TECHNISCHE UNIVERSITÄT DRESDEN

Learning Deterministically Recognizable Tree Series

Andreas Maletti

Dresden, 2007/5/4

00 Motivation

Goal

- Given $\psi\colon T_\Sigma \to A$ with $(A, +, \cdot, 0, 1)$ semifield
- Learn finite representation (here: deterministic wta) of ψ, if possible
- Access to ψ is granted by a certain form of teacher (oracle)

00 Table of Contents

01 Notation

Trees

- T_Σ: trees over ranked alphabet Σ
- C_Σ: contexts (trees with exactly one occurrence of $\Box$) over Σ
- $\mathrm{size}(t)$: number of nodes of a tree t

Tree series

- tree series: mapping of type $T_\Sigma \to A$
- we write (ψ, t) for $\psi(t)$ with $\psi \colon T_\Sigma \to A$
- $A\langle\langle T_\Sigma \rangle\rangle$: set of all mappings of type $T_\Sigma \to A$

02 Syntax

Definition (Borchardt and Vogler ’03)

$(Q, \Sigma, \mathcal{A}, \mu, F)$ is a weighted tree automaton (wta)

- Q is a finite nonempty set (states)
- Σ is a ranked alphabet (of input symbols)
- $\mathcal{A} = (A, +, \cdot, 0, 1)$ is a semifield (of weights)
- $\mu = (\mu_k)_{k \geq 0}$ with $\mu_k \colon \Sigma^{(k)} \to A^{Q^k \times Q}$ (called tree representation)
- $F \subseteq Q$ (final states)

Definition

wta $(Q, \Sigma, \mathcal{A}, \mu, F)$ is deterministic if for every $\sigma \in \Sigma^{(k)}$ and $w \in Q^k$ there exists at most one $q \in Q$ such that $\mu_k(\sigma)_{w,q} \neq 0$.

02 Example wta

$Q = \{\mathsf{S}, \mathsf{VP}, \mathsf{NP}, \mathsf{NN}, \mathsf{ADJ}, \mathsf{VB}\}$ and $F = \{\mathsf{S}\}$

Alice $\xrightarrow{0.5}$ NN Bob $\xrightarrow{0.5}$ NN loves $\xrightarrow{0.5}$ VB hates $\xrightarrow{0.5}$ VB

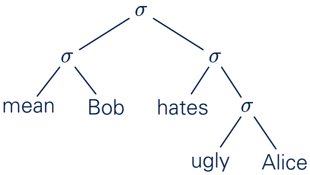
ugly $\xrightarrow{0.25}$ ADJ nice $\xrightarrow{0.25}$ ADJ mean $\xrightarrow{0.25}$ ADJ tall $\xrightarrow{0.25}$ ADJ

$\sigma(\mathsf{NN}, \mathsf{VP}) \xrightarrow{0.5} \mathsf{S}$ $\sigma(\mathsf{NP}, \mathsf{VP}) \xrightarrow{0.5} \mathsf{S}$ $\sigma(\mathsf{VB}, \mathsf{NN}) \xrightarrow{0.5} \mathsf{VP}$

$\sigma(\mathsf{VB}, \mathsf{NP}) \xrightarrow{0.5} \mathsf{VP}$ $\sigma(\mathsf{ADJ}, \mathsf{NN}) \xrightarrow{0.5} \mathsf{NP}$ $\sigma(\mathsf{ADJ}, \mathsf{NP}) \xrightarrow{0.5} \mathsf{NP}$

