Weighted Automata for Proving Termination of String Rewriting

Johannes Waldmann
Hochschule für Technik, Wirtschaft und Kultur (FH) Leipzig
waldmann@imn.htwk-leipzig.de

String rewriting

"…is what the rules of a type-0 grammar do"

- rewriting system $R = \{l_1 \to r_1, \ldots\}$ over Σ is set of pairs of words over Σ
- defines relation $\to_R$ on Σ^* by $u \to_R v \iff \exists x, y \in \Sigma^*, (l \to r) \in R : u = x \cdot l \cdot y, x \cdot r \cdot y = v$

example: $R = \{a^2 \to bc, b^2 \to ac, c^2 \to ab\}$

- allows derivation $bb\boxed{aa} \to_R b\boxed{bb}c \to_R ba\boxed{cc} \to_R b\boxed{aa}b \to_R \boxed{bb}cb \to_R a\boxed{cc}b \to_R aabb \to_R \ldots$
- is there an infinite $\to_R$-chain?

Problems in String Rewriting

given a finite rewrite system R,

- is R terminating?

  there are no infinite $\to_R$ chains

- does R preserve REG? ... preserve CF?

  $R^*(L) := \{v \mid u \in L, u \to_R^* v\}$.
  R preserves $\mathcal{L}$ iff $\forall L \in \mathcal{L} : R^*(L) \in \mathcal{L}$

Focus of this talk: automatic termination
(two meanings: automatically find weighted automata that are certificates of termination)

Plan of this talk

weighted finite automata allow unified view of:

- - Dieter Hofbauer, J. W.: Proving Termination with Matrix Interpretations, submitted, 2006
  - D. H., J. W.: Termination of $\{aa \to bc, bb \to ac, cc \to ab\}$, to appear in IPL, 2006
- - D. H., J. W.: Deleting string rewriting systems preserve regularity, TCS 327(3):301-317, 2004
  - Alfons Geser, D. H., J. W.: Match bounded string rewriting systems, AAECC 15(3-4):149-171, 2004

(Global) Compatibility

general idea: use monotone interpretation into well-founded domain

- A is a V-weighted automaton over Σ, defines a weight function $A : \Sigma^* \to V$
- A is called compatible with relation $\to$ on Σ^* if $u \to v \Rightarrow A(u) > A(v)$.
- $(V, >)$ well-founded and A compatible with $\to$ implies $\to$ is well-founded.

special plan: ensure compatibility of automaton A with rewrite relation $\to_R$ by local conditions on A.

Local compatibility

If $(V, >)$ is ordered semi-ring with

- $(a > b) \Rightarrow (a + c) > (b + c)$
- $(a > b) \wedge (c \neq 0) \Rightarrow (a \cdot c) > (b \cdot c)$

and A over Σ (states Q with i initial, f final)
is locally compatible with R:

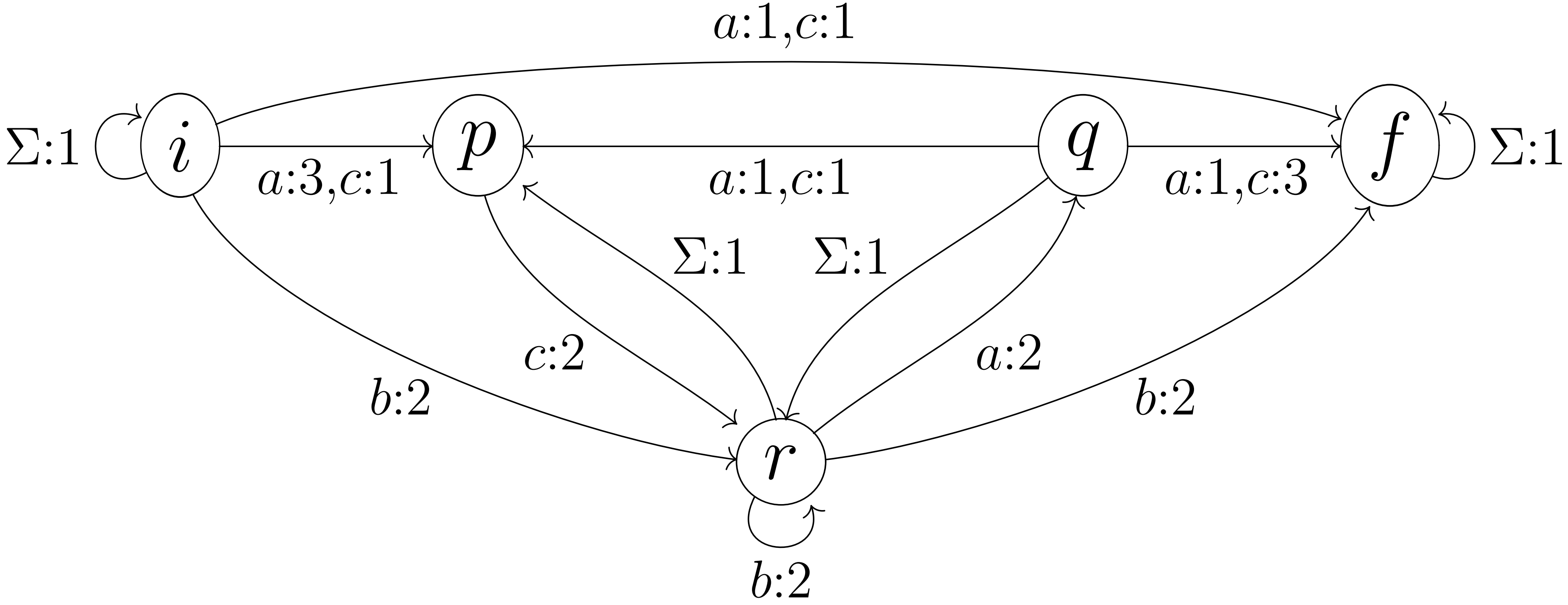
- $\forall x \in \Sigma : A(i, x, i) > 0 \wedge A(f, x, f) > 0$
- $\forall p, q \in Q, (l \to r) \in R : A(p, l, q) \geq A(p, r, q)$
- $\forall (l \to r) \in R : A(i, l, f) > A(i, r, f)$

