Bounded Second-Order Unification is NP-Complete*

Jordi Levy[1], Manfred Schmidt-Schauß[2], and Mateu Villaret[3]

[1] IIIA, CSIC, Campus de la UAB, Barcelona, Spain.
http://www.iiia.csic.es/~levy
[2] Institut für Informatik, Johann Wolfgang Goethe-Universität,
Postfach 11 19 32, D-60054 Frankfurt, Germany.
http://www.ki.informatik.uni-frankfurt.de/persons/schauss/schauss.html
[3] IMA, UdG, Campus de Montilivi, Girona, Spain.
http://ima.udg.es/~villaret

Abstract. Bounded Second-Order Unification is the problem of deciding, for a given second-order equation $t \stackrel{?}{=} u$ and a positive integer m, whether there exists a unifier σ such that, for every second-order variable F, the terms instantiated for F have at most m occurrences of every bound variable.
It is already known that Bounded Second-Order Unification is decidable and NP-hard, whereas general Second-Order Unification is undecidable. We prove that Bounded Second-Order Unification is NP-complete, provided that m is given in unary encoding, by proving that a size-minimal solution can be represented in polynomial space, and then applying a generalization of Plandowski's polynomial algorithm that compares compacted terms in polynomial time.

1 Introduction

Second-order unification (SOU) is a generalization of first-order unification, where variables are permitted also at the position of function symbols, hence they may have arguments. These variables are also called second-order variables. When solving an equation, the second-order variables can stand for an arbitrary first-order term with holes for plugging in the arguments, which must be terms. In lambda-notation, a second-order variable may be instantiated by a term $\lambda x_1 \ldots \lambda x_n \,.\, t$, where t is a first-order term, and the variables x_i also stand for first-order terms. SOU extends the expressivity of first-order unification, and is a restriction of higher-order unification (see [6, 3]). It is known that SOU is undecidable [5], even under severe syntactic restrictions [4, 20, 10, 11].

A decidable variant is bounded second-order unification (BSOU) [17], which restricts the possible instantiations of second-order variables by limiting the number of occurrences of bound variables. However, the described algorithm

* This research has been partially founded by the CICYT research projects iDEAS (TIN2004-04343) and Mulog (TIN2004-07933-C03-01).

for BSOU has non-elementary complexity. Recently, we described an improved algorithm for monadic SOU [9] —which is BSOU where only unary function symbols and constants are permitted— and determined its complexity to be NP-complete.

In this paper we apply and extend methods used in [9] for monadic SOU to improve the BSOU algorithm by compressing the computed solution, and as a main result we prove that BSOU is in NP, which means that it is NP-complete. To obtain this result requires compression techniques and, as a basis, the BSOU-algorithm in [17]. This result shows that BSOU may become a practically useful restriction of higher-order unification, perhaps via using a SAT-solver.

To illustrate the difficulties in proving the membership of BSOU in NP, we will compare this problem with other unification problems. Most general first-order unifiers σ have a very nice property: for every solvable equation E and variable X_i there exists a subterm t_i of the equation E such that σ can be decomposed in the form $\sigma = [X_1 \mapsto t_1] \circ \cdots \circ [X_n \mapsto t_n]$. This representation is polynomial and ensures that the problem is in NP. In [13] it is proved that the search of these subterms can be done very efficiently and the problem is in fact linear. Well-nested context unifiers [8] —this is, context unifiers where instances of variables do not overlap— have the same property, but replacing subterms t_i by subcontexts c_i of the equation. This property is used to prove that well-nested context unification is in NP. However, in this case the search of these subcontexts cannot be done efficiently. The property held by these two problems suggested us to represent substitutions as compositions of instantiations to save space. In monadic SOU we have a weaker property: instead of just one subcontext, we need to compose a bounded number of subcontexts, and in some cases raise them to an exponent. Moreover, we get the instance of *only one* of the variables $[X_1 \mapsto t_1]$. This means that we have to use the same property applied to $[X_1 \mapsto t_1]\, E$ to find the instance of another variable. Notice that $[X_1 \mapsto t_1]\, E$ may be bigger than E, and the size of the instance of X_n could be exponential in n. In [9] it is proved that this is not the case, if we represent such an instance using a context free grammar (CFG). If we have a CFG generating E, to represent a subcontext of E, we have to enlarge the grammar, and in the worst case to duplicate the size, obtaining an exponential representation. To avoid this problem in [9] we propose the conjugation of size and *depth* of the grammar —the depth of the parse tree— , which has an effect similar to balancing conditions. Then, the representation of a subcontext preserves the depth and requires to increase the size of the grammar only on the depth (see Lemma 4). Showing PSPACE [16] as an upper complexity bound for stratified context unification used an ad-hoc compression technique composed of sharing and exponentiation. The algorithm given there does not look for a polynomial-sized solution, and erases partial solution as early as possible to keep the PSPACE-bound.

Compared with monadic SOU, the situation in BSOU is even worse. Given an equation E we can only find a "partial" instance of some variable. This means that we remove a variable, say X, but we have to introduce new variables, say X', by instantiating $[X \mapsto c[X'[\bullet]]]$ where c is a context composed by a bounded

number of subcontexts of E. Moreover, this context is not ground, contrarily to the monadic SOU case. Fortunately, we have a well-founded ordering where $[X \mapsto c[X'[\bullet]]]\, E$ is smaller than E.

This paper proceeds as follows. After some preliminaries, we define an extension of singleton CFG for trees in Section 3. In Section 4 we define an order on equations and show a polynomial upper bound for the length of decreasing sequences. Then, in Section 5 we prove that given an equation E, and a size-minimal solution σ, we can find a polynomial-sized partial description ρ of σ, such that $\rho(E)$ is strictly smaller than E and $\sigma = \sigma' \circ \rho$. Finally, in Section 6 we show how we can get a compact representation of these partial instantiations, and represent σ in polynomial size. Using an extension of Plandowski's [14, 15] result for CFG, we can check in polynomial time if a substitution in such a representation is a solution, proving that BSOU is in NP.

2 Preliminary Definitions

We consider one base (first-order) type o, and second-order types described by the syntax $\tau ::= o \to o \mid o \to \tau$, with the usual convention that $\to$ is associative to the right. We deal with a *signature* $\Sigma = \bigcup_{i \geq 0} \Sigma_i$, where *constants* of Σ_i are i-ary, and a set of *variables* $\mathcal{X} = \bigcup_{i \geq 0} \mathcal{X}_i$, where variables of $\mathcal{X}_i$ are also i-ary. Variables of $\mathcal{X}_0$ are therefore first-order typed and those of $\mathcal{X}_i$, with $i \geq 1$, are second-order typed, and similarly for Σ. We use the convention that X, Y (possibly with primes and subindexes) mean free first-order or second-order variables (unknowns), while constants are denoted by lower-case letters $a, b, \ldots$, for first-order, and $f, g, \ldots$, for second-order ones.

