

# Capturing TAGs Model-Theoretically

Uwe Mönnich and Frank Morawietz

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

<http://tcl.sfs.uni-tuebingen.de/>

Seminar für Sprachwissenschaft

Arbeitsbereich Theoretische Computerlinguistik

Universität Tübingen

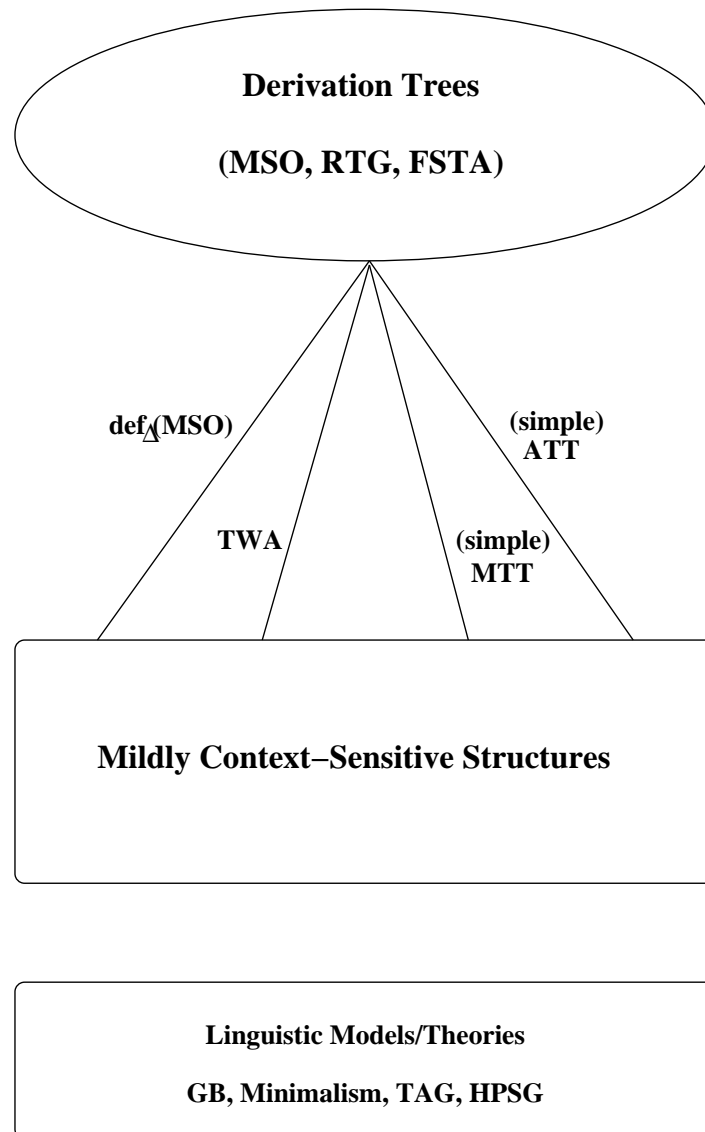


# Outline

- Motivation and Theoretical Background
- TAGs
- (Monadic) Context-Free Tree Grammars
- From (M)CFTGs to Regular Tree Grammars
- Interpreting the Intended Structures



# Overview



# General Structure of the Approach

## Translating TAGs to Monadic CFTGs

- ⇒ Lifting of Monadic Context-Free Tree Languages
- ⇒ Logical Characterization of the lifted Monadic CFTG-Languages
- ⇒ Retrieval of the Intended Structures via MSO-Definable Transductions



# General Structure of the Approach

## Translating TAGs to Monadic CFTGs

- ⇒ Lifting of Monadic Context-Free Tree Languages
- ⇒ Logical Characterization of the lifted Monadic CFTG-Languages
- ⇒ Retrieval of the Intended Structures via MSO-Definable Transductions
- trees as relational structures



# General Structure of the Approach

## Translating TAGs to Monadic CFTGs

- ⇒ Lifting of Monadic Context-Free Tree Languages
- ⇒ Logical Characterization of the lifted Monadic CFTG-Languages
- ⇒ Retrieval of the Intended Structures via MSO-Definable Transductions
  - trees as relational structures
  - trees as elements of a free algebra



# General Structure of the Approach

## Translating TAGs to Monadic CFTGs

- ⇒ Lifting of Monadic Context-Free Tree Languages
- ⇒ Logical Characterization of the lifted Monadic CFTG-Languages
- ⇒ Retrieval of the Intended Structures via MSO-Definable Transductions
  - trees as relational structures
  - trees as elements of a free algebra
  - trees as elements of a free clone (Lawvere theory)





# Tree Adjoining Grammars



# Tree Adjoining Grammars

$$\langle V_N, V_T, S, I, \mathcal{A} \rangle$$

$V_N$  is a finite set of nonterminals

$V_T$  is a finite set of terminals

$S \in V_N$  is the start symbol

$I$  is a finite set of initial trees

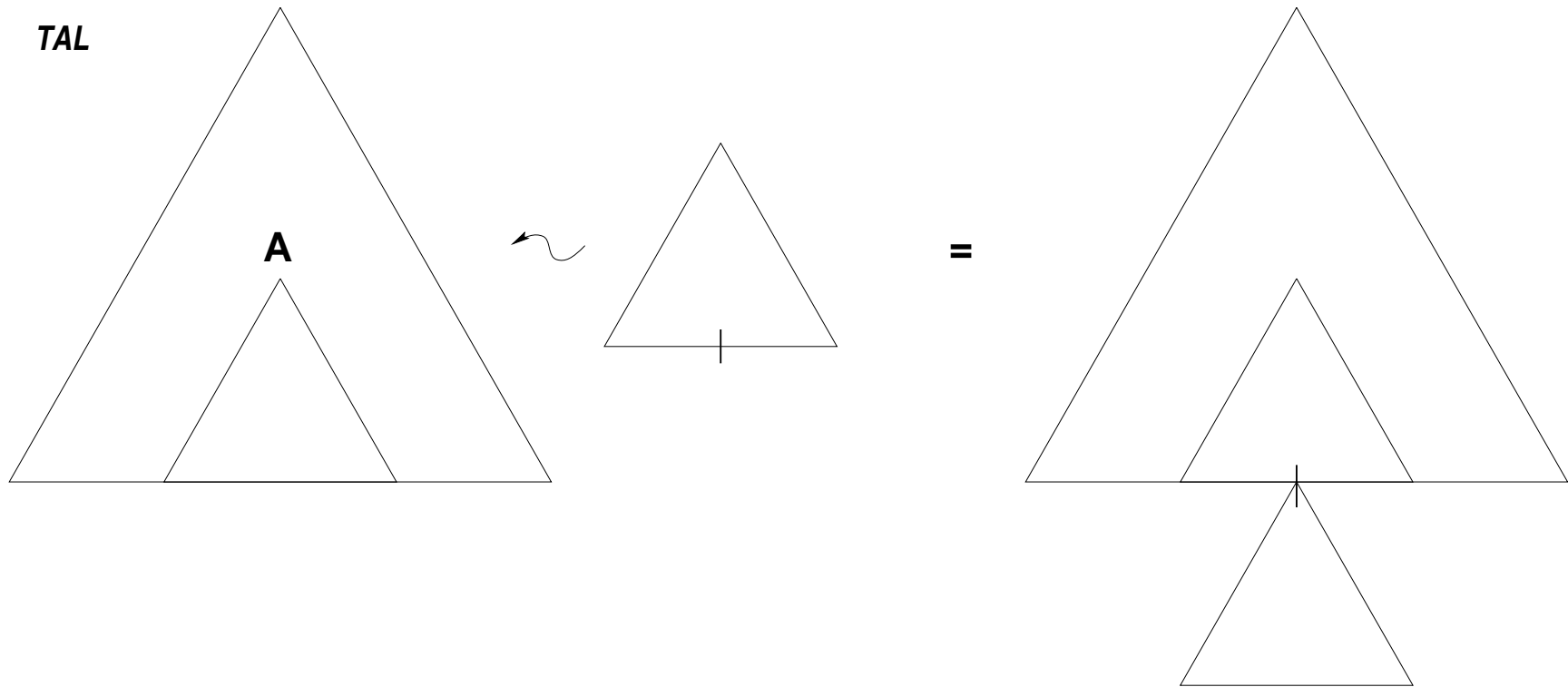
$\mathcal{A}$  is a finite set of auxiliary trees

Def'n from Joshi and Schabes 1997



# Derivation Steps in TAGs

*TAL*



# TAG for $a^n b^n c^n d^n$

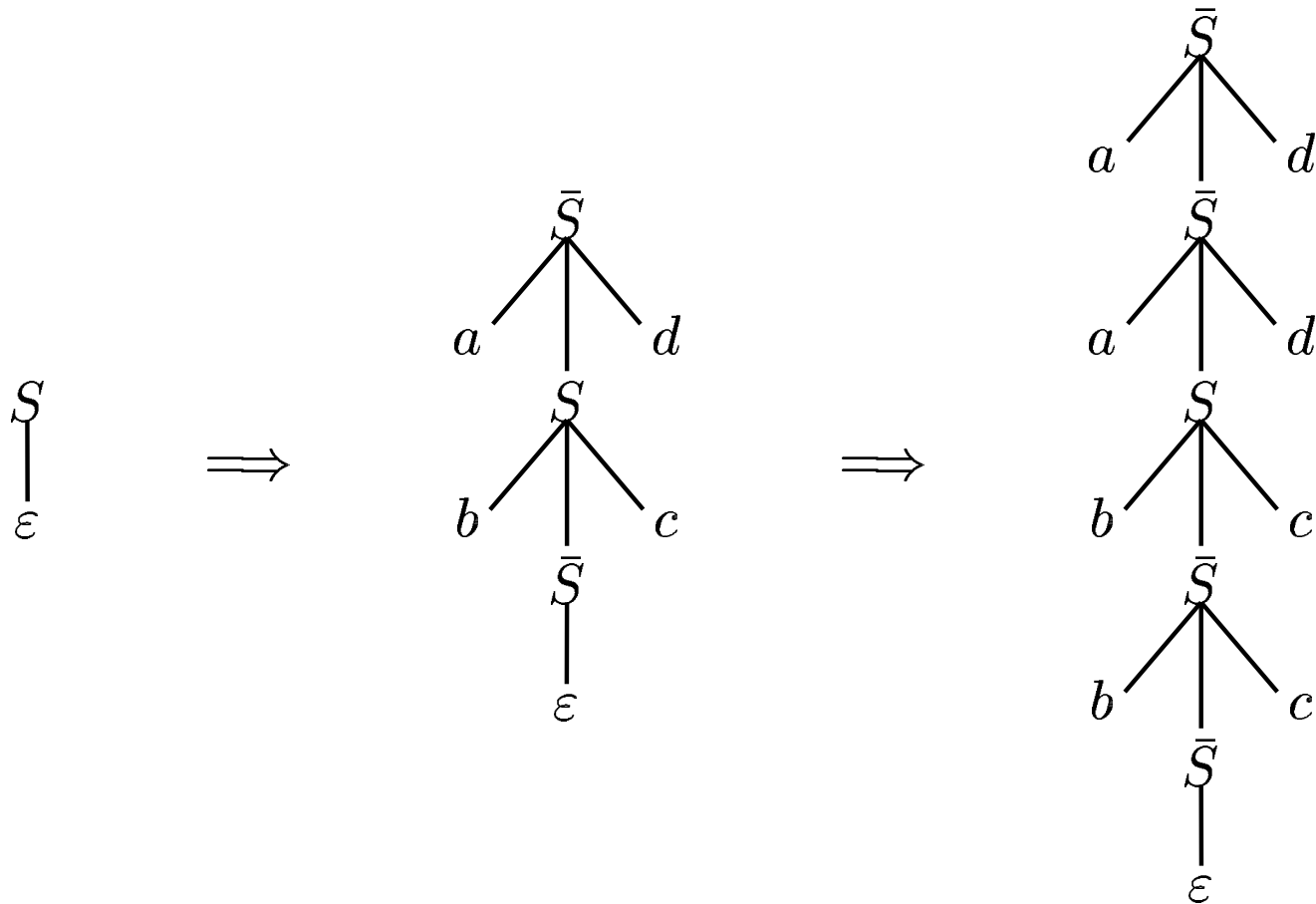
$\langle \{S\}, \{a, b, c, d\}, S, \{\alpha\}, \{\beta\} \rangle$

