Automatic Termination

Johannes Waldmann
HTWK Leipzig, Germany

Automatic **Termination**

- using weighted finite automata
- that can be found automatically

unified view of

- (standard) matrix method (2006)
- arctic matrix method (2008)
- match bound method (2003)

including some history, some alternate views and explanations, some directions for extensions

Termination of Programs

Alan Turing: Checking a large routine, 1949.

…Finally the checker has to verify that the process comes to an end. Here again he should be assisted by the programmer giving a further definite assertion to be verified. This may take the form of a quantity which is certain to decrease continually and vanish when the machine stops. To the pure mathematician it is natural to give an ordinal number … $(n-r)\omega^2 + (r-s)\omega + k$.

http://www.turingarchive.org/browse.php/B/8

Combinatorial Group Theory

Max Dehn (1911): for a group presentation, e.g., $(X = \{a, b\}, R = \{a^2 = 1, b^3 = 1, a^{-1}ba = b^{-1}\}))$, and $w_1, w_2 \in X^*$, decide word problem $w_1 \leftrightarrow^*_R w_2$.

small cancellation theory, Dehn's algorithm,. . .

> . . . let $w = bcd$ . . . find suitable $w' = bt^{-1}d$ of shorter length. A finite number of such reductions . . .

(cited in: Lyndon, Schupp: Combinatorial Group Theory, 1977, Sect. V.4)

Languages and Rewriting (I)

- grammars are rewrite systems
- operations on languages by rewriting (completion) of automata

e.g., REG is closed w.r.t. monadic ($=$ inverse context-free) rewriting (rhs of length ≤ 1)

Proof: given a finite automaton, add transitions (only - not states)

(Ronald Book, Matthias Jantzen, Celia Wrathall, 1982)

Languages and Rewriting (II)

For the solitaire game

- rules $R = \{OXX \to XOO, XXO \to OOX\}$,
- solved positions $L = O^*XO^*$,

the set of solvable positions $R^{-*}(L)$ is a regular language. (Folklore theorem. Exercise in book by Zohar Manna, 1974.)
Exercise in book by Jean Berstel, 1979: the congruence of $\{OXX \to XOO\}$ is a rational transduction (thus REG-preserving)
Bala Ravikumar, 2004: change-bounded rewriting is REG-preserving, R is change-bounded (by 4).

Automata and Paths

an automaton $A = (\Sigma, Q, I, F, \delta)$ determines, for each pair of states, a set of paths.

composition of paths:

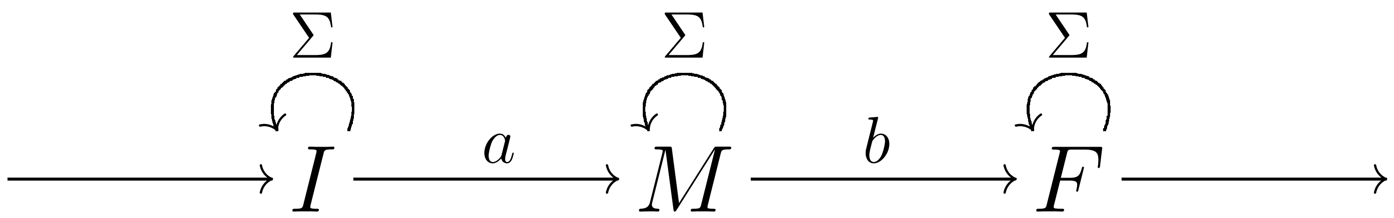
- sequential: $p \xrightarrow{u} q \wedge q \xrightarrow{v} r \Rightarrow p \xrightarrow{u \cdot v} r$
- parallel:

$$\begin{array}{rl} & p_0 \xrightarrow{u_1} p_1 \ldots p_{k-1} \xrightarrow{u_k} p_k \\ \vee & p_0 \xrightarrow{u_1} p'_1 \ldots p'_{k-1} \xrightarrow{u_k} p_k \end{array} \Rightarrow p_0 \xrightarrow{u_1 \cdot \ldots \cdot u_k} p_k$$

A computes a function $Q \times \Sigma^* \times Q \rightarrow \{0, 1\} = \mathbb{B}$

Weighted Automata

count the number of paths:

