A Backward and a Forward Simulation for Weighted Tree Automata

Andreas Maletti

Θεσσαλονίκη — May 20, 2009

Author acknowledges the support of MEC grant JDCI-2007-760.

Overview

Simulations

- "Half" a bisimulation
- Can be used to make automata smaller
- Applications in logic (inseparability by logic formulae)

Rough definition

A state q simulates another state p if any transition involving p is covered by a transition involving q

Class of automata

Here for weighted tree automata over idempotent semirings

Overview

Simulations

- "Half" a bisimulation
- Can be used to make automata smaller
- Applications in logic (inseparability by logic formulae)

Rough definition

A state q simulates another state p if any transition involving p is covered by a transition involving q

Class of automata

Here for weighted tree automata over idempotent semirings

Overview

Simulations

- "Half" a bisimulation
- Can be used to make automata smaller
- Applications in logic (inseparability by logic formulae)

Rough definition

A state q simulates another state p if any transition involving p is covered by a transition involving q

Class of automata

Here for weighted tree automata over idempotent semirings

Table of contents

1. Weighted Tree Automata
2. Backward Simulation
3. Forward Simulation
4. Summary

Overview

Tree Series

- Assigns weight (e.g. a probability) to each tree
- Weights drawn from semiring; e.g. $([0, 1], \max, \cdot, 0, 1)$

Weighted Tree Automaton

- Finite representation of a tree series

Applications

- Re-ranker implementing a language model
- Representation of a parser (or parses)

Overview

Tree Series

- Assigns weight (e.g. a probability) to each tree
- Weights drawn from semiring; e.g. $([0, 1], \max, \cdot, 0, 1)$

Weighted Tree Automaton

- Finite representation of a tree series

Applications

- Re-ranker implementing a language model
- Representation of a parser (or parses)

Overview

Tree Series

- Assigns weight (e.g. a probability) to each tree
- Weights drawn from semiring; e.g. $([0,1], \max, \cdot, 0, 1)$

Weighted Tree Automaton

- Finite representation of a tree series

Applications

- Re-ranker implementing a language model
- Representation of a parser (or parses)

Syntax

Definition

Weighted tree automaton (wta) is tuple $(Q, \Sigma, \mathcal{A}, F, \mu)$ where

- Q: finite set of *states*
- Σ: ranked alphabet of *input symbols*
- $\mathcal{A} = (A, +, \cdot, 0, 1)$: semiring of *weights*
- $F \colon Q \to A$: *final weights*
- $\mu = (\mu_k)_{k \in \mathbb{N}}$ with $\mu_k \colon \Sigma_k \to A^{Q \times Q^k}$

Sample Transition

with weight $\mu_k(\sigma)_{q,q_1 \cdots q_k}$

Syntax

Definition

Weighted tree automaton (wta) is tuple $(Q, \Sigma, \mathcal{A}, F, \mu)$ where

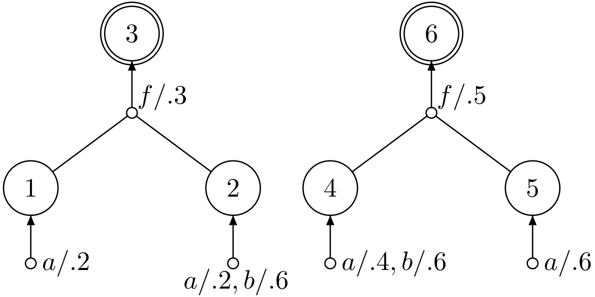
- Q: finite set of *states*
- Σ: ranked alphabet of *input symbols*
- $\mathcal{A} = (A, +, \cdot, 0, 1)$: semiring of *weights*
- $F \colon Q \to A$: *final weights*
- $\mu = (\mu_k)_{k \in \mathbb{N}}$ with $\mu_k \colon \Sigma_k \to A^{Q \times Q^k}$

Sample Transition

$q \to \sigma(q_1, \ldots, q_k)$ with weight $\mu_k(\sigma)_{q,q_1 \cdots q_k}$

Syntax — Illustration

Sample Automaton



Semantics

Definition

Let $t \in T_\Sigma$ and $q \in Q$.

