On Rewrite Constraints and Context Unification

Joachim Niehren [1]

Universität des Saarlandes, Postfach 15 11 50, D-66041 Saarbrücken, Germany

Sophie Tison [1]

LIFL, Unicersité Lille 1, F-59655 Villeneuve d'Ascq cedex, France

Ralf Treinen [1]

LRI, Université Paris-Sud, F-91405 Orsay cedex, France

Abstract

We show that stratified context unification, which is one of the most expressive fragments of context unification known to be decidable, is equivalent to the satisfiability problem of slightly generalized rewriting constraints.

Key words: Automatic Theorem Proving; Theory of Computation; Unification; Rewrite Constraints.

1 Introduction

Context unification (CU) was introduced in rewriting and unification theory [3,14]. CU can be considered as second-order linear unification [6], that is second-order unification where the interpretation of second-order variables is restricted to lambda-terms with exactly one occurrence of the bound variable. Hence, CU is a restriction of higher-order unification (which is undecidable even in the second-order case [5]) and a generalization of string unification (which is decidable [9]). Decidability of CU is still open.

A decidable fragment of CU called *stratified* CU has been introduced in [15]. It is shown in [17] that context unification with two context variables – each

[1] Partially supported by the Esprit Working Group 22457 - CCL II

of which may occur an arbitrary number of times – is decidable. Furthermore, so-called *bounded* second-order unification where lambda-terms may have one *or zero* occurrence of the bound variable is decidable [16]. CU has applications in solving membership constraints in completion of constrained rewriting [3], solving constraints occurring in distributive unification [15], extended critical pairs in bi-rewriting systems [7] and semantics of ellipses in natural language [13,4,11].

The investigation of *(one-step) rewrite constraints* (RC) has been initiated by [1]. Atomic rewrite constraints have the form $x \to y$ *by* R, saying that a ground term denoted by x rewrites by the rewrite system R to a ground term denoted by y (in its most primitive form only one fixed rewrite system R is allowed to occur in a constraint). The original project was to show decidability of the first-order theory of these constraints since such a result would have allowed to generalize known decidability results in rewrite theory. However, undecidability of the $\forall^*\exists^*$-fragment could be shown even for very simple classes of rewrite systems [19,20,10,18]. The question of decidability of the purely existential fragment of positive and negative rewrite constraints remains open, even though some cases for restricted classes of rewrite systems are solved [2,8].

It has been shown in [12] that satisfiability of RC can be expressed as satisfiability of stratified CU and hence is decidable. However, it was not known whether stratified CU really is more difficult than solving RC. In this paper, we propose a minor extension of RC and show that it is in fact equivalent to stratified CU, with linear-time translations in both directions. Our extension concerns a means to compare the positions at which one term rewrites into another. We consider this extension to be insignificant since whenever rewrite constraints such as $x \to y$ *by* $R_1 \wedge x \to z$ *by* R_2 are to be resolved then it is a natural first step to consider the different cases according to the relative positions of the two redeces in x. Hence, in our opinion, any method to solve RC anyway has to cope with comparisons of positions of redeces in a term. In this sense we argue that Stratified Context Unification Problems are essentially equivalent to Rewrite Constraints.

2 The Languages

The syntax of context unification is given in Figure 1. A *CU-term* T is a tree-valued term which is built from tree variables x, y, z, context variables C, D, E, and function symbols from a signature Σ (a is a constant and f a function symbol in Σ). A *tree* over Σ is a *ground* CU-term, i.e. a term without (tree or context) variables.

