 'paths from (a specified vertex)'. So we may as well define a path simply by stating the unique finite sequence of labels of the (directed) edges which it traverses. Also (bar such assertions being trivial consequences of preceding statements), whenever we assert that some $g\in G$ lies in a ball $B(n)$, then g will be expressed as a product in the $x^{\pm1}$'s and $y^{\pm1}$'s. The assertion may then be checked by counting the number of $x^{\pm1}$'s and $y^{\pm1}$'s in this product (and noting that the sum is no more than n). Whenever we assert some $g\in G$ lies in a shell $S(n)$, then g will be expressed as $\sigma_j(p_i(a,b,r,s))$ $(1\le j\le 8$ and $2\le i\le 6)$ for some $a,b,r,s\in\mathbb{N}$. We may then check, by table 5.3.8, that $\| p_i(a,b,r,s)\| = n$ (and, by table 5.3.8 and lemma 5.3.10(iv), we may calculate $\sigma_j(p_i(a,b,r,s))$ ).

Recall that, for each $L\in\mathbb{N}$, we defined the relation $\text{join}(L)$, on the points of $\Gamma$, by $p_1 \;\text{join}(L)\; p_2$ if and only if $p_1$ is joined to $p_2$ by at least one path which stays within $B(\| p_1 \|)$ and has length no more than L. The relation join(L) is neither symmetric nor transitive (unless $L=0$), but we do know that:

$$p_1 \;\text{join}(L)\; p_2 \text{ and } \| p_1\| =\| p_2\| \Rightarrow p_2 \;\text{join}(L)\; p_1 \,.$$

$$p_1 \;\text{join}(L_1)\; p_2 \text{ and } p_2 \;\text{join}(L_2)\; p_3 \Rightarrow p_1 \;\text{join}(L_1+L_2)\; p_3 \,.$$

Finally, supposing $g_1,g_2\in G$, we shall write $g=g_1\circ g_2$ whenever $g=g_1g_2$ with $\| g_1\| = \| g_2\|$.

The next lemma is merely a formulation of a simple technique which we will use frequently to simplify the search for paths.

*5.3.12 Lemma.*

Suppose $g_1, g_2, h\in G$ and integer L are such that $g=g_1\circ g_2$ and $g_2 \;\text{join}(L)\; g_2h$, then $g \;\text{join}(L)\; gh$.

The proof is trivial and we omit it.

5.3.12

The next lemma and its corollary will conclude the preliminaries, they will be referred to, mostly implicitly, to shorten (somewhat) the proof of theorem 5.3.1.

*5.3.13 Lemma.*

Let $g=p_i(a,b,r,s)$ (for some $2\leq i\leq 6$). If $\| g \| \geq \| gz^{-1} \|$, then $g_{\text{ join(10) }} gz^{-1}$.

*Proof:*

Suppose $g = p_2(a,b,r,s)$

$$= x^{(a-r-1)}y^s x y^{(b-s)} x^r \text{ (with } 0\leq r<a,\ 0\leq s\leq b \text{ and } \| g \|=a+b).$$

If b=0 then s= 0 and we have:-

$$g\in S(a), \text{ but } gz^{-1}= \sigma_5(p_3(0,a,0,1))\in S(a+2).$$

If r=s=0 and a≥b, then we have:-

$$g\in S(a+b), \text{ but } gz^{-1}=\sigma_5(p_3(b,a,0,1))\in S(a+b+2\ ).$$

If r=s=0 and a≤b, then we have:-

$$g\in S(a+b), \text{ but } gz^{-1}=\sigma_5(p_4(b,a,0,1))\in S(a+b+2).$$

If s=0 and b,r>0, then we have

$$g= x^{(a-r)}y^b x^r \in S(a+b).$$

So, if r>1 then:-

$$gx^{-1}= x^{(a-r)}y^b x^{(r-1)}\in B(a+b-1),\ gx^{-1}x^{-1}\in B(a+b-2),$$

and therefore,

$$gx^{-1}x^{-1}y^{-1}\in B(a+b-1) \text{ and } gx^{-1}x^{-1}y^{-1}x \in B(a+b).$$

Also:-

$$gx^{-1}x^{-1}y^{-1}xy = gz^{-1}x^{-1}= x^{(a-r)}y^{(b-1)}xyx^{(r-2)}\in B(a+b-1),$$

thus we may traverse $x^{-1},x^{-1},y^{-1},x,y,x$.

If r=1 then:-

$$g=x^{(a-1)}y^b x \in B(a+b),$$

and so we may traverse $x^{-1},y^{-1},x,y$.

We are left with $s>0$.

If $3<r$ then $x^{-3},y^{-1},x^{-1},y,x^{2}$ may be traversed.

If $3\geq r>0$ then $x^{-(r+1)},y^{-1},x,y,x^{r}$ may be traversed.

If $r=0$ and $b>s$ then $y^{-1},x^{-1},y^{-1},x,y,y$ may be traversed.

If $r=0$ and $b=s$ then $x^{-1},y^{-1},x,y$ may be traversed.

Suppose $g=p_3(a,b,r,s)$

$$= x^{-(r+1)}y^{s}xy^{(b-s)}x^{(a+r)} \text{ (with } 0\leq r,\ a+r\leq b,\ 0<s\leq b \text{ and } \| g\|=2(r+1)+a+b).$$

If $s=1$ then we would have:-

$$g= x^{-r}y^{(b-1)}x^{-1}yx^{(a+r+1)}\in S(2(r+1)+a+b),$$

and therefore,

$$gx^{-1}\in B(2r+1+a+b) \text{ and } gx^{-1}y^{-1}\in B(2(r+1)+a+b).$$

Also:-

$$gx^{-1}y^{-1}x= x^{-r}y^{b}x^{(a+r)}y^{-1}\in B(2r+1+a+b),$$

thus $x^{-1},y^{-1},x,y$ may be traversed.

If $s>1$ then we would have:-

$$g= x^{-(r+1)} \circ p_2(a+r+1,b,a+r,s).$$

Also:-

$$p_2(a+r+1,b,a+r,s)\in S(r+1+a+b),$$

$$p_2(a+r+1,b,a+r,s)z^{-1}= p_2(a+r+1,b,a+r,s-1)\in S(r+1+a+b),$$

and we have just proved that in this situation:-

$$p_2(a+r+1,b,a+r,s)\ {}_{\text{join}(10)}\ p_2(a+r+1,b,a+r,s-1).$$

Therefore, by lemma 5.3.12,

$$g\ {}_{\text{join}(10)}\ gz^{-1}.$$

Suppose $g = p_4(a,b,r,s)$

$$= y^{(b+r)}x^{(a-s)}yx^{s}y^{-(r+1)} \text{ (with } 0\leq r,\ b+r\leq a,\ 0<s\leq a \text{ and } \| g\|=2(r+1)+a+b).$$

Provided $r>0$, we can easily see that:-

$$gy\in B(2r+1+a+b),\ gyy\in S(2r+a+b),$$

whence:-

$$gyyx^{-1} \in B(2r+1+a+b),\ gyyx^{-1}y^{-1} \in B(2(r+1)+a+b).$$

Also:-

$$gyyx^{-1}y^{-1}x = gz^{-1}y = y^{(b+r)}x^{(a+1-s)}yx^{(s-1)}y^{-r} \in B(2r+1+a+b),$$

so we may traverse $y,y,x^{-1},y^{-1},x,y^{-1}$.

If $r=0$ then:-

$$g = y^{b}x^{(a-s)}yx^{s}y^{-1} \in S(2+a+b),$$

and we can see that $y,x^{-1},y^{-1},x$ may be traversed.

Suppose $g = p_5(a,b,r,s)$

$$= x^{-(b+r-a)}y^{s}xy^{(b+r-s)}x^{(b+r-1)}y^{-r}$$

(with $0<r$, $a<b+r$, $0<s\leq b+r$ and $\| g \|=4r+3b-a$).

If $s=1$ then we may write g as:-

$$g = x^{-(b+r-a-1)}y^{(b+r)}x^{(b+r-1)}y^{-(r-1)}xy^{-1}x^{-1}.$$

As this product contains precisely $\| g \|$ generators, so we may traverse $x,y,x^{-1},y^{-1}$.

If $s>1$ then we have:-

$$g = x^{-(b+r-a)} \circ p_4(b+r,b,r-1,s).$$

Also:-

$$p_4(b+r,b,r-1,s) \in S(3r+2b),$$

$$p_4(b+r,b,r-1,s)\, z^{-1} = p_4(b+r,b,r-1,s-1) \in S(3r+2b),$$

and we have already proved that in this situation:-

$$p_4(b+r,b,r-1,s) \ {}_{\text{join}(10)}\ p_4(b+r,b,r-1,s)\, z^{-1}.$$

Therefore, by lemma 5.3.12,

$$g \ {}_{\text{join}(10)}\ gz^{-1}.$$

Suppose $g = p_6(a,b,r,s)$

$$= x^{-(b+r+1-a)}y^{s}xy^{(b+r-s)}x^{(b+r)}y^{-r}$$

(with $0<r$, $a\leq b+r$, $0<s<b+r$ and $\| g \|=4r+2+3b-a$).

If $s=1$ then we may write g as:-

$$g = x^{-(b+r-a)}y^{(b+r)}x^{(b+r)}y^{-(r-1)}xy^{-1}x^{-1}.$$

As this product contains precisely $\| g \|$ generators, so we may traverse $x,y,x^{-1},y^{-1}$.

If $s>1$ then we have:-

$$g = x^{-(b+r+1-a)} \circ p_4(b+r+1,b,r-1,s+1).$$

Also:-

$$p_4(b+r+1,b,r-1,s+1) \in S(3r+1+2b),$$

$$p_4(b+r+1,b,r-1,s+1)\, z^{-1} = p_4(b+r+1,b,r-1,s) \in S(3r+1+2b),$$

and we have already proved that in this situation:-

$$p_4(b+r+1,b,r-1,s+1)\ _{\text{join}(10)}\ p_4(b+r+1,b,r-1,s+1)\, z^{-1}.$$

Therefore, by lemma 5.3.12,

$$g\ _{\text{join}(10)}\ gz^{-1}.$$

$\boxed{5.3.13}$

*5.3.14 Corollary.*

Let $gz=p_i(a,b,r,s)$ (for some $2\leq i\leq 6$). If $\| g \| = \| gz \|$, then $g\ _{\text{join}(10)}\ gz$.

*Proof:*

We have $gz$ of the form $p_i(a,b,r,s)$, and $\| (gz)z^{-1}\| = \| g \| = \| gz \|$. So, by 5.3.9, $gz\ _{\text{join}(10)}\ g$, whence $g\ _{\text{join}(10)}\ gz$ (because $\| g \| = \| gz \|$).

$\boxed{5.3.14}$

We are now ready to prove theorem 5.3.1, i.e., with $C=\{ x , x^{-1} , y , y^{-1} \}$, $\Gamma_C(G)$ is almost convex. By theorem 5.1.2, it will suffice to prove that $\Gamma_C(G)$ has the property ac(2), i.e., there is an $\kappa\in \mathbb{N}$ such that, for any $g\in G$ and $c,\acute{c} \in C$:

*(i)* whenever $\| g \| = \| gc \|$, then $g\ _{\text{join}(\kappa)}\ gc$.

*(ii)* whenever $\| g \| = \| gc\acute{c} \|$, then $g\ _{\text{join}(\kappa)}\ gc\acute{c}$.

We immediately prove:

*5.3.15 Lemma.*

(i) cannot occur.

*Proof:*

The relations $(m,n,p)x^{\pm 1} = (m\pm 1, n\pm p, p)$ and $(m,n,p)y^{\pm 1} = (m,n,p\pm 1)$ hold for all $m,n,p \in \mathbb{Z}$. We can easily see from these relations that, for any $g=(m,n,p)\in G$ and $c\in C$, $(g)_1+(g)_3$ and $(gc)_1+(gc)_3$ must differ by precisely 1. By corollary 5.3.4, however, we know that the difference between $(g)_1+(g)_3$ and $\| g \|$ is even, and that the difference between $(gc)_1+(gc)_3$ and $\| gc \|$ is even. So it is not possible for $g\in G$ and $c\in C$ to be such that $\| g \| = \| gc \|$.

$\boxed{5.3.15}$

So we are left with the task of finding some κ with the property that, for any $g\in G$ and $c,\acute{c} \in C$:

$$(ii) \text{ whenever } \| g \| = \| gc\acute{c} \|, \text{ then } g \ {}_{join(\kappa)}\ gc\acute{c}.$$

We will prove that κ=84 is a (rather generous) bound. The proof is laborious; we will catalogue all the possibilities of (ii) as follows.

We first prove (ii) for all $g=p_1(b)$. Then, in proposition 5.3.16, we will prove (ii) for all $g= p_i(a,b,r,s)$ $(2\le i\le 6)$, and all $c\acute{c} \in \{ xy, x^{-1}y, xx, yx^{-1}, yx, yy \}$. In proposition 5.3.18 we will prove (ii) for all $g= p_i(a,b,r,s)$ $(2\le i\le 6)$, and all $c\acute{c} \in \{ y^{-1}x^{-1}, y^{-1}x, x^{-1}x^{-1}, xy^{-1}, x^{-1}y^{-1}, y^{-1}y^{-1} \}$. We conclude the proof of theorem 5.3.1 by describing how 5.3.16 and 5.3.18 imply (ii) for an arbitrary $g\in G$ and all $c\in C$.

So let us prove (ii) for all $g=p_1(b)=y^b$ $(b\in \mathbb{N})$, and all $c,\acute{c}\in C$ (with $c\acute{c}\neq 1$). This is absolutely trivial. If $b=0$ then $g=1$, so $\| g \|=0$ while $\| gc\acute{c} \| =2$. If $b>0$ and $c=y^{-1}$, then $\| g \|=b$, $\| gc \|=b-1$ and so, trivially, $g \ {}_{join(2)}\ gc\acute{c}$. If $b>0$ and $c\neq y^{-1}$, then it is easy (but we do not bother) to prove that $\| g \|=b$ while $\| gc\acute{c} \| =b+2$.

Before beginning propositions 5.3.16 and 5.3.18, we should mention the possibilities of (ii) which are omitted from the cataloging.

We may omit $g=p_i(a,b,r,s)$ post multiplied by $c\acute{c}$ whenever the defining product (page 120) of $p_i(a,b,r,s)$ ends with $c^{-1}$. This is because the defining product of $p_i(a,b,r,s)$ is a product of precisely $\| p_i(a,b,r,s)\|$ generators of $\{x,y\}$. Thus, if $p_i(a,b,r,s)$ ends with $c^{-1}$, then $\| p_i(a,b,r,s)c\| = \| p_i(a,b,r,s)\|-1$ and, trivially, we have $p_i(a,b,r,s) \ {}_{join(2)}\ p_i(a,b,r,s)c\acute{c}$.