Terms are built as usual in simply typed λ-calculus. We assume that they are in $\beta\eta$-long normal form, or are immediately reduced, so we will use a first-order notation, if possible. We denote terms with lower case letters like $t, u, \ldots$.

Contexts are first-order typed terms with *one hole* at some position, notated as $\bullet$. We call *Z-contexts* to the union of first-order terms and contexts, hence they may contain zero or one hole. We denote contexts and Z-contexts by lower case letters: $c, d, \ldots$ for contexts and $c, d, \ldots, t, u, \ldots$ for Z-contexts. If the Z-context d is plugged into the hole of a Z-context c, we denote the result as the Z-context $c[d]$. (In the special case that c is a term, $c[d] = c$). We sometimes abbreviate $c_1[c_2[c_3 \ldots]]$ as $c_1\, c_2\, c_3 \ldots$ and $c[c[c \overset{n}{\ldots}]]$ as c^n. For any pair of Z-contexts c_1 and c_2, if for some Z-context d we have $c_1 = c_2[d]$, then c_2 is said to be a *prefix* of c_1 (notated $c_2 \preceq c_1$ and $c_2 \prec c_1$ for strict prefixes), and if for some context d (with hole) we have $c_1 = d[c_2]$, then c_2 is said to be a *suffix* of c_1. Notice that, if c_2 is a suffix of c_1, then c_1 contains a hole iff c_2 contains a hole. On the contrary, a *subterm* u of a context c does not need to contain a hole. This distinguishes a suffix from a subterm. If c is a prefix of a subterm of d, then c is called a *subcontext* of d. The *size* of a Z-context c is denoted $|c|$, and defined as its number of symbols (including the hole).

We use *positions* in terms, noted p, q, as sequence of positive integers following Dewey's notation. The empty word is notated ϵ, $p \prec q$ notates the prefix relation,

$p \cdot q$ the concatenation, and $t|_p$ the subterm at position p of t. For a context c, its *main path* is the position of the hole. A position p is in the main path of c if p is a prefix of the main path of c.

Second-order *substitutions* are functions from terms to terms, defined as usual. The application of a substitution σ to a term t is written $\sigma(t)$. An instance of the bounded second-order problem (BSOU) is an *equation* $t \stackrel{?}{=} u$, where t and u are first-order terms, and a number m given in unary encoding. The set of variables (unknowns) occurring in an equation E is denoted by $Var(E)$, and the notational size by $|E|$. We assume that equations are symmetric. A substitution σ is said to be a *unifier* of $(t \stackrel{?}{=} u, m)$, iff $\sigma(t) = \sigma(u)$, and for all $X \in Var(E)$ every bound variable in $\sigma(X)$ occurs at most m times. A unifier σ is said to be a *solution* of (E, m), iff $\sigma(t)$ and $\sigma(u)$ are ground (do not contain free variables).

It is easy to see that it suffices to consider only unifiers and solutions built from constant, and function symbols that occur in E. A solution σ of $(t \stackrel{?}{=} u, m)$ is said to be *size-minimal* if it minimizes $|\sigma(t)|$ among all solutions of $(t \stackrel{?}{=} u, m)$.

As already shown in [17], there is an NP-reduction of BSOU to the specialized problem, where $m = 1$, and every second-order variable is unary. Hence in the following, we will only treat this case. In the simplification of the problem we go a step further by considering only second-order variables. To do so we can replace all occurrences of the first-order variable X by the term $X'(a)$ where X' is a fresh (unary) second-order variable and a is any 0-ary constant. This transformation allow us to P-reduce BSOU to BSOU without first-order variables. Therefore, from now on, all variables will have type $o \to o$, and all terms type o, or $o \to o$. Moreover, we will represent second-order typed terms $\lambda y\,.\,t$ as the Z-context resulting from replacing in t the occurrence of y (if any) by the hole. Thus, from now on, we will only deal with Z-contexts, and terms will be assumed to be first-order typed.

We know that size-minimal solutions of a BSOU equation satisfy the exponent of periodicity lemma [12, 7, 19, 17]. However, since we have a slightly different definition of size-minimality, after some encoding by enlarging E, we have a quadratic dependency on $|E|$:

Lemma 1 ([17], Lemma 4.1). *There exists a constant $\alpha \in \mathbb{R}$ such that, for every BSOU-problem E, every size-minimal unifier σ, and every variable X, if d^n is a nonempty subcontext of $\sigma(X)$, then $n \leq 2^{\alpha|E|^2}$.*

3 Singleton Tree Grammars

We generalize singleton context free grammars (SCFG) to trees, since we require a device for a compressed representation of solutions. We extend the expressivity of SCFGs by permitting terms and contexts. The definition is a special case of the context free tree grammars defined in [2].

Definition 1. *A* singleton tree grammar (STG) *is a tree grammar, i.e. a 4-tuple $(\mathcal{TN}, \mathcal{CN}, \Sigma, R)$, where $\mathcal{TN}$ are tree nonterminals, $\mathcal{CN}$ are context nonterminals, and Σ is a signature of terminals symbols (variables and constants), such*

that the sets $\mathcal{TN}$, $\mathcal{CN}$, Σ are pairwise disjoint. The set of nonterminals $\mathcal{N}$ is defined as $\mathcal{N} = \mathcal{TN} \cup \mathcal{CN}$. The rules in R may be of the form:

- $A ::= f(A_1, \ldots, A_n)$, *where* $A, A_i \in \mathcal{TN}$, *and* $f \in \Sigma$ *is an* n*-ary terminal symbol.*
- $A_1 ::= C[A_2]$ *where* $A_1, A_2 \in \mathcal{TN}$, *and* $C \in \mathcal{CN}$.
- $C_1 ::= C_2C_3$, *where* $C_i \in \mathcal{CN}$.
- $C ::= f(A_1, \ldots, A_{i-1}, [\bullet], A_{i+1}, \ldots, A_n)$, *where* $A_i \in \mathcal{TN}$, $C \in \mathcal{CN}$, $[\bullet]$ *is the hole, and* $f \in \Sigma$ *an* n*-ary terminal symbol.*

The tree grammar must be non-recursive (the relation $\xrightarrow{+}$ has no cycles). Furthermore, for every non-terminal N there is exactly one rule having N as left hand side. Give a term t where nonterminals may occur, the derivation by G is an exhaustive iterated replacement of the nonterminals by the corresponding right hand sides.

Definition 2. *The* size *of a grammar (STG) G is the number of its rules and denoted as $|G|$.*
The depth *of a nonterminal D is defined as the maximal number of $\rightarrow_G$-steps from D, where $D' \rightarrow_G D''$ for two nonterminals D', D'', iff $D' ::= T$ is a rule of G, and D'' occurs in T.*
The depth of a grammar *is the maximum of the depths of all nonterminals.*
When a grammar G generates a Z-context t from a non-terminal symbol D (and the grammar is clear from the context) we write $depth(t)$ to denote $depth(D)$.

