Polynomial Bounds for $\mathbb{N}$-weighted word automata

Johannes Waldmann, HTWK Leipzig

Derivational Complexity. . .

- domain D,
- size measure $|\cdot| : D \to \mathbb{N}$,
- (derivation) relation $\to \subseteq D^2$

height (derivational complexity) of $\to$:
maximal length of a chain,
as function of size of its starting point.

$$n \mapsto \sup\{k \mid \exists s, t \in D : |s| \leq n \wedge s \to^k t\}$$

. . . of (String) Rewriting

- $\{0 \to 1\}$ is linear

$$0^k \to^k 1^k$$

- $\{01 \to 10\}$ is quadratic (bubble sort)

$$01^k \to^k 1^k0,\ 0^i1^k \to^{i \cdot k} 1^k0^i$$

- $\{01 \to 110\}$ is exponential

$$01^k \to^k 1^{2k}0,\ 0^i1 \to^* 1^{2^i}0^i$$

- etc.

Rewriting and Weighted Aut.

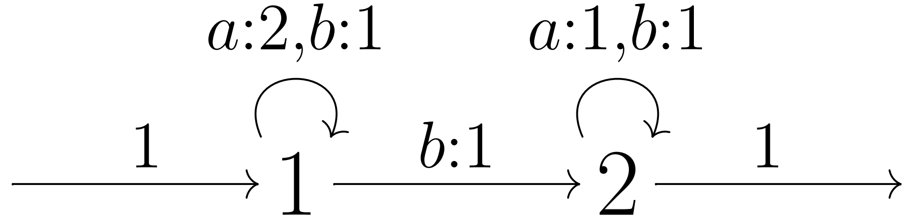
- rewriting system R on Σ
- finite $(\mathbb{N}, +, \times)$-weighted automaton A on Σ
- compatibility: $u \rightarrow_R v \Rightarrow A(u) > A(v)$.

implied by $\forall c \in \Sigma : A(i, c, i) > 0, A(f, c, f) > 0$,
and $\forall (l, r) \in R$:

$$A(i, l, f) > A(i, r, f) \text{ and } A(p, l, q) \geq A(p, r, q)$$

- proves termination of $\rightarrow_R$
- bounds derivational complexity of $\rightarrow_R$
  - in general: exponential
  - under certain conditions: polynomial

Example

weighted automaton 

matrix interpretation $a \mapsto \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, b \mapsto \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$

is compatible with
rewriting system $R = \{ab \to ba\}$, since

$$ab \mapsto \begin{pmatrix} 2 & 2 \\ 0 & 1 \end{pmatrix}, ba \mapsto \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}$$

Note: automaton can be obtained as solution of a (diophantine) constraint system

Constraint Solving

this works surprisingly well:

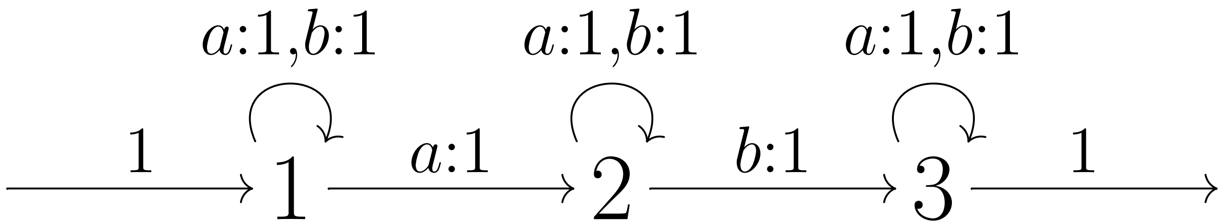
- choose (small) number of states (say, 5)
- choose (small) bit width (say, 3)
- use SAT compiler (e.g., satchmo `http://hackage.haskell.org/package/satchm`) to transform
  - Diophantine constraints for natural numbers
  - to Boolean constraints for their binary digits
- apply (Boolean) SAT solver (e.g., minisat `http://minisat.se/`)

