Solvability of Context Equations with two Context Variables is Decidable

Manfred Schmidt-Schauß
Fachbereich Informatik
Johann Wolfgang Goethe-Universität
Postfach 11 19 32
D-60054 Frankfurt,Germany
schauss@ki.informatik.uni-frankfurt.de

Klaus U. Schulz
CIS
Universität München
Oettingenstr. 67
D-80538 München, Germany
schulz@cis.uni-muenchen.de

January 14, 1999

Abstract

Context unification is a natural variant of second order unification that represents a generalization of word unification at the same time. While second order unification is wellknown to be undecidable and word unification is decidable it is currently open if solvability of context equations is decidable. We show that solvability of systems of context equations with two context variables is decidable. The context variables may have an arbitrary number of occurrences, and the equations may contain an arbitrary number of individual variables as well. The result holds under the assumption that the first order background signature is finite.

1 Introduction

A ground context is a ground term with exactly one occurrence of a special constant that represents a missing argument. Ground contexts can be applied to ground terms, which results in a replacement of the special constant by the given ground term. Similarly ground contexts can be applied to ground contexts. From this perspective, a ground term can be considered to be composed of ground contexts and ground terms in different ways.

Context unification speaks about this form of composition. Given a first-order signature, a set of individual variables and a set of so-called context variables, context terms are built like first order terms with additional context variables. Syntactically, context variables are treated like unary function symbols. A context equation is an equation between context terms. A solution of a context equation is a substitution that maps both sides of the equation to identical ground terms, where substitutions are mappings that assign ground contexts over the given first order signature to context variables. The main result of this paper is the following

Theorem 1.1 (Main Theorem) *Solvability of finite systems of context equations with two context variables is decidable.*

Both context variables may have an arbitrary number of occurrences, and the equations may contain an arbitrary number of individual variables as well. The result holds under the assumption that the first order background signature is finite. The paper provides a partial solution to the *Context Unification Problem* that has recently attracted considerable attention [3, 4, 13, 29, 18, 19, 24]:

Is solvability of arbitrary context equations decidable?

The interest in this problem relies on its close connection to several other well-studied decision problems.

Context unification represents a natural variant of second order unification, which is known to be undecidable ([8, 6, 14]). Ground contexts—the substitution instances of context variables—can be considered as λ-terms with exactly one occurrence of a single λ-bound variable.[1] Second order unification is different in two respects: first, substitution instances of second order variables may have an arbitrary number of λ-bound variables (depending on the arity of the variable). Second, there is no limitation on the number of occurrences of a given bound variable in the substitution term, and in particular this number may be zero. This second property makes an important difference to context unification. A recent result [23] shows that second order unification becomes decidable if an upper bound on the number of occurrences of a given bound variable in the substitution term is fixed. If context unification would turn out to be decidable, the latter result would be a simple consequence. It is known that second order unification is undecidable even for problems with one second order variable only [7]. Hence our results show that with respect to decidable fragments there is

[1] In other formulations of context unification, instances of second order variables may have an arbitrary number of bound variables, each having exactly one occurrence.

at least some significant difference between context unification and second order unification.

Context unification can also be considered as a generalization of word unification [15, 1, 11, 25, 26, 12, 5]. Decidability of word unification had been an open problem for many years. The problem was raised by A. A. Markov in the late 1950's who hoped to prove the undecidability of Hilbert's tenth problem by showing undecidability of the word unification problem. In this context, Y. Matiyasevich [17] gave a simple decision procedure for word unification problems where each variable occurs at most twice. Later, J.I. Hmelevskii (see [10]) proved decidability of word unification for problems with two and three variables with an arbitrary number of occurrences. In his famous paper [15], G. S. Makanin then fully solved the problem, showing that solvability of arbitrary word equations is decidable.

In [25] Makanin's result was generalized in the following way: given a word equation $W_1 = W_2$ with variables in $\{X_1, \ldots, X_n\}$ and constants in the finite alphabet $\mathcal{C}$, and given regular languages $\mathcal{L}_1, \ldots, \mathcal{L}_n$ over $\mathcal{C}$, it is decidable if there exists a solution S of $W_1 = W_2$ where $S(X_i) \in \mathcal{L}_i$, for $i = 1, \ldots, n$. This result will be used here to prove the Main Theorem.

An important role of Makanin's result or its extension in [25] is the following. Given a new decision problem $\mathcal{P}$, a reduction of $\mathcal{P}$ to word unification (with regular constraints) shows that $\mathcal{P}$ is decidable. Conversely a reduction of word unification to $\mathcal{P}$ shows that the problem is hard. See [5, 9] for various applications of this technique. On this background it seems clear that a positive decidability result for context unification would have many interesting consequences. From a practical point of view, context unification is used as a formalism for semantic analysis of natural language utterances [18, 19]. It also occurs in the context of distributive unification [22] and completion of rewrite systems with membership constraints [3, 4].

Some partial results concerning context unification are the following. J. Levy [13] has shown that solvability of context unification problems where each variable occurs at most twice is decidable. In [21] there is an algorithm and a sketch of a proof showing that solvability of so-called stratified context unification problems is decidable; a complete proof for a restricted signature is published in [22]. Stratification imposes strong restrictions on the nesting of context variables. In [24] the authors have given an upper bound on the so-called exponent of periodicity of a minimal solution of a context equation. In the case of word equations a similar bound was a key ingredient of Makanin's decidability result. J. Niehren, M. Pinkal and P. Ruhrberg [18] showed that context unification and so-called "equality up-to constraints" are equally expressive. It is also shown there that one-step rewriting constraints can be expressed by stratified context unification problems. Recently, it was noticed [20] that the converse is also true, which shows that stratified context unification and one-step rewriting constraints are interreducible. It was also noticed in [18] that the first-order theory of context unification is undecidable, using the fact that the first-order theory of one-step rewriting is undecidable [27, 28, 16]. This result was improved by S. Vorobyov [29] who showed that the $\forall\exists^8$ equational theory of context unification is co-recursively enumerable hard.

The proof of Theorem 1.1 uses a series of four non-deterministic translation steps. First we restrict considerations to single context equations. Eventually, a given context equation with two context variables is translated into a finite set of systems of word equations with so-called linear constant restrictions. Linear constant restrictions [2] represent a special type of the regular constraints described above.

1. In the first translation step, which will be described in Section 4, a given context equation is non-deterministically transformed into a so-called generalized context problem. Such a problem essentially gives a partial description of a ground term that represents a (hypothetical) solution of the given equation.

2. In the second step (Section 5), this description is slightly extended by guessing the function symbols at so-called branching points of context variables.

3. Whereas the first two translation steps work for an arbitrary number of context variables, the third translation step in Section 6 is much more delicate. Here some subparts of the given generalized context problem have to be identified, and as a result we may obtain other subparts that are again subject to identification. In general it seems at least very difficult to find a strategy for identification that can guarantee termination. However, in the case of two context variables a special failure condition can be used that solves this problem. The rather technical correctness proof for the third translation is given in the Appendix.

4. In Section 7 finally the resulting problems are translated into systems of word equations with linear constant restriction. This last translation again does not depend on the number of context variables.

In Section 8 we combine the results for the single translation steps and prove Theorem 1.1, first for the case of a single context equation. We show how to extend the decision procedure to finite systems of context equations. A whole bunch of technical notions that is needed will be given in the following section.

2 Formal Preliminaries

Throughout this paper, Σ denotes a finite signature of function symbols having at least one constant. With $ar(f)$ we denote the arity of the function symbol $f \in \Sigma$. Besides the symbols from Σ, a special constant "Ω" that does not belong to Σ will be used. With $\Sigma^{\Omega} := \Sigma \cup \{\Omega\}$ we denote the extended signature where Ω is treated as a constant.

$\mathcal{V}$ denotes an infinite set of *context variables* $X, Y, Z, \ldots$. We shall also use individual variables $x, y, z, \ldots$, and $\mathcal{X}$ denotes the set of individual variables.

2.1 Notions based on term representation

We first give syntax and semantics of context unification. To begin with, *ground terms* over Σ and (occurrences of) *subterms* are defined as usual.

Definition 2.1 A *ground context* is a ground Σ^{Ω}-term t that has exactly one occurrence of the constant Ω, called the "hole" of t. With a *subterm* of a ground context t we always mean a Σ-subterm of t. The ground context Ω is called the *empty ground context*.

Given a ground context s and a ground term/context t we write st for the ground term/context that is obtained from s when we replace the occurrence of Ω in s by t. Note that this form of composition is associative.

Definition 2.2 The set of *context terms* over Σ, $\mathcal{X}$ and $\mathcal{V}$ is inductively defined as follows:

- each constant $a \in \Sigma$ is a context term,
- each individual variable $x \in \mathcal{X}$ is a context term,
- if $t_1, \ldots, t_n$ are context terms and $f \in \Sigma$ is n-ary, then $f(t_1, \ldots, t_n)$ is a context term,
- if t is a context term and $X \in \mathcal{V}$, then $X(t)$ is a context term.

A *context equation* is an equation of the form $s = t$ where s and t are context terms.

Definition 2.3 A *substitution* is a mapping S that assigns a ground context $S(X)$ to each $X \in \mathcal{V}$, and a ground term $S(x)$ to each $x \in \mathcal{X}$. The mapping S is extended to arbitrary context terms as follows

- $S(a) := a$ for each constant $a \in \Sigma$,
- $S(f(t_1, \ldots, t_n) := f(S(t_1), \ldots, S(t_n))$ for n-ary $f \in \Sigma$,
- $S(X(t)) := S(X)S(t)$ for $X \in \mathcal{V}$.

A substitution S is a *solution* of the context equation $s = t$ if $S(s) = S(t)$.

Example 2.4 Let $\Sigma := \{f, g, a\}$ where f is binary, g is unary, and a is a constant. The context equation $X(X(a)) = f(Y(f(Y(a), Z(a))), x)$ is solved by the substitution $X \mapsto f(g(\Omega), g(g(a)))$, $Y \mapsto g(\Omega)$, $Z \mapsto g(g(\Omega)), x \mapsto g(g(a))$ since under this substitution both sides are mapped to $f(g(f(g(a), g(g(a)))), g(g(a)))$.

Definition 2.5 A solution S of the context equation $s = t$ is *positive* iff S maps each context variable appearing in $s = t$ to a non-empty ground context.

Clearly, in order to decide solvability of a given context equation $s = t$ it suffices to have a procedure for deciding positive solvability: we may simply guess which context variables are instantiated by the empty ground context and instantiate these variables by Ω in the equation $s = t$. *In the sequel, with a solution of a context equation we always mean a positive solution.*

Definition 2.6 The ground context s is a *subcontext* of the ground term t iff there exists a ground term t_1 and a ground context r such that $t = rst_1$.

Note that in this situation the root of the given occurrence of t_1 marks the position of the (filled) hole of s. In this sense we can talk about the position of the hole of an occurrence of a ground subcontext, regardless of the fact that the ground term t does not have any occurrence of the constant Ω.

Definition 2.7 The ground context s is a *suffix* of the ground context t iff t can be represented in the form $t = rs$, for some ground context r.

Definition 2.8 Let t be a ground context. The *main path* of t is the path from the root to Ω. The *side area* of t is the set of all nodes that do not belong to the main path.

Following the remarks above we shall also talk about the main path/side area of a given occurrence of a ground context as a subcontext of a ground term. The following definition yields the basis for the correspondence between ground contexts and words that we shall use in Section 7 for the final translaton step.

Definition 2.9 A ground context C is a *letter* iff the hole of C is in depth 1. For each ground context C with hole in depth k there exists a unique sequence of letters $C_1, \ldots, C_k$ such that $C = C_1 \cdots C_k$. The elements $C_1, \ldots, C_k$ are called the *letters* of C.

Let us fix some notational convention for the rest of the paper. With a ground term we always mean a ground Σ-term if not mentioned otherwise. Ground terms and ground contexts will be denoted with letters t, r, s. When we fix a particular occurrence of a ground term/context in a given ground term/context we use expressions with superscripts like $t^{(i)}, r^{(j)}$ etc. The same convention will be used for other expressions.

2.2 Notions based on tree representation

Definition 2.10 A *tree domain* is a finite, non-empty set N of sequences of positive natural numbers of length $n \geq 0$ such that

1. N is *rooted*, i.e., there exists an element $\eta \in N$ such that η is a prefix of each element of N,
2. $\eta_1 \in N$ and $\eta_1 \circ \eta_2$ in N implies $\eta_1 \circ \eta_3 \in N$, for all prefixes η_3 of η_2.

The elements of N are called *nodes*. If the node η is a proper prefix of η', then η' is called a *descendant* of η, and η is an *ancestor* of η'. If the length of η' is the length of η plus 1, then η' is a *child* of η. A node is a *leaf* if it does not have any child, otherwise it is an *inner node*. A node of N is *branching* if it has at least two distinct children in N. Two nodes η_1 and η_2 are *incompatible* if neither η_1 is a prefix of η_2 nor vice versa.

Definition 2.11 A *subtree domain* of a tree domain N is a subset N' of N that is a tree domain.

The *preorder* relationship on a tree domain is the strict linear order where nodes are enumerated using the following recursive traversal order: to traverse a given subtree domain, first traverse the first (second,...) immediate subtree domain, and then visit the root. This relation extends both ancestor relationship and the left-to-right ordering.

Definition 2.12 Let N be a tree domain. A *field* of N is a sequence of nodes $\varphi = (\eta_0, \ldots, \eta_k)$ $(k \geq 1)$ such that η_{i+1} is a child of η_i, for $0 \leq i \leq k-1$. Node η_0 (η_k) is the *initial* (resp. *final*) node of the field, the number k is the *length* of the field. The *side area* of $(\eta_0, \ldots, \eta_k)$ is the set of all nodes of N that are descendants of one of the nodes $\eta_0, \ldots, \eta_{k-1}$, but neither in $\{\eta_0, \ldots, \eta_k\}$ nor in the set of descendants of η_k. Fields of length 1 are called *atomic*. Two fields of N are *branching* if they have a common node and if the final nodes are incompatible. The maximal common prefix of the final nodes is called the *branching point* of the two fields. Let η_1 and η_2 be incompatible nodes. The maximal common prefix of η_1 and η_2 is denoted $\mathrm{MCP}_N(\eta_1, \eta_2)$.

Definition 2.13 A *labeled tree domain* is a pair (N, Lab) where N is a tree domain and $Lab : N \to \Sigma$ is a partial function such that $Lab(\eta) = f \in \Sigma$ implies that η has exactly $k := ar(f)$ children of the form $\eta \circ \langle 1 \rangle, \ldots, \eta \circ \langle k \rangle$.

Note that each ground term in a natural way represents a labeled tree domain with a total labeling function. In the sequel, we do not distinguish between these two notions. We shall also treat ground contexts as totally labeled tree domains (N, Lab) where $Lab : N \to \Sigma^{\Omega}$.

