Hyper-Optimization for Deterministic Tree Automata

Andreas Maletti

Institute for Natural Language Processing
University of Stuttgart, Germany

Halifax, Canada — July 19, 2013

Contents

1. Overview
2. Deterministic Tree Automaton
3. Structure of Hyper-Minimal DTA
4. Hyper-Optimization

Hyper-Optimization

Intuition

- minimize automaton allowing a finite number of errors
- several (non-isomorphic) hyper-minimal automata
- return automaton committing the least number of errors

Hyper-Optimization

Intuition

- minimize automaton allowing a finite number of errors
- several (non-isomorphic) hyper-minimal automata
- return automaton committing the least number of errors

Hyper-Optimization

Intuition

- minimize automaton allowing a finite number of errors
- several (non-isomorphic) hyper-minimal automata
- return automaton committing the least number of errors

n = number of states

m = size of the automaton

model/process	hyper-minimization	hyper-optimization
DFA	$\mathcal{O}(m \log n)$	$\mathcal{O}(mn)$
DBA	$\mathcal{O}(mn)$	???
DCA	$\mathcal{O}(mn)$	???
DTA	$\mathcal{O}(m \log n)$	$\mathcal{O}(mn)$

DTA = deterministic tree automaton
DBA / DCA = deterministic BÜCHI / Co-BÜCHI automaton

Why Hyper-Optimization?

Advantages

- makes the DTA smaller $\rightarrow$ efficiency gain
- reduces spurious, artificial effects
- conservative as it keeps the number of errors minimal

Disadvantages

- reductions sometimes rather small
- no discrimination between errors
- no non-trivial limit on the number of errors

Why Hyper-Optimization?

Advantages

- makes the DTA smaller $\rightarrow$ efficiency gain
- reduces spurious, artificial effects
- conservative as it keeps the number of errors minimal

Disadvantages

- reductions sometimes rather small
- no discrimination between errors
- no non-trivial limit on the number of errors

Contents

1 Overview

2 Deterministic Tree Automaton

3 Structure of Hyper-Minimal DTA

4 Hyper-Optimization

Deterministic Tree Automaton

Definition (GÉCSEG, STEINBY 1984)

(Q, Σ, δ, F) deterministic tree automaton (DTA)

- Q finite set — *states*
- Σ ranked alphabet — *input symbols*
- $\delta \colon \Sigma(Q) \to Q$ — *transitions*
- $F \subseteq Q$ — *final states*

Definition

transition function extends to $\delta \colon T_\Sigma(Q) \to Q$ by

$$\delta(q) = q$$
$$\delta(\sigma(t_1, \ldots, t_k)) = \delta(\sigma(\delta(t_1), \ldots, \delta(t_k)))$$

Deterministic Tree Automaton

Definition (Gécseg, Steinby 1984)

(Q, Σ, δ, F) deterministic tree automaton (DTA)

- Q finite set — *states*
- Σ ranked alphabet — *input symbols*
- $\delta \colon \Sigma(Q) \to Q$ — *transitions*
- $F \subseteq Q$ — *final states*

Definition

transition function extends to $\delta \colon T_\Sigma(Q) \to Q$ by

$$\delta(q) = q$$
$$\delta(\sigma(t_1, \ldots, t_k)) = \delta(\sigma(\delta(t_1), \ldots, \delta(t_k)))$$

Deterministic Tree Automaton

Example

- states q_α, q_β (nonfinal) and q_γ, q_σ (final)
- nullary input symbols α, β, γ and binary σ
- for all nullary symbols π, π'

$$\pi \mapsto q_\pi \qquad \sigma(q_\pi, q_{\pi'}) \mapsto q_\sigma \qquad \sigma(q_\alpha, q_\sigma) \mapsto q_\sigma$$

Evaluating a tree

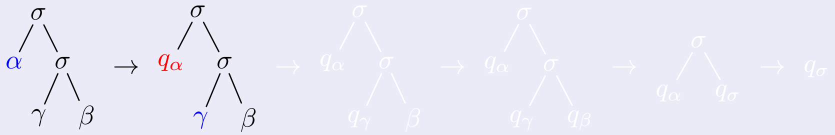
Deterministic Tree Automaton

Example

- states q_α, q_β (nonfinal) and q_γ, q_σ (final)
- nullary input symbols α, β, γ and binary σ
- for all nullary symbols π, π'

$$\pi \mapsto q_\pi \qquad \sigma(q_\pi, q_{\pi'}) \mapsto q_\sigma \qquad \sigma(q_\alpha, q_\sigma) \mapsto q_\sigma$$

