Efficient Finitely Presented Groups Completion Using PATCH Graphs

Christopher Lynch
INRIA Lorraine - CRIN

Polina Strogova
INRIA Lorraine - CRIN and INRIA Rocquencourt

Abstract. Based on a new data structure called PATCH Graph, an efficient completion procedure for finitely presented groups is described. A PATCH Graph represents rules and their symmetrized forms as cycles in a Cayley graph structure. Completion is easily performed directly on the graph, and structure sharing is enforced. The structure of the graph allows us to avoid certain redundant inferences. The PATCH Graph data structure and inference rules complement other extensions of Knuth-Bendix completion for finitely presented groups.

Key words: String Rewriting, Finitely Presented Group, Symmetrization Procedure, PATCH Graph

1. Introduction

1.1. Motivation and Related Work

Important problems in computer algebra have been solved by applying the Knuth-Bendix completion procedure [Knuth and Bendix1970]. In particular it is useful for solving the word problem in a given finite presentation of an algebra. We will focus on computational group theory, where completion has been used to solve other problems. For example, finding a confluent presentation [Gilman1979], testing whether a finitely presented group is a finite group [Gilman1979], verifying nilpotency [Sims1987], and determining whether two finitely presented groups are isomorphic [Holt and Rees1989]. Also see [Linton and Shand1994] for more examples and some benefits of Knuth-Bendix completion over other methods.

Our interest in Knuth-Bendix completion arises from the shortest path routing problem in Cayley Graphs. Cayley Graphs, based on group structure, represent an attractive model for interconnection networks. They are symmetric, dense, and have small diameters. Some classes of Cayley Graphs, such as hypercube, shuffle-exchange and de Bruijn networks for example, are parallel architectures with an effective routing. For all these classes of networks, existing routing algorithms take advantage of specific properties; the concrete form of group presentation for example. In [Fortin *et al.*1994] we described a general routing algorithm for all Cayley Graphs of permutation groups. The algorithm enables us to find a shortest path between two given nodes of a Cayley Graph, starting only with generators of the permutation group. To get a shortest path, we normalize a path, resulting from Todd-Coxeter coset enumeration, with respect to a canonical system describing the graph. Such a canonical system exists, since Knuth-Bendix completion procedure for finite groups, using a total reduction ordering, always succeeds. Applying the standard Knuth-Bendix algorithm were able to complete rewriting systems for groups of order up to 10^4 using RRL [Kapur and Zhang1988], Herky [Zhang1992] and Otter [McCune1994] automated theorem provers. For larger systems, we need a more efficient completion algorithm specialized for group completion.

Several improvements of the Knuth-Bendix completion procedure have been developed for completion of finitely presented groups. Some of these algorithms extend the Knuth-Bendix completion procedure, to create an algorithm that halts for presentations where Knuth-Bendix completion does not. The disadvantage of these algorithms is that they only work for certain classes of groups. In [Madlener *et al.*1991], the confluence property is weakened to only require that a string equivalent to the identity can be rewritten to the identity. An algorithm is given that works for monadic groups. In [Epstein *et al.*1991a], automata theory techniques are used to construct an automata representing an infinite rewrite system for automatic groups.

In this paper, instead of extending the Knuth-Bendix procedure, we choose to make it more efficient. Therefore, our algorithm halts for the same presentations as Knuth Bendix, but it still works for all presentations. Another paper applying this philosophy is [Martin1996], which focuses on finding good orderings. Le Chenadec [Le Chenadec1986] remarks that the completion procedure founded on group reduction relation runs faster than the standard completion procedure. According to group reduction, a string is firstly reduced by a ground relation of group presentation and then simplified by group axioms. A similar relation, called normalized rewriting, is proposed by C. Marché [Marché1993]. Also, Le Chenadec shows how to avoid inferences with the group axioms. He introduces the symmetrization procedure, specifying for groups the techniques of [Bücken1979].

Our procedure builds on Le Chenadec's ideas. We design special data structures so that the inferences can be performed efficiently. These data structures also use automatic structure sharing to store the data efficiently, and allow us to avoid performing certain redundant inferences. Our philosophy is to express the rewriting on the object level, as in [Lynch1995] for paramodulation, and in [Lynch and Strogova1995] for term completion. We think our approach could be combined with the previously mentioned algorithms. In fact, our data structure may even be helpful in creating the automata used in [Epstein *et al.*1991b].

1.2. Our Data Structure

The data structure we propose in this paper is called a *PATCH Graph*. A PATCH Graph represents in a very natural and compact way a string rewriting system for a group. The idea is to store strings as paths, rewriting rules and equations as cycles (this interpretation is inspired by Cayley Graphs), and perform completion inference rules on cycles. Actually, an equation $a \cdot b = c$ can be seen as a cycle formed by concatenation of directed edges $e_1 = (v_1, v_2), e_2 = (v_2, v_3)$ and $e_3 = (v_1, v_3)$ labelled respectively a, b and c. The first advantage of this representation is that the same cycle can be seen as the concatenation of edge e_1, edge e_3, and edge e_2, where e_2 is taken backwards, i.e. as (v_3, v_2). Reading out the edge labels, we get $a = c \cdot b^{-1}$. Equation $a = c \cdot b^{-1}$ is one of the symmetrized forms of equation $a \cdot b = c$.

In this way, if n equals the number of letters in the left-side plus the number of letters in the right-hand side of a rule or an equation R, we can represent all the $O(n)$ symmetrized forms of R by one cycle of length n. Thereby, performing one inference on two cycles of respective lengths m and n, we calculate at one go $O(n \cdot m)$ critical pairs or simplifications for the rules represented by these cycles. This is the second advantage. Finally, from the technical point of view, operating on strings rather than terms avoids additional computations due to the associativity group axiom.

1.3. Group Completion

We formulate inference rules **Symmetrize**, **Patch** and **Eliminate** for group completion, based on PATCH Graphs. As its name prescribes, **Symmetrize** inference rule expresses the symmetrization procedure. **Patch** inference rule gives the law of creation of new cycles. **Eliminate** inference rule encodes cancellation and simplification with respect to group axioms.

A new cycle C in a PATCH Graph is generated from two cycle-ancestors C_1 and C_2, having common paths π_1 and π_2 (i.e. the paths representing the same string). Cycle C is as a combination of path $C_1 \backslash \pi_1$, which complements path π_1 to cycle C_1, and of path $C_2 \backslash \pi_2$, which complements path π_2 to cycle C_2. **Patch** inference rule is performed if and only if paths π_1 and π_2 are maximal non trivial common paths.

Paths π_1 and π_2 are maximal, if they are not strict subpaths of other common paths for C_1 and C_2. Actually, rewriting rules represented by a cycle C, created for maximal common paths π_1 and π_2, simplify all the rules and equations, represented by a cycle C', generated for non-maximal common subpaths of paths π_1 and π_2. Paths π_1 and π_2 are trivial, if $C_1 = C_2$ and $\pi_1 = \pi_2$. A new cycle for trivial common paths π_1 and π_2 represents a critical pair between two symmetrized forms of the same rule. Such a critical pair can be simplified by group axioms to a trivial equation.

Consequently, using our inference rules avoids generating some redundant equations and rules, whereas they are required in standard completion and Le Chenadec's completion procedure. These points enable us to expect a considerable improvement of the completion procedure for groups. We have started an implementation of the algorithm, and we intend to include benchmarks in the final version of the paper.

1.4. Contents of the Paper

The paper is organized in eight sections. Important notations and preliminary results are given in Section 2. After introducing in Section 3 the syntax and the semantics of PATCH Graphs, we formulate in Section 4 the inference rules for the completion procedure based on PATCH Graphs. We explain how do inference rules work on a number of examples. We prove the soundness and the completeness of the inference system in Section 5. In Section 6 we give an idea on the matching procedure on PATCH Graphs, which we use in the completion algorithm formulated in Section 7. In Section 8 we summarize the improvements of the group completion procedure by using PATCH Graphs.

2. Definitions and Preliminary Results

We recall in this section some definitions and results relative to string rewriting and group completion, according to the formalism of [Book and Otto1993] and [Le Chenadec1986], and some basic notions of graph theory.

2.1. String Rewriting

Strings Let G be a finite alphabet of constants called *letters*. Consider a binary *concatenation* operation on G, denoted $\cdot$ in infix notation. Set G^* of all finite concatenations of letters, called *strings*, is the free monoid on G, with the empty string (denoted ε) as neutral element.

For any string s, a string t is a *prefix* of s if there exists a string $u \in G^*$ such that $s = t \cdot u$. A string v is a *suffix* of string s if there exists a string $w \in G^*$ such that $s = w \cdot v$. A string t is a *cyclic permutation* of string s if there exist strings $u, v \in G^*$ such that $s = u \cdot v$ and $t = v \cdot u$. String t is a *substring* of string s if $s = u \cdot t \cdot v$ for some strings $u, v \in G^*$. The *length* of string s, denoted $length(s)$, is the number of letters in s.

Rewriting Systems An *equation* E on G^* is a pair $(l, r) \in G^* \times G^*$, denoted $E: \ l = r$. A *rewriting rule* R on G^* is an ordered pair $(l, r) \in G^* \times G^*$, denoted $R: \ l \to r$. If (l, r) is a rule or an equation, then string l is called the *left-hand side (lhs)* of R, and string r is called *right-hand side (rhs)* of R. A *string rewriting system* (or *rewriting system* for short) $\mathcal{R}$ on G^* is a (finite) set of rewriting rules on G^*.

Relations Induced by Rewriting Systems Let $\mathcal{R}$ be a rewriting system. Relation $\to_{\mathcal{R}}$ such that for all strings $s, t \in G^*$, $s \to_{\mathcal{R}} t$ iff there exists a rule $R: \ l \to r \in \mathcal{R}$, and strings $u, v \in G^*$, such that $s = u \cdot l \cdot v$, and $t = u \cdot r \cdot v$, is called the *single-step reduction relation* induced by $\mathcal{R}$. In this case we say that rule $R: \ l \to r$ *simplifies* string s. Rule $R: \ l \to r$ *simplifies* a rule or an equation $(s, t) \in G^*$ if R simplifies s or t. String s is $\mathcal{R}$-*irreducible* if there is no string t such that $s \to_{\mathcal{R}} t$.

Let $\xrightarrow{*}_{\mathcal{R}}$ denote the reflexive transitive closure of relation $\to_{\mathcal{R}}$. Relation $\xrightarrow{*}_{\mathcal{R}}$ is called the *reduction relation* induced by rewriting system $\mathcal{R}$. The reflexive transitive symmetric closure $\overset{*}{\leftrightarrow}_{\mathcal{R}}$ of relation $\to_{\mathcal{R}}$ is called the *congruence relation*, induced by rewriting system $\mathcal{R}$.

Single-step reduction relation relation $\to_{\mathcal{R}}$ is called *noetherian* if there is no infinite sequence of string reductions $s_1 \to_{\mathcal{R}} s_2 \to_{\mathcal{R}} \ldots$. Rewriting system $\mathcal{R}$ is *noetherian* if single-step reduction relation $\to_{\mathcal{R}}$ is noetherian.

Rewriting system $\mathcal{R}$ is called *Church-Rosser* if for all strings $s, t \in G^*$, $s \overset{*}{\leftrightarrow}_{\mathcal{R}} t$ implies $s \xrightarrow{*}_{\mathcal{R}} u$ and $u \xleftarrow{*}_{\mathcal{R}} t$ for some string $u \in G^*$. A noetherian Church-Rosser rewriting system $\mathcal{R}$ is called *canonical*.

Orderings An irreflexive antisymmetric transitive binary relation on G^*, denoted $>$ in infix notation, is called a *strict partial ordering*. A strict partial ordering $>$ is

- *well-founded* if there is no infinite decreasing sequence $s_1 > s_2 > \ldots$ in G^* ;
- a *reduction ordering* if $s > t$ implies $u \cdot s \cdot v > u \cdot t \cdot v$ for all strings $u, v \in G^*$, and $>$ is well-founded;
- *compatible* with a rewriting system $\mathcal{R}$ if $l > r$ for each rule $R: \ l \to r$ in $\mathcal{R}$.

An example of a well-founded ordering is the *length-lexicographical ordering* $>_{ll}$. Let $g_1 > \ldots > g_n$ be a precedence on letters. Then for all strings s and t, $s >_{ll} t$ iff

- either $length(s) > length(t)$,

– or $length(s) = length(t)$ and there are strings $u, v, w \in G^*$ such that $s = u \cdot g_i \cdot v$ and $t = u \cdot g_j \cdot w$ for $g_i > g_j$.

Completion Procedure Let $R_1 :\ u \to v,\ R_2 :\ l \to r$ be two rewriting rules. A pair $(v \cdot s, p \cdot r)$ such that $u = p \cdot c$ and $l = c \cdot s$ for some strings $c, p, s \in G^*$, is called a *critical pair* between rules R_1 and R_2[1]. A critical pair $(v \cdot s, p \cdot r)$ is said to be *joinable* with respect to rewriting system $\mathcal{R}$, if there is a string $w \in G^*$ such that $v \cdot s \xrightarrow{*}_{\mathcal{R}} w$ and $w \xleftarrow{*}_{\mathcal{R}} p \cdot r$.

A completion procedure for a finite noetherian rewriting system $\mathcal{R}$ tries to construct a canonical rewriting system $\mathcal{R}'$, such that congruence relations $\overset{*}{\leftrightarrow}_{\mathcal{R}}$ and $\overset{*}{\leftrightarrow}_{\mathcal{R}'}$ are the same. The standard Knuth-Bendix procedure [Knuth and Bendix1970] takes a reduction ordering $>$, compatible with rewriting system $\mathcal{R}$, and adds to the system all non joinable critical pairs oriented into rules with respect to ordering $>$. If the completion procedure terminates with success, then the resulting rewriting system is canonical.

2.2. Finitely Presented Groups

Suppose that set G of letters is the disjoint union of sets G^+ and G^-, where $G^+ = \{g_1, \ldots, g_n\}$, such that $g_i \neq \varepsilon\ \forall 1 \leq i \leq n$, and $G^- = \{g_1^{-1}, \ldots g_n^{-1}\}$. Consider function $^{-1} :\ G^* \to G^*$, called *inverse function*, defined as follows:

– For any g_i in G^+, $(g_i)^{-1} = g_i^{-1}, 1 \leq i \leq n$.

– For any g_i^{-1} in G^-, $(g_i^{-1})^{-1} = g_i, 1 \leq i \leq n$.

– For the empty string, $\varepsilon^{-1} = \varepsilon$.

– For all strings s, t in G^*, $(s \cdot t)^{-1} = (t)^{-1} \cdot (s)^{-1}$.

Group axioms Consider canonical rewriting system $\mathcal{A}$ of *group axioms*.

$$\mathcal{A} = \begin{cases} (x \cdot y) \cdot z \to x \cdot (y \cdot z) \\ x \cdot \varepsilon \to x,\ \varepsilon \cdot x \to x \\ x \cdot x^{-1} \to \varepsilon,\ x^{-1} \cdot x \to \varepsilon \\ (x \cdot y)^{-1} \to y^{-1} \cdot x^{-1} \\ x \cdot (x^{-1} \cdot y) \to y \\ x^{-1} \cdot (x \cdot y) \to y \\ (x^{-1})^{-1} \to x \\ \varepsilon^{-1} \to \varepsilon \end{cases}$$

Rewriting system $\mathcal{A}$ is a *term rewriting system*, where binary function $\cdot$ and unary function $^{-1}$ are arbitrary, and x, y and z are variables. Let us take as $\cdot$ binary concatenation operation on strings, and as $^{-1}$ inverse function defined above. Let $\mathcal{R}$ be a string rewriting system. Quotient $\mathcal{G}$ of monoid G^* by congruence relation $\overset{*}{\leftrightarrow}_{\{\mathcal{R},\mathcal{A}\}}$ induced by rewriting system $\{\mathcal{R}, \mathcal{A}\}$, is called a *group*. Elements of set G^+ are said to be *generators* of group $\mathcal{G}$, and the elements of set G^- are said to be *inverses of generators*. Pair $(G^+, \mathcal{R})$ is called a *presentation* of group $\mathcal{G}$. Group $\mathcal{G}$ is *finitely presented* if both set G^+ and rewriting system $\mathcal{R}$ are finite.

Since concatenation operation $\cdot$ is associative, associativity axiom $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ is built in string structure. Therefore we omit parenthesis in group axioms. A string s is said to be in $\mathcal{A}$-*canonical form* if s is $\mathcal{A}$-irreducible. If string s is in $\mathcal{A}$-canonical form and $s \neq \varepsilon$, then s contains no consecutive inverse generators and empty substring ε.

Group Reduction Let $(G^+, \mathcal{R})$ be a presentation of group $\mathcal{G}$. Rewriting relation $\rightarrow_{\mathcal{G}}$ such that $s \rightarrow_{\mathcal{G}} t$ whenever there is a rule $R: \ l \rightarrow r$ in $\mathcal{R}$ and there are strings u, v such that $s = u \cdot l \cdot v$, $u \cdot r \cdot v \xrightarrow{*}_{\mathcal{A}} t$ and t is in $\mathcal{A}$-canonical form, is called the *group reduction.*

The completion procedure implementing group reduction (introduced in [Le Chenadec1986]), runs faster than standard completion procedure.

Trivial Rules and Equations An equation $E: \ s = t$ is called *trivial* if E is an instance of a group axiom, or of reflexivity axiom $x = x$. In other words, a trivial equation E is a group axiom, or reflexivity axiom, where each occurrence of variable x is replaced by a string. A rule $R: \ l \rightarrow r$ is called *trivial* if R is an oriented trivial equation.

Symmetrized Forms According to [Le Chenadec1986], since string rewriting for groups avoids using associativity axiom, computing critical pairs between ground rules (i.e. rules without variables) and group axioms becomes easy.

THEOREM 2.1. *The completion procedure associates to each rule*

$$R: \ a_1 \cdot a_2 \cdot \ldots \cdot a_l \rightarrow b_1 \cdot b_2 \cdot \ldots \cdot b_m,$$

$a_i, b_j \in G$ *the following critical pairs with group axioms, called normal pairs:*

$$\begin{gathered}(a_1 \cdot \ldots \cdot a_{l-1}, b_1 \cdot \ldots \cdot b_m \cdot a_l^{-1}),\\ (a_2 \cdot \ldots \cdot a_l, a_1^{-1} \cdot b_1 \cdot \ldots \cdot b_m),\\ (a_l^{-1} \cdot \ldots \cdot a_1^{-1}, b_m^{-1} \cdot \ldots \cdot b_1^{-1}).\end{gathered}$$

Precompiling normal pairs of each rewriting rule R and adding them directly to the rewriting system increases the efficiency of the completion procedure.

Let $>$ be a reduction ordering. Computing normal pairs of rewriting rule R, normal pairs of oriented normal pairs of R, normal pairs of oriented normal pairs of normal pairs of R etc., we get the set of symmetrized forms of rewriting rule R.

DEFINITION 2.1. Let $R: \ (a_1 \cdot \ldots \cdot a_l, b_1 \cdot \ldots \cdot b_m)$ be a rewriting rule or an equation. Let $Perm(R)$ be the largest set of rewriting rules $l \rightarrow r$ such that $l > r$ and $l \cdot r^{-1}$ is a cyclic permutation of string $a_1 \cdot \ldots \cdot a_l \cdot b_m^{-1} \cdot \ldots \cdot b_1^{-1}$ or of string $b_1 \cdot \ldots \cdot b_m \cdot a_l^{-1} \cdot \ldots \cdot a_1^{-1}$. Let letter g be in set $\{a_i, a_i^{-1}\}_{i=1}^{l} \cup \{b_j, b_j^{-1}\}_{j=1}^{m}$. Let $Perm(g)$ be the subset of set $Perm(R)$, such that for any rule $l \rightarrow r$ in $Perm(g)$ the first letter of string l is g. Denote by $R(g)$ the rule in $Perm(g)$ with shortest left-hand side.

The union for all g in $\{a_i, a_i^{-1}\}_{i=1}^{l} \cup \{b_j, b_j^{-1}\}_{j=1}^{m}$, of rewriting rules $R(g)$ is called the *set of symmetrized forms of* R and is denoted by $Sym(R)$.

We remark that if rewriting rule $R' \in Sym(R)$ then $Sym(R') = Sym(R)$.

LEMMA 2.1. *Let* $R: \ (a_1 \cdot \ldots \cdot a_l, b_1 \cdot \ldots \cdot b_m)$ *be a rewriting rule or an equation. Then any normal pair of* R *is joinable with respect to rewriting system* $\{Sym(R), \mathcal{A}\}$.

