Minimalismus in zwei regulären und logischen Schritten

Uwe Mönnich and Frank Morawietz

SFB 340, A8
Universität Tübingen

{um,frank}@sfs.nphil.uni-tuebingen.de

CIS München, 04. Februar 1999

Outline

- Motivation and Theoretical Background
- Context-Free Tree Grammars
- LIFT-ing
- Coding of LIFT-ed CF Tree Languages
- Back to the Intended Structures
- Coding of the Intended Structures
- Minimalism Reconsidered

Linguistic Background

J. Rogers (1994,98) gives a formalization within MSO tree logic of a main paradigm in Theoretical Linguistics:

The Principle and Parameters Approach

in the tradition of N. Chomsky

+ Natural Formalization of (almost) all Principles
+ Modular
+ Decidable
+ Descriptive Complexity Result
− Undecidability of *Free Indexation* (Grid)
− Non Context-Free Phenomena in Natural Language

Rogers (1994,1998): $L^2_{K,P}$

- K a set of individual constants
- P a set of predicate constants
- immediate dominance: $\lhd$
- reflexive dominance: $\lhd^*$
- proper dominance: $\lhd^+$
- proper precedence: $\prec$
- usual connectives, variables, equality, set membership and quantifiers:
  - a set of *monadic* second-order variables (X,Y, . . .);
  - a set of individual variables (x,y, . . .);
  - boolean connectives: $\neg$, $\wedge$, . . . ;
  - Quantifiers over individuals and sets: $\exists$, $\forall$;
  - equality and membership: $\approx$, $\in$.

Linguistic Motivation

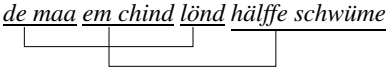
(. . . weil) der Karl die Maria dem Peter den Hans schwimmen lehren helfen liess

German: Palindrome language—CF

(. . . omdat) Karel Marie Piet Jan liet helpen leren zwemmen

Dutch: $a^n b^n$—CF

(. . . wil) mer de maa em chind lönd hälffe schwüme



Swiss-German: $a^n b^m c^n d^m$—Non-CF

Formal Background

Weak Equivalence of Mildly Context-Sensitive Formalisms:

$\mathcal{STR}(\mathcal{HR})$: languages generated by string generating hyperedge replacement grammars

$\mathcal{OUT}(\mathcal{DTWT})$: output languages of deterministic tree-walking tree-to-string transducers

$\mathcal{yDT}_{fc}(\mathcal{REGT})$: yields of images of the regular tree languages under deterministic finite-copying top-down tree transductions

$\mathcal{MCFL}$: languages generated by multiple context-free grammars

$\mathcal{MCTAL}$: languages generated by multi-component tree adjoining grammars

$\mathcal{LCFRL}$: languages generated by linear context-free rewriting systems

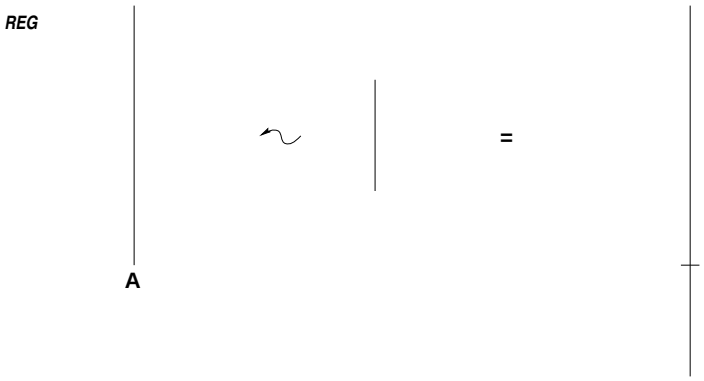
$\mathcal{LUSCL}$: languages generated by local unordered scattered context grammars

General Structure of the Approach

LIFT-ing of Context-Free Tree Languages

$\Rightarrow$ Regular Characterization of the LIFT-ed Languages

$\Rightarrow$ Retrieval of the Intended Structures via Regular Relations



Definition (word models). Let Σ be an alphabet and let $\tau(\Sigma)$ be the vocabulary $\{<\} \cup \{P_a \mid a \in \Sigma\}$, where $<$ is binary and the P_a are monadic. A **word model** for $u \in \Sigma^*$ is a structure of the form

$$M = (B, <, P_a)$$

where $|B| = \mathit{length}(u)$, $<$ is an ordering of B and the P_a correspond to positions in u carrying the label a:

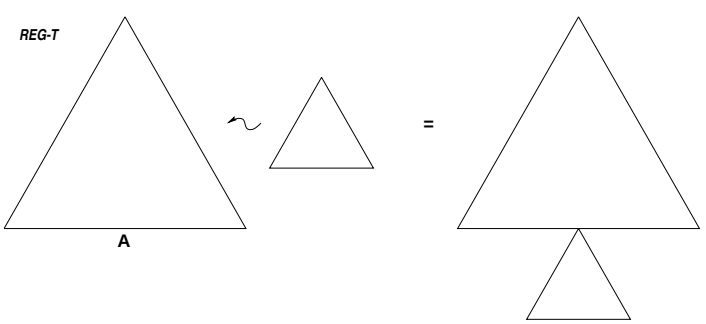
$$P_a = \{b \in B \mid \mathit{label}(b) = a\}.$$

Theorem (Elgot). L is regular over the alphabet Σ iff L is definable in monadic second order logic over the vocabulary $\tau(\Sigma)$. More succinctly:

$L \in \mathcal{L}_{\mathsf{Reg}}$ iff

$\exists A \in \mathit{MSO}(\Sigma)$ such that

$$L = \mathit{Mod}(A) = \{u \in \Sigma^* \mid u \models A\}.$$



Definition (labeled tree models). Let Σ be a ranked alphabet and let $\tau(\Sigma)$ be the vocabulary $\{\mathit{succ}_1, \ldots, \mathit{succ}_n\} \cup \{P_a \mid a \in \Sigma\}$, where the succ_i are binary and the P_a monadic. A **tree model** for $t \in T_\Sigma$ is a structure of the form

$$M = (B, \mathit{succ}_i, <, P_a)_{1 \le i \le n}$$

where B is in one-one correspondence with $\mathit{dom}(t)$, the tree domain of t, the succ_i ($i = 1, \ldots, n$) are the successor functions over the tree domain, $<$ is the prefix order and the P_a correspond to positions with label a:

$$P_a = \{b \in B \mid \mathit{label}(b) = a\}.$$

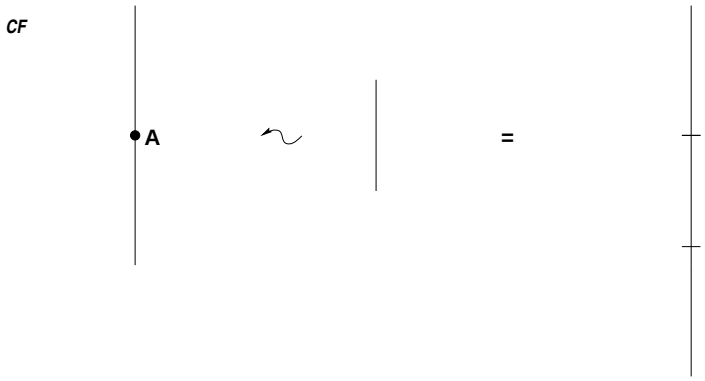
Theorem (Doner, Thatcher–Wright). Let Σ be a ranked alphabet. $L \subseteq \Sigma^*$ is **context-free** iff L is the yield of a set of tree models definable in monadic second order logic over the vocabulary $\tau(\Sigma)$.

$L \in \mathcal{L}_{CF}$ iff

$\exists A \in MSO(\Sigma)$ such that

$$L = \mathrm{yield}(\mathrm{Mod}(A))$$

where $\mathrm{Mod}(A) = \{t \in T_\Sigma \mid t \models A\}$.

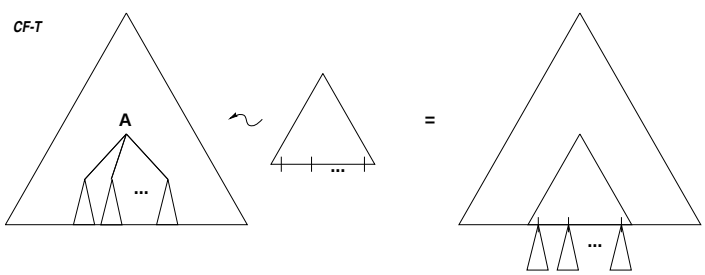


Theorem (Lautemann, Schwentick, Thérien). L is context-free over the alphabet Σ iff L is definable in dyadic second-order logic over the vocabulary $\tau(\Sigma)$ by a Σ^1_1-sentence where the quantified dyadic variables range over matchings. More succinctly:

$L \in \mathcal{L}_{CF}$ iff

$\exists A \in \Sigma^1_1 - DSO(\Sigma)$ such that

$$L = \mathrm{Mod}(A) = \{u \in \Sigma^* \mid u \models A\}.$$



Conjecture. Let Σ be a ranked alphabet. $L \subseteq T_\Sigma$ is a context-free tree language iff L is definable in dyadic second-order logic over the vocabulary $\tau(\Sigma)$ by a Σ^1_1-sentence where the quantified dyadic variables range over matchings. More succinctly:

$L \in \mathcal{L}_{\text{CF-TL}}$ iff

$$\exists A \in \Sigma^1_1 - \mathrm{DSO}(\Sigma) \text{ such that}$$

$$L = \mathrm{Mod}(A) = \{t \in T_\Sigma \mid t \models A\}.$$

Minimalist Grammars (Stabler, 1997)

$$\mathcal{MG} = \langle V, \mathrm{Cat}, \mathrm{Lex}, F \rangle$$

$$\begin{array}{lcl} V & = & P \cup I \\ \mathrm{Cat} & = & \mathsf{base} \cup \mathsf{select} \cup \mathsf{licensee} \cup \mathsf{licensor} \\ \mathrm{Lex} & = & \text{a finite set of trees} \\ F & = & \{\textsc{merge}, \textsc{move}\} \end{array}$$

P		are the phonological features
I		are the interpreted/semantical features
base	=	$\{\mathtt{c}, \mathtt{t}, \mathtt{v}, \mathtt{d}, \mathtt{n}, \ldots\}$
select	=	$\{\mathtt{=x}, \mathtt{=X}, \mathtt{X=} \mid \mathtt{x} \in \mathsf{base}\}$
licensee	=	$\{\mathtt{-case}, \mathtt{-wh}, \ldots\}$
licensor	=	$\{\mathtt{+case}, \mathtt{+CASE}, \mathtt{+wh}, \mathtt{+WH}, \ldots\}$

$$\mathrm{Label} : \mathrm{Leaves} \longrightarrow \mathsf{select}^*(\mathsf{licensor})\mathsf{select}^*\mathsf{licensee}^*(\mathsf{base})P^*I^*$$

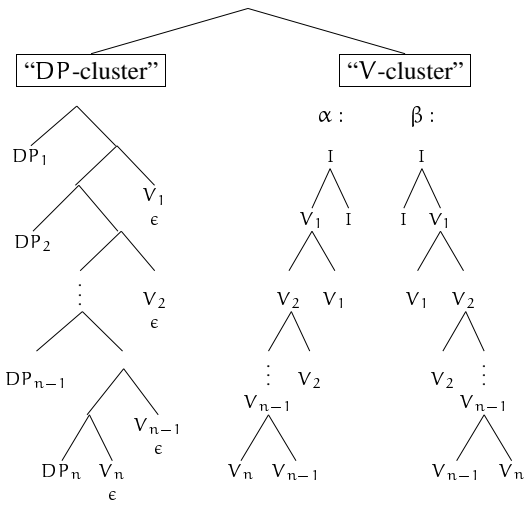
$$\mathcal{L}_{\mathcal{MG}} = \mathrm{Closure}(\mathrm{Lex}, F)$$

"Minimalist" VR

- VR is a lexical property of a certain class of verbs: *lassen-laten-laa, helfen-helpen-hälffe* ... but not *schwimmen-zwemmen-schwüme*;
- VR-verbs take VP-complements;
- VR-verbs (optionally) have a strong V-feature, like the one standardly postulated for the inflectional head I;
- PF serializes complex V_0/I_0 as head-last (German) or head-first (Swiss-German / Dutch).

