

A polynomial time algorithm for the conjugacy problem in $\mathbb{Z}^n \rtimes \mathbb{Z}$

Bren Cavallo

CUNY Graduate Center
bcavallo@gc.cuny.edu

***Delaram Kahrobaei**

CUNY Graduate Center
and
New York City College of
Technology
dkahrobaei@gc.cuny.edu

*Corresponding author

Resum (CAT)

En aquest article introduïm un algorisme que, en temps polinomial, resol el problema de la conjugació (en les seves dues variants, de decisió i de cerca) per a grups de la forma lliure abelià per infinit cíclic, amb els inputs donats en forma normal. Fem això adaptant els resultats de Bogopolski–Martino–Maslakova–Ventura a [1] i de Bogopolski–Martino–Ventura a [2], als grups en qüestió i, en certs casos, usem un algorisme de Kannan–Lipton [7] per a resoldre el problema de l'òrbita a \mathbb{Z}^n en temps polinomial.

Abstract (ENG)

In this paper we introduce a polynomial time algorithm that solves both the conjugacy decision and search problems in free abelian-by-infinite cyclic groups, where the inputs are elements in normal form. We do this by adapting the work of Bogopolski–Martino–Maslakova–Ventura in [1] and Bogopolski–Martino–Ventura in [2], to free abelian-by-infinite cyclic groups, and in certain cases apply a polynomial time algorithm for the orbit problem over \mathbb{Z}^n given by Kannan–Lipton in [7].

Acknowledgement

The second author is partially supported by the Office of Naval Research grant N00014120758, the American Association for the Advancement of Science, a PSC-CUNY grant from the CUNY research foundation, as well as the City Tech foundation.

Keywords: *Conjugacy problem, semidirect product.*

MSC (2010): 20F10, 20E06.

Received: September 13, 2014.

Accepted: October 6, 2014.



1. Introduction

The conjugacy decision problem in a finitely presented group G , is determining if there is a solution to the equation $v = xux^{-1}$ where $u, v, x \in G$. The decision problem also has the search variant, given u and v conjugate, find an explicit x that conjugates u to v . The conjugacy decision problem is in general undecidable [8], whereas the search problem is decidable in every recursively presented group [9].

Due to the rise of applications of group theory to computer science and cryptography, more research has been directed towards studying the algorithmic complexity of group theoretic algorithms rather than solely investigating decidability. Other polynomial time algorithms for the conjugacy problem in solvable groups are due to Vassileva in free solvable groups [13] and Diekert–Miasnikov–Weiß in solvable Baumslag–Solitar groups [3]. Some very related results can also be seen in the work of Sale [11, 12], in which he shows that for a special class of the groups studied in this paper, the conjugacy length function is bounded from above by a linear function. Namely, for any two conjugate elements in these groups, there exists a conjugator of geodesic length less than a constant multiple of the sum of the geodesic lengths of the elements.

In the following sections we introduce a polynomial time algorithm that solves both the conjugacy decision and search problems in free abelian-by-infinite cyclic groups, where elements are given in terms of their normal forms. This family of groups is polycyclic so it is well known that they have a solvable conjugacy problem. This fact is due originally to Formanek [6] and Remesslennikov [10], who independently proved that virtually polycyclic groups are conjugacy separable: for any two $u, v \in G$ that are not conjugate, there exists a finite homomorphic image in which the images of u and v are not conjugate. Conjugacy can be solved in such groups by conjugating u by elements of G and checking if the result is v , while simultaneously enumerating all homomorphisms from G into a finite group and checking if the images of u and v are conjugate. One of the processes is guaranteed to stop which then provides an answer to the problem. This algorithm is brute force and clearly may take very long even in simple cases.

We start the paper with a review of free abelian-by-infinite cyclic groups and the twisted conjugacy problem. We then detail the algorithm due to Bogopolski–Martino–Maslakova–Ventura from [1] and prove that it, along with the solution to the orbit problem due to Kannan–Lipton [7], solves both the conjugacy decision and search problems in polynomially many steps with respect to the lengths of the inputs in normal form. Finally we end with a complexity analysis of the algorithm and discuss how the complexity changes when inputs are considered in their geodesic forms rather than normal forms.