Definition 2.14 If $\varphi = (\eta_0, \ldots, \eta_k)$ is a field of the ground term t, then φ together with its side area defines a unique non-empty ground subcontext s of t. The root of s is given by η_0, and η_k marks the position of the hole. This subcontext is called the *ground subcontext of t with main path φ.*

If $F : N_1 \to N_2$ is a mapping between two labeled tree domains (N_1, Lab_1) and (N_2, Lab_2), we say that F *respects branching points* if $F(\mathrm{MCP}_{N_1}(\eta_1, \eta_2)) = \mathrm{MCP}_{N_2}(F(\eta_1), F(\eta_2))$ for all incompatible nodes η_1, η_2 of N_1. We say that F *respects children relationship for labeled nodes* if each child of a labeled node $\eta \in N_1$ is mapped to a child of $F(\eta)$ under F.

Definition 2.15 Let (N_1, Lab_1) and (N_2, Lab_2) be labeled tree domains. An injective mapping $F : N_1 \to N_2$ is a *labeled tree embedding* iff F respects root, Σ-labels, preorder relationship, branching points, and children relationship for labeled nodes.

If $\varphi = (\eta_0, \ldots, \eta_k)$ is a field of N_1 we write $F(\varphi)$ or $F(\eta_0, \ldots, \eta_k)$ for the subfield of N_2 with initial (resp. final) node $F(\eta_0)$ (resp. $F(\eta_k)$). Note that F respects branching points iff $F(\eta)$ is the branching point of $F(\varphi_1), F(\varphi_2)$, for all branching fields φ_1 and φ_2 of N_1 with branching point η.

3 Generalized Context Problems

We may now define the central data structure of the translation steps to be described later.

Definition 3.1 A *generalized context problem* over the signature Σ is a tuple $T = \langle N, \mathit{Lab}, \mathit{CB}, \mathit{Field}, \mathit{IB}, \mathit{Node}\rangle$ where

1. (N, Lab) is a labeled tree domain.
2. CB is a finite set. The elements of CB are called *context bases*. Each context base has a unique type. The type of a context base is a context variable. Context bases are written in the form cb, or in the more specific form $X^{(i)}$ where i is a natural number and X is the type.
3. Field is a function that assigns to each context base $cb \in \mathit{CB}$ a field $\mathit{Field}(cb)$ of N.
4. IB is a finite set. The elements of IB are called *individual bases*. Each individual base has a unique type, which is an individual variable. Individual bases are written in the form ib, or in the more specific form $x^{(i)}$ where i is a natural number and x is the type.
5. $\mathit{Node} : \mathit{IB} \to N$ is a total function.

The following conditions have to be satisfied:

6. each child of an unlabeled node of T is a non-initial node of the field of a base of T,
7. each leaf η of N is either labeled with an individual constant in Σ or there exists an individual base $ib \in \mathit{IB}$ with $\mathit{Node}(ib) = \eta$.

T is called *first order* if CB and Field are empty.

See Example 3.3 for a graphical representation of a generalized context problem. If T is first order, each inner node of T is labeled, by Condition 6. In this case T can be considered as a first order term where in addition nodes may be decorated with individual variables. More general, given the notion of a solution as introduced below, Condition 6 ensures that the values of the bases uniquely determine the solution.

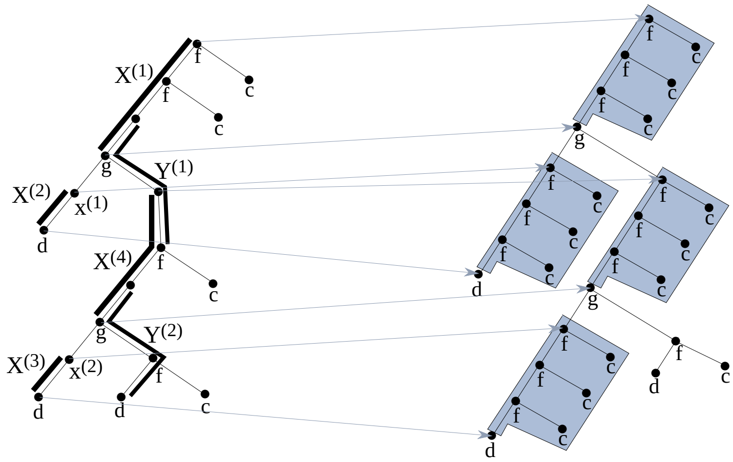
Since in a generalized context problem T each context base has a unique field we may also refer to the final (resp. initial) node, and similarly to the side area of a given context base. Two context bases are *branching* if their fields are branching. The *branching point* of the fields is also called the branching point of the two context bases. A context base cb is *atomic* if $Field(cb)$ is atomic. A context base cb_1 is a *subbase* of cb_2 if $Field(cb_1)$ is a subfield of $Field(cb_2)$.

Definition 3.2 Let $T = \langle N, Lab, CB, Field, IB, Node\rangle$ be a generalized context problem. A *solution* of T is a pair (t, S) where t is a ground term and S is a labeled tree embedding from N to the set of nodes of t such that the following conditions hold:

1. for all context bases $X^{(i)}$ and $X^{(j)}$ of the same type X of T the ground subcontexts of t with main paths $S(Field(X^{(i)}))$ and $S(Field(X^{(j)}))$ respectively are identical, and
2. for all individual bases $x^{(i)}$ and $x^{(j)}$ of the same type x of T the ground subterms of t with roots $S(Node(x^{(i)}))$ and $S(Node(x^{(j)}))$ respectively are identical.

If (t, S) is a solution of T and η is a node of T we write $\hat{S}(\eta)$ for the subterm of t with root $S(\eta)$. If φ is a field of T we write $\hat{S}(\varphi)$ for the ground subcontext of t with main path $S(\varphi)$. If $X^{(i)}$ is a context base of T we write $\hat{S}(X^{(i)})$ for $\hat{S}(Field(X^{(i)}))$. Since all context bases of type X are mapped to the same ground context we also write $\hat{S}(X)$ instead of $\hat{S}(X^{(i)})$. Similary, if $x^{(i)}$ is an individual base of T we write $\hat{S}(x^{(i)})$ for the ground subterm $\hat{S}(Node(x^{(i)}))$ and we write $\hat{S}(x)$ instead of $\hat{S}(x^{(i)})$.

Example 3.3 The following figure represents a generalized context problem T and the solution term t of a solution (t, S). The grey areas represent the ground contexts $\hat{S}(X) = \hat{S}(X^{(1)}) = \cdots = \hat{S}(X^{(4)})$.



Remark 3.4 Let η be a labeled node of the generalized context problem T, and let $\eta_1, \ldots, \eta_n$ be the children of η. If (t, S) is a solution of T, then $\hat{S}(\eta_1), \ldots, \hat{S}(\eta_n)$ is the sequence of the maximal subterms of $\hat{S}(\eta)$.

Remark 3.5 Let η be an unlabeled node of the generalized context problem T, and let η_1 and η_2 be two distinct children of η. If (t, S) is a solution of T, then $S(\eta)$ is the branching point of the fields $S(\eta, \eta_1)$ and $S(\eta, \eta_2)$ in t since S respects branching points. This shows that the number of children of η in T cannot exceed the arity of the label of $S(\eta)$.

The following definition introduces a concept that will become central later.

Definition 3.6 Let $T = \langle N, Lab, CB, Field, IB, Node \rangle$ be a generalized context problem. An atomic subfield (η, η') of the field of a base $cb \in CB$, with labeled node η, is called a *letter description* of T (or of cb) with main node η and label $Lab(\eta)$. If η' is the i-th child of η, then (η, η') has *direction* i.

We use symbols L, L_1 etc. to denote letter descriptions. With $ld(T)$ we denote the set of all letter descriptions of T. Note that if (t, S) is a solution of T and L is a letter description of the context base cb of T, then the ground subcontext $\hat{S}(L)$ is a subletter of $\hat{S}(cb)$. This follows easily from the fact that S preserves children of labeled nodes.

4 Translation of context equations into generalized context problems

In this section we define the notion of a superposition of two generalized context problems and use it for translating context equations into generalized context problems.

Definition 4.1 Let $T_1 = \langle N_1, Lab_1, CB_1, \mathit{Field}_1, IB_1, \mathit{Node}_1 \rangle$ and $T = \langle N, Lab, CB, \mathit{Field}, IB, \mathit{Node} \rangle$ be generalized context problems. An *embedding* of T_1 in T is a labeled tree embedding $F : N_1 \to N$ such that

1. each field of a context base of type X of T_1 is mapped to the field of a context base of the same of type X of T under F, for all $X \in \mathcal{V}$,
2. each node of an individual base of type x of T_1 is mapped to the node of an individual base of the same type x of T under F, for all $x \in \mathcal{X}$.

Lemma 4.2 *If there exists an embedding F of T_1 in T, and if (t, S) is a solution of T, then $(t, F \circ S)$ is a solution of T_1.*

Proof. Obvious. □

Definition 4.3 Let $n \geq 2$. The generalized context problem T is a *superposition* of the generalized context problems $T_1, \ldots, T_n$ iff for $i = 1, \ldots, n$ there are embeddings F_i of T_i in T such that the following conditions hold:

1. For each context base $X^{(r)}$ of T there exists $i \in \{1, \ldots, n\}$ and a context base $X^{(s)}$ of the same type of T_i such that $\mathit{Field}(X^{(r)}) = F_i(\mathit{Field}_i(X^{(s)}))$,
2. For each individual base $x^{(r)}$ of T there exists $i \in \{1, \ldots, n\}$ and an individual base $x^{(s)}$ of the same type of T_i such that $\mathit{Node}(x^{(r)}) = F_i(\mathit{Node}_i(x^{(s)}))$,
3. For each node η of T either there exists $i \in \{1, \ldots, n\}$ and a node η' of T_i such that $\eta = F_i(\eta')$, or there are two distinct indices $i, j \in \{1, \ldots, n\}$ and bases $cb_1 \in CB_i$ and $cb_2 \in CB_j$ such that the images of cb_1 and cb_2 in T branch at η.
4. Modulo renaming of exponents, the set of context (individual) bases of T is the union of the sets of context (individual) bases of the problems $T_1, \ldots, T_n$.

It should be noted that the nodes of individual bases are not necessarily leaves.

Lemma 4.4 *For $i = 1, \ldots, n$, let $T_i = \langle N_i, \mathit{Lab}_i, \mathit{CB}_i, \mathit{Field}_i, \mathit{IB}_i, \mathit{Node}_i \rangle$ be a generalized context problem.*

1. *The set of all superpositions of $T_1, \ldots, T_n$ is (modulo renaming of nodes/exponents) finite.*

2. *If, for some ground term t, each T_i ($1 \leq i \leq n$) has a solution of the form (t, S_i) where the embeddings S_i assign the same ground context (resp. term) to context (individual) bases of the same type, then there exists a superposition T of $T_1, \ldots, T_n$ with a solution of the form (t, S) where $S(S_i(\eta)) = S_i(\eta)$ for all nodes η of T_i ($1 \leq i \leq n$).*

Proof. 1. The first statement is an obvious consequence of Definition 4.3.

2. For $i = 1, \ldots, n$, let (t, S_i) be a solution of T_i where the solutions S_i assign the same ground context (term) to context (individual) bases of the same type. Call a node η of t *relevant* if it has the form $S_i(\eta')$ for some $\eta' \in N_i$ ($i \in \{1, \ldots, n\}$), or if there exist distinct indices $i, j \in \{1, \ldots, n\}$ and context bases $cb_1 \in \mathit{CB}_i$ and $cb_2 \in \mathit{CB}_j$ such that the fields $S_i(\mathit{Field}_i(cb_1))$ and $S_j(\mathit{Field}_j(cb_2))$ branch at η. Note that the root of t is relevant. Let N denote the set of relevant nodes of t. Let Lab denote the labeling function where exactly the nodes $\eta' \in N$ are labeled that have the form $S_i(\eta')$ for some labeled node $\eta' \in N_i$ ($i \in \{1, \ldots, n\}$), and where $\mathit{Lab}(\eta) := \mathit{Lab}_i(\eta')$. Since the mappings S_i are solutions it follows that Lab coincides with the natural labeling of the nodes of the ground term t which in particular implies that Lab is well defined. Now introduce a set of context (individual) bases, CB (resp. IB), and a function Field (resp. Node) as follows: for each base $X^{(r)}$ (resp. $x^{(r)}$) of T_i ($i \in \{1, \ldots, n\}$), CB (resp. IB) contains a base of type X (resp. x) with field $S_i(\mathit{Field}_i(X^{(r)}))$ (resp. node $S_i(\mathit{Node}_i(x^{(r)}))$).

It is obvious that $T := \langle N, \mathit{Lab}, \mathit{CB}, \mathit{Field}, \mathit{IB}, \mathit{Node} \rangle$ is a superposition of $T_1, \ldots, T_n$ where the mappings S_i represent the embeddings ($1 \leq i \leq n$). We show that (t, Id_N) (where Id_N is the identity mapping on N) represents a solution of T. Obviously Id_N respects root, Σ-labels, preorder relationship, and children relationship for labeled nodes.

In order to show that Id_N preserves branching points it suffices to show that for two incompatible relevant nodes η_1 and η_2 of N always $\eta := \mathrm{MCP}(\eta_1, \eta_2)$ is again relevant. We may assume that none of the nodes between η and η_1 (resp. η_2) is relevant. We distinguish two cases.

Case 1: there exists an index $i \in \{1, \ldots, n\}$ and nodes $\eta'_1, \eta'_2 \in T_i$ such that $\eta_j = S_i(\eta'_j)$ for $j = 1, 2$. Since S_i preserves branching points it follows that $\eta = \mathrm{MCP}_t(\eta_1, \eta_2) = S_i(\mathrm{MCP}_{N_1}(\eta'_1, \eta'_2))$ is relevant.

Case 2: In the remaining case there exist distinct indices $i, j \in \{1, \ldots, n\}$ and nodes $\eta'_1 \in N_i$ and $\eta'_2 \in N_j$ such that $\eta_1 = S_i(\eta'_1)$ and $\eta_2 = S_j(\eta'_2)$. We consider the parents η''_1 and η''_2 of the nodes η'_1 and η'_2 respectively. First assume that one of these nodes, say, η''_1, is labeled. Because S_1 preserves children relationship of labeled nodes, then the father of η_1 is labeled and relevant. Our assumptions imply that it coincides with η and we are done. In the second case both η''_1 and η''_2 are unlabeled. Hence, by Condition 6 of Definition 3.1 there exist context bases $cb_1 \in CB_i$ and $cb_2 \in CB_j$ such that $Field_i(cb_1)$ contains η''_1 and η'_1 and $Field_j(cb_2)$ contains η''_2 and η'_2. The image of $Field_i(cb_1)$ under S_i contains η_1 and the image of $Field_j(cb_2)$ under S_j contains η_2. Moreover, our assumptions on relevance of nodes imply that both image fields also contain η. It follows that the fields $S_i(Field_i(cb_1))$ and $S_j(Field_j(cb_2))$ branch at η, which implies that η is relevant.