We may omit $g=p_2(a,b,r,s)$ post multiplied by $c\acute{c}=$ xy, xx, yx or yy. This is because the defining product of $p_2(a,b,r,s)$, i.e., $x^{(a-r-1)}y^sxy^{(b-s)}x^r$ with $0\leq r<a$ and $0\leq s\leq b$, is a product of $a+b$ of the generators of $\{x,y\}$. Thus $p_2(a,b,r,s)$ is a product of $a+b+2$ of the generators of $\{x,y\}$ and so, by 5.3.3, we would have $\| g \|=a+b$ while $\| gc\acute{c} \| =a+b+2$.

*Proposition 5.3.16.*

With $g=p_i(a,b,r,s)$ $(2\leq i\leq 6)$ and $c\acute{c} \in \{ xy, x^{-1}y, xx, yx^{-1}, yx, yy \}$,

$$\text{whenever } \| g \| = \| gc\acute{c} \| \text{, then } g_{\ join(36)}\ gc\acute{c}.$$

*Proof:*

Let $c\acute{c}=xy$.

Suppose $g=p_3(a,b,r,s)$

$$= x^{-(r+1)}y^sxy^{(b-s)}x^{(a+r)} \text{ (with } 0\leq r,\ a+r\leq b,\ 0<s\leq b \text{ and } n=\| g \|=2(r+1)+a+b).$$

If $s>a+r+1$ then we would have:-

$$gxy = p_3(a+1,b+1,r,s-a-r-1) \in S(n+2).$$

So we may now assume $s\leq a+r+1$. We will then have:-

$$gz^{-1} = x^{-(r+1)}y^{(s-1)}xy^{(b+1-s)}x^{(a+r)} \in B(n),$$

and, if $1\leq a+r$,

$$gz^{-1}y = x^{-r}y^{(s+b-a-r)}xy^{(a+r+1-s)}x^{(a+r-1)} \in B(n-1),$$

or, if $0=a+r$,

$$gz^{-1}y = y^{(b+1)} \in B(n-1).$$

Therefore (when $s\leq a+r+1$):-

$$gz^{-1} \in B(n),\ gz^{-1}y \in B(n-1) \text{ and } gz^{-1}y\,x = gxy \in B(n),$$

and so (by 5.3.13) we may traverse $z^{-1},y,x$.

Suppose $g = p_4(a,b,r,s)$

$$= y^{(b+r)}x^{(a-s)}yx^sy^{-(r+1)} \text{ (with } 0\leq r,\ b+r\leq a,\ 0<s\leq a \text{ and } n=\| g \|=2(r+1)+a+b).$$

We have:-

$$gz^{-1} = y^{(b+r)}x^{(a+1-s)}yx^{(s-1)}y^{-(r+1)} \in B(n),$$

and thus

$$gz^{-1}y \in B(n-1),\ gz^{-1}yx = gxy \in B(n).$$

So, by 5.3.13, we may traverse $z^{-1}$,y,x.

Suppose $g = p_5(a,b,r,s)$

$$= x^{-(b+r-a)}y^s xy^{(b+r-s)}x^{(b+r-1)}y^{-r}$$

(with $0<r$, $a<b+r$, $0<s\le b+r$ and $n=\| g \|=4r+3b-a$).

We have:-

$$gz^{-1} = x^{-(b+r-a)}y^{(s-1)}xy^{(b+r+1-s)}x^{(b+r-1)}y^{-r} \in B(n),$$

and thus

$$gz^{-1}y \in B(n-1),\ gz^{-1}yx = gxy \in B(n).$$

So, by 5.3.13, we may traverse $z^{-1}$,y,x.

Suppose $g = p_6(a,b,r,s)$

$$= x^{-(b+r+1-a)}y^s xy^{(b+r-s)}x^{(b+r)}y^{-r}$$

(with $0<r$, $a\le b+r$, $0<s\le b+r$ and $n=\| g \|=4r+2+3b-a$).

We have:-

$$gz^{-1} = x^{-(b+r+1-a)}y^{(s-1)}xy^{(b+r+1-s)}x^{(b+r)}y^{-r} \in B(n),$$

and thus

$$gz^{-1}y \in B(n-1),\ gz^{-1}yx = gxy \in B(n).$$

So, by 5.3.13, we may traverse $z^{-1}$,y,x.$*$

$\boxed{c\acute{c}=xy}$

The maximal path length so far is 12, we now post multiply by $c\acute{c}=x^{-1}y$.

Suppose $g = p_2(a,b,r,s)$

$= x^{(a-r-1)}y^s xy^{(b-s)}x^r$ (with $0\le r<a$, $0\le s\le b$ and $n=\| g \|=a+b$).

Clearly we may assume that $r=0$.

If $b=s$ then $g=x^{(a-1)}y^b x \in S(a+b)$, and therefore $gx^{-1} \in S(a+b-1)$.

If $s<b$ and $b+1\le a-1$ then:-

$$g\in S(a+b), \text{ but } gx^{-1}y=\sigma_5(p_3(b+1,a-1,0,b-s))\in S(a+b+2).$$

If $s<b$ and $b+1\geq a-1$ then:-

$$g\in S(a+b), \text{ but } gx^{-1}y=\sigma_5(p_4(b+1,a-1,0,b-s))\in S(a+b+2).$$

Suppose $g=p_4(a,b,r,s)$

$$=y^{(b+r)}x^{(a-s)}yx^sy^{-(r+1)} \text{ (with } 0\leq r,\ b+r\leq a,\ 0<s\leq a \text{ and } n=\|g\|=2(r+1)+a+b).$$

Note that $n\leq a+a+2(a+1)=4a+2$, so we may as well assume $a\geq 2$ (if $a<2$ then $n\leq 6$ and the distance between any two points of $B(n)$ would be at most 12).

If $s+1\leq a$ then we would have:-

$$gz=p_4(a,b,r,s+1)=y^{(b+r)}x^{(a-s-1)}yx^{(s+1)}y^{-(r+1)}\in S(n),$$

and thus

$$gzy\in B(n-1),\ gzyx^{-1}=gx^{-1}y\in B(n).$$

So, when $s+1\leq a$, we may traverse $z,y,x^{-1}$.

If $s=a$ and $r\geq 1$ then:-

$$g=y^{(b+r+1)}x^ay^{-(r+1)}\in S(2(r+1)+a+b),$$

and so we may traverse $y,x^{-1},y,x^{-1},y,x,y^{-2}$.

If $s=a$ and $r=0$ then:-

$$g=y^{(b+1)}x^ay^{-1}\in S(2+a+b),$$

and so we may traverse $y,x^{-2},y,x,y^{-1}$.

Suppose $g=p_5(a,b,r,s)$

$$=x^{-(b+r-a)}y^sxy^{(b+r-s)}x^{(b+r-1)}y^{-r}$$

(with $0<r$, $a<b+r$, $0<s\leq b+r$ and $n=\|g\|=4r+3b-a$).

If $s<b+r$ then we would have:-

$$gz=p_5(a,b,r,s+1)\in S(n),$$

also

$$gzy=x^{-(b+r-a)}y^{(s+1)}xy^{(b+r-s-1)}x^{(b+r-1)}y^{-(r-1)}\in B(n-1),$$

and so we may traverse $z,y,x^{-1}$.

If $s=b+r$ and $r=1$ then:-

$$g= x^{-(s-a)}y^s x^s y^{-1} \in S(n) \ (n=3s+1-a),$$

and thus

$$gy \in B(n-1),\ gyx^{-1} \in B(n-2),\ gyx^{-1}y^{-1} \in B(n-1),\ gyx^{-1}y^{-1}x^{-1} \in B(n).$$

Also

$$gyx^{-1}y^{-1}x^{-1}y = gx^{-1}yx^{-1} = x^{-(s-a)}y^{(s-1)}x^{-1}yx^{(s-1)} \in B(n-1),$$

so, when $s=b+r$ and $r=1$, we may traverse $y,x^{-1},y^{-1},x^{-1},y,x$.

If $s=b+r$ and $r>1$ then we have:-

$$g = x^{-(s-a)} \circ p_4(s,s-r,r-1,s),$$

also

$$p_4(s,s-r,r-1,s) \in S(2s-r),$$

$$p_4(s,s-r,r-1,s)\, x^{-1}y = p_5(s-1,s-r+1,r-1,r) \in S(2s+r).$$

We have just proved that in this situation:-

$$p_4(s,s-r,r-1,s) {}_{\text{join}(12)} p_4(s,s-r,r-1,s)\, x^{-1}y,$$

therefore, by lemma 5.3.12,

$$g {}_{\text{join}(12)} gz^{-1}.$$

Suppose $g = p_6(a,b,r,s)$

$$= x^{-(b+r+1-a)}y^s x y^{(b+r-s)}x^{(b+r)}y^{-r}$$

(with $0<r$, $a\leq b+r$, $0<s\leq b+r$ and $n=\| g \|=4r+2+3b-a$).

If $s<b+r$ then we have:-

$$gz = p_6(a,b,r,s+1) \in S(n),$$

$$g\,z\,y \ = \ x^{-(b+r+1-a)}y^{(s+1)}xy^{(b+r-s-1)}x^{(b+r)}y^{-(r-1)} \in B(n-1),$$

thus, we may traverse $z,y,x^{-1}$.

If $s=b+r$ and $r=1$ then:-

$$g= x^{-(s-a+1)}y^s x^{(s+1)}y^{-1} \in S(n) \quad (n=3s+3-a),$$

and so

$$gy \in B(n-1),\ gyx^{-1} \in B(n-2),\ gyx^{-1}y^{-1} \in B(n-1),\ gyx^{-1}y^{-1}x^{-1} \in B(n).$$

Also

$$gyx^{-1}y^{-1}x^{-1}y = gx^{-1}yx^{-1} = x^{-(s-a+1)}y^{(s-1)}x^{-1}yx^{s} \in B(n-1),$$

so, when $s=b+r$ and $r=1$, we may traverse $y, x^{-1}, y^{-1}, x^{-1}, y, x$.

If $s=b+r$ and $r>1$ then we have:-

$$g = x^{-(s-a+1)} \circ p_4(s+1, s-r, r-1, s+1),$$

also

$$p_4(s+1, s-r, r-1, s+1) \in S(2s+r+1),$$

$$p_4(s+1, s-r, r-1, s+1)x^{-1}y = p_6(s, s-r+1, r-1, r) \in S(2s+r+1).$$

We have already proved that in this situation:-

$$p_4(s+1, s-r, r-1, s+1) \ {}_{\text{join}(12)} \ p_4(s+1, s-r, r-1, s+1)\, x^{-1}y,$$

therefore, by lemma 5.3.12,

$$g \ {}_{\text{join}(12)} \ gz^{-1}.$$

$\boxed{c\acute{c}=x^{-1}y}$

The maximal path length so far is 12, we now post multiply by $c\acute{c}=xx$.

Suppose $g=p_3(a,b,r,s)$

$= x^{-(r+1)}y^{s}xy^{(b-s)}x^{(a+r)}$ (with $0\le r$, $a+r\le b$, $0<s\le b$ and $n=\| g \|=2(r+1)+a+b$).

If $b\ge a+r+2$ then we have

$$gxx = p_3(a+2, b, r, s) \in S(n+2).$$

Also, if $b<a+r+2$ and $r=0$, then we would have

$$gxx = p_4(a+2, b, 0, s) \in S(n+2).$$

So we may assume $b<a+r+2$ and $r>0$, but, as $b\ge a+r$ anyway, so we are left with ($b=a+r+1$ and $r>0$) or ($b=a+r$ and $r>0$).

If $b=a+r+1$ and $r>0$, then $gxx=p_5(a+2, b, 1, s) \in S(n+2)$.

If $b=a+r$, $r>0$ and $s>1$, then $gxx=p_6(a+2, b, 1, s-1) \in S(n+2)$.

If $b=a+r$, $r>0$ and $s=1$, then we have:-

$$g = x^{-(r+1)}yxy^{(b-1)}x^{b} \in S(2b+r+2),$$

$$gz^{-1} = x^{-r}y^{b}x^{b} \in B(2b+r),$$

and thus

$$gz^{-1}x \in B(2b+r+1),\ gz^{-1}xy \in B(2b+r+2).$$

Also:-

$$gz^{-1}xyx = x^{-(r-1)}y^{(b+1)}x^{(b+1)} \in B(2b+r+1),$$

therefore

$$gz^{-1}xyxy^{-1} = gxx \in B(2b+r+2),$$

and so we may traverse $z^{-1},x,y,x,y^{-1}$.

Suppose $g = p_4(a,b,r,s)$

$= y^{(b+r)}x^{(a-s)}yx^{s}y^{-(r+1)}$ (with $0\leq r$, $b+r\leq a$, $0<s\leq a$ and $n=\| g \|=2(r+1)+a+b$).

If $s\leq r$ then we have:-

$$gx = y^{(b+r)}x^{a}y^{-(r-s)}xy^{-s} \in S(n-1),$$

so we may traverse x,x.

If $s>2r$ then $gxx = p_4(a+2,b,r,s-2r) \in S(n+2)$.

So we may assume $r<s\leq 2r$, and thus $r\geq 1$, $s\geq 2$. We would have:-

$$gz^{-1} = p_4(a,b,r,s-1) \in S(n),$$

$$gz^{-1}z^{-1} = y^{(b+r)}x^{(a+2-s)}yx^{(s-2)}y^{-(r+1)} \in B(n).$$

and thus

$$gz^{-1}z^{-1}y \in B(n-1),\ gz^{-1}z^{-1}yx \in B(n).$$

Also

$$gz^{-1}z^{-1}yxx = y^{(b+r-1)}x^{(2r-s)}yx^{(a+s+2-2r)}y^{-r} \in B(n-1),$$

and thus

$$gxx = gz^{-1}z^{-1}yxxy^{-1} \in B(n\ ),$$

So, when $r<s\leq 2r$, we may traverse (the rather long route of) $z^{-1},z^{-1},y,x,x,y^{-1}$.

Suppose $g = p_5(a,b,r,s)$

$= x^{-(b+r-a)}y^{s}xy^{(b+r-s)}x^{(b+r-1)}y^{-r}$

(with $0<r$, $a<b+r$, $0<s\leq b+r$ and $n=\| g \|=4r+3b-a$).

