An Operational and Denotational Account of Second-Order Substitution

Uwe Mönnich and Frank Morawietz

Seminar für Sprachwissenschaft
Universität Tübingen

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

http://tcl.sfs.nphil.uni-tuebingen.de

Passau, 24. October 2000

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

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: $\triangleleft$
- reflexive dominance: $\triangleleft^*$
- proper dominance: $\triangleleft^+$
- 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

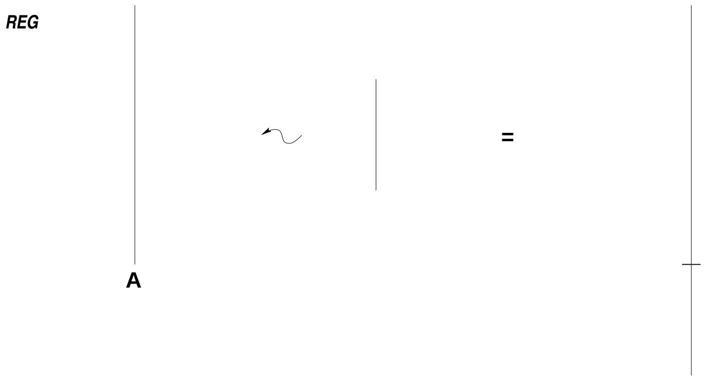
Swiss-German: $a^n b^m c^n d^m$—Non-CF

General Structure of the Approach

LIFT-ing of Context-Free Tree Languages

⇒ Regular Characterization of the LIFT-ed Languages

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

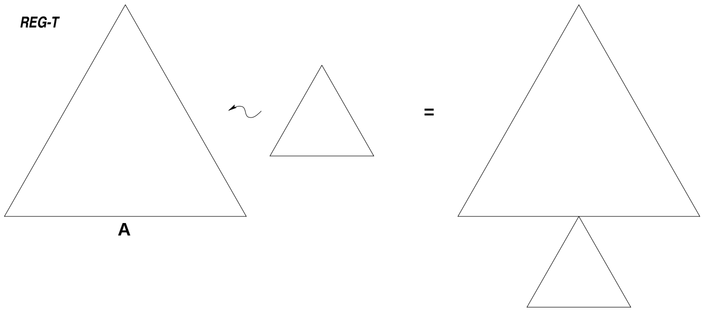
$$P_a = \{b \in B \mid \mathtt{label}(b) = a\}.$$

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

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

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



Definition (labeled tree models). Let Σ be a ranked alphabet and let $\tau(\Sigma)$ be the vocabulary $\{succ_1, \dots, succ_n\} \cup \{P_a \mid a \in \Sigma\}$, 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

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

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

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

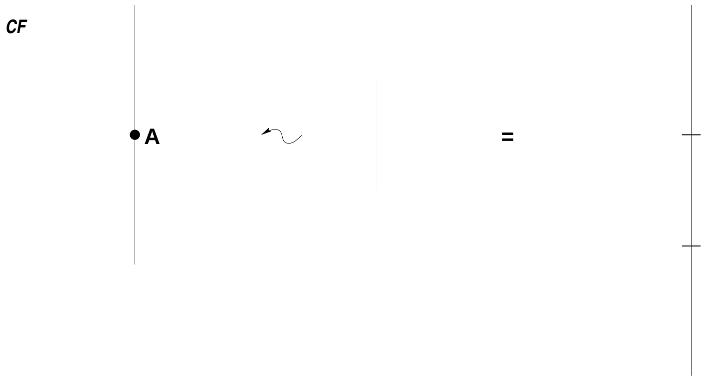
Theorem (Doner, Thatcher–Wright). Let Σ be a ranked alphabet. $L \subseteq \Sigma^*$ is **context-free** iff L is the yield of a set of tree models definable in monadic second order logic over the vocabulary $\tau(\Sigma)$.