A system of CU-equations is a conjunction of equations between CU-terms.

$$\begin{array}{ll} \text{CU-terms} & T ::= C(T) \mid x \mid f(T_1, \ldots, T_n) \\ \text{CU-equation systems} & E ::= T = T' \mid E \wedge E' \end{array}$$

Fig. 1. Terms and equations in context unification

$$\begin{array}{ll} \text{FO-terms} & t ::= x \mid f(t_1, \ldots, t_n) \\ \text{rewrite constraints} & R ::= x \to y \; at \; C \; by \; t \to t' \\ & \quad \mid C{=}\text{id} \mid C \leq C' \mid R \wedge R' \end{array}$$

Fig. 2. First-order terms and rewrite constraints

CU-equations are interpreted in the two sorted algebra where every context-variable is assigned a *context*, that is a λ-term with exactly one occurrence of the bound variable, and where a CU-term t denotes the tree obtained as β-normal form of the λ-term t with his variables replaced by their values.

A *context term* is a sequence of context variables $C_1 \ldots C_n$, $n \geq 0$. The empty sequence is written id. The *second-order prefix of a position* in a term (CU-term or context term) is the context term given by the sequence of context-variables lying on the path from the root of the term to the position. A set of CU-terms is called *stratified* if every two occurrences of the same (tree or context) variable have the same second-order prefix. A CU-equation system E is stratified if the set of all CU-terms used as left or right hand side in an equation of E is stratified.

Example 1 *The system $D(f(a)) = f(D(a))$ is stratified since both occurrences of the context-variable D have the second-order prefix id. The set of solutions for D is $\{(\lambda x.f^n(x)) \mid n \geq 0\}$. The system $D(f(D(a))) = f(D(f(a)))$ is not stratified since the innermost occurrence of D on the left hand side has second-order prefix D but the two other occurrences of D have second-order prefix id. Its only solution is $\lambda x.f(x)$.*

The syntax of rewrite constraints is given in Figure 2. Variables x, y, z denote trees. The rewrite constraint $x \to y \; at \, C \; by \, t \to t'$ means that x rewrites to y at context C by using the rule $t \to t'$. We assume $x, y \notin V$ where $V = V(t) \cup V(t')$. Then, $x \to y \; at \; C \; by \; t \to t'$ is equivalent to $\exists V(x = C(t) \wedge y = C(t))$. Hence, the variables in a rewrite rule should be seen as bound variables having that rewrite rule as scope. The ordering constraint $C \leq D$ means that D denotes an *instance* of C and is equivalent to $\exists E(CE = D)$ where juxtaposition is interpreted by composition.

Example 2 *The rewrite constraint $x \to y$ at id by $f(z) \to z$ is equivalent to $x = f(y)$.*

Our main result is

$$\text{(U1)} \quad \frac{x \rightarrow y \ \mathit{at}\ C \ \mathit{by}\ t \rightarrow t'}{\exists V(x = C(t) \wedge y = C(t'))} \quad \begin{array}{l} V = V(t) \cup V(t') \\ \text{fresh variables} \end{array}$$

$$\text{(U2)} \quad \frac{C = \mathrm{id}}{C(a) = a} \quad a \in \Sigma$$

$$\text{(U3)} \quad \frac{C \leq D}{\exists E(D(t) = C(E(t)) \wedge D(t') = C(E(t')))} \quad \begin{array}{l} t \neq t' \text{ ground} \\ E \text{ fresh} \end{array}$$

Fig. 3. Rewrite Constraints as CU-Equations

Theorem 3 *For every signature, there is a linear time, satisfiability preserving translation which maps a stratified system of CU-equations to a rewrite constraint, and vice versa.*

3 Rewrite Constraints as Stratified CU Equations

It was already shown in [12] that rewrite constraints of the form $x \rightarrow y \ \mathit{by}\ t \rightarrow t'$ can be translated into a stratified system of CU-equations. This translation is extended in Figure 3 to the slightly more general rewrite constraints that we consider in this article. The correctness of the translation of $C \leq D$ by rule (U3) was already proved in [11].