2. Free abelian-by-infinite cyclic groups

We say that a group G is *free abelian-by-cyclic* if G fits into a short exact sequence of the form:

$$1 \rightarrow \mathbb{Z}^n \rightarrow G \rightarrow C \rightarrow 1,$$

where C is a cyclic group. If $C \simeq \mathbb{Z}$, then we say G is *free abelian-by-infinite cyclic*. In this case, G splits as $\mathbb{Z}^n \rtimes_{\phi} \mathbb{Z}$ for some $\phi \in \text{GL}_n(\mathbb{Z})$. Therefore, G has the presentation:

$$\langle g_1, g_2, \dots, g_n, t \mid tg_it^{-1} = \phi(g_i), [g_i, g_j] = 1 \rangle,$$

where $1 \leq i < j \leq n$ and where we view the g_i as the generators of \mathbb{Z}^n and t as the generator of \mathbb{Z} . As such, any $g \in G$ can be written as $w_1 t^{k_1} w_2 t^{k_2} \dots w_m t^{k_m}$ where each $w_i \in \mathbb{Z}^n$ and $k_i \in \mathbb{Z}$. Applying the relations of the form $tg_it^{-1} = \phi(g_i)$ multiple times, one can move all the t^{k_i} over to the right side of the

word, thus representing each element as wt^k where $w \in \mathbb{Z}^n$ and $k \in \mathbb{Z}$. For any $g \in G$ we call such a representative its *normal form*. Multiplication in normal forms can then be carried out as:

$$wt^k \cdot w't^{k'} = w\phi^k(w')t^{k+k'}.$$

Namely, every time we need to move t^k to the right, over a word in \mathbb{Z}^n , we can do so at the price of applying ϕ^k . It can additionally be seen (see [4]) that each group element's normal form is unique.

For the remainder of this paper, we will be working entirely with elements in their normal forms and as such assume in the following algorithm that elements are given in their normal form. We also define a length function, $|\cdot|$, over elements of G where if $g =_G wt^k$, then:

$$|g| = |wt^k| = |w|_{\mathbb{Z}^n} + |k|,$$

where $|w|_{\mathbb{Z}^n}$ is the standard geodesic length of $w \in \mathbb{Z}^n$.

3. The twisted conjugacy problem

Definition 3.1. Given a finitely presented group G , an automorphism $\phi \in \text{Aut}(G)$, and $u, v \in G$ we say u and v are *twisted conjugate* by ϕ if there exists $x \in G$ such that

$$v = xu\phi(x^{-1}).$$

If u and v are twisted conjugate by ϕ we write: $u \sim_\phi v$.

Notice that the standard conjugacy problem is a special case of the twisted conjugacy problem by taking ϕ to be the identity.

In [1] Bogopolski–Martino–Maslakova–Ventura introduced an algorithm that relates the conjugacy problem in free-by-infinite cyclic groups to the twisted conjugacy problem in free groups. Following that work, Bogopolski–Martino–Ventura [2] adapted the algorithm from [1] to solve the conjugacy problem in a variety of groups created by extensions. What follows is an adaptation of their algorithm for free abelian-by-cyclic groups.

4. The algorithm

The following lemma and proof is taken directly from the beginning of section 2 in [1] and adapted to free abelian-by-infinite cyclic groups.

Lemma 4.1. Let $u = wt^s$ and $v = xt^r$ in $\mathbb{Z}^n \rtimes_\phi \mathbb{Z}$ be conjugate. Then $s = r$ and there exists $e \in \mathbb{Z}$ such that $\phi^e(w) \sim_{\phi^s} x$ in \mathbb{Z}^n . Additionally, if $\phi^s = \phi^r$ is the identity, then $x = \phi^e(w)$ for some $e \in \mathbb{Z}$.

Proof. Let $a = bt^e \in \mathbb{Z}^n \rtimes_\phi \mathbb{Z}$ be such that $v = aua^{-1}$. Therefore,

$$xt^r = (bt^e)wt^s(bt^e)^{-1} = bt^e wt^s t^{-e} b^{-1} = b\phi^e(w)t^s b^{-1} = b\phi^e(w)\phi^s(b^{-1})t^s.$$

As such, we have $xt^r = b\phi^e(w)\phi^s(b^{-1})t^s$, which implies $s = r$ and $\phi^e(w) \sim_{\phi^s} x$ by b . □

Given u and v as above, the lemma shows that there are two cases one must consider to solve the conjugacy decision and search problems in \mathbb{Z}^n -by- \mathbb{Z} groups. First, check if $s = r$. If not, then u and v are not conjugate. If the exponents are the same, then there are two cases:

- If ϕ^s is trivial, we have to decide whether $\exists e \in \mathbb{Z}$ such that $x = \phi^e(w)$.
- Otherwise, we have to decide if there exists e such that $\phi^e(w) \sim_{\phi^s} x$.

