Two-sided unification is NP-complete

Tatyana A. Novikova[1] and Vladimir A. Zakharov[2]

[1] Kazakhstan Branch of Lomonosov Moscow State University
[2] Lomonosov Moscow State University
(taniaelf@mail.ru, zakh@cs.msu.su)

Abstract

It is generally accepted that to unify a pair of substitutions θ_1 and θ_2 means to find out a pair of substitutions η' and η'' such that the compositions $\theta_1\eta'$ and $\theta_2\eta''$ are the same. Actually, unification is the problem of solving linear equations of the form $\theta_1 X = \theta_2 Y$ in the semigroup of substitutions. But some other linear equations on substitutions may be also viewed as less common variants of unification problem. In this paper we introduce a two-sided unification as the process of bringing a given substitution θ_1 to another given substitution θ_2 from both sides by giving a solution to an equation $X\theta_1 Y = \theta_2$. Two-sided unification finds some applications in software refactoring as a means for extracting instances of library subroutines in arbitrary pieces of program code. In this paper we study the complexity of two-sided unification and show that this problem is NP-complete by reducing to it the bounded tiling problem.

1 Introduction

To unify a pair of expressions E_1 and E_2 means to compute such instances of these expressions that are identical (syntactical unification) or have the same meaning (semantical unification). Such common instances of E_1 and E_2 can be obtained by replacing some variables in E_1 and E_2 by appropriate terms, i.e. by applying some substitutions to these expressions. Unification algorithms have found a wide utility in theorem proving, logic programming, term rewriting, type inference, language processing, etc. (see [1, 2]). In [9] it was shown that unification problem is also meaningful and efficiently decidable when expressions E_1 and E_2 are some formal models in imperative programs. If the programs are unifiable then their behaviors are somewhat similar; therefore, some results of the analysis of one program (proofs of its correctness, termination, etc.) can be easily adapted to the other.

But a similarity of programs can be formalized differently. Suppose that one has a library subroutine $\pi_0(\vec{x}\!:\!input;\ \vec{y}\!:\!output)$ with a set of formal input arguments $\vec{x}$ and a set of formal output parameters $\vec{y}$. Given some piece of program code π_1 one may wonder if it is possible to replace it with an appropriate subroutine call. Such a replacement would make the program both succinct and uniform which is very much helpful for program understanding and analysis. To this end one could try to find such instantiation η'' of input arguments $\vec{x}$ and such specialization η' of output parameters $\vec{y}$ as to make the composition of η', π_0, and η'' equivalent to π_1. In some formal models of programs (see [4, 8, 9]) a behavior of a program π can be specified by a substitution θ_π which assigns terms on input arguments $\vec{x}$ to output parameters $\vec{y}$. Thus, we can set up the following problem: given a pair of substitutions θ_{π_0} and θ_{π_1} find a pair of substitutions η'' (input instantiation) and η' (output specialization) such that the composition $\eta'\theta_{\pi_0}\eta''$ is equal to θ_{π_1}, or, in other words, solve the equation $X\theta_{\pi_0}Y = \theta_{\pi_1}$ in the semigroup of substitutions. It is worth noticing that the conventional unification problem may be regarded as that of solving linear equations of the form $\theta_1 X = \theta_2 Y$ in the semigroup of substitutions when both unknown substitutions are applied to θ_1 and θ_2 from the one side. Therefore, we call the solving of equations of the form $X\theta_0 Y = \theta_1$ when unknowns appear on both sides of

θ_0 *two-sided unification* of substitutions θ_0 and θ_1. In this paper we show that the problem of two-sided unification for first-order substitutions is NP-complete.

The paper is organized as follows. In Section 2 we recall briefly the basic notions concerning first-order substitutions, set up formally two-sided unification problem, and show that it is in NP. Afterward, in Section 3 we consider BOUNDED TILING problem which is widely used in complexity theory (see [3]) as an alternative to SATISFIABILITY. Finally, in Section 3 we prove that BOUNDED TILING is reducible to two-sided unifiability problem.

2 Preliminaries.

We deal with the first-order language over some fixed sets of functional symbols $\mathcal{F}$. Letters $\mathcal{U}, \mathcal{X}, \mathcal{Y}, \mathcal{Z} \ldots$ will be used for pairwise disjoint finite sets of variables. The set of terms $Term[\mathcal{X}]$ over a set of variables is defined as usual.

Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ and $\mathcal{Y} = \{y_1, y_2, \ldots\}$ be two sets of variables. A $\mathcal{X}$-$\mathcal{Y}$-*substitution* is any mapping $\theta : \mathcal{X} \to Term[\mathcal{Y}]$. Every such mapping can be represented as a set of bindings $\theta = \{x_1/\theta(x_1), \ldots x_n/\theta(x_n)\}$. We write $Subst[\mathcal{X}, \mathcal{Y}]$ for the set of all $\mathcal{X}$-$\mathcal{Y}$-substitutions. An *application* of a substitution θ to a term $t(x_1, \ldots, x_n)$ yields the term $t\theta = t(\theta(x_1), \ldots, \theta(x_n))$ obtained from t by replacing all occurrences of every variable $x_i,\ 1 \leq i \leq n$, with the term $\theta(x_i)$. A *composition* of a $\mathcal{X}$-$\mathcal{Y}$-substitution θ and a $\mathcal{Y}$-$\mathcal{Z}$-substitution η is a $\mathcal{X}$-$\mathcal{Z}$-substitution ξ such that the equality $x\xi = (x\theta)\eta$ (or, in other notation, $\xi(x) = (\theta(x))\eta$) holds for every $x,\ x \in \mathcal{X}$. To denote the composition of θ and η we use an expression $\theta\eta$; since $t(\theta\eta) = (t\theta)\eta$ holds for every term $t,\ t \in Term[\mathcal{X}]$, this notation makes it possible to skip parentheses when writing $t\theta\eta$ for the application of a composition of substitutions to a term. A $\mathcal{X}$-$\mathcal{X}$-substitution ρ is called a *renaming* iff ρ is a bijection on the set of variables $\mathcal{X}$. Two $\mathcal{X}$-$\mathcal{Z}$-substitutions θ_1 and θ_2 are *equivalent* if $\theta_1 = \theta_2\rho$ for some $\mathcal{X}$-$\mathcal{X}$-renaming ρ. If θ_1 is a composition of θ_2 and η then θ_1 is called an *instant* of θ_2, and θ_2 is called a *pattern* of θ_1.

Let θ_0 be a $\mathcal{X}$-$\mathcal{Y}$-substitution and θ_1 be a $\mathcal{Z}$-$\mathcal{U}$-substitution. Then a pair of substitutions η' and η'' from $Subst[\mathcal{Z}, \mathcal{X}]$ and $Subst[\mathcal{Y}, \mathcal{U}]$ respectively is called a *two-sided unifier* of (θ_0, θ_1) iff $\eta'\theta_0\eta'' = \theta_1$. Two-sided unification problem is that of finding, given a pair of substitutions (θ_0, θ_1), a two-sided unifier (η', η'') of (θ_0, θ_1). It must be noticed that two-sided unification, unlike usual unification, is asymmetric, since substitutions θ_0 and θ_1 play different roles in the equation $X\theta_0 Y = \theta_1$. Another important aspect of two-sided unification to be emphasized is that a substitution η'' does not affect directly the variables from $\mathcal{X}$ but only through the terms from θ_0 via the set of variables $\mathcal{Y}$. This is due to the software engineering application two-sided unification problem stems from: η'' only initializes input variables of θ_0 but does not interfere in the computation of θ_0.

Two-sided unification problem for a pair of substitutions (θ_0, θ_1) may have several solutions. For example, if $\theta_0 = \{x_1/f(y_1, y_2), x_2/y_3\}, \theta_1 = \{z/f(f(u, u), f(u, u))\}$ then two-sided unifiers of (θ_0, θ_1) are non-equivalent pairs $(\eta' = \{z/f(x_1, x_1)\}, \eta'' = \{y_1/u, y_2/u\})$, $(\eta' = \{z/f(f(x_2, x_2), x_1)\}, \eta'' = \{y_1/u, y_2/u\})$ and $(\eta' = \{z/x_1\}, \eta'' = \{y_1/f(u, u), y_2/f(u, u)\})$. Since the first component η' of every such pair is a pattern of θ_1 and the set of non-equivalent patterns of every substitution is finite, the set of non-equivalent two-sided unifiers of every pair of substitutions (θ_0, θ_1) is also finite.

