Context Unification and Traversal Equations*

Jordi Levy[1] and Mateu Villaret[2]

[1] IIIA, CSIC, Campus de la UAB, Barcelona, Spain.
http://www.iiia.csic.es/~levy
[2] IMA, UdG, Campus de Montilivi, Girona, Spain.
http://www.ima.udg.es/~villaret

Abstract. Context unification was originally defined by H. Comon in ICALP'92, as the problem of finding a unifier for a set of equations containing first-order variables and *context variables*. These context variables have arguments, and can be instantiated by *contexts*. In other words, they are second-order variables that are restricted to be instantiated by linear terms (a linear term is a λ-expression $\lambda x_1 \cdots \lambda x_n . t$ where every x_i occurs exactly once in t).

In this paper, we prove that, if the so called *rank-bound conjecture* is true, then the context unification problem is decidable. This is done reducing context unification to solvability of *traversal equations* (a kind of word unification modulo certain permutations) and then, reducing traversal equations to *word equations with regular constraints.*

1 Introduction

Context unification is defined as the problem of finding a unifier for a finite set of equations where, in addition to first-order variables, we also consider *context variables.* These variables are applied to terms, and can be instantiated by contexts, i.e. by *linear second-order terms.* A linear second-order term is a λ-expression $\lambda x_1 \cdots \lambda x_n . t$ where x_1,..., x_n are first-order bound variables and occur *exactly once* in t. Therefore, context unification can be considered as a variant of second-order unification where possible instances of second-order variables are restricted to be linear. Sometimes, context variables are required to be unary. However, this restriction does not help to prove the decidability of the problem, and it will not be used in this paper. Given an instance of the problem, if it has a solution considered as a context unification problem, then it has also a solution as second-order unification problem. Obviously, the converse is not true.

The context unification problem was originally formulated by H. Comon in [Com92,Com98]. There it is proved that context unification is decidable when, for any context variable, all its occurrences have the same argument. Later, it was proved [SS96,SS98,SS99b] that the problem is also decidable when context variables are stratified, i.e. when, for any variable, the list of context variables

* This work has been partially supported by the CICYT research projects DENOC (BFM 2000-1054-C02-01) and MODELOGOS (TIC 97-0579-C02-01).

A. Middeldorp (Ed.): RTA 2001, LNCS 2051, pp. 169–184, 2001.

we find going from the root of the term to any occurrence of this variable is always the same. It was also proved [Lev96] that a generalization of the problem –the *linear second-order unification problem*, where third-order constants are also allowed– is decidable when no variable occurs more than twice. Recently, it has been proved [SSS99] that context unification is also decidable for problems containing no more than two context variables. The relationship between the context unification problem and the linear second-order unification problem is studied in [LV00b].

Decidability of context unification would have important consequences in different research areas. For instance, some partial decidability results are used in [Com92] to prove decidability of membership constraints, in [SS96] to prove decidability of distributive unification, in [LA96] to define a completion procedure for bi-rewriting systems. In [ENRX98] it is proved that *parallelism constraints* –a kind of partial description of trees– are equivalent to context unification. *Dominance constraints* are a subset of parallelism constraints, and their solvability is decidable [KNT98]. Other application areas of context unification include computational linguistics [NPR97b]. The common assumption is that context unification is decidable. This is because the various restrictions that make context unification decidable, when they are applied to second-order unification, they do not make it decidable [Lev98,LV00a].

In [Lev96] there is a description of a sound and complete context unification procedure, based on Pietrzykowski's procedure [Pie73] for second-order unification. Like Pietrzykowski's procedure, this procedure does not always terminate. The linearity restriction makes some trivially solvable second-order unification problems, like $X(a) \stackrel{?}{=} X(b)$, unsolvable when we only consider context unifiers. Notice that this problem has only one unifier $[X \mapsto \lambda x \,.\, Y]$ which is not linear because x does not occur once in Y. In particular, flexible-flexible pairs, which are always solvable in second-order unification, now are not necessarily solvable.

The *bounded second-order unification problem* is another variant of second-order unification, similar to context unification. There, instances of second-order variables are required to use their arguments a bounded number of times. We can easily reduce any k-bounded second-order unification problem, like $X(Y(a,b)) \stackrel{?}{=} Y(X(a),b)$, to a context unification problem, like

$$\begin{gathered} X(Y(a, \overset{q}{\dots}, a, b, \overset{r}{\dots}, b), \overset{p}{\dots}, Y(a, \overset{q}{\dots}, a, b, \overset{r}{\dots}, b)) \stackrel{?}{=} \\ \stackrel{?}{=} Y(X(a, \overset{p}{\dots}, a), \overset{q}{\dots}, X(a, \overset{p}{\dots}, a), b, \overset{r}{\dots}, b) \end{gathered}$$

nondeterministically, for any possible choice of $p, q, r \leq k$ satisfying the bound. The converse reduction does not seem easy to find. The bounded second-order unification problem has recently been proved decidable [SS99a].

The relationship between context unification and word unification [Mak77] was originally suggested in [Lev96]. In [SSS98] it is proved that the exponent of periodicity lemma also holds for context unification. We can easily reduce word unification to context unification by encoding any word unification problem, like $F\,a\,G \stackrel{?}{=} G\,a\,F$, as a monadic context unification problem $F(a(G(b))) \stackrel{?}{=} G(a(F(b)))$, where b is a new constant. This paper suggests that the opposite reduction may also be possible. In the following Section we motivate this statement using a naive reduction. Although it does not work, we will see in the rest of the paper how it could be adapted properly.

2 A Naive Reduction

Given a signature where every symbol has a fixed arity, we can encode a term using its *pre-order traversal* sequence. We can use this fact to encode a context unification problem, like the following one

$$X(Y(a,b)) \stackrel{?}{=} Y(X(a),b) \tag{1}$$

as the following word unification problem

$$X_0\, Y_0\, a\, Y_1\, b\, Y_2\, X_1 \stackrel{?}{=} Y_0\, X_0\, a\, X_1\, Y_1\, b\, Y_2 \tag{2}$$

We can prove easily that *if* the context unification problem (1) is solvable, *then* its corresponding word unification problem (2) is also solvable. In our example, the solution corresponding to the following unifier

$$\begin{array}{l} X \mapsto \lambda x\, .\, f(f(x,b),b) \\ Y \mapsto \lambda x\, .\, \lambda y\, .\, f(f(f(x,b),y),b) \end{array} \tag{3}$$

is

$$\begin{array}{ll} X_0 \mapsto f\, f & Y_0 \mapsto f\, f\, f \\ X_1 \mapsto b\, b & Y_1 \mapsto b \\ & Y_2 \mapsto b \end{array}$$

Unfortunately, the converse is not true. We can find a solution of the word unification problem which does not correspond to the pre-order traversal of any instantiation of the original context unification problem (consider the unifier that instantiates X_0, X_1, Y_0, Y_1 and Y_2 by the empty word). Word unification is decidable [Mak77], and given a solution of the word unification problem we can check if it corresponds to a solution of the context unification problem. Unfortunately, word unification is also infinitary, and we can not repeat this test for infinitely many word unifiers.

