The conjugacy problem in right-angled Artin groups and their subgroups

John Crisp, Eddy Godelle, Bert Wiest

The conjugacy problem in right-angled Artin groups and their subgroups

John Crisp, Eddy Godelle, Bert Wiest

Institut de Mathémathiques de Bourgogne, CNRS UMR 5584
Université de Bourgogne, B.P. 47870, 21078 Dijon cedex, France
`jcrisp@gmail.com`
and
Laboratoire de Mathématiques Nicolas Oresme, CNRS UMR 6139
Université de Caen, 14032 Caen cedex, France
`eddy.godelle@math.unicaen.fr`
and
IRMAR, Campus de Beaulieu, CNRS UMR 6625
Université de Rennes 1, 35042 Rennes, France
`bertold.wiest@univ-rennes1.fr`

Abstract We prove that the conjugacy problem in right-angled Artin groups (RAAGs), as well as in a large and natural class of subgroups of RAAGs, can be solved in linear-time. This class of subgroups contains, for instance, all graph braid groups (i.e. fundamental groups of configuration spaces of points in graphs), many hyperbolic groups, and it coincides with the class of fundamental groups of "special cube complexes" studied independently by Haglund and Wise.

AMS Classification 20F36; 20F10, 20F65.

Keywords right-angled Artin group, partially commutative group, graph group, graph braid group, conjugacy problem, cubed complex, special cube complex.

1 Introduction

It is well known that the conjugacy problem in free groups can be solved in linear-time by a RAM (random access memory) machine. This result has been generalized in two different directions. On the one hand, Epstein and Holt [14] have shown that the conjugacy problem is linear in all word-hyperbolic groups. On the other hand, Liu, Wrathall and Zeger have proved the analogue result for all right-angled Artin groups ([23], based on [29]). Note that these groups are also called "partially commutative groups" or "graph groups" in the literature.

The aim of the present paper is to extend the second approach, in order to prove linearity of the conjugacy problem in a large class of subgroups of right-angled Artin groups. Very roughly speaking, the subgroups in question are fundamental groups of cubical complexes, sitting inside the right-angled Artin group in a convex fashion. This class of groups has previously been studied by Crisp and Wiest [11, 12], and independently by Haglund and Wise [19], as fundamental groups of so-called *special cube complexes* (or, more precisely, A-special cube complexes).

The class of groups considered in this paper contains in particular all graph braid groups [1, 2, 15, 16, 24] and more generally all state complex groups [3, 17]. These classes of groups have attracted considerable interest recently, which stems partially from their close relations to robotics [2, 17]. Indeed, our results can be interpreted as giving very efficient algorithms for motion planning of periodic robot movements. However, our results also apply to the various word-hyperbolic groups discussed in [11, 12] – in particular, to all surface groups except the three simplest non-orientable ones.

The present paper raises the stakes on the conjecture of Haglund and Wise [19] that all Artin groups (e.g. braid groups) are virtually fundamental groups of special cube complexes. If this conjecture was known to be true, then our work would imply that Artin groups have finite index subgroups where the conjugacy problem can be solved in linear time.

The plan of the paper is as follows. In the second section we present an alternative approach to the conjugacy problem in right-angled Artin groups, different from the one of Liu, Wrathall and Zeger, but rather close in spirit to the methods of Lalonde and Viennot [22, 28]. In the third section we prove that isometrically embedded subgroups of right-angled Artin group inherit a linear-time solution to the conjugacy problem from their supergroups.

2 The conjugacy problem in RAAGs is linear-time

We recall that a right-angled Artin group is a group given by a finite presentation, where every relation states that some pair of generators commutes. Graphically, a right-angled Artin group A can be specified by a simple graph Γ_A, where the generators of A correspond to the vertices of Γ_A, and a pair of generators commutes if and only if the corresponding vertices are *not* connected by an edge. Note that the opposite convention (connecting *commuting* generators by an edge) is also very common, but in the present paper we shall stick to this convention.

Right-angled Artin groups have been widely studied in the last decades – see [10] for an excellent survey. Several solutions to the word and conjugacy problem have been found. It seems to be difficult to have a complete bibliography of the large number of articles on these two problems. The first solutions to the word and the conjugacy problem was obtained by Servatius in [25]. In [27], Van Wyk constructed a normal form in right-angled Artin groups and proved that these groups are biautomatic. Indeed, even thought our point of view is very different from Van Wyk's, the normal form constructed in the present paper is the very similar to his. One of the main papers regarding the algorithmic complexity of these two problems is [23] (based on [29]) by Liu, Wrathall and Zeger, which proves that they are both of linear complexity.

The word problem in partially commutative *monoids* has also been widely studied and numerous papers appeared on that topics. Several approaches appeared to be successful. In [9], Cartier and Foata constructed a normal form on partially commutative monoids, and then obtained the first solution to the word problem. This normal form is the restriction of the normal form obtained in [27]. More recently, Viennot introduced in [28] a new tool, the so-called *Viennot's piling*, based on a geometrical representation of partially commutative monoids. Several works deal with this tool (see for instance [13] and [22]). The Viennot piling method associates a piling to each element of a partially commutative monoid and thereby provides a linear-time solution to the words problem in such a monoid. As remarked by Krob, Mairesse, and Michos in [21], this piling is canonically related to the normal form constructed in [9]. In [22], Lalonde introduces and uses the notion of a *pyramid* in order to study the conjugacy problem in partially commutative monoids. In the present paper, we are going to extend the notions of a piling and of a pyramid to the context of right-angled Artin *groups*, and use them in order to obtain a linear-time solution (to the word problem and) to the conjugacy problem. This leads us to introduce the notion of a cyclic normal form.

In order to get an intuition for the nature of the conjugacy problem in right-angled Artin groups, let us first consider the relatively easy case of free groups. Given two cyclic words of length ℓ_1 and ℓ_2 respectively, there is a two step algorithm which can be performed in time $O(\ell_1 + \ell_2)$ on a RAM machine: first each word can be cyclically reduced in time $O(\ell_1)$ and $O(\ell_2)$, respectively. If the reduced words have different lengths, then they are not conjugate. If they have the same length ℓ, then they can be compared in time $O(\ell)$ using standard pattern matching algorithms, like the Knuth-Morris-Pratt algorithm, the Boyer-Moore algorithm, or algorithms based on suffix-tree methods – see [20, 7, 4, 18, 26]. It should be stressed that on a Turing machine these algorithms take time $O(\ell \log(\ell))$.

In the sequel, we assume that A is a fixed right-angled Artin group given by a fixed presentation. We denote by $\{a_1, \cdots, a_N\}$ the generating set of A associated with this presentation.

The aim of this section is to provide an algorithm which does, very roughly speaking, the following: given a word w, another word w' with smaller or equal length is created in linear time such that w and w' represent conjugate elements of A. Furthermore, the word w' depends only on the conjugacy class in A of the element represented by w, up to a cyclic permutation of its letters. This yields a linear-time solution to the conjugacy problem in A because, given words w and v we can compute the canonical cyclic words w' and v' representing their conjugacy classes, and compare those by one of the algorithms mentioned above.

2.1 The word problem is linear-time

We start by recalling the following classical lemma.

Lemma 1 *[25] Any element of A can be represented by a* reduced *word (one which does not contain a subword of the form $a_i^{\pm 1} x a_i^{\mp 1}$, where all letters of x commute with a_i). Moreover, any two reduced representatives of the same element are related by a finite number of commutation relations – no insertions/deletions of trivial pairs are needed.*

Now we introduce our main tool, the notion of a *piling*.

Definition 2 An *abstract piling* is a collection of N words, one for each generator a_i of A, over the alphabet with three symbols $\{+, -, 0\}$.

The word associated with the generator a_i will be called the a_i*-stack* of the abstract piling. The product of two abstract pilings is defined as the piling obtained by concatenation of the corresponding stacks.

We define a function $\pi^\star$ on the set $\{a_1^{\pm 1}, \ldots, a_N^{\pm 1}\}^*$ of words on the $2n$ letters $a_1, a_1^{-1}, \ldots, a_N, a_N^{-1}$ that associates an abstract piling to every word in the following way: starting with the empty piling, we read the word from left to right. When a letter a_i^ϵ is read, we check what the last letter of the a_i-stack of the piling is. If this letter is different from $-\epsilon$ (the no-cancellation cases: the a_i-stack is empty, or finishes either with 0 or ϵ), then we append a letter $+$ or $-$ at the end of the a_i-stack of the piling (the sign of ϵ). Moreover, we also append a letter 0 at the end of each of the a_j-stacks associated with a generator a_j that does not commute with the generator a_i. On the other hand, if the last letter of the a_i-stack is $-\epsilon$ (the cancellation case), then we erase this last letter, and we also erase the terminal letter of each of the a_j-stacks of the piling associated with a generator a_j that does not commute with the generator a_i – note that the terminal letter of the a_j-stack is necessarily "0".

Definition 3 A *piling* is an abstract piling in the image of the function $\pi^\star$. The set of pilings is denoted Π.

