Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 188 (2007) 105–115
www.elsevier.com/locate/entcs

Proving Termination of Context-Sensitive Rewriting with MU-TERM

Beatriz Alarcón, Raúl Gutiérrez, José Iborra and Salvador Lucas[1,2]

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Valencia, Spain

Abstract

Context-sensitive rewriting (CSR) is a restriction of rewriting which forbids reductions on selected arguments of functions. Proving termination of CSR is an interesting problem with several applications in the fields of term rewriting and programming languages. Several methods have been developed for proving termination of CSR. The new version of MU-TERM which we present here implements all currently known techniques. Furthermore, we show how to combine them to furnish MU-TERM with an *expert* which is able to automatically perform the termination proofs. Finally, we provide a first experimental evaluation of the tool.

Keywords: Context-sensitive rewriting, term rewriting, program analysis, termination.

1 Introduction

Restrictions of rewriting can eventually *achieve* termination of rewriting computations by pruning all infinite rewrite sequences issued from every term. However, such kind of improvements can be difficult to prove. Context-sensitive rewriting (*CSR* [17,18]) is a restriction of rewriting which is useful for describing semantic aspects of programming languages (e.g., Maude, OBJ2, OBJ3, or CafeOBJ) and analyzing termination of the corresponding programs (see [8,9,13,18,22] for further motivation). In *CSR*, a *replacement map* μ discriminates, for each symbol of the signature, the argument positions $\mu(f)$ on which rewritings are allowed. In this way, for a given Term Rewriting System (TRS), we obtain a restriction of the rewrite relation which

[1] This work has been partially supported by the EU (FEDER) and the Spanish MEC, under grants TIN 2004-7943-C04-02 and HA 2006-0007, the Generalitat Valenciana under grant GV06/285, and the ICT for EU-India Cross-Cultural Dissemination ALA/95/23/2003/077-054 project. Beatriz Alarcón was partially supported by the Spanish MEC under FPU grant AP2005-3399.

[2] Email: {balarcon,rgutierrez,jiborra,slucas}@dsic.upv.es

doi:10.1016/j.entcs.2007.05.041

we call *context-sensitive rewriting*. A TRS $\mathcal{R}$ together with a replacement map μ is often called a CS-TRS and written $(\mathcal{R}, \mu)$.

Proving termination of *CSR* is an interesting problem with several applications in the fields of term rewriting and programming languages (see [22]). There are two main approaches to prove termination of a CS-TRS $(\mathcal{R}, \mu)$:

- *direct proofs* use adapted versions of term orderings such as RPOs and polynomial orderings to compare the left- and right-hand sides of the rules [4,11,20,21]; and
- *transformations* which obtain a transformed TRS $\mathcal{R}^{\mu}_{\Theta}$ (where Θ represents the transformation). If we are able to prove *termination* of $\mathcal{R}^{\mu}_{\Theta}$ (using the standard methods), then termination of the CS-TRS is ensured (see [22] for a recent survey).

MU-TERM was the first tool implementing techniques for proving termination of *CSR* [19]. The tool is available here:

http://www.dsic.upv.es/~slucas/csr/termination/muterm

Nowadays, the tool AProVE [15] also accepts context-sensitive termination problems specified in the TPDB format [3]. However, AProVE's proofs of termination of *CSR* are based on using transformations (i.e., no direct proof method is currently available). The new version of MU-TERM which we present here implements all currently known techniques. The new contributions which we report in this paper are the following:

(i) We have implemented the *context-sensitive recursive path ordering* described in [4].

(ii) We have implemented the *context-sensitive dependency pairs approach* described in [2].

(iii) On the basis of recent theoretical and experimental results (see [22]), we have developed a termination expert for *CSR* which combines the different existing techniques for proving termination of *CSR* without any interaction with the user.

Finally, we want to mention that the Maude Termination Tool [9]:

http://www.lcc.uma.es/~duran/MTT

which transforms proofs of termination of Maude programs into proofs of termination of *CSR* uses MU-TERM's expert as an auxiliary tool.

We assume a basic knowledge about term rewriting, see [24] for missing definitions and more information. In Section 2 we briefly describe the new features which have been added to MU-TERM. Section 3 discusses the termination expert. Section 4 provides an experimental evaluation of the new version of MU-TERM. Section 5 concludes and discusses future work.