02 Computation using wta

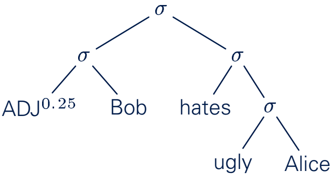
Example



$$\text{mean} \xrightarrow{0.25} \text{ADJ}$$

02 Computation using wta

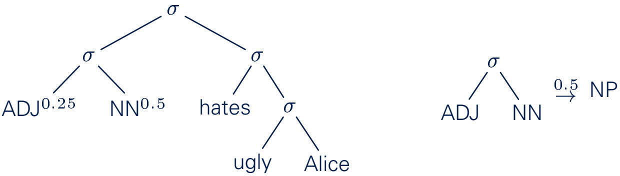
Example



Bob $\xrightarrow{0.5}$ NN

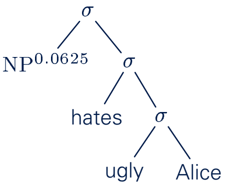
02 Computation using wta

Example



02 Computation using wta

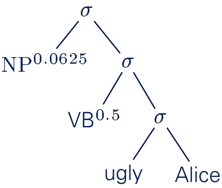
Example



hates $\stackrel{0.5}{\rightarrow}$ VB

02 Computation using wta

Example

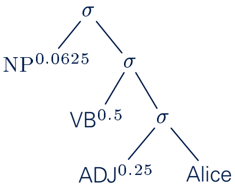


ugly $\xrightarrow{0.25}$ ADJ

TECHNISCHE UNIVERSITÄT DRESDEN

02 Computation using wta

Example

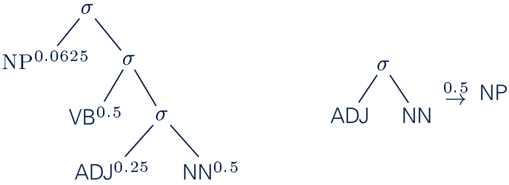


$$\text{Alice} \xrightarrow{0.5} \text{NN}$$

TECHNISCHE UNIVERSITÄT DRESDEN

02 Computation using wta

Example



02 Computation using wta

Example

02 Computation using wta

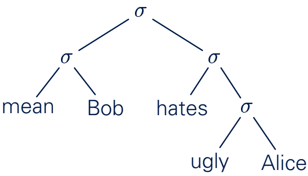
Example

02 Computation using wta

Example

$$S^{0.000244140625}$$

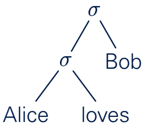
So the tree



is accepted with weight 0.000244140625.

02 Computation using wta (cont'd)

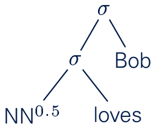
Example



Alice $\xrightarrow{0.5}$ NN

02 Computation using wta (cont'd)

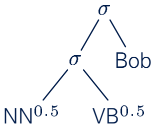
Example



$$\text{loves} \xrightarrow{0.5} \text{VB}$$

02 Computation using wta (cont'd)

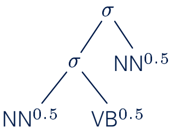
Example



$\text{Bob} \xrightarrow{0.5} \text{NN}$

02 Computation using wta (cont'd)

Example

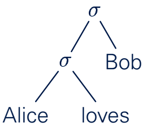


?

02 Computation using wta (cont'd)

Example

So the tree



is rejected (accepted with weight 0).

02 Deterministically recognizable

Definition

A tree series $\psi \in A\langle\langle T_\Sigma \rangle\rangle$ is deterministically recognizable if there exists a deterministic wta accepting ψ.

03 Definition

In the sequel, let $\psi \in A\langle\langle T_\Sigma \rangle\rangle$.

Definition (Borchardt '03)

Two trees $t, u \in T_\Sigma$ are equivalent if there exists $a \in A \setminus \{0\}$ such that for every context $c \in C_\Sigma$

$$a \cdot (\psi, c[t]) = (\psi, c[u]) \ .$$

This equivalence relation is denoted by $\equiv$.

03 Definition

In the sequel, let $\psi \in A\langle\langle T_\Sigma \rangle\rangle$.

Definition (Borchardt '03)

Two trees $t, u \in T_\Sigma$ are equivalent if there exists $a \in A \setminus \{0\}$ such that for every context $c \in C_\Sigma$

$$a \cdot (\psi, c[t]) = (\psi, c[u]) \enspace .$$

This equivalence relation is denoted by $\equiv$.

Example

Let $\psi = \mathrm{size}$ and $\mathcal{A} = (\mathbb{Z} \cup \{\infty\}, \min, +, \infty, 0)$. Then $t \equiv u$ for every $t, u \in T_\Sigma$ because with $a = \mathrm{size}(u) - \mathrm{size}(t)$

$$a + \mathrm{size}(c[t]) = a + \mathrm{size}(c) - 1 + \mathrm{size}(t) = \mathrm{size}(u) + \mathrm{size}(c) - 1 = \mathrm{size}(c[u])$$

03 Myhill-Nerode theorem

Lemma (Borchardt ’03)

$\equiv$ *is a congruence on* (T_Σ, Σ).

