Satisfiability of word equations with constants is in PSPACE

Wojciech Plandowski
Institute of Informatics, Warsaw University,
Banacha 2, 02-097, Warsaw, Poland.
E-mail: wojtekpl@mimuw.edu.pl.
Supported by KBN grant 8 T11C 039 15.

Abstract

We prove that satisfiability problem for word equations is in PSPACE.

1. Introduction

Satisfiability problem for word equations has a simple formulation: Find out whether or not an input word equation has a solution. The decidability of the problem was proved by Makanin [11]. His decision procedure is one of the most complicated algorithms existing in the literature. There were several attempts to simplify it [2, 12]. The algorithm has been implemented [1]. During last 20 years its complexity has been improved several times: 4-NEXPTIME (composition of four exponential functions) [7, 17], 3-NEXPTIME [9], 2-EXPSPACE [4], EXPSPACE [8]. The exact complexity of the algorithm is still not known. Current version of the algorithm, the full version of which can be found in [5], is still very complicated.

Recently, another algorithm has been proposed in [15]. It works nondeterministically in time polynomial in $n \log N$ where n is the size of an input equation and N is the size of a minimal solution of the input equation. It is much more simple than Makanin's algorithm but its full version requires an estimation for N and the algorithm in [13]. Till this year the only estimation for N was a consequence of Makanin's algorithm. The one which can be concluded from the last version of the algorithm is triple exponential (Corrollary 1 in [8]). Very recently a double exponential estimation for N has been proved [14]. The proof does not use Makanin's approach but uses the optimal bound for the index of periodicity of a minimal solution [9]. With this estimation the algorithm in [15] places the problem in NEXPTIME.

We propose the third algorithm. The full version of the algorithm requires only a proof of the upper bound for index of periodicity of a minimal solution [9]. Our algorithm is the first one which is proved to work in polynomial space.

A lower bound for the problem is NP [3, 9]. The best algorithms for NP-hard problems run in single exponential deterministic time. Each algorithm in PSPACE can be implemented to run also in single exponential deterministic time. So our algorithm is optimal unless faster algorithms are developed for NP-hard problems.

An important generalization of satisfiabitity problem for word equations is the problem of satisfiability of word equations with regular constraints. It consists in deciding whether a word equation e has a solution h such that $h(X) \in L_1$, $h(Y) \in L_2$, ... where X, Y ... are variables of e and L_1, L_2 ... are given regular languages. The decidability of the problem has been proved in [17]. The best upper bound so far is NEXPTIME [6]. The proof is based on extension the reasoning of [14, 15]. The lower bound for the problem is PSPACE-hard. This follows from the fact that the emptiness problem of an intersection of several regular languages is PSPACE-hard [10]. Using the tools of this paper one can prove.

Theorem 1 ([16]). *The problem of satisfiability of word equations with regular constraints is PSPACE-complete.*

2. The algorithm - overview

Given a word equation e we construct a relation $\rightarrow$ on "short" equations. The relation satisfies

$$(a = a) \rightarrow^* e \text{ iff } e \text{ has a solution (Lemma 14).}$$

where $\rightarrow^*$ is a transitive and reflexive closure of $\rightarrow$.

Our algorithm is nondeterministic:

eq:= $(a = a)$.
while $(eq \neq e)$ **do**
    nondeterministically generate next equation eq'
        such that $eq \rightarrow eq'$;
    eq:=eq'

The algorithm is in NPSPACE since the equations are "short" i.e. of polynomial length with respect to the length of e and because the generation of a next equation can be done in NPSPACE. Since NPSPACE=PSPACE the result follows.

3. Preliminaries

The length of a word w is denoted by $|w|$. A subword of w starting at position i and ending at position j is denoted by $w[i..j]$. A period of a word is a number p such that for all i, $w[i] = w[i+p]$ whenever both sides of the equation are defined. A fundamental result dealing with periods is the following.