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

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

$$L = yield(Mod(A))$$

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

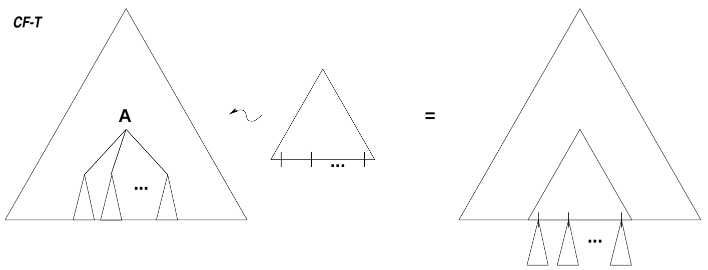


Theorem (Lautemann, Schwentick, Thérien). L is context-free over the alphabet Σ iff L is definable in dyadic second-order logic over the vocabulary $\tau(\Sigma)$ by a Σ_1^1-sentence where the quantified dyadic variables range over matchings. More succinctly:

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

$$\exists A \in \Sigma_1^1 - \mathrm{DSO}(\Sigma) \text{ such that}$$

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



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:

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

$$\exists A \in \Sigma_1^1 - \mathrm{DSO}(\Sigma) \text{ such that}$$

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

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.

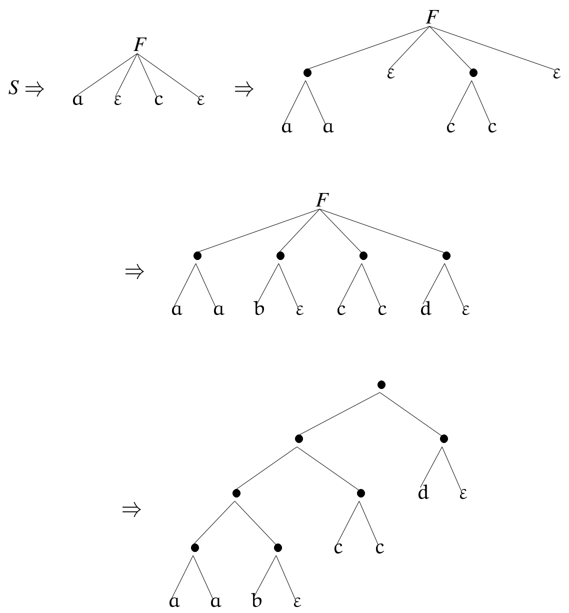
Tree grammars with $\mathbf{F}_n = \emptyset, n \neq 0$, are called *regular*.

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

$$\Sigma_0 = \{\varepsilon, a, b, c, d\}$$
$$\Sigma_2 = \{\bullet\}$$
$$\mathbf{F}_0 = \{\, S \,\} \qquad \mathbf{F}_4 = \{\, F \,\}$$
$$S = S \qquad \mathbf{X} = \{x_1, x_2, x_3, x_4\}$$

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

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 $k \geq 0$, $\text{LIFT}^\Sigma_k : T(\Sigma, \mathbf{X}_k) \to T(\Sigma^L, k)$ 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: $\mathcal{G}'$

$\Sigma_0^L = \{\varepsilon, a', b', c', d', S'\}$ $\quad \Sigma_2^L = \{\bullet'\}$

$\Sigma_4^L = \{F'\} \cup \{\pi_1, \pi_2, \pi_3, \pi_4\}$ $\quad \Sigma_{n,k}^L = \{c\}$ for simplicity

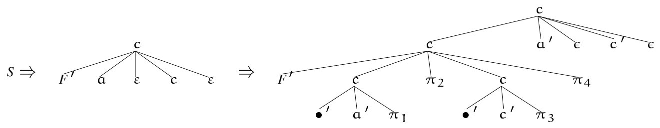
$F_0 = \{\, S', F' \,\} \qquad S = S' \qquad X = \emptyset$

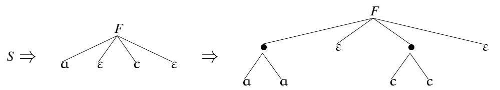
$$P = \left\{ \begin{array}{lcl} S' & \longrightarrow & \varepsilon \\ S' & \longrightarrow & c(F', a', \varepsilon, c', \varepsilon) \\ S' & \longrightarrow & c(F', \varepsilon, b', \varepsilon, d') \\ F' & \longrightarrow & c(F', c(\bullet', a', \pi_1), \pi_2, c(\bullet', c', \pi_3), \pi_4) \\ F' &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