When the complexity issues of decision problems for substitutions are concerned, the representation of terms in a set of bindings $\theta = \{x_1/t_1, \ldots x_n/t_n\}$ is of prime importance. We will assume that terms $t_1, \ldots, t_n$ in every substitution θ are represented by labeled trees. A representation of a composition $\theta\eta$ can be obtained from representations of θ and η just by attaching the terms from η to the corresponding leaves in the representations of terms from θ.

Lemma 1. *The problem of two-sided unifiability of pairs of substitutions represented by trees is in NP.*

Proof. It is easy to see that two-sided unifiability of (θ_0, θ_1) can be non-deterministically checked in polynomial time. It is sufficient

1. to guess a cut of tree representation T_{θ_1} of θ_1 into three pieces T_1, T_2, and T_3 in such a way that the leaves of every tree in T_1 and T_2 become the roots in the tree representations of T_2 and T_3 respectively.

2. to assign consistently variables from $\mathcal{X}$ and $\mathcal{Y}$ to all leaves of T_1 and T_2 respectively (the same variable can be assigned to different leaf nodes v_1, v_2 of a piece $T_i, i = 1, 2$, only if v_1 and v_2 are the roots of equal subtrees in T_{i+1}), and

3. to check that all trees from the middle piece T_2 represent only terms from θ_0.

Clearly, such cut of T_1, T_2, and T_3 of T_{θ_1} do exist iff $\theta_1 = \eta'\theta_0\eta''$ for some substitutions η' and η''. It is easy to see that the consistency of variable assignment and the inclusion of T_2 in T_{θ_0} can be checked in polynomial time. □

NP-hardness of two-sided unifiability problem follows from NP-completeness of BOUNDED TILING problem which is formally defined in the next section.

3 Bounded tiling problem

To define the bounded tiling problem imagine 1×1 square tiles whose edges are coloured. Suppose that only finitely many types of tiles are available. Consider a $n \times m$ rectangular area whose border is divided into segments of length 1 and assume that all such segments are also colored. The problem is to determine if it is possible to cover this area with the tiles (i.e. make a tiling) in such a manner that every pair of adjacent edges of two tiles has the same colour and every border segment has the same colour as the edge of a tile adjacent to it.

Formally BOUNDED TILING problem is specified as follows. Let $Colours = \{1, 2, \ldots, K\}$ be a finite set of *colours*. A *tile* is a quadruple $tile = \langle a_1, a_2, a_3, a_4\rangle$ of colours. The components of *tile* are denoted by $tile[0, -1], tile[-1, 0], tile[0, 1], tile[1, 0]$ respectively; they identify the colours of the top, right, bottom and left edges of the tile. A $n \times m$ *area* is the set of pairs $Area = \{(i, j) \ : \ 0 \le i \le n+1, 0 \le j \le m+1\}$; the elements of this set are called *squares*. The set of squares $Inter = \{(i, j) \ : \ 1 \le i \le n, 1 \le j \le m\}$ is the *interior* of the area. The *border* of the area is the set of squares $Border = Area \setminus Inter$. Two squares (i_1, j_1) and (i_2, j_2) in the *Area* are called *adjacent* iff $|i_1 - i_2| + |j_1 - j_2| = 1$. A *boundary constraint* is any mapping $B : Border \to Colours$. If $B(i, j) = a$ then this means that the "innermost" edge of a border square (i, j) is painted colour a. Let $Tiles = \{tile_1, \ldots, tile_L\}$ be a finite set of tiles. Then a *tiling* of an *Area* is any mapping $T : Inter \to Tiles$. Given a boundary constraint B, a tiling T is called B-*consistent* if the following two requirements are satisfied:

1. for every pair of adjacent interior squares (i_1, j_1) and (i_2, j_2) the equality $T(i_1, j_1)[i_1 - i_2, j_1 - j_2] = T(i_2, j_2)[i_2 - i_1, j_2 - j_1]$ holds; this equality means that the adjacent edges of the tiles inserted on these squares have the same colour;

2. for every interior square (i_1, j_1) which is adjacent to a border square (i_2, j_2) the equality $T(i_1, j_1)[i_1 - i_2, j_1 - j_2] = B(i_2, j_2)$ holds; this equality means that the colour of the border segment matches the colour of the adjacent edge of the tile.