The idea to overcome this difficulty comes from the notion of *rank of a term.* In figure 1 there are some examples of terms (trees) with different ranks. Notice that terms with rank bounded by zero are isomorphic to words, and those with rank bounded by one are caterpillars. For signatures of binary symbols, the rank of a term can be defined as follows

$$\begin{array}{l} \mathrm{rank}(a) = 0 \\ \mathrm{rank}(f(t_1,t_2)) = \begin{cases} 1 + \mathrm{rank}(t_1) & \text{if } \mathrm{rank}(t_1) = \mathrm{rank}(t_2) \\ \max\{\mathrm{rank}(t_1), \mathrm{rank}(t_2)\} & \text{if } \mathrm{rank}(t_1) \neq \mathrm{rank}(t_2) \end{cases} \end{array}$$

Alternatively, the rank of a binary tree can also be defined as the depth of the greatest complete binary tree that is embedded in the tree, using the standard embedding of trees.

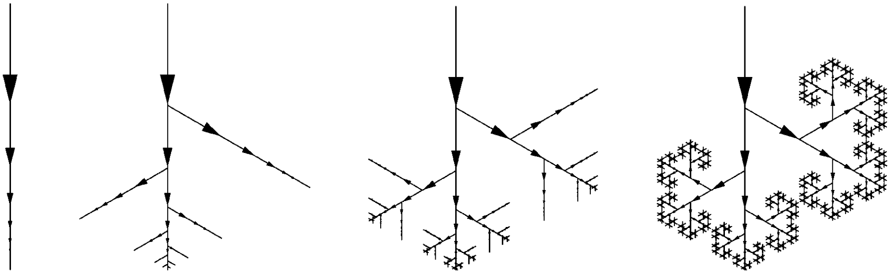


Fig. 1. Examples of trees with ranks equal to 0, 1, 2 and ∞.

We conjecture that there is a computable function Φ such that, for every solvable context unification problem $t \overset{?}{=} u$, there exists a ground unifier σ, such that the rank of $\sigma(t)$ is bounded by the size of the problem: $\text{rank}(\sigma(t)) \leq \Phi(\text{size}(t \overset{?}{=} u))$.

The other idea is to generalise pre-order traversal sequences to a more general notion of *traversal sequence*, by allowing subterms to be traversed in different orders. Then, any rank-bounded term has a traversal sequence belonging to a regular language. We also introduce a new notion of *traversal equation*, noted $t \equiv u$, and meaning t and u are traversal sequences of the same term. We prove that a variant of these constraints can be reduced to word equations with regular constraints, that are decidable [Sch91].

The rest of this paper proceeds as follows. In Section 3 we introduce basic notation. In Section 4 we define the notions of traversal sequence, rank of a traversal sequence, rank of a term, and normal traversal sequence. Traversal equations are introduced in Section 5. There, we prove that solvability of rank- and permutation-bounded traversal equations is decidable, by reducing the problem to solvability of word equations with regular constraints. In Section 6, we state the rank-bound conjecture. Finally, in Section 7 we show how, if the conjecture is true, context unification could be reduced to rank- and permutation-bounded traversal systems.

3 Preliminary Definitions

In this section, we introduce some definitions and notations. Most of them are standard and can be skipped.

We define terms over a *second-order signature* $\langle \Sigma, \mathcal{X} \rangle$ of constants $\Sigma = \bigcup_{i\geq 0} \Sigma_i$ and variables $\mathcal{X} = \bigcup_{i\geq 0} \mathcal{X}_i$, where any constant $f \in \Sigma_i$ or variable $X \in \mathcal{X}_j$ has a fixed arity: $\text{arity}(f) = i$, $\text{arity}(X) = j$. Constants from Σ_0 are called *first-order constants* whereas constants from $\Sigma \backslash \Sigma_0$ are called second-order constants or *function symbols*. Similarly, variables from $\mathcal{X}_0$ are *first-order variables*, and those from $\mathcal{X} \backslash \mathcal{X}_0$ are *context variables*. *First-order terms* $T^1(\Sigma, \mathcal{X})$ and *second-order terms* $T^2(\Sigma, \mathcal{X})$ are defined as usual. The set of *free variables* of a term t is denoted by $\mathrm{Var}(t)$. The *size* of a first-order term is defined inductively by $\mathrm{size}(f(t_1,...,t_n)) = 1 + \sum_{i\in[1..n]} \mathrm{size}(t_i)$ being f either a n-ary constant or variable. The arity of a ($\beta\eta$-normalised) second-order term is defined by $\mathrm{arity}(\lambda x_1 \cdots \lambda x_n \,.\, t) = n$. A second-order term $\lambda x_1 \cdots \lambda x_n \,.\, t$ is said to be *linear* if any bound variable x_i occurs exactly once in t. As far as first-order terms do not contain bound variables, any first-order term is linear.

A *position* within a term is defined, using Dewey decimal notation, as a sequence of integers $i_1 \cdots i_n$, being λ the empty sequence. The concatenation of two sequences is denoted by $p_1 \cdot p_2$. The concatenation of an integer and a sequence is also denoted by $i \cdot p$, standing $i, j,...$ for integers and $p, q,...$ for sequences. The subterm of t at position p is denoted by $t|_p$. By $t[u]_p$ we denote the term t where the subterm at position p has been replaced by u.

The group of permutations of n elements is denoted by Π_n. A permutation ρ of n elements is denoted as a sequence of integers $[\rho(1),...,\rho(n)]$.

A *context unification problem* is a finite sequence of equations $\{t_i \stackrel{?}{=} u_i\}_{i\in[1..n]}$, being an *equation* $t \stackrel{?}{=} u$ a pair of first-order terms $t, u \in T^1(\Sigma, \mathcal{X})$. The *size of a problem* is defined by $\mathrm{size}(\{t_i \stackrel{?}{=} u_i\}_{i\in[1..n]}) = \sum_{i\in[1..n]}(\mathrm{size}(t_i) + \mathrm{size}(u_i))$ A position within a problem or an equation is defined by

$$\begin{aligned}
&\{t_i \stackrel{?}{=} u_i\}_{i\in[1..n]}|_{j\cdot p} = (t_j \stackrel{?}{=} u_j)|_p\\
&(t \stackrel{?}{=} u)|_{1\cdot p} = t|_p\\
&(t \stackrel{?}{=} u)|_{2\cdot p} = u|_p
\end{aligned}$$

A *second-order substitution* is a finite sequence of pairs of variables and terms $\sigma = [X_1 \mapsto s_1,...,X_m \mapsto s_m]$, where X_i and s_i are restricted to have the same arity. A *context substitution* is a second-order substitution where the s_i's are linear terms. A substitution $\sigma = [X_1 \mapsto s_1,...,X_n \mapsto s_n]$ defines a mapping from terms to terms. A substitution σ_1 is said to be *more general* than another σ_2, if there exist another substitution ρ such that $\sigma_2 = \rho \circ \sigma_1$.

