Slide 1

On Formalizing Syntax

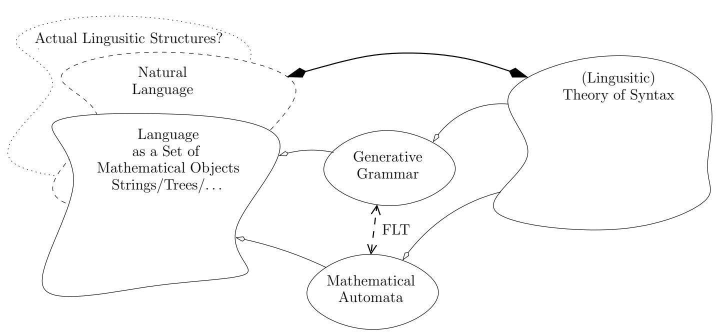
James Rogers

Dept. of Computer Science, Earlham College

`jrogers@cs.earlham.edu`

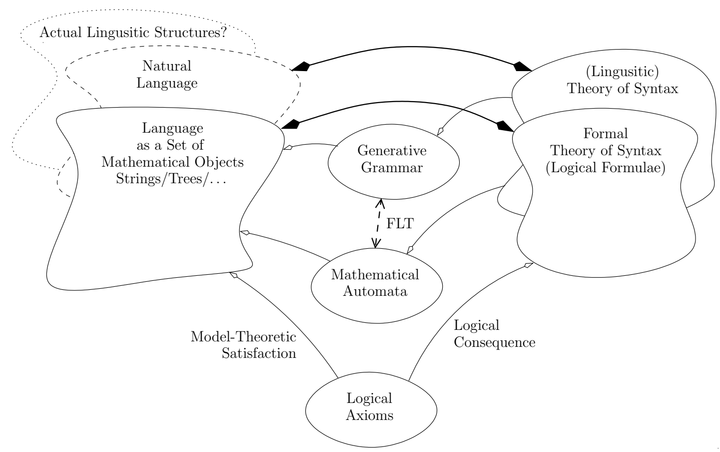
Slide 2

Formalization of Syntax



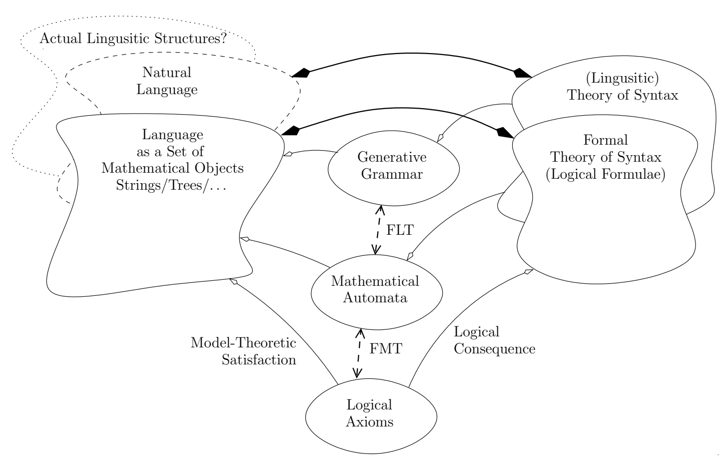
Slide 3

Formalization of Syntax



Slide 4

Formalization of Syntax



Slide 5

Word Models

$(<) \quad \langle \mathcal{D}, \triangleleft, \triangleleft^+, P_\sigma \rangle_{\sigma \in \Sigma} \qquad (+1) \quad \langle \mathcal{D}, \triangleleft, P_\sigma \rangle_{\sigma \in \Sigma}$

$$\begin{array}{rcl} \mathcal{D} & — & \text{Finite} \\ \triangleleft^+ & — & \text{Linear order on } \mathcal{D} \\ \triangleleft & — & \text{Successor wrt } \triangleleft^+ \\ P_\sigma & — & \text{Partition } \mathcal{D} \end{array}$$

$$\begin{array}{rcl} w \in \Sigma^* & \equiv & \langle \mathcal{D}^w, (\triangleleft)^w, (\triangleleft^+)^w, P_\sigma^w \rangle_{\sigma \in \Sigma} \\ \mathcal{D}^w & \stackrel{\text{def}}{=} & \{i \mid 0 \leq i < |w|\} \\ (\triangleleft)^w & \stackrel{\text{def}}{=} & \{\langle i, i+1 \rangle \mid 0 \leq i < |w| - 1\} \\ (\triangleleft^+)^w & \stackrel{\text{def}}{=} & \{\langle i, j \rangle \mid 0 \leq i < j < |w|\} \\ P_\sigma^w & \stackrel{\text{def}}{=} & \{i \mid w = u \cdot \sigma \cdot v,\ |u| = i\} \end{array}$$

$$\mathcal{A} \cdot \mathcal{B} \stackrel{\text{def}}{=} \langle \mathcal{D}^\mathcal{A} \uplus \mathcal{D}^\mathcal{B}, (\triangleleft)^{\mathcal{A} \cdot \mathcal{B}}, (\triangleleft^+)^\mathcal{A} \cup (\triangleleft^+)^\mathcal{B} \cup (\mathcal{D}^\mathcal{A} \times \mathcal{D}^\mathcal{B}), P_\sigma^\mathcal{A} \uplus P_\sigma^\mathcal{B} \rangle$$

Slide 6

k-grams

k-factors

$$F_k(w) \stackrel{\text{def}}{=} \begin{cases} \{w\}, & \text{if } |w| < k \\ \{y \mid w = x \cdot y \cdot z,\ |y| = k\}, & \text{otherwise.} \end{cases}$$

$$F_k(L) \stackrel{\text{def}}{=} \{F_k(w) \mid w \in L\}$$

Strictly k-Local Definitions

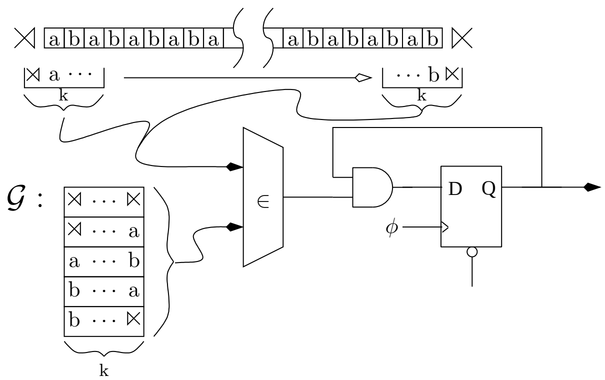
$$\mathcal{G} \subseteq F_k(\{\rtimes\} \cdot \Sigma^* \cdot \{\ltimes\})$$

$$w \models \mathcal{G} \stackrel{\text{def}}{\iff} F_k(\rtimes \cdot w \cdot \ltimes) \subseteq \mathcal{G}$$

