Myhill-Nerode Theorem for Recognizable Tree Series — Revisited

Andreas Maletti

LATIN — 7 April 2008

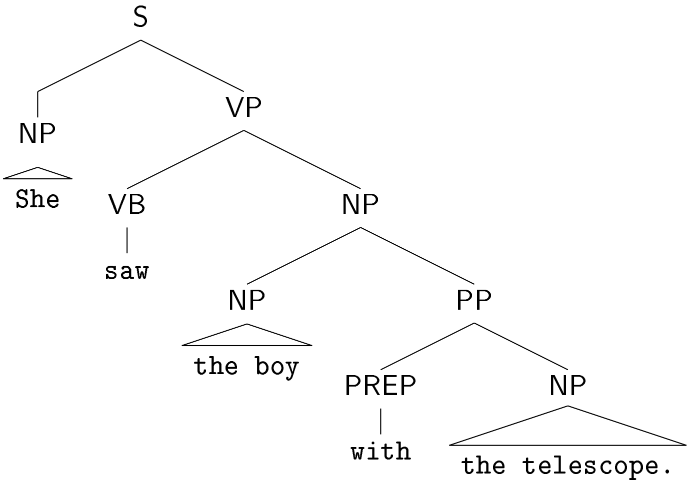
Representation of parses

Input sentence

`She saw the boy with the telescope.`

Representation of parses

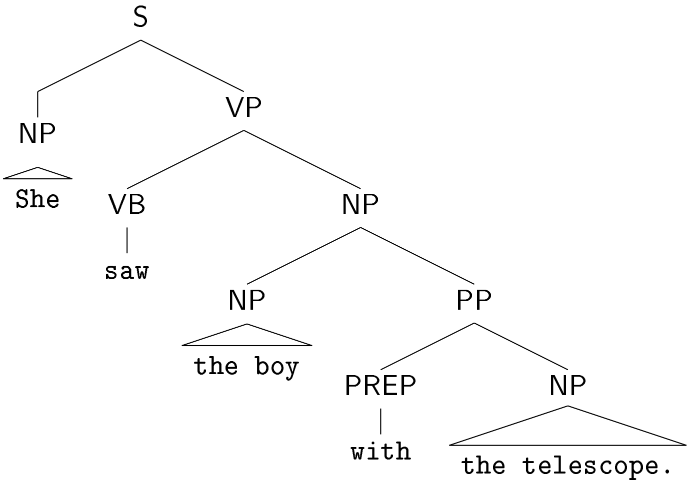
Parses



maybe

Representation of parses

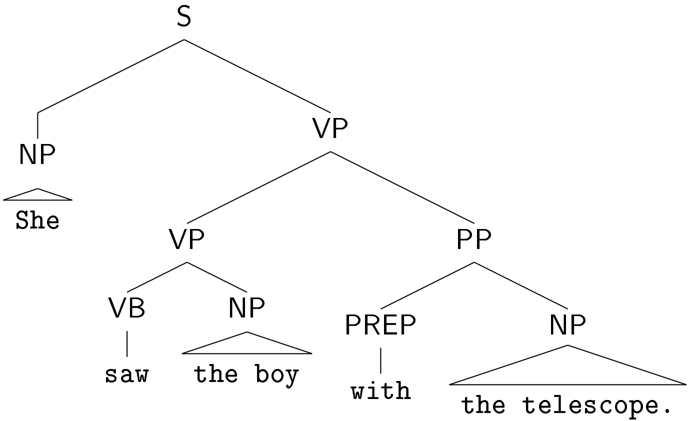
Parses



0.33

Representation of parses

Parses

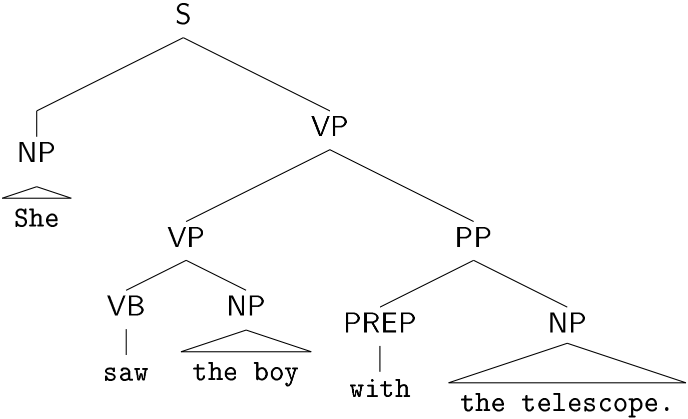


more likely

Representation of parses

Parses

0.66



Tree Series

Prerequisites

- Semiring structure on weights

Tree Series

Prerequisites

- Semiring structure on weights; e.g. $(\mathbb{R}, +, \cdot, 0, 1)$

Tree Series

Prerequisites

- Semiring structure on weights
- Commutative semiring; i.e. $\cdot$ commutative

Tree Series

Prerequisites

- Semiring structure on weights
- Commutative semiring; i.e. $\cdot$ commutative
- A mapping f assigning weights to infinitely many trees

Tree Series

Prerequisites

- Semiring structure on weights
- Commutative semiring; i.e. $\cdot$ commutative
- A mapping f assigning weights to infinitely many trees

Question

How to finitely represent such maps f?

Tree Series

Prerequisites

- Semiring structure on weights
- Commutative semiring; i.e. $\cdot$ commutative
- A mapping f assigning weights to infinitely many trees

Question

How to finitely represent such maps f?

Immediate answer

Non-default value ($\neq 0$) for only finitely many trees

Tree Series

Prerequisites

- Semiring structure on weights
- Commutative semiring; i.e. $\cdot$ commutative
- A mapping f assigning weights to infinitely many trees

Question

How to finitely represent such maps f?

Better answer

Finite-state automaton computes map

Recognizable Tree Series

Determinism

- For efficiency we prefer deterministic devices
- Single run for each input

Recognizable Tree Series

Determinism