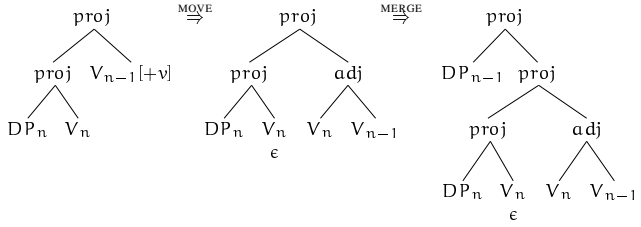
$$\begin{array}{lcl} \mathtt{proj}(x, y) & := & \textit{head-of-}y(x, y) \\ \mathtt{adj}(x, y) & := & [\textit{head-of-}y, \textit{head-of-}y](x, y) \end{array}$$

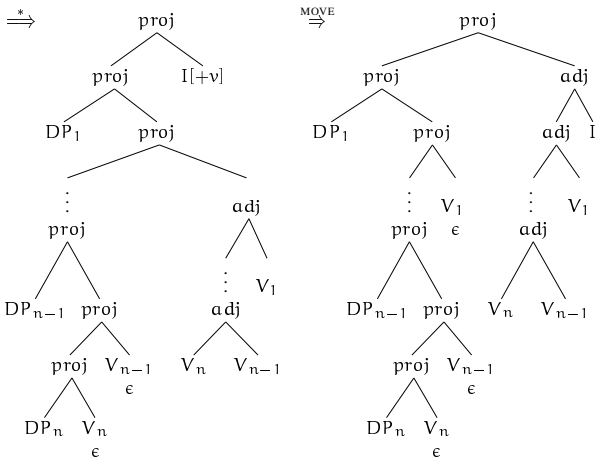
$$\mathtt{adj}(x, y) = \textit{SpellOut(German)} \Rightarrow \mathtt{xy}$$
$$\mathtt{adj}(x, y) = \textit{SpellOut(Dutch, Swiss-German)} \Rightarrow \mathtt{yx}$$



	overt CASE	V-cluster
German	+	α
Dutch	−	β
Swiss-German	+	β

Some steps in a derivation





Minimalist VR: Two Approaches

dp	/DP$_i$/
=dp v	/V$_n$/
V= =dp v	/V$_{n-i}$/
V= i	/I/

dp	/DP$_i$/
=dp v v−	/V$_n$/
=v +V =dp v v−	/V$_{n-i}$/
=v +V i	/I/

Tree Grammars

Definition (Tree Grammar). A *context-free tree grammar* Γ is a 5-tuple $\langle \Sigma, \mathbf{F}, S, \mathbf{X}, \mathbf{P} \rangle$, where

- $\Sigma, \mathbf{F}$ are the ranked alphabets of *inoperatives* and *operatives*;
- $S \in \mathbf{F}$ is the starting symbol;
- $\mathbf{X}$ is a countable set of variables; and
- $\mathbf{P}$ is a set of productions.

The $p \in \mathbf{P}$ have the form

$$F(x_i, \cdots, x_n) \longrightarrow t$$

with $F \in \mathbf{F}$, $x_i, \cdots, x_n \in \mathbf{X}$, and t a term over $\Sigma \cup \mathbf{F} \cup \{x_i, \cdots, x_n\}$.

An application of a rule $F(x_i, \cdots, x_n) \longrightarrow t$ "rewrites" a tree rooted in F as the tree t with its respective variables substituted by F's daughters.

Tree grammars with $\mathbf{F}_n = \emptyset, n \neq 0$, are called *regular*.

Macro-VR

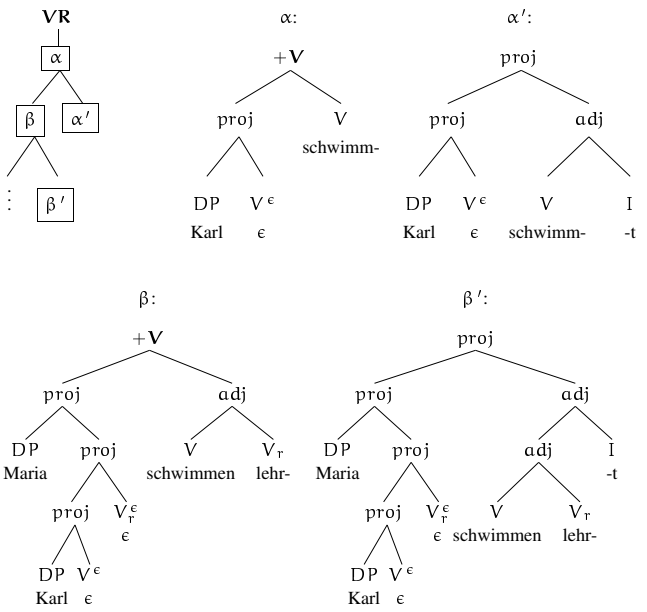
$$\Sigma_0 \text{ (the Lexicon)}$$

$$\Sigma_1 = \{ V, V^{\epsilon}, V_r, V_r^{\epsilon}, DP, I \} \qquad \Sigma_2 = \{ \mathrm{proj}, \mathrm{adj} \}$$

$$\mathbf{F}_0 = \{ VR \} \qquad \mathbf{F}_2 = \{ +V \} \qquad S = VR \qquad \mathbf{V} = \{x, y\}$$

$$\mathbf{P} = \left\{ \begin{array}{rcl} VR & \Rightarrow & +V(\mathrm{proj}(DP, V^{\epsilon}), V) \\ +V(x, y) & \Rightarrow & \mathrm{proj}(x, \mathrm{adj}(y, I)) \\ +V(x, y) & \Rightarrow & +V(\mathrm{proj}(DP, \mathrm{proj}(x, V_r^{\epsilon})), \mathrm{adj}(y, V_r)) \end{array} \right\}$$