We observe that the number of letters $+$ and $-$ occuring in the piling $\pi^\star(w)$ is at most equal to the length of the word w. Moreover, it is immediate from the description of the function $\pi^\star$ that, given a word w of length ℓ, the piling $\pi^\star(w)$ can be calculated in time $O(\ell)$ (linear-time).

It may be helpful to keep in mind the following physical interpretation of a piling: we have N vertical sticks, labelled by the generators $a_1, \dots, a_n$, with beads on it; the beads are labelled by $+$, $-$ or 0 such that when reading from bottom to top the sequence of labels of the beads on the a_i-stick, we obtain the a_i-stack of the piling. A letter a_i or a_i^{-1} of the word w corresponds to a set of beads (which we call a *tile*), consisting of one bead labelled $+$ or $-$ on the corresponding stick, and one bead labelled 0 on each of the sticks corresponding to generators of A which do not commute with a_i; each 0 labelled bead is connected to the $\pm$ labelled bead by a thread. The rule is: on a stick, adjacent 0-beads can commute with ("slide through") each other, but 0-beads do not commute with $\pm$-beads. In this physical model, we construct the image of a word by adding beads from the top, and removing opposed tiles when one obtains on a stick two adjacent $\pm$-beads with opposite signs. In fact, when we are dealing with the word problem we can forget about the threads between the beads, but they are helpful for thinking about the conjugacy problem.

Example 4 In the group A with group presentation

$$\langle a_1, a_2, a_3, a_4 \mid a_1a_4 = a_4a_1 \ ; \ a_2a_3 = a_3a_2 \ ; \ a_2a_4 = a_4a_2 \rangle$$

we can calculate the piling p of the word $a_2^{-2}a_4^{-1}a_3a_2a_4a_1a_2a_1^{-1}a_2^2a_4^{-1}$ as indicated in Figure 1.

The map $\pi^\star$ induces a well-defined function $\pi\colon A \to \Pi$ because words representing the same element of A have the same image under $\pi^\star$: the image of a word is unchanged by applying a commutation relation, and by inserting or deleting a trivial pair $a_ia_i^{-1}$ or $a_i^{-1}a_i$. Now, from the definitions it is immediate that no cancellation occurs during the construction of the piling $\pi^\star(w)$ of a reduced word w. Then, the identity of A is the unique element of A whose

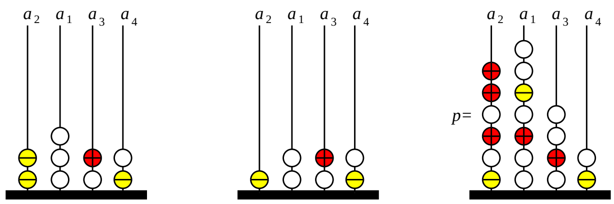

Figure 1: The pilings of the prefixes $a_2^{-2}a_4^{-1}a_3$ and $a_2^{-2}a_4^{-1}a_3a_2$, and of the full word $a_2^{-2}a_4^{-1}a_3a_2a_4a_1a_2a_1^{-1}a_2^2a_4^{-1}$

image by π is the trivial piling, and therefore the word problem is solved in linear-time: a word w represents the identity if and only if its piling $\pi^\star(w)$ is trivial; this piling can be built in linear-time.

The following notion will be extremely useful in the next section when we consider the conjugacy problem.

Definition 5 Let w be a reduced word.

(i) We say that w is *initially normal* when w is trivial or when the index of its first letter is greater or equal to the index of the first letter of any equivalent reduced word.

(ii) We say that w is *normal* when all its suffixes are initially normal.

We remark that all the factors of a normal word are normal words.

Proposition 6 *Any element of A has a unique normal reduced representative word.*

Proof For any reduced word $w = a_{i_1}^{\varepsilon_1} \cdots a_{i_k}^{\varepsilon_k}$, where $\varepsilon_j = \pm 1$, we set

$$\Omega(w) = \{(r, s) \mid 1 \leq r < s \leq k \text{ and } i_r < i_s\}.$$

Let a be in A. In order to prove that a has normal reduced representative word, we choose, among all words representing a, a word w for which the number $\#\Omega(w)$ is as small as possible (possibly equal to zero). This word w is minimal.

We shall prove uniqueness of the normal representative by induction on the length. If a is of length 1, i.e. if $a = a_i^\varepsilon$ for $\epsilon = \pm 1$, then uniqueness is obvious.

Now suppose that a has two normal reduced representatives $w = a_{i_1}^{\varepsilon_1} \cdots a_{i_k}^{\varepsilon_k}$ and $w' = a_{i'_1}^{\varepsilon'_1} \cdots a_{i'_k}^{\varepsilon'_k}$. Since the the suffixes of length $k-1$ of w and w' are again normal, it is, by induction hypothesis, sufficient to prove that $a_{i_1}^{\varepsilon_1} = a_{i'_1}^{\varepsilon'_1}$. Since w and w' are normal, we have $i_1 = i'_1$. Now, the exponents also have to be equal by Lemma 1: we can not transform the word $a_{i_1}^{\varepsilon_1} \cdots a_{i_k}^{\varepsilon_k}$ into the word $a_{i_1}^{-\varepsilon_1} a_{i'_2}^{\varepsilon'_2} \cdots a_{i'_k}^{\varepsilon'_k}$ by using commutation relations only: starting from the reduced w, no word of the form $u a_{i_1}^{\varepsilon_1} a_{i_1}^{-\varepsilon_1} u'$ can appear by any sequence of commutation relations. □

In the sequel, we call this unique normal reduced word representing a the *normal form* of a.

Proposition 7 *There is a linear-time algorithm that associates to each piling p a normal word $\sigma^\star(p)$ such that $\pi^\star(\sigma^\star(p)) = p$. Furthermore, for any element a of A the word $\sigma^\star(\pi(a))$ is the normal form of a.*

Example 8 Using the notation of Example 4, the word $\sigma^\star(p)$ is equal to $a_4^{-1} a_3$ $a_2^{-1} a_1 a_2 a_1^{-1} a_2 a_2$. The calculation is shown in Figure 2.

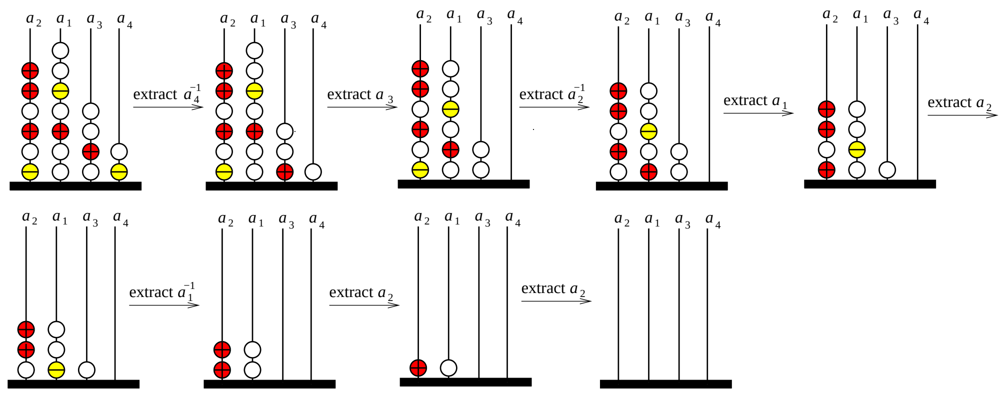

Figure 2: The word $\sigma^\star(p)$ associated to a piling p

Proof of Proposition 7 Let p be a piling. By definition, this means that there exists an element a of A such that $\pi(a) = p$. In order to prove Proposition 7, it suffices to find an algorithm for constructing in linear time a word $\sigma^\star(p)$, and to prove that $\sigma^\star(p)$ is a normal reduced representative of a.

We start with the observation that the element a has a reduced representative starting with the letter $a_i^{\pm 1}$ if and only if the a_i-stack of the piling is nonempty and starts with the letter $+$ or $-$, respectively (not with the letter 0).

We associate to p a normal reduced word $\sigma^\star(p)$ by induction on the number of letters $+$ and $-$ in p in the following way. If p is empty then $\sigma^\star(p)$ is the empty word. Otherwise, let i be the largest index with the property that the a_i-stack of p is nonempty and starts with the letter $+$ or $-$, not with 0. Then, according to this sign, we define the first letter of $\sigma^\star(p)$ to be a_i or a_i^{-1}, respectively. Then we remove the tile consisting of the first letter ($+$ or $-$) of the a_i-stack, and of the initial letter (which has to be 0) of each of the a_j-stacks associated with a generator a_j that does not commute with a_i. What remains is a piling p_1 with strictly fewer letters. Thus the word $\sigma^\star(p_1)$ is already defined, by induction hypothesis, and we define the word $\sigma^\star(p)$ by concatenation $\sigma^\star(p) = a_i^{\pm 1}\sigma^\star(p_1)$.

