The Termination Competition

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

The Termination Competition

Claude Marché[1,2] and Hans Zantema[3]

[1] INRIA Futurs, ProVal, Parc Orsay Université, F-91893
[2] Lab. de Recherche en Informatique, Univ Paris-Sud, CNRS, Orsay, F-91405
[3] Department of Computer Science, Technische Universiteit Eindhoven
P.O. Box 513, 5600 MB, Eindhoven, The Netherlands
Claude.Marche@inria.fr, h.zantema@tue.nl

Abstract. Since 2004, a Termination Competition is organized every year. This competition boosted a lot the development of automatic termination tools, but also the design of new techniques for proving termination. We present the background, results, and conclusions of the three first editions, and discuss perspectives and challenges for the future.

1   Motivation and History

In a landmark paper in 1970, Manna & Ness [1] proposed a criterion for proving termination of rewrite systems, based on *reduction* orderings. Since then, many techniques for proving termination have been proposed, by providing means of defining classes of reduction orderings: path orderings, polynomial interpretations, etc. A few implementations were developed [2,3,4] but practical results in proposed sets of benchmark problems [5,6] were quite poor.

A disruptive progress came up in 1997, with the *Dependency Pair* criteria proposed by Arts & Giesl [7], allowing to prove termination with a larger class of orderings than reduction orderings. This brought up a new interest towards automation of termination proofs. Since around 2000 several tools were developed for this goal, and in 2003, for the Workshop on Termination in Valencia, Albert Rubio organized a special session for comparing tools. Tool authors gathered, proposed a few challenging examples, and each of them manually ran their own tool and told what they were able to prove. Participants were AProVE [8], Cariboo [9], CiME [10], MatchBox [11], Temptation [12] and TTT [13] on rewrite systems problems; and TALP [14], TerminWeb [15] and Hasta-La-Vista [16] for logic programs. MatchBox was only dealing with string rewriting.

Stimulated by the enthusiasm of the participants it was decided to organize an annual competition. Participants agreed on a common syntax of problems, in order to build a shared database called the TPDB (Termination Problem Data Base, http://www.lri.fr/~marche/tpdb/). The main idea was that the competition must be run fully automatically, to demonstrate the ability of tools to solve termination problems, without requiring any expertise use, such as setting clever options or parameters as what the author's tool can do. C. Marché took care of the organization and the development of the required utilities for running the competition automatically and making results available online. The main objectives for such a competition were and remain:

- to stimulate research in this area, shifting emphasis towards automation,
- to provide a standard to compare termination techniques.

The first full competition in this style ran in May 2004, the week before the Workshop on Termination in Aachen, where the results were reported. There were three categories, corresponding to different input syntax: term rewriting (TRS, 5 participating tools), string rewriting (SRS, 5 tools) and logic programs (LP, 2 tools). With respect to the first event in 2003, there was only one new tool: TORPA [17], specialized to SRSs. Some other tools were not able to participate in this automatic competition: Termptation, TerminWeb and Hasta-La-Vista. AProVE had a new module allowing it to participate to the LP category. The TRS category has been subdivided into 5 sub-categories corresponding to standard rewriting, rewriting modulo theories, innermost strategy, context-sensitive strategy and conditional rewriting.

In 2005, Hans Zantema joined the organization of the competition, and the second competition ran in April 2005, the week before RTA in Nara. This time there were only two categories: TRS and SRS. In TRS category, Cariboo did not participate but 2 new tools joined: TPA [18] and TEPARLA [19], leading to a total of 6 tools. They also participated to the SRS category, together with the new tool JamBox [20], thus there were 8 participants. A new sub-category for relative termination was introduced, both for term rewriting and string rewriting. In the meantime, the size of the TPBD grew significantly. Several termination proofs were given for systems where the 2004 versions failed, showing improvements of the tools. The ranking of the tools were quite similar as the 2004 edition.

In 2006, the competition was held in June, two months before WST and RTA in Seattle. In the TRS category, there were eight participants: TTT was replaced by a variation called TTTbox [21], and two new tools joined: MU-Term [22] and JamBox [20] (which was only in SRS category in the previous year). Nine tools were in the SRS category: a new tool MultumNonMulta [23] joined. This year, the rankings have been significantly modified.

In the following, we first describe in Section 2 the rules of the competition. Then in Section 3 we summarize the results and comment their evolution over the years. We draw some conclusions and perspectives in Section 4 and we provide a list of challenges for future competitions in Section 5.