Proposition 2 (Fine and Wilf). *Let p, q be two periods of a word w. If $p + q \leq |w|$ then $gcd(p, q)$ is also a period of w, where gcd stands for the greatest common divisor.*

We say that a word v *occurs* in a word w at position p if $w = w_1 v w_2$ for some words w_1, w_2 such that $p = |w_1| + 1$. In this case we will say that p is a *starting position of an occurrence of the word v in w*. We say that an occurrence of v at position p in w *covers* a position k in w if $p \leq k \leq p + |v|$. We will use the following property of words.

Lemma 3 ([14]). *Let $i < j < k$ be three consecutive starting positions of occurrences of a word v in w. If $i + |v| \geq k$ (ie. each of the occurrences of v covers the position k in w) then $k - j = j - i$ and $k - j$ is a period of a word $w[i..k + |v| - 1]$.*

Let Σ and Ξ be two disjoint alphabets: alphabet of constants and alphabet of variables. A *word equation* is a pair of words (u, v) (usually denoted by $u = v$) over the alphabet $\Sigma \cup \Xi$. A *length* of a word equation $e : u = v$ is defined by $|u| + |v|$ and denoted by $|e|$. A *solution* of the word equation is a morphism $h : (\Xi \cup \Sigma)^* \to \Sigma^*$ such that $h(a) = a$, for $a \in \Sigma$, and $h(u) = h(v)$. Note that a morphism being a solution of a word equation is uniquely determined by its values on variables. A solution h of $u = v$ is *minimal* if for all solutions g of $u = v$, $|h(u)| \leq |g(u)|$. There can be several minimal solutions of an equation.

An *index of periodicity* of a word w is a maximal number $p(w)$ such that $u^{p(w)}$, for a nonempty word u, is a subword of w. An *index of periodicity $p(h)$ of a solution h* of $u = v$ is $p(h(u))$.

Proposition 4 ([9]). *There is a constant c such that for each minimal solution h of a word equation e, $p(h) \leq 2^{c|e|}$.*

The original proof of Proposition 4 uses slightly different definition of a minimal solution and of an index of periodicity but the same proof works also with our definitions.

A cardinality of a set S is denoted by $|S|$.

4. Quasi-factorizations

A *factorization of a word* w is a sequence of nonempty words $w_1, w_2, \ldots, w_k$ such that $w = w_1 w_2 \ldots w_k$. The words w_i are called *factors* of $\mathcal{P}$. Factorization of a word is not able to distinguish between two factors which are the same words. This is why we introduce quasi-factorizations. A *quasi-factorization of a word* w with indices I is a sequence of pairs

$$(w_1, i_1), (w_2, i_2), \ldots (w_k, i_k)$$

such that $w = w_1 w_2 \ldots w_k$, and, for $1 \leq s \leq k$, $i_s \in I$, and w_s is a nonempty word. The pairs (w_s, i_s) are called *quasi-factors*. The *length* of a quasi-factor (w, i) is $|w|$.

Since we deal with sequences we introduce some notation dealing with them. For a sequence $\mathcal{P} = a_1, \ldots a_k$, the number k is the *length* of $\mathcal{P}$ and is denoted by $\mathrm{length}(\mathcal{P})$. Denote by $\mathrm{last}(\mathcal{P})$ the last element of $\mathcal{P}$ and by $\mathrm{cut\_last}(\mathcal{P})$ the sequence $\mathcal{P}$ without last element of $\mathcal{P}$. For a positive integer t denote by $(\mathcal{P})^t$ a sequence consisting of t repetitions of $\mathcal{P}$. An *exponential expression* is an expression on sequences in which the only allowed operations are repetition and catenation of sequences. Exponential expression represents a sequence. The *size* of an exponential expression is its denotational length, e.g. the size of the expression $(a, b, c)^{100}$, b is 10 (we assumme that the denotational length of an exponent and an element of the sequence are one) although it represents a sequence of length 301. A *height* of an exponent expression is defined recursively. For an element a we define $\mathrm{height}(a) = 0$. For sequences $\mathcal{P}$, $\mathcal{R}$ and an exponent t we define

$$\begin{aligned} \mathrm{height}(\mathcal{P}, \mathcal{R}) &= \max(\mathrm{height}(\mathcal{P}), \mathrm{height}(\mathcal{R})), \\ \mathrm{height}((\mathcal{P})^t) &= \mathrm{height}(\mathcal{P}) + 1 \end{aligned}$$

The expressions we consider are mostly of height one. When we write of the expressions of other height we say their height. Usually we identify exponential expression with the sequence it represents.