Given a context unification problem $\{t_i \stackrel{?}{=} u_i\}_{i\in[1..n]}$, a context [second-order] substitution $\sigma = [X_1 \mapsto s_1,...,X_m \mapsto s_m]$, is said to be a *context [second-order] unifier* if $\sigma(t_i) = \sigma(u_i)$, for any $i \in [1..n]$. A unifier σ is said to be *most general*, m.g.u. for short, if no other unifier is strictly more general than it. It is said to be *ground* if $\sigma(t_i)$ does not contain variables, for any $i \in [1..n]$. A context unification problem is said to be *solvable* if it has a context unifier.

The *context unification problem* is defined as the problem of deciding if, given context unification problem, does it have a context unifier or not.

Without loss of generality, we can assume that the unification problem only contains just one equation $t \stackrel{?}{=} u$. We will also assume that the signature Σ is finite, and that it contains, at least, a first-order constant, and a binary function symbol. This ensures that any solvable context unification problem has a ground unifier, and we can guess constant symbols in non-deterministic computations. If nothing is said, the signature of a problem is the set of symbols occurring in the problem, plus a first-order and a binary constant, if required.

In the appendix we include a variant of the sound and complete context unification procedure described in [Lev96], and adapted to our actual settings.

This procedure can be used to find most general unifiers, and a variant of it, to find minimal ground unifiers.

4 Terms and Traversal Sequences

The solution to the problems pointed out in the introduction comes from generalising the definition of pre-order traversal sequence. It will allow us to traverse the branches of a tree, i.e. the arguments of a function, in any possible order. In order to reconstruct the term from the traversal sequence, we have to annotate the permutation we have used in this particular traversal sequence. For this purpose, we define a new signature Σ_Π containing $n!$ symbols f^ρ for each n-ary symbol $f \in \Sigma$, where $\rho \in \Pi_n$ and Π_n is the group of permutations of n elements.

Definition 1. *Given a signature $\Sigma = \bigcup_{i\geq 0} \Sigma_i$, we define the* extended signature

$$\Sigma_\Pi = \{f^\rho \mid f \in \Sigma \ \wedge \ \rho \in \Pi_{\mathrm{arity}(f)}\}$$

where Π_n is the group of permutations over n elements.
For any $f^\rho \in \Sigma_\Pi$, and its corresponding $f \in \Sigma$, we define $\mathrm{arity}(f^\rho) = \mathrm{arity}(f)$.
A sequence $s \in (\Sigma_\Pi)^*$ is said to be a traversal sequence *of a term $t \in T(\Sigma)$ if:*

1. *$t \in \Sigma_0$, and $s = t$ (the permutation is omitted for first-order constants); or*
2. *$t = f(t_1,...,t_n)$, for any $i \in [1..n]$, there exists a sequence s_i such that it is a traversal sequences of t_i, and there exists a permutation $\rho \in \Pi_n$ such that $s = f^\rho s_{\rho(1)} \cdots s_{\rho(n)}$.*

Definition 2. *Given a sequence of symbols $a_1 \cdots a_n \in (\Sigma_\Pi)^*$, we define its* width *as*

$$\begin{aligned} \mathrm{width}(a) &= \mathrm{arity}(a) - 1 \\ \mathrm{width}(a_1 \cdots a_n) &= \textstyle\sum_{i\in[1..n]} \mathrm{width}(a_i) \end{aligned}$$

This definition can be used to characterize *traversal sequences.*

Lemma 3. *A sequence of symbols $a_1 \cdots a_n \in (\Sigma_\Pi)^*$ is a* traversal sequence, *of some term $t \in T(\Sigma)$, if, and only if,*

$$\begin{aligned} &\mathrm{width}(a_1 \cdots a_n) = -1, \textit{ and} \\ &\mathrm{width}(a_1 \cdots a_i) \geq 0, \textit{ for any } i \in [1..n-1]. \end{aligned}$$

Now we define the rank of a traversal sequence, and by extension, the rank of a term as the minimal rank of its traversal sequences. This definition coincides with the definition given in the introduction for the rank of a term for binary signatures.

Definition 4. *Given a sequence of symbols $a_1 \cdots a_n \in (\Sigma_\Pi)^*$, we define its* rank *as*

$$\mathrm{rank}(a_1 \cdots a_n) = \max\{\mathrm{width}(a_i \cdots a_j) \mid i, j \in [1..n]\}$$

Given a term $t \in T(\Sigma)$, we define its rank *as*

$$\mathrm{rank}(t) = \min\{\mathrm{rank}(w) \mid w \textit{ is a traversal of } t\}$$

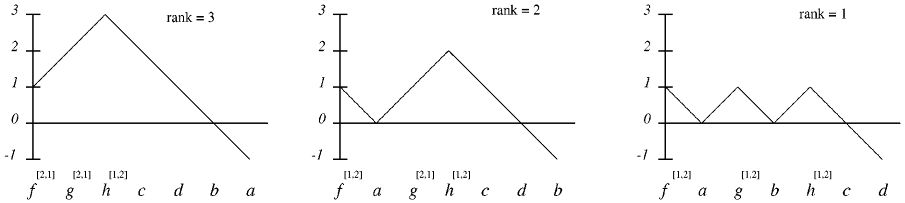


Fig. 2. Representations of the function $f(i) = \text{width}(a_1 \cdots a_i)$, for some traversal sequences of $f(a, g(b, h(c, d)))$.

In general, a term has more than one traversal sequence associated. The rank of the term is always smaller or equal to the rank of its traversals, and for at least one of them we have equality. These rank-minimal traversals are relevant for us, and we choose one of them as the *normal traversal sequence.* In figure 2, the third traversal sequence $f^{[1,2]}\, a\, g^{[1,2]}\, b\, h^{[1,2]}\, c\, d$ is the normal one.