An instance of the BOUNDED TILING problem is a tuple $BT = (n, m, Tiles, B)$; this instance is accepted iff there exists a B-consistent tiling of $n \times m$ $Area$ with tiles from $Tiles$. For the first time the TILING problem has been introduced in [10]. The complexity of this problem depends on the area to be tiled. Thus, in [11] it has been shown that if $Area$ is a quadrant of infinite plane then TILING problem is undecidable. In [3] it has been proved that BOUNDED TILING problem is NP-complete. We use this fact to prove NP-hardness of the two-sided unifiability problem.

4 NP-hardness of two-sided unification

Let $Colours = \{1, \ldots, K\}$. Consider an instance of the BOUNDED TILING problem $BT = (n, m, Tiles, B)$, where $Tiles = \{tile_1, \ldots, tile_L\}$. We show how to build such a pair of substitutions θ_0 and θ_1 that their two-sided unification (η', η''), if any, gives a solution to BT. The bindings of $\mathcal{X}$-$\mathcal{Y}$-substitution θ_0 represent the boundary constraint B and all possible insertions of tiles from $Tiles$ onto interior squares of the $Area$. The $\mathcal{Z}$-$\mathcal{U}$-substitution θ_1 represents the tiling of the same area with monochromatic tiles whose edges are painted colour K. The first component η' of a two-sided unifier specifies a choice of some possible tiling T of the $Area$, and the second component η'' checks the consistency of this tiling by simulating an attempt to "re-paint" consistently the edges of all tiles and border segments to achieve monochromatic tiling. The key feature of the substitution θ_0 is that the terms in its bindings share variables in such a manner that the colours of the adjacent edges of tiles can be changed only in common and by the same value. Therefore, a monochromatic "re-painting" η'' is possible only for B-consistent tilings.

To define θ_0 and θ_1 formally we introduce a set of functional symbols $\mathcal{F}$ which includes

- a binary function $g^{(2)}$ to build a $n \times m$ area,
- a 6-ary function $h^{(6)}$ to construct border constraints and instances of tiles,
- a unary function $f^{(1)}$ to enumerate colours and squares.

As for the sets of variables the substitutions θ_0 and θ_1 operate with, we assume that

- $\mathcal{X} = \mathcal{X}' \cup \mathcal{X}''$, where
  - $\mathcal{X}' = \{x'_{i,j} : (i,j) \in Border\}$: every variable $x'_{i,j}$ is associated with a border square (i,j) in the $Area$,
  - $\mathcal{X}'' = \{x''_{i,j,\ell} : (i,j) \in Interior, 1 \le \ell \le L\}$: every variable $x''_{i,j,\ell}$ is associated with an instant of a tile $tile_\ell$ inserted onto the square (i,j);
- $\mathcal{Y} = \{y_0\} \cup \mathcal{Y}'$, where
  - y_0 is a common "dummy" variable;
  - $\mathcal{Y}' = \{y_{i_1,j_1,i_2,j_2} : 0 \le i_1 \le i_2 \le n+1, 0 \le j_1 \le j_2 \le m+1, |i_1 - i_2| + |j_1 - j_2| = 1\}$: every variable y_{i_1,j_1,i_2,j_2} is associated with a pair of adjacent squares (i_1, j_1) and (i_2, j_2) in the $Area$;
- $\mathcal{Z} = \{z\}$, and $\mathcal{U} = \{u\}$.

By means of functional symbol $f^{(1)}$ we define recursively *numerals* $f_n(y)$ for every integer n as follows: $f_0(y) = y$ and $f_{n+1}(y) = f(f_n(y))$ for every $n,\ n \geq 0$. Numerals will be used to enumerate squares and colours. Clearly, $f_n(f_m(y)) = f_{n+m}(y)$ holds for every pair of integers n, m. We will say that a numeral $f_n(y)$ has a *rank* n.

The terms that represent the boundary constraint B and all possible insertions of individual tiles from $Tiles$ onto interior squares are defined as follows.

