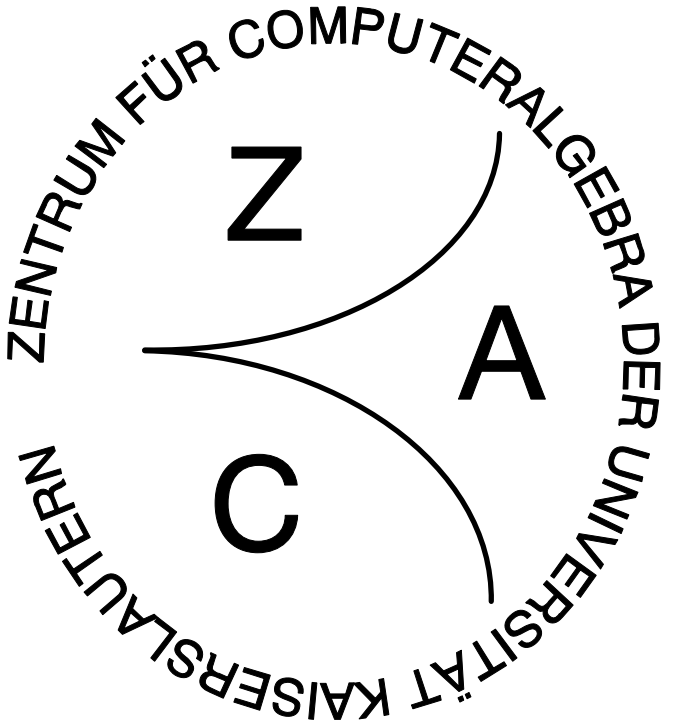
UNIVERSITÄT KAISERSLAUTERN
Zentrum für Computeralgebra

REPORTS ON COMPUTER ALGEBRA
NO. 23



Observations on coset enumeration

by

Birgit Reinert

Nov 1998

The Zentrum für Computeralgebra (Centre for Computer Algebra) at the University of Kaiserslautern was founded in June 1993 by the Ministerium für Wissenschaft und Weiterbildung in Rheinland-Pfalz (Ministry of Science and Education of the state of Rheinland-Pfalz). The centre is a scientific institution of the departments of **Mathematics, Computer Science, and Electrical Engineering** at the University of Kaiserslautern.
The goals of the centre are to advance and to support the use of Computer Algebra in industry, research, and teaching. More concrete goals of the centre include

- the development, integration, and use of software for Computer Algebra
- the development of curricula in Computer Algebra under special consideration of interdisciplinary aspects
- the realisation of seminars about Computer Algebra
- the cooperation with other centres and institutions which have similar goals

The present coordinator of the Reports on Computer Algebra is:
Olaf Bachmann (email: `obachman@mathematik.uni-kl.de`)

Zentrum für Computeralgebra
c/o Prof. Dr. G.-M. Greuel, FB Mathematik
Erwin-Schrödinger-Strasse
D-67663 Kaiserslautern; Germany
Phone: 49 - 631/205-2850 Fax: 49 - 631/205-5052
email: `greuel@mathematik.uni-kl.de`
URL: `http://www.mathematik.uni-kl.de/~zca/`

Observations on Coset Enumeration

Birgit Reinert
Fachbereich Informatik
Universität Kaiserslautern
67663 Kaiserslautern
Germany
reinert@informatik.uni-kl.de

Abstract

Todd and Coxeter's method for enumerating cosets of finitely generated subgroups in finitely presented groups (abbreviated by Tc here) is one famous method from combinatorial group theory for studying the subgroup problem. Since prefix string rewriting is also an appropriate method to study this problem, prefix string rewriting methods have been compared to Tc. We recall and compare two of them briefly, one by Kuhn and Madlener [4] and one by Sims [15]. A new approach using prefix string rewriting in free groups is derived from the algebraic method presented by Reinert, Mora and Madlener in [14] which directly emulates Tc. It is extended to free monoids and an algebraic characterization for the "cosets" enumerated in this setting is provided.

Keywords. coset enumeration, subgroup problem, prefix string rewriting, Gröbner bases in monoid and group rings.

1 Introduction

A group $\mathcal{G}$ is called **finitely presented** if there is a finite set of **generators** Σ and a finite set of **relators** R such that $\mathcal{G}$ is isomorphic to the quotient of the free group generated by Σ modulo the congruence generated by R. Let $\bar{\Sigma} = \Sigma \cup \Sigma^{-1}$ where $\Sigma^{-1} = \{a^{-1} \mid a \in \Sigma\}$ denotes the set of formal inverses for the generators. The group elements then are represented as words on $\bar{\Sigma}$ with the empty word λ representing the one.
In 1911 Dehn stated decision problems for groups: The **word problem** for a group is to decide whether two representations describe the same group element. The **subgroup problem** for a group is to decide for a group element and a finitely generated subgroup of the group whether the element is in fact a member of the subgroup.
Both problems are undecidable in general, but become decidable when restricted to special classes of groups. For finitely generated free groups the word problem can be solved by free reduction, i.e. by deleting occurrences of subwords of the form aa^{-1} and $a^{-1}a$ for $a \in \Sigma$. The subgroup problem can be solved using Nielsen reduced sets due to the fact that there is a lot of crucial information on the maximal parts of words which can cancel each other when multiplying generating elements of subgroups. A well established procedure for dealing with both problems in the case of *arbitrary* finitely presented groups is the Todd-Coxeter coset enumeration method (Tc): Given a set of defining relators for the group $\mathcal{G}$ and a set of generators of the subgroup $\mathcal{H}$ (as words in the generators of $\mathcal{G}$) Tc enumerates the cosets of $\mathcal{H}$ in $\mathcal{G}$. Of course this process can only stop in case $\mathcal{H}$ has finite index in $\mathcal{G}$ and then Tc also provides the coset table, i.e. all multiples of cosets by generators. Now given a word w in the generators of $\mathcal{G}$ we have that $w \in \mathcal{H}$ if and only if w is in the coset of the identity. Hence Tc provides a semi-decision procedure by determining, while enumerating cosets, whether w is in one of the cosets enumerated so far, and answering "yes" in case it is in the coset of the identity. It is obvious that the answer "no" can only be given in case the procedure terminates, since as long as more cosets are enumerated there is the possibility of cosets collapsing, i.e., even if w is found in a coset which is not the identity it might later on be derived that the coset coincides with the coset of the identity. Notice that when choosing the trivial group as the subgroup $\mathcal{H}$, Tc in fact enumerates all elements of the group $\mathcal{G}$ and terminates if and only if $\mathcal{G}$ is finite.
Group presentations can be interpreted as string rewriting systems and this

field is well studied. Its most important procedure is due to Knuth and Bendix and allows computing convergent[1] presentations for groups. In case such a presentation is additionally finite it can be used to compute unique normal forms for the group elements and hence to decide the word problem for the group. The advantage is that this method is often still applicable to infinite groups. For an overview see e.g. [2] and [5].

