Two-sided unification is NP-complete

Tatyana A. Novikova, Astana, Kazakhstan

Vladimir A. Zakharov, Moscow, Russia

taniaelf@mail.ru, zakh@cs.msu.su
**Unification as solving equations**

To unify a pair of expressions $E_1$ and $E_2$ means to compute a common instance of these expressions (syntactical unification), or, in other words, to solve the equation $E_1X = E_2X$, where a variable $X$ takes its value in some set of substitutions $\text{Subst}$.

If $E_1 = \theta$ and $E_2 = \eta$ are substitutions themselves then we have the equation

$$\theta X = \eta X$$

in a semigroup of substitutions.

When $\text{Subst}$ is the set of first-order substitutions, equations of this type are solvable in linear time [M.S. Paterson, M.N. Wegman, 1978].
Equations on substitutions

And what about the solvability of some other equations of the same kind in the semigroup of first-order substitutions?

\[ X_{1}^{\sigma_{1}} \theta X_{2}^{\sigma_{2}} = X_{3}^{\sigma_{3}} \eta X_{4}^{\sigma_{4}}, \]

where \( \sigma_{i} \in \{0, 1\} \),
\( X^{1} = X \),
\( X^{0} = \varepsilon \) (empty substitution).
Equations on substitutions

Some equations of this type are trivially solvable for every pair of substitutions $\theta$, $\eta$.

$$X_1 \theta X_2 = X_3 \eta X_4 :$$

$X_1 = \eta$, $X_4 = \theta$, $X_2 = X_3 = \varepsilon$.

$$X_1 \theta = X_3 \eta X_4 :$$

$X_1 = \eta$, $X_4 = \theta$, $X_3 = \varepsilon$.

$$X_1 \theta X_2 = \eta X_4 :$$

$X_1 = \eta$, $X_4 = \theta$, $X_2 = \varepsilon$. 
Equations on substitutions

Equations

\[ \theta X_2 = \eta X_4 \]

and

\[ \theta X_2 = \eta \]

correspond to conventional unification and matching problems; these equations are solvable in linear time.
Equations on substitutions

Equations

\[ X_1 \theta = X_3 \eta \]

and

\[ X_1 \theta = \eta \]

are decidable in polynomial time.

These equations appear in some equivalence checking techniques for sequential programs [Zakharov, 2000]
Equations on substitutions

And, finally, we arrive at the equations of the form

\[ X_1 \theta X_2 = \eta. \]

A pair of substitutions \((\xi_1, \xi_2)\) which gives a solution to this equation is called

a **two-sided unifier** of \(\theta\) and \(\eta\).

What does two-sided unification [matching] problem look like?

And what two-sided unifiers are good for?
String searching problem

Suppose that all functional symbols in the terms of $\theta$ and $\eta$ are unary, i.e. all terms are words.

Then the equation

$$X_1 \theta X_2 = \eta$$

is but an algebraic definition of string searching problem — find a place where one string (pattern) $\theta$ occurs within another string (text) $\eta$.

Thus, two-sided unification problem may be viewed as a generalization of string searching problem.
**Program refactoring**

Let $\pi_0(\vec{x}:\text{input}; \vec{y}:\text{output})$ a library subroutine with a set of formal input arguments $\vec{x}$ and a set of formal output parameters $\vec{y}$. Suppose that one wants to make program refactoring by replacing some piece of program code $\pi_1$ with an appropriate call of the subroutine.

To this end one could try to find such instantiation $\eta''$ of input arguments $\vec{x}$ and such specialization $\eta'$ of output parameters $\vec{y}$ as to make the composition of $\eta'$, $\pi_0$, and $\eta''$ equivalent to $\pi_1$.

In some formal models of programs a behavior of a program $\pi$ can be specified by a substitution $\theta_\pi$ which assigns terms on input arguments $\vec{x}$ to output parameters $\vec{y}$.

Thus, we arrive at **two-sided unification problem** : given a pair of substitutions $\theta_{\pi_0}$ and $\theta_{\pi_1}$ find a pair of substitutions $\eta''$ (**input instantiation**) and $\eta'$ (**output specialization**) such that

$$\eta'\theta_{\pi_0}\eta'' = \theta_{\pi_1}.$$
Two-sided unifiability is NP-complete

Lemma 1. Two-sided unification problem is in NP.

Proof: Given a tree representation of a pair of substitutions $(\theta, \eta)$ ...
Two-sided unifiability is NP-complete

**Lemma 1.** Two-sided unification problem is in NP.

**Proof:** ...guess an arbitrary double cut in the tree representation of substitution $\eta$, ...

![Diagram of substitutions $\eta$, $\eta_1$, $\eta_2$, $\eta_3$, and $\theta$.]
Two-sided unifiability is NP-complete

**Lemma 1.** Two-sided unification problem is in NP.

**Proof:** ...and check that substitution $\theta$ includes all terms from the middle section.
BOUNDED TILING problem

Proposed by Hao Wang in 1961.
Unbounded variant is undecidable [R. Berger, 1966].
Bounded variant is NP-complete [C.H. Lewis, 1978] and widely used as a "NP-yardstick".
BOUNDDED TILING problem

BOUNDDED TILING problem is specified by

- a size \( n \times m \) of the area to be tiled;
- a colouring \( B \) of the boarder segments (boundary constraint);
- a set of available tiles \( Tiles \).