$$L(\mathcal{G}) \stackrel{\text{def}}{=} \{w \mid w \models \mathcal{G}\}$$

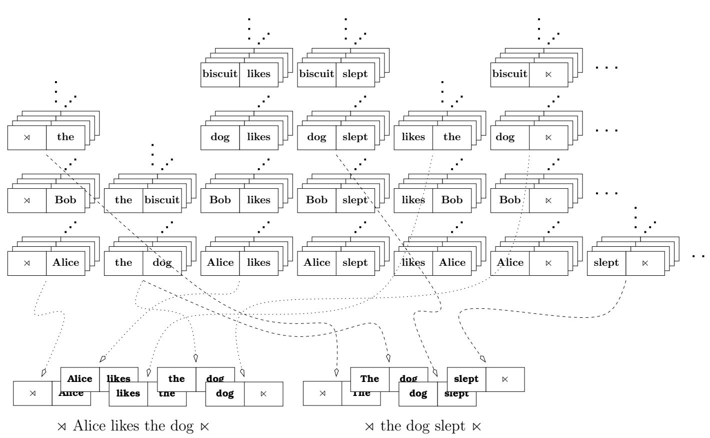
Slide 7

Scanners



Slide 8

Strictly Local Generation



Slide 9

Character of Strictly 2-Local Sets

Theorem (Suffix Substitution Closure):
A stringset L is strictly 2-local iff whenever there is a word x and strings w, y, v, and z, such that

$$\begin{array}{ccccc} w & \cdot & x & \cdot & y \in L \\ v & \cdot & x & \cdot & z \in L \end{array}$$

then it will also be the case that

$$w \cdot x \cdot z \in L$$

Example:

The dog	·	likes	·	the biscuit	$\in L$
Alice	·	likes	·	Bob	$\in L$
The dog	·	likes	·	Bob	$\in L$

Slide 10

Character of (General) Strictly Local Sets

Theorem (General Suffix Substitution Closure):
a stringset l is Strictly Local iff there is some k such that whenever there is a string x of length $k-1$ and strings w, y, v, and z, such that

$$\begin{array}{ccccc} w & \cdot & x & \cdot & y \in L \\ v & \cdot & x & \cdot & z \in L \end{array}$$

then it will also be the case that

$$w \cdot x \cdot z \in L$$

Slide 11

k-Expressions

$$
\begin{array}{ccc}
f \in F_k(\rtimes \cdot \Sigma^* \ltimes) & w \models f & \stackrel{\text{def}}{\iff} & f \in F_k(\rtimes \cdot w \cdot \ltimes) \\
\varphi \wedge \psi & w \models \varphi \wedge \psi & \stackrel{\text{def}}{\iff} & w \models \varphi \text{ and } w \models \psi \\
\neg\varphi & w \models \neg\varphi & \stackrel{\text{def}}{\iff} & w \not\models \varphi
\end{array}
$$

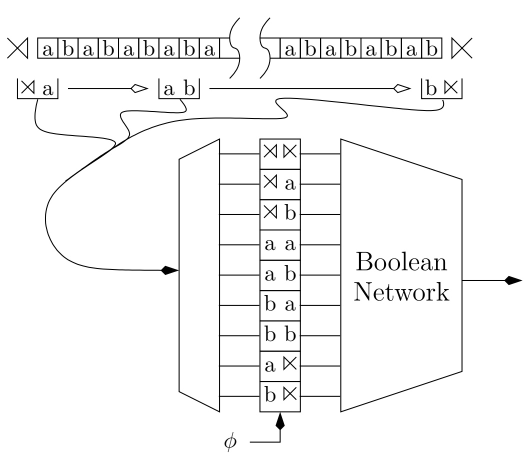
Locally k-Testable Languages (LT_k):

$$L(\varphi) \stackrel{\text{def}}{=} \{w \mid w \models \varphi\}$$

$$\mathrm{SL}_k \equiv \bigwedge_{f_i \notin \mathcal{G}} [\neg f_i] \subsetneq \mathrm{LT}_k$$

Slide 12

LT Automata



Slide 13

Character of Locally Testable Sets

Locally Testable Sets
A stringset L over Σ is *Locally Testable* iff (by definition) there is some k-expression φ over Σ (for some k) such that L is the set of all strings that satisfy φ.

$$L_\varphi = \{x \in \Sigma^* \mid x \models \varphi\}$$

Theorem (k-Test Invariance):
A stringset L is Locally Testable iff

there is some k such that, for all strings x and y,

if $\rtimes \cdot x \cdot \ltimes$ and $\rtimes \cdot y \cdot \ltimes$ have exactly the same set of k-factors

then either both x and y are members of L or neither is.

Slide 14

FO(<) (Strings)

$\langle \mathcal{D}, \triangleleft, \triangleleft^+, P_\sigma \rangle_{\sigma \in \Sigma}$

First-order Quantification over positions in the strings

$x \triangleleft y$	$w, [x \mapsto i, y \mapsto j] \models x \triangleleft y$	$\overset{\text{def}}{\iff}$	$j = i + 1$
$x \triangleleft^+ y$	$w, [x \mapsto i, y \mapsto j] \models x \triangleleft^+ y$	$\overset{\text{def}}{\iff}$	$i < j$
$P_\sigma(x)$	$w, [x \mapsto i] \models P_\sigma(x)$	$\overset{\text{def}}{\iff}$	$i \in P_\sigma$
$\varphi \wedge \psi$	$\vdots$		
$\neg\varphi$	$\vdots$		
$(\exists x)[\varphi(x)]$	$w, s \models (\exists x)[\varphi(x)]$	$\overset{\text{def}}{\iff}$	$w, s[x \mapsto i] \models \varphi(x)$ for some $i \in \mathcal{D}$

Slide 15

Locally Testable with Order (LTO_k)

LT_k plus

$$\varphi \bullet \psi \qquad w \models \varphi \bullet \psi \overset{\text{def}}{\iff} w = w_1 \cdot w_2, \quad w_1 \models \varphi \text{ and } w_2 \models \psi.$$

Definition 1 (Star-Free Set) *The class of* Star-Free Sets *(SF) is the smallest class of languages satisfying:*

- $\emptyset \in SF$, $\{\varepsilon\} \in SF$, *and* $\{\sigma\} \in SF$ *for each* $\sigma \in \Sigma$.
- *If* $L_1, L_2 \in SF$ *then:* $L_1 \cdot L_2 \in SF$,
  $L_1 \cup L_2 \in SF$,
  $\overline{L_1} \in SF$.

Theorem 1 (McNauthton and Papert) *A set of strings is k-Locally Testable with Order (LTO_k) iff it is* Star-Free.

