Well-nested Context Unification

Jordi Levy, Joachim Niehren, Mateu Villaret

Well-Nested Context Unification.
Proc. of the 20th Int. Conf. on Automated Deduction (CADE-20).

Well-Nested Context Unification*

Jordi Levy[1], Joachim Niehren[2], and Mateu Villaret[3]

[1] IIIA, CSIC, Barcelona, Spain.
[2] INRIA Futurs, LIFL, Lille, France.
[3] IMA, Universitat de Girona, Spain.

Abstract. Context unification (CU) is the open problem of solving context equations for trees. We distinguish a new decidable variant of CU – *well-nested CU* – and present a new unification algorithm that solves well-nested context equations in non-deterministic polynomial time. We show that minimal well-nested solutions of context equations can be composed from the material present in the equation (see Theorem 1). This property is wishful when modeling natural language ellipsis in CU.

1 Introduction

Context unification (CU) is the problem of solving equations with context variables over the structure of finite trees [Com92,Lev96,NPR97,SSS99,SS02]. CU is the natural generalisation of *word unification* (WU) [Mak77]. It can equally be understood as a variant of *linear second-order unification* [Lev96,LV00].

Whether CU is decidable is a long-standing open problem. So far, only a number of fragments of CU could be shown decidable. The most prominent fragment is WU proved decidable in [Mak77] and PSPACE in [Pla99]. Two further decidability results were shown for stratified CU [SS02] and for the 2 variable fragment [SSS99]. All these decidable fragments are defined over syntactic properties of the equations considered. In contrast to these results, the present paper introduces a new decidable variant, *well-nested context unification*, whose definition relies on semantic properties of solutions.

The well-nestedness restriction is motivated by an applications of CU [NPR97] in the field of compositional semantics of natural language, whose original goal was to improve on previous approaches to ellipsis resolution based on higher-order unification [DSP91,GK96]. The equational language of CU here serves as a uniform modelling language for ellipsis and scope underspecification phenomena. This approach then led to the development of the *constraint language for lambda structures* (CLLS) [EKN01]. The *parallelism constraints* feature of the CLLS, has been shown equally expressive than CU [NK01], hence its decidability is still unknown.

* This research has been partially supported by the CYCIT projects iDEAS (TIN2004-04343), Mulog (TIN2004-07933-C03-01) and Asilan (TIN2004-07672-C03-01), the Mostrare project of INRIA Futurs, the Universities of Lille 1 and 3, and the projects Tralala and ACIMDD of the ACI masses de données.

The first decidable fragment of CLLS that is sufficiently expressive for the envisaged application to computational semantics (even though not for all aspects of ellipsis modelling) is the language of *well-nested parallelism constraints* [EN03], whose satisfiability problem is NP-complete. Modelling languages with lower algorithmic complexity are not known for these applications.

Unfortunately, however, the NP-algorithm for well-nested parallelism constraints remains questionable in practice. It relies on repeatedly solving *full dominance constraints*, which is exponentially less efficient in theory and practice than solving *normal dominance constraints* [BDMN04]. Normal dominance constraints in turn are sufficient for modelling pure scope underspecification.

Well-nested CU is properly less expressive than well-nested parallelism constraints because the later subsume (full) dominance constraints, whereas well-nested CU does not. We show that minimal well-nested solutions of CU equations can always be composed from the material present in the equations (see Theorem 1). This surprising property is wanted for ellipsis modelling, where it means that elided parts of ellipsis can always be reconstructed from what was uttered elsewhere. CLLS does not satisfy this condition. In particular, some well-nested parallelism constraints that we can not express in well-nested CU fail to have this property.

We contribute a new unification algorithm that solves well-nested CU equations in non-deterministic polynomial time. Our algorithm guesses how to construct minimal well-nested solutions from the given equation. We show NP-hardness of well-nested CU satisfiability by encoding string matching [Ang80].

The paper proceeds as follows, in Section 2 we illustrate how well-nested CU can solve ellipses, in Sections 3 and 4 we provide basic definitions to introduce in Section 5 well-nested CU and prove that size-minimal solutions have a polynomial representation. In Section 6 we show its NP-completeness proving that guessed solutions can be checked in polynomial time.

2 Ellipsis Resolution

To illustrate the usage of CU in ellipsis resolution, we consider an example from [Sag76] which contains two VP-ellipsis with a nice nesting structure, but without scope underspecification:

(1) Mary can't go to Princeton in the fall, (2) but she can in the spring, (3) although if she does then ...

Sentence *(2)* means that Mary can go to Princeton in the spring, so the phrase *go to Princeton* is elided. Sentence *(3)* states that something will happen if Mary goes to Princeton in the spring; *go to Princeton in the spring* is elided here.

Following [Mon73], we can represent the semantics of sentences by lambda terms, i.e., in higher-order logics:

$$x_1 = mary@\lambda m(in\_the\_fall@(can't@m@go\_to\_princeton))$$
$$x_2 = mary@\lambda m(in\_the\_spring@(can@m@go\_to\_princeton))$$
$$x_3 = (mary@\lambda m(in\_the\_spring@(do@m@go\_to\_princeton))) \rightarrow x_4$$

The variables x_1, x_2, x_3 denote the respective meanings of the three sentences, while x_4 stands for the meaning of the consequence that Mary fears.

A non-trivial question is how to resolve the nested ellipses automatically. This problem is typically split into two part [DSP91,GK96,NPR97,EKN01,EN03]. First one has to infer the nesting structure of the ellipsis, second, one has to reconstruct the elided parts from the nesting structure. In this paper, we only treat the second problem, resolving ellipsis given their nesting structure. The nesting structure of the example is indicated in Fig. 1.

The dashed box represents the first elided part, the larger dotted box the second one. Note that one occurrence of the dashed box is nested inside of the dotted box, while the second one is completely outside of it. These boxes are well-nested because they do not properly overlap.

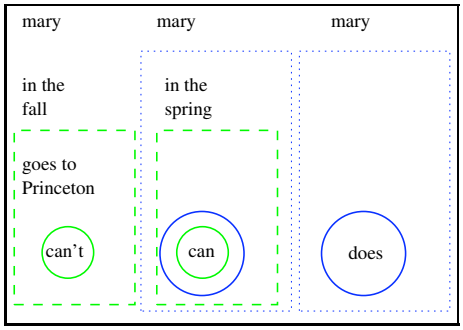

Fig. 1. Nesting sketches for 2

When looking at the trees formed by the abstract syntax of the lambda-terms, the boxes become contexts, i.e., trees where a subtree has been substituted by a hole marker $\bullet$. The lambda terms can then be described by context equations of CU. Let X be a context variable for the dashed box and Y for the dotted box. We can then describe the meaning of the ellipsis by the following equations:

$$x_1 \stackrel{?}{=} mary@\lambda m(in\_the\_fall@X(can't)) \qquad X \stackrel{?}{=} (\bullet@m)@go\_to\_princeton$$
$$x_2 \stackrel{?}{=} mary@\lambda m(Y(can)) \qquad Y \stackrel{?}{=} in\_the\_spring@X$$
$$x_3 \stackrel{?}{=} (mary@\lambda m(Y(do))) \to x_4$$

Similar semantical descriptions can be inferred compositionally from the syntax of the sentence [EKN01,NPR97]. Resolving the ellipsis amounts to find minimal size unifiers of the equations. Such solutions are well-nested, i.e., correspond to instantiating context variables with contexts that do not properly overlap, moreover, they are made of the material already present in the equations.