Summing up, we have seen that Id_N is a labeled tree embedding. If $X^{(r)}$ and $X^{(s)}$ are two context bases of T, with fields φ_r and φ_s, say, then there exist bases $X^{(r')}$ and $X^{(s')}$ of problems T_i and T_j with fields $\varphi_{r'}$ and $\varphi_{s'}$ such that $S_i(\varphi_{r'}) = \varphi_r$ and $S_j(\varphi_{s'}) = \varphi_s$. Let $\varphi_{r''}$ and $\varphi_{s''}$ denote the fields of t that represent the images of $\varphi_{r'}$ (resp. $\varphi_{s'}$) under S_i (resp. S_j). Our assumptions on S_i and S_j show that the two ground subcontexts of t with main paths $\varphi_{r''}$ and $\varphi_{s''}$ coincide. On the other hand, by construction, the fields $\varphi_{r''}$ and $\varphi_{s''}$ of t represent the images of φ_r and φ_s under Id_N. Hence we have $\hat{Id}_N(X^{(r)}) = \hat{Id}_N(X^{(s)})$.

In the same way it follows that $\hat{Id}_N(x^{(r)}) = \hat{Id}_N(x^{(s)})$ for all individual bases $x^{(r)}$ and $x^{(s)}$ of the same type of T. It follows that (t, Id_N) is in fact a solution of T. Obviously we have $Id_N(S_i(\eta)) = S_i(\eta)$ for all nodes η of T_i $(1 \leq i \leq n)$. $\square$

Definition 4.5 For $i = 1, \ldots, n$, let $T_i = \langle N_i, Lab_i, CB_i, Field_i, IB_i, Node_i \rangle$ be a generalized context problem. Assume that each T_i $(1 \leq i \leq n)$ has a solution of the form (t, S_i) where the S_i assign the same ground context (term) to context (individual) bases of the same type. The superposition T of $T_1, \ldots, T_n$ described in Part 3 of the previous proof will be called the *superposition given by the joint image of* $T_1, \ldots, T_n$ *in* t *under* $S_1, \ldots, S_n$, and $S := Id_N$ is the *canonical solution of* T *extending* $S_1, \ldots, S_n$.

The following translation lemma makes use of the fact that each context term t represents in a natural way a generalized context problem, $T := (N_t, Lab_t, CB_t, Field_t, IB_t, Node_t)$, where N_t (Lab_t) is the set of positions (resp. labeling function) of the term t, where CB_t (res. IB_t) is given by the set of all occurrences of context (resp. individual) variables in t, where $Field_t(X^{(i)})$ has as its initial (resp. final) node the position of the i-th occurrence of X in t (resp. the position of the head symbol of its argument term), and where $Node(x^{(i)})$ is the position of the i-th occurrence of x in t. Note that $(N_t, Lab_t, CB_t, Field_t, IB_t, Node_t)$ is trivially solvable in the sense that each assignment of non-empty ground contexts (ground terms) to the context variables (individual variables) in t defines a solution of $(N_t, Lab_t, CB_t, Field_t, IB_t, Node_t)$.

Lemma 4.6 (Transl1) *For each context equation E it is possible to compute a finite set $\mathcal{T}$ of generalized context problems such that E has a solution iff some $T \in \mathcal{T}$ has a solution. If E has occurrences of k context variables only, the same holds for the problems in $\mathcal{T}$.*

Proof. Consider a context equation $t_1 \doteq t_2$. As described above, each of the context terms t_i can be considered as a trivially solvable generalized context problem T_i. Let $\mathcal{T}$ denote the set of superpositons of T_1 and T_2. If $t_1 \doteq t_2$ has a solution[2], then T_1 and T_2 have solutions (t, S_1) and (t, S_2) that satisfy the conditions given in Lemma 4.4. The lemma shows that an element of $\mathcal{T}$ has a solution. If some member T of $\mathcal{T}$ has a solution, then it follows from Lemma 4.2 that T_1 and T_2 have solutions of the form (t, S_1) and (t, S_2) respectively where S_1 and S_2 assign the same ground context to context (resp. individual) variables of the same type. It follows that $t_1 \doteq t_2$ has a solution. □

5 Transparent generalized context problems

In this section we describe a simple translation that assigns to a given generalized context problem T a finite set of generalized context problems T' where the branching points of bases are always labeled nodes. Given an unlabeled branching point, we essentially just guess the label of the branching point. Here it is essential that the given first-order signature Σ is finite.

[2] Recall that with a solution of a context equation we always mean a positive solution.

Definition 5.1 A generalized context problem T is *transparent* iff each branching point of two bases of T is always labeled.

The following simple observation will be used in the correctness proof of the final translation into word equations with linear constant restriction in Section 7.

Lemma 5.2 *Let T be a transparent generalized context problem. Each unlabeled inner node of T has exactly one child.*

Proof. Let η be an unlabeled inner node of the transparent generalized context problem T. By transparency, η cannot be a branching point of two bases of T. Part 6 of Definition 3.1 shows that η has exactly one child. $\square$

We may now give the translation procedure.

Definition 5.3 [**Procedure (Transl2)**] The *input* of this procedure is a generalized context problem T. If there exists a branching point η of two bases of T that is unlabeled, with children $\eta_1, \ldots, \eta_m$, say, then

1. non-deterministically choose a function symbol $f \in \Sigma$, of arity $n \geq m$, and label η with f,
2. introduce n new children in the left-to-right order $\eta'_1, \ldots, \eta'_n$.
3. Now each old child η_i is nondeterministically either identified with some new child η'_j, or appended as the unique child of η'_j. In this step, distinct new children η'_j are used for distinct old children η_i, and the left-to-right ordering is respected.
4. Each new child η'_j that is not used in the previous step represents a leaf. We introduce an individual base *ib of new type* and define $Node(ib) = \eta'_j$. We always use distinct types $x \in \mathcal{X}$ for individual bases of distinct leaves.
5. Repeat Steps 1-4 until all branching nodes are labeled.
6. Update the field function accordingly.

The *output* of the procedure consists of the set $\mathcal{T}$ of all generalized context problems that are reached by suitable choices in the nondeterministic steps.

Definition 5.4 Let $L = (\eta, \eta')$ be a letter description of the generalized context problem T with label f of arity n. The $n-1$ children of η that are different from η' are called the *top side nodes* of L. We write $\langle \eta_1, \ldots, \eta_{n-1} \rangle = tsn(L)$ if $\eta_1, \ldots, \eta_{n-1}$ is the sequence of all top side nodes of L in the natural left-to-right ordering. In this situation, η_i is called the i-th top side node of L. A node η is called a top side node of T if η is a top side node of a letter description of T. With $tsn(T)$ we denote the set of all top side nodes of T. If L and L' are two letter descriptions with the same label and direction, and if η (resp. η') is the i-th top side node of L (resp. L'), then η and η' are called *corresponding* top side nodes of L and L'.

Lemma 5.5 ((Transl2)) *Assume that (Transl2) generates the output set $\mathcal{T}$, given the generalized context problem T as input. Then*

1. *$\mathcal{T}$ is (modulo renaming of new individual bases) finite,*
2. *each element of $\mathcal{T}$ is a transparent generalized context problem. If T has only context bases of k types, then the same holds for the generalized context problems in $\mathcal{T}$.*
3. *if T has a solution, then some $T' \in \mathcal{T}$ has a solution,*
4. *if some $T' \in \mathcal{T}$ has a solution, then T is solvable.*

Proof. 1. Part 1 is obvious. 2. Let $T' \in \mathcal{T}$. In order to show that T' is a generalized context problem we verify that T' satisfies Condition 6 of Definition 3.1. All other conditions are trivially satisfied. Consider a new child η'_j. If η'_j represents an unlabeled node of the new problem, then either η'_j is identified with an unlabeled child η_i, or η'_j has exactly one child that represents an old child η_i. In the first case it follows that η'_j and its children satisfy Condition 6 of Definition 3.1. In the second case note that η_i, as a child of the unlabeled node η, is a non-initial node of some base of T. It follows again that η'_j and his unique child η_i satisfy Condition 6 of Definition 3.1. It follows from Part 3 of (Transl2) that the procedure does not introduce new branching points. Hence T' is transparent. Obviously, if T has only context bases of two types, then the same holds for the generalized context problems in $\mathcal{T}$.

3. Let (t, S) be a solution of T. Consider a branching point η of two bases of T that is unlabeled. If η_i and η_j are two distinct children of η, then the largest common prefix of $S(\eta_i)$ and $S(\eta_j)$ in t is $\mathrm{MCP}_t(S(\eta_i), S(\eta_j)) = S(\mathrm{MCP}_T(\eta_i, \eta_j)) = S(\eta)$. If follows easily that for one of the problems T'

generated by (Transl2) there exists an embedding S' such that (t, S') solves T'.

4. Part 4 follows from the fact that for each of the new problems T' there exist an embedding of T in T', using Lemma 4.2. $\square$

6 Identification of letter descriptions

We now come to the most difficult translation step. The basic idea is very simple. We want to guess which letter descriptions of a given transparent generalized context problem T are mapped to the same ground letter under a given (hypothetical) solution of T, and we want to identify these letter descriptions. We shall proceed in an indirect way and guess (roughly) which top side nodes of T are mapped to identical ground terms. The subtrees of these top side nodes are replaced by a common superposition, which means that each solution of the new problem will always map these subtrees to the same ground term. If for given letter descriptions L_1 and L_2 of the new problem each pair of corresponding top side nodes is identified in this sense, this also means that L_1 and L_2 are mapped to the same ground letter under any solution.

Given this simple idea, there are essentially two complications. First, a closer look at the technical details shows that we cannot simultaneously treat all equivalence classes of top side nodes that we would like to identify. The difficulty arises from the fact that a letter description may be part of the side area of another letter description. We proceed in an iterative way, identifying top side nodes in "top-down" manner.

Second, when we superimpose the subtrees of two given top side nodes we may obtain new branching points of bases in the superposition, which means that we may produce new letter descriptions and new top side nodes at such a step. As long as we do not restrict the number of context variables (i.e., the number of types of context bases), we see no way to guarantee termination. For this reason we consider input problems that only have context bases of two types. In this case, termination can be enforced.

The procedure will also identify the subtrees of all nodes that represent the image of the same individual variable.

Before we can give the algorithm, several concepts are needed. The first notion explains when two generalized context problems can be considered to be essentially identical.

Definition 6.1 For $i = 1, 2$, let $T_i = \langle N_i, Lab_i, CB_i, Field_i, IB_i, Node_i\rangle$ be a generalized context problem. T_1 and T_2 are *strictly isomorphic* iff there exists a bijection $F : N_1 \to N_2$ such that F is an embedding (cf. Def. 4.1) of T_1 in T_2 and F^{-1} is an embedding of T_2 in T_1. Each pair of the form $(\eta, F(\eta))$ $(\eta \in N_1)$ is called a pair of *corresponding nodes* of T_1 and T_2.

The following definition formalizes the concept of the subproblem of a generalized context problem T given by a particular node of T.

Definition 6.2 Let η be a node of the generalized context problem $T = \langle N, Lab, CB, Field, IB, Node\rangle$. The *subproblem of T defined by η* is the generalized context problem $T^\eta = \langle N^\eta, Lab^\eta, CB^\eta, Field^\eta, IB^\eta, Node^\eta\rangle$ with the following components: N^η is the set of descendants of η, together with η. Lab^η is the restriction of Lab to N^η. Let cb be a context base of CB such that $|Field(cb) \cap N^\eta| \geq 2$. If $Field(cb) \subseteq N^\eta$, then cb is a context base of CB^η with the same field and type as in T. If $Field(cb) \not\subseteq N^\eta$ we introduce a "placeholder" base cb' with field $Field^\eta(cb') := Field(cb) \cap N^\eta$ in CB^η that *represents* the suffix $Field(cb) \cap N^\eta$ of cb. The context base cb' receives a new context variable as its type. For each individual base ib of T with $Node(ib) \in N^\eta$ the set IB^η inherits a base ib of the same type with $Node^\eta(ib) := Node(ib)$.

Definition 6.3 A transparent generalized context problem T is *marked* iff for every atomic subfield φ of the field of a context base of T there exist a context base cb of T such that $Field(cb) = \varphi$.

The following lemma clarifies the role of markedness.

Lemma 6.4 *Let η_1 and η_2 be two nodes of the generalized context problem T with solution (t, S). If T is marked and if the problems T^{η_1} and T^{η_2} are strictly isomorphic, then $\hat{S}(\eta_1') = \hat{S}(\eta_2')$ for each pair of corresponding nodes (η_1', η_2') of T^{η_1} and T^{η_2}. In particular $\hat{S}(\eta_1) = \hat{S}(\eta_2)$.*

Proof. Simple, cf. Lemma 4.2. □

Definition 6.5 Let L and L' be two letter descriptions of the generalized context problem T with the same label and direction. L and L' are *strictly isomorphic* iff for all corresponding top side nodes η and η' of L and L' the subproblems T^η and $T^{\eta'}$ are strictly isomorphic.

The next proposition is a simple consequence of Condition 6 of Definition 3.1.

Proposition 6.6 *Let L_1 and L_2 be strictly isomorphic letter descriptions of the generalized context problem T. If (t, S) is a solution of T, then $\hat{S}(L_1) = \hat{S}(L_2)$.*

Definition 6.7 Let T be a marked generalized context problem. A solution (t, S) of T is *rigid* iff it satisfies the following condition for all letter descriptions L_1 and L_2 of T: $\hat{S}(L_1) = \hat{S}(L_2)$ iff L_1 and L_2 are strictly isomorphic.

Rigid solutions are introduced since in the last translation step only *rigid* solvability of a marked generalized context problem ensures solvability of its translation (a system of word equations with linear constant restriction.)

Definition 6.8 Let $T = \langle N, Lab, CB, Field, IB, Node\rangle$ be a transparent generalized context problem. Given an individual variable $x \in X$ we say that $\eta \in N$ is an *x-node of T* iff T has an individual base *ib* of type x with $Node(ib) = \eta$. Let $\mathcal{Y} \subseteq \mathcal{X}$ be a set of individual variables. The subset π of N is called $\mathcal{Y}$-closed iff, for each $y \in \mathcal{Y}$ the following condition holds: if π contains an y-node, then π contains each y-node of T.

We may now give the translation algorithm. In the following procedure we use two sets Π_i and Δ_i. Intuitively, Π_i collects the equivalence classes of all the nodes that have been identified already, and Δ_i represents the set of all nodes that are still subject to identification. In the sequel, let $rel(T)$ denote the set of all top side nodes and of all nodes of individual bases of T.