Evaluating a tree



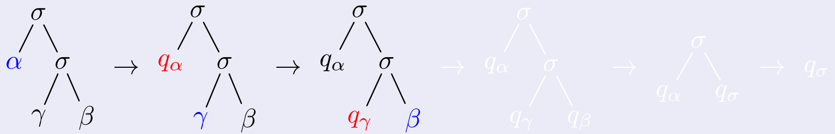
Deterministic Tree Automaton

Example

- states q_α, q_β (nonfinal) and q_γ, q_σ (final)
- nullary input symbols α, β, γ and binary σ
- for all nullary symbols π, π'

$$\pi \mapsto q_\pi \qquad \sigma(q_\pi, q_{\pi'}) \mapsto q_\sigma \qquad \sigma(q_\alpha, q_\sigma) \mapsto q_\sigma$$

Evaluating a tree



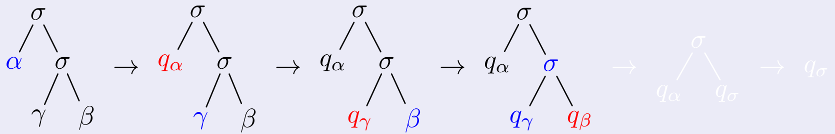
Deterministic Tree Automaton

Example

- states q_α, q_β (nonfinal) and q_γ, q_σ (final)
- nullary input symbols α, β, γ and binary σ
- for all nullary symbols π, π'

$$\pi \mapsto q_\pi \qquad \sigma(q_\pi, q_{\pi'}) \mapsto q_\sigma \qquad \sigma(q_\alpha, q_\sigma) \mapsto q_\sigma$$

Evaluating a tree



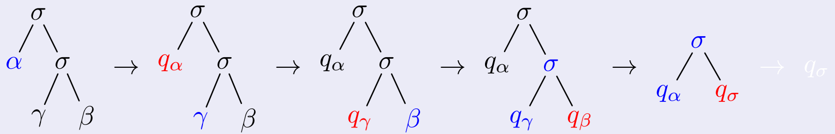
Deterministic Tree Automaton

Example

- states q_α, q_β (nonfinal) and q_γ, q_σ (final)
- nullary input symbols α, β, γ and binary σ
- for all nullary symbols π, π'

$$\pi \mapsto q_\pi \qquad \sigma(q_\pi, q_{\pi'}) \mapsto q_\sigma \qquad \sigma(q_\alpha, q_\sigma) \mapsto q_\sigma$$

Evaluating a tree



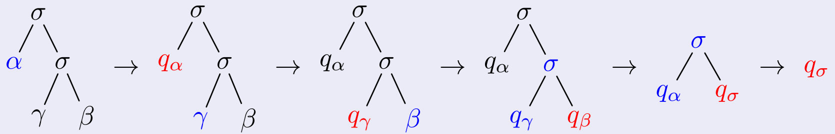
Deterministic Tree Automaton

Example

- states q_α, q_β (nonfinal) and q_γ, q_σ (final)
- nullary input symbols α, β, γ and binary σ
- for all nullary symbols π, π'

$$\pi \mapsto q_\pi \qquad \sigma(q_\pi, q_{\pi'}) \mapsto q_\sigma \qquad \sigma(q_\alpha, q_\sigma) \mapsto q_\sigma$$

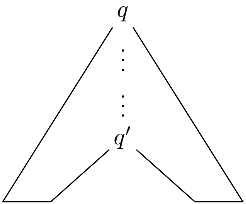
Evaluating a tree



Deterministic Tree Automaton

Shorthands

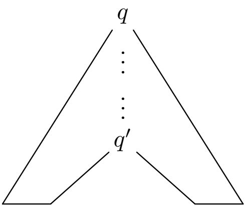
- $L(M)^q_{q'} = \{c \in C_\Sigma \mid \delta(c[q']) = q\}$
- $L(M)_{q'} = \bigcup_{f \in F} L(M)^f_{q'}$
- $L(M)^q = \delta^{-1}(q) \cap T_\Sigma$

