Logged Rewriting for Monoids

Anne Heyworth [a,1] , Michael Johnson [b] ,

[a] *Department of Computer Science, Leicester University, LE1 7RH, U.K.*

[b] *Information and Communication Sciences, Macquarie University, NSW 2109 Australia.*

Abstract

The original motivation for rewriting is to provide decision procedures for equational theories. A string-rewriting system is usually associated with a monoid presentation. At the first level the problem is to decide which combinations of the generators are equivalent under the given rules; Knuth-Bendix completion of the string-rewriting system is one of the most successful mechanisms for solving this problem. At the second level, the problem involves determining which combinations of rules are equivalent. Logged rewriting is a technique which not only transforms strings but records the transformation in terms of the original system of rules. The relations between combinatorial, homotopical and homological finiteness conditions for monoids prompt us to consider using computer-friendly rewriting systems to calculate homotopical and homological structure from monoid presentations.

Key words: Rewriting system, Knuth-Bendix completion, monoid presentation, crossed module, identity among relations, derivation scheme.
PACS: 08A59, 18D05, 18D10, 68Q42.

1 Introduction

The background to this work includes the papers by Squier, Lafont, Prouté, Otto, Cremanns, Anick, Kobayashi, Pride and others on finiteness conditions

Email addresses: a.heyworth@mcs.le.ac.uk (Anne Heyworth), mike@ics.mq.edu.au (Michael Johnson).

[1] EPSRC GR/R29604/01 'Kan - A Categorical Approach to Computer Algebra'.

for monoids. The 'combinatorial' finiteness condition is that a monoid has a finite complete presentation. This implies the homological finiteness condition FP_∞ (Anick, 1986; Squier, 1987; Kobayashi, 1990) and also the 'homotopical' finiteness condition FDT (Squier, 1994). It is also known that the weaker homological condition FP_3 is not sufficient for either FDT or the existence of a complete rewriting system in the case of monoids (Squier, 1994).

Our aim is to use enhanced rewriting procedures to explicitly provide:

(1) A finite complete rewriting system (combinatorial specification)
(2) A finite set of homotopy generators for $P^{(2)}\Gamma$ (homotopical specification)
(3) A (small) finitely generated resolution (homological specification)

Logged rewriting for group presentations (Heyworth and Wensley, 1999) gives procedures for representing consequences of the relations of the presentation as elements of a pre-crossed module and algorithms for computing generators of the modules of identities among relations. In the monoid case the relations are given by pairs of terms and the structure of a crossed module is not appropriate to represent consequences of relations. It is well known (Stell, 1994; Street, 1992) that sesquicategories can be used to model rewriting systems. It has been proved (Pride, 1995; Gilbert, 1996) that when the rewriting system comes from a group presentation, the sesquicategory can be identified with the pre-crossed module of the presentation and factoring out by the interchange law to obtain the 2-category corresponds to factoring by the Peiffer rules to obtain the crossed module.

In the special case of groups, various results are known. In particular there are methods for calculating a set of generators for the kernel Π_2 of the crossed module (of 'consequences'), which is useful for constructing resolutions and calculating (co)homology. We examine the analogue of Π_2 when the rewriting system is not a group presentation, detailing its algebraic structure, an algorithm for computing a set of generators for it and provide justification that the constructions we make give combinatorial, homotopical and (co)homological detail in the same spirit as Π_2.

In the case of monoids, logged rewriting techniques have futher applications. Specifically, we have so far examined applications to the analysis of coset and double coset systems and used logged rewriting to provide an alternative to the Reidemeister-Schreier algorithm for finding presentations of subgroups (Brown et al, 2004; Ghani and Heyworth, 2003). Additionally, logged rewriting techniques are easily generalised to Kan extensions where they provide a proof method for a wide range of decidablility problems (Brown and Heyworth, 2000).

2 Logged Rewrite Systems

A *monoid presentation* is given as a pair $\mathcal{P} = \mathrm{mon}\langle X|R\rangle$ where X is a set of generators and R is a set of pairs (l, r) of elements of the free monoid X^*. The monoid presented, M is the quotient monoid obtained by factoring X^* by $=_R$, the congruence generated by R. The quotient monoid morphism will be denoted $\theta : X^* \to M$.

We will assume that the reader is familiar with standard string-rewriting as in (Book and Otto, 1993). The notation we use follows the usual conventions. The set R is a *rewriting system* for the monoid M and its elements are referred to as *rules*. The *reduction relation generated by* R on the free monoid X^* is denoted by $\to_R$, the reflexive, transitive closure is denoted $\overset{*}{\to}_R$ and the reflexive, symmetric, transitive closure $\overset{*}{\leftrightarrow}_R$ coincides with the congruence $=_R$. For convenience we assume that R is compatible with an admissible well-ordering $>$; i.e. for all pairs $(l, r) \in R$, we have $l > r$. This ensures that the relation $\to_R$ is Noetherian.

We shall now define a *logged rewrite system* for $\mathcal{P}$. Such a system must not only reduce any word in X^* to an irreducible word (unique if the rewrite system is complete) but must also express the actual reduction as a consequence of the original monoid relations. Consequences of group relations are represented by elements of a crossed module. Consequences of monoid relations cannot be represented in that framework; essentially this is because the free monoid does not have inverses. However, it is well known that general string-rewriting systems may be modelled by sesquicategories or 2-categories (Benson, 1975; Stell, 1994; Street, 1992). Therefore to every monoid presentation we associate a sesquicategory. Its 2-cells correspond to possible sequences of rewrites and inverse rewrites between strings in the free monoid, with respect to the given rewrite system. Formally:

Definition 2.1 (Sesquicategory of Rewrites)
The sesquicategory C_2 of a monoid presentation $\mathcal{P}$ *has a single 0-cell* $*$, *its 1-cells are the words of* X^* *and its 2-cells are generated by the rewrite system* R *in the following way:*

If $\alpha = (l, r)$ *is a rewrite rule in* R *then we define* $\mathrm{src}(\alpha) = l$ *and* $\mathrm{tgt}(\alpha) = r$.

For each string w *of the free monoid the identity rewrite is denoted* 1_w *and* $\mathrm{src}(1_w) = \mathrm{tgt}(1_w) = w$.

There are left and right actions of X^* *known as 'whiskering'. To keep notational devices minimal, we write the whiskers either side of the 2-cell as if it were a string; i.e. for any rewrite* α *and any elements* u *and* v *of the free monoid we say that* $u\alpha v$ *is a rewrite and* $\mathrm{src}(u\alpha v) = u\mathrm{src}(\alpha)v$ *and*

$\mathrm{tgt}(u\alpha v) = u\mathrm{tgt}(\alpha)v$. *Whiskering identity rewrites has the property that*

$$u \cdot 1_w \cdot v = 1_{uwv}$$

for all u, v in X^*.