2   The Rules

The following rules were applied to the 2006 termination competition.

- Submission of new problems for TPDB is open until a few weeks before the competition, when this new TPDB is publicly available for testing and tuning the tools.
- Just before the competition participants submit
  - final versions of the tools, and
  - secret problems, up to ten per participant per category, that are added to the version of TPDB used for the competition, but not accessible for other participants before the competition.

- All tools apply on all problems in the corresponding TPDB categories, all
  on the same machine. The required output of every tool is
    - "YES", followed by the text of a termination proof, or
    - "NO", followed by the text of a non-termination proof, or
    - anything else, interpreted as "DON'T KNOW".
- Execution of more than one minute for any tool on any termination problem
  causes a time-out, interpreted as "DON'T KNOW".
- All results are reported on-line, including generated proof text, and statistics
  about scores and running time.
- Any tool generating a wrong answer is "disqualified" (see Section 3.4).
- There are no formal rules and consequences of being a "winner", apart from
  the honour of having a high or the highest score in some (sub)category.

These rules were designed in such a way that participants also being organizer
had no advantage of being organizer. In 2006, categories were subdivided in the
following eight subcategories:

- Standard term rewriting.
- Innermost term rewriting. This means that only rewrite steps are allowed
  for which all proper subterms of the redex are in normal form.
- Context-sensitive strategy. This means that for every operation symbol it is
  specified under which position rewriting is allowed.
- Term rewriting modulo theory. This means that apart from the rewrite rules
  also equations are specified (usually associativity and commutativity) mod-
  ulo which rewriting is done.
- Relative termination of term rewriting. This means that two rewrite systems
  R, S are specified for which termination of $\to_S^* \cdot \to_R \cdot \to_S^*$ has to be proved.
- Standard string rewriting. This coincides with standard term rewriting in
  which all symbols have arity one.
- Relative termination of string rewriting.
- Logic programs.

3   Competitions Results

We present here a summary of the results in the three first editions of the compe-
tition. We only present and discuss on the two main categories: standard string
rewriting and standard term rewriting. More detailed results including all termi-
nation problems, all generated proofs, executable code of the tools, and measured
execution times and statistics are available from `http://www.lri.fr/~marche/`
`termination-competition/`.

3.1   SRS Category

The following table summarizes the results of the SRS category, for the three
editions of the competition. In the second column we give the number of problems
submitted to tools. Then for both YES and NO answers, we give the total number

of problems solved, then the three best tools with their respective score with respect to this total. We also give the number of problems that remain unsolved by any tool.

year	# pbs	answer	overall	1st		2nd		3rd	
2004	104	yes	89	TORPA	88	AProVE	87	MatchBox	49
		no	7	MatchBox	7	TORPA	6	AProVE	5
		unk.	8						
2005	153	yes	139	TORPA	126	AProVE	114	JamBox	102
		no	9	JamBox and MatchBox			9	TORPA	5
		unk.	5						
2006	322	yes	263	JamBox	251	TORPA	201	MatchBox	176
		no	25	JamBox	25	AProVE and MatchBox			12
		unk.	34						

After its short win in 2004, TORPA has been a clear winner in 2005. But a big surprise happened in 2006: the new JamBox tool was significantly improved and became clearly more efficient than TORPA. The reason is the new technique of matrix interpretations implemented in JamBox, discussed below. It is also noticeable that the number of problems significantly grew in 2006, incorporating sets of problems from new research groups. This clearly made the competition exciting, and the next competition will be interesting to follow.

3.2   TRS Category

year	# pbs	answer	overall	1st		2nd		3rd	
2004	521	yes	426	AProVE	410	TTT	397	CiME	297
		no	22	AProVE	22	MatchBox	21	TTT	12
		unk.	73						
2005	773	yes	588	AProVE	576	TTT	509	TPA	407
		no	94	AProVE	94	MatchBox	80	TEPARLA	15
		unk.	91						
2006	865	yes	686	AProVE	638	JamBox	626	TPA	422
		no	103	AProVE	103	MatchBox	85	TEPARLA	15
		unk.	76						

The clearly visible fact is that the AProVE tool has been constantly the best tool each year, both for proving and disproving termination. However, this should not hide that this required strong improvement each year: for example, the AProVE 2005 version would not have won the 2006 edition. Indeed, the JamBox tool made the 2006 competition very exciting, showing how efficient was the new technique of matrix interpretation implemented in JamBox. It is noticeable too that each year, there were a significant amount of problems not solved by the winner but solved by others.