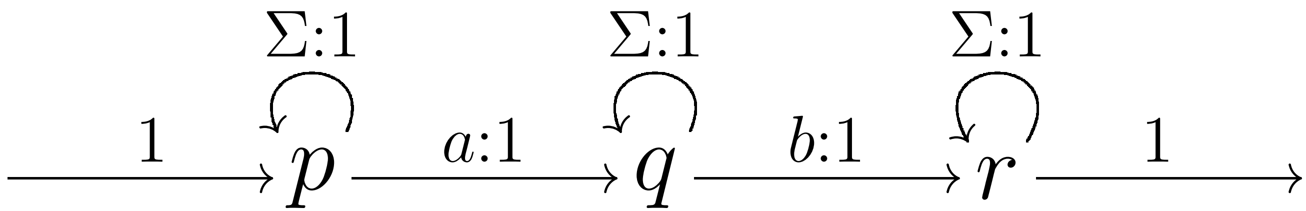


attach weights to transitions and combine:
sequential: multiplication, parallel: addition

computes a function $Q \times \Sigma^* \times Q \to \mathbb{N}$
e.g., $A(I, aabb, F) = 4, A(I, abab, F) = 3$.

with initial and final weights $I, F : Q \to \mathbb{N}$,
global weight function $A : \Sigma^* \to \mathbb{N}$ given by
$A(w) = \sum\{I(p) \cdot A(p, w, q) \cdot F(q) \mid p, q \in Q\}$

Automata = Matrices



$$I = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \quad \Bigg| \quad [a] = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \Bigg| \quad [b] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad \Bigg| \quad F = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

algebraic view of automaton: $A(w) = I \cdot ([w] \cdot F)$

algebra domain: weight vectors $D = (Q \to \mathbb{N})$,

algebra operations: for $x \in \Sigma, [x] : D \to D$

Weighted Automata/Languages

- Marcel Schützenberger, On a Definition of a Family of Automata, 1961.
- Jean Berstel, Christophe Reutenauer: Rational Series and Their Languages, 1988.

  formal power series in non-commuting variables:
  coefficient of monomial $=$ weight of word
- Manfred Droste, Werner Kuich, Heiko Vogler: Handbook of Weighted Automata, 2009.

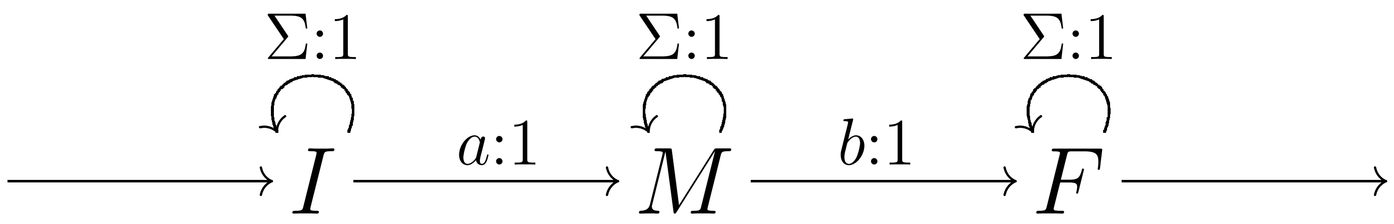
Automata and Rewriting

A is (globally) compatible with a rewrite system R:

$\forall u, v \in \Sigma^* : u \to_R v \Rightarrow A(u) > A(v)$

if weight domain $(\mathbb{N}, >)$ is well-founded,

then automaton A certifies termination of R



is compatible with $R = \{ab \to ba\}$

e.g., $A(I, aabb, F) = 4, A(I, abab, F) = 3$.

but how to prove this?

Testing Compatibility

- $\forall p, q \in Q : A(p, l, q) \geq A(p, r, q)$
- $\forall u, v \in \Sigma^* : \exists i, p, q, f \in Q :$

$$i \xrightarrow{u} p \overset{l}{\underset{r}{\rightrightarrows}} q \xrightarrow{v} f \;\wedge\; A(p, l, q) > A(p, r, q)$$

Let $A' = A$ plus $\{p \xrightarrow{D} q \mid A(p, l, q) > A(p, r, q)\}$
then check $\Sigma^* \cdot D \cdot \Sigma^* \subseteq \mathrm{supp}(A')$
where $\mathrm{supp} : (\mathbb{N}, +, \cdot, 0, 1) \longrightarrow (\mathbb{B}, \vee, \wedge, 0, 1)$

This test is PSPACE.