$\alpha =$  
$$\begin{array}{c} S \\ | \\ \varepsilon \end{array}$$

$\beta =$  
$$\begin{array}{c} \bar{S} \\ / \quad | \quad \backslash \\ a \quad \quad d \\ | \\ S \\ / \quad | \quad \backslash \\ b \quad \quad c \\ | \\ \bar{S} \end{array}$$



# An example derivation





# Tree Grammars

# Tree Grammars (CFTG & RTG)

A *context-free tree grammar*  $\Gamma$  is a 5-tuple  $\langle \Sigma, \mathbf{F}, S, \mathbf{X}, \mathbf{P} \rangle$ , with

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



# Tree Grammars (CFTG & RTG)

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

$$F(x_1, \dots, x_n) \longrightarrow t$$

with  $F \in \mathbf{F}$ ,  $x_1, \dots, x_n \in \mathbf{X}$ , and  $t$  a term over  $\Sigma \cup \mathbf{F} \cup \{x_1, \dots, x_n\}$ .

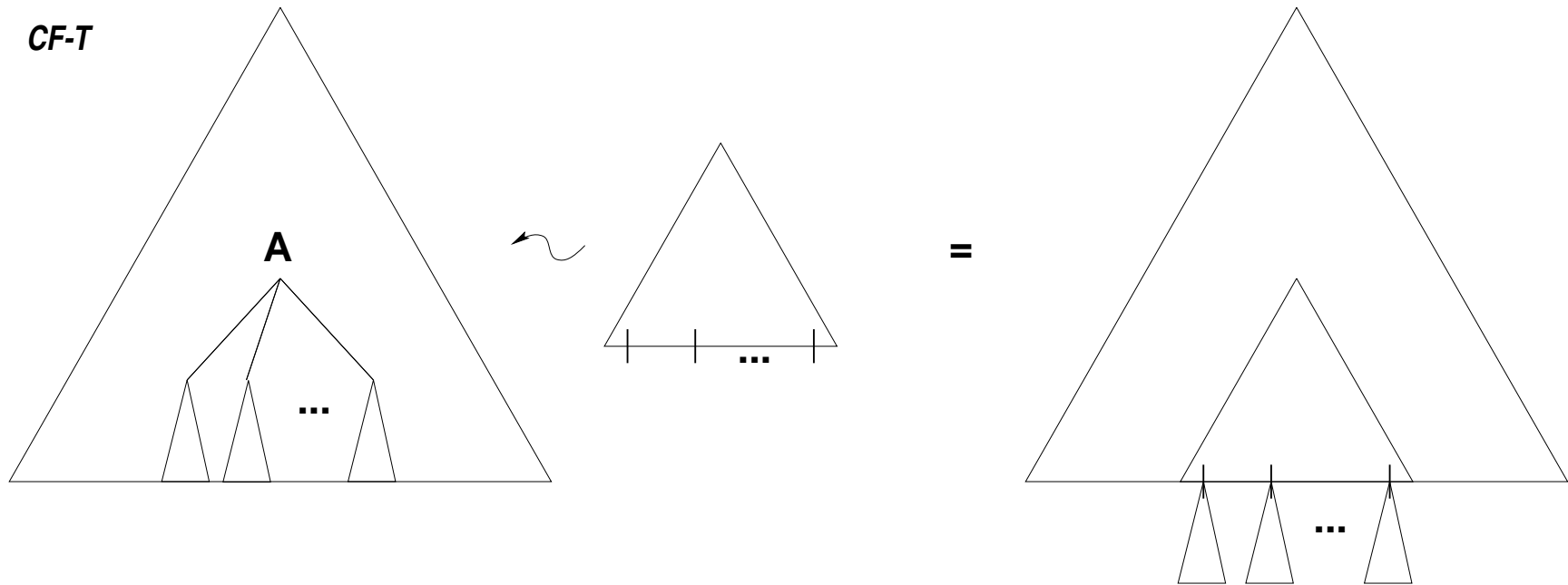
An application of a rule  $F(x_1, \dots, x_n) \longrightarrow t$  “rewrites” a subtree rooted in  $F$  as the tree  $t$  with its respective variables substituted by  $F$ ’s daughters.

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



# Derivation Steps in CFTGs

*CF-T*





# *Monadic CFTGs*

$$A \longrightarrow a$$

$$A \longrightarrow B(C)$$

$$A(x) \longrightarrow a(B_1, \dots, B_{i-1}, x_i, B_{i+1}, \dots, B_n)$$

$$A(x) \longrightarrow B_1(B_2(\dots B_n(x) \dots))$$



**Theorem** (Mönnich 1997, Fujiyoshi & Kasai 2000)

TAGs are equivalent to monadic CFTGs.



# Monadic CFTG $\mathcal{G}$ for $a^n b^n c^n d^n$

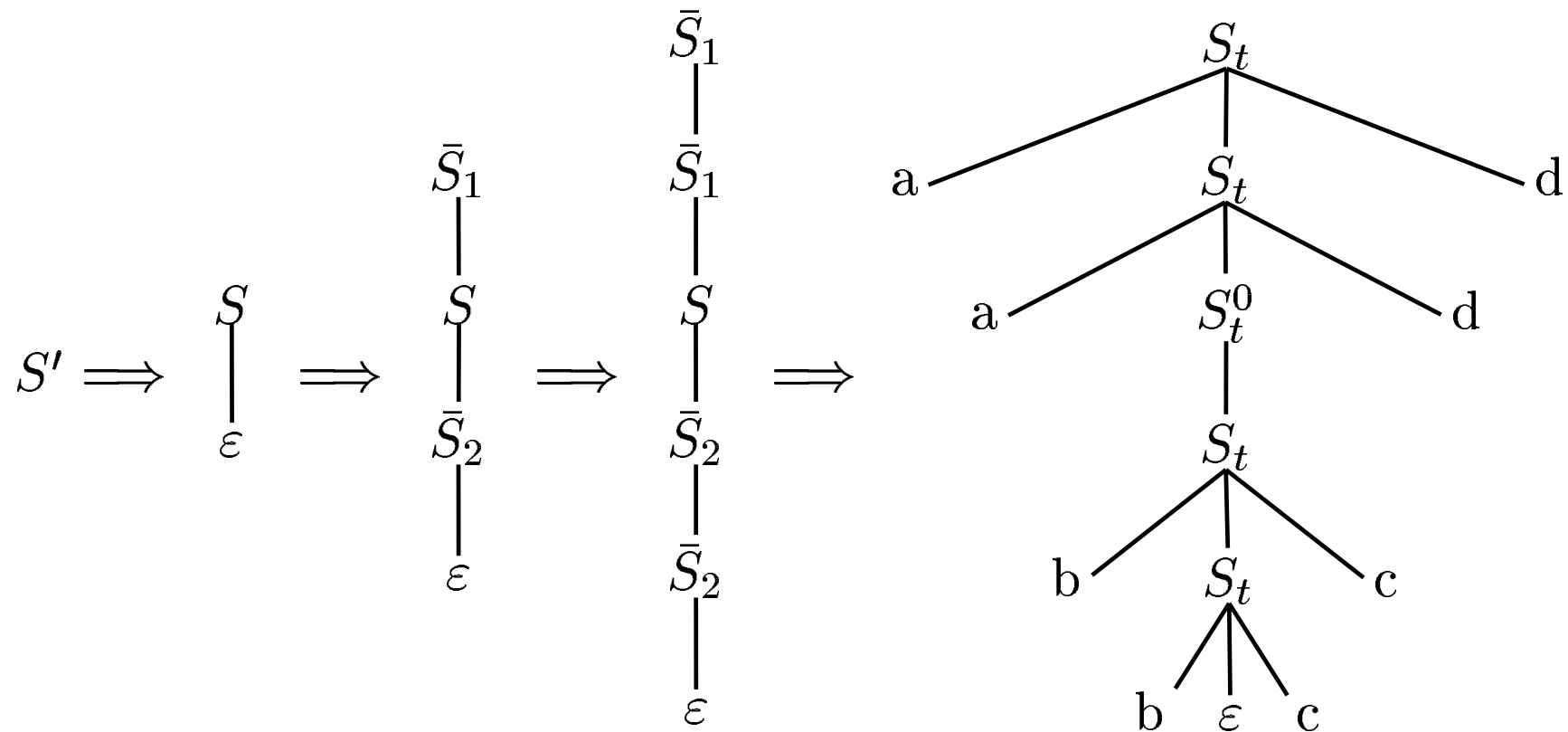
$$\mathcal{G} = \langle \{a, b, c, d, \varepsilon, S_t, S_t^0\}, \{S, S', \bar{a}, \bar{b}, \bar{c}, \bar{d}, \bar{S}_1, \bar{S}_2\}, S', \{x\}, \mathbf{P} \rangle$$

$$\begin{aligned} S' &\longrightarrow S(\varepsilon) \\ S(x) &\longrightarrow \bar{S}_1(S(\bar{S}_2(x))) \\ S(x) &\longrightarrow S_t^0(x) \\ \bar{S}_1(x) &\longrightarrow S_t(\bar{a}, x, \bar{d}) \\ \bar{S}_2(x) &\longrightarrow S_t(\bar{b}, x, \bar{c}) \end{aligned}$$

$$\begin{aligned} \bar{a} &\longrightarrow a \\ \bar{b} &\longrightarrow b \\ \bar{c} &\longrightarrow c \\ \bar{d} &\longrightarrow d \end{aligned}$$