Definition 5. *Given a term t, its* normal traversal sequence $\mathrm{NF}(t)$ *is defined recursively as follows:*

1. *If $t = a$ then $\mathrm{NF}(t) = a$.*
2. *If $t = f(t_1,\dots,t_n)$ then let $\rho \in \Pi_n$ be the permutation satisfying*

$$i < j \Rightarrow \begin{cases} \text{rank}(t_{\rho(i)}) < \text{rank}(t_{\rho(j)}) \\ \vee \\ \text{rank}(t_{\rho(i)}) = \text{rank}(t_{\rho(j)}) \wedge \rho(i) < \rho(j) \end{cases}$$

Then, $\mathrm{NF}(t) = f^{\rho}\ \ \mathrm{NF}(t_{\rho(1)})\ \cdots\ \mathrm{NF}(t_{\rho(n)})$.

Lemma 6. *For any term, its normal traversal sequence has minimal rank, i.e.* $\text{rank}(t) = \text{rank}(\mathrm{NF}(t))$.

Rank-upper bounded traversal sequences define a regular language. The construction of associated automata can be found in [LV00b].

Lemma 7. *Given an extended signature Σ_Π and a constant k, the following set is a regular language.*

$$R^k_\Sigma = \{s \in (\Sigma_\Pi)^* \mid \text{rank}(s) \leq k\ \wedge\ s \textit{ is a traversal}\}$$

Proof. We can define R^k_Σ inductively as follows:

$$R^0_\Sigma = (\Sigma_1)^*\ \Sigma_0$$
$$R^k_\Sigma = R^{k-1}_\Sigma \cup \Big(\bigcup_{n\geq 1} \Sigma_n\ R^{k-n+1}_\Sigma \cdots R^{k-1}_\Sigma\Big)^*\ \Sigma_0$$

5 Traversal Equations

In this section we introduce *traversal equations*. Solvability of traversal equations is still an open question, but we prove that a variant of them (the so called *rank- and permutation-bounded traversal equations*) can be reduced to word equations with regular constraints [Sch91], which are decidable. This reduction is somehow inspired in the reduction from trace equations to word equations used in [DMM97] to prove decidability of trace equations. Later in Section 7, we will reduce context unification to solvability of traversal equations. We need the rank-bound conjecture to prove that the reduction can be done to rank- and permutation-bounded traversal equations.

Definition 8. *A* traversal system *over an extended signature with word variables $(\Sigma_\Pi, \mathcal{W})$ is a conjunction of literals, where every literal has the form* $w_1 \stackrel{?}{=} w_2$ (word equation)*,* $w_1 \equiv w_2$ (traversal equation) *or* $w \in R$ (regular constraint)*, being $w_i \in (\Sigma_\Pi \cup \mathcal{W})^*$ words with variables and $R \subseteq (\Sigma_\Pi)^*$ a regular language.*
A solution *of a traversal system is a word substitution $\sigma : \mathcal{W} \to (\Sigma_\Pi)^*$ such that*

1. *$\sigma(w_1) = \sigma(w_2)$ for any word equation $w_1 \stackrel{?}{=} w_2$,*
2. *$\sigma(w_1)$ and $\sigma(w_2)$ are both traversal sequences of the same term, for any traversal equation $w_1 \equiv w_2$,*
3. *and $\sigma(w)$ belongs to R, for any regular constraint $w \in R$.*

Definition 9. *A traversal system is said to be* rank-bounded *if, for every traversal equation $w_1 \equiv w_2$, there exist two constants k_1 and k_2, and two regular constraints $w_1 \in R_\Sigma^{k_1}$ and $w_2 \in R_\Sigma^{k_2}$ in the system, where R_Σ^k is the (regular) set of k-bounded traversal sequences.*

We can transform rank-bounded traversal systems into equivalent traversal systems using the following transformation rules.

Definition 10. *The following rules define a non-deterministic translation procedure from rank-bounded traversal systems into word equations with regular constraints.*

Rule 1: *For some n-ary symbol $f \in \Sigma$ and permutations $\rho_1, \rho_2 \in \Pi_n$, we replace the traversal equation $w_1 \equiv w_2$ and the corresponding regular constraints $w_1 \in R_\Sigma^{k_1}$ and $w_2 \in R_\Sigma^{k_2}$ by*

$$\begin{array}{lcl}
 & & w_1 \in R_\Sigma^{k_1} \\
 & & w_2 \in R_\Sigma^{k_2} \\
w_1 \equiv w_2 & & w_1 \stackrel{?}{=} X_1 \; f^{\rho_1} \; Y_{\rho_1(1)} \cdots Y_{\rho_1(n)} \; X_2 \\
w_1 \in R_\Sigma^{k_1} & \Longrightarrow & w_2 \stackrel{?}{=} X_1 \; f^{\rho_2} \; Y'_{\rho_2(1)} \cdots Y'_{\rho_2(n)} \; X_2 \\
w_2 \in R_\Sigma^{k_2} & & \left.\begin{array}{l} Y_i \equiv Y'_i \\ Y_{\rho_1(i)} \in R_\Sigma^{k_1-n+i} \\ Y'_{\rho_2(i)} \in R_\Sigma^{k_2-n+i} \end{array}\right\} \textit{for any } i \in [1..n]
\end{array}$$

where X_1, X_2 and $\{Y_i, Y'_i\}_{i \in [1..n]}$ are fresh word variables.

Rule 2: *We replace the traversal equation $w_1 \equiv w_2$ and the corresponding regular constraints $w_1 \in R_\Sigma^{k_1}$ and $w_2 \in R_\Sigma^{k_2}$ by*

$$\begin{array}{l} w_1 \equiv w_2 \\ w_1 \in R_\Sigma^{k_1} \\ w_2 \in R_\Sigma^{k_2} \end{array} \Longrightarrow \begin{array}{l} w_1 \stackrel{?}{=} w_2 \\ w_1 \in R_\Sigma^{\min\{k_1,k_2\}} \end{array}$$

If the rank of a traversal sequence $f^\rho\ w_1 \cdots w_n$ is bounded by k_1, then, for any $i \in [1..n]$, the rank of w_i is bounded by $k_1 - n + i$. These are the values of the exponents used in the regular restrictions of the right-hand side of Rule 1. Rank-boundness is crucial in order to ensure soundness of Rule 2. For instance, the traversal equation $X\,a\,a\,Y \equiv Y\,a\,a\,X$ has no solution, whereas the word equation $X\,a\,a\,Y \stackrel{?}{=} Y\,a\,a\,X$ is solvable. Notice that some substitutions, like $X, Y \mapsto a$, give equal sequences, but they are not traversal sequences.