We claim that the word $\sigma^\star(p)$ is a normal reduced representative of a; indeed, in the above construction we see that the first letter of $\sigma^\star(p)$ is also the first letter of some reduced representative of a. By induction, the whole word $\sigma^\star(p)$ is a reduced representative of a. Moreover, the word $\sigma^\star(p)$ is initially normal, by construction, and by induction its suffix $\sigma^\star(p_1)$ is normal. Hence the whole word $\sigma^\star(p)$ is normal. $\square$

2.2 Cyclic normal forms and pyramidal pilings

We are now ready to attack the conjugacy problem.

2.2.1 Cyclically reduced words and cyclically reduced pilings

We recall that a *cycling* of a reduced word w is the operation of removing the first letter of the word, and placing it at the end of the word. A word is called *cyclically reduced* if it is reduced and if any word obtained from it by a sequence of cyclings and commutations is still reduced – in other words, if it is not of the form $x_1 a_i^{\pm 1} x_2 a_i^{\mp 1} x_3$, where all the letters of x_1 and x_3 commute with a_i. As far as we know, all known solutions to the conjugacy problem in RAAGs are based on the following lemma.

Lemma 9 *Two cyclically reduced words represent conjugate elements of A if and only if they are related by a sequence of cyclings and commutation relations.*

Therefore two reduced words w_1, w_2 with letters in $\{a_1^{\pm 1}, \ldots, a_n^{\pm 1}\}$ represent conjugate elements of A if and only if there is a sequence of words

$$w_1 \xrightarrow{\text{red}} v_1 \leftrightarrow v_2 \xleftarrow{\text{red}} w_2$$

where the two arrows labelled "red" represent two sequences of cyclic reductions down to cyclically reduced words and the arrow $\leftrightarrow$ represents a finite sequence of cyclings and commutation relations.

Definition 10 If, in a piling p, the a_i-stack starts (*resp.* finishes) with a letter $+$ or $-$, the *bottom* a_i*-tile* (*resp.* the *top* a_i*-tile*) of p is the *sub-piling* formed by the first (*resp.* last) letter of the a_i-stack and the first (*resp.* last) letter of the a_j-stacks such that a_i and a_j do not commute in A.

Example 11 With the notation of Example 4, Figure 3 gives an example of top and bottom tiles of a piling.

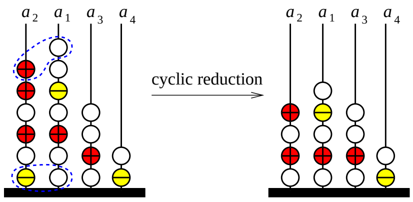

Figure 3: a top a_2-tile and a bottom a_2-tile, and the associated cyclic reduction

Definition 12 If in a piling p the a_i-stack starts with the letter $+$ and ends with $-$, or vice versa, a *cyclic reduction* is the act of removing both top and bottom a_i-tiles. We say that the piling is *cyclically reduced* if no cyclic reduction is possible.

Note that cyclically reducing a piling yields again a piling. We remark that there is an obvious linear-time algorithm for transforming any piling into a cyclically reduced one by a finite sequence of cyclic reductions. We also observe that for a reduced word $w \in \{a_1^{\pm 1}, \dots, a_N^{\pm 1}\}^*$, cycling of w corresponds to a cycling of its piling, and that w is cyclically reduced if and only if the piling $\pi^\star(w)$ is.

Now we have a fast algorithm for cyclically reducing words and pilings. In contrast to the case of free groups, however, the reduced words which we can obtain are *not* unique up to cyclic permutation. In order to circumvent this problem, we shall introduce in the sequel the notion of a *cyclic normal form*.

2.2.2 Non-split words and non-split pilings

Our first objective is to restrict the conjugacy problem to the case of *non-split cyclically reduced words* (or *pilings*). We recall that a graph Γ_A is associated to the right-angled Artin group A.

Definition 13 Let w be a reduced word different from 1, and let p be its image by $\pi^\star$. Consider $\Delta(p)$ (or $\Delta(w)$) the full subgraph of Γ_A whose vertices are those whose correponding stacks contain at least one bead different from 0 (in other words, the letters a_i such that $a_i^{\pm 1}$ occurs in w). Then, the word w and the piling p are said to be *non-split* when the graph $\Delta(p)$ is connected.

In other words, w is non-split if and only if its set of letters cannot be separated in two disjoint subsets such that every letter of one of the subset commutes in A with every letter of the other subset. Clearly, it takes linear-time to obtain the set of vertices of the graph $\Delta(p)$, and constant time (which depends on the graph Γ_A) to decide if $\Delta(p)$ is connected. If it is not, it takes still constant time to determine the connected components $\Delta_1(w), \ldots, \Delta_k(w)$ of $\Delta(w)$. Figure 4 (which still uses the notation of Example 4) contains examples of both split and non-split pilings.

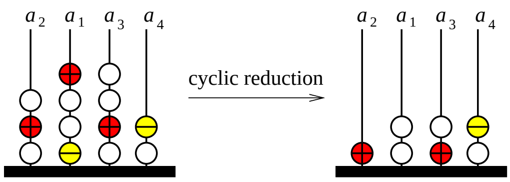

Figure 4: The word $a_1^{-1}a_2a_3a_1a_4^{-1}$ is not split, but cyclic reduction yields a word which is split: $a_2(a_3a_4^{-1}) = (a_3a_4^{-1})a_2$

Now, if w is a cyclically reduced word that is split, then it is equivalent to a product $w_1 \cdots w_k$ of non-split cyclically reduced words, one for each connected component $\Delta_i(w)$ of the graph $\Delta(w)$; the graph $\Delta_i(w)$ is equal to $\Delta(w_i)$. Furthermore, once that the connected components $\Delta_1(w), \ldots, \Delta_k(w)$ of $\Delta(w)$ are computed, appropriate words $w_1, \ldots, w_k$ can be obtained in linear-time.

Remark 14 The following observation will be crucial: if v is another cyclically reduced word, then then w and v represent conjugate elements if and only if two conditions are satisfied: firstly the graph $\Delta(v)$ is equal to $\Delta(w)$; secondly, if $v_1, \ldots, v_k$ are words such that $\Delta(v_i) = \Delta_i(v)$ and such that v is equivalent to the product $v_1 \cdots v_k$, then for each index i the words w_i and v_i represent conjugate elements.

Therefore, in order to obtain a solution to the conjugacy problem in linear-time it is enough to consider the case of cyclically reduced non-split words.

2.2.3 Pyramidal piling and cyclic normal form

To solve the conjugacy problem, we associate in the sequel a *cyclic normal word* to each cyclically reduced non-split word. We first do the analogue of this in the framework of pilings: to each non-split cyclically reduced piling, we associate a *pyramidal* piling.

Definition 15 Let p be a non-empty piling, and denote by i the smallest index such that the a_i-stack contains an $a_i^{\pm}$-bead. We say that the piling p is *pyramidal* if the first bead of every a_j-stack except the a_i-stack is either empty or starts with the letter 0. In that case, we say that a_i is the *apex* of the pyramidal piling.

Note that a pyramidal piling has to be non-split.

Lemma 16 *(i) Let p is a non-empty piling and denote by i the smallest index such that the a_i-stack of p contains an $a_i^{\pm}$-bead; then there exists a unique decomposition $p_0 \cdot p_1$ of p such that p_1 is a pyramidal piling with a_i as apex, and p_0 is a piling without $a_i^{\pm}$-beads. Furthermore, one has the equality of words $\sigma^\star(p) = \sigma^\star(p_0)\sigma^\star(p_1)$.*

(ii) The above decomposition $p_0 \cdot p_1$ can be computed in linear-time on the number of $\pm$-beads of the piling p.

Example 17 Using the notation of Example 4, Figure 5 gives an example of a decomposition of a piling.

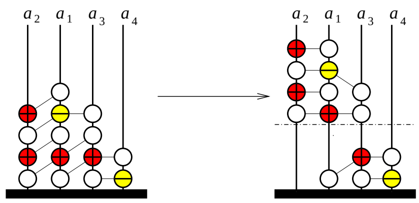

Figure 5: Decomposition of a piling as $p_0 \cdot p_1$

Proof of Lemma 16 We start by exhibiting a linear-time algorithm for finding such a decomposition of a given non-empty piling p. Let p_0 be the empty piling. Reading all the stacks (in the index order), obtain in linear-time the smallest index i for which the a_i-stack contains a bead distinct from 0. Then, apply iteratively the following recipe: consider the largest index j (necessarily greater than i) for which the a_j-stack starts with a letter $+$ or $-$; then remove all the beads in the bottom a_j-tile, and add them to the top of the piling p_0. When no more beads can be extracted from the bottom of the piling p, then the construction of the factor p_0 is complete, and what remains is the piling p_1. This proves the existence part of (i), as well as part (ii). The formula $\sigma^\star(p) = \sigma^\star(p_0)\sigma^\star(p_1)$ is now immediate by construction. For the uniqueness part of (i), we notice that in any decomposition $p = p_0 \cdot p_1$, the factor p_0 has to contain exactly those tiles that can be extracted on the bottom from p without extracting any apex bead. $\square$

We call the piling p_0 the 0*-factor* of p. Thus the piling p is pyramidal if and only if its 0-factor is empty.