The following theorem is a generalization to trees of Plandowski's one in [14, 15].

Theorem 1 ([1, 18]). *Given an STG G, and two tree nonterminals from G, it is decidable in polynomial time depending on $|G|$ whether they generate the same tree or not.*

The following lemmas state how the size and the depth of a grammar are increased by extending it with concatenations, exponentiation, prefixes and suffixes of Z-contexts. Proofs may be adapted from the extended version of [9].

Lemma 2. *Let G be an STG defining the Z-contexts $d_1, \ldots, d_n$ for $n \geq 1$. Then there exists an STG $G' \supseteq G$ that defines the Z-context $d_1 \cdots d_n$ and satisfies $|G'| \leq |G| + n - 1$ and $depth(d_1 \cdots d_n) \leq \max\{depth(d_1), \ldots, depth(d_n)\} + \lceil \log n \rceil$.*

Lemma 3. *Let G be an STG defining the context d. For any $n \geq 1$, there exists an STG $G' \supseteq G$ that defines the context d^n and satisfies $|G'| \leq |G| + 2 \lfloor \log n \rfloor$ and $depth(d^n) \leq depth(d) + \lceil \log n \rceil$.*

Lemma 4. *Let G be an STG defining the context d. For any nontrivial prefix or suffix context d' of d, there exists an STG $G' \supseteq G$ that defines d' and satisfies $|G'| \leq |G| + depth(d)$ and $depth(d') \leq depth(d)$.*
Similarly if d is a Z-context and d' is a subterm of d.

Lemma 5. *Let G be an STG defining the term t. For any nontrivial prefix context d of the term t, there exists an STG $G' \supseteq G$ that defines d and satisfies $|G'| \leq |G| + 2\ depth(t)\ (\log(depth(t)) + 1)$ and $depth(d) \leq depth(t) + 2 + \log(depth(t))$.*

Notice that for prefixes of contexts we get better bounds than for prefixes of terms.

4 A Well-Founded Ordering on Equations

In this section we define an ordering on the equations. This order is similar to the one proposed in [17] to prove the decidability of BSOU. However, in our case, the order is not only well-founded: we prove that the length of any strictly decreasing sequence is polynomially bounded on the size of the first element.

Definition 3. *We say that p is a* surface position *of t if there are no variable occurrences strictly above p.*

Given an equation $E = (t \stackrel{?}{=} u)$, the relation $\approx_E \subseteq Var(E) \times Var(E)$ is the reflexive-transitive closure of the relation defined by: if X occurs at the surface position p of t and Y occurs at the same surface position p of u, then $X \approx_E Y$.

The relation $\succ_E \subseteq Var(E) \times Var(E)$ is the relation defined by: if X occurs at the surface position p of t and, for some nonempty sequence q, Y occurs at the surface position $p \cdot q$ of u, then $X \succ_E Y$. We extend this relation to classes of equivalences with if $X \succ_E Y$ then $\overline{X} \succ_E \overline{Y}$.

If p is a surface position of t and of u, then $t|_p \stackrel{?}{=} u|_p$ is called a subequation *of $t \stackrel{?}{=} u$.*

In first-order unification all variable occurrences are at surface positions. Moreover, if $\succ_E^+$ is not irreflexive then there is occur-check and the equation is unsolvable. In second-order unification this is not the case, $\succ_E^+$ may be not irreflexive and E solvable.

Definition 4. *A cycle in an equation $E = (t \stackrel{?}{=} u)$ is a sequence of variables $X_1, \ldots, X_n$ and pairs of positions $\langle p_1, p_1 \cdot q_1 \rangle, \ldots, \langle p_n, p_n \cdot q_n \rangle$, such that, for $i = 1, \ldots, n$, X_i is at the surface position p_i of t, and X_{i+1} is at the surface position $p_i \cdot q_i$ of u, and there is at least one nonempty q_i.*[1]

The length of the cycle is n.

Notice that an equation E contains a cycle iff the relation $\prec_E^+$ for classes of equivalences is not irreflexive. The shortest cycle in an equation E is shorter than $|\,Var(E)|$.

Definition 5. *Given an equation E, the measure $\mu(E)$ is a lexicographic combination $\langle \mu_1(E), \mu_2(E), \mu_3(E) \rangle$ of the following components:*

1. *$\mu_1(E) = |\,Var(E)|$ is the number of variables occurring in E.*

[1] When the length n of the cycle is clear from the context, all indexes i greater than n are replaced by $((i-1) \mod n) + 1$.

2. $\mu_2(E)$ *is the length of the shortest cycle in* E*, or* ∞ *if there are no cycles.*
3. $\mu_3(E)$ *is zero, if* E *contain cycles, otherwise*

$$\mu_3(E) = |\,Var(E)| - |\,Var(E)/\approx_E| + 2|\succ_E| = \sum_{C\,\in\,Var(E)/\approx_E} (|C|-1) \quad + \sum_{\substack{X,Y\in Var(E)\\ X\succ_E Y}} 2$$

Lemma 6. *Any decreasing sequence of equations* $\{E_i\}_{i\geq 1}$*, i.e. where* $\mu(E_i) > \mu(E_{i+1})$*, terminates in at most* $2\,|\,Var(E_1)|^3$ *steps.*

Proof: Let $n = |\,Var(E_1)|$. The first component of $\mu(E_i)$ can have values from $j = n, \ldots, 1$. When the first component is j, the second component can have values from $\infty, j, \ldots, 1$. When there are no cycles, the third component is maximal when all the equivalence classes are singletons, and is $j(j-1)$. Therefore, the set of possible values of $\mu(E_i)$ is smaller than $\sum_{j=1}^{n} j + j\,(j-1) + 1 = 1/3n^3 + 1/2n^2 + 7/6n \leq 2n^3$. ■

5 Finding the Partial Instance of Some Variable

In this section we show how, given an equation E and a minimal solution σ, we can find an instantiation $[X \mapsto t]$ or a partial instantiation $[X \mapsto c[X'(\bullet)]]$ for every variable $X \in Var(E)$ such that the composition ρ of all them satisfies $\sigma = \sigma' \circ \rho$, where σ' is a size-minimal solution of $\rho(E)$, and the new equation $\rho(E)$ is smaller than E w.r.t. μ. Moreover the (partial) instantiation can be built up from a linear number of pieces (subcontexts) of E, which as we show in the next section, ensures that it can be efficiently represented.

Lemma 7 (Partial instance). *Given an equation* E *and a size-minimal solution* σ*, with exponent of periodicity bounded by* e*, there exist substitutions* $\rho = \rho_2 \circ \rho_1$ *such that the* ρ_i*'s have the form*

$$[X_1 \mapsto c_1[X_1'(\bullet)]\ , \ldots,\ X_n \mapsto c_n[X_n'(\bullet)]]$$

such that:

1. $n \leq |\,Var(E)|$,
2. $X_1', \ldots, X_n'$ *are fresh variables not occurring in* E,
3. *the Z-contexts* c_i *can be constructed taking* $\mathcal{O}(n)$*-many subcontexts of* E *[or of* $\rho_1(E)$ *in the case of* ρ_2*], composing them, raising the result to some exponent smaller than* e *and taking a prefix,*
4. ρ *is coherent with* σ*, i.e.* σ *decomposes as* $\sigma = \sigma' \circ \rho$*, for some* σ'*, and*
5. $\mu(E) > \mu(\rho(E))$.