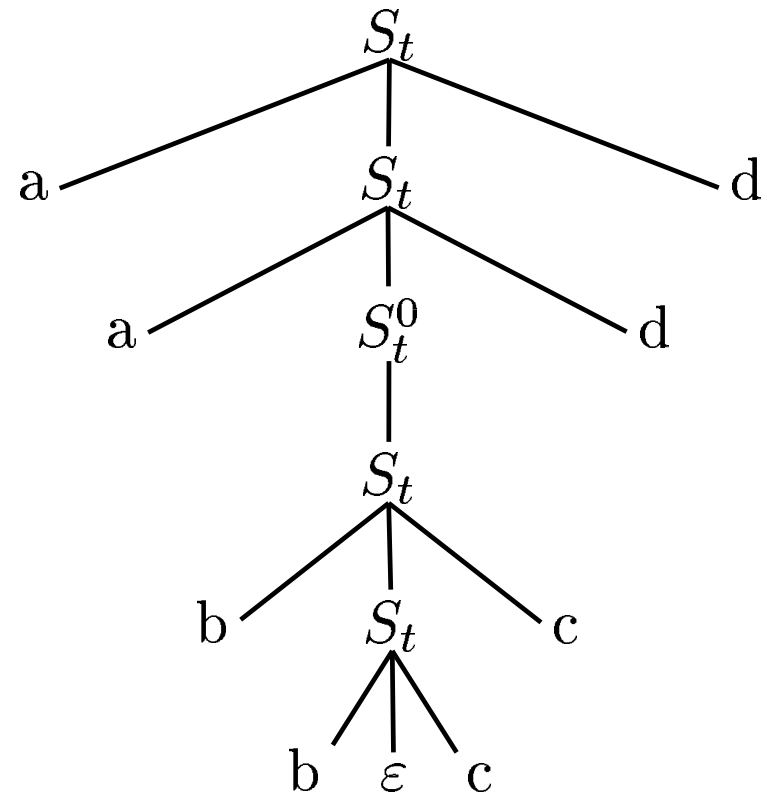
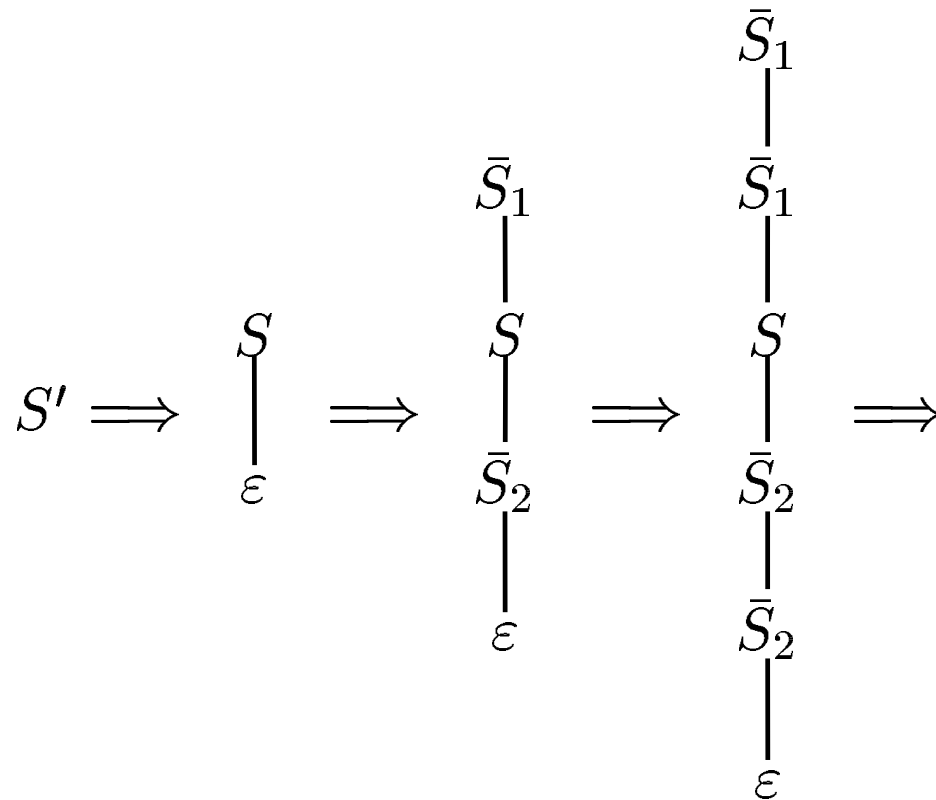
# An example derivation



$$S' \longrightarrow S(\epsilon)$$



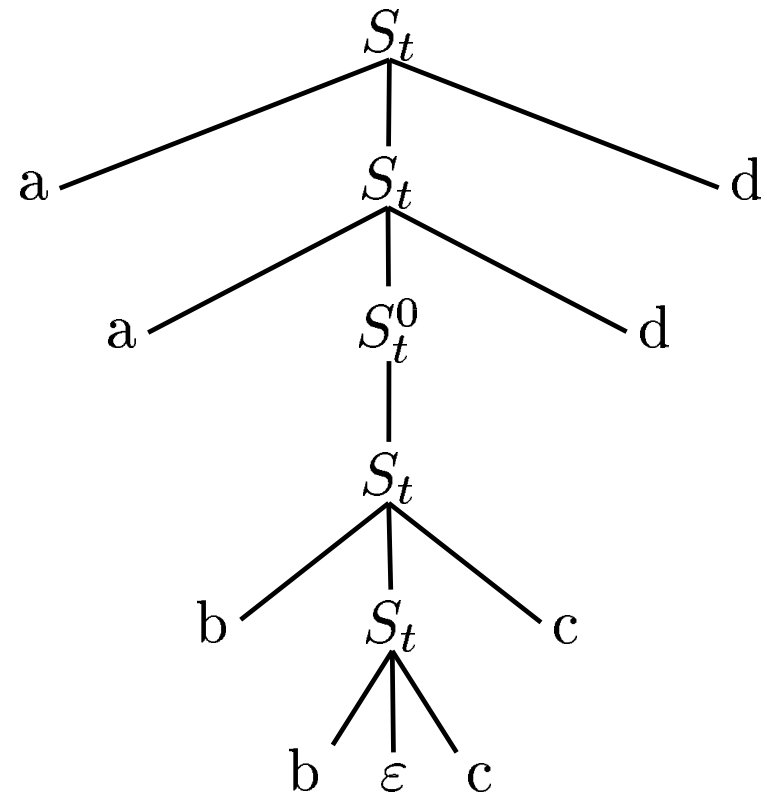
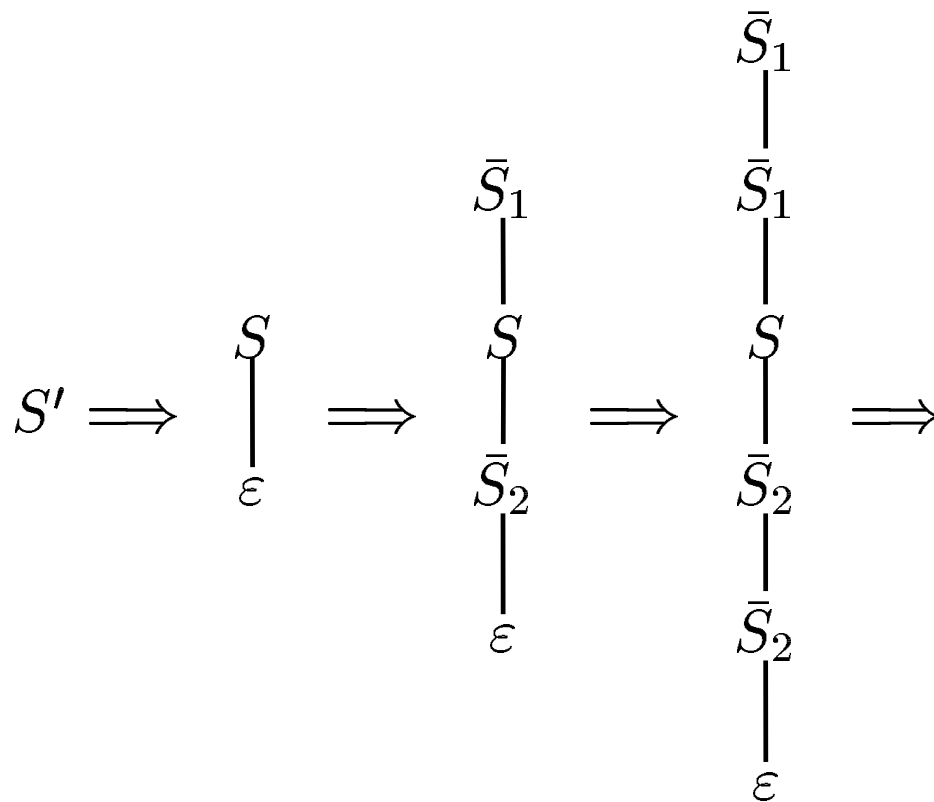
# An example derivation



$$S(x) \longrightarrow \bar{S}_1(S(\bar{S}_2(x)))$$



# An example derivation



$$S(x) \longrightarrow S_t^0(x)$$

$$\bar{S}_1(x) \longrightarrow S_t(\bar{a}, x, \bar{d})$$

$$\bar{S}_2(x) \longrightarrow S_t(\bar{b}, x, \bar{c})$$



# Regular Tree Grammars

$\Gamma = \langle \Sigma, F_0, S, P \rangle$  where

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

$$F_0 = \langle F_{\varepsilon,s} \mid s \in \mathcal{S} \rangle$$

$S \in F_0$  is the starting symbol

$P$  is a set of productions

The  $p \in P$  have the form

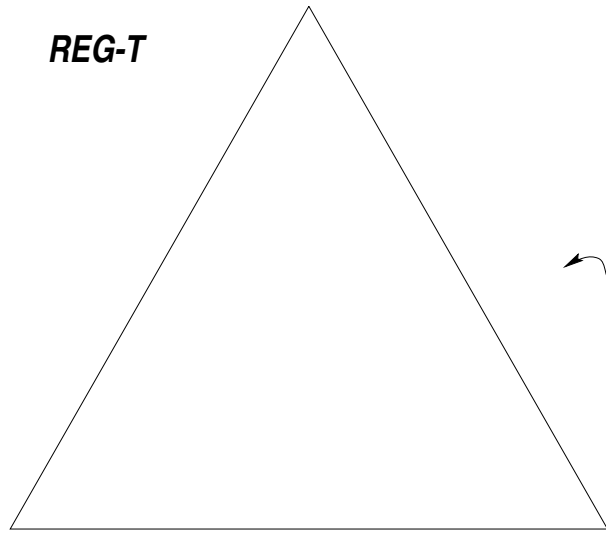
$$F \longrightarrow t$$

with  $F \in F_{\varepsilon,s}$ , and  $t$  a term over  $T(\Sigma \cup F_0)$ .