03 Myhill-Nerode theorem

Lemma (Borchardt '03)

$\equiv$ is a congruence on (T_Σ, Σ).

Theorem (Borchardt '03)

The following are equivalent:

- *ψ is deterministically recognizable.*
- *$\equiv$ has finite index.*

03 Approximating the Myhill-Nerode relation

Definition

Let $C \subseteq C_\Sigma$. Two trees $t, u \in T_\Sigma$ are C-equivalent if there exists $a \in A \setminus \{0\}$ such that for every context $c \in C$

$$a \cdot (\psi, c[t]) = (\psi, c[u]) \ .$$

The C-equivalence relation is denoted by $\equiv_C$.

03 Approximating the Myhill-Nerode relation

Definition

Let $C \subseteq C_\Sigma$. Two trees $t, u \in T_\Sigma$ are C-equivalent if there exists $a \in A \setminus \{0\}$ such that for every context $c \in C$

$$a \cdot (\psi, c[t]) = (\psi, c[u]) \ .$$

The C-equivalence relation is denoted by $\equiv_C$.

Lemma

- *$\equiv$ and $\equiv_{C_\Sigma}$ coincide.*
- *If $\equiv$ has finite index, then there exists finite $C \subseteq C_\Sigma$ such that $\equiv$ and $\equiv_C$ coincide.*

04 Supervised learning

Goal

Learn a small $C \subseteq C_\Sigma$ such that $\equiv$ and $\equiv_C$ coincide.

Definition (Drewes and Vogler '07)

A maximally adequate teacher answers two types of queries:

- Coefficient query: Given $t \in T_\Sigma$ the teacher supplies (ψ, t).
- Equivalence query: Given wta M the teacher supplies either
  - $\perp$ if $S(M) = \psi$; or
  - some $t \in T_\Sigma$ such that $(S(M), t) \neq (\psi, t)$.

04 Main data structure

Definition

(E, T, C) is an observation table if

- E and T are finite subsets of T_Σ; C is a finite subset of C_Σ
- $E \subseteq T \subseteq \Sigma(E) = \{\sigma(t_1, \ldots, t_k) \mid \sigma \in \Sigma^{(k)}, t_1, \ldots, t_k \in E\}$
- $\square \in C$
- $T \cap L_C = \emptyset$ where $L_C = \{t \in T_\Sigma \mid \forall c \in C\colon (\psi, c[t]) = 0\}$
- $\mathrm{card}(E) = \mathrm{card}(E/\equiv_C)$

04 Main data structure

Definition

(E, T, C) is an observation table if

- E and T are finite subsets of T_Σ; C is a finite subset of C_Σ
- $E \subseteq T \subseteq \Sigma(E) = \{\sigma(t_1, \ldots, t_k) \mid \sigma \in \Sigma^{(k)}, t_1, \ldots, t_k \in E\}$
- $\Box \in C$
- $T \cap L_C = \emptyset$ where $L_C = \{t \in T_\Sigma \mid \forall c \in C \colon (\psi, c[t]) = 0\}$
- $\mathrm{card}(E) = \mathrm{card}(E/_{\equiv_C})$

$T \cap L_C = \emptyset$ means: “No dead states”

04 Main data structure

Definition

(E, T, C) is an observation table if

- E and T are finite subsets of T_Σ; C is a finite subset of C_Σ
- $E \subseteq T \subseteq \Sigma(E) = \{\sigma(t_1, \ldots, t_k) \mid \sigma \in \Sigma^{(k)}, t_1, \ldots, t_k \in E\}$
- $\square \in C$
- $T \cap L_C = \emptyset$ where $L_C = \{t \in T_\Sigma \mid \forall c \in C\colon (\psi, c[t]) = 0\}$
- $\mathrm{card}(E) = \mathrm{card}(E/_{\equiv_C})$

$\mathrm{card}(E) = \mathrm{card}(E/_{\equiv_C})$ means: “No equivalent states”

04 Main data structure

Definition

(E, T, C) is an observation table if

- E and T are finite subsets of T_Σ; C is a finite subset of C_Σ
- $E \subseteq T \subseteq \Sigma(E) = \{\sigma(t_1, \ldots, t_k) \mid \sigma \in \Sigma^{(k)}, t_1, \ldots, t_k \in E\}$
- $\square \in C$
- $T \cap L_C = \emptyset$ where $L_C = \{t \in T_\Sigma \mid \forall c \in C\colon (\psi, c[t]) = 0\}$
- $\mathrm{card}(E) = \mathrm{card}(E/_{\equiv_C})$

Definition

observation table (E, T, C) complete if $\mathrm{card}(E) = \mathrm{card}(T/_{\equiv_C})$.