$\Sigma_0 = \{\varepsilon, a, b, c, d\} \quad \Sigma_2 = \{\bullet\} \quad F_0 = \{S\} \quad F_4 = \{F\} \quad S = S \quad X = \{x_1, x_2, x_3, x_4\}$

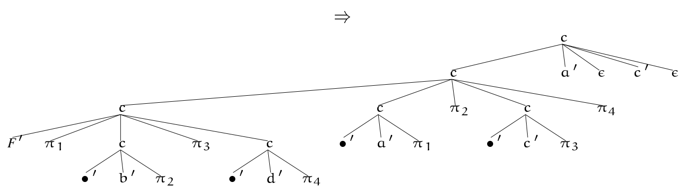
$$P = \left\{ \begin{array}{rcl} S & \longrightarrow & \varepsilon \\ S & \longrightarrow & F(a, \varepsilon, c, \varepsilon) \\ S & \longrightarrow & F(\varepsilon, b, \varepsilon, d) \\ F(x_1, x_2, x_3, x_4) & \longrightarrow & F(\bullet(a, x_1), x_2, \bullet(c, x_3), x_4) \\ F(x_1, x_2, x_3, x_4)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

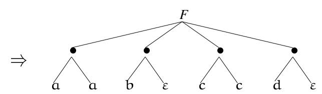
A sample derivation I



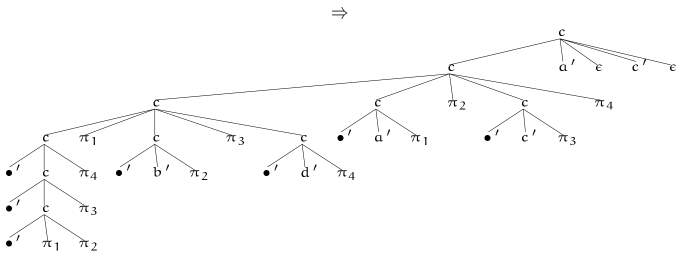


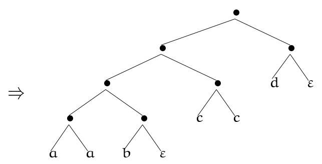
A sample derivation II





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.

$$\begin{aligned} h_\delta(\varepsilon) &= a_0 \\ h_\delta(\sigma(t_1, \ldots, t_n)) &= \delta(h_\delta(t_1), \ldots, h_\delta(t_n), \sigma) \\ & \quad t_i \in T_\Sigma, 1 \leq i \leq n, \sigma \in \Sigma_n \end{aligned}$$

