Minimization of Weighted Automata

Andreas Maletti

Universitat Rovira i Virgili
Tarragona, Spain

Wrocław — May 19, 2010

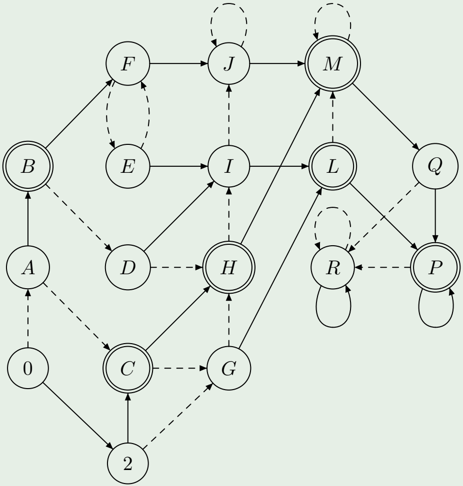
In Collaboration with

- Zoltán Ésik, University of Szeged, Hungary
- Johanna Högberg, Umeå University, Sweden
- Markus Holzer, Universität Giessen, Germany
- Jonathan May, USC, Los Angeles, CA, USA
- Heiko Vogler, TU Dresden, Germany

Table of Contents

Unweighted automata

Example (Unweighted automaton)



Unweighted automata (Cont'd)

Minimization methods

- HOPCROFT's algorithm for DFA $O(m \log n)$ [HOPCROFT 1971]
- TARJAN's algorithm for NFA $O(m \log n)$ (bisimulation minimization) [TARJAN 1987]
- NFA minimization is PSPACE-complete [JIANG, RAVIKUMAR 1993]

Special cases

- Cover automata $O(m \log n)$ [KÖRNER 2002]
- Hyper-minimization $O(m \log n)$ [GAWRYCHOWSKI, JEŻ 2009; HOLZER, $\sim$ 2010]

Unweighted automata (Cont'd)

Minimization methods

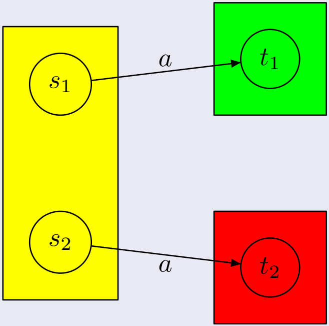
- HOPCROFT's algorithm for DFA $O(m \log n)$ [HOPCROFT 1971]
- TARJAN's algorithm for NFA $O(m \log n)$ (bisimulation minimization) [TARJAN 1987]
- NFA minimization is PSPACE-complete [JIANG, RAVIKUMAR 1993]

Special cases

- Cover automata $O(m \log n)$ [KÖRNER 2002]
- Hyper-minimization $O(m \log n)$ [GAWRYCHOWSKI, JEŻ 2009; HOLZER, $\sim$ 2010]

Minimization Procedures

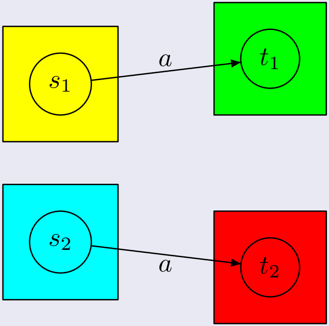
Hopcroft (for DFA)



Tarjan (for NFA)

Minimization Procedures

Hopcroft (for DFA)

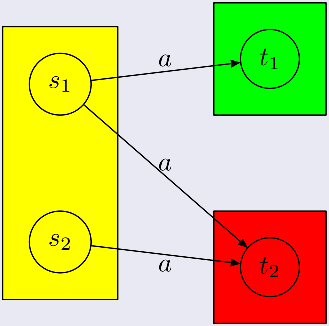


Tarjan (for NFA)

Minimization Procedures

Hopcroft (for DFA)

Tarjan (for NFA)



Minimization Procedures

Hopcroft (for DFA)

Tarjan (for NFA)

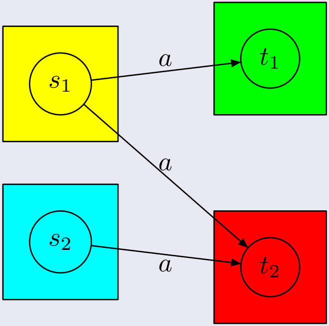
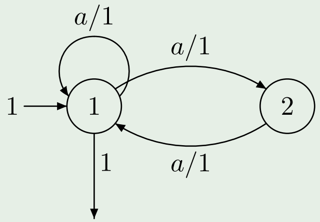


Table of Contents

Syntax

Example



Definition (Weighted automaton)

(Q, Σ, μ, I, F)