Tight bounds

For $R = \{ab \to ba\}$, previous automaton is compatible, but not tight:

$$[a^k b] = \begin{pmatrix} 2^k & 2^k \\ 0 & 1 \end{pmatrix}$$ but $\mathrm{dc}_{\to_R}(a^k b) = k$

“better” automaton:



this interpretation is quadratically bounded (the automaton exactly counts the inversions)

Upper triangular form

$m \in \mathbb{N}^{d \times d}$ is upper triangular (U) if
$\forall i, j : (i > j \Rightarrow m_{i,j} = 0) \wedge (i = j \Rightarrow m_{i,j} \in \{0, 1\})$
Example (previous slide):

$$a \mapsto \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, b \mapsto \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

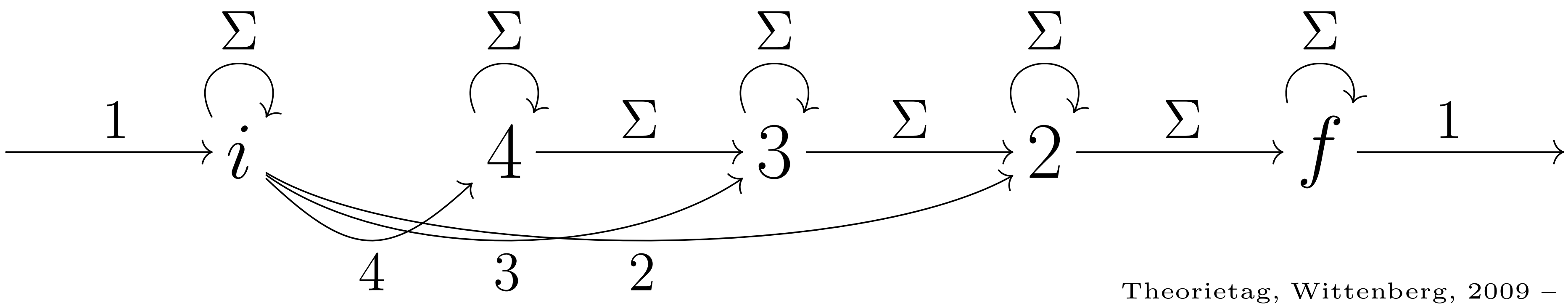
Prop: Let $[\cdot] : \Sigma \to U$. Then
$(n \mapsto \max\{[w]_{i,j} \mid w \in \Sigma^n\}) \in O(n \mapsto n^{\max(j-i,0)})$.
Cor: upper triangular interpretation gives polynomial bound on derivational complexity
Note: easy modification of constraint system

Polynomial Derivations (Ex.)

$R_d = \{ki \to jk \mid j < k\}$ over $\Sigma = \{1, 2, \ldots, d\}$.
E.g. $R_2 = \{21 \to 12, \ldots\}$,
$R_3 = R_2 \cup \{31 \to 23, 32 \to 13, \ldots\}$
For $d \geq 2$, derivation with $\Theta(n^d)$ steps:

$$w = d^n(d-1)^n \ldots 1^n \to^* \text{reverse}(w)$$

compatible (upper triangular) $\mathbb{N}$-automaton (all weights are 1)