Proof. Let us prove the lemma for normal pair $(a_1 \cdot \ldots \cdot a_{l-1}, b_1 \cdot \ldots \cdot b_m \cdot a_l^{-1})$. Proofs for two other normal pairs are similar. If $a_1 \cdot \ldots \cdot a_{l-1} > b_1 \cdot \ldots \cdot b_m \cdot a_l^{-1}$ then rewriting rule $R_1 : \ a_1 \cdot \ldots \cdot a_{l-1} \to b_1 \cdot \ldots \cdot b_m \cdot a_l^{-1}$ from set $Perm(a_1)$ either coincides with rule $R(a_1)$, or has a longer lhs than rule $R(a_1)$. In both cases, normal pair $(a_1 \cdot \ldots \cdot a_{l-1}, b_1 \cdot \ldots \cdot b_m \cdot a_l^{-1})$ is joinable with respect to rewriting system $\{R(a_1), \mathcal{A}\}$. This result is straightforward in first case. In second case, $R(a_1) : \ a_1 \cdot \ldots \cdot a_i \to b_1 \cdot \ldots \cdot b_m \cdot a_l^{-1} \cdot \ldots \cdot a_{i+1}^{-1}$ for some $1 \le i \le l-2$. Thereby,

$$a_1 \cdot \ldots \cdot a_{l-1} \to_{R(a_1)} (b_1 \cdot \ldots \cdot b_m \cdot a_l^{-1} \cdot \ldots \cdot a_{i+1}^{-1}) \cdot a_{i+1} \cdot \ldots a_{l-1} \xrightarrow{*}_{\mathcal{A}} b_1 \cdot \ldots \cdot b_m \cdot a_l^{-1}.$$

If $a_1 \cdot \ldots \cdot a_{l-1} < b_1 \cdot \ldots \cdot b_m \cdot a_l^{-1}$ then rewriting rule $R_1 : \ b_1 \cdot \ldots \cdot b_m \cdot a_l^{-1} \to a_1 \cdot \ldots \cdot a_{l-1}$ from set $Perm(b_1)$ either coincides with rule $R(b_1)$, or has a longer lhs than rule $R(b_1)$. Normal pair $(a_1 \cdot \ldots \cdot a_{l-1}, b_1 \cdot \ldots \cdot b_m \cdot a_l^{-1})$ is joinable with respect to rewriting system $\{R(b_1), \mathcal{A}\}$. The result is obvious in first case. In second case, $R(b_1) : \ b_1 \cdot \ldots \cdot b_j \to a_1 \cdot \ldots \cdot a_l \cdot b_m^{-1} \cdot \ldots \cdot b_{j+1}^{-1}$ for some $1 \le j \le m-1$. Then,

$$b_1 \cdot \ldots \cdot b_m \cdot a_l^{-1} \to_{R(b_1)} (a_1 \cdot \ldots \cdot a_l \cdot b_m^{-1} \cdot \ldots \cdot b_{j+1}^{-1}) \cdot b_{j+1} \cdot \ldots b_m \cdot a_l^{-1} \xrightarrow{*}_{\mathcal{A}} a_1 \cdot \ldots \cdot a_{l-1}.$$

□

2.3. Graphs

A *graph* is defined by a set of *vertices* V and a set of *edges* E.

Edges An edge e is associated with a pair of vertices (v, v'). Different edges can be associated with the same pair of vertices. We write $e = (v, v')$, when it is not important, which edge we take among all the edges associated with the pair (v, v'). Vertices v and v' are said to be *ends* of edge $e = (v, v')$, and edge e is said to be *adjacent to vertices* v and v'.

An edge e is said to be *directed* if pair (v, v') is ordered; in this case v is called the *initial vertex*, or the *beginning vertex* of edge e, and v' the *final vertex*, or the *end vertex* of edge e.

If $e = (v, v')$ is a directed edge, then we say that edge e *goes from* vertex v *to* vertex v' for *standard direction*. The same edge e is said to be taken in *opposite direction* if e is considered as going from v' to v. Edge e in standard direction is denoted by e^+, and edge e in opposite direction is denoted by e^-. By default, notation e means e^+.

Paths A *path* π (denoted $\pi = e_1^{\alpha_1}.e_2^{\alpha_2}. \ \ldots \ .e_k^{\alpha_k}$) is the concatenation of a finite sequence of edges $e_1^{\alpha_1}, e_2^{\alpha_2}, \ldots, e_k^{\alpha_k}$, such that

(1) $e_1^{\alpha_1} = (v_1, v_2), e_2^{\alpha_2} = (v_2, v_3), \ldots, e_k^{\alpha_k} = (v_k, v_{k+1})$, and

(2) $\alpha_i \in \{+, -\}$.

The *length* of a path $\pi = e_1^{\alpha_1}.e_2^{\alpha_2}. \ \ldots \ .e_k^{\alpha_k}$ is k. The *empty path*, denoted by ϵ, is the empty sequence of edges. The length of the empty path ϵ is 0.

Let $\pi = e_1^{\alpha_1}.e_2^{\alpha_2}. \ \ldots \ .e_k^{\alpha_k}$ be a path satisfying properties (1) and (2). We say vertices $v_1, \ldots, v_{k+1}$ to *belong to path* π and denote it by $v_1, \ldots, v_{k+1} \in \pi$. We say that vertex v_i *precedes* vertex v_j in path π if $i < j$. We also say that edge $e_i^{\alpha_i}$ *precedes* edge $e_j^{\alpha_j}$ in path π if $i < j$.

Vertices v_1 and v_k are said to be *ends* of path π. Vertex v_1 is called the *initial vertex*, or *beginning vertex* of path π, and vertex v_{k+1} is called the *final vertex*, or *end vertex* of path π. We also say that π *goes from v_1 to v_{k+1}* for *standard direction*. Considered as going from v_{k+1} to v_1, path π is taken in *opposite direction*. Path π in standard direction is denoted by π^+, and by π^- in opposite direction. When superscripts are omitted, we suppose that π stands for π^+. Note that if edge e precedes edge e' in path π^+, then edge e' precedes edge e in path π^- and vice versa. Also, if vertex v precedes vertex v' in path π^+, then vertex v' precedes vertex v in path π^- and vice versa. To obtain path π^- it is sufficient to reverse the order of edges in path π^+ and their direction, and vice versa.

A path π_1 is a *subpath* of a path $\pi = e_1^{\alpha_1}. \ \ldots \ .e_k^{\alpha_k}$ if path π_1 has the form $\pi_1 = e_i^{\alpha_i}.e_{i+1}^{\alpha_{i+1}}. \ \ldots \ .e_j^{\alpha_j}$ for some $1 \leq i \leq j \leq k$. We denote it by $\pi_1 \subseteq \pi$. The empty path ϵ is a subpath of every path π.

If path $\pi_1 = e_i^{\alpha_i}.e_{i+1}^{\alpha_{i+1}}. \ \ldots \ .e_j^{\alpha_j}$ is a subpath of a path $\pi = e_1^{\alpha_1}. \ \ldots \ .e_k^{\alpha_k}$, then path π is the *concatenation* of paths π_2, π_1 and π_3 (denoted by $\pi = \pi_2.\pi_1.\pi_3$), where

- path π_2 has the form $\pi_2 = e_1^{\alpha_1}. \ \ldots \ .e_{i-1}^{\alpha_{i-1}}$ and
- path π_3 has the form $\pi_3 = e_{j+1}^{\alpha_{j+1}}. \ \ldots \ .e_k^{\alpha_k}$.

Let path π_1 be a subpath of a path π. Path π_1 is a *prefix of path* π if $\pi = \pi_1.\pi_2$ for some path π_2. Path π_1 is a *suffix of path* π if $\pi = \pi_3.\pi_1$ for some path π_3.

An edge e is said to be *adjacent to a path* π if e is adjacent to an end of path π.

Cycles Let $\pi = e_1^{\alpha_1}. \ \ldots \ .e_k^{\alpha_k}$ be a path such that $v_{k+1} = v_1$ and π is not empty. Set $[\pi] = \{\pi_1, \ldots, \pi_{2k}\}$ of paths, such that for any path π_i in $[\pi]$, the sequence of edges of π_i is a cyclic permutation of the sequence of edges of path π^+ or of path π^-, is called a *cycle*. The length of path π is also the *length of cycle* $[\pi]$.

A path π' is a *subpath of a cycle* $[\pi]$, if $\pi' \subseteq \pi_i$ for some $i \in \{1, \ldots, 2k\}$. It is denoted by $\pi' \subseteq [\pi]$. The length of a subpath π of a cycle $[\pi]$ is less or equal than the length of cycle $[\pi]$. A vertex v *belongs to a cycle* $[\pi]$, if $v \in \pi_i$ for some $i \in \{1, \ldots, 2k\}$. We denote it by $v \in [\pi]$. Actually, $v \in [\pi]$ implies $v \in \pi_i$ for all $i \in \{1, \ldots, 2k\}$.
Thereafter we denote a cycle $\mathrm{C} = [\pi]$ by any path $e_1^{\alpha_1}. \ \ldots \ .e_k^{\alpha_k}$ in set $[\pi]$, writing C: $e_1^{\alpha_1}. \ \ldots \ .e_k^{\alpha_k}$. An ordered pair of paths (π_1, π_2) is called a *cut* of a cycle C, if there is a path π in C such that $\pi = \pi_1^+.\pi_2^-$. In particular, if (π_1, π_2) is a cut of cycle C, then initial vertices of paths π_1 and π_2 coincide, and final vertices paths π_1 and π_2 coincide too. If (π_1, π_2) is a cut of C, then we write $\pi_1 = \mathrm{C} \backslash \pi_2$.

3. Patch Graphs

In Section 3.1 we explain intuitively, how a finite string rewriting system for a group can be represented by a PATCH Graph. In Section 3.2 we give an idea how a rewriting system can be completed in graph representation. One of the central notions of the paper, the notion of *common paths*, is introduced. Section 3.3 gives a formal definition of PATCH Graph syntax, and Sect.3.4 defines the PATCH Graph semantics.

3.1. Informal Description of Graph Representation of a String Rewriting System for a Finitely Presented Group

Let $\mathcal{R}$ be a rewriting system on a set G. The idea is to consider strings on G as paths, and rules as pairs of different paths with common beginning and common end. To a letter g or $g^{-1} \in G$ we associate a directed edge e labelled by g, called a *concatenation edge*. A string $s \in G^*$ is associated with the path, composed of concatenation edges, corresponding to generators and inverses of generators in s. These edges are oriented depending on their "sign". For example, to string $g_1 \cdot g_2^{-1}$ we make correspond path $\pi = e_1^+.e_2^-$, where $e_1 = (v_1, v_2), e_2 = (v_3, v_2)$, e_1 is labelled by g_1, and e_2 is labelled by g_2. A rewriting rule $l \to r$ corresponds to a cycle C and a cut (π_1, π_2) of C, such that π_1 corresponds to l and π_2 corresponds to r. We mark by **B** initial vertex of π_1 and π_2, and by **E** final vertex of them. **B** stands for *beginning* and **E** stands for *end.* A rewriting system $\mathcal{R}$ is represented as the union of cycles corresponding to all its rules.

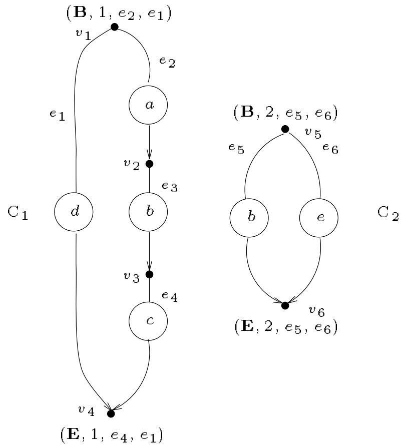

Figure 1. Graph representation of $\mathcal{R} = \{R_1 : \ a \cdot b \cdot c \to d, \ R_2 : \ b \to e\}$

For example, system $\mathcal{R} = \{R_1 : \ a \cdot b \cdot c \to d, \ R_2 : \ b \to e\}$, can be represented by graph in Figure 1. There are two cycles: C_1 and C_2 representing R_1 and R_2 respectively. Let us detail the construction of C_1. Let e_1, e_2, e_3 and e_4 be edges labelled d and a, b, c respectively, such that they form a cycle $e_1^+.(e_2.e_3.e_4)^-$. Let $e_1 = (v_1, v_4), e_2 = (v_1, v_2), e_3 = (v_2, v_3)$ and $e_4 = (v_3, v_4)$. Then v_1 is initial vertex of paths e_1 and $e_2.e_3.e_4$, and v_2 is final vertex of them. We put label $(\mathbf{B}, 1, e_2, e_1)$ on v_1 indicating the beginning of paths and the orientation of rule R_1. Actually, edge e_2 is the first edge of path $e_2.e_3.e_4$ representing the lhs $a \cdot b \cdot c$ and e_1 is the first edge of path e_1 representing the rhs d. Number 1 in the label stands for rule number one.

3.2. Informal Description of Completion of a String Rewriting System in Graph Representation

The reason for storing rewriting rules as cycles is that one cycle can represent the set of all symmetrized forms of a rule, sharing concatenation edges. Let $>_{ll}$ be the length-lexicographical ordering with precedence

$$a > a^{-1} > b > b^{-1} > c > c^{-1} > d > d^{-1} > e > e^{-1}.$$

The set of all symmetrized forms of R_1 is the following:

$$\left\{ \begin{array}{l} R_{(1,1)}:\ a \cdot b \rightarrow d \cdot c^{-1} \\ R_{(1,2)}:\ a^{-1} \cdot d \rightarrow b \cdot c \\ R_{(1,3)}:\ b^{-1} \cdot a^{-1} \rightarrow c \cdot d^{-1} \\ R_{(1,4)}:\ c^{-1} \cdot b^{-1} \rightarrow d^{-1} \cdot a \end{array} \right\}.$$

Putting on C_1 vertices the following labels:

- $\{(\mathbf{B}, 1, 1, e_2, e_1), (\mathbf{E}, 1, 3, e_2, e_1)\}$ on v_1,
- $\{(\mathbf{B}, 1, 2, e_2, e_3), (\mathbf{E}, 1, 4, e_3, e_2)\}$ on v_2,
- $\{(\mathbf{B}, 1, 3, e_3, e_4), (\mathbf{E}, 1, 1, e_3, e_4)\}$ on v_3, and
- $\{(\mathbf{B}, 1, 4, e_4, e_1), (\mathbf{E}, 1, 2, e_1, e_4)\}$ on v_4

(see Fig.2), we can represent by cycle C_1 all four rules $R_{(1,i)}$, sharing concatenation edges. Second component of each 5-uple is number 1 of cycle C_1, and third component is number $i \in \{1, 2, 3, 4\}$ of rule $R_{(1,i)}$. Each pair $(\mathbf{B}, 1, i, *, *)$ and $(\mathbf{E}, 1, i, *, *)$ of 5-uples[2], shows the beginning and the end of rule $R_{(1,i)}$ in cyclic representation.

In the same way, since the set of all symmetrized forms of R_2 is

$$\{R_{(2,1)}:\ b \rightarrow e, R_{(2,2)}:\ b^{-1} \rightarrow e^{-1}\},$$

putting label $\{(\mathbf{B}, 2, 1, e_5, e_6), (\mathbf{E}, 2, 2, e_5, e_6)\}$ on vertex v_5, and label $\{(\mathbf{B}, 2, 2, e_5, e_6), (\mathbf{E}, 2, 1, e_5, e_6)\}$ on vertex v_6 we can represent both rules $R_{(2,1)}$ and $R_{(2,2)}$ by cycle C_2.

Precompiling the set of all symmetrized forms of a rewriting rule and putting corresponding labels on vertices of the cycle representing this rule, we perform the *symmetrization procedure*, which is a part of the completion procedure. Let us explain how to perform completion procedure on a rewriting system transforming the graph representing it.

To discover rules giving a critical pair or a simplification, we find in the graph cycles having common paths.

DEFINITION 3.1. Let C_1 and C_2 be cycles in Γ. Every pair π_1, π_2 of paths, such that $\pi_1 \subseteq \mathrm{C}_1$, $\pi_2 \subseteq \mathrm{C}_2$, and π_1 and π_2 represent the same string s, are called *common paths* for C_1 and C_2.

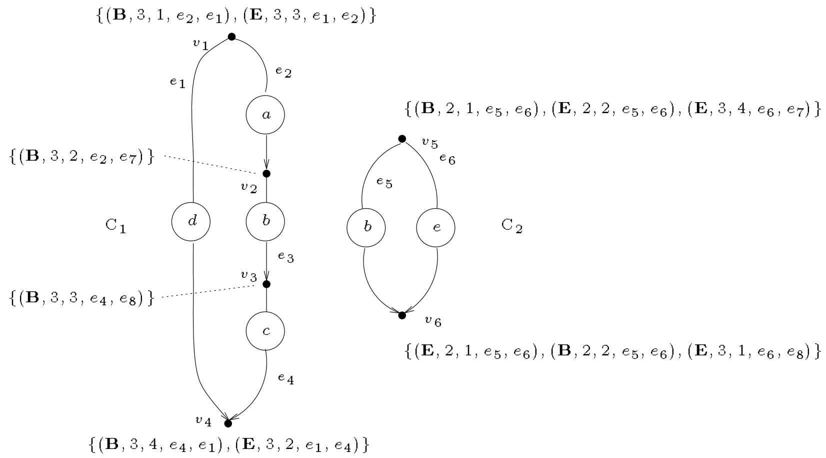

Figure 2. Graph representation of symmetrized forms of rewriting rules

If cycles C_1 and C_2 have a common path, then it is possible that there is a rule R_1 in the set of rewriting rules represented by C_1 and there is a rule R_2 in the set of rewriting rules represented by C_2, such that R_1 simplifies R_2, or R_1 and R_2 give a critical pair. In Section 4, in the premise of **Patch** inference rule, we formulate the condition when this happens.

In the example above, edges e_3 and e_5, both representing b, are common paths for C_1 and C_2. Adding new edges e_7 and e_8, labelled ε, going from v_2 to v_5 and from v_6 to v_3 respectively, we obtain a new cycle $\mathrm{C}_3 : e_2^+.e_7^+.e_6^+.e_8^+.e_4^+.e_1^-$. Reading edge labels through the cycle, taking for example cut $(e_2^+.e_7^+.e_6^+.e_8^+.e_4^+, e_1^+)$, we see that C_3 represents equation $a \cdot e \cdot c = d$. The set of all symmetrized forms of this equation with respect to $>_{ll}$ is the following [3]:

$$\begin{cases} R_{(3,1)} : \; a \cdot e \rightarrow d \cdot c^{-1} \\ R_{(3,2)} : \; a^{-1} \cdot d \rightarrow e \cdot c \\ R_{(3,3)} : \; c \cdot d^{-1} \rightarrow e^{-1} \cdot a^{-1} \\ R_{(3,4)} : \; c^{-1} \cdot e^{-1} \rightarrow d^{-1} \cdot a \end{cases}$$

We add the following vertex labels:

- $\{(\mathbf{B}, 3, 1, e_2, e_1), (\mathbf{E}, 3, 3, e_1, e_2)\}$ to v_1,
- $\{(\mathbf{B}, 3, 2, e_2, e_7)\}$ to v_2,
- $\{(\mathbf{E}, 3, 4, e_6, e_7)\}$ to v_5,
- $\{(\mathbf{E}, 3, 1, e_6, e_8)\}$ to v_6,
- $\{(\mathbf{B}, 3, 3, e_4, e_8)\}$ to v_3, and
- $\{(\mathbf{B}, 3, 4, e_4, e_1), (\mathbf{E}, 3, 2, e_1, e_4)\}$ to v_4.

Since $b > e$, and e_5 represents the lhs of both rules $R_{(2,1)}$ and $R_{(2,2)}$, the rules represented by C_1 such that e_3 is a subpath of the path representing their lhs or rhs, are simplified either by $R_{(2,1)}$, or by $R_{(2,2)}$. Consequently, we can delete them from the system, removing corresponding vertex labels along C_1. We can not remove edges of C_1 because some of them are shared by C_1 and C_3. Resulting graph is presented by Fig.3. It corresponds to the following canonical rewriting system

$$\left\{\begin{array}{c} R_{(2,1)}:\ b \rightarrow e, R_{(2,2)}:\ b^{-1} \rightarrow e^{-1}, \\ R_{(3,1)}:\ a \cdot e \rightarrow d \cdot c^{-1}, R_{(3,2)}:\ a^{-1} \cdot d \rightarrow e \cdot c, \\ R_{(3,3)}:\ c \cdot d^{-1} \rightarrow e^{-1} \cdot a^{-1}, R_{(3,4)}:\ c^{-1} \cdot e^{-1} \rightarrow d^{-1} \cdot a \end{array}\right\}.$$

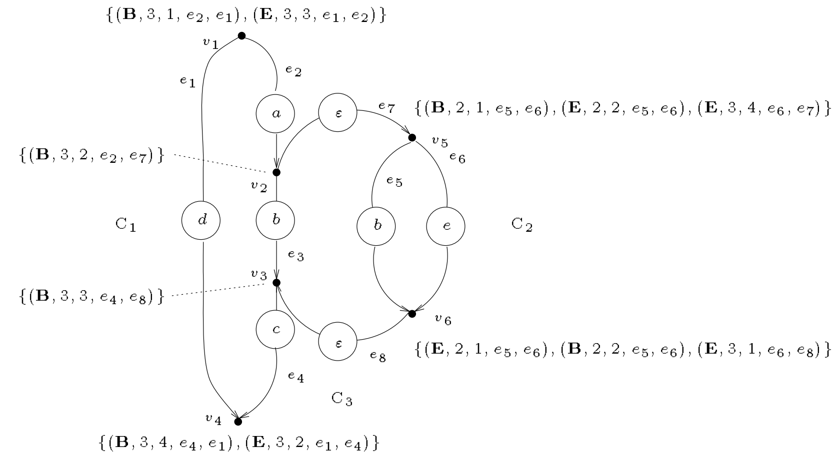

Figure 3. Graph representation of canonical rewriting system

As the reader can see, the computation of common paths between cycles is an important part of completion procedure we propose. Afterwards we will indicate common paths by *matching edges*. For each pair of common paths, we will put a pair of matching edges, one between initial, and the other between final vertices of common paths. Positions and labels of matching edges will be computed by *matching procedure* we introduce in Section 6.