If a border square (i, j) is such that $(i, j) \in \{(0, 0), (n + 1, 0), (n + 1, m + 1), (0, m + 1)\}$ (i.e. (i, j) is a corner square of $Area$) then we assign the term

$$t_{i,j} = h(f_i(y_0), f_j(y_0), f_K(y_0), f_K(y_0), f_K(y_0), f_K(y_0))$$

to the variable $x_{i,j}$ associated with the square (i, j). This term indicates that all edges of this square are painted colour K.

Suppose that $(i, j) \in Border \setminus \{(0, 0), (n + 1, 0), (n + 1, m + 1), (0, m + 1)\}$ and $B(i, j) = k$. Then there exists the only interior square (i', j') which is adjacent to (i, j). Let y_{i_1,j_1,i_2,j_2} be the variable from $\mathcal{Y}'$ which is associated with the pair (i, j), (i', j'). Then we assign the term

$$t_{i,j} = h(f_i(y_0), f_j(y_0), f_K(y_0), f_K(y_0), f_K(y_0), f_k(y_{i_1,j_1,i_2,j_2}))$$

to the variable $x_{i,j}$ associated with the border square (i, j). This term indicates that the interior edge of this square (border segment) is painted colour $B(i, j)$, whereas all other edges are painted colour K.

Suppose that $(i, j) \in Interior$. Then there are exactly four squares in the $Area$ that are adjacent to the square (i, j) on the top, on the right, on the bottom, and on the left. Let $y_{i_1,j_1,i'_1,j'_1}, y_{i_2,j_2,i'_2,j'_2}, y_{i_3,j_3,i'_3,j'_3}$, and y_{i_4,j_4,i'_4,j'_4} be all those variables from $\mathcal{Y}'$ that are associated with these pairs of adjacent squares respectively. Then for every tile $tile_\ell = \langle k_1, k_2, k_3, k_4\rangle,\ 1 \leq \ell \leq L$, from $Tiles$ we assign the term

$$t_{i,j,\ell} = h(f_i(y_0), f_j(y_0), f_{k_1}(y_{i_1,j_1,i'_1,j'_1}), f_{k_2}(y_{i_2,j_2,i'_2,j'_2}), f_{k_3}(y_{i_3,j_3,i'_3,j'_3}), f_{k_4}(y_{i_4,j_4,i'_4,j'_4}))$$

to the variable $x_{i,j,\ell}$ associated with the interior square (i, j) and the tile $tile_\ell$.

With terms $t_{i,j}$ and $t_{i,j,\ell}$ at hand, we define the substitution θ_0:

$$\theta_0 = \{x_{i,j}/t_{i,j} : (i, j) \in Border\} \cup \{x_{i,j,\ell}/t_{i,j,\ell} : (i, j) \in Interior, 1 \leq \ell \leq L\}.$$

It worth noticing that every variable from the set $\mathcal{Y}'$ occurred as an argument of numerals exactly in two terms from the range of substitution θ_0. We say that an occurrence of a variable y has a *depth* n iff n is the maximal rank of a numeral which includes this occurrence of y.

Using functional symbol $g^{(2)}$ we can build a (arbitrary) term t_{area} which has $(n+2)(m+2)$ argument positions (leaves in the tree representation of the term); every argument position in this term stands for a square in the $Area$. For every square (i, j) in the $Area$ we introduce the term $\widehat{t}_{i,j} = h(f_i(u), f_j(u), f_K(u), f_K(u), f_K(u), f_K(u))$ and define the substitution $\theta_1 = \{z/t_{area}(\widehat{t}_{0,0}, \widehat{t}_{0,1}, \ldots, \widehat{t}_{n+1,m+1})\}$ (monochromatic tiling of $Area$).