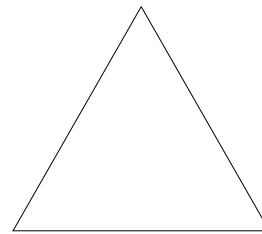


# Derivation Steps in RTGs

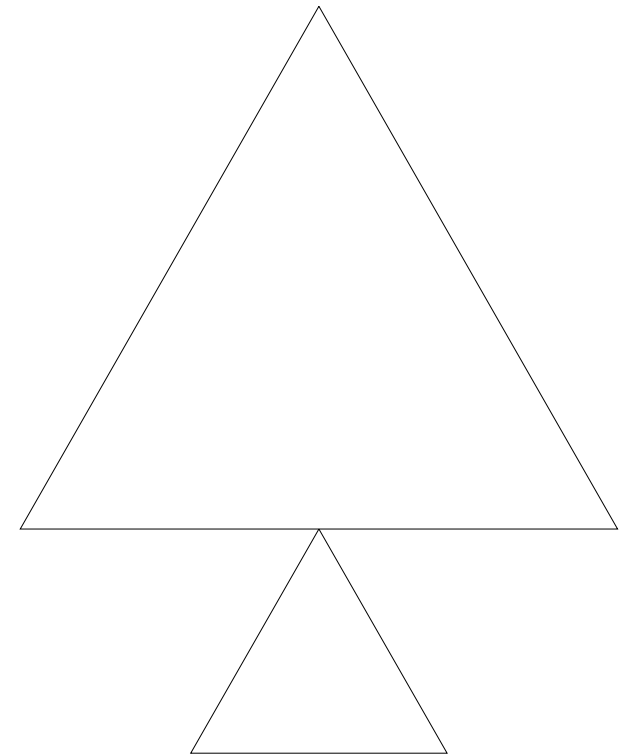
*REG-T*



**A**



=







# From (M)CFTGs to RTGs

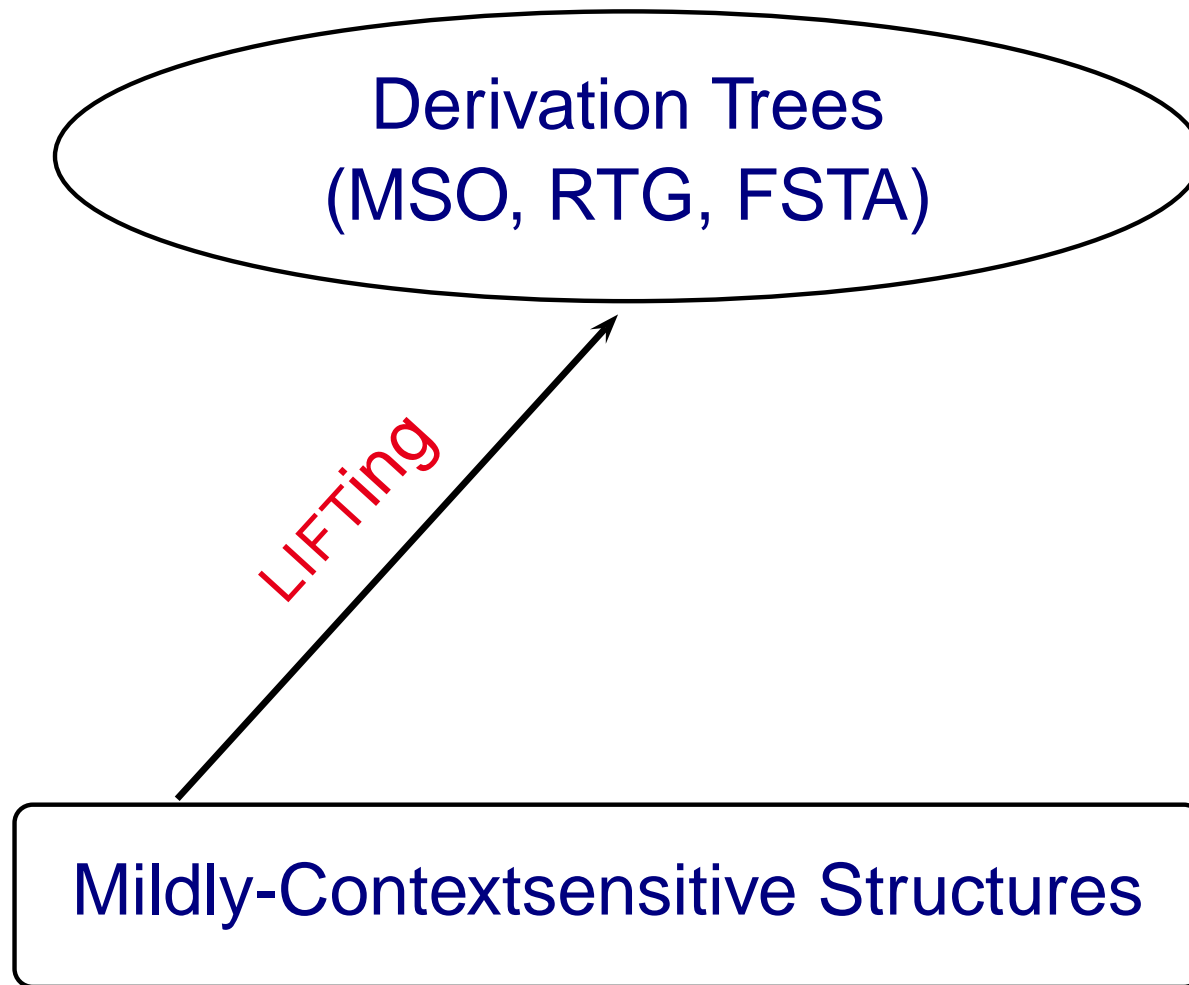
# Overview of the Two-Step Approach

Derivation Trees  
(MSO, RTG, FSTA)

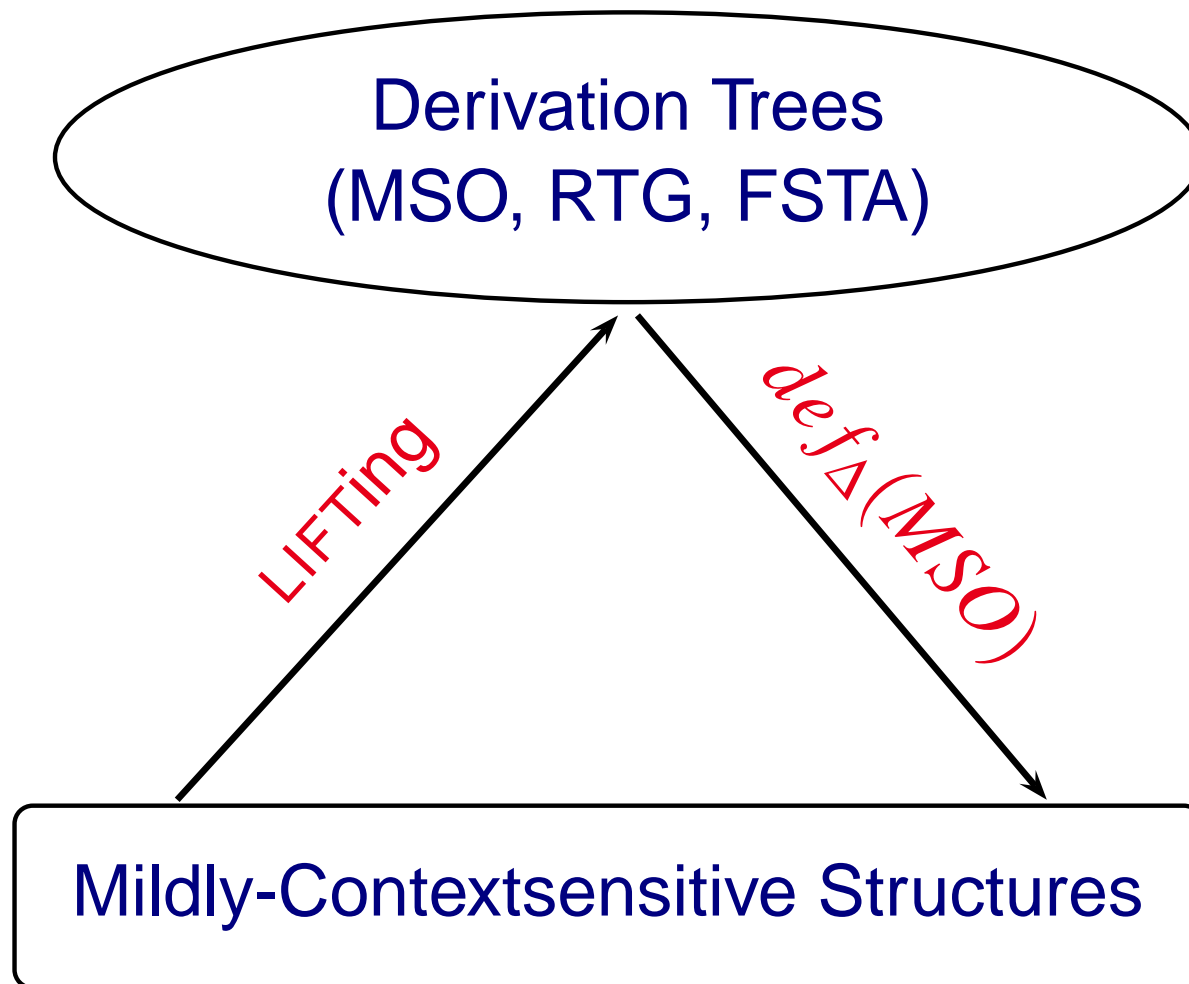
Mildly-Contextsensitive Structures



# Overview of the Two-Step Approach



# Overview of the Two-Step Approach



# From (M)CFTGs to RTGs: Intuition

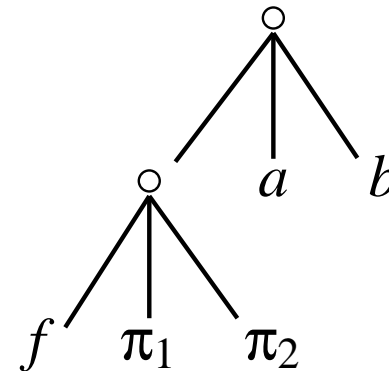
- simple primitive-recursive function
- control structure is made visible
- all function symbols become constants
- the resulting grammar is *regular*,
- and therefore expressible by an MSO formula
- and can be translated into an FSTA.