In our considerations a *quasi-factorization* $\mathcal{F}$ is a function which takes a word and returns a quasi-factorization of this word. Denote by $\mathcal{F}(w)[i, j]$ a sequence of words which is obtained from $\mathcal{F}(w)$ by cutting off the part of it which corresponds to $w[i..j]$. More precisely, let $\mathcal{F}(w) = (w_1, j_1), \ldots, (w_k, j_k)$ and let $i_t = |w_1 w_2 \ldots w_{t-1}| + 1$ for $1 \leq t \leq k + 1$ (i_t, for $t < k + 1$, is a starting position of an occurrence of the word w_t in w). Let s, r be such that $i_r \leq i < i_{r+1}$ and $i_s \leq j < i_{s+1}$. If $s = r$ then $\mathcal{F}(w)[i, j] = (w[i..j], j_r)$. Otherwise,

$$\begin{aligned} \mathcal{F}(w)[i, j] = {} & (w', j_r), (w_{r+1}, j_{r+1}), (w_{r+2}, j_{r+2}), \\ & \ldots, (w_{s-1}, j_{s-1}), (w'', j_s) \end{aligned}$$

where w' is a suffix of w_r of length $i_{r+1}-i$ and w'' is a prefix of w_s of length $j-i_s+1$. We say that the subsequence $(w_r, j_r), (w_{r+1}, j_{r+1}), \ldots, (w_s, j_s)$ of $\mathcal{F}(w)$ (which is usually different from $\mathcal{F}(w)[i,j]$) *corresponds* to the interval $[i,j]$ in $\mathcal{F}(w)$.

Let $\mathcal{D}$ be a set of words of the same length. *A $\mathcal{D}$-factorization* is a quasi-factorization with indices $\mathcal{D} \cup \{\$\}$ which is defined as follows. (The symbol $\$$ is a special symbol which does not occur in $\mathcal{D}$.) If no word of $\mathcal{D}$ occurs in a word w, then $\mathcal{D}(w) = (w, \$)$. Otherwise, let $i_1 < i_2 < \ldots < i_k$ be the set of all starting positions of occurrences of the words of $\mathcal{D}$ in w and let $u_1, \ldots u_k$ be the words in $\mathcal{D}$ which occur at these positions. Then

$$\begin{aligned}\mathcal{D}(w) &= (w[1..i_1-1], u_1), \ldots, \\ &\quad (w[i_{k-1}, i_k-1], u_k), (w[i_k..|w|], \$).\end{aligned}$$

Lemma 5. *Let $\mathcal{D}$ be a set of words of the same length t. Let $i<j<k$ be three consecutive occurrences of a word $v \in \mathcal{D}$ in a word w. Assume that $i+t \geq k$. Then $\mathcal{D}(w)[i, j-1] = \mathcal{D}(w)[j, k-1]$.*

Proof. By Lemma 3, $k-j = j-i$ and $k-j$ is a period of $u = w[i..k+t-1]$. It is enough to prove that for $0 \leq p < j-i$ the words of length t starting at positions $i+p$ and $j+p$ in w are identical. This is true since 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 6. *Let $\mathcal{D}$ be a set of words of the same length t. Let $i<k$ be occurrences of two words $v, u \in \mathcal{D}$ in a word w. Assume that $i+t \geq k$. Then $\mathcal{D}(w)[i, k-1]$ can be represented by an exponential expression of size $O(|\mathcal{D}|^2)$.*