The presentation of a finitely generated free group in terms of the trivial relators can be interpreted as a convergent string rewriting system and free reduction is exactly reduction using this string rewriting system.

In [1] it is outlined how Tc and Kb are related for the special case of the *trivial* subgroup, i.e. for *finite* groups: A modified version of Tc is presented, which represents the cosets by appropriate words on $\bar{\Sigma}$ and uses a certain strategy to replace cosets when new equations are obtained. On termination the output of Kb is a subset of the rules corresponding to the equations generated by the modified version of Tc. What now are the essential differences between Tc and Kb in this case? If Tc terminates so will a specialized version of Kb but the converse does not hold. This difference in behaviour is due to the fact that Tc, when viewed as a rewriting procedure, does not apply ordinary string rewriting but *prefix string rewriting*[2]. Now in case the index is not finite, Tc will not terminate although the set of rules corresponding to the equations generated by Tc so far might include a finite convergent presentation of the group.

In [9] a similar idea is applied to finite monoids: Given a string rewriting system, the procedure terminates if and only if the monoid is finite and then yields the multiplication table of the monoid.

Variants of prefix rewriting have a long tradition when studying subgroups using string rewriting techniques. However, there are two main differences in these approaches: While Kuhn et. al. [4, 5] require the group to have a convergent presentation, Sims [15] like Tc allows *any* presentation. The output gained by the prefix string rewriting completion techniques in [4, 5] is a *description* of cosets of the subgroup in the group while Tc *enumerates* cosets of the subgroup generated by the original subgroup generators and the normal closure of the relators in the corresponding free group. This difference explains why the former prefix approach can also handle cases where the

[1] Convergent presentations for groups are string rewriting systems which are terminating and confluent.

[2] A rule $\ell \longrightarrow r$ can be used to prefix (string) rewrite a word u if $u \equiv \ell w$, i.e. ℓ is a prefix of u as a word. u is then replaced by rw.

subgroup has infinite index, but a "nice" finite description for the cosets can be found, e.g. by an automaton. For a convergent group presentation Sims' approach [15] is equivalent to the one by Kuhn et. al. [4, 5]. Else it additionally computes a convergent presentation for the group and hence, contrary to Tc, will not always terminate for finite index. For the special case where the group completion diverges while the index is finite, additional knowledge has to be used to determine that the completion process can be stopped (see Section 3.10 in [15] for more details). We overcome this problem by translating the algebraic characterization of Tc as presented by Reinert, Mora and Madlener in [14] into a prefix string rewriting procedure which can be generalized to the case of monoids.
The paper is organized as follows: First we give short introductions to string rewriting theory, the subgroup problem and the Todd-Coxeter coset enumeration method. Then in Section 5 we outline the approach of Kuhn and Madlener for studying the subgroup problem using prefix string rewriting methods and its connections to Tc by giving an example. In Section 6 we sketch Sims' simulation of prefix string rewriting using ordinary string rewriting and compare his results for the subgroup problem to the ones by Kuhn and Madlener, and Tc. Additional knowledge has to be incorporated into the completion procedure to make it terminating for *all* finite index situations. In Section 7 we give a procedure where this is no longer necessary. Finally in Section 8 we generalize this procedure to free monoids and give an algebraic characterization of the structure it computes. It turns out that we exactly get the "blocks" described by Neumann in [12] as a generalization for cosets of submonoids in monoids. These results can be compared to the more general work of Labonté [6] and Linton [7].
Acknowledgments. The author thanks Klaus Madlener, Teo Mora and Steve Linton for valuable discussions on some ideas of this paper.

2 String Rewriting Methods

First we give some basic definitions for string rewriting systems (for a general reference of the terms and techniques described here see [2]). A **string rewriting system** T over a finite alphabet Γ is a subset of $\Gamma^* \times \Gamma^*$. The elements $\ell \longrightarrow r$ of T are called **rules**. The **single-step reduction relation** on Γ^* induced by T is defined as follows: For any u, v in Γ^*, $u \longrightarrow_T v$ if and only if there exist x, y in Γ^* and $\ell \longrightarrow r$ in T such that $u \equiv x\ell y$

and $v \equiv xry$. v is then called a proper descendant of u. The reflexive transitive symmetric closure is denoted by $\overset{*}{\longleftrightarrow}_T$. If $u \overset{*}{\longrightarrow}_T v$ holds then one says that u **reduces** to v. In case u has no proper descendant u is called **irreducible**. An irreducible descendant of u is called a **T-normal form**. Such normal forms need neither exist nor be unique. The reduction relation $\longrightarrow_T$ is called **Noetherian** or **terminating** if and only if there is no infinite chain $u \longrightarrow_T v_1 \longrightarrow_T v_2 \longrightarrow_T \ldots$. In this case normal forms always exist. It is called **confluent** if for all u, v, w in Γ^*, $u \overset{*}{\longrightarrow}_T v$ and $u \overset{*}{\longrightarrow}_T w$ imply the existence of z in Γ^* such that $v \overset{*}{\longrightarrow}_T z$ and $w \overset{*}{\longrightarrow}_T z$. Then normal forms are unique in case they exist. A string rewriting system is called **convergent** or **complete** if it is both, Noetherian and confluent, i.e., unique normal forms exist. By Newman's lemma we know that under the hypothesis that a reduction relation is Noetherian, a string rewriting system is confluent if and only if it is locally confluent, i.e., for all u, v, w in Γ^*, $u \longrightarrow_T v$ and $u \longrightarrow_T w$ imply the existence of z in Γ^* such that $v \overset{*}{\longrightarrow}_T z$ and $w \overset{*}{\longrightarrow}_T z$. For finite string rewriting systems the global property of being locally confluent can be localized to enable a finite confluence test by standard critical pair sets. The process of trying to turn a Noetherian string rewriting system into a convergent one by resolving the not locally confluent situations is called completion and is performed by the Knuth-Bendix completion procedure (abbreviated by KB) which is based on resolving critical pairs. Since the word problem for string rewriting systems is undecidable, such a completion procedure in general will not terminate. Nevertheless, using a fair[3] strategy, it always enumerates a convergent string rewriting system presenting the same monoid as the input system.

Henceforth in this paper, we will only consider string rewriting systems (Γ, T) such that for $\ell \longrightarrow r \in T$ we have $\ell > r$ for some well-founded admissible ordering $>$ on Γ^*. This ordering will be the length-lexicographic in the examples given here.