Definition 6.9 [**Procedure (Transl3)**] The *input* is a transparent generalized context problem $T = \langle N_0, Lab_0, CB_0, Field_0, IB_0, Node_0\rangle$ with context bases of type X or Y only. Let $\mathcal{X}_0$ denote the set of all individual variables $x \in \mathcal{X}$ such that T has an x-node. In a first step, T is transformed into the problem $T_0 = \langle N_0, Lab_0, CB_0, Field_0, IB_0, Node_0, \Pi_0, \Delta_0\rangle$ where $\Pi_0 := \emptyset$ represents the empty partition and $\Delta_0 := rel(T_0)$.

I. Assume that we have reached after i steps the problem

$$T_i = \langle N_i, Lab_i, CB_i, Field_i, IB_i, Node_i, \Pi_i, \Delta_i\rangle$$

where $\Pi_i = \{\pi_1, \ldots, \pi_i\}$ is a partition of a subset of $rel(T_i)$ and where $\Delta_i = (rel(T_i) \setminus \bigcup \Pi_i)$. If $\Delta_i = \emptyset$, then go to II, otherwise go to III.

II. If T_i is yet not completely marked, then we add appropriate bases of distinct type until a problem T' is reached that is completely marked. Now T' represents an *output problem* of (Transl3).

III. Choose a non-empty subset $\pi_{i+1} = \{\eta_1, \ldots, \eta_m\}$ of Δ_i that satisfies the following conditions:

(a) π_{i+1} does not contain two labeled nodes with distinct label,

(b) π_{i+1} is a set of maximal elements of Δ_i in the sense that π_{i+1} does not have any element that is a descendant of another node in Δ_i,

(c) π_{i+1} is $\mathcal{X}$-closed.

If this is not possible, then fail. Otherwise

1. nondeterministically choose a superposition T^{S_0} of the problems $T_i^{\eta_1}, \ldots, T_i^{\eta_m}$ defined by $\eta_1, \ldots, \eta_m$.

2. If π_{i+1} does not have any x-node for some $x \in \mathcal{X}_0$, then we apply the following *Failure Condition*: If T^{S_0} contains two context bases cb_1 and cb_2 such that cb_1 has a node that falls in the side area of cb_2, then fail.

3. If the superposition T^{S_0} contains any pair of branching bases where the branching node is unlabeled, then introduce a label using the same procedure as in (Transl2). If at this step we introcude new individual bases (cf. Step 4 of (Transl2)), then it is important to use a new type z that is not in $\mathcal{X}_0$. Repeat this until the superposition represents a transparent problem T^{S_1}.

4. With each atomic subfield of a context base of T^{S_1} associate a new context base $Z^{(0)}$, using distinct base types Z for distinct fields. Let T^S denote the resulting problem.

5. Replace each problem $T_i^{\eta_j}$ $(1 \leq j \leq m)$ by a strictly isomorphic copy T_j^S of T^S. If the (placeholder) base cb' of T^S represents a (suffix of a) base cb in $T_i^{\eta_j}$, then (the suffix of) cb and its corresponding

(placeholder) base receive the same field in T_j^S. For each context base of the superposition T^S that is not a placeholder base, including the new bases of the form $Z^{(0)}$ introduced in Step 4, we use a new base of the same type in T_j^S, context bases that represent the same context base of T^S receive corresponding fields in the problems T_j^S. Similarly, for each individual base of T^S we use a new base of the same type in T_j^S, individual bases that represent the same individual base of T^S receive corresponding nodes in the problems T_j^S.

Let

$$T_{i+1} = \langle N_{i+1}, \mathit{Lab}_{i+1}, \mathit{CB}_{i+1}, \mathit{Field}_{i+1}, \mathit{IB}_{i+1}, \mathit{Node}_{i+1}, \Pi_{i+1}, \Delta_{i+1} \rangle$$

denote the problem obtained in this way where $\Pi_{i+1} := \Pi_i \cup \{\pi_{i+1}\}$ and $\Delta_{i+1} := \mathit{rel}(T_{i+1}) \setminus \bigcup \Pi_{i+1}$. Go to I.

Note that if the input problem T_0 is first order, then (Transl3) reduces to a first order unification procedure where the failure condition before Step 1 of Case III corresponds to the occur-check. For the following two lemmas we always assume that we use as input for (Transl3) a transparent generalized context problem T with bases of type X or Y. In the Appendix we prove (cf. Lemma 9.4)

Lemma 6.10 *The procedure (Transl3) terminates.*

Also the rather technical proof of the following theorem will be given in the Appendix (cf. Theorem 9.18).

Theorem 6.11 *Let $\mathcal{T}$ denote the output set of (Transl3).*

1. *$\mathcal{T}$ is finite,*
2. *Each $T' \in \mathcal{T}$ is a marked generalized context problem.*
3. *If T has a solution, then there exists a problem $T' \in \mathcal{T}$ such that T' has a rigid solution.*
4. *If in T' two individual bases ib_1 and ib_2, with nodes η_1 and η_2, say, have the same type, then the subproblems T'^{η_1} and T'^{η_2} are strictly isomorphic.*
5. *If some $T' \in \mathcal{T}$ has a solution, then T is solvable.*

7 Translation into word equations with linear constant restriction

We come now to the final translation step. In this section we fix a transparent and marked generalized context problem $T = \langle N, Lab, CB, Field, IB, Node\rangle$ that satisfies the following condition (†): If in T two individual bases ib_1 and ib_2, with nodes η_1 and η_2, say, have the same type, then the subproblems T^{η_1} and T^{η_2} are strictly isomorphic. Recall that the output problems of (Transl3) have this property, by Theorem 6.11. For simplicity we also assume that for each field φ of T there exists at most one base cb of T with field φ. It is simple to see that a given generalized context problem can always be "normalized" in this sense without changing any of the relevant properties and preserving (rigid) solvability in both directions. In fact, whenever two bases cb_1 and cb_2 of T, say, of type X and Y, have the same field, we may erase cb_2 and afterwards assign the new type X to all other bases of type Y.

We consider the equivalence relation on the set of all letter descriptions of T that is given by strict isomorphism. To each equivalence class we assign a letter C, using distinct letters for distinct classes. Each member L of the class is said to have *letter-type* C, and L is called an *occurrence* of the letter-type C in T. With $\mathcal{C}$ we denote the set of all letter-types of T.

If S is a solution of T, then all occurrences L of the letter-type $C \in \mathcal{C}$ receive the same image $\hat{S}(L)$ under S, by Proposition 6.6. This letter is denoted as $\hat{S}(C)$.

In the sequel, $\mathcal{V}_T$ denotes the set of context variables (i.e., types of context bases) occurring in T. We say that the letter-type C *occurs in* $X \in \mathcal{V}_T$ if C has an occurrence L that is a letter description of a context base $X^{(i)}$ of T of type X. Conversely we say that $X \in \mathcal{V}_T$ *occurs in the side area of* $C \in \mathcal{C}$ if there exists a context base $X^{(i)}$ of T of type X that is in the side area of some occurrence L of C.

The translation of T will be a pair $(\mathcal{W}_T, <)$ where $\mathcal{W}_T$ is a system of word equations and "$<$" is a linear constant restriction for $\mathcal{W}_T$. Here $\mathcal{W}_T$ is the result of a (deterministic) translation of context bases and letter descriptions of T into word equations, to be described below. The choice of the linear ordering "$<$" represents a non-deterministic step, details are given below.

Step 1: Translation of context bases and letter descriptions

To each context base cb of T, say, of type X, with field $(\eta_0, \ldots, \eta_k)$, we assign the word equation

$$X = Z_0, \ldots, Z_{k-1}$$

where Z_i is the unique (see above) context variable such that (η_i, η_{i+1}) is the field of a base of type Z_i, for $i = 0, \ldots, k-1$. To each letter description L of T, say, of type C, with field (η, η'), we assign the word equation

$$C = Z$$

where Z is the unique context variable such that (η, η') is the field of a base of type Z.

Let $\mathcal{W}_T$ denote the set of all word equations assigned to the context bases and letter descriptions of T in this way.

Step 2: Choice of linear constant restriction

Given $X \in \mathcal{V}_T$, a letter-type $C \in \mathcal{C}$, and nodes $\eta, \eta' \in N$ we define

$$\begin{aligned}
C <_1 X \quad &:\Leftrightarrow \quad C \text{ occurs in } X,\\
X <_1 C \quad &:\Leftrightarrow \quad X \text{ occurs in the side area of } C,\\
\eta <_1 X \quad &:\Leftrightarrow \quad \eta \text{ is in the side area of an occurrence of } X,\\
\eta <_1 C \quad &:\Leftrightarrow \quad \eta \text{ is in the side area of an occurrence of } C,\\
X <_1 \eta \quad &:\Leftrightarrow \quad X \text{ occurs in the subtree of } T \text{ with root } \eta,\\
C <_1 \eta \quad &:\Leftrightarrow \quad C \text{ occurs in the subtree of } T \text{ with root } \eta,\\
\eta <_1 \eta' \quad &:\Leftrightarrow \quad \eta \text{ is a descendant of } \eta'.
\end{aligned}$$

If "$<_1$" contains any cycle, we stop with failure. In the other case, let C_0 be a new symbol. Nondeterministically choose a linear ordering "$<$" on $\mathcal{V}_T \cup \mathcal{C} \cup \{C_0\}$ that extends the restriction of "$<_1$" to $\mathcal{V}_T \cup \mathcal{C}$ and has C_0 as its minimal element.

Output

The set of *output problems* of the translation procedure is the set of all pairs of the form $(\mathcal{W}_T, <)$ described above. In each case $\mathcal{W}_T$ represents a system of word equations over the alphabet of variables $\mathcal{V}_T$ and the set of constants

$\mathcal{C} \cup \{C_0\}$, and "$<$" represents a linear constant restriction for $\mathcal{W}_T$ [3]. In more detail, we demand that a solution of $(\mathcal{W}_T, <)$ does not instantiate any variable $X \in \mathcal{V}_T$ with the empty word.

Completeness and soundness of the translation

We first show completeness.

Lemma 7.1 *If T has a rigid solution, then there exists an output problem $(\mathcal{W}_T, <)$ that has a solution.*

Proof. Let S be a rigid solution of T. Recall that S assigns the same letter $\hat{S}(C)$ to all occurrences of a given letter-type C in T, while letters $\hat{S}(C_1)$ and $\hat{S}(C_2)$ are distinct for $C_1 \neq C_2$. Similarly $\hat{S}$ assigns a unique ground context (resp. ground term) to each base (resp. node) of T. For $X \in \mathcal{V}_T$, $C \in \mathcal{C}$ and nodes η, η' of T define

$$\begin{array}{rcl}
C \prec_1 X & :\Leftrightarrow & \hat{S}(C) \text{ is a letter of } \hat{S}(X), \\
X \prec_1 C & :\Leftrightarrow & \hat{S}(X) \text{ is a proper subcontext of } \hat{S}(C), \\
\eta \prec_1 X & :\Leftrightarrow & \hat{S}(X) \text{ has a subterm } \hat{S}(\eta), \\
\eta \prec_1 C & :\Leftrightarrow & \hat{S}(C) \text{ has a subterm } \hat{S}(\eta), \\
X \prec_1 \eta & :\Leftrightarrow & \hat{S}(\eta) \text{ has a subcontext } \hat{S}(X), \\
C \prec_1 \eta & :\Leftrightarrow & \hat{S}(\eta) \text{ has a subcontext } \hat{S}(C), \\
\eta \prec_1 \eta' & :\Leftrightarrow & \hat{S}(\eta) \text{ is a proper subterm of } \hat{S}(\eta').
\end{array}$$

Consider one of these relations $\kappa_1 \prec_1 \kappa_2$ defined above. For $i = 1, 2$, let $\#_\Sigma \hat{S}(\kappa_1)$ denote the number of nodes of $\hat{S}(\kappa_i)$ that are labeled with a symbol in Σ. If $\kappa_1 = C \in \mathcal{C}$ and $\kappa_2 = X \in \mathcal{V}_T$, then $\#_\Sigma \hat{S}(\kappa_1) \leq \#_\Sigma \hat{S}(\kappa_2)$. In all other cases it is easy to see that $\#_\Sigma \hat{S}(\kappa_1) < \#_\Sigma \hat{S}(\kappa_2)$. It follows that $\prec_1$ does not have any cycle. It is also clear that "$\prec_1$" extends the relation $<_1$ defined in Step 2 above. Hence there exists a linear ordering "$<$" on

[3] A *word equation* over the alphabet of constants $\mathcal{C}$ and the alphabet of variables $\mathcal{V}$ is an expression of the form $W_1 = W_2$ where W_1 and W_2 are words over the joint alphabet $\mathcal{C} \cup \mathcal{V}$. Let S be a mapping that assigns a word $S(X)$ to each variable X in the equation. S is a *solution* of $W_1 = W_2$ if both sides of the equation become identical when we replace each occurrence of a variable X by the word $S(X)$. A *linear constant restriction* is given by a linear ordering "$<$" on the set of constants and variables occurring in the equation. The solution S of $W_1 = W_2$ respects the linear constant restriction "$<$" if $X < c$ implies that c does not occur in $S(X)$, for each variable X and each constant c occurring in $W_1 = W_2$.

$\mathcal{V}_T \cup \mathcal{C} \cup \{C_0\}$ that extends "$\prec_1$" and represents a possible choice in Step 2. We consider the output problem $(\mathcal{W}_T, <)$.

For $X \in \mathcal{V}_T$, let $s_{X,1}, \ldots, s_{X,n_X}$ denote the sequence of letters of the ground context $\hat{S}(X)$. We replace each letter $s_{X,i}$ of the form $\hat{S}(C)$ for some $C \in \mathcal{C}$ by the constant C, and each of the remaining letters by C_0. By rigidness of S, the replacement instance of each letter is welldefined. Let $S'(X)$ be the resulting word in the alphabet $\mathcal{C} \cup \{C_0\}$.

To see that S' is a solution of $\mathcal{W}_T$ we first consider a word equation of $\mathcal{W}_T$ of the form $X = Z_0, \ldots, Z_{k-1}$. Since S is a solution of T we have

$$\hat{S}(X) = \hat{S}(Z_0) \ldots \hat{S}(Z_{k-1}).$$

It follows that

$$S'(X) = S'(Z_0) \ldots \mathcal{S}'(Z_{k-1})$$

which shows that S' solves the above equation. Consider now a word equation of $\mathcal{W}_T$ of the form $C = Z$. Since S is a solution of T we have

$$\hat{S}(C) = \hat{S}(Z).$$

It follows that

$$C = S'(Z)$$

which shows that S' solves the above equation. Summing up, we have seen that S' solves $\mathcal{W}_T$.