Remark 1. Notice that Lemma 7 and 6 allow us to decompose $\sigma = \rho_m \circ \cdots \circ \rho_1$, where m is polynomial on the size of the original equation E, and ρ_i can be represented polynomially on the size of $\rho_{i-1} \circ \cdots \circ \rho_1(E)$ using singleton tree grammars. From this we can only conclude that σ has a representation bounded

by a composition of a polynomial number of polynomials, i.e. that σ has an exponential-size representation. Obviously, this is not enough for proving the NP-completeness of BSOU. We need an important result that will be proved in Section 6.

Lemma 7 is proved in the following subsections. We also need the following Lemma.

Lemma 8. *If σ is a size-minimal solution of E, and σ decomposes as $\sigma = \sigma' \circ \rho$, then σ' is a size-minimal solution of $\rho(E)$.*

5.1 There Are Cycles in the Set of Equations

If $E = (t \stackrel{?}{=} u)$ contains a cycle defined by $X_1, \ldots, X_n$ and $\langle p_1, p_1 \cdot q_1 \rangle, \ldots, \langle p_n, p_n \cdot q_n \rangle$, then, for every $i = 1, \ldots, n$, we have a subequation $t|_{p_i} \stackrel{?}{=} u|_{p_i}$ of the form

$$X_i(v_i) \stackrel{?}{=} c_i[X_{i+1}(w_i)]$$

for some terms v_i and w_i, and some context c_i that has its hole at position q_i and has no variables in its main path. Note that there is at least one context c_i different from $\bullet$. The unifier σ of $t \stackrel{?}{=} u$ has to solve all these subequations.

Now we find how long each variable "stays" in the cycle: For $i = 1, \ldots, n$, let r_i be the longest prefix of $(q_i \cdot \ldots \cdot q_n \cdot q_1 \cdot \ldots \cdot q_{i-1})^\infty$ such that, if $\sigma(X_i)$ has no hole, then r_i is a position inside the term $\sigma(X_i)$ and, if $\sigma(X_i)$ has a hole, then this hole must be below or at position r_i.

We select a minimal r_i: Let $\mathit{minlength} = \min_{i \in \{1,\ldots,n\}} |r_i|$, and assume w.l.o.g. that r_1 is minimal, i.e. $\mathit{minlength} = |r_1|$.

We make all variables copy along this distance: For $i = 1, \ldots, n$, let s_i be the prefix of $(q_i \cdot \ldots \cdot q_n \cdot q_1 \cdot \ldots \cdot q_{i-1})^\infty$ of length *minlength*, and let d_i be the context resulting from putting a hole at position s_i of $(c_i \ldots c_n\, c_1 \ldots c_{i-1})^\infty$. Note that, since the exponent of periodicity of σ does not exceed e, then d_i has the form $(c_i \ldots c_n\, c_1 \ldots c_{i-1})^{e_i}\, d'_i$ where $e_i \leq e$ and the context d'_i is a prefix of $c_i \ldots c_n\, c_1 \ldots c_{i-1}$.

Since d_i is a prefix of $\sigma(X_i)$, the substitution $\rho_1 = [X_1 \mapsto d_1[X'_1(\bullet)], \ldots, X_n \mapsto d_n[X'_n(\bullet)]]$ is coherent with σ. Moreover, the sequences $X'_1, \ldots, X'_n$ and $\langle p_1 \cdot s_1,\ p_1 \cdot q_1 \cdot s_2 \rangle,\ \langle p_2 \cdot s_2,\ p_2 \cdot q_2 \cdot s_3 \rangle,\ \ldots,\ \langle p_n \cdot s_n,\ p_n \cdot q_n \cdot s_1 \rangle$ define a cycle in $\rho_1(E)$ of the same length as the original cycle. Now, we define a new substitution ρ_2 such that $\rho = \rho_2 \circ \rho_1$ is coherent with σ and $\mu(\rho(E)) < \mu(E)$. There are three cases:

Case 1: If $\sigma(X_1)$ does not contain any hole, then r_1 corresponds to the position of a first-order constant in $\sigma(X_1)$. Since r_1 and $q_1 \cdot r_2$ are both prefixes of $(q_1 \cdot \ldots \cdot q_n)^\infty$ and $|r_1| \leq |r_2|$, r_1 is a prefix of $q_1 \cdot r_2$. Since σ solves $X_1(v_1) = c_1[X_2(w_1)]$, and r_2 is a position inside $\sigma(X_2)$, $\sigma(X_2)$ has a first-order constant at position r_2 and $r_1 = q_1 \cdot r_2$. Therefore, since $|r_1| \leq |r_2|$, we have $q_1 = \epsilon$ and $c_1 = \bullet$. Thus, $|r_2| = \mathit{minlength}$ and $\sigma(X_2)$ also has a constant at position r_2. Repeating this argument we would get $c_i = \bullet$, for

every $i = 1, \ldots, n$, which contradicts the assumption that we have a cycle (for some i, $c_i \neq \bullet$). Therefore this situation is not possible.

Case 2: If for some $i = 1, \ldots, n$, s_i corresponds to the position of the hole in $\sigma(X_i)$ then take $\rho_2 = [X'_i \mapsto \bullet]$. The variable X_i is completely instantiated, and the first component of μ decreases. This situation corresponds to some variable that "finishes inside the cycle, i.e. it is completely instantiated".

Case 3: Otherwise, r_1 corresponds to some *proper* prefix of the hole position of $\sigma(X_1)$. Let m be the minimal index such that $c_1 = \cdots = c_{m-1} = \bullet$ and $c_m \neq \bullet$. Notice that $q_1, \ldots, q_{m-1}$ are empty, $r_1 = s_1 = \cdots = s_{m-1}$, and, for $j = 2, \ldots m-1$, $r_1 \prec r_j$ strictly. For $i = 1, \ldots, m-1$, let $l_i \in \mathbb{N}$ satisfy: the hole of $\sigma(X_i)$ is below or at $r_1 \cdot l_i$, if $\sigma(X_i)$ has a hole, or $l_i = 1$, otherwise. Let $l \in \mathbb{N}$ satisfies $r_1 \cdot l \preceq q_m \cdot s_{m+1}$. The equation $\rho_1(E)$ contains as subequations $\{X'_1(\rho_1(v_1)) \stackrel{?}{=} X'_2(\rho_1(w_1)), \cdots, X'_{m-1}(\rho_1(v_{m-1})) \stackrel{?}{=} X'_m(\rho_1(w_{m-1})), X'_m(\rho_1(v_m)) \stackrel{?}{=} \rho_1(c_m[d_{m+1}[X'_{m+1}(w_m)]]|_{r_1})\}$ where r_1 is a proper prefix of the main path of $c_m[d_{m+1}[\bullet]]$, i.e. $r_i \prec q_m \cdot s_{m+1}$. Let f be the constant at the root of $c_m[d_{m+1}[X'_{m+1}(w_m)]]|_{r_1}$. We take