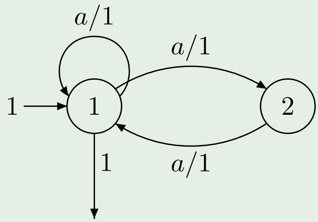
- Q finite set of states
- Σ alphabet
- $\mu \colon Q \times \Sigma \times Q \to A$
- $I \colon Q \to A$ initial weights
- $F \colon Q \to A$ final weights

References

- BERSTEL, REUTENAUER: *Rational series and their languages*. Springer, 1988
- KUICH, SALOMAA: *Semirings, automata, languages*. Springer, 1986

Syntax

Example



Definition (Weighted automaton)

(Q, Σ, μ, I, F)

- Q finite set of states
- Σ alphabet
- $\mu \colon Q \times \Sigma \times Q \to A$
- $I \colon Q \to A$ initial weights
- $F \colon Q \to A$ final weights

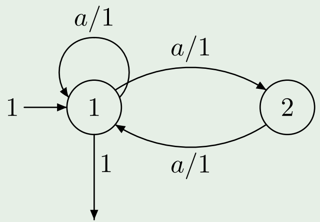
References

- BERSTEL, REUTENAUER: *Rational series and their languages*. Springer, 1988
- KUICH, SALOMAA: *Semirings, automata, languages*. Springer, 1986

Semantics

Weight structure: Semiring $A = (A, +, \cdot, 0, 1)$

Example



Definition (Semantics)

$h_\mu \colon \Sigma^* \to A^Q$

$$h_\mu(\varepsilon)_q = I(q)$$

$$h_\mu(wa)_q = \sum_{p \in Q} h_\mu(w)_p \cdot \mu(p, a, q)$$

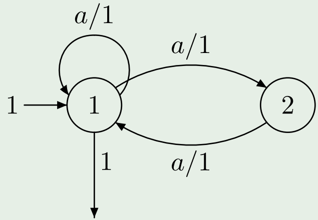
Example (using natural numbers)

$$h_\mu(a) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad h_\mu(aa) = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \qquad h_\mu(aaa) = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Semantics

Weight structure: Semiring $\mathcal{A} = (A, +, \cdot, 0, 1)$

Example



Definition (Semantics)

$h_\mu \colon \Sigma^* \to A^Q$

$$h_\mu(\varepsilon)_q = I(q)$$

$$h_\mu(wa)_q = \sum_{p \in Q} h_\mu(w)_p \cdot \mu(p, a, q)$$

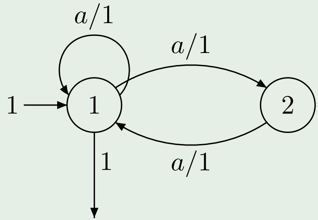
Example (using natural numbers)

$$h_\mu(a) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad h_\mu(aa) = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad h_\mu(aaa) = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Semantics

Weight structure: Semiring $A = (A, +, \cdot, 0, 1)$

Example



Definition (Semantics)

$h_\mu \colon \Sigma^* \to A^Q$

$$h_\mu(\varepsilon)_q = I(q)$$

$$h_\mu(wa)_q = \sum_{p \in Q} h_\mu(w)_p \cdot \mu(p, a, q)$$

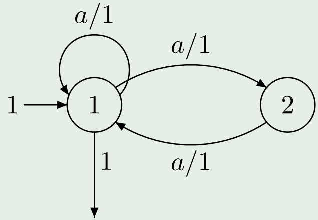
Example (using natural numbers)

$$h_\mu(a) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad h_\mu(aa) = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad h_\mu(aaa) = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Semantics

Weight structure: Semiring $A = (A, +, \cdot, 0, 1)$

Example



Definition (Semantics)

$h_\mu \colon \Sigma^* \to A^Q$

$$h_\mu(\varepsilon)_q = I(q)$$

$$h_\mu(wa)_q = \sum_{p \in Q} h_\mu(w)_p \cdot \mu(p, a, q)$$

Example (using natural numbers)

$$h_\mu(a) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad h_\mu(aa) = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad h_\mu(aaa) = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Overview

Results

Method	Nondet.	Det.	Complexity	Reference
Pushing & HOPCROFT	–	x	$O(m \log n)$	MOHRI
Forward Bisimulation	x	x	$O(m \log n)$	BUCHHOLZ
Backward Bisimulation	x	–	$O(m \log n)$	BUCHHOLZ
Backward Simulation	x	–	$O(mn)$	RANZATO, ...
Full minimization	x	x	$O(mn^2)$	BERSTEL, ...