Finally, notice that although the number of problems increased by 92 between 2005 and 2006, the number of undecided problems decreased by 15: this is a clear evidence that the tools efficiency considerably improved.

3.3   Other Categories

To summarize briefly the other categories: AProVE remained winner in every sub-category, except relative termination which it does not support. JamBox won relative termination subcategories for TRS and SRS in 2006.

In the Logic Programs category, AProVE won both 2004 and 2006 editions. The evolution in this category is poorly significant, it seems that the lack of participants does not encourage efforts in this direction. In other words, the competition cannot reach its goal of stimulating research if existing other tools are not willing to participate.

3.4   Remarks

soundness issue Since all tools execute complicated tasks it is likely that they contain bugs. In 2006, for two tools (CiME and MU-Term) we detected some obviously incorrect generated proofs. The tools have been "disqualified" in the sense that their scores are not taken into account (results above would not be different anyway). We emphasize that it does not imply that all termination proofs generated by the remaining tools are correct: we cannot check all thousands of generated termination proofs. As a long-term objective we see an automatic formal correctness check of the generated proofs.

timing issue Most proofs are found within less than a second. For most tools the average time to find a termination proof was a few seconds. In 2006, we experimented running a second round for problems not solved in a one minute timeout, giving a time limit of five minutes instead. In the category of term rewriting it occurred a few times that a termination proof was found by a tool in the second round where all tools failed in the first round: 3 times for standard rewriting and once for context-sensitive and modulo theory subcategories. In the string rewriting category, the tools JamBox, MultumNonMulta and TPA found termination proofs in a second round where all tools failed in the first round. The total number of these systems was 5, both in the subcategories standard (2) and relative termination (3). The time limit has a small influence on the set of unsolved problems, emphasizing the fact that it is not easy to know when a tool does not solve a problem, whether it is just because of lack of time, or because of an intrinsic insufficiency of the techniques implemented.

non-termination Since the first competition in 2004, it was decided to evaluate the ability of proving non-termination. Before that, no tool except maybe MatchBox was implementing any technique for disproving termination. We emphasize that the competition stimulated both the development of new techniques for disproving termination, and of course implementations (but still not all tools have facilities for this). The non-termination proofs found were all generated by presenting a looping reduction. For logic programs, no tool is able to give non-termination proofs. For the rewriting categories, only very few proofs of non-termination are obtained for other sub-categories than standard rewriting.

challenging problems After the 2004 competition, we tried to designate a
subset of unsolved problems that could represent challenges. We chose some
modifications of Toyama's system, and they were solved in the 2005 compe-
tition. Again, in 2005 we presented termination of the SRS `Zantema-z086`
consisting of the three rules $aa \rightarrow bc$, $bb \rightarrow ac$, $cc \rightarrow ab$ as an open problem,
since no tool solved this system in the two first editions. It was also proposed
as a new problem in the RTA open problem list. This has been first solved
by Hofbauer and Waldmann [24], and without any doubt it stimulated the
development of the new matrix interpretation method [25,26], by which the
problem was automatically proved terminating by JamBox in 2006.

4   Conclusions and Perspectives

The three first edition of the Termination Competition have been really exciting
due to new developments of termination techniques and new implementations.
Dependency graph criteria[7], together with efficient techniques to search for ar-
gument filterings, path orderings and polynomial interpretations, were shown to
be the most useful techniques. The most powerful new technique whose discovery
was motivated by the competition is the matrix method [25,26]. Regarding im-
plementations of this technique, we emphasize that JamBox and MatchBox both
use an external SAT solver for searching for suitable interpretations. AProVE
also uses a SAT solver to search for path orderings and polynomial interpre-
tations [27]. It seems to be a very efficient way to benefit from the very good
progress made by SAT solvers, which maybe indeed due to the existing compe-
tition (`http://www.satcompetition.org/`) of this kind.

We also emphasize that although AProVE won all three competitions in stan-
dard TRS category, it couldn't remained winner each year without major im-
provements in the techniques implemented: handling applicative TRSs [28], poly-
nomials with negative coefficients[29], subterm criterion [30], match-bounds for
term rewriting [31], etc.

We consider that the termination competition has been very successful so far,
justifying annual continuation:

– It provides an objective way to compare the power of various implementa-
  tions and techniques for proving termination.
– New challenges emerge from the competition, stimulating the development
  of new powerful techniques.