[3] See http://www.lri.fr/~marche/tpdb/format.html

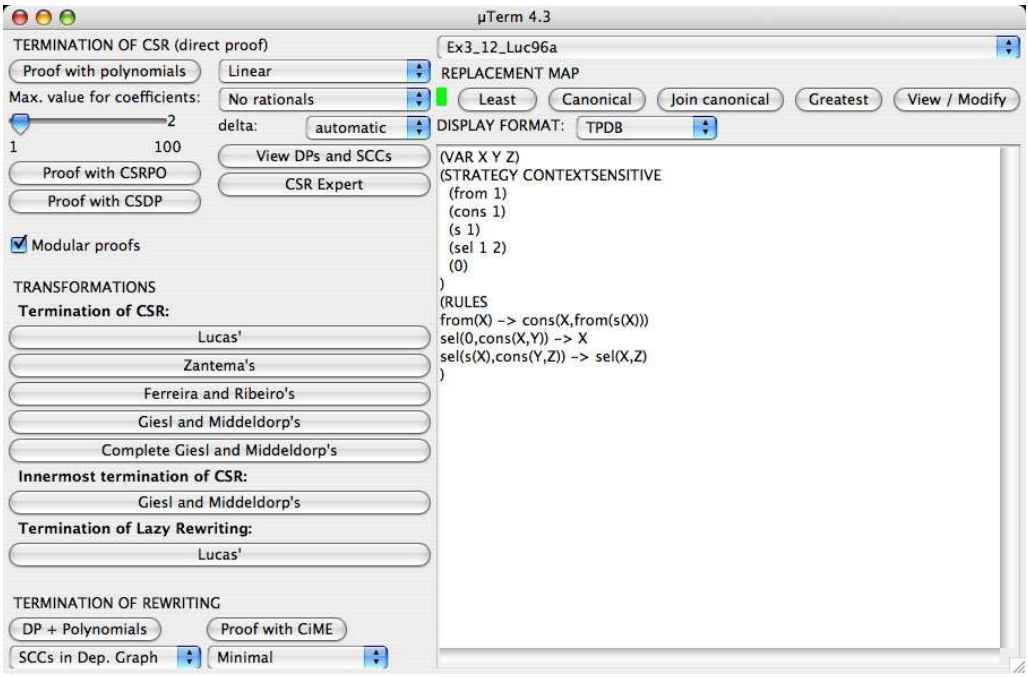

Fig. 1. Screenshot of the main window of MU-TERM 4.3

2 New features in MU-TERM

MU-TERM is written in Haskell [4], and wxHaskell [5] has been used to develop the graphical user interface. The system consists of more than 45 Haskell modules containing more than 14000 lines of code. Compiled versions in several platforms (Linux, Mac OSX, and Windows) and instructions for the installation are available on the MU-TERM WWW site. A recent hybrid (Haskell/C#) version of the tool is also available for the .NET platform [3].

MU-TERM has a graphical user interface (see Figure 1) whose details (menu structure, supported formats, etc.) are given in [19]. Let us briefly recall the main features of the tool.

- Modularity: If the *modular proofs* are activated, then MU-TERM attempts a *safe* decomposition of the TRS in such a way that the components satisfy the modularity requirements described in [10]. If it succeeds in performing a non-trivial decomposition (i.e., MU-TERM obtains more than one component), then individual proofs of termination of *CSR* are attempted for each component.
- Direct methods: MU-TERM implements the use of polynomial interpretations as described in [20,21]. An interesting feature of MU-TERM is that it generates polynomial interpretations with *rational* coefficients.
- Transformations: MU-TERM also implements a number of transformations for proving termination of *CSR* (see [13,22]).

[4] See http://haskell.org/.

[5] See http://wxhaskell.sourceforge.net.

In the following, we briefly describe the new features implemented in the current version of MU-TERM.