Theorem 11. *The rules of Definition 10 describe a sound and complete decision procedure for rank-bounded traversal systems. In other words, for any rank-bounded traversal system S,*

1. *if $S \Longrightarrow^* S'$ and the substitution σ is a solution of S', then σ is also a solution of S, and*
2. *if the substitution σ is a solution of S, then there exists a word unification problem with regular constraints S', a transformation sequence $S \Longrightarrow^* S'$, and an extension σ' of σ, such that σ' is a solution of S'.*

Unfortunately, this nondeterministic transformation procedure does not always terminate. Notice that we can have $\rho_1(n) = \rho_2(n) = r$, and in such case we obtain a traversal equation $Y_r \equiv Y'_r$ with the same bounds $Y_r \in R_\Sigma^{k_1}$ and $Y'_r \in R_\Sigma^{k_2}$ as the original one. However, these transformation rules can be used to find solutions σ of equations $w_1 \equiv w_2$, such that $\sigma(w_1)$ and $\sigma(w_2)$ are traversal sequences for the same term, and they are "similar", where "similar" means that they only differ in a bounded number of permutations.

Definition 12. *Given two traversal sequences v and w over Σ_Π, we say that they differ in n permutations if, either*

1. *$v = f^\rho\, r_1 \cdots r_m$ and $w = f^\rho\, s_1 \cdots s_m$, for any $i \in [1..m]$, r_i and s_i differ in n_i permutations, and $\sum_{i=1}^m n_i = n$, or*
2. *$v = f^\rho\, r_{\rho(1)} \cdots r_{\rho(m)}$ and $w = f^\tau\, s_{\tau(1)} \cdots s_{\tau(m)}$, where $\rho \neq \tau$, for any $i \in [1..m]$, r_i and s_i differ in n_i permutations, and $\sum_{i=1}^n n_i = n - 1$.*

Definition 13. *A* permutation-bounded traversal equation,*noted $w_1 \equiv_k w_2$, is a tuple of two words with variables w_1 and w_2, and an integer k.*

A substitution σ is said to be a solution of a permutation-bounded traversal equation $w_1 \equiv_k w_2$ if $\sigma(w_1)$ and $\sigma(w_2)$ are both traversal sequences of the same term, and they only differ in at most k permutations.

A permutation- and rank-bounded traversal system *is a rank-bounded traversal system where all traversal equations are permutation-bounded.*

Theorem 14. *Solvability of permutation- and rank-bounded traversal systems is decidable.*

Proof. We can reduce the problem to an equivalent word unification problem with regular constraints using a variant of the rules of Definition 10 for permutation-bounded equations, finitely many times.

When we apply Rule 1 with $\rho_1 = \rho_2$, we transform $w_1 \equiv_k w_2$ into $\{Y_i \equiv_{k_i} Y'_i\}_{i\in[1..n]}$ where $\sum_{i=1}^n k_i = k$. We can require the existence of $i, j \in [1..n]$, such that $i \neq j$, $k_i \neq 0$ and $k_j \neq 0$ without loosing completeness. When we apply this rule with $\rho_1 \neq \rho_2$, we transform $w_1 \equiv_k w_2$ into $\{Y_i \equiv_{k_i} Y'_i\}_{i\in[1..n]}$ where $\sum_{i=1}^n k_i = k - 1$.

Rule 2 can be applied to transform $w_1 \equiv_k w_2$ into $w_1 \stackrel{?}{=} w_2$, for any k.

It is easy to prove that this transformation process always terminates using a multiset ordering on the multisets of bounds of the traversal equations.

6 The Rank-Bound Conjecture

In this section we introduce the rank-bound conjecture. This is the base of the reduction of context unification to permutation- and rank-bounded traversal systems that we describe in the next section. As we will see, this conjecture is essential in order to prove that the traversal equations that we find in the reduction are both permutation-bounded and rank-bounded.

Conjecture 15 (Rank-Bound Conjecture). *There exists a computable function Φ such that, for any solvable context unification problem $t \stackrel{?}{=} u$ there exists a ground unifier σ satisfying*

$$\mathrm{rank}(\sigma(t)) \leq \Phi(\mathrm{size}(t \stackrel{?}{=} u))$$

The validity of the conjecture is still an open question. In fact, we think that the conjecture is true, not only for *just one* ground unifier, but for *any* most general unifier. This stronger version of the conjecture is not true for second-order unification, because we can have most general second-order unifiers with arbitrarily large rank, as the following example shows.

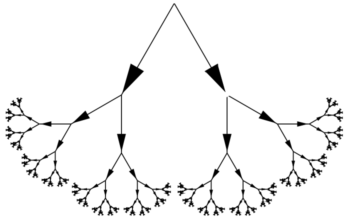
Example 16. The second-order unification problem

$$F(f(a,a)) \stackrel{?}{=} f(F(a), F(a))$$

has only one context unifier $\sigma = [F \mapsto \lambda x\,.\,x]$. However, it has infinitely many second-order unifiers which are not context unifiers, like

$$\sigma = [F \mapsto \lambda x\,.\,f(f(f(x,x), f(x,x)), f(f(x,x), f(x,x)))]$$

For any $n \geq 0$, there is a second-order unifier where bound variable x occurs 2^n many times in the body of the function, and the rank of $\sigma(F(f(a,a)))$ is equal to $n+1$. This term $\sigma(F(f(a,a)))$ can be represented as follows for $n = \infty$.



In the following Lemma we prove that the conjecture is true for first-order unification.

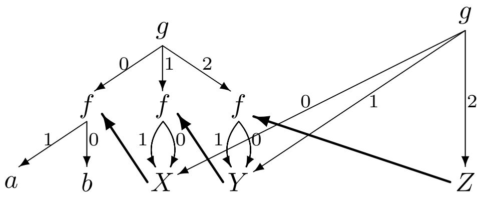
Lemma 17. *Given a solvable first-order unification problem* $t \stackrel{?}{=} u$*, its m.g.u.* σ *satisfies*

$$\mathrm{rank}(\sigma(t)) \leq \mathrm{size}(t) + \mathrm{size}(u)$$

Proof. Suppose we have an unification problem $t \stackrel{?}{=} u$ like

$$g(f(a,b), f(X,X), f(Y,Y)) \stackrel{?}{=} g(X,Y,Z)$$

We can represent it by a directed acyclic graph (DAG) where we have two initial nodes (one for each side of the equation), and a unique node per variable. We can solve the unification problem by re-addressing the arrows pointing to a variable, when this variable is instantiated. Therefore we can represent $\sigma(t)$ by means of a DAG D, where $\mathrm{size}(D) \leq \mathrm{size}(t) + \mathrm{size}(s)$, being the size of a DAG its number of arrows. This is the representation of the DAG corresponding to our example (where, for simplicity, we have added a thick arrow instead of re-addressing arrows pointing to variables):