Slide 16

FO(<) over Strings and LTO

$$\begin{aligned} w \models ab \quad &\Leftrightarrow \quad w \models (\exists x, y)[x \triangleleft y \wedge P_a(x) \wedge P_b(y)] \\ w \models \varphi \bullet \psi \quad &\Leftrightarrow \quad w \models (\exists x)[\varphi^{<x}(x) \wedge \psi^{\geq x}(x)] \end{aligned}$$

$$\begin{aligned} w \models P_\sigma(\mathsf{max}) \quad &\Leftrightarrow \quad w \models \sigma\ltimes \\ w \models \mathsf{max} \approx \mathsf{max} \quad &\Leftrightarrow \quad w \models f \vee \neg f \\ w \models \mathsf{max} \approx \mathsf{min} \quad &\Leftrightarrow \quad w \models \textstyle\bigvee_{\sigma\in\Sigma}[\rtimes\sigma\ltimes] \\ w \models (\exists x)[\varphi(x)] \quad &\Leftrightarrow \quad w \models (\exists x)[\textstyle\bigvee_{\langle\varphi_i,\psi_i\rangle\in S_\varphi}[\varphi_i^{<x}(x) \wedge \psi_i^{\geq x}(x)]\ ] \\ &S_\varphi \ \text{ finite, } \ \mathrm{qr}(\varphi_i), \mathrm{qr}(\psi_i) < \mathrm{qr}((\exists x)[\varphi(x)]). \end{aligned}$$

Theorem 2 (McNauthton and Papert) *A set of strings is First-order definable over* $\langle \mathcal{D}, \triangleleft, \triangleleft^+, P_\sigma\rangle_{\sigma\in\Sigma}$ *iff it is* Star-Free.

Slide 17

Character of First-Order Definable Sets

Theorem (McNaughton and Papert):
A stringset L is Star-Free iff it is recognized by a finite-state automaton that is *non-counting* (that has an *aperiodic* syntactic monoid), that is, iff:

there exists some $n > 0$ such that

for all strings u, v, w over Σ

if $uv^n w$ occurs in L

then $uv^{n+i}w$, for all $i \geq 1$, occurs in L as well.

E.g. $(n = 2)$

$$\frac{\text{my}\,\overbrace{\text{father's}}\,\overbrace{\text{father's}}\,\text{father resembled my father} \quad \in L}{\text{my}\,\overbrace{\text{father's}}\,\overbrace{\text{father's}}\,\overbrace{(\text{father's})}^{\geq 1}\,\text{father resembled my father} \quad \in L}$$

Slide 18

FO(+1) (Strings)

$\langle \mathcal{D}, \triangleleft, P_\sigma \rangle_{\sigma \in \Sigma}$

First-order Quantification (over positions in the strings)

Theorem 3 (Thomas) *A set of strings is First-order definable over* $\langle \mathcal{D}, \triangleleft^+, P_\sigma \rangle_{\sigma \in \Sigma}$ *iff it is* Locally Threshold Testable.

Definition 2 (Locally Threshold Testable) *A set* L *is* Locally Threshold Testable *(LTT) iff there is some* k *and* t *such that, for all* $w, v \in \Sigma^*$*:*

if for all $f \in F_k(\rtimes \cdot w \cdot \ltimes) \cup F_k(\rtimes \cdot v \cdot \ltimes)$

either $|w|_f = |v|_f$ *or both* $|w|_f \geq t$ *and* $|v|_f \geq t$,

then $w \in L \iff v \in L$.

Slide 19

MSO (Strings)

$\langle \mathcal{D}, \triangleleft, \triangleleft^{+}, P_{\sigma} \rangle_{\sigma \in \Sigma}$

First-order Quantification (positions)

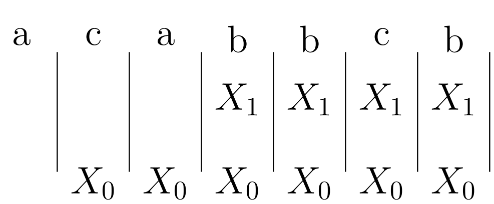
Monadic Second-order Quantification (sets of positions)

$\triangleleft^{+}$ is MSO-definable from $\triangleleft$.

Slide 20

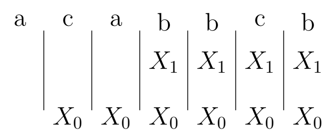
MSO Example

$$
\begin{aligned}
(\exists X_0, X_1)[\quad & (\forall x, y)[(X_0(x) \wedge x \triangleleft y) \to X_0(y)] \wedge \\
& (\forall x)[C(x) \to X_0(x)] \wedge (\exists x)[X_0(x) \wedge B(x)] \quad \wedge \\
& (\forall x, y)[(X_1(x) \wedge x \triangleleft y) \to X_1(y)] \wedge \\
& (\forall x)[B(x) \to X_0(x)] \wedge \neg(\exists x)[A(x) \wedge X_1] \qquad]
\end{aligned}
$$



Slide 21

Automata for MSO

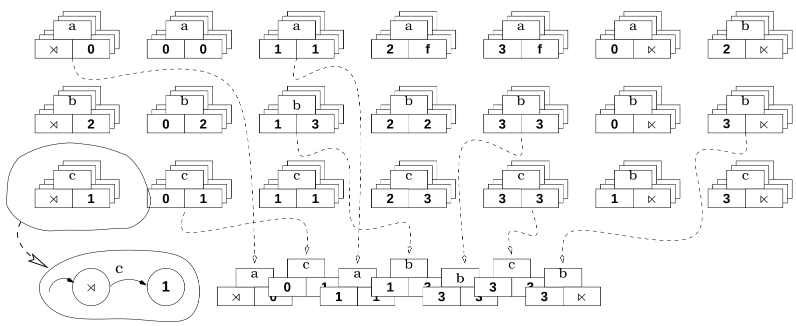


0 – $\emptyset$

1 – $\{X_0\}$

2 – $\{X_1\}$

3 – $\{X_0, X_1\}$



Slide 22

Theorem 4 (Chomsky Shützenberger) *A set of strings is Regular iff it is a homomorphic image of a Strictly 2-Local set.*

Definition (Nerode Equivalence): *Two strings w and v are* Nerode Equivalent *with respect to a stringset L over Σ (denoted $w \equiv_L v$) iff for all strings u over Σ, $wu \in L \Leftrightarrow vu \in L$.*

Theorem 5 (Myhill-Nerode) *: A stringset L is recognizable by a FSA (over strings) iff $\equiv_L$ partitions the set of all strings over Σ into finitely many equivalence classes.*

Theorem 6 (Büchi, Elgot) *A set of strings is MSO-definable over $\langle \mathcal{D}, \triangleleft, \triangleleft^+, P_\sigma \rangle_{\sigma \in \Sigma}$ iff it is regular.*

Theorem 7 *MSO = ∃MSO over strings.*

$$\text{SL} \lneq \text{FO}(+) = \text{LTT} \lneq \text{FO}(<) = \text{SF} \lneq \text{MSO} = \text{Reg. (strings)}$$