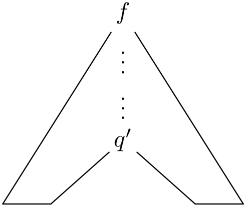


Deterministic Tree Automaton

Shorthands

- $L(M)^q_{q'} = \{c \in C_\Sigma \mid \delta(c[q']) = q\}$
- $L(M)_{q'} = \bigcup_{f \in F} L(M)^f_{q'}$
- $L(M)^q = \delta^{-1}(q) \cap T_\Sigma$

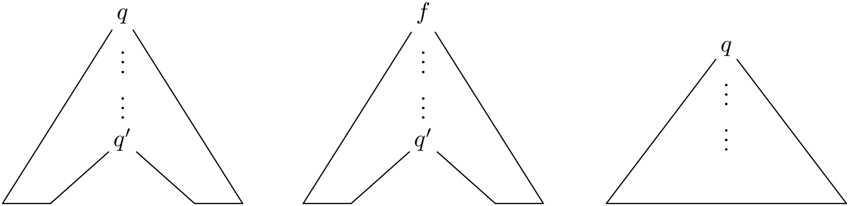




Deterministic Tree Automaton

Shorthands

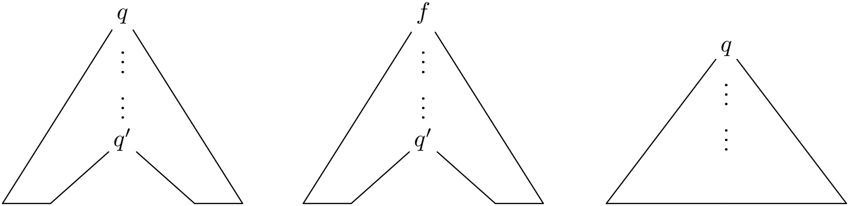
- $L(M)^{q}_{q'} = \{c \in C_\Sigma \mid \delta(c[q']) = q\}$
- $L(M)_{q'} = \bigcup_{f \in F} L(M)^{f}_{q'}$
- $L(M)^{q} = \delta^{-1}(q) \cap T_\Sigma$



Deterministic Tree Automaton

Shorthands

- $L(M)^{q}_{q'} = \{c \in C_\Sigma \mid \delta(c[q']) = q\}$
- $L(M)_{q'} = \bigcup_{f \in F} L(M)^{f}_{q'}$
- $L(M)^{q} = \delta^{-1}(q) \cap T_\Sigma$



Definition (Recognized tree language)

$L(M) = \bigcup_{f \in F} L(M)^{f}$

Hyper-Minimization

Definition

states q and q' are

- equivalent if $L(M)_q = L(M)_{q'}$
- almost equivalent if $L(M)_q$ and $L(M)_{q'}$ are almost equal

Theorem (known)

- *trim DTA is minimal*
  $\iff$ *no different, but equivalent states*
- *minimal DTA is hyper-minimal*
  $\iff$ *no different, but almost equivalent special states*

Hyper-Minimization

Definition

states q and q' are

- equivalent if $L(M)_q = L(M)_{q'}$
- almost equivalent if $L(M)_q$ and $L(M)_{q'}$ are almost equal

Theorem (known)

- *trim DTA is minimal*
  $\iff$ *no different, but equivalent states*
- *minimal DTA is hyper-minimal*
  $\iff$ *no different, but almost equivalent special states*

Hyper-Minimization

Definition

states q and q' are

- equivalent if $L(M)_q = L(M)_{q'}$
- almost equivalent if $L(M)_q$ and $L(M)_{q'}$ are almost equal

Theorem (known)

- *trim DTA is minimal*
  $\iff$ *no different, but equivalent states*
- *minimal DTA is hyper-minimal*
  $\iff$ *no different, but almost equivalent special states*

Contents

1 Overview

2 Deterministic Tree Automaton

3 Structure of Hyper-Minimal DTA

4 Hyper-Optimization

Almost Equivalent DTA

Definition

DTA M and N are almost equivalent
if they recognize almost equal tree languages

Definition

- state q is a kernel state if $L(M)^q$ is infinite
- $\text{Ker}(M) = \{q \in Q \mid q \text{ kernel state}\}$
- $\text{Pre}(M) = Q - \text{Ker}(M)$ preamble states