3 Well-Nested Segments

Given a signature Γ of symbols, we write $\mathcal{T}_\Gamma$ for the set of terms over Γ. The *size* $|t|$ of a term t is the number of its symbols in Γ. We identify positions in terms by their relative address from the root, using the Dewey's decimal notation. We denote the set of positions of a term t by $\mathsf{pos}(t)$. The word's prefix ordering $p \preceq q$ is also known as the *dominance ordering* on positions, which holds if p is an ancestor or equal to q. For all terms t and positions $p \in \mathsf{pos}(t)$, we let $\mathsf{lab}_t(p)$ denote the symbol of Γ at position p of t.

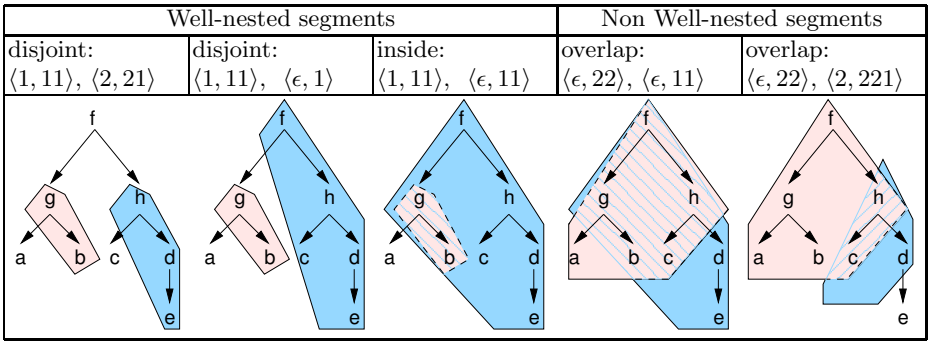

Fig. 2. Relationships between segments

A *segment* $\langle p, q\rangle$ is a pair of positions such that $p \preceq q$. A segment $\langle p, q\rangle$ belongs to a term t if $p, q \in \mathsf{pos}(t)$. Every segment of a term t distinguishes a set of positions of t: $\mathsf{pos}_t(\langle p, q\rangle) = \{r \in \mathsf{pos}(t) \mid p \preceq r, q \not\preceq r\}$

Definition 1. *Let S_1 and S_2 be two segments of t. We say that S_1 is* inside *of S_2 in t, if $\mathsf{pos}_t(S_1) \subseteq \mathsf{pos}_t(S_2)$. The segments S_1 and S_2 are* disjoint *in t, if $\mathsf{pos}_t(S_1) \cap \mathsf{pos}_t(S_2) = \emptyset$. The segments S_1 and S_2 are* well-nested*, if one of them lies inside the other in t, or if they are disjoint.*

Examples for the possible segment relationship are given in Fig. 2. A non-empty segment $\langle p, q\rangle$ of t is inside of a segment $\langle p', q'\rangle$ of t if, and only if, $p' \preceq p$ and $q \preceq q'$. Notice that $\mathsf{pos}_t(\langle p, p\rangle) = \emptyset$ for any position $p \in \mathsf{pos}(t)$.

4 Context Unification

We introduce the variant of context unification (CU) that we will use. Let Σ be a signature of constants: $f, g, \ldots$ with non-zero arity, and $a, b, \ldots$ with arity zero. Let Vars be a countable infinite set of *context variables* $X, Y, Z, \ldots$ with arity one[4], and let $\bullet$ be a symbol of arity 0 that we call the *hole marker*. The signature of CU-terms $s, t, \ldots$ over Σ is $\Gamma(\Sigma) = \Sigma \uplus \mathsf{Vars} \uplus \{\bullet\}$, i.e.:

$$s, t ::= a \mid \bullet \mid X(t) \mid f(t_1, \ldots, t_n)$$

The set of variables occurring in a term t is $\mathsf{Var}(t)$.

A *tree* over Σ is a term over $\Gamma(\Sigma)$ that does not contain the hole marker (but possibly variables). A *context* over Σ is a term over $\Gamma(\Sigma)$ that contains a *unique* occurrence of the hole marker, at a position denoted by $\mathsf{hole}(t)$. We write Tree_Σ for the set of trees over Σ and Cont_Σ for the set of contexts over Σ.

Given a tree t and a segment $\langle p, q\rangle$ of t, we write $t\langle p, q\rangle$ for the context in t that starts at p and has its hole at q. For instance $g(f(a, b), c))\langle 1, 1\cdot 2\rangle = f(a, \bullet)$. Note that $\mathsf{pos}_t(\langle p, q\rangle) = \{p \cdot r \mid r \in \mathsf{pos}(t\langle p, q\rangle)\} \setminus \{q\}$.

[4] Note that we could add variables of arbitrary arities, but, although the adaptation of this results to this more general framework is not straight forward, we do not do so for sake of simplicity.

The hole marker is like a λ-bound variable, for instance the context $f(g(a,\bullet))$ corresponds to the linear higher-order term $\lambda x.f(f(a,x))$. This view is formally adapted in linear second-order unification [Lev96,LV00].

While avoiding more general higher-order syntax here, we will nevertheless use contexts $t \in \mathsf{Cont}_\Sigma$ as functions $t : \mathcal{T}_{\Gamma(\Sigma)} \to \mathcal{T}_{\Gamma(\Sigma)}$ which map terms s to terms $t(s)$ by substituting the hole marker in t by s. For instance,

$$f(g(a,\bullet))(b) = f(g(a,b)) \qquad \text{or} \qquad f(g(a,\bullet))(f(\bullet)) = f(g(a,f(\bullet)))$$

Note that $t(\mathsf{Tree}_\Sigma) \subseteq \mathsf{Tree}_\Sigma$ and $t(\mathsf{Cont}_\Sigma) \subseteq \mathsf{Cont}_\Sigma$ since the unique hole of t will be filled by either a tree or a context.

A *context substitution* is a function $\sigma : \mathsf{Vars} \to \mathsf{Cont}_\Sigma$ such that $\sigma(X) \neq X$ only for a finite set of variables $X \in \mathsf{Vars}$. We lift substitutions from variables to functions on terms $\sigma : \mathcal{T}_{\Gamma(\Sigma)} \to \mathcal{T}_{\Gamma(\Sigma)}$ while using contexts as functions:

$$\begin{array}{ll} \sigma(X(s)) = \sigma(X)(\sigma(s)), & \sigma(a) = a, \\ \sigma(f(t_1,\ldots,t_n)) = f(\sigma(t_1),\ldots,\sigma(t_n)), & \sigma(\bullet) = \bullet. \end{array}$$

Contexts $\sigma(X)$, substituted for variables X, are immediately applied so that their holes are filled. Thus, $\sigma(\mathsf{Tree}_\Sigma) \subseteq \mathsf{Tree}_\Sigma$ and $\sigma(\mathsf{Cont}_\Sigma) \subseteq \mathsf{Cont}_\Sigma$. The composition of two substitutions σ_1 and σ_2, written as $\sigma_1 \circ \sigma_2$, is defined by $(\sigma_1 \circ \sigma_2)(t) = \sigma_1(\sigma_2(t))$. Substitutions are usually represented as $[X_1 \mapsto \sigma(X_1), \ldots, X_n \mapsto \sigma(X_n)]$, where the X_i's are the variables for which $\sigma(X_i) \neq X_i$. As we will see in the next section, we can get more compact representations of substitutions by means of composition.