plus the obvious rules and symbols for Σ_1



A CF-Tree Grammar for $a^n b^m c^n d^m$: $\mathcal{G}$

$$\Sigma_0 = \{\varepsilon, a, b, c, d\}$$

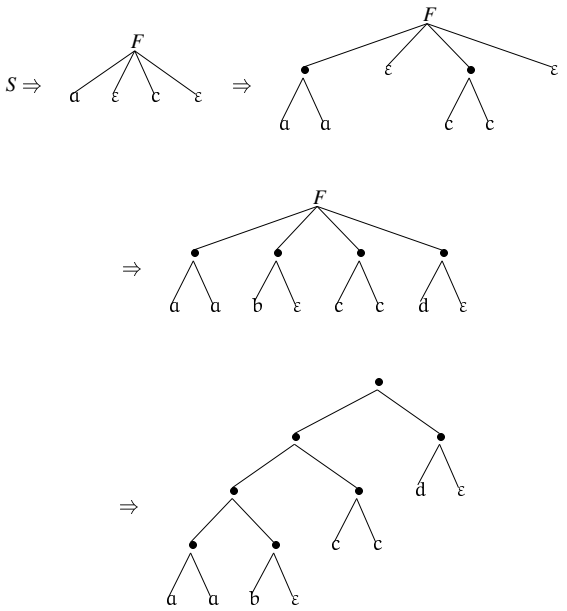
$$\Sigma_2 = \{ \bullet \}$$

$$\mathbf{F}_0 = \{ S \} \qquad \mathbf{F}_4 = \{ F \}$$

$$S = S \qquad \mathbf{X} = \{x_1, x_2, x_3, x_4\}$$

$$\mathbf{P} = \left\{ \begin{array}{rcl} S & \longrightarrow & \varepsilon \\ S & \longrightarrow & F(a, \varepsilon, c, \varepsilon) \\ S & \longrightarrow & F(\varepsilon, b, \varepsilon, d) \\ F(x_1, x_2, x_3, x_4) & \longrightarrow & \\ \multicolumn{3}{c}{F(\bullet(a, x_1), x_2, \bullet(c, x_3), x_4)} \\ F(x_1, x_2, x_3, x_4) & \longrightarrow & \\ \multicolumn{3}{c}{F(x_1, \bullet(b, x_2), x_3, \bullet(d, x_4))} \\ F(x_1, x_2, x_3, x_4) & \longrightarrow & \\ \multicolumn{3}{c}{\bullet(\bullet(\bullet(x_1, x_2), x_3), x_4)} \end{array} \right\}$$

An example derivation of aabccd



Definition (LIFT). Suppose that Σ is a ranked alphabet. The *derived* $\mathbb{N}$-sorted alphabet Σ^{L} is defined as follows: For each $n \geq 0$, $\Sigma'_n = \{f' | f \in \Sigma_n\}$ is a new set of symbols of sort n; for each $n \geq 1$ and each $i, 1 \leq i \leq n$, π^n_i is a new symbol, the ith projection symbol of sort n; for each $n \geq 0, k \geq 0$ the new symbol $c_{n,k}$ is the (n,k)th composition symbol. The set of all $c_{n,k}$ will be denoted by $\mathbf{C}$.

$$\Sigma^{L}_0 = \Sigma'_0$$
$$\Sigma^{L}_n = \Sigma'_n \cup \{\pi^n_i | 1 \leq i \leq n\} \text{ for } n \geq 1$$
$$\Sigma^{L}_{n,k} = \{c_{n,k}\} \text{ for } n, k \geq 0$$
$$\Sigma^{L}_i = \emptyset \text{ otherwise}$$

For $k \geq 0$, $\text{LIFT}^{\Sigma}_k : T(\Sigma, X_k) \to T(\Sigma^{L}, k)$ is defined as follows:

$$\text{LIFT}^{\Sigma}_k(x_i) = \pi^k_i$$
$$\text{LIFT}^{\Sigma}_k(f) = c_{0,k}(f') \text{ for } f \in \Sigma_0$$
$$\text{LIFT}^{\Sigma}_k(f(t_1, \ldots, t_n)) = c_{n,k}(f', \text{LIFT}^{\Sigma}_k(t_1), \ldots, \text{LIFT}^{\Sigma}_k(t_n)) \text{ for } f \in \Sigma_n, n \geq 1$$

A LIFT-ed Regular Tree Grammar: $\mathcal{G}'$

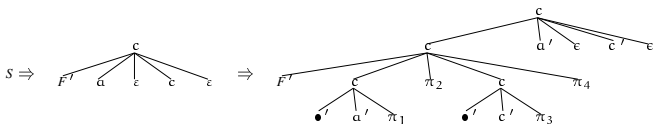
$\Sigma^{L}_0 = \{\varepsilon, a', b', c', d', S'\}$ $\quad \Sigma^{L}_2 = \{\bullet'\}$

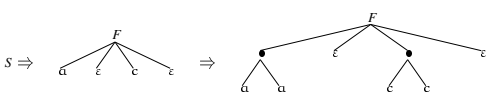
$\Sigma^{L}_4 = \{F'\} \cup \{\pi_1, \pi_2, \pi_3, \pi_4\}$ $\quad \Sigma^{L}_{n,k} = \{c\}$ for simplicity

$\mathbf{F}_0 = \{ S', F' \}$ $\quad S = S'$ $\quad \mathbf{X} = \emptyset$

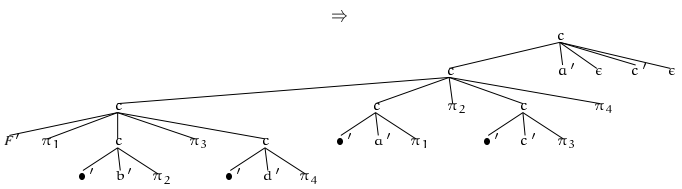
$$\mathbf{P} = \left\{ \begin{array}{lcl} S' & \longrightarrow & \varepsilon \\ S' & \longrightarrow & c(F', a', \varepsilon, c', \varepsilon) \\ S' & \longrightarrow & c(F', \varepsilon, b', \varepsilon, d') \\ F' & \longrightarrow & c(F', c(\bullet', a', \pi_1), \pi_2, c(\bullet', c', \pi_3), \pi_4) \\ F' & \longrightarrow & c(F', \pi_1, c(\bullet', b', \pi_2), \pi_3, c(\bullet', d', \pi_4)) \\ F' & \longrightarrow & c(\bullet', c(\bullet', c(\bullet', \pi_1, \pi_2), \pi_3), \pi_4) \end{array} \right\}$$

