Satisfiability of Equations in Free Groups is in PSPACE

Claudio Gutiérrez*
Computer Science Group, Dept. Mathematics, Wesleyan University, U.S.A.
and
Departamento de Ingeniería Matemática, Universidad de Chile, Chile
cgutierrez@wesleyan.edu

ABSTRACT

We prove that the computational complexity of the problem of deciding if an equation in a free group has a solution is PSPACE.

The problem was proved decidable in 1982 by Makanin, whose algorithm was proved later to be non primitive recursive: this was the best upper bound known for this problem. Our proof consists in reducing equations in free groups to equations in free semigroups with antiinvolution, and presenting an algorithm for deciding equations in free semigroups with antiinvolution.

1. INTRODUCTION

Let $\Sigma = \{a_1, \dots, a_n\}$ be an alphabet. An *equation* in the free group G generated by Σ with unknowns $x_1, \dots, x_m$ is an equality of the form $w(x_1, \dots, x_m, a_1, \dots, a_n) = 1$, where w is a word formed from the letters $x_1, \dots, x_m, a_1, \dots, a_n$ and their inverses. A *solution* of such an equation is a list $v_1, \dots, v_m$ of words in $a_1, \dots, a_n, a_1^{-1}, \dots, a_n^{-1}$ such that $w(v_1, \dots, v_m, a_1, \dots, a_n) = 1$ in the group G. In this paper we prove that the problem of deciding if such an equation has a solution is in PSPACE.

In the early 60's Markov, studying algorithmic problems of semigroups and groups, posed the following question: Is there an algorithm for solving arbitrary equations in free groups? (or in unification language: is the unification problem for groups decidable?). This problem and the related one for free semigroups has lately attracted much attention from the theoretical computer science community, see for example [2], [8], [9], [3], [16], [17], [18]. Special particular cases were answered positively by Lyndon [12], Lorents [10], Kmelevskii [6], [7]. In 1982 Makanin [14] (corrections in [15]) presented an algorithm that solves the general case, still the only one known. Koscielski and Pacholski [9], by showing that 'contrary to the common belief' this algorithm is not primitive recursive, stated the current upper bound for this problem. As for lower bounds, Durnev [2] showed a NP-hard lower bound. On related algorithmic aspects of equations on free groups we can mention the work of Razborov [19] who presented an algorithm for generating all the solutions to a given group equation, and Durnev [2] which proves the undecidability of several related problems.

Summarizing, the current complexity of the problem of satisfiability of equations in free groups is between NP-hard (see [2]) and PSPACE (this paper).

Overview of the paper

In [4] we reduced the problem of satisfiability of equations in free groups to that of satisfiability of equations in a simpler theory, namely in free *semigroups with antiinvolution* (SGA), via a PSPACE translation. In this paper we prove that satisfiability of equations in free SGA is in PSPACE, hence giving a PSPACE upper bound for the case of free groups. The theory SGA is 'in between' that of semigroups and groups, and is defined by the equations $x(yz) = (xy)z$, $(xy)^{-1} = y^{-1}x^{-1}$ and $(x^{-1})^{-1} = x$. A free SGA over the set Σ is the set of words over the alphabet $\Sigma \cup \{a^{-1} : a \in \Sigma\}$ together with an operator $(\ )^{-1}$ which reverses a word and changes the exponent of the base letters.

Makanin in [14] reduces satisfiability of equations in free groups to the satisfiability of a special kind of equations in free SGA, namely those whose solutions are non-contractible. A *contractible* word is, roughly speaking, one which does not contain any factor of the kind cc^{-1} or $c^{-1}c$ for c constant. Then he applies to these special equations a methodology similar to that of his famous previous algorithm on word equations by defining generalized equations and the corresponding transformations.

We followed a different path, whose schema can be summarized as follows:

1. Reduce satisfiability of equations in groups to satisfiability of equations in SGA with non-contractible solution.

   Claim 1: For each equation E in free groups we get a set of equations $E'_1, \dots, E'_m$ in SGA such that E has a solution iff one of the E'_i has a non-contractible solution.

   (This first step is the same as in Makanin [14]; from here on, the approaches differ completely.)

2. Reduce satisfiability of equations in free SGA with non-contractible solutions to satisfiability of equations in free SGA (i.e. no restriction on the solutions).

   Claim 2: For each equation E' in free SGA there is

*Partially supported by FONDAP, Matemáticas Aplicadas.

STOC 2000 Portland Oregon USA

a set $E_1'', \dots, E_k''$ of equations in free SGA such that E' has a non-contractible solution iff one of E_j'' has a (ordinary) solution.

3. Generalize the method used in [18] for deciding satisfiability of word equations to a method for deciding satisfiability of equations in free SGA.
Claim: Satisfiability of equations in free SGA is in PSPACE.

The size of the set of equations in Step 1 is exponentially bigger than the size of E. Same for Step 2. The good news is that they can be generated non-deterministically in polynomial space. So we can conclude that satisfiability of equations in free groups is in NPSPACE, hence in PSPACE. As we said, Claim 1 is in Makanin's paper [14]. We proved Claim 2 in [4]. From these proofs it is straightforward to conclude that these sets can be generated non-deterministically in polynomial space. For the sake of completeness, we will state the relevant theorems from these papers in the Appendix.
What remains is Claim 3, which is what we essentially present in this paper. This generalization follows the seminal Plandowski's paper [18], and is a combinatorial proof. The idea is to define a non-deterministic transformation $\longrightarrow$ among equations which preserves satisfiability. The algorithm consists in generating non-deterministically equations from the simple (satisfiable) equation $(c = c)$ for a constant c. The difficult part is to prove that this process can be done in polynomial space. The reader familiar with [18] will recognize our indebtedness to that paper.