3.3. Syntax

Before giving the formal definition of a PATCH Graph, let us introduce the notion of conjugate common paths.

DEFINITION 3.2. Let π_1 and π_2 be paths. They are called *conjugate common paths* if there exist paths π_3, π_5 (possibly empty) and π_4 such that $\pi_1^+ = \pi_3^+.\pi_4^+.\pi_5^-$, π_2^+ and π_4^+ are common paths, and paths π_3^+ and π_5^+ represent the same string.

We need the notion of *conjugate* common paths, because the notion of common paths is not sufficient to define a PATCH Graph. We introduce in Sect.4 inference rules **Patch**, **Symmetrize** and **Eliminate** for completion of rewriting systems in PATCH Graph representation. The reader will see that **Patch** and **Symmetrize** transform a PATCH Graph into another PATCH Graph, even if common paths are used instead of conjugate common paths, but **Eliminate** does not.

DEFINITION 3.3 (PATCH Graph). Let $\Gamma = (V, E, \lambda, \eta)$ be a finite labelled graph, where V is a set of vertices, E is a set of edges, λ is the labelling function on edges, and η is the labelling function on vertices. Γ is called a *PATCH Graph relative to a set of generators* G^+ iff

- set E is a disjoint union of set E_C of *concatenation edges* (*C-edges*) and set E_M of *matching edges* (*M-edges*). Concatenation edges are directed, matching edges are undirected.
  Label $\lambda(e)$ of every M-edge e is a set of triples (n, n_1, n_2) for some $n, n_1, n_2 \in \mathbf{N}$. An unordered pair (e_1, e_2) of M-edges such that $e_1 \neq e_2$ and set $\lambda(e_1)$ and set $\lambda(e_2)$ both contain the same triple (n, i, j), is called a *matching bridge* (*M-bridge*) *identified* (n, i, j). Triple (n, i, j) is said to be the *identification* of M-bridge (e_1, e_2). There is at most one M-bridge with the same n as first component of its identification.

- $\Gamma = \bigcup_{n=1}^{m} \mathrm{C}_n \cup E_M$, where for each $n \in \{1, \ldots, m\}$ C_n is a cycle number n in Γ.
  C_n is a *cycle number n* in Γ iff it is a minimal cycle, such that either

  - for any edge e of C_n, e is a C-edge and $\lambda(e) = (g_i, n)$ for some $g_i \in G^+$ and $n \in \mathbf{N}$. Such a cycle is called an *initial cycle*; or
  - there exist C-edges e_1 and e_2 in C_n,
    there exist cycles C_{n_1} and C_{n_2}, and
    there are cuts (π_1, π_3) of C_{n_1} and (π_2, π_4) of C_{n_2}, such that:
    * paths π_1 and π_2 are conjugate common paths, and
    * $\lambda(e_1) = (\varepsilon, n, n_1, n_2)$ and $\lambda(e_2) = (\varepsilon, n, n_2, n_1)$,
      $e_1 = (v_1, v_2)$ and $e_2 = (v_3, v_4)$, where
      v_1 and v_4 are respectively initial and final vertices of π_1 and π_3, and
      v_2 and v_3 are respectively initial and final vertices of π_2 and π_4 (see Figure 4).

    In this case cycle number n is $\mathrm{C}_n : e_1^+.\pi_4^+.e_2^+.\pi_3^-$. Pair (e_1, e_2) is called a *concatenation bridge* (*C-bridge*) between cycles C_{n_1} and C_{n_2}, relative to conjugate common paths π_1 and π_2. Cycles C_{n_1} and C_{n_2} are said to be *ancestors* of cycle C_n. We also say that C-bridge (e_1, e_2) *defines* cycle C_n.

- For any vertex v label $\eta(v)$ is a (possibly empty) set of *beginning* and *end markers*, such that if $\eta(v)$ contains a marker $(\mathbf{B}, n, m, e_3, e_4)$ or $(\mathbf{E}, n, m, e_3, e_4)$ for some $m \in \mathbf{N}$, then $v \in \mathrm{C}_n$ and edges $e_3, e_4 \in \mathrm{C}_n$ are C-edges adjacent to v. For all n, there is exactly one $\mathbf{B}$-marker and one $\mathbf{E}$-marker with the same m.

The role of C-edges is to form cycles, their labels giving information on the number of first (with respect to order of creation) cycle where they appear.

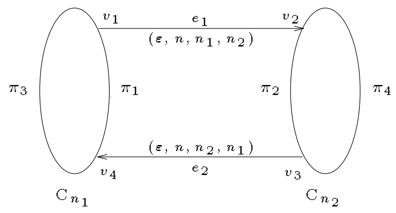

Figure 4. Cycle in a PATCH Graph

The role of M-edges is to make reference to places where a new cycle can be created. Their labels give numbers n_1 and n_2 of cycles-ancestors and the number n of a potential cycle.

In this notation, edge labels of the graph given by Figure 3 are: $\lambda(e_1) = (d, 1)$, $\lambda(e_2) = (a, 1)$, $\lambda(e_3) = (b, 1)$, $\lambda(e_4) = (c, 1)$, $\lambda(e_5) = (b, 2)$, $\lambda(e_6) = (e, 2)$, $\lambda(e_7) = (\varepsilon, 3, 1, 2)$ and $\lambda(e_8) = (\varepsilon, 3, 2, 1)$.

Constructing an initial PATCH graph Γ, representing initial rewriting system $\mathcal{R}$ we have to complete, we first cyclically reduce rewriting rules of $\mathcal{R}$ and then create for each reduced rule an initial cycle. Each new cycle receives a fresh number n^4, as in our example in Figure 1.

3.4. Semantics

Now we give the semantics of PATCH Graphs. Function Δ defines inductively what a PATCH Graph, relative to a set of generators G^+, represents.

DEFINITION 3.4. Let $\Gamma = (\mathrm{V}, \mathrm{E}, \lambda, \eta)$ be a PATCH Graph relative to a set G^+. Function Δ is the *semantic function* of Γ such that

1. For any edge $e \in E_C$ such that $\lambda(e) = (g, n)$ for some $g \in G^+$ and $n \in \mathbf{N}$, $\Delta(e^+) = g$, and $\Delta(e^-) = g^{-1}$.

2. For any edge $e \in E_C$ such that $\lambda(e) = (\varepsilon, n, n_1, n_2)$ for some $n, n_1, n_2 \in \mathbf{N}$, $\Delta(e^+) = \Delta(e^-) = \varepsilon$.

3. For the empty path ϵ $\Delta(\epsilon) = \varepsilon$.

4. For each non-empty path[5] π and a C-edge e

   - $\Delta(\pi.e) = \Delta(\pi)$ if $\Delta(e) = \varepsilon$ and $\Delta(\pi.e) = \Delta(\pi) \cdot \Delta(e)$ else,
   - $\Delta(e.\pi) = \Delta(\pi)$ if $\Delta(e) = \varepsilon$ and $\Delta(e.\pi) = \Delta(e) \cdot \Delta(\pi)$ else.

5. For each cycle $\mathrm{C}_n : \pi$ in Γ, such that for any vertex $v \in \mathrm{C}_n$, neither of markers $(\mathbf{B}, n, *, *, *)$ and $(\mathbf{E}, n, *, *, *)$ belongs to label $\eta(v)$, $\Delta(\mathrm{C}_n)$ is the equation $\Delta(\pi) = \varepsilon$.

6. For each cycle C_n in Γ, such that there is a vertex $v \in \mathrm{C}_n$ such that label $\eta(v)$ contain a marker $(\mathbf{B}, n, *, *, *)$ or $(\mathbf{E}, n, *, *, *)$, let (π_1, π_2) be a cut of C_n such that for some i

- label $\eta(v_1)$ of initial vertex v_1 of paths π_1 and π_2 contains beginning marker $(\mathbf{B}, n, i, e_1, e_2)$, where e_1 (e_2) is the first edge of π_1 (resp. π_2), and
- label $\eta(v_2)$ of final vertex v_2 of paths π_1 and π_2 contains end marker $(\mathbf{E}, n, i, e_3, e_4)$, where e_3 (e_4) is the last edge of π_1 (resp. π_2).

Then $\Delta(\mathrm{C}_n, (\pi_1, \pi_2))$ is the rule $R_{(n,i)} : \ \Delta(\pi_1) \to \Delta(\pi_2)$.

7. For each cycle $\mathrm{C}_n \in \Gamma$, and set $\{(\pi_1, \pi_1'), \ldots, (\pi_k, \pi_k')\}$ of all cuts of C_n, such that for any $i \in \{1, \ldots, k\}$ cut (π_i, π_i') satisfies the conditions of item 6, $\Delta(\mathrm{C}_n, \{(\pi_1, \pi_1'), \ldots, (\pi_k, \pi_k')\}) = \{R_{(n,i)}\}_{i=1}^{k}$.
8. Let cycles $\mathrm{C}_1, \ldots, \mathrm{C}_m$ are all cycles in graph Γ and for each $n \in \{1, \ldots, m\}$ set $\{(\pi_{k_1}, \pi_{k_1}'), \ldots, (\pi_{k_n}, \pi_{k_n}')\}$ of cuts of cycle C_n satisfies the conditions of item 7. Then $\Delta(\Gamma) = \{\Delta(\mathrm{C}_n, \{(\pi_{k_1}, \pi_{k_1}'), \ldots, (\pi_{k_n}, \pi_{k_n}')\})\}_{n=1}^{m}$.

4. Inference Rules

In this section we give inference rules of a new completion procedure for string rewriting systems for groups represented by PATCH Graphs, and illustrate each of them by examples. We have three inference rules: **Symmetrize**, **Patch** and **Eliminate**. All of them are formulated as transformations of cycles in a PATCH Graph. **Symmetrize** expresses the symmetrization procedure, relative to a total reduction ordering $>$ on G^*. **Symmetrize** is applied to each new cycle, created by a **Patch** application. **Patch** is the central rule, giving the law of creation of new cycles. **Patch** inference rule generalizes the critical pair and the simplification computations. **Eliminate** defines the transformation of cycles encoding cancellation and simplification with respect to group axioms $x \cdot x^{-1} \to \varepsilon$ and $x^{-1} \cdot x \to \varepsilon$. **Eliminate** is applied to some of the new cycles, created by a **Patch** inference rule application.

4.1. **Symmetrize** Inference Rule

This inference rule encodes the symmetrization procedure for rewriting rules represented by a cycle of a PATCH Graph.

Symmetrize

For each new cycle C_n, we put along C_n beginning and end markers $(\mathbf{B}, n, j, *, *)$, $(\mathbf{E}, n, k, *, *)$, corresponding to the set of all symmetrized forms of $\Delta(\mathrm{C}_n)$.

Proposition 4.1 explains how to compute symmetrized forms of a rewriting rule or an equation represented by a cycle in a PATCH Graph.

PROPOSITION 4.1. *Let C be a cycle in a PATCH Graph. Let $\{(\pi_1, \pi_1'), \ldots, (\pi_k, \pi_k')\}$ be the largest set of cuts of C such that for all $i, j \in \{1, \ldots, k\}$ such that $i \neq j$, $\pi_i \not\subset \pi_j$ and $\Delta(\pi_i) > \Delta(\pi_i')$. Then*

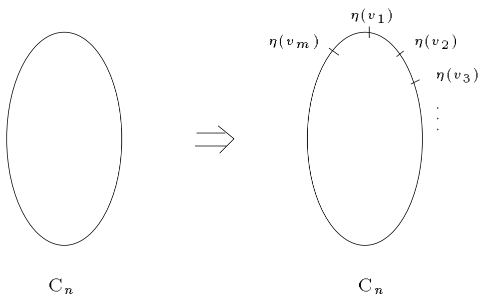

Figure 5. Symmetrize inference rule

- *for any edge $e \in C$, there is an $i \in \{1, \ldots, k\}$ such that e is the first edge of path π_i, and*
- *set $\Delta(C, \{(\pi_1, \pi'_1), \ldots, (\pi_k, \pi'_k)\})$ of rewriting rules coincides with set $Sym(\Delta(C))$ of all symmetrized forms of $\Delta(C)$.*

Proof. Let cycle C: $e_1^{\alpha_1}.e_2^{\alpha_2}. \ \ldots \ .e_l^{\alpha_l}$. Let $\Delta(e_i) = s_i, 1 \leq i \leq l$. Equation $\Delta(\mathrm{C})$ is $\Delta(\mathrm{C}): \ s_1^{\alpha_1} \cdot \ldots \cdot s_l^{\alpha_l} = \varepsilon$.

For any $i \in \{1, \ldots, l\}$, denote by $Cut(e_i^+)$ the set of all cuts (σ, σ') of cycle C such that e_i^+ is the first edge of path σ, and $\Delta(\sigma) > \Delta(\sigma')$. Also, denote by $Cut(e_i^-)$ the set of all cuts (θ, θ') of cycle C such that e_i^- is the first edge of path θ, and $\Delta(\theta) > \Delta(\theta')$. Let (Σ_i, Σ'_i) be the cut such path Σ_i is a shortest path among all first components of cuts from set $Cut(e_i^+)$. Let (Θ_i, Θ'_i) be the cut such path Θ_i is a shortest path among all first components of cuts from set $Cut(e_i^-)$.

Then set $\{(\pi_1, \pi'_1), \ldots, (\pi_k, \pi'_k)\} = \{(\Sigma_i, \Sigma'_i), (\Theta_i, \Theta'_i)\}_{i=1}^{l}$, which proves first assertion.

Rewriting rule $R(s_i^+)$ (from Definition 2.1 of symmetrized forms) is $\Delta(\mathrm{C}, (\Sigma_i, \Sigma'_i))$ and rewriting rule $R(s_i^-)$ is $\Delta(\mathrm{C}, (\Theta_i, \Theta'_i))$. Then $Sym(\Delta(\mathrm{C})) = \Delta(\mathrm{C}, \{(\pi_1, \pi'_1), \ldots, (\pi_k, \pi'_k)\})$. □

Remark 4.1. We can give an alternative definition of set of symmetrized forms of a rewriting rule or an equation.

DEFINITION 4.1 (Alternative to Definition 2.1). Let $R: \ (a_1 \cdot \ldots \cdot a_l, b_1 \cdot \ldots \cdot b_m)$ be a rewriting rule or an equation. Let $Perm(R)$ be the largest set of rewriting rules $l \to r$ such that $l > r$ and $l \cdot r^{-1}$ is a cyclic permutation of string $a_1 \cdot \ldots \cdot a_l \cdot b_m^{-1} \cdot \ldots \cdot b_1^{-1}$ or of string $b_1 \cdot \ldots \cdot b_m \cdot a_l^{-1} \cdot \ldots \cdot a_1^{-1}$. Let letter g be in set $\{a_i, a_i^{-1}\}_{i=1}^{l} \cup \{b_j, b_j^{-1}\}_{j=1}^{m}$. Let $Perm(g)$ be the subset of set $Perm(R)$, such that for any rule $l \to r$ in $Perm(g)$ the last letter of string l is g. Denote by $R(g)$ the rule in $Perm(g)$ with shortest left-hand side.

The union for all g in $\{a_i, a_i^{-1}\}_{i=1}^{l} \cup \{b_j, b_j^{-1}\}_{j=1}^{m}$, of rewriting rules $R(g)$ is called the *set of symmetrized forms of* R and is denoted by $Sym(R)$.

Proposition 4.2, analogous to Proposition 4.1, give a new property of set $Sym(\Delta(\mathrm{C}))$ of symmetrized forms.

PROPOSITION 4.2. Let C be a cycle in a PATCH Graph. Let $\{(\pi_1, \pi_1'), \ldots, (\pi_k, \pi_k')\}$ be the largest set of cuts of C such that for all $i, j \in \{1, \ldots, k\}$ such that $i \neq j$, $\pi_i \not\subset \pi_j$ and $\Delta(\pi_i) > \Delta(\pi_i')$. Then

- for any edge $e \in$ C, there is an $i \in \{1, \ldots, k\}$ such that e is the last edge of path π_i, and
- set $\Delta(\mathrm{C}, \{(\pi_1, \pi_1'), \ldots, (\pi_k, \pi_k')\})$ of rewriting rules coincides with set $Sym(\Delta(\mathrm{C}))$ of all symmetrized forms of $\Delta(\mathrm{C})$.

4.2. Example

Recall that $>$ is supposed to be a length-lexicographical ordering with precedence

$$a > a^{-1} > b > b^{-1} > \ldots > e > e^{-1} > \varepsilon.$$

EXAMPLE 4.1. Consider a new cycle $\mathrm{C}_1 : e_1.e_2.e_3.e_4$, shown in Figure 6.

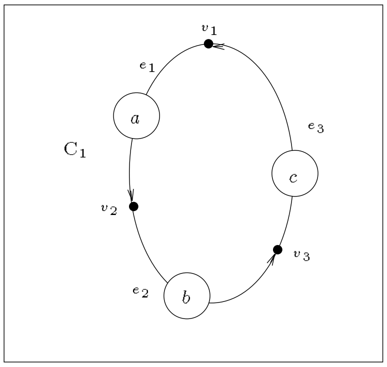

Figure 6. Example 4.1: A new cycle C_1

Compute the set of all cuts of C_1. This is the set of 18 following cuts:

$$\left\{\begin{array}{lll}
(e_1^+, e_3^-.e_2^-), & (e_1^+.e_2^+, e_3^-), & (e_1^+.e_2^+.e_3^+, \epsilon) \\
(e_1^-, e_2^+.e_3^+), & (e_1^-.e_3^-, e_2^+), & (e_1^-.e_3^-.e_2^-, \epsilon) \\
(e_2^+, e_1^-.e_3^-), & (e_2^+.e_3^+, e_1^-), & (e_2^+.e_3^+.e_1^+, \epsilon) \\
(e_2^-, e_3^+.e_1^+), & (e_2^-.e_1^-, e_3^+), & (e_2^-.e_1^-.e_3^-, \epsilon) \\
(e_3^+, e_2^-.e_1^-), & (e_3^+.e_1^+, e_2^-), & (e_3^+.e_1^+.e_2^+, \epsilon) \\
(e_3^-, e_1^+.e_2^+), & (e_3^-.e_2^-, e_1^+), & (e_3^-.e_2^-.e_1^-, \epsilon)
\end{array}\right\},$$

where, we recall, ϵ denotes the empty path. We select cuts $(\pi_i, \pi_i'), 1 \leq i \leq 18$ such that $\Delta(\pi_i) > \Delta(\pi_i')$. For length-lexicographical ordering $>$ this is equivalent to select cuts such that the length of path π_i is bigger than the length of path π_i'. These are second and third cuts in each line:

$$\left\{\begin{array}{ll} (e_1^+.e_2^+, e_3^-), & (e_1^+.e_2^+.e_3^+, \epsilon) \\ (e_1^-.e_3^-, e_2^+), & (e_1^-.e_3^-.e_2^-, \epsilon) \\ (e_2^+.e_3^+, e_1^-), & (e_2^+.e_3^+.e_1^+, \epsilon) \\ (e_2^-.e_1^-, e_3^+), & (e_2^-.e_1^-.e_3^-, \epsilon) \\ (e_3^+.e_1^+, e_2^-), & (e_3^+.e_1^+.e_2^+, \epsilon) \\ (e_3^-.e_2^-, e_1^+), & (e_3^-.e_2^-.e_1^-, \epsilon) \end{array}\right\}.$$

Finally, we select cuts (π_i, π'_i) and (π_j, π'_j), such that $\pi_i \not\subset \pi_j, i \neq j$. We get the set of 6 cuts. We enumerate them as follows:

$$\left\{\begin{array}{ll} cut\ 1 & (e_1^+.e_2^+, e_3^-) \\ cut\ 2 & (e_1^-.e_3^-, e_2^+) \\ cut\ 3 & (e_2^+.e_3^+, e_1^-) \\ cut\ 4 & (e_2^-.e_1^-, e_3^+) \\ cut\ 5 & (e_3^+.e_1^+, e_2^-) \\ cut\ 6 & (e_3^-.e_2^-, e_1^+) \end{array}\right\}.$$

We put beginning and end markers corresponding to these cuts along cycle C_1 (see Figure 7. For *cut* 2 $(e_1^-.e_3^-, e_2^+)$, we put beginning marker $(\mathbf{B}, 1, 2, e_1, e_2)$ on vertex v_2. Components of the marker have the following meanings:

- $\mathbf{B}$: vertex v_2 is beginning vertex of paths $e_1^-.e_3^-$ and e_2^+,
- number 1: vertex v_2 belongs to cycle number 1 C_1,
- number 2: vertex v_2 is one of ends of paths from *cut* 2,
- e_1: edge e_1 is the first edge of first path in the cut,
- e_2: edge e_2 is the first edge of second path in the cut.

We also put end marker $(\mathbf{E}, 1, 2, e_3, e_2)$ on vertex v_3. Components of this marker mean:

- $\mathbf{E}$: vertex v_3 is end vertex of paths $e_1^-.e_3^-$ and e_2^+,
- number 1: same as before,
- number 2: same as before,
- e_3: edge e_3 is the last edge of first path in the cut,
- e_2: edge e_2 is the last edge of second path in the cut.

We put other five beginning and five end markers in the same way. The set of all markers for the same vertex v forms label $\eta(v)$.