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

A Tree Automaton for the LIFT-ed CF Tree Grammar

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

$$\delta : \bigcup_{1 \leq n \leq 5} Q^n \times \Sigma \to Q$$

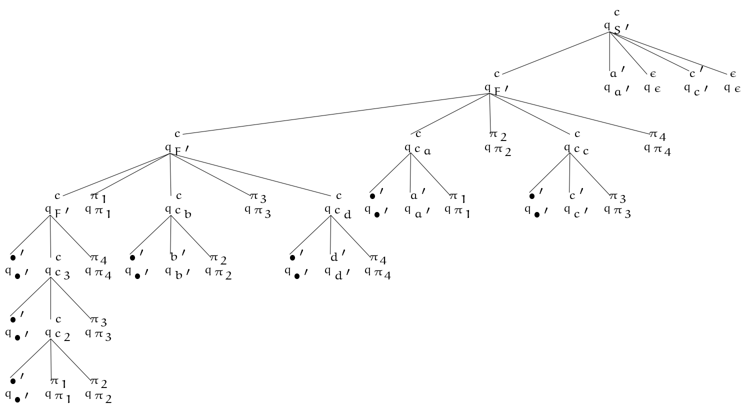
$$\begin{array}{lll} (q_0, \epsilon) \to q_\epsilon & & (q_{\bullet'}, q_{a'}, q_{\pi_1}, c) \to q_{C_a} \\ (q_0, a') \to q_{a'} & & (q_{\bullet'}, q_{c'}, q_{\pi_3}, c) \to q_{C_c} \\ (q_0, b') \to q_{b'} & & (q_{\bullet'}, q_{b'}, q_{\pi_2}, c) \to q_{C_b} \\ (q_0, c') \to q_{c'} & & (q_{\bullet'}, q_{d'}, q_{\pi_4}, c) \to q_{C_d} \\ (q_0, d') \to q_{d'} & & (q_{\bullet'}, q_{\pi_1}, q_{\pi_2}, c) \to q_{C_2} \\ (q_0, \pi_1) \to q_{\pi_1} & & (q_{\bullet'}, q_{C_2}, q_{\pi_3}, c) \to q_{C_3} \\ (q_0, \pi_2) \to q_{\pi_2} & & (q_{\bullet'}, q_{C_3}, q_{\pi_4}, c) \to q_{F'} \\ (q_0, \pi_3) \to q_{\pi_3} & & (q_{F'}, q_{\pi_1}, q_{C_b}, q_{\pi_3}, q_{C_d}, c) \to q_{F'} \\ (q_0, \pi_4) \to q_{\pi_4} & & (q_{F'}, q_{C_a}, q_{\pi_2}, q_{C_c}, q_{\pi_4}, c) \to q_{F'} \\ (q_0, \bullet') \to q_{\bullet'} & & (q_{F'}, q_\epsilon, q_b, q_\epsilon, q_d, c) \to q_{S'} \\ (q_0, \epsilon) \to q_{S'} & & (q_{F'}, q_a, q_\epsilon, q_c, q_\epsilon, c) \to q_{S'} \end{array}$$

Coding RTGs in MSO Logic

- Define a tree automaton $\mathfrak{A}_{\mathcal{G}'}$ from the RTG $\mathcal{G}'$
- Code its behaviour in $\varphi_{\mathfrak{A}_{\mathcal{G}'}}$

$$\begin{aligned} &(\exists X_0, \ldots, X_m)[\bigwedge_{i \neq j}(\neg\exists y)[y \in X_i \wedge y \in X_j] \wedge \\ &\quad (\forall x)[(\neg\exists y)[x \lhd y] \to x \in X_0] \wedge \\ &\quad (\forall x_1, \ldots, x_n, y)[\bigvee_{\substack{(i_1, \ldots, i_n, \sigma, j) \in \alpha \\ 1 \leq k \leq n}} x_k \in X_{i_k} \wedge y \lhd x_k \wedge y \in X_j \wedge y \in P_\sigma] \\ &\quad \bigvee_{i \in F}(\exists x \forall y)[x \lhd^* y \wedge x \in X_i] \end{aligned}$$

An Example Run



Goal:

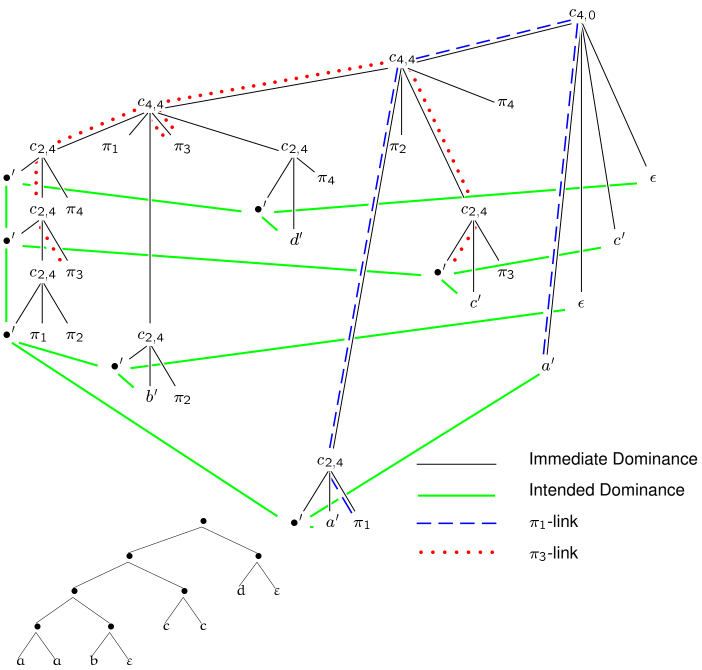
Define a set of relations
$R^I = \{\blacktriangleleft^\star$ *(dominance)*, $\blacktriangleleft, \blacktriangleleft^+, \lhd$ *(precedence)*, *c-command*, $\ldots\}$
holding between the nodes $n \in N^L$ of the explicit tree T^L
which carry a *"linguistic"* label $l \in \mathcal{T}^L \cup \mathcal{N}^L$ ($\subseteq \bigcup_{n \geq 0} \Sigma_n$)
in such a way, that when interpreting