Tiling is any map
\[
T : \{(i,j) : 1 \leq i \leq n, 1 \leq j \leq m\} \rightarrow Tiles
\]

Tiling \( T \) is consistent if colors on adjacent tile edges and boarder segments match.
Two-sided unifiability is NP-complete

Lemma 2. BOUNDED TILING is log-space reducible to two-sided unification problem.

Prof: Encode all components of BOUNDED TILING problem by terms.

\[ f_1(y) : f(y) \]

\[ f_2(y) : f(f(y)) \]

\[ f_3(y) : f(f(f(y))) \]
Two-sided unifiability is NP-complete

Instances of tiles

\[
t_{i,j,\ell_1} : h(f_i(y_0), f_j(y_0), f_1(y_{i+0.5,j}), f_2(y_{i,j+0.5}), f_3(y_{i-0.5,j}), f_2(y_{i,j-0.5}))
\]
Two-sided unifiability is NP-complete

Instances of neighboring tiles

\[ j \quad j + 1 \]

Tiles \( \ell_1 \) and \( \ell_2 \)

\[ t_{i,j,\ell_1} : \]
\[ h(f_i(y_0), f_j(y_0), f_1(y_{i+0.5,j}), f_2(y_{i,j+0.5}), f_3(y_{i-0.5,j}), f_4(y_{i,j-0.5})) \]

\[ t_{i,j,\ell_2} : \]
\[ h(f_i(y_0), f_{j+1}(y_0), f_4(y_{i+0.5,j+1}), f_3(y_{i,j+1.5}), f_1(y_{i-0.5,j+1}), f_2(y_{i,j+0.5})) \]
Two-sided unifiability is NP-complete

Proof: The substitution $\theta$ (in the equation $X_1 \theta X_2 = \eta$) specifies all possible insertions of tiles from Tiles onto the squares of the Area.

$$\theta = \{x_{i,j,\ell}/t_{i,j,\ell} : 1 \leq i \leq n, 1 \leq j \leq m, \ell \in \text{Tiles}\}$$
Two-sided unifiability is NP-complete

Using functional symbol $g^{(2)}$ build an arbitrary term $t_{\text{area}}$ which has $nm$ argument positions (leaves in the tree representation of the term); every argument position in this term stands for a square in the $\text{Area}$.
Two-sided unifiability is NP-complete

*Proof:* For every square \((i, j)\) in the *Area* define the term

\[
\hat{t}_{i,j} = h(f_i(u), f_j(u), f_K(u), f_K(u), f_K(u), f_K(u))
\]

It corresponds to an insertion of a monochromatic tile onto the square \((i, j)\) of the *Area* — all edges are painted maximal colour \(K\).

The substitution \(\eta\) (in the equation \(X_1 \theta X_2 = \eta\)) specifies the consistent covering of the *Area* with \(K\)-monochromatic tiles.

\[
\eta = \{z/ t_{\text{area}}(\hat{t}_{1,1}, \hat{t}_{1,2}, \ldots, \hat{t}_{n,m})\}
\]
Two-sided unifiability is NP-complete

Proof: A pair of substitutions $(\xi_1, \xi_2)$ pretending to be a two-sided unifier of the pair $(\theta, \eta)$ must
1) make a tiling $T$ of $\text{Area}$ by choosing appropriate arguments $x_{i,j,\ell}$ of the term $t_{\text{area}}$:
$$\xi_1 = \{ z / t_{\text{area}}(x_{1,1}, T(1,1), x_{1,2}, T(1,2), \ldots, x_{n,m}, T(n,m)) \}$$
2) repaint accordingly and monochromatically all tiles of $T$:
$$\xi_2 = \{ y_0/u, \ldots, y_{i,j+0.5} / f_{K-c(i,j+0.5)}(u), \ldots \},$$
where $c(i,j+0.5)$ is the colour of the adjacent edges of neighboring tiles $T(i,j)$ and $T(i,j+1)$. 
Two-sided unifiability is NP-complete

**Proof:** Conclusion:

A two-sided unifier \((\xi_1, \xi_2)\) of the pair \((\theta, \eta)\) exists iff the BOUNDED TILING problem \((n, m, K, \text{Tiles})\) has a solution (consistent tiling). 

\(\square\)
Final remark

Two-sided unification as a hybrid problem

Unification (matching) problem
Complexity: $O(n)$

Substring Searching problem
Complexity: $O(n)$

Two-sided unification problem
Complexity: NP-complete
Thank you for attention