- For efficiency we prefer deterministic devices
- Single run for each input

Questions

- Which mappings can be computed in this way?

Recognizable Tree Series

Determinism

- For efficiency we prefer deterministic devices
- Single run for each input

Questions

- Which mappings can be computed in this way?
- Can a given map f be computed in this way?

Recognizable Tree Series

Determinism

- For efficiency we prefer deterministic devices
- Single run for each input

Questions

- Which mappings can be computed in this way?
- Can a given map f be computed in this way?
- How many states are needed to compute a map f?

Recognizable Tree Series

Determinism

- For efficiency we prefer deterministic devices
- Single run for each input

Questions

- Which mappings can be computed in this way?
- Can a given map f be computed in this way?
- How many states are needed to compute a map f?

Answer

The MYHILL-NERODE congruence relation

Table of Contents

Table of Contents

Syntax

Definition (Borchardt and Vogler '03)

Weighted tree automaton: (Q, Σ, A, μ, F)

- Q finite set of states
- Σ ranked alphabet of input symbols
- $A = (A, +, \cdot, 0, 1)$ commutative semiring of weights
- $\mu = (\mu_k)_{k \geq 0}$ with $\mu_k \colon Q^k \times \Sigma^{(k)} \times Q \to A$ transition weights
- $F \colon Q \to A$ final weights

Syntax

Definition (Borchardt and Vogler '03)

Weighted tree automaton: (Q, Σ, A, μ, F)

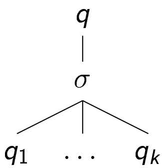
- Q finite set of states
- Σ ranked alphabet of input symbols
- $A = (A, +, \cdot, 0, 1)$ commutative semiring of weights
- $\mu = (\mu_k)_{k \geq 0}$ with $\mu_k \colon Q^k \times \Sigma^{(k)} \times Q \to A$ transition weights
- $F \colon Q \to A$ final weights

Definition

deterministic wta: for every $(w, \sigma) \in Q^k \times \Sigma^{(k)}$ there exists exactly one $q \in Q$ such that $\mu_k(w, \sigma, q) \neq 0$