If $s>2r$ and $a+1<b+r$, then $gxx=p_5(a+2,b,r+1,s-2r)\in S(n+2)$.

If $s>2r$ and $a+1=b+r$, then $gxx=p_4(a+2,b,r,s-2r)\in S(n+2)$.

If $r\geq s$ then we would have:-

$$gx=x^{-(b+r-a-1)}y^{(s+b)}xy^{(r-s)}x^{(b+r-1)}y^{-r}\in S(n-1),$$

and so we may traverse x,x.

We are left with the possibility of $r<s\leq 2r$, but then we would have:-

$$gz^{-1}=p_5(a,b,r,s-1)\in S(n),$$

also

$$gz^{-1}z^{-1}=x^{-(b+r-a)}y^{(s-2)}xy^{(b+r+2-s)}x^{(b+r-1)}y^{-r}\in B(n),$$

and thus

$$gz^{-1}z^{-1}y\in B(n-1),\ gz^{-1}z^{-1}yx\in B(n).$$

As

$$gz^{-1}z^{-1}yxx=x^{-(b+r-a-1)}y^{(s+b-r)}xy^{(2r-s)}x^{(b+r)}y^{-(r-1)}\in B(n-1),$$

so we may traverse $z^{-1},z^{-1},y,x,x,y^{-1}$.

Suppose $g=p_6(a,b,r,s)$
$\quad =x^{-(b+r+1-a)}y^{s}xy^{(b+r-s)}x^{(b+r)}y^{-r}$

(with $0<r$, $a\leq b+r$, $0<s\leq b+r$ and $n=\|g\|=4r+2+3b-a$).

If $s>2r+1$ and $a+1\leq b+r$, then $gxx=p_6(a+2,b,r+1,s-2r-1)\in S(n+2)$.

If $s>2r+1$ and $a+1=b+r$, then $gxx=p_4(a+2,b,r+1,s-2r-1)\in S(n+2)$.

If $r\geq s$ then we would have:-

$$gx=x^{-(b+r-a)}y^{(b+r)}x^{(b+r)}y^{-(r-s)}\in S(n-1),$$

and so we may traverse x,x.

So we may now assume $r<s\leq 2r+1$.

If $a=b+r$ and $s=2r+1$, then:-

$$gxx=p_4(a+2,b,r,s-2r)\in S(n+2),$$

and so, as $a\leq b+r$ anyway, we can also assume either $a<b+r$ or ($a=b+r$ and $s<2r+1$).

If $a<b+r$ then:-

$$gz^{-1}z^{-1}yxx=x^{-(b+r-a-1)}y^{(s+b-r)}xy^{(2r+1-s)}x^{(b+r)}y^{-r}\in B(n-1),$$

or, if $a=b+r$ and $s<2r+1$, then:-

$$gz^{-1}z^{-1}yxx = y^{(b+r-1)}x^{(2r-s)}yx^{(s+b+2-r)}y^{-(r-1)} \in B(n-1).$$

Whichever, we also have:-

$$gz^{-1} = p_6(a,b,r,s-1) \in S(n),$$

$$gz^{-1}z^{-1} = x^{-(b+r+1-a)}y^{(s-2)}xy^{(b+r+2-s)}x^{(b+r)}y^{-r} \in B(n)$$

and therefore

$$g_{\text{ join}(20)}\ gz^{-1}z^{-1},\ gz^{-1}z^{-1}y \in B(n-1),\ gz^{-1}z^{-1}yx \in B(n).$$

As $gz^{-1}z^{-1}yxx \in B(n-1)$, so we may traverse $z^{-1},z^{-1},y,x,x,y^{-1}$.

$\boxed{c\acute{c}=xx}$

The maximal path length so far is 24, we now post multiply by $c\acute{c}=yx^{-1}$.

Suppose $g = p_2(a,b,r,s)$

$$= x^{(a-r-1)}y^{s}xy^{(b-s)}x^{r} \text{ (with } 0\leq r<a,\ 0\leq s\leq b \text{ and } n=\|g\|=a+b).$$

If $s+br<b+1$ and $a-1\leq b+1$ then:-

$$gyx^{-1} = \sigma_4(p_4(b+1,a-1,0,b+1-s-br)) \in S(a+b+2).$$

If $s+br<b+1$ and $a-1\geq b+1$ then:-

$$gyx^{-1} = \sigma_5(p_3(b+1,a-1,0,b+1-s-br)) \in S(a+b+2).$$

So we can now assume $s+br\geq b+1$, and therefore, as $0<s\leq b$ anyway, $b>0$ and $r>0$.

If $s=0$ then $r>1$ and we would have:-

$$gz^{-1} = x^{(a-r)}y^{(b-1)}xyx^{(r-1)} \in B(a+b),$$

or, if $s>0$ then we would have:-

$$gz^{-1} = x^{(a-r-1)}y^{(s-1)}xy^{(b+1-s)}x^{r} \in B(a+b).$$

Whichever, we can see that:-

$$gz^{-1} \in B(a+b),\ gz^{-1}x^{-1} = gyx^{-1}y^{-1} \in B(a+b-1),$$

and so we may traverse $z^{-1},x^{-1},y$.

Suppose $g=p_3(a,b,r,s)$

$$= x^{-(r+1)}y^{s}xy^{(b-s)}x^{(a+r)} \text{ (with } 0\leq r,\ a+r\leq b,\ 0<s\leq b \text{ and } n=\|g\|=2(r+1)+a+b).$$

We have:-

$$gz^{-1}=x^{-r}y^{(b+1-s)}x^{-1}y^{(s-1)}x^{(a+r+1)}\in B(n),$$

and thus

$$gz^{-1}x^{-1}\in B(n-1),\ gz^{-1}x^{-1}y=gyx^{-1}y^{-1}\in B(n).$$

So we may traverse $z^{-1},x^{-1},y$.

$\boxed{c\acute{c}=yx^{-1}}$

The maximal path length so far is 24, we now post multiply by $c\acute{c}=yx$.

Suppose $g=p_3(a,b,r,s)$

$$=x^{-(r+1)}y^sxy^{(b-s)}x^{(a+r)}\text{ (with }0\leq r,\ a+r\leq b,\ 0<s\leq b\text{ and }n=\|g\|=2(r+1)+a+b).$$

If $a+r<s$ then $gyx=p_3(a+1,b+1,r,s-a-r)\in S(n+2)$.

If $a+r\geq s$ then we would have:-

$$gy=x^{-r}y^bx^{(a+r-s)}yx^s\in B(n-1),$$

and so we may traverse y,x.

$\boxed{c\acute{c}=yx}$

The maximal path length so far is 24, we now post multiply by $c\acute{c}=yy$.

Suppose $g=p_3(a,b,r,s)$

$$=x^{-(r+1)}y^sxy^{(b-s)}x^{(a+r)}\text{ (with }0\leq r,\ a+r\leq b,\ 0<s\leq b\text{ and }n=\|g\|=2(r+1)+a+b).$$

If $s\leq a+r$ then:-

$$gy=x^{-r}y^bx^{(a+r-s)}yx^s\in B(n-1),$$

and so we could traverse y,y.

If $0=a+r$ then $gy\,y=p_3(0,b+2,0,s)\in S(n+2)$.

So we can now assume $0<a+r<s$.

If $a>0$ then we would have:-

$$g\in S(a+b+2(r+1)),$$

and

$$gy\,x^{-1}=p_3(a-1,b+1,r,s-a-r)\in S(a+b+2(r+1)).$$

Also, we can certainly assume, without loss of generality, that

$$gy\,x^{-1}(xy) = gy\,y \in S(a+b+2(r+1)).$$

Thus, we would already have proved

$$g \;_{\text{join}(24)}\; gy\,x^{-1},\; gy\,x^{-1} \;_{\text{join}(12)}\; gy\,y,$$

and so (when a=0)

$$g \;_{\text{join}(36)}\; gy\,y.$$

If a=0 and 2r<s, then we would have $gyy = p_3(0,b+2,r,s-2r) \in S(n+2)$.

If a=0 and 2r≥s, then r>0 (because a+r>0) and so, also, 2r ≥ s. We would now have:-

$$g = x^{-r} \circ p_3(r,b,0,s).$$

Also

$$p_3(r,b,0,s) \in S(r+a+b),$$
$$p_3(r,b,0,s)yy = p_2(r,b+2,r-1,b+s-2r+2) \in S(r+2+b)$$

and we have just proved that in this situation

$$p_3(r,b,0,s) \;_{\text{join}(36)}\; p_2(r,b+2,r-1,b+s-2r+2).$$

Therefore, by lemma 5.3.8,

$$g \;_{\text{join}(36)}\; gyy.$$

$\boxed{\text{cć=yy and 5.3.16}}$

*Corollary 5.3.17*

Suppose g∈G, cć ∈ { $y^{-1}x^{-1}$, $y^{-1}x$, $x^{-1}x^{-1}$, $xy^{-1}$, $x^{-1}y^{-1}$, $y^{-1}y^{-1}$ } and ‖ g ‖ = ‖ gcć ‖ with gcć=(a,d,b) for some a,b,d∈ ℕ, then $g \;_{\text{join}(36)}$ gcć.

*Proof:*

By 5.3.11(i), we may write gcć = $p_1(b)$ or $p_i(a,b,r,s)$ (2≤i≤6) for some r,s∈ ℕ. Also, ‖ (gcć)$(c'c)^{-1}$ ‖ = ‖ gcć ‖ (=‖ g ‖ ) with $(c'c)^{-1}$ ∈ { xy, $x^{-1}y$, xx, $yx^{-1}$, yx, yy }, thus, by proposition 5.3.16, gcć $_{\text{join}(36)}$ g. As ‖ g ‖ = ‖ gcć ‖, so we have $g \;_{\text{join}(36)}$ gcć.

$\boxed{\text{5.3.17}}$

*Proposition 5.3.18.*

With $g=p_i(a,b,r,s)$ $(2\leq i\leq 6)$ and $c\acute{c} \in \{ y^{-1}x^{-1}, y^{-1}x, x^{-1}x^{-1}, xy^{-1}, x^{-1}y^{-1}, y^{-1}y^{-1} \}$,

*(ii)* whenever $\| g \| = \| gc\acute{c} \|$, then $g \; {}_{\text{join}(84)} \; gc\acute{c}$.

*Proof:*

We note that, by corollary 5.3.17, (ii) would already have been proved if $gc\acute{c}=(a,d,b)$ for some $a,b,d\in \mathbb{N}$.

Let $c\acute{c}=y^{-1}x^{-1}$.

Suppose $g = p_2(a,b,r,s)$

$= x^{(a-r-1)}y^sxy^{(b-s)}x^r$ (with $0\leq r<a$, $0\leq s\leq b$ and $n=\| g \|=a+b$).

If $s<b$ and $r=0$ then:-

$$gy^{-1} = x^{(a-r-1)}y^sxy^{(b-s-1)} \in B(n-1),$$

and so we may traverse $y, x^{-1}$.

If $s<b$ and $r>0$ then:-

$$gz = p_2(a,b,r,s+1) \in S(n),$$

$$gzx^{-1} = x^{(a-r-1)}y^{(s+1)}xy^{(b-s-1)}x^{(r-1)} \in B(n-1)$$

and so we may traverse $z,x^{-1},y^{-1}$.

If $s=b$ then $gy^{-1}x^{-1}=(a-1,br+1,b-1)$ with $a-1,br+1\geq 0$. So, by 5.3.17, we may as well assume $b=0$, but then $g = x^a \in S(a)$ and $gy^{-1}x^{-1} \in S(a+2)$.

Suppose $g=p_3(a,b,r,s)$

$= x^{-(r+1)}y^sxy^{(b-s)}x^{(a+r)}$ (with $0\leq r$, $a+r\leq b$, $0<s\leq b$ and $n=\| g \|=2(r+1)+a+b$).

If $s<b$ then we have:-

$$gz = p_3(a,b,r,s+1) \in S(n),$$

and so, by corollary 5.3.13,

$$g \; {}_{\text{join}(10)} \; gz.$$

Also

$$gzx^{-1} = gy^{-1}x^{-1}y = x^{-r}y^{(b-s-1)}x^{-1}y^{(s+1)}x^{(a+r)} \in B(n-1),$$

and so we may traverse $z,x^{-1},y^{-1}$.

If $s=b$ then $gy^{-1}x^{-1}=(a-1,b-1,b(a+r)+1)$ with $b-1,b(a+r)+1\geq 0$. So, by 5.3.17, we can assume $a=0$. If $r=0$ then we may traverse $x^{-1},y^{-1},x,y^{-1},x^{-1},y$. If $r=1$ then we may traverse $x^{-1},y^{-1},y^{-1},x^{-1},y,x$. If $r>1$ then we may traverse $x^{-1},y^{-1},x^{-1},y^{-1},x^{-1},y,x,x$.

Suppose $g = p_4(a,b,r,s)$

$$= y^{(b+r)}x^{(a-s)}yx^sy^{-(r+1)} \text{ (with } 0\leq r,\ b+r\leq a,\ 0<s\leq a \text{ and } n=\|g\|=2(r+1)+a+b).$$

Note that $n\leq 2(a+1)+a+a=4a+2$, and so we may as well assume $a>2$ (if $a\leq 2$ then $n\leq 10$ and the distance between any two points of $B(n)$ would be at most 20).

Also, $gy^{-1}x^{-1}=(a-1,s+a(b+r)-b+1),b-1)$ with $a-1\geq 0$. So by 5.3.17, we may as well assume $b=0$ or $s+a(b+r)<b-1$. As $s+a(b+r)<b-1 \Rightarrow b+r=0$, so we can restrict to $b=0$. Thus, we are assuming $a>2$ and $b=0$.

If $r+1\leq a-2$ and $a<s+r+2$ then $gy^{-1}x^{-1}=\sigma_6(p_3(1,a-1,r+1,s+r+2-a))\in S(n+2)$.

If $r+1\leq a-2$ and $a\geq s+r+2$, we would have:-

$$gz=p_4(0,a,r,s+1),$$

and thus, by 5.3.14,

$$g_{\text{ join}(10)}\, gz.$$

Also,

$$gzx^{-1}=gy^{-1}x^{-1}y=y^rx^{(a-s-r-2)}yx^{(s+r+1)}y^{-(r+1)}\in B(n-1),$$

so we may traverse $z,x^{-1},y^{-1}$.

We are left with $r+1\geq a-2$, but as $r\leq a-b$ anyway, and we are assuming $b=0$, so we have $r=a$, $a-1$ or $a-2$.

