Rewrite Properties and Presburger Logic

Johannes Waldmann (HTWK Leipzig)

Workshop Proof Theory and Rewriting
Kanazawa, 2013

Outline

- properties of rewrite systems:
  - non-termination, termination
  - lower/upper bounds on derivational complexity
- certificates for these properties:
  - (families of) derivations $\Rightarrow$ lower bounds
  - interpretations $\Rightarrow$ upper bounds
- Presburger logic for:
  - defining the certificate
  - defining the validity of a certificate
- applications:
  - decide validity of certificate
  - unify (non)termination proof methods
- extensions (work in progress):
  - enlarge the class of certificates
  - finding certificates automatically

Rewrite Systems and Properties

- string rewriting system: $R \subseteq \Sigma^* \times \Sigma^*$

  Ex. $R = \{(al, ar), (rb, br)\}$
- derivation relation $u \to_R v :=$
  $\exists p, s \in \Sigma^*, (x, y) \in R : u = pxs \wedge pys = v$

  Ex. $\underline{al}bb \to_R a\underline{rb}b \to_R ab\underline{rb} \to_R abbr$
- properties:
  - R is non-terminating $:=$ there is an infinite path $w_0 \to_R w_1 \to_R w_2 \to \dots$
  - R is terminating $:= \neg$ (R is non-terminating)

Non-Termination

Specifying families of derivations

example: (non-looping) non-termination for

- $R = \{al \to ar, rb \to br, re \to ble, bl \to lb\}$
- for each $i \geq 0$: $alb^i e \to arb^i e \to^i ab^i re \to ab^{i+1} le \to^{i+1} alb^{i+1} e$
- inf. derivation $ale \to^3 albe \to^5 alb^2 e \to^7 \ldots$

formal proof of non-termination of R:

- specify sub-derivation $alb^i e \to^{2i+2} alb^{i+1} e$
- verify that sub-derivations are correct and can be linked
- using Presburger Logic
  (= first-order logic with $+$ and $=$ over $\mathbb{N}$)

Specifying families of derivations (II)

$$alb^i e \to arb^i e \to^i ab^i re \to ab^{i+1} le \to^{i+1} alb^{i+1} e$$

using relations $D_x(i,t,p)$ for $i,t,p \in \mathbb{N}, x \in \Sigma_\perp$
t = time, p = position, x = letter or blank ($\perp$)

$$
\begin{array}{rcl}
D_a(i,t,p) & = & (p = 0) \\
D_e(i,t,p) & = & (t \leq i+1 \wedge p = i) \vee (t > i+1 \wedge p = i+1) \\
D_l(i,t,p) & = & \begin{array}{cl} & (t = 0 \wedge p = 1) \\ \vee & (t > i+2 \wedge t+p = 2i+4) \end{array} \\
D_r(i,t,p) & = & 0 < t \wedge t \leq i+1 \wedge t = p \\
D_\perp(i,t,p) & = & (t \leq i+1 \wedge p > i+2) \vee (t > i+1 \wedge p > i+3) \\
D_b(i,t,p) & = & \neg \bigvee_{x \in \{a,b,l,r,\perp\}} D_x(i,t,p)
\end{array}
$$

Validating families of derivations

relations D_x specify a non-terminating deriv. if

- (encode)
  - $\forall i, t, p$: exactly one of $\{D_x(i, t, p) \mid x \in \Sigma_\perp\}$
  - $\forall i, t, p : D_\perp(i, p, t) \Rightarrow D_\perp(i, p, t + 1)$
- (step) $\forall i, t : \exists q : \bigvee_{(l,r) \in R}$
  "rule (l, r) was applied at position q"
  - left of q: no change,
  - in $q \ldots q + |l| - 1$: change according to rule,
  - right of $q + |l|$: shift by $|r| - |l|$
- (link) $\forall i : \exists t : t > 0 \wedge$
  $\forall p : \bigwedge_{x \in \Sigma_\perp} D_x(i, t, p) \iff D_x(i + 1, 0, p)$

Validating families of derivations (II)

since we use *Presburger Logic*, which is decidable,

- to define D_x (= the family of derivations),
- to define validity of D_x;

we have

- D_x is a certificate for non-termination, validity of certificate is decidable,
- this generalizes loops, and self-embedding rewrite closures ($pu^i s \to^+ pu^{i+1} s$)
- reduction strategies can be included if the redex selection criterion is Presburger-definable (e.g., leftmost)

Termination

Presburger definable interpretations

apply the general termination proof method

- monotone Σ-algebra A
  with well-founded carrier $(D, >)$:
  for each $f \in \Sigma$, a function $f_A : D \to D$
  with $\forall x, y \in D : x > y \Rightarrow f_A(x) > f_A(y)$.
- …compatible with R:
  $\forall (l, r) \in R, x \in D : l_A(x) > r_A(x)$

  where $w_A(x) = w_{1_A}(\cdots w_{n_A}(x) \cdots)$
  for $w = w_1 \ldots w_n$

- for A that are Presburger-definable,
  e.g., $D = \mathbb{N}^k$, $>_A := >_\mathbb{N} \times \geq_\mathbb{N}^{k-1}$,
  f_A using $+$, max, ….