A *context equation* e is a term $s \stackrel{?}{=} t$ with $s,t \in \mathsf{Tree}_\Sigma$. Equations are terms over the signature $\Sigma \cup \mathsf{Var} \cup \{\stackrel{?}{=}\}$, therefore the notions of positions and segments apply to equation, as to all other types of terms. We apply substitutions to equations such that the result of $\sigma(s \stackrel{?}{=} t)$ is the equation $\sigma(s) \stackrel{?}{=} \sigma(t)$.

A *unifier* of an equation $s \stackrel{?}{=} t$ is a substitution σ satisfying $\sigma(s) = \sigma(t)$. A unifier σ of e is said to be *ground* if $\sigma(e)$ does not contain any variable, and *size-minimal* if it minimizes $\sum_{X \in \mathsf{Var}(e)} |\sigma(X)|$ while satisfying $\sigma(X) = X$ for all $X \notin \mathsf{Var}(e)$.

CU is often seen as the satisfiability problem of context equations, i.e.: given a fixed signature Σ, the problem of deciding whether a given system of context equations over Σ has a unifier. For the sake of simplicity, we restrict ourselves to a single context equation as input. This does not affect expressiveness.

CU can alternatively be considered as a constraint solving problem, i.e., the problem to enumerate all unifiers of a given equation. Rather than enumerating all unifiers, one usually prefers to enumerate only the most general unifiers (from which all others can be obtained by instantiation). In this paper, we will consider a similar variant. We will be able to enumerate all size-minimal unifiers in a compact representation by using compositions of substitutions.

Definition 2 (Correspondence function). *Let e be an equation with unifier σ. Positions in e and $\sigma(e)$ correspond through the function $\mathsf{c}^e_\sigma : \mathsf{pos}(e) \to$*

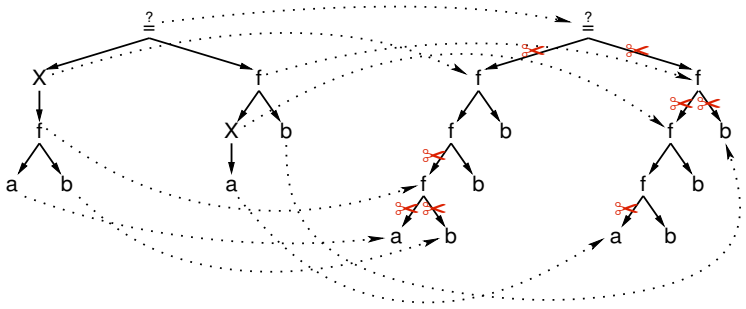

Fig. 3. The equation $e = X(f(a,b)) \stackrel{?}{=} f(X(a),b)$ and $\sigma(e)$, where σ is the non-well nested unifier $[X \mapsto f(f(\bullet,b),b)]$, with the correspondence function c^e_σ and the "cuts".

$\mathsf{pos}(\sigma(e))$ *that satisfies for all positions* $p \cdot i \in \mathsf{pos}(e)$ *and variables* $X \in \mathsf{Var}(e)$*:*

$$\mathsf{c}^e_\sigma(\epsilon) = \epsilon \qquad \textit{and} \qquad \mathsf{c}^e_\sigma(p \cdot i) = \begin{cases} \mathsf{c}^e_\sigma(p) \cdot i & \textit{if } \mathsf{lab}_e(p) \in \Sigma \\ \mathsf{c}^e_\sigma(p) \cdot \mathsf{hole}(\sigma(X)) & \textit{if } \mathsf{lab}_e(p) \in \mathsf{Var} \end{cases}$$

Figure 3 represents the correspondence function of a given equation and unifier. Notice that if $\langle p, q\rangle$ is a segment of e then $\langle \mathsf{c}^e_\sigma(p), \mathsf{c}^e_\sigma(q)\rangle$ is a segment of the equation $\sigma(e)$. Furthermore, if $\mathsf{lab}_e(p) = X$ then $\sigma(X) = \sigma(e)\langle \mathsf{c}^e_\sigma(p), \mathsf{c}^e_\sigma(p{\cdot}1)\rangle$.

We next define an equivalence relation on positions of $\sigma(e)$ that must carry the same labels. An occurrence of variable X in an equation e is a position p that satisfies $\mathsf{lab}_e(p) = X$.

Definition 3 (Equivalent positions). *Let* e *be an equation with unifier* σ*. Let* $\approx^e_\sigma \subseteq \mathsf{pos}(\sigma(e)) \times \mathsf{pos}(\sigma(e))$ *be the least equivalence relation satisfying:*

1. *corresponding positions in both sides of the instantiated equation are equivalent: all position* p *such that* $1 \cdot p, 2 \cdot p \in \mathsf{pos}(\sigma(e))$ *satisfy* $\quad 1 \cdot p \approx^e_\sigma 2 \cdot p$
2. *corresponding positions in instances of different occurrences of the same variable are equivalent: all* $X \in \mathsf{Var}(e)$*,* $q \in \mathsf{pos}(\sigma(X)) \setminus \mathsf{hole}(\sigma(X))$*, and occurrences* p_1 *and* p_2 *of* X *in* e *satisfy* $\quad \mathsf{c}^e_\sigma(p_1) \cdot q \approx^e_\sigma \mathsf{c}^e_\sigma(p_2) \cdot q$

Lemma 1. *If* σ *is a solution of equation* e *then every pair of equivalent positions* $p_1 \approx^e_\sigma p_2$ *has the same label, i.e.* $\mathsf{lab}_{\sigma(e)}(p_1) = \mathsf{lab}_{\sigma(e)}(p_2)$*.*

5 Well-Nested Unifiers

In this section we define well-nested unifiers, and show how to represent well-nested unifiers of equations in polynomial space depending on the size of the equation.