Proof. Denote $\mathcal{P} = \mathcal{D}(w)[i, k-1]$. Let $i_1 = i < i_2 < \ldots < i_s = k$ be all starting positions of occurrences of words of $\mathcal{D}$ in w such that $i \leq i_r \leq k$. Note that for three consecutive occurrences of the same word in these occurrences Lemma 5 becomes applicable. Scan occurrences from left to right searching for earliest two occurrences of the same word. If words occurs only once then $s = \mathrm{length}(\mathcal{P}) \leq |\mathcal{D}|$ and we are done. Otherwise, let $j_1 < j_2$ be found starting positions of occurrences of a word v. Let $j_3 < \ldots < j_p$ be all other occurrences of v. Then by Lemma 5 $\mathcal{P} = (\mathcal{P}_1)^1, (\mathcal{P}'_1)^{p-1}, \mathcal{P}'$ where $\mathrm{length}(\mathcal{P}_1)$, $\mathrm{length}(\mathcal{P}'_1) \leq |\mathcal{D}|$ and $\mathcal{P}'$ is determined by j_p and occurrences starting to the right of j_p. Note that among them there are no occurrences of v. We continue scanning starting from the position to the right of j_p. In this way we obtain

$$\begin{aligned}\mathcal{P} &= (\mathcal{P}_1)^1, (\mathcal{P}'_1)^{p_1}, (\mathcal{P}_2)^1, (\mathcal{P}'_2)^{p_2}, \\ &\quad \ldots, (\mathcal{P}_{q-1})^1, (\mathcal{P}'_{q-1})^{p_{q-1}}, (\mathcal{P}_q)^1\end{aligned}$$

It is enough to prove that $q \leq |\mathcal{D}|$. After each phase of scanning the set of words from $\mathcal{D}$ which occurs at remaining positions loses at least one element. This justifies the inequality $q \leq |\mathcal{D}|$. This completes the proof. □

The most important property of Lemma 6 is that the size of the expression **does not depend** on t but only on the size of $\mathcal{D}$.

Let $\mathcal{P}$ be a sequence of quasi-factors $(w_1, i_1), \ldots, (w_k, i_k)$. Denote $\mathrm{concat}(\mathcal{P}) = w_1 \ldots w_k$.

Lemma 7. *Let $\mathcal{D}$ be a set of words of the same length. Then*

$$\mathcal{D}(w)[i,j] = \mathrm{cut\_last}(\mathcal{D}(w[i,j])), \mathcal{R}$$

where $\mathcal{R}$ is an exponential expression of size at most $O(|\mathcal{D}|^2)$, and $\mathrm{concat}(\mathcal{R}) = \mathrm{last}(\mathcal{D}(w[i,j]))$.

Proof. The quasi-factorizations $\mathcal{D}(w)$ of w and $\mathcal{D}(w[i,j])$ of $w[i,j]$ are based on occurrences of the words in $\mathcal{D}$ in w and $w[i,j]$, respectively. $\mathcal{D}(w)[i,j]$ differs from $\mathcal{D}(w[i,j])$ because of possible occurrences of words from $\mathcal{D}$ which cover the position j in w. The result follows now from Lemma 6. □

5. Quasi-factorizations of solutions

Fix a word equation $e : u = v$ and its minimal solution h. Denote $n = |e|$. A border is defined for a graphical representation of a word being a sequence of symbols written along a straight line. A *border* in a word w is a space between two consecutive symbols of w or a space before or after the word. Each word w contains $|w|+1$ borders. Let $u = u_1u_2$ for some words u_1, u_2. A border between $h(u_1)$ and $h(u_2)$ in $h(u)$ is called *left cut*. Similarly, let $v = v_1v_2$ for some words v_1, v_2. A border between $h(v_1)$ and $h(v_2)$ in $h(v) = h(u)$ is called *right cut*. A *cut* in $h(u)$ is a border being left cut or right cut. Note that the borders before $h(u)$ and after it are left and right cuts and therefore the total number of cuts is at most n. A *characteristic* word of $h(u)$ is a subword w of $h(u)$ such that there is an occurrence of w in $h(u)$ the first symbol of which is to the right of a cut. Formally, w is a characteristic word if $h(u)$ can be written in form pws where p and s are such that there is a cut between p and ws.