- Run on t: map $r\colon \text{pos}(t) \to Q$
- Weight of r

$$\text{wt}(r) = \prod_{w \in \text{pos}(t)} \mu_k(t(w))_{r(w),r(w1)\cdots r(wk)}$$

- Weight of t in q

$$\text{wt}(t, q) = \sum_{r \text{ run on } t, r(\varepsilon)=q} \text{wt}(r)$$

- Recognized tree series: $(\|M\|, t) = \sum_{p \in Q} \text{wt}(t, p) \cdot F(p)$

Semantics

Definition

Let $t \in T_\Sigma$ and $q \in Q$.

- Run on t: map $r \colon \text{pos}(t) \to Q$
- Weight of r

$$\text{wt}(r) = \prod_{w \in \text{pos}(t)} \mu_k(t(w))_{r(w), r(w1) \cdots r(wk)}$$

- Weight of t in q

$$\text{wt}(t, q) = \sum_{r \text{ run on } t, r(\varepsilon) = q} \text{wt}(r)$$

- Recognized tree series: $(\|M\|, t) = \sum_{p \in Q} \text{wt}(t, p) \cdot F(p)$

Semantics

Definition

Let $t \in T_\Sigma$ and $q \in Q$.

- Run on t: map $r \colon \operatorname{pos}(t) \to Q$
- Weight of r

$$\operatorname{wt}(r) = \prod_{w \in \operatorname{pos}(t)} \mu_k(t(w))_{r(w), r(w1) \cdots r(wk)}$$

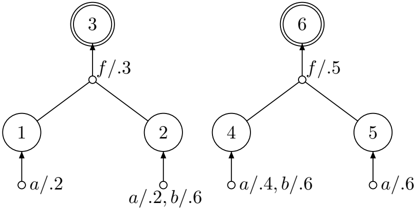
- Weight of t in q

$$\operatorname{wt}(t, q) = \sum_{r \text{ run on } t, r(\varepsilon) = q} \operatorname{wt}(r)$$

- Recognized tree series: $(\|M\|, t) = \sum_{p \in Q} \operatorname{wt}(t, p) \cdot F(p)$

Semantics — Illustration

Sample Automaton



Sample Runs for Input tree: $f(a, a)$

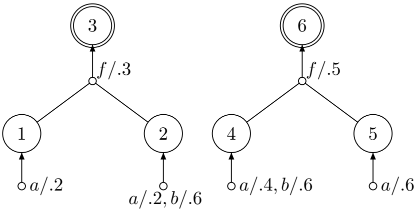
$\mathrm{wt}(6(1,2)) = 0$

$\mathrm{wt}(3(1,2)) = 0.2 \cdot 0.2 \cdot 0.3 = 0.012$

$\mathrm{wt}(6(4,5)) = 0.4 \cdot 0.6 \cdot 0.5 = 0.12$

Semantics — Illustration

Sample Automaton



Sample Runs for Input tree: $f(a, a)$

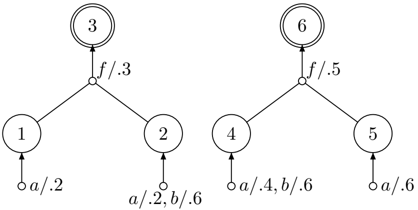
$\mathrm{wt}(6(1,2)) = 0$

$\mathrm{wt}(3(1,2)) = 0.2 \cdot 0.2 \cdot 0.3 = 0.012$

$\mathrm{wt}(6(4,5)) = 0.4 \cdot 0.6 \cdot 0.5 = 0.12$

Semantics — Illustration

Sample Automaton



Sample Runs for Input tree: $f(a, a)$

$\text{wt}(6(1,2)) = 0$
$\text{wt}(3(1,2)) = 0.2 \cdot 0.2 \cdot 0.3 = 0.012$
$\text{wt}(6(4,5)) = 0.4 \cdot 0.6 \cdot 0.5 = 0.12$

Idempotent semiring

Definition

Idempotent semiring: $1 + 1 = 1$

Definition

Natural order: $a \sqsubseteq b$ if and only if $a + b = b$

Idempotent semiring

Definition

Idempotent semiring: $1 + 1 = 1$

Definition

Natural order: $a \sqsubseteq b$ if and only if $a + b = b$

Table of contents

1 Weighted Tree Automata

2 Backward Simulation

3 Forward Simulation

4 Summary

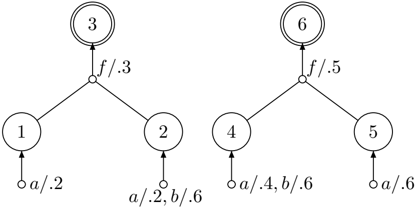
Backward simulation

Definition

A quasi-order $\preceq \subseteq Q^2$ is a backward simulation if for every $p \preceq q$, $\sigma \in \Sigma_k$, and $p_1, \ldots, p_k \in Q$ there exist $p_1 \preceq q_1, \ldots, p_k \preceq q_k$ such that