For any labelling of the original DAG, the same labels in the DAG resulting from instantiation represent a traversal sequence of $\sigma(t)$ and a traversal sequence of $\sigma(u)$. Defining the rank of a node as the addition of the label in the path from the root to this node, the rank of the traversal sequence will be the maximal of the rank of all leaves. In our example, this rank is 5 and it is obtained from the following path

$$g \xrightarrow{2} f \xrightarrow{1} f \xrightarrow{1} f \xrightarrow{1} a$$

The rank of a path never exceeds the number of arrows of the DAG, i.e. its size, because, to avoid occur check, we can not repeat nodes in a path. Therefore, when we use an arrow with label n, there are at least n other arrows (the ones with the same origin) that can not be contained in the same path. We can conclude that the traversal sequence of $\sigma(t)$ represented in the path satisfies $\mathrm{rank}(s) \leq \mathrm{size}(t) + \mathrm{size}(u)$, thus $\mathrm{rank}(\sigma(t)) \leq \mathrm{size}(t) + \mathrm{size}(u)$.

7 Reducing Context Unification to Traversal Equations

In this section we prove that context unification can be reduced to solvability of traversal systems. Moreover, we also prove that if the rank-bound conjecture is true, then this reduction can be done to permutation- and rank-bounded traversal systems. Therefore, if the conjecture is true, then context unification is decidable.

The reduction is very similar to the naive reduction described in Section 2: First-order variables X are encoded as word variables X' such that, if σ is a solution of the context unification problem, and σ' is the corresponding solution of the equivalent word unification problem, then $\sigma'(X') = \mathrm{NF}(\sigma(X))$.

For every n-ary context variable F, we would need $n+1$ word variables $F'_0,\dots,F'_n$, such that $\sigma'(F'_0\, a\, F'_1\, a \cdots F'_{n-1}\, a\, F'_n) = \mathrm{NF}(\sigma(F(a,\dots,a)))$. However, this simple translation does not work. If a term t contains two occurrences of a first-order variable X, then $\mathrm{NF}(\sigma(t))$ will contain two occurrences of $\mathrm{NF}(\sigma(X))$. However, two different occurrences of a context variable can have different arguments, and this means that the context $\sigma(F)$ can be traversed in different ways, depending on the arguments. Notice that, in general, even if $\mathrm{NF}(t[a]) = w_0\, a\, w_1$, we can have $\mathrm{NF}(t[u]) \neq w_0\ \mathrm{NF}(u)\, w_1$. Fortunately, the different ways in which the occurrences of $\sigma(F)$ are traversed in the normal form of $\sigma(t)$ are not very different, i.e. they differ in at most a bounded number of permutations.

Example 18. Let $\sigma(F) = \lambda x\,.\,f(f(x,t_1),t_2)$, where $\mathrm{rank}(t_1) < \mathrm{rank}(t_2)$, and $w_i = \mathrm{NF}(t_i)$, for $i = 1,2$. Depending on the argument u, we have

$$\mathrm{NF}(\sigma(F(u))) = \begin{cases} f^{[1,2]}\, f^{[1,2]}\ \mathrm{NF}(\sigma(u))\, w_1\, w_2 & \text{if } \mathrm{rank}(\sigma(u)) \leq \mathrm{rank}(t_1) \\ f^{[1,2]}\, f^{[2,1]}\, w_1\ \mathrm{NF}(\sigma(u))\, w_2 & \text{if } \mathrm{rank}(t_1) < \mathrm{rank}(\sigma(u)) \leq \mathrm{rank}(t_2) \\ f^{[2,1]}\, w_2\, f^{[2,1]}\, w_1\ \mathrm{NF}(\sigma(u)) & \text{if } \mathrm{rank}(t_2) < \mathrm{rank}(\sigma(u)) \end{cases}$$

For any u and u', $\mathrm{NF}(\sigma(F(u)))$ and $\mathrm{NF}(\sigma(F(u')))$ only differ in at most 2 permutations.

Lemma 19. *Let F be a context variable and σ a substitution. For any two terms $F(t_1,\dots,t_n)$ and $F(u_1,\dots,u_n)$, there exist sequences $v_0,\dots,v_n, w_0,\dots,w_n$ and permutations $\rho,\tau \in \Pi_n$, such that*

$$\begin{aligned} \mathrm{NF}(\sigma(F(t_1,\dots,t_n))) &= v_0\ \mathrm{NF}(\sigma(t_{\rho(1)}))\, v_1 \cdots v_{n-1}\ \mathrm{NF}(\sigma(t_{\rho(n)}))\, v_n \\ \mathrm{NF}(\sigma(F(u_1,\dots,u_n))) &= w_0\ \mathrm{NF}(\sigma(u_{\tau(1)}))\, w_1 \cdots w_{n-1}\ \mathrm{NF}(\sigma(u_{\tau(n)}))\, w_n \end{aligned}$$

and, for any sequence of constants $\{a_i\}_{i\in[1..n]}$,

$$\begin{gathered} v_0\, a_{\rho(1)}\, v_1 \cdots v_{n-1}\, a_{\rho(n)}\, v_n \\ w_0\, a_{\tau(1)}\, w_1 \cdots w_{n-1}\, a_{\tau(n)}\, w_n \end{gathered}$$

are both traversal sequences of $\sigma(F(a_1,\dots,a_n))$, and they only differ in at most $n \cdot \mathrm{rank}(\sigma(F(a_1,\dots,a_n)))$ permutations.

Notice that we need the rank-bound conjecture in order to bound the value of $\mathrm{rank}(\sigma(F(a_1,\ldots,a_n)))$, i.e. to prove that these two traversal sequences differ in a bounded number of permutations.

In the rest we describe how a context unification problem could be effectively translated into an equivalent system of traversal equations.

Theorem 20. *Context unification can be reduced to solvability of traversal systems.*
If the Rank-Bound Conjecture is true, then context unification can be reduced to solvability of permutation- and rank-bounded traversal systems.

Proof. Let $t \stackrel{?}{=} u$ be the original context unification problem, and $(\Sigma, \mathcal{X})$ be the original signature. We assume that Σ is finite, and contains at least $2 \cdot n$ distinct first-order constants $a_1,\ldots,a_n,b_1,\ldots,b_n$, where $n = \max\{\mathrm{arity}(F) \mid F \in \mathrm{Var}(t \stackrel{?}{=} u)\}$, and a binary symbol f, and that $a_1,\ldots,a_n,b_1,\ldots,b_n$ do not occur in $t \stackrel{?}{=} u$. Therefore, if a problem is solvable, it has a ground unifier.