A partial 'vertical' composition of rewrites is defined so that $\alpha \cdot \beta$ is a rewrite with $\mathrm{src}(\alpha \cdot \beta) = \mathrm{src}(\alpha)$ and $\mathrm{tgt}(\alpha \cdot \beta) = \mathrm{tgt}(\beta)$ whenever $\mathrm{tgt}(\alpha) = \mathrm{src}(\beta)$.

The formal inverse of any rewrite α is denoted α^{-1} where $\mathrm{src}(\alpha^{-1}) = tgt(\alpha)$ and $\mathrm{tgt}(\alpha^{-1}) = \mathrm{src}(\alpha)$.

Thus 2-cells are represented by sequences

$$\alpha = u_1\alpha_1^{\epsilon_1}v_1 \cdot \cdots \cdot u_n\alpha_n^{\epsilon_n}v_n$$

where $u_1, \ldots, u_n, v_1, \ldots, v_n \in X^*$, $\alpha_1, \ldots, \alpha_n \in R \cup \{1\}$ and $\epsilon_1, \ldots, \epsilon_n = \pm 1$ such that $u_i \,\mathrm{tgt}(\alpha_i^{\epsilon_i})\, v_i = u_{i+1} \,\mathrm{src}(\alpha_{i+1}^{\epsilon_{i+1}})\, v_{i+1}$ for $i = 1, \ldots, n-1$.

The sesquicategory axioms hold with respect to vertical composition and the whiskering action give the structure of the sesquicategory.

Further we shall allow rewrites to be cancelled with their inverses:

$$\alpha \cdot \alpha^{-1} = 1_{\mathrm{src}(\alpha)} \text{ for all rewrites } \alpha.$$

This gives $\cdot$ a groupoid structure, so we may refer to the sesquigroupoid C_2.

In the case where we can apply the rule α to one substring of a string and the rule β to another substring which is completely disjoint from the first, we regard the order in which the rules are actually applied as immaterial. This interchangability of non-overlapping rewrites is captured by the interchange law on the sesquicategory, giving us a 2-category.

Definition 2.2 (2-category of Rewrites) *The 2-category of rewrites is obtained by factoring the 2-cells C_2 of the sesquicategory/sesquigroupoid by the interchange law*

$$\alpha \,\mathrm{src}(\beta) \cdot \mathrm{tgt}(\alpha)\, \beta = \mathrm{src}(\alpha)\, \beta \cdot \alpha \,\mathrm{tgt}(\beta)$$

which defines a horizontal composition of the rewrites denoted $\alpha \circ \beta$ and generates a congruence $=_I$ on C_2.

In the case of term rewriting one does not always wish to factor out by the interchange law, in order to preserve the notion of length (number of steps) of a rewrite. In the case of string rewriting here we do not have to worry about notions of length of derivation, thus we use the 2-category. However, it should be noted that rewrites are represented uniquely in the sesquicategory, whereas the word problem for the 2-category is generally unsolvable (generalisation of a crossed module). We shall denote rewrites as α since it is more convenient than

$[\alpha]_I$, but regard two rewrites or 2-cells $\alpha, \beta \in C_2$ to be equivalent whenever $\alpha =_I \beta$.

In the context of groups, the seaquicategory associated to a monoid presentation is well known. In particular the following theorem has been proved elsewhere (Gilbert, 1996; Pride, 1995).

—— **Question: should this part be more detailed - or less?** ——

Theorem 1 (Interchange corresponds to Peiffer)
Let $\mathcal{P} = grp\langle X|R\rangle$ be a group presentation where $R \subseteq F(X)$. Let $\bar{X} = \{x^+ : x \in X\} \cup \{x^- : x \in X\}$. Let $\theta : \bar{X} \to F(X)$ be defined by $\theta(x^+) = x$, $\theta(x^-) = x^{-1}$ and let τ be a section of θ. Then let $\bar{R} = \{(\tau(w), 1) : w \in R\}$. A monoid presentation of the same group is

$$\bar{\mathcal{P}} = mon\langle \bar{X}|\bar{R}, \{(x^+x^-, 1) : x \in X\}, \{(x^-x^+, 1) : x \in X\ \}\rangle$$

The interchange law on the 2-cells of the sesquicategory $SQ(\bar{\mathcal{P}})$ associated to $\bar{\mathcal{P}}$ is identified with the Peiffer rule on the elements of the usual pre-crossed $F(X)$-module $PCM(\mathcal{P})$ associated to the group presentation $\mathcal{P}$.

Outline Proof Define $\Phi : SQ(\bar{\mathcal{P}}) \to PCM(\mathcal{P})$ by $\Phi(u\alpha v) = \alpha^v$. Then in the sesquicategory, the interchange law is as follows:

$$\alpha\, \mathrm{src}(\beta) \cdot \mathrm{tgt}(\alpha)\, \beta = \mathrm{src}(\alpha)\, \beta \cdot \alpha\, \mathrm{tgt}(\beta),$$

which maps under Φ to:

$$\alpha^{\mathrm{src}(\beta)^{-1}}\beta = \beta\alpha^{\mathrm{tgt}(\beta)^{-1}},$$

or, since $\mathrm{tgt}(\beta) = 1$ and $\mathrm{tgt}(\beta) = \delta(\beta)$:

$$\alpha^{\delta(\beta)} = \beta^{-1}\alpha\beta,$$

which is the Peiffer rule in the pre-crossed module as required. □

Definition 2.3 (Logged Rewrite System)
A logged rewrite system *for a presentation $\mathcal{P}$ of a monoid M is a collection of 2-cells (rewrites)*

$$\mathcal{L} = \{\alpha_1, \ldots, \alpha_n\}$$

of the associated 2-category $\mathcal{C}_2$ so that the underlying rewrite system

$$R_{\mathcal{L}} = \{(\mathrm{src}(\alpha_1), \mathrm{tgt}(\alpha_1)), \ldots, (\mathrm{src}(\alpha_n), \mathrm{tgt}(\alpha_n))\}$$

is a rewrite system for M.

The logged reduction of a string by a 2-cell α is recorded as: $u\alpha v : u\mathrm{src}(\alpha)v \to u\mathrm{tgt}(\alpha)v$. We shall use the terms 'logged rewrite' and '2-cell' interchangeably.