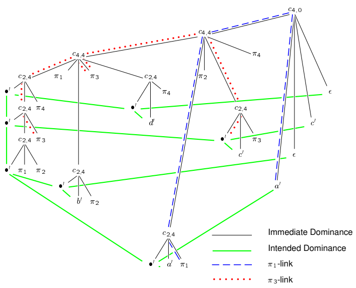
$\blacktriangleleft^\star \in R^I$ as a tree order on the set of *"linguistic"* nodes and
$\lhd \in R^I$ as the precedence relation on the resulting structure,

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

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

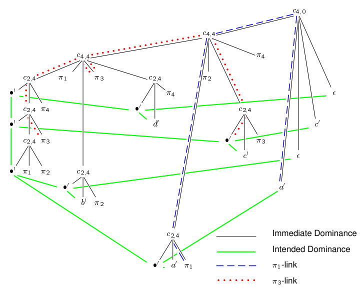
Reconstructing the Intended Structures





A few general facts

1. Our trees feature three families of labels: the "linguistic" symbols $\mathbf{L} = \mathcal{T}^L \cup \mathcal{N}^L$, where $\mathcal{T} = \Sigma_0$ of the underlying macro-grammar and $\mathcal{N} = \bigcup_{n \geq 1} \Sigma_n$; the "composition" symbols $\mathbf{C} = \bigcup_{n,k \geq 0} C_{n,k}$; and the "projection" symbols Π.
2. All non-terminal nodes in T^L are labeled by some $c \in \mathbf{C}$. 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 $\pi \in \Pi_i$.



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

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

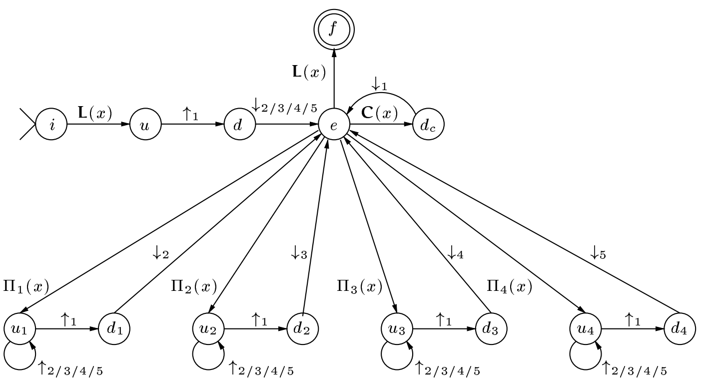
where for all states $q_i, q_j \in Q$ and nodes x, y in t

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

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 $W_{\blacktriangleleft}$

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

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

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

Reflexive transitive closure in MSO (on finite structures)

$$\mathsf{R}\text{-}closed(X) \equiv (\forall x, y)[x \in X \wedge \mathsf{R}(x, y) \ \rightarrow \ y \in X]$$
$$\mathsf{R}^*(x, y) \equiv (\forall X)[\mathsf{R}\text{-}closed(X) \wedge x \in X \ \rightarrow \ y \in X]$$