Notation

- m: number of transitions
- n: number of states

Pushing & HOPCROFT [MOHRI]

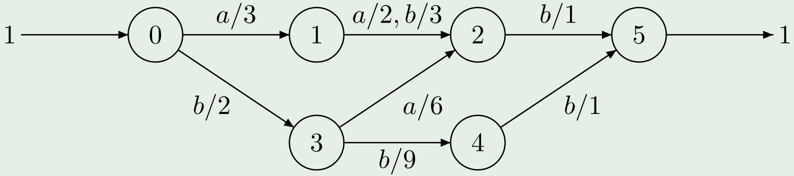
1. Push

Move weights toward the front

2. Minimize

Minimize as unweighted automaton; treat weight as part of label

Example (ala Eisner)



Pushing & HOPCROFT [MOHRI]

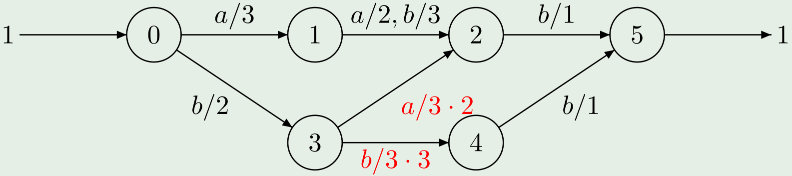
1. Push

Move weights toward the front

2. Minimize

Minimize as unweighted automaton; treat weight as part of label

Example (ala Eisner)



Pushing & HOPCROFT [MOHRI]

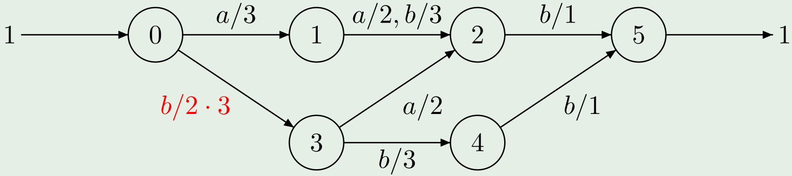
1. Push

Move weights toward the front

2. Minimize

Minimize as unweighted automaton; treat weight as part of label

Example (ala Eisner)



Pushing & HOPCROFT [MOHRI]

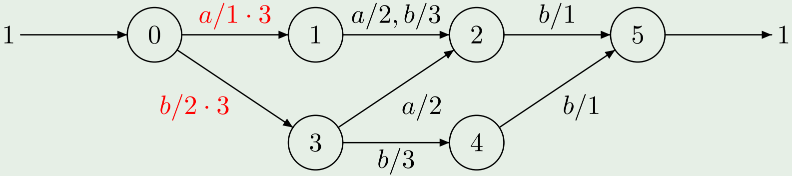
1. Push

Move weights toward the front

2. Minimize

Minimize as unweighted automaton; treat weight as part of label

Example (ala Eisner)



Pushing & HOPCROFT [MOHRI]

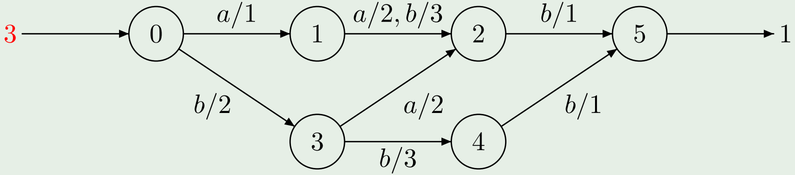
1. Push

Move weights toward the front

2. Minimize

Minimize as unweighted automaton; treat weight as part of label

Example (ala Eisner)



Pushing & HOPCROFT [MOHRI]

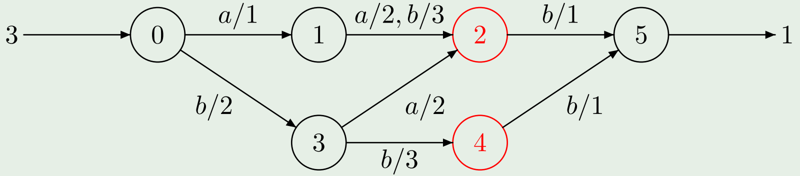
1. Push

Move weights toward the front

2. Minimize

Minimize as unweighted automaton; treat weight as part of label

Example (ala Eisner)



Pushing & HOPCROFT [MOHRI]

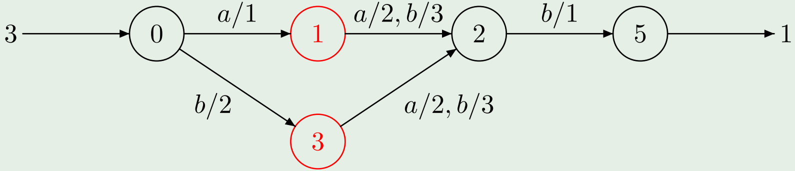
1. Push

Move weights toward the front

2. Minimize

Minimize as unweighted automaton; treat weight as part of label

Example (ala Eisner)



Pushing & HOPCROFT [MOHRI]

