The Termination Competition 2006

Claude Marché[1] and Hans Zantema[2]

[1] LRI, Université Paris-sud 11
F-91405 ORSAY Cedex, France
Claude.Marche@lri.fr

[2] Department of Computer Science, Technische Universiteit Eindhoven
P.O. Box 513, 5600 MB, Eindhoven, The Netherlands
h.zantema@tue.nl

Abstract. The third Termination Competition took place in June 2006. We present the background, results and conclusions of this competition.

1 Motivation and history

In the past decades several techniques have been developed to prove termination of programs and rewrite systems. In the late nineties the emphasis in this research shifted towards automation: for a new technique the final goal was not to use it by hand in order to prove termination of a number of systems, but to implement it in such a way that termination proofs could be found fully automatically using a computer. Since around 2000 several tools were developed for this goal. In 2003 the idea came up to organize a competition on these tool by developing an extensive set of termination problems called TPDB (termination problem data base), and run the tools on them and compare the results. The main objectives for such a competition were and are:

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

At the Workshop on Termination in 2003 in Valencia a preliminary competition was organized by Albert Rubio, together with the initial development of TPDB. Stimulated by the enthusiasm of the participants it was decided to organize an annual competition, in which all tools run on one central computer and results are reported on-line. Claude Marché took care of the organization and the development of the required tools.

All details on the termination competition including past editions, rules and all results, are found on http://www.lri.fr/~marche/termination-competition/

The first full competition in this style run 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, string rewriting and logic programs; respectively having 5, 5 and 2 participating tools. Apart from standard rewriting the category of term rewriting had some subcategories: termination modulo a theory, innermost strategy, outermost strategy, context-sensitive rewriting and conditional rewriting. In standard term rewriting the tool AProVE had the highest score of 410 termination proofs, out of 514 rewrite systems, while TTT was second with 397. Not all of the 514 systems were terminating: for 22 of them AProVE found a proof of non-termination. In string rewriting the tool TORPA had the highest score of 88 termination proofs out of 104, directly followed by AProVE with 87. In logic programs and all term rewriting subcategories AProVE had the highest score.

In 2005 Hans Zantema joined the organization of the competition, and the second competition run in April 2005, the week before RTA in Nara. Since there was no Workshop on Termination in 2005, the results of this competition were reported at RTA. This time there were only two categories: term rewriting and string rewriting, respectively having 6 and 8 participating tools. Since string rewriting can be seen as a special case of term rewriting, all tools for term rewriting participated for the string rewriting category too. A new sub-category for relative termination was introduced, both for term rewriting and string rewriting. On the other hand subcategories were restricted to subcategories having at least two participants — otherwise it is hard to speak about a competition. For term rewriting these were standard, relative, innermost and modulo theory; for string rewriting these were standard and relative. In standard term rewriting again the tool AProVE had the highest score of 576 termination proofs, out of 773 rewrite systems, while TTT was again second, with 509. In string rewriting again the tool TORPA had the highest score of 126 termination proofs out of 153, again followed by AProVE, with 114. For both categories the improvements were not only due to extensions of TPDB. Also several termination proofs were given for systems where the 2004 version failed, showing improvements of the tools.

For both term rewriting and string rewriting the tools ending at the first place were the same in 2004 and 2005: AProVE and TORPA.

The same holds for the second place: TTT and AProVE. So one might expect that the strong tools will remain strong in next editions, and it is unlikely for new tools to take over the role of strongest tools. Surprisingly, due to strong new techniques and the strong new tool Jambox the 2006 termination competition resulted in this unlikely scenario. Before going into details first we describe the rules as they applied for the 2006 edition of the competition.

2 The rules

The following rules were applied to the 2006 termination competition. They were announced several months in advance.

- 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 10 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”.
- For termination problems that are not solved within 10 seconds by any tool, a second round is hold with the same rules except that the time-out is five minutes rather than one minute. The time-out may be used as a parameter for the tool, by which the tools may use different policies or heuristics for different time-outs.
- All results are reported on-line, including generated proof text, and statistics about scores and running time.
- Any tool generating a wrong answer will be disqualified.
- 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. Just like in 2004 and 2005 Claude Marché took care of the actual running of the competition. After a short delay it started on June 12, 2006. It took around ten days to run the full competition, due to ten participating tools and nearly 2000 termination problems.