then A is (globally) compatible with $\to_R$.

Example (1)

$R = \{aa \to bc, bb \to ac, cc \to ab\}, \Sigma = \{a, b, c\}$

$V = (\mathbb{N}, +, \cdot, 0, 1)$ and standard ordering $>$



$A(i, aa, f) = 2 > 1 = A(i, bc, f)$

$A(r, bb, r) = 4 \geq 4 = A(r, ac, r)$.

How to find such automata

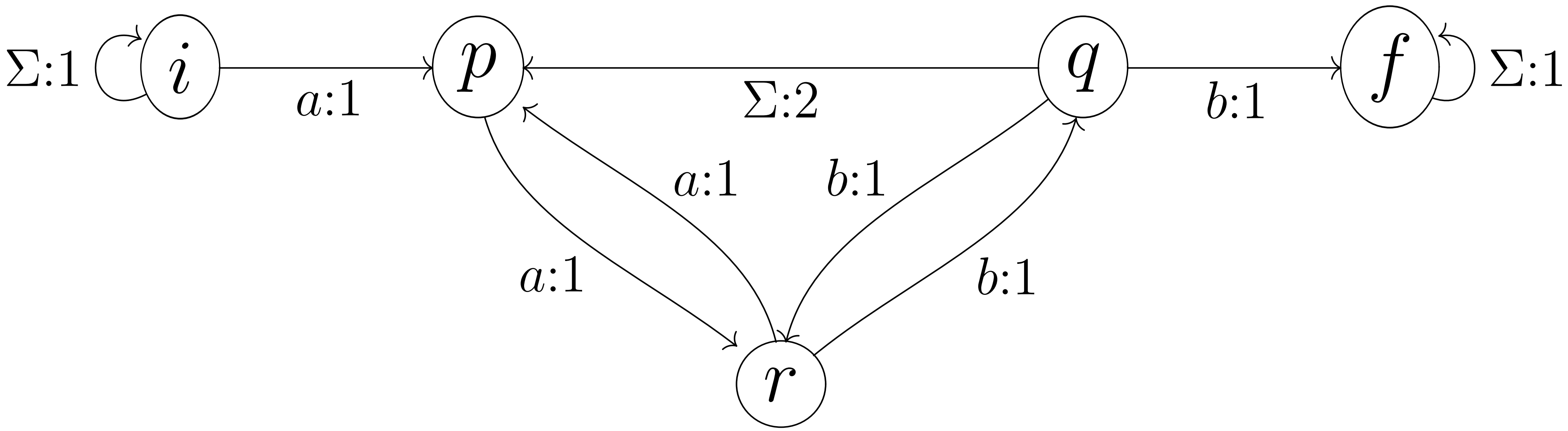
- fix number d of states, say 5.
  automaton is mapping $t : \Sigma \to \mathbb{N}^{d \times d}$
- local compatibility $\Rightarrow$ constraint system with $|\Sigma| \cdot d^2$ unknowns and $|R| \cdot d^2$ constraints
- fix maximal value for entries, say 7.
  $\Rightarrow$ finite domain constraint system
- represent unknowns in binary $\Rightarrow$ boolean satisfiability problem, (15.000 variables, 90.000 clauses, 300.000 literals) $\Rightarrow$ solve by SAT solver (SateliteGTI) (takes 7 seconds)

Example (2)

standard test case for automated termination

$R = \{a^2b^2 \to b^3a^3\}, \Sigma = \{a, b\}$

$V = (\mathbb{N}, +, \cdot, 0, 1)$ and standard ordering $>$



$A(i, a^2b^2, f) = 1 > 0 = A(i, b^3a^3, f)$

$A(q, a^2b^2, p) = 4 \geq 1 = A(q, b^3a^3, p)$.