As remarked in Section 3.4, an important objective for the future is an auto-
matic formal correctness check of generated proofs. Achieving this both requires
a lot of work and agreement about formats of the proofs. But recent progress has
been made by two research groups using the Coq proof assistant for formalizing
a soundness proof checker: CoLoR [32] and A3PAT/CiME [33]. It is likely that a
sub-category for certified termination proof will appear in the next competition.

The emphasis in the competition is in rewriting rather than termination of
programs. For logic programs, even if recent progresses have been made, the

participating tools restricted to the specific technique of transforming the logic program to a term rewriting system and then prove termination of the latter. We should like to have participation by other tools not focusing on rewriting, such as the TERMINATOR tool [34]. For the next competition we plan to add new categories for functional programs (Haskell, ML) and for some kind of imperative programs. Another aspect, which was forgotten when focusing on rewrite systems, is the support for numerical computations, using for example built-in integers. This was indeed a important originality of the Hasta-La-Vista tool for logic programs. This aspect should be considered again in the future, to support such numerical computations in functional or imperative programs.

5   Challenges

To stimulate further research, we now present some challenging problems. All references in typewriter font refer to TBDB. Right after the end of the 2006 competition, it was emphasized that the SRS `SRS/Waldmann-jw1`

$$aaa \rightarrow bab \qquad bbb \rightarrow aaa$$

was probably the shortest TPDB problem unsolved by any tool. In the meantime, it has been solved by A. Nogin and C. Witty (`http://lists.lri.fr/pipermail/termtools/2006-August/000295.html`) and now can be solved automatically [35].

5.1   Longstanding Open Problems

Challenge 1 (Hercules and Hydra battle). *It is problem* `TRS/D33-33`, *the famous Hercules and Hydra battle. It is the only problem of old benchmarks [5,6] which remain unsolved by any tool.*

$$h(z, e(x)) \rightarrow h(c(z), d(z, x)) \qquad d(z, g(0, 0)) \rightarrow e(0)$$
$$d(z, g(x, y)) \rightarrow g(e(x), d(z, y)) \qquad g(e(x), e(y)) \rightarrow e(g(x, y))$$
$$d(c(z), g(g(x, y), 0)) \rightarrow g(d(c(z), g(x, y)), d(z, g(x, y)))$$

At several occasions, some tool author pretended to be able to solve it, but it never happened during the competition, and it is impossible to know whether it was requiring specific user interaction, or even if it was not a bug in the tools.

Several other TRSs have been proposed in the literature, for various purposes, for which the termination was left as an open problem. In 1997, a TRS for integer arithmetic [36] was presented, with a very complicated ad-hoc proof. It has been shown terminating using MSPO in 2003, and in the same year solved automatically by CiME using dependency pairs modulo AC and polynomial interpretations. In the same article, a large TRS for rational arithmetic is presented, whose termination remains open. In 2000, Deplagne [37] introduced a TRS for sequent calculus modulo, leaving again open its termination. It is now shown

terminating both by APrOVE and CiME. Bonelli proposed a TRS for explicit subsitutions, and was unable to prove its termination. It has been introduced as a secret problem in the 2005 competition (`TRS/secret2005-cime1`), and was first shown to be terminating by TEPARLA. Another system related to explicit substitution is `TRS/Zantema-z10`. Several papers were devoted only to proving termination of this system; a shorter proof based on semantic labelling [38] was first found manually, and later on by TPA.

A very long-standing open termination problem appeared in 1991: a TRS proposed by Cohen and Watson [39] for arithmetic:

Challenge 2 (Cohen-Watson system for arithmetic). *This is the 22-rules system `TRS/secret2006-cime1` of the TPDB. Its termination status is unknown, and is indeed the problem #65 in the RTA List of Open Problems.*

5.2 Semantic Decreasing Argument

A large class of problems which remain unsolved in the TPDB seem to require a decreasingness argument that is "semantic": it depends on a value of a normal form of a subterm (which is more or less a return value of some auxiliary function call). This is especially true for TRSs coming from automatic translations of context-sensitive TRSs, or Maude or OBJ programs. This phenomenon may occur also when dealing with functional or imperative programs.

Challenge 3 (While loop). *As an instance of this problem, we mention `TRS/Zantema06-while`:*

$$f(t, x, y) \rightarrow f(g(x, y), x, s(y)) \qquad g(s(x), 0) \rightarrow t \qquad g(s(x), s(y)) \rightarrow g(x, y)$$

which encodes the obviously terminating loop `while` $x > y$ `do` $y := y + 1$. *Using the semantics, e.g., by semantic labelling [38], a termination proof can be given, but until now no tool could solve it.*

5.3 String Rewriting Systems

In 2006, three series of randomly generated SRSs where added, and those problems are poorly solved by tools, hence those series provide a set of challenging problems. We propose below a challenging problem with a more natural computational content.