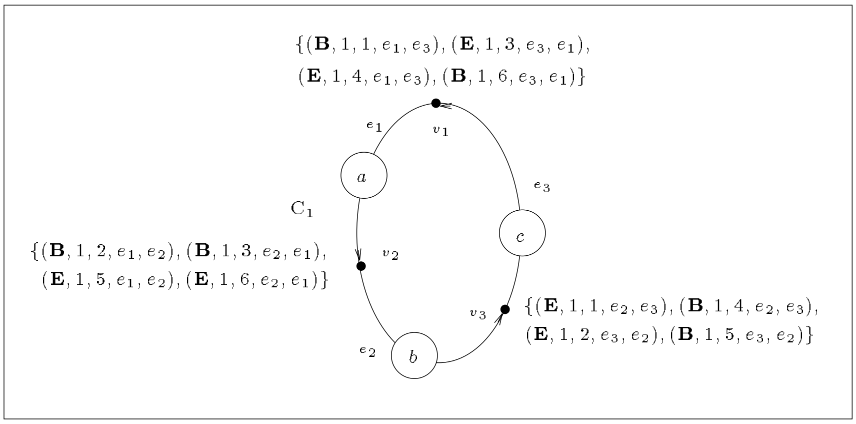

Figure 7. Example 4.1: Cycle C_1 with beginning and end markers

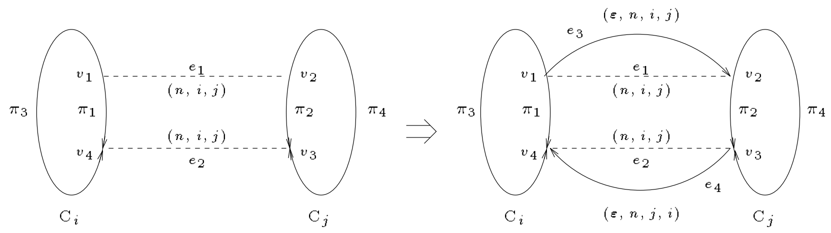

Figure 8. Patch inference rule

4.3. **Patch** Inference Rule

Patch

The premise of the inference rule shows an M-bridge (e_1, e_2) relative to common paths π_1 and π_2 for cycles C_i and C_j (see Figure 8). Matching edges are drawn as dashed, and concatenation edges as solid lines.

DEFINITION 4.2. Let $\pi_1 \subseteq C_i$ and $\pi_2 \subseteq C_j$ be common paths. An M-bridge (e_1, e_2) is said to be *relative to common paths* π_1 *and* π_2 if the following condition holds. Let v_1 be initial vertex and v_4 final vertex of π_1, and v_2 be initial vertex and v_3 final vertex of π_2. Then $e_1 = (v_1, v_2)$, $e_2 = (v_3, v_4)$, and $(n, i, j) \in \lambda(e_1)$ and $(n, i, j) \in \lambda(e_2)$ for some natural number n.

Let us denote by π_3 path $C_i \backslash \pi_1$, and by π_4 path $C_j \backslash \pi_2$.

If one of the following conditions is satisfied:

1. $\Delta(\pi_1) > \Delta(\pi_3)$ and $\Delta(C_i, (\pi_1, \pi_3))$ is a rule $R_{(i,l)} : \Delta(\pi_1) \rightarrow \Delta(\pi_3)$, and $\Delta(\pi_2)$ is a substring of the lhs or the rhs of a rule $R_{(j,m)}$,

2. $\Delta(\pi_1) \not> \Delta(\pi_3)$ or there are no rules $R_{(i,l)}$ and $R_{(j,m)}$ as in case 1, but there exist a rule $R_{(i,p)}$, and a rule $R_{(j,q)}$, such that $\Delta(\pi_1)$ is a suffix of the lhs of $R_{(i,p)}$ and $\Delta(\pi_2)$ is a prefix of the lhs of $R_{(j,q)}$,

then inference **Patch** creates a new cycle C_n from ancestors C_i and C_j, adding a new C-bridge (e_3, e_4), where $e_3 = (v_1, v_2), e_4 = (v_3, v_4)$, $\lambda(e_1) = (\varepsilon, n, i, j), \lambda(e_2) = (\varepsilon, n, j, i)$. If first condition is satisfied, beginning and end markers defining all the rules $R_{(j,m)}$, such that $\Delta(\pi_2)$ is a substring of the lhs or the rhs of $R_{(j,m)}$, are removed. **Patch** inference rule can be applied only once to any M-bridge (e_1, e_2) relative to common paths π_1 and π_2.

We remark that it is easy to verify the conditions of **Patch** inference rule application, analyzing beginning and end markers of C_i and C_j. Actually, first condition is satisfied when $(\mathbf{B}, i, l, e_5, *) \in \eta(v_1)$ and $(\mathbf{E}, i, l, e_6, *) \in \eta(v_4)$, where e_5 and e_6 are respectively the first and last edges of π_1, and beginning and end markers $(\mathbf{B}, j, m, *, *)$ and $(\mathbf{E}, j, m, *, *)$, defining $R_{(j,m)}$, are labels of vertices belonging to π_4. Second condition is satisfied when $(\mathbf{E}, i, l, e_6, *) \in \eta(v_4)$ and $(\mathbf{B}, i, l, *, *)$ appears in a label of a vertex belonging to π_3, and $(\mathbf{B}, j, m, e_7, *) \in \eta(v_2)$ (where e_7 is the first edge of π_2) and $(\mathbf{E}, j, m, *, *)$ appears in a label of a vertex belonging to π_4. Also, $\Delta(\pi_2)$ is a substring of the lhs or the rhs of $R_{(j,m)}$ if an only if the vertex containing beginning marker $(\mathbf{B}, j, m, *, *)$ precedes in path π_4 the vertex containing end marker $(\mathbf{E}, j, m, *, *)$.

Patch inference rule computes a critical pair or a simplification for each pair of rules represented by cycles C_i and C_j respectively. Actually, first case corresponds to a simplification of $R_{(j,m)}$ by $R_{(i,l)}$, and second case corresponds to a critical pair between $R_{(i,p)}$ and $R_{(j,q)}$.

Beginning and end markers along cycle C_j, defining rules $R_{(j,m)}$ which can be simplified, are removed. We can not remove paths representing simplified strings, because these paths are shared between C_j and C_n. If symmetric conditions hold for C_i, then by symmetric transformation, beginning and end markers along cycle C_i, defining rules $R_{(i,l)}$, which can be simplified, are removed. We remark that when $\Delta(\pi_1) > \Delta(\pi_3)$ and $\Delta(\pi_2) > \Delta(\pi_4)$ (actually this case corresponds to a critical pair on the top), we never remove beginning and end markers in both C_i and C_j, but only the markers of C_i, when $\Delta(\pi_3) > \Delta(\pi_4)$, or only the markers of C_j, when $\Delta(\pi_4) > \Delta(\pi_3)$. Then we keep smaller rewriting rules in the system represented by a PATCH graph.

Inference **Patch** is performed only for maximal non-trivial common paths.

DEFINITION 4.3. Let $\pi_1 \subseteq \mathrm{C}_i$ and $\pi_2 \subseteq \mathrm{C}_j$ be common paths. They are said to be *maximal common paths*, if there are no common paths $\pi_3 \subseteq \mathrm{C}_i$ and $\pi_4 \subseteq \mathrm{C}_j$ such that:

– $\pi_3 = \pi_5.\pi_1.\pi_6$ and $\pi_4 = \pi_7.\pi_2.\pi_8$,

– $\Delta(\pi_5) = \Delta(\pi_7)$ and $\Delta(\pi_6) = \Delta(\pi_8)$,

– π_5 or π_6 (and, consequently, π_7 or π_8) is not empty.

Common paths π_1 and π_2 are called *trivial common paths* if $\mathrm{C}_i = \mathrm{C}_j$ and $\pi_1 = \pi_2$.

Since this restriction is imposed, inference **Patch** does not produces some redundant cycles representing redundant equations.

LEMMA 4.1. *Let $\pi_1, \pi_3 \subseteq C_i$ and $\pi_2, \pi_4 \subseteq C_j$. Let π_1, π_2 and π_3, π_4 be common paths such that $\pi_3 = \pi_5.\pi_1.\pi_6$, $\pi_4 = \pi_7.\pi_2.\pi_8$, and $\Delta(\pi_5) = \Delta(\pi_7)$, $\Delta(\pi_6) = \Delta(\pi_8)$. Let π_5 or π_6 be non-empty. Denote by C_k (resp. C_l) the new cycle, generated applying* **Patch** *inference rule to the M-bridge relative to common paths π_1 and π_2 (resp. relative to common paths π_3 and π_4).*

Then for any cut (π_9, π_{10}) of cycle C_k, there is some cut (π_{11}, π_{12}) of cycle C_l, such that

- *rewriting rule $R_2 = \Delta(C_l, (\pi_{11}, \pi_{12}))$ belongs to set $Sym(\Delta(C_l))$, and*

- *rule $R_1 = \Delta(C_k, (\pi_9, \pi_{10}))$ can be simplified to a trivial equation by rewriting rule R_2 and group axioms $\mathcal{A}$.*

Proof. Let $\Delta(\pi_1) = t$, $\Delta(\pi_5) = s$ and $\Delta(\pi_6) = u$. Denote by π_{13} path $\mathrm{C}_i \backslash \pi_3$ and by π_{14} path $\mathrm{C}_i \backslash \pi_4$. Let $\Delta(\pi_{13}) = v$ and $\Delta(\pi_{14}) = w$.

Let v_1 be initial vertex of paths π_9 and π_{10}, and v_2 be their final vertex. Thereafter the proof differs following to the positions of vertices v_1 and v_2 in cycle C_k. Let us suppose that $v_1 \in \pi_4$ and $v_2 \in \pi_{14}$ (see Figure 9). Other cases are similar. We do not detail them because they need a lot of new notation.

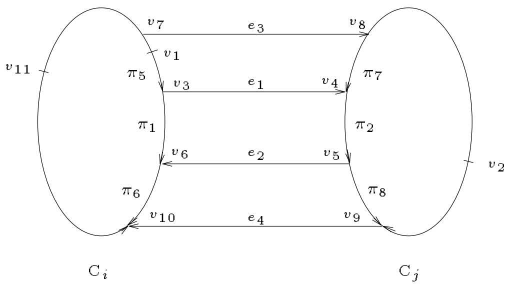

Figure 9. Illustration of Lemma 4.1

Let (e_1, e_2) be C-bridge defining cycle C_k, and (e_3, e_4) be C-bridge defining cycle C_l. Let $e_1 = (v_3, v_4)$, $e_2 = (v_5, v_6)$, $e_3 = (v_7, v_8)$ and $e_4 = (v_9, v_{10})$.

Denote by π_{15} the prefix of path π_5, going from vertex v_7 to vertex v_1, and by π_{16} the suffix of path π_5, going from vertex v_1 to vertex v_3. Let $\Delta(\pi_{15}) = p$ and $\Delta(\pi_{16}) = q$. Then path $\pi_5 = \pi_{15}.\pi_{16}$ and string $s^{-1} = q^{-1} \cdot p^{-1}$. Also, let π_{17} be the prefix of path π_{14}, going from vertex v_8 to vertex v_2, and let π_{18} be the suffix of path π_{14}, going from vertex v_2 to vertex v_9. Let $\Delta(\pi_{17}) = l$ and $\Delta(\pi_{18}) = r$. Then path $\pi_{14} = \pi_{17}.\pi_{18}$ and string $w = l \cdot r$.

Rule R_1 is $R_1 : \ q \cdot s^{-1} \cdot l \rightarrow p^{-1} \cdot v \cdot u^{-1} \cdot u \cdot r^{-1}$.

Consider the cut (π_{11}, π_{12}) of cycle C_l, such that vertex v_2 is final vertex of path π_{11}, last edge of π_{11} is the same that last edge of path π_{10} and rewriting rule $R_2 = \Delta(\mathrm{C}_l, (\pi_{11}, \pi_{12}))$ belongs to set $Sym(\Delta(\mathrm{C}_l))$. By Proposition 4.2, such a cut of cycle C_l exists.

Denote by v_{11} initial vertex of paths π_{11} and π_{12}. Suppose that vertex $v_{11} \in \pi_{13}$. The proof in other cases is analogous.

Let path $\pi_{13} = \pi_{19}.\pi_{20}$, where path π_{19} is the prefix of π_{13} going from vertex v_7 to vertex v_{11}, and path π_{20} is the suffix of π_{13} going from vertex v_{11} to vertex v_{10}. Let $\Delta(\pi_{19}) = g$ and $\Delta(\pi_{20}) = d$. Then string $v = g \cdot d$ and rule R_2 is $R_2 :\ d \cdot r^{-1} \to g^{-1} \cdot l$.

$$q \cdot s^{-1} \cdot l = q \cdot q^{-1} \cdot p^{-1} \cdot l \to_{\mathcal{A}} p^{-1} \cdot l, \text{ and}$$
$$p^{-1} \cdot v \cdot u^{-1} \cdot u \cdot r^{-1} \to_{\mathcal{A}} p^{-1} \cdot v \cdot r^{-1} = p^{-1} \cdot g \cdot d \cdot r^{-1} \to_{R_2} p^{-1} \cdot g \cdot g^{-1} \cdot l \to_{\mathcal{A}} p^{-1} \cdot l.$$

□

LEMMA 4.2. *Inference* **Patch***, performed for an M-bridge relative to trivial common paths π_1 and π_2, generates a new cycle C such that for every cut (π_3, π_4) of C, rewriting rule $R = \Delta(C, (\pi_3, \pi_4))$ can be simplified by group axioms to a trivial equation.*

Proof. Let $\pi_1 \in \mathrm{C}_i$, $\pi_2 \in \mathrm{C}_j$. Since π_1 and π_2 are trivial common paths, we have $\mathrm{C}_i = \mathrm{C}_j$ and $\pi_1 = \pi_2$. Let (e_1, e_2) be the C-bridge, defining cycle C, generated by **Patch** rule.

Consider a cut (π_3, π_4) of C. Let v_1 be initial, and v_2 be final vertex of path π_1, and let v_3 be initial vertex, and v_4 be final vertex of paths π_3 and π_4. We remark that both of vertices v_3 and v_4 belong to path $C_i \backslash \pi_1$, which coincides with path $C_j \backslash \pi_2$. Let us suppose that vertex v_3 precedes vertex v_4 in path $C_i \backslash \pi_1$. The proof in other case is analogous.

Let π_5 and π_6 be the subpaths of cycle C_i such that π_5 goes from vertex v_3 to vertex v_1, and π_6 goes from vertex v_2 to vertex v_4 (see Figure 10). Then $\pi_3 = \pi_5^+.e_1^+.(\mathrm{C}_j \backslash \pi_2)^+.e_2^+.\pi_6^+$ and $\pi_4 = (\mathrm{C}_i \backslash (\pi_5.\pi_1.\pi_6))^+$.

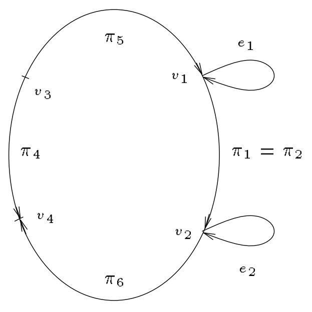

Figure 10. Illustration of Lemma 4.2

Let $\Delta(\pi_5) = s, \Delta(\pi_1) = t, \Delta(\pi_6) = u$ and $\Delta(C_i \backslash (\pi_5.\pi_1.\pi_6)) = v$. Rule $R = \Delta(C, (\pi_3, \pi_4))$ is $R :\ s \cdot s^{-1} \cdot v \cdot t^{-1} \cdot t \to v$. String $s \cdot s^{-1} \cdot v \cdot t^{-1} \cdot t \xrightarrow{*}_{\mathcal{A}} v$. □

4.4. Examples

We illustrate different cases, when **Patch** inference rule can or can not be applied to a pair of cycles in a PATCH graph. For all examples of this section let $>$ be the length-lexicographical ordering with precedence

$$a > a^{-1} > b > b^{-1} > \ldots > e > e^{-1} > \varepsilon.$$

EXAMPLE 4.2 (Simplification Case). Let us recall the example we have just given in Sect.3:

$$\mathcal{R} = \{R_1: \ a \cdot b \cdot c \to d, R_2: \ b \to e\}.$$

Figure 11 shows cycles C_1 and C_2 representing the set of symmetrized forms of rule R_1 and the set of symmetrized forms of rule R_2 respectively, and M-bridge (e_7, e_8) relative to maximal common paths $\pi_1 = e_3$ and $\pi_2 = e_5$ for cycles C_1 and C_2. The labels of edges e_7 and e_8 are supposed to be $\lambda(e_7) = \lambda(e_8) = (3, 1, 2)$.

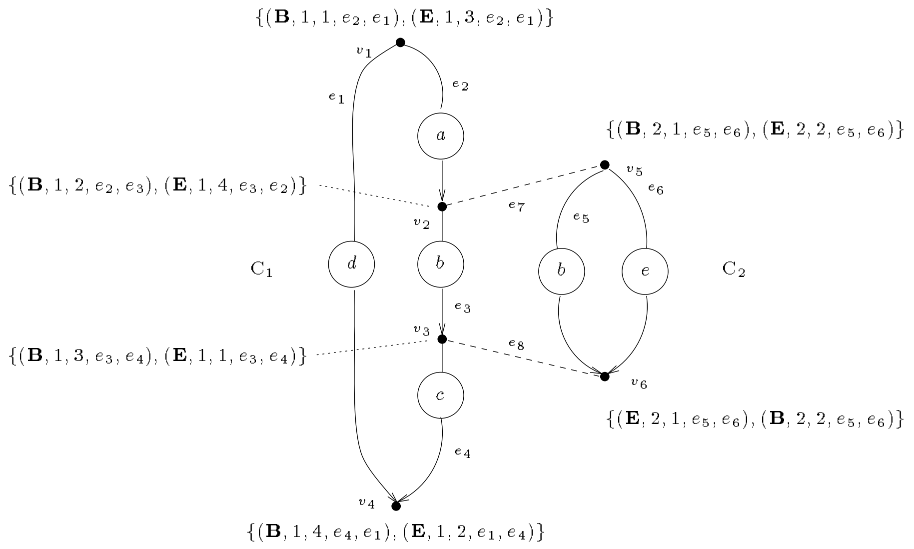

Figure 11. Example 4.2: Cycles C_1 and C_2 with M-bridge (e_7, e_8)

Let us denote by π_3 path $C_1 \backslash \pi_1$, and by π_4 path $C_2 \backslash \pi_2$. Considering cycle C_2, we have:

– $\Delta(\pi_2^+) = b$, $\Delta(\pi_4^+) = e$ and $b > e$. Then $\Delta(C_2, (\pi_2^+, \pi_4^+))$ is rule $R_{(2,1)}: \ b \to e$.

  Since in path π_3^+ is $e_2^-.e_1^+.e_4^-$, and vertex v_1 such that $(\mathbf{B}, 1, 1, e_2, e_1) \in \eta(v_1)$ precedes vertex v_3 such that $(\mathbf{E}, 1, 1, e_3, e_4) \in \eta(v_3)$, we deduce that $\Delta(\pi_1^+)$ is a substring of the lhs of rule $R_{(1,1)}$, and that first condition of **Patch** premise is satisfied.

Patch inference rule is applicable.

  Rule $R_{(2,1)}$ simplifies rule $R_{(1,1)}$. In the same way, since vertex v_2 having beginning marker $(\mathbf{B}, 1, 2, e_2, e_3)$ precedes in path π_3^+ vertex v_4, having end marker $(\mathbf{E}, 1, 2, e_1, e_4)$, string $\Delta(\pi_1^+)$ is a substring of the rhs of rule $R_{(1,2)}$. Thus, rule $R_{(2,1)}$ simplifies rule $R_{(1,2)}$.

Taking cycle C_1, we remark that $\Delta(\pi_1^+) = b, \Delta(\pi_3^+) = a^{-1} \cdot d \cdot c^{-1}$ and $b \not> a^{-1} \cdot d \cdot c^{-1}$. Therefore, markers $(\mathbf{B}, 1, 1, e_2, e_1), (\mathbf{E}, 1, 1, e_3, e_4), (\mathbf{B}, 1, 2, e_2, e_3)$ and $(\mathbf{E}, 1, 2, e_1, e_4)$ have to be removed.

– $\Delta(\pi_2^-) = b^{-1}$, $\Delta(\pi_4^-) = e^{-1}$ and $b^{-1} > e^{-1}$. Then $\Delta(C_2, (\pi_2^-, \pi_4^-))$ is rule $R_{(2,2)} : \; b^{-1} \to e^{-1}$.

Since in path $\pi_3^- = e_4^+.e_1^-.e_2^+$, and vertex v_3 such that $(\mathbf{B}, 1, 3, e_3, e_4) \in \eta(v_3)$ precedes vertex v_1 such that $(\mathbf{E}, 1, 3, e_2, e_1) \in \eta(v_1)$, we deduce that $\Delta(\pi_1^-)$ is a substring of the lhs of rule $R_{(1,3)}$. Rule $R_{(2,2)}$ simplifies rule $R_{(1,3)}$. In the same way, since vertex v_4 having beginning marker $(\mathbf{B}, 1, 4, e_4, e_1)$ precedes in path π_3^- vertex v_2, having end marker $(\mathbf{E}, 1, 4, e_3, e_2)$, string $\Delta(\pi_1^-)$ is a substring of the rhs of rule $R_{(1,4)}$. Consequently, rule $R_{(2,2)}$ simplifies rule $R_{(1,4)}$.