A rewriting system $\to_R$ is said to be *confluent* if for any string w there exists a unique irreducible string $\bar{w}$ such that $w \xrightarrow{*}_R \bar{w}$. If a rewriting system is *complete* i.e. both Noetherian and confluent. This is a desirable property, since any pair of strings w_1, w_2 can be reduced in a finite number of steps to their irreducible forms $\bar{w_1}$ and $\bar{w_2}$ which will be equal if and only if $w_1 =_R w_2$; i.e. the *word problem* is decidable. The Knuth-Bendix algorithm attempts to convert an arbitrary rewriting system into a complete one, by using an ordering on strings to ensure that the system is Noetherian and adding rules to the system to try to force confluence. The key concept here is that of *critical pairs*; i.e. pairs of reductions which can be applied to the same string to obtain two different results. The important critical pairs are associated with the *overlaps* of a rewriting system; i.e. strings w of X^*, such that either $w = l_1 v = u l_2$ or $w = u l_1 v = l_2$ for some $u, v \in X^*$. Here, (l_1, r_1), $(l_2, r_2) \in R$ are referred to as a *critical pair of rules of* R and the *critical pair of strings* corresponding to the overlap is $(r_1 v, u r_2)$ in the first overlap case and $(u r_1 v, r_2)$ in the second.

Definition 2.4 (Logged Critical Pairs)
An overlap occurs between the logged rewrites $\alpha_1 : l_1 \to r_1$ *and* $\alpha_2 : l_2 \to r_2$ *of* $\mathcal{L}$ *whenever one of the following is true:*

$$i) \quad u_1 l_1 v_1 = l_2, \quad ii) \quad u_1 l_1 = l_2 v_2, \quad iii) \quad l_1 v_1 = u_2 l_2, \quad iv) \quad l_1 = u_2 l_2 v_2.$$

for some $u_1, v_1, u_2, v_2 \in X^*$. *The* logged critical pair *resulting from the overlap is a whiskered pair* $(u_1 \alpha_1 v_1, v_2 \alpha_2 u_2)$ *for the appropriate* $u_1, v_1, u_2, v_2 \in X^*$.

3 Logged Completion

Given a monoid presentation $\mathcal{P} = mon\langle X|R\rangle$ we can associate to it an *initial logged rewrite system* $\mathcal{L}_{init}$ which consists of one 2-cell α for each rule (l, r) of R such that $\mathrm{src}(\alpha) = l$ and $\mathrm{tgt}(\alpha) = r$. These 2-cells are the generators of sesquigroupoid associated to the presentation.

If the initial logged rewrite system is not complete then we can attempt to transform it into a complete logged rewrite system by adding 2-cells which will make the underlying rewrite system complete in a version of the Knuth-Bendix algorithm which records information that is usually discarded. Clearly, this recorded completion terminates exactly when the usual completion procedure would terminate.

The immediate application for logged rewrite systems is in the provision of witnesses for computation. An ordinary complete rewrite system can determine whether or not two strings s_1 and s_2 represent the same element of the monoid; a logged rewrite system produces a proof in terms of a sequence of specific applications of the original monoid relations which will transform s_1 into s_2. This is a fairly shallow application, although variations on it are useful in more complex algorithms such as (Heyworth and Wensley, 1999).

Algorithm 3.1 (Logged Knuth-Bendix Procedure)

LKB1: *(Input) Let $\mathcal{P}$ be a presentation of a monoid with generators X and relations $(l_1, r_1), \ldots, (l_n, r_n)$ where $l_1 > r_1, \ldots, l_n > r_n$ for some well-ordering on the free monoid X^*. Define $\mathcal{L}_{init}$ to be the set of 2-cells or logged rules $\{\alpha_1, \ldots, \alpha_n\}$, where $\mathrm{src}(\alpha_i) = l_i$ and $\mathrm{tgt}(\alpha_i) = r_i$ for $i = 1, \ldots, n$.*

LKB2: *(Initialise) Set $\mathcal{L}_{all} = \mathcal{L}_{init}$; $\mathcal{L}_{new} = \mathcal{L}_{init}$; and let C be the empty list.*

LKB3: *(Search for Overlaps and Record Critical Pairs) Whenever an overlap occurs between the rewrites $\alpha_a \in \mathcal{L}_{all}$ and $\alpha_n \in \mathcal{L}_{new}$, record the associated critical pair by adding the element $(u_a, \alpha_a, v_a, u_n, \alpha_n, v_n)$ to the list C (where u_a, v_a, u_n or v_n may be the identity element).*

LKB4: *(Attempt to Resolve Critical Pairs) Set $\mathcal{L}_{new} = \emptyset$. For every element of C consider the pair $(u_a r_a v_a, u_n r_n v_n)$, reducing each string by $\mathcal{L}_{all}$ to the irreducible strings z_a and z_n respectively. If $z_a = z_n$ then the critical pair is said to resolve and it can be removed from C. Otherwise we must add a new logged rule to the system. If β_a and β_n are the logs of the reductions to z_a and z_n then the new logged rule is $\gamma = \beta_a^{-1} \cdot u_a\alpha_a^{-1}v_a \cdot u_n\alpha_n v_n \cdot \beta_n$ if $z_a > z_n$ and $\gamma = \beta_n^{-1} \cdot u_n\alpha_n^{-1}v_n \cdot u_a\alpha_a v_a \cdot \beta_a$ if $z_n > z_a$. Add γ to $\mathcal{L}_{new}$ and to $\mathcal{L}_{all}$.*

LKB5: *(Loop) If $\mathcal{L}_{new}$ is non-empty then loop to* LKB3. *Otherwise the procedure terminates: all critical pairs of $\mathcal{L}_{all}$ have been tested, and resolve.*

LKB6: *(Output) Output $\mathcal{L}_{all}$, a complete logged rewrite system for $\mathcal{P}$.*

---

4 Endorewrites

Deeper information about the presentation can be gained by studying the interaction of the relations with each other, known in group theory as the *identities among relations*. In the case of monoids, the structure is less simple than the kernel of a crossed module map. The identities themselves represent rewrite sequences which start at a word, send it through various transformations and return it to its original form. For the monoid case, we decided to refer to such rewrites as endorewrites.

Definition 4.1 (Endorewrites)
A 2-cell $\alpha \in C_2$ is an endorewrite *on a string w if* $\mathrm{src}(\alpha) = \mathrm{tgt}(\alpha) = w$.

The set of all endorewrites is actually the equaliser object of the two maps $\mathrm{src}, \mathrm{tgt} : C_2 \to X^*$ in the category of sets. We denote it EQ. Note that we continue our practice of writing $\alpha \in EQ \subseteq C_2$ and assuming the interchange law.