Let $\mathcal{F}_l$ be the set of characteristic words of length l. The set $\mathcal{F}_l$ is not empty for $1 \leq l \leq |h(u)|$. Clearly, for each l, $|\mathcal{F}_l| \leq n$.

Proposition 8 ([15],Lemma 6). *If h is minimal then each subword of $h(u)$ has an occurrence over a cut.*

In the following we prove that quasi-factors in $\mathcal{F}_l(h(u))$ are short.

Lemma 9. *Each quasi-factor in $\mathcal{F}_l(h(u))$ is of length at most $l(n+2)$.*

Proof. The factorization $\mathcal{F}_l(h(u))$ is determined by occurrences of characteristic words of length l in $h(u)$. Suppose there is a quasi-factor (w,v) in $\mathcal{F}_l(h(u))$ such that $|w| > l(n+2)$. By the definition of quasi-factorization $\mathcal{F}_l$ the prefix of w of length l is in $\mathcal{F}_l$ and there is no other occurrence of a word from $\mathcal{F}_l$ in w. By Proposition 8 w has an occurrence over a cut. The cut divides the word w into two parts p,r. We have $w = pr$. Since w does not contain an occurrence of word in $\mathcal{F}_l$, we have $|r| < l$. Then $p > l(n+1)$ and again by Proposition 8 there is an occurrence of p (which is a prefix of w) over another cut. Then $p = p'r'$ with $|p'| > (l-1)n$. We repeat the above with p' (which is a prefix of w) and so on up to the moment when we find a prefix of w which occurs over a cut which was hit earlier. Since there are at most n cuts this prefix say t is of length at least $2l$. The cut divides t into parts t' and t'' with $|t'| > l$. t' is contained in a prefix of w which hit the cut earlier in particular it is contained in w. It is impossible since t' as a prefix of w of length at least l starts as w with a word in $\mathcal{F}_l$. A contradiction. □

6. From factorizations to factor equations

A *factor equation over a set $\mathcal{D}$* is a word equation containing exponential expressions over constants in which an alphabet of constants is $\Sigma^* \times \mathcal{D}$ and an alphabet of variables is $\mathcal{G}$. The variables in $\mathcal{G}$ are called *factor variables*. One constant of the equation will correspond to a quasi-factor. Hence, a solution of a factor equation is a substitution of sequences of quasi-factors onto factor variables. We assume that each variable X of the word equation e has a corresponding factor variable $\mathcal{X}$.

Consider the sequence of quasi-factors $s = \mathcal{F}_l(h(u))$. The sequence s corresponds to a factor equation $\mathcal{E} : \mathcal{U} = \mathcal{V}$ over $\mathcal{F}_l \cup \{\$\}$ which we construct as follows. First, we construct $\mathcal{U}$ on the basis of u. Consider an occurrence of a variable X in u, i.e. $u = u_1 X u_2$. Consider the occurrence of the word $h(X)$ in $h(u)$ at position $|h(u_1)|+1$. In the quasi-factorization s of $h(u)$, this occurrence corresponds to some sequence of quasi-factors $(f_1,u_1), (f_2,u_2), \ldots (f_k,u_k)$. By Lemma 7, applied with $\mathcal{D} = \mathcal{F}_l$ and $w = h(u)$ the sequence is of the form $\mathrm{cut\_last}(h(X)), \mathcal{R}$ where $\mathcal{R}$ is of size $O(n^2)$. If $\mathrm{cut\_last}(h(X)) = \emptyset$ (the empty sequence) then we replace the subsequence $(f_1,w_1), (f_2,w_2) \ldots, (f_k,w_k)$ in s by the expression $\mathcal{R}$. Otherwise, we replace it by the sequence: $\mathcal{X}, \mathcal{R}$ where $\mathcal{X}$ is the factor variable corresponding to X. Similarly, we treat all occurrences of variables in u. The sequence obtained from s in this way is $\mathcal{U}$. Similarly, we construct $\mathcal{V}$ on the basis of $h(v) = h(u)$, h and v.

