Termination

Johannes Waldmann
HTWK Leipzig, Germany

September 18, 2015

Termination

- abstract: a relation $\to$ on A is *terminating* iff each $\to$-chain is finite
- which of these are terminating?
  $>$ on $\mathbb{N}$, $>$ on $\mathbb{Q}_{\geq 0}$, $\{(2x, 3x) \mid x \in \mathbb{N}_{>0}\}$,
  $\{(2x, x) \mid x \in \mathbb{N}_{>0}\} \cup \{(2x+1, 3x+2) \mid x \in \mathbb{N}_{>0}\}$
- concrete: $\to$ is the small-step semantics of some program in some model of computation
  examples: state transition, $\to_\beta$, rewriting
- "$\to$ is terminating" is:
  - undecidable in general
  - important for applications

Termination of Imperative Programs

- example program:
  ```
  while (x>0) { x-- ; y++ ; }
  ```
- computation state is $\mathbb{N}^2$ (slightly cheating)
- computation step is state transition $(x+1, y) \to (x, y+1)$ where $x \ge 0$
- terminates because 1st component decreases and is bounded from below (by 0)
- represent number x by term $S^x(Z)$ e.g., $0 = Z, 2 = S(S(Z))$
- represent program by *term rewriting system* $\{P(S(x), y) \to P(x, S(y))\}$.

Termination of Functional Programs

- example program (first order)
  is term rewriting system

  ```
  m(x,y+1) = p(m(x,y),x); m(x,0) = 0;
  p(x+1,y) = 1 + p(x,y); p(0,y) = y;
  ```
- computation state is *tree* (term)
  containing nested function calls
  e.g., `m(3,2)`, `p(3,p(3,m(3,0)))`
- consider all possible computations
  (all evaluation strategies)

TRS that are Not Programs

in (TRS for) (FO) functional programs:

- clear separation of *function* and *data* symbols (in previous ex.: function: P, M, data: S, Z)
- each left-hand side (lhs) of a rule has exactly one function symbol (at the top)

in term rewriting:

- lhs can contain several “function” symbols
- this is motivated by transformation of programs (optimization), simplification of expressions, e.g., $x \wedge (y \vee z) \to (x \wedge y) \vee (x \wedge z)$,

Rewriting

- pattern replacement in context
  $C[l\sigma] \to C[r\sigma]$ for rule (l, r)
  for: graphs, DAGs, terms (trees), strings (paths)
- term rewriting is a language for both
  computation (apply rules to data) ...
- ... and *deduction*
  (apply rules to statements and their proofs)
- e.g., for proving/deriving types of programs

String Rewriting

- a *string* is a finite sequence of symbols
- equiv.: . . . a term (tree) where all symbols are unary (all nodes have one child)
- string rewriting systems are actually well-known (rules of formal grammars of Chomsky type 0)
- rewrite system $R \subseteq \Sigma^* \times \Sigma^*$ defines rewrite relation $\to_R$ as $\{(plq, prq) \mid p, q \in \Sigma^*, (l, r) \in R\}$
- example $R = \{(ab, ba)\}$, $aabb \to_R abab$.
- string rewriting is still hard (Turing complete)
- and illustrates a lot of term rewriting

Groups and String Rewriting

- represent groups by *relations* on *generators*
- e.g., the symmetry group of the rectangle:
  $\langle H, V \mid H^2 = V^2 = (HV)^2 = 1\rangle$
  (Klein's 4-group) (cf. Erlangen Program 1872)
- computations with group elements $\Rightarrow$
  computations on representations ($=$ strings)
  e.g., $VH = H^2VH = H^2VHV^2 = H(HV)^2V = HV$
- orient equations, obtain *semi-Thue* system
  ($=$ string rewriting system)
  (named after Axel Thue 1863–1922,
  student of Sophus Lie 1842–1899,
  successor of Felix Klein 1849–1925 at Leipzig)

Examples for SRS Termination

- $R_1 = \{ab \to a\}$ is terminating since
  $u \to_{R_1} v$ implies $|u|_b > |v|_b$.
- $R_2 = \{ab \to ba\}$ is terminating since
  number of inversions
  $I(u) = \#\{(p,q) \mid p < q, u_p = a, u_q = b\}$
  decreases: $u \to_{R_2} v$ implies $I(u) = 1 + I(v)$.
- $R_3 = \{ab \to bba\}$
  long computations: $ab^k \to b^{2k}a$, $a^k b \to b^{2^k} a^k$
  is terminating since … (interpretation)
- $R_4 = \{ab \to b^2 a^2\}$ is non-terminating

Termination Competitions

- since 2003, yearly, for programs with spec:
- input: rewrite system R,
  out: YES (R terminates), NO, MAYBE/timeout

extensions:

- variants of rewriting (strategies, modulo AC,...)
- programming languages (Haskell,Prolog,Java,C)
- complexity (derivation lengths)
- certification (of proofs of (non) termination)

termcomp 2015:

- 10 participants, 10^4 problems, 10^7 sec (CPU)
- 10 h (wall), `http://www.starexec.org/`

Well-founded Monotone Algebras

- Σ-algebra $[\cdot]$ with domain D
  maps each letter $c \in \Sigma$ to a function $[c] : D \to D$
  and the string $u_1 \dots u_n$ to $[u_1](\dots([u_n](0))\dots)$
- assume well-founded (terminating) order $>$ on D
- $[\cdot]$ is *monotone* if
  $\forall c \in \Sigma, x, y \in D : x > y \Rightarrow [c](x) > [c](y)$.
- $[\cdot]$ is *compatible* with R if
  $\forall (l, r) \in R, x \in D : [l](x) > [r](x)$.
- Thm (Manna and Ness): R terminating $\iff$ R admits compatible well-founded monotone alg.
- Ex: for $\{ab \to ba\}$: $[a](x) = 2x, [b](x) = x + 1$.
  for $\{ab \to bba\}$, take ...