# From (M)CFTGs to RTGs: Intuition

- simple primitive-recursive function
- control structure is made visible
- all function symbols become constants
- the resulting grammar is *regular*,
- and therefore expressible by an MSO formula
- and can be translated into an FSTA.

$$f(a, b) \rightsquigarrow (f \circ (\pi_1, \pi_2)) \circ (a, b) \rightsquigarrow$$



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

$$\text{LIFT}_k^\Sigma(x_i) = \pi_i^k$$

$$\text{LIFT}_k^\Sigma(f) = c_{(0,k)}(f') \text{ for } f \in \Sigma_0$$

$$\text{LIFT}_k^\Sigma(f(t_1, \dots, t_n)) =$$

$$c_{(n,k)}(f', \text{LIFT}_k^\Sigma(t_1), \dots, \text{LIFT}_k^\Sigma(t_n))$$

for  $f \in \Sigma_n, n \geq 1$



# The Lifted Example Grammar $\mathcal{G}'$

$$S' \longrightarrow c_{(1,0)}(S, \varepsilon)$$

$$S \longrightarrow c_{(1,1)}(\bar{S}_1, c_{(1,1)}(S, c_{(1,1)}(\bar{S}_2, \pi_1^1)))$$

$$S \longrightarrow c_{(1,1)}(S_t^0, \pi_1^1)$$

$$\bar{S}_1 \longrightarrow c_{(3,1)}(S_t, a, \pi_1^1, d)$$

$$\bar{S}_2 \longrightarrow c_{(3,1)}(S_t, b, \pi_1^1, c)$$

$$S' \longrightarrow S(\varepsilon)$$

$$S(x) \longrightarrow \bar{S}_1(S(\bar{S}_2(x)))$$

$$S(x) \longrightarrow S_t^0(x)$$

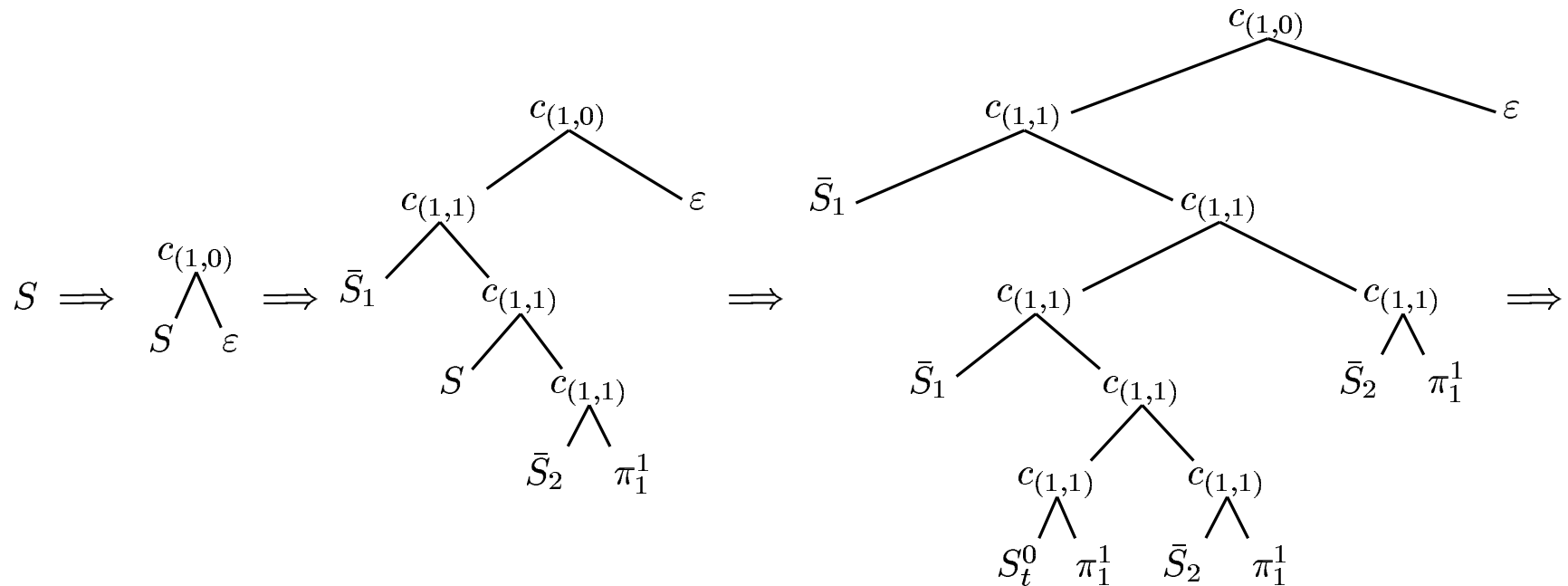
$$\bar{S}_1(x) \longrightarrow S_t(\bar{a}, x, \bar{d})$$

$$\bar{S}_2(x) \longrightarrow S_t(\bar{b}, x, \bar{c})$$





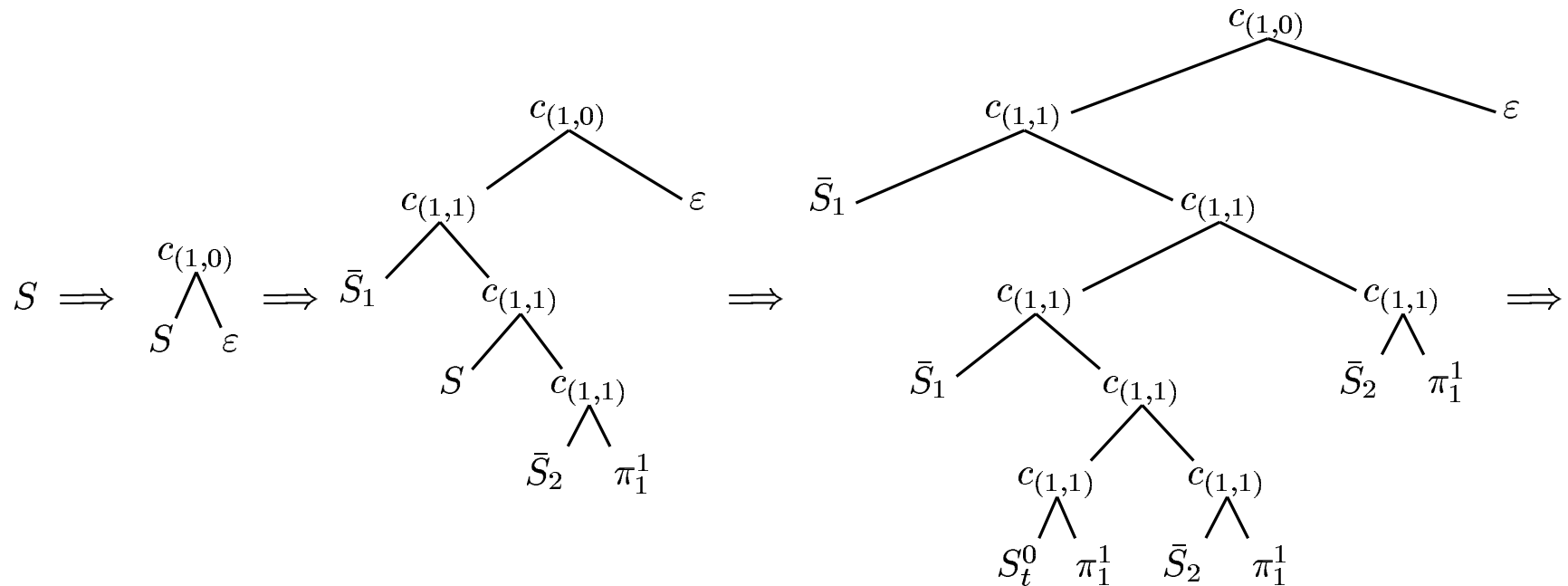
# An example derivation of $\mathcal{G}'$



$$S' \longrightarrow c_{(1,0)}(S, \varepsilon)$$



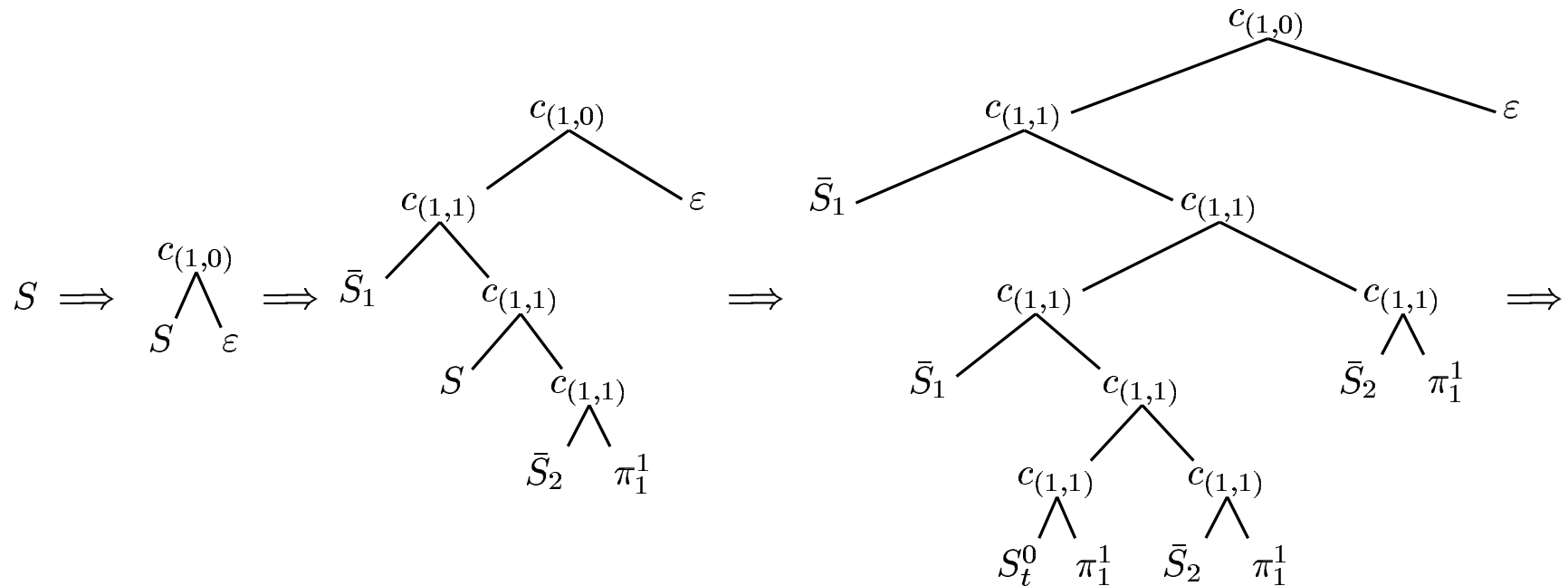
# An example derivation of $\mathcal{G}'$



$$S \longrightarrow c_{(1,1)}(\bar{S}_1, c_{(1,1)}(S, c_{(1,1)}(\bar{S}_2, \pi_1^1)))$$