2. PRELIMINARIES AND NOTATIONS

2.1 Equations in SGA

A *semigroup with anti-involution* (SGA) is an algebra with a binary associative operation (written as concatenation) and a unary operation $(\ )^{-1}$ with the equational axioms

$$(xy)z = x(yz) \quad (1)$$
$$(xy)^{-1} = y^{-1}x^{-1} \quad (2)$$
$$x^{-1-1} = x. \quad (3)$$

A *free* semigroup with anti-involution is an initial algebra for this variety. It is not difficult to check that for a given alphabet A, the set of words over $A \cup A^{-1}$ together with the operator $(\ )^{-1}$, which reverses a word (changing also the exponent of the letters), is a free algebra for SGA over A.

2.1.1 Equations and solutions

Let Σ and V be two disjoint alphabets of constants and variables respectively. Denote by $\Sigma^{-1} = \{c^{-1} : c \in \Sigma\}$. Similarly for V^{-1}. An *equation* E in free SGA with constants Σ and variables V is a pair (w_1, w_2) of words over the alphabet $\mathcal{A} = \Sigma \cup \Sigma^{-1} \cup V \cup V^{-1}$. The number $|E| = |w_1| + |w_2|$ is the *length* of the equation. These equations are also known as *equations in a paired alphabet.*
A map $h : V \longrightarrow (\Sigma \cup \Sigma^{-1})^*$ can be uniquely extended to a SGA-homomorphism $\bar{h} : \mathcal{A}^* \longrightarrow (\Sigma \cup \Sigma^{-1})$ by defining $h(c) = c$ for $c \in \Sigma$ and $h(u^{-1}) = (h(u))^{-1}$ for $u \in \Sigma \cup V$. We will use the same symbol h for the map h and the SGA-homomorphism $\bar{h}$. A *solution* h of the equation E is (the unique SGA-homomorphism defined by) a map $h : V \longrightarrow \Sigma \cup \Sigma^{-1}$ such that $h(w_1) = h(w_2)$. The length of the solution h is $|h(w_1)|$. By $h(E)$ we denote the word $h(w_1)$ (which is the same as $h(w_2)$).
The exponent of periodicity of a word w is the maximal integer p such that $w = xy^pz$ for x, y, z words and y non-empty. By the exponent of periodicity of a solution h we mean the exponent of periodicity of $h(E)$. The next is an important theorem.

THEOREM 1. *Let E be an equation in free SGA. Then, the exponent of periodicity of a minimal solution of E is bounded by $2^{\mathcal{O}(|E|)}$.*

PROOF. The proof is a straightforward generalization to SGA of the result proved in [8] for words; a sketch of the proof can be found in [4]. □

2.2 Sequences of words

Given sequences $S_1 = w_1, \dots, w_n$, $S_2 = v_1, \dots, v_m$ of elements of Σ^*, the composition S_1, S_2 denotes the sequence $w_1, \dots, w_n, v_1, \dots, v_m$. In general, for S_i sequences of Σ, we define inductively $S_1, \dots, S_n$ as the composition of the sequence $S_1, \dots, S_{n-1}$ with S_n. By S^t we denote the sequence $S, \dots, S$ consisting of t repetitions of S. Also $S^{-1} = w_n^{-1}, \dots, w_1^{-1}$.
Given a sequence $S = w_1, \dots, w_n$, we will give special names to the following objects: $\text{conc}(S) = w_1 \cdots w_n$, $\text{length}(S) = n$; $\text{first}(S) = w_1$; $\text{last}(S) = w_n$; $\ker(S) = w_2, \dots, w_{n-1}$ if $\text{length}\, S > 2$, otherwise $\ker(S) = \epsilon$. If R is another sequence, then the *substitution* in S of w_j by R is the composition of the sequences $w_1, \dots, w_{j-1}, R, w_{j+1}, \dots, w_n$. The sequence S is a *refinement* of R if $\text{conc}(S) = \text{conc}(R)$ and there are indices $i_1 < \cdots < i_k$ such that

$$R = \text{conc}(w_1, \dots, w_{i_1}), \text{conc}(w_{i_1+1}, \dots, w_{i_2}), \dots \\ \dots, \text{conc}(w_{i_k+1}, \dots, w_n).$$

2.2.1 Exponential expressions

Given a word w and a positive integer t, we will denote the sequence $w, \dots, w$ (t-times) by w^t. If we extend the definition of sequence allowing these kind of expressions we get what is called an *exponential expression.* So we can codify sequences by *exponential expressions* in the obvious way. For example $ab, ab, ab, ab, a, a, a, b$ can be codified as $(ab)^4, a^3, b$, etc. The *height* of an expression is defined recursively as follows: $\text{height}(w) = 0$ for a word w, $\text{height}(S_1, S_2) = \max(\text{height}\, S_1, \text{height}\, S_2)$ and $\text{height}(S^t) = 1 + \text{height}(S)$. We will deal most of the time with sequences of height no bigger than 1. The *size* of an exponential expression is defined as follows: $s(w) = 1$ for a word w, $s(S_1, S_2) = s(S_1) + s(S_2)$ and $s(S^t) = 1 + s(S)$.
For our purposes what will be important are not the particular words in a sequence, but the pattern of their occurrences. So we define two exponential expressions S, R to be *isomorphic* if for the sequences they represent, say $w_1, \dots, w_m$ and $v_1, \dots, v_n$ respectively, it holds $m = n$ and there is a bijection $\varphi : \{w_1, \dots, w_m\} \longrightarrow \{v_1, \dots, v_n\}$ such that $v_i = \varphi(w_i)$ and $\varphi(w^{-1}) = (\varphi(w))^{-1}$. The following Lemma is due to Plandowski [18]:

LEMMA 1. *The isomorphism of two exponential expressions of polynomial size can be checked in polynomial time.*

2.3 Facts from word combinatorics

Given a word w, the subword starting at position i and ending at position j is denoted by $w[i,j]$; we will write $w[i]$ for $w[i,i]$. A *period* of w is a number p such that for all i, $w[i] = w[i+p]$ whenever both sides are defined.
The following result uses essentially a well known result by Fine and Wilf about periodicity, and appears in [18]:

LEMMA 2. *Let $i < j < k$ be three consecutive starting positions of occurrences of a word v in w. If $i + |v| \geq k$ then $k - j = j - i$ and $k - j$ is a period of a word $w[i, k + |v| - 1]$.*

The following is an easy result on conjugate words, see e.g. [11]:

LEMMA 3. *If $u_1 w = w u_2$ then there are words v_1, v_2 such that $u_1 = v_1 v_2$ and $u_1 w = v_1(v_2 v_1)^m$ for some integer m.*

3. FACTORIZATIONS

DEFINITION 1. *1. A* factorization *$F(w)$ of a word w is a sequence of non-emtpy words*

$$F(w) = w_1, w_2, \ldots, w_n \tag{4}$$

such that $w = \mathrm{conc}(w_1, \ldots, w_n)$.

2. For positions $1 \leq i,j \leq |w|$ of w, we define the partition $F(w)[i,j]$, the restriction *of the partition $F(w)$ to $w[i,j]$, as follows:*

$$F(w)[i,j] = w[i, p_{s+1} - 1], w_{s+1}, \ldots, w_f, w[p_{f+1} - 1, j],$$

where $p_1 < \cdots < p_k$ are the starting positions of $w_1, \ldots, w_n$ in the factorization (4), that is $p_j = |w_1 \cdots w_{j-1}| + 1$, and s, f are the subindices such that $p_s \leq i < p_{s+1}$ and $p_f < j \leq p_{f+1}$.

We will be mostly interested in the following kind of factorizations:

DEFINITION 2 (D-FACTORIZATION). *Let D be a set of words of the same even length $2t > 0$ and w any word. Let $1 \leq p_1 < \cdots < p_k < |w|$ be the set of starting positions of all the occurrences of words of D in w. Let $v_j = w[p_j, p_j + 2t - 1]$ for $j = 1, \ldots, k$.*

*1. The D-*factorization *of w is defined as:*

$$F_D(w) = w[1, p_1 + t - 1], w[p_1 + t, p_2 + t - 1], \ldots \\ \ldots, w[p_k + t, |w|]. \tag{5}$$

If no word of D occurs in w, then we define $F_D(w) = w$.

2. For each $j (1 \leq j < k)$, the pair of words v_j, v_{j+1} determine the factor $u_j = w[p_j + t, p_{j+1} + t - 1]$ of (5). The triple (u_j, v_j, v_{j+1}) is called the extended factor *of the factor u_j.*

For the cases $u_0 = w[1, p_1 + t - 1]$ and $u_k = w[p_k + t, |w|]$, the extended factor is defined as $(u_0, \$, v_1)$ and $(u_k, v_k, \$)$ respectively, where $\$$ is a new symbol. If $F_D(w) = w$ then $(w, \$, \$)$ is its extended factor.

3. For a subsequence S of $F_D(w)$, we will denote by $(S)^e$ the sequence of extended factors obtained from S by replacing each factor by its extended factor.

Remark. The factorization $F_D(w)$ above factors w along the boundaries marked by the 'middle' of the words in D, hence we need words of even length. (In [18] the beginning of the words signal the marks for the factorization.) Typically D will be a finite set of words of the same even length closed under converse, *i.e.*, if $w \in D$ then $w^{-1} \in D$.

LEMMA 4. *Let D be a set of words of the same length $2t$. Let $i < j < k$ be starting positions of three consecutive occurrences of a word $v \in D$ in w such that $i + 2t \geq k$. Then*

$$(F_D(w)[i+t, j+t-1])^e = (F_D(w)[j+t, k+t-1])^e.$$

PROOF. Along the same lines as in [18]. By Lemma 2, $k - j = j - i$ and $k - j$ is a period of $u = w[i, k + 2t - 1]$. It is enough to prove that for $0 \leq p < j - i$ the words of length $2t$ starting at positions $i + t + p$ and $j + t + p$ in w are identical. This is true because these two words are wholy contained in u and the distance between their occurrences in u is equal to $j - i$ which is a period of u. □

LEMMA 5. *Let D be a set of words of the same length $2t$. Let $i < k$ be occurrences of two words $u, v \in D$ in a word w. Assume that $i + 2t \geq k$.*
Then $(F_D(w)[i+t, k+t-1])^e$ can be represented by an exponential expression of size $\mathcal{O}(|D|^2)$.

PROOF. Along the same lines as in [18]. □

The key point in Lemma 5 is the fact that the size of the expression *does not depend* on t, but only on the size of the set D.