Walking language $\Rightarrow$ MSO-formula

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

$$\begin{aligned}
edge_n(x, y) = (\exists x_1, \ldots, x_{n-1})[&x \lhd x_1 \wedge \cdots \wedge x \lhd x_{n-1} \wedge \\
&x \lhd y \wedge x_1 \prec x_2 \wedge \cdots \wedge x_{n-1} \prec y \wedge \\
&(\forall w)[(x \lhd w \wedge w \not\approx x_1 \wedge \cdots \wedge w \not\approx x_{n-1} \wedge \\
&w \not\approx y) \rightarrow y \prec w]]
\end{aligned}$$

Intended Structures via MSO transduction

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

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

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

Top-Down Tree Transducer

A top-down tree transducer (TDTT) is a a tuple $\mathsf{T} = \langle Q, \Sigma, \Omega, q_0, P\rangle$ with states Q, ranked alphabets Σ and Ω (input and output), initial state q_0 and a finite set of productions P of the form

$$q(\sigma(x_1, \ldots, x_n)) \longrightarrow t$$

where $n \geq 0$, $\sigma \in \Sigma_n$ and $t \in T(\Omega \cup \Sigma(X) \cup Q)$.

The transition relation ($\overset{T}{\Longrightarrow}$) is defined as usual. The transduction realized by a top-down tree transducer T is then defined to be $\{(t_1, t_2) \in T(\Sigma) \times T(\Omega) \mid q_0(t_1) \overset{T}{\Longrightarrow}{}^* \ t_2\}$.

$$q_0(a(x_1, x_2)) \longrightarrow b(q_0(x_1), c, q_0(x_2))$$
$$q_0(p) \longrightarrow q$$

Complex Right-Hand-Sides

Let Σ and Ω be ranked alphabets and $n, m \geq 0$. The set of right hand sides $\mathrm{RHS}(\Sigma, \Omega, n, m)$ over Σ and Ω with n variables and m parameters is the smallest set $rhs \subseteq T(\Sigma \cup \Omega, X_n \cup Y_m)$ such that

1. $Y_m \subseteq rhs$
2. For $\omega \in \Omega_k$ with $k \geq 0$ and $\varphi_1, \ldots, \varphi_k \in rhs$, $\omega(\varphi_1, \ldots, \varphi_k) \in rhs$
3. For $q \in Q_{k+1}$ with $k \geq 0$, $x_i \in X_n$ and $\varphi_1, \ldots, \varphi_k \in rhs$, $q(x_i, \varphi_1, \ldots, \varphi_k) \in rhs$

Macro Tree Transducer

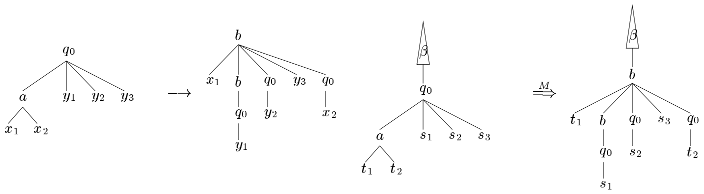
A macro tree transducer (MTT) is a five-tuple $M = \langle Q, \Sigma, \Omega, q_0, P \rangle$ with Q a ranked alphabet of states, ranked alphabets Σ and Ω (input and output), initial state q_0 of rank 1, and a finite set of productions P of the form

$$q(\sigma(x_1, \ldots, x_n), y_1, \ldots, y_m) \longrightarrow t$$

where $n, m \geq 0$, $q \in Q_{m+1}$, $\sigma \in \Sigma_n$ and $t \in \mathrm{RHS}(\Sigma, \Omega, n, m)$.

The productions $p \in P$ of M are used as term rewriting rules in the usual way. The transition relation of M is denoted by $\stackrel{M}{\Longrightarrow}$. The transduction realized by M is the function $\{(t_1, t_2) \in T(\Sigma) \times T(\Omega) \mid (q_0, t_1) \stackrel{M}{\Longrightarrow}^* t_2\}$.