It is well known that most general first-order unifiers can be exponentially sized on the size of the problem. Take as example the problem $g(X_1, \ldots, X_{n-1}) \stackrel{?}{=} g(f(X_2, X_2), \ldots, f(X_n, X_n))$. Fortunately, these unifiers can be represented in polynomial space as a composition of substitutions. In our example: $[X_{n-1} \mapsto f(X_n, X_n)] \circ \cdots \circ [X_1 \mapsto f(X_2, X_2)]$. Moreover, the terms on

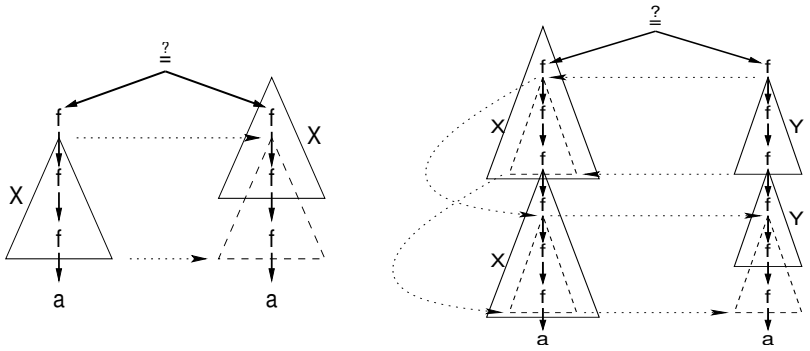

Fig. 4. Forbidden overlaps

the right of the arrows, $f(X_i, X_i)$ in our example, are subterms of the original unification problem. These two properties are used by practical implementations of first-order unification. Theorem 1 states a similar property for well-nested CU, that does not hold neither for general CU nor for word unification.

Size-minimal and most general well-nested unifiers do not introduce new constants not occurring in the original equation. In the application to ellipsis resolution, this is expected to hold for all size-minimal unifiers. In general CU, however, this property does not hold. For instance, the equation $X(a) \stackrel{?}{=} Y(b)$ has as many size-minimal unifiers of the form $[X \mapsto f(\bullet, b), Y \mapsto f(a, \bullet)]$ as binary symbols $f \in \Sigma$. This makes CU dependent of the signature.

If we allow the use of n-ary context variables, like in linear second-order unification, then we get this property back, but only for most general unifiers (not necessarily size-minimal ones). For instance, we get a unique most general unifier $[X \mapsto Z(\bullet, b), Y \mapsto Z(a, \bullet)]$ for our example, that uses a fresh binary context variable Z, but that does not introduce new constants.

Roughly speaking, a unifier σ of an equation e is well-nested if all segments of $\sigma(e)$ that encompass variable occurrences in e are pairwise well-nested. Consider for instance the equation $f(X(a)) \stackrel{?}{=} X(f(a))$. Well-nestedness forbids unifiers like $[X \mapsto f(f(\bullet))]$ since the segments encompassing the two occurrences of X overlap when overlaying the two sides of the instantiated equation. The situation is illustrated in Fig. 4 (left)

More indirect overlaps are raised by "reflection" in the instances of other variable occurrences. Consider for instance the equation $X(X(a)) \stackrel{?}{=} f(Y(Y(f(a))))$ and the unifier $[X \mapsto f(f(f(\bullet))), Y \mapsto f(f(\bullet))]$ in Fig. 4 (right). Here the instances of the occurrences of Y are nested into those of X. However, if we overlay the both instances of X, then we see that the two instances of Y overlap. Thus, the unifier is not well-nested.

In order to formally define well-nested unifiers we need to extend the equivalence relation $\approx^e_\sigma$ on positions in $\sigma(e)$ to an equivalence relation $\equiv^e_\sigma$ on segments of $\sigma(e)$.

Definition 4 (Equivalent segments). *Let $\equiv^e_\sigma$ be the least equivalence relation on segments of $\sigma(e)$ that satisfies:*

1. corresponding segments on both sides of the instantiated equation are equivalent: *for all segments* $\langle 1 \cdot p, 1 \cdot q\rangle$ *of* $\sigma(e)$, $\quad \langle 1 \cdot p, 1 \cdot q\rangle \equiv^e_\sigma \langle 2 \cdot p, 2 \cdot q\rangle$
2. segments encompassing different occurrences of the same variable are equivalent: *all occurrence* p_1, p_2 *of the same variables* $X \in \mathsf{Var}(e)$ *in* e *satisfy* $\langle \mathsf{c}^e_\sigma(p_1), \mathsf{c}^e_\sigma(p_1 \cdot 1)\rangle \equiv^e_\sigma \langle \mathsf{c}^e_\sigma(p_2), \mathsf{c}^e_\sigma(p_2 \cdot 1)\rangle$
3. corresponding subsegments in equivalent segments are equivalent: *for all equivalent segments* $\langle p_1, q_1\rangle \equiv^e_\sigma \langle p_2, q_2\rangle$ *of* $\sigma(e)$*, and all segments* $\langle p, q\rangle$ *satisfying that* $\langle p_1 \cdot p, p_1 \cdot q\rangle$ *is a segment of* $\sigma(e)$ *inside* $\langle p_1, q_1\rangle$, $\quad \langle p_1 \cdot p, p_1 \cdot q\rangle \equiv^e_\sigma \langle p_2 \cdot p, p_2 \cdot q\rangle$

If $\langle p, q\rangle$ is a segment of $\sigma(e)$ just containing one position (this must be p), then $\langle p, q\rangle \equiv^e_\sigma \langle p', q'\rangle$ if, and only if, $p \approx^e_\sigma p'$. This shows that the equivalence on segments indeed extends on the equivalence on positions.

Definition 5 (Well-nested CU). *Let* σ *be a unifier of an equation* e*. The* $\langle \sigma, e\rangle$*-images of variables in* e *are the segments* $\langle q, r\rangle$ *of* $\sigma(e)$ *such that there exists a variable-labeled position* p *in the equation* e *whose corresponding segment in* $\sigma(e)$ *is equivalent:*

$$\langle q, r\rangle \ \equiv^e_\sigma \ \langle \mathsf{c}^e_\sigma(p), \mathsf{c}^e_\sigma(p \cdot 1)\rangle$$

We call a unifier σ *of* e well-nested *if the set of all* $\langle \sigma, e\rangle$*-images of variables in* e *are pairwise well-nested.* Well-nested CU *is the problem of deciding whether a given equation has a well-nested unifier or not.*

In what follows, we will show how to solve this problem by enumerating all well-nested size-minimal ground unifier of a given input equation.

Definition 6 (Normal unifer). *We say that a well-nested unifier is* normal *if it is size-minimal* among *all well-nested unifiers.*

It is important that normal unifiers are required to be size-minimal *among* well-nested unifiers, since size-minimal ground unifiers may not always need to be well-nested. As a consequence of the fact that non-ground well-nested unifiers can be made smaller by instantiating variables to the empty context, we have:

Lemma 2. *Normal unifiers are ground.*

We next recall a nice property of WU, that we will extend to CU, in order to prove the existence of compact representations of normal CU unifiers. We do not actually know who was the first in stating this property for WU, but to our knowledge, it was first proved in [PR98]. The set of positions in the equation e define "cuts" in $\sigma(e)$ throughout the function c^e_σ: formally, a position $p \in \sigma(e)$ is a *cut* if it is in the range of c^e_σ. These "cuts" limit the possible subwords that a size-minimal unifier may contain:

Proposition 1 (Lemma 6 of [PR98]). *If* σ *is a minimal unifier of a word equation* $t \stackrel{?}{=} u$*, then each subword of* $\sigma(t)$ *has an occurrence "over a cut".*

In [PR98] cuts are located *between* two consecutive positions, *cutting* the word into two pieces (see the scissors in Fig. 3). We *locate* the cuts in the positions on the range of c^e_σ (in Fig. 3, just bellow the scissors, where the dotted arrows representing c^e_σ points). In WU, "having an occurrence over a cut" means that there is an(other) occurrence of *the same* subword containing a cut in an inner point or in one of their extremes.