Simplified Compatibility (I)

"E/P" method: $I = (1, \ldots), F = (\ldots, 1)^T$,

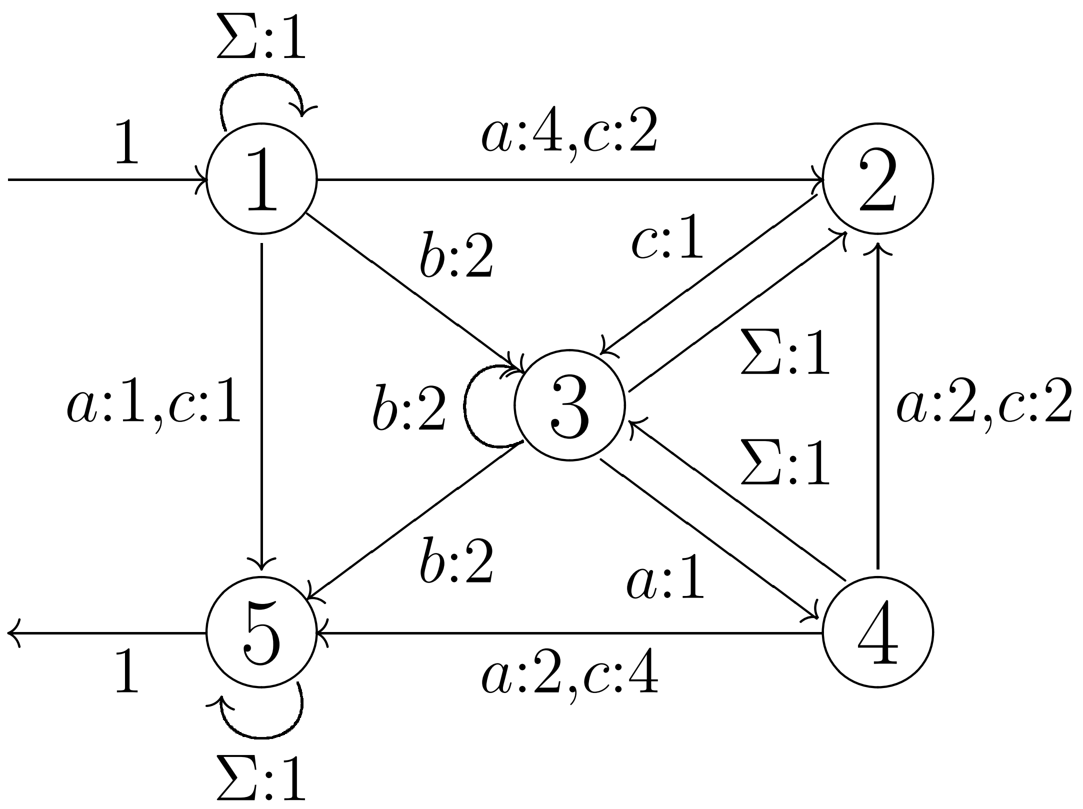
$$[x] \in \begin{pmatrix} 1 & \ldots & * \\ \vdots & \ddots & \vdots \\ * & \ldots & 1 \end{pmatrix}, [l] - [r] \in \begin{pmatrix} \ldots & 1 \\ & \vdots \end{pmatrix}$$

corresponds to (extended) monotone algebra

- domain: $(\ldots, 1)^T$,
- strict order: $x_1 > y_1 \wedge x_2 \geq y_2 \wedge \ldots \wedge x_d \geq y_d$
- weak order: $x_1 \geq y_1 \wedge \ldots \wedge x_d \geq y_d$

Endrullis, W., Zantema 2006

An Example Automaton



(Dieter Hofbauer, J.W., 2006) strictly compatible w.

$\{a^2 \to bc, b^2 \to ac, c^2 \to ab\}$ (Hans Zantema, 2003)

Simplified Compatibility (II)

"M/M" method: $[x] \in M \wedge [l] - [r] \in M$

where $M = \{m \mid \forall i \exists j : m_{ij} > 0\}$

Hofbauer/W 2006;

Pierre Courtieu, Gladys Gbedo, Olivier Pons 2009

what is the corresponding algebra (domain, order)?

