The Termination Competition 2007

Claude Marche, INRIA Futurs, ProVal, Parc Orsay Universite
Johannes Waldmann, HTWK Leipzig
Hans Zantema, Dept. of CS, TU Eindhoven

Idea of the Competition

Specification for *termination prover* program:

- Input: a *termination problem*
  ("is this rewrite relation/program terminating?")
- Output:
  - *answer* (YES/NO)
  - *reason*, readable/verifiable by
    - informed human (from the beginning)
    - *machine* (since this year)

Challenges for provers:

- find many correct answers (quickly)

The Termination Problem DB

different sources, different purposes

- “classical” problems (e.g. Ackermann function)
- applications (e.g. Haskell Prelude)
- problems that illustrate power of certain methods/implementations
- hard problems (cannot currently be solved by any automated method)
  note: “hard” now may become “easy” next year
- open problems (e.g. $3n + 1$ problem)

huge, open, unbalanced collection

- randomly select 128 per subgroup (directory)

Categories and Provers 2007

- *standard term rewriting:*
  Aprove, TTT2, Jambox, NTI
- *context-sensitive term rewriting:* Muterm, A.
- *standard string rewriting:*
  Matchbox, Torpa, TTT2, Jambox, A., MultumNonMulta, NTI
- *relative string rewriting:* J-x, M-x, M-N-M, Torpa,
- *logic programming:* Aprove, Polytool, Talp, NTI
- *functional progr. (demonstration):* Aprove
  Note: consisting exclusively of “real” problems (from Haskell Prelude and Standard Libraries)

Certified Termination

(currently, only for standard term rewriting)

- TPA $\longrightarrow$ Rainbow $\longrightarrow$ Color/Coq
- Cime $\longrightarrow$ Coccinelle/Coq
- TTT2 $\longrightarrow$ Rainbow $\longrightarrow$ Color/Coq

this was new in 2007, and it is a major progress.
($\approx$ half of the TRS proofs can be certified now)

Ongoing work on TPG: unified format (XML)
for certifiable termination proofs:
independent of prover, and of certifier.
(Rainbow translates TPG to Color/Coq)

Stunning Proofs (TRS)

- Aprove proves termination of

$$\left\{ \begin{array}{l} f(t,x,y) \longrightarrow f(g(x,y),x,s(y)), \\ g(s(x),0) \longrightarrow t, g(s(x),s(y)) \longrightarrow g(x,y) \end{array} \right\}$$

simulating the program

```
while (x > y) do y := y + 1;
```

Method: *bounded increase*
works for several such TRS
that were presented as challenges in 2006

Stunning Proofs (TRS)

- TTT(box/2)
  generalizes $*$–bounds to non-left-linear systems, introduces quasi-deterministic tree automata
- Muterm
  uses *non-integer polynomial interpretations* (but proofs removed due to an unrelated accident)

Stunning Proofs (SRS)

- TTT2 proves (DP, self labelling) z090 (Touzet)

  ```
  s b -> b s s s, s b s -> b t, t b -> b s, t b s -> u t b,
  b u -> b s, t s -> t t, t u -> u t, s u -> s s
  ```

  is implementation of (infinite) SRS over $\mathbb{N}$
  $[x+1, y] \to [x, y+3], [x+1, y, z] \to [x, z, y]$

  complexity is *not primitive recursive*.

  challenge: try jw3: $[x+1, 0] \to [x, 0, x+1]$
- Torpa (QPI) and Matchbox (arctic matrices) prove jw1 $\{b^3 \to a^3, a^3 \to bab\}$ (open in 2006)

Non-Termination

(earlier)

- find loops by brute force enumeration of derivations
- restrict to forward closures (Jambox, Matchbox, Torpa)
- use DP/SCC analysis (Aprove)

(this year)

- compressed loops (NTI (new entry), Matchbox)
- (nontermination using SAT (TTT2) — why not in competition?)

Conclusion

merits of the termination competition:

- objective comparison
  of different (methods and) implementations
- stimulates new research (to solve “solved” problems faster, to solve “unsolved” problems.)
- emphasizes certification

issues for future competitions:

- (several),

discuss at WST business meeting