The analogous result fails for general CU but carries over in the well-nested case (see Lemma 5). Moreover, we generalise it by restricting the relation "has an(other) occurrence" to "there is an(other) $\equiv^e_\sigma$ equivalent segment", and, if the segment contains two or more positions, by restricting cuts to properly cut the context into two *non-empty* terms, i.e., discarding cuts on the extremes. To prove this result, we need two previous lemmas. Lemma 4 plays the same role as the Lemma 4 of [PR98].

Lemma 3. *Given a term t, and a marking function $m : \mathsf{pos}(t) \to \{\mathsf{rem}, \mathsf{pres}\}$, let $\mathsf{pres}(t, m)$ be a term for which there exists an embedding from the set of preserved nodes of t to $\mathsf{pos}(\mathsf{pres}(t, m))$, i.e. a bijective morphism*

$$f_{t,m} : \{p \in \mathsf{pos}(t) \mid m(p) = \mathsf{pres}\} \to \mathsf{pos}(\mathsf{pres}(t, m))$$

preserving the tree structure (ancestor, brother and label relations). Then, if such term $\mathsf{pres}(t, m)$ exists, then it is unique.
Given an equation e and a well-nested unifier σ, let $m : \mathsf{pos}(\sigma(e)) \to \{\mathsf{rem}, \mathsf{pres}\}$ be a marking function satisfying:

1. *For any pair of positions $p, q \in \mathsf{pos}(\sigma(e))$, if $p \approx^e_\sigma q$ then $m(p) = m(q)$, i.e., equivalent positions are both removed or both preserved.*
2. *For any $p \in \mathsf{pos}(e)$, we have $m(\mathsf{c}^e_\sigma(p)) = \mathsf{pres}$, i.e. cuts are preserved.*
3. *The term $\mathsf{pres}(\sigma(e), m)$ exists, i.e. after the removing process you get a term.*

Then, there exists a well-nested unifier σ' of e such that $\sigma'(e) = \mathsf{pres}(\sigma(e), m)$. Moreover, if there exists some removed node, then σ' is size-smaller than σ.

Proof. First, notice that if $\mathsf{pres}(t, m)$ exists there must be a unique outermost preserved position in t, and the function $f^{-1}_{t,m}$ has to map ϵ to it. Moreover, for any $p \in \mathsf{pos}(\mathsf{pres}(t, m))$ there must be $\mathrm{arity}(\mathsf{lab}_t(f^{-1}_{t,m}(p)))$ many preserved outermost bellow $f^{-1}_{t,m}(p)$ positions in t, and $f^{-1}_{t,m}$ has to map $p \cdot i$ to the i-th of them. Therefore, $f^{-1}_{t,m}$ is unique, so its inverse, and $\mathsf{pres}(t, m)$.

Now, notice that a unifier σ of e can be characterised by e, the term $\sigma(e)$ and the correspondence function c^e_σ. In our case, we characterise σ' by the correspondence function $\mathsf{c}^e_{\sigma'} = f_{\sigma(e),m} \circ \mathsf{c}^e_\sigma$.

Since cuts are not removed, this function maps any position of e to a position in $\mathsf{pres}(\sigma(e), m)$, which on the other side, is a well-formed term.

For any $p, q \in \mathsf{pos}(e)$ with $\mathsf{lab}_e(p) = \mathsf{lab}_e(q) \in \mathsf{Var}(e)$ we have $\sigma(e)\langle \mathsf{c}^e_\sigma(p), \mathsf{c}^e_\sigma(p \cdot 1)\rangle = \sigma(e)\langle \mathsf{c}^e_\sigma(q), \mathsf{c}^e_\sigma(q \cdot 1)\rangle$. Now, since the positions of these two contexts are $\approx^e_\sigma$-equivalent one to one, and equivalent positions are both removed or preserved, we have $\mathsf{pres}(\sigma(e), m)\langle \mathsf{c}^e_{\sigma'}(p), \mathsf{c}^e_{\sigma'}(p \cdot 1)\rangle = \mathsf{pres}(\sigma(e), m)\langle \mathsf{c}^e_{\sigma'}(q), \mathsf{c}^e_{\sigma'}(q \cdot 1)\rangle$. Therefore,

we can define a substitution σ' mapping X to $\mathsf{pres}(\sigma(e), m)\langle \mathsf{c}^e_{\sigma'}(p), \mathsf{c}^e_{\sigma'}(p \cdot 1)\rangle$, for any of the occurrences p of X.

It can also be proved that $p \approx^e_{\sigma'} q \Leftrightarrow f^{-1}_{\sigma(e),m}(p) \approx^e_{\sigma} f^{-1}_{\sigma(e),m}(q)$. This makes $\langle \sigma, e\rangle$-images to correspond to $\langle \sigma', e\rangle$-images throughout $f_{\sigma(e),m}$. Therefore, if σ is well-nested, then σ' is well-nested, too. □

Lemma 4. *If σ is a normal unifier of an equation e, then every equivalence class C of $\approx^e_\sigma$ is cut by some constant in e, i.e., there is a position $p \in \mathsf{pos}(e)$ such that $\mathsf{c}^e_\sigma(p) \in C$ and $\mathsf{lab}_e(p) \in \Sigma$.*

Proof. All positions of C are labeled alike in $\sigma(e)$ according to Lemma 1. Let $f = \mathsf{lab}_{\sigma(e)}(p)$ for all positions $p \in C$. Now, assume that the statement does not hold, so all positions in C belong to segments that are denotations of some variables. There are three cases:

If f is unary, the proof is quite similar to the proof given in [PR98]: if C does not contain cuts of a unary constant of e, then we can remove these occurrences of the constant f in C, which would result into a smaller unifier. As it is stated in Lemma 3, this replacement only involves changes in the instantiations of the variables, by deleting the nodes of the equivalence class. Moreover, this removing process preserves the well-nestedness property, because the relative position between segments do not change with the process.

If arity$(f) \geq 2$, the proof is more subtle, and requires the unifier to be well-nested. For every $p \in C$, consider the longest path q_p in e such that $\mathsf{c}^e_\sigma(q_p) \preceq p$. First, notice that, since we have assumed that C does not contain cuts corresponding to constants, all q_p correspond to variables $X_p = \mathsf{lab}_e(q_p)$ and, second, notice that p is inside the segment $\langle \mathsf{c}^e_\sigma(q_p), \mathsf{c}^e_\sigma(q_p \cdot 1)\rangle$.