Observe, that in $\mathcal{E}$ there are only variables such that $\mathrm{cut\_last}(h(X)) \neq \emptyset$ and that the substitution $\mathcal{X} = \mathrm{cut\_last}(h(X))$, for all X which satisfy $\mathrm{cut\_last}(h(X)) \neq \emptyset$, is a solution of $\mathcal{E}$. Observe also that although the length of $\mathcal{E}$ can be long because of exponential expressions which can represent long sequences of quasi-factors, it is represented by an exponential expression (containing factor variables) of size $O(n^3)$. This does not mean, however, that the equation can be represented in PSPACE since the size does not include the real size of quasi-factors and exponents. The equation was constructed from a quasi-factorization of a minimal solution of an equation, so by Proposition 4 each exponent in it can be encoded on cn bits. Quasi-factors are constants in $\mathcal{E}$ so the only important information on them is which of them are equal. Therefore, we have

Lemma 10. *The factor equation $\mathcal{E}$ is isomorphic to a factor equation which can be represented in space polynomial in n.*

Consider now two sequences $s_1 = \mathcal{F}_{l+1}(h(u))$ and $s_2 = \mathcal{F}_l(h(u))$. Since the words in $\mathcal{F}_l$ are prefixes of the words in $\mathcal{F}_{l+1}$, the set of starting positions of occurrences of words from $\mathcal{F}_{l+1}$ in $h(u)$ is a subset of the set of starting positions of occurrences of words from $\mathcal{F}_l$ in $h(u)$. Hence, the sequence s_2 is obtainable from the sequence s_1 by replacing its quasi-factors by sequences of quasi-factors which occur in s_2. Moreover, since each quasi-factor (w,v) "remembers" the word $v \in \mathcal{F}_{l+1}$ which occurs just after w in $h(u)$, two identical quasi-factors in s_1 are replaced by the same sequence of quasi-factors of s_2. Moreover, as the following lemma says, the sequence can be represented by an exponential expression of size $O(n^3)$.

Lemma 11. *Let the quasi-factor (w,v) be replaced by a sequence of quasi-factors $\mathcal{S} = (w_1,u_1), (w_2,u_2), \ldots (w_k,u_k)$. Then the sequence $\mathcal{S}$ can be represented by an exponential expression of size $O(n^3)$.*

Proof. Clearly, $w = w_1 w_2 \ldots w_k$. Moreover $u = u_k a$, for some $a \in \Sigma$, or $v = u_k = \$$. Assume, that wv occurs at position i in $h(u)$. By Lemma 9, $|w| \leq (l+1)(n+2)$. We mark in the occurrence of w at i in $h(u)$ positions $i+l, i+2l, i+3l, \ldots$. There are at most $\frac{(l+1)(n+2)}{l} + 1 \leq 2(n+2)$ marked positions. Since $|u_i| = l$, each occurrence of u_i covers at most one marked position. By Lemma 6, the sequence of quasi-factors $(w_i,u_i), \ldots (w_j,u_j)$, for $i < j$, such that the occurrences of u_i,...,u_j in $h(u)$ cover the same marked position can be represented by an exponential expression of size $O(n^2)$. Since the number of marked positions is linear in n, the result follows. □