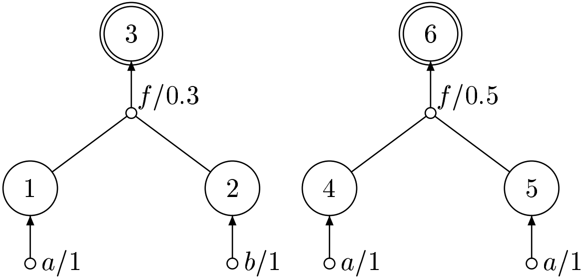
Example — Syntax

Example



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

Example



Semantics

Definition

$h_\mu : \mathrm{Trees}(\Sigma) \to A^Q$

$$h_\mu\left(\begin{array}{c}\sigma\\ \diagup \;\; | \;\; \diagdown\\ t_1 \quad \dots \quad t_k\end{array}\right)_q = \sum_{q_1\cdots q_k\in Q^k} \mu_k\left(\begin{array}{c} q\\ |\\ \sigma\\ \diagup \;\; | \;\; \diagdown \\ q_1 \quad \dots \quad q_k\end{array}\right)\cdot\prod_{i=1}^{k} h_\mu(t_i)_{q_i}$$

Semantics

Definition

$h_\mu : \mathrm{Trees}(\Sigma) \to A^Q$

$$h_\mu\left(\begin{array}{c} \sigma \\ \diagup \quad | \quad \diagdown \\ t_1 \quad \ldots \quad t_k \end{array}\right)_q = \sum_{q_1 \cdots q_k \in Q^k} \mu_k\left(\begin{array}{c} q \\ | \\ \sigma \\ \diagup \quad | \quad \diagdown \\ q_1 \quad \ldots \quad q_k \end{array}\right) \cdot \prod_{i=1}^{k} h_\mu(t_i)_{q_i}$$

Semantics

$$\|M\|(t) = \sum_{q \in Q} F(q) \cdot h_\mu(t)_q$$

Table of Contents

Recognizability

Definition

recognizable f: there exists wta M such that $\|M\| = f$

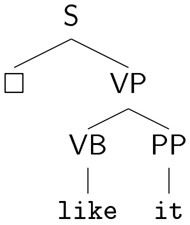
Recognizability

Definition

recognizable f: there exists wta M such that $\|M\| = f$

Notation

- Context: tree with exactly one occurrence of $\square$



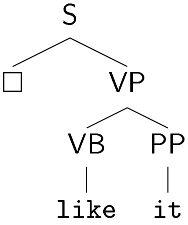
Recognizability

Definition

recognizable f: there exists wta M such that $\|M\| = f$

Notation

- Context: tree with exactly one occurrence of $\square$



Definition

For every $t \in \mathrm{Trees}(\Sigma)$ let $t^{-1}f \colon \mathrm{Contexts}(\Sigma) \to A$ with

$$t^{-1}f(c) = f(c[t])$$

Example — Recognizability

Notation

- size: number of nodes in a tree

Example

Given two trees t and u

$$t^{-1}\,\mathsf{size}(c) = \mathsf{size}(c[t]) = \mathsf{size}(c) - 1 + \mathsf{size}(t)$$
$$u^{-1}\,\mathsf{size}(c) = \mathsf{size}(c[u]) = \mathsf{size}(c) - 1 + \mathsf{size}(u)$$

Example — Recognizability

Notation

- size: number of nodes in a tree

Example

Given two trees t and u

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

Hence $t^{-1}\,\mathrm{size}(c) = u^{-1}\,\mathrm{size}(c) + \mathrm{size}(t) - \mathrm{size}(u)$

Example — Recognizability

Notation

- size: number of nodes in a tree

Example

Given two trees t and u

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

Hence $t^{-1}\,\mathrm{size}(c) = u^{-1}\,\mathrm{size}(c) + \mathrm{size}(t) - \mathrm{size}(u)$

- Suppose that A is a field
- V_f sub-vectorspace generated by $t^{-1}f$ for all t

Example — Recognizability

Notation

- size: number of nodes in a tree

Example

Given two trees t and u

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

Hence $t^{-1}\,\mathrm{size}(c) = u^{-1}\,\mathrm{size}(c) + \mathrm{size}(t) - \mathrm{size}(u)$

- Suppose that A is a field
- V_f sub-vectorspace generated by $t^{-1}f$ for all t
- $t^{-1}\,\mathrm{size}$ and $\vec{1}$ are basis of V_{size} and $\dim V_{\mathrm{size}} = 2$

Recognizability (cont'd)

Theorem (Bozapalidis, Louscou-Bozapalidou '83)