Taking cycle C_1, we get $\Delta(\pi_1^-) = b^{-1}, \Delta(\pi_3^-) = c \cdot d^{-1} \cdot a$ and $b^{-1} \not> c \cdot d^{-1} \cdot a$. Then, markers $(\mathbf{B}, 1, 3, e_3, e_4), (\mathbf{E}, 1, 3, e_2, e_1), (\mathbf{B}, 1, 4, e_4, e_1)$ and $(\mathbf{E}, 1, 4, e_3, e_2)$ have to be removed.

Thereby, all rules $R_{(1,1)}, \ldots, R_{(1,4)}$, represented by cycle C_1, are simplified by rules $R_{(2,1)}$ and $R_{(2,2)}$, represented by cycle C_2.

Applying **Patch** inference rule, we create a new cycle C_3, adding to the graph new C-bridge (e_9, e_{10}), relative to common paths π_1 and π_2. Edge e_9 goes from vertex v_2 to vertex v_5, and edges e_{10} from vertex v_6 to vertex v_3. Their labels are $\lambda(e_9) = (\varepsilon, 3, 1, 2)$ and $\lambda(e_{10}) = (\varepsilon, 3, 2, 1)$.

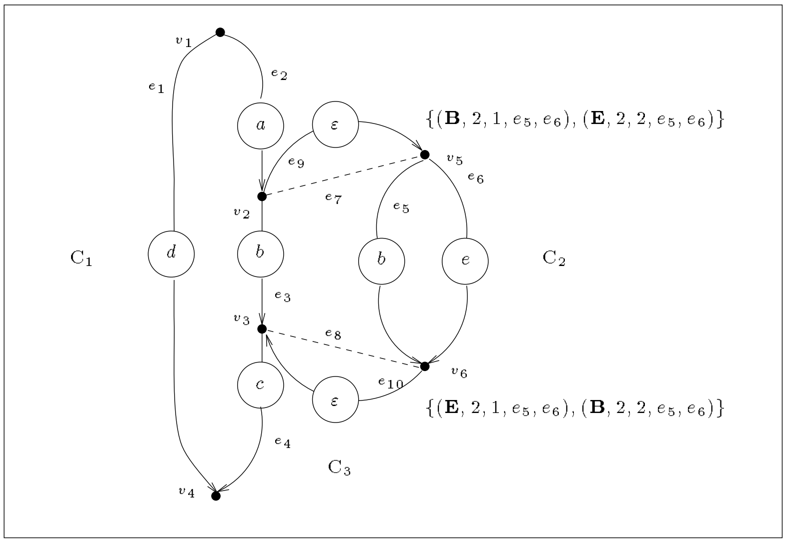

Figure 12. Example 4.2: Cycles C_1, C_2 and C_3

Figure 12 shows the PATCH Graph after **Patch** inference application.

EXAMPLE 4.3 (Reciprocal Simplification Case). Consider rewriting system

$$\mathcal{R} = \{R_1 : \ a \cdot b \to c \cdot d, R_2 : \ a \cdot b \to e\}.$$

Figure 13 shows cycles C_1 and C_2 which represent respectively the set of symmetrized forms of rule R_1:

$$\left\{\begin{array}{l} R_{(1,1)} : \ a \cdot b \to c \cdot d \\ R_{(1,2)} : \ a^{-1} \cdot c \to b \cdot d^{-1} \\ R_{(1,3)} : \ b^{-1} \cdot a^{-1} \to d^{-1} \cdot c^{-1} \\ R_{(1,4)} : \ c^{-1} \cdot a \to d \cdot b^{-1} \end{array}\right\}$$

and the set of symmetrized forms of rule R_2:

$$\left\{\begin{array}{l} R_{(2,1)} : \ a \cdot b \to e \\ R_{(2,2)} : \ b \cdot e^{-1} \to a^{-1} \\ R_{(2,3)} : \ e^{-1} \cdot a \to b^{-1} \\ R_{(2,4)} : \ e \cdot b^{-1} \to a \\ R_{(2,5)} : \ a^{-1} \cdot e \to b \\ R_{(2,6)} : \ b^{-1} \cdot a^{-1} \to e^{-1} \end{array}\right\}.$$

M-edges e_8 and e_9 form an M-bridge relative to maximal common paths $\pi_1 = e_3.e_4$ and $\pi_2 = e_5.e_6$ for cycles C_1 and C_2. We suppose that labels of e_8 and e_9 are $\lambda(e_8) = \lambda(e_9) = (3, 1, 2)$.

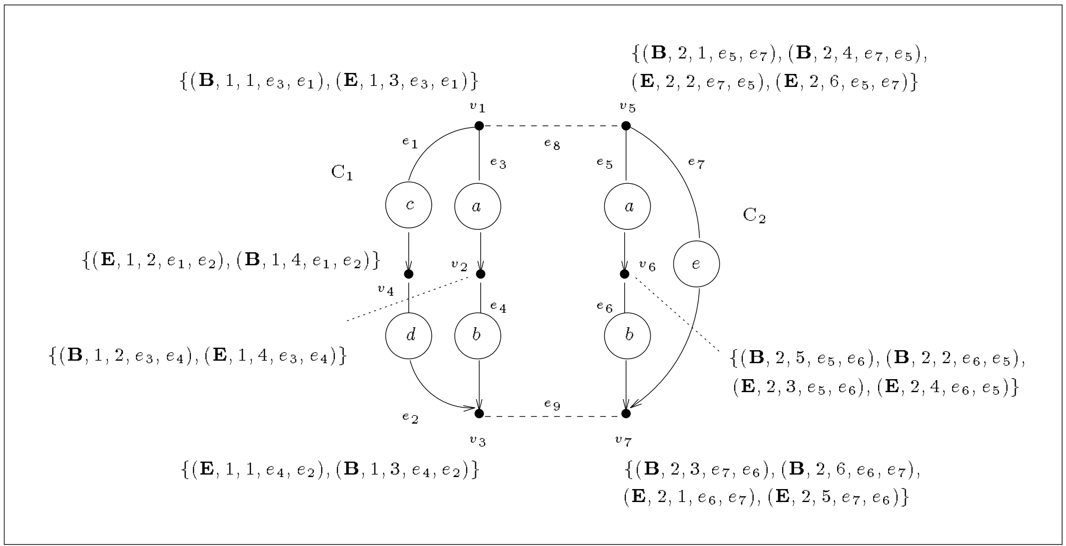

Figure 13. Example 4.3: Cycles C_1 and C_2 with M-bridge (e_8, e_9)

Let π_3 be path $C_1 \backslash \pi_1$ and π_4 be path $C_2 \backslash \pi_2$. Considering cycle C_1, we see that

- $\Delta(\pi_1^+) = a \cdot b, \Delta(\pi_3^+) = c \cdot d$ and $a \cdot b > c \cdot d$. Therefore, rule $\Delta(C_1, (\pi_1^+, \pi_3^+))$ is $R_{(1,1)} : \ a \cdot b \to c \cdot d$.

Path $\pi_4^+ = e_7^+$ contains two vertices: v_5 and v_7, and v_5 precedes v_7. Since vertex v_5 has beginning marker $(\mathbf{B}, 2, 1, e_5, e_7)$, and vertex v_7 has end marker $(\mathbf{E}, 2, 1, e_6, e_7)$, string $\Delta(\pi_2^+)$ coincides with the lhs of rule $R_{(2,1)}$. Consequently, first condition of the premise of **Patch** inference holds, and

Patch rule can be applied.

Moreover, rule $R_{(2,1)}$ can be simplified by rule $R_{(1,1)}$.

Also, $\Delta(\pi_2^+) = a \cdot b$, $\Delta(\pi_4^+) = e$ and $a \cdot b > e$. In path $\pi_3^+ = e_1^+.e_2^+$, vertex v_1 having beginning marker $(\mathbf{B}, 1, 1, e_3, e_1)$, precedes vertex v_3, having end marker $(\mathbf{E}, 1, 1, e_4, e_2)$. Thereby, string $\Delta(\pi_1^+)$ is the lhs of rule $R_{(1,1)}$, represented by cycle C_1. Rule $R_{(1,1)}$ can be simplified by rule $R_{(2,1)}:\ a \cdot b \to e$.

Comparing strings $\Delta(\pi_3^+)$ and $\Delta(\pi_4^+)$, we get $c \cdot d > e$. Thus, markers $(\mathbf{B}, 1, 1, e_3, e_1)$ and $(\mathbf{E}, 1, 1, e_4, e_2)$, defining rule $R_{(1,1)}$ have to be removed, and markers $(\mathbf{B}, 2, 1, e_5, e_7)$ and $(\mathbf{E}, 2, 1, e_6, e_7)$, defining rule $R_{(2,1)}$ have to be kept (see Figure 14).

– $\Delta(\pi_1^-) = b^{-1} \cdot a^{-1}$, $\Delta(\pi_3^-) = d^{-1} \cdot c^{-1}$ and $b^{-1} \cdot a^{-1} > d^{-1} \cdot c^{-1}$.

Path $\pi_4^- = e_7^-$ contains two vertices: v_5 and v_7, and v_7 precedes v_5. Since vertex v_7 has beginning marker $(\mathbf{B}, 2, 6, e_6, e_7)$, and vertex v_5 has end marker $(\mathbf{E}, 2, 6, e_5, e_7)$, string $\Delta(\pi_2^-)$ coincides with the lhs of rule $R_{(2,6)}$. Then rule $R_{(2,6)}$ can be simplified by rule $R_{(1,3)}:\ b^{-1} \cdot a^{-1} \to d^{-1} \cdot c^{-1}$.

Also, $\Delta(\pi_2^-) = b^{-1} \cdot a^{-1}$, $\Delta(\pi_4^-) = e^{-1}$ and $b^{-1} \cdot a^{-1} > e^{-1}$. In path $\pi_3^- = e_2^-.e_1^-$, vertex v_3 having beginning marker $(\mathbf{B}, 1, 3, e_4, e_2)$, precedes vertex v_1, having end marker $(\mathbf{E}, 1, 3, e_3, e_1)$. Then string $\Delta(\pi_1^-)$ is the lhs of rule $R_{(1,3)}$. Rule $R_{(2,6)}$ can be simplified by rule $R_{(1,3)}$.

Comparing strings $\Delta(\pi_3^-)$ and $\Delta(\pi_4^-)$, we have $d^{-1} \cdot c^{-1} > e^{-1}$. Then markers $(\mathbf{B}, 2, 6, e_6, e_7)$ and $(\mathbf{E}, 2, 6, e_5, e_7)$, defining rule $R_{(2,6)}$ have to be kept, and markers $(\mathbf{B}, 1, 3, e_4, e_2)$ and $(\mathbf{E}, 1, 3, e_3, e_1)$, defining rule $R_{(1,3)}$ have to be removed.

We remark that in this example only rules $R_{(1,1)}$ and $R_{(1,3)}$, represented by cycle C_1, are simplified.

By application of **Patch** inference, a new cycle C_3 is created. A new C-bridge (e_{10}, e_{11}) defines C_3. Edge e_{10} goes from vertex v_1 to vertex v_5, it is labelled by $(\varepsilon, 3, 1, 2)$. Edge e_{11} goes from vertex v_7 to vertex v_3, and it is labelled by $(\varepsilon, 3, 2, 1)$. The updated PATCH Graph is presented in Figure 14.

Example 4.4 shows that one graph transformation by **Patch** inference application encodes the computation of several critical critical pairs or simplifications. Note that the goal of **Patch** is not to compute all possible critical pairs between each pair of rules represented by cycles C_i and C_j respectively, but do not compute such of them, that will be certainly simplified later. Thus, the critical pair between rule $R_{(2,4)}:\ b^{-1} \cdot f^{-1} a^{-1} \to c^{-1} \cdot b^{-1} \cdot d^{-1}$ and rule $R_{(1,2)}:\ a^{-1} \cdot d \cdot e \to b \cdot c \cdot f^{-1}$ is not calculated, because this pair will be simplified by a rule represented by cycle C_3.

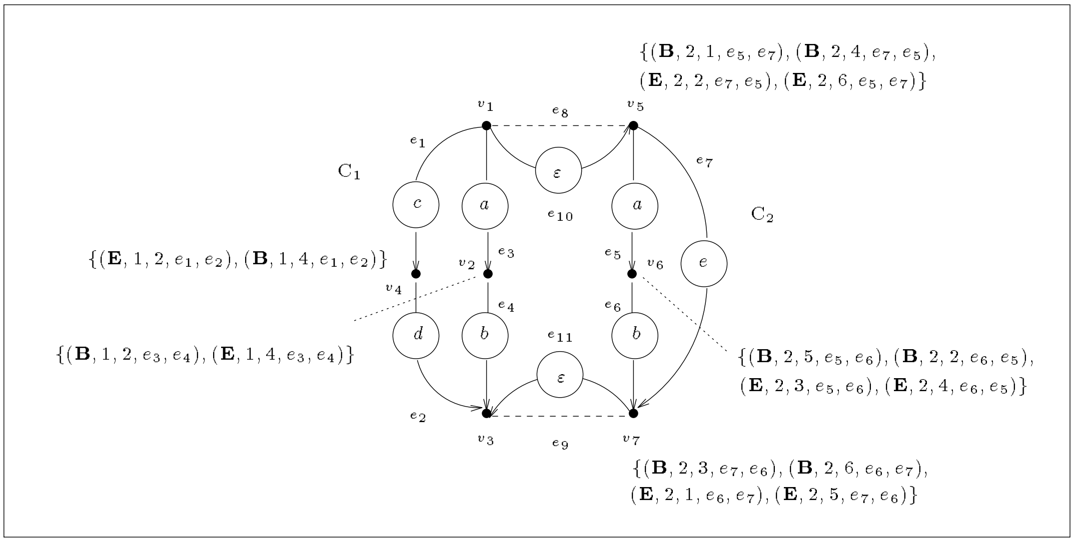

Figure 14. Example 4.3: Cycles C_1, C_2 and C_3

EXAMPLE 4.4 (Critical Pair Case). Consider rewriting system

$$\mathcal{R} = \{R_1 : \ a \cdot b \cdot c \to d \cdot e \cdot f, R_2 : \ d \cdot b \cdot c \to a \cdot f \cdot b\}.$$

We show cycle C_1, representing set

$$\left\{ \begin{array}{l} R_{(1,1)} : a \cdot b \cdot c \to d \cdot e \cdot f \\ R_{(1,2)} : a^{-1} \cdot d \cdot e \to b \cdot c \cdot f^{-1} \\ R_{(1,3)} : b^{-1} \cdot a^{-1} \cdot d \to c \cdot f^{-1} \cdot e^{-1} \\ R_{(1,4)} : c^{-1} \cdot b^{-1} \cdot a^{-1} \to f^{-1} \cdot e^{-1} \cdot d^{-1} \\ R_{(1,5)} : e^{-1} \cdot d^{-1} \cdot a \to f \cdot c^{-1} \cdot b^{-1} \\ R_{(1,6)} : d^{-1} \cdot a \cdot b \to e \cdot f \cdot c^{-1} \end{array} \right\}$$

of symmetrized forms of rule R_1, and cycle C_2, representing set

$$\left\{ \begin{array}{l} R_{(2,1)} : a \cdot f \cdot b \to d \cdot b \cdot c \\ R_{(2,2)} : b \cdot c \cdot b^{-1} \to d^{-1} \cdot a \cdot f \\ R_{(2,3)} : b^{-1} \cdot d^{-1} \cdot a \to c \cdot b^{-1} \cdot f^{-1} \\ R_{(2,4)} : b^{-1} \cdot f^{-1} \cdot a^{-1} \to c^{-1} \cdot b^{-1} \cdot d^{-1} \\ R_{(2,5)} : b \cdot c^{-1} \cdot b^{-1} \to f^{-1} \cdot a^{-1} \cdot d \\ R_{(2,6)} : a^{-1} \cdot d \cdot b \to f \cdot b \cdot c^{-1} \end{array} \right\}$$

of symmetrized forms of rule R_2 in Figure 15. Pair (e_{13}, e_{14}) of M-edges is the M-bridge relative to maximal common paths $\pi_1 = e_5.e_4$ and $\pi_2 = e_8.e_9$ for cycles C_1 and C_2.

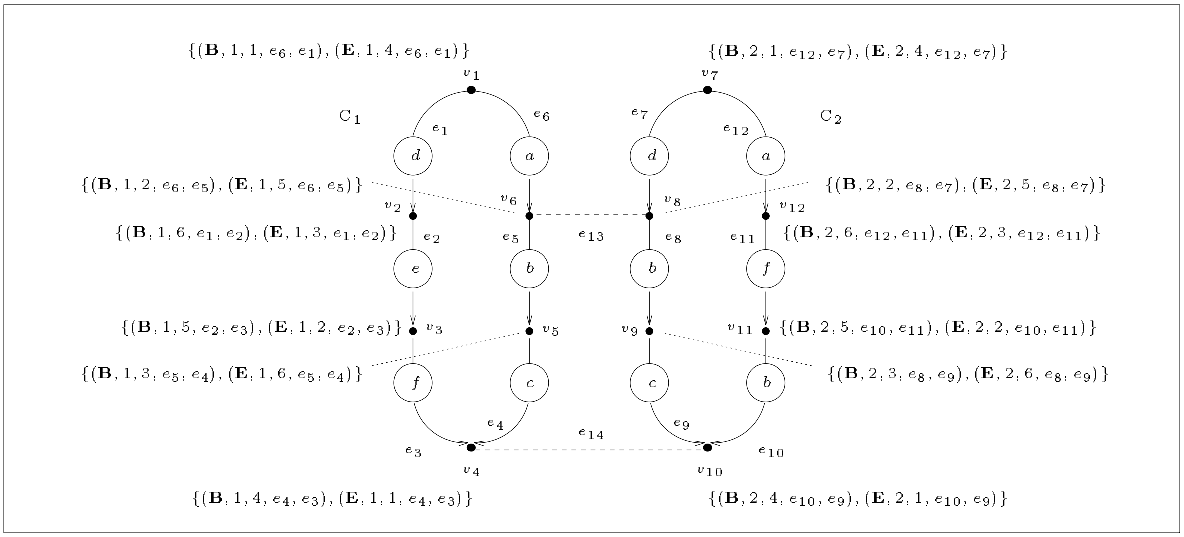

Figure 15. Example 4.4: Cycles C_1, C_2 and M-bridge (e_{13}, e_{14})

Denote by π_3 path $\mathrm{C}_1 \backslash \pi_1$, and by π_4 path $\mathrm{C}_2 \backslash \pi_2$. We have: $\pi_3 = e_6^-.e_1^+.e_2^+.e_3^+$ and $\pi_4 = e_7^-.e_{12}^+.e_{11}^+.e_{10}^+$. Also, vertex v_6 is initial vertex of paths π_1 and π_3, and vertex v_4 is final vertex of them. Vertex v_8 is initial vertex of paths π_2 and π_4, and vertex v_{10} is their final vertex.

Consider cycle C_1. We have

– $\Delta(\pi_1^+) = b \cdot c, \Delta(\pi_3^+) = a^{-1} \cdot d \cdot e \cdot f$ and $b \cdot c \not\succ a^{-1} \cdot d \cdot e \cdot f$.

  Vertex v_4 contains end marker $(\mathbf{E}, 1, 1, e_4, e_3)$, and vertex v_1, which belong to path π_3^+, contains beginning marker $(\mathbf{B}, 1, 1, e_6, e_1)$. In addition, vertex v_8 has beginning marker $(\mathbf{B}, 2, 2, e_{12}, e_7)$, and vertex v_{11}, belonging to path π_4^+, has end marker $(\mathbf{E}, 2, 2, e_{10}, e_{11})$.

  Then second condition of **Patch** inference rule premise is satisfied.

  Patch rule can be applied.

– $\Delta(\pi_1^-) = c^{-1} \cdot b^{-1}, \Delta(\pi_3^-) = f^{-1} \cdot e^{-1} \cdot d^{-1} \cdot a$ and $c^{-1} \cdot b^{-1} \not\succ f^{-1} \cdot e^{-1} \cdot d^{-1} \cdot a$.

  Vertex v_6 contains end marker $(\mathbf{E}, 1, 5, e_6, e_5)$, and vertex v_3, which belong to path π_3^-, contains beginning marker $(\mathbf{B}, 1, 5, e_2, e_3)$. Also, vertex v_{10} has beginning marker $(\mathbf{B}, 2, 4, e_{10}, e_9)$, and vertex v_7, belonging to path π_4^-, has end marker $(\mathbf{E}, 2, 4, e_{12}, e_7)$.

We deduce that there are critical pairs between rule $R_{(1,1)}: \ a \cdot b \cdot c \rightarrow d \cdot e \cdot f$ and rule $R_{(2,2)}: \ b \cdot c \cdot b^{-1} \rightarrow d^{-1} \cdot a \cdot f$, and between rule $R_{(1,5)}: \ e^{-1} \cdot d^{-1} \cdot a \rightarrow f \cdot c^{-1} \cdot b^{-1}$ and rule $R_{(2,4)}: \ b^{-1} \cdot f^{-1} \cdot a^{-1} \rightarrow c^{-1} \cdot b^{-1} \cdot d^{-1}$.

Consider cycle C_2. In the same way, we discover that

– $\Delta(\pi_2^-) = c^{-1} \cdot b^{-1}, \Delta(\pi_4^-) = b^{-1} \cdot f^{-1} \cdot a^{-1} \cdot d$ and $c^{-1} \cdot b^{-1} \not\succ b^{-1} \cdot f^{-1} \cdot a^{-1} \cdot d$.