Challenge 4 (Power 2 to power 3). *This is problem `SRS/Zantema-z079`:*

$$caa \rightarrow ac \qquad acb \rightarrow adb \qquad ad \rightarrow daaa \qquad bd \rightarrow bc$$

which essentially rewrites 2^n *to* 3^n, *more precisely* $bca^{2n}b$ *rewrites to* $bca^{3n}b$. *No tool could solve it, for any of the three editions of the competition.*

5.4   Non-termination

There are also challenging problems for showing non-termination. An example
of these is `TRS/HofWald-6`:

$$f(f(a, x), y) \rightarrow f(f(x, f(a, y)), a)$$

where a is a constant and x, y are variables, which is not solved by any tool,
although there seems to be a very simple and easy implementable way to ob-
serve non-termination: every ground instance of the right hand side contains an
instance of the left hand side. Another example is the following SRS,

$$al \rightarrow la \qquad ra \rightarrow ar \qquad bl \rightarrow bar \qquad rb \rightarrow lb$$

being `SRS/Zantema-z073`. It is obviously non-terminating since $ba^n lb \rightarrow^* ba^{n+1} lb$
for all n, but no tool can solve it. Generally speaking, techniques should be de-
veloped for automatically proving non-looping non-termination.

Acknowledgements. Special thanks to Albert Rubio who brought up the idea
of a termination competition, and set up the first TPDB. Thanks to all partici-
pants all over the years.

References

1. Manna, Z., Ness, S.: On the termination of Markov algorithms. In: ICSS. pp.
   789–792 (1970)
   `http://perso.ens-lyon.fr/pierre.lescanne/not_accessible.html`
2. Lescanne, P.: Computer experiments with the REVE term rewriting system gen-
   erator. In: Proc. POPL'83 (1983)
3. Forgaard, R., Detlefs, D.: Reve 2.4: A program for generating and analyzing term
   rewriting systems. Massachusetts Institute (1984)
4. Steinbach, J.: Generating polynomial orderings. Information Processing Letters 49,
   85–93 (1994)
5. Dershowitz, N.: 33 examples of termination. In: Comon, H., Jouannaud, J.-P.
   (eds.) Spring School of Theoretical Computer Science. LNCS, vol. 909, pp. 16–
   26. Springer, Heidelberg (1995)
6. Steinbach, J., Kühler, U.: Check your ordering – termination proofs and open
   problems. Technical Report SEKI Report SR-90-25, Univ. Kaiserslautern (1990)
7. Arts, T., Giesl, J.: Termination of term rewriting using dependency pairs. Theo-
   retical Computer Science 236, 133–178 (2000)
8. Giesl, J., Schneider-Kamp, P., Thiemann, R.: The AProVE tool (RWTH, Aachen,
   Germany) `http://aprove.informatik.rwth-aachen.de/`
9. Fissore, O., Gnaedig, I., Kirchner, H.: CARIBOO: A multi-strategy termination
   proof tool based on induction. In: Rubio, A., ed.: WST, pp. 77–79 (2003)
10. Contejean, E., Marché, C., Urbain, X.: The CiME rewrite toolbox (LRI, Orsay,
    France) `http://cime.lri.fr`
11. Waldmann, J.: The MatchBox tool (HTWK, Leipzig, Germany)
    `http://dfa.imn.htwk-leipzig.de/matchbox/`