$$\mu_k(\sigma)_{p,p_1\cdots p_k} \sqsubseteq \mu_k(\sigma)_{q,q_1\cdots q_k}$$

Example



$1 \preceq 2 \preceq 4$ and $1 \preceq 5$

Principal results

Theorem

There exists a greatest backward simulation for M.

Lemma (Main lemma)

For every $t \in T_\Sigma$ *and* $p \preceq q$

$$\mathrm{wt}(t, p) \sqsubseteq \mathrm{wt}(t, q)$$

Corollary

For every $t \in T_\Sigma$ *and* $p \preceq q \preceq p$

$$\mathrm{wt}(t, p) = \mathrm{wt}(t, q)$$

Principal results

Theorem

There exists a greatest backward simulation for M.

Lemma (Main lemma)

For every $t \in T_\Sigma$ and $p \preceq q$

$$\mathrm{wt}(t, p) \sqsubseteq \mathrm{wt}(t, q)$$

Corollary

For every $t \in T_\Sigma$ and $p \preceq q \preceq p$

$$\mathrm{wt}(t, p) = \mathrm{wt}(t, q)$$

Principal results

Theorem

There exists a greatest backward simulation for M.

Lemma (Main lemma)

For every $t \in T_\Sigma$ and $p \preceq q$

$$\mathrm{wt}(t, p) \sqsubseteq \mathrm{wt}(t, q)$$

Corollary

For every $t \in T_\Sigma$ and $p \preceq q \preceq p$

$$\mathrm{wt}(t, p) = \mathrm{wt}(t, q)$$

Reducing the automaton

Definition

Let $\simeq \, = \, \preceq \cap \preceq^{-1}$. Let $(M/{\simeq}) = (Q', \Sigma, \mathcal{A}, F', \mu')$ with

- $Q' = (Q/{\simeq})$
- $F'(P) = \sum_{q \in P} F(q)$ for every $P \in Q'$
- for every $\sigma \in \Sigma_k$ and $P, P_1, \ldots, P_k \in Q'$

$$\mu'_k(\sigma)_{P,P_1 \cdots P_k} = \sum_{q_1 \in P_1, \ldots, q_k \in P_k} \mu_k(\sigma)_{\min(P), q_1 \cdots q_k}$$

Theorem

M and $(M/{\simeq})$ are equivalent.

Reducing the automaton

Definition

Let $\simeq \; = \; \preceq \cap \preceq^{-1}$. Let $(M/{\simeq}) = (Q', \Sigma, \mathcal{A}, F', \mu')$ with

- $Q' = (Q/{\simeq})$
- $F'(P) = \sum_{q \in P} F(q)$ for every $P \in Q'$
- for every $\sigma \in \Sigma_k$ and $P, P_1, \ldots, P_k \in Q'$

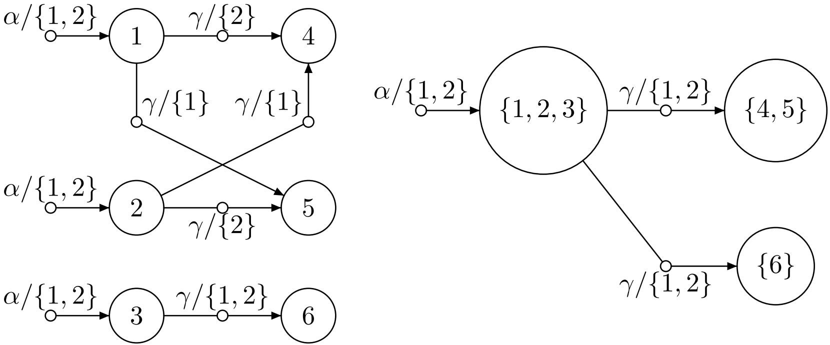
$$\mu'_k(\sigma)_{P, P_1 \cdots P_k} = \sum_{q_1 \in P_1, \ldots, q_k \in P_k} \mu_k(\sigma)_{\min(P), q_1 \cdots q_k}$$

Theorem

M and $(M/{\simeq})$ are equivalent.