Proposition 4 *Given a rewrite constraint the rules (U1)–(U3) in Figure 3 terminate and yield a satisfaction equivalent stratified system of CU-equation in linear time.*

4 Stratified CU-Equations as Rewrite Constraints

It remains to show that stratified systems of CU-equations can be translated to rewrite constraints. We proceed in three steps: We first show that we can restrict ourselves to *normalized CU-equations*, that is equations of the form $x = T$ where T is a CU-term without tree variables. Second, we translate normalized CU-equations into contextual constraints - an expressive generalization of rewrite constraints - such that stratification is preserved. Third, we map stratified contextual constraints to rewrite constraints.

Proposition 5 *For every signature Σ there exists a signature Σ' with a single constant such that CU-equations over Σ can be translated in linear time by preserving satisfiability and stratification into CU-equations over Σ'.*

Proof: For any signature Σ let Σ' be the signature consisting of all non-constant symbols of Σ, plus the constants of Σ considered as unary function

context terms	$\Delta ::= \Delta C \mid \mathrm{id}$
contextual constraints	$S ::= x \to y \ at\ \Delta\ by\ t \to t' \mid S \wedge S'$

Fig. 4. Contextual constraints

$$\text{(C1)} \quad \frac{x = \Delta(f(T_1, \ldots, T_n))}{\bigwedge_{i=1,\ldots,n} \exists x_i (x_i = \Delta(T_i) \ \wedge\ x \to x_i \ at\ \Delta\ by\ f(u_1, \ldots, u_n) \to u_i)} \quad n \neq 0$$

$$\text{(C2)} \quad \frac{x = \Delta(a)}{x \to x \ at\ \Delta\ by\ a \to a} \quad a \text{ constant}$$

Fig. 5. Normal CU-equations into contextual constraints

symbols, plus a new constant a. Analogously, we can transform a system of context equations E into a system E' by replacing every constant c by $c(a)$. Now it is easy to see that E is satisfiable over Σ iff E' is satisfiable over Σ'. Note that we can obtain, from an arbitrary solution of E' over Σ', a solution of E over Σ simply be replacing $c(a)$ by the constant c and by removing all remaining new unary function symbols c. □

Proposition 6 *Every CU-equation can be normalized in linear time such that stratification and satisfiability are preserved.*

Proof: According to Proposition 5 we can assume that the signature Σ contains only one constant a. For any tree variable x, we fix a new context-variable C_x and replace all occurrences of x by $C_x(a)$. This transformation preserves satisfiability since all ground terms have to contain the constant a. It also preserves stratification since the occurrences of C_x have the same second-order prefixes as the occurrences of x before. Finally, we replace an equation $t = s$ by $x = t \wedge x = s$ for some fresh variable x. □

In Figure 4 we present *contextual constraints* which are much more expressive than rewrite constraints in that they allow to specify the rewrite position by a context term Δ. A contextual constraint $x \to y \ at\ \Delta\ by\ t \to t'$ is equivalent to $\exists V(x = \Delta(t) \ \wedge\ y = \Delta(t'))$ where all variables in $V = V(t) \cup V(t')$ are supposed to be fresh. We call a system of contextual constraints *stratified* if its set of context terms is stratified.

Proposition 7 *A normalized system of CU-equations can be translated in linear time to a contextual constraint such that stratification and satisfiability are preserved.*

Proof: Given a normalized system of CU-equations, the rules (C1)–(C2) in Figure 5 yield a satisfaction equivalent contextual constraint. The rules terminate in linear time: Both rules replace one CU-equation by one contextual constraint plus one CU-equation per subterm. It is obvious that both rules are

A stratified system of CU-equations:

$$x = D(f(E(g(a)))) \qquad x = D(h(E(b), F(c)))$$

Translation to a stratified contextual constraint:

$$\begin{array}{ll} x \to x_1 \ \textit{at}\ D\ \textit{by}\ f(u) \to u & x_1 \to x_1 \ \textit{at}\ DE\ \textit{by}\ g(a) \to g(a) \\ x \to x_2 \ \textit{at}\ D\ \textit{by}\ h(u_1,u_2) \to u_1 & x_2 \to x_2 \ \textit{at}\ DE\ \textit{by}\ b \to b \\ x \to x_3 \ \textit{at}\ D\ \textit{by}\ h(u_1,u_2) \to u_2 & x_3 \to x_3 \ \textit{at}\ DF\ \textit{by}\ c \to c \end{array}$$