Slide 23

Modal Logics—Strings—$\mathcal{L}_{\mathbf{word}}$

$\langle T, \triangleleft, \triangleleft^{+}, P_{\sigma}\rangle_{\sigma\in\Sigma}$ as Frame and Valuation

$$\boldsymbol{\mathcal{L}}_{\text{word}} \quad \varphi : \quad P,\ \top,\ \neg\varphi,\ \varphi\wedge\psi,\ \langle\rightarrow\rangle\varphi,\ \langle\leftarrow\rangle\varphi$$

$$L(\varphi) \stackrel{\text{def}}{=} \{\mathcal{T} \mid \forall(t\in T)[\mathcal{T}, t \models \varphi]\}$$

$$L(\varphi\vee\psi) \neq L(\varphi)\cup L(\psi).$$

$$\mathcal{L}_{\text{word}} = \text{SL} \quad \text{(strings)}$$

Slide 24

Modal Logics—Strings—Adding $\rightarrow^{*}$

$$\boldsymbol{\mathcal{L}}_{\rightarrow^{*}} \quad \varphi : \quad P,\ \top,\ \neg\varphi,\ \varphi\wedge\psi,\ \langle\rightarrow\rangle\varphi,\ \langle\rightarrow^{*}\rangle\varphi$$

$$\mathcal{T}, t \models \langle\rightarrow\rangle\varphi \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad (\exists t')[\langle t, t'\rangle \in \mathcal{T}^{\triangleleft} \text{ and } \mathcal{T}, t' \models \varphi]$$
$$\mathcal{T}, t \models \langle\rightarrow^{*}\rangle\varphi \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad (\exists t')[(t' \approx t \text{ or } \langle t, t'\rangle \in \mathcal{T}^{\triangleleft^{+}}) \text{ and } \mathcal{T}, t' \models \varphi]$$

$$L(\varphi) \stackrel{\text{def}}{=} \{\mathcal{T} \mid \mathcal{T}, \varepsilon \models \varphi\}$$

$$L_{\text{word}} = \text{SL} \ \lneq\ \text{LT} \ \lneq\ \mathcal{L}_{\rightarrow^{*}} \ \lneq\ \text{FO}(<) \ \text{(strings)}$$

Slide 25

Modal Logics—Strings—PTL

$$\boldsymbol{\mathcal{L}}_{\mathbf{until}} \quad \varphi : \quad P,\ \top,\ \neg\varphi,\ \varphi \wedge \psi,\ \langle\rightarrow\rangle\varphi,\ \langle\rightarrow^*\rangle\varphi,\ \mathcal{U}(\varphi,\psi)$$

$$\mathcal{T}, t \models \mathcal{U}(\varphi,\psi) \overset{\text{def}}{\Longleftrightarrow} (\exists t')[\quad t \triangleleft^* t' \text{ and } \mathcal{T}, t' \models \varphi \text{ and } (\forall s)[t \triangleleft^* s \triangleleft^* t' \Rightarrow \mathcal{T}, s \models \psi]]$$

$$\mathcal{L}_{\text{word}} = \text{SL} \lneq \mathcal{L}_{\rightarrow^*} \lneq \mathcal{L}_{\text{until}} = \text{FO}(<) = \text{SF (strings)}$$

Slide 26

Modal Logics—Strings—PDL

$$\boldsymbol{\mathcal{L}}_{\mathbf{pdl}} \quad \varphi : \quad P,\ \top,\ \neg\varphi,\ \varphi \wedge \psi,\ \langle\pi\rangle\varphi$$
$$\pi : \quad \rightarrow,\ ?\varphi,\ \pi_1;\pi_2,\ \pi_1 \cup \pi_2,\ \pi^*$$

$$\mathcal{T}, t \models \langle\pi\rangle\varphi \overset{\text{def}}{\Longleftrightarrow} (\exists t')[\langle t, t'\rangle \in R^{\mathcal{T}}_{\pi} \text{ and } \mathcal{T}, t' \models \varphi]$$

$$R^{\mathcal{T}}_{\rightarrow} \overset{\text{def}}{=} \triangleleft^{\mathcal{T}} \qquad R^{\mathcal{T}}_{?\varphi} \overset{\text{def}}{=} \{\langle t, t\rangle \mid \mathcal{T}, t \models \varphi\}$$
$$R^{\mathcal{T}}_{\pi_1;\pi_2} \overset{\text{def}}{=} R^{\mathcal{T}}_{\pi_1} \circ R^{\mathcal{T}}_{\pi_2} \qquad R^{\mathcal{T}}_{\pi_1\cup\pi_2} \overset{\text{def}}{=} R^{\mathcal{T}}_{\pi_1} \cup R^{\mathcal{T}}_{\pi_2}$$

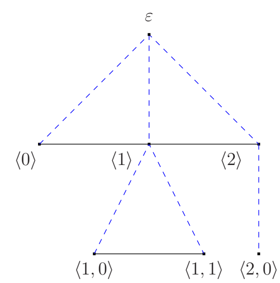
$$L_{\text{word}} = \text{SL} \lneq \mathcal{L}_{\rightarrow^*} \lneq \mathcal{L}_{\text{until}} = \text{FO}(<) \lneq \mathcal{L}_{\text{pdl}} = \text{MSO} = \text{Reg. (strings)}$$

Slide 27

Tree Models

$\langle T, \triangleleft_1, \triangleleft_1^+, \triangleleft_2, \triangleleft_2^+, P_\sigma \rangle_{\sigma \in \Sigma}$

$T \subseteq$	—	Finite Tree domain
$\triangleleft_1$	—	Immediate left-of (global)
$\triangleleft_1^+$	—	Left-of (global)
$\triangleleft_2$	—	Immediate domination
$\triangleleft_2^+$	—	Proper domination
P_σ	—	Partition $\mathcal{D}$

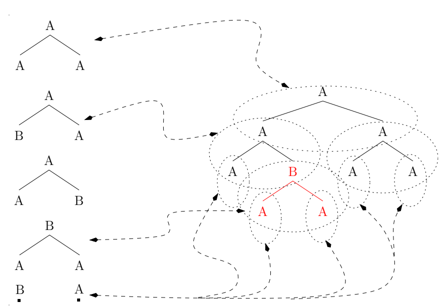


Σ-labeled Tree:

$$\mathcal{T} = \langle T, \tau \rangle, \ \tau : T \to \Sigma = \{x \mapsto \sigma \mid x \in P_\sigma\}$$

Slide 28

Local Tree Grammars



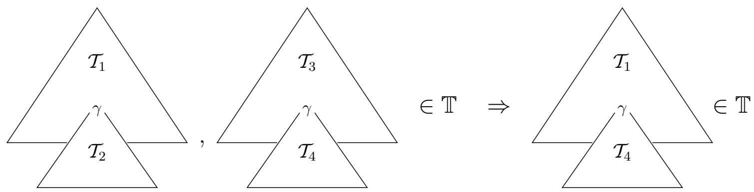
A *Local Tree Grammar* $\mathcal{G}$ over Σ is a finite set of local (height ≤ 1) Σ-labeled trees.

The set of Σ-labeled trees licensed by $\mathcal{G}$ relative to some set of start labels $S \subseteq \Sigma$ is: $\mathcal{G}(S) \stackrel{\text{def}}{=} \{\mathcal{T} \mid \text{LT}(\mathcal{T}) \subseteq \mathcal{G}, \ \tau(\varepsilon) \in S\}$

$$\text{LTG} \leq \text{FO}(\triangleleft_2^+)$$

Slide 29

Subtree Substitution Closure

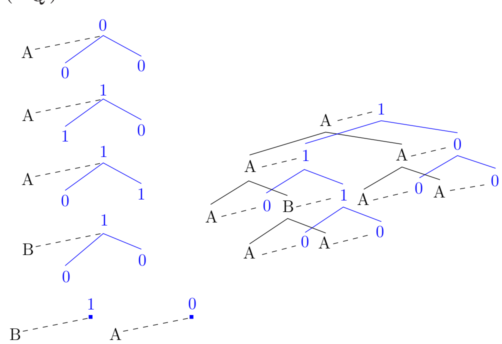


Theorem 8 *A set of labeled trees is Local iff it is closed under substitution of subtrees rooted at similarly labeled points.*

Slide 30

Tree Automata

A *Tree Automaton* over alphabet Σ and state set Q is a finite set $\mathcal{A} \subseteq \Sigma \times \mathrm{LT}(\mathbb{T}_Q)$.