Full example

Semiring $(\mathcal{P}(\{1,2\}), \cup, \cap, \emptyset, \{1,2\})$



$1 \preceq 2 \preceq 3 \preceq 1$ and $4 \preceq 5 \preceq 4 \preceq 6$

Algorithm computing the coarsest simulation

$R_0 \leftarrow Q \times Q$
$i \leftarrow 0$
repeat
  $j \leftarrow i$
  for all $\sigma \in \Sigma_k$ and $p_1, \ldots, p_k \in Q$ **do**
    $R_{i+1} \leftarrow \{(p, q) \in R_i \mid \exists (p_1, q_1), \ldots, (p_k, q_k) \in R_i :$
      $\mu_k(\sigma)_{p,p_1 \cdots p_k} \sqsubseteq \mu_k(\sigma)_{q,q_1 \cdots q_k}\}$
    $i \leftarrow i + 1$
  end for
until $R_i = R_j$

Table of contents

1 Weighted Tree Automata

2 Backward Simulation

3 Forward Simulation

4 Summary

Forward simulation

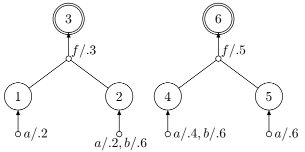
Definition

A quasi-order $\preceq \subseteq Q^2$ is a forward simulation if for every $p \preceq q$

- $F(p) \sqsubseteq F(q)$ and
- for every $\sigma \in \Sigma_k$, $1 \leq i \leq k$, and $p', q_1, \ldots, q_k \in Q$ there exist $q' \in Q$ such that $p' \preceq q'$ and

$$\mu_k(\sigma)_{p',q_1 \cdots q_{i-1} p q_{i+1} \cdots q_k} \sqsubseteq \mu_k(\sigma)_{q',q_1 \cdots q_{i-1} q q_{i+1} \cdots q_k} \quad .$$

Example



no similar states

Principal results

Theorem

There exists a greatest forward simulation for M.

Lemma (Main lemma)

For every $c \in C_\Sigma$, $p \preceq q$, and up-set $B \subseteq Q$

$$\sum_{q' \in B} \mathrm{wt}(C[p], q') \sqsubseteq \sum_{q' \in B} \mathrm{wt}(C[q], q')$$

Corollary

For every $C \in C_\Sigma$, $p \preceq q \preceq p$, and up-set $B \subseteq Q$

$$\sum_{q' \in B} \mathrm{wt}(C[p], q') = \sum_{q' \in B} \mathrm{wt}(C[q], q')$$

Principal results

Theorem

There exists a greatest forward simulation for M.

Lemma (Main lemma)

For every $c \in C_\Sigma$, $p \preceq q$, and up-set $B \subseteq Q$

$$\sum_{q' \in B} \mathrm{wt}(C[p], q') \sqsubseteq \sum_{q' \in B} \mathrm{wt}(C[q], q')$$

Corollary

For every $C \in C_\Sigma$, $p \preceq q \preceq p$, and up-set $B \subseteq Q$

$$\sum_{q' \in B} \mathrm{wt}(C[p], q') = \sum_{q' \in B} \mathrm{wt}(C[q], q')$$

Principal results

Theorem

There exists a greatest forward simulation for M.

Lemma (Main lemma)

For every $c \in C_\Sigma$, $p \preceq q$, *and up-set* $B \subseteq Q$

$$\sum_{q' \in B} \operatorname{wt}(C[p], q') \sqsubseteq \sum_{q' \in B} \operatorname{wt}(C[q], q')$$

Corollary

For every $C \in C_\Sigma$, $p \preceq q \preceq p$, *and up-set* $B \subseteq Q$

$$\sum_{q' \in B} \operatorname{wt}(C[p], q') = \sum_{q' \in B} \operatorname{wt}(C[q], q')$$

Reducing the automaton

Definition

Let $\simeq = \preceq \cap \preceq^{-1}$. Let $(M/\simeq) = (Q', \Sigma, \mathcal{A}, F', \mu')$

- $Q' = (Q/\simeq)$
- $F'(P) = F(\min(P))$ for every $P \in Q'$
- for every $\sigma \in \Sigma_k$ and $P, P_1, \ldots, P_k \in Q'$

$$\mu'_k(\sigma)_{P, P_1 \cdots P_k} = \sum_{q \in P} \mu_k(\sigma)_{q, \min(P_1), \ldots, \min(P_k)}$$

Theorem

M and $(M/\simeq)$ are equivalent.