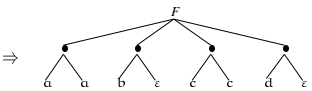
A sample derivation I



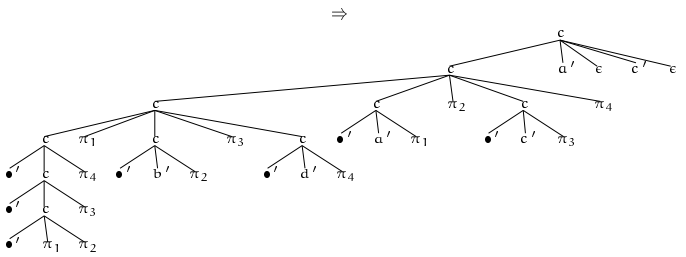


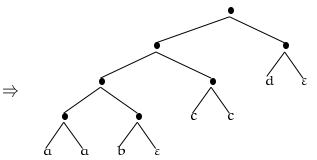
A sample derivation II





A sample derivation III





Definition (Bottom-Up Tree Automaton).

$$\mathfrak{A} = \langle A, \Sigma, \delta, a_0, F \rangle$$

- A the (finite) set of states,
- Σ a ranked alphabet,
- $a_0 \in A$ the initial state,
- $F \subseteq A$ the final states and
- $\delta : (A^n \times \Sigma_n) \to A$ the transition function.

$$h_\delta(\varepsilon) = a_0$$
$$h_\delta(\sigma(t_1, \ldots, t_n)) = \delta(h_\delta(t_1), \ldots, h_\delta(t_n), \sigma)$$
$$t_i \in T_\Sigma, 1 \le i \le n, \sigma \in \Sigma_n$$

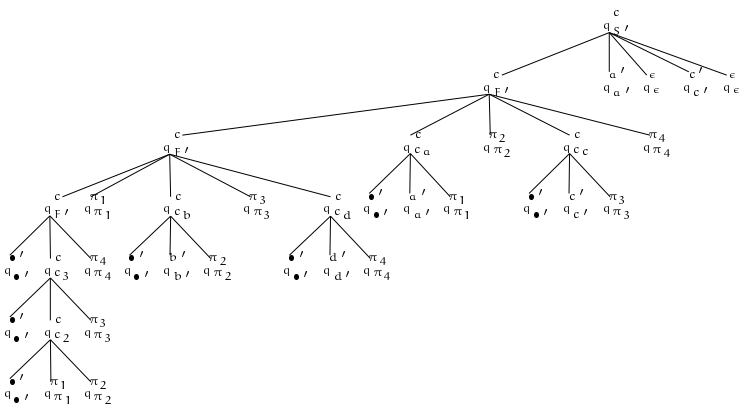
$$T(\mathfrak{A}) = \{t \mid h_\delta(t) \in F\}$$

A Tree Automaton for the LIFT-ed CF Tree Grammar

$$\mathfrak{A}_{G'} = \langle Q, \Sigma, \delta, q_0, \{q_{S'}\} \rangle$$

$$\begin{array}{llll}
(q_0,q_0,q_0,q_0,q_0,\epsilon) \to q_\epsilon & (q_0,q_0,q_{\bullet'},q_{a'},q_{\pi_1},c) \to q_{c_a} \\
(q_0,q_0,q_0,q_0,q_0,a') \to q_{a'} & (q_0,q_0,q_{\bullet'},q_{c'},q_{\pi_3},c) \to q_{c_c} \\
(q_0,q_0,q_0,q_0,q_0,b') \to q_{b'} & (q_0,q_0,q_{\bullet'},q_{b'},q_{\pi_2},c) \to q_{c_b} \\
(q_0,q_0,q_0,q_0,q_0,c') \to q_{c'} & (q_0,q_0,q_{\bullet'},q_{d'},q_{\pi_4},c) \to q_{c_d} \\
(q_0,q_0,q_0,q_0,q_0,d') \to q_{d'} & (q_0,q_0,q_{\bullet'},q_{\pi_1},q_{\pi_2},c) \to q_{c_2} \\
(q_0,q_0,q_0,q_0,q_0,\pi_1) \to q_{\pi_1} & (q_0,q_0,q_{\bullet'},q_{c_2},q_{\pi_3},c) \to q_{c_3} \\
(q_0,q_0,q_0,q_0,q_0,\pi_2) \to q_{\pi_2} & (q_0,q_0,q_{\bullet'},q_{c_3},q_{\pi_4},c) \to q_{F'} \\
(q_0,q_0,q_0,q_0,q_0,\pi_3) \to q_{\pi_3} & (q_{F'},q_{\pi_1},q_{c_b},q_{\pi_3},q_{c_d},c) \to q_{F'} \\
(q_0,q_0,q_0,q_0,q_0,\pi_4) \to q_{\pi_4} & (q_{F'},q_{c_a},q_{\pi_2},q_{c_c},q_{\pi_4},c) \to q_{F'} \\
(q_0,q_0,q_0,q_0,q_0,\bullet') \to q_{\bullet'} & (q_{F'},q_\epsilon,q_b,q_\epsilon,q_d,c) \to q_{S'} \\
(q_0,q_0,q_0,q_0,q_0,\epsilon) \to q_{S'} & (q_{F'},q_a,q_\epsilon,q_c,q_\epsilon,c) \to q_{S'}
\end{array}$$

An Example Run



Goal:

Define a set of relations

$\mathsf{R}^{\mathsf{I}} = \{\blacktriangleleft^\star \textit{ (dominance)}, \blacktriangleleft, \blacktriangleleft^+, \lhd \textit{ (precedence)}, \textit{ c-command}, \ldots\}$