The first case, namely given two vectors $w, x \in \mathbb{Z}^n$ and $\phi \in \text{GL}_n(\mathbb{Z})$ determine if there exists $e \in \mathbb{Z}$ such that $x = \phi^e(w)$, is known as the orbit problem over \mathbb{Z}^n . In [7], Kannan–Lipton provide a polynomial time algorithm that solves the orbit problem over \mathbb{Q}^n . Since the orbit problem over \mathbb{Z}^n is a special case of their work, this algorithm provides a polynomial time solution to the twisted conjugacy problem over \mathbb{Z}^n in the case that ϕ^s is trivial. If such an e is found satisfying the orbit problem, then we have that $v = t^e u t^{-e}$.

For the second case, we use the fact from the lemma that $\exists b \in \mathbb{Z}^n, e \in \mathbb{Z}$ such that $x = b\phi^e(w)\phi^s(b^{-1})$. Before we begin the algorithm, we state [1, Lemma 1.7].

Lemma 4.2. *For any group G , $\phi \in \text{Aut}(G)$, and $u \in G$, $u \sim_{\phi} \phi(u)$.*

Proof. $\phi(u) = u^{-1}u\phi(u)$. Therefore u is twisted conjugate over ϕ to $\phi(u)$, by u^{-1} . \square

As such, $\phi^e(w) \sim_{\phi^s} \phi^{e \pm ks}(w)$ for any $k \in \mathbb{Z}$. Therefore, if there exists an e that satisfies the equation $\phi^e(w) \sim_{\phi^s} x$, then we can find such an e among $\{0, 1, \dots, |s| - 1\}$. This is where it is important that we are in the second case and so, $s \neq 0$.

We can now proceed with the full algorithm. Due to the fact that $x, w \in \mathbb{Z}^n$ and $\phi \in \text{GL}_n(\mathbb{Z})$ it is more convenient to put the equation $x = b\phi^e(w)\phi^s(b^{-1})$ into additive notation. As such we write $x = b + \phi^e(w) - \phi^s(b)$. This gives the equation

$$x - \phi^e(w) = (\text{Id}_n - \phi^s)b,$$

where Id_n is the $n \times n$ identity matrix. In this way, each e yields a system of linear equations, which we solve for the vector b . There will be a solution to the conjugacy problem, as long as there is some e for which the solution b is in \mathbb{Z}^n . Moreover, we know that if there is a solution to the conjugacy problem, such an e must lie in the set $\{0, 1, \dots, |s| - 1\}$. If there exists such an e , $u \sim v$ and bt^e is a conjugator. As such, we proceed by solving the system of linear equations given by each of the possible e 's and then checking if the solution, b , is in \mathbb{Z}^n . In the case that $\text{Id}_n - \phi^s$ is invertible, namely, ϕ^s does not have 1 as an eigenvalue, then we can also write:

$$b = (\text{Id}_n - \phi^s)^{-1}(x - \phi^e(w)).$$

For a complete description of the algorithm in pseudo-code on inputs $wt^s, xt^r \in \mathbb{Z}^n \rtimes_{\phi} \mathbb{Z}$, see Algorithm 1. We have the algorithm return **FALSE** if the elements are not conjugate, and a conjugating element if they are.

5. Complexity analysis

In the algorithm above we have two cases each of which can be dealt with in polynomially many steps with respect to n and $|s|$. If $s = r \neq 0$, we find solutions of an $n \times n$ linear system at most $|s|$ times. On the

Algorithm 1 Conjugacy Algorithm for $\mathbb{Z}^n \rtimes_{\phi} \mathbb{Z}$

```

if  $s \neq r$  then
  return FALSE
else if  $\phi^s$  is the identity then
  Run Kannan-Lipton algorithm.
  if Kannan-Lipton returns  $k$  then
    return  $t^k$ 
  else return FALSE
  end if
else
   $e := 0$ 
  while  $e < |s|$  do
    if  $\exists b \in \mathbb{Z}^n$  such that  $x - \phi^e(w) = (\text{Id}_n - \phi^s)b$  then
      return  $bt^e$ 
    else  $e := e + 1$ 
    end if
  end while
  return FALSE
end if

```

other hand, if $s = r = 0$, we use Kannan–Lipton algorithm, which runs in polynomial time. Therefore, this algorithm is at most polynomial in terms of n and the lengths of the input words.