$$\rho_2 = [X'_i \mapsto f(w_i^1, \ldots, w_i^{l_i-1}, X''_i(\bullet), w_i^{l_i+1}, \ldots, w_i^{arity(f)})]_{i \in \{1,\ldots,m\}}$$

where, for $k \neq l_m$, $w_m^k = \rho_1(c_m[d_{m+1}[X'_{m+1}(w_m)]]|_{r_1 \cdot k})$. And, for every $i = m-1, \ldots, 1$, let $X'_i(\rho_1(v_i)) \stackrel{?}{=} X'_{i+1}(\rho_1(w_i))$ be the subequation of $\rho_1(E)$. For every $k \neq l_i, l_{i+1}$, we have $w_i^k = w_{i+1}^k$ and, if $l_i \neq l_{i+1}$, $w_i^{l_{i+1}} = X'_{i+1}(\rho_1(w_i))$. The new equation $\rho_2 \circ \rho_1(E)$ contains a cycle defined by the variables $X'_{m+1}, \ldots, X'_n$ and the variables X''_i that do not leave the cycle, i.e. that satisfy $l_i = l$. Among the pairs of positions we have $\langle p_n \cdot s_n,\ p_n \cdot q_n \cdot q_m \cdot s_{m+1} \rangle$. This cycle is shorter than the original one because $l_1 \neq l$. This situation corresponds to some variable that "leaves the cycle". Notice that in the special situation where $n = 1$, we always fall into case 2.

5.2 There Are No Cycles

If the surface positions of variables in t are the same as in u, then either $\sigma(X) = a$ or $\sigma(X) = \bullet$, for every $X \in Var(E)$. Therefore if we take $\rho = \sigma$ we fulfil the requirement of the lemma. Notice that the size-minimality of σ is only needed in this point and in the exponent of periodicity lemma.

Otherwise, there exists a $\prec_E^*$-maximal $\approx_E$-equivalence class $\{X_1, \ldots, X_n\}$ such that, there exists a variable (assume w.l.o.g. that it is X_1) and a surface position q of X_1 in t, satisfying $u|_q$ has not a variable in the root. Let $v = u|_q$, then $X_1(\ldots) \stackrel{?}{=} v$ is a subequation of E. We consider two cases:

Case 1 If, for all $i = 1, \ldots, n$, $\sigma(X_i)$ does not contain the hole, then take $\rho = [X_1 \mapsto v, \ldots, X_n \mapsto v]$. It is easy to prove that ρ is coherent with σ, and since it completely instantiates some variable, $\mu(\rho(E)) < \mu(E)$.

Case 2 Otherwise, let p be the largest sequence such that, for all $i = 1, \ldots, n$

1. if $\sigma(X_i)$ contains a hole, then p is a prefix of this hole occurrence,
2. if $\sigma(X_i)$ does not contain a hole, then p is inside $\sigma(X_i)$, and

3. for any q and r, if q is a surface occurrence of X_i in t, and $u|_q$ has not a variable on the root, and $q \cdot r$ is a surface occurrence of a variable in u, then $r \not\succ p$.

Notice that p is a position of v, and there are not variables in v above or at p. Roughly speaking, p is the result of following the main path of the Z-contexts $\sigma(X_i)$ until they split, or someone finish, or we find another variable below. Let c be the context resulting of putting a hole at position p of v. Then $\rho_1 = [X_1 \mapsto c[X_1'(\bullet)], \ldots, X_n \mapsto c[X_n'(\bullet)]]$ is coherent with σ. Moreover $X_1', \ldots, X_n'$ belong to the same equivalence class of $\rho_1(E)$, and $X_1'(\ldots) \stackrel{?}{=} \rho_1(v|_p)$ is a subequation of $\rho_1(E)$. Now there are three possibilities:

Case 2a For some $i = 1, \ldots, n$, the hole of $\sigma(X_i)$ is at position p. Then take $\rho = [X_i' \mapsto \bullet] \circ \rho_1$. The first component of μ decreases. This situation corresponds to the case when one of the main paths finish.

Case 2b If there exists a surface position q of some variable X_i in t, and $q \cdot p$ is a surface position of some variable Y in u (hence $X_i \succ_E Y$) then take $\rho = \rho_1$. This situation corresponds to the case when we have found a variable Y (belonging to a smaller equivalence class) before two main paths split or some one finishes.

In this situation $\rho(E)$ either contains a cycle, or the equivalence class $C = \{X_1, \ldots, X_n\}$ is merged getting $C' = \{X_1', \ldots, X_n'\} \cup \overline{Y} \cup \ldots$. In the second case, $|C'| > |C|$, but we pass from $X_i \succ_E Y$ to $X_i' \not\succ_{\rho(E)} Y$, and no new $\succ$ related pairs are added. The increasing in the first term of μ_3 is strictly compensated by the decreasing in the second term of μ_3.

Case 2c For every $i = 1, \ldots, n$, let $l_i \in \mathbb{N}$ satisfy: if $\sigma(X_i)$ has a hole, then it is below or at $p \cdot l_i$, otherwise, $l_i = 1$. If cases 2a and 2b does not apply, then there exists at least two distinct l_i's. This situation corresponds to the case when two main paths of variable instantiations split.

Let f be the constant symbol at the root of $v|_p$.We take $\rho = \rho_2 \circ \rho_1$ where

$$\rho_2 = [X_i' \mapsto f(w_i^1, \ldots, w_i^{l_i - 1}, X_i''(\bullet), w_i^{l_i + 1}, \ldots, w_i^{arity(f)})]_{i \in \{1, \ldots, n\}}$$

Now we show how the subterms w_i^k are chosen. First $w_1^j = \rho_1(v|_{p \cdot j})$, for every $j \neq l_1$, being $X_1(\ldots) \stackrel{?}{=} v$ a subequation of E. Then, for every $i, j = 1, \ldots, n$, if $X_i(w') \stackrel{?}{=} X_j(w'')$ is a subequation of E, then $w_i^k = w_j^k$, for any $k \neq l_i, l_j$, and, if $l_i \neq l_j$, then $w_i^{l_j} = X_j''(w'')$ and $w_j^{l_i} = X_i''(w')$. The existence of a connection between any pair of variables of the same equivalence class ensures that we define all the w_i^k's. We can prove that ρ is coherent with σ. Moreover ρ_2 can be built up from a linear number of pieces of $\rho_1(E)$, and ρ_1 from a linear number of pieces of E.