If $r=a$ then we have:-

$$g\in S(3a+2), \text{ but } gy^{-1}x^{-1}=\sigma_6(p_5(1,a-1,2,s))\in S(3a+4).$$

If $r=a-1$ then we have:-

$$g\in S(3a), \text{ but } gy^{-1}x^{-1}=\sigma_6(p_6(1,a-1,1,s))\in S(3a+2).$$

If $r=a-2$ then we have:-

$$g\in S(3a-2), \text{ but } gy^{-1}x^{-1}=\sigma_6(p_5(1,a-1,1,s))\in S(3a).$$

Suppose $g = p_5(a,b,r,s)$

$$= x^{-(b+r-a)}y^{s}xy^{(b+r-s)}x^{(b+r-1)}y^{-r}$$

(with $0<r$, $a<b+r$, $0<s\leq b+r$ and $n=\| g \|=4r+3b-a$).

Note that $n\leq 4(b+r)$, so we can assume $b+r>3$.

Also, we have $gy^{-1}x^{-1} = (a-1\ ,\ s+(b+r)(b+r-1)-b+1\ ,\ b-1)$ with $s+(b+r)(b+r-1)\geq b+1$. So, by corollary 5.3.17, we can assume $a=0$ or $b=0$.

If $a=0$ and $s<b+r$ then we would have:-

$$gz= p_5(a,b,r,s+1),$$

and so, by corollary 5.3.14,

$$g \ {}_{\text{join}(10)}\ gz.$$

Also (as $b+r>3$),

$$gzx^{-1} = gy^{-1}x^{-1}y = y^{-r}x^{-(b+r)}y^{(s+1)}xy^{(b+r-s-1)}x^{(b+r-2)} \in B(n-1),$$

so we may traverse $z,x^{-1},y^{-1}$.

If $a=0$ and $s=b+r$, then we may traverse $x^{-1},y^{-1},x^{-1},y^{-1},x^{-1},y,x,x$.

If $b=0$ and $a+s>r$ then $gy^{-1}x^{-1}=\sigma_6(p_6(1,a-1,r+1-a,a+s-r)) \in S(n+2)$.

If $b=0$ and $a+s\leq r$ then we would have:-

$$gy^{-1}=y^{(r-1)}x^{(a+s)}y^{-1}x^{(r-a-s)}y^{-(r-1)}x^{-(r-a)} \in B(n-1),$$

and so we could simply traverse $y^{-1}, x^{-1}$.

Suppose $g = p_6(a,b,r,s)$

$$= x^{-(b+r+1-a)}y^{s}xy^{(b+r-s)}x^{(b+r)}y^{-r}$$

(with $0<r$, $a\leq b+r$, $0<s\leq b+r$ and $n=\| g \|=4r+2+3b-a$).

Note that $n\leq 4(b+r)+2$, so we can assume $b+r>3$ (if $b+r\leq 3$ then $n\leq 10$ and the distance between any two points of $B(n)$ would be at most 20).

Also $gy^{-1}x^{-1} = (\ a-1\ ,\ s+(b+r)^2-b+1\ ,\ b-1\ )$ with $s+(b+r)^2\geq b-1$. So, by corollary 5.3.17, we may as well assume $a=0$ or $b=0$.

If $a=0$ and $s<b+r$ then we would have:-

$$gz = p_6(a,b,r,s+1),$$

and so, by corollary 5.3.14,

$$g \ {}_{\text{join}(10)}\ gz.$$

Also (as $b+r>3$),

$$gzx^{-1}=gy^{-1}x^{-1}y=y^{-r}x^{-(b+r+1-a)}y^{(s+1)}xy^{(b+r-s-1)}x^{(b+r-1)}\in B(n-1),$$

so we may traverse $z,x^{-1},y^{-1}$.

If $a=0$ and $s=b+r$, then we may traverse $x^{-1},y^{-1},x^{-1},y^{-1},x^{-1},y,x,x$·

If $b=0$ and $a+s>r$, then $gy^{-1}x^{-1}=\sigma_6(p_5(1,a-1,r+2-a,a+s-r))\in S(n+2)$.

If $b=0$ and $a+s\leq r$, then we would have:-

$$gy^{-1}=y^rx^{(a+s)}y^{-1}x^{(r-a-s)}y^{-r}x^{-(r-a)}\in B(n-1),$$

and so we could simply traverse $y^{-1}$, $x^{-1}$.

$\boxed{c\acute{c}=y^{-1}x^{-1}}$

The maximal path length so far is 36, we now post multiply by $c\acute{c}=y^{-1}x$.

Suppose $g=p_2(a,b,r,s)$

$$=x^{(a-r-1)}y^sxy^{(b-s)}x^r \quad (\text{with } 0\leq r<a,\ 0\leq s\leq b \text{ and } n=\|g\|=a+b).$$

We have $gy^{-1}x=(a+1\ ,\ s+br+b-1\ ,\ b-1\ )$. So, by corollary 5.3.17, we can assume $b=0$, but then $s=0$, $g=x^a\in S(a)$ and $gy^{-1}x\in S(a+2)$.

Suppose $g=p_3(a,b,r,s)$

$$=x^{-(r+1)}y^sxy^{(b-s)}x^{(a+r)} \ (\text{with } 0\leq r,\ a+r\leq b,\ 0<s\leq b \text{ and } n=\|g\|=2(r+1)+a+b).$$

We have $gy^{-1}x=(a+1\ ,\ s+b(a+r)\ ,\ b-1\ )$. So, by corollary 5.3.17, we can assume $b=0$, but then $a=r=0$, $n=2$ and the distance between any two points of $B(n)$ is at most 4.

Suppose $g=p_4(a,b,r,s)$

$$=y^{(b+r)}x^{(a-s)}yx^sy^{-(r+1)} \quad (\text{with } 0\leq r,\ b+r\leq a,\ 0<s\leq a \text{ and } n=\|g\|=2(r+1)+a+b).$$

We have $gy^{-1}x=(a+1\ ,\ s+a(b+r)+b-1\ ,\ b-1\ )$, and so, by corollary 5.3.17, we may as well assume $b=0$ .

If $s>r+1$ then $gy^{-1}x=\sigma_3(p_3(1,a+1,r,s-r-1))\in S(n+2)$.

If $s\leq r+1$, then we would have:-

$$gz^{-1}=y^rx^ay^{-(r+1-s)}xy^{-(s-1)}x^{-1}\in B(n),$$

and thus,

$$gz^{-1}x = gy^{-1}xy \in B(n-1),$$

so we could traverse $z^{-1}, x, y^{-1}$.

Suppose $g = p_5(a,b,r,s)$

$$= x^{-(b+r-a)}y^{s}xy^{(b+r-s)}x^{(b+r-1)}y^{-r}$$

(with $0<r$, $a<b+r$, $0<s\leq b+r$ and $n=\| g \|=4r+3b-a$).

We have $gy^{-1}x =( a+1 , s+a(b+r)(b+r-1)+b-1 , b-1 )$, and so, by corollary 5.3.17, we may as well assume $b=0$. We would then have:-

$$gz^{-1} = x^{-(r-a)}y^{(s-1)}xy^{(r-s)}x^{r}y^{-(r-1)}x^{-1} \in B(n),$$

and thus

$$gz^{-1}x = gy^{-1}xy \in B(n-1),$$

so we could traverse $z^{-1}, x, y^{-1}$.

Suppose $g = p_6(a,b,r,s)$

$$= x^{-(b+r+1-a)}y^{s}xy^{(b+r-s)}x^{(b+r)}y^{-r}$$

(with $0<r$, $a\leq b+r$, $0<s\leq b+r$ and $n=\| g \|=4r+2+3b-a$).

We have $gy^{-1}x =( a+1 , s+a(b+r)^2+b-1 , b-1 )$, and so, by corollary 5.3.17, we may as well assume $b=0$. We would then have:-

$$gz^{-1} = x^{-(r-a)}y^{(s-1)}xy^{(r+1-s)}x^{r}y^{-r}x^{-1} \in B(n),$$

and thus

$$gz^{-1}x = gy^{-1}xy \in B(n-1),$$

so we could traverse $z^{-1}, x, y^{-1}$.

$\boxed{c\acute{c}=y^{-1}x}$

The maximal path length so far is 36, we now post multiply by $c\acute{c}=x^{-1}x^{-1}$.

Suppose $g = p_2(a,b,r,s)$

$= x^{(a-r-1)}y^{s}xy^{(b-s)}x^{r}$ (with $0\leq r<a$, $0\leq s\leq b$ and $n=\| g \|=a+b$).

Clearly we can assume $r=0$, but then:-

$$g = x^{(a-1)}y^{s}xy^{(b-s)} \in S(a+b),$$

and so we can also assume $b>s$.

If $a=1$, then $gx^{-1}x^{-1} = \sigma_2(p_3(1,b,0,b-s)) \in S(n+2)$,

so we may assume $a>1$.

Supposing $a+s<2b+2$.

If $b<a-2$, then $gx^{-1}x^{-1} = \sigma_5(p_3(b,a-2,1,2b+2-a-s)) \in S(n+2)$.

If $b=a-2$, then $gx^{-1}x^{-1} = \sigma_5(p_5(b,a-2,1,b-s)) \in S(n+2)$.

If $b>a-2$, then $gx^{-1}x^{-1} = \sigma_5(p_4(b,a-2,1,b-s)) \in S(n+2)$.

So we are left with $a>1$ and $a+s\geq 2b+2$, we would have:-

$$gx^{-1}x^{-1} = x^{(a+s-2b-2)}y^{-1}x^{(2b-s)}y^{(b+1)} \in B(a+b).$$

If $b=1$ we would have $s=0$, because we are, anyway, assuming $b>s$. Thus $a\geq 2b+2=4$, $g = x^{a}y$ and we could traverse $y^{-1},x^{-4},y^{-1},x^{2},y^{2}$.

If $b>1$, then we could traverse $y^{-1},x^{-1},y^{-1},x^{-1},y,x^{-1},y^{-1},x^{-1},y^{-1},x^{2},y^{3}$.

Suppose $g=p_3(a,b,r,s)$

$= x^{-(r+1)}y^{s}xy^{(b-s)}x^{(a+r)}$ (with $0\leq r$, $a+r\leq b$, $0<s\leq b$ and $n=\| g \|=2(r+1)+a+b$).

Apparently the only non trivial situation is $a+r=0$, but then

$$gx^{-1} = y^{(b-s)}x^{-1}y^{s} \in S(n-1),$$

and so we could still traverse $x^{-1},x^{-1}$.

Suppose $g = p_4(a,b,r,s)$

$= y^{(b+r)}x^{(a-s)}yx^{s}y^{-(r+1)}$ (with $0\leq r$, $b+r\leq a$, $0<s\leq a$ and $n=\| g \|=2(r+1)+a+b$).

We have $gx^{-1}x^{-1} = (\, a-2\, ,\, s+b(a-2)+ar\, ,\, b\, )$. So by corollary 5.3.17, we may as well assume $a<2$, but then $n= 2r+2+a+b \leq 3a+2 \leq 5$ (and so the distance between any two points of $B(n)$ would be at most 10).

Suppose $g = p_5(a,b,r,s)$

$= x^{-(b+r-a)}y^{s}xy^{(b+r-s)}x^{(b+r-1)}y^{-r}$

(with $0<r$, $a<b+r$, $0<s\leq b+r$ and $n=\| g \|=4r+3b-a$).

If $a=0$, we would have:-

$$gx^{-1} = y^{-r}x^{-(b+r-1)}y^{(b+r-s)}x^{-1}y^{s}x^{(b+r-1)} \in B(n-1).$$

If a>0 and b≥s, we would have:-

$$gx^{-1} = x^{-(b+r-1-a)}y^{(b-s)}x^{-1}y^{(s+r)}x^{(b+r-1)}y^{-r} \in B(n-1).$$

If a>0 and b<s, we would have:-

$$g \in S(n),$$

$$gx^{-1}y = p_5(a-1,b+1,r-1,s-b) \in S(n).$$

Also, there is certainly no loss of generality in assuming

$$gx^{-1}y\,(y^{-1}x^{-1}) = gx^{-1}x^{-1} \in S(n),$$

and so we would already have proved:-

$$g \;_{\text{join}(12)}\; gx^{-1}y, \;\; gx^{-1}y \;_{\text{join}(36)}\; gx^{-1}x^{-1}.$$

Therefore

$$g \;_{\text{join}(48)}\; gx^{-1}x^{-1}.$$

Suppose $g = p_6(a,b,r,s)$

$$= x^{-(b+r+1-a)}y^{s}xy^{(b+r-s)}x^{(b+r)}y^{-r}$$

(with $0<r$, $a\leq b+r$, $0<s\leq b+r$ and $n=\| g \|=4r+2+3b-a$).

If a=0, we would have:-

$$gx^{-1} = y^{-r}x^{-(b+r+1)}y^{s}xy^{(b+r-s)}x^{(b+r-1)} \in B(n-1).$$

If a>0 and b≥s, we would have:-

$$gx^{-1} = x^{-(b+r-a)}y^{(b-s)}x^{-1}y^{(s+r)}x^{(b+r)}y^{-r} \in B(n-1).$$

If a>0 and b<s, we would have:-

$$g \in S(n),$$

$$gx^{-1}y = p_6(a-1,b+1,r-1,s-b) \in S(n).$$

Also, there is no loss of generality in assuming

$$gx^{-1}y\,(y^{-1}x^{-1}) = gx^{-1}x^{-1} \in S(n),$$

and so we would already have proved:-

$$g \;_{\text{join}(12)}\; gx^{-1}y, \;\; gx^{-1}y \;_{\text{join}(36)}\; gx^{-1}x^{-1}.$$

Therefore

$$g \;_{\text{join}(48)}\; gx^{-1}x^{-1}.$$

$\boxed{c\acute{c}=x^{-1}x^{-1}}$

The maximal path length so far is 48, we now post multiply by $c\acute{c}=xy^{-1}$.

Suppose $g = p_2(a,b,r,s)$

$$= x^{(a-r-1)}y^s xy^{(b-s)}x^r \text{ (with } 0\le r<a,\ 0\le s\le b \text{ and } n=\| g \|=a+b).$$

We have that $gxy^{-1} = ( a+1 , s+br+b ,b-1 )$. So, by corollary 5.3.17, we can assume that $b=0$, but then , as $s\le b$ anyway, so $s=0$ and we would have:-

$$g = x^a \in S(a) \text{ while } gxy^{-1} = x^{(a+1)}y^{-1} \in S(a+2).$$

Suppose $g=p_3(a,b,r,s)$

$$= x^{-(r+1)}y^s xy^{(b-s)}x^{(a+r)} \text{ (with } 0\le r,\ a+r\le b,\ 0<s\le b \text{ and } n=\| g \|=2(r+1)+a+b).$$

We have $gxy^{-1} = ( a+1 , s+b(a+r) ,b-1 )$. So, by corollary 5.3.17, we may as well assume $b=0$ - but $b>0$ anyway.

