Weighted Tree Automata as Certificates for Termination of Term Rewriting

Adam Koprowski, TU Eindhoven

Johannes Waldmann, HTWK Leipzig

Introduction: Rewriting

term rewriting system R is set of rules,
rule is pair of terms (with variables),
left hand side describes pattern,
right hand side describes replacement.
system R defines

- top rewrite relation $\stackrel{\text{top}}{\longrightarrow}_R$
- rewrite relation $\longrightarrow_R$ (context closure of $\stackrel{\text{top}}{\longrightarrow}_R$)

model of (parallel, nondeterministic) computation

one important question: is $\stackrel{\text{top}}{\longrightarrow}_R, \longrightarrow_R$ well-founded?

Introduction: Termination

undecidable in general (Turing completeness)
various semi-algorithms. basic ideas:

- syntactic methods: find well-founded ordering $>$ on $\mathrm{Term}(\Sigma)$ with $s \to_R t \Rightarrow s > t$
  Knuth-Bendix-Ordering (1970), (lexicographic, multiset) path ordering (Dershowitz 1982)
- interpretation $[\cdot]$ from term algebra $\mathrm{Term}(\Sigma)$ into some well-founded Σ-algebra $(A, >)$ such that $s \to_R t \Rightarrow [s] > [t]$.
  polynomial interpretations (into $\mathbb{N}$) (Manna, Ness 1970; Lankford 1975)

Introduction: Automation

find termination proofs automatically

- find precedence/ordering
- find (polynomial) interpretation
- use transformations: e.g. dependency pairs transformation (Arts, Giesl 2000)

  termination of $\to_R \iff$ termination of $\stackrel{\text{top}}{\longrightarrow}_{R'}$

typically, produces step-wise proofs:

- splitting (into independent sub-problems)
- removal of rules

Introduction: Certification

there are several (rather advanced) termination "provers",

regular Termination Competition since 2003.

is the output of such a program really a proof?

yes: if it is accepted by a proof checker (Coq, Isabelle)

uses library of termination proof methods (Color: Blanqui, Koprowski 2006 . . .).

Interpretations (I)

interpretation of function symbol $f \in \Sigma_k$
by a function $[f] : A^k \to A$
can be extended to terms with variables:
if $t \in \mathrm{Term}(\Sigma, V)$, then $[t] : (V \to A) \to A$.
for rewriting system R, ordering $>$ on A, define

- $[\cdot]$ is compatible with R if $\forall (l \to r) \in R$,
  $\sigma : \mathrm{Var}(l) \cup \mathrm{Var}(r) \to A : [l]\sigma > [r]\sigma$
- $[\cdot]$ is monotonic (closed w.r.t. contexts) if
  $\forall f \in \Sigma_k, v_1, \ldots, v_i' > v_i, \ldots v_k \in A :$
  $[f](\ldots, v_i', \ldots) > [f](\ldots, v_i, \ldots)$

Interpretations (II)

- If $(A, >)$ is well-founded
  and $[\cdot]$ into $(A, >)$ is compatible with R,
  then $\stackrel{\text{top}}{\longrightarrow}_R$ is terminating.
- If $(A, >)$ is well-founded
  and $[\cdot]$ into $(A, >)$ is compatible with R and monotonic,
  then $\longrightarrow_R$ is terminating.

Now: define $[\cdot]$ by a weighted tree automaton,
take $A = (Q \to W)$.

Weighted Tree Automata (WTA)

WTA A consists of ranked signature Σ,
weight semi-ring $(W, +, \cdot, 0, 1)$, set of states Q,

weighted transitions: for $f \in \Sigma_k : \mu_f : Q^{k+1} \to W$,
defines an interpretation $\mathrm{Term}(\Sigma) \to (Q \to W)$ by
$[f(t_1, \ldots, t_k)] = q \mapsto$
$\sum_{q_1,\ldots,q_k \in Q} \mu_f(q_1, \ldots, q_k, q) \cdot [t_1](q_1) \cdot \ldots \cdot [t_k](q_k)$

final weight vector $\gamma : Q \to W$,
defines $A(t) = \sum_{q \in Q} \gamma(q) \cdot [t](q)$.

classical instance: $W =$ Boolean semi-ring.

Separated (Linear?) WTA

in the general model, a WTA interprets a function symbol $f \in \Sigma_k$ by a function

$$[f] : (Q \to W)^k \to (Q \to W)$$

that is multi-linear (a tensor).

simplification (restriction): functions of shape
$[f](v_1, \ldots, v_k) = (M_0+)M_1 \cdot v_1 + \ldots + M_k v_k$
where M_0 vector, $M_1, \ldots, M_k$ square matrices.

Note: (M_0+) by de-homogenization (assume last vector component is $= 1$)
Note: closed under composition (substitution).

WTA over $\mathbb{N}$

(the “matrix method”, Endrullis, Hofbauer, Waldmann, Zantema 06)
domain $A = (Q \to \mathbb{N})$, ordering
$u > v \iff u_1 > v_1 \wedge u_2 \geq v_2 \wedge \ldots \wedge u_n \geq v_n$.