To check validity of the linear constant restriction, assume that the letter $C \in \mathcal{C}$ occurs in $S'(X)$. Then, by definition of S', $\hat{S}(C)$ is a letter of $\hat{S}(X)$ and we have $C <_1 X$ by our choice in Step 2. This shows that S' satisfies the linear constant restriction imposed by "$<$". $\square$

We may now show soundness.

Lemma 7.2 *If an output problem $(\mathcal{W}, <)$ has a solution, then T has a solution.*

Proof. Assume that $(\mathcal{W}, <)$ has a solution S. It follows from the failure condition of Step 2 that there exists a linear ordering "$<_2$" on $\mathcal{C} \cup \mathcal{V}_T \cup N \cup \{C_0\}$ that extends "$<$" and has C_0 as minimal element. Let t_0 be an arbitrary letter, and let b be a fixed constant in Σ. We shall now construct, by simultaneous induction on "$<_2$",

- a mapping S_1 that assigns a ground term $S_1(\eta)$ to each node $\eta \in N$,

- a mapping S_2 that assigns letters (resp. ground contexts) to the elements of $\mathcal{C} \cup \{C_0\}$ (resp. $\mathcal{V}_T$).

The idea behind the definition of S_1 is to use the terms assigned to the children of a node η for constructing $S_1(\eta)$. As a matter of fact, if η_i is a child of η, then $S_1(\eta)$ may have various subterms of the form $S_1(\eta_i)$, but just exactly one of these occurrences has its origin in the use of $S_1(\eta_i)$ in the construction of $S_1(\eta)$. In order to distinguish this occurrence notationally, we denote it in the form $t^{(\eta_i)}$.

Let $\eta \in N$ be a leaf. If η is labeled with $a \in \Sigma$, then $S_1(\eta) := a$. If η is the node of an individual base, then we define $S_1(\eta) := b$. In addition, let $S_2(C_0) := t_0$. Now assume that S_1 and S_2 have been defined, up to a certain element of the linear ordering "$<_2$". We consider the first element κ of "$<_2$" where S_1 or S_2 respectively is yet not defined.

1. if $\kappa = \eta$ is an inner node that is labeled with the n-ary function symbol f, and if $\eta_1, \ldots, \eta_n$ are its children, then we define $S_1(\eta) := f(S_1(\eta_1), \ldots, S_1(\eta_n))$. Using the notational convention explained above, this term will be written in the form $t^{(\eta)} = f(t^{(\eta_1)}, \ldots, t^{(\eta_n)})$.

2. if $\kappa = \eta$ is an unlabeled inner node, then η is a non-final node of a base (cf. Def. 3.1, 6.), and η has exactly one child η' (cf. Lemma 5.2). Moreover, (η, η') is the field of a unique base cb, say, of type Z. Since $Z <_1 \eta$ we have also $Z <_2 \eta$ and we may define $t^{\eta} = S_1(\eta) := S_2(Z)(S_1(\eta')) = S_2(Z)(t^{\eta'})$.

3. if $\kappa = X$ is a context variable, let $S(X) = C_{X,1} \cdots C_{X,n_X}$. Since S respects the linear constant restriction imposed by "$<$" the constants $C_{X,i} \in \mathcal{C} \cup \{C_0\}$ are smaller than X with respect to "$<_2$". Accordingly, for each constant $C_{X,i}$, the letter $S_2(C_{X,i})$ has been defined by induction hypothesis. We define $S_2(X) := S_2(C_{X,1}) \cdots S_2(C_{X,n})$.

4. if $\kappa = C$ is a letter-type, let $L = (\eta, \eta')$ denote an occurrence of C in T. Let f (resp. i) be the label (resp. direction) of L, let $(\eta_1, \ldots, \eta_n)$ be the sequence of all children of η. By assumption, $S_1(\eta_j)$ has been defined for $1 \leq j \leq n$. We define $S_2(C) := f(S_1(\eta_1), \ldots, S_1(\eta_{i-1}), \Omega, S_1(\eta_{i+1}), \ldots, S_1(\eta_n))$.

Now let $\eta_\top$ denote the root of T. We want to show that $(t^{(\eta_\top)}, S')$ is a solution of T, where for all $\eta \in N$ the node $S'(\eta)$ is given by the position of the root of $t^{(\eta)}$.

Since S assigns a non-empty word to each variable $X \in \mathcal{V}_T$ a trivial induction shows that $S_2(X)$ is always a non-empty ground context. From this it follows immediately that S' is an injective mapping from N to the set of nodes of $t^{(\eta_T)}$. By construction S' preserves root, Σ-labels, children of Σ-labeled nodes, and preorder relations.

In order to show that S' respects branching points it suffices to show that whenever η_1 and η_2 are two distinct children of a node η of T, then $S'(\eta_1)$ and $S'(\eta_2)$ are distinct children of $S'(\eta)$ in $t^{(\eta_\top)}$. Since unlabeled nodes of T never have two distinct children this follows from Step 1 above.

We now show that $\hat{S}'$ assigns the same ground context to the fields of bases of the same type X, for all $X \in \mathcal{V}_T$. This follows directly from the following two claims, which will be proven by simultaneous induction on "$<_2$".

C1. if (η, η') is an occurrence of $C \in \mathcal{C}$ in T, then $\hat{S}'(\eta, \eta') = S_2(C)$,

C2. if $X^{(i)}$ is an occurrence of $X \in \mathcal{V}_T$ in T, then $\hat{S}'(X^{(i)}) = S_2(X)$.

Assume that Claims 1 and 2 have been shown for all predecessors of $\kappa \in \mathcal{C} \cup \mathcal{V}_T$ with respect to "$<_2$".

First assume that $\kappa = C \in \mathcal{C}$. It follows from the definition of $S_2(C)$ in Step 4 above that the occurrence (η, η') of C that has been used in this step for defining $S_2(C)$ satisfies Condition C1. By induction hypothesis for C2 we know that all occurrences of the same context variable X in the side area of an occurrence of C — which all are smaller than C with respect to $<_2$ — are mapped to the same ground context $S_2(X)$ under $\hat{S}'$. Since all occurrences of C in T are strictly isomorphic it follows easily that all other occurrences of C in T are mapped to the same letter under $\hat{S}'$, which proves Condition C1 for C.

Now assume that $\kappa = X \in \mathcal{V}_T$. Let $X^{(s)}$ be an occurrence of $X \in \mathcal{V}_T$ in T, and let $\mathit{Field}(X^{(s)}) = (\eta_i, \ldots, \eta_j)$. In each atomic subfield (η_l, η_{l+1}) of $(\eta_i, \ldots, \eta_j)$, node η_l is either labeled or unlabeled. For simplicity we assume that $(\eta_i, \ldots, \eta_j)$ has the form $(\eta_i, \eta_{i+1}, \eta_{i+2})$ where η_i is labeled and η_{i+1} is unlabeled.

The field (η_i, η_{i+1}) represents a letter description of X, say, of type C. Note that $C <_1 X$ and hence $C <_2 X$. Let $Z^{(r)}$ denote the unique base of T with field (η_{i+1}, η_{i+2}). It follows from the definition of $\mathcal{W}_T$ that S solves the equation $X = CZ$. Hence from the definition of S_2 (Case 3 above) we see that

$$S_2(X) = S_2(C)S_2(Z).$$

Moreover, since $C <_2 X$ we know by induction hypothesis C1 for C that $\hat{S}'(\eta_i, \eta_{i+1}) = S_2(C)$. In addition it follows from Case 2 of the definition of S_1 that $\hat{S}'(Z^{(r)}) = S_2(Z)$.

It follows now that $\hat{S}'(\eta_i, \eta_{i+1}, \eta_{i+2})$, which is the composition of $\hat{S}'(\eta_i, \eta_{i+1})$ and $\hat{S}'(\eta_{i+1}, \eta_{i+2})$, has the form $S_2(C)S_2(Z) = S_2(X)$, which proves that $\hat{S}'(X^{(i)}) = S_2(X)$.

By our special assumption (†) on individual variables it follows from Lemma 6.4 that $\hat{S}'$ assigns the same ground term to nodes η_1 and η_2 whenever there are two individual bases of the same type with nodes η_1 and η_2 respectively. □

8 Summing up

We are now able to prove the Main Theorem (Theorem 1.1).

Theorem 8.1 *It is decidable if a finite system of context equations with two context variables and an arbitrary number of individual variables has a solution.*

Proof. We first treat the case where we just have one input equation. Let $s = t$ be a context equation with two context variables. Combining the results of Lemma 4.6, Lemma 5.5, Lemma 6.10, Theorem 6.11, Lemma 7.1, and Lemma 7.2 it follows that we may effectively compute a finite set M of multi-word equations with linear constant restriction such that $s = t$ has a solution if and only if a multi-word equation with linear constant restriction in M has a solution. The results in [25] on regular solutions of word equations show that solvability of multi-word equations with linear constant restriction is decidable. Hence the result follows.

Let us now consider the case where we have a finite system of context equations, $\{s_1 = t_1, \ldots, s_n = t_n\}$ as input. We show how to reduce it to the first situation. We may assume that there is at least one function symbol "f" of arity $n > 1$ in the signature Σ (since otherwise we are in the monadic case where context unification problems directly translate into word equations). For simplicity we assume that "f" has arity 2 (if the arity is greater than 2 we may use essentially the same encoding where other arguments of f are filled with a fixed constant $a \in \Sigma$). Obviously, $\{s_1 = t_1, \ldots, s_n = t_n\}$ has a solution if and only if the context equation $f(s_1, f(s_2, f(\ldots))) = f(t_1, f(t_2, f(\ldots)))$ has a solution. □

References

[1] H. Abdulrab and J.-P. Pecuchet. Solving word equations. *J. Symbolic Computation*, 8:499–521, 1989.

[2] F. Baader and K. U. Schulz. Unification in the union of disjoint equational theories: Combining decision procedures. *J. of Symbolic Computation*, 21:211–243, 1996.

[3] H. Comon. Completion of rewrite systems with membership constraints. In *Int. Coll. on Automata, Languages and Programming, ICALP'92*, volume 623 of *LNCS*. Springer-Verlag, 1992.

[4] H. Comon. Completion of rewrite systems with membership constraints, part I: Deduction rules and part II: Constraint solving. Technical Report, CNRS and LRI, Université de Paris Sud, 1993. to appear in JSC.

[5] V. Diekert. Makanin's Algorithm. In M. Lothaire, editor, *Algebraic Combinatorics on Words*. 1998. preliminary version, Fakultät für Informatik, University of Stuttgart.

[6] W. Farmer. Simple second-order languages for which unification is undecidable. *Theoretical Computer Science*, 87:173–214, 1991.

[7] H. Ganzinger, F. Jacquemard, and M. Veanes. Rigid reachability. Draft, Max-Planck-Institut für Informatik, Germany, 1998.

[8] W. Goldfarb. The undecidability of the second-order unification problem. *Theoretical Computer Science*, 13:225–230, 1981.

[9] Y. Gurevich and A. Voronkov. Monadic simultaneous rigid E-unification and related problems. UPMAIL Technical Report 137, Uppsala University, Computing Science Department, Sweden, 1997.

[10] J. I. Hmelevskii. Equations in free semigroups. *Trudy Mat. Inst. Steklov*, 107, 1971. English translation in: Proceedings of the Steklov Institute of Mathematics, 107, Am. Math. Soc., 1976.

[11] J. Jaffar. Minimal and complete word unification. *J. of the ACM*, 37(1):47–85, 1990.

[12] A. Kościelski and L. Pacholski. Complexity of Makanin's algorithms. *Journal of the Association for Computing Machinery*, 43:670–684, 1996.

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

[14] J. Levy and M. Veanes. On the undecidability of second-order unification. *Submitted to Information and Computation*, 1999.

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

[16] J. Marcinkowski. Undecidability of the first order theory of one-step right ground rewriting. In H. Comon, editor, *International Conference on Rewriting Techniques and Applications*, Lecture Notes in Computer Science. Springer-Verlag, June 1997.

[17] Y. Matiyasevich. A connection between systems of word equations and length equations and Hilbert's tenth problem. *Sem. MAt. V. A. Steklov Math. Inst. Leningrad*, 8:132–144, 1968. Englisc translation in: Seminars in Mathematics, V. A. Steklov Mathematical Institute 8, 1970.

[18] J. Niehren, M. Pinkal, and P. Ruhrberg. On equality up-to constraints over finite trees, context unification, and one-step rewriting. In *Proc. of the Int. Conf. on Automated Deduction*, volume 1249 of *LNCS*, pages 34–48, 1997.

[19] J. Niehren, M. Pinkal, and P. Ruhrberg. A uniform approach to underspecification and parallelism. Technical Report, 1997.

[20] J. Niehren, S. Tison, and R. Treinen. On stratified context unification and rewriting constraints. Talk at CCL'98 Workshop, 1998.

[21] M. Schmidt-Schauß. Unification of stratified second-order terms. Internal Report 12/94, Fachbereich Informatik, J.W. Goethe-Universität Frankfurt, Germany, 1994.

[22] M. Schmidt-Schauß. An algorithm for distributive unification. *Theoretical Computer Science*, 208:111–148, 1998.

[23] M. Schmidt-Schauß. Decidability of bounded second order unification. Draft, Fachbereich Informatik, J.W. Goethe-Universität Frankfurt, Germany, 1998.

[24] M. Schmidt-Schauß and K. U. Schulz. On the exponent of periodicity of minimal solutions of context equations. In *Rewriting Techniques and Applications, Proc. RTA'98*, volume 1379 of *LNCS*, pages 61–75. Springer-Verlag, 1998.

[25] K. U. Schulz. Makanin's algorithm - two improvements and a generalization. In *Proc. of IWWERT 1990*, volume 572 of *LNCS*, pages 85–150. Springer-Verlag, 1990.

[26] K. U. Schulz. Word unification and transformation of generalized equations. *J. Automated Reasoning*, pages 149–184, 1993.

[27] R. Treinen. The first-order theory of one-step rewriting is undecidable. In H. Ganzinger, editor, *7th International Conference on Rewriting Techniques and Applications*, number 1103 in Lecture Notes in Computer Science, pages 276–286, Rutgers University, NJ, USA, July 1996. Springer-Verlag.

[28] S. Vorobyov. The first-order theory of one step rewriting in linear noetherian systems is undecidable. In H. Comon, editor, *International Conference on Rewriting Techniques and Applications*, Lecture Notes in Computer Science. Springer-Verlag, June 1997.

[29] S. Vorobyov. The $\forall\exists^8$-equational theory of context unification is co-recursively enumerable hard. Talk at CCL'98 Workshop, 1998.

9 Appendix

It remains to give the proofs for Lemma 6.10 and Theorem 6.11.