In our physical interpretation, if i is the smallest index such that the a_i-stack contains a $a_i^\pm$-bead, we can lift up the first $a_i^\pm$-bead along its stick to the first floor. Then some part of the piling stays on the ground, while some beads are lifted up. Here it is essential to keep in mind that each 0-bead is connected by a thread to a $\pm$-bead, and that adjacent 0-beads on a stick can slide through each other. The factor that stays down is p_0, the factor that is lifted up is p_1. This latter factor has the structure of an upside-down pyramid supported by one of the apex-beads, hence the names.

If in a cyclically reduced piling, the a_i-stack starts with a letter $+$ or $-$, then one can perform a *cycling* of the bottom tile containing that bead to the top of the piling, i.e., one can move the initial letter $+$ on the a_i-stack, and the initial letters 0 on the stacks corresponding to letters that do not commute with a_i, to the end of their respective stacks. A physical interpretation (see Figure 6) of this procedure is obtained by replacing the sticks by concentric hula hoops. A cycling of a bottom tile corresponds to the operation of cycling the corresponding tile along the hula hoops.

Proposition 18 *There is an algorithm which takes as its input any non-split cyclically reduced piling p and which outputs a pyramidal piling that is obtained from the input piling by a finite sequence of cyclings. If the piling has ℓ beads, then the algorithm requires $O(\ell)$ cyclings, so its computational complexity is $O(\ell)$.*

Proof The basic procedure of the algorithm is in two steps; given a cyclically reduced piling p, first determine the 0-factor p_0 of the canonical decomposition

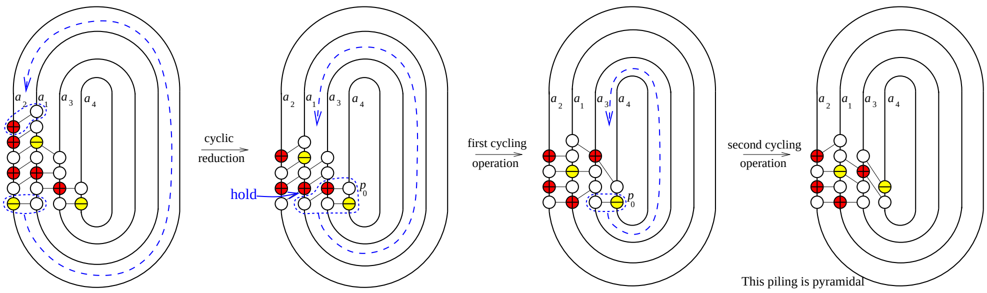

Figure 6: The calculation of a pyramidal piling

(by the method of Lemma 16). Secondly, cycle all the tiles belonging to p_0 in order to obtain a new piling. This procedure takes time $O(\ell)$. The algorithm is simply to iterate this basic procedure until the factor p_0 is empty. It remains to prove that there is a bound on the number of iterations which depends only on the group A, not on the piling p. In fact, if we denote i the smallest index such that p contains an a_i-tile, and $\Delta(p)$ the full subgraph of the defining graph Γ defined above, we claim that $\max_{a_j \in \Delta(p)} \; dist_\Delta(a_i, a_j)$ is an upper bound on the number of iterations, where each edge of $\Delta(p)$ has length 1. This quantity is finite, because $\Delta(p)$ is connected, and is bounded above by N, the number of generators of the group A (which does not depend on the piling p). This fact is obvious from the geometrical representation, and the proof is a straightforward induction: after the first iteration of the basic procedure, no $a_j^{\pm}$-beads such that a_j is at distance 1 from a_i in $\Delta(p)$ appear in the 0-factor; after a second iteration no $a_j^{\pm}$-beads such that a_j is at distance at most 2 from a_i in $\Delta(p)$ appear in the 0-factor, and so on. □

Now, if w is a non-split reduced word, we can apply the above algorithm to the piling $\pi(w)$ to obtain a pyramidal piling p. Then, the words $\sigma^\star(p)$ and w represent conjugate elements.

Definition 19 Let w be a word in $\{a_1^{\pm 1}, \ldots, a_n^{\pm 1}\}^*$ that is reduced and cyclically reduced. We say that the word w is a *cyclic normal form* if it is normal and all its cyclically conjugate words are normal.

Intuitively, if we regard w as a cyclic word, and we start reading anywhere in the word, then the first letter that we read must always be the largest-index letter that can be extracted on the left. For instance, with the notation of Example 4, the word $a_4^{-1}a_3a_2^{-1}a_1a_2a_1^{-1}a_2a_2$ is not a cyclic normal form: starting from the last letter and reading cyclically, we read out $a_2 a_4^{-1} \ldots$, which is already illegal, because the letters commute, and a_4^{-1} has a larger index than a_2, so a_4^{-1} should come first. Another example: the word $a_1 a_2 a_1^{-1} a_3 a_4^{-1} a_2$ is a cyclic normal form. Our linear-time solution to the conjugacy problem is based on the two following results.

Proposition 20 *If p is a non-split cyclically reduced pyramidal piling, then $\sigma^\star(p)$ is a cyclic normal form.*

Proposition 21 *Two cyclic normal forms represent conjugate elements if and only if they are equal up to a cyclic permutation.*

Proof of Proposition 20 Firstly, we remark that a consequence of Lemma 1 is the following fact: if a^ϵ, b^η are letters ($\epsilon, \eta = \pm 1$) and w is a reduced word such that $b^{-\eta} w$ and $w a^\epsilon$ are both reduced (*i.e.* no word equivalent to w starts and finishes with b^η and $a^{-\epsilon}$, respectively) but the word $b^{-\eta} w a^\epsilon$ is not reduced (*i.e.* $w a^\epsilon$ is equivalent to some word that starts with b^η), then $a^\epsilon = b^\eta$ and all the letters of w commute with a. Now, we know that $\sigma^\star(p)$ is a normal cyclically reduced word. For a cyclically reduced word w, the word ww is cyclically reduced (this follows directly from the above fact, or from the piling representation), and all the words cyclically conjugate to the former are subwords of the latter. Therefore, in order to prove the result, it is enough to prove that the word $\sigma^\star(p)\sigma^\star(p)$ is normal. Assume that this is not the case. Since $\sigma^\star(p)$ is normal, we can then write $\sigma^\star(p) = w_1 a_j^\eta w_2 = v_1 a_i^\epsilon v_2$ such that $a_j^\eta w_2 v_1$ is initially normal but $a_j^\eta w_2 v_1 a^\epsilon$ is not. In particular, there exists a_k^ν, with $k > j$, such that $a_k^{-\nu} a_j^\eta w_2 v_1$ is reduced but $a_k^{-\nu} a_j^\eta w_2 v_1 a_i^\epsilon$ is not. Since $a_j^\eta w_2 v_1 a_i^\epsilon$ is a subword of $\sigma^\star(p)\sigma^\star(p)$, it is reduced. Using the above fact, we get that $a_i^\epsilon = a_k^\nu$, and a_k^ν commute with all the letter of $a_j^\eta w_2 v_1$. In particular, the word $\sigma^\star(p)$ is equivalent to $a_k^\nu v_1 v_2$. This is impossible because k is greater that j, and p is pyramidal. Therefore, $\sigma^\star(p)\sigma^\star(p)$ is normal. □

Proof of Proposition 21 The "if" implication is obvious, we have to prove the "only if" part.

Let w and w' be two cyclic normal forms that represent conjugate elements. Let i be the smallest index that appears in w and choose a distinguished letter a_i^ε in w. As the words w and w' are cyclically reduced, there exists a sequence of words $w_0 = w \to w_1 \to \cdots w_r = w'$ that transforms w into w', such that w_{i+1} is obtained from w_i by a commutation or a cycling transformation.

We can keep track of the distinguished letter a_i^ε along the transformations: write $w_j = w'_j a_i^\varepsilon w''_j$. Assume the number ℓ of commutations that involve the distinguished letter is positive. Since w is a normal word, the first commutation $w_j \to w_{j+1}$ that involves a_i^ε is "from left to right", i.e. it is of the following form: $w_j = w'_{j+1} a_{i'}^{\varepsilon'} a_i^\varepsilon w''_j$ and $w_{j+1} = w'_{j+1} a_i^\varepsilon a_{i'}^{\varepsilon'} w''_j$ with $i' > i$.