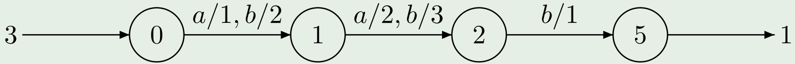
1. Push

Move weights toward the front

2. Minimize

Minimize as unweighted automaton; treat weight as part of label

Example (ala Eisner)



Pushing & HOPCROFT [MOHRI]

Prerequisites (ala EISNER)

1. Automaton deterministic
2. Semiring multiplicatively cancellative
3. Semiring allows greedy factorization

Definition (Greedy factorization)

There exists a mapping $f\colon A^2 \to A$ such that for all $a, b, c \in A$ with $c \neq 0$:

If $a|c$ and $b|c$, then $\dfrac{c}{a \cdot f(a,b)} = \dfrac{c}{b \cdot f(b,a)}$

$$\begin{tikzcd}
& \bullet \arrow[dl, "a^{-1}"'] \arrow[dr, "b^{-1}"] & \\
\bullet \arrow[dr, "f(a,b)^{-1}"'] & & \bullet \arrow[dl, "f(b,a)^{-1}"] \\
& \bullet &
\end{tikzcd}$$

Forward Bisimulation [BUCHHOLZ]

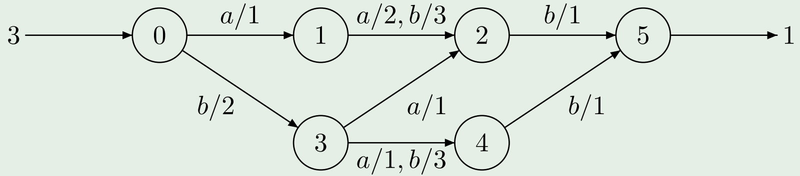
Definition (Forward bisimulation)

Equivalence relation $\equiv$ on states such that $F(p) = F(p')$ and

$$\sum_{r \in [q]} \mu(p, a, r) = \sum_{r \in [q]} \mu(p', a, r)$$

for every $p \equiv p'$, state q, and symbol a.

Example



Forward Bisimulation [Buchholz]

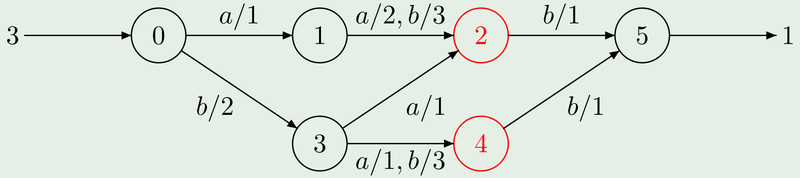
Definition (Forward bisimulation)

Equivalence relation $\equiv$ on states such that $F(p) = F(p')$ and

$$\sum_{r \in [q]} \mu(p, a, r) = \sum_{r \in [q]} \mu(p', a, r)$$

for every $p \equiv p'$, state q, and symbol a.

Example



Forward Bisimulation [BUCHHOLZ]

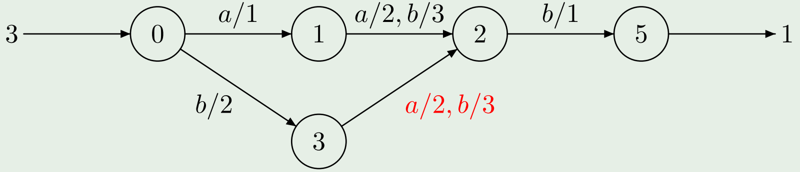
Definition (Forward bisimulation)

Equivalence relation $\equiv$ on states such that $F(p) = F(p')$ and

$$\sum_{r \in [q]} \mu(p, a, r) = \sum_{r \in [q]} \mu(p', a, r)$$

for every $p \equiv p'$, state q, and symbol a.

Example



Forward Bisimulation [Buchholz]

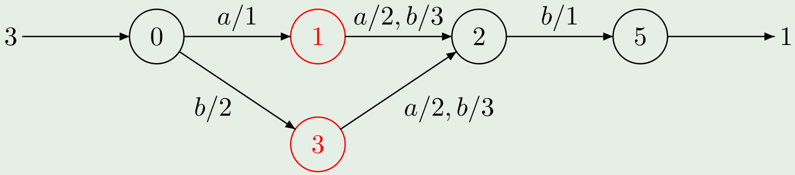
Definition (Forward bisimulation)

Equivalence relation $\equiv$ on states such that $F(p) = F(p')$ and

$$\sum_{r \in [q]} \mu(p, a, r) = \sum_{r \in [q]} \mu(p', a, r)$$

for every $p \equiv p'$, state q, and symbol a.