Lemma 9.1 *Let $z \in X$ be an individual variable. Assume that T_{i+1} has an z-node, but T_i doesn't. Then each z-node of T_{i+1} is a top side node of T_{i+1}. Moreover, if η and η' are two z-nodes of T_{i+1}, then π_{i+1} contains nodes η_1 and η'_1 such that $T_{i+1}^{\eta_1}$ and $T_{i+1}^{\eta'_1}$ are strictly isomorphic and have corresponding and strictly isomorphic letter descriptions L and L' where η_1 and η'_1 are corresponding top side nodes of L and L'.*

Proof. It is easy to see that an individual base of new type z can only arise from Part 3 of the subprocedure of Case III where we apply Step 4 of (Transl2). The nodes of the individual bases *ib* that are introduced at this step are the children of the labeled branching point of two context bases, which means that they are top side nodes of the superposition T^{S_1}. All the new individual bases receive a new type. By construction, all the variants T_j^S of T^S that are created in Step 5 of Case III are strictly isomorphic problems. The lemma follows easily. □

Lemma 9.2 *Let $\eta \in \bigcup \Pi_i$ and $\eta' \in \Delta_i$. Then η is not a descendant of η'.*

Proof. We use induction. Assume that the statement holds for Π_i and Δ_i. Then Condition (b) for the choice of π_{i+1} ensures that the condition holds for Π_{i+1} and Δ_{i+1} as well since in the Steps 1-5 new relevant nodes are only created below the nodes of π_{i+1}. □

Lemma 9.3 *Each $\pi \in \Pi_i$ is $\mathcal{X}$-closed.*

Proof. This holds trivially for $i = 0$ where Π_0 is empty. Assume that the statement is correct for T_i and Π_i. First let $\eta \in \pi_{i+1}$ be an x-node of T_{i+1}. Since in Step 3 of Case III always new individual variables are used it follows easily that in this case π_{i+1} also contains an x-node η' of the predecessor problem T_i. Hence π_{i+1} contains all x-nodes of T_i, by Condition (c). When building T_{i+1}, new x-nodes always belong to π_{i+1} (cf. Step 5). It follows that π_{i+1} contains all x-nodes of T_{i+1}.

Now let $\eta \in \pi \in \Pi_i$ be an x-node of T_{i+1}. When building T_{i+1}, new z-nodes (for some $z \in \mathcal{X}$) are only created below—or at—the nodes of π_{i+1}. Assume that η is not an x-node of T_i. Since $\eta \notin \pi_{i+1}$ this would mean that η

is a descendant of an element η' of π_{i+1}. This is impossible, by Lemma 9.2. We have seen that η is an x-node of T_i. By induction hypothesis, π contains all x-nodes of T_i. Since, as we saw, no new x-nodes are created when building T_{i+1} it follows that π contains all x-nodes of T_{i+1}. $\square$

We may now give the proof of Lemma 6.10:

Lemma 9.4 *The procedure (Transl3) terminates.*

Proof. We consider the measure μ on $T_0, T_1, \ldots$ that is given by the lexicographic order with the following components:

(1) The number of individual variables $x \in \mathcal{X}_0$ such that Δ_i has an x-node,

(2) the number of top side nodes η of Δ_i where T_i^η is not first order,

(3) the total number of top side nodes in Δ_i.

Clearly this measure is well-founded. We show that each call to Case III, where we move from T_i to T_{i+1}, reduces the measure.

Case 1. If π_{i+1} contains an x-node for some $x \in \mathcal{X}_0$, then, by Lemma 9.3, π_{i+1} contains all x-nodes of T_{i+1}. Hence, by Lemma 9.3, the first component of μ decreases.

Case 2. If π_{i+1} does not contain any x-node of T_i for $x \in \mathcal{X}_0$ we apply the Failure Condition of Step 2. The condition shows that in this case the superposition T^{S_0} does not have any branching bases.

First consider the subcase where for some $\eta_j \in \pi_{i+1}$ the subproblem $T_i^{\eta_j}$ is not first order. Consider any new top side node $\eta \in \mathit{tsn}(T_{i+1}) \setminus \mathit{tsn}(T_i)$. Clearly η is a descendant of a node $\eta' \in \pi_{i+1}$ and there exists a context base cb with $|\mathit{Field}_{i+1}(cb) \cap T_{i+1}^{\eta'}| \geq 2$ such that η is in the side area of cb. But now the Failure Condition implies that the subproblem T_{i+1}^{η} is first order. Hence we decreased the second component of μ while leaving the first component unchanged.

Now assume that all the subproblems $T_i^{\eta_j}$ for $\eta_j \in \pi_{i+1}$ are first order. Obviously the superposition T^S does not have any top side node in this case. Hence the total number of top side nodes in Δ_i decreases and the first two components of μ are left unchanged. $\square$

Definition 9.5 Let T be a transparent generalized context problem and let X be a context variable where T has a base of type X. The set of context bases of *suffix type* X of T is recursively defined as follows:

1. each context base of type X is of suffix type X,
2. if cb_1 and cb_2 are context bases of T of the same type, if the context base cb_3 of T has suffix type X and if $Field(cb_2)$ is a suffix of $Field(cb_3)$, then cb_1 has suffix type X.

Obviously, if (t, S) is a solution of T and cb has suffix type X, then $\hat{S}(cb)$ is a suffix of $\hat{S}(X)$.

The following property characterizes the context bases that are generated in (Transl3), as we shall see below.

Definition 9.6 Let X and Y be two context variables. T is called X-Y-*binary* iff every context base cb of T is a (proper or improper) subbase of a context base of suffix type X or Y.

Lemma 9.7 *Each of the problems T_i generated by (Transl3) and each output problem T' is X-Y-binary.*

Proof. Obviously T_0 is X-Y-binary. As an induction hypothesis, assume that T_i is X-Y-binary. Let cb be a context base of T_{i+1}. It may be

(1) a base of T_i,
(2) a copy of a base of T_i (cf. Step 5 of Case III),
(3) a placeholder base of T_i (construction of $T_i^{\eta_j}$),
(4) a copy of a placeholder base of T_i (cf. Step 5 of Case III),
(5) a subbase $Z^{(h)}$ of a base of type 1–4 (cf. Steps 4, 5 of Case III),

1. If cb has type (1), then by induction hypothesis cb is a subbase of a base of suffix type X or Y of T_i. Clearly the same property holds in T_{i+1} as well.
2. If $cb = cb_j$ has type (2) and is the copy of the base cb_k of type (1), let $T_{i+1}^{\eta_j}$ and $T_{i+1}^{\eta_k}$ denote the variants of T^S that contain these bases. By 1, cb_k is a subbase of a base of suffix type X or Y of T_{i+1}. Since $T_{i+1}^{\eta_j}$ and $T_{i+1}^{\eta_k}$ are strictly isomorphic the same holds for cb_j as well.
3. If cb has type (3), assume that cb represent the suffix of the base cb' of T_i. By 1, cb' is a subbase of a base of suffix type X or Y of T_{i+1}. Since cb

is in T_{i+1} a subbase of cb' also cb is a subbase of a base of suffix type X or Y of T_{i+1}.
4. If $cb = cb_j$ has type (4) and is the copy of the placeholder base cb_k of type (3), let $T_{i+1}^{\eta_j}$ and $T_{i+1}^{\eta_k}$ denote the variants of T^S that contain these bases. By 3, cb_k is a subbase of a base of suffix type X or Y of T_{i+1}. Since $T_{i+1}^{\eta_j}$ and $T_{i+1}^{\eta_k}$ are strictly isomorphic the same holds for cb_j as well.
5. If cb has type (5), then it is a subbase of a base of T_{i+1} of one of the types (1)-(4). It follows from the previous cases that cb is a subbase of a base of suffix type X or Y of T_{i+1}.

We have seen that each of the problems T_i generated by (Transl3) is X-Y-binary. Obviously this implies that each output problem T' is X-Y-binary. $\square$

Definition 9.8 Let T be an X-Y-binary transparent generalized context problem. A set Δ of top side nodes of T is called *X-Y-stable* iff for each $\eta \in \Delta$ and each context base cb of T such that $|Field(cb) \cap T^\eta| \geq 2$ either

(1) $Field(cb) \subseteq T^\eta$ and cb has type X or Y, or
(2) $Field(cb) \not\subseteq T^\eta$ and cb has suffix type X or Y, or
(3) cb is a subbase of a context base of type (1) or (2).

Lemma 9.9 *For each of the problems T_i generated by (Transl3) the set Δ_i is X-Y-stable.*

Proof. Obviously Δ_0 is X-Y-stable. As an induction hypothesis, assume that Δ_i is X-Y-stable w.r.t. T_i. Let $\eta \in \Delta_{i+1}$ and let cb be a context base of T_{i+1} such that $|Field_{i+1}(cb) \cap T_{i+1}^\eta| \geq 2$.

First we assume that η is not a descendant of any node in π_{i+1}. Lemma 9.2 implies that η does not have a descendant in π_{i+1}. Both properties together imply that cb is a base of T_i with $Field_i(cb) = Field_{i+1}(cb)$. In addition it follows that $\eta \in \Delta_i$. By induction hypothesis, either

(1) $Field_i(cb) \subseteq T_i^\eta$ and cb has type X or Y in T_i, or
(2) $Field_i(cb) \not\subseteq T_i^\eta$ and cb has suffix type X or Y in T_i, or
(3) cb is a subbase of a context base of type (1) or (2) of T_i.

It follows that cb has in T_{i+1} the corresponding property as well.

Now let η be a descendant of $\eta_j \in \pi_{i+1}$ in $T_{i+1}^{\eta_j}$. As in the previous proof we distinguish the cases where cb is

(1) a base of T_i,
(2) a copy of a base of T_i (cf. Step 5 of Case III),
(3) a placeholder base of T_i (construction of $T_i^{\eta_j}$),
(4) a copy of a placeholder base of T_i (cf. Step 5 of Case III),
(5) a subbase $Z^{(h)}$ of a base of type 1–4 (cf. Steps 4, 5 of Case III),

1. If cb has type (1), then $|\mathit{Field}_i(cb) \cap T_i^{\eta_j}| \geq 2$. Since $\eta_j \in \Delta_i$, by induction hypothesis, either

(1) $\mathit{Field}_i(cb) \subseteq T_i^{\eta_j}$ and cb has type X or Y in T_i, or
(2) $\mathit{Field}_i(cb) \not\subseteq T_i^{\eta_j}$ and cb has suffix type X or Y in T_i, or
(3) cb is a subbase of a context base of type (1) or (2) of T_i.

But then, since η is a descendant of η_j in T_{i+1} it is easy to see that either

(1') $\mathit{Field}_{i+1}(cb) \subseteq T_{i+1}^{\eta}$ and cb has type X or Y in T_{i+1}, or
(2') $\mathit{Field}_{i+1}(cb) \not\subseteq T_{i+1}^{\eta}$ and cb has suffix type X or Y in T_{i+1},
(3') or cb is a subbase of a context base of type (1) or (2) of T_{i+1}.

2. If $cb = cb_j$ has type (2) and is the copy of the base cb' of type (1), let $T_{i+1}^{\eta_k}$ denote the variant of T^S that contains cb', and let η' denote the node of $T_{i+1}^{\eta_k}$ that corresponds to η. Since $T_{i+1}^{\eta_k}$ and $T_{i+1}^{\eta_j}$ are strictly isomorphic we know that $|\mathit{Field}_{i+1}(cb') \cap T_{i+1}^{\eta'}| \geq 2$. It follows from the previous case that either

(1') $\mathit{Field}_{i+1}(cb') \subseteq T_{i+1}^{\eta'}$ and cb' has type X or Y in T_{i+1}, or
(2') $\mathit{Field}_{i+1}(cb') \not\subseteq T_{i+1}^{\eta'}$ and cb' has suffix type X or Y in T_{i+1},
(3') or cb' is a subbase of a context base of type (1) or (2) of T_{i+1}.

Because of the strict isomorphism between $T_{i+1}^{\eta_k}$ and $T_{i+1}^{\eta_j}$ the same holds for cb and η as well.
3. If cb has type (3) and represents the suffix of the context base cb', then cb' has type (1). It follows from Case 1 that cb' has one of the possible types in T_{i+1}. Hence the suffix cb, which starts at the predecessor η_j of η, has one of the possible types, too.

4. If $cb = cb_j$ has type (4) and is the copy of the placeholder base cb' of type (3), let $T_{i+1}^{\eta_k}$ denote the variant of T^S that contains cb', and let η' denote the node of $T_{i+1}^{\eta_k}$ that corresponds to η. Since $T_{i+1}^{\eta_k}$ and $T_{i+1}^{\eta_j}$ are strictly isomorphic we know that $|Field_{i+1}(cb') \cap T_{i+1}^{\eta'}| \geq 2$. It follows from the previous case that either

(1') $Field_{i+1}(cb') \subseteq T_{i+1}^{\eta'}$ and cb' has type X or Y in T_{i+1}, or

(2') $Field_{i+1}(cb') \not\subseteq T_{i+1}^{\eta'}$ and cb' has suffix type X or Y in T_{i+1},

(3') or cb' is a subbase of a context base of type (1) or (2) of T_{i+1}.

Because of the strict isomorphism between $T_{i+1}^{\eta_k}$ and $T_{i+1}^{\eta_j}$ the same holds for cb and η as well.
5. If cb has type (5), then it is a subbase of a base cb' of the form treated in the previous cases. We have seen that cb' has one of the three possible types. It follows that its subbase cb has one of the three possible types. □

The following lemmas are needed for justifying the use of Condition 2 (Failure Condition) in Case III of (Transl3). First, some criteria for the unsolvability of a generalized context problem are given that generalize the occur-check in first-order syntactic unification.

Lemma 9.10 *Let s_1 and s_2 be two suffixes of the same ground context s, and let $s_1^{(1)}$ and $s_2^{(2)}$ be occurrences of s_1 and s_2 respectively in the ground term t. If $s_1^{(1)}$ is completely contained in the side area of $s_2^{(2)}$, then s_1 is a proper suffix of s_2.*

Proof. Otherwise s_2 would be a suffix of s_1. This would mean that the suffix $s_2^{(2)}$ of s_1 properly contains an occurrence $s_1^{(1)}$ of s_1, which obviously is impossible. □

Corollary 9.11 *Let s_1 and s_2 be two suffixes of the ground context s, and let $s_1^{(1)}$ and $s_2^{(2)}$ be occurrences of s_1 and s_2 respectively in the ground term t. Then the main paths of $s_1^{(1)}$ and $s_2^{(2)}$ cannot represent branching fields.*

Proof. Assume that the main paths of $s_1^{(1)}$ and $s_2^{(2)}$ are branching at point η. Let $r_1^{(1)}$ and $r_2^{(2)}$ denote the suffixes of $s_1^{(1)}$ and $s_2^{(2)}$ with root η, and let $t_1^{(1)}$ and $t_2^{(2)}$ denote the suffixes of $r_1^{(1)}$ and $r_2^{(2)}$ that start at the two children of η on the main paths of these contexts. By the previous lemma,

$t_1^{(1)}$ is a proper suffix of $r_2^{(2)}$ and $t_2^{(2)}$ is a proper suffix of $r_1^{(1)}$. This implies that $r_1 = r_2$ and $r_1^{(1)} = r_2^{(2)}$, a contradiction. $\square$

Definition 9.12 Let T be a transparent generalized context problem. Let (t, S) be a solution of T. With "$\equiv_S$" we denote the equivalence relation on $ld(T)$ defined by $L \equiv_S L'$ iff $\hat{S}(L) = \hat{S}(L)$. With $[L]_S$ we denote the equivalence class of $L \in ld(T)$ with respect to "$\equiv_S$". For $\eta, \eta' \in tsn(T)$ we define $\eta \sim_S^1 \eta'$ iff there exist letter descriptions $L \equiv_S L'$ and an index i such that η (resp. η') represents the i-th top side node of L (resp. L'). The equivalence relation "$\sim_S$" generated by "$\sim_S^1$" is called the *equivalence relation on $tsn(T)$ induced by S*. With $[\eta]_S$ we denote the equivalence class of $\eta \in tsn(T)$ with respect to "$\sim_S$".