# An example derivation of $\mathcal{G}'$

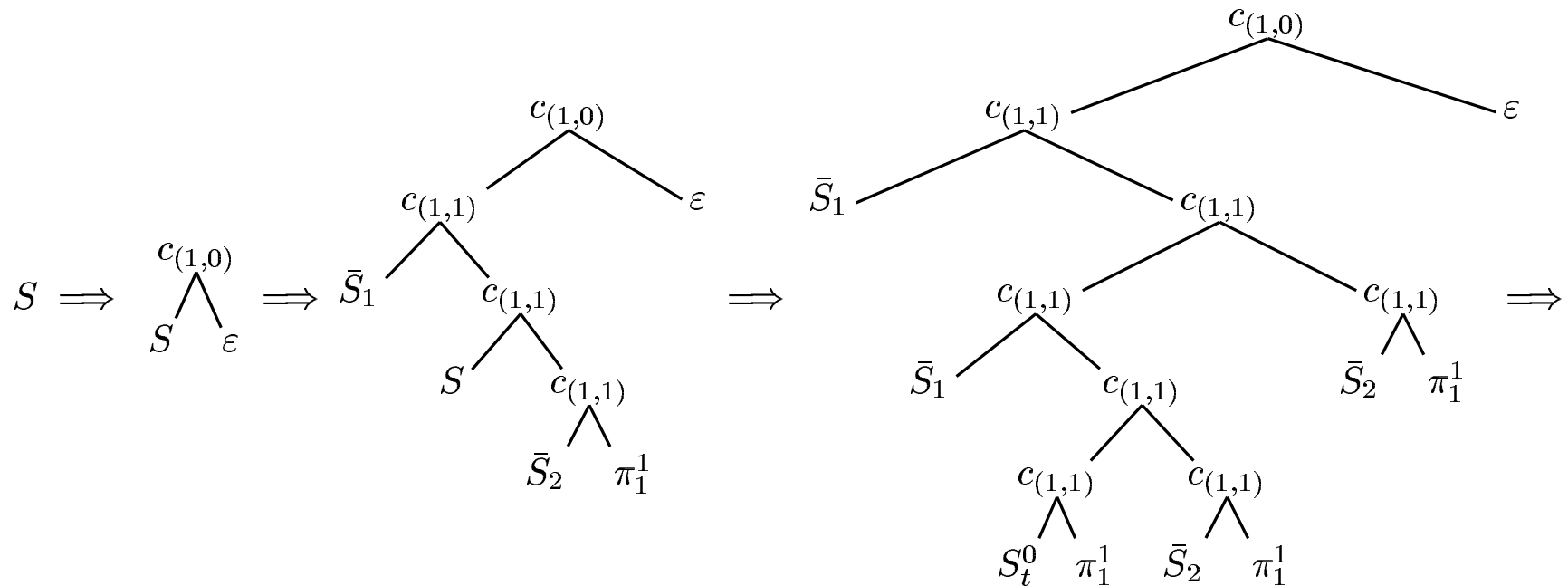


$$S \longrightarrow c_{(1,1)}(\bar{S}_1, c_{(1,1)}(S, c_{(1,1)}(\bar{S}_2, \pi_1^1)))$$

$$S \longrightarrow c_{(1,1)}(S_t^0, \pi_1^1)$$



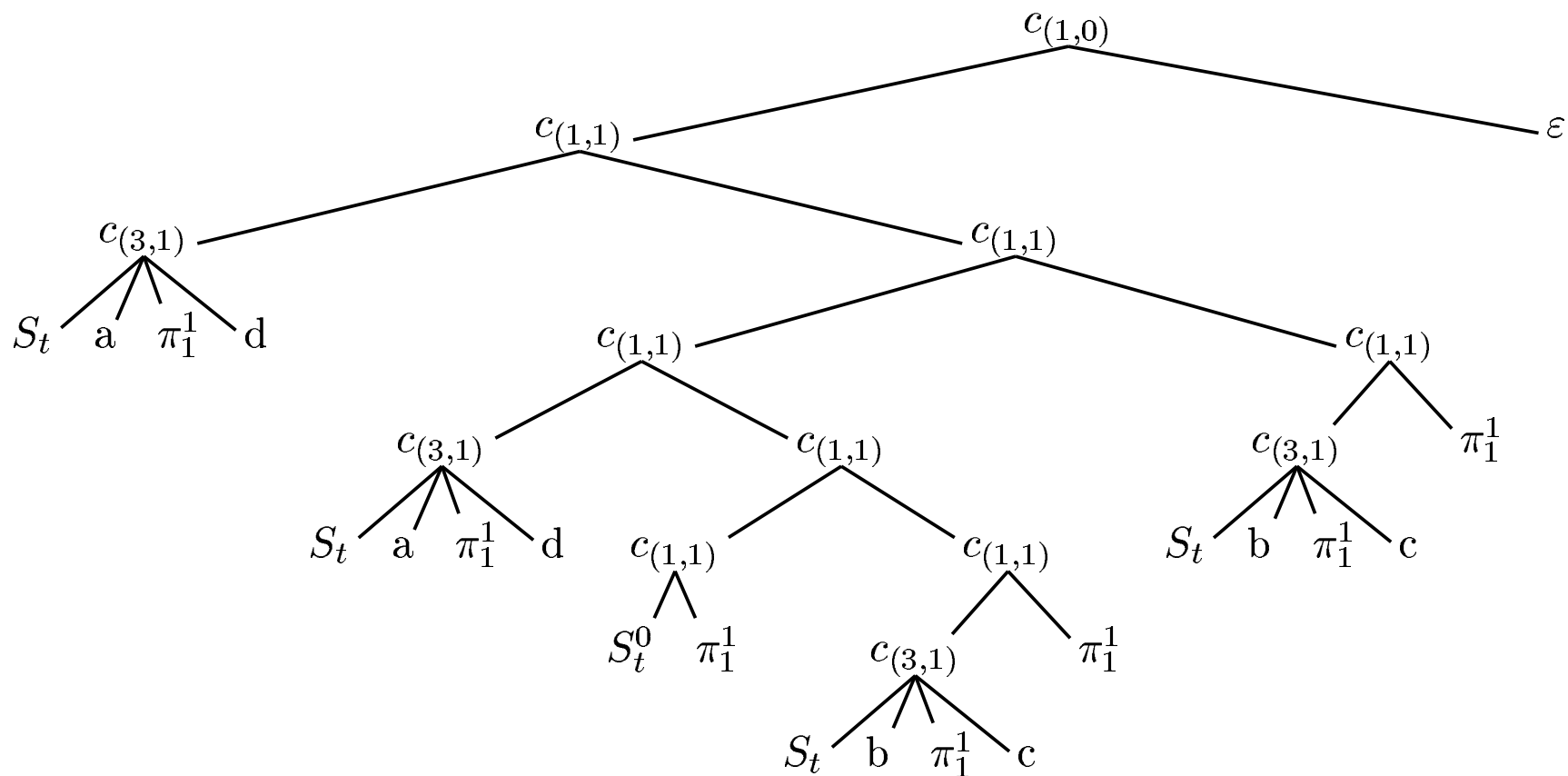
# An example derivation of $\mathcal{G}'$



$$\bar{S}_1 \longrightarrow c_{(3,1)}(S_t, a, \pi_1^1, d)$$

$$\bar{S}_2 \longrightarrow c_{(3,1)}(S_t, b, \pi_1^1, c)$$

# An example derivation of $\mathcal{G}'$ (cont'd)



# Coding RTGs in MSO Logic (Thomas 1997)

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

$$(\exists X_0, \dots, X_m) \left[ \bigwedge_{i \neq j} (\neg \exists y) [y \in X_i \wedge y \in X_j] \wedge \right.$$

$$(\forall x) [(\neg \exists y) [x \triangleleft y] \rightarrow x \in X_0] \wedge \quad \% \text{ Initial State}$$

$$\bigwedge_{1 \leq n \leq m} (\forall x_1, \dots, x_n, y) \left[ \bigvee_{\substack{(i_1, \dots, i_n, \sigma, j) \in \alpha \\ 1 \leq k \leq n}} x_k \in X_{i_k} \wedge y \triangleleft x_k \wedge y \in X_j \wedge y \in P_\sigma \right]$$

$$\bigvee_{i \in F} (\exists x \forall y) [x \triangleleft^* y \wedge x \in X_i] \quad \% \text{ Root}$$



# Interpreting the Intended Structures

# Reconstructing the Intended Structures

- The intended tree is contained in the lifted tree
- Homomorphism:

$$h(f') = f(x_1, \dots, x_n) \text{ for } f \in \Sigma_n$$

$$h(\pi_i^n) = x_i$$

$$h(c_{(n,k)}(t, t_1, \dots, t_n)) = h(t)[h(t_1), \dots, h(t_n)]$$

- Goal: Definition of a set of intended relations:

$$R^I = \{ \triangleleft^* \text{ (Dominance)}, \triangleleft, \triangleleft^+, \triangleleft \text{ (Precedence)}, \dots \}$$

⇒ MSO-Transduction





# MSO definable transductions

$$\mathcal{R} \rightsquigarrow Q$$

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

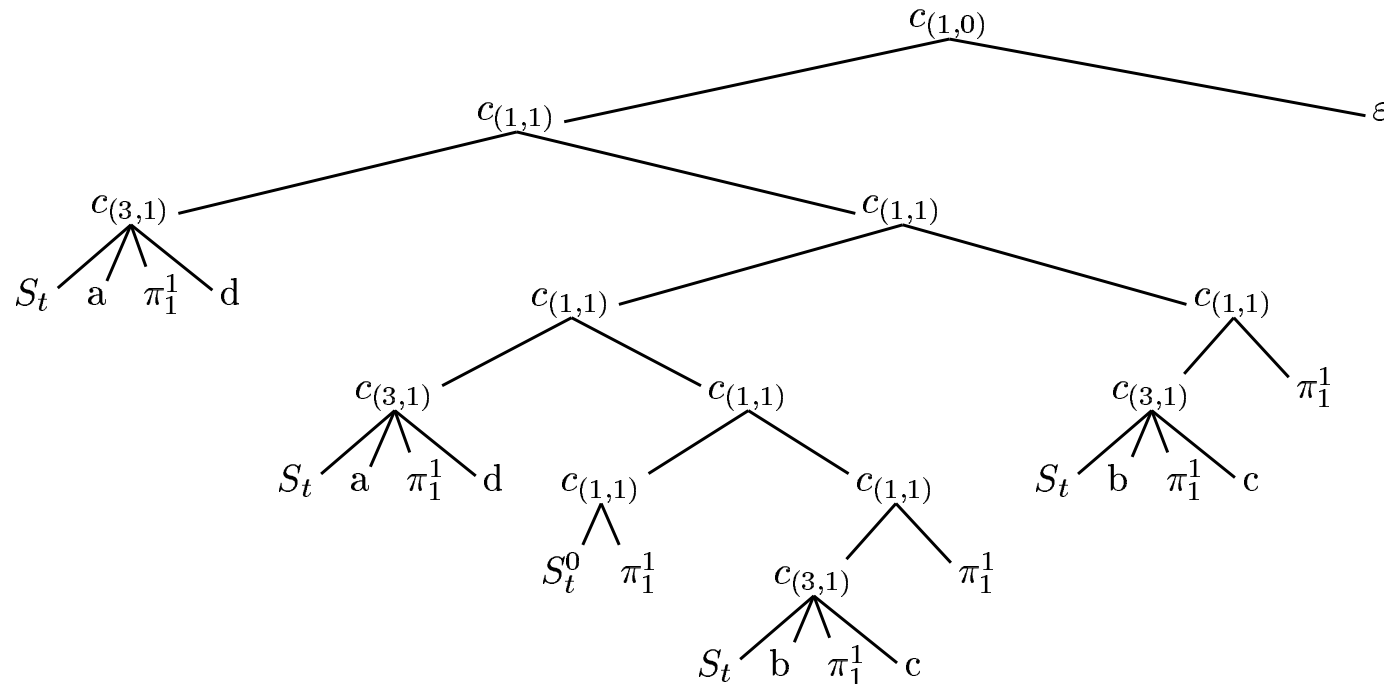
$\varphi$  the domain of the transduction