Now any group presentation in terms of generators Σ and relators R gives rise to a string rewriting system by setting $\Gamma = \bar{\Sigma}$ and $T = T_I \cup T_R$ where $T_I = \{aa^{-1} \longrightarrow \lambda, a^{-1}a \longrightarrow \lambda \mid a \in \Sigma\}$ and $T_R = \{r \longrightarrow \lambda \mid r \in R\}$. Running KB on this input results in a (possibly infinite) completion of the presentation with respect to the chosen completion ordering on $\bar{\Sigma}^*$. If the resulting convergent system is finite many questions concerning the group

[3]A fair strategy will ensure that all elements of the critical pair test sets are considered at some time by the procedure.

can be answered (see e.g. [2]).
Another specialized string rewriting technique used to study subgroups of groups is **prefix string rewriting**. Any string rewriting system (Γ, T) can be interpreted as a prefix string rewriting system with a single-step reduction relation on Γ^* induced by T as follows: For any u, v in Γ^*, $u \longrightarrow_T^{\mathrm{p}} v$ if and only if there exist y in Γ^* and $\ell \longrightarrow r$ in T such that $u \equiv \ell y$ and $v \equiv ry$. The notions irreducible, T-normal form, Noetherian, confluent and convergent with respect to prefix string rewriting carry over naturally. A completion procedure with respect to prefix string rewriting is nothing else than an interreduction of the respective set of rules by prefix rewriting and in contrary to KB always terminates for the finite string rewriting systems we are looking at in this paper.

3 The Subgroup Problem

This section outlines the subgroup problem for groups and its connections to string rewriting techniques.

Definition 1 *Given a subgroup $\mathcal{H}$ of a group $\mathcal{G}$ the* **generalized word problem** *or the* **subgroup problem** *for $\mathcal{H}$ is to determine, given $w \in \mathcal{G}$, whether $w \in \mathcal{H}$.*

For a finite subset S of a group $\mathcal{G}$ and $S^{-1} = \{s^{-1} \mid s \in S\}$ let $\langle S \rangle = \{s_1 \circ \ldots \circ s_n \mid n \in \mathbb{N}, s_i \in S \cup S^{-1}\}$ denote the subgroup generated by S. A subgroup $\mathcal{H}$ of a group $\mathcal{G}$ is called **finitely generated** if there exists a finite subset S of $\mathcal{G}$ such that $\mathcal{H} = \langle S \rangle$. We say a group $\mathcal{G}$ has solvable generalized word problem if for every finite subset S of $\mathcal{G}$ the subgroup problem for $\langle S \rangle$ is decidable.
The word problem for a group $\mathcal{G}$ is just the generalized word problem for the trivial subgroup in $\mathcal{G}$ since $u = v$ holds in $\mathcal{G}$ if and only if $u \circ v^{-1} = \lambda$ holds in $\mathcal{G}$, i.e. $u \circ v^{-1} \in \langle \lambda \rangle$. Thus the existence of a group with undecidable word problem yields undecidability for the generalized word problem for this group as well. On the other hand, decidable word problem for a group does not imply decidable generalized word problem (for an overview on various decision problems for groups see e.g. [10]).
Subgroups of groups can be characterized by one-sided congruences on the group. In the following we restrict ourselves to the case of right congruences

(left congruences can be introduced in a similar fashion). Let $\mathcal{H}$ be a subgroup of a group $\mathcal{G}$. Then for $u, v \in \mathcal{G}$ we can define

$$u \sim_{\mathcal{H}} v \text{ if and only if } \mathcal{H}u = \mathcal{H}v$$

where $\mathcal{H}u = \{g \circ u \mid g \in \mathcal{H}\}$. It is easy to prove that $\sim_{\mathcal{H}}$ is a right congruence. The subgroup $\mathcal{H}$ itself is the congruence class of λ. This right congruence is a congruence if and only if $\mathcal{H}$ is a normal subgroup.
Let us take a look at how the right congruence of the subgroup $\mathcal{H}$ generated by S in the group $\mathcal{G}$ presented by $(\bar{\Sigma}, T = T_I \cup T_R)$ can be described using string rewriting techniques. To S we associate the set of rules $T_S = \{s \longrightarrow \lambda \mid s \in S\}$. We study the rewriting relation induced by the combined reduction relation $\Longrightarrow_{S,T} = \longrightarrow_{T_S}^{\mathrm{p}} \cup \longrightarrow_T$ which presents the right congruence $\sim_{\mathcal{H}}$ in that $\sim_{\mathcal{H}} = \overset{*}{\Longleftrightarrow}_{S,T}$. Since we have $w \in \mathcal{H}$ if and only if $w \overset{*}{\Longleftrightarrow}_{S,T} \lambda$, this problem becomes immediately solvable for appropriate reduction relations, e.g. in the case of λ-confluence for $\Longrightarrow_{S,T}$. Following a short presentation of TC, three different string rewriting approaches will be outlined and their output compared to TC.

4 The Todd-Coxeter Coset Enumeration Procedure

Todd-Coxeter coset enumeration (TC) is a famous method from combinatorial group theory for studying finitely presented groups. It is based on the following fundamental observations: Presenting a group $\mathcal{G}$ in terms of generators Σ and relators R corresponds to viewing it as the quotient of the free group $\mathcal{F}$ (generated by Σ) by the normal subgroup $\mathcal{N}$ generated by R. $\mathcal{N}$ can be viewed as the subgroup of $\mathcal{F}$ generated by $N(R) = \{w \circ r \circ w^{-1} \mid w \in \mathcal{F}, r \in R\}$. Notice that if R is finite, $\mathcal{N}$, while finitely generated as a normal subgroup of $\mathcal{F}$, need not be finitely generated as a subgroup.
Now given a subgroup $\mathcal{H}$ of $\mathcal{G}$ generated by a set $S \subseteq \mathcal{G}$ the index of $\mathcal{H}$ in $\mathcal{G}$ is the same as the index of the subgroup $\mathcal{U}$ generated by $S \cup N(R)$ in $\mathcal{F}$. While it is undecidable whether a subgroup has finite index in a group, TC attempts to *verify* whether the index is finite.
In the following we will always assume that the group $\mathcal{G}$ and the subgroup $\mathcal{H}$ are finitely presented respectively generated, i.e. the sets Σ, R and S are finite. Detailed descriptions of TC procedures can be found e. g. in [11, 3, 15].

We only list some of the properties and their interpretations here: If the index of $\mathcal{H}$ in $\mathcal{G}$ is finite the procedure halts and produces a set of coset representatives and a coset table with entries $c \circ a$ for each coset c and each $a \in \bar{\Sigma}$. The (unique) coset representative for any word in $\bar{\Sigma}^*$ can be computed by tracing it through the coset table starting with λ. Moreover, given a total well-founded ordering $>$ on $\bar{\Sigma}^*$ which is additionally compatible with right concatenation, we can associate to each coset c the representative $w \in \bar{\Sigma}^*$ of minimal length[4]. The coset table gives rise to a convergent prefix string rewriting system as follows: To each coset w, each $a \in \bar{\Sigma}$ and the respective coset w_a corresponding to $w \circ a$, associate a rule[5] of the form $wa \longrightarrow w_a$. This prefix string rewriting system then can be used to determine the coset of a word in $\bar{\Sigma}^*$ by prefix reduction.
Let us illustrate these findings with an example from [3], page 71:

Example 2 *Let $\mathcal{G}$ be the Dyck group $D(3,3,2)$ presented by $\Sigma = \{a, b\}$ and $R = \{aaa, bbb, abab\}$, and $\mathcal{H}$ the subgroup of $\mathcal{G}$ generated by $\{a\}$. The index of $\mathcal{H}$ in $\mathcal{G}$ is 4 and* Tc *(using the length lexicographical ordering induced by $a \prec b \prec a^{-1} \prec b^{-1}$) computes the coset representatives $\{\lambda, b, b^{-1}, ba^{-1}\}$ and the coset table*

	a	b	a^{-1}	b^{-1}
λ	λ	b	λ	b^{-1}
b	b^{-1}	b^{-1}	ba^{-1}	λ
b^{-1}	ba^{-1}	λ	b	b
ba^{-1}	b	ba^{-1}	b^{-1}	ba^{-1}

giving rise to the prefix string rewriting system $R_{TC} = \{\ a \longrightarrow \lambda,\ a^{-1} \longrightarrow \lambda,\ ba \longrightarrow b^{-1},\ bb \longrightarrow b^{-1},\ bb^{-1} \longrightarrow \lambda,\ b^{-1}a \longrightarrow ba^{-1},\ b^{-1}b \longrightarrow \lambda,\ b^{-1}a^{-1} \longrightarrow b,\ b^{-1}b^{-1} \longrightarrow b,\ ba^{-1}a \longrightarrow b, ba^{-1}b \longrightarrow ba^{-1},\ ba^{-1}a^{-1} \longrightarrow b^{-1},\ ba^{-1}b^{-1} \longrightarrow ba^{-1}\ \}$.
The coset representative of the word aba can be deduced by either tracing the coset table: $\lambda \circ \mathbf{a} = \lambda$, $\lambda \circ \mathbf{b} = b$ and $b \circ \mathbf{a} = b^{-1}$, or by prefix reduction: $\underline{a}ba \longrightarrow^{\mathrm{p}}_{a \longrightarrow \lambda} \underline{ba} \longrightarrow^{\mathrm{p}}_{ba \longrightarrow b^{-1}} b^{-1}$ In both cases we find that aba lies in the coset represented by b^{-1} which is in fact the minimal representative of this coset with respect to the chosen ordering.

[4]There are strategies for Tc to produce exactly these cosets by using words as coset representatives and bonus and collapse equations as oriented "rules".

[5]Notice that there are trivial rules among these where the left and right hand sides coincide as words. They correspond to the defining equations in Tc and of course have to be omitted in order to make the system terminating.

5 Kuhn and Madlener's Approach

In [4] a procedure for completing the reduction relation $\Longrightarrow_{S,T}$ as described in Section 3 is provided for the case that the group presentation $(\bar{\Sigma}, T = T_I \cup T_R)$ is *convergent*. The basic idea is to interpret the rules defining the group as the (possibly infinite) set of prefix string rewriting rules $P_T = \{x\ell \longrightarrow xr \mid x \in \Gamma^*, \ell \longrightarrow r \in T,$ no proper prefix of $x\ell$ is $\longrightarrow_T$ $-$reducible$\}$. Then $\overset{*}{\Longleftrightarrow}_{S,T} = \overset{*}{\longleftrightarrow}^{\mathrm{p}}_{T_S \cup P_T}$. A completion procedure for this reduction relation has T_S and T as input and on termination a set T_C as output which is constructed as follows: T_C is initialized as T_S. The completion step then considers critical situations between rules $\ell_1 \longrightarrow r_1, \ell_2 \longrightarrow r_2 \in T_C$ of the form $\ell_1 \equiv \ell_2 y$ or between rules $\ell_1 \longrightarrow r_1 \in T$ and $\ell_2 \longrightarrow r_2 \in T_C$ of the form $x\ell_1 \equiv \ell_2 y$ with $|x| < |\ell_2|$. The overlaps between rules in T can be omitted as T is already convergent. If critical situations cannot be resolved, the newly arising rules are added to T_C. On termination the (still possibly infinite) set $T_C \cup P_T$ is convergent as a prefix string rewriting system. It is known that for subgroups of finite index the procedure terminates. In this case the interreduced version of the set $T_C \cup P_T$ is finite and corresponds to the set R_{TC} arising from the coset table for Tc as described in the previous section. Of course in order to treat Example 2 we have to use a convergent group presentation for $\mathcal{G}$ instead of the defining relators presented in the previous section. Completing $\{aaa \longrightarrow \lambda, bbb \longrightarrow \lambda, abab \longrightarrow \lambda\} \cup \{aa^{-1} \longrightarrow \lambda, a^{-1}a \longrightarrow \lambda, bb^{-1} \longrightarrow \lambda, b^{-1}b \longrightarrow \lambda\}$ with respect to the length-lexicographical ordering induced by the precedence $b^{-1} \succ a^{-1} \succ b \succ a$ gives us the convergent string rewriting system $(\{a, b, a^{-1}, b^{-1}\}, \{\ aa^{-1} \longrightarrow \lambda,\ a^{-1}a \longrightarrow \lambda,\ bb^{-1} \longrightarrow \lambda,\ b^{-1}b \longrightarrow \lambda,\ aa \longrightarrow a^{-1},\ bab \longrightarrow a^{-1},\ bb \longrightarrow b^{-1},\ a^{-1}a^{-1} \longrightarrow a,\ a^{-1}b^{-1} \longrightarrow ba,\ b^{-1}a^{-1} \longrightarrow ab,\ b^{-1}b^{-1} \longrightarrow b,\ b^{-1}ab \longrightarrow ba^{-1},\ bab^{-1} \longrightarrow a^{-1}b,\ aba \longrightarrow b^{-1},\ a^{-1}ba \longrightarrow ab^{-1},\ aba^{-1} \longrightarrow b^{-1}a,\ ba^{-1}b \longrightarrow ab^{-1}a,\ b^{-1}ab^{-1} \longrightarrow ab^{-1}a,\ a^{-1}ba^{-1} \longrightarrow ab^{-1}a\ \})$ and on input $T_S = \{a \longrightarrow \lambda\}$ on termination we get the set $T_C = \{a \longrightarrow \lambda, a^{-1} \longrightarrow \lambda, ba \longrightarrow b^{-1}, b^{-1}a \longrightarrow ba^{-1}\}$ and the convergent prefix string rewriting system $T_C \cup P_T$ can be prefix interreduced to the set of rules R_{TC} in Example 2.

While sometimes also successful for subgroups of infinite index, this method is no longer applicable to groups given by arbitrary presentations. This would also require that on completing T_C we additionally have to resolve critical situations arising from overlaps of rules in T. This is essentially what happens in Sims' approach.