Lemma 4.2 (Endorewrite Structure)
The set of all endorewrites EQ is the disjoint union of the sets EQ_w for $w \in X^*$ where

$$EQ_w = \{\alpha : \mathrm{src}(\alpha) = \mathrm{tgt}(\alpha) = w\}.$$

Each EQ_w is closed under vertical composition; and their union EQ is additionally closed under horizontal composition and whiskering.

Vertical composition is defined only within subsets EQ_w. Horizontal composition is defined across subsets: if $\alpha \in EQ_w$ and $\alpha' \in EQ_{w'}$ then $\alpha \circ \alpha' \in EQ_{ww'}$. Whiskering means that for any substring s of a string w there is an injective mapping $EQ_s \to EQ_w$ defined by $\alpha \mapsto u\alpha v$ where $usv = w$.

Lemma 4.3 (Conjugate Endorewrites)
If $\theta(w_1) = \theta(w_2)$ then for every $\beta \in C_2$ such that $\mathrm{src}(\beta) = w_1$ *and* $\mathrm{tgt}(\beta) = w_2$ *there exists a bijection $\Phi_\beta : EQ_{w_1} \to EQ_{w_2}$ defined by $\alpha \mapsto \beta^{-1} \cdot \alpha \cdot \beta$.*

Thus the elements of $EQ_{w'}$ are all conjugates of elements of EQ_w so it becomes logical that we should only seek generators for endorewrites EQ_w of one representative string w for each monoid element $\theta(w)$. The next lemma helps to make this concrete.

Lemma 4.4 (Partial Action of Rewrites on Endorewrites)
A partial action of the monoid $(C_2, \cdot)$ on the set EQ can be defined by $\alpha^\beta = \beta^{-1}\alpha\beta$ for $\alpha \in EQ$ and $\beta \in C_2$ such that $\mathrm{src}(\alpha) = \mathrm{src}(\beta)$. *The partial action satisfies the following properties:*

i) $u\alpha^\beta v = (u\alpha v)^{u\beta v}$ *for* $u, v \in X^*$,
ii) $(\alpha_1 \cdot \alpha_2)^\beta = \alpha_1^\beta \cdot \alpha_2^\beta$.

Proof To verify that the partial action is well defined, we check that $(\alpha)^{(\beta_1 \cdot beta_2)} = (\alpha_1^\beta)_2^\beta$, which follows by the definition of the action together with the groupoid structure of $\cdot$. Further, $\alpha^{1_{\mathrm{src}\alpha}} = \alpha$ since $1_{\mathrm{src}\alpha}$ is an identity arrow. The other properties follow directly. □

Intuitively, α is like a circular walk; conjugating by β just means that we walk down a certain path to reach the start of the circular walk, and once the circle is done, retrace our steps back along that initial path. Clearly the circular walk is not much more interesting for having this initial path added to it and a tourist guidebook which promoted all conjugates of α as distinct and fascinating jaunts could be regarded as rather a swizz. Thus we factor EQ by this action: consider α to be equivalent to all its possible conjugates.

Lemma 4.5 (Classes of Endorewites)
Let C_2 be the 2-category of rewrites for a monoid presentation and let EQ be

the set of endorewrites. Then define

$$J = \{(\alpha, \beta^{-1} \cdot \alpha \cdot \beta) : \alpha \in EQ, \beta \in C_2 \textit{ and } \mathrm{src}\alpha = \mathrm{src}\beta\}.$$

Let $=_{I+J}$ be the smallest congruence with respect to $\cdot$ and whiskering which contains both J and the interchange law I. Then the quotient $EQ/=_{I+J}$ is well-defined, preserving both vertical composition and whiskering.

Note: as before, we shall write elements of EQ as $\alpha, \beta \in EQ \subseteq C_2$ with α and β being considered to represent the same endorewrite if $\alpha =_{I+J} \beta$; i.e. if they are equivalent under the interchange law together with the conjugation equivalence. To conclude this section we observe the following lemma.

Lemma 4.6 (Endorewrites form a $\mathbb{Z}M$-bimodule)
The conjugacy classes of endorewrites EQ^J form a $\mathbb{Z}M$-bimodule with respect to vertical composition and whiskering that is isomorphic to the image of EQ under $\Psi : C_2 \to \mathbb{Z}M[R]\mathbb{Z}M$ – the free $\mathbb{Z}M$-bimodule generated by R.

Proof Vertical composition of conjugacy classes of endorewrites gives an abelian group structure: if $\alpha_1, \alpha_2 : w \to w$ then

$$\alpha_1 \cdot \alpha_2 =_{I+J} \alpha_2 \cdot \alpha_1 \cdot \alpha_2 \cdot \alpha_2^{-1} = \alpha_2 \cdot \alpha_1.$$

The left and right whiskering actions of X^* on EQ^J resrtict to well-defined left and right actions of M since:

$$u_1\, \alpha\, v_1 =_{I+J} u_2\, \alpha v_2$$

when $\theta(u_1) = \theta(u_2)$ and $\theta(v_1) = \theta(v_2)$ since:

$$u_1\, \alpha\, v_1 = 1_{u_1} \cdot \alpha \cdot 1_{v_1} =_{I+J} 1_{u_2} \cdot \alpha \cdot 1_{v_2} = u_2\, \alpha\, v_2.$$

Compatibility of the addition $\cdot$ and the actions of whiskering were proved in the previous lemma. □

5 Critical Pairs

In this chapter we shall prove the intuitively reasonable idea that all distinct circular routes come from examining the reconnection of non-trivial diverging paths and thus provide a method for identifying all the interesting endorewrites of any completable rewrite system.

Lemma 5.1 (Resolved Critical Pairs Yield Endorewrites)
Let C be the set of all logged critical pairs and let EQ be the set of all endorewrites of a complete logged rewrite system. Then there is a well defined map $\delta : C \to EQ$.

Proof Suppose $c = (\alpha_1, \alpha_2)$ is a logged critical pair; i.e. there is a string w which may be rewritten in two ways – $\alpha_1 : w \to w_1$ and $\alpha_2 : w \to w_2$, where $\alpha_1, \alpha_2 \in C_2$. Since the pair can be resolved, there exists a string z so that $\beta_1 : w_1 \to z$ and $\beta_2 : w_2 \to z$, for some rewrite sequences β_1, β_2 in C_2. It is immediate that $\delta(c) = \alpha_1 \cdot \beta_1 \cdot \beta_2^{-1} \cdot \alpha_2^{-1}$ is an endorewrite on w. □

The following discussion will use the mapping δ, defined in the above lemma, to associate an endorewrite with every critical pair. Our main result requires that we first identify exactly what we mean by 'a generating set of endorewrites'.