First step. The order of the arguments in F and in $\sigma(F)$ are not necessarily the same. In this first step we guess a permutation $\rho_F \in \Pi_{\mathrm{arity}(F)}$ for any context variable and transform $t \stackrel{?}{=} u$ into $\sigma_0(t) \stackrel{?}{=} \sigma_0(u)$ where

$$\sigma_0 = \bigcup_{F \in \mathrm{Var}(t \stackrel{?}{=} u)} [F \mapsto \lambda x_1 \cdots x_n \,.\, F'(x_{\rho_F(1)},\ldots,x_{\rho_F(n)})]$$

Now, we can assume that F' and its instance have the arguments in the same order. Moreover, as far as σ_0 is simply a renaming substitution, $t \stackrel{?}{=} u$ and $\sigma_0(t) \stackrel{?}{=} \sigma_0(u)$ are equivalent problems.

Second step. We introduce a word variable $X' \in \mathcal{W}$ *for every first order variable* $X \in \mathcal{X}$, and $\mathrm{arity}(F) + 1$ many word variables $F^p_0,\ldots,F^p_{\mathrm{arity}(F)} \in \mathcal{W}$ *for every occurrence* p *of a context variable* F in the problem (notice that in this case we use different word variables for every occurrence).

We guess a permutation ρ_p for any occurrence of a constant function f or of a context variable F, with arity greater or equal than two, in a position p of the problem.

We define the following translating function $\mathcal{T}$ that given a subterm $t \in T^1(\Sigma, \mathcal{X})$ of the problem, and its position p, returns its translation in terms of words with variables $w \in (\Sigma_\Pi \cup \mathcal{W})^*$.

For any first-order constant a, or variable X,

$$\mathcal{T}(a,p) = a$$
$$\mathcal{T}(X,p) = X'$$

For every n-ary function symbol f, or context variable F, occurring at position p, let $w_i = \mathcal{T}(t_i,\, p \cdot i)$, and ρ_p be the permutation conjectured for this position, then

$$\mathcal{T}(f(t_1,\ldots,t_n),p) = f^{\rho_p}\, w_{\rho_p(1)} \cdots w_{\rho_p(n)}$$
$$\mathcal{T}(F(t_1,\ldots,t_n),p) = F^p_0\, w_{\rho_p(1)}\, F^p_1 \cdots F^p_{n-1}\, w_{\rho_p(n)}\, F^p_n$$

Finally, the traversal system will contain the following equations:

1. A word equation for the original problem $t \stackrel{?}{=} u$
$$\mathcal{T}(t,1) \stackrel{?}{=} \mathcal{T}(u,2)$$

2a. For any two occurrences $F(t_1,\ldots,t_n)$ and $F(u_1,\ldots,u_n)$ of a context variable F at positions p and q, we introduce the following traversal equations and regular constraints:[1]
$$\begin{array}{ll} \mathcal{T}(F(a_1,\ldots,a_n),p) \equiv_k \mathcal{T}(F(a_1,\ldots,a_n),q) & \mathcal{T}(F(b_1,\ldots,b_n),p) \equiv_k \mathcal{T}(F(b_1,\ldots,b_n),q) \\ \mathcal{T}(F(a_1,\ldots,a_n),p) \in R^{k_1}_{\Sigma_\Pi} & \mathcal{T}(F(b_1,\ldots,b_n),p) \in R^{k_1}_{\Sigma_\Pi} \\ \mathcal{T}(F(a_1,\ldots,a_n),q) \in R^{k_2}_{\Sigma_\Pi} & \mathcal{T}(F(b_1,\ldots,b_n),q) \in R^{k_2}_{\Sigma_\Pi} \end{array}$$
where $k = \text{arity}(F) \cdot \Phi(\text{size}(t \stackrel{?}{=} u))$
$k_1 = k_2 = \Phi(\text{size}(t \stackrel{?}{=} u))$
and Φ is the computable function introduced in the rank-bound conjecture.

2b. In case we want to reduce context unification to (non-bounded) traversal systems, we will introduce
$$\mathcal{T}(F(a_1,\ldots,a_n),p) \equiv \mathcal{T}(F(a_1,\ldots,a_n),q)$$
$$\mathcal{T}(F(b_1,\ldots,b_n),p) \equiv \mathcal{T}(F(b_1,\ldots,b_n),q)$$
In this second case, we do not need the conjecture to fix k, k_1 and k_2.

The duplication of traversal equations with distinct constants a_i and b_i ensures that these constants occur in the place of the arguments. Otherwise, if we only introduce a traversal equation $X_0\,a\,X_1 \equiv X'_0\,a\,X'_1$, we can get solutions like $\sigma = [X_0 \mapsto f^{[1,2]}\,a][X_1 \mapsto \lambda][X'_0 \mapsto f^{[1,2]}][X'_1 \mapsto a]$, that do not satisfy $\sigma(X_0\,b\,X_1) \equiv \sigma(X'_0\,b\,X'_1)$, and leads to incompatible definitions of $\sigma(F) = \lambda x\,.\,f(a,x)$ and $\sigma(F) = \lambda x\,.\,f(x,a)$.

Corollary 21. *If the Rank-Bound Conjecture is true, then Context Unification is decidable.*

Example 22. To conclude, let's see how problem $X(Y(a,b)) \stackrel{?}{=} Y(X(a),b)$ could be translated into a traversal system.

We guess σ_0 equals to identity in the first step. In second step, we introduce the word variables X_0, X_1, X'_0, X'_1 for the two occurrences of X, and $Y_0, Y_1, Y_2, Y'_0, Y'_1, Y'_2$ for Y. For both occurrences of Y, the only symbol with arity 2 or greater, we guess the same permutation $\rho_{1\cdot 1} = \rho_2 = [2,1]$.

The translation of the unification problem results then into:
$$X_0\,Y_0\,b\,Y_1\,a\,Y_2\,X_1 \stackrel{?}{=} Y'_0\,b\,Y'_1\,X'_0\,a\,X'_1\,Y'_2$$
$$\begin{array}{ll} X_0\,a_1\,X_1 \equiv_k X'_0\,a_1\,X'_1 & X_0\,b_1\,X_1 \equiv_k X'_0\,b_1\,X'_1 \\ X_0\,a_1\,X_1 \in R^k_{\Sigma_\Pi} & X_0\,b_1\,X_1 \in R^k_{\Sigma_\Pi} \\ X'_0\,a_1\,X'_1 \in R^k_{\Sigma_\Pi} & X'_0\,b_1\,X'_1 \in R^k_{\Sigma_\Pi} \end{array}$$

[1] We can avoid to introduce a context variable occurrence in more than two traversal equation. If we have $p_1,\ldots,p_n$ occurrences of F, we can introduce an equation relating p_1 and p_2, p_2 and p_3,..., p_{n-1} and p_n.