6 Sims' Approach

In Section 2.8. of [15] Sims gives an approach similar to the one in the previous section but allowing *arbitrary* presentations $(\bar{\Sigma}, T = T_I \cup T_R)$ for the group $\mathcal{G}$. Instead of defining a specialized completion procedure for prefix string rewriting he encodes the set T_S into an ordinary string rewriting system by adding a new symbol \$ and transforming the rules to a new set $T_{\$S} = \{\$s \longrightarrow \$ \mid s \in S\}$. Then the extended string rewriting system $(\bar{\Sigma} \cup \{\$\}, T_{\$S} \cup T_I \cup T_R)$ is completed using KB.

For the cosets of the subgroup of the Dyck group $D(3,3,2)$ as presented in Example 2 we get the following situation: $\bar{\Sigma} = \{a, b, a^{-1}, b^{-1}\}$, $T_R \cup T_I = \{aaa \longrightarrow \lambda, bbb \longrightarrow \lambda, abab \longrightarrow \lambda\} \cup \{aa^{-1} \longrightarrow \lambda, a^{-1}a \longrightarrow \lambda, bb^{-1} \longrightarrow \lambda, b^{-1}b \longrightarrow \lambda\}$, and $T_{\$S} = \{\$a \longrightarrow \$\}$. Completing the string rewriting system $(\{a, a^{-1}, b, b^{-1}, \$\}, T_R \cup T_I \cup T_{\$S})$ with respect to the length-lexicographical ordering induced by the precedence $b^{-1} \succ a^{-1} \succ b \succ a \succ \$$ using KB results in the convergent set of rules $R_{KB} = \{aa^{-1} \longrightarrow \lambda, bb^{-1} \longrightarrow \lambda, b^{-1}b \longrightarrow \lambda, \$a \longrightarrow \$, aa \longrightarrow a^{-1}, a^{-1}a \longrightarrow \lambda, bab \longrightarrow a^{-1}, bb \longrightarrow b^{-1}, \$a^{-1} \longrightarrow \$, a^{-1}a^{-1} \longrightarrow a, a^{-1}B \longrightarrow ba, b^{-1}a^{-1} \longrightarrow ab, b^{-1}b^{-1} \longrightarrow b, b^{-1}ab \longrightarrow ba^{-1}, bab^{-1} \longrightarrow a^{-1}b, aba \longrightarrow b^{-1}, \$ba \longrightarrow \$b^{-1}, a^{-1}ba \longrightarrow ab^{-1}, aba^{-1} \longrightarrow b^{-1}a, \$b^{-1}a \longrightarrow \$ba^{-1}, ba^{-1}b \longrightarrow ab^{-1}a, b^{-1}ab^{-1} \longrightarrow ab^{-1}a, a^{-1}ba^{-1} \longrightarrow ab^{-1}a\}$. For the sets T_C and T from the previous section we then have $T_C = \{\ell \longrightarrow r \mid \$\ell \longrightarrow \$r \in R_{KB}\}$ and $T = R_{KB} \backslash T_C$.

Let us continue with a comparison of TC and KB as presented in this setting: When running KB for the free group on $T_I \cup T_{\$S}$ we are in fact computing a Nielsen reduced set for the subgroup generated by S. The situation is slightly different in the general case, as in contrary to TC, although we are simulating coset enumeration, it no longer must terminate for finite index. This is due to the fact that in any case KB will try to complete the defining relators $T_R \cup T_I$ and there are examples where we have finite index but no finite convergent system for $T_R \cup T_I$ exists. However, if we know that the index is finite, it is possible to find a bound on how far we have to run KB to gain enough information to describe the cosets. More information on this can be found in Section 3.10. in [15] where the following example is taken from. Notice that this example cannot be handled by the approach of Kuhn and Madlener in the previous section for the chosen string rewriting system presenting the group is not convergent.

Example 3 *Let our group $\mathcal{G}$ be presented by the string rewriting system*

$\Gamma = \{x, y, Y\}$[6] *and the set of rules* $T = \{xx \longrightarrow \lambda, yY \longrightarrow \lambda, Yy \longrightarrow \lambda, yxyxyxy \longrightarrow xYxYxYx\}$. *The subgroup is encoded by* $T_{\$S} = \{\$y \longrightarrow \$x, \$yxYxyxYx \longrightarrow \$xyxYxyxY\}$. *Running* KB *on input* $\Gamma \cup \{\$\}$ *and* $T \cup T_{\$S}$ *with the length-lexicographical ordering induced by the precedence* $Y \succ y \succ x \succ \$$ *diverges. However Sims outlines how to use the knowledge that we have finite index to stop the calculation after considering all overlaps up to length 17 and verifying that there are 24 cosets.*

Next we provide a procedure which can handle the finite index case *without* applying additional knowledge.

7 Simulating Todd-Coxeter with Prefix String Rewriting

In this section we want to present a procedure which emulates TC. It is based on the algebraic procedure presented in [14] for simulating TC using prefix Gröbner bases methods.
Informally the procedure works as follows:
The input consists of the group presentation in terms of the sets $\bar{\Sigma}$, $T_I = \{aa^{-1} \longrightarrow \lambda, a^{-1}a \longrightarrow \lambda \mid a \in \Sigma\}$ and $T_R = \{r \longrightarrow \lambda \mid r \in R\}$, and the generators of the subgroup encoded in the set of rules $T_S = \{s \longrightarrow \lambda \mid s \in S\}$. Additionally we will use the following sets:

- A set $N \subseteq \mathcal{F}$ of potential coset representatives of the subgroup generated by $S \cup N(R)$ in $\mathcal{F}$.
- A set $B \subseteq \mathcal{F}$ that serves as a test set for possible coset representatives.
- A set H that is used to approximate the infinite generating set $N(R)$.
- A prefix interreduced set of rules G that is used to decide whether or not the elements in B are indeed coset representatives of the respective approximated subgroup.

The set N, which on termination will contain the coset representatives, is initialized as the set containing the empty string only, which is the coset

[6] We can omit the formal inverse X of x as completion would lead to an identification of the two.

representative of the subgroup itself. During the computation this set remains prefix-closed. The set B of possible candidates for further cosets is initialized as $B := \{a \mid a \in \bar{\Sigma}\}$. The set G, which on termination will encode the non-trivial part of the coset table, is computed by prefix interreducing the set $T_I \cup T_R \cup T_S$. Hence, the rules in G can be used to decide the membership for the subgroup of $\mathcal{F}$ that is generated by $S \cup R$. This completes the initialization phase.
Now as long as there still are candidates for new cosets in B the following actions are performed: We choose and remove the smallest element from B (with respect to the length-lexicographical ordering) and call it τ. If τ is not prefix-reducible by G, then it is added to N and all elements of the form τa are added to B, where $a \in \bar{\Sigma}$. Next $H := \{\tau\ell \longrightarrow \tau r \mid \ell \longrightarrow r \in T_I \cup T_R\}$ is determined, which in TC corresponds to the process of marking the first and the last slot of each relator table with the newly found coset representative τ. Finally the set $G \cup H$, which corresponds to the subgroup of $\mathcal{F}$ that is generated by $S \cup \{\tau \circ r \circ \tau^{-1} \mid \tau \in N, r \in R\}$, is prefix interreduced. Hence, we approximate the potentially infinite generating set of the subgroup $\langle S \cup N(R) \rangle$ of $\mathcal{F}$. Based on the new set G some elements of N may become prefix reducible. These have to be removed from N, as they are no longer coset representatives. This corresponds to the collapse of cosets in TC.