LEMMA 6. *Let D be a set of words of the same length $2t$.*

1. *If $\ker F_D(w[i,j])$ is empty, then $((F_D w)[i,j])^e$ can be represented by an exponential expression of size $\mathcal{O}(|D|^2)$.*
2. *If $\ker F_D(w[i,j])$ is not empty, then*

$$((F_D(w))[i,j])^e = (R_1)^e, (\ker F_D(w[i,j]))^e, (R_2)^e$$

where R_1^e and R_2^e can be represented by exponential expressions of size at most $\mathcal{O}(|D|^2)$, and $\mathrm{conc}(R_1) = \mathrm{first}(F_D(w[i,j]))$ and $\mathrm{conc}(R_2) = \mathrm{last}(F_D(w[i,j]))$.

PROOF. The factorization $F_D(w)$ of w and $F_D(w[i,j])$ of $w[i,j]$ are based on occurrences of the words of D in w and $w[i,j]$, respectively. $(F_D(w))[i,j]$ differs from $F_D(w[i,j])$ on possible occurrences of words from D which either cover the positions i or j in w. Apply then Lemma 5. □

We will need also the following result in the case of words with converse:

LEMMA 7. *Let D be a set of words of the same length $2t$ closed under converse, and w a word such that $\ker F_D(w)$ is not empty. Then if*

$$(\ker F_D(w))^e = (w_1, v_{11}, v_{12}), \ldots, (w_n, v_{n1}, v_{n2})$$

it holds

$$(\ker F_D(w^{-1}))^e = (w_n^{-1}, v_{n2}^{-1}, v_{n1}^{-1}), \ldots, (w_1^{-1}, v_{12}^{-1}, v_{11}^{-1}).$$

PROOF. Just note that if there is a word u in D and $u = w[p - t + 1, p + t]$, then u^{-1} is also in D and $u^{-1} = w^{-1}[|v| - p - t + 1, |v| - p + t]$. The result follows then immediately. □

4. FACTORIZATIONS OF SOLUTIONS OF EQUATIONS

From now on we are going to fix a satisfiable equation $E = (u, v)$ in free SGA and a minimal solution h of it. Denote $|E| = |u| + |v|$. A *boundary* of a word w is a pair $(p, p+1)$ of consecutives positions. By extension we define $(0, 1)$ and $(|w|, |w|+1)$ as the initial and final boundaries respectively. Note that for each boundary $(p, p+1)$ of u (resp. v) there is a unique 'image' boundary in $h(u)$ (resp. $h(v)$), namely $(q, q+1)$, where $q = |h(u[1, p])|$, which is called a *cut* of h. Because $h(u) = h(v)$ there are no more than $|E|$ cuts.

The following proposition about cuts is a straightforward generalization for free SGA of the similar result for words due to Rytter and Plandowski [16]. The proof can be found in [4].

PROPOSITION 1 (LEMMA 2 IN [4]). *Assume S is a minimal (w.r.t. length) solution of E. Then*

1. *For each subword $w = h(E)[i, j]$ with $|w| > 1$, there is and occurrence of w or w^{-1} which contains a cut of h which is neither the initial nor the final boundary of that occurrence.*
2. *For each letter $c = h(E)[i]$ of $h(E)$, there is an occurrence of c or c^{-1} in E.*

We will need D-factorizations with a special set D as introduced in the next definition.

DEFINITION 3 (THE SET OF WORDS D_l). *For each natural number $l \geq 1$ define, from E and h, the set D_l of words as follows: $w \in D_l$ if and only if either*

1. *$w = h(u)[q - l + 1, q + l]$ for some cut $(q, q + 1)$ of h.*
2. *w is the converse of a word in (1).*

These sets D_l (parameterized by $l \geq 1$) are going to play a key role in what follows. Observe that $|D_l| \leq 2|E|$.

Notation. Given a word w, if no confusion arises, we will write $F_l w$ for the factorization $F_{D_l}(w)$.

We will prove next that the factors in $F_l h(u)$ have a small representation.

LEMMA 8. *Each factor in $F_l h(u)$ is of the form*

$$w_1 w_2^p w_3$$

where $|w_1|, |w_2|, |w_3| < 2l|E|$ and $p \in 2^{\mathcal{O}(|E|)}$.

PROOF. The factorization of $F_l h(u)$ is determined by occurrences of words of D_l in $h(u)$. Consider a factor w of $F_l h(u)$, and w.l.o.g suppose $|w| > 6ln$, and let $w = h(u)[i, j]$. By definition of factorization, $h(u)[i - l, i + l - 1]$ and $h(u)[j - l + 1, j + l]$ are in D_l and there are no other occurences of words of D_l in $h(u)[i - l, j + l]$.

By Proposition 1, w or w^{-1} has an occurrence over a cut; w.l.o.g. suppose that w occurs over a cut in $h(u)$. The cut divides w into w', w'', and $|w'| < l$ or $|w''| < l$ (otherwise $h(u)[i + |w'| - l, i + |w'| + l - 1] \in D_l$ and w would not be a factor).

Suppose $|w''| < l$. Then consider $w_1 = w'$ and by Proposition 1, w_1 has an occurrence over a cut. The cut divides w_1 into w_1', w_1'', and $|w_1'| < l$ or $|w_1''| < l$.