Remark 9.13 In the situation of the previous definition we have $\eta \sim_S \eta'$ iff there exists a sequence of pairs $(L_1, \eta_1), \ldots, (L_n, \eta_n)$ of letter descriptions L_j and top side nodes η_j of L_j in T, with $\eta = \eta_1$ and $\eta' = \eta_n$, such that for all consecutive pairs $(L_j, \eta_j), (L_{j+1}, \eta_{j+1})$ $(1 \leq j \leq n-1)$ either

(a) $L_j \equiv_S L_{j+1}$ and η_j and η_{j+1} are corresponding top side nodes of L_j and L_{j+1}, or

(b) $\eta_j = \eta_{j+1}$ and $L_j \neq L_{j+1}$ belong to branching context bases where η_j is a child of the branching point.

Definition 9.14 Let T be an X-Y-binary transparent generalized context problem. A context base cb of T is *X-Y-normal* for the top side node η of T iff $|Field(cb) \cap T^\eta| \geq 2$ and if either

(1) $Field(cb) \subseteq T^\eta$ and cb has type X or Y, or

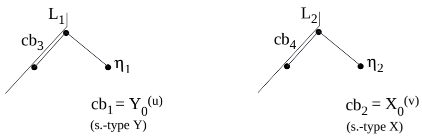
(2) $Field(cb) \not\subseteq T^\eta$ and cb has suffix type X or Y.

Lemma 9.15 *Let T be a generalized context problem with solution (t, S) that is X-Y-binary. Let η be a top side node of the letter description L for the base cb of suffix type X. If the context base cb' is X-Y-normal for η, then cb' has suffix type Y.*

Proof. Assume that cb' has suffix type X. If $Field(cb') \subseteq T^\eta$, then cb' has type X. Then Lemma 9.10 yields a contradiction. In the other case, cb and cb' are branching bases and Corollary 9.11 yields a contradiction. $\square$

Lemma 9.16 *Let T be an X-Y-binary generalized context problem with solution (t, S). Let $L_1 \equiv_S L_2$ be two letter descriptions of T, let η_1 and η_2 be corresponding top side nodes of L_1 and L_2. If the context bases cb_1 and cb_2 of T are X-Y-normal for η_1 and η_2 respectively, then either both cb_1 and cb_2 have suffix type X or both have suffix type Y.*

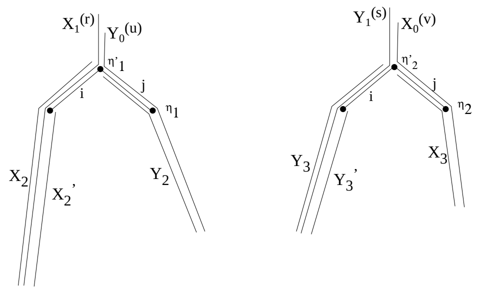
Proof. Assume, to get a contradiction, that $cb_1 = Y_0^{(u)}$ has suffix type Y and $cb_2 = X_0^{(v)}$ has type suffix X. Let L_1 (resp. L_2) be a letter description of the base cb_3 (resp. cb_4). Since T is X-Y-binary we may assume that cb_3 and cb_4 have suffix type X or Y.



Lemma 9.15 shows that cb_3 has suffix type X and cb_4 has suffix type Y. Let $cb_3 = X_1^{(r)}$ and $cb_4 = Y_1^{(s)}$. We claim that the field of $Y_0^{(u)}$ cannot be completely contained in T^{η_1}. Otherwise, since $Y_0^{(u)}$ is X-Y-normal for η_1, $Y_0^{(u)}$ would have type Y and $\hat{S}(Y)$ would be a proper subcontext of $\hat{S}(L_1) = \hat{S}(L_2)$ which is a subletter both of $\hat{S}(X)$ and $\hat{S}(Y)$ and we obtain a contradiction. Likewise, the field of $X_0^{(v)}$ cannot be completely contained in the side area of L_2. We conclude that the main node η_1' of L_1 is the branching point of $Y_0^{(u)}$ and $X_1^{(r)}$, and the main node η_2' of L_2 is the branching point of $X_0^{(v)}$ and $Y_1^{(s)}$. We consider the following suffixes of the given bases:

1. the suffix X_2 of $X_1^{(r)}$ starting at η_1',
2. the suffix Y_2 of $Y_0^{(u)}$ starting at η_1',
3. the suffix X_3 of $X_0^{(v)}$ starting at η_2',
4. the suffix Y_3 of $Y_1^{(s)}$ starting at η_2',
5. the suffix X_2' of X_2 starting at the respective child of η_1',
6. the suffix Y_3' of Y_3 starting at the respective child of η_2'.

We have the following situation.



Obviously $\hat{S}(X_2)$ and $\hat{S}(X_3)$ are distinct. Hence either $\hat{S}(X_2)$ is a proper suffix of $\hat{S}(X_3)$ or vice versa. Since $\hat{S}(L_1) = \hat{S}(L_2)$ Lemma 9.10 shows that $\hat{S}(X_3)$ is a proper suffix of $\hat{S}(X_2)$ and, since η_1' is labeled, a suffix of $\hat{S}(X_2')$. Symmetrically it follows that $\hat{S}(Y_2)$ is a suffix of $\hat{S}(Y_3')$. The observations show that have now the following chain, where the symbol "$\subset$" (resp. "$\subseteq$") denote proper (non-strict) subcontext relationship:

$$\hat{S}(Y_3') \subset \hat{S}(X_3) \subseteq \hat{S}(X_2') \subset \hat{S}(Y_2) \subseteq \hat{S}(Y_3').$$

This yields a contradiction. □

Lemma 9.17 *Let T be an X-Y-binary generalized context problem with solution (t, S). Let $\eta_1 \sim_S \eta_2$ be two top side nodes of T. If the context bases cb_1 and cb_2 of T are X-Y-normal for η_1 and η_2 respectively, then either both cb_1 and cb_2 have suffix type X or both have suffix type Y.*

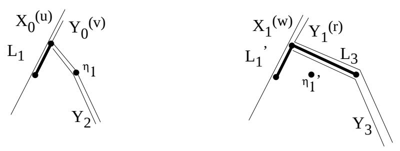
Proof. If there exist two letter descriptions $L_1 \equiv_S L_2$ such that η_1 and η_2 are corresponding top side nodes of L_1 and L_2, then we are in the situation of Lemma 9.16 and we are done. We shall now show that the other situation does not occur, which proves the lemma.

Assume that we are in the remaining case. For $i = 1, 2$, let η_1 be a top side node of the letter description L_i. Since $\eta_1 \sim_S \eta_2$, but $L_1 \not\equiv_S L_2$, Remark 9.13 shows that there exist two distinct letter descriptions L_1' and L_3 with $L_1 \equiv_S L_1'$ such that the top side node η_1' of L_1' that corresponds to η_1 is also a top side node of L_3. L_1' and L_3 have different direction and $\mathcal{L}_1 \not\equiv_S L_3$.

Since T is X-Y-binary we may assume that L_1 is a letter description of a base $X_0^{(u)}$ of suffix type X. By Lemma 9.15, $cb_1 = Y_0^{(v)}$ has suffix type Y. We distinguish two cases:

Case 1: L_1' is a letter description of a base $X_1^{(w)}$ of suffix type X. In this case, L_3 is a letter description of a base $Y_1^{(r)}$ of suffix type Y, by Corollary 9.11. If $\mathit{Field}(cb_1) \subseteq T^{\eta_1}$, then cb_1, which is X-Y-normal for η_1, has type Y and because of $L_1 \equiv_S L_1'$ Lemma 9.10 yields a contradiction. It follows that the main node of L_1 is in $\mathit{Field}(cb_1)$.

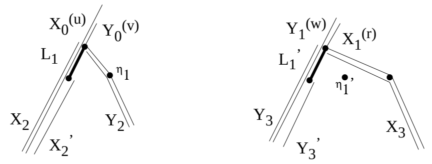
Let Y_2 denote the suffix of $Y_0^{(v)}$ starting at the main node of L_1, and let Y_3 denote the suffix of $Y_1^{(r)}$ starting at the main node of L_1' (or L_3). We have the following picture:



Obviously $\hat{S}(Y_2) \neq \hat{S}(Y_3)$. If $\hat{S}(Y_2)$ is a proper suffix of $\hat{S}(Y_3)$, then $L_1 \equiv_S L_1'$ shows that $\hat{S}(Y_2)$ has a subcontext of the form $\hat{S}(Y_2)$ in its own side area, which yields a contradiction. Conversely, if $\hat{S}(Y_3)$ is a proper suffix of $\hat{S}(Y_2)$, then $L_1 \equiv_S L_1'$ shows that $\hat{S}(Y_3)$ has a subcontext of the form $\hat{S}(Y_3)$ in its own side area, which yields a contradiction. Hence this case cannot occur.

Case 2: L_1' is a letter description of a base $Y_1^{(w)}$ of suffix type Y. If $\mathit{Field}(cb_1) \subseteq T^{\eta_1}$, then cb_1, which is X-Y-normal for η_1, has type Y and because of $L_1 \equiv_S L_1'$ Lemma 9.10 yields a contradiction. It follows that the main node of L_1 is in $\mathit{Field}(cb_1)$. Let Y_2 denote the suffix of $Y_0^{(v)}$ starting at the main node of L_1, and let Y_3 denote the suffix of $Y_1^{(w)}$ starting at the main node of L_1'. Because of $L_1 \equiv_S L_1'$ Lemma 9.10 shows that $\hat{S}(Y_2)$ is a proper suffix of $\hat{S}(Y_3)$.

In addition we know in the present situation that L_3 is a letter description of a base $X_1^{(r)}$ of suffix type X. Let X_2 denote the suffix of $X_0^{(u)}$ starting at the main node of L_1, and let X_3 denote the suffix of $X_1^{(r)}$ starting at the main node of L_1'. Lemma 9.10 shows that $\hat{S}(X_3)$ is a proper suffix of $\hat{S}(X_2)$. With X_2' (resp. Y_3') we denote the suffixes of X_2 (resp. Y_3) starting at the respective successor of the main node of L_1 (resp. L_1'). We have the following picture:



Now $\hat{S}(X_2')$ is a proper subcontext of $\hat{S}(Y_2)$ which is a suffix of $\hat{S}(Y_3')$. Furthermore $\hat{S}(Y_3')$ is a proper subcontext of $\hat{S}(X_3)$ which is a suffix of $\hat{S}(X_2')$. Hence $\hat{S}(X_2')$ has a proper subcontext of the form $\hat{S}(X_2')$ which yields a contradiction. Hence this case can also not occur. $\square$

We now prove Theorem 6.11:

Theorem 9.18 *Let $\mathcal{T}$ denote the output set of (Transl3), where we use the transparent generalized context problem T with bases of type X or Y as input.*

1. *$\mathcal{T}$ is finite,*
2. *Each $T' \in \mathcal{T}$ is a marked generalized context problem.*
3. *If T has a solution, then there exists a problem $T' \in \mathcal{T}$ such that T' has a rigid solution.*
4. *If in T' two individual bases ib_1 and ib_2, with nodes η_1 and η_2, say, have the same type, then the subproblems T'^{η_1} and T'^{η_2} are strictly isomorphic.*
5. *If some $T' \in \mathcal{T}$ has a solution, then T is solvable.*

Proof. 1. Since, by Lemma 9.4, there are only a finite number of iterations of the procedure described in Case III of (Transl3), it follows easily from Part 1 of Lemma 4.4 that $\mathcal{T}$ is finite.

2. Clearly each output problem is a generalized context problem. By assumption, the input problem T_0 is transparent. A simple induction shows that each problem T_i reached by iterations of the procedure of Case III is transparent (cf. Step 3 of (Transl3)). The procedure of Case II cannot introduce branching points. Hence each output problem T' is transparent. But then, obviously the procedure that is applied in Case II of (Transl3) implies that T' is marked.

3. Let (t, S) be a solution of T. We have to describe possible choices in the non-deterministic steps that lead to a generalized context problem that has a rigid solution. Let us introduce the following definition: Let (t, S_i) be a solution of T_i. A subset π of *rel*(T_i) is called $\sim_{S_i}$*-closed* if it satisfies that following condition: if π contains the top side node η_1, and if $\eta_1 \sim_{S_i} \eta_2$, then π contains η_2,

As an induction hypothesis, assume that for some $i \geq 0$ we have found a problem $T_i = \langle N_i, Lab_i, CB_i, Field_i, IB_i, Node_i, \Pi_i, \Delta_i \rangle$ such that

(0) there exists an embedding of T_0 in T_i,

(1) $\Pi_i = \{\pi_1, \ldots, \pi_i\}$ is a partition of a subset of $rel(T_i)$. If η_1 and η_2 belong to the same class of Π_i, then $T_i^{\eta_1}$ and $T_i^{\eta_2}$ are strictly isomorphic.

(2) If η and η' are distinct y-nodes of T_i for some $y \notin \mathcal{X}_0$, then Π_i contains a set π with nodes η_1 and η_1' and strictly isomorphic letter descriptions L and L' with corresponding positions in $T_i^{\eta_1}$ and $T_i^{\eta_1'}$ such that η and η' are corresponding top side nodes of L and L'.

(3) For each $\eta \in \bigcup \Pi_i$ and for each atomic subfield φ in T_i^{η}: if φ is a subfield of a context base, then there exists an atomic base cb with field φ in T_i.

We also assume that there exists a solution (t, S_i) of T_i such that

(4) Each element π of Π_i is $\sim_{S_i}$-closed and $\mathcal{X}$-closed.

(5) for $\eta \in \bigcup \Pi_i$ and $\eta' \in \Delta_i$ the ground term $\hat{S}_i(\eta)$ is not a proper subterm of $\hat{S}_i(\eta')$.