Generalized Tc Simulation
Given: $(\bar{\Sigma}, T = T_I \cup T_R)$, T_S

$N := \{\lambda\}$;
$B := \{a \mid a \in \bar{\Sigma}\}$;
$G :=$ prefix.interreduce$(T \cup T_S)$;
while $B \neq \emptyset$ **do**
    $\tau :=$ min$_<(B)$;
    $B := B \backslash \{\tau\}$;
    if τ is not prefix reducible by G
    then $N := N \cup \{\tau\}$;
        $B := B \cup \{\tau a \mid a \in \bar{\Sigma}\}$;
        $H := \{\tau\ell \longrightarrow \tau r \mid \ell \longrightarrow r \in T\}$;
        $G :=$ prefix.interreduce$(G \cup H)$;
        $N := N \backslash \{w \in N \mid w$ is reducible by G$\}$;
    endif
endwhile

The following theorem is a direct conclusion of the results stated in [14].

Theorem 4 *Let R and S be as specified above. Procedure* Generalized Tc Simulation *terminates iff the subgroup generated by $S \cup N(R)$ has finite index in $\mathcal{F}$.*

Applying this procedure to Example 3 it terminates with 24 cosets in N and the set G encodes the non-trivial part of the coset table.

8 TC in Monoids

Notice that procedure Generalized Tc Simulation in fact can also be applied to monoids by giving as input a string rewriting system (Σ, T) where T no longer needs to contain the inverse rules. The question now is, what does the procedure compute in this more general setting?
Given a finite subset S of a monoid $\mathcal{M}$, we let $\langle S \rangle = \{s_1 \circ \ldots \circ s_n \mid n \in \mathbb{N}, s_i \in S\}$ denote the submonoid generated by S. A submonoid $\mathcal{U}$ of a monoid $\mathcal{M}$ is called **finitely generated** if there exists a finite subset S of $\mathcal{M}$ such that $\mathcal{U} = \langle S \rangle$.
As in group theory we can define a right congruence as follows:

$$u \sim_{\mathcal{U}} v \text{ if and only if } \mathcal{U}u = \mathcal{U}v$$

for $u, v \in \mathcal{M}$. But, we can no longer characterize the submonoid by a special congruence class of this right congruence as the following example shows:

Example 5 *Let $\Sigma = \{a, b\}, T = \{ab \longrightarrow \lambda\}$ be a string rewriting system presenting of a monoid $\mathcal{M}$, the bicyclic monoid. Let $\mathcal{U} = \{a^n \mid n \in \mathbb{N}\}$ be the submonoid of $\mathcal{M}$ generated by $S = \{a\}$. For the right congruence class of λ we find $[\lambda]_{\sim_{\mathcal{U}}} = \{\lambda\}$, i.e. even $a \notin [\lambda]_{\sim_{\mathcal{U}}}$, while $\mathcal{U} = \{a^n \mid n \in \mathbb{N}\}$.*

So when defining "cosets" of submonoids in monoids we meet the difficulty that they no longer have such nice properties as cosets of subgroups in groups have. In [12] Neumann gives a different structure which is as close as possible with respect to the desired properties: Instead of adopting the analogue of a single coset of a subgroup in a group, he looks at the set of *all* cosets: Given a right congruence on monoids, i.e. an equivalence relation that admits right multiplication by elements of the monoid, the equivalence classes are called **blocks**. If B is such a block, then the elements $m \in \mathcal{M}$ such that $Bm \subseteq B$

form a submonoid of $\mathcal{M}$ called the **fixing submonoid** of B. Different non-void fixing submonoids that arise from the blocks of a right congruence can then be considered as conjugate.
Now starting with a submonoid $\mathcal{H}$ of $\mathcal{M}$ we are looking for the least right congruence that contains $\mathcal{H}$ in a block: then $\mathcal{H}$ will be contained in the fixing submonoid of that block[7]. This is the nearest one can get to "cosets" of $\mathcal{H}$ in $\mathcal{M}$.
We can define the block containing $\mathcal{H}$, called $B_{\mathcal{H}}$, recursively as follows:

- $B_0 = \mathcal{H}$
- $B_i = B_{i-1} \cup \{b \circ x \mid b \in B_{i-1}, x \in \mathcal{M}$ s.t. there exist $b_1, b_2 \in B_{i-1}$ with $b_1 = b_2 \circ x\}$
- $B_{\mathcal{H}} = \bigcup_{i \in \mathbb{N}} B_i$.

Of course if $\mathcal{H}$ is a subgroup of a group $\mathcal{M}$, then $B_{\mathcal{H}} = \mathcal{H}$ as $b_1 = b_2 \circ x$ implies $x = b_2^{-1} \circ b_1 \in \mathcal{H}$.
Notice that when in defining B_i, if some x are used in extending B_{i-1} because there are $b_1, b_2 \in B_{i-1}$ such that $b_1 = b_2 \circ x$, then $x \in B_i$ as well since we have $\lambda \in B_0$.
Reviewing the case of the bicyclic monoid in Example 5 we get $B_{\langle a \rangle} = \mathcal{M}$ since for $i, j \in \mathbb{N}$ we have $a^i \circ b^i a^j = a^j \in \langle a \rangle = B_0$ and hence $b^i a^j \in B_{\langle a \rangle}$. Procedure Generalized Tc Simulation computes the sets $N = \{\lambda\}$ and $G = \{a \longrightarrow \lambda, b \longrightarrow \lambda\}$. On the other hand, if we look at the submonoid generated by b we get $B_{\langle b \rangle} = \{b^n \mid n \in \mathbb{N}\}$ and procedure Generalized Tc Simulation computes the infinite sets $N = \{a^n \mid n \in \mathbb{N}\}$ and $G = \{a^{n+1}b \longrightarrow a^n \mid n \in \mathbb{N}\} \cup \{b \longrightarrow \lambda\}$.
In order to link for some generating set S the block $B_{\langle S \rangle}$ to procedure Generalized Tc Simulation, we need the algebraic structure of one-sided ideals in monoid rings. It is easy to show that there is a one to one correspondence between the congruences generated by sets of rules $\{\ell_i \longrightarrow r_i \mid 1 \leq i \leq n\}$ on Σ^* and the congruences of (one-sided) ideals generated by sets of binomials $\{\ell_i - r_i \mid 1 \leq i \leq n\}$ in $\mathbb{K}[\Sigma^*]$ (see e.g. [8]). Using this fact, we get the following algebraic characterization of $B_{\langle S \rangle}$ in terms of right ideals in monoid rings.

[7]See the appendix for a proof.