Example



Forward Bisimulation [BUCHHOLZ]

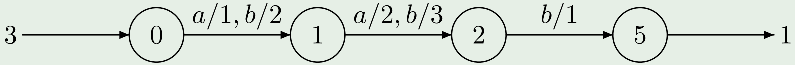
Definition (Forward bisimulation)

Equivalence relation $\equiv$ on states such that $F(p) = F(p')$ and

$$\sum_{r \in [q]} \mu(p, a, r) = \sum_{r \in [q]} \mu(p', a, r)$$

for every $p \equiv p'$, state q, and symbol a.

Example



Unweighted Forward Simulation [ABDULLA et. al.]

Definition (Forward bisimulation unweighted)

Reflexive, symmetric, and transitive relation $\equiv$ on states such that $F(p) = F(p')$ and

$$\exists q' \equiv q \colon \mu(p, a, q) \leq \mu(p', a, q')$$

for every $p \equiv p'$, state q, and symbol a.

Unweighted Forward Simulation [ABDULLA et. al.]

Definition (Forward bisimulation unweighted)

Reflexive, symmetric, and transitive relation $\equiv$ on states such that $F(p) = F(p')$ and

$$\exists q' \equiv q \colon \mu(p, a, q) \leq \mu(p', a, q')$$

for every $p \equiv p'$, state q, and symbol a.

Unweighted Forward Simulation [ABDULLA et. al.]

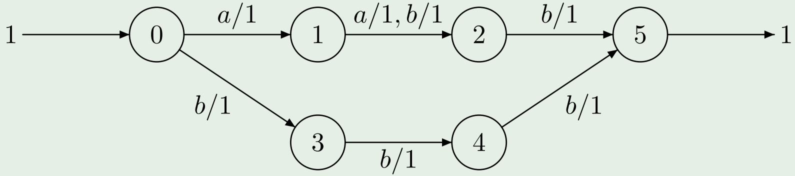
Definition (Forward simulation)

Pre-order $\leq$ on states such that $F(p) \leq F(p')$ and

$$\exists q' \geq q \colon \mu(p, a, q) \leq \mu(p', a, q')$$

for every $p \leq p'$, state q, and symbol a.

Example



$0 \quad 3 \leq 1 \quad 4 \leq 2 \leq 4 \quad 5$

Unweighted Forward Simulation [ABDULLA et. al.]

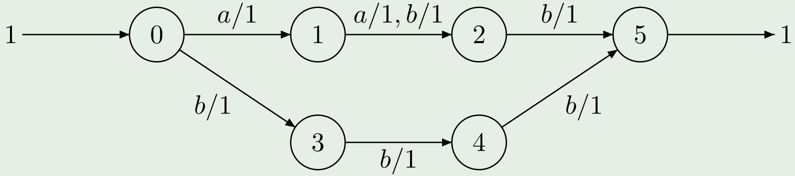
Definition (Forward simulation)

Pre-order $\leq$ on states such that $F(p) \leq F(p')$ and

$$\exists q' \geq q \colon \mu(p, a, q) \leq \mu(p', a, q')$$

for every $p \leq p'$, state q, and symbol a.

Example



$0 \quad 3 \leq 1 \quad 4 \leq 2 \leq 4 \quad 5$

Unweighted Forward Simulation [ABDULLA et. al.]

Definition (Forward simulation)

Pre-order $\leq$ on states such that $F(p) \leq F(p')$ and

$$\exists q' \geq q \colon \mu(p, a, q) \leq \mu(p', a, q')$$

for every $p \leq p'$, state q, and symbol a.

Note

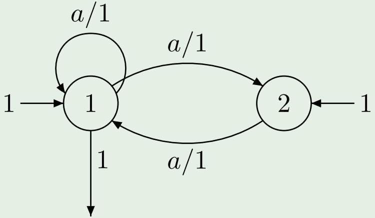
Does in general not preserve the language!

Backward Simulation [ABDULLA et. al.]

Definition (Backward (bi)simulation)

A forward (bi)simulation on the reversed automaton.

Example



$1 \leq 2 \leq 1$

Theorem

Reducing automaton by $\leq \cap \geq$ with $\leq$ a backward simulation preserves the language.

Note

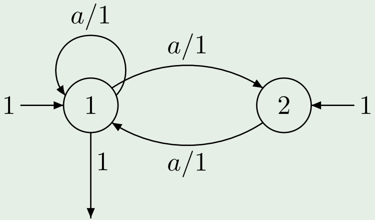
Slightly more general than backward bisimulation.

Backward Simulation [ABDULLA et. al.]

Definition (Backward (bi)simulation)

A forward (bi)simulation on the reversed automaton.

Example



$1 \leq 2 \leq 1$

Theorem

Reducing automaton by $\leq \cap \geq$ with $\leq$ a backward simulation preserves the language.

Note

Slightly more general than backward bisimulation.

Full Minimization [Berstel et. al.]