2.1 Context-Sensitive Recursive Path Ordering (CSRPO)

CSRPO extends the recursive path ordering (RPO [7]) to context-sensitive terms [4]. The first idea that comes in mind to extend RPO to *CSR* (CSRPO) is *marking* the symbols which are in blocked positions and consider them smaller than the active ones. Therefore, terms in blocked positions become smaller. However, marking all symbols in non-replacing positions can unnecessarily weaken the resulting ordering. Thus, in addition to the usual precedence [6] $\succeq_{\mathcal{F}}$ on the symbols of the signature $\mathcal{F}$ of the TRS, a *marking map*, denoted by $\mathtt{m}$, is also used. The marking map defines, for every symbol and every blocked position, the set of symbols that should be marked. By $\underline{\mathcal{F}}$ we denote the set of marked symbols corresponding to $\mathcal{F}$. Given a k-ary symbol f in $\mathcal{F} \cup \underline{\mathcal{F}}$ and $i \in \{1, \ldots, k\}$, a marking map $\mathtt{m}$ provides the subset of symbols in $\mathcal{F}$ that should be marked, i.e. $\mathtt{m}(f, i) \subseteq \mathcal{F}$. Marking maps are intended to mark *only* blocked arguments, i.e., $\mathtt{m}(f, i) = \varnothing$ if $i \in \mu(f)$ for all $f \in \mathcal{F}$. In this way, we mark only the necessary symbols (in blocked positions), see [4] for a thorough discussion.

Example 2.1 Consider the following TRS $\mathcal{R}$:

```
from(X) -> cons(X,from(s(X)))
sel(0,cons(X,Y)) -> X
sel(s(X),cons(Y,Z)) -> sel(X,Z)
```

together with $\mu(\mathtt{cons}) = \mu(\mathtt{s}) = \mu(\mathtt{from}) = \{1\}$ and $\mu(\mathtt{sel}) = \{1, 2\}$. The μ-termination of $\mathcal{R}$ can be proved by the CSRPO induced by the following precedence and marking map (computed by MU-TERM, see Figure 2):

$$\mathtt{sel} \succ_{\mathcal{F}} \mathtt{from} \succ_{\mathcal{F}} \mathtt{cons} \succ_{\mathcal{F}} \mathtt{s}$$

$$\mathtt{m}(\mathtt{cons}, 2) = \mathtt{m}(\underline{\mathtt{cons}}, 2) = \{\mathtt{from}\}, \quad \mathtt{m}(\underline{\mathtt{from}}, 1) = \varnothing$$

and lexicographic status for all function symbols.

Although the μ-termination of $\mathcal{R}$ in Example 2.1 can be proved by using the following polynomial interpretation:

$$[\mathtt{from}](X) = 3X + 2 \qquad [\mathtt{cons}](X, Y) = X + \tfrac{1}{3}Y \qquad [0] = 0$$
$$[\mathtt{s}](X) = 2X + 1 \qquad [\mathtt{sel}](X, Y) = 2X^2Y + X + Y + 1$$

the proof using CSRPO is much faster.

2.2 Context-Sensitive Dependency Pairs (CSDPs)

Recently, the *dependency pairs approach* [1], one of the most powerful techniques for proving termination of rewriting, has been generalized to be used in proofs of

[6] By a precedence, we mean a reflexive and transitive relation.

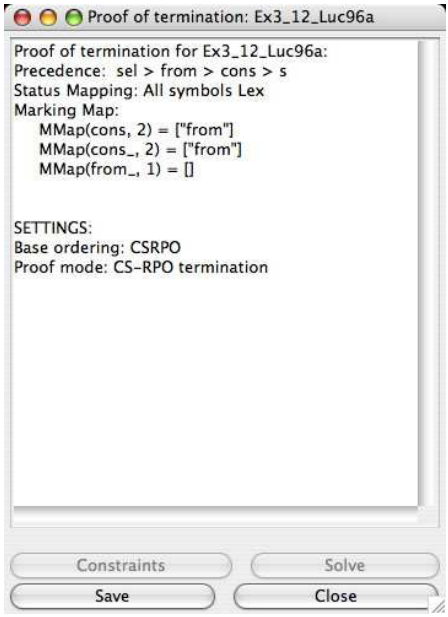

Fig. 2. Termination of *CSR* usingCSRPO

termination of *CSR* [2].