Let A field and $f : \mathrm{Trees}(\Sigma) \to A$

$$f \text{ recognizable} \iff \dim V_f \text{ finite}$$

Notes

- String case by [Reutenauer '80]
- Refined by [Arz '83] to identify requirements for direction
- Led to necessary and/or sufficient conditions of recognizability

Recognizability (cont'd)

Theorem (Bozapalidis, Louscou-Bozapalidou '83)
Let A field and $f : \mathrm{Trees}(\Sigma) \to A$

$$f \text{ recognizable} \iff \dim V_f \text{ finite}$$

Notes

- String case by [Reutenauer '80]
- Refined by [Arz '83] to identify requirements for direction
- Led to necessary and/or sufficient conditions of recognizability
- Tree case: no refinement yet

Deterministic recognizability

Definition

det. recognizable f: there is det. wta M such that $\|M\| = f$

Definition (Myhill-Nerode congruence)

$t \equiv_f u$: there is nonzero $a \in A$ such that $t^{-1}f = a \cdot u^{-1}f$

$$f(c[t]) = a \cdot f(c[u]) \qquad \forall \text{ contexts } c$$

Deterministic recognizability

Definition

det. recognizable f: there is det. wta M such that $\|M\| = f$

Definition (Myhill-Nerode congruence)

$t \equiv_f u$: there is nonzero $a \in A$ such that $t^{-1}f = a \cdot u^{-1}f$

$$f(c[t]) = a \cdot f(c[u]) \qquad \forall \text{ contexts } c$$

Example

$t \equiv_{\mathsf{size}} u$ iff $\mathsf{size}(t) = \mathsf{size}(u)$ because

$$t^{-1}\,\mathsf{size}(c) = \mathsf{size}(c) - 1 + \mathsf{size}(t)$$
$$u^{-1}\,\mathsf{size}(c) = \mathsf{size}(c) - 1 + \mathsf{size}(u)$$

Deterministic recognizability

Definition

det. recognizable f: there is det. wta M such that $\|M\| = f$

Definition (Myhill-Nerode congruence)

$t \equiv_f u$: there is nonzero $a \in A$ such that $t^{-1}f = a \cdot u^{-1}f$

$$f(c[t]) = a \cdot f(c[u]) \qquad \forall \text{ contexts } c$$

Example

$t \equiv_{\mathsf{size}} u$ iff $\mathsf{size}(t) = \mathsf{size}(u)$ because

$$t^{-1}\,\mathsf{size}(c) = \mathsf{size}(c) - 1 + \mathsf{size}(t)$$
$$u^{-1}\,\mathsf{size}(c) = \mathsf{size}(c) - 1 + \mathsf{size}(u)$$

Index of $\equiv_{\mathsf{size}}$ infinite

Deterministic recognizability (cont'd)

Theorem (Borchardt '03)

Let A semifield and $f\colon \mathrm{Trees}(\Sigma) \to A$

$$f \text{ det. recognizable} \iff \equiv_f \text{ finite index}$$

Deterministic recognizability (cont'd)

Theorem (Borchardt '03)

Let A semifield and $f\colon \mathrm{Trees}(\Sigma) \to A$

$$f \text{ det. recognizable} \quad \Longleftrightarrow \quad \equiv_f \text{ finite index}$$

Notes

- So size is not det. recognizable
- Refinements only for smaller classes (all-accepting wta)

Refinement

Definition (Borchardt '05)

$t \equiv_f u$: there exist nonzero $a, b \in A$ such that $a \cdot t^{-1}f = b \cdot u^{-1}f$

$$a \cdot f(c[t]) = b \cdot f(c[u]) \qquad \forall \text{ contexts } c$$

Refinement

Definition (Borchardt '05)

$t \equiv_f u$: there exist nonzero $a, b \in A$ such that $a \cdot t^{-1}f = b \cdot u^{-1}f$

$$a \cdot f(c[t]) = b \cdot f(c[u]) \qquad \forall \text{ contexts } c$$

Definition

Zero-divisor free A: $a \cdot b = 0$ implies $0 \in \{a, b\}$

Refinement

Definition (Borchardt '05)