Now, consider the last operation $w_p \to w_{p+1}$ such that a letter a_k^η is exchanged with the distinguished letter a_i^ε from left to right: we have $w_p = w'_{p+1} a_k^\eta a_i^\varepsilon w''_p$ and $w_{p+1} = w'_{p+1} a_i^\varepsilon a_k^\eta w''_p$. We can also keep track of the distinguished letter a_k^η. As long as the two letters do not cross each other again in the opposite direction, we have $w''_q w'_q = y_q a_k^\eta z_q$ such that all the letters of y_q commute with a_k (where q satisfies $q > p$). In particular, $a_i^\varepsilon w''_q w'_q$ is not initially normal. But w' is normal, so the two distinguished letters have to cross each other again in the opposite direction: there exists s, with $p < s < r$, such that $w_s = w'_s a_i^\varepsilon a_k^\eta w''_{s+1}$ and $w_{s+1} = w'_s a_k^\eta a_i^\varepsilon w''_{s+1}$. Hence, we have a sequence

$$w_p = w'_{p+1} a_k^\eta a_i^\varepsilon w''_p \to v'_{p+1} a_k^\eta a_i^\varepsilon v''_{p+1} \to \cdots \to v'_{s-1} a_k^\eta a_i^\varepsilon v''_{s-1} \to v'_s a_k^\eta a_i^\varepsilon v''_s \to w_{s+1}$$

such that each word $v''_q v'_q$ is equal to the word $y_q z_q w'_q$. Thus we obtain a new sequence from w to w' with only $\ell - 2$ commutations that involve the distinguished letter a_i^ε.

It follows that we can assume that no commutation involves the distinguished letter a_i^ε along the sequence $w_0 = w \to w_1 \to \cdots w_r = w'$. But this implies that the words $a_i^\varepsilon w''_1 w'_1$ and $a_i^\varepsilon w''_r w'_r$ are equivalent. As they are both cyclic normal forms, they are normal words. Therefore they are equal by Proposition 6. Hence, the words w and w' are equal up to a cyclic permutation. □

Summing up, in order to decide whether two nonsplit cyclically reduced words represent conjugate elements, it suffices to decide whether their cyclic normal forms are equal (as cyclic words), and these cyclic normal forms can be calculated in linear time. More formally, we have

Theorem 22 *The conjugacy problem in a right-angled Artin group A is linear-time on the sum of the lengths of the two input words.*

Proof Here is a summary of the algorithm:

Given any two words w and v,

(i) produce the piling $\pi^\star(w)$, and then by cyclic reduction a cyclically reduced piling p; similarly for the word v produce first the piling $\pi^\star(v)$, and cyclically reduce it to a piling q;

(ii) factorize each of the pilings p and q into non-split factors. If the collection of subgraphs $\Delta_i(p)$ and $\Delta_i(q)$ of the defining graph Γ_A do not coincide, output "NO, w and v do not represent conjugate elements" and stop. Otherwise,

(iii) if $p = p^{(1)} \cdot \ldots \cdot p^{(k)}$ and $q = q^{(1)} \cdot \ldots \cdot q^{(k)}$ are the factorizations found in step (ii), then for $i = 1, \ldots, k$ do the following

(a) transform the non-split cyclically reduced pilings $p^{(i)}$ and $q^{(i)}$ into pyramidal pilings $\widetilde{p}^{(i)}$ and $\widetilde{q}^{(i)}$, using a sequence of cyclings. Then produce the words in cyclic normal form $\sigma^\star(p^{(i)})$ and $\sigma^\star(q^{(i)})$;

(b) decide whether the words in cylic normal form found in the previous steps are the same up to cyclic permutation (in linear-time, using a standard algorithm). If they are not, answer "NO" and stop.

(iv) answer "YES". □

2.3 Calculating the centralizer of an element

The centralizer of a cyclically reduced element of A has a canonical finite generating set: suppose that w is a cyclically reduced word, written as a product of cyclically reduced non-split words $w = w_1 \cdots w_k$, c.f. Section 2.2.2. Then, according to [6], for each i in $\{1, \ldots, k\}$ there exists a unique maximal infinite-cyclic subgroup of A containing $[w_i]$, generated by some cyclically reduced element $[z_i]$, and by [25] the centralizer of $[w]$ in A is generated by

(1) the elements $[z_i]$, and

(2) the generators of A which commute with all the generators occurring in w.

In the next section we will need to algorithmically determine explicit representatives of these generators, in the special case where the words w_i are cyclic normal forms.

Proposition 23 *There is a linear-time algorithm which takes as its input a cyclically reduced word w, decomposed as a product of words in cyclic normal form $w = w_1 \cdots w_k$, and which outputs the canonical generating set of the centralizer of w.*

Proof It takes linear time to determine the graph $\Delta(w)$, and then constant time to deduce from this the generators of type (2).

Now we turn to the generators of type (1), i.e. the minimal roots $[z_i]$ of the elements $[w_i]$. As a first step, we claim that periodicity of elements is visible in their cyclic normal form. More precisely, if one of the words w_i is equivalent to a word of the form $\widetilde{z}_i^r$ for some word $\widetilde{z}_i$ and some integer r, then the word w_i itself is of the form z_i^r, for some word z_i. In order to prove this claim, we observe that $\widetilde{z}_i$ is equivalent to a word z_i in cyclic normal form (because the 0-factor of $p(\widetilde{z}_i)$ must divide the 0-factor of $p(w_i)$, which is the trivial word). Now the word z_i^r is still in cyclic normal form (c.f. the proof of Proposition 20), and it is equivalent to w_i. Therefore we have $z_i^r = w_i$.

We claim that for each of the factors w_i, the desired minimal root z_i of w_i is detectable in linear-time: we can calculate a pair (z_i, r), where z_i is a word and r an integer with $z_i^r = w_i$, and r is maximal among all such pairs. Indeed, this algorithm works as follows: consider the word w_i^* obtained by removing the first letter from the word $w_i w_i$. Then find the starting point of the first occurrence of w_i as a subword of w_i^* – this can be done by standard algorithms, like the Boyer-Moore algorithm, in time $O(\ell_i)$, where ℓ_i denotes the length of w_i. If this starting point is at the ℓ_ith letter of w_*, then there is no periodicity. If on the other hand the starting point is at the tth letter with $t < \ell_i$, then let z_i be the prefix of w_i of length t. By construction we have an equality of words $z_i w_i = w_i z_i$. This implies that the words w_i and z_i have a common root. By the choice of z_i, this root has to be z_i itself and for $r := \ell_i/t$ we have an equality of words $w_i = z_i^r$. Finally, by the choice of t, no prefix of w_i of length less than t can be a root of w_i, so z_i is indeed the minimal root. □

3 The conjugacy problem in subgroups of RAAGs

In the previous section we saw that the conjugacy problem in a fixed right-angled Artin group can be solved in linear-time on a RAM-machine with constant that depends only on the group. In this section we shall prove analogue results for a large class of subgroups of right-angled Artin groups, namely those considered in the papers [11, 12], as well as in [19].

3.1 A class of subgroups of RAAGs

Every right-angled Artin group A admits a finite $K(A,1)$, called the Salvetti complex of A, which we shall denote Y and which can be constructed explicitely from the presentation of A. It is a cubed complex which has one single vertex, and one edge of length 1 for every generator of A. Moreover, for every n-tuple of mutually commuting generators of A, there is one $(n+1)$-torus in Y. We equip every cell, of any dimension, of this complex with the flat metric, in the sense that in the universal cover $\widetilde{Y}$ every cell is a Euclidean cube of sidelength 1. Then the complex is locally CAT(0), and its universal cover $\widetilde{Y}$ is CAT(0). For instance, for the group $A = \mathbb{Z}^2 = \langle a_1, a_2 \mid [a_1, a_2] = 1\rangle$, the complex Y is a torus, constructed out of one vertex, two edges, and one square which glued to the 1-skeleton according to the commutation relation. See [11] for details. The reader should note that as soon as an orientation is chosen on each edge (*i.e.* simple loop) of Y, one obtains an explicit isomorphism between A and $\pi_1(Y)$ such that the image of each generator a_i of A is represented by the simple loop labelled by a_i traversed in the positive direction.

Now, suppose that X is a finite locally $CAT(0)$ cubed complex, and consider a cubical map $\Phi\colon X \to Y$, sending each open cube of X bijectively and locally isometrically to a cell of the same dimension in Y. (Here Y still denotes the Salvetti complex of some right-angled Artin group.) If one of the vertices of X is designated as its basepoint, then such a mapping induces a homomorphism $\Phi_*\colon \pi_1(X) \to \pi_1(Y)$. See Figure 7 for an example where X and Y are 1-dimensional complexes.

We need some more notation: for any vertex x of X, we denote by $\Phi_{lk}\colon lk(x,X) \to lk(\Phi(x), Y)$ the induced map from the link of x in X to the link of $\Phi(x)$ in Y We shall be interested in the following two properties which our map Φ may have:

- The convexity property: for any vertex x of X, and any two vertices of $lk(\Phi(x), Y)$ which belong to the image $\Phi_{lk}(lk(x,X))$ and which are connected by an edge, the connecting edge belongs to the image $\Phi_{lk}(lk(x,X))$, as well.
- The injectivity property: the map of universal covers $\widetilde{\Phi}\colon \widetilde{X} \to \widetilde{Y}$ is injective. In particular, $\Phi_*\colon \pi_1(X) \to \pi_1(Y)$ is a monomorphism.