$[\cdot]$ compatible with rule $l \to r$ is implied by
$[l]_0 > [r]_0$ and $\forall i > 0 : [l]_i \geq [r]_i$.
$[\cdot]$ monotonic is implied by $\forall i > 0 : (M_i)_{1,1} > 0$.

$$[f](v_1, \ldots, v_k) = M_0 + M_1 \cdot v_1 + \ldots + M_k \cdot v_k$$

Arctic WTA

$\mathbb{A} = (-\infty \cup \mathbb{N}, \max, +, -\infty, 0)$
domain $\mathbb{N} \cdot \mathbb{A} \cdot \ldots \cdot \mathbb{A}$
ordering: component-wise extension of $>_0$
where $x >_0 y \iff x > y \vee x = y = -\infty$.

Note: $a >_0 b \wedge c >_0 d \Rightarrow \max(a, c) >_0 \max(b, d)$
and $a >_0 b \wedge a \in \mathbb{N} \Rightarrow a \neq b$.

$[\cdot]$ compatible with rule $l \to r$
is implied by $\forall i \geq 0 : [l]_i >_0 [r]_i$.

$$[f](v_1, \ldots, v_k) = M_0 + M_1 \cdot v_1 + \ldots + M_k v_k$$

Arctic WTA (II)

$[f]$ must not leave the domain

$\mathbb{N} \cdot \mathbb{A} \cdot \ldots \cdot \mathbb{A}$

- e.g. require that $\forall f : \exists i : ([f]_i)_{1,1} \in \mathbb{N}$.
- in fact only $[l]\sigma$ must be in the domain: require that $([l]_0)_1 \in \mathbb{N}$.

$$[f](v_1, \ldots, v_k) = M_0 + M_1 \cdot v_1 + \ldots + M_k v_k$$

Arctic Monotonicity?

$$[f](v_1, \ldots, v_k) = M_0 + M_1 \cdot v_1 + \ldots + M_k v_k$$

if $k > 1$, then no such $[f]$ is monotonic.
($\Rightarrow$ no “deep” termination proofs, “only” top termination proofs.)

for $k = 1$ (string rewriting),
$[f]$ is monotonic if $M_0 = -\infty^Q$ and $(M_1)_{1,1} \in \mathbb{N}$.
this is the Matchbox 2007 method.

Arctic WTA . . . below zero

$\mathbb{A}_\pm = (-\infty \cup \mathbb{Z}, \max, +, -\infty, 0)$
domain ${\mathbb{A}_\pm}^Q \cap \{v \mid v_1 \geq k\}$ for some $k > -\infty$
ordering and compatibility as before.

Keeping the domain:
Does not work: $\forall f : \exists i : ([f]_i)_{1,1} \in \mathbb{Z}$.
but this works: require that $([l]_0)_1 \geq k$.
in fact, require only $\forall (l \to r) \in R : ([l]_0)_1 > -\infty$
and then take $k = \min\{([l]_o)_1 \mid (l \to r) \in R\}$.

$$[f](v_1, \ldots, v_k) = M_0 + M_1 \cdot v_1 + \ldots + M_k v_k$$

Results

- implementation transforms to SAT problem (other approaches: complete, randomly, evolutionary)
- $\mathbb{N}$ matrices for string and term rewriting: 2006
- $\mathbb{A}$ matrices for string rewriting (Matchbox): won the 2007 Termination competition
- $\mathbb{A}$ and $\mathbb{A}_{\pm}$ matrices for term rewriting: will take part in 2008 (Matchbox/TPA)
- formal proofs (for Coq) are being worked on, extending the existing proofs for $\mathbb{N}$ matrices in the Color/Rainbow framework

WTA properties

previous conditions (on $[f]$) are in fact crude approximations to limitedness problems in WTA:

- input: WTA A over $(W, >)$, regular language L
- question: $\inf\{A(t) \mid t \in L\} > 0$

for $W =$ Boolean: decidable ($L \setminus L(A) \stackrel{?}{=} \emptyset$),
for $W = \mathbb{A}$: decidable (use Boolean WTA that recognizes the support of $L(A)$)
but decision algorithm not easy for a constraint solver.

Open Questions

automata theory:

- decidability of $\mathbb{A}_{\pm}$ limitedness
- compare languages of linear WTA to those of full (multilinear) WTA
- . . . e.g. for the Boolean case

rewriting:

- compare proving power of $\mathbb{N}$, $\mathbb{A}$, $\mathbb{A}_{\pm}$ WTA
- . . . w.r.t. number of states

implementation:

- find WTA compatible with given R quickly

WTA limitedness

our problem:

- input: WTA A over $\mathbb{A}_{\pm}$
- question: $-\infty < \inf\{A(t) \mid t \in \mathrm{Term}(\Sigma)\}$

cf. limitedness problem for tropical (min,plus) automata (Hashiguchi 1982, Leung 1991, Kirsten 200?) but

- strings $\longrightarrow$ trees
- domain $\mathbb{N} \cup \infty \longrightarrow$ domain $\mathbb{Z} \cup \infty$

(Comment by D. Kirsten: it follows from a result by Krob that the above problem is undecidable)