Theorem 6 *Let S be a subset of $\mathcal{M}$ and $P_S = \{s - 1 \mid s \in S\}$ a subset of $\mathbb{K}[\mathcal{M}]$ associated to S. Then the following statements are equivalent for $w \in \mathcal{M}$:*

(1) $w \in B_{\langle S \rangle}$.

(2) $w - 1 \in \mathsf{ideal}_r^{\mathbb{K}[\mathcal{M}]}(P_S)$.

Exploiting the connections between string rewriting systems and special binomial ideal bases, it follows that on termination the set G computed by procedure Generalized Tc Simulation corresponds to a prefix Gröbner basis $\{\ell - r \mid \ell \longrightarrow r \in G\}$ of the right ideal generated by the set P_S in $\mathbb{K}[\mathcal{M}]$ where $\mathcal{M}$ is the monoid presented by the string rewriting system (Σ, T) (see [13, 8] for more details):

Corollary 7 *For S a subset of Σ^* let $P_S = \{s - 1 \mid s \in S\}$ and for T let $P_T = \{x\ell - xr \mid x \in \Sigma^*, \ell \longrightarrow r \in T,$ no proper prefix of $x\ell$ is $\longrightarrow_T$ $-$reducible$\}$. Then the following statements are equivalent:*

(1) $w \in B_{\langle S \rangle}$.

(2) $w - 1 \in \mathsf{ideal}_r^{\mathbb{K}[\Sigma^*]}(P_S \cup P_T)$.

Moreover, if procedure Generalized Tc Simulation *terminates on input S and T with output G, then $w \in B_{\langle S \rangle}$ if and only if $w \xrightarrow{*}{}^{\mathrm{p}}_{G} 1$.*

9 Conclusions

In this paper we have recalled some methods used to study subgroups of groups and compared them in terms of prefix string rewriting. A procedure based on prefix string rewriting was presented which emulates Todd-Coxeter coset enumeration naturally hence bringing Tc and string rewriting into a much closer context. This procedure was generalized to the setting of monoids where it is related to the more general concept of blocks as introduced by Neumann in [12].

References

[1] B. Benninghofen, S. Kemmerich, and M.M. Richter. *Systems of Reductions*. LNCS 277. Springer, 1987.

[2] R. Book and F. Otto. *String Rewriting Systems*. Springer, 1993.

[3] D. L. Johnson. *Presentation of Groups*. Cambridge University Press, 1976.

[4] N. Kuhn and K. Madlener. A method for enumerating cosets of a group presented by a canonical system. In G. Gonnet, editor, *Proc. ISSAC'89*, pages 338–350. ACM, 1989.

[5] N. Kuhn, K. Madlener, and F. Otto. Computing presentations for subgroups of polycyclic groups and of context-free groups. *Applicable Algebra in Engineering, Communication and Computing*, 5:287–316, 1994.

[6] G. Labonté. An algorithm for the construction of matrix representations for finitely presented non-comutative algebras. *Journal of Symbolic Computation*, 9:27–38, 1990.

[7] S. Linton. Constructing matrix representations of finitely presented groups. *Journal of Symbolic Computation*, 12:427–438, 1991.

[8] K. Madlener and B. Reinert. String rewriting and Gröbner bases – a general approach to monoid and group rings. In M. Bronstein, J. Grabmeier, and V. Weispfenning, editors, *Proceedings of the Workshop on Symbolic Rewriting Techniques, Monte Verita, 1995*, volume 15 of *Progress in Computer Science and Applied Logic*, pages 127–180. Birkhäuser, 1998.

[9] R. McNaughton. The finiteness of finitely presented monoids. Technical report, Rensselaer Polytechnic Institute, 1997.

[10] C. F. Miller. Decision problems for groups – survey and reflections. In *Algorithms and Classification in Combinatorial Group Theory*, pages 1–60. Springer, 1991.

[11] J. Neubüser. An elementary introduction to coset table methods in computational group theory. In C. M. Campbell and E. F. Robertson, editors, *Groups St. Andrews 1981*, L.M.S. Lecture Notes 71, pages 1–45. Cambridge University Press, 1982.

[12] B. H. Neumann. Some remarks on semigroup presentations. *Canadian Journal of Mathematics*, 19:1018–1026, 1967.

[13] B. Reinert. *On Gröbner Bases in Monoid and Group Rings*. PhD thesis, Universität Kaiserslautern, 1995.

[14] B. Reinert, T. Mora, and K. Madlener. A note on nielsen reduction and coset enumeration. In O. Gloor, editor, *Proc. ISSAC'98*, pages 171–178. ACM, 1998.

[15] C. Sims. *Computation with Finitely Presented Groups*. Cambridge University Press, 1994.

10 Appendix

Lemma 8 *Let $\sim$ be the least right congruence such that $\mathcal{H}$ is contained in a block B. Then for all $h \in \mathcal{H}$ we have $Bh \subseteq B$, i.e. $\mathcal{H}$ is contained in the fixing monoid of B.*

Proof :
We show that for each $b \in B$ we have $b \circ h \in B$. Since $b, \lambda \in B$, we have $b \sim \lambda$, and hence as $\sim$ is a right congruence $b \circ h \sim \lambda \circ h = h$ holds. Now $h \in B$ implies $b \circ h \in B$ and we are done.

q.e.d.

The following lemma is necessary to prove Theorem 6

Lemma 9 *For some $h \in \mathcal{H}$ and $x \in \mathcal{M}$ we have $h \circ x \in B_{\mathcal{H}}$ if and only if $x \in B_{\mathcal{H}}$.*

Proof :
Since $h, \lambda \in \mathcal{H}$ and $\sim$ is a right congruence, $h \sim \lambda$ implies $h \circ x \sim x$, and as $h \circ x \in B_{\mathcal{H}}$ we can conclude $x \in B_{\mathcal{H}}$.
On the other hand we show that $x \in B_{\mathcal{H}}$ implies $h \circ x \in B_{\mathcal{H}}$ by showing $hB_i \subseteq B_i$ by induction on i where $x \in B_i$. If $i = 0$ we find $h, x \in B_0 = \mathcal{H}$ and hence $h \circ x \in B_0$. Hence let us assume that for all $h \in \mathcal{H}$, $x \in B_{i-1}$ we have $h \circ x \in B_{i-1}$. Now take $x \in B_i \backslash B_{i-1}$ and hence $x = b \circ x'$ for some $b \in B_{i-1}$, $x' \in B_i$. The induction hypothesis implies $h \circ b \in B_{i-1}$ and hence

$(h \circ b) \circ x' = h \circ x \in B_i$.

q.e.d.