A *generating set* for EQ is a set of endorewrites E such that any other endorewrite of EQ is equivalent under the interchange law together with the conjugacy congruence $=_{I+J}$, to a product of whiskered elements and inverse elements of E.

Definition 5.2 (Generating Set for EQ)
A generating set *for the endorewrites EQ associated with a monoid presentation is a set $E \subseteq EQ$ such that for any $\gamma \in EQ$ there exist $\alpha_1, \ldots, \alpha_n \in E$ such that $\gamma =_J u_1\alpha_1 v_1 \cdot \cdots \cdot u_n \alpha_n v_n$ for some $u_1, \ldots, u_n, v_1, \ldots, v_n \in X^*$ and $\epsilon_1, \ldots, \epsilon_n \in \{-1, 1\}$.*

Our main theorem claims that a set of generating endorewrites E, can be produced from the critical pairs which result from overlaps of the completed rewrite system. In order to prove the theorem we use digraph arguments, a digraph being associated with each endorewrite coming from a critical pair in the following way:

Lemma 5.3 (Digraphs and Endorewrites)
Given two words w and z, any pair of logged reductions $\alpha_1, \alpha_2 : w \to z$ can be represented by a labelled digraph which is associated uniquely with an endorewrite.

Proof

Let $\alpha_1, \alpha_2 : w \to z$ in C_2. Then we have a digraph $D(\alpha_1, \alpha_2)$, as shown, and the associated endorewrite $\delta(\alpha_1, \alpha_2) = \alpha_1 \alpha_2^{-1} \in EQ_w$; can be obtained by reading the labels anticlockwise from the edges, beginning at the vertex which is greatest with respect to $>$ on X^*.

$$\begin{tikzcd} w \arrow[d, bend right, "\alpha_1"'] \arrow[d, bend left, "\alpha_2"] \\ z \end{tikzcd}$$

□

We now observe that endorewrites resulting from critical pairs are trivial if the critical pair did not involve an overlapping of rules and a conjugate of the endorewrite resulting from the reduction of the minimal string on which the same overlap occurs, otherwise.

Lemma 5.4 (Overlaps and Endorewrites)
If $\alpha_1 : l_1 \to r_1$ and $\alpha_2 : l_2 \to r_2$ are rules of a complete logged rewriting system $\mathcal{L}$, which may both be applied to a string w then:

i) if the rules do not overlap then resolution of the critical pair yields the trivial identity,

ii) if the rules do overlap then the endorewrite of the critical pair is a whiskering of the endorewrite that is given by a resolution of the same pair of rules applied to the minimal string on which the same overlap occurs.

Proof

In the first case the rules do not overlap on w. Then there exist $x, y, z \in X^*$ such that $w = xl_1yl_2z$ and the logged reductions shown in the digraph on the right apply. This yields the endorewrite $x\alpha_1yl_2z \cdot xr_1y\alpha_2z \cdot xl_1y\alpha_2z \cdot x\alpha_1yr_2z$ which is equivalent under the interchange law to 1_w.

$$\begin{tikzcd}
& w \arrow[dl, bend right, "x\alpha_1 y l_2 z"'] \arrow[dr, bend left, "x l_1 y \alpha_2 z"] & \\
x r_1 y l_2 z \arrow[dr, bend right, "x r_1 y \alpha_2 z"'] & 1_w & x l_1 y r_2 z \arrow[dl, bend left, "x \alpha_1 y r_2 z"] \\
& x r_1 y r_2 z &
\end{tikzcd}$$

In the second case, the rules do overlap on w and there exist $u_1, v_1, v_2, x, y, z \in X^*$ such that $w = xyz$ and either $y = u_1l_1v_1 = l_2$ or $y = u_1l_1 = l_2v_2$. In either case we can write $y = u_1l_1v_1 = l_2v_2$ and the logged reductions of y are $u_1\alpha_1v_1 : y \to u_1r_1v_1$ and $\alpha_2v_2 : y \to r_2v_2$. By completeness there are logged reductions $\beta_1 : u_1r_1v_1 \to \bar{w}$ and $\beta_2 : r_2v_2 \to \bar{w}$ such that $\gamma = u_1\alpha_1v_1 \cdot \beta_1 \cdot \beta_2^{-1} \cdot \alpha_2^{-1}u_2$ is an endorewrite – the digraph is shown the right. The critical pair of reductions on w are $xu_1\alpha_1v_1z : w \to xu_1r_1v_1z$ and $x\alpha_2u_2z : w \to xr_2v_2z$. They can be resolved by $x\beta_1z : xu_1r_1v_1z \to x\bar{w}z$ and $x\beta_2z : xr_2v_2z \to x\bar{w}z$. The digraph is shown on the right and the associated endorewrite is $x\gamma z$.

$$\begin{tikzcd}
& w \arrow[dl, bend right, "u_1\alpha_1 v_1"'] \arrow[dr, bend left, "\alpha_2 v_2"] & \\
x u_1 r_1 v_1 \arrow[dr, bend right, "\beta_1"'] & \gamma & r_2 v_2 \arrow[dl, bend left, "\beta_2"] \\
& \bar{w} &
\end{tikzcd}$$

$$\begin{tikzcd}
& xwz \arrow[dl, bend right, "x u_1 \alpha_1 v_1 z"'] \arrow[dr, bend left, "x \alpha_2 v_2 z"] & \\
x u_1 r_1 v_1 z \arrow[dr, bend right, "x\beta_1 z"'] & x\gamma z & x r_2 v_2 z \arrow[dl, bend left, "x\beta_2 z"] \\
& x\bar{w}z &
\end{tikzcd}$$

□

Lemma 5.5 (Digraph of Reduction Sequences)
For any critical pair of logged reduction sequences, there exists a finite digraph which is the union of digraphs resulting from resolving critical pairs as in Lemma 5.4.

Proof Given two logged reduction sequences $\alpha : w \to w_1 \to \cdots \to w_m \to z$ and $\alpha' : w \to w_{m+1} \to \cdots \to w_n \to z$, we define a digraph D. The vertices $V(D)$ are the distinct words occurring in these sequences, and there is an edge labelled α_i from w_i to w_j if $w_i \to w_j$ is a reduction step labelled by α_i in one of the two given reduction sequences. The pair of reduction sequences (α, α') yield the endorewrite $\gamma = \delta(\alpha, \alpha')$ in the way described in Lemma 5.4. We now add to the graph (if the graph is drawn, this looks like subdivison into small confluence diagrams, the proof was originally phrased in 'diamonds'). Note that the vertices are ordered with respect to $>$ in X^*.