Now, we can prove the following property: for any $p, p' \in C$, and integer numbers i, i', if $p \cdot i \preceq \mathsf{c}^e_\sigma(q_p \cdot 1)$ and $p' \cdot i' \preceq \mathsf{c}^e_\sigma(q_{p'} \cdot 1)$, then $i = i'$. To prove it, assume $i \neq i'$. Since p and p' are related by the $\approx^e_\sigma$ equivalence relation, and this is the transitive closure of a more restrictive relation, we can find two positions $p'', p''' \in C$, related by this more restrictive relation, and such that $p'' \cdot i \preceq \mathsf{c}^e_\sigma(q_{p''} \cdot 1)$ and $p''' \cdot i' \preceq \mathsf{c}^e_\sigma(q_{p'''} \cdot 1)$. According to the $\approx^e_\sigma$ definition, for these new positions, either $X_{p''} = X_{p'''}$, or $p'' = 1 \cdot r$ and $p''' = 2 \cdot r$, for some r. In the first case, we have $i = i'$. In the second case, if $i \neq i'$, we would have two bad-nested variables $X_{p''}$ and $X_{p'''}$. Notice that, in this point of the proof, we make use of the well-nestedness property.

We can conclude that there exists a number i such that, for every $p \in C$, every integer $j \neq i$, and every sequence r, there are not cuts of the form $p \cdot j \cdot r$. (Notice that this does not imply that there exist cuts of the form $p \cdot i \cdot r$). Now, using again Lemma 3, we can replace all the contexts $\sigma(e)\langle p, p \cdot i\rangle$ by the empty context, because they do not contain cuts. Again this removing process preserves the well-nestedness property, and results in a smaller well-nested unifier, which contradicts the assumption.

If arity$(f) = 0$ we can not apply the previous reasoning because $\sigma(e)\langle p, p \cdot i\rangle$ is not a context. However, if C does not contains cuts corresponding to constants of e, then C does not contain cuts at all (it could only contain cuts corresponding

to first-order variables, but we do not consider them). Then, the set of parents of $p \in C$, i.e., the set of positions $\{q \mid \exists p \in C \,.\, \exists j \in \mathbb{N} \,.\, p = q \cdot j\}$ is an equivalence class of positions C'. This class of positions C' can contain cuts corresponding to context variables, but not to constants (this would imply that C contains cuts). Therefore, since the nodes of C' are labelled with a nonzero arity constant, we can apply some of the two previous cases to C', and reach a contradiction. □

We want to remark that Lemma 4 does not hold for general CU. For instance, the context unification equation $e = X(a) \stackrel{?}{=} Y(b)$ has a size-minimal unifier $\sigma = [X \mapsto f(\bullet, b), Y \mapsto f(a, \bullet)]$. The set of positions $\{1, 2\}$ is an $\approx^e_\sigma$-equivalence class (all of them corresponding to the same constant f), but there exists no f-labeled node in e that could generate a f-labelled cut in $\sigma(e)$. This size-minimal unifier, however, is not well-nested.

Lemma 5. *If σ is a normal unifier of an equation e, then, for any non-empty segment $\langle p, q\rangle$ of $\sigma(e)$, there exists an $\equiv^e_\sigma$ equivalent segment $\langle p', q'\rangle$ over a cut $\mathsf{c}^e_\sigma(r)$, i.e. there exist $r \in \mathsf{pos}(e)$ and $\langle p', q'\rangle \equiv^e_\sigma \langle p, q\rangle$ such that $\mathsf{c}^e_\sigma(r)$ is inside $\langle p', q'\rangle$.*
Moreover, if $\langle p, q\rangle$ contains just one position, then we can restrict $\mathsf{lab}_e(p')$ to be a unary constant, and if $\langle p, q\rangle$ contains two or more positions, then we can restrict r to satisfy $\mathsf{c}^e_\sigma(r) \neq p'$.

Proof. Consider a segment $\langle p, q\rangle$ containing just one position, this must be p, and $q = p \cdot 1$. Consider the $\approx^e_\sigma$-equivalence class defined by p. By Lemma 4, there exists a cut $p' \approx^e_\sigma p$ corresponding to a constant $f = \mathsf{lab}_e(\mathsf{c}^e_\sigma{}^{-1}(p'))$ in the same equivalence class. Then, the segment $\langle p', p' \cdot 1\rangle$ fulfils our requirements, because $p \approx^e_\sigma p'$ implies $\langle p, p \cdot 1\rangle \equiv^e_\sigma \langle p', p' \cdot 1\rangle$.

Consider a segment $\langle p, q\rangle$ containing more than one position. Suppose that the lemma does not hold, then any segment $\langle p', q'\rangle$, in the same $\equiv^e_\sigma$-equivalence class as $\langle p, q\rangle$, does not contain any cut, except for possibly p'. Comparing the definitions of $\approx^e_\sigma$ and $\equiv^e_\sigma$, in such conditions, we have $\langle p, p \cdot s\rangle \equiv^e_\sigma \langle p', p' \cdot s\rangle$ implies $p \approx^e_\sigma p'$. Now, by Lemma 4, there exists a position $p' \approx^e_\sigma p$ over a cut corresponding to a constant. Therefore, for any integer i, the position $p' \cdot i$ is also a cut. Now, since $\langle p', q'\rangle$ contains more than one position, it must contain $p' \cdot i$, for some i, hence it contains another cut, apart from p', which contradicts the initial supposition. □

Lemma 6. *If σ is a normal unifier of an equation e, then, for every variable $X \in \mathsf{Var}(e)$, there exists a segment $\langle p, q\rangle$ in e such that $\sigma(X) = \sigma(e\langle p, q\rangle)$, where the segment $e\langle p, q\rangle$ is not of the form $Y(\bullet)$.*

Proof. There are several cases. If $\sigma(X)$ is the empty context $\bullet$ then, for any empty segment $\langle p, p\rangle$ of e, we have $\sigma(X) = \sigma(e\langle p, p\rangle) = \sigma(\bullet) = \bullet$.

If $\sigma(X) = f(\bullet)$, we can assume that X occurs in e, say at position p, and $\sigma(X)$ occurs in $\sigma(e)$, at segment $\langle \mathsf{c}^e_\sigma(p), \mathsf{c}^e_\sigma(p \cdot 1)\rangle = \langle \mathsf{c}^e_\sigma(p), \mathsf{c}^e_\sigma(p) \cdot 1\rangle$, otherwise σ would not be size-minimal. Now, using Lemma 5, we have an occurrence r in e such that $\langle \mathsf{c}^e_\sigma(r), \mathsf{c}^e_\sigma(r) \cdot 1\rangle \equiv^e_\sigma \langle p, p \cdot 1\rangle$ and $\mathsf{lab}_e(r) = f$. Therefore, $\sigma(X) = \sigma(e\langle r, r \cdot 1\rangle) = f(\bullet)$.