Almost Equivalent DTA

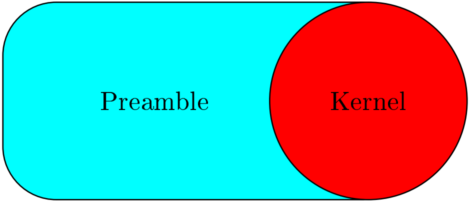
Definition

DTA M and N are almost equivalent
if they recognize almost equal tree languages

Definition

- state q is a kernel state if $L(M)^q$ is infinite
- $\mathrm{Ker}(M) = \{q \in Q \mid q \text{ kernel state}\}$
- $\mathrm{Pre}(M) = Q - \mathrm{Ker}(M)$ preamble states

Kernel and Preamble States



Almost equivalent DTA

DTA $M = (Q, \Sigma, \delta, F)$ $\qquad N = (P, \Sigma, \mu, G)$

Theorem

If M and N are hyper-minimal and almost equivalent, then there exists a bijection $h\colon Q \to P$ such that

1. *h is bijective on kernels* $\qquad \mathrm{Ker}(M) \times \mathrm{Ker}(N)$
2. $h(q) \in G \iff q \in F$ $\qquad$ *for all* $q \in \mathrm{Ker}(M)$
3. $h(\delta(s)) = \mu(h(s))$ *for every* $s \in \Sigma(Q) - \{s \in \Sigma(\mathrm{Pre}(M)) \mid \delta(s) \in \mathrm{Ker}(M)\}$

Almost equivalent DTA

DTA $M = (Q, \Sigma, \delta, F)$ $N = (P, \Sigma, \mu, G)$

Theorem

If M and N are hyper-minimal and almost equivalent, then there exists a bijection $h \colon Q \to P$ such that

1. *h is bijective on kernels* $\mathrm{Ker}(M) \times \mathrm{Ker}(N)$
2. $h(q) \in G \iff q \in F$ *for all* $q \in \mathrm{Ker}(M)$
3. $h(\delta(s)) = \mu(h(s))$ *for every* $s \in \Sigma(Q) - \{s \in \Sigma(\mathrm{Pre}(M)) \mid \delta(s) \in \mathrm{Ker}(M)\}$

Almost equivalent DTA

DTA $M = (Q, \Sigma, \delta, F)$ $N = (P, \Sigma, \mu, G)$

Theorem

If M and N are hyper-minimal and almost equivalent, then there exists a bijection $h \colon Q \to P$ such that

1. *h is bijective on kernels* $\mathrm{Ker}(M) \times \mathrm{Ker}(N)$
2. $h(q) \in G \iff q \in F$ *for all* $q \in \mathrm{Ker}(M)$
3. $h(\delta(s)) = \mu(h(s))$ *for every* $s \in \Sigma(Q) - \{s \in \Sigma(\mathrm{Pre}(M)) \mid \delta(s) \in \mathrm{Ker}(M)\}$

Almost equivalent DTA

DTA $M = (Q, \Sigma, \delta, F)$ $N = (P, \Sigma, \mu, G)$

Theorem

If M and N are hyper-minimal and almost equivalent, then there exists a bijection $h \colon Q \to P$ such that

1. *h is bijective on kernels* $\text{Ker}(M) \times \text{Ker}(N)$
2. $h(q) \in G \iff q \in F$ *for all* $q \in \text{Ker}(M)$
3. $h(\delta(s)) = \mu(h(s))$ *for every* $s \in \Sigma(Q) - \{s \in \Sigma(\text{Pre}(M)) \mid \delta(s) \in \text{Ker}(M)\}$

Intuition

hyper-minimal and almost equivalent DTA can differ in

1. the finality of preamble states
2. transitions from exclusively preamble states to a kernel state

Almost equivalent DTA

DTA $M = (Q, \Sigma, \delta, F)$ $\quad N = (P, \Sigma, \mu, G)$

Theorem

If M and N are hyper-minimal and almost equivalent, then there exists a bijection $h \colon Q \to P$ such that

1. *h is bijective on kernels* $\quad \mathrm{Ker}(M) \times \mathrm{Ker}(N)$
2. $h(q) \in G \iff q \in F$ $\quad$ *for all* $q \in \mathrm{Ker}(M)$
3. $h(\delta(s)) = \mu(h(s))$ *for every* $s \in \Sigma(Q) - \{s \in \Sigma(\mathrm{Pre}(M)) \mid \delta(s) \in \mathrm{Ker}(M)\}$