Digraph Filling/Construction

D1: (Initialise) Given D be as defined above, set $V_1 = V(D)$ and $i = 1$.

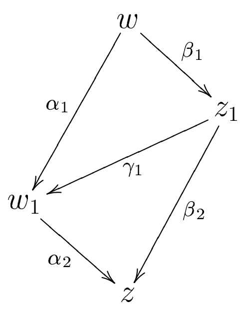
D2: (Add to the Digraph) Remove the maximum vertex v_i from V_i. If v_i is the source of two distinct arrows of D then we consider the corresponding two reductions $\beta_{i,1} : v_i \to v_{i,1}$ and $\beta_{i,2} : v_i \to v_{i,2}$ The critical pair can be resolved since $\mathcal{L}$ is a complete rewrite system so we have $\beta_{i,1} : v_{i,1} \to z_i$ and $\beta_{i,2} : v_{i,2} \to z_i$ Define D_i to be the digraph consisting of all the vertices and arrows considered in this step. Add D_i to D, identifying the vertices which have the same labels.

D3: (Loop) Increment i and repeat step D2.

We note firstly that $\mathcal{L}$ is finite, so there are only finitely many rules which can be applied; secondly, any finite word can only be reduced in a finite number of ways; finally, the system is Noetherian, so there are no infinite reduction sequences. This means that the procedure will terminate, giving a finite digraph D which is the union of the digraphs D_i, which are all of the type in Lemma 5.4. □

Lemma 5.6 (Digraph Compositions)
The product (at the base point) of the endorewrites associated (in the sense of Lemma 5.3) with the sub-digraphs is equivalent under the interchange law to the endorewrite associated with the original digraph.

Proof Consider the composition of digraphs of the type described, remembering that each edge is associated uniquely to a particular log of the reduction.



The endorewrites associated to the two digraphs are $\alpha_1{\cdot}\gamma_1^{-1}{\cdot}\beta_1^{-1}$ and $\gamma_1{\cdot}\alpha_2{\cdot}\beta_2^{-1}$. Composing them from the base point w gives us $\alpha_1{\cdot}\gamma_1^{-1}{\cdot}\beta_1^{-1}{\cdot}\beta_1{\cdot}(\gamma_1{\cdot}\alpha_2{\cdot}\beta_2^{-1}){\cdot}\beta_1^{-1}$ which is equivalent in the sesquigroupoid to $\alpha_1 \cdot \alpha_2 \cdot \beta_2^{-1} \cdot \beta_1^{-1}$, the endorewrite given by taking the boundary of the composite. The fact that the order of the digraph endorewrites is not important corresponds with the fact that EQ is abelian. □

Combining Lemma 5.5 and Lemma 5.6 with Lemma 5.4 we can deduce that any digraph can be identified with a product of whiskered endorewrites and inverse endorewrites of E. This allows us to prove the main theorem:

Theorem 5.7 (Critical Pairs give a Set of Generators for EQ)
Let $\mathcal{L}_{init}$ be the initial logged rewrite system for a monoid presentation, and let $\mathcal{L}_{comp}$ be a completion. Let C be the set of all logged critical pairs resulting from overlaps of the system $\mathcal{L}_{comp} \cup \mathcal{L}_{init}$. Then

$$E = \{\delta(c) : c \in C\}$$

is a generating set of endorewrites.

Proof Let γ be an endorewrite on some string w. Then consider the critical pair $(\gamma, 1_w)$. Using Algorithm 5.5 we can construct a digraph D whose associated endorewrite is γ and whose sub-digraphs yield a product of whiskered elements of E and their inverses which is equivalent to γ by Lemma 5.6. □

6 Example

In this example we show how a complete set of generators for the endorewrites of a monoid presentation can be constructed from the overlaps of a complete logged rewriting system. We use Lemma 4.6 to enable a reduction in the size of this generating set. Note: we omit the operation symbol $\cdot$ when writing the whiskers, it being clear which part of the expression is a string and which is a 2-cell.

Consider the monoid presentation

$$mon\langle e, s \mid e^2 = e, s^3 = s, s^2e = e, es^2 = e, sese = ese, eses = ese\rangle.$$

Using the short-lex ordering with $s > e$, labelling the relations $\alpha_1, \ldots, \alpha_6$ we have the complete logged rewriting system $\alpha_1 : e^2 \to e$, $\alpha_2 : s^3 \to s$, $\alpha_3 : s^2e \to e$, $\alpha_4 : e^2s \to e$, $\alpha_5 : sese \to ese$, $\alpha_6 : eses \to ese$.

By considering all the overlaps of the logged system the following 26 endorewrities can be deduced.

The endorewrites of e are: $\alpha_2 se \cdot s^2\alpha_3^{-1}$, $\alpha_1 s^2 \cdot \alpha_1^{-1} \cdot e\alpha_4^{-1}$, $\alpha_1 e \cdot e\alpha_1^{-1}$, $\alpha_3 e \cdot \alpha_1 \cdot \alpha_3^{-1} \cdot s^2\alpha_1^{-1}$, $\alpha_3 s^2 \cdot \alpha_4 \cdot \alpha_3^{-1} \cdot s^2\alpha_4^{-1}$, $\alpha_4 s^2 \cdot es\alpha_2^{-1}$ and $\alpha_4 e \cdot e\alpha_3^{-1}$.

The endorewrites of ese are: $\alpha_1 ses \cdot \alpha_6 \cdot \alpha_1^{-1} se \cdot e\alpha_6^{-1}$, $s\alpha_5 \cdot \alpha_5 \cdot \alpha_3^{-1} se$, $\alpha_3 ses \cdot s^2\alpha_6^{-1}$, $\alpha_4 se \cdot es\alpha_3^{-1}$, $\alpha_5 e \cdot ses\alpha_1^{-1}$, $\alpha_5 s^2 \cdot es\alpha_4 \cdot \alpha_5^{-1} \cdot ses\alpha_4^{-1}$ and $\alpha_5 s \cdot \alpha_6 \cdot \alpha_5^{-1} \cdot s\alpha_6^{-1}$.