We remark that a map Φ satisfying the two hypotheses is a local isometry. Now, the subgroups of the right-angled Artin group $A \cong \pi_1(Y)$ for which we shall solve the conjugacy problem are the fundamental groups $\pi_1(X)$ of cubical complexes X which admit a cubical map $\Phi\colon X \to Y$ with the convexity and injectivity property.

Remark If X and Y are both known to be $CAT(0)$ cube complexes then the convexity property implies the injectivity property – cf. [11], Theorem 1 and the remark following. Conversely, the two conditions, together with the knowledge that Y is $CAT(0)$, imply that the complex X is itself $CAT(0)$.

The reader unfamiliar with the geometrical language used in stating the conditions should remember that the convexity and injectivity properties are satisfied by all the subgroups of right-angled Artin groups discussed in Theorem 1 of [11]. So some typical examples to keep in mind are those given in this paper. More generally, in order to get a mental image of the class of subgroups satisfying the two hypotheses, one can think of a subgroup whose Cayley graph sits in the Cayley graph of A in a "flat" way. Moreover, as proven by Haglund and Wise ([19], Theorem 4.2), for a cubed complex X, the property of admitting map Φ to a RAAG with the convexity and injectivity property can be characterized purely in terms of certain combinatorial conditions on the complex X – they call such complexes *special*.

General Notation and Conventions for the rest of the section

- We fix once and for all a right-angled Artin group A given by a presentation with generators $a_1, \dots, a_N$, and we denote by Y the cubed complex associated with A. We fix an orientation on every edge of Y and identify A with $\pi_1(Y)$, using the chosen orientations.
- We also fix a finite cubed complex X and $\Phi\colon X \to Y$ a cubical map satisfying the convexity and injectivity condition. Finally, we fix a label $x_1, x_2, x_3, \dots$ for each vertex of X.

Roughly speaking, our main result is the following

Claim: Using the General Notation and Conventions of this section, the conjugacy problem in the group $\pi_1(X)$, with respect to any finite set of generators of $\pi_1(X)$, is solvable in linear-time.

Phrased in this way, however, this statement is somewhat dissatisfying, because we have not even stated how the generators of $\pi_1(X)$ are specified. A more precise statement will be given in Theorem 25 below.

In fact, we will not directly solve the conjugacy problem in the fundamental group of X, but a more general problem, namely the conjugacy problem in the fundamental *groupoid* of X, in linear-time. First, we explain what precisely that means.

Let us fix a (positive) orientation for each edge of the complex X by pulling back along Φ the orientation of edges in Y. An element of the fundamental groupoid is, by definition, a homotopy class of paths (with fixed endpoints) from some vertex x_i to some vertex x_j. Such an element of the fundamental groupoid can be represented by a finite sequence of successive directed edges, which may be traversed in the positive or in the negative direction. We shall call such a sequence an *edge path* from x_i to x_j. Similarly in Y we have an analogue notion of an edge path as a homotopy class of path specified as a sequence of positively or negatively directed edges.

We shall use the following very convenient way of coding edge paths in X and Y: in Y, we shall simply identify closed edge paths with words in the letters $a_1^{\pm 1}, \ldots, a_N^{\pm 1}$. As for X, the map Φ gives rise to a coding of edge paths in X by *based words.*

Definition 24 A *based word* is a word of the form $x_i w x_j$, where x_i and x_j are vertices of X, and w is the image under Φ of an edge path in X starting at x_i and ending at x_j. The vertex x_i is called the *base vertex* of the based word.

In other words, the edge path $x_i w x_j$ is by definition the pullback to X of the path w in Y which starts at x_i and ends at x_j. Notice that not every word of the form $x_i w x_j$, with x_i and x_j vertices of X and w a word with letters in $\{a_1^{\pm 1}, \ldots, a_N^{\pm 1}\}$, is a based word. However, when it is, then it *uniquely* determines an edge path in X, because of the injectivity property. For instance, if $x_i w x_j$ is a based word, and if the word w can be written as a concatenation $w = w_1 w_2$, then there exists a unique vertex x_k such that $x_i w_1 x_k$ and $x_k w_2 x_j$ are based words. For an example of based words, see again Figure 7.

Two elements of the fundamental groupoid of X can be multiplied if the terminal vertex of the first coincides with the initial vertex of the second. In terms of based words, $(x_i w_1 x_j) \cdot (x_j w_2 x_k) = x_i w_1 w_2 x_k$. Two loops in X are freely homotopic if and only if they represent conjugate elements of the fundamental groupoid. If the loops are represented by based words $x_1 w x_1$ and $x_2 v x_2$, then this equivalent to the existence of a based word $x_1 u x_2$ such that the elements of the fundamental groupoid represented by $x_1 u v u^{-1} x_1$ and $x_1 w x_1$ coincide.

Our main result can now be stated precisely. The proof will occupy the whole rest of the paper:

Theorem 25 *Using the General Notation and Conventions of this section, given two based words $x_1 w x_1$ and $x_2 v x_2$, one can decide whether they represent freely homotopic loops in X. Moreover, if w and v have length ℓ_1 and ℓ_2, respectively, the decision can be performed by an algorithm which takes time $O(\ell_1 + \ell_2)$ on a RAM machine, where the linear constants depend on X, Y and Φ only.*

3.2 Base points and homotopies in the cubical complex X

Why did we pass to the fundamental groupoid, rather than sticking to the fundamental group? In other words, why do we pay so much attention to basepoint issues? By the way of motivation, let us look at a wrong "proof" of Theorem 25, and see how how we get into trouble if we don't make basepoints explicit at every step.

Wrong Claim Let α, β be two closed edge paths in X based at a common vertex x. Then the loops α and β represent conjugate elements of $\pi_1(X)$ if and only if the words $\Phi(\alpha)$ and $\Phi(\beta)$ represent conjugate elements of A.

Wrong proof of the Wrong Claim The implication "$\Rightarrow$" is obvious. For "$\Leftarrow$", we suppose that the words $\Phi(\alpha)$ and $\Phi(\beta)$ represent conjugate elements of $\pi_1(Y)$, so the loops $\Phi(\alpha)$ and $\Phi(\beta)$ in Y are freely homotopic. Thus we can apply sequences of free reductions, cyclings, and commutation relations (homotopies across squares) in Y to each of the two loops so as to transform both of them into some loop Γ in Y. By the injectivity- and convexity hypothesis, these transformations can be pulled back to free homotopies of the original loops α and β in X. *Therefore* α and β are both freely homotopic to some loop γ in X, i.e. they are freely homotopic. $\square$

This proof is almost correct, and our real proof of Theorem 25 shall follow this outline. The mistake, however, is the conclusion in the very last sentence: we can only conclude that α and β are freely homotopic to some loops γ and γ', respectively, where $\Phi(\gamma) = \Gamma = \Phi(\gamma')$. Intuitively, the loops γ and γ' in X may look like two different "liftings" of Γ, we did not pay attention to basepoints!

An explicit counterexample to the Wrong Claim illustrating the base point problem is given in Figure 7.

In order to prepare the proof of Theorem 25, let us study what homotopies of paths in X look like.

If α is an edge path in X giving rise to a based word $x_i w x_j$, and if $x_i \widetilde{w} x_j$ is a based word obtained from $x_i w x_j$ by one application of a commutation relation

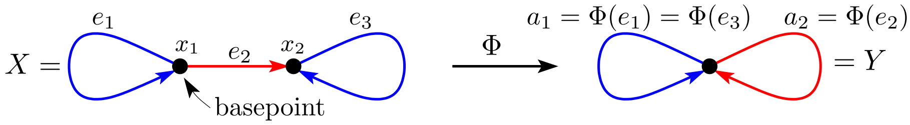

Figure 7: $\pi_1(Y)$ is the free group on two generators $a_1 = \Phi(e_1) = \Phi(e_3)$ and $a_2 = \Phi(e_3)$. The loops e_1 and $e_2e_3e_2^{-1}$ are not conjugate as elements of $\pi_1(X)$, whereas their images in $\pi_1(Y)$ are. In order to describe the loops in X it is better to use the based words $x_1a_1x_1$ and $x_1a_2a_1a_2^{-1}x_1$. The latter is conjugate to $x_2a_1x_2$.

(corresponding to a homotopy of a path in Y across a square) then there exists an edge path $\widetilde{\alpha}$ in X, starting from the same vertex as α and homotopic to α, which gives rise to the based word $x_i\widetilde{w}x_j$ – this is an immediate consequence of the convexity condition. Similarly, free cancellations in w can be realised by cancellations of backtracking path segments in α.

Let us summarize the situation in even more geometric language. Given a vertex x of X, it is in general not true that every loop in Y is the image under Φ of a path in X starting at x. However, when such a pullback of the loop exists, then it is unique. Moreover, in that case all homotopies of the loop in Y, except length-increasing ones, can be pulled back to based homotopies of the path in X.

Let us now look more generally at *free* homotopies of loops in X, i.e., homotopies that move the basepoint.

Definition 26 Suppose that xwx is a based word. A *parallel transport* of xwx is a replacement of the vertex x by a vertex x', where x' is obtained from x by walking along an oriented edge e with the property that the element $\Phi(e)$ of A commutes with all the generators of A occurring in the word w.

