WST'2004 Competitition/Exhibition of termination tools

C. Marché and A. Rubio

March 8, 2004

1 Overview of the events

At WST'2004, two events involving systems for proving termination will take place: an exhibition and a competition.

The exhibition will be special session of the workshop programme, where authors of termination provers will make demos of their tools, and where any participant may propose a termination problem to solve.

The competition will be a completely automatic process that will be run without user interaction, during the workshop or a bit before, so that results can be announced during the exhibition. During this competition, a large set of termination problems, extracted from the termination problem database (TPDB), will be considered, and each of these problems will be submitted to each tool willing to participate. Given a termination problem, each tool is supposed to answer whether the given system/program is terminating. Each tool must run without any user interaction.

The precise rules of the competition are detailed in the next section. Those who are willing to participate should register to the mailing dedicated to the competition, by sending a mail at

```
majordomo@lri.fr
```

with

```
subscribe termtools
```

in the body of the message.

2 Rules of the competition

A tool participating to the competition will be evaluated by a completely automatic process. There will be three categories: term rewrite systems, string rewrite systems, and logic programs. A tool may participate to any number of categories.

The tool must be available as an executable that takes as argument the name of a file describing a termination problem, and an integer giving the maximal CPU time in second allowed to give an answer. The tool must run without any user interaction, and the answer must finally be printed on standard output.

For the term or string rewrite system category, the input file will be in the common format of the TPDB, available as a separate document. The expected answer must be either YES or NO, meaning that the given rewrite system is terminating (respectively, not terminating) under the STRATEGY given in the input file. If there is no strategy annotation, strong termination should be considered.

If the tool is not able to decide any termination property on the given system, it should simply answer nothing. If a system is still running when the maximal CPU time allowed is reached, its process will be killed. Notice that even if a tool is not able to handle the given strategy, it of course may try to prove strong termination. On the other hand, if a tool is not able to handle a THEORY annotation given, then it must give up, giving no answer.

For the logic program category, the input file will be a Prolog program in a standard syntax as in the TPDB. Additionally, in the command line will be given an *input query*, of the form $id(x_1, \ldots, x_n)$ where each x_i is either *i* or *o*. The tool should answer whether the given program terminates on every query with that *moding*.

For each category, no termination proof trace, or counter-examples in case of negative answers, are required, but if it appears that for some example, a wrong answer is given, then the tool will be "disqualified".

A score table will be computed for each category, by giving one point for each (correct) answer. Classifications will be computed for each category, several classifications will be made in the rewriting categories, with respect to strategies.

There is nothing to win, except of being declared as "the best automatic termination prover in the world in 2004" in the given category.

3   Technical details for participants

If you are willing to participate to the competition, here is what you have to do in details:

- One the authors of the tool (aka the corresponding author) must send a mail to Claude Marché (`Claude.Marcheatlri.fr`) giving the following informations :

  1. name of the tool
  2. URL of the web page of the tool
  3. author(s)
  4. short description of the tool features (20 lines max)

- One or more of the authors must register to the `termtools` mailing list as described above

- As soon as possible, send the program itself, in a mail attachment or via an URL. The preferred form is directly a stand-alone executable for i386/Linux, since the evaluation will be performed on this architecture. If it is not possible for you to provide this, discuss with `Claude.Marcheatlri.fr` to decide what to do.

The program must follow this specification:

- The program which has to be run has to be called `runme`. It could be a shell script which runs the actual tool.

- It runs without user interaction, only in "batch" mode writing on its standard output and standard error.

- It takes as command line arguments the file name containing the termination problem to solve, a maximum allowed CPU time (an integer, in seconds), and for logic programs, an input query.

- The answer must be given on the first line of the standard output, which must be either "YES" if termination has been established, or "NO" if non-termination has been established. Any other first line of answer is considered as "DON'T KNOW". The remaining lines of answer can be anything, for convenience it could be a kind of proof trace. (The full answer will be accessible from the result table).

  Making the first of answer be "YES" or "NO" can of course be done by the `runme` script.

- The program may create temporary files, but only in its current directory, and they must be erased after execution.