Suppose $g = p_4(a,b,r,s)$

$$= y^{(b+r)}x^{(a-s)}yx^s y^{-(r+1)} \text{ (with } 0\le r,\ b+r\le a,\ 0<s\le a \text{ and } n=\| g \|=2(r+1)+a+b).$$

We have $gxy^{-1} = ( a+1 , s+b(a+r)+b ,b-1 )$, so, by corollary 5.3.17, we can assume that $b=0$. Then, if $s\le r$, we would have $gx = y^r x^a y^{-(r-s)} xy^{-s} \in B(n-1)$. If $s>r$, we would have $gxy^{-1} = \sigma_6(p_3(1,a+1,r,s-r)) \in S(n+2)$.

Suppose $g = p_5(a,b,r,s)$

$$= x^{-(b+r-a)}y^s xy^{(b+r-s)}x^{(b+r-1)}y^{-r}$$

$$\text{(with } 0<r,\ a<b+r,\ 0<s\le b+r \text{ and } n=\| g \|=4r+3b-a).$$

We have $gxy^{-1} = ( a+1 , s+(b+r)(b+r-1)+b ,b-1 )$. So, by corollary 5.3.17, we can assume that $b=0$, but then $gx = x^{-(b+r-a-1)}y^s xy^{(b+r-s)}x^{(b+r-1)}y^{-r} \in B(n-1)$.

Suppose $g = p_6(a,b,r,s)$

$$= x^{-(b+r+1-a)}y^s xy^{(b+r-s)}x^{(b+r)}y^{-r}$$

$$\text{(with } 0<r,\ a\le b+r,\ 0<s\le b+r \text{ and } n=\| g \|=4r+2+3b-a).$$

We have $gxy^{-1} = ( a+1 , s+(b+r)^2+b ,b-1 )$. So, by corollary 5.3.17, we can assume that b=0, but then $gx = x^{-(b+r-a)}y^{s}xy^{(b+r-s)}x^{(b+r)}y^{-r} \in B(n-1)$.

$\boxed{c\acute{c}=xy^{-1}}$

The maximal path length so far is 48, we now post multiply by $c\acute{c}=x^{-1}y^{-1}$.

Suppose $g = p_2(a,b,r,s)$

$= x^{(a-r-1)}y^{s}xy^{(b-s)}x^{r}$ (with $0\leq r<a$, $0\leq s\leq b$ and $n=\| g \|=a+b$).

Clearly we may assume r=0.

If s>0, we would have:-

$$gz^{-1} = p_2(a,b,r,s-1) \in S(n),$$

also there is no loss of generality in assuming

$$gz^{-1}(y^{-1}x^{-1}) = gx^{-1}y^{-1} \in S(n).$$

So we would already have already proved:-

$$g_{\ join(10)}\ gz^{-1} \text{ and } gz^{-1}{}_{\ join(36)}\ gx^{-1}y^{-1},$$

thus

$$g_{\ join(46)}\ gx^{-1}y^{-1}.$$

We are left with s=0, but then $g = x^{a}y^{b} \in S(a+b)$, so we also assume b>0.

If a>b=1, then we may traverse $y^{-1},x^{-2},y^{-1},x,y$.

If a>b>1, then we may traverse $y^{-1},x^{-1},y^{-1},x^{-1},y^{-1},x,y^{2}$.

If a≤b then n≤2b so we may as well assume b≥2. Then, if a<b, $gx^{-1}y^{-1} = \sigma_5(p_4(b-1,a-1,1,1)) \in S(a+b+2)$. If a=b, then $gx^{-1}y^{-1} = \sigma_5(p_5(b-1,b-1,1,1)) \in S(2b+2)$.

Suppose $g=p_3(a,b,r,s)$

$= x^{-(r+1)}y^{s}xy^{(b-s)}x^{(a+r)}$ (with $0\leq r$, $a+r\leq b$, $0<s\leq b$ and $n=\| g \|=2(r+1)+a+b$).

Clearly, we may assume a+r=0. Then we will have:-

$$g \in S(b+2),\ gz^{-1} = x^{-1}y^{(s-1)}xy^{(b+1-s)} \in B(b+2),$$

and thus

$$gz^{-1}y^{-1} \in B(b+1).$$

So we may traverse $z^{-1},y^{-1},x^{-1}$.

Suppose $g = p_4(a,b,r,s)$

$= y^{(b+r)}x^{(a-s)}yx^{s}y^{-(r+1)}$ (with $0 \leq r$, $b+r \leq a$, $0<s \leq a$ and $n = \| g \| = 2(r+1)+a+b$).

If $s=1$, then:-

$$gz^{-1} = y^{(b+r)}x^{a}y^{-r} \in B(2r+a+b),$$

and so we may traverse $z^{-1}, y^{-1}, x^{-1}$.

If $s>1$, then we would have:-

$$gz^{-1} = p_4(a,b,r,s-1) \in S(n),$$

and, as there is no loss of generality in assuming

$$gz^{-1}(y^{-1}x^{-1}) = gx^{-1}y^{-1} \in S(n),$$

so, we have already proved

$$g \ \text{join}_{(10)} \ gz^{-1} \text{ and } gz^{-1} \ \text{join}_{(36)} \ gx^{-1}y^{-1}.$$

Thus

$$g \ \text{join}_{(46)} \ gx^{-1}y^{-1}.$$

Suppose $g = p_5(a,b,r,s)$

$= x^{-(b+r-a)}y^{s}xy^{(b+r-s)}x^{(b+r-1)}y^{-r}$

(with $0<r$, $a<b+r$, $0<s \leq b+r$ and $n = \| g \| = 4r+3b-a$).

If $s=1$, then we would have:-

$$gz^{-1} = x^{-(b+r-a-1)}y^{(b+r)}x^{(b+r-1)}y^{-r} \in B(4r+3b-a-2),$$

and so we may traverse $z^{-1}, y^{-1}, x^{-1}$.

If $s>1$, then we would have:-

$$gz^{-1} = p_5(a,b,r,s-1) \in S(n),$$

and, as there is no loss of generality in assuming

$$gz^{-1}(y^{-1}x^{-1}) = gx^{-1}y^{-1} \in S(n),$$

so we have already proved

$$g \ \text{join}_{(10)} \ gz^{-1} \text{ and } gz^{-1} \ \text{join}_{(36)} \ gx^{-1}y^{-1}.$$

Thus

$$g \ \text{join}_{(46)} \ gx^{-1}y^{-1}.$$

Suppose $g = p_6(a,b,r,s)$

$$= x^{-(b+r+1-a)}y^s x y^{(b+r-s)} x^{(b+r)} y^{-r}$$

(with $0<r$, $a\leq b+r$, $0<s\leq b+r$ and $n=\| g \|=4r+2+3b-a$).

If $s=1$, then we would have:-

$$gz^{-1} = x^{-(b+r-a)}y^{(b+r)}x^{(b+r)}y^{-r} \in B(4r+3b-a),$$

and so we may traverse $z^{-1},y^{-1},x^{-1}$.

If $s>1$, then we would have:-

$$gz^{-1} = p_6(a,b,r,s-1) \in S(n),$$

and, as there is no loss of generality in assuming

$$gz^{-1}(y^{-1}x^{-1}) = gx^{-1}y^{-1} \in S(n),$$

so we have already proved

$$g \text{ join}_{(10)}\ gz^{-1} \text{ and } gz^{-1} \text{ join}_{(36)}\ gx^{-1}y^{-1}.$$

Thus

$$g \text{ join}_{(46)}\ gx^{-1}y^{-1}.$$

$\boxed{c\hat{c}=x^{-1}y^{-1}}$

The maximal path length so far is 48, we now post multiply by $c\hat{c}=y^{-1}y^{-1}$.

Suppose $g = p_2(a,b,r,s)$

$= x^{(a-r-1)}y^s x y^{(b-s)} x^r$ (with $0\leq r<a$, $0\leq s\leq b$ and $n=\| g \|=a+b$).

We have $gy^{-1}y^{-1} = (\ a\ ,\ s+br\ ,b-2\ )$, so by 5.3.17, we may as well assume $b=0$ or $b=1$.
If $b=1$ then $g\in S(a+1)$ but $gy^{-1}y^{-1} = \sigma_6(p_2(1,a,0,s+r)) \in S(a+3)$.
If $b=0$ then, as $s\leq b$ anyway, $s=0$, so $g = x^a \in S(a)$ and $gy^{-1}y^{-1} \in S(a+2)$.

Suppose $g = p_3(a,b,r,s)$

$= x^{-(r+1)}y^s x y^{(b-s)} x^{(a+r)}$ (with $0\leq r$, $a+r\leq b$, $0<s\leq b$ and $n=\| g \|=2(r+1)+a+b$).

We have $gy^{-1}y^{-1} = (\ a\ ,\ s+b(a+r)\ ,\ b-2\ )$. So by, 5.3.17, we may as well assume $b\leq 1$, but then $n = 2(r+1)+a+b \leq 3b+2 \leq 5$ (so the distance between any two points of $B(n)$ would be at most 10).

Suppose $g = p_4(a,b,r,s)$

$= y^{(b+r)}x^{(a-s)}yx^sy^{-(r+1)}$ (with $0 \leq r$, $b+r \leq a$, $0 < s \leq a$ and $n = \| g \| = 2(r+1)+a+b$).

Note that $n \leq 2(a+1)+a+a = 4a+2$, so we can assume $a \geq 2$. Also $gy^{-1}y^{-1} = ( a , s+a(b+r) , b-2 )$, so, by 5.3.17, we may as well assume $a \geq 2$ and $b \leq 1$.

If $b=0$ and $a \geq r+2$, then $gy^{-1}y^{-1} = \sigma_6(p_3(2,a,r,s)) \in S(n+2)$.

If $b=0$ and $a < r+2$, then, as $b+r \leq a$ anyway, we must have $r=a-1$ or $r=a$.

If $r=a-1$, then $gy^{-1}y^{-1} = \sigma_6(p_5(2,a,1,s)) \in S(n+2)$.

If $r=a$ and $s>1$, then $gy^{-1}y^{-1} = \sigma_6(p_6(2,a,1,s-1)) \in S(n+2)$.

If $r=a$ and $s=1$, then we would have:-

$$g = y^a x^{(a-1)} y x^s y^{-(a+1)} \in S(3a+2),$$

$$gy^{-1}y^{-1} = y^{(a-1)} x^{(a+1)} y^{-(a+1)} x^{-1} \in S(3a+2),$$

and we could traverse $y, x^{-1}, y^{-1}, x, y^{-1}, x, y^{-1}, x^{-1}$.

If $b=1$ and $a>r+1$, then $gy^{-1}y^{-1} = \sigma_6(p_3(1,a,r+1,s)) \in S(n+2)$.

If $b=1$ and $a=r+1$, then $gy^{-1}y^{-1} = \sigma_6(p_5(1,a,1,s)) \in S(n+2)$.

Suppose $g = p_5(a,b,r,s)$

$= x^{-(b+r-a)}y^sxy^{(b+r-s)}x^{(b+r-1)}y^{-r}$

(with $0<r$, $a<b+r$, $0<s \leq b+r$ and $n = \| g \| = 4r+3b-a$).

We have $gy^{-1}y^{-1} = ( a , s+(b+r)(b+r-1) , b-2 )$. So by, 5.3.17, we may as well assume $b \leq 1$.

If $b=0$ and $a+s \leq r$, then we have:-

$$gy^{-1} = y^{(r-1)} x^{(a+s)} y^{-1} x^{(r-a-s)} y^{-(r-1)} x^{-(r-a)} \in B(n-1).$$

If $b=0$, $a+s>r$ and $2r<s+2a$, then we have

$$gy^{-1}y^{-1} = \sigma_6(p_5(2,a,r+1-a,s+2a-2r)) \in S(n+2).$$

If $b=0$, $a+s>r$ and $2r \geq s+2a$, so (as $a+s>r$ and $r=b+r \geq s$ anyway) $s>2$. We would then have:-

$$gz^{-1} = p_5(a,0,r,s-1) \in S(4r-a),$$

and

$$gz^{-1}z^{-1} = y^{(s-2)} x y^{(r+2-s)} x^{(r-1)} y^{-r} x^{-(r-a)} \in B(4r-a),$$

therefore, by 5.3.13,

$$g \ {}_{\text{join}(10)} \ gz^{-1} \text{ and } gz^{-1} \ {}_{\text{join}(10)} \ gz^{-1}z^{-1}.$$

Also, we can see that,

$$gz^{-1}z^{-1}x \in B(4r-a-1), \text{ and so } gz^{-1}z^{-1}xy^{-1} \in B(4r-a).$$

So, as

$$\begin{aligned} gz^{-1}z^{-1}xy^{-1}y^{-1} &= gy^{-1}y^{-1}x \\ &= y^{(r-1)}x^{(s+2a-r)}y^{-1}x^{(2r-s-2a)}y^{-r}x^{-(r-a-1)} \in B(4r-a-1), \end{aligned}$$

we may traverse (the lengthy route of) $z^{-1},z^{-1},x,y^{-1},y^{-1},x^{-1}$.

If $b=1$ then we would have:-

$$g \in S(4r+3-a),$$

and, if $a<r$,

$$gy^{-1}x = p_5(a+1,0,r+1,s) \in S(4r+3-a).$$

or, if $a=r$,

$$gy^{-1}x = p_4(a+1,0,r,s) \in S(4r+3-a).$$

Whichever, there is no loss of generality in assuming

$$gy^{-1}x(x^{-1}y^{-1}) = gy^{-1}y^{-1} \in S(4r+3-a),$$

so we would have already proved

$$g \ {}_{\text{join}(36)} \ gy^{-1}x \text{ and } gy^{-1}x \ {}_{\text{join}(48)} \ gy^{-1}y^{-1},$$

therefore

$$g \ {}_{\text{join}(84)} \ gy^{-1}y^{-1}.$$

Suppose $g = p_6(a,b,r,s)$

$$= x^{-(b+r+1-a)}y^{s}xy^{(b+r-s)}x^{(b+r)}y^{-r}$$

(with $0<r$, $a \le b+r$, $0<s \le b+r$ and $n=\| g \|=4r+2+3b-a$).

We have $gy^{-1}y^{-1} = (a, s+(b+r)^2, b-2)$. So by, 5.3.17, we may as well assume $b \le 1$.