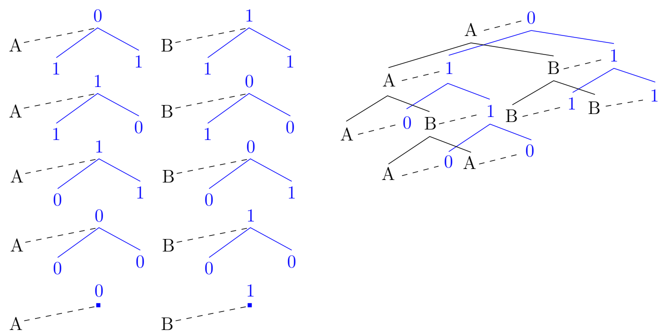


OneB: $\mathcal{A}(\{1\}) = \{\mathcal{T} \in \mathbb{T}_{\{A,B\}} \mid |\mathcal{T}|_B = 1\}$

$$\mathrm{LTG} \lneq \mathrm{FO}(\triangleleft_2^+)$$

Slide 31

Tree Automata



EvenB: $\mathcal{A}(\{0\}) = \{\mathcal{T} \in \mathbb{T}_{\{A,B\}} \mid |\mathcal{T}|_B \equiv 0 \pmod 2\}$

$$\text{LTG} \lneq \text{FO}(\triangleleft_2^+) \lneq \text{Reg (trees)}$$

Slide 32

A Myhill-Nerode Characterization

Theorem 9 *Suppose $\mathbb{T} \subseteq \mathbb{T}_\Sigma$. For all $\mathcal{T}_1, \mathcal{T}_2 \in \mathbb{T}_\Sigma$, let $\mathcal{T}_1 \equiv_\mathbb{T} \mathcal{T}_2$ iff, for every tree $\mathcal{T} \in \mathbb{T}_\Sigma$ and point s in the domain of $\mathcal{T}$, the result of substituting $\mathcal{T}_1$ at s in $\mathcal{T}$ is in $\mathbb{T}$ iff the result of substituting $\mathcal{T}_2$ is:*

$$\mathcal{T} \overset{s}{\leftarrow} \mathcal{T}_1 \in \mathbb{T} \iff \mathcal{T} \overset{s}{\leftarrow} \mathcal{T}_2 \in \mathbb{T}.$$

Then $\mathbb{T}$ is recognizable iff $\equiv_\mathbb{T}$ has finite index.

Slide 33

FO, MSO—Trees

Theorem 10 (Thatcher) *A set of Σ-labeled trees is recognizable iff it is a projection of a local set of trees.*

Theorem 11 (Thatcher and Wright, Doner) *A set of Σ-labeled trees is definable in MSO over trees iff it is recognizable.*

$$\text{LTG} \lneq \text{FO}(\triangleleft_2^+) \lneq \text{MSO}(\triangleleft_2^+) = \text{Reg} \qquad \text{(trees)}$$

Theorem 12 (Thatcher) *A set of strings L is the yield of a local set of trees (equivalently, is the yield of a recognizable set of trees) iff it is Context-Free.*

Corollary 1 *A set of strings L is the yield of a MSO (or FO) definable set of trees iff it is Context-Free.*

Slide 34

Parsing Model-Theoretic Grammars

Parsing string grammars

$$L(\varphi) = \{w \mid w \models \varphi\}$$

Parsing = satisfaction (model checking)

Parsing tree grammars

$$L(\varphi) = \{\text{Yield}(\mathcal{T}) \mid \mathcal{T} \models \varphi\}$$

Let: $\psi_w \stackrel{\text{def}}{=}$ "yield of $\mathcal{T}$ is w".

Then: $\{\mathcal{T} \mid \mathcal{T} \models \psi_w \wedge \varphi\} =$ parse forest for w.

Recognition = satisfiability of $\psi_w \wedge \varphi$

Slide 35

FO—Trees

FO(+1): $\langle T, \triangleleft_1, \triangleleft_1^+, \triangleleft_2, P_\sigma \rangle_{\sigma \in \Sigma}$

Theorem 13 (Benedikt and Segoufin) *A regular set of trees is definable in FO(+1) over trees iff it is Locally Threshold Testable.*

Theorem 14 (Benedikt and Segoufin) *A regular set of trees is definable in FO(+1) over trees iff it is aperiodic.*

FO(mod):

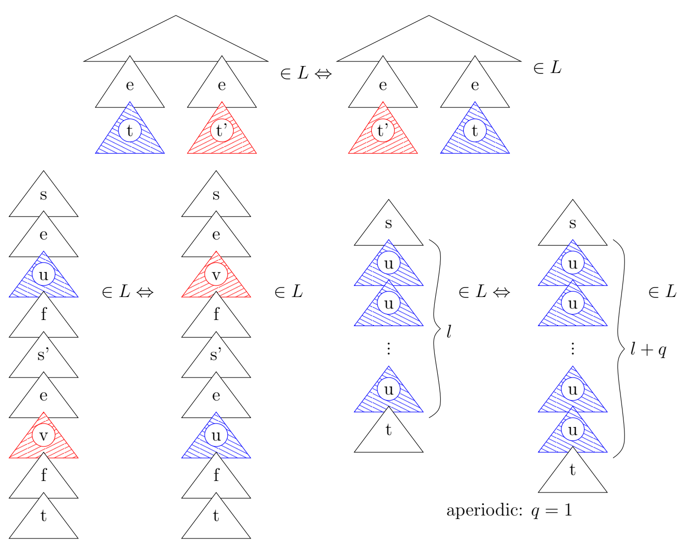
$$\mathcal{T} \models (\exists^{r,q} x)[\varphi(x, \vec{y})] \overset{\text{def}}{\iff}$$
$$\mathbf{card}(\{a \mid \mathcal{T} \models \varphi(x, \vec{y})[x \mapsto a]\}) \equiv r \pmod{q}$$

Theorem 15 (Benedikt and Segoufin) *A regular set of trees is definable in FO(mod) over trees iff it is q-periodic.*

$$\text{LTG} \lneq \text{FO}(+) \lneq \text{FO}(\text{mod}) \lneq \text{FO}(<) \lneq \text{MSO} = \text{Reg.} \quad \text{over trees}$$

Slide 36

Aperiodic/q-periodic Regular Tree Languages



Slide 37

MSO and SF—trees

Theorem 16 (Thatcher and Wright, Doner)
MSO over trees = ∃MSO over trees.

Theorem 17 (Thomas)
MSO = "Anti-chain" MSO over trees without unary branching.
MSO = "Frontier" MSO over trees without unary branching.

Theorem 18 (Thomas)
Every Regular tree language without unary branching is Star-Free.
Regular tree languages without unary branching are of uniformly bounded dot depth.