Proof of Theorem 6:
$1 \Longrightarrow 2:$ Let $w \in B_{\langle S \rangle}$, i.e., $w \in B_i$ for some $i \in \mathbb{N}$. We show our claim by induction on i. If $w \in B_0$ this implies $w \in \mathcal{H}$ and hence $w - 1 \in P_S \subseteq \mathsf{ideal}_r(P_S)$ and we are done. Let us assume that for all $w \in B_{i-1}$ we have $w - 1 \in \mathsf{ideal}_r(P_S)$. Then for $w \in B_i \backslash B_{i-1}$ there must exist $b_1, b_2, b \in B_{i-1}$ and $x \in \mathcal{M}$ such that $b_1 = b_2 \circ x$ and $w = b \circ x$. Then we get

$$\begin{aligned} w - 1 &= b \circ x \underbrace{- b_2 \circ x + b_2 \circ x}_{=0} - 1 \\ &= (b - b_2) \circ x + \underbrace{b_2 \circ x - 1}_{= b_1 - 1} \end{aligned}$$

and $b - b_2, b_1 - 1 \in \mathsf{ideal}_r(P_S)$ imply $w - 1 \in \mathsf{ideal}_r(P_S)$.
$2 \Longrightarrow 1:$ We have to show that $w - 1 \in \mathsf{ideal}_r(P_S)$ implies $w \in B_{\langle S \rangle}$. We know $w - 1 = \sum_{j=1}^{n} \alpha_j \cdot (s_j - 1) * x_j$, where $\alpha_j \in \mathbb{K}^*$, $s_j \in S$, $x_j \in \mathcal{M}$. Therefore, by showing the following stronger result we are done: A representation $w - b = \sum_{j=1}^{m} p_j$ where $p_j = \alpha_j \cdot (s_j - 1) \circ x_j$ or $p_j = \alpha_j \cdot (u_j - v_j)$ with $\alpha_j \in \mathbb{K}^*$, $s_j \in \mathcal{H}$, $x_j \in \mathcal{M}$ and $b, u_j, v_j \in B_{\langle S \rangle}$ implies $s_j \circ x_j, x_j \in B_{\langle S \rangle}$ and moreover $w \in B_{\langle S \rangle}$. Now, let $w - b = \sum_{j=1}^{m} p_j$ be such a representation. Depending on this representation we define K as the number of polynomials p_j containing b as a term. We will show our claim by induction on (m, K), where $(m', K') < (m, K)$ if and only if $m' < m$ or ($m' = m$ and $K' < K$). In case $m = 0$, $w - b = 0$ implies $w = b \in B_{\langle S \rangle}$. Thus let us assume $m > 0$.
In case $K = 1$, let p_m be the polynomial containing b and wlog let us assume $p_m = b - b'$. Then $b' \in B_{\langle S \rangle}$ as either this follows directly from the definition or $p_m = (s_m - 1) \circ x_m$ and then Lemma 9 is applicable and either $b = s_m \circ x_m$ or $b = x_m$. Now we get $w - b - p_m = w - b' = \sum_{j=1}^{m-1} p_j$ and we are done as m is decreased.
In case $K > 1$ there are at least two polynomials say p_m, p_{m-1} in the corresponding representation containing b and without loss of generality we can assume $p_m = \alpha_m \cdot (b - b_m)$ and $p_{m-1} = \alpha_{m-1} \cdot (b - b_{m-1})$. As before we can conclude that $b_m, b_{m-1} \in B_{\langle S \rangle}$. If then $b_m = b_{m-1}$ we can immediately decrease m by substituting the occurrence of $p_m + p_{m-1}$ by $(\alpha_m + \alpha_{m-1}) \cdot p_{m-1}$.

Otherwise we can proceed as follows:

$$\begin{aligned} p_m + p_{m-1} &= p_m \underbrace{-\alpha_m \cdot \alpha_{m-1}^{-1} \cdot p_{m-1} + \alpha_m \cdot \alpha_{m-1}^{-1} \cdot p_{m-1}}_{=0} + p_{m-1} \\ &= \alpha_m \cdot (b_{m-1} - b_m) + (\alpha_m \cdot \alpha_{m-1}^{-1} + 1) \cdot p_{m-1} \end{aligned}$$

and while m is not decreased unless $(\alpha_m \cdot \alpha_{m-1}^{-1} + 1) = 0$, K is.

q.e.d.

List of papers published in the Reports on Computer Algebra series

[RCA:29] Greuel, G.-M. Computer Algebra and Algebraic Geometry – Achievements and Perspectives. Feb 2000.

[RCA:28] Siebert, T. Recursive Computation of Free Resolutions and a Generalized Koszul Complex. Jan 2000.

[RCA:27] Bachmann, O. and Gräbe, H.-G. The SymbolicData Project: Towards an Electronic Repository of Tools and Data for Benchmarks of Computer Algebra Software. Jan 2000.

[RCA:26] B. Reinert. Solving One-Sided Equations in Integer Monoid Rings. Jan 2000.

[RCA:25] B. Reinert and D. Zeckzer. Coset Enumeration using Prefix Gröbner Bases in Mrc – An Experimental Approach. Nov 1999.

[RCA:23] Birgit Reinert. Observations on coset enumeration. Nov 1998.

[RCA:22] K. Madlener and F. Otto. Some Applications Of Prefix-Rewriting In Monoids, Groups, And Rings. November 1998.

[RCA:21] G.-M. Greuel, G. Pfister, and H. Schönemann. Singular version 1.2 user manual. June 1998.

[RCA:20] B. Reinert and D. Zeckzer. MRC – A System for Computing Gröbner Bases in Monoid and Group Rings. July 1998.

[RCA:19] B. Reinert, K. Madlener, and T. Mora. A note on nielsen reduction and coset enumeration. February 1998.

[RCA:18] O. Bachmann and H. Schönemann. Monomial Representations for Gröbner Bases Computations. January 1998.

[RCA:17] Thomas Siebert. An algorithm for constructing isomorphisms of modules. January 1998.

[RCA:16] K. Madlener and B. Reinert. String Rewriting and Gröbner Bases – A General Approach to Monoid and Group Rings. October 1997.

[RCA:15] B. Martin and T. Siebert. Splitting Algorithm for vector bundles. September 1997.

[RCA:14] K. Madlener and B. Reinert. Relating rewriting techniques on monoids and rings: Congruences on monoids and ideals in monoid rings. September 1997.

[RCA:12] O. Bachmann, S. Gray, and H. Schönemann. MP Prototype Specification. Dec 1997.

[RCA:11] O. Bachmann. Effective simplification of cr expressions. January 1997.

[RCA:10] O. Bachmann, S. Gray, and H. Schönemann. A proposal for syntactic data integration for math protocols. January 1997.

[RCA:09] B. Reinert. Introducing reduction to polycyclic group rings – a comparison of methods. October 1996.

[RCA:08] T. Siebert. On strategies and implementations for computations of free resolutions. September 1996.

[RCA:02] H. Schönemann. Algorithms in singular. June 1996.

[RCA:01] H. Grassmann, G.-M. Greuel, B. Martin, W. Neumann, G. Pfister, W. Pohl, H. Schönemann, and T. Siebert. Standard bases, syzygies and their implementation in singular. July 1996.