Other Matrix Forms

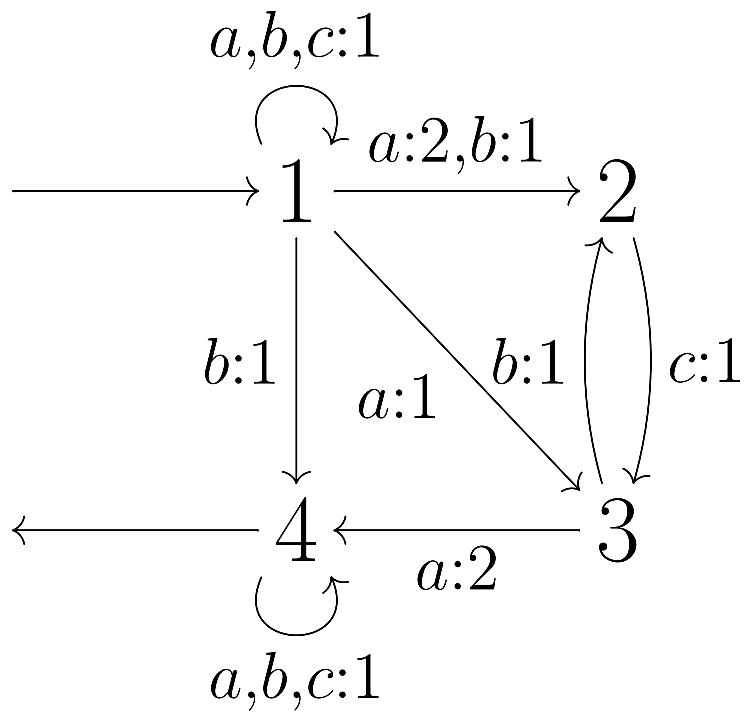
there are matrix interpretations with polynomial growth but not of upper triangular form. Example:

$$a \mapsto \begin{pmatrix} 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$b \mapsto \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$c \mapsto \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

as weighted automaton:



Related: DT0L Growth

$\mathbb{N}$-weighted word automaton
$\equiv$ action of DT0L system on Parikh vectors

- 0 : context-free, D : deterministic $\Rightarrow$ morphisms
- T : tabled $\Rightarrow$ several morphisms

$$a \mapsto \begin{pmatrix} 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (p \mapsto pqqr, q \mapsto \epsilon, r \mapsto ss, s \mapsto s)$$

$$b \mapsto \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (p \mapsto pqs, q \mapsto \epsilon, r \mapsto q, s \mapsto s),$$

$$c \mapsto \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (p \mapsto p, q \mapsto r, r \mapsto \epsilon, s \mapsto s)$$

Is $\mathbb{N}$-Automaton polynomial?

Decision procedure:

1. compute strongly connected components $A_1, \ldots, A_k$ of underlying graph.
2. if there is any arrow with weight > 1 inside one component, then growth is exponential.
3. from each component A_i, construct a (classical) automaton (all states initial and final)
4. if any A_i is ambiguous, then A is exponential.
5. Otherwise, A has polynomial growth.

Notes: degree is $<$ maximal number of SCCs on a chain of SCCs, this bound is not sharp

Ambiguity

Def: A is non-ambiguous iff each $w \in L(A)$ has exactly one accepting path.
Thm: A is non-ambiguous iff

- the reduced form (all states reachable and productive)
- of $A \times A$ (cartesian product construction)
- consists of the main diagonal only.

(e.g. Sakarovitch: Theorie des Automates)

Constraints for Ambiguity

- define $M \subseteq Q^4$: move relation of $A \times A$:
  $M = \{((p_1, p_2), (q_1, q_2)) \mid$
  $\exists c \in \Sigma : p_1 \to_c q_1 \wedge p_2 \to_c q_2)\}$
- unknown set $R \subseteq Q^2$:
  states in $A \times A$ reachable from diagonal
  diag $\subseteq R \wedge M(R) \subseteq R$
- unknown set $P \subseteq Q^2$:
  states in $A \times A$ reaching the diagonal
  diag $\subseteq P \wedge M^-(P) \subseteq P$
- reduced automaton consists of diagonal only:
  $R \cap P \subseteq$ diag

Constraints for SCCs

- $C \subseteq Q^2$ "reachable":
  - $p \xrightarrow{c:w, w>0}_A q \Rightarrow C(p,q)$,
  - C is transitive
- $S \subseteq Q^2$ "strongly connected":
  - $S = C \cap C^-$,
  - $p \xrightarrow{c:w, w>1}_A q \Rightarrow \neg S(p,q)$,
- height of $T := C \setminus S$ is $\leq b$

Constraints for Height

- height of relation $T \subseteq Q^2$:
  maximal length of a T-chain.
- express “height$(T) \leq b$”
  by constraints on $H \subseteq Q \times \{1, \ldots, b\}$
  where $H(p, h) \iff$ height of p in T is $\geq h$
- implementation:
  - $H(p, h + 1) \Rightarrow H(p, h)$,
  - $T(p, q) \Rightarrow \exists h : H(p, h) \wedge \neg H(q, h)$