$t \equiv_f u$: there exist nonzero $a, b \in A$ such that $a \cdot t^{-1}f = b \cdot u^{-1}f$

$$a \cdot f(c[t]) = b \cdot f(c[u]) \qquad \forall \text{ contexts } c$$

Definition

Zero-divisor free A: $a \cdot b = 0$ implies $0 \in \{a, b\}$

Lemma

If A zero-divisor free, then $\equiv_f$ congruence of term algebra $\mathrm{Trees}(\Sigma)$

Refinement (cont'd)

Theorem (Necessary condition)

If A zero-divisor free, then

$$f \text{ det. recognizable} \quad \Longrightarrow \quad \equiv_f \text{ finite index}$$

Refinement (cont'd)

Theorem (Necessary condition)

If A zero-divisor free, then

$$f \text{ det. recognizable} \quad \Longrightarrow \quad \equiv_f \text{ finite index}$$

Theorem

If A zero-divisor free, then every det. wta recognizing f has at least $\mathrm{index}(\equiv_f)$ *states*

Refinement (cont'd)

Theorem (Necessary condition)

If A zero-divisor free, then

$$f \text{ det. recognizable} \quad \Longrightarrow \quad \equiv_f \text{ finite index}$$

Theorem

If A zero-divisor free, then every det. wta recognizing f has at least $\operatorname{index}(\equiv_f)$ *states*

Corollary

height *(longest path) not det. recognizable using addition*

Refinement (cont'd)

Question

What about

$$f \text{ det. recognizable} \quad \Longleftarrow \quad \equiv_f \text{ finite index}$$

Refinement (cont'd)

Question

What about

$$f \text{ det. recognizable} \quad \Longleftarrow \quad \equiv_f \text{ finite index}$$

Notes

- Holds for semifields [Borchardt '03]

Refinement (cont'd)

Question

What about

$$f \text{ det. recognizable} \quad \Longleftarrow \quad \equiv_f \text{ finite index}$$

Notes

- Holds for semifields [Borchardt '03]
- In the string case:
  Refinement for certain cancellative semirings by [Eisner '03]
- In the tree case: Open

All-accepting wta

Definition (Drewes and Vogler '07)

all-accepting wta: $F = \vec{1}$

All-accepting wta

Definition (Drewes and Vogler '07)

all-accepting wta: $F = \vec{1}$

Definition

subtree-closed f: $f(t) \neq 0$ implies $f(u) \neq 0$ for all subtrees u of t

All-accepting wta

Definition (Drewes and Vogler '07)

all-accepting wta: $F = \vec{1}$

Definition

subtree-closed f: $f(t) \neq 0$ implies $f(u) \neq 0$ for all subtrees u of t

Lemma

f det. aa-recognizable iff f det. recognizable and subtree-closed

All-accepting wta

Definition (Drewes and Vogler '07)

all-accepting wta: $F = \vec{1}$

Definition

subtree-closed f: $f(t) \neq 0$ implies $f(u) \neq 0$ for all subtrees u of t

Lemma

f det. aa-recognizable iff f det. recognizable and subtree-closed

Theorem

If A cancellative, then

$$f \text{ det. aa-recognizable} \quad \Longleftrightarrow \quad \equiv_f \text{ finite index and } f \text{ subtree-closed}$$

Notes

- Improves on a similar statement for semifield

References (1/2)

J. Arz.
Syntactic congruences and syntactic algebras.
RAIRO Theoretical Informatics, 17(3):231–238, 1983.

Björn Borchardt.
The Myhill-Nerode theorem for recognizable tree series.
In *Proc. DLT*, volume 2710 of LNCS, pages 146–158. Springer, 2003.

Björn Borchardt and Heiko Vogler.
Determinization of finite state weighted tree automata.
J. Autom. Lang. Combin., 8(3):417–463, 2003.

Symeon Bozapalidis and Olympia Louscou-Bozapalidou.
The rank of a formal tree power series.
Theoret. Comput. Sci., 27:211–215, 1983.

References (2/2)

- Frank Drewes and Heiko Vogler.
  Learning deterministically recognizable tree series.
  J. Autom. Lang. Combin., 2007.
- C. Reutenauer.
  Séries formelles et algébres syntactiques.
  J. Algebra, 66:448–483, 1980.

Thank You!