Summary (so far)

- new automated termination method for string rewriting with powerful implementation (can solve problems that no other method can)
- developed in joint work with Dieter Hofbauer
- generalized to term rewriting in joint work with Jörg Endrullis and Hans Zantema
- can't handle more than 5 states well via SAT solver, more synthetic construction of automata (matrices) would be much desirable

Part 2: we show a variant of this method where we already have a synthetic construction

A Multi-Set Semi-Ring

Idea: given V-weighted automaton A over Σ.

- For a path in A labelled $(w_1/v_1)(w_2/v_2)\ldots$, consider multi-set of weights $\{v_1, v_2, \ldots\}$.
- For a word w over Σ, consider lowest weight-multi-set of paths with $w = w_1 w_2 \ldots$

$\mathbb{M}(V) = \top \cup \mathbb{N}^V$ (with finite support) is semi-ring:

- $0 := \top$, $1 := \emptyset$
- $A + B := \min_{\gg}(A, B)$ (multiset extension of $>$)
- $A \cdot B := A \cup B$ (adding weights), $A \cdot 0 := 0$

Multi-set ordering

given $(V, >)$, define $\gg$ on V-multi-sets as $\gg_1^+$ for
$(x > y_1 \wedge \ldots \wedge x > y_n) \Rightarrow (A \backslash x \gg_1 B \cup \{y_1, \ldots, y_n\})$

- if $>$ total, then $\gg$ total
- if $>$ well-founded, then $\gg$ well-founded

$(\mathbb{M}(V), \gg)$ is ordered semi-ring (make $\top$ maximal)

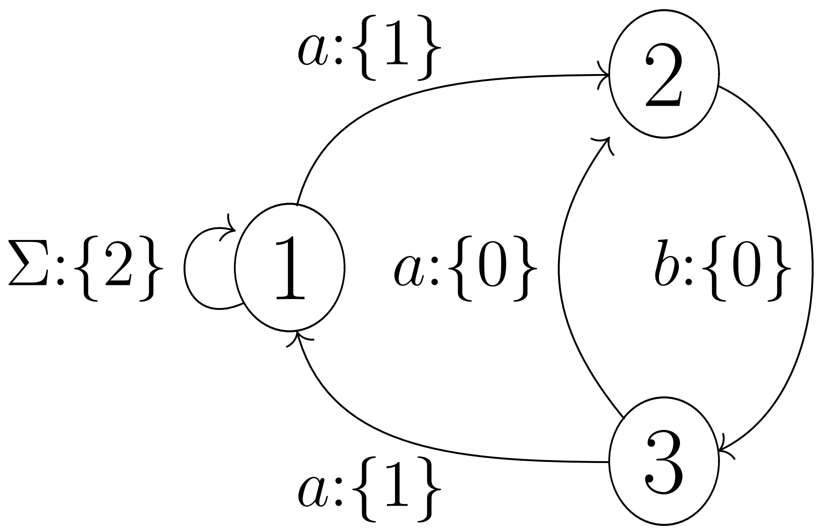
An alternative picture

$\ldots$ of this ordered semi-ring of multi-sets:

- domain is $\mathbb{N}^*$ (but no leading 0): multiplicities, starting with largest element

  for $V = \{a > b > c > d\}$,
  $\{a, c, c\} \mapsto 1020$ and $\{b, c, d\} \mapsto 111$.
- ordering is length-lexicographic: $1020 > 111$
- multiplication is point-wise addition, right-aligned: $1020 \cdot 111 = 1131$
- addition is minimum w.r.t. ordering $1020 + 111 = 111$

A $\mathbb{M}(V)$-weighted Automaton



$$A(aa) = \begin{pmatrix} \{2,2\} & \{2,1\} & \top \\ \top & \top & \top \\ \{1,2\} & \{1,1\} & \top \end{pmatrix}$$

$$A(aba) = \begin{pmatrix} \{1,0,1\} & \{1,0,0\} & \top \\ \top & \top & \top \\ \{0,0,1\} & \{0,0,0\} & \top \end{pmatrix}$$

For $A(1, aba, 1)$ note $\{2,2,2\} \gg \{1,0,1\}$ etc.