Note that T_0 vacuously satisfies these conditions since Π_0 is empty and T_0 does not contain any y-node for $y \notin \mathcal{X}_0$.

We first consider the situation where $\bigcup \Pi_i = rel(T_i)$, i.e., Case II of (Transl3). We want to show that (t, S_i) is a rigid solution of T_i. Let L_1 and L_2 be letter descriptions of T_i and assume that $\hat{S}_i(L_1) = \hat{S}_i(L_2)$. Let η_1 (resp. η_2) be the j-th top side node of L_1 (resp. L_2). Since $L_1 \equiv_{S_i} L_2$ we have $\eta_1 \sim_{S_i} \eta_2$. Condition (4) shows that η_1 and η_2 belong to the same class of Π_i. Condition (1) shows that $T_i^{\eta_1}$ and $T_i^{\eta_2}$ are strictly isomorphic. It follows that L_1 and L_2 are strictly isomorphic. We have seen that (t, S_i) is a rigid solution of T_i. Condition (3) ensures that the letter descriptions of T_i are not affected by the final marking (cf. Case II). Hence (t, S_i) is also a rigid solution of the output problem.

We now treat the situation where $\bigcup \Pi_i \neq rel(T_i)$, i.e., Case III of (Transl3).

Selection of π_{i+1}. Consider a minimal $\sim_{S_i}$-closed and $\mathcal{X}_0$-closed set $\pi \subseteq rel(T_i)$. By (4), either $\pi \subseteq \Delta_i$ or π is a subset of an element of Π_i. Note also that the nodes of a *minimal* $\sim_{S_i}$-closed and $\mathcal{X}_0$-closed set π are mapped to identical ground terms under S_i. Let us denote this ground term in the form

$\hat{S}_i(\pi)$. Among all minimal $\sim_{S_i}$-closed and $\mathcal{X}_0$-closed subsets of $rel(T_i)$ that are subsets of Δ_i we choose as π_{i+1} one set where $\hat{S}_i(\pi_{i+1})$ is a maximal[4] ground term. Given this choice of π_{i+1} it is obvious that conditions (a) and (b) are satisfied. By choice, π_{i+1} is $\mathcal{X}_0$-closed. Let η and η' be two y-nodes of T_i for $y \notin \mathcal{X}_0$. By (1), η and η' are corresponding top side nodes of strictly isomorphic letter descriptions L and L' of T_i. It follows that $\eta \sim_{S_i} \eta'$. This shows that π_{i+1} is $\mathcal{X}$-closed. Clearly, since T_i is solvable, the selection of π cannot lead to failure before Step 1. By (5) and choice of π_{i+1} we have

(6) $\quad \forall \eta \in \pi_{i+1},\ T_i^{\eta}$ does not contain a node in $\bigcup \Pi_i$, and

Step 1, superposition. Let $\pi_{i+1} =: \{\eta_1, \ldots, \eta_m\}$. As T^S we choose the superposition which is given by the joint embedding (cf. Def. 4.5) of the problems $T_i^{\eta_1}, \ldots, T_i^{\eta_m}$ in $\hat{S}_i(\eta_1)$ under the restrictions of S_i to $T_i^{\eta_1}, \ldots, T_i^{\eta_m}$ respectively. We have to show that the Failure Condition (Step 2 of Case III) does not apply.

Assume that π_{i+1} does not have any x-node for $x \in \mathcal{X}_0$. Let $\eta_j, \eta_k \in \pi_{i+1}$. Let cb_j and cb_k be two bases of T_i such that $|T_i^{\eta_j} \cap Field_i(cb_j)| \geq 2$ and $|T_i^{\eta_k} \cap Field_i(cb_k)| \geq 2$. By Lemma 9.7, $\pi_{i+1} \subseteq \Delta_i$ is X-Y-stable. Hence cb_j and cb_k are subbases of (not necessarily distinct) context bases cb'_j and cb'_k such that cb'_j is X-Y-normal for η_j and cb'_k is X-Y-normal for η_k. In addition we know that $\eta_j \sim_{S_i} \eta_k$ by (2) and choice of π_{i+1}. By Lemma 9.17 we may assume without loss of generality that both cb'_j and cb'_k have suffix type X.

Assume, to get a contradiction, that in the superposition T^{S_0} the context base cb'_j (or its placeholder base) has a node in the side area of cb'_k (or its placeholder base) or vice versa. Then either cb'_j and cb'_k (or their placeholder bases) are branching in T^{S_0}, in which case Corollary 9.11 yields a contradiction, or cb'_j, which is normal for η_j, has type X, in which case Lemma 9.10 yields a contradiction.

We have seen that in T^{S_0} neither cb'_j (or its placeholder base) has a node in the side area of cb'_k (or its placeholder base) nor vice versa. It follows that in T^{S_0} neither cb_j (or its placeholder base) has a node in the side area of cb_k (or its placeholder base) nor vice versa. This shows that Failure Condition 2 of Case III does not apply. Let

$$T_{i+1} = \langle N_{i+1}, Lab_{i+1}, CB_{i+1}, Field_{i+1}, IB_{i+1}, Node_{i+1}, \Pi_{i+1}, \Delta_{i+1} \rangle$$

[4] maximal w.r.t other ground terms of the form $\hat{S}_i(\pi)$ for minimal $\sim_{S_i}$-closed and $\mathcal{X}_0$-closed $\pi \subseteq \Delta_i$.

be defined as described in Case III. We have to show that the above Conditions (0)-(5) hold for T_{i+1}.

(0') Obviously there exists a canonical embedding of T_i in T_{i+1}. By (0) there exists an embedding of T_0 in T_{i+1}.

In the sequel, we do not distinguish between nodes of T_i and their images under the canonical embedding in T_{i+1}.

(1') Obviously $\Pi_{i+1} = \{\pi_1, \ldots, \pi_i, \pi_{i+1}\}$ is a partition of a subset of $rel(T_{i+1})$. Let η and η' belong to the same class of Π_{i+1}. If $\eta, \eta' \in \pi_{i+1}$, then by construction (cf. Steps 3 and 4 in Case III) T_{i+1}^{η} and $T_{i+1}^{\eta'}$ are strictly isomorphic. If η, η' belong to the same class of Π_i, then by assumption (1) T_i^{η} and $T_i^{\eta'}$ are strictly isomorphic. Both η and η' do not belong to π_{i+1}, and by (6), neither η nor η' is a descendant of a node in π_{i+1}. Hence T_i^{η} and $T_i^{\eta'}$ are only modified if some descendant of η or of η' is in π_{i+1}. If $\eta_j \in \pi_{i+1}$ falls in T_i^{η}, say, and if η_j' is the corresponding node of $T_i^{\eta'}$, then it follows from the strict isomorphism (see (1)) between T_i^{η} and $T_i^{\eta'}$ and from $\eta_j \in \pi_{i+1}$ that either both η_j and η_j' are top side nodes where $\eta_j \sim_{S_i} \eta_j'$, or both are x-nodes for some $x \in \mathcal{X}_0$. Hence by choice of π_{i+1} we have $\eta_j' \in \pi_{i+1}$. This observation shows that the subproblems $T_i^{\eta_j}$ and $T_i^{\eta_j'}$ of T_i^{η} and $T_i^{\eta'}$ are replaced by the steps of Case III by strictly isomorphic subproblems. It follows that T_{i+1}^{η} and $T_{i+1}^{\eta'}$ are strictly isomorphic.

(2') Let η and η' be distinct y-nodes of T_{i+1} for some $y \notin \mathcal{X}_0$. If η and η' are y-nodes of T_i, then by assumption (2) there exists $\pi \in \Pi_i$ with nodes η_1 and η_1' in π and strictly isomorphic letter descriptions L and L' with corresponding positions in $T_i^{\eta_1}$ and $T_i^{\eta_1'}$ such that η and η' are corresponding top side nodes of L and L'. By (1), $T_i^{\eta_1}$ and $T_i^{\eta_1'}$ are strictly isomorphic. As in (1) it follows that $T_{i+1}^{\eta_1}$ and $T_{i+1}^{\eta_1'}$ are again strictly isomorphic. Hence the images of L and L' in T_{i+1} are strictly isomorphic and we are done.
In the other case T_i does not have a y-node. In this case the statement (2') follows from Lemma 9.1.

(3') Let $\eta \in \bigcup \Pi_{i+1}$ and let φ be an atomic subfield of T_{i+1}^{η} that is a subfield of a context base. If $\eta \in \pi_{i+1}$ then it follows from Steps 3 and 4 in Case III that there exists an atomic base cb with field φ in T_{i+1}. If

$\eta \in \bigcup \Pi_i$ and if φ is a field of a subproblem $T_i^{\eta_j}$ for some $\eta_j \in \pi_{i+1}$ then it follows again from Steps 3 and 4 in Case III that there exists an atomic base cb with field φ in T_{i+1}. In the remaining case it follows from the induction hypothesis that there exists an atomic base cb with field φ in T_{i+1}.

Let S_{i+1} denote the canonical extension of S_i which is given by the joint embedding of the problems $T_i^{\eta_1}, \ldots, T_i^{\eta_m}$ in $\hat{S}_i(\eta_1)$ (cf. Definition 4.5). This means that $(*)$ for $L \in ld(T_i)$ (resp. $\eta \in tsn(T_i)$) we have $\hat{S}_i(L) = \hat{S}_{i+1}(L)$ (resp. $\hat{S}_i(\eta) = \hat{S}_{i+1}(\eta)$)).

(4') We first show that the sets in Π_{i+1} are $\mathcal{X}$-closed. Let $\pi \in \Pi_i$. Since by assumption (4) π is $\mathcal{X}$-closed in T_i it follows from (6) that π is $\mathcal{X}$-closed in T_{i+1}. The set π_{i+1} is $\mathcal{X}$-closed by restriction (c) (cf. Case III of (Transl3)). If π_{i+1} contains an x-node, then all x-nodes of T_{i+1} are in π_{i+1}. It remains to show that the sets in Π_{i+1} are $\sim_{S_{i+1}}$-closed, i.e., with $\eta \in \pi_k \in \{\pi_1, \ldots, \pi_{i+1}\}$ and $\eta \sim_{S_{i+1}} \eta'$ we have $\eta' \in \pi_k$, for all $\eta' \in tsn(T_{i+1})$. By Remark 9.13, since $\eta \sim_{S_{i+1}} \eta'$ there exists a sequence of pairs $(L_1, \eta_1), \ldots, (L_n, \eta_n)$ of letter descriptions L_j and top side nodes η_j of L_j in T_{i+1}, with $\eta = \eta_1$ and $\eta' = \eta_n$, such that for all consecutive pairs $(L_j, \eta_j), (L_{j+1}, \eta_{j+1})$ $(1 \leq j \leq n-1)$ either

(a) $L_j \equiv_{S_{i+1}} L_{j+1}$ and η_j and η_{j+1} are corresponding top side nodes of L_j and L_{j+1}, or

(b) $\eta_j = \eta_{j+1}$ and $L_j \neq L_{j+1}$ belong to branching context bases.

We use a subinduction to show that for all $1 \leq j \leq n$ always L_j is a letter description of T_i, η_j is a top side node of T_i and $\eta_1 \sim_{S_i} \eta_j$. This shows that $\eta \sim_{S_i} \eta'$ and, by (4) and choice of π_{i+1}, that $\eta' \in \pi_k$.

For $j = 1$ we only have to show that L_1 is in T_i. Otherwise there would be an ancestor η_0 of η that belongs to π_{i+1}. By choice of π_{i+1} this would mean that η is in $\pi_1 \cup \ldots \cup \pi_i$. But then, by $(*)$, $\hat{S}_i(\eta)$ is a proper subterm of $\hat{S}_i(\eta_0)$, which contradicts (5).

For the induction step, assume that L_j (η_j) is a letter description (top side node) of T_i and $\eta_1 \sim_{S_i} \eta_j$. First assume (a) that $L_j \equiv_{S_{i+1}} L_{j+1}$ and η_j and η_{j+1} are corresponding top side nodes of L_j and L_{j+1}. If L_{j+1} is not a letter description of T_i, or if η_{j+1} is not a top side node of T_i, then there exists an ancestor η_0 of η_{j+1} that belongs to π_{i+1}. Then, by induction hypothesis, $\hat{S}_{i+1}(\eta_1) = \hat{S}_{i+1}(\eta_{j+1})$ would be a proper subterm of $\hat{S}_{i+1}(\eta_0)$. By $(*)$, $\hat{S}_i(\eta_1)$ would be a proper subterm of $\hat{S}_i(\eta_0)$, which contradicts (5). In the second case (b) $\eta_j = \eta_{j+1}$ and $L_j \neq L_{j+1}$ belong to branching context bases. As above the assumption that L_{j+1} does not belong to T_i leads to a contradiction.

(5') Let $\eta \in \bigcup \Pi_{i+1}$ and $\eta' \in \Delta_{i+1}$. First assume that $\eta' \in tsn(T_i)$. In particular $\eta' \in \Delta_i$. If $\eta \in \bigcup \Pi_i$, then by (5), $\hat{S}_i(\eta) = \hat{S}_{i+1}(\eta)$ is not a proper subterm of $\hat{S}_i(\eta') = \hat{S}_{i+1}(\eta')$. If $\eta \in \pi_{i+1}$ the choice of π_{i+1} guarantees that $\hat{S}_i(\eta) = \hat{S}_{i+1}(\eta)$ is not a proper subterm of $\hat{S}_i(\eta') = \hat{S}_{i+1}(\eta')$. Now assume that $\eta' \notin tsn(T_i)$, which means that η' is a new top side node of T_{i+1}. Then η' is a descendant of a node $\eta_j \in \pi_{i+1}$. Since, by (5), $\hat{S}_i(\eta) = \hat{S}_{i+1}(\eta)$ is not a proper subterm of $\hat{S}_i(\eta_j) = \hat{S}_{i+1}(\eta_j)$ it follows that $\hat{S}_{i+1}(\eta)$ is not a proper subterm of $\hat{S}_{i+1}(\eta')$.

4. Let T_i be the last problem that is reached before we come to Case II. Let η and η' be two x-nodes of T_i, for some $x \in \mathcal{X}$. Both are in $rel(T_i)$, hence in $\bigcup \Pi_i$. It follows from (4) of the previous step that η and η' belong to the same class of Π_i. By (1), the problems T_i^{η} and $T_i^{\eta'}$ are strictly isomorphic. By (3), the final marking of Case II does not modify T_i^{η} and $T_i^{\eta'}$. Hence the subproblems of the output T' given by the nodes η and η' are strictly isomorphic as well.

5. Let $T' \in \mathcal{T}$ be solvable. In Part 3 above we have seen there exists an embedding of T in T'. By Lemma 4.2, T is solvable. $\square$