Continue on for $4|E| + 1$ steps. Because there are no more than $|E|$ cuts in h, there must be two indices $i_0 < j_0 \leq 4|E| + 1$ such that w_{i_0} and w_{j_0} (or $w_{i_0}^{-1}$ and $w_{j_0}^{-1}$) hit the same cut, say $(q, q+1)$ of $h(u)$, and either $|w_{i_0}'|, |w_{j_0}'| < l$ or $|w_{i_0}''|, |w_{j_0}''| < l$. Suppose w.l.o.g. that w_{i_0} and w_{j_0} hit the same cut, and $|w_{i_0}'|, |w_{j_0}'| < l$ (see Figure 1). We know that

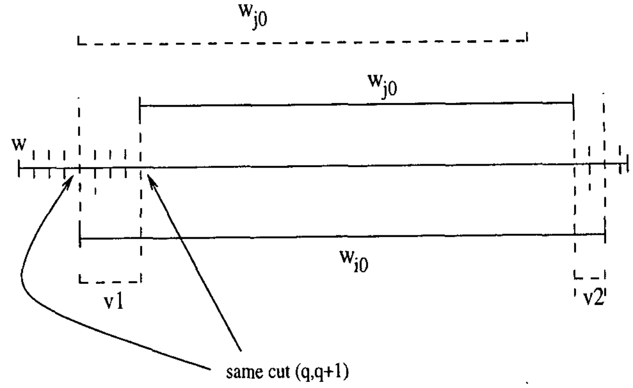

Figure 1: Visualization of proof of Lemma 8.

$w_{i_0} = v_1 w_{j_0} v_2$ and $|v_1| < (j_0 - i_0)l$ and also that

$$v_1 w_{j_0} = h[q + 1, q + 1 + |v_1 w_{j_0}|] = w_{j_0} v_1'$$

for some v_1, v_1'. Then by Lemma 3, $w_{j_0} = u_0(v_0 u_0)^p$ for certain $p \geq 0$, and $|v_0 u_0| = |v_1|$. The statement of the lemma follows from the fact that h is a minimal solution, hence by Proposition 1, $p \leq 2^{c|E|}$, and so can be encoded by $c|E|$ bits. $\square$

LEMMA 9. *Let w be a factor of $F_{l+1} h(u)$. Then the following hold:*

1. *It is refined in $F_l h(u)$ by a sequence of factors S and $(S)^e$ can be represented by an exponential expression of size $\mathcal{O}(|E|^3)$.*
2. *Moreover, any two occurrences of w in $F_{l+1} h(u)$ which have the same extended factor are refined in $F_l h(u)$ by the same sequence of extended factors.*

PROOF. Part 1. By Lemma 8, $w = w_1 w_2^p w_3$ with $|w_i| \leq 2l|E|$ and p can be encoded by $c|E|$ bits. First, let us remark that the proof of Lemma 11 in [18] works for the general case $|w| < (a|E| + b)l + c$, where a, b, c are positive integers. We are going to use this case below. If $|w_2^p| \leq 2l$ then $|w| \leq (4|E| + 2)l$, and proceed as in the proof of Lemma 11 in [18]. Otherwise, let us write $w_2^p = v_1 v_2 v_3$ with $|v_1| = |v_3| = l$; so $w = w_1 v_1 v_2 v_3 w_2$.

Because $|w_1 v_1| \leq 2|E|l + 1$ and $|v_3 w_3| \leq 2|E|l + 1$, we can apply Lemma 11 in [18] to these pieces. As for v_2, we can write $v_2 = z_1 w_2^{p'} z_2$ with $|z_i| < |w_2|$. The key point now is the observation that the D_l-factors of v_3 are periodic: If certain word of D_l occurs in v_3 determining a boundary in certain copy of w_2, then that same word of D_l determines a boundary in each copy w_2 in v_3 (thus the choice of v_1, v_2). Hence the extended factorization of the middle part $w_2^{p'}$ is just the extended factorization of any copy of w_2 raised to the power p'. Because $p' \leq p \leq 2^{c|E|}$ and $|w_2| \leq 2|E|l$, it follows that can be represented in space $\mathcal{O}(|E|^3)$.

For Part 2, just notice that both occurrences of w must occur inside identical contexts $w_1 w w_2$ with $|w_i| = l + 1$. $\square$

5. FACTOR EQUATIONS

It will be convenient to view free SGA equations as sequences of words instead of words themselves. So for example, the equation $(xay^{-1}, abxx)$ can be thought of as the pair of sequences $(x, a, y^{-1}), (a, b, x, x)$. A *factor equation* is a pair (U, V) of sequences of of elements of $(\Sigma^* \cup V)$. A solution is an assignment $h : \mathcal{V} \longrightarrow \mathcal{S}$, where $\mathcal{S}$ is the set of sequences of elements of Σ^* such that the substitution $h(x)$ for the variables x occurring in U or V make both sequences equal (i.e. both sequences have same i-th factors). Two factor equations (U_1, V_1) and (U_2, V_2) are *isomorphic* if the sequence $U_1, =, V_1$ is isomorphic to $U_2, =, V_2$, where '$=$' is a new symbol.

Notice that a free SGA equation over Σ is naturally a factor equation over Σ: the sequences built by transforming the pair of words into a pair of sequences (each symbol is transformed into an element of the sequence). In what follows we will talk only of factor equations, and identify a free SGA equation (via the above inclusion) with the corresponding factor equation.