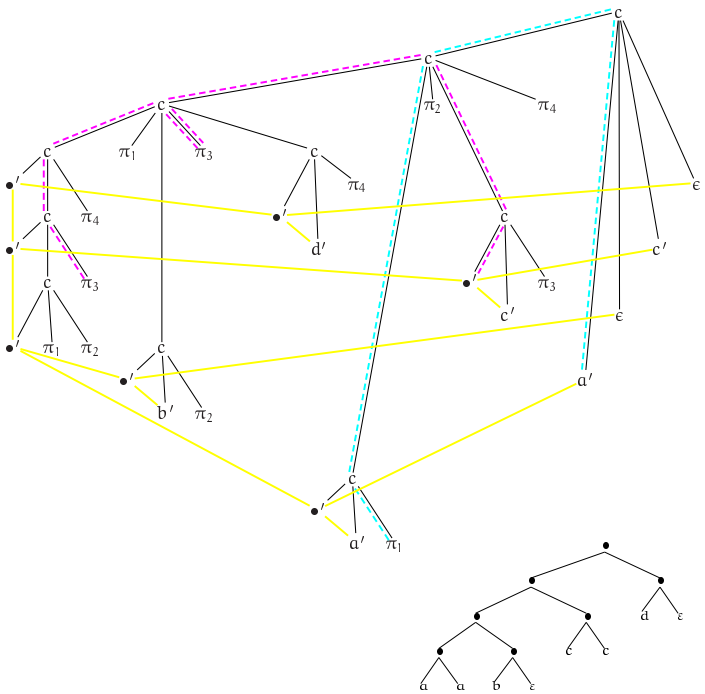
holding between the nodes $\mathfrak{n} \in \mathsf{N}^{\mathsf{L}}$ of the explicit tree T^{L} which carry a *"linguistic"* label $\mathfrak{l} \in \mathcal{T}^{\mathsf{L}} \cup \mathcal{N}^{\mathsf{L}}$ $(\subseteq \bigcup_{n \geq 0} \Sigma_n)$ in such a way, that when interpreting

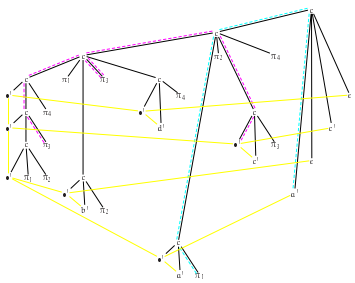
$\blacktriangleleft^\star \in \mathsf{R}^{\mathsf{I}}$ as a tree order on the set of *"linguistic"* nodes and $\lhd \in \mathsf{R}^{\mathsf{I}}$ as the precedence relation on the resulting structure,

$\langle \{\mathfrak{n} \mid \mathfrak{n} \in \mathsf{N}^{\mathsf{L}} \wedge \ell(\mathfrak{n}) \in \mathcal{T}^{\mathsf{L}} \cup \mathcal{N}^{\mathsf{L}}\}, \blacktriangleleft^\star, \lhd \rangle$

is in fact the intended tree corresponding to T^{L}.

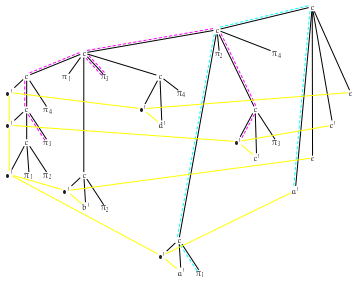
Reconstructing the Intended Structures





A few general facts

1. Our trees feature three families of labels: the "linguistic" symbols $\mathbf{L} = \mathcal{T}^{\mathsf{L}} \cup \mathcal{N}^{\mathsf{L}}$, where $\mathcal{T} = \Sigma_0$ of the underlying macro-grammar and $\mathcal{N} = \bigcup_{n \geq 1} \Sigma_n$; the "composition" symbols $\mathbf{C} = \bigcup_{n,k \geq 0} \mathsf{C}_{n,k}$; and the "projection" symbols Π.
2. All non-terminal nodes in T^{L} are labeled by some $\mathsf{c} \in \mathbf{C}$. No terminal node is labeled by some c.
3. The terminal nodes in T^{L} are either labeled by some "linguistic" symbol or by some "projection" symbol $\pi \in \Pi_{\mathfrak{i}}$.



A few facts (cont'd.)

4. Any "linguistic" node dominating anything in the intended tree is on some left branch in T^L, i.e., it is the left-most daughter^L of some $c \in \mathbf{C}$.
5. For any node p labeled with some "projection" symbol $\pi \in \Pi_i$ in T^L there is a unique node n (labeled with some $c \in \mathbf{C}$ by (2.)) which properly dominates p and whose i-th sister will eventually evaluate to the value of π. Moreover, n will be the first node properly dominating p which is on a left branch.

MSO definable transductions: $\mathcal{R} \rightsquigarrow \mathcal{Q}$

$$(\varphi, \psi, (\theta_q)_{q \in Q})$$

φ the domain of the transduction

ψ the resulting domain of $\mathcal{Q}$

θ_q the new relations

Definition (Tree-Walking Automaton). A *tree-walking automaton (with tests)* over some ranked alphabet Σ is a finite automaton $\mathfrak{A} = (Q, \Delta, \delta, I, F)$ with states Q, directives Δ, transitions $\delta : Q \times \Delta \to Q$ and the initial and final states $I \subseteq Q$ and $F \subseteq Q$ which traverses a tree using three kinds of directives:

$\uparrow_i$ "move up to the mother of the current node (if it has one and it is its i-th daughter)",

$\downarrow_i$ "move to the i-th daughter of the current node (if it exists)",

$\varphi(x)$ "verify that φ holds at the current node".

For any tree t, such an automaton $\mathfrak{A}$ computes a node relation

$$R_t(\mathfrak{A}) = \{(x, y) \mid (x, q_i) \stackrel{*}{\Longrightarrow} (y, q_f) \text{ for some } q_i \in I \text{ and some } q_f \in F\}$$

where for all states $q_i, q_j \in Q$ and nodes x, y in t

$$(x, q_i) \Rightarrow (y, q_j) \text{ iff } \exists d \in \Delta : (q_i, d, q_j) \in \delta \text{ and } y \text{ is reachable from } x \text{ in } t \text{ via } d.$$

If all the tests $\varphi(x)$ of $\mathfrak{A}$ are MSO definable, $\mathfrak{A}$ specifies a *regular tree-node relation*, which is itself MSO definable (B & E 1997).