$$\begin{array}{ll}
Y_0\, a_2\, Y_1\, a_1\, Y_2 \equiv_{2k} Y'_0\, a_2\, Y'_1\, a_1\, Y'_2 & Y_0\, b_2\, Y_1\, b_1\, Y_2 \equiv_{2k} Y'_0\, b_2\, Y'_1\, b_1\, Y'_2 \\
Y_0\, a_2\, Y_1\, a_1\, Y_2 \in R^k_{\Sigma_\Pi} & Y_0\, b_2\, Y_1\, b_1\, Y_2 \in R^k_{\Sigma_\Pi} \\
Y'_0\, a_2\, Y'_1\, a_1\, Y'_2 \in R^k_{\Sigma_\Pi} & Y'_0\, b_2\, Y'_1\, b_1\, Y \in R^k_{\Sigma_\Pi}
\end{array}$$

where $k = \varPhi(8)$, and $\varPhi$ is the function introduced by the rank-bound conjecture.

8 Conclusions and Further Work

In this paper we prove that, if the rank-bound conjecture is true, then context unification is decidable. The decidability of context unification is still an open question, and a positive answer would have important implications in very different research areas. Additionally, we define *traversal equations* and *rank- and permutation-bounded traversal equations*, and prove that solvability of the second ones is decidable.

We are currently trying to prove the rank-bound conjecture, and finding a reduction from traversal equations to context unification, to prove the equivalence of both problems.

Acknowledgements. We are in debt with M. Schmidt-Schauß and K.U. Schulz who carefully read a preliminary version of this paper and found some errors. We also acknowledge M.L. Bonet for her valuable comments and suggestions, and all the anonymous referees.

References

[Com92] Hubert Comon. Completion of rewrite systems with membership constraints. In *Int. Coll. on Automata, Languages and Programming, ICALP'92*, volume 623 of *LNCS*, Vienna, Austria, 1992.

[Com98] Hubert Comon. Completion of rewrite systems with membership constraints. *Journal of Symbolic Computation*, 25(4):397–453, 1998.

[DMM97] Volker Diekert, Yuri Matiyasevich, and Anca Muscholl. Solving trace equations using lexicographical normal forms. In *Int. Colloquium on Automata, Languages and Programming, ICALP'97*, pages 336–346, Bologna, Italy, 1997.

[ENRX98] Markus Egg, Joachim Niehren, Peter Ruhrberg, and Feiyu Xu. Constraints over lambda-structures in semantic underspecification. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (ACL'98)*, pages 353–359, Montreal, Quebec, Canada, 1998.

[KNT98] Alexander Koller, Joachim Niehren, and Ralf Treinen. Dominance constraints: Algorithms and complexity. In *Proceedings of the Third Conf. on Logical Aspects of Computational Linguistics*, Grenoble, France, 1998.

[LA96] Jordi Levy and Jaume Agustí. Bi-rewrite systems. *Journal of Symbolic Computation*, 22(3):279–314, 1996.

[Lev96] Jordi Levy. Linear second-order unification. In *Proceedings of the 7th Int. Conf. on Rewriting Techniques and Applications (RTA'96)*, volume 1103 of *LNCS*, pages 332–346, New Brunswick, New Jersey, 1996.

[Lev98] Jordi Levy. Decidable and undecidable second-order unification problems. In *Proceedings of the 9th Int. Conf. on Rewriting Techniques and Applications (RTA'98)*, volume 1379 of *LNCS*, pages 47–60, Tsukuba, Japan, 1998.

[LV98] Jordi Levy and Margus Veanes. On unification problems in restricted second-order languages. In *Annual Conf. of the European Ass. of Computer Science Logic (CSL98)*, Brno, Czech Republic, 1998.

[LV00a] Jordi Levy and Margus Veanes. On the undecidability of second-order unification. *Information and Computation*, 159:125–150, 2000.

[LV00b] Jordi Levy and Mateu Villaret. Linear second-order unification and context unification with tree-regular constraints. In *Proceedings of the 11th Int. Conf. on Rewriting Techniques and Applications (RTA'00)*, volume 1833 of *LNCS*, pages 156–171, Norwich, UK, 2000.

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

[NPR97a] Joachim Niehren, Manfred Pinkal, and Peter Ruhrberg. On equality up-to constraints over finite trees, context unification, and one-step rewriting. In *Proceedings of the 14th Int. Conference on Automated Deduction (CADE-14)*, volume 1249 of *LNCS*, pages 34–48, Townsville, North Queensland, Australia, 1997.

[NPR97b] Joachim Niehren, Manfred Pinkal, and Peter Ruhrberg. A uniform approach to underspecification and parallelism. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL'97)*, pages 410–417, Madrid, Spain, 1997.

[NTT00] Joachim Niehren, Sophie Tison, and Ralf Treinen. On rewrite constraints and context unification. *Information Processing Letters*, 74(1-2):35–40, 2000.

[Pie73] Tomasz Pietrzykowski. A complete mechanization of second-order logic. *J. of the ACM*, 20(2):333–364, 1973.

[Sch91] Klaus U. Schulz. Makanin's algorithm, two improvements and a generalization. Technical Report CIS-Bericht-91-39, Centrum für Informations und Sprachverarbeitung, Universität München, 1991.

[SS96] Manfred Schmidt-Schauß. An algorithm for distributive unification. In *Proceedings of the 7th Int. Conf. on Rewriting Techniques and Applications (RTA'96)*, volume 1103 of *LNCS*, pages 287–301, New Jersey, USA, 1996.

[SS98] Manfred Schmidt-Schauß. A decision algorithm for distributive unification. *Theoretical Computer Science*, 208:111–148, 1998.

[SS99a] Manfred Schmidt-Schauß. Decidability of bounded second-order unification. Technical Report Frank-report-11, FB Informatik, J.W. Goethe Universität Frankfurt, 1999.

[SS99b] Manfred Schmidt-Schauß. A decision algorithm for stratified context unification. Technical Report Frank-report-12, FB Informatik, J.W. Goethe Universität Frankfurt, 1999.

[SSS98] Manfred Schmidt-Schauß and Klaus U. Schulz. On the exponent of periodicity of minimal solutions of context equations. In *Proceedings of the 9th Int. Conf. on Rewriting Techniques and Applications (RTA'98)*, volume 1379 of *LNCS*, pages 61–75, Tsukuba, Japan, 1998.

[SSS99] Manfred Schmidt-Schauß and Klaus U. Schulz. Solvability of context equations with two context variables is decidable. In *Proceedings of the 16th Int. Conf. on Automated Deduction (CADE-16)*, LNAI, pages 67–81, 1999.