Roughly speaking, given a TRS $\mathcal{R}$, the dependency pairs $u \to v$ associated to $\mathcal{R}$ conform a new TRS $\mathsf{DP}(\mathcal{R})$ which (together with $\mathcal{R}$) determines the so-called *dependency chains* whose finiteness or infiniteness characterize termination of $\mathcal{R}$. The dependency pairs can be presented as a *dependency graph*, where the nodes of the graph are dependency pairs and the absence of infinite chains can be analyzed by considering the *cycles* in the graph. Two dependency pairs $u \to v$ and $u' \to v'$ in the graph are connected by an arc if there is a substitution σ which makes possible a (possibly empty) rewrite sequence (in $\mathcal{R}$) from $\sigma(v)$ to $\sigma(u')$. These ideas are generalized (with a number of non-trivial changes) to *CSR*.

Example 2.2 Consider the following non-terminating TRS $\mathcal{R}$ borrowing the well-known Toyama's example [12, Example 1]:

```
f(a,b,X) -> f(X,X,X)      c -> a      c -> b
```

together with $\mu(\mathtt{f}) = \{3\}$. The only dependency pair for this system is:

```
F(a,b,X) -> F(X,X,X)
```

where `F` is a 'marked' version (often called a *tuple symbol*) of `f` and we further assume that $\mu(\mathtt{F}) = \{3\}$. It is not difficult to see that there is no substitution σ which is able to originate a (possibly empty) *context-sensitive* rewrite sequence (with $\mathcal{R}$!) from $\sigma(\mathtt{F(X,X,X)})$ to $\sigma(\mathtt{F(a,b,X)})$. The replacement restriction $\mu(\mathtt{F}) = \{3\}$ is essential for this. Furthermore, this fact can be easily checked as explained in [2] and so it is implemented in MU-TERM.

A proof of μ-termination of $\mathcal{R}$ in Example 2.2 is *not* possible by using either CSRPO or polynomials with non-negative coefficients (see [11]). Also, as shown by Giesl and Middeldorp (see also [13]), among all the existing transformations for

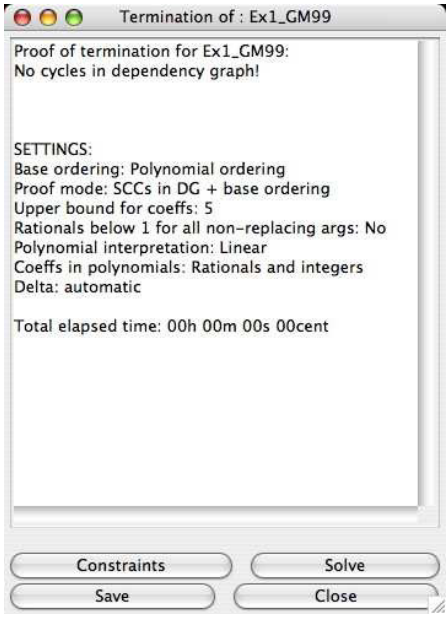

Fig. 3. Termination of *CSR* using dependency pairs

proving termination of *CSR*, only the *complete* Giesl and Middeldorp's transformation [13] (yielding a TRS $\mathcal{R}_C^\mu$) could be used in this case, but no concrete proof of termination for $\mathcal{R}_C^\mu$ is known yet. Furthermore, $\mathcal{R}_C^\mu$ has 13 dependency pairs and the dependency graph contains many cycles. In contrast, the CS-TRS has only *one* context-sensitive dependency pair and the corresponding dependency graph has *no* cycle! Thus, a direct and automatic proof of μ-termination of $\mathcal{R}$ is easy now (see Figure 3).

Although the subterms in the right-hand sides of the rules which are considered to build the context-sensitive dependency pairs are *μ-replacing* terms, considering *only non-variable subterms* (as in Arts and Giesl's approach [1]) is *not* sufficient to obtain a correct approximation. As discussed in [2], in general we also need to consider dependency pairs with *variables* in the right-hand sides.

Example 2.3 Consider the TRS $\mathcal{R}$ [26, Example 5]:

```
if(true,X,Y) -> X        f(X) -> if(X,c,f(true))
if(false,X,Y) -> Y
```

with $\mu(\mathtt{if}) = \{1, 2\}$. There are two dependency pairs:

```
F(X) -> IF(X,c,f(true))
IF(false,X,Y) -> Y
```

with μ extended by $\mu(\mathtt{F}) = \{1\}$ and $\mu(\mathtt{IF}) = \{1, 2\}$.