3 The tools and the categories

There were eleven participating tools:

- AProVE, developed at RWTH in Aachen, Germany, coordinated by Jürgen Giesl.
- CiME, developed at LRI, Orsay, France, coordinated by Claude Marché.
- Jambox, developed by Jörg Endrullis, starting in Leipzig, Germany, and continued at Free University in Amsterdam, The Netherlands.
- Matchbox, developed by Johannes Waldmann, at HTWK in Leipzig, Germany. Just like AProVE and Jambox this tool is not standalone, but makes use of the SAT solver SatELite/MiniSat.
- MultumNonMulta, developed by Dieter Hofbauer in Kassel, Germany. Uses the `glpsol` solver from the GNU Linear Programming Kit.
- MU-Term, developed at Universidad Politécnica in Valencia, Spain, coordinated by Salvador Lucas. It makes use of CiME for polynomial constraint solving.
- TALP, developed by Claus Claves and Enno Ohlebusch. It makes use of the tool CiME for proving termination by polynomial interpretations.
- TEPARLA, developed by Jeroen van der Wulp at Technische Universteit Eindhoven, The Netherlands.
- TORPA, developed by Hans Zantema at Technische Universteit Eindhoven, The Netherlands.
- TPA, developed by Adam Koprowski at Technische Universteit Eindhoven, The Netherlands.
- TTTbox, developed by Martin Korp at Innsbruck Universität, Austria.

There were three categories, subdivided in the following eight subcategories:

- Standard term rewriting.
- Context-sensitive term rewriting. This means that for every operation symbol it is specified inside which position rewriting is allowed. If for every symbol every position is allowed, then this coincides with standard term rewriting.
- Term rewriting modulo theory. This means that apart from the rewrite rules also equations are specified (usually associativity and commutativity) modulo which rewriting is done. Having no equations coincides with standard term rewriting.
- 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.
- Innermost term rewriting. This means that only rewrite steps are allowed for which all proper subterms of the redex are in normal form.
- Standard string rewriting. This coincides with standard term rewriting in which all symbols have arity one.
- Relative termination of string rewriting.
- Logic programs.

The following table indicates which tool applies on which (sub)category:

tool	term rewriting					string rewr.		logic progr.
	stand.	cont. sens.	mod. th.	rel. term.	innermost	stand.	rel. term.	
AProVE	×	×	×		×	×		×
CiME	×	×	×		×	×		
Jambox	×			×		×	×	
Matchbox	×	×	×	×	×	×	×	
MultumNonM.						×	×	
MU-Term	×	×						
TALP								×
TEPARLA	×			×		×	×	
TORPA						×	×	
TPA	×			×		×	×	
TTTbox	×					×		

4 The results

Detailed results including

- all termination 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/2006/`

In this section we restrict to the main observations.

Since all tools execute complicated tasks it is likely that they contain bugs. For two tools (CiME and MU-Term) we detected some obviously incorrect generated proofs. We give their results below anyway, but if someone wants to use these tools, we recommend to contact the authors to get a bug-fixed version. We want to emphasize that this does not imply that all termination proofs generated by the remaining tools are correct: we did not check all thousands of generated termination proofs. As a long-term objective we see an automatic formal correctness check of the generated proofs.

The following table indicates the number of generated termination proofs for the remaining tools, divided over all (sub)categories:

	term rewriting					string rewr.		logic progr.
	stand.	cont. sens.	mod. th.	rel. term.	inner-most	stand.	rel. term.	
total number	865	133	71	45	69	322	45	325
AProVE	**638**	**60**	**53**		**66**	164		**225**
CiME	*345*	*16*	*40*		*12*	*44*		
Jambox	626			**27**		**251**	**36**	
Matchbox	395	16	8	22	14	176	33	
MultumNonM.						129	27	
MU-Term	*279*	*51*						
TALP								170
TEPARLA	355			19		101	21	
TORPA						201	28	
TPA	422			22		95	18	
TTTbox	193					75		

In this table for every category the highest score is printed in bold. Results of disqualified tools are in italics.

The largest category was the category of standard term rewriting, consisting of 865 termination problems. In this category AProVE was the strongest tool with 638 termination proofs, being 93 % of the 686 problems for which termination was proved by any tool. This result coincides with the results of 2005 and 2005 when AProVE had the highest score in this category too.

The great surprise was the tool Jambox, being a good second with 626 termination proofs. In 2004 and 2005 the second place was for TTT, which unfortunately did not participate this year. But in 2005 the difference between AProVE and TTT was nearly 100 systems, by which we may conclude safely that Jambox is stronger than TTT now. With a big distance TPA and Matchbox are third and fourth, with 422 and 395 termination proofs, respectively.

In the other subcategories of term rewriting AProVE had the highest score, just like in 2004 and 2005, except for relative termination for which AProVE did not participate. Also for logic programs AProVE was the winner.

In standard string rewriting, and for relative termination, both for term rewriting and string rewriting the highest scores were achieved by Jambox. In particular for standard string rewriting this was a surprise since both in 2004 and 2005 TORPA had the highest score. Now TORPA was second with 201 termination proofs, far behind Jambox with 251. Both in 2004 and 2005 AProVE was second in standard string rewriting, now only fourth after Matchbox.