Denote by $\mathcal{E}_l$ the factor equation which corresponds to the sequence of quasi-factors $\mathcal{F}_l(h(u))$. We will show now how to transform $\mathcal{E}_{l+1}$ onto $\mathcal{E}_l$, nondeterministically. First we replace constants of $\mathcal{E}_{l+1}$ by exponential expressions as in Lemma 11. In this way we obtain an equation $\mathcal{E}'$ containing exponential expressions of height at

most two. The set of constants of $\mathcal{E}'$ is now the same as the set of constants of $\mathcal{E}_l$. In $\mathcal{E}_l$ may occur variables which does not occur in $\mathcal{E}'$. Such a variable $\mathcal{X}$ has the property that a corresponding variable X in e satisfies $\mathrm{cut\_last}(\mathcal{F}_{l+1}(h(X)) = \emptyset$ and $\mathrm{cut\_last}(\mathcal{F}_l(h(X)) \neq \emptyset$. We replace sequences $\mathrm{cut\_last}(\mathcal{F}_l(h(X))$ in $\mathcal{E}'$ which correspond to occurrences of $\mathcal{X}$ in $\mathcal{E}'_l$ by $\mathcal{X}$ obtaining equation $\mathcal{E}''$. Now the sets of factor variables and constants of $\mathcal{E}_l$ and $\mathcal{E}''$ are the same. Now we take care of the other factor variables, ie. such that their corresponding variables X satisfy $\mathrm{cut\_last}(\mathcal{F}_{l+1}(h(X)) \neq \emptyset$ and $\mathrm{cut\_last}(\mathcal{F}_l(h(X)) \neq \emptyset$. The word $\mathrm{concat}(\mathrm{cut\_last}(\mathcal{F}_{l+1}(h(X)))$ can be a proper prefix of $\mathrm{concat}(\mathrm{cut\_last}(\mathcal{F}_l(h(X)))$. In this case the variable $\mathcal{X}$ in $\mathcal{E}''$ expands to the right, ie. for some exponential expression $\mathcal{R}$ to the right of each occurrence of $\mathcal{X}$ in $\mathcal{E}''$ there is an expression representing a sequence which starts from $\mathcal{R}$. The occurrences of $\mathcal{X}$,$\mathcal{R}$ are thus replaced by $\mathcal{X}$. In this way we obtain an equation $\mathcal{E}'''$ with expressions of height at most two which is isomorphic to $\mathcal{E}_l$.

We omit the proof of the following lemma.

Lemma 12. *The isomorphism of two equations one with expressions of height at most two and the second one with expressions of height at most one can be checked in polynomial time.*

A more general result can be concluded from the result in [13], namely that isomorphism of two equations with exponential expressions of arbitrary height can be decided in polynomial time.

Denote $\mathcal{E}_0 = e$. Observe that the equation e is obtainable from $\mathcal{E}_1$ in the same way as $\mathcal{E}_l$ from $\mathcal{E}_{l+1}$.

7. The relation $\rightarrow$

The definition of the relation $\rightarrow$ follows the transformation of $\mathcal{E}_{l+1}$ to $\mathcal{E}_l$. It is defined on equations with exponential expressions of size $O(n^3)$. $e_1 \rightarrow e_2$ if e_2 is isomorphic to an equation which can be obtained from e_1 in the following way:

1. replace constants of e_1 by exponential expressions of size $O(n^3)$ with exponents at most 2^{cn},
2. replace fragments: $\mathcal{X}$, $\mathcal{R}$, for an exponential expression $\mathcal{R}$ which can be a part of other expression, by $\mathcal{X}$ (for each variable this transformation should be done for all occurrences of it),
3. replace some sequences of constants by new variables (for the same variables the sequences should be the same).

The resulting equation may contain exponential expressions of height two, but the isomorphism of it with e_2 can be checked in polynomial time.