Vertex v_8 contains end marker $(\mathbf{E}, 2, 5, e_8, e_7)$, and vertex v_{11}, which belong to path π_4^-, contains beginning marker $(\mathbf{B}, 2, 5, e_8, e_7)$. Vertex v_4 has beginning marker $(\mathbf{B}, 1, 4, e_4, e_3)$, and vertex v_1, belonging to path π_3^-, has end marker $(\mathbf{E}, 1, 4, e_6, e_1)$.

Consequently, there is a critical pair between rule $R_{(2,5)}: \ b \cdot c^{-1} \cdot b^{-1} \to f^{-1} \cdot a^{-1} \cdot d$ and rule $R_{(1,4)}: \ c^{-1} \cdot b^{-1} \cdot a^{-1} \to f^{-1} \cdot e^{-1} \cdot d^{-1}$.

Applying inference rule **Patch**, we create a new C-bridge (e_{15}, e_{16}), such that $e_{15} = (v_6, v_8), e_{16} = (v_{10}, v_4)$, and $\lambda(e_{15}) = (\varepsilon, 3, 1, 2), \lambda(e_{16}) = (\varepsilon, 3, 2, 1)$ (see Fig.16). This C-bridge defines new cycle C_3.

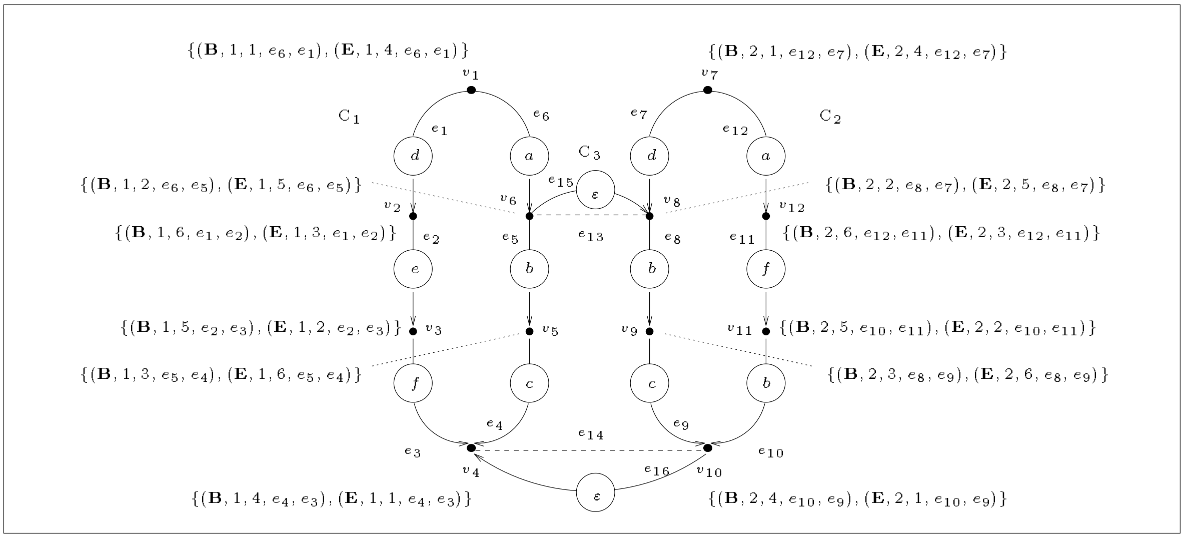

Figure 16. Example 4.4: Cycles C_1, C_2 and C_3

EXAMPLE 4.5 (**Patch** is not applicable). Consider system of rewriting rules

$$\mathcal{R} = \{R_1: \ a \cdot b \to c \cdot d, R_2: \ b \cdot c \to c \cdot d\}.$$

Cycle C_1 in Fig.17 represents the set of symmetrized forms of rule R_1:

$$\left\{ \begin{array}{l} R_{(1,1)}: \ a \cdot b \to c \cdot d \\ R_{(1,2)}: \ a^{-1} \cdot c \to b \cdot d^{-1} \\ R_{(1,3)}: \ b^{-1} \cdot a^{-1} \to d^{-1} \cdot c^{-1} \\ R_{(1,4)}: \ c^{-1} \cdot a \to d \cdot b^{-1} \end{array} \right\},$$

and cycle C_2 represents the set of symmetrized forms of rule R_2:

$$\left\{ \begin{array}{l} R_{(2,1)} : b \cdot c \to c \cdot d \\ R_{(2,2)} : c^{-1} \cdot b \to d \cdot c^{-1} \\ R_{(2,3)} : c^{-1} \cdot b^{-1} \to d^{-1} \cdot c^{-1} \\ R_{(2,4)} : b^{-1} \cdot c \to c \cdot d^{-1} \end{array} \right\}.$$

M-bridge (e_9, e_{10}) is relative to maximal common paths $\pi_1 = e_3.e_4$ and $\pi_2 = e_5.e_6$. Path π_1 goes from vertex v_1 to vertex v_3, and path π_2 goes from vertex v_5 to vertex v_7.

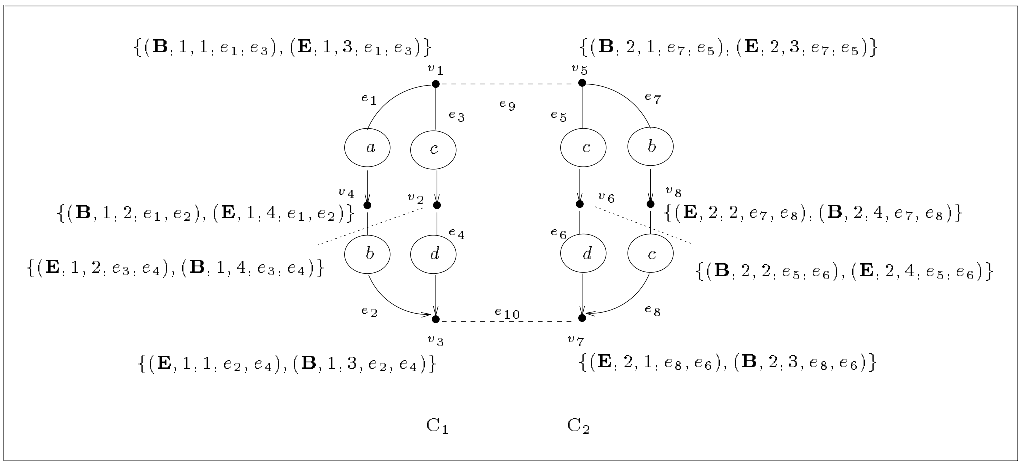

Figure 17. Example 4.5: Cycles C_1, C_2 and M-bridge (e_9, e_{10})

Let π_3 denote path $\mathrm{C}_1 \backslash \pi_1$, and π_4 denote path $\mathrm{C}_2 \backslash \pi_2$.

- Considering cycle C_1, we see that $\Delta(\pi_1^+) = c \cdot d, \Delta(\pi_3^+) = a \cdot b$ and $c \cdot d \not\succ a \cdot b$. Also $\Delta(\pi_1^-) = d^{-1} \cdot c^{-1}, \Delta(\pi_3^-) = b^{-1} \cdot a^{-1}$ and $d^{-1} \cdot c^{-1} \not\succ b^{-1} \cdot a^{-1}$, Analyzing labels $\eta(v_1), \eta(v_3), \eta(v_5)$ and $\eta(v_7)$, we do not discover any pair of rules R_{1_l} and R_{2_m} satisfying second condition in the premise of **Patch**.
- Considering cycle C_1, we obtain that $\Delta(\pi_2^+) = c \cdot d, \Delta(\pi_4^+) = b \cdot c$ and $c \cdot d \not\succ b \cdot c$. In addition, $\Delta(\pi_2^-) = d^{-1} \cdot c^{-1}, \Delta(\pi_4^-) = c^{-1} \cdot b^{-1}$ and $d^{-1} \cdot c^{-1} \not\succ c^{-1} \cdot b^{-1}$. Analyzing labels of vertices v_1, v_3, v_5 and v_7, we see that there is no rules R_{1_l} and R_{2_m}, satisfying any condition of **Patch** premise.

Our conclusion is that

Inference **Patch** can not be applied.

4.5. **Eliminate** Inference Rule

Inference rule **Eliminate** encodes the cancellation and the simplification by group axioms $x \cdot x^{-1} \to \varepsilon$ and $x^{-1} \cdot x \to \varepsilon$ of rewriting rules represented by a cycle in a PATCH Graph.

Eliminate

Let C_n be a non-initial cycle, and let (e'_1, e'_2) be the C-bridge between cycles C_i and C_j, defining cycle C_n. Let labels of edges e'_1 and e'_2 be $\lambda(e'_1) = (\varepsilon, n, i, j)$ and $\lambda(e'_2) = (\varepsilon, n, j, i)$. Let

$$\pi = e_1^{\alpha_1}.e_2^{\alpha_2}. \ \ldots \ .e_{k-1}^{\alpha_{k-1}}.e_k^{\alpha_k},$$

$\alpha_i \in \{+, -\}, 1 \leq i \leq k$, be a subpath of C_n satisfying the following properties:

(1) edges e_1 and e_k are labelled by the same $g \in G^+$, and edges $e_2, \ldots, e_{k-1}$ are labelled by ε,

(2) $\alpha_k = -(\alpha_1)$, i.e. either $\alpha_1 = +$ and $\alpha_k = -$, or $\alpha_1 = -$ and $\alpha_k = +$.

As will be shown later, we can suppose that both of edges e'_1 and e'_2 belong to π. Let l and m be numbers such that $1 < l < m < k$ and $e'_1 = e_l, e'_2 = e_m$.

Inference rule **Eliminate** consists in removing cycle C_n from the graph, and adding a new cycle $\mathrm{C}_{n'}$, where n' is a fresh number. Precisely, we delete C-bridge (e'_1, e'_2) between cycles C_i and C_j, defining C_n, and add a new C-bridge (e'_3, e'_4) between C_i and C_j, defining $\mathrm{C}_{n'}$. Let $e_i^{\alpha_i} = (v_i, v_{i+1}), 1 \leq i \leq k$. Then $e'_3 = (v_1, v_{l+1})$. Initial vertex of edge e'_3 is the same as initial (when $\alpha_1 = +$) or final (when $\alpha_1 = -$) vertex of edge e_1, and final vertex of edge e'_3 is the same as final vertex of edge e'_1. Edge $e'_4 = (v_m, v_{k+1})$ goes from vertex v_m, which is initial vertex of edge e'_2, to vertex v_{k+1}, which is initial (when $\alpha_1 = +$) or final (when $\alpha_1 = -$) vertex of edge e_k. Labels of new edges are: $\lambda(e'_3) = (\varepsilon, n', i, j)$ and $\lambda(e'_4) = (\varepsilon, n', j, i)$. New cycle is

$$\mathrm{C}_{n'} : (\mathrm{C}_n \backslash \pi)^+.(e'_4)^-.e_{l+1}^-. \ \ldots \ .e_{m-1}^-.(e'_3)^-.$$

LEMMA 4.3. *Let (e'_1, e'_2) be the C-bridge, relative to common paths π_1 and π_2, defining cycle C_n. Let $\lambda(e'_1) = (\varepsilon, n, i, j)$ and $\lambda(e'_2) = (\varepsilon, n, j, i)$. Let edge e'_1 belong to path π, and edge e'_2 belong to path $C_n \backslash \pi$. Then paths π_1 and π_2 are not maximal common paths for cycles C_i and C_j.*

Proof. Let path $\pi_3 \subseteq \mathrm{C}_i$ and $\pi_4 \subseteq \mathrm{C}_j$ be paths such that $\pi = \pi_3^+.(e'_1)^+.\pi_4^+$. Since $\pi_1 \subset \mathrm{C}_i$ and $\pi_2 \subset \mathrm{C}_j$ are common paths, last edge of path π_3 is adjacent to one of ends of path π_1, say, to its initial vertex, and first edge of path π_4 is adjacent to initial vertex of path π_2 (see Figure 18).

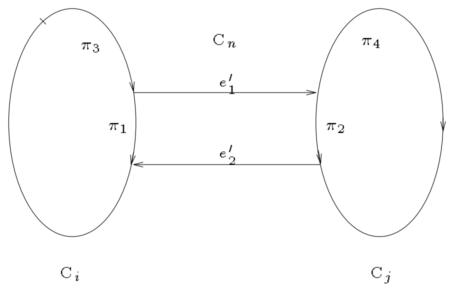

Figure 18. Illustration of Lemma 4.3

Moreover, $\Delta(\pi_3^+) = \Delta(\pi_4^-) = g$ or g^{-1}, depending on the values of α_1 and α_k. Then π_1 is the subpath of path $\pi_3^+.\pi_1^+$ and π_2 is the subpath of path $\pi_4^-.\pi_2^+$, such that $\Delta(\pi_3^+.\pi_1^+) = \Delta(\pi_4^-.\pi_2^+)$.Consequently, common paths π_1 and π_2 are not maximal. □

We perform **Eliminate** on each new cycle C_n as it is produced by application of **Patch** inference to cycles C_i and C_j. Then we suppose that neither C_i, nor C_j has a subpath satisfying properties (1) and (2). Since **Patch** inference is applied only to maximal common paths, our supposition in the premise of **Eliminate** inference, that both of edges e'_1 and e'_2 belong to π, is true.

We remark that if, as in Lemma 4.3, $\pi_1 \subseteq C_i$ and $\pi_2 \subseteq C_j$ are common paths, relative to C-bridge (e'_1, e'_2), then new C-bridge (e'_3, e'_4) is relative to common paths $e_1^{\alpha_1}. \ldots .e_{l-1}^{\alpha_{l-1}}.\pi_1.e_{m+1}^{\alpha_{m+1}} \ldots .e_k^{\alpha_k} \subset C_i$ and π_2, conjugate by string g (or g^{-1}, according to the values of α_1 and α_k).

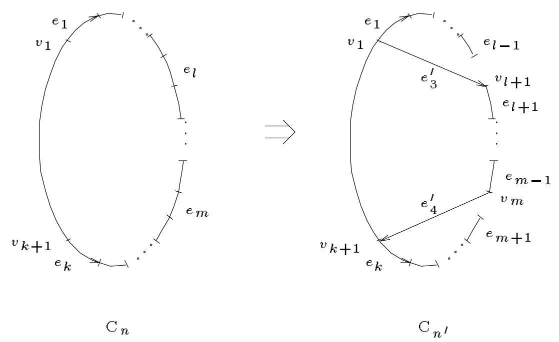

Figure 19. Eliminate inference rule

Figure 19 illustrates inference rule **Eliminate** when $\alpha_1 = +$ and $\alpha_k = -$. Case $\alpha_1 = -$ and $\alpha_k = +$ is similar.

EXAMPLE 4.6 (Eliminate). Consider cycle

$$C_3 : e_3^+.e_9^+.e_{10}^+.e_6^-.e_2^-.e_1^-,$$

shown in Figure 20-a. Cycle C_3 is defined by the C-bridge (e_9, e_{10}). Edge e_9 goes from vertex v_6 to vertex v_7, and edge e_{10} goes from vertex v_7 to vertex v_4. Labels of edges e_9 and e_{10} are $\lambda(e_9) = (\varepsilon, 3, 1, 2)$ and $\lambda(e_{10}) = (\varepsilon, 3, 2, 1)$.

Cycle C_3 is produced by application of **Patch** inference to following cycles:

- cycle $C_1 : e_1^-.e_3^+.e_4^+.e_5^+.e_6^-.e_2^-$, and
- cycle $C_2 : e_7^+.e_8^+$.

Since edge e_3^+ and edge e_6^+ are labelled by a, path $\pi = e_3^+.e_9^+.e_{10}^+.e_6^-$ satisfies conditions of the premise of inference rule **Eliminate**. Applying the inference, we remove edges e_9 and e_{10}, and

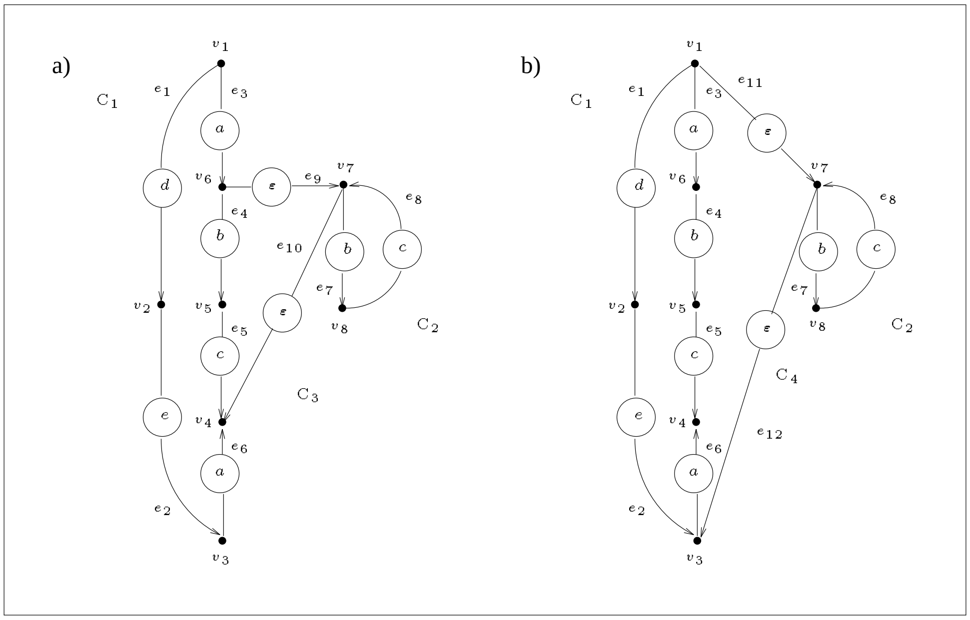

Figure 20. Example 4.6: a) Cycles C_1, C_2 and C_3. b) Cycles C_1, C_2 and C_4

add new C-bridge (e_{11}, e_{12}). Edge e_{11} goes from vertex v_1, which is initial vertex of edge e_3, to vertex v_7. Edge e_{12} goes from vertex v_7 to vertex v_3, which is initial vertex of edge e_6 (see Figure 20-b). Labels of new edges are: $\lambda(e_{11}) = (\varepsilon, 4, 1, 2)$ and $\lambda(e_{12}) = (\varepsilon, 4, 2, 1)$. New cycle C_4 is

$$C_4 : e_{11}^+.e_{12}^+.e_2^-.e_1^- .$$

Take cut (π_1, π_2) of cycle C_3, where $\pi_1 = e_1^-.e_3^+.e_9^+.e_{10}^+.e_6^-.e_2^-$ and π_2 is the empty path. Rule $\Delta(C_3, (\pi_1, \pi_2))$ is $d \cdot a^{-1} \cdot a \cdot e^{-1} \to \varepsilon$. Take cut (π_3, π_4) of cycle C_3, with $\pi_3 = e_1^-.e_{11}^+.e_{12}^+.e_2^-$ and π_4 the empty path. Rule $\Delta(C_4, (\pi_3, \pi_4))$ is $d \cdot e^{-1} \to \varepsilon$. We remark that rule $\Delta(C_4, (\pi_3, \pi_4))$ coincides with rule $\Delta(C_3, (\pi_1, \pi_2))$, which lhs is simplified by instances $a^{-1} \cdot a \to \varepsilon$ and $\varepsilon \cdot e^{-1} \to e^{-1}$ of group axioms.

Also, consider cut (π_5, π_6) of C_3 and cut (π_7, π_8) of C_4, such that $\pi_5 = e_1^-.e_3^+.e_9^+$, $\pi_6 = e_2^+.e_6^+.e_{10}^-$, $\pi_7 = e_1^-.e_{11}^+$ and $\pi_8 = e_2^+.e_{12}^-$. We see that rule $\Delta(C_4, (\pi_7, \pi_8)) :\ d^{-1} \to e$ coincides with rule $\Delta(C_3, (\pi_5, \pi_6)) :\ d^{-1} \cdot a \to e \cdot a$, cancelled on the right.

5. Completeness and Soundness of the Inference Rules System

In this section we prove the system of inference rules **Patch**, **Symmetrize** and **Eliminate** for completion modulo group theory to be sound and complete.

Remark that the inference rules preserve PATCH Graph structure. In other words, each of inference rules **Patch**, **Symmetrize** and **Eliminate**, applied to a PATCH Graph relative to a set of generators G, produces another PATCH Graph relative to the set G.

THEOREM 5.1. *Inference rules* **Patch**, **Symmetrize** *and* **Eliminate** *for finitely presented group completion are sound, i.e. they do not change the equational theory defined by rewriting rules and equations represented by a PATCH Graph* Γ *and group axioms.*

Proof. For each inference rule Inf, let us prove that if the graph $\Gamma' = Inf(\Gamma)$ is the result of inference rule Inf application, then congruence relation $\overset{*}{\leftrightarrow}_{\{\Delta(\Gamma'),\mathcal{A}\}}$ is the same that congruence relation $\overset{*}{\leftrightarrow}_{\{\Delta(\Gamma),\mathcal{A}\}}$.

– **Patch** inference rule. Graph Γ' is $\Gamma \cup \mathrm{C}_n$, where C_n is a new cycle. Let cycles C_i and C_j be ancestors of C_n. Let (e_1, e_2) be the C-bridge, which defines cycle C_n, and let $\pi_1 \subseteq \mathrm{C}_i$ and $\pi_2 \subseteq \mathrm{C}_j$ denote relative maximal common paths. Let edge e_1 link initial vertices of paths π_1 and π_2, and edge e_2 link their final vertices. Let label $\lambda(e_1)$ be (ε, n, i, j) and label $\lambda(e_2)$ be (ε, n, j, i). Let π_3 denote the path $\mathrm{C}_i \backslash \pi_1$ and π_4 denote the path $\mathrm{C}_j \backslash \pi_2$.