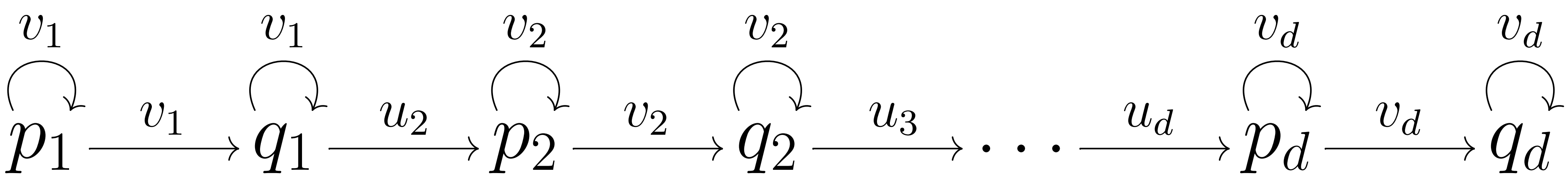
A Sharp Bound (I)

O.H. Ibarra, B. Ravikumar: *On Sparseness, ambiguity and other decision problems for acceptors and transducers*, STACS 1986.

A. Weber, H. Seidl: *On the degree of ambiguity of finite automata*, MFCS 1986, TCS 1991.

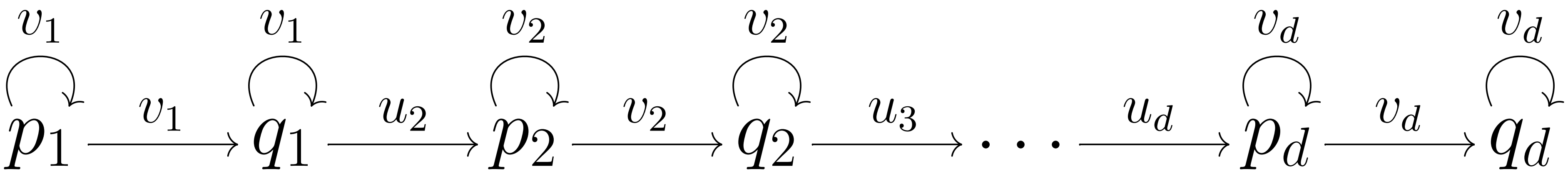
(cited in: Allauzen, Mohri, Rastogi: *General Algorithms for Testing the Ambiguitiy of Finite Automata*, 2008arXiv0802.3254A)

Thm: automaton contains graph



$\iff$ ambiguity is at least n^d.

A Sharp Bound (II)



- components can be encoded by $(p_i, p_i, q_i) \longrightarrow^* (p_i, q_i, q_i)$ in $A \times A \times A$
- $q_i \longrightarrow^* p_{i+1}$ is reachability in A

allows similar encoding as before
(bound the length of chains of components)
$|Q|^6$ unknowns, $|Q|^9$ constraints: too much for current SAT solvers

String $\to$ Term Rewriting

- same question: bound derivational complexity,
- use path-separated weighted tree automata, where interpretation of k-ary function symbol is $(\vec{x_1}, \ldots, \vec{x_k}) \mapsto M_1\vec{x_1} + \ldots + M_k\vec{x_k} + \vec{a}$
- interpretation of term (tree) t is sum of interpretations of paths (strings)
- tree growth $=$ size $\times$ path growth
- compute bound for corresponding word automaton, increase degree by one.

Summary, Discussion

summary:

- polynomial $\mathbb{N}$-automaton growth is decidable
- can be encoded as constraint system

open/todo:

- completeness (polynomially bounded R always has compatible polynomially bounded A?)
- hierarchy by size, by degree
- is $\{a^2 \to bc, b^2 \to ac, c^2 \to ab\}$ polynomial?
- lower bounds for derivational complexity
- apply in software analysis