If we compare $\approx_E$ and $\succ_E$ with $\approx_{\rho(E)}$ and $\succ_{\rho(E)}$, we see that the equivalence class $C = \{X_1, \ldots, X_n\}$ has been split into $arity(f)$ (possibly empty) subsets $C_k = \{X_i'' \mid l_i = k\}$. The existence of two distinct l_i's ensures that there are at least two nonempty of such equivalence classes, and the first term of μ_3 has decreased. There can also be merges between equivalence classes, but then the second term of μ_3 decreases and compensates the increasing in the first term of μ_3. There can also appear cycles, but then μ_2 decreases.

6 Compacting the Solutions

One of the key ideas to compact the representation of a unifier is notating it as a composition of instantiations $[X_1 \mapsto v_1] \circ \cdots \circ [X_n \mapsto v_n]$. Another key idea is representing the Z-contexts v_i using a STG. Finally, the representation of the instance of a variable may involve the computation of subcontext of a term t represented as $t = [X_1 \mapsto v_1] \circ \cdots \circ [X_n \mapsto v_n]\, u$. In this section we show how this can be done efficiently without increasing very much the depth of the grammar.

To understand the main ideas, assume that the v_i's, t and u are words, and we have a grammar G that generates $A_i \to^* v_i$ and $A_0 \to^* u$. We can get a grammar G' that generates $B \to^* t$ replacing in G the variables X_i by the nonterminals A_i. This preserves the size of the grammar, but not the depth: in the worst case $\mathit{depth}(B) = \sum_{i=0}^{n} \mathit{depth}(A_i)$. This means that, to represent a prefix t' of t, we have to enlarge G' in $\mathit{depth}(B)$. A less expensive solution is finding a prefix v_i' of v_i and u' of u such that $t' = [X_1 \mapsto v_1] \circ \cdots \circ [X_n \mapsto v_n]\, u'\, v_n' \ldots v_1'$, and enlarge G in order to generate $B \to^* A_0'\, A_n' \cdots A_1' \to^* u'\, v_n' \ldots v_1'$. Then, in the worst case the depth is only $\mathit{depth}(B) = \log n + \max_{i=0}^{n}\{\mathit{depth}(A_i)\}$.

Definition 6. *We say that a term t is compactable as $t = [X_1 \mapsto v_1] \circ \cdots \circ [X_n \mapsto v_n] u$ with a grammar G, if*

1. *$X_i \neq X_j$, when $i \neq j$,*
2. *X_i does not occur in $v_1, \ldots, v_{i-1}$,*
3. *G generates v_i, for $i = 1, \ldots, n$, and u.*

Similarly when t and u are equations.

The following is a technical lemma used to handle the proof by induction of Lemma 10.

Lemma 9. *Let $\sigma = [X_1 \mapsto v_1] \circ \cdots \circ [X_n \mapsto v_n]$. For any context t compactable as $t = \sigma(X_i(u))$ with a grammar G, and any prefix $c \preceq t$ satisfying $\sigma(X_i) \not\preceq c$, c is compactable as $c = \sigma(d)$ with a grammar $G' \supseteq G$ satisfying*

$$\begin{aligned} &\mathit{depth}(d) \leq 3\,i + M \\ &|G'| \leq |G| + i^2 + 3\,i + 2\,i\,M \end{aligned}$$

where $M = \max\{\mathit{depth}(u), \mathit{depth}(v_1), \ldots, \mathit{depth}(v_n)\}$.

Proof: We proceed by induction on i.

In the base case $i = 1$ we have $c \preceq t = \sigma(X_1(u)) = v_1[\sigma(u)]$. The position of the hole of c must correspond to some position inside v_1 (otherwise $\sigma(X_1) = v_1 \prec c$, contrarily to the assumptions). Therefore, either c does not contain any part of $\sigma(u)$ or contains it completely. So, there exists a prefix d of $v_1[u]$ such that $c = \sigma(d)$. Now, by Lemmas 2 and 4 we can generate any prefix d of $v_1[u]$ with $\mathit{depth}(d) \leq \mathit{depth}(v_1[u]) = 1 + \max\{\mathit{depth}(v_1), \mathit{depth}(u)\}$ using a grammar G' with size $|G'| \leq |G| + 2 + \max\{\mathit{depth}(v_1), \mathit{depth}(u)\}$.

In the induction case $i > 1$ we have $c \preceq \sigma(X_i(u)) = \sigma(v_i[u])$. Let d_i be the largest prefix of $v_i[u]$ such that $\sigma(d_i) \preceq c$. This prefix is uniquely defined.

By Lemmas 2 and 4, since $d_i \preceq v_i[u]$, we can generate d_i with $\mathit{depth}(d_i) \leq 1 + \max\{\mathit{depth}(v_i), \mathit{depth}(u)\}$ using a grammar G'' of size $|G''| \leq |G| + 2 + \max\{\mathit{depth}(v_i), \mathit{depth}(u)\}$.

If $\sigma(d_i) = c$, taking $d = d_i$ and $G' = G''$ we fulfil the requirements of the lemma.

Otherwise $\sigma(d_i) \prec c$, and the hole of c fall inside the instance of some variable X_k occurring in v_i, with $k < i$ (remember that $\sigma(X_i) \not\preceq c$). This position may be in the main path of v_i or not. In the first case, we have $v_i[u] = d_i[X_k[v'_i[u]]]$, for some suffix Z-context v'_i of v_i, i.e. d_i does not contain any part of u. In the second case, we have $v_i[u] = d_i[X_k[v'_i]]$, for some subterm v'_i of v_i, i.e. d_i completely contains u.

In the first case, we can decompose $c = \sigma(d_i)[\hat{c}]$, for some Z-context $\hat{c}$ satisfying $\hat{c} \prec \sigma(X_k[v'_i[u]])$ and $\sigma(X_k) \not\preceq \hat{c}$. Using Lemmas 2 and 4, we see that there exists a grammar $\hat{G}$ deriving $X_k[v'_i[u]]$ with $\mathit{depth}(X_k[v'_i[u]]) \leq 2+\max\{\mathit{depth}(v_i), \mathit{depth}(u)\}$ and satisfying $|\hat{G}| \leq |G''|+2+\mathit{depth}(v_i)$. Moreover, $\sigma(X_k[v'_i[u]])$ is compactable with $\hat{G}$. Since $k < i$, by induction hypothesis, we can compact $\hat{c} = \sigma(\hat{d})$ with a grammar $\hat{G}'$ that generates $\hat{d}$ with

$$\begin{aligned}\mathit{depth}(\hat{d}) &\leq 3k + \max\{\mathit{depth}(X_k[v'_i[u]]), \mathit{depth}(v_1), \ldots, \mathit{depth}(v_n)\} \\ &\leq 3k + 2 + M\end{aligned}$$

and has size

$$\begin{aligned}|\hat{G}'| &\leq |\hat{G}| + k^2 + 5\,k + +2\,k\,\max\{\mathit{depth}(v_1), \ldots, \mathit{depth}(v_n), \mathit{depth}(v'_i[u])\} \\ &\leq |\hat{G}| + k^2 + 3\,k + 2\,k\,(1 + M)\end{aligned}$$

Now since $c = \sigma(d_i)[\hat{c}]$ and $\hat{c} = \sigma(\hat{d})$, we have $c = \sigma(d_i[\hat{d}])$. Therefore, by Lemmas 2 and 4, we can find a grammar G' with $|G'| \leq |\hat{G}'| + 1$ that generates $d = d_i[\hat{d}]$ with $\mathit{depth}(d) = 1 + \max\{\mathit{depth}(d_i), \mathit{depth}(\hat{d})\}$ and allow us to compact c.