If $b=0$ and $a+s \le r$, then we have

$$gy^{-1} = y^{(r)}x^{(a+s)}y^{-1}x^{(r-a-s)}y^{-r}x^{-(r-a)} \in B(n-1).$$

If $b=0$, $a+s>r$ and $2r+1<s+2a$, then we have

$$gy^{-1}y^{-1} = \sigma_6(p_6(2,a,r+1-a,s+2a-2r-1)) \in S(n+2).$$

If $b=0$, $a+s>r$, $2r+1\geq s+2a$ and $s=1$ (then $a+s>r$ and $r+1\geq a$), so $a=r$. We would have -

$$g\in S(3r+2),$$

and

$$gy^{-1}x=\sigma_6(p_5(1,r+1,0,1)\in S(3r+2).$$

Also, as there is no loss of generality in assuming

$$gy^{-1}x(x^{-1}y)=gy^{-1}y^{-1}\in S(3r+2),$$

so we would have already proved

$$g\ {}_{\mathrm{join}(36)}\ gy^{-1}x \text{ and } gy^{-1}x\ {}_{\mathrm{join}(48)}\ gy^{-1}y^{-1}.$$

Therefore

$$g\ {}_{\mathrm{join}(84)}\ gy^{-1}y^{-1}.$$

If $b=0$, $a+s>r$, $2r+1\geq s+2a$ and $s>1$, then we would have:-

$$gz^{-1}=p_6(a,0,r,s-1)\in S(4r+2-a),$$

$$gz^{-1}z^{-1}=y^{(s-2)}xy^{(r+2-s)}x^{r-1}y^{-r}x^{-(r+1-a)}\in B(4r+2-a),$$

and so, by 5.3.13,

$$g\ {}_{\mathrm{join}(10)}\ gz^{-1} \text{ and } gz^{-1}\ {}_{\mathrm{join}(10)}\ gz^{-1}z^{-1}.$$

Also, we can see that,

$$gz^{-1}z^{-1}x\in B(4r+1-a), \text{ and so } gz^{-1}z^{-1}xy^{-1}\in B(4r+2-a).$$

So, as

$$\begin{aligned} gz^{-1}z^{-1}xy^{-1}y^{-1}&=gy^{-1}y^{-1}x\\ &=y^{(r-1)}x^{(s+2a-r)}y^{-1}x^{(2r+1-s-2a)}y^{-r}x^{-(r-a)}\in B(4r+1-a), \end{aligned}$$

we may traverse $z^{-1},z^{-1},x,y^{-1},y^{-1},x^{-1}$.

If $b=1$ then we would have:-

$$g\in S(4r+5-a),$$

and, if $a\leq r$,

$$gy^{-1}x=p_5(a+1,0,r+1,s)\in S(4r+3-a).$$

or, if $a=r+1$,

$$gy^{-1}x=p_4(a+1,0,r+1,s)\in S(4r+3-a).$$

Whichever, there is no loss of generality in assuming

$$gy^{-1}x(x^{-1}y^{-1}) = gy^{-1}y^{-1} \in S(4r+3-a),$$

so we would have already proved

$$g \ _{\mathrm{join}(36)}\ gy^{-1}x \ \text{ and } \ gy^{-1}x \ _{\mathrm{join}(48)}\ gy^{-1}y^{-1},$$

therefore

$$g \ _{\mathrm{join}(84)}\ gy^{-1}y^{-1}.$$

$\boxed{c\acute{c}=y^{-1}y^{-1} \text{ and } 5.3.18}$

We need to prove (for $\Gamma_C(G)$ to be almost convex) that, for any $g \in G$ and $c, \acute{c} \in C$:

*(ii)* whenever $\| g \| = \| gc\acute{c} \|$, then $g \ _{\mathrm{join}(84)}\ gc\acute{c}$.

By propositions 5.3.16 and 5.3.18, we know that (ii) holds provided $g = p_1(b)$ or $g = p_i(a,b,r,s)$ (for some $2 \leq i \leq 6$ and $a,b,r,s \in \mathbb{N}$).

We now take any $g \in G$ and $c, \acute{c} \in C$ with $\| g \| = \| gc\acute{c} \|$, and prove that $g \ _{\mathrm{join}(84)}\ gc\acute{c}$. By lemma 5.4. 11(ii), we know that

$$g = \sigma_j(p_1(b)) \text{ or } \sigma_j(p_i(a,b,r,s)) \text{ (for some } 1 \leq j \leq 8,\ 2 \leq i \leq 6 \text{ and } a,b,r,s \in \mathbb{N}).$$

As the automorphism $\sigma_j$ simply permutes the generators of C, so $\sigma_j$ and its inverse are norm preserving, and we have:-

$$\begin{aligned} \| \sigma_j^{-1}(g) \| &= \| g \| \\ &= \| gc\acute{c} \| \\ &= \| \sigma_j^{-1}(gc\acute{c}) \| . \end{aligned}$$

Whence:-

$$\| \sigma_j^{-1}(g) \| = \| \sigma_j^{-1}(g)\, \sigma_j^{-1}(c)\, \sigma_j^{-1}(\acute{c}) \| ,$$

with

$$\sigma_j^{-1}(g) = p_1(b) \text{ or } p_i(a,b,r,s),$$

and

$$\sigma_j^{-1}(c),\ \sigma_j^{-1}(\acute{c}) \in C.$$

So, by propositions 5.3.16 and 5.3.18, we will have:-

$$\sigma_j^{-1}(g) \ _{\mathrm{join}(84)}\ \sigma_j^{-1}(g)\, \sigma_j^{-1}(c)\, \sigma_j^{-1}(\acute{c}),$$

whence

$$g \ _{\mathrm{join}(84)}\ gc\acute{c},$$

because $\sigma_j$ is norm preserving.

$\boxed{5.3.1}$

*5.3.19 Corollary.*

With $A=\{x, x^{-1}, y, y^{-1}, z, z^{-1}\}$, $\Gamma_A(G)$ is almost convex.

*Proof:*

We know that $z\in$ centre(G), but, also, $z^{20}=y^5x^4y^{-5}x^{-4}$. Thus, whenever (arbitrary) $g\in G$ is expressed as a product, p, in a minimal (i.e. $\|g\|_A$) number of the generators of A, then there cannot be more than 19 z's or 19 $z^{-1}$'s occuring in p. So, by replacing every z or $z^{-1}$ of p by, respectively, $yxy^{-1}x^{-1}$ or $xyx^{-1}y^{-1}$, we may express g as a product of at most $\|g\|_A+19\times4$ of the generators from C. Whence $\|g\|_A \leq \|g\|_C \leq \|g\|_A+76$, and so, by proposition 5.1.4, $\Gamma_A(G)$ will be almost convex.

$\boxed{5.3.19}$

## *5.4*

## *Abelian by finite groups are almost convex.*

In this section we will give an alternative, and slightly generalized, proof of Cannon's theorem that free abelian by finite groups are almost convex (Cannon).

A discrete group of euclidean isometries (acting on some euclidean space T say) is free abelian by finite. This is the gist of Cannon's (geometrical) proof that these groups are almost convex. The map $\Psi : \Gamma_C(G) \longrightarrow T$ is defined by mapping g to g(0), if $g \in G$, and, by mapping the edge beginning at $g \in G$, and labelled by $c \in C$, to the euclidean segment between g(0) and gc(0). By referring to Benson's 'factorization lemma' (cf (Benson) or lemma 5.4.3 of this thesis), the following 'quasi isometry lemma' can be proved. There are integers N(1) and N(2) so that, if $\rho$ is a path of $\Gamma$, then $|\Psi(\rho)| \leq N(1)|\rho|$ ; also, if $x,y \in \Gamma$, then there is a path $\rho$, of $\Gamma$, which joins x to y, has length $\leq N(2)(1+d(\Psi(x),\Psi(y))$, and which stays (pointwise) within the N(2) neighborhood of the geodesic segment between $\Psi(x)$ and $\Psi(y)$. (The latter is probably the hardest part of the proof.) Then, with reference to Benson's lemma (again), Cannon is able to define convex euclidean polyhedra, P(n), for all $n \in \mathbb{N}$, so as to prove the following 'convexity lemma'. The polyhedra are good approximations to the n-balls of $\Gamma$; good approximations in the sense that there is a number N(3), independent of n, so that $\Psi(B(n))$ lies (pointwise) within the N(3) neighborhood of $\Psi(\Gamma) \cap P(n)$, and P(n) lies (pointwise) within the N(3) neighborhood of $\Psi(B(n))$. By referring to the 'quasi isometry' and 'convexity' lemmas, we can , after about two pages of calculations, prove that $\Gamma_C(G)$ is almost convex (with respect to any (inverse closed) generating set).

As opposed to Cannon's geometric proof, our proof is basically algebraic. We believe it to be the simpler proof, although it does rely on theorem 2.1.5 (finitely generated abelian groups are automatic) and theorem 5.2.9 (llb-automatic groups are almost convex).

*5.4.1 Theorem.*

Let G be an an extension of an abelian group, A, by a finite group, F, of order f. If C is any finite generating set of G, then $\Gamma_C(G)$ is almost convex.

*Proof:*

*Comment.* To avoid some rather clumsy notation, we will be constructing free monoids on several subsets of the underlying set of G. It should be clear from the context when we are taking a product, $p$, (in the set G) to be a product in the group G, or as a product in a free monoid, but, for emphasis, we may sometimes write $\gamma(p)$ to mean that the product is being taken as a group product .

Basically, we prove 5.4.1 by using theorem 2.1.5 to derive an automatic structure for (G,C) with word acceptor W satisfying the hypothesis of theorem 5.2.9 (i.e., we shall prove that (G,C) is llb-automatic).

We let $C^*$ denote the free monoid on the generators C, and assume throughout the proof that the words of $C^*$ are ordered first by the length function $|\ |$, and then lexicalgraphically (i.e., by a ShortLex ordering). If $g\in G$, then we define rep(g) to be the least word, $w$, of $C^*$ such that $\gamma(w)=g$.

We fix a set of right coset representatives of A in G, say $\{r_i\}_{1\le i\le f}$ (with $r_1=1$). Note that:

$$(1)\ \text{if } g_1,\ldots,g_n\in G \text{ with } n>f, \text{ then } g_i\, g_{(i+1)} \cdots g_j\in A \text{ for some } 1\le i\le j\le n.$$

Then, as a trivial consequence of (1), we can see that every word in the set:

$$\mathcal{S}=\{\text{ shortest words } s\in C^* \mid \gamma(s)\in A \text{ but } \gamma(a)\notin A \text{ whenever } a \text{ is a proper subword of } s\}$$

will have length at most f. Thus:-

$$S=\gamma(\mathcal{S}) \text{ and } B=\{\, r_i\, s\, r_i^{-1} \mid 1\le i\le f \text{ and } s\in S\,\}$$

are both finite generating sets of A.

We let $B^*$ denote the free monoid on the generators B. Also, for each $b\in B$, we define:

$$\omega(b)=\{\text{ the least word } s\in\mathcal{S} \text{ such that } b=r_i\gamma(s)r_i^{-1} \text{ for some } 1\le i\le f\,\},$$

and then

$$\Lambda(b)=|\omega(b)|,$$

$$\iota(b)=\{\text{ smallest i such that } b=r_i\,\gamma(\omega(b))r_i^{-1}\,\}.$$

Now we fix any ordering $\le$ of B with the proviso:

$$(2)\ b<\bar{b} \text{ whenever } \iota(b)<\iota(\bar{b}).$$

We remind the reader that $<_\Lambda$ denotes the ordering of the words of $B^*$ first by the length function $\Lambda^*$, and then lexicalgraphically according to the ordering $<$.

By theorem 2.1.5, we know that (A,B) will be automatic with a word acceptor, W(A) say, accepting the (prefix closed) language of the $<_A$least words. Moreover, the proviso (2) on the ordering < guarantees that each word $\ell \in$ lan(W(A)) can be *uniquely* factorized as follows:

$$(3)\quad \ell = (b_{(1,1)}b_{(1,2)}\cdots b_{(1,n_1)})(b_{(2,1)}b_{(2,2)}\cdots b_{(2,n_2)})\cdots(b_{(f,1)}b_{(f,2)}\cdots b_{(f,n_f)})$$

with, for all $1 \le i \le f$ and $1 \le j \le n_i$, $b_{(i,j)} \in B$ and $\iota(b_{(i,j)}) = i$.

Whence, we may define the map $\beta$: lan(W(A)) $\longrightarrow C^*$ by:

$$(4)\quad \beta(\ell) = \mathrm{rep}(r_1)\,\omega(b_{(1,1)})\,\omega(b_{(1,2)})\ldots\omega(b_{(1,n_1)})\,\mathrm{rep}(r_1{}^{-1})$$
$$\mathrm{rep}(r_2)\,\omega(b_{(2,1)})\,\omega(b_{(2,2)})\ldots\omega(b_{(2,n_2)})\,\mathrm{rep}(r_2{}^{-1})\ldots$$
$$\ldots\mathrm{rep}(r_f)\,\omega(b_{(f,1)})\,\omega(b_{(f,2)})\ldots\omega(b_{(f,n_f)})\,\mathrm{rep}(r_f{}^{-1}).$$

Also, we note that, as $\iota(b_{(i,j)}) = i$ (for $1 \le i \le f$ and $1 \le j \le n_i$), so, by the definitions of $\omega(b_{(i,j)})$ and $\iota(b_{(i,j)})$,

$$b_{(i,j)} = r_i\,\gamma(\omega(b_{(i,j)}))r_i{}^{-1},$$

and therefore, by (3) and (4),

$$(5)\quad \gamma(\ell) = \gamma(\beta(\ell)) \text{ for all words } \ell \in \mathrm{lan}(W_{(A)}).$$

*5.4.2 Lemma.*

If $\ell \in$ lan(W(A)) is defined as in (3), then $\beta(\ell)$ cannot be expressed as a shorter word,

$$(6)\quad w = \mathrm{rep}(r_1)\,s_{(1,1)}\,s_{(1,2)}\cdots s_{(1,\bar{n}_1)}\,\mathrm{rep}(r_1{}^{-1})$$
$$\mathrm{rep}(r_2)\,s_{(2,1)}\,s_{(2,2)}\cdots s_{(1,\bar{n}_2)}\,\mathrm{rep}(r_2{}^{-1})\ldots$$
$$\ldots\mathrm{rep}(r_f)\,s_{(f,1)}\,s_{(f,2)}\cdots s_{(f,\bar{n}_f)}\,\mathrm{rep}(r_f{}^{-1})$$

with all the $s_{(i,j)}$ belonging to $S$.