Lemma 2. *An instance of the BOUNDED TILING problem $BT = (n, m, Tiles, B)$ is acceptable iff the substitutions θ_0 and θ_1 defined above are two-sided unifiable.*

Proof. 1) Suppose that the instance BT is acceptable. Then there exists a B-consistent tiling T of $Area$ with the tiles from the set $Tiles$. For every pair of adjacent squares (i, j) and (i', j') in the interior of the area (assuming that $i \leq i',\ j \leq j'$) denote by $c(i, j, i', j')$ the common colour of the adjacent edges of the tiles $T(i, j)$ and $T(i', j')$ installed onto these squares. The

same notation will be used for the common colour of a tile's edge and an adjacent segment of the boarder. By the definition of the terms $t_{i,j,\ell}$ both occurrences of a variable $y_{i,j,i',j'}$ in terms $t_{i,j,T(i,j)}$ and $t_{i',j',T(i',j')}$ have the same depth $c(i,j,i',j')$. Then a two-sided unification of (θ_0, θ_1) is a pair (η', η'') such that

$$\begin{array}{lcl} \eta' & = & \{z/t_{area}(x_{0,0}, x_{0,1}, \ldots, x_{0,m+1}, x_{1,0}, x_{1,1,T(1,1)}, \ldots, x_{1,m,T(1,m)}, x_{1,m+1}, \ldots, x_{n+1,m+1})\}, \\ \eta'' & = & \{y_0/u, y_{0,1,1,1}/f_{K-c(0,1,1,1)}(u), \ldots, y_{i,j,i',j'}/f_{K-c(i,j,i',j')}(u), \ldots\}. \end{array}$$

In substitution η' every argument of the term t_{area} corresponding to a square (i,j) is either a variable $x_{i,j}$ in the event that (i,j) is a boarder square, or a variable $x_{i,j,T(i,j)}$ in the event that (i,j) is an interior square. In the latter case the variable $x_{i,j,T(i,j)}$ indicates that a tile $tile_{T(i,j)}$ is placed onto the square (i,j). The substitution η'' assigns to every variable $y_{i,j,i',j'}$ associated with a pair of adjacent edges of two squares a numeral $f_{K-c(i,j,i',j')}(u)$ to complement the common colour $c(i,j,i',j')$ of the adjacent edges of the tiles $T(i,j)$ and $T(i',j')$ to the maximal colour K. By taking into account the fact that the tiling T is B-consistent we arrive at the conclusion that $\theta_1 = \eta'\theta_0\eta''$.

2) Suppose that $\theta_1 = \eta'\theta_0\eta''$ holds for a pair of substitutions (η', η''). Consider a sequence of functional symbols assigned to the nodes in an arbitrary branch in a tree representation of substitution θ_1. As it follows from the definition of θ_1, this sequence is $g, g, \ldots, g, h, f \ldots, f$. Moreover, for every square (i,j) the term in the range of θ_1 contains the only term of the form $h(f_i(y_0), f_j(y_0), \ldots)$. At the same time all terms in the range of θ_0 contain only functional symbols h and f. Thus, the substitution η' takes the form:

$$\eta' = \{z/t_{area}(x_{0,0}, x_{0,1}, \ldots, x_{0,m+1}, x_{1,0}, x_{1,1,\ell_{1,1}}, \ldots, x_{1,m,\ell_{1,m}}, x_{1,m+1}, , x_{n+1,m+1})\},$$

and η'' is a substitution of the form:

$$\eta'' = \{y_0/u, y_{0,1,1,1}/f_{k_{0,1,1,1}}(u), \ldots, y_{i,j,i',j'}/f_{k_{i,j,i',j'}}(u), \ldots\}.$$

Consider a tiling T such that $T(i,j) = \ell_{i,j}$ holds for every square (i,j) iff some term of the substitution η' includes a variable $x_{i,j,l_{i,j}}$. We show that this tiling is B-consistent.

Assume the contrary. Then there exists a pair of squares (i_1, j_1) and (i_2, j_2) in the *Area* such that either the adjacent edges of the tiles $T(i_1, j_1)$ and $T(i_2, j_2)$ inserted into these squares have different colours, or the adjacent edges of the tile $T(i_1, j_1)$ and the boarder square (i_2, j_2) are coloured differently. Without loss of generality we consider only the former case. Then the occurrences of the shared variable y_{i_1,j_1,i_2,j_2} in the terms $t_{i_1,j_1,\ell_{i_1 j_1}}$ and $t_{i_2,j_2,\ell_{i_2,j_2}}$ have different depths. Therefore, both occurrences of y_{i_1,j_1,i_2,j_2} in the terms of substitutions $\eta'\theta_0 = \{z/t_{area}(x_{0,0}, \ldots, x_{n+1,m+1})\theta_0\}$ also have different depths. Since η'' substitutes the same term instead of both occurrences of y_{i_1,j_1,i_2,j_2}, the numerals that indicate the colour of adjacent edges in the terms of composition $\eta'\theta_0\eta''$ have different ranks as well. In view of the fact that $\theta_1 = \eta'\theta_0\eta''$ the latter seems contrary to the definition of θ_1: all numerals that indicate the colour of edges must have the same rank K.