12. Borralleras, C., Rubio, A.: The termptation tool (Universitat Politecnica de Catalunya, Spain) `http://www.lsi.upc.es/~albert/term.html`
13. Hirokawa, N., Middeldorp, A.: Tyrolean Termination Tool (Innsbruck Universität, Austria) `http://colo6-c703.uibk.ac.at/ttt/`
14. Claves, C., Ohlebusch, E.: The TALP tool (University of Bielefeld, Germany) `http://bibiserv.techfak.uni-bielefeld.de/talp/`
15. Codish, M., Taboch, C.: the TerminWeb tool (Ben Gurion University, Israel) `http://www.cs.bgu.ac.il/~mcodish/TerminWeb/`
16. Serebrenik, A., De Schreye, D.: The Hasta-La-Vista tool (K.U. Leuven, Belgium)
17. Zantema, H.: The TORPA tool (Technische Universteit Eindhoven, The Netherlands) `http://www.win.tue.nl/~hzantema/torpa.html`
18. Koprowski, A.: The TPA tool (Technische Universteit Eindhoven, The Netherlands) `http://www.win.tue.nl/tpa`
19. van der Wulp, J.: The TEPARLA tool (Technische Universteit Eindhoven, The Netherlands) `http://www.win.tue.nl/~hzantema/torpa.html`
20. Endrullis, J.: The JamBox tool (Free University in Amsterdam, The Netherlands) `http://joerg.endrullis.de/`
21. Korp, M.: The TTTbox (Innsbruck Universität, Austria) `http://homepage.uibk.ac.at/~csad2836/TTTbox.html`
22. Lucas, S.: Mu-term, a tool for proving termination of rewriting with replacement restrictions (2003) `http://www.dsic.upv.es/~slucas/csr/termination/muterm/`
23. Hofbauer, D.: The MultumNonMulta tool (Kassel, Germany) `http://www.theory.informatik.uni-kassel.de/~dieter/multum/`
24. Hofbauer, D., Waldmann, J.: Termination of $\{aa \rightarrow bc, bb \rightarrow ac, cc \rightarrow ab\}$. Information Processing Letters 98(4), 156–158 (2006)
25. Hofbauer, D., Waldmann, J.: Termination of string rewriting with matrix interpretations. In: Pfenning, F. (ed.) RTA 2006. LNCS, vol. 4098, Springer, Heidelberg (2006)
26. Endrullis, J., Waldmann, J., Zantema, H.: Matrix interpretations for proving termination of term rewriting. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, Springer, Heidelberg (2006)
27. Codish, M., Schneider-Kamp, P., Lagoon, V., Thiemann, R., Giesl, J.: Sat solving for argument filterings. In: Hermann, M., Voronkov, A. (eds.) LPAR 2006. LNCS (LNAI), vol. 4246, pp. 30–44. Springer, Heidelberg (2006)
28. Giesl, J., Thiemann, R., Schneider-Kamp, P.: Proving and disproving termination of higher-order functions. In: Gramlich, B. (ed.) FroCoS. LNCS (LNAI), vol. 3717, pp. 216–231. Springer, Heidelberg (2005)
29. Hirokawa, N., Middeldorp, A.: Polynomial interpretations with negative coefficients. In: Buchberger, B., Campbell, J.A. (eds.) AISC 2004. LNCS (LNAI), vol. 3249, pp. 185–198. Springer, Heidelberg (2004)
30. Hirokawa, N., Middeldorp, A.: Dependency pairs revisited. In: van Oostrom, V. (ed.) RTA 2004. LNCS, vol. 3091, pp. 249–268. Springer, Heidelberg (2004)
31. Geser, A., Hofbauer, D., Waldmann, J., Zantema, H.: On tree automata that certify termination of left-linear term rewriting systems. Information and Computation 205(4), 512–534 (2007)
32. Blanqui, F.: CoLoR project (INRIA Lorraine, France) `http://color.loria.fr/`
33. Urbain, X., Forest, J., Courtieu, P.: The A3PAT project (Cédric, CNAM, Paris, France) `http://www3.ensiie.fr/~urbain/a3pat`
34. Podelski, A.: the TERMINATOR project (Microsoft Research, Cambridge, UK) `http://research.microsoft.com/TERMINATOR/`

35. Zantema, H., Waldmann, J.: Termination by quasi-periodic interpretations. In: Baader, F. (ed.) RTA'07. LNCS, Springer, Heidelberg (2007)
36. Contejean, E., Marché, C., Rabehasaina, L.: Rewrite systems for natural, integral, and rational arithmetic. In: Comon, H. (ed.) Rewriting Techniques and Applications. LNCS, vol. 1232, Springer, Heidelberg (1997)
37. Deplagne, É.: Sequent Calculus Viewed Modulo. In: Pilière, C. (ed.) ESSLLI Student Session, Univ. Birmingham, pp. 66–76 (2000)
38. Zantema, H.: Termination of term rewriting by semantic labelling. Fundamenta Informaticae 24, 89–105 (1995)
39. Cohen, D., Watson, P.: An efficient representation of arithmetic for term rewriting. In: Book, R.V. (ed.) RTA'91. LNCS, vol. 488, pp. 240–251. Springer, Heidelberg (1991)