Without unary branching:

$$\text{LTG} \lneq \text{FO}(+1) \lneq \text{FO}(\text{mod}) \lneq \text{FO}(<) \lneq \text{SF} = \text{MSO} = \text{Reg.}$$

Slide 38

Modal Logics—Trees—$\mathcal{L}_{\textbf{core}}$

$\langle T, \triangleleft_1, \triangleleft_1^+, \triangleleft_2, \triangleleft_2^+, P_\sigma \rangle_{\sigma\in\Sigma}$ as Frame and Valuation

$$\begin{array}{lll} \boldsymbol{\mathcal{L}}_{\textbf{core}} & \varphi : & P,\ \top,\ \neg\varphi,\ \varphi \wedge \psi,\ \langle\pi\rangle \\ & \pi : & \rightarrow,\ \downarrow,\ \leftarrow,\ \uparrow,\ \pi^* \end{array}$$

$$\mathcal{T}, t \models \langle\pi\rangle\varphi \overset{\text{def}}{\Longleftrightarrow} (\exists t')[\langle t, t'\rangle \in R_\pi^{\mathcal{T}} \text{ and } \mathcal{T}, t' \models \varphi]$$

$$\begin{array}{lll@{\qquad}lll} R_{\rightarrow}^{\mathcal{T}} & \overset{\text{def}}{=} & \triangleleft_1{}^{\mathcal{T}}\,|_{\{\langle s\cdot i, s\cdot j\rangle\}} & R_{\downarrow}^{\mathcal{T}} & \overset{\text{def}}{=} & \triangleleft_2{}^{\mathcal{T}} \\ R_{\rightarrow^*}^{\mathcal{T}} & \overset{\text{def}}{=} & \triangleleft_1^{*\mathcal{T}}\,|_{\{\langle s\cdot i, s\cdot j\rangle\}} & R_{\downarrow^*}^{\mathcal{T}} & \overset{\text{def}}{=} & \triangleleft_2^{*\mathcal{T}} \\ R_{\leftarrow}^{\mathcal{T}} & \overset{\text{def}}{=} & (R_{\rightarrow}^{\mathcal{T}})^{-1} & & \text{etc.} & \end{array}$$

$$L(\varphi) \overset{\text{def}}{=} \{\mathcal{T} \mid \mathcal{T}, \varepsilon \models \varphi\}$$

Slide 39

Modal Logics—Trees—$\mathcal{L}_{\text{until}}$, $\mathcal{L}_{\text{pdl}}$ and $\mathcal{L}_{\text{cp}}$

$$\boldsymbol{\mathcal{L}}_{\textbf{until}} \quad \varphi : \quad P,\ \top,\ \neg\varphi,\ \varphi \wedge \psi,$$
$$\mathcal{U}_{\rightarrow}(\varphi,\psi),\ \mathcal{U}_{\leftarrow}(\varphi,\psi),\ \mathcal{U}_{\downarrow}(\varphi,\psi),\ \mathcal{U}_{\uparrow}(\varphi,\psi)$$

$$\mathcal{T}, t \models \mathcal{U}_{\downarrow}(\varphi,\psi) \overset{\text{def}}{\iff} (\exists t')[\quad t \triangleleft_2^* t' \text{ and } \mathcal{T}, t' \models \varphi \text{ and}$$
$$(\forall s)[t \triangleleft_2^* s \triangleleft_2^* t' \Rightarrow \mathcal{T}, s \models \psi]]$$

$$\boldsymbol{\mathcal{L}}_{\textbf{pdl}} \quad \varphi : \quad P,\ \top,\ \neg\varphi,\ \varphi \wedge \psi,\ \langle\pi\rangle\varphi$$
$$\pi : \quad \rightarrow,\ \leftarrow,\ \downarrow,\ \uparrow,\ ?\varphi,\ \pi_1;\pi_2,\ \pi_1 \cup \pi_2,\ \pi^*$$

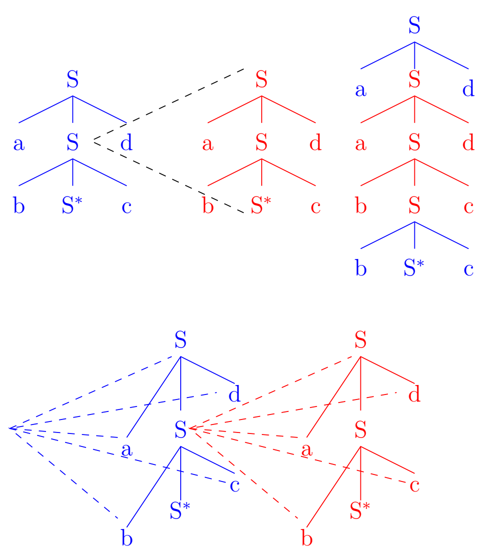
$$R^{\mathcal{T}}_{?\varphi} \overset{\text{def}}{=} \{\langle t,t\rangle \mid \mathcal{T}, t \models \varphi\} \quad R^{\mathcal{T}}_{\pi_1;\pi_2} \overset{\text{def}}{=} R^{\mathcal{T}}_{\pi_1} \circ R^{\mathcal{T}}_{\pi_2} \quad R^{\mathcal{T}}_{\pi_1 \cup \pi_2} \overset{\text{def}}{=} R^{\mathcal{T}}_{\pi_1} \cup R^{\mathcal{T}}_{\pi_2}$$

$$\boldsymbol{\mathcal{L}}_{\textbf{cp}} \quad \varphi : \quad P,\ \top,\ \neg\varphi,\ \varphi \wedge \psi,\ \langle\pi\rangle\varphi$$
$$\pi : \quad \rightarrow,\ \leftarrow,\ \downarrow,\ \uparrow,\ \varphi?;\pi,\ \pi^*$$

LTG $\lneq \mathcal{L}_{\text{core}} \lneq \mathcal{L}_{\text{until}} = \mathcal{L}_{\text{cp}} = \text{FO}(<) \lneq \mathcal{L}_{\text{pdl}} \lneq$ MSO = Reg. (trees)