Geometrically, this move corresponds to replacing a closed path based at x by a parallel one based at x', where x and x' are joined by an edge e. The two paths together bound an annulus-shaped region of X. Notice that, under Φ, the two paths have the same image w in Y. Another way of moving the basepoint of a loop is to push it along the loop:

Definition 27 Suppose that $w = xy_1y_2\dots y_\ell x$ is a based word, and denote by e the unique edge of X that has one of its extremities equal to x and such that $\Phi(e) = y_1$. A *based cycling* of the based word w is its replacement by the word $x'y_2\dots y_\ell y_1x'$, such that the vertex x' is the second extremity of the edge e.

Geometrically, if α is a loop in X based at a vertex x, and described by a based word $xy_1y_2\dots y_\ell x$, and if we apply a cycling operation (in the sense of section 2) to the word $y_1y_2\dots y_\ell$, then this cycling can be pulled back to X to a based cycling of the based word, yielding a loop $\widetilde{\alpha}$, which looks exactly like α, except that it based at a different vertex x', "one notch further along the loop".

Example 28 In the example of Figure 7, we can apply a based cycling to the based word $x_1a_2a_1a_2^{-1}x_1$, yielding $x_2a_1a_2^{-1}a_2x_2$. After a cancellation, we obtain the based word $\gamma_2 = x_2a_1x_2$. We note that this is different from the based word $\gamma_1 = x_1a_1x_1$, which was also discussed in that example – in fact, the based words $x_2a_1x_2$ and $x_1a_1x_1$ are not even related by parallel transport (because $\Phi(e_2) = a_2$ does not commute with a_1). As we shall see in Lemma 31, this implies that the two loops e_1 and $e_2e_3e_2^{-1}$ are not freely homotopic in X.

Also note that a cyclic reduction of a word on the generators of A and their inverses can be decomposed as a cycling, followed by a usual cancellation of letters, and each of these operations can be pulled back to operations on the loop in X. Summarizing the last few paragraphs, we have the following

Key Observation 29 If α is a loop in X then all non-length-increasing free homotopies of the loop $\Phi(\alpha)$ in Y can be pulled back to free homotopies of α. Thus for a based word xwx, all cancellations, applications of commutation relations, cyclings, and cyclic reductions of the word w can be pulled back to analogue cancellations, commutation relations, and based cyclings of the based word. Similarly, if x_1wx_2 is a based word, and if the word w can be transformed into a word w' by applying cancellations and commutation relations, then $x_1w'x_2$ is again a based word.

3.3 The linear-time solution to the conjugacy problem

The aim of this subsection is to prove Theorem 25. We recall that we are considering two based words x_1wx_1 and x_2vx_2 representing two loops in X traversing ℓ_1 and ℓ_2 edges, respectively. A necessary condition for these loops being conjugate in the fundamental groupoid of X is that the words w_1 and w_2 represent conjugate elements of the right-angled Artin group A. In geometric terms, for the two loops to be freely homotopic in X, their images under Φ in Y must be freely homotopic. This is a condition which we can check in time $O(\ell_1+\ell_2)$ by the results of Section 2. However, this condition is not sufficient, as seen in Example 28. So let us now try to refine this approach.

Proposition 30 *There is an algorithm with running time $O(\ell_1 + \ell_2)$ whose input consists of two based words $x_1 w x_1$ and $x_2 v x_2$ of lengths ℓ_1 and ℓ_2, and which outputs*

(1) *either the information that they do not represent freely homotopic loops in X, or*

(2) *two based words $x_1' \widetilde{w}_1 \ldots \widetilde{w}_k x_1'$ and $x_2' \widetilde{w}_1 \ldots \widetilde{w}_k x_2'$, representing two loops in X which are respectively freely homotopic to the original two, and where the $\widetilde{w}_i$ are mutually commuting cyclic normal forms.*

Proof of Proposition 30 As seen in Section 2 we can decide in linear-time whether w and v represent conjugate elements of A. If they do not, then the two based words do not represent conjugate elements of the fundamental groupoid either, and it suffices to output this information (case (1)).

For the rest of the proof we have to deal with the case where w and v do represent conjugate elements of A.

We already know from Section 2 that the word w can, by a sequence of cancellations, commutation relations and cyclings be transformed into a word w' with the required decomposition $w' = w_1' \ldots w_k'$. Moreover, we know how to calculate the word w' in linear-time.

We also know from the Key Observation 29 above that the transformation of the word w into the word w' can be pulled back to a transformation of the based word $x_1 w x_1$ into a based word $x_3 w' x_3$. Our next task is to determine the corresponing base vertex x_3 in linear-time.

We shall fulfill this task by "carrying along information about the base vertex in X during the algorithm". While running the algorithm of Section 2, the only steps that affect the base vertex are the cyclings of pilings (including cyclic reductions of pilings, which can be decomposed as cyclings, followed by cancellations of tiles): when we cycle an $a_j^{\pm}$-tile, we have to determine how the base vertex is affected. However, this can be done simply by a lookup in a finite, precalculated list: for every vertex x of X, for every generator a_j of A, and for every $\epsilon \in \{-1, 1\}$, this list must tell us at which vertex of X we arrive if we pull back the loop $a_j^{\epsilon} \in A = \pi_1(Y)$ to a path in X starting at x (if that is possible). Since the algorithm of Section 2 performs a linearly bounded number of cyclings, we can calculate the new base vertex x_3 in time $O(\ell_1)$.

In a similar manner we can algorithmically transform the based word $x_2 v x_2$ into a word $x_2' \widetilde{w} x_2'$, where $\widetilde{w}$ is equipped with an analogue decomposition $\widetilde{w} = \widetilde{w}_1 \cdots \widetilde{w}_j$.

But since w and v represented conjugate elements of A, we have, by the results of Section 2, that the words w' and $\widetilde{w}$ are in fact the same, at least after a reordering of the factors of w' and a linearly bounded number of cyclings of each factor w'_i; in particular, we have $j = k$. Moreover, the Boyer-Moore algorithm tells us how many letters from each factor we have to cycle in order to achieve this. Thus we can transform the based word $x_3 w' x_3$ into the based word $x'_1 \widetilde{w} x'_1$ for some vertex x'_1, using a reordering of the factors (which does not affect the base vertex) and a linearly bounded number of based cyclings. □

Thus in order to prove Theorem 25, it is enough to prove it for the special case $v = w = \widetilde{w}_1 \dots \widetilde{w}_k$, where the words $\widetilde{w}_1, \dots, \widetilde{w}_k$ are mutually commuting cyclic normal forms. (For instance, this is the situation of Example 28, where we need to decide if the based words $x_1 a_1 x_1$ and $x_2 a_1 x_2$ represent freely homotopic loops in X.) For the rest of the proof of Theorem 25 we fix such a word $\widetilde{w}$, with such a decomposition.

Suppose a based word $x_1 \widetilde{u} x_2$ is such that $x_1 \widetilde{u} \widetilde{w} \widetilde{u}^{-1} x_1$ and $x_1 \widetilde{w} x_1$ represent the same element of the fundamental groupoid. Then in particular the elements of A represented by $\widetilde{u}$ and $\widetilde{w}$ commute: we have $[\widetilde{u}][\widetilde{w}][\widetilde{u}]^{-1} = [\widetilde{w}]$ in A.

As seen in Section 2.3, and using the notation of this section, the word $\widetilde{u}$ is equivalent to another word u of the form

$$u = z_1^{p_1} \dots z_k^{p_k}\, \zeta$$

where $p_1, \dots, p_k$ are integers and ζ is a word whose letters are generators of A which commute with, but are different from, all the generators occurring in w, and their inverses. We shall call such a word u a word in *preferred form*. We define the *norm* $\|u\|$ of u by

$$\|u\| = \sum_{i=1}^{k} |p_i| + \mathrm{length}(\zeta)$$

We are now ready state an algorithmically checkable criterion for $x_1 \widetilde{w} x_1$ and $x_2 \widetilde{w} x_2$ representing conjugate elements (i.e. representing freely homotopic loops in X):

Lemma 31 *The two based words $x_1 \widetilde{w} x_1$ and $x_2 \widetilde{w} x_2$ represent conjugate elements in the fundamental groupoid if and only if there exists a based word $x_1 u x_2$ such that u is a word in preferred form with*

$$\|u\| \leqslant \#\{\textit{vertices of } X\} \tag{1}$$

Proof We first suppose that an edge path $x_1 u x_2$ exists, where u is a word in preferred form. Then the word $u\widetilde{w}u^{-1}$ can be transformed into the word $\widetilde{w}$ by a finite number of commutation relations and cancellations (but no length-increasing transformations). By Key Observation 29, this homotopy can be pulled back to X, to yield a based homotopy between the paths in X represented by the based words $x_1 u\widetilde{w}u^{-1}x_1$ and $x_1\widetilde{w}x_1$. In other words, the elements $x_1\widetilde{w}x_1$ and $x_2\widetilde{w}x_2$ are conjugate, with conjugating element x_1ux_2.

