Symbolic Enumeration of One-Rule String Rewriting Systems

Alfons Geser, Johannes Waldmann, Mario Wenzel
HTWK Leipzig

Workshop on Termination, Obergurgl, 2016

Motivation: Size Does Matter

looking at small hard examples (for anything, really)

- highlights features of existing methods and implementations (strengths, weaknesses)
- invites invention of new methods and implementations (use small examples as “coffee table problems”)

specifically, termination of one/few-rule string rewriting

- rule shape $0^*1^* \to \{0,1\}^* \Rightarrow$ decision procedure (Sénizergues 1996)
- Zantema’s problem $\{a^2b^2 \to b^3a^3\} \Rightarrow$ matchbounds (2003)
- Zantema’s “other problem” $\{a^2 \to bc, b^2 \to ac, c^2 \to ab\} \Rightarrow$ matrix interpretations (2006)

Any sufficiently complex decision problem *must have* small hard instances. Termination of one-rule string rewriting *could* be decidable.

Traditional Approach: Explicit Enumeration

- enumerate canonical representatives
  (w.r.t. permuting letters, mirroring words, permuting rules)
- filter
  (w.r.t. "easy" criteria that imply termination or non-termination, e.g., overlaps, count letters)

drawback:

- time-consuming (generate-and-test . . . many tests!)
- more clever generator (less tests) $\Rightarrow$ more complex program (deal with several criteria at once)

history:

- Kurth 1990 (one-rule, rhs size ≤ 6)
- Geser 2001 (one-rule, rhs size ≤ 9)
- Waldmann 2007 (total size ≤ 12)

Filter Criteria: Redundance

redundant $=$ has (lexicographically) smaller equivalent system

- permute letters, reverse words, (permute rules)
  equivalence class of $\{10 \to 011\}$
  is $\{\{10 \to 011\}, \{01 \to 100\}, \{01 \to 110\}, \{10 \to 001\}\}$.
- *borders* (common prefix and suffix)
  Ex.: $abba \to abaaba$ is bordered by a,
  $[bb] \to [b][][b]$ is shorter, and equivalent for termination
- codes (inverse morphism)
  Ex..for $bca \to aabc$, use code $\{a, bc\}$,
  reduce termination problem to $[bc][a] \to [a][a][bc]$.
  code must be free of overlaps

Filter Criteria: Ease

termination is implied

- counting letters
- Kurth's non-overlap criterion D

nontermination is implied

- loops of length 1 (embedding)
- loops of length 2 (overlap patterns)

decision procedure is known

- (McNaughton) $\exists$ inhibitor $i \in \Sigma(r) \setminus \Sigma(l)$
- (Sénizergues) $l \in 0^*1^*$
- (Geser) grid criterion $\exists a \in \Sigma : |l|_a = |r|_a > 0$

Our *New* Approach: Symbolic Enumeration

key points:

- represent set of (interesting) SRS *symbolically*, as set of models of a binary decision diagram (BDD)
- fix $\Sigma, |l|, |r|$, one-hot encoding for letters
- construct BDD by Boolean operations (conjunction) from (encodings of) interesting properties

advantages:

- orthogonality: encode each criterion on its own
- counting, inclusion check without enumeration
- arbitrary boolean combinations

drawback:

- not everything can be encoded efficiently (quantification is expensive, since it needs to be expanded)

Implementation

done by Mario Wenzel

- use cudd BDD library, Haskell API
- Haskell main program
- filter locally with matchbox and ttt2 (low timeout) (20.000 CPU hours)
- filter on starexec (larger timeout)
- submit remaining systems for TPDB
- with small modifications, do the same for cycle rewriting

technical observation:

- "canonicity after reversal and renaming"
  implemented by enumerating all permutations of letters,
  this is exponential in $|\Sigma|$

Observations and Expected Results

- pure-matchbox (used for filtering) did:
  RFC matchbounds, forward closure enumeration
- after observing performance on these one-rule SRS, extended by
  - “strip symbols” (Torpa had it? AProVE has it, and it helps),
  - transport systems (Matchbox already had this at some point)
- for cycle rewriting, use full matchbounds, and adapt transport systems
- if there was a one-rule SRS/cycles category, matchbox should currently win it...

Concrete Examples

smallest one-rule systems unsolved in tests on starexec:

- strings:

  ·●·●●●·●●·● → ·●●·●●●·●·●●●·

  ·●·●·●··●·●· → ·●·●··●·●··●·●

  ·●·●·●··●·●· → ·●··●·●··●·●·●

  `http://termcomp.imn.htwk-leipzig.de/competitions/168`

- cycles

  ·●··●· → ··●·●··●

  ·●··● → ··●·●·●··●

  ·●·●●·● → ·●●·●·●●·

  `http://termcomp.imn.htwk-leipzig.de/competitions/167`

Is Termination Decidable . . .

. . . for string rewriting with only one rule? (Geser 2001)

Some say “yes”. Two approaches:

- - non-terminating $\iff$ has loop
  - there is a computable bound on the length of a shortest loop
- - terminating $\iff$ RFC-matchbounded . . .
  - after stripping common prefix/suffix . . .
  - and codes (inverse homomorphism) . . .
  - with a condition that allows harmless overlaps

And, for cycle rewriting?