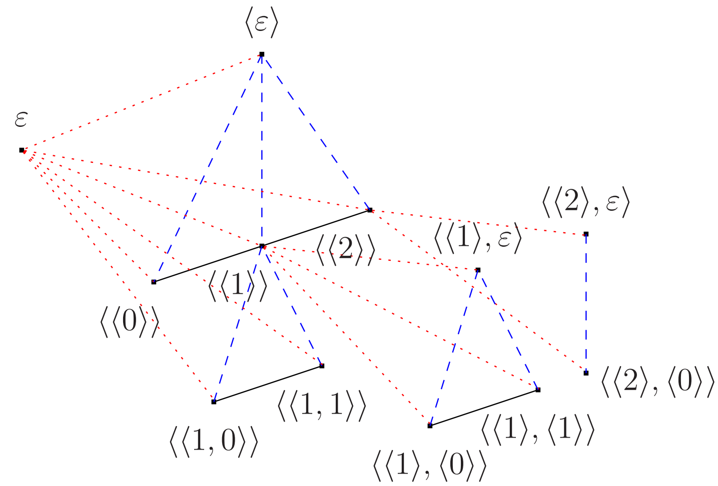
Slide 40

Beyond CFLs



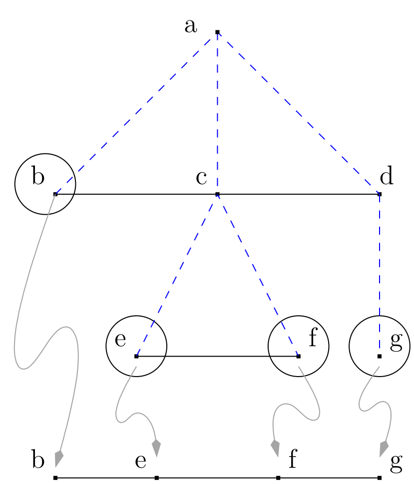
Slide 41

3-Dimensional Domains



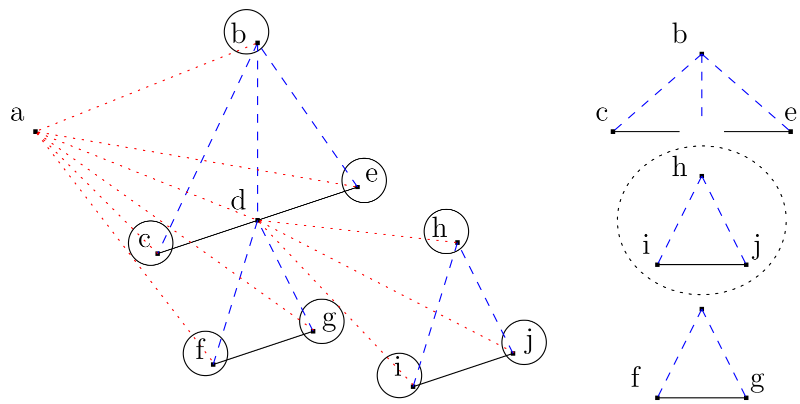
Slide 42

Yields of T2



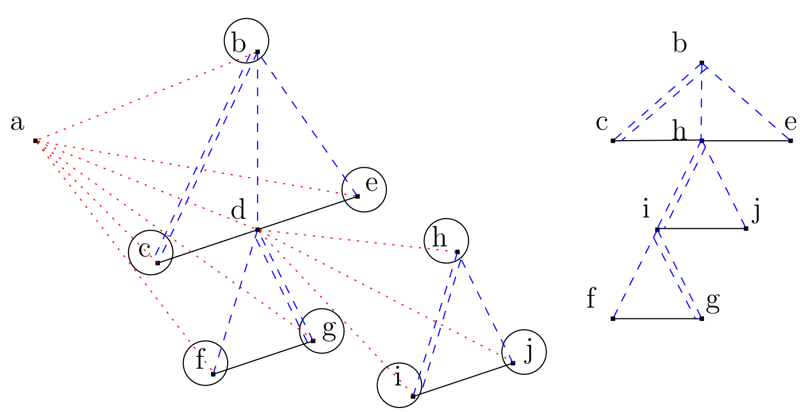
Slide 43

Yields of T3



Slide 44

Headed Structures



Slide 45

Σ-Labeled Headed T3

Definition 3 *A* Σ-Labeled Headed T3 *is a structure:*

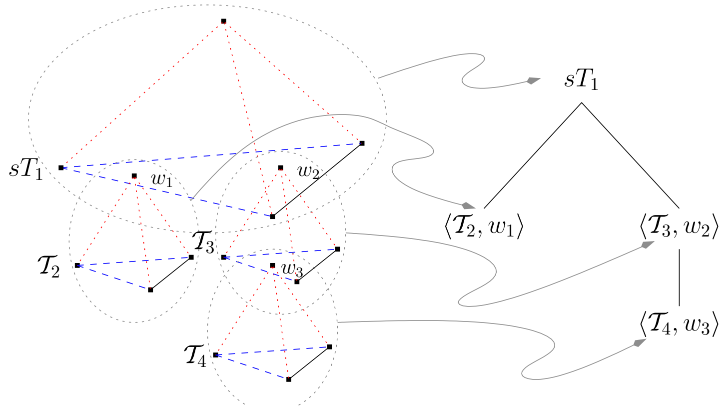
$$\mathcal{T} = \langle T, \triangleleft_i^+, R_i, H_i, P_\sigma \rangle_{1 \leq i \leq 3, \sigma \in \Sigma},$$

- P_σ—points labeled σ.
- R_i—roots of i-dimensional component structures.
- H_i—i-dimensional heads,
  - one on the principle spine of each $(i-1)$-dimensional component.
- $\triangleleft_i^+$—"inherited" proper domination

Theorem 19 *A set of* Σ*-labeled Headed T3 is MSO definable iff it is recognizable.*

Slide 46

Local Sets and Derivation Trees



Slide 47

Non-Strict TAGs and T3-Automata

Theorem 20 *A set of Σ-labeled trees is the yield of a recognizable set of Σ-labeled T3 iff it is generated by a non-strict TAG with adjoining constraints.*

T3 Automata and Non-Strict TAGs with adjoining constraints are, in essence, just notational variants.

Slide 48

Feasibility

- While complexity of translation algorithm is non-elementary, in many actual cases it is practical [Basin and Klarlund'95, Henriksen et al.'95, Morawietz and Cornell'95, '98].
- In many cases it isn't. (viz. indexation) [Morawietz and Cornell'95, '98].
- Restricting to tractable formulae:
  - Limit the total number of free variables
  - Limit the quantifier depth
  - Limit the overall size of formulae.
    - Morawietz: CLP over recognizable sets of trees

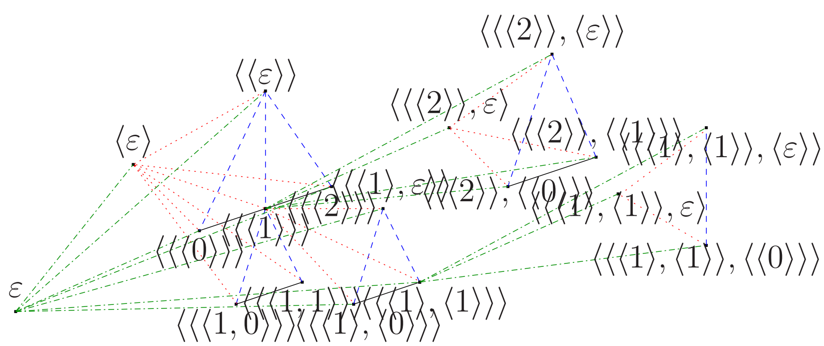
Slide 49

Feasibility and TAG

- TAG is index-free.
- All agreement is local to elementary trees
  - reduces number of variables needed for feature passing.
- Factorization pushes quantifiers inward
  - Conjunction/disjunction of relatively simple formulae.
- Factorizations express constraints on elementary trees
  - filters on local trees of the grammar.