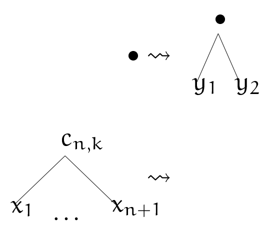
$$q_0(a(x_1, x_2), y_1, y_2, y_3) \longrightarrow b(x_1, b(q_0(y_1)), q_0(y_2), y_3, q_0(x_2))$$



Constructing an MTT

$$\sigma \rightsquigarrow \sigma$$

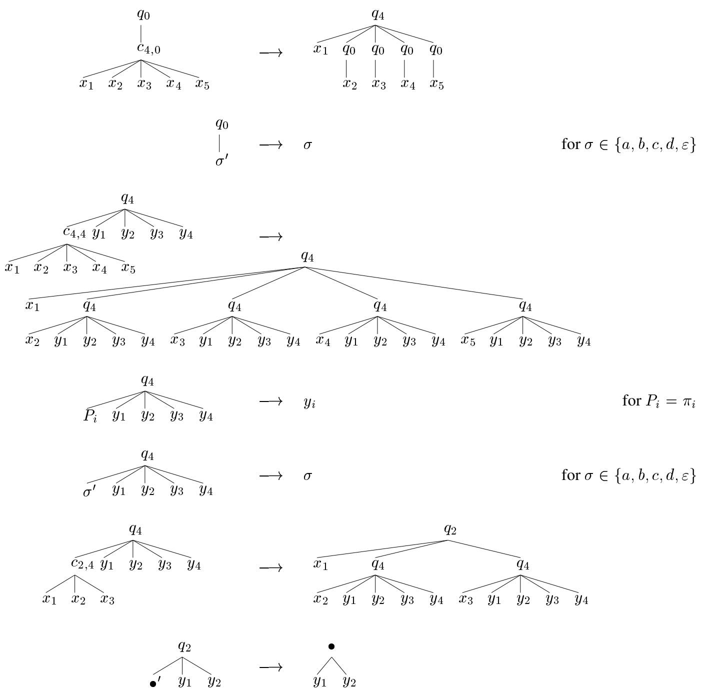
$$\pi_i \rightsquigarrow y_i$$



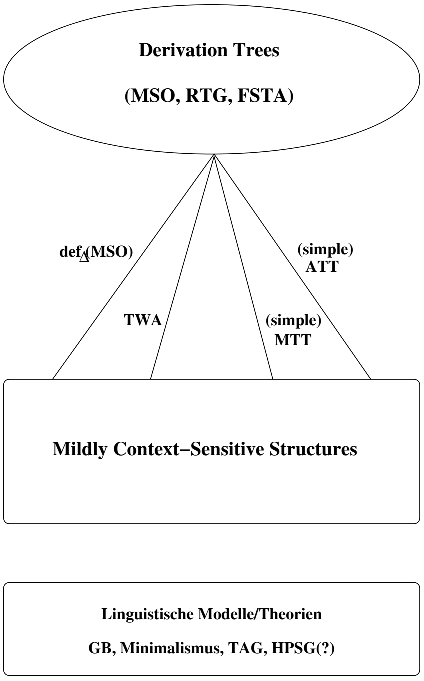
$$q_k(\underbrace{c_{n,k}(x_1, \ldots, x_{n+1}), y_1, \ldots, y_k}_{k+1}) \longrightarrow$$

$$q_n(\underbrace{x_1, q_k(x_2, y_1, \ldots, y_k), \ldots, q_k(x_{n+1}, y_1, \ldots, y_k)}_{n+1})$$

The Example MTT for $\mathcal{G}'$



Conclusion



Outlook

Weak Equivalence of Mildly Context-Sensitive Formalisms:

STR(HR): languages generated by string generating hyperedge replacement grammars

OUT(DTWT): output languages of deterministic tree-walking tree-to-string transducers

yDT_{fc}(REGT): yields of images of the regular tree languages under deterministic finite-copying top-down tree transductions

MCFL: languages generated by multiple context-free grammars

MCTAL: languages generated by multi-component tree adjoining grammars

LCFRL: languages generated by linear context-free rewriting systems

LUSCL: languages generated by local unordered scattered context grammars

Conjecture: "Simple" (linear, non-inverting) CFTGs followed by a "simple" macro tree transduction are weakly equivalent to the formalisms given above.