Intuition

hyper-minimal and almost equivalent DTA can differ in

1. the finality of preamble states
2. transitions from exclusively preamble states to a kernel state

Almost equivalent DTA

DTA $M = (Q, \Sigma, \delta, F)$ $\qquad$ $N = (P, \Sigma, \mu, G)$

Theorem

If M and N are hyper-minimal and almost equivalent, then there exists a bijection $h \colon Q \to P$ such that

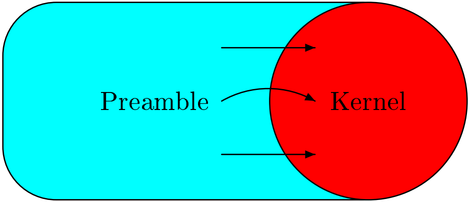
1. *h is bijective on kernels* $\qquad$ $\mathrm{Ker}(M) \times \mathrm{Ker}(N)$
2. $h(q) \in G \iff q \in F$ $\qquad$ *for all* $q \in \mathrm{Ker}(M)$
3. $h(\delta(s)) = \mu(h(s))$ *for every* $s \in \Sigma(Q) - \{s \in \Sigma(\mathrm{Pre}(M)) \mid \delta(s) \in \mathrm{Ker}(M)\}$

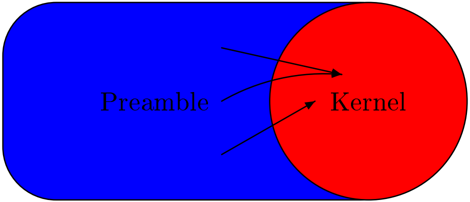
Intuition

hyper-minimal and almost equivalent DTA can differ in

1. the finality of preamble states
2. transitions from exclusively preamble states to a kernel state

Almost equivalent DTA





Hyper-Optimization

Rough Outline

given DTA M:

1. minimize M — $\mathcal{O}(m \log n)$
2. hyper-minimize M — $\mathcal{O}(m \log n)$
3. optimize hyper-minimal DTA M — $\mathcal{O}(mn)$

Hyper-Optimization

Rough Outline

given DTA M:

1. minimize M — $\mathcal{O}(m \log n)$
2. hyper-minimize M — $\mathcal{O}(m \log n)$
3. optimize hyper-minimal DTA M — $\mathcal{O}(mn)$

Hyper-Optimization

Rough Outline

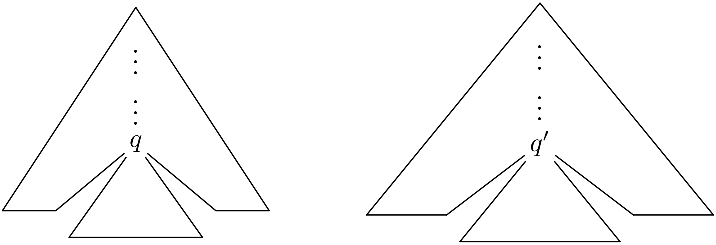
given DTA M:

1. minimize M $\mathcal{O}(m \log n)$
2. hyper-minimize M $\mathcal{O}(m \log n)$
3. optimize hyper-minimal DTA M $\mathcal{O}(mn)$

State Merging

Definition

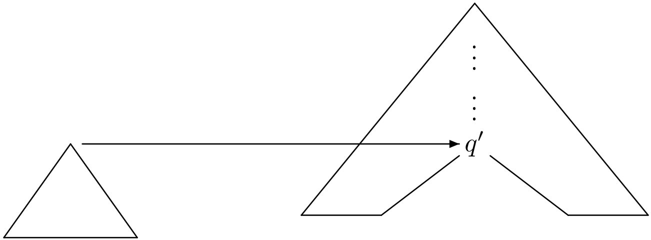
merge of q into q': redirect all transitions leading to q into q'



State Merging

Definition

merge of q into q': redirect all transitions leading to q into q'



Hyper-Minimization

Lemma (known)

Merging q into q' yields an almost equivalent DTA if

- *q and q' are almost equivalent*
- *q is a preamble state*

Theorem (known)

DTA hyper-minimal $\iff$ no different, but almost equivalent states involving a preamble state

Hyper-Minimization

Lemma (known)