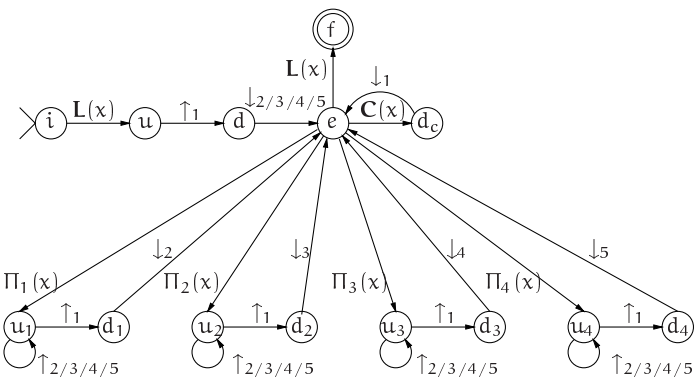
The tree-walking automaton $A_{\blacktriangleleft}$

$$A_{\blacktriangleleft} = (Q, \Delta, \delta, I, F)$$

$$Q = \{i, e, d, d_1, d_2, d_3, d_4, d_c, u, u_1, u_2, u_3, u_4, f\}$$

$$I = \{i\} \qquad F = \{f\}$$

$$\Delta = \{\downarrow_1, \downarrow_2, \downarrow_3, \downarrow_4, \downarrow_5, \uparrow_1, \uparrow_2, \uparrow_3, \uparrow_4, \uparrow_5, \mathbf{L}(x), \mathbf{C}(x), \Pi_1(x), \Pi_2(x), \Pi_3(x), \Pi_4(x)\}$$



The walking language $W_{\blacktriangleleft}$

$$W_{\blacktriangleleft} = \mathbf{L}(x) \cdot \uparrow_1 \cdot (\downarrow_2 \cup \downarrow_3 \cup \downarrow_4 \cup \downarrow_5) \cdot (W_{\mathbf{C}} \cup W_{\Pi_1} \cup W_{\Pi_2} \cup W_{\Pi_3} \cup W_{\Pi_4})^* \cdot \mathbf{L}(x)$$

$$\begin{aligned}
W_{\mathbf{C}} &= \mathbf{C}(x) \cdot \downarrow_1 \\
W_{\Pi_1} &= \Pi_1(x) \cdot (\uparrow_2 \cup \uparrow_3 \cup \uparrow_4 \cup \uparrow_5)^* \cdot \uparrow_1 \cdot \downarrow_2 \\
W_{\Pi_2} &= \Pi_2(x) \cdot (\uparrow_2 \cup \uparrow_3 \cup \uparrow_4 \cup \uparrow_5)^* \cdot \uparrow_1 \cdot \downarrow_3 \\
W_{\Pi_3} &= \Pi_3(x) \cdot (\uparrow_2 \cup \uparrow_3 \cup \uparrow_4 \cup \uparrow_5)^* \cdot \uparrow_1 \cdot \downarrow_4 \\
W_{\Pi_4} &= \Pi_4(x) \cdot (\uparrow_2 \cup \uparrow_3 \cup \uparrow_4 \cup \uparrow_5)^* \cdot \uparrow_1 \cdot \downarrow_5
\end{aligned}$$

$W_{\blacktriangleleft}$ can be inductively translated into a MSO-formula $\mathit{trans}_{W_{\blacktriangleleft}}(x, y)$.

Reflexive transitive closure in MSO (on finite structures)

$$\begin{aligned}
R\text{-}\mathit{closed}(X) &\equiv (\forall x, y)[x \in X \wedge R(x, y) \rightarrow y \in X] \\
[R^*(x, y) &\equiv (\forall X)[R\text{-}\mathit{closed}(X) \wedge x \in X \rightarrow y \in X]
\end{aligned}$$

Walking language $\Rightarrow$ MSO-formula

$$\begin{aligned}
\mathit{trans}_{\emptyset}(x, y) &= \bot \\
\mathit{trans}_{\downarrow_i}(x, y) &= \mathit{edge}_i(x, y) \\
\mathit{trans}_{\uparrow_i}(x, y) &= \mathit{edge}_i(y, x) \\
\mathit{trans}_{\varphi(x)}(x, y) &= \varphi(x) \wedge x = y \\
\mathit{trans}_{W_1 \cup W_2}(x, y) &= \mathit{trans}_{W_1}(x, y) \vee \mathit{trans}_{W_2}(x, y) \\
\mathit{trans}_{W_1 \cdot W_2}(x, y) &= \exists z\,(\mathit{trans}_{W_1}(x, z) \wedge \mathit{trans}_{W_2}(z, y)) \\
\mathit{trans}_{W^*}(x, y) &= \mathit{trans}^*_W(x, y) \\
\mathit{trans}^*_W(x, y) &= \forall X\,(\forall v, w\,(v \in X \wedge \mathit{trans}_W(v, w) \rightarrow w \in X) \\
&\qquad \wedge x \in X \rightarrow y \in X)
\end{aligned}$$

Intended Structures via MSO transduction

$$(\varphi, \psi, (\theta_q)_{q \in Q})$$

$$Q = \{\blacktriangleleft, \blacktriangleleft^\star, \blacktriangleleft^+, \lhd, \dots\}$$

$$\begin{aligned}
\varphi &= \text{ the formula coding } \mathfrak{A}_{\mathcal{G}'} \\
\psi &= (\exists y)[x \blacktriangleleft y \vee y \blacktriangleleft x] \\
\theta_{\blacktriangleleft} &= \text{ the formula for the walking language of } \mathfrak{A}_{\blacktriangleleft} \\
\theta_{\blacktriangleleft^\star} &= (\forall X)[\blacktriangleleft\text{-}\mathit{closed}(X) \vee x \in X \rightarrow y \in X] \\
\theta_{\blacktriangleleft^+} &= x \blacktriangleleft^\star y \vee x \not\approx y \\
\theta_{\lhd} &= \text{ another tree-walking automaton} \\
\theta_{\text{labels}} &= \text{ as in the old structure } R
\end{aligned}$$

Outlook

Theorem (Courcelle). The image of an MSO-definable class of structures under a definable transduction is not MSO definable in general.

But which class of structures is captured using regular means/MSO logic combined with regular/MSO definable binary node relations?

Is there a machine model for the MSO transduction such that we can implement the process?

Derivational Minimalism

Theorem (Michaelis). Each Minimalist Grammar (MG) in the sense of Stabler (1997) can be translated into a weakly equivalent multiple context-free grammar (MCFG) and is therefore mildly context-sensitive.

Basic Definitions I

Let S be a set of sorts. A *many-sorted signature* Σ is an indexed family

$$\langle \Sigma_{w,s} \mid w \in S^*, s \in S \rangle$$

of disjoint sets. A symbol in $\Sigma_{w,s}$ is called an:

operator of type $\langle w, s \rangle$,
arity w,
sort s and
rank $\ell(w)$.