Let us recall some facts which will be useful in what follows. $E = (u, v)$ denotes a satisfiable free SGA equation, and h a minimal solution of it. Let us assume that $u = u_1 \cdots u_n$ and $v = v_1 \cdots v_m$, for $u_i, v_j \in \Sigma \cup V$. If u_k is a variable or the inverse of a variable, say x, and $i \leq j$ are such that $h(u_k) = h(u)[i, j]$, then from Lemma 6 we know that if $\ker F_l h(x)$ is not empty,

$$((F_l h(u))[i, j])^e = (R_1)^e, (\ker F_l h(x))^e, (R_2)^e \qquad (6)$$

where $(R_1)^e, (R_2)^e$ can be represented by exponential expressions of size $\mathcal{O}(|D|^2)$. In particular, $(\ker F_l h(x))^e$ is the same sequence for all occurrences of the variable x in E. Also, from Lemma 7 we know that if $h(x)$ and $h(x^{-1})$ occur in $F_l h(u)$,

$$(\ker F_l\ h(x^{-1}))^e = ((\ker F_l\ h(x))^{-1})^e. \qquad (7)$$

DEFINITION 4 (FACTOR EQUATIONS $E_l(h)$). *Let $l \geq 1$ be an integer, and E and h as before.*

1. *For each extended factor (w, v_1, v_2) in $(F_l h(u))^e$ define a fresh constant $c_{(w,v_1,v_2)}$. Also, if $(w, v_1, v_2) \neq (w^{-1}, v_2^{-1}, v_1^{-1})$ and both occur both in $(F_l h(u))^e$, identify the constants $c_{(w^{-1},v_2^{-1},v_1^{-1})}$ and $c^{-1}_{(w,v_1,v_2)}$.*

2. *Let (w, v_1, v_2) be an extended factor in $(F_l h(u))^e$. Define the map $(\ )^*$ as follows:*

   (a) *$(w, v_1, v_2)^* = c_{(w,v_1,v_2)} c^{-1}_{(w,v_1,v_2)}$ if $(w, v_1, v_2) = (w^{-1}, v_2^{-1}, v_1^{-1})$,*

   (b) *$(w, v_1, v_2)^* = c_{(w,v_1,v_2)}$ otherwise.*

3. *Define U_l as follows (the case for V_l is similar): consider the extended factorization $(F_l h(u))^e$. Note that for each symbol u_k of u which is a variable (say x) and $\ker F_l h(x)$ is not empty, $(\ker F_l h(x))^e$ occurs as a subsequence of $(F_l h(u))^e$. Then U_l is built from $(F_l h(u))^e$ by replacing each such subsequence by the one-element sequence consisting of the corresponding variable x.*

4. *Define U_l^* from U_l by replacing each element (w, v_1, v_2) of U_l which is not a variable by $(w, v_1, v_2)^*$. Similarly for V_l^*.*

Then define $E_l(h)$ as the pair (U_l^*, V_l^*).
For $l = 0$ we make the convention that $U_0 = u_1, \ldots, u_n$ and $V_0 = v_1, \ldots, v_n$, i.e., $E_0(h)$ is E.

LEMMA 10. *Same notations as before. For each integer $l \geq 0$ it holds:*

1. *$E_l(h)$ is satisfiable.*
2. *$E_l(h)$ can be represented by an exponential expression of size $\mathcal{O}(|E|^3)$.*

PROOF. For Part 1 consider the map h' defined for constants as $h'(c) = c$, and for variables as $h'(x) = ((\ker F_l h(x))^e)^*$, that is $(\ker F_l h(x))^e$ with $(\ )^*$ applied to each component. Observe that $((\ker F_l h(x))^e)^*$ does not depend on the occurrence of the variable x. Also $h'(x) = (h'(x^{-1}))^{-1}$ follows from (7) and the identification of some constants in part 1 of Definition 4. Finally it is clear from the definition of $E_l(h)$ that $h'(U_l^*) = h'(V_l^*)$.

For Part 2 just note that U_l consists of: (1) possibly all the constants of u (no more than those of E), and (2) the extended factorization of each $h(u_j)$ for u_j variables, with $(\ker F_l h(u_j))^e$ replaced by one symbol when it is not empty. Then use Equation (6) and Lemma 6 to conclude that U_l can be represented by an exponential expression of size $\mathcal{O}(|E|^3)$. Hence U_l^*, V_l^* have also a small representation. $\square$

DEFINITION 5 (NON-DETERM. TRANSFORMATION $\longrightarrow$). *Let E_1, E_2 be exponential expressions representing factor equations. Then $E_1 \longrightarrow E_2$ if and only if E_2 is isomorphic to an equation obtained from E_1 as follows*

1. *Replace constants of E_1 by exponential expressions of size $\mathcal{O}(|E|^3)$ with exponents at most $2^{c|E|}$ consistently, i.e., if $a := exp_a$ then $a^{-1} := (exp_a)^{-1}$.*

2. *Suppose u is a variable and S_1, S_2 sequences. If all occurrences of u (resp. u^{-1}) in E_1 are in the context of subsequences of the form S_1, u, S_2 (resp. $S_2^{-1}, u^{-1}, S_1^{-1}$) and they do not overlap, then replace all occurrences of S_1, u, S_2 (resp. $S_2^{-1}, u^{-1}, S_1^{-1}$) by u (resp. u^{-1}).*

3. *Replace some occurrences of a subsequence S by a new variable y and S^{-1} by y^{-1} (for the same variables, the sequences replaced should be the same).*

LEMMA 11. *Let E_1, E_2 be exponential expressions representing factor equations. If $E_1 \longrightarrow E_2$ and E_1 is satisfiable, then E_2 is satisfiable.*

PROOF. Let h_1 be a solution of E_1. For each constant a, denote by exp_a the expression which replaces a in Step 1 of the definition of $\longrightarrow$. This replacement defines a morphism σ with $\sigma(w^{-1}) = (\sigma(w))^{-1}$ such that $\sigma(a) = exp_a$ for each constant a of E_1.