Merging q into q' yields an almost equivalent DTA if

- *q and q' are almost equivalent*
- *q is a preamble state*

Theorem (known)

DTA hyper-minimal $\iff$ *no different, but almost equivalent states involving a preamble state*

Contents

1 Overview

2 Deterministic Tree Automaton

3 Structure of Hyper-Minimal DTA

4 Hyper-Optimization

Error Attribution

hyper-minimal DTA N minimal input DTA M

Intuition

each error tree $t \in L(M) \ominus L(N)$ occurs either at

- a preamble state of N $\mu(t) \in \mathrm{Pre}(N)$
- a kernel state of N $\mu(t) \in \mathrm{Ker}(N)$

Observations

- $\delta(t) \in \mathrm{Pre}(M)$ if $\mu(t) \in \mathrm{Pre}(N)$
- $L(N)^{\mu(t)} = \bigcup_{q \in B} L(M)^q$

$$B = \{q \in Q \mid q \text{ almost equivalent to } \delta(t)\}$$

Error Attribution

hyper-minimal DTA N minimal input DTA M

Intuition

each error tree $t \in L(M) \ominus L(N)$ occurs either at

- a preamble state of N $\mu(t) \in \mathrm{Pre}(N)$
- a kernel state of N $\mu(t) \in \mathrm{Ker}(N)$

Observations

- $\delta(t) \in \mathrm{Pre}(M)$ if $\mu(t) \in \mathrm{Pre}(N)$
- $L(N)^{\mu(t)} = \bigcup_{q \in B} L(M)^q$

$$B = \{q \in Q \mid q \text{ almost equivalent to } \delta(t)\}$$

Error Attribution

hyper-minimal DTA N minimal input DTA M

Intuition

each error tree $t \in L(M) \ominus L(N)$ occurs either at

- a preamble state of N $\mu(t) \in \mathrm{Pre}(N)$
- a kernel state of N $\mu(t) \in \mathrm{Ker}(N)$

Observations

- $\delta(t) \in \mathrm{Pre}(M)$ if $\mu(t) \in \mathrm{Pre}(N)$
- $L(N)^{\mu(t)} = \bigcup_{q \in B} L(M)^q$

$$B = \{q \in Q \mid q \text{ almost equivalent to } \delta(t)\}$$

Preamble State Errors

Theorem

The preamble state $p \in \mathrm{Pre}(N)$ *causes*

- $\sum_{q \in B \cap F} |L(M)^q|$ *errors if* $p \notin G$
- $\sum_{q \in B - F} |L(M)^q|$ *errors if* $p \in G$

where $t \in L(N)^p$ *and* $B = \{q \in Q \mid q \text{ almost equivalent to } \delta(t)\}$.

Observations

- $B \subseteq \mathrm{Pre}(M)$
- $|L(M)^q|$ easily computable for $q \in \mathrm{Pre}(M)$ $\mathcal{O}(m)$

Preamble State Errors

Theorem

The preamble state $p \in \mathrm{Pre}(N)$ *causes*

- $\sum_{q \in B \cap F} |L(M)^q|$ *errors if* $p \notin G$
- $\sum_{q \in B - F} |L(M)^q|$ *errors if* $p \in G$

where $t \in L(N)^p$ *and* $B = \{q \in Q \mid q \text{ almost equivalent to } \delta(t)\}$.

Observations

- $B \subseteq \mathrm{Pre}(M)$
- $|L(M)^q|$ easily computable for $q \in \mathrm{Pre}(M)$ $\mathcal{O}(m)$

Preamble State Errors

Theorem

The preamble state $p \in \mathrm{Pre}(N)$ *causes*

- $\sum_{q \in B \cap F} |L(M)^q|$ *errors if* $p \notin G$
- $\sum_{q \in B - F} |L(M)^q|$ *errors if* $p \in G$

where $t \in L(N)^p$ *and* $B = \{q \in Q \mid q \text{ almost equivalent to } \delta(t)\}$.