A direct and automatic proof of μ-termination of $\mathcal{R}$ is possible with CSDPs by using an auxiliary polynomial ordering generated by a linear polynomial interpretation (computed by MU-TERM, see Figure 4).

A proof of μ-termination of $\mathcal{R}$ in Example 2.3 is *not* possible by using CSRPO. Furthermore, the μ-termination of $\mathcal{R}$ cannot be proved by using a polynomial or-

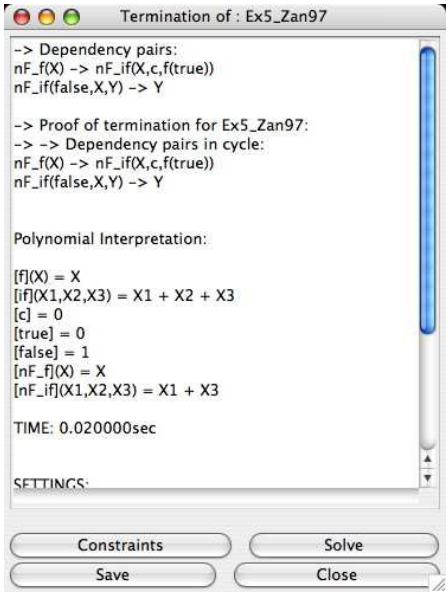

Fig. 4. Termination of *CSR* using dependency pairs and polynomials

dering based on a linear polynomial interpretation.

3 Automatically proving termination of *CSR* with MU-TERM

On the basis of recent theoretical and experimental results, we have developed a *termination expert* for *CSR* which combines the different existing techniques in a sequence of proof attempts which do not require any user interaction. The sequence of techniques which are tried by the expert is as follows:

(i) Context-sensitive dependency pairs with auxiliary polynomial orderings based on polynomial interpretations using either:
  (a) linear interpretations whose coefficients are taken from (1) $\{0,1\}$, (2) $\{0,1,2\}$, or (3) $\{0,\frac{1}{2},1,2\}$, in this order; or
  (b) simple-mixed interpretations linear whose coefficients are taken from (1) $\{0,1\}$, (2) $\{0,1,2\}$, or (3) $\{0,\frac{1}{2},1,2\}$, again in this order.

(ii) Context-sensitive recursive path ordering.

(iii) Polynomial orderings generated from either linear or simple-mixed polynomial interpretations whose coefficients are rational numbers of the form $\frac{p}{q}$ where $0 \leq p,q \leq 5$ and $q > 0$.

(iv) Transformations which obtain a TRS whose termination is proved by using the standard dependency pairs approach [1]. The transformations are attempted according to the decision tree in Figure 5 (explained below).

In the following, we motivate some of the choices we made for obtaining the concrete configuration of the previous sequence.