In the second case we obtain lower bounds. Finally, all the inequalities allow us to conclude

$$\begin{aligned}\mathit{depth}(d) &= 1 + \max\{\mathit{depth}(d_i), \mathit{depth}(\hat{d}\} \\ &\leq 1 + \max\{\,1 + \max\{\mathit{depth}(v_i), \mathit{depth}(u)\}\ , \\ &\qquad\qquad 3k + \max\{\mathit{depth}(X_k[v'_i[u]]), \mathit{depth}(v_1), \ldots, \mathit{depth}(v_n)\}\} \\ &\leq 1 + \max\{1 + M,\ 3k + \max\{2 + M, M\}\} \\ &= 3(k+1) + M \leq 3i + M\end{aligned}$$

$$\begin{aligned}|G'| &\leq |\hat{G}'| + 1 \\ &\leq |\hat{G}| + k^2 + 3\,k + 2\,k\,(M+1) + 1 \\ &\leq |G''| + 2 + M + k^2 + 3\,k + 2\,k\,(M+1) + 1 \\ &\leq |G| + 2 + M + 2 + M + k^2 + 3\,k + 2\,k\,(M+1) + 1 \\ &= |G| + (k+1)^2 + 3\,(k+1) + 1 + 2\,(k+1)\,M \leq |G| + i^2 + 3\,i + 2\,i\,M\end{aligned}$$

■

Lemma 10. *For any Z-context t compactable as $t = [X_1 \mapsto v_1] \circ \cdots \circ [X_n \mapsto v_n]u$ with a grammar G, any prefix, subterm or subcontext t' of t, is also compactable as $t' = [X_1 \mapsto v_1] \circ \cdots \circ [X_n \mapsto v_n]u'$, for some Z-context u', with a grammar $G' \supseteq G$ satisfying*

$$\begin{aligned} &\mathit{depth}(u') \leq M + \mathcal{O}(n) \\ &|G'| \leq |G| + \mathcal{O}(n\,M) \end{aligned}$$

where $M = \max\{\mathit{depth}(u), \mathit{depth}(v_1), \ldots \mathit{depth}(v_n)\}$.

PROOF: We only show the proof when t' is a prefix of t, and t is a context. We can write $t = [X_1 \mapsto v_1] \circ \cdots \circ [X_n \mapsto v_n]u$ as $t = [X_1 \mapsto v_1] \circ \cdots \circ [X_n \mapsto v_n] \circ [X_{n+1} \mapsto u]X_{n+1}(\bullet)$ for any fresh variable X_{n+1}. Then we can apply Lemma 9.

For subterms we need a variant of Lemma 9, and for subcontexts the application of a subterm and then a prefix. For prefixes of terms we need a variant of Lemma 9 based on Lemma 5. These proofs exceeds the length of this paper. ■

Lemma 11. *For any equation E, and any substitution $\tau = [X \mapsto c[X'(\bullet)]]$, where c is a Z-context not containing X, and built up using $\mathcal{O}(|\,Var(E)|)$ subcontexts of E, and one exponentiation to e, if E is compactable as*

$$E = [X_1 \mapsto v_1] \circ \cdots \circ [X_n \mapsto v_n]E'$$

with a grammar G, then, for some Z-context d, some $m \in \{0, \ldots, n\}$, and some permutation π, $\tau(E)$ is also compactable as

$$\begin{aligned} \tau(E) = [X_{\pi(1)} \mapsto v_{\pi(1)}] \circ \cdots \circ [X_{\pi(m)} \mapsto v_{\pi(m)}] \circ [X \mapsto d] \\ \circ [X_{\pi(m+1)} \mapsto v_{\pi(m+1)}] \circ \cdots \circ [X_{\pi(n)} \mapsto v_{\pi(n)}]\,E' \end{aligned}$$

with a grammar $G' \supseteq G$ deriving d and satisfying

$$\begin{aligned} &\mathit{depth}(d) \leq M + \mathcal{O}(|\,Var(E)|\,n + \log e) \\ &|G'| \leq |G| + \mathcal{O}(|\,Var(E)|\,n\,M + \log e) \end{aligned}$$

where $M = \max\{\mathit{depth}(E), \mathit{depth}(v_1), \ldots, \mathit{depth}(v_n)\}$.

PROOF: By Lemma 10, we can compact each one of the $\mathcal{O}(Var(|E|)$ subcontexts c_i of E that compound c as $c_i = \sigma(d_i)$ with the same grammar G' increasing the size of G in $\mathcal{O}(Var\,|E|)\,\mathcal{O}(n\,M)$ and the depth of the symbols generating d_i being at most $M + \mathcal{O}(Var\,|E|)\,\mathcal{O}(n)$. Let d be constructed from the pieces d_i as c is constructed from the pieces c_i.

By Lemmas 3 and 2, applied as many-times as pieces we have to assemble, we can prove that there exists a grammar $G'' \supseteq G'$ that generates d with depth $M + \mathcal{O}(|\,Var(E)|\,n + \log e)$, and satisfying $|G''| = |G'| + \mathcal{O}(\log e)$. Using this grammar G'', we can compact $\tau(E)$ as

$$\begin{aligned} \tau(E) &= \big[X \mapsto [X_1 \mapsto v_1] \circ \cdots \circ [X_n \mapsto v_n]\,d\big]E \\ &= \big[X \mapsto [X_1 \mapsto v_1] \circ \cdots \circ [X_n \mapsto v_n]\,d\big] \circ [X_1 \mapsto v_1] \circ \cdots \circ [X_n \mapsto v_n]\,E' \\ &= [X_1 \mapsto v_1] \circ \cdots \circ [X_n \mapsto v_n] \circ [X \mapsto d] \circ [X_1 \mapsto v_1] \circ \cdots \circ [X_n \mapsto v_n]\,E' \end{aligned}$$