Weighted Tree Automata

- Σ-algebra of weight vectors $D = (Q \to \mathbb{N})$
- interpret $f \in \Sigma_k$ by multilinear $[f] : D^k \to D$
  e.g., $(x, y) \mapsto x + x \cdot y + y$
- restrict to sums of unary linear functions:
  path-separated weighted tree automaton

  $$[f] : (\vec{v_1}, \ldots, \vec{v_k}) \mapsto M_1 \cdot \vec{v_1} + \ldots + M_k \cdot \vec{v_k} + \vec{a}$$

  weight of tree is sum of weights of paths.

Jörg Endrullis, Hans Zantema, J.W.:
Matrix Interpretations for TRS, 2006.

Exotic semirings

- generalize weight domain: $\mathbb{B}, \mathbb{N}, \ldots$ semirings
- tropical $(\mathbb{N} \cup \{+\infty\}, \min, +, +\infty, 0)$
  named after Imre Simon, Univ. Sao Paulo
  investigated Finite Power Property, and Star Height Problem of regular languages
  (Limited Subsets of a Free Monoid, 1978)
- artic $(\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$
  the “opposite” of tropical
- “fuzzy” $(\mathbb{N} \cup \{-\infty, +\infty\}, \min, \max, +\infty, -\infty)$

Motivation, Applications

degrees of polynomials: addition $\mapsto$ max, multiplication $\mapsto$ plus ($=$ the arctic semiring)

limits of "warped" classical operations:

$x \oplus_b y = \log_b(b^x + b^y), x \otimes y = x + y = \log_b(b^x \cdot b^y)$

for $b \to +\infty$: arctic, for $b \to +0$: tropical

(large deviation theory, tropical algebraic geometry, idempotent analysis)

tropical $=$ (min,plus) $=$ shortest-path algebra

Arctic Termination

need a different local compatibility condition:

standard plus is strict: $y > y' \Rightarrow x + y > x + y'$

arctic plus ($=$ max) is not: $\max(3, 2) = \max(3, 1)$

but it is half strict:
$x > x' \wedge y > y' \Rightarrow \max(x, y) > \max(x', y')$

use $x \gg y := x > y \vee x = y = -\infty$

and require $[l] \gg [r]$ pointwise.

Adam Koprowski, J.W., 2008
including formal verification in Color/Coq.

Quasi-Periodic $\to$ Arctic

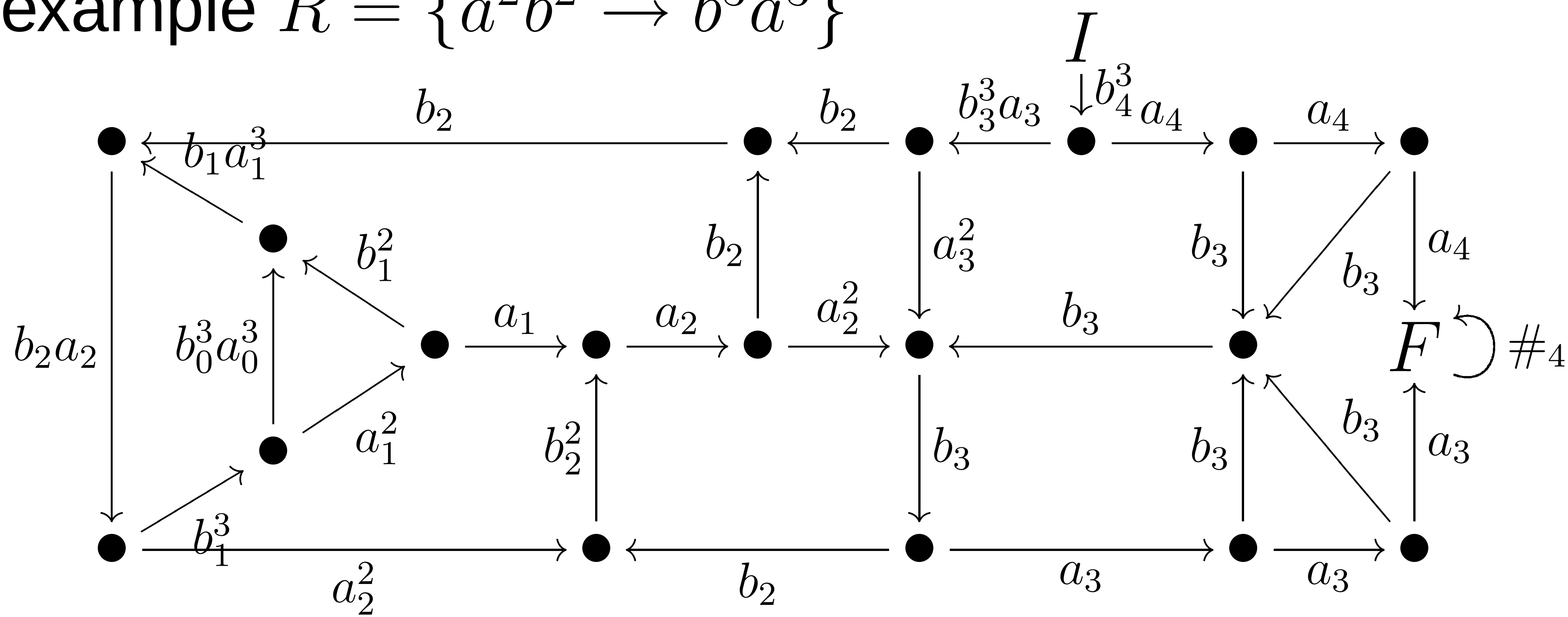
$R = \{bab \to a^3, a^3 \to b^3\}$, proof by Aleksey Nogin, Carl Witty 2006; Hans Zantema, J.W., 2007