Thus, the tiling T defined above is B-consistent. □

Lemma 3. *BOUNDED TILING problem is log − space reducible to the problem of two-sided unifiability of first-order substitutions.*

Proof. Suppose that an instance of the BOUNDED TILING problem $BT = (n, m, Tiles, B)$ has a size N. As it can be seen from the definition of terms $t_{i,j}$, $t_{i,j,\ell}$, and $\widehat{t}_{i,j}$, a tree representation of every such term can be built by a deterministic procedure which operates on an auxiliary space of the size $O(\log N)$. Hence, tree representation of substitutions θ_0 and θ_1 as defined above can be also built within the same space. □

The main theorem follows from Lemmas 1 and 3.

Theorem 1. *Two-sided unifiability problem for first-order substitutions is NP-complete.*

5 Conclusion

This theorem completes the complexity picture in the study of solvability problem for equations of the form $X_1^{\sigma_1}\theta X_2^{\sigma_2} = X_1^{\sigma_3}\eta X_4^{\sigma_4}$ in the semigroup of first-order substitutions, where $\sigma_i \in \{0,1\}$, and X^σ is either X in the case of $\sigma = 1$, or empty substitution in the case of $\sigma = 0$. It is obvious that equations $X_1\theta X_2 = X_3\eta X_4$, $X_1\theta X_2 = \eta X_4$, and $X_1\theta X_2 = X_3\eta$ are trivially solvable for every pair of substitutions θ, η. Equations $\theta X_2 = \eta X_4$ and $\theta X_2 = \eta$ correspond to conventional unification problem; it is known that they are decidable in almost linear time (see [5, 6, 7]). Equations $X_1\theta = X_3\eta$ and $X_1\theta = \eta$ appeared in [12] with regard to equivalence checking problem in some class of sequential programs; they are decidable in polynomial time. Finally, in this paper we prove that only the solvability of equations of the form $X_1\theta X_2 = \eta$ (two-sided unification) is NP-complete problem.

References

[1] F. Baader, W. Snyder: Unification theory. In J.A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, 2001, v. 1, p. 447-533.

[2] K. Knight: Unification: a multidisciplinary survey. *ACM Computing Surveys*, 1989, v. 21, N 1, p. 93-124.

[3] C.H. Lewis: Complexity of solvable cases of the decision problem for predicate calculus. Proceedings of the 19-th Annual Symposium on Foundations of Computer Science, 1978, p. 35-47.

[4] D.C. Luckham, D.M. Park, M.S. Paterson: On formalized computer programs. *Journal of Computer and System Science*, 1970, v. 4, N 3, p. 220-249.

[5] Z. Manna, R. Waldinger: Deductive synthesis of the unification algorithm. *Science of Computer Programming*. 1981, v. 1, N 1-2, p. 5-48.

[6] A. Martelli, U. Montanari: An efficient unification algorithm. *ACM Transactions on Programming Languages and Systems*, 1982, v. 4, N 2, p. 258-282.

[7] M.S. Paterson, M.N. Wegman: Linear unification. *The Journal of Computer and System Science*, v. 16, N 2, 1978, p. 158–167.

[8] V.K. Sabelfeld: The logic-termal equivalence is polynomial-time decidable. *Information Processing Letters*, 1980, v. 10, N 2, p. 57-62.

[9] T.A. Novikova, V.A. Zakharov: Is it possible to unify programs?. The 27-th International Workshop on Unification, Epic Series, v. 19, 2013, p. 35-45.

[10] Wang Hao: Proving theorems by pattern recognition. *Bell System Technical Journal*. 1961, v. 40, N 1, p. 1-41.

[11] R. Berger: The undecidability of domino problem. *Memoirs of American Mathematical Society*, v. 66.

[12] V.A. Zakharov: On the decidability of the equivalence problem for orthogonal sequential programs. *Grammars*, v 2, N 3, p. 271-281.