$\psi$  the resulting domain of  $Q$

$\theta_q$  the new relations



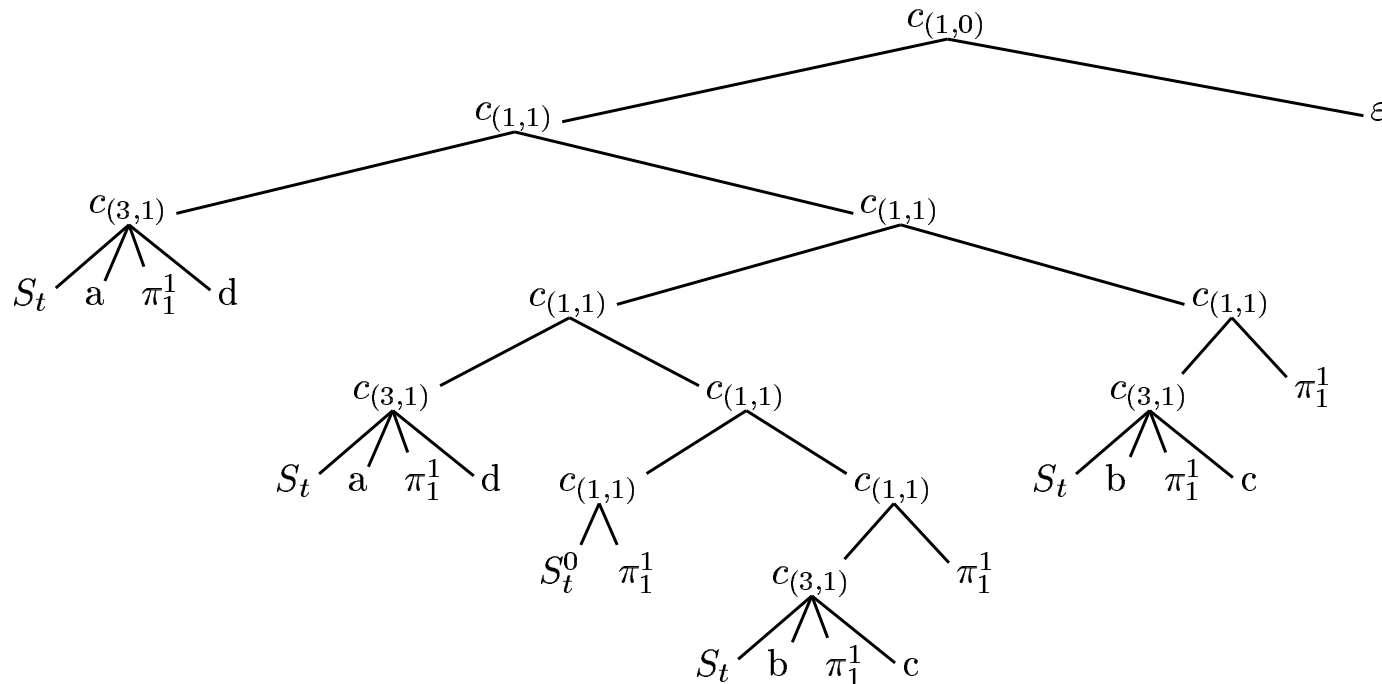
# A few general facts



1. Our trees feature three families of labels: the “linguistic” symbols  $L = \mathcal{T}^L \cup \mathcal{N}^L$ , where  $\mathcal{T} = \Sigma_0$  of the underlying (M)CFTG and  $\mathcal{N} = \bigcup_{n \geq 1} \Sigma_n$ ; the “composition” symbols  $C = \bigcup_{n,k \geq 0} c_{n,k}$ ; and the “projection” symbols  $\Pi$ .



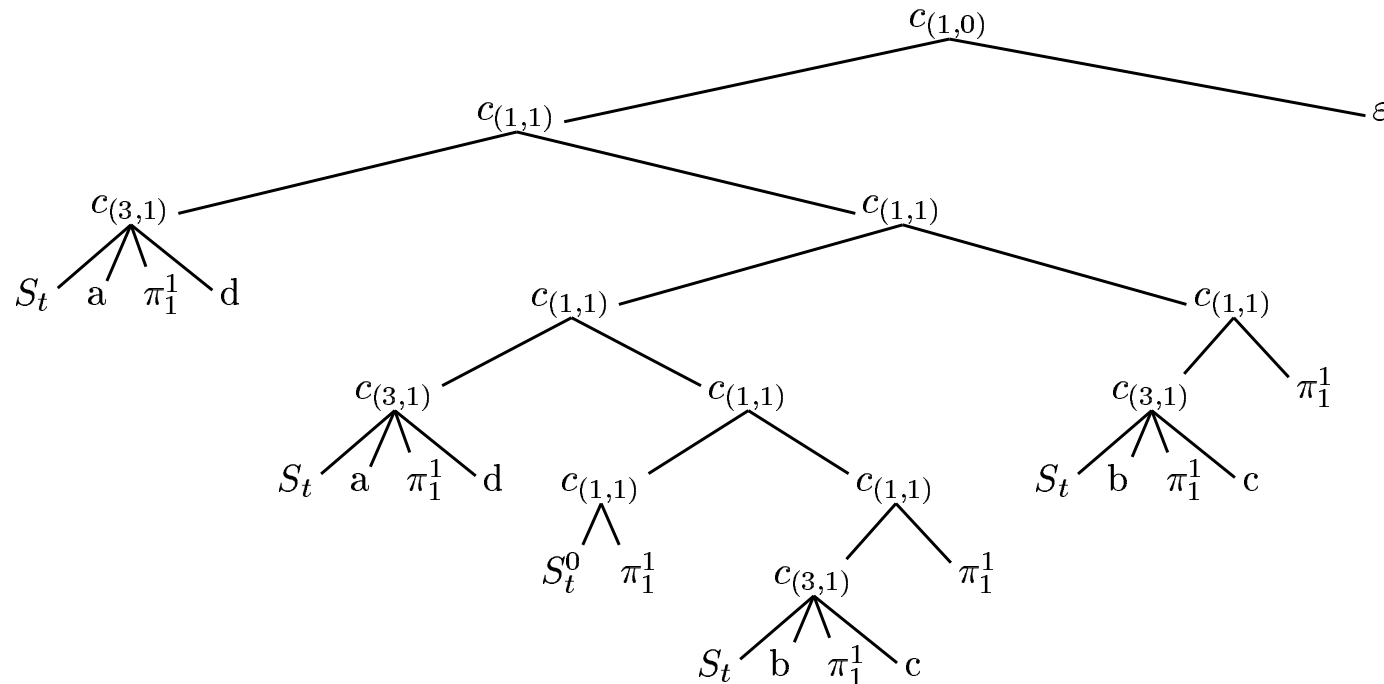
# A few general facts



2. All non-terminal nodes in  $T^L$  are labeled by some  $c \in C$ . No terminal node is labeled by some  $c$ .



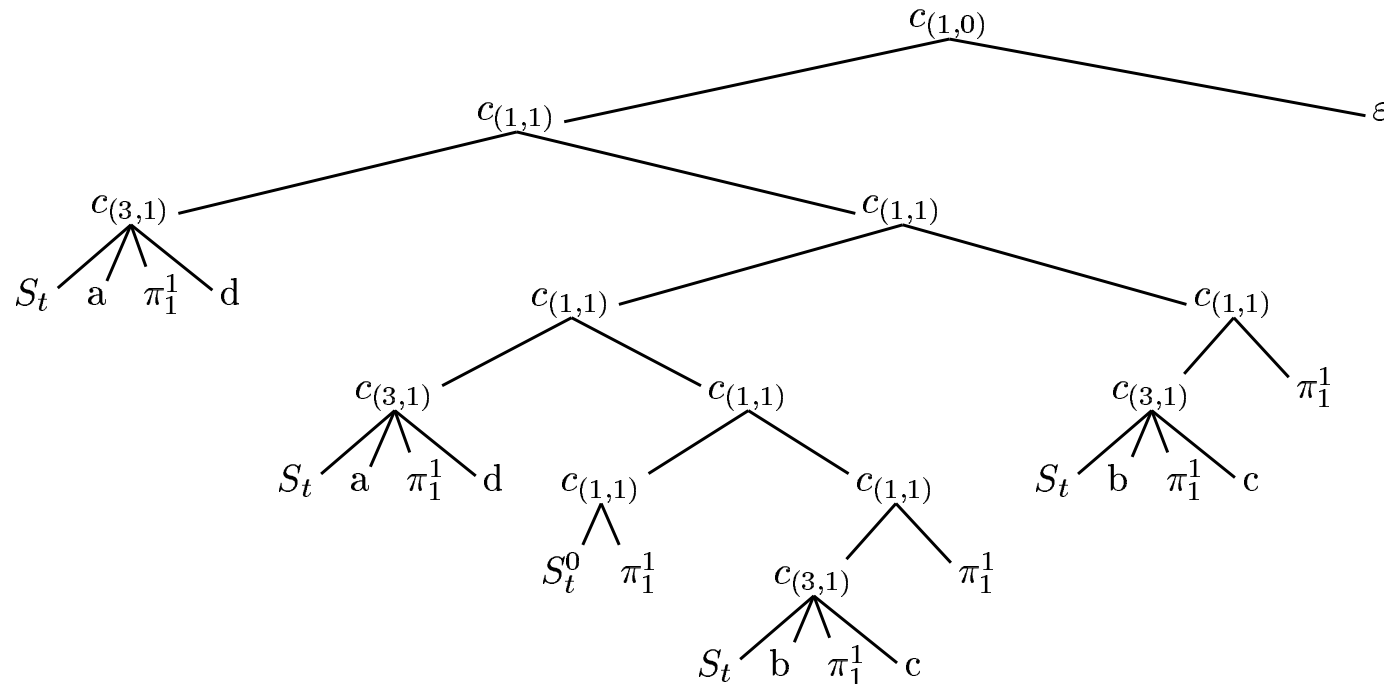
# A few general facts



3. The terminal nodes in  $T^L$  are either labeled by some “linguistic” symbol or by some “projection” symbol  $\pi \in \Pi_i$ .