*Proof:*

Suppose we have (6) with $\gamma(\beta(\ell)) = \gamma(w)$. We define, for each $1 \le i \le f$ and $1 \le j \le \bar{n}_i$, $\tilde{b}_{(i,j)} \in B$ by:

$$(7)\quad \tilde{b}_{(i,j)} = r_i\,\gamma(s_{(i,j)})\,r_i{}^{-1}.$$

We will then have:–

$$\Lambda(\tilde{b}_{(i,j)}) = |\omega(\tilde{b}_{(i,j)})|,$$

by the definition of $\Lambda(\tilde{b}_{(i,j)})$. Whence:-

$$(8)\ \ \Lambda(\tilde{b}_{(i,j)}) \leq |s_{(i,j)}|.$$

because $\tilde{b}_{(i,j)} = r_i\, \gamma(s_{(i,j)})\, r_i^{-1}$, while, by definition, $\omega(\tilde{b}_{(i,j)})$ is the least word, $s$, of $S$ such that $\tilde{b}_{(i,j)} = r_\kappa \gamma(s) r_\kappa^{-1}$ for some $1 \leq \kappa \leq f$.

Also, we will have:-

$\gamma(\ell) = \gamma(\beta(\ell)),$

by (5),

$= \gamma(w),$

and thus

$$(9)\ \ \gamma(\ell) = (\tilde{b}_{(1,1)}\tilde{b}_{(1,2)} \ldots \tilde{b}_{(1,\tilde{n}_1)})(\tilde{b}_{(2,1)}\tilde{b}_{(2,2)} \ldots \tilde{b}_{(2,\tilde{n}_2)}) \ldots (\tilde{b}_{(f,1)}\tilde{b}_{(f,2)} \ldots \tilde{b}_{(f,\tilde{n}_f)}).$$

by (6) and (7).

As $\ell \in \mathrm{lan}(W(\Lambda))$, so, by definition, $\ell$ is a least word of $B^*$, with respect to the length function $\Lambda^*$. Therefore, by (9), we must have:-

$$(10)\ \ \Lambda^*(\ell) \leq \sum_{i=1}^{i=f} \sum_{j=1}^{j=\tilde{n}_i} \Lambda(\tilde{b}_{(i,j)}).$$

We now have:-

$$\sum_{i=1}^{i=f} \sum_{j=1}^{j=n_i} |\omega(b_{(i,j)})| = \sum_{i=1}^{i=f} \sum_{j=1}^{j=n_i} \Lambda(b_{(i,j)}),$$

by definition of the $\Lambda(b_{(i\ j)})$,

$$= \Lambda^*(\ell),$$

by definition (3) (of $\ell$),

$$\leq \sum_{i=1}^{i=f} \sum_{j=1}^{j=\tilde{n}_i} \Lambda(\tilde{b}_{(i,j)}),$$

by (10),

$$\leq \sum_{i=1}^{i=f} \sum_{j=1}^{j=n_i} |s_{(i,j)}|.$$

by (8).

Then, by comparing (4) and (6), we can see that the length of $\beta(\ell)$ is no more than the length of the word $w$ (as required).

$\boxed{5.4.2}$

Let us now define:

$$\textit{(11)}\ \ \mathcal{LAN} = \{\ \beta(\ell)\, \mathrm{rep}(r_t) \mid \ \ell \in \mathrm{lan}(W(A)) \text{ and } 1 \leq t \leq f\ \}.$$

Then, as $\gamma(\mathrm{lan}(W(A))) = A$ and $\{r_i\}_{1 \leq i \leq f}$ is a set of right coset representatives of A in G, so $\gamma(\mathcal{LAN}) = G$.

Actually, $\mathcal{LAN}$ will be our candidate for the regular language of the word acceptor of (G,C). We will need to prove that, for any $w \in \mathcal{LAN}$, $|w| - \|\gamma(w)\|_C$ is bounded. This is a corollary of the next lemma which is due to Benson and appears, in a slightly weaker version, in (Cannon) where it is referred to as the 'factorization lemma'.

*5.4.3 Lemma. (Benson)*

Each $g \in G$ can be written as a product

$$(r_1 s_{(1,1)} s_{(1,2)} \cdots s_{(1,n_1)} r_1^{-1})\,(r_2 s_{(2,1)} s_{(2,2)} \cdots s_{(2,n_2)} r_2^{-1}) \ldots (r_f s_{(f,1)} s_{(f,2)} \cdots s_{(f,n_f)} r_f^{-1}) r_t$$

with all the $s_{(i,j)}$ in S, and with the difference between $\|g\|_C$ and $\sum_{i=1}^{i=f} \sum_{j=1}^{j=n_i} \|s_{(i,j)}\|_C$ being bounded.

*Proof:*

A product $p = g_1 g_2 \ldots g_n$ (with the $g_i \in G$) is said to be a *geodesic* product if

$$\|p\|_C = \sum_{i=1}^{i=n} \|g_i\|_C.$$

Let

$$\textit{(12)}\ \ g = c_1 c_2 \ldots c_n$$

be a geodesic product with all the $c_i$ belonging to (the generating set) C (so $n=\|g\|_C$).

Recall that (1) stated that, if $g_1,\dots,g_n \in G$ with $n \ge f$, then $g_i\, g_{(i+1)} \cdots g_j \in A$ for some $1 \le i \le j \le n$. Also, we defined

$\mathcal{S}$={ shortest words $s \in C^*$ | $\gamma(s) \in A$ but $\gamma(a) \notin A$ whenever $a$ is a proper subword of $s$ },

and

$$S=\gamma(\mathcal{S}).$$

We may as well assume that g has norm >f so that, by (1), the $c_i$ of (12) can be grouped together to derive a geodesic factorization of g of the form:

$$\textit{(13)}\ \ p_1w_1p_2w_2 \cdots p_\kappa w_\kappa$$

with all the $p_i$ being products in terms of S, and with at least one $p_i$ of norm >0.

Supposing κ>f then, by (1), there would be some $i \le j < \kappa$ with

$$p_iw_i\, p_{(i+1)}\, w_{(i+1)} \cdots p_jw_j \in A.$$

Then this product would commute with $p_{(i+1)}$, and so we could consolidate $p_j$ with $p_{(j+1)}$ and $w_j$ with $w_{(i+1)}$ to derive a factorization of the form (13), but with κ reduced by one. Repeating this process, if necessary, we can reduce κ to no more than f. In short, given any product of the form (13) we can assume, solely by rearranging terms, that κ≤f.

So let us begin with a geodesic factorization of g of the form (13). Then, if one of the $w_i$ has norm >f, we replace this $w_i$ with a product of the form (13). The composite factorization of g would again be of the form (13), but with the sum of the norms of the abelian factors, $p_i$, strictly increased. We then rearrange the terms of this factorization so that κ≤f. Clearly, by repeating this process a finite number of times, we will derive a geodesic factorization of g of the following form.

$$\textit{(14)}\ \ g = p_1w_1p_2w_2 \cdots p_{(f-1)}w_f$$

where, for all $1 \le i \le f$:

$$p_i = s_{(i,1)}\, s_{(i,2)} \cdots s_{(i,n_i)}$$

is a geodesic product in the generators of S, and

$$\|w_i\|_C \le f.$$

Thus,

$$\textit{(15)}\ \ \|w_1w_2 \ldots w_f\|_C \le f^2,$$

and, as (14) is a geodesic factorization of g, so we have:-

$$(16)\ \ \| g\|_C \text{ and } \sum_{i=1}^{i=f}\sum_{j=1}^{j=n_i} \| s_{(i,j)}\|_C \text{ do not differ by more than } f^2.$$

If we now now define $\sigma(i)$, $1 \leq i \leq f-1$, by $Ar_{\sigma(i)} = Aw_1w_2 \ldots w_i$, then a simple calculation yields:–

$$p_0w_1p_1w_2 \cdots p_{(f-1)}w_f = p_0(r_{\sigma(1)}\, p_1\, r_{\sigma(1)}{}^{-1}) \ldots (r_{\sigma(f-1)}\, p_{(f-1)}\, r_{\sigma(f-1)}{}^{-1})w_1w_2 \cdots w_f .$$

By consolidating the like factors of, and then rearranging the terms of the latter factorization, we derive a factorization of g of the following form.

$$(17)\ \ g = (\, r_1\, s_{(1,1)}s_{(1,2)} \cdots s_{(1,n_1)}\, r_1{}^{-1}\,)\,(\, r_2\, s_{(2,1)}s_{(2,2)} \cdots s_{(2,n_2)}\, r_2{}^{-1}\,) \ldots$$
$$\ldots (\, r_f\, s_{(f,1)}s_{(f,2)} \cdots s_{(f,n_1)}\, r_f{}^{-1}\,)w_1w_2 \cdots w_f$$

where the $s_{(i,j)}$ are now the (possibly) reindexed $s_{(i,j)}$ of the factorization (14).

As the $r_i$ are right coset representatives of A in G, and A is finitely generated by S, we may choose $r_t$ and $\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_n \in S$ so that:–

$$(18)\ \ \bar{s}_1\bar{s}_2 \ldots \bar{s}_n\, r_t = w_1w_2 \cdots w_f$$

with n as small as possible. By choosing n as small as possible we guarantee

$$n \leq \| w_1w_2 \ldots w_f\|_A + \| r_t\|_A ,$$

so, by (15), n is bounded. Then n being bounded (certainly) ⇒

$$(19)\ \ \| \bar{s}_1\|_C + \| \bar{s}_2\|_C + \ldots + \| \bar{s}_n\|_C \text{ is bounded.}$$

So, by (17) and (18), we have:–

$$(20)\ \ g = (\, r_1\bar{s}_1\bar{s}_2 \ldots \bar{s}_n\, s_{(1,1)}s_{(1,2)} \cdots s_{(1,n_1)}\, r_1{}^{-1}\,)\,(\, r_2\, s_{(2,1)}s_{(2,2)} \cdots s_{(2,n_2)}\, r_2{}^{-1}\,) \ldots$$
$$\ldots (\, r_f\, s_{(f,1)}s_{(f,2)} \cdots s_{(f,n_1)}\, r_f{}^{-1}\,)\, r_t$$

with all the $\bar{s}_i$ and $s_{(i,j)}$ (being the reindexed $s_{(i,j)}$ of (14)) belonging to S. Also, by (16) and (19), we see that the difference between

$$\| g\|_C \text{ and } \left( \| \bar{s}_1\|_C + \| \bar{s}_2\|_C + \ldots + \| \bar{s}_n\|_C + \sum_{i=1}^{i=f}\sum_{j=1}^{j=n_i} \| s_{(i,j)}\|_C \right)$$

is bounded (so (20) is the required factorization).

$\boxed{5.4.3}$

*5.4.4 Corollary.*

If $w \in \mathcal{LAN}$, then $|w| - \|\gamma(w)\|_C$ is bounded.

*Proof:*

By the definition of $\mathcal{LAN}$, $w \equiv \beta(\ell)\mathrm{rep}(r_i)$ for some $\ell \in \mathrm{lan}(W(A))$ and $1 \le i \le f$. Then, by lemma 5.4.2, $\beta(\ell)$ will be the shortest word corresponding to $\gamma(\ell)$ of the form:

$$(6)\ \mathrm{rep}(r_1)\, s_{(1,1)}\, s_{(1,2)} \ldots s_{(1,n_1)}\, \mathrm{rep}(r_1^{-1})\, \mathrm{rep}(r_2)\, s_{(2,1)}\, s_{(2,2)} \ldots s_{(1,n_2)}\, \mathrm{rep}(r_2^{-1}) \ldots$$
$$\ldots \mathrm{rep}(r_f)\, s_{(f,1)}\, s_{(f,2)} \ldots s_{(f,n_f)}\, \mathrm{rep}(r_f^{-1})$$

with all the $s_{(i,j)}$ belonging to $\mathcal{S}$.

However, by lemma 5.4.3, we can find a word, $\bar{\ell}$ say, of the form (6), with $\gamma(\bar{\ell}) = \gamma(\ell)$ and $|\bar{\ell}| - \|\gamma(\ell)\|_C$ being bounded. So it must be that $|\bar{\ell}| \ge |\ell| \ge \|\gamma(\ell)\|_C$, and so $|\ell| - \|\gamma(\ell)\|_C$ is also bounded. Whence $|w| - \|\gamma(w)\|_C$ is bounded (as required).

$\boxed{5.4.4}$

We will now begin the task of proving that (G,C) is automatic with a word acceptor accepting $\mathcal{LAN}$. To start with we must prove that $\mathcal{LAN}$ is actually a regular subset of $C^*$.

*5.4.5 Lemma.*

$\mathcal{LAN} = \{\, \beta(\ell)\, \mathrm{rep}(r_i) \mid \ell \in \mathrm{lan}(W(A)) \text{ and } 1 \le i \le f \,\}$ is a regular subset of $C^*$.

*Proof:*

It will suffice to prove that $\beta(\mathrm{lan}(W(A)))$ is a regular subset of $C^*$. We suppose $W(A)$ has transition function $\tau$.

Recall that (the coset representative) $r_1 = 1$. Thus $\mathrm{rep}(r_1) \equiv \mathrm{rep}(r_1^{-1}) \equiv \varepsilon$, and, by defining $r_{(f+1)} = 1$ (so that $\mathrm{rep}(r_{(f+1)}^{-1}) \equiv \varepsilon$), we can easily see, from definition (4), that a word is a prefix of a word in $\beta(\mathrm{lan}(W(A)))$ if and only if it can be factorized (not necessarily uniquely) as follows.

$$(21)\ \mathrm{rep}(r_1)\, \omega(b_{(1,1)})\, \omega(b_{(1,2)}) \ldots \omega(b_{(1,n_1)})\, \mathrm{rep}(r_1^{-1})$$
$$\mathrm{rep}(r_2)\, \omega(b_{(2,1)})\, \omega(b_{(2,2)}) \ldots \omega(b_{(2,n_2)})\, \mathrm{rep}(r_2^{-1}) \ldots$$
$$\ldots \mathrm{rep}(r_i)\, \omega(b_{(i,1)})\, \omega(b_{(i,2)}) \ldots \omega(b_{(i,n_i)})\, p$$

where, for all $1 \leq i \leq f$ and all $1 \leq j \leq n_i$: the $b_{(i\,j)}$ belong to B;

$$\tau_{(A)}(\, b_{(1,1)} b_{(1,2)} \cdots b_{(1,n_1)})(b_{(2,1)} b_{(2,2)} \cdots b_{(2,n_2)}) \ldots (b_{(i,1)} b_{(i,2)} \cdots b_{(i,n_i)})\,) = h$$

is a halt state of W(A); and $p$ is a proper prefix of some $\omega(b)$ ($b \in B$), or a proper prefix of $\mathrm{rep}(r_i^{-1})\mathrm{rep}(r_{(i+1)})$.