04 Completing an observation table

Algorithm: COMPLETE

Require: an observation table (E, T, C)

Ensure: return a complete observation table (E', T, C) such that $E \subseteq E'$

for all $t \in T$ **do**
    if $t \not\equiv_C e$ for every $e \in E$ **then**
        $E \leftarrow E \cup \{t\}$
return (E, T, C)

04 The outer structure

Algorithm: Main

$\mathcal{T} \leftarrow (\emptyset, \emptyset, \{\square\})$ {initial observation table}
loop
    $M \leftarrow \mathcal{M}(\mathcal{T})$ {construct new wta}
    $t \leftarrow \textsc{Equal?}(M)$ {ask equivalence query}
    if $t = \bot$ **then**
        return M {return the approved wta}
    else
        $\mathcal{T} \leftarrow \textsc{Extend}(\mathcal{T}, t)$ {extend the observation table}

04 The workhorse: EXTEND

Algorithm: EXTEND

Require: a complete observation table $\mathcal{T} = (E, T, C)$ and a counterexample $t \in T_\Sigma$

Ensure: return a complete observation table $\mathcal{T}' = (E', T', C')$ such that $E \subseteq E'$ and $T \subseteq T'$ and one inclusion is strict

Decompose t into $t = c[u]$ where $c \in C_\Sigma$ and $u \in \Sigma(E) \setminus E$
if $u \in T$ **then**
  if $u \equiv_{C \cup \{c\}} \mathcal{T}(u)$ **then**
    return EXTEND$(\mathcal{T}, c[\mathcal{T}(u)])$ {normalize and continue}
  else
    return COMPLETE$(E \cup \{u\}, T, C \cup \{c\})$ {c separates $\mathcal{T}(u)$ and u}
else
  return COMPLETE$(E, T \cup \{u\}, C \cup \{c\})$ {u not yet reachable}

05 A full example

Let us try to learn the series recognized by the example wta.

05 A full example

$\mathcal{T} \leftarrow (\emptyset, \emptyset, \{\square\})$ $\quad \{\mathcal{T} = (\emptyset, \emptyset, \{\square\})\}$

loop

$\quad M \leftarrow \mathcal{M}(\mathcal{T})$

$\quad t \leftarrow \textsc{Equal?}(M)$

$\quad$ **if** $t = \bot$ **then**

$\quad\quad$ **return** M

$\quad$ **else**

$\quad\quad \mathcal{T} \leftarrow \textsc{Extend}(\mathcal{T}, t)$

05 A full example

$\mathcal{T} \leftarrow (\emptyset, \emptyset, \{\square\})$ $\qquad \{\mathcal{T} = (\emptyset, \emptyset, \{\square\})\}$

loop

$\quad M \leftarrow \mathcal{M}(\mathcal{T})$ $\qquad \{M = \ldots\}$

$\quad t \leftarrow \text{EQUAL?}(M)$

$\quad$ **if** $t = \bot$ **then**

$\qquad$ **return** M

$\quad$ **else**

$\qquad \mathcal{T} \leftarrow \text{EXTEND}(\mathcal{T}, t)$

05 A full example

```
T ← (∅, ∅, {□})
loop
    M ← 𝓜(T)
    t ← EQUAL?(M)
    if t = ⊥ then
        return M
    else
        T ← EXTEND(T, t)
```