x	0	1	2	3	4	5	...
$[a](x)$	1	2	3	4	5	6	...
$[b](x)$	0	3	3	3	6	6	...
$[aab](x)$	3	6	6	6	9	9	...
$[aaa](x)$	3	4	5	6	7	8	...
$[abb](x)$	0	3	3	3	6	6	...

translation to arctic (period $=$ dimension) (Adam Koprowski, J.W., 2009), used by Matchbox

Match bounds

$\forall (l,r) \in R : \forall p \xrightarrow{l}_A q : \exists p \xrightarrow{r}_A q$
such that max of l labels $>$ max of r labels.
example $R = \{a^2 b^2 \to b^3 a^3\}$



Alfons Geser, Dieter Hofbauer, J.W. 2003

Match bounds and Semirings

- the automaton is actually weighted in the semiring $(\mathbb{N} \cup \{-\infty, +\infty\}, \min, \max, +\infty, -\infty)$
- there is an implicit “min” in the notion of “match-compatibility”
- local compatibility does not imply compatibility $A(ulv) > A(urv)$, since $\operatorname{range} A$ is finite.
- globally, use semiring of multisets of labels.
- matchbounds for relative termination. relative rules may keep $-\infty$. (J.W. 2007)
- observation: R/S match-bounded $\wedge\ S$ match-bounded $\Rightarrow R$ match-bounded.

Find Compatible Automata

- constraint solving: for fixed dimension,
  - describe compatibility by constraint system
    standard: `QF_NIA`, arctic: `QF_LIA`,
    match: `QF_IDL`
  - apply SMT solver (Barcelogic, Yices, Z3, . . .)
  - for fixed bit width, transform to `QF_BV`
    - then SMT solver (Boolector, . . .)
    - transform to CNF,
      then SAT solver (Minisat, . . .)
- completion