Requirements

- weight structure is a field

Procedure

- prefix compression: select prefix-closed set W such that all $h_\mu(w)$ with $w \in W$ are linearly independent
- suffix compression

References

- Berstel, Reutenauer: *Rational series and their languages*. Springer, 1988
- Berstel, Reutenauer: *Noncommutative Rational Series With Applications*, Cambridge Univ. Press, 2010

Simulation vs. Equivalence

Theorem

Two equivalent automata can be joined by forward and backward bisimulation (and weight inversion).

Extensions

- natural numbers
- integers
- rings, ...

References

- BERSTEL, REUTENAUER: *Rational series and their languages*. Springer, 1988
- BEAL, LOMBARDY, SAKAROVITCH: *On the Equivalence of $\mathbb{Z}$-Automata*, ICALP 2005

Simulation vs. Equivalence

Theorem

Two equivalent automata can be joined by forward and backward bisimulation (and weight inversion).

Extensions

- natural numbers
- integers
- rings, ...

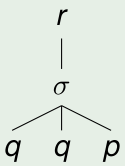
References

- BERSTEL, REUTENAUER: *Rational series and their languages*. Springer, 1988
- BEAL, LOMBARDY, SAKAROVITCH: *On the Equivalence of $\mathbb{Z}$-Automata*, ICALP 2005

Table of Contents

Syntax

Example (Transition)



Definition (Weighted tree automaton)

$(Q, \Sigma, (\mu_k)_{k \in \mathbb{N}}, F)$

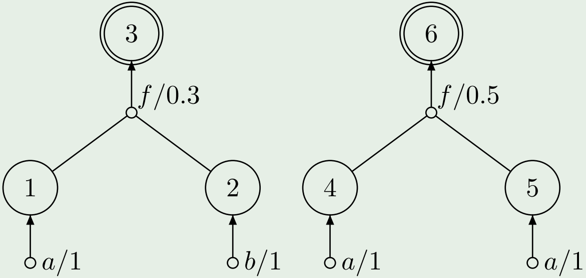
- Q finite set of states
- Σ ranked alphabet
- $\mu_k \colon Q^k \times \Sigma_k \times Q \to A$
- $F \colon Q \to A$ final weights

References

- BERSTEL, REUTENAUER: *Recognizable formal power series on trees*. Theor. Comput. Sci. 18, 1982
- BORCHARDT: *The theory of recognizable tree series*. PhD thesis, 2004

Syntax — Illustration

Example



Semantics

Definition

Let $t \in T_\Sigma(Q)$ and $W = \text{pos}(t)$.

- Run on t: map $r\colon W \to Q$ with $r(w) = t(w)$ if $t(w) \in Q$
- Weight of r

$$\text{wt}(r) = \prod_{\substack{w \in W \\ t(w) \in \Sigma}} \mu_k(r(w1), \dots, r(wk), t(w), r(w))$$

- Recognized tree series

$$(\|M\|, t) = \sum_{r \text{ run on } t} F(r(\varepsilon)) \cdot \text{wt}(r)$$

Semantics

Definition

Let $t \in T_\Sigma(Q)$ and $W = \text{pos}(t)$.

- Run on t: map $r\colon W \to Q$ with $r(w) = t(w)$ if $t(w) \in Q$
- Weight of r

$$\text{wt}(r) = \prod_{\substack{w \in W \\ t(w) \in \Sigma}} \mu_k(r(w1), \ldots, r(wk), t(w), r(w))$$

- Recognized tree series

$$(\|M\|, t) = \sum_{r \text{ run on } t} F(r(\varepsilon)) \cdot \text{wt}(r)$$

Semantics

Definition

Let $t \in T_\Sigma(Q)$ and $W = \text{pos}(t)$.

- Run on t: map $r \colon W \to Q$ with $r(w) = t(w)$ if $t(w) \in Q$
- Weight of r

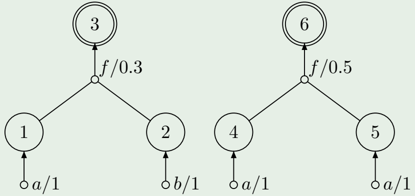
$$\text{wt}(r) = \prod_{\substack{w \in W \\ t(w) \in \Sigma}} \mu_k(r(w1), \ldots, r(wk), t(w), r(w))$$

- Recognized tree series

$$(\|M\|, t) = \sum_{r \text{ run on } t} F(r(\varepsilon)) \cdot \text{wt}(r)$$

Semantics — Illustration

Example (Automaton)



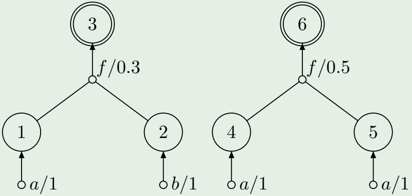
Example (Runs)

Input tree: $f(a, b)$

Runs: $6(1, 2)$ with weight 0

Semantics — Illustration

Example (Automaton)