We will use W(A) to construct a (partial) non-deterministic automaton, W, which has language $\beta(\mathrm{lan}(W(A)))$ (we refer the reader to the definition and terminology of a non-deterministic finite state automaton on page 7).

The states of W will consist of all those triples

$$(i,q,p)$$

$$\text{where} \begin{cases} 1 \leq i \leq f. \\ q \text{ is a state of } W(A). \\ p \text{ is a proper prefix of some } s \in \mathcal{S} \text{ or a proper prefix of } \mathrm{rep}(r_i{}^{1})\mathrm{rep}(r_{(i+1)}). \end{cases}$$

We wish to define W so as to satisfy the following hypothesis.

*(22)* $w \in C^*$ can be factorized as (21) if and only if W has a path of arrows with label $w$ and target $(i,h,p)$.

This is the description of W. We let W have start state $(1,q_0,\varepsilon)$ where $q_0$ is the start state of W(A). Then, for all $1 \leq i \leq f$, all $c \in C$, and all states q of W(A) :

W has arrow:

$$(\,(i,q,p)\,,\,c\,,\,(i,q,pc)\,)$$

if and only if $pc$ is a proper prefix of some $s \in \mathcal{S}$ or a proper prefix of $\mathrm{rep}(r_i^{-1})\mathrm{rep}(r_{(i+1)})$.

W has arrow:

$$(\,(i,q,p)\,,\,c\,,\,(i+1,q,\varepsilon)\,)$$

if and only if $pc = \mathrm{rep}(r_i^{-1})\mathrm{rep}(r_{(i+1)})$.

W has arrow:

$$(\,(i,q,p)\,,\,c\,,\,(i,\bar{q},\varepsilon)\,)$$

if and only if $pc = s \in \mathcal{S}$, and there is a transition $\tau(q,b) = \bar{q}$ with $\omega(b) = s$ and $\iota(b) = i$.

We will want the halt states of W to be all those states $(f+1,h,\varepsilon)$ where h is a halt state of W(A).

The construction of W is reasonably self-explanatory. A rigorous proof of hypothesis (22) (by induction on the length of $w$) is straightforward, but would be a laborious formality and is best omitted.

$\boxed{5.4.5}$

*5.4.6 Lemma.*

(G,C) is automatic with word acceptor W (accepting $\mathcal{LAN}$).

*Proof:*

Let c be an arbitrary generator of C and recall that we defined the set of word differences of c to be:

$$\left\{ \gamma(w_0(1,r))^{-1}\gamma(w_1(1,r)) \mid \text{ for all } w_0, w_1 \in \text{lan}(W) \text{ satisfying } \gamma(w_0 c)=\gamma(w_1), \text{ and all } r \in \mathbb{N} \right\}.$$

Also, by theorem 2.1.3, to prove that (G,C) is automatic with word acceptor W, it suffices to prove that the set of word differences of (all such) c is finite, i.e., we want to prove that, whenever

$$\textit{(23)}\quad w_0, w_1 \in \mathcal{LAN},\ \gamma(w_0 c)=\gamma(w_1) \text{ and } 0 \le r \in \mathbb{N},$$

then $\| \gamma(w_0(1,r))^{-1}\gamma(w_1(1,r)) \|_C$ is bounded (independently of $w_0$, $w_1$ or r).
So let us suppose (23) holds, and note that there is no loss of generality in assuming that $|w_0| \ge |w_1|$, r.

By the definition of $\mathcal{LAN}$, we will have $w_0 = \beta(\ell_0)\text{rep}(r_{t_0})$ and $w_1 = \beta(\ell_1)\text{rep}(r_{t_1})$ for some $\ell_0, \ell_1 \in \text{lan}(W(A))$, and $1 \le t_0, t_1 \le f$. Also, recall that the alphabet of W(A) is B, and, by definition (3), any word, $\ell$ of (prefix closed) lan(W(A)) can be uniquely factorized as

$$\ell = (b_{(1,1)}b_{(1,2)}\cdots b_{(1,n_1)})(b_{(2,1)}b_{(2,2)}\cdots b_{(2,n_2)})\ldots(b_{(f,1)}b_{(f,2)}\cdots b_{(f,n_f)})$$

for some $b_{(i,j)} \in B$ with $\iota(b_{(i,j)})=i$ for all $1 \le i \le f$ and $1 \le j \le n_i$. Thus, we were able to define the map $\beta$: lan(W(A)) $\longrightarrow C^*$ by:

$$\begin{aligned}\beta(\ell) = \text{rep}(r_1)\, \omega(b_{(1,1)})\, \omega(b_{(1,2)})\ldots\omega(b_{(1,n_1)})\, \text{rep}(r_1^{-1})\\ \text{rep}(r_2)\, \omega(b_{(2,1)})\, \omega(b_{(2,2)})\ldots\omega(b_{(2,n_2)})\, \text{rep}(r_2^{-1})\ldots\\ \ldots\text{rep}(r_f)\, \omega(b_{(f,1)})\, \omega(b_{(f,2)})\ldots\omega(b_{(f,n_f)})\, \text{rep}(r_f^{-1}).\end{aligned}$$

By (23) we have:-

$$\gamma(\beta(\ell_0)\mathrm{rep}(r_{\ell_0})c)=\gamma(\beta(\ell_1)\mathrm{rep}(r_{\ell_1})),$$

i.e.,

$$\gamma(\beta(\ell_0)r_{\ell_0}c\, r_{\ell_1}{}^{-1})=\gamma(\beta(\ell_1)),$$

whence:-

$$\textit{(26)}\ \ \gamma(\ell_0)r_{\ell_0}c\, r_{\ell_1}{}^{-1}=\gamma(\ell_1),$$

because, by (5), $\gamma(\ell)=\gamma(\beta(\ell))$ for all words $\ell\in \mathrm{lan}(W(A))$.

As, A is automatic with word acceptor W(A), so, by (26) and corollary 2.1.4, we know that $\|\gamma(\ell_0(1,\kappa))^{-1}\gamma(\ell_1(1,\kappa))\|_B$ will be bounded (independently of $\ell_0$, $\ell_1$ or $\kappa\in\mathbb{N}$). Because $\gamma(\ell_0(1,\kappa))=\gamma(\beta(\ell_0(1,\kappa)))$ and $\gamma(\ell_1(1,\kappa))=\gamma(\beta(\ell_1(1,\kappa)))$, so $\|\gamma(\beta(\ell_0(1,\kappa)))^{-1}\gamma(\beta(\ell_1(1,\kappa)))\|_B$ is bounded, whence:-

$$\textit{(27)}\ \ \|\gamma(\beta(\ell_0(1,\kappa)))^{-1}\gamma(\beta(\ell_1(1,\kappa)))\|_C \text{ is bounded}$$

(independently of $\ell_0$, $\ell_1$ or $\kappa\in\mathbb{N}$).

Now, by the definition of $\beta$, we can see that $\kappa\in\mathbb{N}$ can be chosen so that $|\beta(\ell_0(1,\kappa))|$ and r differ by at most:

$$\textit{(28)}\ \ \max(|\omega(b)|)_{b\in B}+\sum_{i=1}^{i=f}|\mathrm{rep}(r_i)|+|\mathrm{rep}(r_i^{-1})|\ .$$

However, for (27) to hold, we must have the difference between $|\beta(\ell_0(1,\kappa))|$ and $|\beta(\ell_1(1,\kappa))|$ bounded.So, also, the difference between $|\beta(\ell_1(1,\kappa))|$ and r will be bounded.

We finish by noting that, by the definition of $\beta$, $\beta(\ell_0(1,\kappa))$ and $u_0=\beta(\ell_0)\mathrm{rep}(r_{\ell_0})$ have a common prefix of length bounded to $|\beta(\ell_0(1,\kappa))|$ by at most (28). Thus, as the difference between $\beta(\ell_0(1,\kappa))$ and r is, anyway, bounded, so $\beta(\ell_0(1,\kappa))$ and $(\beta(\ell_0)\mathrm{rep}(r_{\ell_0}))(1,r)$ have a common prefix of length bounded to $|\beta(\ell_0(1,\kappa))|$. Similarly, $\beta(\ell_1(1,\kappa))$ and $(\beta(\ell_1)\mathrm{rep}(r_{\ell_1}))(1,r)$ have a common prefix of length bounded to $|\beta(\ell_1(1,\kappa))|$. Whence, by (27), $\|\gamma((\beta(\ell_0)\mathrm{rep}(r_{\ell_0}))(1,r))^{-1}\gamma((\beta(\ell_1)\mathrm{rep}(r_{\ell_1}))(1,r))\|_C$ is bounded (as required).

$\boxed{5.4.6}$

By 5.4.4 and 5.4.6, (G,C) is llb-automatic (with word acceptor W), so, by theorem 5.2.9, we know that $\Gamma_C(G)$ is almost convex.

$\boxed{5.4.1}$

# *Bibliography*

Bachmair, Dershowitz.

L. Bachmair, N. Dershowitz. *Critical Pair Criteria for the Knuth-Bendix Completion procedure* Unpublished Manuscript, Dept. of Computer Science of Illinois at Ulbana-Champain, also SYMSAC 1989

Benson.

M.L. Benson. *Growth Series of finite extensions of $\mathbb{Z}^n$ are rational.* Inventiones Mathematicae, vol 73 (1983), pp 251-269.

Book.

R.V Book. *Thue Systems as Rewriting Systems.* J. Symbolic Computation, vol 3 (Feb/April 1987), pp 39-68.

Bauer, Otto.

G. Brauer, F. Otto. *Finite complete rewriting systems and the complexity of the word problem.* Acta Inf., vol 21 (1984), pp 521-540.

Book, O' Dunlaing.

R.V. Book, C.P. Ó' Dunlaing. *Testing for the Church-Rosser Property.* (Note) Theoretical Computer Science, vol 16 (1981), pp 223-229.

Buchberger.

B. Buchberger. *History and Basic Features of the Critical Pair/Completion Procedure* J. Symbolic Computation, vol 3 (Feb/April 1987), pp 3-38.

Cannon.

J.W. Cannon. *Almost Convex Groups.* 1984 preprint.

CEHPT.

J.W. Cannon, D.B.A. Epstein, D.F. Holt, M.S. Paterson, W.P. Thurston. *Word Processing and Group Theory.* University of Minnesota Supercomputer Institute Research Report UMSI 91/57, 1991. Alternatively:

D.B.A Epstein, J.W. Cannon, D.F. Holt, S. Levy, M.S. Paterson, W.P. Thurston. *Word Processing* *in Groups.* Jones and Bartlett, Boston 1992.

Le Chenadec.

P. Le Chenadec. *Canonical Forms in Finitely Presented Algebras.* Research Notes in Theoretical Computer Science, Pitman 1989.

Alternatively:

P. Le Chenadec. *A Catalouge of Complete Group Presentations.* J. Symbolic Computation, vol 2 (1986), pp 363-381.

Epstein, Holt, Rees.

D.B.A Epstein, D.F. Holt, S. Rees. *The use of Knuth-Bendix methods to solve the word problem in automatic groups.* University of Warwick preprint 1989, to appear in Journal of Symbolic Computation.

Gilman.

R. Gilman. *Presentations of Groups and Monoids.* Journal of Algebra, vol 57 (1979), pp 544-554.

Gilman, R. *Enumerating Infinitely Many Cosets.* Computational Group Theory - Proceedings of the London Mathematical Society Symposium on Computational Group Theory, edited by D. Atkinson. Acedemic Press 1984, pp 51-55.

Groves, Smith.

J.R.J. Groves, G.C. Smith. *Rewriting Systems and Soluble Groups.* Bath Computer Science Technical Report 89-90.

Hayashi..

C. Hayashi. *The Word Problem for Groups with Regular Relations (Improvement of the Knuth-Bendix Algorithm).*1991 preprint.

Huet.

G. Huet. *A Complete proof of the Knuth Bendix Algorithm.* J. Computer Systems Science, vol 23 (1981), pp 11-21.

Jantzen.

M. Jantzen. *Confluent string rewriting and congruences.* Bull. EATCS vol 28 (1986), pp 52-72.

Kapur, Musser, Narendon.

D. Kapur, D.R. Musser, P. Narendran. *Only Prime Superpositions need to be considered in the Knuth-Bendix Completion Procedure.* J. of Symbolic Computation, vol 6 (1988), pp 19-36.

Kapur, Narendon.

[1] D. Kapur, P. Narendon. *A finite Thue System with decidable word problem and without equivalent finite canonical system.* (Note) Theoretical Computer Science, vol 35 (1985), pp 337-344.

[2] D. Kapur, P. Narendon. *The Knuth-Bendix Procedure and Thue Systems.* SIAM, J. Computing, vol 14 (1985), pp 1052-1072.

Kernighan, Ritchie.

B.W. Kernighan, D.M. Ritchie. *The C Programming Language, 2nd edition.* Prentice Hall Software Series 1988.

Kfoury, Moll, Arbib.

A.J. Kfoury, R.N. Moll, M.A. Arbib. *A Programming Approach to Computability.* Texts and Monographs in Computer Science, Springer Verlag 1983, Chapter 9.

Knuth, Bendix.

D. Knuth, P.G. Bendix. *Simple Word Problems in Universal Algebras* Computational Problems in Abstract Algebra, ed. Leech, J. Oxford:Pergamon Press (1970), pp 263-269.

Lyndon, Schupp.

R.C. Lyndon, P.E. Schupp. *Combinatorial Group Theory.* Springer Verlag 1976.

Ó' Dunlaing

C.P. Ó' Dunlaing. *Infinite Monadic Thue Systems.* Theoretical Computer Science, vol 25, pp 171-192.

Rayward Smith.

V.J. Rayward Smith. *A 1st course in formal language theory.* Blackwell Scientific Publications Computer Science Texts (1983), chapter 3.

Salomaa.

M. Salomaa. *Formal languages.* (A)ssociation for (C)omputing (M)achinary Inc. Monograph Series, Academic Press 1973, Chapter 4.

Squier.

C. Squier. *Word Problems and a Homological Finiteness Condition for Monoids.* Journal of Pure and Applied Algebra, vol 49 (1987), pp 201-217.

Winkler, Buchberger.

F. Winkler, B. Buchberger. *A criterion for eliminating unnecessary reductions in the Knuth-Bendix algorithm.* Proc. Coll. on Algebra, Combinatorics and Logic in Computer Scienece, Györ, Hungary (1983).