It is worth pointing out that unlike many of the algorithms group theorists study, this algorithm takes as inputs words in their polycyclic normal forms as opposed to in their geodesic form or just in any general form. This affects the complexity of the algorithm as all forms have different lengths. It is worth noting that the geodesic form of a word in a polycyclic group can be logarithmic with respect to the length in normal form. For instance in the group:

$$G = \mathbb{Z}^2 \rtimes_{\phi} \mathbb{Z} = \langle g_1, g_2, t \mid [g_1, g_2], tg_1t^{-1} = g_1^2g_2, tg_2t^{-1} = g_1g_2 \rangle,$$

where $\phi(t) = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$, we have that:

$$t^n abt^{-n} = a^{F(2n+2)} b^{F(2n+1)},$$

where $F(n)$ is the n -th element of the Fibonacci sequence $F = \{1, 1, 2, 3, 5, \dots\}$. In this way, normal forms in G can be exponentially longer than their geodesic forms. As such, collecting words in geodesic form and then performing the algorithm would take an exponential number of steps with respect to the geodesic length since the process of collecting involves writing out a word that is exponentially longer than the original word. On the other hand, in a practical setting, converting words to normal forms is fast (see [5]) and the main complexity involved in the algorithm has to do with the exponent above the generator t after collection, which is just the sum of the exponents above the t 's in a general word. As such, after collection, the exponent above t contributes to the length of the word at most what it contributed prior to collection. In that vein, even though a word may grow in size exponentially after collection, most of the additional steps are involved in collection rather than in actually solving the conjugacy problem.

Acknowledgements. We would like to thank the Universitat Politècnica de Catalunya where much of this research was conducted and our host Enric Ventura. Finally, we would like to thank the referee for his/her comments.

References

- [1] O Bogopolski, A Martino, O Maslakova, and E Ventura. The conjugacy problem is solvable in free-by-cyclic groups. *Bulletin of the London Mathematical Society*, 38(05):787–794, 2006.
- [2] Oleg Bogopolski, Armando Martino, and Enric Ventura. Orbit decidability and the conjugacy problem for some extensions of groups. *Transactions of the American Mathematical Society*, 362(4):2003–2036, 2010.
- [3] Volker Diekert, Alexei Miasnikov, and Armin Weiß. Conjugacy in Baumslag’s group, generic case complexity, and division in power circuits. *arXiv preprint arXiv:1309.5314*, 2013.
- [4] B. Eick. *Algorithms for Polycyclic Groups*. Habilitationsschrift, Universität Kassel, 2001.
- [5] Bettina Eick and Delaram Kahrobaei. Polycyclic groups: A new platform for cryptology? *arXiv preprint math/0411077*, 2004.
- [6] Edward Formanek. Conjugate separability in polycyclic groups. *Journal of Algebra*, 42(1):1–10, 1976.
- [7] Ravindran Kannan and Richard J Lipton. Polynomial-time algorithm for the orbit problem. *Journal of the ACM (JACM)*, 33(4):808–821, 1986.
- [8] Charles F Miller III. Decision problems for groups, survey and reflections. In *Algorithms and classification in combinatorial group theory*, pages 1–59. Springer, 1992.
- [9] Alexei Myasnikov, Vladimir Shpilrain, and Alexander Ushakov. *Group-based Cryptography*. Springer, 2008.
- [10] Vladimir N Remeslennikov. Conjugacy in polycyclic groups. *Algebra and Logic*, 8(6):404–411, 1969.
- [11] Andrew W Sale. Short conjugators in solvable groups. *arXiv preprint arXiv:1112.2721*, 2011.
- [12] Andrew W Sale. Conjugacy length in group extensions. *arXiv preprint arXiv:1211.3144*, 2012.
- [13] Svetla Vassileva. Polynomial time conjugacy in wreath products and free solvable groups. *Groups, Complexity, Cryptology*, 3(1):105–120, 2011.