Automata by Completion

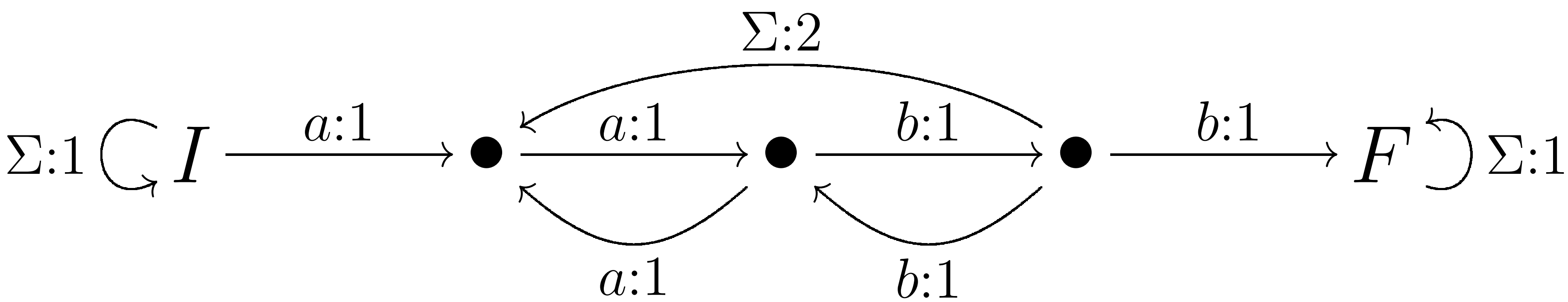
- problem: for automaton A, rewrite system R, compute B with $\mathrm{L}(B) = R^*(\mathrm{L}(A))$.
- algorithm: for each $(l \to r) \in R, p \xrightarrow{l}_A q$, ensure that $p \xrightarrow{r}_A q$ by adding paths
- for weighted automata: condition $A(p, l, q) > A(p, r, q)$.
- problems:
  - non-termination because of added states
  - non-linearities (for TRS)

Completion for Matchbounds

- can construct language closure w.r.t. rewriting at the same time
- this is used in the RFC method (prove match-boundedness on right hand sides of forward closures)
- state re-use heuristics (Alfons Geser, Dieter Hofbauer, Hans Zantema, J.W.)
- for non-linear TRS: quasi-deterministic tree automata (Martin Korp and Aart Middeldorp)

Completion for (plus,times)

termination proof for $\{a^2b^2 \to b^3a^3\}$
by MultumNonMulta (Dieter Hofbauer, 2006):
start with redex path, add edges, increase weights.



note change of direction:

- fuzzy semiring: zero ($= +\infty$) is highest, completion goes forward (redex $\to$ reduct)
- in standard semiring on $\mathbb{N}$: zero is lowest, completion goes backward (reduct $\to$ redex)

Exact construction of automata

R is deleting w.r.t. $>$ on Σ: for each $(l \to r) \in R$, there is $x \in l$ larger than each $y \in r$.

R match-bounded $\Rightarrow$ annotated R is deleting.

R deleting $\Rightarrow$ exists C, E with $\to_R^* = \to_C^* \circ \to_E^*$

where C terminating and context-free (SNCF) and E inverse CF (rhs of length ≤ 1, in fact $= 0$)

Corr.: R preserves REG, R^- preserves CF

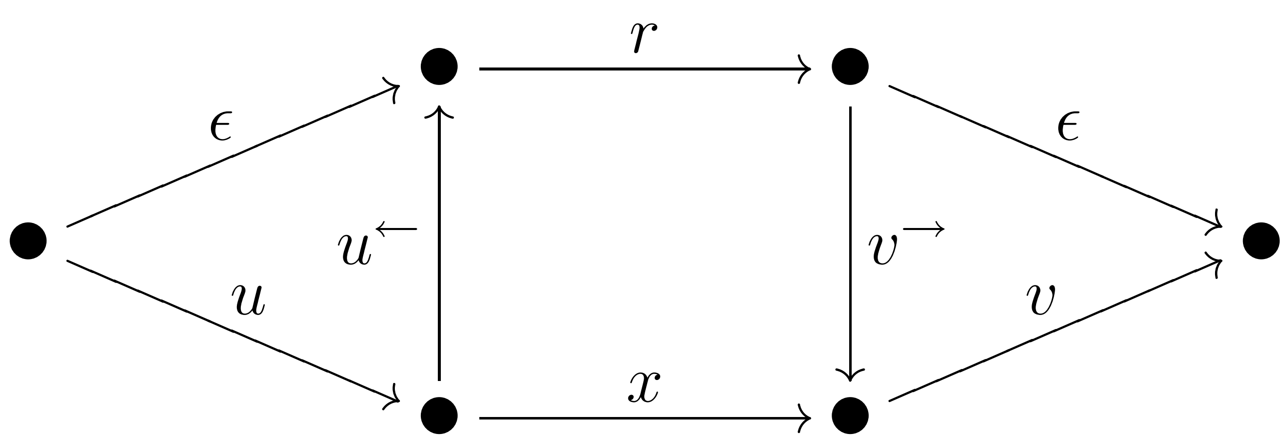
(Jörg Endrullis, Dieter Hofbauer, J.W., WST 2006)