Translation to a rewrite constraint:

$$\begin{array}{ll} x \to x_1 \ \textit{at}\ D\ \textit{by}\ f(u) \to u & x_1 \to x_1 \ \textit{at}\ C_1\ \textit{by}\ g(a) \to g(a) \\ x \to x_2 \ \textit{at}\ D\ \textit{by}\ h(u_1,u_2) \to u_1 & x_2 \to x_2 \ \textit{at}\ C_1\ \textit{by}\ b \to b \\ x \to x_3 \ \textit{at}\ D\ \textit{by}\ h(u_1,u_2) \to u_2 & x_3 \to x_3 \ \textit{at}\ C_2\ \textit{by}\ c \to c \\ & D \leq C_1 \ \wedge \ D \leq C_2 \end{array}$$

Fig. 6. Translation of a stratified CU-equations by example

sound. They preserve stratification since deletion of function symbols does not change second-order prefixes. □

In fact, we could generalize rule (C2) be allowing an arbitrary ground term instead of a constant a. An example for the translation of a stratified system of normalized CU-equations into a stratified contextual constraint is given in Figure 6.

Proposition 8 *A stratified contextual constraint can be transformed in linear time into a satisfaction equivalent rewrite constraint.*

Proof: Given a contextual constraint, we replace all its context terms $\Delta_1, \ldots, \Delta_n$ by fresh variables $C_1, \ldots, C_n$, always using the same variable for replacing multiple occurrences of the same context term. We obtain a rewrite constraint plus a system of equations $\bigwedge_{i=1}^n C_i = \Delta_i$ such that 1) for all $i, j \in \{1, \ldots, n\}$: C_i does not occur in Δ_j, 2) all Δ_i are pairwise distinct, 3) the set $\{\Delta_1, \ldots, \Delta_n\}$ is stratified.

Let Δ_j be a term of maximal length in this set. If $\Delta_j = \mathrm{id}$ then all equations in $\bigwedge_{i=1}^n C_i = \Delta_i$ are of the form $C_i = \mathrm{id}$ and hence rewrite constraints. Otherwise, $\Delta_j = \Delta_j' D$ for some context term Δ_j' and context variable D. We next show that D cannot occur elsewhere in the equation system. If $\Delta_i = \Delta^1 D \Delta^2$ for some i, Δ^1, Δ^2 then $\Delta^1 = \Delta_j'$ by stratification and $\Delta^2 = \mathrm{id}$ due to maximality. Since all terms Δ_i are distinct, the occurrences of D in Δ_j and $\Delta^1 D \Delta^2$ must be equal.

If our equation system does not contain an equation $C = \Delta'_j$ for some C than we add one for a fresh variable C. Given that D occurs only once, we can safely replace the equation $C_j = \Delta'_j D$ by $\exists D(C_j = \Delta'_j D)$ and thus by $C \leq C_j$, and continue the process. □

Example 9 *The following stratified system of equations*

$$\begin{array}{l} C_1 = id \ \wedge \ C_2 = D \ \wedge \ C_3 = DE \ \wedge \\ C_4 = DF \ \wedge \ C_5 = DEG \ \wedge \ C_6 = DEH \end{array}$$

is satisfaction equivalent to the following system of ordering constraints:

$$C_1{=}id \ \wedge \ C_1 \leq C_2 \ \wedge \ C_2 \leq C_3 \ \wedge \ C_2 \leq C_4 \ \wedge \ C_3 \leq C_5 \ \wedge \ C_3 \leq C_6$$

References

[1] A.-C. Caron, J.-L. Coquidé, and M. Dauchet. Encompassment properties and automata with constraints. In *5th Int. Conference on Rewriting Techniques and Applications*, volume 690 of *LNCS*, pages 328–342, 1993.