Conversely, let us suppose that a conjugating element in the fundamental groupoid exists, and is represented by a based word $x_1\widetilde{u}x_2$. This means that there exists an edge path in X from x_1 to x_2 such that reading out the edge labels along the path yields the word $\widetilde{u}$. As seen before, $[\widetilde{u}]$ belongs to the subgroup of A generated the elements $[z_1], \dots, [z_k]$ and $[a_{j_1}], \dots, [a_{j_m}]$. Thus there is a word u in preferred form which can be obtained from $\widetilde{u}$ by a sequence of reductions and commutation relations. By Key Observation 29, x_1ux_2 is also a based word, i.e. it also represents an edge path in X.

We have shown the existence of a based word x_1ux_2 with u a word in preferred form, and without loss of generality we can suppose that u is chosen so that $\|u\|$ is minimal among all such based words.

Now for t in $\{0, \dots, \|u\|\}$ let us denote by $x(t)$ the vertex of X obtained by a walk in X starting at x_1 and following the edges of X according to the t first subwords. Now, if this function

$$\{0, 1, \dots, \|u\|\} \longrightarrow \{\text{vertices of } X\}\ , \quad t \mapsto x(t)$$

is not injective (for instance, if $\|u\|$ is larger than the number of vertices of X), then there exists a strictly shorter edge path in X represented by a based word $x_1u'x_2$ with u' also in preferred form, obtained by cutting out some segment of the previous edge path (c.f. the paragraph following Definition 24). This is in contradiction to the choice of u, and we can conclude that we have $\|u\| \leqslant \#\{\text{vertices of } X\}$ □

Let us now prove that the condition of Lemma 31 can be checked algorithmically in linear-time, i.e. in time $O(\ell)$, where ℓ is the length of the word w.

Firstly, recalling that the centralizer of $[\widetilde{w}]$ is generated by a finite number of elements (some of them represented by the words $z_1, \dots, z_k$ and the others equal to certain generators of A), we observe that there is a universal upper bound on the number of generators, namely the number of generators of A. Moreover, as seen in Proposition 23, words representing these generators can be determined in linear time.

Now there is a very simpleminded linear-time algorithm to check for the existence of a conjugating element: for *all* words u in preferred form satisfying condition (1) check whether $x_1 u x_2$ is a based word, i.e. whether there exists an edge path in X represented by the based word $x_1 u x_2$. Indeed, there is a universal bound on the number of words to be checked, and for each word u the check takes linear time (since the length of the words z_i can grow linearly with the length of $\widetilde{w}$).

Here is a summary of the whole algorithm: given two based words $x_* w x_*$ and $x_* v x_*$ representing loops α and β in X,

(1) Apply steps (i) and (ii) of the algorithm of Section 2.2, always carrying along the base vertex, to find graphs $\Delta_j(w)$ $(j = 1, \dots, k)$, $\Delta_j(v)$ $(j = 1, \dots, k')$, base vertices x_1, x_2, and based words $x_1 w_1 \dots w_k x_1$ and $x_2 v_1 \dots v_{k'} x_2$ representing loops that are freely homotopic to α and β.

(2) If $k \neq k'$, or if the collections of full subgraphs $\Delta_j(w)$ and $\Delta_j(v) \subset \Gamma$ are not the same, or if for some j between 1 and k the words v_j and w_j do not have the same length ℓ_j, return "NO".

(3) Apply step (iii)(a) of the algorithm of Section 2.2 to each of the k factors, always carrying along the base vertices, to transform $x_1 w_1 \dots w_k x_1$ into a based word $x_3 w'_1 \dots w'_k x_3$ and similarly $x_2 v_1 \dots v_k x_2$ into $x'_2 \widetilde{w}_1 \dots \widetilde{w}_k x'_2$, where all words w'_i and $\widetilde{w}_i$ are cyclic normal forms.

(4) For each factor, use a standard pattern matching algorithm to decide if $w'_i = \widetilde{w}_i$ as cyclic words. If no, return "NO". If yes, keep in mind how many cyclings of each factor w'_i are required to achieve equality $w'_i = \widetilde{w}_i$ as (non-cyclic) words.

(5) Perform the required *based* cyclings of $x_3 w'_1 \dots w'_k x_3$ to obtain a based word of the form $x'_1 \widetilde{w}_1 \dots \widetilde{w}_k x'_1$.

(6) Calculate the minimal roots z_i of the words $\widetilde{w}_i$, as explained in Section 2.3. Also determine the set of generators that commute with all the letters occurring in the words $\widetilde{w}_i$, but do not occur in any of them.

(7) Check, for all words u in preferred from satisfying condition (1), whether there exists an edge path in X represented by the based word $x'_1 u x'_2$. If for one of the words u the answer is affirmative, then return "YES". Otherwise return "NO".

Acknowledgement We thank Sam Sang-Hyun Kim, Tim Hsu, Lucas Sabalka, and Michah Sageev for interesting conversations.

References

[1] **A Abrams,** *Configuration spaces of colored graphs*, Geometriae Dedicata 92 (2002) 185–194.

[2] **A Abrams, R Ghrist,** *Finding topology in a factory: configuration spaces*, Amer. Math. Monthly 109(2) (2002) 140–150.

[3] **A Abrams, R Ghrist,** *State Complexes for Metamorphic Robots*, The International Journal of Robotics Research 23 (2004) 809–824.

[4] **A V Aho, J E Hopcroft, J D Ullman,** *The design and analysis of computer algorithms*, Addison-Wesley (1974).

[5] **A Baudisch**, *Kommutationsgleichungen in semifreien Gruppen,* Acta Math. Acad. Sci. Hungar. 29 (1977) 235–249.

[6] **A Baudisch**, *Subgroups of semifree groups,* Acta Math. Acad. Sci. Hungar. 38 (1981) 19–28.

[7] **R S Boyer, J S Moore,** *A fast string searching algorithm*, Comm. of the ACM 20 (1977) 762–772.

[8] **M R Bridson, A Haefliger,** *Metric spaces of non-positive curvature*, Springer Grundlehren Series, Vol. 319 (1999).

[9] **P Cartier, D Foata,** *Problèmes combinatoires de commutation et réarrangements*, Lecture Notes in Math. 85, Springer-Verlag (1969).

[10] **R Charney,** *An introduction to right-angled Artin groups*, Geometriae Dedicata 125 (2007) 141–158.

[11] **J Crisp, B Wiest,** *Embeddings of graph braid and surface groups in right-angled Artin groups and braid groups*, Algebr. Geom. Toplogy 4 (2004) 439–472.

[12] **J Crisp, B Wiest,** *Quasi-isometrically embedded subgroups of braid and diffeomorphism groups*, Trans. A.M.S. 359 (2007) 5485–5503.

[13] **C Duboc,** *Commutation dans les monoïdes libres, un cadre théorique pour l'étude du parallélisme*, Thèse de doctorat, Paris (1986).

[14] **D B A Epstein, D Holt,** *The linearity of the conjugacy problem in word-hyperbolic groups* preprint 2005.

[15] **D Farley, L Sabalka,** *Discrete Morse theory and graph braid groups,* Alg. and Geom. Topology 5 (2005) 1075–1109.

[16] **D Farley, L Sabalka,** *On the cohomology rings of tree braid groups*, preprint `arXiv:math.GR/0602444`.

[17] **R Grist, V Peterson,** *The geometry and topology of reconfiguration,* Advances in Applied Mathematics 38 (2007) 302–323.

[18] **D Gusfield,** *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology,* Cambridge University press (1997).

[19] **F Haglund, D Wise,** *Special cube complexes*, to appear in Geom. Funct. Anal.

[20] **D E Knuth, J H Morris, V R Pratt,** *Fast pattern-matching algorithms*, SIAM J Computing 6(2) (1977) 323–350.

[21] **D Krob, J Mairesse, and I Michos,** *Computing the average parallelism in trace monoids*, Discrete Math. 273 (2003) 131–162.

[22] **F Lalonde,** *Contribution à l'étude des empilements*, Doctoral thesis, Montréal (1990).

[23] **H-N Liu, C Wrathall, K Zeger,** *Efficient solution of some problems in free partially commutative monoids*, Information and Computation 89 (1990) 180–198.

[24] **L Sabalka,** *Embedding right-angled Artin groups into graph braid groups*, preprint `arXiv:math.GR/0506253`.

[25] **H Servatius,** *Automorphisms of Graph Groups*, J. Algebra 126 (1987) 34–60.

[26] **G Steven,** *String searching algorithms,* World Scientific (1994).

[27] **L Van Wyk,** *Graph groups are biautomatic*, J. Pure Appl. Algebra 94 (1994) 341–352.

[28] **X Viennot,** *Algèbres de Lie libre et monoïdes libres*, Lecture Notes in Math. 691, Springer-Verlag (1978).

[29] **C Wrathall,** *Free partially commutative groups*, in "Combinatorics, computing and complexity" (Ed. D-Z Du and G Hu) 195–216, Kluwer academic/Science press, Norwell, MA (1989).