Let $\sqsubset$ be the transitive closure of the relation: if X_i occurs in v_j then $X_i \sqsubset X_j$. By definition of compaction this relation is irreflexive. Extend this relation considering $X \sqsubset X_i$ when X occurs in v_i and and $X_i \sqsubset X$ when X_i occurs in d. Then, for every $i = 1, \ldots, n$, either $X_i \not\sqsubset X$ or $X \not\sqsubset X_i$. (Otherwise we would get $X \sqsubset X$ and either c_1 or c_2 would contain X, contrarily to our assumption). Now,for every i, if $X_i \not\sqsubset X$ we can remove $[X_i \mapsto v_i]$ from the left of $[X \mapsto d]$, and if $X \not\sqsubset X_i$ we can remove $[X_i \mapsto v_i]$ from the right of $[X \mapsto d]$. In this way we obtain the desired compaction. Notice that we have to re-order the X_i according to the extension of $\sqsubset$, i.e. $X_{\pi(1)} < \cdots < X_{\pi(m)} < X < X_{\pi(m+1)} < \cdots < X_{\pi(n)}$ is a total ordering of the variables compatible with the partial ordering $\sqsubset$. ■

Theorem 2. *If σ is a size-minimal solution of $E = (t \stackrel{?}{=} u)$, then $\sigma(t)$ is compactable as $\sigma(t) = [X_1 \mapsto v_1] \circ \cdots \circ [X_m \mapsto v_m]\, t'$ with a grammar of depth $\mathcal{O}(|E|^9)$ and size $\mathcal{O}(|E|^{18})$, where $m = \mathcal{O}(|E|^4)$.*

Similarly for u.

Proof: Using Lemmas 7 and 8 inductively, we can get a decomposition $\sigma = \rho_n \circ \cdots \circ \rho_1$ such that $\mu(\rho_i \circ \cdots \circ \rho_1(E)) < \mu(\rho_{i-1} \circ \cdots \circ \rho_1(E))$. Therefore, by Lemma 6, we have $n = \mathcal{O}(|E|^3)$. Moreover, each one of the ρ_i's is the composition of at most $|\,Var(E)|$ many (partial) instantiations of just one variable. So, there are $m = \mathcal{O}(|E|^4)$ of these instantiations.

Each one of these partial instantiations τ_j fulfill the requirements of Lemma 11. So, using this Lemma 11 inductively, we can prove that $\tau_i \circ \cdots \circ \tau_1(E)$ is compactable with a grammar G_i such that the maximal depth d_i of a Z-context derived by G_i is $d_i \leq d_{i-1} + \mathcal{O}(|\,Var(E)|\, i + \log e)$, i.e. $d_i = \mathcal{O}(|\,Var(E)|\, i^2 + i \log e)$, and for the size $|G_i| \leq |G_{i-1}| + \mathcal{O}(|\,Var(E)|\, i\, d_i + \log e) = |G_{i-1}| + \mathcal{O}(|\,Var(E)|^2\, i^3 + |\,Var(E)|\, i^2 \log e)$, i.e. $|G_i| = \mathcal{O}(|\,Var(E)|^2\, i^4 + |\,Var(E)|\, i^3 \log e)$.

We have $i \leq \mathcal{O}(|E|^4)$. The exponent of periodicity lemma ensures that $\log e = \mathcal{O}(|E|^2)$. We have also $|\,Var(E)| = \mathcal{O}(|E|)$.

Finally, composing all the bounds we get the polynomial bounds stated in the Theorem. ■

Corollary 1. *Bounded Second-Order Unification is NP-complete.*

Proof: For any equation E, and any size-minimal solution σ, there exists a STG of polynomial size in $|E|$ that generates $\sigma(X)$, for every $X \in Var(E)$. Notice that we represent σ as a composition of substitutions, and the grammar can generate each one of the compositions, but replacing variables by non-terminal symbols of the grammar, we can (increasing the depth, but without increasing the size) generate σ. A small enlargement of the grammar allow us to generate $\sigma(t)$ and $\sigma(u)$.

Now, a nondeterministic algorithm, guessing a representation of the substitution σ not exceeding the polynomial bound, and using Theorem 1 to check if $\sigma(t) = \sigma(u)$ can decide if $t \stackrel{?}{=} u$ is solvable or not.

NP-hardness is proved in [17]. ■

References

1. G. Busatto, M. Lohrey, and S. Maneth. Efficient memory representation of XML documents. In *DBPL'05*, volume 3774 of *LNCS*, pages 199–216, 2005.
2. H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: `http://www.grappa.univ-lille3.fr/tata`, 1997. release 1.10.2002.
3. G. Dowek. Higher-order unification and matching. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 16, pages 1009–1062. Elsevier Science, 2001.
4. W. M. Farmer. Simple second-order languages for wich unification is undecidable. *Theoretical Computer Science*, 87:173–214, 1991.
5. W. D. Goldfarb. The undecidability of the second-order unification problem. *Theoretical Computer Science*, 13:225–230, 1981.
6. G. Huet. A unification algorithm for typed λ-calculus. *Theoretical Computer Science*, 1:27–57, 1975.
7. A. Kościelski and L. Pacholski. Complexity of Makanin's algorithm. *J. ACM*, 43(4):670–684, 1996.
8. J. Levy, J. Niehren, and M. Villaret. Well-nested context unification. In *CADE'05*, volume 3632 of *LNCS*, pages 149–163, 2005.
9. J. Levy, M. Schmidt-Schauß, and M. Villaret. Monadic second-order unification is NP-complete. In *RTA'04*, volume 3091 of *LNCS*, pages 55–69, 2004.
10. J. Levy and M. Veanes. On the undecidability of second-order unification. *Information and Computation*, 159:125–150, 2000.
11. J. Levy and M. Villaret. Currying second-order unification problems. In *RTA'02*, volume 2378 of *LNCS*, pages 326–339, 2002.
12. G. S. Makanin. The problem of solvability of equations in a free semigroup. *Math. USSR Sbornik*, 32(2):129–198, 1977.
13. A. Martelli and U. Montanari. An efficient unification algorithm. *ACM TOPLAS*, 4(2):258–282, 1982.
14. W. Plandowski. Testing equivalence of morphisms in context-free languages. In *ESA'94*, volume 855 of *LNCS*, pages 460–470, 1994.
15. W. Plandowski. *The Complexity of the Morphism Equivalence Problem for Context-Free Languages*. PhD thesis, Department of Mathematics, Informatics and Mechanics, Warsaw University, 1995.
16. M. Schmidt-Schauß. Stratified context unification is in PSPACE. In *CSL'01*, volume 2142 of *LNCS*, pages 498–512, 2001.
17. M. Schmidt-Schauß. Decidability of bounded second order unification. *Information and Computation*, 188(2):143–178, 2004.
18. M. Schmidt-Schauß. Polynomial equality testing for terms with shared substructures. Frank report 21, Institut für Informatik. FB Informatik und Mathematik. J. W. Goethe-Universität Frankfurt am Main, November 2005.
19. M. Schmidt-Schauß and K. U. Schulz. On the exponent of periodicity of minimal solutions of context equations. In *RTA'98*, volume 1379 of *LNCS*, pages 61–75, 1998.
20. A. Schubert. Second-order unification and type inference for church-style polymorphism. In *POPL'98*, pages 279–288, 1998.