Observations

- $B \subseteq \mathrm{Pre}(M)$
- $|L(M)^q|$ easily computable for $q \in \mathrm{Pre}(M)$ $\mathcal{O}(m)$

Error Attribution

hyper-minimal DTA N minimal input DTA M

Intuition

Each error tree $t \in L(M) \ominus L(N)$ occurs either at

- a preamble state of N $\mu(t) \in \mathrm{Pre}(N)$
- a kernel state of N $\mu(t) \in \mathrm{Ker}(N)$

Observations

If $\mu(t) \in \mathrm{Ker}(N)$, then there exists a left-most position $p \in \mathrm{pos}(t)$ such that

- $\delta(t|_p) \in \mathrm{Ker}(N)$
- $\delta(t|_{pw}) \in \mathrm{Pre}(N)$ for all $w \neq \varepsilon$

Error Attribution

hyper-minimal DTA N minimal input DTA M

Intuition

Each error tree $t \in L(M) \ominus L(N)$ occurs either at

- a preamble state of N $\mu(t) \in \mathrm{Pre}(N)$
- a kernel state of N $\mu(t) \in \mathrm{Ker}(N)$

Observations

If $\mu(t) \in \mathrm{Ker}(N)$, then there exists a left-most position $p \in \mathrm{pos}(t)$ such that

- $\delta(t|_p) \in \mathrm{Ker}(N)$
- $\delta(t|_{pw}) \in \mathrm{Pre}(N)$ for all $w \neq \varepsilon$

Kernel State Errors

Example

DTA $M = (Q, \Sigma, \delta, \{q_\alpha\})$ with $L(M) = T_\Sigma - \{\beta, \sigma(\beta, \beta)\}$

- $Q = \{q_\alpha, q_\beta, q_\sigma\}$
- $\Sigma = \{\alpha^{(0)}, \beta^{(0)}, \sigma^{(2)}\}$
- for all $(q, q') \in Q^2 - \{(q_\beta, q_\beta)\}$

$$\delta(\alpha) = q_\alpha \qquad \delta(\beta) = q_\beta \qquad \delta(\sigma(q_\beta, q_\beta)) = q_\sigma \qquad \delta(\sigma(q, q')) = q_\alpha$$

Almost equivalent hyper-minimal (single-state) DTA N with $L(N) = T_\Sigma$

Observation

- error $\sigma(\beta, \beta)$ processed in kernel state $\top$ of N
- both $\beta \to \top$ transitions switch from exclusively preamble states to a kernel state

Kernel State Errors

Example

DTA $M = (Q, \Sigma, \delta, \{q_\alpha\})$ with $L(M) = T_\Sigma - \{\beta, \sigma(\beta, \beta)\}$

- $Q = \{q_\alpha, q_\beta, q_\sigma\}$
- $\Sigma = \{\alpha^{(0)}, \beta^{(0)}, \sigma^{(2)}\}$
- for all $(q, q') \in Q^2 - \{(q_\beta, q_\beta)\}$

$$\delta(\alpha) = q_\alpha \qquad \delta(\beta) = q_\beta \qquad \delta(\sigma(q_\beta, q_\beta)) = q_\sigma \qquad \delta(\sigma(q, q')) = q_\alpha$$

Almost equivalent hyper-minimal (single-state) DTA N with $L(N) = T_\Sigma$

Observation

- error $\sigma(\beta, \beta)$ processed in kernel state $\top$ of N
- both $\beta \to \top$ transitions switch from exclusively preamble states to a kernel state

Kernel State Errors

Definition

left-most switch contexts:

$$\mathrm{LC} = \{c \in C_\Sigma \mid \forall w \in \mathrm{pos}(c)\colon\ w \sqsubset \mathrm{pos}_\Box(c) \text{ implies } [\delta(c|_w)] \subseteq \mathrm{Pre}(M)\}$$

Lemma

For all $q \sim q'$ we have

Kernel State Errors

Definition

left-most switch contexts:

$$\mathrm{LC} = \{c \in C_\Sigma \mid \forall w \in \mathrm{pos}(c) \colon w \sqsubset \mathrm{pos}_\square(c) \text{ implies } [\delta(c|_w)] \subseteq \mathrm{Pre}(M)\}$$

$$\overline{C}_M = C_\Sigma(\mathrm{Pre}(M)) \cap \Sigma(Q \cup \{\square\})$$

Lemma

For all $q \sim q'$ we have

Kernel State Errors

Definition

left-most switch contexts:

$$\mathrm{LC} = \{c \in C_\Sigma \mid \forall w \in \mathrm{pos}(c) \colon w \sqsubset \mathrm{pos}_\square(c) \text{ implies } [\delta(c|_w)] \subseteq \mathrm{Pre}(M)\}$$