  Graph Γ' represents the system of rewriting rules and equations $\Delta(\Gamma') = \{\Delta(\Gamma), \Delta(\mathrm{C}_n)\}$. Let new cycle C_n be $\mathrm{C}_n : e_1^+.\pi_4^+.e_2^+.\pi_3^-$.

  Let $\Delta(\pi_1) = \Delta(\pi_2) = u$, $\Delta(\pi_3) = w$ and $\Delta(\pi_4) = v$. Then $\Delta(\mathrm{C}_n)$ is the equation $v \cdot w^{-1} = \varepsilon$. Since cycles C_i and C_j are in Γ, we deduce $u \overset{*}{\leftrightarrow}_{\{\Delta(\Gamma),\mathcal{A}\}} w$ and $u \overset{*}{\leftrightarrow}_{\{\Delta(\Gamma),\mathcal{A}\}} v$. Then $v \cdot w^{-1} \overset{*}{\leftrightarrow}_{\{\Delta(\Gamma),\mathcal{A}\}} u \cdot u^{-1} \overset{*}{\rightarrow}_{\mathcal{A}} \varepsilon$.

– **Symmetrize** inference rule. Let $\mathrm{C}_n : e_1^{\alpha_1}. \ \ldots \ .e_k^{\alpha_k}$ be a new cycle to which the inference rule is applied. Since C_n is a new cycle, its vertex labels contain no markers $(\mathbf{B}, n, *, *)$ and $(\mathbf{E}, n, *, *)$. Let $\Delta(e_i^{\alpha_i}) = s_i$. Then, before **Symmetrization** rule application, $\Delta(\mathrm{C}_n)$ is the equation $s_1 \cdot \ldots \cdot s_k = \varepsilon$.

  Let $\{(\pi_1, \pi_1'), \ldots, (\pi_k, \pi_k')\}$ be the set of cuts of C, such that the set

  $$\{R_{(n,i)}\}_{i=1}^k = \Delta(\mathrm{C}_n, \{(\pi_1, \pi_1'), \ldots, (\pi_k, \pi_k')\})$$

  is the set of symmetrized forms of $\Delta(\mathrm{C}_n)$. If graph Γ is the union of cycles $\mathrm{C}_1, \ldots, \mathrm{C}_n$, such that cycle C_n have no beginning markers $(\mathbf{B}, n, *, *)$ and end markers $(\mathbf{E}, n, *, *)$, then graph Γ' is the union of cycles $\mathrm{C}_1, \ldots, \mathrm{C}_n$, such that the labels of vertices of C_n contain such markers. Consequently, the set of rewriting rules $\Delta(\Gamma')$ is equal to

  $$\{ \ (\Delta(\Gamma) \backslash s_1 \cdot \ldots \cdot s_k = \varepsilon) \ , \ \{R_{(n,i)}\}_{i=1}^k \ \}.$$

  The equational theory defined by $\{\Delta(\Gamma'), \mathcal{A}\}$ is the same that the theory defined by $\{\Delta(\Gamma), \mathcal{A}\}$ if

  - for any i, the lhs of the rule $R_{(n,i)}$ is congruent to its rhs in the theory defined by $\{\Delta(\Gamma), \mathcal{A}\}$, and

- the string $s_1 \cdot \ldots \cdot s_k$ is congruent to ε in the theory defined by $\{\Delta(\Gamma'), \mathcal{A}\}$.

Actually, each rule $R_{(n,i)}$ has the form

$$s_{i+1} \cdot \ldots \cdot s_j \to s_i^{-1} \cdot \ldots \cdot s_1^{-1} \cdot s_k^{-1} \cdot \ldots \cdot s_{j+1}^{-1} \text{ or } s_i^{-1} \cdot \ldots \cdot s_1^{-1} \cdot s_k^{-1} \cdot \ldots \cdot s_{j+1}^{-1} \to s_{i+1} \cdot \ldots \cdot s_j$$

for some i, j such that $1 \le i \le j \le k$. When $i = j$, string $s_{i+1} \cdot \ldots \cdot s_j$ is empty.
Then, firstly, for each i

$$\begin{aligned} s_{i+1} \cdot \ldots \cdot s_j &\overset{*}{\leftrightarrow}_{\mathcal{A}} s_i^{-1} \cdot \ldots \cdot s_1^{-1} \cdot (s_1 \cdot \ldots \cdot s_k) \cdot s_k^{-1} \cdot \ldots \cdot s_{j+1}^{-1} \\ &\overset{*}{\leftrightarrow}_{\Delta(\Gamma)} s_i^{-1} \cdot \ldots \cdot s_1^{-1} \cdot \varepsilon \cdot s_k^{-1} \cdot \ldots \cdot s_{j+1}^{-1} \\ &\to_{\mathcal{A}} s_i^{-1} \cdot \ldots \cdot s_1^{-1} \cdot s_k^{-1} \cdot \ldots \cdot s_{j+1}^{-1}. \end{aligned}$$

Secondly, string

$$s_1 \cdot \ldots \cdot s_k \leftrightarrow_{R_{(n,i)}} s_1 \cdot \ldots \cdot s_i \cdot (s_i^{-1} \cdot \ldots \cdot s_1^{-1} \cdot s_k^{-1} \cdot \ldots \cdot s_{j+1}^{-1}) \cdot s_{j+1} \cdot \ldots \cdot s_k \overset{*}{\to}_{\mathcal{A}} \varepsilon.$$

– **Eliminate** inference rule. Let C_n be the new cycle to which we apply the inference rule. Let $\pi = e_1^{\alpha_1}.e_2^{\alpha_2}. \ \ldots \ e_k^{-\alpha_1}$ be the path satisfying the premises of the inference rule. Denote by π' the path $\mathrm{C}_n \backslash \pi$. Let $\Delta(\pi') = s$.

  Cycle C_n is a new cycle, then its vertex labels contain no beginning and end markers $(\mathbf{B}, n, *, *)$ and $(\mathbf{E}, n, *, *)$. Cycle $\mathrm{C}_n : \pi^+.(\pi')^-$ represents the equation $g^{\alpha_1} \cdot g^{-\alpha_1} \cdot s^{-1} = \varepsilon$. By application of **Eliminate** inference rule, cycle C_n is removed and a new cycle $\mathrm{C}_{n'}$: $(\pi')^+.(e'_4)^-.e_{l+1}^-. \ \ldots \ .e_{m-1}^-.(e'_3)^-$ is created (see Figure 19). Cycle $\mathrm{C}_{n'}$ is defined by a new C-bridge (e'_3, e'_4). Both of edges e'_3 and e'_4 are labelled by ε. Consequently, equation $\Delta(\mathrm{C}_{n'})$ is $s = \varepsilon$.

  Graph Γ' is $\Gamma \backslash \mathrm{C}_n \cup \mathrm{C}_{n'}$. Actually,

  1. String s is congruent to ε in the theory defined by $\{\Delta(\Gamma), \mathcal{A}\}$:

  $$s \leftrightarrow_{\mathcal{A}} \varepsilon \cdot s \leftrightarrow_{\Delta(\Gamma)} (g^{\alpha_1} \cdot g^{-\alpha_1} \cdot s^{-1}) \cdot s \overset{*}{\to}_{\mathcal{A}} \varepsilon.$$

  2. String $g^{\alpha_1} \cdot g^{-\alpha_1} \cdot s^{-1}$ is congruent to ε in the theory defined by $\{\Delta(\Gamma'), \mathcal{A}\}$:

  $$g^{\alpha_1} \cdot g^{-\alpha_1} \cdot s^{-1} \to_{\mathcal{A}} \varepsilon \cdot s^{-1} \leftrightarrow_{\Delta(\Gamma')} s \cdot s^{-1} \to_{\mathcal{A}} \varepsilon.$$

$\square$

Let $\mathcal{R}$ be a string rewriting system. Let $>$ be a reduction ordering, compatible with system $\{\mathcal{R}, \mathcal{A}\}$. Let Γ be a PATCH Graph representing $\mathcal{R}$. Let be given a Matching Procedure, which outputs all matching bridges relative to maximal common paths for cycles of Γ.

The *Completion Procedure of PATCH Graph* Γ transforms graph Γ, applying as much as possible inference rules **Patch**, **Eliminate** and **Symmetrize**. New cycle are generated by **Patch** inference, possibly reduced by **Eliminate** inference, and their vertices are labelled by **Symmetrize** inference. Let Γ_0 be initial graph Γ, and for any natural j, graph $\Gamma_{j+1} = Inf(\Gamma_j)$ be the graph generated by application of an inference rule $Inf \in \{\textbf{Patch}, \textbf{Eliminate}, \textbf{Symmetrize}\}$ to graph Γ_j. Graph $\Gamma^\infty = \bigcup_{i\geq 0}\bigcap_{j>i}\Gamma_j$ is called *persisting graph*. Graph Γ^∞ is saturated with respect to inference rules **Patch**, **Eliminate** and **Symmetrize**, i.e. no inferences can be applied to Γ^∞. Denote by $\mathcal{R}^\infty$ rewriting system $\Delta(\Gamma^\infty)$, represented by persisting graph.

THEOREM 5.2. *The system of inference rules* **Patch**, **Symmetrize** *and* **Eliminate** *for finitely presented group completion is complete, i.e. rewriting system* $\{\mathcal{R}^\infty, \mathcal{A}\}$ *is canonical.*

We need following two lemmas:

LEMMA 5.1. *Let C_1 and C_2 be cycles in a PATCH Graph.*
Let (π_1, π_2) be a cut of cycle C_1 and (π_3, π_4) be a cut of cycle C_2, such that rewriting rule $R_2 = \Delta(C_2, (\pi_3, \pi_4))$ from set $Sym(R_2)$ simplify rule $R_1 = \Delta(C_1, (\pi_1, \pi_2))$ from set $Sym(R_1)$. Let path $\pi_1 = \pi_5.\pi_6.\pi_7$ or path $\pi_2 = \pi_5.\pi_6.\pi_7$, such that paths π_6 and π_3 are maximal common paths for cycles C_1 and C_2. Let C_3 be new cycle produced by **Patch** *inference rule application to the M-bridge relative to paths π_6 and π_3.*

Then for all strings s and t such that $s \rightarrow_{R_1} t$, there is a rewriting rule $R_3 \in Sym(\Delta(C_3))$ such that $s \xrightarrow{*}_{\{R_2,R_3,\mathcal{A}\}} t$.

Proof of LEMMA 5.1. We prove the lemma when $\pi_1 = \pi_5.\pi_6.\pi_7$. The case $\pi_2 = \pi_5.\pi_6.\pi_7$ is analogous.

Let $R_1 : \ l_1 \rightarrow r_1$ and $R_2 : \ l_2 \rightarrow r_2$. Since path $\pi_1 = \pi_5.\pi_6.\pi_7$, string $l_1 = u_1 \cdot l_2 \cdot w_1$, where $u_1 = \Delta(\pi_5)$ and $w_1 = \Delta(\pi_7)$.

Let (e_1, e_2) be the C-bridge which defines cycle C_3 (see Figure 21).

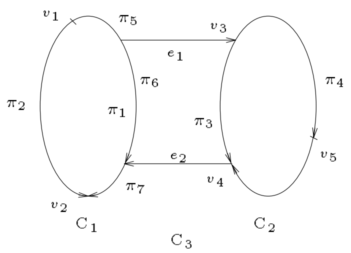

Figure 21. Illustration of Lemma 5.1

Cycle C_3 is $\mathrm{C}_3 : \pi_5^+.e_1^+.\pi_4^+.e_2^+.\pi_7^+.\pi_2^-$, and equation $\Delta(\mathrm{C}_3)$ is $E : \ u_1 \cdot r_2 \cdot w_1 \cdot r_1^{-1} = \varepsilon$.

Denote by v_1 initial vertex of paths π_1 and π_2 and by v_2 their final vertex. Also, denote by v_3 initial vertex of paths π_3 and π_4 and by v_4 final vertex of paths π_3 and π_4. Take rule R_3 in set $Sym(E)$ such that $R_3 = \Delta(\mathrm{C}_3, (\pi_8, \pi_9))$ for some cut (π_8, π_9) of cycle C_3 such that initial vertex of paths π_8 and π_9 coincides with v_1, and the first edge of path π_8 is the same that the first edge of path π_1 (such a rule R_3 exists, according to Proposition 4.1). Let v_5 be final vertex of paths π_8 and π_9. Thereafter the proof differs following to the place of vertex v_5 in cycle C_3. Let us suppose that $v_5 \in \pi_4$ (other cases are similar, we do not detail them). Denote by π_{10} the prefix of path π_4, going from vertex v_3 to vertex v_5, and by π_{11} the suffix of path π_4, going from vertex v_5 to vertex v_4. Let $\Delta(\pi_{10}) = u_3$ and $\Delta(\pi_{10}) = w_3$. Then string $r_2 = u_3 \cdot w_3$ and rule R_3 is $R_3: \ u_1 \cdot u_3 \to r_1 \cdot w_1^{-1} \cdot w_3^{-1}$.

Let strings s and t be such that $s \to_{R_1} t$. Then there are strings u_2 and w_2 such that $s = u_2 \cdot l_1 \cdot w_2$ and $t = u_2 \cdot r_1 \cdot w_2$.

$$\begin{aligned} s &= u_2 \cdot l_1 \cdot w_2 = u_2 \cdot u_1 \cdot l_2 \cdot w_1 \cdot w_2 \\ &\to_{R_2} u_2 \cdot u_1 \cdot r_2 \cdot w_1 \cdot w_2 = u_2 \cdot u_1 \cdot u_3 \cdot w_3 \cdot w_1 \cdot w_2 \\ &\to_{R_3} u_2 \cdot r_1 \cdot w_1^{-1} \cdot w_3^{-1} \cdot w_3 \cdot w_1 \cdot w_2 \\ &\xrightarrow{*}_{\mathcal{A}} u_2 \cdot r_1 \cdot w_2 = t. \end{aligned}$$

□

LEMMA 5.2. *Let C_1 be a cycle in a PATCH Graph.*
Let (π_1, π_2) be a cut of cycle C_1 such that rewriting rule $R_1 = \Delta(C_1, (\pi_1, \pi_2))$ belong to set $Sym(R_1)$. Let $\pi \subseteq C_1$ be a path satisfying the premises of the **Eliminate** *inference rule. Denote by C_2 the cycle resulting from application of rule* **Eliminate** *to cycle C_1.*

Then for all strings s and t such that $s \to_{R_1} t$, there is a rewriting rule $R_2 \in Sym(\Delta(C_2))$ such that $s \xrightarrow{*}_{\{R_2, \mathcal{A}\}} t$.

Proof of LEMMA 5.2. Let $\pi = e_1^{\alpha_1}.e_2^{\alpha_2}. \ \ldots. \ e_k^{-\alpha_1}$. Two situations are possible: either edge e_1 belongs to path π_1 and edge e_2 belongs to path π_2, or both of edges e_1 and e_2 belong to path π_1 or to path π_2. Let us consider the first case. The second case is similar, we do not detail it to avoid introducing a lot of new notations.

Let $R_1: \ l_1 \to r_1$. Denote by v_1 initial vertex of paths π_1 and π_2 and by v_2 their final vertex. Let v_3 be initial vertex of edge e_1, and v_4 be final vertex of edge e_k. Denote by π_3 the prefix of path π_1 going from vertex v_1 to vertex v_3, and by π_4 the prefix of path π_2 going from vertex v_1 to vertex v_4. Then string $l_1 = p \cdot g^{\alpha_1}$ and string $r_1 = q \cdot g^{\alpha_1}$, where $p = \Delta(\pi_3)$ and $q = \Delta(\pi_4)$.

New cycle C_2 is $\mathrm{C}_2 : (\mathrm{C}_1 \backslash \pi)^+.(e_4')^-.e_{l+1}^-. \ \ldots \ .e_{m-1}^-.(e_3')^-$, where (e_3', e_4') is a new C-bridge (see Figure 19). Take rewriting rule R_2 in set $Sym(\Delta(\mathrm{C}_2))$ such that $R_2 = \Delta(\mathrm{C}_2, (\pi_5, \pi_6))$ for some cut (π_5, π_6) of cycle C_2 such that v_1 is initial vertex of paths π_5 and π_6, and the first edge of path π_5 is the same that the first edge of path π_1 (such a rule R_2 exists by Proposition 4.1). Let v_5 be final vertex of paths π_5 and π_6. Thereafter the proof differs following to the place of vertex v_5 in cycle C_2. We explain the case $v_5 \in (\mathrm{C}_1 \backslash \pi)$ (see Figure 22). The proofs in other cases are similar.

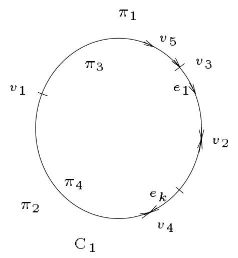

Figure 22. Illustration of Lemma 5.2

Let $R_2: \ l_2 \to r_2$. Let path π_7 be the suffix of path π_1 going from vertex v_5 to vertex v_3. Let $\Delta(\pi_7) = s$. Then $p = l_2 \cdot s$ and $r_2 = q \cdot s^{-1}$.

Let strings s and t be such that $s \to_{R_1} t$. Then there are strings u and w such that $s = u \cdot l_1 \cdot w$ and $t = u \cdot r_1 \cdot w$.

$$\begin{aligned} s &= u \cdot l_1 \cdot w = u \cdot p \cdot g^{\alpha_1} \cdot w = u \cdot l_2 \cdot s \cdot g^{\alpha_1} \cdot w \\ &\to_{R_2} u \cdot r_2 \cdot s \cdot g^{\alpha_1} \cdot w = u \cdot q \cdot s^{-1} \cdot s \cdot g^{\alpha_1} \cdot w \\ &\to_{\mathcal{A}} u \cdot q \cdot g^{\alpha_1} \cdot w = u \cdot r_1 \cdot w = t. \end{aligned}$$

□

Proof of THEOREM 5.2. We remark that rewriting system $\{\mathcal{R}^\infty, \mathcal{A}\}$ is noetherian. Since, for every cycle of PATCH graph, beginning and end markers of its vertices are computed with respect to reduction ordering $>$, and ordering $>$ is compatible with initial rewriting system $\mathcal{R}$ and group axioms $\mathcal{A}$, there is no infinite sequence of reductions by rewriting rules from system $\{\mathcal{R}^\infty, \mathcal{A}\}$.

Let us prove that any critical pair between two rules in system $\{\mathcal{R}^\infty, \mathcal{A}\}$ is joinable with respect to $\{\mathcal{R}^\infty, \mathcal{A}\}$. Let system $\mathcal{R}^j = \Delta(\Gamma_j), \ j \geq 0$.

Since only **Patch** inference rule (in the case of a simplification) or **Eliminate** inference rule application can remove rewriting rules from set $\mathcal{R}^j$, we conclude, summarizing the results of Lemmas 5.1 and 5.2, that if a critical pair is joinable with respect to system $\{\mathcal{R}^J, \mathcal{A}\}$ for some J, then it is still joinable with respect to system $\{\mathcal{R}^{J+1}, \mathcal{A}\}$. By induction, we deduce that if a critical pair is joinable with respect to system $\{\mathcal{R}^J, \mathcal{A}\}$ for some J, then it is joinable with respect to system $\{\mathcal{R}^\infty, \mathcal{A}\}$. Let us find such a number J for any critical pair.

System $\mathcal{A}$ of group axioms is canonical, consequently, only two kinds of critical pairs can occur:

1. critical pairs between a group axiom and a ground rule, and

2. critical pairs between two ground rules.

Case 1. Let rewriting rule $R: \ a_1 \cdot \ldots \cdot a_l \to b_1 \cdot \ldots \cdot b_m$ be in system $\mathcal{R}^\infty$. Let j be the minimal number such that $R \in \mathcal{R}^j$. According to [Le Chenadec1986], each critical pair between a group axiom and rule R is a normal pair.

Let C_1 be the cycle in graph Γ_j representing rule R. Let $J > j$ be the minimal number such that in graph Γ_J all the beginning and end markers corresponding to rules from set $Sym(R)$ are put along cycle C_1 by **Symmetrize** inference rule application. By Lemma 2.1, each normal pair of rule R is joinable with respect to system $\{Sym(R), \mathcal{A}\} \subseteq \{\mathcal{R}^J, \mathcal{A}\}$.

Case 2. Let $R_1: \ u \to v$ and $R_2: \ l \to r$ be two rewriting rules from system $\mathcal{R}^\infty$ giving a critical pair. Let cycle C_1 represent rule R_1 and cycle C_2 represent rule R_2. Let i and j be minimal numbers such that $\mathrm{C}_1 \in \Gamma_i$ and $\mathrm{C}_2 \in \Gamma_j$.