$\{\mathcal{T} = (\emptyset, \emptyset, \{\Box\})\}$

$\{M = \dots\}$

$\{t = \sigma(A, \sigma(l, B))\}$

05 A full example

$\mathcal{T} \leftarrow (\emptyset, \emptyset, \{\Box\})$ $\{\mathcal{T} = (\emptyset, \emptyset, \{\Box\})\}$
loop
  $M \leftarrow \mathcal{M}(\mathcal{T})$ $\{M = \ldots\}$
  $t \leftarrow \textsc{Equal?}(M)$ $\{t = \sigma(A, \sigma(l, B))\}$
  if $t = \bot$ **then**
    return M
  else
    $\mathcal{T} \leftarrow \textsc{Extend}(\mathcal{T}, t)$

05 A full example

Recall that $\mathcal{T} = (\emptyset, \emptyset, \{\Box\}) = (E, T, C)$ and $t = \sigma(A, \sigma(l, B))$

Decompose t into $t = c[u]$ with $u \in \Sigma(E) \setminus E$ $\{c = \sigma(\Box, \sigma(l, B)); u = A\}$

if $u \in T$ **then**
    if $u \equiv_{C \cup \{c\}} \mathcal{T}(u)$ **then**
        return EXTEND$(\mathcal{T}, c[\mathcal{T}(u)])$
    else
        return COMPLETE$(E \cup \{u\}, T, C \cup \{c\})$
else
    return COMPLETE$(E, T \cup \{u\}, C \cup \{c\})$

05 A full example

Recall that $\mathcal{T} = (\emptyset, \emptyset, \{\Box\}) = (E, T, C)$ and $t = \sigma(A, \sigma(l, B))$

```
Decompose t into t = c[u] with u ∈ Σ(E) \ E          {c = σ(□, σ(l, B)); u = A}
if u ∈ T then
    if u ≡_{C∪{c}} 𝒯(u) then
        return EXTEND(𝒯, c[𝒯(u)])
    else
        return COMPLETE(E ∪ {u}, T, C ∪ {c})
else
    return COMPLETE(E, T ∪ {u}, C ∪ {c})
```

TECHNISCHE UNIVERSITÄT DRESDEN

05 A full example

Recall that $E = \emptyset$ and $T = \{A\}$ and $C = \{\Box, \sigma(\Box, \sigma(l, B))\}$

for all $t \in T$ **do**
2: **if** $t \not\equiv_C e$ for every $e \in E$ **then**
$E \leftarrow E \cup \{t\}$ $\{E = \{A\}\}$
4: **return** (E, T, C)

05 A full example

Recall that $E = \emptyset$ and $T = \{A\}$ and $C = \{\Box, \sigma(\Box, \sigma(l, B))\}$

for all $t \in T$ **do**
    if $t \not\equiv_C e$ for every $e \in E$ **then**
        $E \leftarrow E \cup \{t\}$ $\{E = \{A\}\}$
return (E, T, C) $\{E = T = \{A\}; C = \{\Box, \sigma(\Box, \sigma(l, B))\}\}$

05 A full example

$\mathcal{T} \leftarrow (\emptyset, \emptyset, \{\Box\})$
loop
    $M \leftarrow \mathcal{M}(\mathcal{T})$
    $t \leftarrow$ EQUAL?(M)
    if $t = \bot$ **then**
        return M
    else
        $\mathcal{T} \leftarrow$ EXTEND$(\mathcal{T}, t)$

$\{\mathcal{T} = (\{A\}, \{A\}, \{\Box, \sigma(\Box, \sigma(l, B))\})\}$

05 A full example

$\mathcal{T} \leftarrow (\emptyset, \emptyset, \{\square\})$
loop
    $M \leftarrow \mathcal{M}(\mathcal{T})$ $\{M = \dots\}$
    $t \leftarrow$ EQUAL?(M)
    if $t = \bot$ **then**
        return M
    else
        $\mathcal{T} \leftarrow$ EXTEND$(\mathcal{T}, t)$ $\{\mathcal{T} = (\{A\}, \{A\}, \{\square, \sigma(\square, \sigma(l, B))\})\}$

05 A full example

$\mathcal{T} \leftarrow (\emptyset, \emptyset, \{\square\})$
loop
  $M \leftarrow \mathcal{M}(\mathcal{T})$ $\{M = \dots\}$
  $t \leftarrow$ EQUAL?(M) $\{t = \sigma(A, \sigma(l, B))\}$
  if $t = \bot$ **then**
    return M
  else
    $\mathcal{T} \leftarrow$ EXTEND$(\mathcal{T}, t)$ $\{\mathcal{T} = (\{A\}, \{A\}, \{\square, \sigma(\square, \sigma(l, B))\})\}$