Most proofs were found within a few seconds. For most tools the average time to find a termination proof was a few seconds; only MU-Term, TORPA and TTTbox were substantially faster. This year we had a second round for hard problems, having a time-out of five minutes rather than one minute. For the category of logic programs not a single new termination proof was found in this second round, applied on several dozens of problems. 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).

Apart from termination proofs also non-termination proofs were generated, all by presenting a looping reduction. Not all tools had facilities for this. For logic programs no non-termination proofs were given; and for the rewriting categories hardly outside the standard subcategories. In the subcategory of standard term rewriting AProVE found the most: 103 non-termination proofs, two of which were found in the second round. The tool Matchbox was second with 85. In the subcategory of standard string rewriting Jambox found the most: 25 non-termination proofs, where AProVE and Matchbox share the second place with 12 each.

5 Conclusions and challenges

The Termination Competition 2006 was really exciting due to new developments. The most powerful new technique applied was the matrix method [2,1]. The basic idea of this technique is the same as of polynomial interpretations: find interpretations such that by doing a rewrite step the interpretation strictly decreases. The difference with polynomials is that terms are interpreted as vectors over natural numbers rather than natural numbers, and that symbols are interpreted based on matrix multiplication rather than polynomials. This technique has been implemented in Jambox, Matchbox and MultumNonMulta, among which Jambox and Matchbox use a SAT solver for searching for suitable interpretations. Among these tools Jambox was the strongest by far, due to the fact that in Jambox also many other techniques have been implemented, including quite advanced instances of the dependency pair method. In the category of term rewriting Jambox ended second, close after the winner AProVE, and in the category of string rewriting Jambox ended first, far before TORPA, the second in this category.

New this year was the second round: after finishing a first round with time limit of one minute a second round was held for the hard problems with time limit five minutes. The effect of this second round was quite limited: only for a few rewrite systems a termination proof was found where the first round failed for all tools, and similarly a few non-termination proofs.

As an important objective for the future we see an automatic formal correctness check of generated proofs. However, achieving this both requires a lot of work and agreement about formats of the proofs. As a very preliminary step this year we required full proofs as generated proofs, including references to underlying theory. However, we did not yet have facilities for verifying this.

In 2005 we presented termination of the string rewriting system consisting of the three rules $aa \to bc$, $bb \to ac$, $cc \to ab$ as an open problem, since no tool solved this system SRS/Zantema-z086, also formulated as problem 104 in the RTA open problem list. Without any doubt this challenge has stimulated the development of the new strong matrix method, by which it was solved indeed this year by Jambox.

We see two important reasons for considering the termination competition to be successful and justifying annual continuation:

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

Again this year there were several termination problems that could not be solved by any of the tools, and can serve as a new challenge. As a new pearl we want to mention the string rewriting system SRS/Waldmann-jw1, consisting of the two rules

$$bbb \to aaa, \quad aaa \to bab.$$

Termination of this system is open: neither any tool can solve it nor a proof by hand has been found. In the category of string rewriting there are several more very small systems that could be solved by none of the tools, all being added in 2006, but this one is the smallest and most symmetric.

In the category of term rewriting we want to mention the single rule TRS/HofWald-6:

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

in which a is a constant and x, y are variables. It is easily seen to be non-terminating, but no tool can prove it. This example, several other single term rewrite rules in TPDB, and the above mentioned string rewriting systems were found by randomly generating small systems and filtering out systems that could be solved by a number of tools. But there are also rewrite systems having some more meaning for which termination can not be proved by any of the tools. A classical one is TRS/D33-33 describing a coding of the battle of Hydra and Hercules. As a final example we mention TRS/Zantema06-while consisting of the three rules

$$f(t,x,y) \to f(g(x,y),x,s(y)), \quad g(s(x),0) \to t, \quad g(s(x),s(y)) \to g(x,y).$$

Here x, y are variables, g stands for *greater than* and t stands for *true*, by which the second and third rule are the standard rules for *greater than* over the naturals composed from 0 and s(successor). The first rule describes the obviously terminating loop `while` $x > y$ `do` $y := y + 1$. Both for TRS/D33-33 and TRS/Zantema06-while termination proofs have been found by hand.

The emphasis in the competition is in rewriting rather than termination of programs. Even in the category of logic programs 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. For the next competition we plan to add new categories for Haskell programs, and for some kind of imperative programs.

We conclude by stating that everybody is welcome to suggest new problems for addition to TPDB.

References

1. J. Endrullis, J. Waldmann, and H. Zantema. Matrix interpretations for proving termination of term rewriting. In U. Furbach and N. Shankar, editors, *Proceedings of the third International Joint Conference on Automated Reasoning (IJCAR)*, Lecture Notes in Computer Science. Springer, 2006.
2. D. Hofbauer and J. Waldmann. Termination of string rewriting with matrix interpretations. In F. Pfenning, editor, *Proceedings of the 17th International Conference on Rewriting Techniques and Applications (RTA)*, Lecture Notes in Computer Science. Springer, 2006.