Example (Runs)

Input tree: *f*

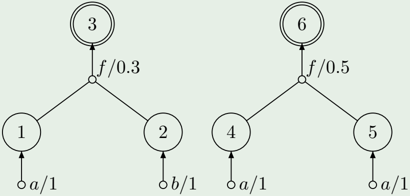
a *b*

Runs: *f* with weight

a *b*

Semantics — Illustration

Example (Automaton)



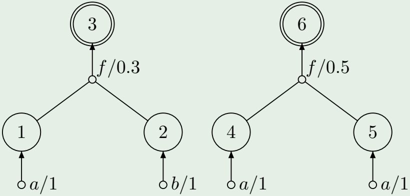
Example (Runs)

Input tree: $f(a, b)$

Runs: $f(1, b)$ with weight 1

Semantics — Illustration

Example (Automaton)



Example (Runs)

Input tree: f

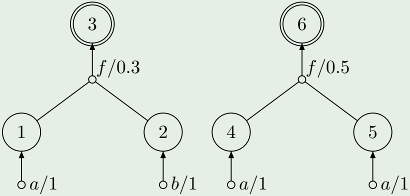
$a \quad b$

Runs: f with weight 1

1 2

Semantics — Illustration

Example (Automaton)



Example (Runs)

Input tree: $f(a, b)$

Runs: $\textcolor{red}{3}(1, 2)$ with weight 0.3

Overview

Results

Method	Nondet.	Det.	Complexity	Reference
Det. Minimization	–	x	$O(rmn)$	$\sim$
Forw. Bisimulation	x	x	$O(rm \log n)$	HÖGBERG, ...
Backw. Bisimulation	x	–	$O(r^2 m \log n)$	HÖGBERG, ...
Backw. Simulation	x	–	$O(r^2 mn)$	ABDULLA, ...
Full minimization	x	x	P	BOZAPALIDIS

Notation

- m: number of transitions
- n: number of states
- r: maximal rank of the input symbols

Forward Bisimulation [Högberg, et. al.]

Definition (Forward bisimulation)

Equivalence relation $\equiv$ on states such that $F(p) = F(p')$ and

$$\sum_{r \in [q]} \mu(\dots, p, \dots, a, r) = \sum_{r \in [q]} \mu(\dots, p', \dots, a, r)$$

for every $p \equiv p'$, symbol a, and states q and $\dots$.

Backward Bisimulation [HÖGBERG, et. al.]

Definition (Backward bisimulation)

Equivalence relation $\equiv$ on states such that

$$\sum_{q_1 \cdots q_k \in B_1 \cdots B_k} \mu(q_1, \ldots, q_k, a, p) = \sum_{q_1 \cdots q_k \in B_1 \cdots B_k} \mu(q_1, \ldots, q_k, a, p')$$

for every $p \equiv p'$, symbol a, and blocks $B_1, \ldots, B_k$.

Det. Minimization — Overview

Applicability

- Deterministic wta
- Commutative semifield (i.e. multiplicative inverses)

Roadmap

- MYHILL-NERODE congruence relation [BORCHARDT 2003]
- Determine signs of life
- Refinement

MYHILL-NERODE congruence

Definition

$p \equiv q$: there exists nonzero a such that for every context C

$$(\|M\|, C[p]) = a \cdot (\|M\|, C[q])$$

Notes

- Semifields are zero-divisor free
- Element a is unique if p is not dead

MYHILL-NERODE congruence

Definition

$p \equiv q$: there exists nonzero a such that for every context C

$$(\|M\|, C[p]) = a \cdot (\|M\|, C[q])$$

Notes

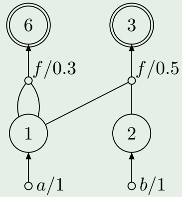
- Semifields are zero-divisor free
- Element a is unique if p is not dead

Signs of Life

Definition

Sign of life of $q \in Q$: context C such that $(\|M\|, C[q]) \neq 0$

Example



State	Sign of life	State	Sign of life
1	$f(\square, b)$	2	$f(1, \square)$
3	$\square$	6	$\square$

Stages

Definition

Stage $(\Pi, \mathrm{sol}, f, r)$:

(i) $\equiv$ refinement of $\equiv_\Pi$

(ii) $\mathrm{sol}(F) = \{\Box\}$

(iii) for live q with $p = r([q])$

$$(\|M\|, \mathrm{sol}(p)[q]) = f(q) \cdot (\|M\|, \mathrm{sol}(p)[p])$$

(iv) $\equiv_\Pi$ congruence

(v) for symbol σ and context C with live $\delta_\sigma(C[q])$

$$f(q)^{-1} \cdot c_\sigma(C[q]) \cdot f(\delta_\sigma(C[q])) = c_\sigma(C[p]) \cdot f(\delta_\sigma(C[p]))$$

where $\delta_\sigma \colon Q^k \to Q$ and $c_\sigma \colon Q^k \to A$

Stages

Definition

Stable stage $(\Pi, \mathrm{sol}, f, r)$:

(i) $\equiv$ refinement of $\equiv_\Pi$

(ii) $\mathrm{sol}(F) = \{\Box\}$

(iii) for live q with $p = r([q])$

$$(\|M\|, \mathrm{sol}(p)[q]) = f(q) \cdot (\|M\|, \mathrm{sol}(p)[p])$$

(iv) $\equiv_\Pi$ congruence

(v) for symbol σ and context C with live $\delta_\sigma(C[q])$

$$f(q)^{-1} \cdot c_\sigma(C[q]) \cdot f(\delta_\sigma(C[q])) = c_\sigma(C[p]) \cdot f(\delta_\sigma(C[p]))$$

where $\delta_\sigma \colon Q^k \to Q$ and $c_\sigma \colon Q^k \to A$

Refining a Stage

Definition

Refinement of $(\Pi, \mathrm{sol}, f, r)$: Partition Π' with $p \equiv_{\Pi'} q$ if

(i) $p \equiv_\Pi q$

(ii) $\delta_\sigma(C[p]) \equiv_\Pi \delta_\sigma(C[q])$

(iii) if $\delta_\sigma(C[p])$ is live, then

$$f(p)^{-1} \cdot c_\sigma(C[p]) \cdot f(\delta_\sigma(C[p])) = f(q)^{-1} \cdot c_\sigma(C[q]) \cdot f(\delta_\sigma(C[q]))$$

for states p and q, symbol σ, and context C

Complete Algorithm

Algorithm

$(\Pi', \mathrm{sol}, D) \leftarrow \textsc{ComputeSol}(M)$
2: **repeat**
$\quad (\Pi, \mathrm{sol}, f, r) \leftarrow \textsc{Complete}(M, \Pi', \mathrm{sol}, D)$
4: $\quad \Pi' \leftarrow \textsc{Refine}(M, \Pi, \mathrm{sol}, f, r, D)$
until $\Pi' = \Pi$
6: **return** minimized wta

Notes

- Algorithm runs in $O(rmn^4)$
- Can be optimized to run in $O(rmn)$ [$\sim$ 2008]
- Returns equivalent minimal deterministic wta

Complete Algorithm

Algorithm

$(\Pi', \mathrm{sol}, D) \leftarrow \textsc{ComputeSol}(M)$
repeat
  $(\Pi, \mathrm{sol}, f, r) \leftarrow \textsc{Complete}(M, \Pi', \mathrm{sol}, D)$
  $\Pi' \leftarrow \textsc{Refine}(M, \Pi, \mathrm{sol}, f, r, D)$
until $\Pi' = \Pi$
return minimized wta

Notes

- Algorithm runs in $O(rmn^4)$
- Can be optimized to run in $O(rmn)$ [$\sim$ 2008]
- Returns equivalent minimal deterministic wta

Simulation vs. Equivalence

Theorem

Two equivalent tree automata over fields can be joined by forward and backward bisimulation (and weight inversion).

Extensions

- natural numbers
- integers
- rings, ...

References

- BOZAPALIDIS: *Effective Construction of the Syntactic Algebra of a Recognizable Series on Trees.* Acta Inf. 1991
- ÉSIK, $\sim$: *Simulations of weighted tree automata*, 2010

Simulation vs. Equivalence

Theorem

Two equivalent tree automata over fields can be joined by forward and backward bisimulation (and weight inversion).

Extensions

- natural numbers
- integers
- rings, ...

References

- BOZAPALIDIS: *Effective Construction of the Syntactic Algebra of a Recognizable Series on Trees.* Acta Inf. 1991
- ÉSIK, $\sim$: *Simulations of weighted tree automata*, 2010

Table of Contents

Experiments

State Count

Original	Minimal	Reduction to
98	68	69%
394	308	78%
497	381	77%
727	515	71%
2701	1993	74%
3686	1766	48%

State & Transition Count

Error	Original	Minimal	Reduction to
10^{-4}	$(727, 6485)$	$(629, 6131)$	$(87\%, 95\%)$
10^{-2}	$(727, 6485)$	$(525, 3425)$	$(72\%, 53\%)$

Experiments

State Count

Original	Minimal	Reduction to
98	68	69%
394	308	78%
497	381	77%
727	515	71%
2701	1993	74%
3686	1766	48%

State & Transition Count

Error	Original	Minimal	Reduction to
10^{-4}	$(727, 6485)$	$(629, 6131)$	$(87\%, 95\%)$
10^{-2}	$(727, 6485)$	$(525, 3425)$	$(72\%, 53\%)$

The End

Thank You!