05 A full example

$\mathcal{T} \leftarrow (\emptyset, \emptyset, \{\Box\})$

loop

  $M \leftarrow \mathcal{M}(\mathcal{T})$ $\{M = \ldots\}$

  $t \leftarrow$ EQUAL?(M) $\{t = \sigma(A, \sigma(l, B))\}$

  if $t = \bot$ **then**

    return M

  else

    $\mathcal{T} \leftarrow$ EXTEND$(\mathcal{T}, t)$ $\{\mathcal{T} = (\{A\}, \{A\}, \{\Box, \sigma(\Box, \sigma(l, B))\})\}$

05 A full example

Recall that $\mathcal{T} = (\{A\}, \{A\}, \{\Box, \dots\}) = (E, T, C)$ and $t = \sigma(A, \sigma(l, B))$

Decompose t into $t = c[u]$ with $u \in \Sigma(E) \setminus E$ $\{c = \sigma(A, \sigma(l, \Box)); u = B\}$

if $u \in T$ **then**
  if $u \equiv_{C \cup \{c\}} \mathcal{T}(u)$ **then**
    return $\textsc{Extend}(\mathcal{T}, c[\mathcal{T}(u)])$
  else
    return $\textsc{Complete}(E \cup \{u\}, T, C \cup \{c\})$
else
  return $\textsc{Complete}(E, T \cup \{u\}, C \cup \{c\})$

05 A full example

Recall that $\mathcal{T} = (\{A\}, \{A\}, \{\Box, \dots\}) = (E, T, C)$ and $t = \sigma(A, \sigma(l, B))$

Decompose t into $t = c[u]$ with $u \in \Sigma(E) \setminus E$ $\{c = \sigma(A, \sigma(l, \Box)); u = B\}$

if $u \in T$ **then**

  if $u \equiv_{C \cup \{c\}} \mathcal{T}(u)$ **then**

    return EXTEND$(\mathcal{T}, c[\mathcal{T}(u)])$

  else

    return COMPLETE$(E \cup \{u\}, T, C \cup \{c\})$

else

  return COMPLETE$(E, T \cup \{u\}, C \cup \{c\})$

05 A full example

Recall that $E = \{A\}$ and $T = \{A, B\}$ and $C = \{\square, \sigma(\square, \sigma(l, B)), \sigma(A, \sigma(l, \square))\}$

for all $t \in T$ **do**
    if $t \not\equiv_C e$ for every $e \in E$ **then**
        $E \leftarrow E \cup \{t\}$
return (E, T, C)

05 A full example

Recall that $E = \{A\}$ and $T = \{A, B\}$ and $C = \{\Box, \sigma(\Box, \sigma(l, B)), \sigma(A, \sigma(l, \Box))\}$

```
for all t ∈ T do
    if t ≢_C e for every e ∈ E then
        E ← E ∪ {t}
return (E, T, C)
```

$\{E = \{A\}; T = \{A, B\}; C = \dots\}$

05 A full example

$\mathcal{T} \leftarrow (\emptyset, \emptyset, \{\Box\})$
loop
  $M \leftarrow \mathcal{M}(\mathcal{T})$
  $t \leftarrow$ EQUAL?(M)
  if $t = \bot$ **then**
    return M
  else
    $\mathcal{T} \leftarrow$ EXTEND$(\mathcal{T}, t)$ $\{\mathcal{T} = (\{A\}, \{A, B\}, \{\Box, \ldots\})\}$

05 A full example

$\mathcal{T} \leftarrow (\emptyset, \emptyset, \{\Box\})$
loop
  $M \leftarrow \mathcal{M}(\mathcal{T})$ $\{M = \ldots\}$
  $t \leftarrow$ EQUAL?(M)
  if $t = \bot$ **then**
    return M
  else
    $\mathcal{T} \leftarrow$ EXTEND$(\mathcal{T}, t)$ $\{\mathcal{T} = (\{A\}, \{A, B\}, \{\Box, \ldots\})\}$