If $\sigma(X)$ contains two or more positions, let p be an occurrence of X in e, and $\langle \mathsf{c}^e_\sigma(p), \mathsf{c}^e_\sigma(p \cdot 1)\rangle$ be the corresponding occurrence of $\sigma(X)$ in $\sigma(e)$. By Lemma 5 there exists an occurrence $\langle p', q'\rangle$ of $\sigma(X)$ over a cut: $\langle p', q'\rangle \equiv^e_\sigma \langle \mathsf{c}^e_\sigma(p), \mathsf{c}^e_\sigma(p \cdot 1)\rangle$, and there exists a position r in e such that $\mathsf{c}^e_\sigma(r)$ is inside $\langle p', q'\rangle$ and $\mathsf{c}^e_\sigma(r) \neq p'$. Now, we will prove that p' and q' are also cuts.

Suppose that p' is not a cut. Let $s \in \mathsf{pos}(e)$ be the longest position such that $\mathsf{c}^e_\sigma(s) \preceq p'$. Then, $\mathsf{c}^e_\sigma(s) \prec p' \prec \mathsf{c}^e_\sigma(s \cdot 1)$, hence s corresponds to a variable occurrence. Since $\langle \mathsf{c}^e_\sigma(s), \mathsf{c}^e_\sigma(s{\cdot}1)\rangle$ does not contain other cuts than $\mathsf{c}^e_\sigma(s)$, we have $\mathsf{c}^e_\sigma(s{\cdot}1) \preceq \mathsf{c}^e_\sigma(r)$. This gives two overlapping segments $\langle p', q'\rangle$ and $\langle \mathsf{c}^e_\sigma(s), \mathsf{c}^e_\sigma(s{\cdot}1)\rangle$ that violate the well-nestedness condition. A similar argument allows us to prove that q' is a cut.

We can conclude that $\langle \mathsf{c}^{e\,-1}_\sigma(p'), \mathsf{c}^{e\,-1}_\sigma(q')\rangle$ is a segment of e containing a position r with $r \neq \mathsf{c}^{e\,-1}_\sigma(p')$ and $r \neq \mathsf{c}^{e\,-1}_\sigma(q')$, therefore fulfilling the conditions of the lemma. □

Theorem 1. *Normal unifiers σ of a context equation e have a representation of the form:*

$$\sigma = [X_1 \mapsto e\langle p_1, q_1\rangle] \circ \cdots \circ [X_n \mapsto e\langle p_n, q_n\rangle]$$

where $\langle p_i, q_i\rangle$ are segments of e and X_i variables of $\mathsf{Var}(e)$ that do not occur in $e\langle p_j, q_j\rangle$ for $j \leq i$.
This representation is polynomial in $|e|$, and does not use constants not occurring in e.

Proof. By Lemma 6, for any variable X_i of the equation, we can find a segment $\langle p_i, q_i\rangle$ of the equation such that $\sigma(X_i) = \sigma(e\langle p_i, q_i\rangle)$, where $\langle p_i, q_i\rangle$ is something else than just one variable. For any variable X_j occurring in $e\langle p_i, q_i\rangle$ we say that $X_i > X_j$. The transitive closure of this relation results into an irreflexive relation. Now, Let $X_1 > X_2 > \cdots > X_n$ be a total ordering of the variables of the equation compatible with this transitive and irreflexive ordering. This ordering is used to express the σ as it is stated in the theorem. □

6 Well-nested Context Unification is in NP

In this section we prove that given an equation e, and a substitution σ that can be represented in polynomial space on $|e|$ in the form $[X_1 \mapsto t_1] \circ \cdots \circ [X_n \mapsto t_n]$, we can check if σ is a unifier of e in polynomial time on $|e|$. This result is not trivial, since $|\sigma(e)|$ can be exponential in $|e|$. For instance, the equation:

$$X_n(X_{n-1}(\cdots X_1(a)\cdots)) \stackrel{?}{=} X_{n-1}(X_{n-1}(X_{n-2}(X_{n-2}(\cdots f(a))\cdots))))$$

has as unique unifier that we can represent as follows:

$$\sigma = [X_1 \mapsto f(\bullet)] \circ [X_2 \mapsto X_1(X_1(\bullet))] \circ \cdots \circ [X_n \mapsto X_{n-1}(X_{n-1}(\bullet))]$$

This representation has linear size on n, like the equation. However $\sigma(e) = f^{2^n-1}(a) \stackrel{?}{=} f^{2^n-1}(a)$ has exponential size on n.

To overcome this problem we construct a context free grammar generating a preorder traversal of the term $\sigma(t)$, and another for $\sigma(u)$. Both grammars will be of polynomial size (not so the word generated by them). Then, we can use a result (see Lemma 7) due to Plandowski that allows us to check the equality of the words generated by the two grammars in polynomial time on the size of the grammars.

A *context-free grammar (CFG)* is a 4-tuple (Σ, N, P, S), where Σ is an alphabet of *terminal* symbols, N is an alphabet of *non-terminal* symbols, P is a finite set of rules, and $S \in N$ is the *start symbol*. We will not distinguish a particular start symbol, and we will represent a context free grammars as a 3-tuple (Σ, N, P).

Definition 7. *A context free grammar $G = (\Sigma, N, P)$* generates *a word $w \in \Sigma^*$ if there exists a non-terminal symbol $A \in N$ such that w belongs to the language defined by (Σ, N, P, A). In such case, we also say that A generates w.*

We say that a context free grammar is a singleton CFG *if it is not recursive and every non-terminal symbol occurs in the left-hand side of exactly one rule. Then, every non-terminal symbol $A \in N$ generates just one word, noted w_A.*

Plandowski [Pla94,Pla95] defines singleton grammars, but he calls them *grammars defining set of words*. He proves the following result.

Lemma 7 ([Pla95], Theorem 33). *The word equivalence problem for singleton context-free grammars is defined as follows: given a grammar and two non-terminal symbols A and B, to decide whether $w_A = w_B$. This problem can be solved in polynomial worst-case time on the size of the grammar.*

In order to translate trees and contexts into sequences, we will use a pair of nonterminal symbols X^L and X^R for each context variable X. Then, the translation function is defined by:

$$\begin{array}{ll} \mathsf{trans}(a) = a & \mathsf{trans}(f(t_1, \ldots, t_n)) = f\,\mathsf{trans}(t_1) \cdots \mathsf{trans}(t_n) \\ \mathsf{trans}(\bullet) = \bullet & \mathsf{trans}(X(t)) = X^L\,\mathsf{trans}(t)\,X^R \end{array}$$

Given a unification equation $t \stackrel{?}{=} u$, and a guessed substitution $[X_1 \mapsto v_1] \circ \cdots \circ [X_n \mapsto v_n]$, where $v_i \in \mathsf{Cont}_\Sigma$, we generate the following grammar:

$$\left.\begin{array}{l|l} A \to \mathsf{trans}(t) & X_i^L \to \mathsf{left}(\mathsf{trans}(v_i)) \\ B \to \mathsf{trans}(u) & X_i^R \to \mathsf{right}(\mathsf{trans}(v_i)) \end{array}\right\} \text{for every } i = 1, \ldots, n$$

where $\mathsf{left}(\alpha \bullet \beta) = \alpha$ and $\mathsf{right}(\alpha \bullet \beta) = \beta$.