Lemma 13. *If $e_1 \rightarrow e_2$ and e_1 is satisfiable then e_2 is satisfiable, too.*

Proof. Let h be a solution of e_1. Using it we define a solution g of e_2. The definition uses the sequences which occur in definition of the relation $\rightarrow$. For each constant a, denote by exp_a the exponential expression which replaced a constant a in step 1. This replacement defines a morphism σ such that $\sigma(a) = exp_a$, for each constant a of e_1. Denote by $\mathcal{R}_\mathcal{X}$ the exponential expression which is used as $\mathcal{R}$ in step 2 for the variable $\mathcal{X}$. Denote by $\mathcal{P}_\mathcal{Y}$ the sequence of constants which is replaced by a new variable $\mathcal{Y}$ in step 3. The solution g of e_2 is defined by

$$g(\mathcal{X}) = \begin{cases} \sigma(h(\mathcal{X})), \mathcal{R}_\mathcal{X} & \text{if } \mathcal{X} \text{ occurs in } e_1 \text{ and } e_2 \\ \mathcal{P}_\mathcal{Y} & \text{if } \mathcal{Y} \text{ occurs in } e_2 \\ & \text{but not in } e_1 \end{cases}$$

□

Lemma 14. *$(a = a) \rightarrow^* e$ iff e is satisfiable.*

Proof. If $e : u = v$ is solvable then we take its minimal solution h. We construct factor equations $\mathcal{E}_l$ corresponding to sequences $\mathcal{F}_l(h(u))$. We have $\mathcal{E}_{|h(u)|} \rightarrow^* \mathcal{E}_0$. Since $\mathcal{E}_{|h(u)|}$ is isomorphic to $a = a$ and $\mathcal{E}_0$ is e the result follows.

If $(a = a) \rightarrow^* e$ then the result follows from Lemma 13. □

Theorem 15. *The problem of satisfiability of word equations is in PSPACE.*

References

[1] Abdulrab H., *Resolution d'equations sur le mots: etude et implementation LISP de l'algorithme de Makanin*, Ph.D. Thesis, Universite de Rouen, 1987.

[2] Abdulrab H., Pecuchet J.P., Solving word equations, *Journal of Symbolic Computation* **8**(1989), 499-521.

[3] Angluin D., Finding pattern common to a set of string, *in Proc. STOC'79*, 130-141, 1979.

[4] Diekert V., personal communication, 1998.

[5] Diekert V., Makanin's algorithm, a Chapter *in* Algebraic aspects of combinatorics on words (Ed.: J. Berstel and D.Perrin), 1999, to appear.

[6] Diekert V., Hagenach C., Word equations with regular constraints, in: *Preproceedings of DLT'99*, 35-41, 1999.

[7] Jaffar J., Minimal and complete word unification, *Journal of the ACM* **37**(1), 47-85, 1990.

[8] Gutierrez C., Satisfiability of word equations with constants is in exponential space, in: *Proc. FOCS'98*, IEEE Computer Society Press, Palo Alto, California.

[9] Koscielski A., Pacholski L., Complexity of Makanin's Algorithm, *Journal of the ACM* **43**(4), 670-684.

[10] Kozen D., Lower bounds for natural proof systems, in: *Proc. FOCS'77*, 254-266, 1977.

[11] Makanin G. S., The problem of solvability of equations in a free semigroup, *Mat. Sb.*, **103**(2), 147-236. In Russian; English translation in: *Math. USSR Sbornik*, **32**, 129-198, 1977.

[12] Pecuchet J.P., *Equations avec constantes et algorithme de Makanin*, Ph.D. Thesis, Laboratoire d'informatique, Rouen, 1981.

[13] Plandowski W., Testing equivalence of morphisms on context-free languages, in: *Proc. ESA'94*, LNCS, 1994.

[14] Plandowski W., Satisfiability of word equations is in NEXPTIME, in: *Proc. STOC'99*, to appear.

[15] Plandowski W., Rytter W., Application of Lempel-Ziv encodings to the solution of word equations, in: *Proc. ICALP'98*, LNCS 1443, 731-742, 1998.

[16] Rytter W., Personal comunication, 1999.

[17] Schulz K.U., Makanin's algorithm for word equations: two improvements and a generalization, in: *Proc. IWWERT'90*, LNCS 572, 85-150, 1992.