$$\overline{C}_M = C_\Sigma(\mathrm{Pre}(M)) \cap \Sigma(Q \cup \{\square\})$$

Lemma

For all $q \sim q'$ we have

$$d(q, q) = 0$$

Kernel State Errors

Definition

left-most switch contexts:

$$\mathrm{LC} = \{c \in C_\Sigma \mid \forall w \in \mathrm{pos}(c)\colon w \sqsubset \mathrm{pos}_\square(c) \text{ implies } [\delta(c|_w)] \subseteq \mathrm{Pre}(M)\}$$

$$\overline{C}_M = C_\Sigma(\mathrm{Pre}(M)) \cap \Sigma(Q \cup \{\square\})$$

Lemma

For all $q \sim q'$ we have

$$d(q, q) = 0$$

$$d(q, q') = \Big(\sum_{\substack{c \in \overline{C}_M \\ c = \sigma(q_1, \ldots, q_i, \square, q_{i+1}, \ldots, q_k) \\ [q_1], \ldots, [q_i] \subseteq \mathrm{Pre}(M)}} a_{q_1} \cdot \ldots \cdot a_{q_k} \cdot d\big(\delta(c[q]), \delta(c[q'])\big) \Big) + I$$

Kernel State Errors

Lemma

For every $s = \sigma(p_1, \dots, p_k) \in \Sigma(\mathrm{Pre}(N))$ *with* $\mu(s) \in \mathrm{Ker}(N)$

$$|E_s| = \sum_{\substack{q_1 \in [\delta(u_{p_1})] \\ \cdots \\ q_k \in [\delta(u_{p_k})]}} |L(M)^{q_1}| \cdot \ldots \cdot |L(M)^{q_k}| \cdot d\big(\delta(\sigma(q_1, \dots, q_k)),\, q\big)$$

where $u_p \in L(N)^p$ *for every* $p \in \mathrm{Pre}(N)$ *and* $q \equiv \mu(s)$

Main Result

$m = |M|$ $n = |Q|$

Theorem

- *Given hyper-minimal DTA N, almost equivalent to M, we can determine $|L(M) \ominus L(N)|$ in time $\mathcal{O}(mn)$.*
- *We can compute a hyper-optimal DTA in time $\mathcal{O}(mn)$.*

Main Result

$m = |M|$ $n = |Q|$

Theorem

- *Given hyper-minimal DTA N, almost equivalent to M, we can determine $|L(M) \ominus L(N)|$ in time $\mathcal{O}(mn)$.*
- *We can compute a hyper-optimal DTA in time $\mathcal{O}(mn)$.*

Conclusion

Solved problems

- Structural characterization of almost equivalent hyper-minimal DTA
- Hyper-optimization algorithm $\mathcal{O}(mn)$

Open problems

- Can hashes be avoided in hyper-minimization?
- Sub-quadratic error optimization?

Conclusion

Solved problems

- Structural characterization
  of almost equivalent hyper-minimal DTA
- Hyper-optimization algorithm $\mathcal{O}(mn)$

Open problems

- Can hashes be avoided in hyper-minimization?
- Sub-quadratic error optimization?

References

- BADR, GEFFERT, SHIPMAN: *Hyper-minimizing minimized deterministic finite state automata*. ITA 43, 2009
- BADR: *Hyper-minimization in $O(n^2)$*. IJFCS 20, 2009
- GAWRYCHOWSKI, JEŻ: *Hyper-minimisation made efficient*. MFCS 2009
- GAWRYCHOWSKI, JEŻ, MALETTI: *On minimising automata with errors*. MFCS 2011
- GÉCSEG, STEINBY: *Tree automata*. Akadémiai Kiadó, 1984
- HÖGBERG, MAY, MALETTI: *Backward and forward bisimulation minimization of tree automata*. TCS 410, 2009
- HOLZER, JAKOBI *From qquivalence to almost-equivalence, and beyond — minimizing automata with errors*. DLT 2012
- HOLZER, MALETTI: *An $n \log n$ algorithm for hyper-minimizing states in a (minimized) deterministic automaton*. TCS 411, 2010
- MALETTI, QUERNHEIM: *Optimal hyper-minimization*. IJFCS 22, 2011
- SCHEWE: *Beyond hyper-minimisation — minimising DBAs and DPAs is NP-complete*. FSTTCS 2010