[2] A.-C. Caron, F. Seynhaeve, S. Tison, and M. Tommasi. Deciding the satisfiability of quantifier free formulae on one-step rewriting. In *10th Int. Conference on Rewriting Techniques and Applications*, volume 1631 of *LNCS*, pages 103–117, 1999.

[3] H. Comon. Completion of rewrite systems with membership constraints. In *Coll. on Automata, Languages and Programming*, volume 623 of *LNCS*, 1992.

[4] M. Egg, J. Niehren, P. Ruhrberg, and F. Xu. Constraints over lambda-structures in semantic underspecification. In *Proc. of COLING/ACL*, pages 253–359, 1998.

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

[6] J. Lévy. Linear second order unification. In *7th Int. Conference on Rewriting Techniques and Applications*, volume 1103 of *LNCS*, pages 332–346, 1996.

[7] J. Levy and J. Agust. Bi-rewriting systems. *Journal of Symbolic Computation*, 22(3):279–314, Sept. 1996.

[8] S. Limet and P. Réty. A new result about the decidability of the existential one-step rewriting theory. In *10th Int. Conference on Rewriting Techniques and Applications*, volume 1631 of *LNCS*, pages 118–132, 1999.

[9] G. Makanin. The problem of solvability of equations in a free semigroup. *Soviet Akad. Nauk SSSR*, 223(2), 1977.

[10] J. Marcinkowski. Undecidability of the first order theory of one-step right ground rewriting. In *8th Int. Conference on Rewriting Techniques and Applications*, volume 1232 of *LNCS*, pages 241–253, 1997.

[11] J. Niehren and A. Koller. Dominance constraints in context unification. In *Third International Conference on Logical Aspects of Computational Linguistics*, Grenoble, France, Dec. 1998. To appear in LNCS.

[12] J. Niehren, M. Pinkal, and P. Ruhrberg. On equality up-to constraints over finite trees, context unification and one-step rewriting. In *14th Int. Conference on Automated Deduction*, volume 1249 of *LNAI*, pages 34–48, 1997.

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

[14] M. Schmidt-Schauß. Unification of stratified second-order terms. Technical Report 12/94, J. W. Goethe Universität, Frankfurt, 1994.

[15] M. Schmidt-Schauß. A unification algorithm for distributivity and a multiplicative unit. *J. of Symbolic Computation*, 22(3):315–344, 1997.

[16] M. Schmidt-Schauß. Decidability of bounded second order unification. Internal Report Frank-11, Universität Frankfurt, Frankfurt, Germany, 1999. Available at `http://www.ki.informatik.uni-frankfurt.de/papers/articles.html`.

[17] M. Schmidt-Schauß and K. Schulz. Solvability of context equations with two context variables is decidable. In *16th Int. Conference on Automated Deduction*, volume 1632 of *LNCS*, 1999.

[18] F. Seynhaeve, M. Tommasi, and R. Treinen. Grid structures and undecidable constraint theories. In *Theory and Practice of Software Development*, volume 1214 of *LNCS*, pages 357–368, 1997. Extended Version to appear in *Theoretical Computer Science*.

[19] R. Treinen. The first-order theory of one-step rewriting is undecidable. In *7th Int. Conference on Rewriting Techniques and Applications*, volume 1103 of *LNCS*, pages 276–286, 1996. Extended Version in *Theoretical Computer Science* 208, Nov. 1998, pp. 149-177.

[20] S. Vorobyov. The first-order theory of one step rewriting in linear noetheran systems is undecidable. In *8th Int. Conference on Rewriting Techniques and Applications*, volume 1232 of *LNCS*, pages 254–268, 1997.