Compatibility

for $\mathbb{M}(V)$, we have

- $(a \gg b) \wedge (c \neq 0) \Rightarrow (a \cdot c) \gg (b \cdot c)$

we do not have

- $(a \gg b) \Rightarrow (a + c) \gg (b + c)$

instead, will use

- $(a \gg b) \wedge (c \gg d) \Rightarrow (a + c) \gg (b + d)$

to infer global compatibility (of a $\mathbb{M}(V)$-automaton with $\longrightarrow_R$), need something sharper than local compatibility.

Strict local compatibility

If $(V,>)$ is ordered semi-ring with

- $(a > b) \wedge (c > d) \Rightarrow (a+c) > (b+d)$
- $(a > b) \wedge (c \neq 0) \Rightarrow (a \cdot c) > (b \cdot c)$

and A over Σ (states Q with i initial and final) is strictly locally compatible with R:

- $\forall x \in \Sigma : A(i,x,i) \neq 0$
- $\forall p,q \in Q, (l \to r) \in R :$
  $A(p,l,q) = 0 \vee A(p,l,q) > A(p,r,q)$

then A is (globally) compatible with $\to_R$.

Flattening the Multi-sets

Given $(V, >)$, consider $V' = V \cup \{-\infty, +\infty\}$ and semi-ring $(V', -\infty, +\infty, \min_>, \max_>)$.

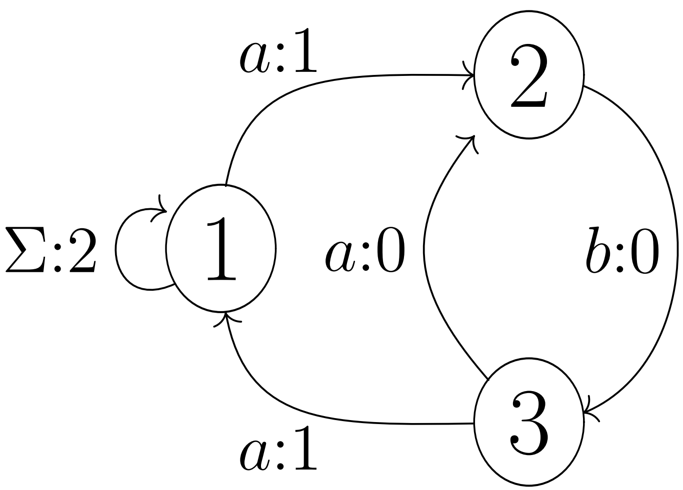
$\mathrm{flat} : \mathbb{M}(V) \to V' : B \mapsto \max B, \top \mapsto +\infty$
is a morphism of ordered semi-rings.

- $(\mathrm{flat}\, B > \mathrm{flat}\, C) \Rightarrow (B \gg C)$ (but not "$\Leftarrow$")
- $(\mathrm{flat}\, B \geq \mathrm{flat}\, C) \Leftarrow (B \gg C)$

$\ldots$ will use the stronger ordering via flat

Strict "flat" compatibility

If the $(V', >)$-weighted automaton A is strictly locally compatible with R, then its "lifted" $(\mathbb{M}(V), \gg)$-weighted automaton is compatible with $\to_R$ (...but A itself is not)



$$\operatorname{flat} A(aa) = \begin{pmatrix} 2 & 2 & + \\ + & + & + \\ 2 & 1 & + \end{pmatrix} \qquad \operatorname{flat} A(aba) = \begin{pmatrix} 1 & 1 & + \\ + & + & + \\ 1 & 0 & + \end{pmatrix}$$

this is the concept of match-boundedness.

Match-Bounded Rewriting

Annotate letters by numbers ("match heights").
In each rewrite step $x \cdot l \cdot y \to_R x \cdot r \cdot y$,

- annotate each letter in r
  by ($1+$ minimal annotation in l).

Example $R = \{aa \to aba\}, a_2\underline{a_3a_0} \to a_2a_1b_1a_1$
If heights (starting from 0) are bounded, then

- R is terminating
- R effectively preserves REG
- R has certificate automaton (see prev. slide!)
- R^- effectively preserves CF

Summary, Open Questions

two termination methods using weighted automata:

- weights in $(\mathbb{N}, +, \cdot)$: “matrix method”, automata are “guessed” (finite domain constraint system)
- weights in $(\mathbb{N}, \min, \max)$: match bounds, (huge) automata can be efficiently constructed

Questions:

- efficient construction of $(\mathbb{N}, +, \cdot)$ automata?
- existence of $(\mathbb{N}, \min, \max)$ automaton $\Rightarrow$ existence of $(\mathbb{N}, +, \cdot)$ automaton?
- other semi-rings for termination?