Slide 50

Higher-Dimensional Domains



Slide 51

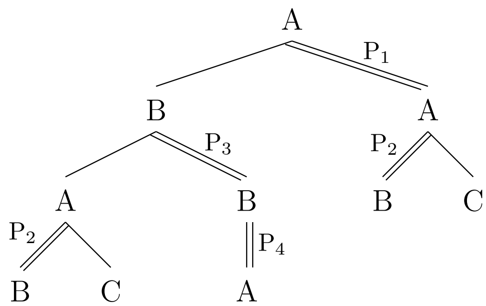
Labeled Distinguished Grammars

$P_1: \quad A \longrightarrow B\underline{A}$

$P_2: \quad A \longrightarrow \underline{B}C$

$P_3: \quad B \longrightarrow A\underline{B}$

$P_4: \quad B \longrightarrow \underline{A}$



$CW(T) = \{P_1P_2, P_3P_4, P_2\}$

Slide 52

The Control Language Hierarchy (Weir'92)

$$L(G, C) \stackrel{\text{def}}{=} \{\text{Yield}(\mathcal{T}) \mid \mathcal{T} \in T(G) \text{ and } \text{CW}(\mathcal{T}) \subseteq C\}$$

$\mathcal{C}_1$: CFL ($= L(G, C)$ for C Regular).

$\mathcal{C}_{i+1}$: $L(G, C)$ for $C \in \mathcal{C}_i$.

Theorem 21 *A string language is* $\textit{Yield}^1_d(\mathbb{T})$ *for some* $\mathbb{T}$*, a recognizable set of Td,* $d \geq 2$*, iff it is in* $\mathcal{C}_{d-1}$.

Slide 53

Higher-Dimensional Grammars

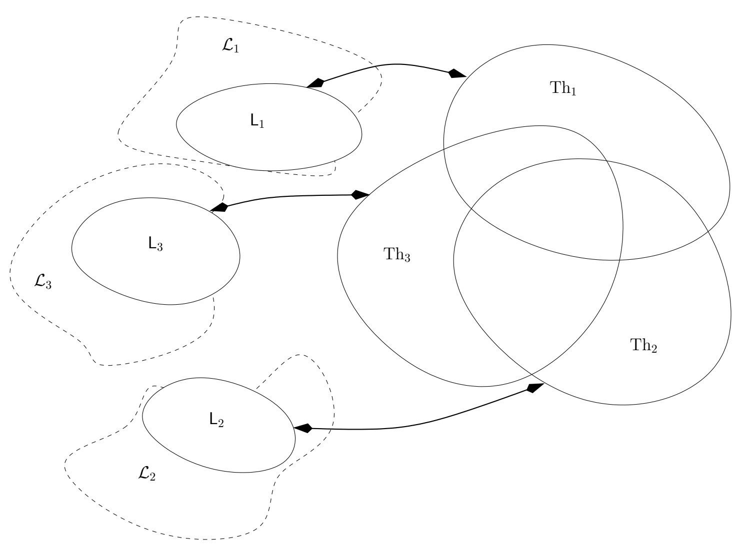
Theorem 22 (Recognizable Sets and the CLH) *A string language is* $Yield_d^1(\mathbb{T})$ *for some* $\mathbb{T}$*, a recognizable set of Td,* $d \geq 2$*, iff it is in* $\mathcal{C}_{d-1}$*.*

Theorem 23 *A set of* Σ*-labeled Headed Td is MSO definable iff it is recognizable.*

Corollary 2 *A string language is* $Yield_d^1(\mathbb{T})$ *for some* $\mathbb{T}$*, a MSO definable set of Td,* $d \geq 2$*, iff it is in* $\mathcal{C}_{d-1}$*.*

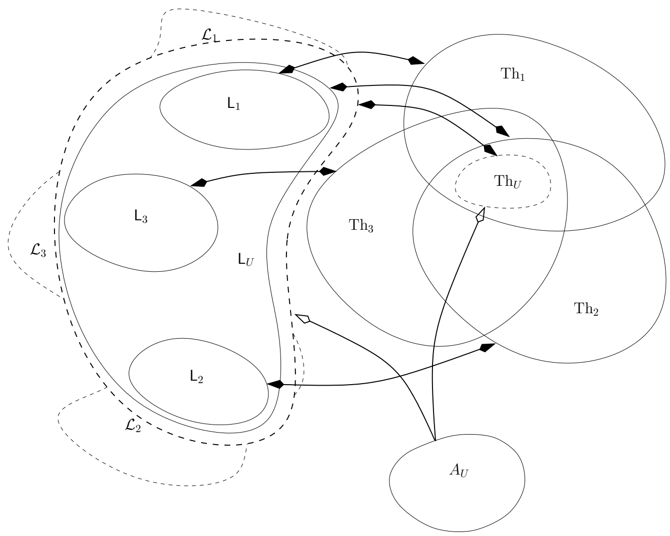
Slide 54

Linguistic Theories v.s. Logical Theories



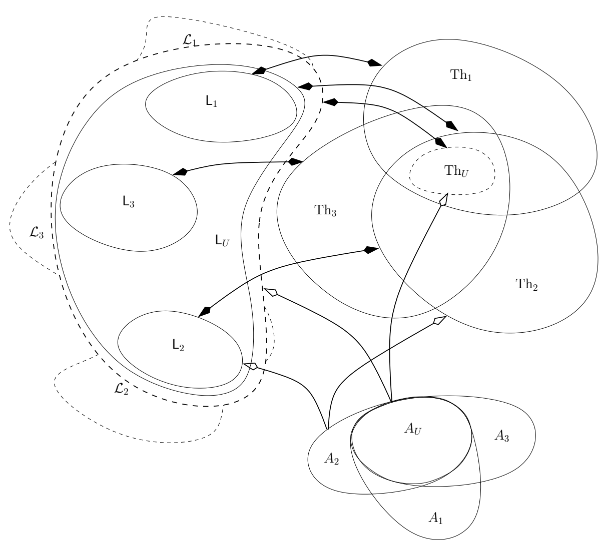
Slide 55

Universal Theories



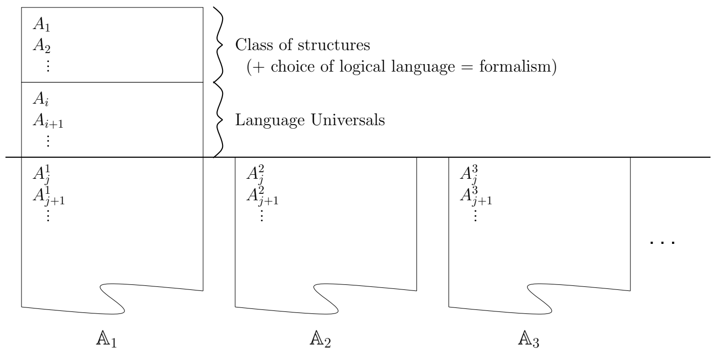
Slide 56

Language Variation



Slide 57

Structure of Axioms



Slide 58

Relevance of FLT to Formal Syntax

- *It's too soon to formalize*
  - Every hypothetical constraint defines a partial theory.
- *Properties of FLT classes are irrelevant to natural language*
  - FLT classes characterize certain fundamental logical languages/classes of structures.
  - *Any* class of structures definable in those logical terms will, consequently, exhibit those properties.
  - But they are not the properties that determine the defined class of structures—the FLT characterizations are consequences of definability.