In the case of a single-sorted signature we write $\Sigma_{s^n,s}$ as Σ_n.

Basic Definitions II

The set of k-ary trees $T(\Sigma, X_k)$ over such a single-sorted signature Σ is built up from a finite set $X_k = \{x_1, \ldots, x_k\}$ of variables using the operators in the expected way:

If $\sigma \in \Sigma_n$ and $t_1, \ldots, t_n$ are trees, then

$$\sigma(t_1, \ldots, t_n) \text{ is a } k\text{-ary tree.}$$

All $x_i \in X_k$ are 0-ary trees.

Basic Definitions III

A Σ-*algebra* $\mathbb{A}$ consists of an S-indexed family of sets $A = \langle A_s \rangle_{s \in S}$ and for each operator $\sigma \in \Sigma_{w,s}$ a function

$$\sigma_{\mathbb{A}} : A^w \to A^s$$
$$A^w = A^{w_1} \times \cdots \times A^{w_n}$$
$$w = w_1 \cdots w_n.$$

The set of k-ary trees $T(\Sigma, X_k)$ can be made into a Σ-algebra $\mathbb{T}$ by specifying the operations as follows.

For every $\sigma \in \Sigma_n$ and every $t_1, \ldots, t_n \in T(\Sigma, X_k)$ we identify

$$\sigma_{\mathbb{T}}(t_1, \ldots, t_n) \text{ with } \sigma(t_1, \ldots, t_n).$$

Lawvere Algebras I

Given a set of sorts S, a Lawvere theory, as an algebra, is an $S^* \times S^*$-sorted algebra $\mathbb{T}$, whose carriers $\langle T(u, v) | u, v \in S^* \rangle$ consist of the morphisms of the theory and whose operations are of the following types:

projection: $\pi_i^u \in T(u, u_i) \qquad (u = u_1 \ldots u_n \in S^*)$

composition: $c_{u,v,w} \in T(u, v) \times T(v, w) \to T(u, w) \qquad (u, v, w \in S^*)$

target tupling: $(\quad)_{(u,v)} \in T(u, v_1) \times \cdots \times T(u, v_n) \to T(u, v) \qquad (u, v = v_1 \ldots v_n \in S^*)$

The projections and the operations of target tupling are required to satisfy the obvious identities for products and the composition operations must satisfy associativity.

Lawvere Algebras II

With S being a singleton, the power set $\wp(T(\Sigma, X_k))$ of k-ary trees constitutes the central example of interest for formal language theory. The carriers

$$\langle \wp(T(k, m)) \mid k, m \in w \rangle$$

consist of sets of m-tuples of k-ary trees $\{(t_1, \ldots, t_m)\}$.

MCFG

$G = \langle N, T, F, P, S \rangle$

nonterminals N

terminals T

basic morphisms F

start symbol $S \in N$

productions P

$$A \longrightarrow f(A_0, \ldots, A_{n-1})$$
$$A, A_0, \ldots, A_{n-1} \in N$$
$$f \in F : (T^*)^k \longrightarrow (T^*)^l$$
$$k = \Sigma_{i=0}^{n-1} k_i \, (k_i \text{ the rank of } A_i)$$
l the rank of A

An Example Grammar

$$\mathcal{G} = \langle \{S, A\}, \{a_1, a_2, a_3\}, S, P \rangle$$

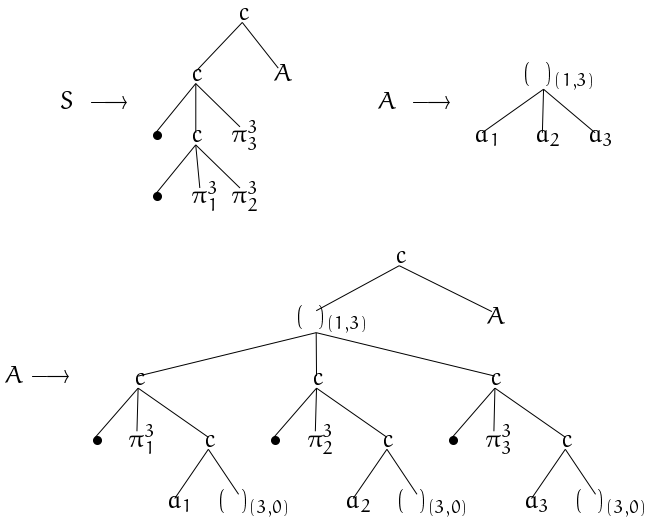
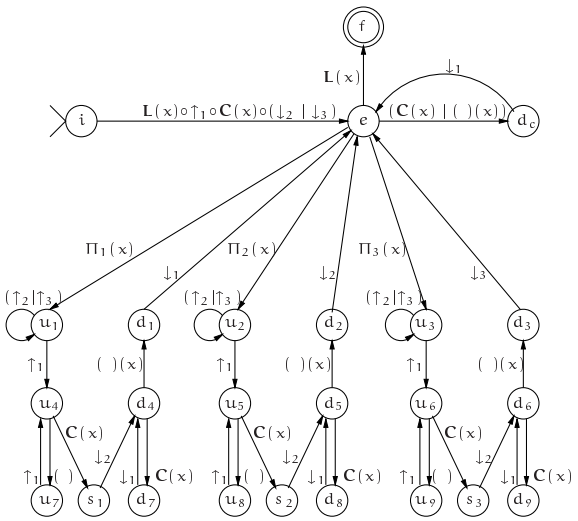
$$P = \left\{ \begin{array}{lcl} S & \rightarrow & g(A) \\ A & \rightarrow & f() \\ A & \rightarrow & h(A) \end{array} \right\}$$

$$\begin{array}{lcl} g(x,y,z) & = & xyz \\ f() & = & (a_1, a_2, a_3) \\ h(x,y,z) & = & (xa_1, ya_2, za_3) \end{array}$$

$$\begin{array}{lcl} a_1, a_2, a_3 & : & (0,1) \\ A & : & (0,3) \\ \bullet, \pi_1^3, \pi_2^3, \pi_3^3 & : & (3,1) \end{array}$$

$$\mathcal{L}(\mathcal{G}) = a_1^n a_2^n a_3^n$$

The Translated Example Grammar



The Tree-Walking Automaton for Dominance