Reducing the automaton

Definition

Let $\simeq \, = \, \preceq \cap \preceq^{-1}$. Let $(M/{\simeq}) = (Q', \Sigma, \mathcal{A}, F', \mu')$

- $Q' = (Q/{\simeq})$
- $F'(P) = F(\min(P))$ for every $P \in Q'$
- for every $\sigma \in \Sigma_k$ and $P, P_1, \ldots, P_k \in Q'$

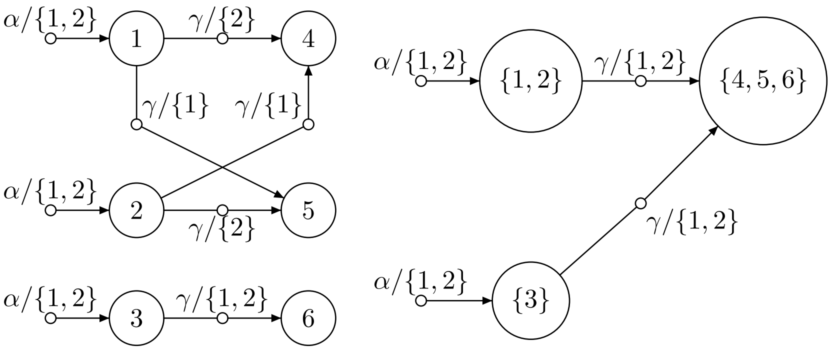
$$\mu'_k(\sigma)_{P,P_1 \cdots P_k} = \sum_{q \in P} \mu_k(\sigma)_{q,\min(P_1),\ldots,\min(P_k)}$$

Theorem

M and $(M/{\simeq})$ are equivalent.

Full example

Semiring $(\mathcal{P}(\{1,2\}), \cup, \cap, \emptyset, \{1,2\})$



$1 \preceq 2 \preceq 1 \preceq 3$ and $4 \preceq 5 \preceq 6 \preceq 4$

Algorithm computing the coarsest simulation

$R_0 \leftarrow \{(p, q) \in Q \times Q \mid F(p) \sqsubseteq F(q)\}$
$i \leftarrow 0$
repeat
$j \leftarrow i$
for all $\sigma \in \Sigma_k$, $1 \leq n \leq k$, and $p', q_1, \ldots, q_k \in Q$ **do**
$R_{i+1} \leftarrow \{(p, q) \in R_i \mid \exists (p', q') \in R_i :$
$\mu_k(\sigma)_{p', q_1 \cdots q_{n-1} p q_{n+1} \cdots q_k} \sqsubseteq \mu_k(\sigma)_{q', q_1 \cdots q_{n-1} q q_{n+1} \cdots q_k}\}$
$i \leftarrow i + 1$
end for
until $R_i = R_j$

Summary

Property	*Simulation* Backward	Forward
Generalization		
unweighted simulation	✓	✓
weighted bisimulation	✗	✗
Computation		
Admits greatest simulation	✓	✓
Easy to compute	✓	✓
Deterministic wta		
Useful	✗	✓
Better reduction	✗	✗

References

- ABDULLA, BOUAJJANI, HOLÍK, KAATI, VOJNAR
  Computing simulations over tree automata.
  Proc. TACAS, LNCS 4963, 2008.
- HÖGBERG, MALETTI, MAY
  Backward and forward bisimulation minimization of tree automata.
  Theoret. Comput. Sci., to appear.
- BORCHARDT
  The theory of recognizable tree series.
  PhD thesis, TU Dresden, 2005

Thank you for your attention!

Author acknowledges the support of MEC grant JDCI-2007-760.

References

- ABDULLA, BOUAJJANI, HOLÍK, KAATI, VOJNAR
  Computing simulations over tree automata.
  Proc. TACAS, LNCS 4963, 2008.
- HÖGBERG, MALETTI, MAY
  Backward and forward bisimulation minimization of tree automata.
  Theoret. Comput. Sci., to appear.
- BORCHARDT
  The theory of recognizable tree series.
  PhD thesis, TU Dresden, 2005

Thank you for your attention!

Universitat Rovira i Virgili
La universitat pública de Tarragona

Author acknowledges the support of MEC grant JDCI-2007-760.