The endorewrites of s are: $\alpha_5 ses \cdot \alpha_6 es \cdot es\alpha_1 s \cdot \alpha_6 \cdot \alpha_5^{-1} \cdot s\alpha_1^{-1} se \cdot se\alpha_5^{-1} \cdot ses\alpha_6^{-1}$, $\alpha_2 s^2 \cdot s^2\alpha_2^{-1}$ and $\alpha_6 se \cdot \alpha_6 e \cdot ese\alpha_3^{-1}$.

The endorewrites of s^2 are: $\alpha_6 s^2 \cdot es\alpha_4 \cdot \alpha_6^{-1} \cdot ese\alpha_2^{-1}$, $\alpha_2 s \cdot s\alpha_2^{-1}$ and $\alpha_6 s \cdot es\alpha_4^{-1}$.

The endorewrites of es are: $\alpha_6 e \cdot es\alpha_1 \cdot \alpha_1^{-1} se \cdot e\alpha_5^{-1}$, $\alpha_4 s \cdot e\alpha_2^{-1}$ and $\alpha_6 ese \cdot ese\alpha_5$.

Finally, the endorewrites of se are: $\alpha_6 es \cdot es\alpha_1 s \cdot \alpha_6 \cdot es\alpha_1^{-1} \cdot \alpha_6^{-1} e \cdot es\alpha_6^{-1}$, $\alpha_2 e \cdot s\alpha_3^{-1}$ and $\alpha_5 se \cdot e\alpha_5 \cdot \alpha_1 se \cdot \alpha_5^{-1} \cdot s\alpha_1^{-1} se \cdot se\alpha_5^{-1}$.

7 Homotopical and Homological Interpretations

We promised, in the introduction, that our results would enable homotopical and homological specifications of the monoid. It is well known that the existence of a finite complete rewriting system for a monoid presentation implies the homological finiteness conditions FP_3 (Squier, 1987) and the stronger condition FP_∞ (Anick, 1986; Kobayashi, 1990) as well as the homotopical condition of having finite derivation type (FDT) (Cremanns, 1995; Squier, 1994). The addition made by this paper, in considering logged rewriting systems, is that our algorithms enable the specification of the structures which the properties are based upon.

In the homotopical case, it is immediate to observe that the set E of generating endorewrites suffices as a set of homotopy generators in the sense of (Cremanns, 1995). In detail: if α is any cycle of the graph whose objects are all strings and whose invertible edges are all rewrites, then α corresponds to the digraph of an endorewrite and it turns out that the product of the subdigraphs is homotopically equivalent to α for the same reasons as the associated endorewrite is equivalent to the composite of the endorewrites of the subdigraphs.

In terms of homology, the specification of E, similar to the analogous case of Π_2 for groups, enables us to construct a resolution. Specifically, we have an exact sequence of free, finitely generated $\mathbb{Z}M$-modules:

$$C_2 \xrightarrow{\delta_2} C_1 \xrightarrow{\delta_1} C_0 \xrightarrow{\delta_0} \mathbb{Z} \longrightarrow 0$$

Given our specification of a finite set of homotopy generators, further details of the resolution can be found in (Cremanns, 1995) in the proof of the fact that FDT implies FP_3.

Potentially more interesting is to consider the generalisation to monoids of a crossed resolution for groups. Recall the group case:

A *crossed complex (over groupoids)* is a sequence C

$$\cdots \xrightarrow{\delta_{n+1}} C_n \xrightarrow{\delta_n} C_{n-1} \qquad \cdots \xrightarrow{\delta^3} C_2 \xrightarrow{\delta_2} C_1 \xrightarrow{\delta_1} C_0$$

such that

i) C_1 is a groupoid with C_0 as its set of vertices and δ^1, δ^0 as its source and target maps.

ii) For $n \geqslant 2$, C_n is a totally disconnected groupoid over C_0 and for $n \geqslant 3$, the groups at the vertices of C_n are abelian.

iii) The groupoid C_1 operates on the right of each C_n for $n \geqslant 2$ by an action denoted $(x, a) \mapsto x^a$.

iv) For $n \geqslant 2$, $\delta_n : C_n \to C_{n-1}$ is a morphism of groupoids over C_0 and C_1 acts on itself by conjugation.

v) $\delta_n \delta_{n-1} = 0 : C_n \to C_{n-2}$ for $n \geqslant 3$ and $\delta_2 \delta^0 = \delta_2 \delta^1 : C_2 \to C_0$.

vi) If $c \in C_2$ then $\delta_2(c)$ operates trivially on C_n for $n \geqslant 3$ and operates on C_2 by conjugation by c.

A crossed complex C is *free* if C_1 is a free groupoid (on some graph Γ_1) and C_2 is a free crossed C_1-module (for some $\lambda : \Gamma_2 \to C_1$) and for $n \geqslant 3$, C_n is a free $\pi_1 C$-module on some Γ_n where $\pi_1 C$ is the fundamental groupoid of the crossed complex; i.e. the quotient of the groupoid C_1 by the normal, totally disconnected subgroupoid $\delta_2(C_2)$.

A crossed complex C is *exact* if for $n \geqslant 2$

$$Ker(\delta_n : C_n \to C_{n-1}) = Im(\delta_{n+1} : C_{n+1} \to C_n).$$

If C is an free exact crossed complex and G is a groupoid then C together with an isomorphism $\pi_1 C \to G$ (or, equivalently, C with a quotient morphism $C_1 \to G$ whose kernel is $\delta_2(C_2)$) is called a *corssed resolution of* G. It is a *free crossed resolution* if C is also free.

In the case of monoids, we propose a similar structure, but instead of a right action of C_1 we have a two-sided action; instead of a crossed module $\delta_2 : C_2 \to C_1$ we have a 2-category $\mathrm{src}, \mathrm{tgt} : C_2 \to C_1$ and instead of C_1 being a groupoid, it is a category.

This section needs finishing still - any suggestions/comments welcome – currently studying Steve Pride's papers on lower dimensional topology for monoids.

8 Generalised Logged String Rewriting

In (Brown and Heyworth, 2000) it was shown that the familiar string rewriting methods can be applied to problems of computing left Kan extensions over the category of sets. Structures such as monoid and category presentations, induced actions of groups and monoids, equivalence and conjugacy classes, equalisers and pushouts all turn out to be special cases of left Kan extensions over Sets and thus string rewriting methods can be applied to all these variations on the word problem.

Since string rewriting for Kan extensions can be achieved by embedding in a monoid, it is unnecessary to go through the detail of the sesquigroupoid whose 2-cells possess the structure for the logged rules. However, since we don't need to embed in a monoid in order for the string rewriting methods to work, we briefly outline the alternative sesquigroupoid.