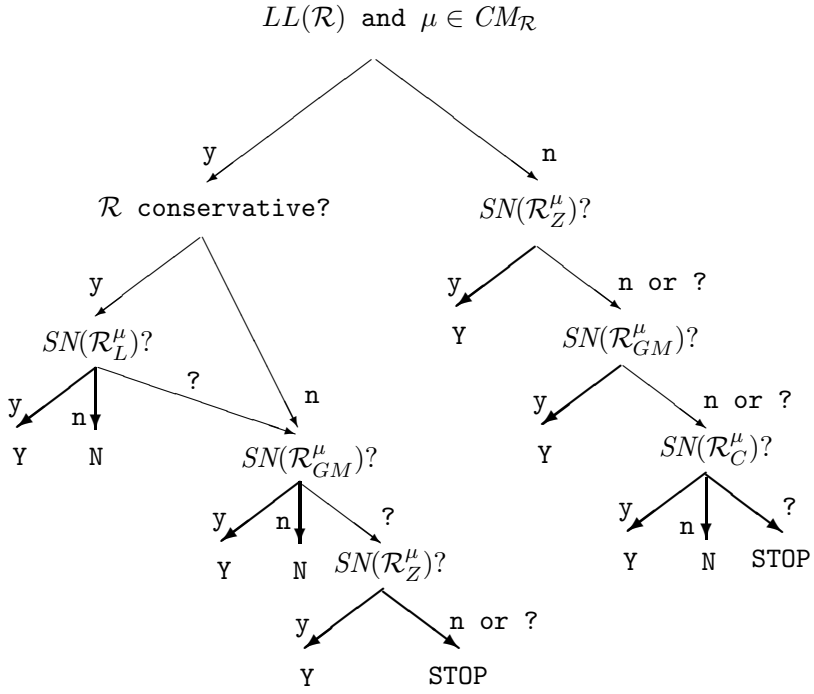

Fig. 5. Decision graph for proving termination of *CSR* by transformation

3.1 Use of polynomial interpretations

As shown in [21,23], the use of rational (or real) coefficients in polynomial interpretations can be helpful to achieve proofs of termination of (context-sensitive) rewriting. In this setting, in order to obtain a proof of μ-termination of a TRS $\mathcal{R} = (\mathcal{F}, R)$, we use *parametric* polynomial interpretations for the symbols $f \in \mathcal{F}$, whose indeterminate coefficients are intended to be *real* (or *rational*) instead of natural or integer numbers. The termination problem is rephrased as a set of polynomial constraints on the indeterminate coefficients. This set of constraints is intended to be solved *in the domain of the real numbers*. Although such polynomial constraints over the reals are decidable [25], the difficulty of the procedure depends on the degree and composition of the parametric polynomials that we use for this. As in [6], we consider classes of polynomials which are well-suited for automatization of termination proofs: linear and simple-mixed polynomial interpretations.

The automatic generation of rational coefficients can be computationally expensive. For instance, MU-TERM manages rational (nonnegative) coefficients $c \in \mathbb{Q}$ in polynomial interpretations as pairs numerator/denominator, i.e., $c = \frac{p}{q}$, where $p, q \in \mathbb{N}$ and $q > 0$. Thus, each rational coefficient c involves *two* integers. This leads to a huge search space in the corresponding constraint solving process [6,21]. For this reason, MU-TERM is furnished with three main *generation modes* [21]:

(i) *No rationals*: here, no rational coefficient is allowed.

(ii) *Rationals and integers*: here, since rational coefficients are intended to introduce non-monotonicity, we only use them with arguments $i \notin \mu(f)$.

(iii) *All rationals*: where all coefficients of polynomials are intended to be rational numbers.

These generation modes are orderly used by the expert to try different polynomial interpretations.

Regarding the *range* of the coefficients, we follow the usual practice in similar termination tools, where coefficients are bounded to take values 0, 1, or 2 (see [6,15,16,27]). Note that (as in those related tools) this choice is *heuristic*, usually based on the experience. We do not know of any theoretical or empirical investigation which tries to guide the choice of appropriate bounds for the coefficients depending on the concrete termination problem. From our side, we just added the value $\frac{1}{2}$ which enables a minimal (but still fruitful) use of rational coefficients. Again, these generation modes are orderly used by the expert to try different polynomial interpretations.

3.2 Use of transformations

In [22] we have investigated how to combine the different transformations for proving termination of *CSR*. Figure 5 provides a concrete decision tree for using the different transformations. Here, $LL(\mathcal{R})$ meas that $\mathcal{R}$ is left-linear, $CM_{\mathcal{R}}$ is the set of replacement maps which are not more restrictive than the *canonical* replacement map $\mu^{can}_{\mathcal{R}}$ of the TRS $\mathcal{R}$. This replacement map has a number of interesting properties (see [17,18]) and can be automatically computed for each TRS (for instance, the tool MU-TERM can do that) thus giving the user the possibility of using *CSR* without explicitly introducing hand-crafted replacement restrictions. Finally, $SN(\mathcal{R})$? represeents a check of *termination* of the TRS $\mathcal{R}$. More details can be found in [22].