Decomposition f. Deleting SRS

for rule $(l = uxv \to r) \in R$, use $(x \to u^{\leftarrow} r v^{\rightarrow}) \in C$,

$E = \{x x^{\leftarrow} \to \epsilon, x^{\rightarrow} x \to \epsilon \mid x \in \Sigma\}$

$\to_R^* = \to_C^* \circ \to_E^*$ by this diagram (and confluence)



and C terminating iff R deleting.

very efficient implementation in Jambox (2005)

Relative Termination

- $R \,\Big\vdash\, S$ for rewriting systems R, S with $R \supseteq S$ and $(R \setminus S)$ is terminating relative to S, that is, from R, all non-S rules "could be removed".
- $\Big\vdash$ is transitive, $R \,\Big\vdash^{*}\, \emptyset$ implies R terminating.
- $\Big\vdash^{\mathfrak{M}(W,n)} := \{(R, S) \mid$ there is a W-weighted automaton with $\leq n$ states strictly compatible with $R \setminus S$ and weakly compatible with $S\ \}$

Matrix Termination Hierarchy

hierarchy $\left|\!\frac{\mathfrak{M}(W,d)}{}\right.^{s}$ for (pairs of) rewrite systems

- semiring W (standard, arctic, fuzzy)
- dimension d
- number of proof steps s

the usual questions about hierarchies:

- is it infinite (in d, in s)?
- are the levels decidable?
- which of the obvious inclusions are strict?

Some Results on the Hierarchy (I

easy observation: these are decidable:

- $\mathfrak{M}(\mathbb{R}_{\geq 0}, d)$ Tarski, QEPCAD
- $\mathfrak{M}(\mathrm{arctic}, d)$ linear inequalities
- $\mathfrak{M}(\mathrm{matchbounds}, d)$ difference logic

relations between proofs over different domains:

- obvious inclusions $\mathbb{N} \subseteq \mathbb{Q}_{\geq 0} \subseteq \mathrm{Alg}_{\geq 0} \subseteq \mathbb{R}_{\geq 0}$
- matchbounds $\rightarrow$ multisets $\rightarrow$ weight functions $\rightarrow$ tropical

Some Results on the Hierarchy (II)

Andreas Gebhardt and J.W. (WATA 2008)

- $\mathfrak{M}(\mathbb{N}, 0) \subset \mathfrak{M}(\mathbb{N}, 1) \subset \mathfrak{M}(\mathbb{N}, 2) \subset \mathfrak{M}(\mathbb{N}, 3)$
- Amitsur-Levitski Theorem
  (polynomial identities in matrix rings)
  $\Rightarrow$ dimension hierarchy is infinite
- derivation lengths
  $\Rightarrow$ proof length hierarchy is infinite
- $\mathfrak{M}(\mathbb{N}, *)^* \subset \mathfrak{M}(\mathbb{Q}_{\geq 0}, *)^*$
  ... for relative termination

Derivational Complexity

- $\mathrm{dc}_R(s) = \sup\{k \mid \exists t, t' : |t| \le s \wedge t \to_R^k t'\}$
- termination proof by interpretation automatically gives complexity information
- general plan: from proof method, infer complexity class. (Andreas Weiermann, Dieter Hofbauer, and others)
- recent interest in polynomial bounds (Georg Moser and others).

D.C. and Weighted Automata

growth bounds for matrix interpretations:

- arctic, tropical, fuzzy (matchbounds): linear
- standard (no restrictions: exponential)
  - upper triangular shape:
    $\forall i > j : m_{ij} = 0 \land \forall i : m_{ii} \leq 1$
    implies polynomial, degree $\leq$ size $- 1$
  - easy improvement:
    degree $\leq \#\{i : \exists x \in \Sigma : [x]_{ii} > 0\}$

    all diagonals $= (1, 0, \ldots, 0, 1) \Rightarrow$ linear

Challenges in Matrix Complexity

- polynomial matrix growth is decidable

  (ET0L growth functions, formal power series)

  $\Rightarrow$ implement decision procedure
  as constraint system (Matchbox/poly)

- other semirings for growth information

- relate to density of (regular) languages

  $D_L(n) = |L \cap \Sigma^n|$

proof of the pudding: $\{a^2 \to bc, b^2 \to ac, c^2 \to ab\}$