Denote by S_1^x, S_2^x the exponential expressions introduced in Step 2 of the definition of $\longrightarrow$ for the variable x. Denote by P^y the sequence of constants which is replaced by a new variable y. Then it is not difficult to check that

$$h_2(x) = \begin{cases} S_1^x, \sigma(h_1(x)), S_2^x & \text{if } x \text{ occurs in } E_1 \text{ and } E_2 \\ P^y & \text{if } y \text{ does not occur in } E_1 \end{cases}$$

is a solution of E_2. Observe that $h_2(x^{-1}) = (h_2(x))^{-1}$ because $(\sigma h_1(x))^{-1} = \sigma(h_1(x^{-1}))$ and $P^{y^{-1}} = (P^y)^{-1}$. $\square$

PROPOSITION 2. *For each integer $l \geq 0$, it holds $E_{l+1}(h) \longrightarrow E_l(h)$.*

PROOF. For $l = 0$, it is easy to check that $E_1(h) \longrightarrow E_0(h)$.
For $l \geq 1$, by Lemma 9, $(F_l h(u))^e$ can be got from $(F_{l+1} h(u))^e$ by replacing each extended factor of the sequence $(F_{l+1} h(u))^e$ by a sequence of extended factors representable by an exponential expression of size $\mathcal{O}(|E|^3)$, and moreover (Part 2 of the lemma) two factors with identical extended factors are replaced by the same sequence of extended factors.
Now recall that U_{l+1} differs from $(F_{l+1} h(u))^e$ in that for each occurrence of a variable x in E with $\ker_{l+1} h(x)$ not empty, the corresponding occurrence of $(\ker_{l+1} h(x))^e$ is replaced by x. Also recall that if $\ker_l h(x)$ is not empty, then $(\ker_l h(x))^e$ is a subsequence S of $(F_l h(u))^e$. Moreover, if additionally $\ker_{l+1} h(x)$ is not empty, then $(\ker_l h(x))^e = S_1, (K)^e, S_2$ where K is a refinement of $\ker_{l+1} h(x)$ and S_1, S_2 are some sequences of extended factors.
From these facts, it is clear that $E_{l+1}(h) \longrightarrow E_l(h)$ is got by doing the steps 1,2,3 (in that order) in Definition 5. □

Remark. Observe that if E'_l (resp. E'_{l+1}) is an exponential expression representing $E_l(h)$ (resp. $E_{l+1}(h)$), then $E'_{l+1} \longrightarrow E'_l$. (Same replacements, sequences, etc. used in $E_{l+1}(h) \longrightarrow E_l(h)$ work here.)

LEMMA 12. *$(a = a) \longrightarrow^* E$ if and only if E is satisfiable.*

PROOF. If $E = (u, v)$ is satisfiable, let h be a minimal solution. Then $E_{|h(u)|}(h) \longrightarrow^* E_0(h)$ by Proposition 2, and observe that $E_{|h(u)|}(h)$ is isomorphic to $(a = a)$ and $E_0(h)$ is E.
If E is not satisfiable, then from Lemma 11 it follows that the equation $(a = a)$, which is trivially satisfiable, cannot rewrite to E. □

THEOREM 2. *Satisfiability of equations in free SGA is in PSPACE.*

PROOF. Consider the space M of exponential expressions of size $\mathcal{O}(|E|^3)$ representing factor equations. Consider $(a = a)$ and apply nondeterministically $\longrightarrow$. That the algorithm is correct follows from Lemma 12 and the fact that the chain $(a = a) \longrightarrow^* E$ can be done in M, which follows from Lemma 10, Part 2, the remark above, and Lemma 1. □

6. SATISFIABILITY OF EQUATIONS IN FREE GROUPS

As we mentioned in the introduction, the first step to decide satisfiability of equations in free groups is the following reduction:

THEOREM 3 (THEOREM 9, [4]). *For each equation E in a free group G with generators C there is a finite set Q of equations in a free semigroup with anti-involution G' with generators $C \cup \{c_1, c_2\}$, $c_1, c_2 \notin C$, such that the following hold:*

1. *E is satisfiable in G if and only if one of the equations in Q is satisfiable in G'.*
2. *There is $c > 0$ constant such that for each $E' \in Q$, it holds $|E'| \leq c|E|^3$.*

This theorem is proved in [4]. For the sake of completeness we will indicate the steps of the proof given in [4] using the Propositions in the Appendix.

1. From E generate a finite list of system of equations in SGA with properties as in Proposition 4.
2. From each of these systems, using Proposition 3 build a non-contractible equation in SGA.
3. From each non-contractible equation got in (2), generate a list of systems of equations in SGA with properties as of Proposition 5.
4. Again use Proposition 3 to obtain from each system in (3) and equivalent equation in SGA.

Remark. The equations in the set Q can be generated non-deterministically in polynomial space.
Finally the main result of this paper:

THEOREM 4. *Satisfiability of equations in free groups is in PSPACE.*

PROOF. The algorithm works as follows: From an equation E generate non-deterministically an SGA-equation E' in the set Q (as in Theorem 3). Then use Theorem 2. □

After Theorem 4, the current complexity of the problem of satisfiability of equations in free groups is between NP-hard (see [2]) and PSPACE (this paper).

6.1 Comparison with other work

The only published upper bound on the complexity of equations in free groups is [9], which is non primitive recursive. The problem of equations in free SGA was stated in [4], where the problem about its decidability is asked. It seems that nothing was known before about this problem. Diekert and Hagenah [1] have recently proved independently of us its decidability. The lower bound NP-hard is proved in [4]. Theorem 2 gives a tight upper bound. As for the methodology in proving Theorem 2, Theorem 1 generalizes [8], and Lemmas 4, 5, 6, 8, 10, 9, 11, 12 and Prop. 2 have their counterparts in [18].