A critical pair between R_1 and R_2 has the form $(v \cdot s, p \cdot r)$ where $u = p \cdot c$ and $l = c \cdot s$ for some strings p, c and s. Let (π_1, π_2) be a cut of cycle C_1 such that $\Delta(\mathrm{C}_1, (\pi_1, \pi_2)) = R_1$, and let (π_3, π_4) be a cut of cycle C_2 such that $\Delta(\mathrm{C}_2, (\pi_3, \pi_4)) = R_2$. Let vertex $v_1 \in \mathrm{C}_1$ be initial vertex of paths π_1 and π_2, and let vertex $v_2 \in \mathrm{C}_1$ be their final vertex. Also, let $v_3 \in \mathrm{C}_2$ be initial vertex of paths π_3 and π_4, and let vertex $v_4 \in \mathrm{C}_2$ be final vertex of π_3 and π_4. Remark that $\Delta(\pi_1) = u$,

$\Delta(\pi_2) = v$, $\Delta(\pi_3) = l$ and $\Delta(\pi_4) = r$. Let path π_5 be the prefix of path π_1 such that $\Delta(\pi_5) = p$, and let path π_6 be the suffix of path π_1 such that $\Delta(\pi_6) = c$. Also, let path π_7 be the prefix of path π_3 such that $\Delta(\pi_7) = c$, and let path π_8 be the suffix of path π_3 such that $\Delta(\pi_8) = s$. Actually, $\pi_1 = \pi_5.\pi_6$ and $\pi_3 = \pi_7.\pi_8$. There are two possibilities:

2-a There is an M-bridge (e_1, e_2) in graph Γ^∞ (generated by the Matching Procedure), relative to paths π_6 and π_7, identified $(n, 1, 2)$ (or $(n, 2, 1)$) for some n.

2-b There is no M-bridge relative to paths π_6 and π_7, identified $(n, 1, 2)$ or $(n, 2, 1)$, in graph Γ^∞.

Case 2-a. Paths π_6 and π_7 satisfy the second condition in **Patch** inference rules premises. Let $k > max(i, j)$ be the minimal number, such that graph Γ_k contains cycle C_n, which results from **Patch** inference rule application to M-bridge (e_1, e_2).

Let (e_3, e_4) be the C-bridge which defines cycle C_n, such that $\lambda(e_3) = (\varepsilon, n, 1, 2)$ and $\lambda(e_4) = (\varepsilon, n, 2, 1)$. Cycle C_n is $\mathrm{C}_n : e_3^+.\pi_4^+.\pi_8^-.e_4^+.\pi_2^-.\pi_5^+$.

Let J be a minimal number such that in graph Γ_J all the beginning and end markers corresponding to rules from set $Sym(\Delta(\mathrm{C}_n))$ are put along cycle C_n by **Symmetrize** inference rule application.

Let $R_3 \in Sym(\Delta(\mathrm{C}_n))$ be a rule, and (π_9, π_{10}) be a cut of cycle C_n such that $R_3 = \Delta(\mathrm{C}_n, (\pi_9, \pi_{10}))$, v_1 is initial vertex of paths π_9 and π_{10}, and the first edge of path π_9 is the same that the first edge of path π_1. Rule R_3 exists by Proposition 4.1. Obviously, rule $R_3 \in \mathcal{R}^J$.

Let v_5 be final vertex of paths π_9 and π_{10}. Thereafter the proof differs following to the place of vertex v_5 in cycle C_n. Let us suppose that $v_5 \in \pi_4$ (see Figure 23). Other cases are similar.

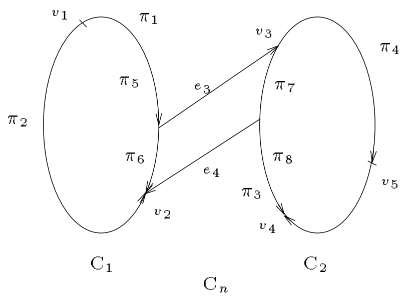

Figure 23. Illustration of Case 2-a

Denote by π_{11} the prefix of path π_4, going from vertex v_3 to vertex v_5, and by π_{12} the suffix of path π_4, going from vertex v_5 to vertex v_4. Let $\Delta(\pi_{11}) = t$ and $\Delta(\pi_{12}) = w$. Then $\pi_4 = \pi_{11}.\pi_{12}$, $r = t \cdot w$ and $R_3 :\; p \cdot t \to v \cdot s \cdot w^{-1}$.

$$p \cdot r = p \cdot t \cdot w \to_{R_3} v \cdot s \cdot w^{-1} \cdot w \to_{\mathcal{A}} v \cdot s.$$

Case 2-b. There is no M-bridge relative to common paths π_6 and π_7 for cycles C_1 and C_2. Then paths π_6 and π_7 are either trivial, or non-maximal common paths.

In last case, let $\pi_9 \subseteq \mathrm{C}_1$ and $\pi_{10} \subseteq \mathrm{C}_2$ be maximal common paths for cycles C_1 and C_2, such that $\pi_9 = \pi_{11}.\pi_6.\pi_{12}$ and $\pi_{10} = \pi_{13}.\pi_7.\pi_{14}$ for some paths $\pi_{11}, \pi_{12}, \pi_{13}$ and π_{14}, $\Delta(\pi_{11}) = \Delta(\pi_{13})$ and $\Delta(\pi_{12}) = \Delta(\pi_{14})$. Let C_m be the cycle resulting from **Patch** inference rule application to the M-bridge identified (m, i, j) (or (m, j, i)) for some m, relative to paths π_9 and π_{10}.

Let J be a minimal number such that in graph Γ_J all the beginning and end markers corresponding to rules from set $Sym(\Delta(\mathrm{C}_m))$ are put along cycle C_m by **Symmetrize** inference rule application. By Lemma 4.1, critical pair $(v \cdot s, p \cdot r)$ is joinable with respect to system $\{\mathcal{R}^J, \mathcal{A}\}$.

In former case, $\mathrm{C}_1 = \mathrm{C}_2 = \mathrm{C}$. Let J be a minimal number such that in graph Γ_J all the beginning and end markers corresponding to rules from set $Sym(\Delta(\mathrm{C}))$ are put along cycle C by **Symmetrize** inference rule application. By Lemma 4.2, critical pair $(v \cdot s, p \cdot r)$ is joinable with respect to system $\{\mathcal{R}^J, \mathcal{A}\}$.

$\square$

6. Matching

In this section we define the Matching Procedures, creating an M-bridge for each pair of maximal common paths π_1, π_2 for cycles C_i and C_j.

6.1. Matching Procedure for Initial Cycles

Procedure **MatchInit**

Input: Initial cycles $\mathrm{C}_1, \ldots, \mathrm{C}_n$

Output: Set E_M of M-edges forming the set of all M-bridges between maximal non-trivial common paths for cycles $\mathrm{C}_1, \ldots, \mathrm{C}_n$

`begin`

$E_M = \emptyset, N = n$
`for all` $i, j \in \{1, \ldots, n\}$ such that $i \leq j$
    `for all` ordered pairs (e_1, e_2) such that edge $e_1 \in \mathrm{C}_i$ and edge $e_2 \in \mathrm{C}_j$ `do`
        `if` $(\lambda(e_1) = (a, i)$ and $\lambda(e_2) = (a, j))$
        `then if` $(i = j$ and $e_1 = e_2)$ /**trivial common paths**/ `then continue`;
            `else do` $V_1 = InitialVertex(e_1);\ V_2 = InitialVertex(e_2)$;
                $V_3 = FinalVertex(e_1);\ V_2 = FinalVertex(e_2)$;
                $v_1 = V_1;\ v_2 = V_2;\ v_3 = V_3;\ v_4 = V_4$;
                $\pi_1 = e_1;\ \pi_2 = e_2$;
                `while` (edge $e_3 \in \mathrm{C}_i$ adjacent to v_1 and edge $e_4 \in \mathrm{C}_j$ adjacent to v_2 are

```
        such that λ(e_3) = (b, i) and λ(e_4) = (b, j))
do v_1 = OtherEnd(e_3, v_1);
   v_2 = OtherEnd(e_4, v_2);
   if (v_1 = V_3 or v_2 = V_4) /*whole cycle*/ then break;
while (edge e_5 ∈ C_i adjacent to v_3 and edge e_6 ∈ C_j adjacent to v_4 are
        such that λ(e_5) = (c, i) and λ(e_6) = (c, j))
do v_3 = OtherEnd(e_5, v_3);
   v_4 = OtherEnd(e_6, v_4);
   if (v_3 = v_1 or v_4 = v_2) /*whole cycle*/ then break;
N = N + 1;
e_7 = (v_1, v_2) and e_8 = (v_3, v_4) M-edges;
λ(e_7) = (N, i, j); λ(e_8) = (N, i, j);
E_M = Add(e_7, E_M); Add(e_8, E_M);

end
```

For any pair C_i and C_j of initial cycles, we look for a pair of edges e_1 and e_2 labelled by the same generator. If such edges are found, we put $\pi_1 = e_1, \pi_2 = e_2$ and grow up common paths π_1 and π_2, until we obtain maximal common paths.

Procedure **MatchInit** uses the following functions:

– functions $InitialVertex(e)$ and $FinalVertex(e)$, which give respectively initial and final vertex of an edge e,

– function $OtherEnd(e, v)$, where vertex v is one of ends of edge e, which returns the other end of edge e, and

– function $Add(e, E_M)$, which adds edge e to set E_M of matching edges.

6.2. Matching Procedure Involving a New Cycle

Adding an M-bridge between a new cycle and another cycle means either creating a pair of new M-edges, or adding new triples to labels of a pair of already existing M-edges.

Lemma 6.1 shows that a maximal common path between a new cycle and an old (i.e. already existing in the graph) cycle is a combination of maximal common paths for each of the ancestors of the new cycle and the old cycle. Lemma 6.1 gives the sufficient condition for the addition of a new M-edge.

LEMMA 6.1. *Let C_i, C_j and C_k be old cycles, and let C_n be a new cycle, such that cycles C_i and C_j are its ancestors. Let (e_1, e_2) be the C-bridge between C_i and C_j which defines C_n. Let $\pi_1 \subset C_i$ and $\pi_2 \subset C_j$ be maximal common paths relative to C-bridge (e_1, e_2). Let $\pi_3 \subset C_n$ and $\pi_4 \subset C_k$ be a pair of maximal common paths for cycle C_n and cycle C_k. Then*

1. *Each of paths π_3 and π_4 is a combination of some maximal common paths for cycles C_i and C_k, and of some maximal common paths for cycles C_j and C_k.*

2. Let (e_3, e_4) be the M-bridge relative to maximal common paths π_3 and π_4. Then e_3 or e_4 is a new M-edge in the graph only if corresponding maximal common paths for cycles C_j and C_k have a common edge with path π_1 or π_2.

The proof of Lemma 6.1 is easy, but needs a long analysis of different situations depending on the place of initial and final vertices of path π_3. We do not include it in the present version of the paper.

We suppose that **MatchNew** is a matching procedure which computes all M-bridges between a new cycle and old cycles in a PATCH Graph, as soon as the new cycle is created. We also suppose that old M-edges are reused, and new M-edges are added only when it is necessary.

7. Completion Algorithm Based on PATCH Graph Representation

7.1. Completion Algorithm

In this section we give an algorithm of completion based on the PATCH Graph structure. Using the Inference System described in Section 4 and the Matching Procedures **MatchInit** and **MatchNew** discussed in Section 6, we formulate the algorithm in a compact and simple way.

Let $\mathcal{R}_0 = \{R_1, \ldots, R_n\}$ be the initial string rewriting system. Let Γ_0 be the initial PATCH graph. Graph Γ_0 is the disjoint union $\bigcup_{i=1}^{n} \mathrm{C}_i$ of initial cycles $\mathrm{C}_1, \ldots, \mathrm{C}_n$. Each cycle C_i, $1 \leq i \leq n$ represents the rule R_i, cyclically reduced and simplified with respect to group axioms $x \cdot x^{-1} \to \varepsilon, x^{-1} \cdot x \to \varepsilon, \varepsilon \cdot x \to x$ and $x \cdot \varepsilon \to x$.

ALGORITHM 1 (**Completion Algorithm for Groups).**

Input : PATCH Graph Γ_0

Output : PATCH Graph Γ^{∞}

`begin`

$\Gamma = \Gamma_0$; $E_M = \emptyset$;
$E_M =$ **MatchInit**$(\mathrm{C}_1, \ldots, \mathrm{C}_n)$;
`for each` $i \in \{1, \ldots, n\}$ `do`
    $\mathrm{C}_i =$ **Symmetrize**(C_i);
$N =$ maximal number such that there is an M-bridge identified(N, i, j) for some i and j;
`while` (there is a non-empty label $\lambda(e)$ of an M-edge e) `do`
    Choose an M-bridge (e_1, e_2) identified (l, i, j) for some$l$, i and j;
    $\lambda(e_1) = \lambda(e_1) \backslash (l, i, j)$;
    $\lambda(e_2) = \lambda(e_2) \backslash (l, i, j)$;
    `if` (**Patch** is applicable) `then` $\mathrm{C}_l =$ **Patch**(e_1, e_2) a new cycle;

```
                    L = l;
                    while(Eliminate is applicable)
                          do C_{N+1} = Eliminate(C_N);
                             N = N + 1;
                             L = N + 1;
                    C_L = Symmetrize(C_L);
                    E_M = Add(MatchNew(C_L),E_M);
      else continue;
return(Γ^∞ = Γ);

end
```

Performing the algorithm, we first compute matching bridges for initial cycles in graph Γ. Second, we symmetrize each initial cycle.

Taking an M-bridge (e_1, e_2) identified (l, i, j), we verify the premises of the **Patch** inference rule, and if possible, apply it to the M-bridge, creating a new cycle C_l. We apply to C_l, while possible, the inference rule **Eliminate**. Each apllication of **Eliminate** creates a new cycle. Number L denotes the number of the last created cycle. When **Eliminate** is no longer applicable, we apply the inference rule **Symmetrize**. Finally, we compute all the M-bridges between the new cycle and all the cycles in current graph Γ, applying matching procedure **MatchNew**. According to the values of new M-bridges, we either add new identifications to the labels of some already existing M-edges, or create new M-edges. This is done by function *Add*.

We choose M-bridges in a fair way, taking them in the order they are produced by the matching procedures **MatchInit** and **MatchNew**, without ignoring any M-bridge. After M-bridge (e_1, e_2) is being processed, we remove the identification (l, i, j) from sets $\lambda(e_1)$ and $\lambda(e_2)$.

We can not evaluate the precise complexity of our algorithm, because the number of new cycles depends on the form of the given rewriting rules and on the given reduction ordering. The efficiency of the algorithm also relates to a strategy of computation of M-bridges.

We have started an implementation of the algorithm in C++ object-oriented language using LEDA Library [Naher1993]. We intend to include benchmarks in the final version of the paper.

8. Concluding Remarks

We have developed PATCH Graphs, a new data structure, representing rewriting systems for groups in a compact way. All the symmetrized forms of a rule or an equation are represented by a cycle in a PATCH Graph. Then each transformation performed on a cycle corresponds to a transformation of a class of rewriting rules by the standard completion procedure or the group completion procedure by Le Chenadec [Le Chenadec1986].

We have formulated inference rules **Symmetrize**, **Patch** and **Eliminate** for completion of finitely presented groups using PATCH Graphs. The inference rules can be efficiently performed on cycles of a PATCH Graph. **Symmetrize** inference rule gives the symmetrization procedure for groups. **Eliminate** inference rule expresses the simplification by group axioms and the cancellation of ground rules. **Patch** inference rule generalizes the critical pair and the simplification computations. Actually, inference rule **Patch** translates **Deduce**, **Simplify**, **Compose** and **Collapse** inference rules of the standard completion procedure [Dershowitz and Jouannaud1990] into the formalism of PATCH Graphs. Each application of **Patch** inference to cycles C_1 and C_2 in a PATCH Graph corresponds to a serie of applications of **Deduce**, **Simplify**, **Compose** and **Collapse** inference rules to symmetrized forms of rewriting rules represented by cycles C_1 and C_2.

Our inference rules avoid generating cycles giving some redundant rewriting rules. A cycle C_3 created by **Patch** inference rule for any pair of maximal non trivial common paths π_1 and π_2 represent rewriting rules which simplify equations or rewriting rules, defined by a cycle C_4, created by **Patch** inference rule for some non maximal common subpaths of paths π_1 and π_2. Every equation or rewriting rule, defined by a cycle C_5, generated by **Patch** inference rule applied to any pair of trivial common paths can be simplified to a trivial equation by group axioms. In the standard algorithm as well as in the group completion algorithm by Le Chenadec, equations and rewriting rules represented by such cycles C_4 and C_5 are introduced into the rewriting system, and the corresponding simplifications are done when they are discovered.

We believe that our work will be useful in solving open problems in the field of finitely presented groups. The simplicity, compactness, and avoidance of redundancies will help to focus the Knuth-Bendix completion procedure. Our PATCH Graph data structure complements other improvements of the Knuth-Bendix completion procedure, and we believe the combination of ideas is what will be most useful in solving the open problems.

Acknowledgements

We thank Dominique Fortin and Claude Kirchner for fruitful discussions on PATCH Graphs

Notes

[1] This notion differs from the standard notion of critical pair, given in [Book and Otto1993]. We specially distinguish between the case of simplification by a rewriting rule and the case of critical pair between rewriting rules, which are both included into the critical pair case in the standard definition

[2] Here and below we denote by asterisks non-significant components of labels

[3] The result of the computation of the set of symmetrized forms does not depend of the choice of a cycle cut

[4] $n = N + 1$, where N is the number of cycles in current graph

[5] Here and below we suppose a non-empty path to be composed of only C-edges

References

Book and Otto1993. R. V. Book and F. Otto. *String-Rewriting Sytsems.* Springer-Verlag, 1993.

Bücken1979. Hans Bücken. Reduction-systems and small cancellation theory. In *Proceedings of 4th Workshop on Automated Deduction*, pages 53–59, 1979.

Dershowitz and Jouannaud1990. N. Dershowitz and J.-P. Jouannaud. *Handbook of Theoretical Computer Science*, volume B, chapter 6: Rewrite Systems, pages 244–320. Elsevier Science Publishers B. V. (North-Holland), 1990. Also as: Research report 478, LRI.

Epstein *et al.*1991a. D. B. A. Epstein, D. F. Holt, and S. Rees. The use of knuth-bendix methods to solve the word problem in authomatic groups. *Journal of Symbolic Computation*, 12:397–414, 1991.

Epstein *et al.*1991b. D. B. A. Epstein, D. F. Holt, and S. Rees. The use of knuth-bendix methods to solve the word problem in automatic groups. *Journal of Symbolic Computation*, 12:397–414, 1991.

Fortin *et al.*1994. D. Fortin, C. Kirchner, and P. Strogova. Routing in Regular Networks Using Rewriting. In J. Slaney, editor, *Proceedings of the CADE international workshop on automated reasoning in algebra (ARIA)*, pages 5–8, June 1994.

Gilman1979. R. H. Gilman. Presentations of groups and monoids. *J. of Algebra*, 57:544–554, 1979.

Holt and Rees1989. D. Holt and S. Rees. Testing for isomorphism between finitely presented groups. In *Proceedings of the Conference on Groups and Combinatorics*, Cambridge, 1989. Cambridge University Press.

Kapur and Zhang1988. D. Kapur and H. Zhang. RRL: A rewrite rule laboratory. In *Proceedings 9th International Conference on Automated Deduction, Argonne (Ill., USA)*, volume 310 of *Lecture Notes in Computer Science*, pages 768–769. Springer-Verlag, 1988.

Knuth and Bendix1970. Donald E. Knuth and P. B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, Oxford, 1970.

Le Chenadec1986. Ph. Le Chenadec. *Canonical Forms in Finitely Presented Algebras.* John Wiley & Sons, 1986.

Linton and Shand1994. S. Linton and D. Shand. Some group theoretic examples with completion theorem provers. Technical Report CS/94/14, Dep. of Math. & Comp. Sciences, Univ. of St Andrew, 1994.

Lynch and Strogova1995. Christopher Lynch and Polina Strogova. Sour graphs for efficient completion. Technical Report 95-R-343, CRIN, 1995.

Lynch1995. Christopher Lynch. Paramodulation without duplication. In Dexter Kozen, editor, *Proceedings of LICS'95*, San Diego, June 1995.

Madlener *et al.*1991. K. Madlener, P. Narendran, and F. Otto. A specialized completion procedure for monadic string-rewriting systems presenting groups. In *Proceedings 18th ICALP Conference, Madrid (Spain)*, volume 516 of *Lecture Notes in Computer Science*, pages 279–290. Springer-Verlag, 1991.

Marché1993. C. Marché. *Réécriture modulo une théorie présentée par un système convergent et décidabilité du problème du mot dans certaines classes de théories équationnelles.* Thèse de Doctorat d'Université, Université de Paris-Sud, Orsay (France), October 1993.

Martin1996. U. Martin. Theorem proving with group presenations: examples and questions. In *Proceedings of CADE-13*, 1996. To appear.

McCune1994. W. W. McCune. Otter 3.0: Reference manual and guide. Technical Report 6, Argonne National Laboratory, 1994.

Naher1993. Stefan Naher. Leda manual. Technical Report MPI-I-93-109, mpi, 1993.

Sims1987. C. C. Sims. Verifying nilpotence. *Journal of Symbolic Computation*, 3:231–247, 1987.

Zhang1992. H. Zhang. Herky: High-performance rewriting techniques in rrl. In D. Kapur, editor, *Proceedings of 1992 International Conference of Automated Deduction*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 696–700. Springer-Verlag, 1992.

Address for correspondence: Technopôle de Nancy-Brabois, B.P. 101, 54602 Villers-lès-Nancy Cedex, France
phone: (+33) 83 59 30 00, fax: (+33) 83 27 83 19, e-mail: lynch, strogova@loria.fr