In the case of our example, the grammar would be:

$$\begin{array}{l} A \to X_n^L\,X_{n-1}^L \cdots X_1^L\,a\,X_1^R \cdots X_{n-1}^R\,X_n^R \\ B \to X_{n-1}^L\,X_{n-1}^L\,X_{n-2}^L\,X_{n-2}^L \cdots f\,a \cdots X_{n-2}^R\,X_{n-2}^R\,X_{n-1}^R\,X_{n-1}^R \\ \begin{array}{l|l} X_n^L \to X_{n-1}^L\,X_{n-1}^L & X_n^R \to X_{n-1}^R\,X_{n-1}^R \\ X_{n-1}^L \to X_{n-2}^L\,X_{n-2}^L & X_{n-1}^R \to X_{n-2}^R\,X_{n-2}^R \\ \cdots & \cdots \\ X_1^L \to f & X_1^R \to \epsilon \end{array} \end{array}$$

Lemma 8. *Given a context equation e, and a substitution of the form:*

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

where X_i are distinct context variables, v_i are contexts, and X_i does not occurs in $v_1 \dots v_i$, we can check if it is a unifier in polynomial time on $|e| + |\sigma|$.

Proof. Since every constant has a unique arity, then for every pair of terms $t, u \in \mathcal{T}_{\Gamma(\Sigma)}$, $t = u$ if, and only if $\mathsf{trans}(t) = \mathsf{trans}(u)$.

Now, we can prove, by induction on k, that the grammar

$$\left.\begin{array}{l} A \to \mathsf{trans}(t) \\ B \to \mathsf{trans}(u) \end{array}\right| \quad \left.\begin{array}{l} X_i^L \to \mathsf{left}(\mathsf{trans}(v_i)) \\ X_i^R \to \mathsf{right}(\mathsf{trans}(v_i)) \end{array}\right\} \text{ for every } i = n-k, \dots, n$$

using A as the start symbol, can generate $\mathsf{trans}([X_{n-k} \mapsto v_{n-k}] \circ \cdots \circ [X_n \mapsto v_n]t)$, and similarly for B and u. Finally, using Lemma 7, we simply have to check if A and B generate the same sequence, in polynomial time on the size of the grammar, that is polynomial on $|e| + |\sigma|$. □

Theorem 2. *Deciding if a context equation has a well-nested unifier is NP-complete.*

Proof. Given an equation e, we can guess a representation for normal unifiers in polynomial time (Theorem 1), and then check in polynomial time whether it is represents indeed a unifier (Lemma 8). Finally, it can be proved that checking if the unifier represented in the form of Theorem 1 is well-nested can also be done in polynomial time using similar techniques.

For NP-hardness, it is sufficient to note that well-nested CU subsumes string matching [Ang80]. This is since the standard encoding of string matching will produce context equations that only have well-nested unifiers. □

7 Conclusions

Well-nested context unifiers are a kind of context unifiers with interest in computational linguistics. Here we prove that decidability of the existence of a well-nested context unifier is NP-complete. Additionally, we prove that well-nested size-minimal unifiers can be represented in polynomial space, as a composition of substitutions where each one of them instantiate a context variable by a *segment* of the original equation (a similar result holds for most general unifiers in first-order unification). As a direct consequence, well-nested size-minimal unifiers do not use constants not occurring in the original equation, a wishful property for computational linguistic applications. All these results are extensible to word unification.

In the future, we plan to study the relationship between well-nested context unification and well-nested parallelism constraints, and to look for a more efficient algorithm to compute well-nested unifiers than the brute force guessing.

Acknowledgments: We acknowledge the suggestions of the anonymous referees and the fruitful discussions with Katrin Erk.

References

[Ang80] D. Angluin. Finding patterns common to a set of strings. *Journal of Computer and Systems Sciences*, 21:46–62, 1980.

[BDMN04] M. Bodirsky, D. Duchier, S. Miele, and J. Niehren. A new algorithm for normal dominance constraints. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 54–78, 2004.

[Com92] H. Comon. Completion of rewrite systems with membership constraints. In *Int .Coll. on Automata Languages and Progr.*, vol. 623 of *LNCS*, 1992.

[DSP91] M. Dalrymple, S. Shieber, and F. Pereira. Ellipsis and higher-order unification. *Linguistics & Philosophy*, 14:399–452, 1991.

[EKN01] M. Egg, A. Koller, and J. Niehren. The constraint language for lambda structures. *Journal of Logic, Language, and Information*, 10:457–485, 2001.

[EN03] K. E. Erk and J. Niehren. Well-nested parallelism constraints for ellipsis resolution. In *11th Conf. of the European Chapter of the Ass. of Comp. Ling.*, pages 115–122, 2003.

[GK96] C. Gardent and M. Kohlhase. Higher–order coloured unification and natural language semantics. In *34th Meet. of the Ass. for Comput. Ling.*, 1996.

[Lev96] J. Levy. Linear second-order unification. In *7th Int. Conf. on Rewriting Techniques and Applications*, vol. 1103 of *LNCS*, pages 332–346, 1996.

[LV00] J. Levy and M. Villaret. Linear second-order unification and context unification with tree-regular constraints. In *11th Int. Conf. on Rewriting Techniques and Applications*, vol. 1833 of *LNCS*, pages 156–171, 2000.

[Mak77] G. S. Makanin. The problem of solvability of equations in a free semigroup. *Math. USSR Sbornik*, 32(2):129–198, 1977.

[Mon73] R. Montague. The proper treatment of quantification in ordinary English. In *Approaches to Natural Language*. Dordrecht, 1973.

[NK01] J. Niehren and A. Koller. Dominance constraints in context unification. In *Logical Aspects of Computational Linguistics*, vol. 2014 of *LNAI*, pages 199–218, 2001.

[NPR97] J. Niehren, M. Pinkal, and P. Ruhrberg. A uniform approach to underspecification and parallelism. In *35th Meeting of the Association of Computational Linguistics*, pages 410–417, 1997.

[Pla94] W. Plandowski. Testing equivalence of morphisms in context-free languages. In J. van Leeuwen, editor, *Proc. of the 2nd Annual European Symposium on Algorithms*, vol. 855 of *LNCS*, pages 460–470, 1994.

[Pla95] W. Plandowski. *The Complexity of the Morphism Equivalence Problem for Context-Free Languages*. PhD thesis, Department of Mathematics, Informatics and Mechanics, Warsaw University, 1995.

[Pla99] W. Plandowski. Satisfiability of word equations with constants is in PSPACE. In *40th IEEE Found. of Comp. Science*, pages 495–500, 1999.

[PR98] W. Plandowski and W. Rytter. Application of lempel-ziv encodings to the solution of words equations. In *25th ICALP*, vol. 1443 of *LNCS*, pages 731–742, 1998.

[Sag76] I. Sag. *Deletion and logical form*. PhD thesis, MIT, Cambridge, 1976.

[SS02] M. Schmidt-Schauß. A decision algorithm for stratified context unification. *Journal of Logic and Computation*, 12:929–953, 2002.

[SSS99] M. Schmidt-Schauß and K. U. Schulz. Solvability of context equations with two context variables is decidable. In *16th Int. Conf. on Automated Deduction*, LNAI, pages 67–81, 1999.