Let (E, ϵ) be the left Kan extension of the category action $X : \mathsf{A} \to \mathsf{Sets}$ along the functor $F : \mathsf{A} \to \mathsf{B}$. We assume that the data for the Kan extension is given as a finite presentation, consisting of generating graphs for A and B, a set of relations for B and the action of functors F and X being defined for every object and arrow of the generating graph of A. The sesquicategory C_2 associated with the presentation of the Kan extension has 0-cells $\bigcup_{A \in \mathrm{Ob}\mathsf{A}} XA \cup \mathrm{Ob}\mathsf{B}$ and 1-cells $\{(s_x : x \to FA) \,|\, x \in XA, A \in \mathrm{Ob}\mathsf{A}\} \cup \mathrm{Arr}\mathsf{B}$. The 2-cells are the rewrites and inverse rewrites, with vertical composition as before, but clearly, whiskering and horizontal compositions are partial operations dependent on whether paths can be composed.

In conjunction with (Brown and Heyworth, 2000), this observation enables logged rewriting techniques to be applied to a wide range of problems, including category presentations, equivalence relations, induced actions, pushouts and coset systems. Interpretations and potential applications of the endorewrites in each case requires further investigation.

9 Implementations and Further Applications

Techniques of logged rewriting have been implemented by the first author as GAP functions which will eventually be submitted as a package. Applications of logged rewriting were explored in (Heyworth and Wensley, 1999) where the group version was implemented, providing a new algorithmic method for the construction of crossed resolutions of groups; in (Ghani and Heyworth, 2003) where the logged completion methods give an alternative to the Reidemeister-Schreier method of computing a subgroup presentation; and in (Brown et al, 2004) we show how endorewrites for double coset rewriting systems reveal information about the subgroups.

Further work will pursue other potential applications, including in Petri nets, concurrency and the analysis of knot quandles; as well as generalising the techniques to Gröbner bases where the endorewrites can be identified with syzygies.

References

D. J. Anick, *On the Homology of Associative Algebras,* Transactions of the American Mathematical Society, vol.296 p641-59 1986.

F. Baader and T. Nipkow, *Term Rewriting and All That,* Cambridge University Press 1998.

D.B. Benson, *The Basic Algebraic Structures in Categories of Derivations,* Information and Control, vol. 28, no. 1, Academic Press 1975.

R. Book and F. Otto, *String-Rewriting Systems,* Springer-Verlag, New York 1993.

R. Brown and A. Heyworth, *Using Rewrite Systems to Compute Kan Extensions and Induced Actions of Categories*, J. Symbolic Computation 29 5-31, 2000.

R. Brown, N. Ghani, A. Heyworth and C.D. Wensley, *String-rewriting for Double Coset Systems,* submitted J. Symbolic Computation 2004.

R. Brown and A Razak Salleh, *Free Crossed Resolutions of Groups and Presentations of Modules of Identities Among Relations,* Journal of Mathematical Computation, LMS, vol.2 p28-61 1999.

B. Buchberger and F. Winkler, *Gröbner Bases and Applications,* "33 Years of Gröbner Bases" RISC-Linz 2-4 Feb 1998, Proc. London Math. Soc. vol.251.

D. Cohen, *String Rewriting and Homology of Monoids,* Math. Struct. Comput. Sci. 7 207-240 1997

R. Cremanns, *Finiteness Conditions for Rewriting Systems* PhD thesis, Universität Gesamthochschule Kassel, 1995.

R. Cremanns and F. Otto, *Finite Derivation Type implies the Homological Finiteness Condition FP_3,* Journal of Symbolic Computation, vol.18 p91-112 1994.

R. Cremanns and F. Otto, *Finite Derivation Type is equivalent to FP_3 for Groups,* Journal of Symbolic Computation, vol.22 p155-177 1996.

The GAP Group, *GAP – Groups, Algorithms, and Programming, Version 4*, Aachen, St Andrews, 1998. `http://www.gap.dcs.st-and.ac.uk/~gap`.

N. Ghani and A. Heyworth, *A Rewriting Alternative to the Reidemeister-Schreier Procedure*, RTA 2003.

N. D. Gilbert, *Monoid Presentations and Associated Groupoids,* Int. J. Algebra and Computation, 8 141-152 1998.

J. R. J. Groves, *An Algorithm for Computing Homology Groups, Journal of Algebra, vol.94 p331-361* 1997.

A. Heyworth and C.D. Wensley, *Logged Rewriting and Identities Among Relators*, in Groups St Andrews 2001 in Oxford, eds. C.M. Campbell, E.F. Robertson, G.C. Smith, London Math. Soc. Lecture Note Ser. 304, C.U.P. p256-76 2003.

Y. Kobayashi, *Complete Rewriting Systems and Homology Of Monoid Algebras,* Journal of Pure and Applied Algebra, vol.65 p263-275 1990.

Y. Lafont, *A New Finiteness Condition for Monoids Presented by Complete Rewriting Systems (after Craig C. Squier),* Journal of Pure and Applied Algebra, vol.98 p229-244 1995.

Y. Lafont and A. Prouté, *Church Rosser Property and Homology Of Monoids,* Mathematical Structures in Computer Science vol.1, Cambridge University Press p297-326 1991.

B. Mitchell, *The 2-category of a Set of Relations,* in 'Rings with several objects', Advances in Math. 8 p74-77 1972.

S. J. Pride, *Low-dimensional Homotopy Theory for Monoids,* International Journal of Algebra and Computation, vol.5 p631-649 1993.

S. J. Pride, *Low-dimensional Homotopy Theory for Monoids II,* Preprint, University of Glasgow 1995.

B. Reinert, *On Gröbner Bases in Monoid and Group Rings,* PhD Thesis, Universität Kaiserslautern 1995.

B. Reinert and D. Zecker, *MRC - A System for Computing Gröbner Bases in Monoid and Group Rings,* Universität Kaiserslautern Preprint 1998.

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

C. C. Squier, *Word Problems and a Homological Finiteness Condition for Monoids,* Journal of Pure and Applied Algebra, vol.49 p201-17 1987.

C. C. Squier, F. Otto and Y Kobayashi, *A Finiteness Condition for Rewriting Systems,* Theoretical Computer Science, vol.131 p271-94 1994.

J. G. Stell, *Modelling Term Rewriting Systems by Sesquicategories,* Technical Report TR94-02, University of Keele 1994.

R. Street, *Categorical Structures,* in 'Handbook of Algebra', M. Hazelwinkel (ed) vol.1 1992.

A. Tonks, *Theory and Applications of Crossed Complexes,* PhD thesis, University of Wales, Bangor 1993.