Totally Ordered Algebras

- for algebras over $(\mathbb{N}, >)$, always $[x] \geq x$.
  Proof: trivially $[0] \geq 0$,
  $x + 1 > x \Rightarrow [x + 1] > [x] \geq x$.
- $\{aa \to aba\}$ is terminating,
  (count occurences of *aa*)
  but does not admit compatible algebra over $(\mathbb{N}, >)$.

A Non-Totally Ordered Alg. of Vectors

- domain $D_k = \mathbb{N}^{k-1} \times \mathbb{N}_{>0}$, well-founded order
  $x > y \iff x_1 > y_1 \wedge x_2 \geq y_2 \wedge \cdots \wedge x_k \geq y_k$
- interpret letter c by matrix $[c] \in \mathbb{N}^{k \times k}$
  must map D_k into D_k, hence $[c]_{k,k} > 0$
  must be monotone, hence $[c]_{1,1} > 0$.
  compatible with (l, r) if $[l] \geq [r] \wedge [l]_{1,k} > [r]_{1,k}$
- $[a] = \begin{pmatrix} \boxed{1} & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & \boxed{1} \end{pmatrix}$, $[b] = \begin{pmatrix} \boxed{1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \boxed{1} \end{pmatrix}$,
  $[aa] = \begin{pmatrix} 1 & 1 & \boxed{1} \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$, $[aba] = \begin{pmatrix} 1 & 1 & \boxed{0} \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$,

Linear Interpretations as Matrices

- had $[a](x) = 3x, [b](x) = x + 1$ for $\{ab \to bba\}$
- this can be written as $[a] = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}, [b] = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$
  then $[ab] = \begin{pmatrix} 3 & 3 \\ 0 & 1 \end{pmatrix}, [bba] = \begin{pmatrix} 3 & 2 \\ 0 & 1 \end{pmatrix}.$
- for $\{ab \to baa\}$, there is no linear int. over $(\mathbb{N}, >)$
- but we can take
  $[a] = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}, [b] = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$
  then $[ab] = \begin{pmatrix} 1 & 3 \\ 0 & 3 \end{pmatrix}, [bba] = \begin{pmatrix} 1 & 2 \\ 0 & 3 \end{pmatrix}.$

What a Difference an *a* Makes

decide termination of

- $R_1 = \{ba \to acb, bc \to abb\}$,
- $R_2 = \{ba \to acb, bc \to cbb\}$,
- $R_3 = \{ba \to aab, bc \to cbb\}$.

$\{ba \to aab, bc \to cbb\}$

- we have $b^k a \to^* a^{2^k} b^k$ and $bc^k \to^* c^k b^{2^k}$
- from $bc^k a$, doubly exponential derivation lengths
- $\Rightarrow$ there is no compatible matrix interpretation
- $[a]_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, [b]_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, [c]_1 = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}$
  $[a]_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, [b]_2 = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}, [c]_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix};$
  if $u \to_{bc \to cbb} v$, then $[u]_1 > [v]_1$
  if $u \to_{ba \to aab} v$, then $[u]_1 = [v]_1 \land [u]_2 > [v]_2$
- $[u] = ([u]_1, [u]_2)$ is lexicographically decreasing, the lexicographic product of wf orders is wf.
- $[\cdot]_1$ "removes rule" $bc \to cbb$ (cf. Relative Termination, Alfons Geser 199?)

$\{ba \to acb, bc \to cbb\}$

- $b^k a \to^* a(cb)^k \to^* ac^k b^{2^k-1}$
- $b^2 a^k \to^* \ldots$ multiply exponential
- admits no lexicographic matrix proof since each rule is applied more that exponentially often

prove termination by showing $\to \subseteq >_{a,c,b}$ where

- $u >_{x,y,\ldots} v$ iff $u = u_0 x u_1 x \ldots x u_m$, $v = v_0 x v_1 x \ldots x v_n$ with $x \notin u_i, x \notin v_i$ and $[u_0, \ldots, u_m] > [v_0, \ldots, v_n]$ length-lexicographically w.r.t. $>_{y,\ldots}$
- this is the lexicographic path order (Nachum Dershowitz, 198?) for precedence $a > c > b$.

$\{ba \to acb, bc \to abb\}$

- simple form of non-termination is *loop* $u \to^+ puq$
- loops can be found by *explicit enumeration*

we show here an *implicit* loop detector:

- observe $\forall x \in \{a, b, c\} : bx \to^* \phi(x)b$ where $\phi : a \mapsto ac, b \mapsto b, c \mapsto ab$.
- hence, $\forall k : b^k x \to^* \phi^k(x)b^k$
- find x and k such that $\phi^k(x)$ contains $b^k x$ as scattered subword $\Rightarrow$ loop
- Parikh matrix of ϕ is $P = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$, P has eigenvalue > 1, entries in P^k grow exponentially, claim follows.

Where to Go From Here

Termination

- strings $\to$ terms, terms $\to$ programs
- matrices over $\mathbb{N} \to$ matrices over exotic semirings: (max,plus), (min,plus), (min,max)
- constraint programming for finding matrices

Complexity

- each termination proof method bounds derivation lengths (e.g., matrices $\Rightarrow$ exponential)
- special interest in polynomial bounds

there's much more to Rewriting: equational reasoning (completion), higher order, graphs,. . .