4 Experimental evaluation

As remarked in the introduction, besides MU-TERM, AProVE is currently the only tool which is able to prove termination of *CSR* by using (non-trivial) transformations. AProVE is currently the most powerful tool for proving termination of TRSs and implements most existing results and techniques regarding DPs and related techniques. AProVE implements a termination expert which successively tries different transformations for proving termination of *CSR* and uses a variety of different and complementary techniques for proving termination of rewriting, see [15,14]. We have considered the (Linux-based, completely automatic) WST'06-version of AProVE and the set of 90 termination problems for *CSR* which have been used in the 2006 termination competition:

`http://www.lri.fr/~marche/termination-competition/2006`

A summary of the benchmarks can be found here:

`http://www.dsic.upv.es/~rgutierrez/muterm/benchmarks.html`

The benchmarks were executed on a PC equipped with an AMD Athlon XP processor at 2.4 GHz and 512 MB of RAM, running Linux (kernel 2.6.12). Both AProVE

and MU-TERM succeeded (running in a completely automatic way and with a time-out of 1 minute) on 56 examples; furthermore, the total elapsed time was almost the same for both tools. The MU-TERM expert used CSDPs in 45 of the 56 cases (80.4%); CSRPO in 7 cases (12.5%), and transformations in only 4 cases (7.1%, three of them using Zantema's transformation and one of them using Giesl and Middeldorp's incomplete transformation).

5 Conclusions and Future work

We have presented MU-TERM, a tool for proving termination of *CSR*. The tool has been improved with the implementation of new direct techniques for proving termination of *CSR* (the context-sensitive dependency pairs and the context-sensitive recursive path orderings) and an 'expert' for automatically proving termination of *CSR*. The new features perform quite well and have been shown useful in comparison with previously implemented techniques.

Future extensions of the tool will address the problem of efficiently using negative coefficients in polynomial interpretations (see [21] for further motivation). More research is also necessary to make the use of rational coefficients in proofs of termination much more efficient.

The current implementation of CSRPO is based on an ad-hoc incremental constraint solver which could be improved in many different directions. We plan to explore the reduction of the problem to a SAT-solving format, as described in [5]. We also plan to develop algorithms to solve polynomial constraints over the reals yielding exact (but not necessarily rational) solutions.

Finally, we want to improve the generation of reports and the inclusion of new, richer formats for input systems (e.g., Conditional TRSs, Many sorted TRSs, TRSs with AC symbols, etc.).

References

[1] T. Arts and J. Giesl. Termination of Term Rewriting Using Dependency Pairs *Theoretical Computer Science*, 236:133-178, 2000.

[2] B. Alarcón, R. Gutiérrez, and S. Lucas. Context-Sensitive Dependency Pairs. In N. Garg and S. Arun-Kumar, editors *Proc. of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS'06*, LNCS 4337:297-308, Springer-Verlag, Berlin, 2006.

[3] B. Alarcón and S. Lucas. Building .NET GUIs for Haskell applications. In *Proc. of .NET'06*, pages 57-66, University of West-Bohemia, 2006.

[4] C. Borralleras, S. Lucas, and A. Rubio. Recursive Path Orderings can be Context-Sensitive. In A. Voronkov, editor *Proc. of 18th International Conference on Automated Deduction, CADE'02*, LNAI 2392:314-331, Springer-Verlag, Berlin, 2002.

[5] M. Codish, V. Lagoon, and P. Stuckey . Solving Partial Order Constraints for LPO Termination. In F. Pfenning, editor, *Proc. of 17th International Conference on Rewriting Techniques and Applications, RTA'06*, LNCS 4098, 2006.

[6] E. Contejean, C. Marché, A.-P. Tomás, and X. Urbain. Mechanically proving termination using polynomial interpretations. *Journal of Automated Reasoning*, 32(4):315-355, 2006.

[7] N. Dershowitz. Orderings for term rewriting systems. *Theoretical Computer Science*, 17(3):279–301, 1982.

[8] F. Durán, S. Lucas, J. Meseguer, C. Marché, and X. Urbain. Proving Termination of Membership Equational Programs. In P. Sestoft and N. Heintze, editors, *Proc. of ACM SIGPLAN 2004 Symposium on Partial Evaluation and Program Manipulation, PEPM'04*, pages 147-158, ACM Press, New York, 2004.