Definition of an Interpretation

$f : \mathbb{N} \to \mathbb{N}$ is p-quasi-periodic (of slope 1)
iff $\forall x : f(x+p) = p + f(x)$

Ex. ($p = 3$)

	0	1	2	3	4	5
a	2	3	4	5	6	7
b	3	3	3	6	6	6
A	2	4	5	5	7	8
B	3	4	4	6	7	7

writing functions as relations:

$$b(i,o) \;:=\; \exists t : (o = t+t+t) \wedge (i < o) \wedge (o \le i+3)$$

$$A(i,o) \;:=\; \begin{array}{ll} & (\exists t : i = 3t) \Rightarrow i+2 = o \\ \wedge & \neg(\exists t : i = 3t) \Rightarrow i+3 = o \end{array}$$

Properties of Interpretations

- weak monotonicity
  $\forall i_1, i_2, o_1, o_2 :$
  $i_1 \leq i_2 \land a(i_1, o_1) \land a(i_2, o_2) \Rightarrow o_1 \leq o_2$
- compatibility, e.g., with $(l, r) = (Aaa, Bab)$

$$l(i, o) := \exists p, q : a(i, p) \land a(p, q) \land A(q, o)$$
$$r(i, o) := \exists p, q : b(i, p) \land a(p, q) \land B(q, o)$$
$$\forall i, o_1, o_2 : l(i, o_1) \land l(i, o_2) \Rightarrow o_1 \geq o_2$$

Presburger interpretations (II)

- the termination certificate (the interpretation) is Presburger definable, e.g.,
  - built from basic functions (const, id, mod k)
  - by max, plus, composition
- the validity of the certificate is Presburger definable
  - monotonicity
  - compatibility with R
- validity of certificate is decidable

not Presburger definable in general:

- validity of the *method*,
- *existence* of a certificate

Presburger interpretations (III)

- generalizes quasi-periodic interpretations; arctic, tropical, fuzzy matrix interpretations
- can extend to $\mathbb{N}^k$-valued interpretations, using Presburger definable functions on components
- Q: what is the implied upper bound on derivational complexity? (exponential)
- can be combined with transformations (e.g., relative (top) termination, Dependency Pairs method)

Implementation, Extensions

Deciding Presburger Formulas

- represent set of models of (sub-)formula F with free variables $x_1, \ldots, x_n$ as language over $\{0, 1\}^n$ (with stacked binary encoding)
- realize logical connectives as operations on finite automata
- for efficiency, use compressed representation of automata (by BDDs)
- DISCUSS: omega automata not needed.

Application:

- the certificate (derivation, interpretation) can be represented directly by the automaton (instead of the formula)

Automatic Relations/Interpretations

the decision method works in a (slightly) more general setting:

- formula can refer to any *automatic* relation
- some are not Presburger-definable, like $E(n) :=$ "the number of 1-bits of n is even"

possible applications in rewriting:

- more complicated derivations
- more involved interpretations
  e.g., using weakly monotone function
  $f : \mathbb{N} \to \mathbb{N} : x \mapsto 2^{\lfloor \log_2 x \rfloor}$

How to find Models for Formulas

How to find the (non) termination certificate?

- input: F with free relation symbols $R_1, \ldots$
- output: for each R_i, a Presburger definable/automatic relation R_{i_A}

such that F is true in A. — Methods of solution:

- enumerate formulas for R_i
- enumerate automata for R_i
- derive a (Boolean) constraint system for A from F. — This is somewhat ambitious: it requires a propositional encoding of the Presburger decision method.

Hybrid Search for Interpretations

propositional encoding for

- DAG that describes interpretation functions (weakly) monotone $\mathbb{N} \to \mathbb{N}$
  - leaves: base functions (const 0, succ, ...)
  - branch nodes: operations (composition, ...)
- table: node $\mapsto$ prefix of list of values
- weak/strict compatibility with rules

algorithm:

- if SAT solver finds a solution candidate C (looking at the table only),
- it can be verified by Presburger decision method (looking at the DAG/terms only)
- if verification fails, add $<_{\text{lex}} C$ (or $>_{\text{lex}} C$, resp.) as a constraint, and repeat

Outline

- properties of rewrite systems:
  - non-termination, termination
  - lower/upper bounds on derivational complexity
- certificates for these properties:
  - (families of) derivations $\Rightarrow$ lower bounds
  - interpretations $\Rightarrow$ upper bounds
- Presburger logic for:
  - defining the certificate
  - defining the validity of a certificate
- applications:
  - decide validity of certificate
  - unify (non)termination proof methods
- extensions (work in progress):
  - enlarge the class of certificates
  - finding certificates automatically

References

- Mojżesz Presburger, *Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, . . .* , 1929
- Reddy, Loveland, *Presburger arithmetic with bounded quantifier alternation*, STOC 78
- Geser, Zantema, *Non-looping string rewriting*, ITA, 99
- Oppelt, *Automatische Erkennung von Ableitungsmustern. . .* , diploma thesis, 08
- Waldmann, Zantema, *Termination by Quasi-Periodic Interpretations*, RTA 07