Acknowledgements. The people of DIM, Universidad de Chile, especially Marcos Kiwi and Martín Matamala, heard a first version of this paper. Useful remarks where also provided by Dan Dougherty and Volker Diekert. Discussions with Alexander Razborov led to several improvements. Thanks to them all.

7. REFERENCES

[1] V. Diekert, Personal communication, 8 Oct. 1999.

[2] V. Durnev, *Studying Algorithmic Problems for Free Semi-groups and Groups*, Proceedings of the 4th International Symposium in Logical Foundations of Computer Science, Yaroslavl 1997. LNCS 1234.

[3] C. Gutiérrez, *Satisfiability of Word Equations with Constants is in Exponential Space*, in Proceedings FOCS'98, IEEE Computer Soc. Press, Palo Alto, California, 1998.

[4] C. Gutiérrez, *Equations in free Semigroups with anti-involution and their relation to equations in free Groups*, Proceedings of Latin American Theoretical INformatics, LATIN'2000, to appear in LNCS 1776. Also available as Technical Report MA-99-B-474, Departamento de Ingeniería Matemática, Universidad de Chile; and in `http://www.wesleyan.edu/~cgutierrez/stoc00.ps`

[5] Y.I. Kmelevskii, *Equations in a free semigroup*, Trudy Mat. Inst. Steklov 107(1971); English trans. Proc. Steklov Inst. Math. 107(1971).

[6] Y.I. Kmelevskii, *Systems of equations in a free group I*, Izv. Akad. Nauk. SSSR Ser. Mat. 35(1971), 1237-1268; English trans. in Math USSR Izv. 5(1971).

[7] Y.I. Kmelevskii, *Systems of equations in a free group II*, Izv. Akad. Nauk. SSSR Ser. Mat. 36(1972), 110-179; English trans. in Math USSR Izv. 6(1972).

[8] A. Kościelski, L. Pacholski, *Complexity of Makanin's algorithm*, J. Assoc. Comput. Mach. 43 (1996) 670-684.

[9] A. Kościelski, L. Pacholski, *Makanin's algorithm is not primitive recursive*, Theoretical Computer Science 191 (1998) 145-156.

[10] A.A. Lorents, *Representation of solution sets of systems of equations with one unknown in free groups*, Dokl. Akad. Nauk SSSR 178(1968), 290-292; English trans. in Soviet Math. Dokl. 9 (1968).

[11] Lothaire, M. *Combinatorics on Words*, Cambridge Mathematical Texts, reprinted 1998.

[12] R.C. Lyndon, *Equations in free groups*, Trans. Amer. Math. Soc. 96(1960), 445-457.

[13] G.S. Makanin, *The problem of solvability of equations in a free semigroup*, Mat. Sbornik 103, 147-236 (in Russian). English translation in Math. USSR Sbornik 32, 129-198.

[14] G.S. Makanin. *Equations in a free group*, Izvestiya NA SSSR 46(1982), 1199-1273; English translation in Math USSR Izvestiya, 21 (1983), 483-546.

[15] G.S. Makanin. *Decidability of the universal and positive theories of a free group*, Izvestiya NA SSSR 48(1984), 735-749; English translation in Math USSR Izvestiya, 25 (1985), 75-88.

[16] W. Rytter and W. Plandowski, *Applications of Lempel-Ziv encodings to the solution of word equations*, In Proceedings of the 25th. ICALP, 1998.

[17] Plandowski, W., *Satisfiability of word equations with constants is in NEXPTIME*, in Proc. STOC'99.

[18] Plandowski, W., *Satisfiability of word equations with constants is in PSPACE*, in Proc. FOCS'99.

[19] A.A. Razborov, *On systems of equations in a free group*, Izvestiya AN SSSR 48 (1984) 779-832 (in Russian). English translation in Math. USSR Izvestiya 25 (1985) 115-162.

Appendix

The first proposition is an old observation of Kmelevskii [5] for free semigroups which extends easily to free SGA:

PROPOSITION 3 (PROPOSITION 4, [4]). *For each system of equations Σ in free SGA with generators C, there is an equation E in free SGA with generators $C \cup c$, $c \notin (C \cup C^{-1})$, such that*

1. *S is a solution of E if and only if S is a solution of Σ.*
2. *$|E| \leq 4|\Sigma|$.*

Moreover, if the equations in Σ are non-contractible, the E is non-contractible.

PROPOSITION 4 (LEMMA 1.1 IN [14]). *For any non contractible equation E in the free group G with generators C we can construct a finite list of systems of non-contractible equations in the free SGA G' with generators C $\Sigma_1, \ldots, \Sigma_k$ such that the following conditions are satisfied:*

1. *E has a non-contractible solution in G if and only if $k > 0$ and some system Σ_j has a non-contractible solution in G'.*
2. *There is a constant $c > 0$ such that $|\Sigma_i| \leq c|E|^3$ for each $i = 1, \ldots, k$.*
3. *$k \leq 2^{c|E|^3}$ for some constant $c > 0$.*

PROPOSITION 5 (PROPOSITION 3, [4]). *For each non contractible equation E there is a finite list of systems of equations $\Sigma_1, \ldots, \Sigma_k$ such that the following conditions hold:*

1. *E has a non-contractible solution if and only if some of the Σ_i has a solution.*
2. *$k \leq 2^{c|E|^2}$, for $c > 0$ a constant.*
3. *There is a constant $c > 0$ such that for each i, $|\Sigma_i| \leq c|E|$.*