[9] F. Durán, S. Lucas, J. Meseguer, C. Marché, and X. Urbain. Proving Operational Termination of Membership Equational Programs. *Higher-Order and Symbolic Computation*, to appear, 2007.

[10] B. Gramlich and S. Lucas. Modular termination of context-sensitive rewriting. In C. Kirchner, editor, *Proc. of 4th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, PPDP'02*, pages 50-61, ACM Press, New York, 2002.

[11] B. Gramlich and S. Lucas. Simple termination of context-sensitive rewriting. In B. Fischer and E. Visser, editors, *Proc. of 3rd ACM SIGPLAN Workshop on Rule-Based Programming, RULE'02*, pages 29-41, ACM Press, New York, 2002.

[12] J. Giesl and A. Middeldorp. Transforming Context-Sensitive Rewrite Systems. In P. Narendran and M. Rusinowitch, editors, *Proc. of 10th International Conference on Rewriting Techniques and Applications, RTA'99*, LNCS 1631:271-285, Springer-Verlag, Berlin, 1999.

[13] J. Giesl and A. Middeldorp. Transformation techniques for context-sensitive rewrite systems. *Journal of Functional Programming*, 14(4): 379-427, 2004.

[14] J. Giesl, R. Thiemann, and P. Schneider-Kamp. The Dependency Pair Framework: Combining Techniques for Automated Termination Proofs. In F. Baader and A. Voronkov, editors *Proc. of 11th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning, LPAR'04*, LNCS 3452:301-331, Springer-Verlag, Berlin, 2004.

[15] J. Giesl, P. Schneider-Kamp, and R. Thiemann. AProVE 1.2: Automatic Termination Proofs in the Dependency Pair Framework. In U. Furbach and N. Shankar, editors, *Proc. of Third International Joint Conference on Automated Reasoning, IJCAR'06*, LNAI 4130:281-286, Springer-Verlag, Berlin, 2006. Available at `http://www-i2.informatik.rwth-aachen.de/AProVE`.

[16] N. Hirokawa and A. Middeldorp. Tyrolean Termination Tool. In J. Giesl, editor, *Proc. of 16th International Conference on Rewriting Techniques and Applications, RTA'05*, LNCS 3467:175-184, 2005.

[17] S. Lucas. Context-sensitive computations in functional and functional logic programs. *Journal of Functional and Logic Programming*, 1998(1):1-61, January 1998.

[18] S. Lucas. Context-sensitive rewriting strategies. *Information and Computation*, 178(1):293-343, 2002.

[19] S. Lucas. MU-TERM: A Tool for Proving Termination of Context-Sensitive Rewriting In V. van Oostrom, editor, *Proc. of 15h International Conference on Rewriting Techniques and Applications, RTA'04*, LNCS 3091:200-209, Springer-Verlag, Berlin, 2004.

[20] S. Lucas. Polynomials for proving termination of context-sensitive rewriting. In I. Walukiewicz, editor, *Proc. of 7th International Conference on Foundations of Software Science and Computation Structures, FOSSACS'04*, LNCS 2987:318-332, Springer-Verlag, 2004.

[21] S. Lucas. Polynomials over the reals in proofs of termination: from theory to practice. *RAIRO Theoretical Informatics and Applications*, 39(3):547-586, 2005.

[22] S. Lucas. Proving termination of context-sensitive rewriting by transformation. *Information and Computation*, 204(12):1782-1846, 2006.

[23] S. Lucas. On the relative power of polynomials with real, rational, and integer coefficients in proofs of termination of rewriting. *Applicable Algebra in Engineering, Communication and Computing*, 17(1):49-73, 2006.

[24] E. Ohlebusch. Advanced Topics in Term Rewriting. Springer-Verlag, Berlin, 2002.

[25] A. Tarski. A Decision Method for Elementary Algebra and Geometry. Second Edition. University of California Press, Berkeley, 1951.

[26] H. Zantema. Termination of Context-Sensitive Rewriting. In H. Comon, editor, *Proc. of 8th International Conference on Rewriting Techniques and Applications, RTA'97*, LNCS 1232:172-186, Springer-Verlag, Berlin, 1997.

[27] H. Zantema. TORPA: Termination of String Rewriting Systems. *Journal of Automated Reasoning*, 34:105-39, 2006.