05 A full example

$\mathcal{T} \leftarrow (\emptyset, \emptyset, \{\Box\})$

loop

$M \leftarrow \mathcal{M}(\mathcal{T})$ $\{M = \dots\}$

$t \leftarrow \text{EQUAL?}(M)$ $\{t = \sigma(A, \sigma(l, B))\}$

if $t = \bot$ **then**

return M

else

$\mathcal{T} \leftarrow \text{EXTEND}(\mathcal{T}, t)$ $\{\mathcal{T} = (\{A\}, \{A, B\}, \{\Box, \dots\})\}$

05 A full example

$\mathcal{T} \leftarrow (\emptyset, \emptyset, \{\Box\})$
loop
    $M \leftarrow \mathcal{M}(\mathcal{T})$ $\{M = \ldots\}$
    $t \leftarrow \text{EQUAL?}(M)$ $\{t = \sigma(A, \sigma(l, B))\}$
    if $t = \perp$ **then**
        return M
    else
        $\mathcal{T} \leftarrow \text{EXTEND}(\mathcal{T}, t)$ $\{\mathcal{T} = (\{A\}, \{A, B\}, \{\Box, \ldots\})\}$

05 A full example

Recall that $\mathcal{T} = (\{A\}, \{A, B\}, \{\square, \dots\}) = (E, T, C)$ and $t = \sigma(A, \sigma(l, B))$

```
    Decompose t into t = c[u] with u ∈ Σ(E) \ E        {c = σ(A, σ(l, □)); u = B}
    if u ∈ T then
        if u ≡_{C∪{c}} 𝒯(u) then
            return EXTEND(𝒯, c[𝒯(u)])
        else
            return COMPLETE(E ∪ {u}, T, C ∪ {c})
    else
        return COMPLETE(E, T ∪ {u}, C ∪ {c})
```

05 A full example

Recall that $\mathcal{T} = (\{A\}, \{A, B\}, \{\square, \dots\}) = (E, T, C)$ and $t = \sigma(A, \sigma(l, B))$

```
Decompose t into t = c[u] with u ∈ Σ(E) \ E          {c = σ(A, σ(l, □)); u = B}
if u ∈ T then
    if u ≡_{C∪{c}} 𝒯(u) then
        return EXTEND(𝒯, c[𝒯(u)])                    {EXTEND(𝒯, σ(A, σ(l, A)))}
    else
        return COMPLETE(E ∪ {u}, T, C ∪ {c})
else
    return COMPLETE(E, T ∪ {u}, C ∪ {c})
```

05 Learned wta

$Q = \{\mathsf{NP}, \mathsf{VB}, \mathsf{VP}, \mathsf{S}, \mathsf{ADJ}\}$ and $F = \{\mathsf{S}\}$

$$\text{Alice} \xrightarrow{1} \mathsf{NP} \qquad \text{Bob} \xrightarrow{1} \mathsf{NP} \qquad \text{loves} \xrightarrow{1} \mathsf{VB} \qquad \text{hates} \xrightarrow{1} \mathsf{VB}$$

$$\text{ugly} \xrightarrow{1} \mathsf{ADJ} \qquad \text{nice} \xrightarrow{1} \mathsf{ADJ} \qquad \text{mean} \xrightarrow{1} \mathsf{ADJ} \qquad \text{tall} \xrightarrow{1} \mathsf{ADJ}$$

$$\sigma(\mathsf{NP}, \mathsf{VP}) \xrightarrow{0.03125} \mathsf{S} \qquad \sigma(\mathsf{VB}, \mathsf{NP}) \xrightarrow{1} \mathsf{VP} \qquad \sigma(\mathsf{ADJ}, \mathsf{NP}) \xrightarrow{0.125} \mathsf{NP}$$

TECHNISCHE UNIVERSITÄT DRESDEN

05 Thank you for your attention!

References

- Borchardt, Vogler: Determinization of finite state weighted tree automata. J. Autom. Lang. Combin. 8(3):417–463, 2003
- Borchardt: The Myhill-Nerode theorem for recognizable tree series. Proc. 7th Int. Conf. Developments in Language Theory, LNCS 2710, p. 146–158, Springer 2003
- Drewes, Vogler: Learning deterministically recognizable tree series. J. Autom. Lang. Combin., to appear, 2007