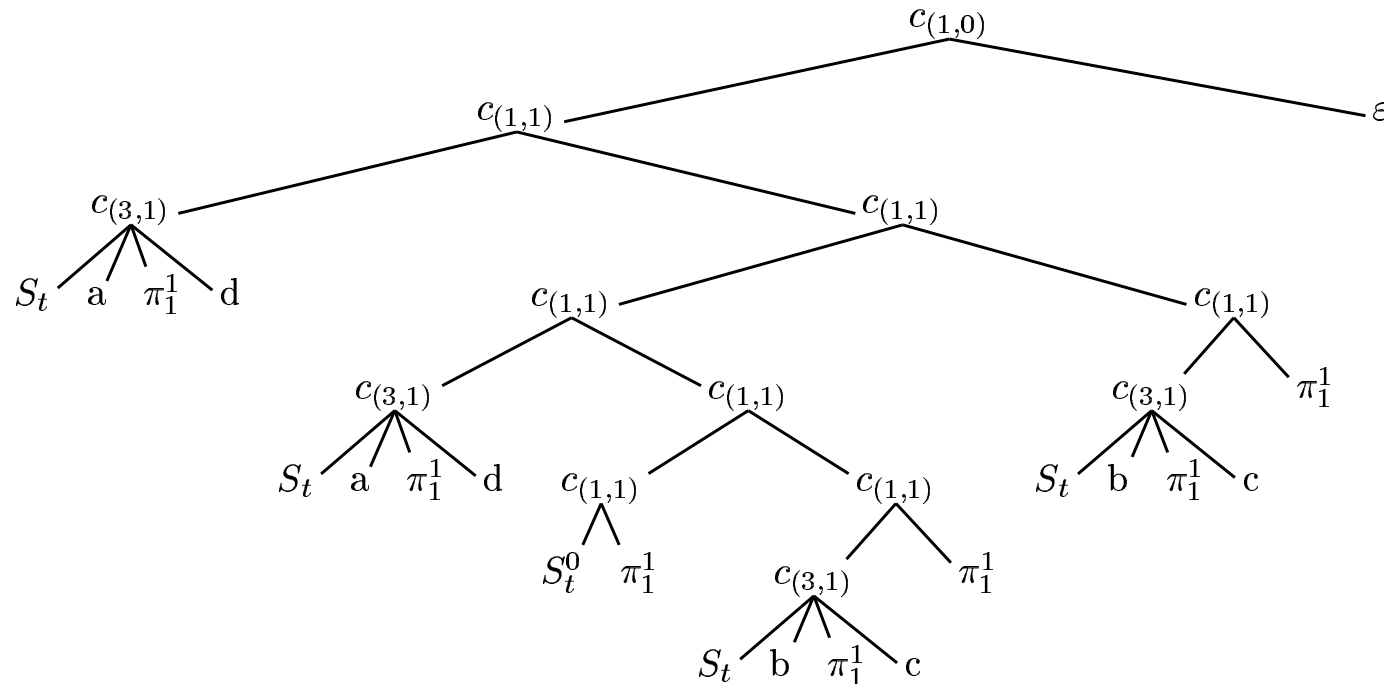
# A few general facts



- 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 of some  $c \in C$ .

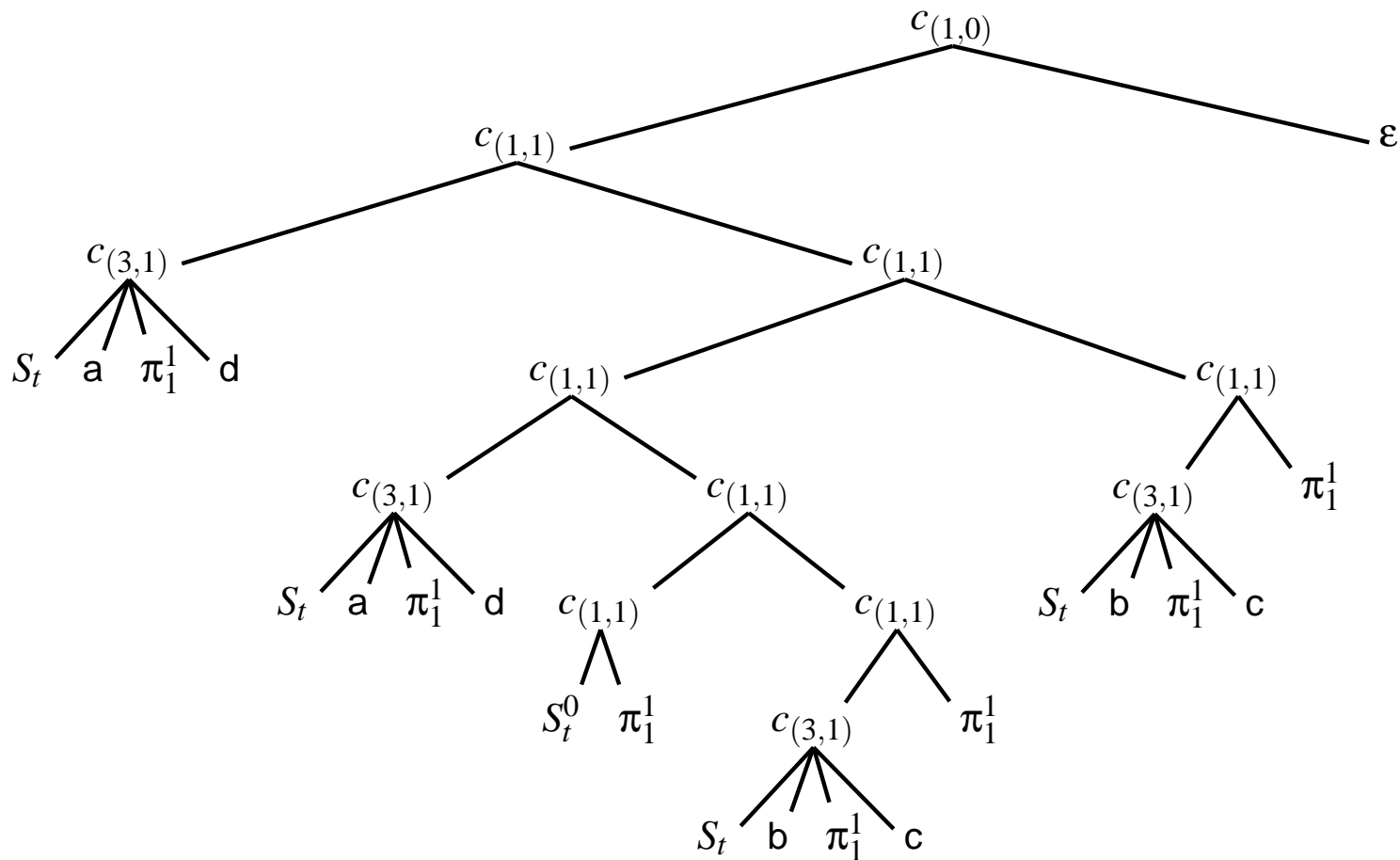


# A few general facts

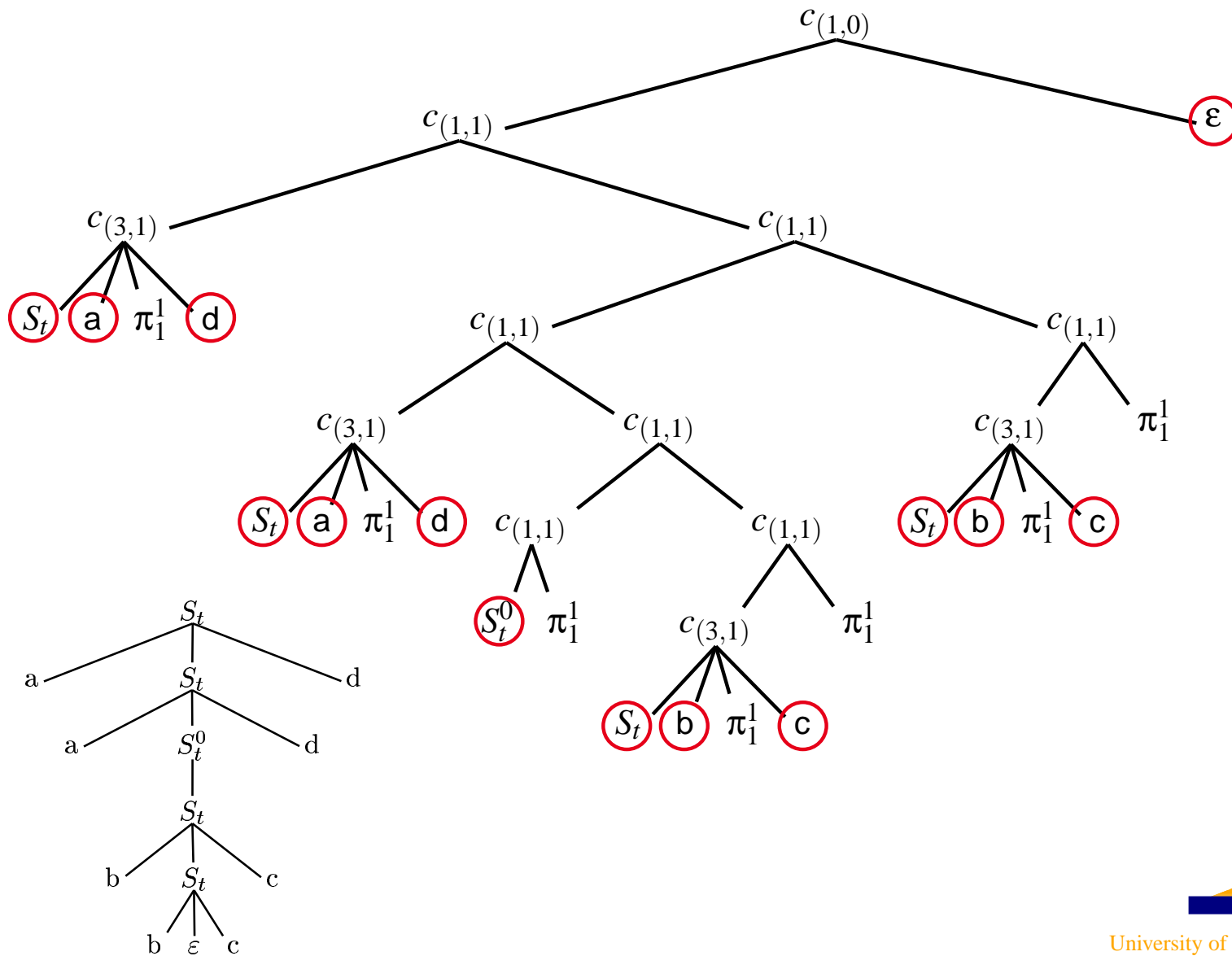


5. For any node  $p$  labeled with some “projection” symbol  $\pi \in \Pi_i$  in  $T^L$  there is a unique node  $n$  (labeled with some  $c \in C$  by (2.)) which properly dominates  $p$  and whose  $i$ -th sister (to the right) will eventually evaluate to the value of  $\pi$ . Moreover,  $n$  will be the first node properly dominating  $p$  which is on a left branch.

# Reconstructing the Intended Structures



# Reconstructing the Intended Structures





EBERHARD KARLS UNIVERSITÄT TÜBINGEN



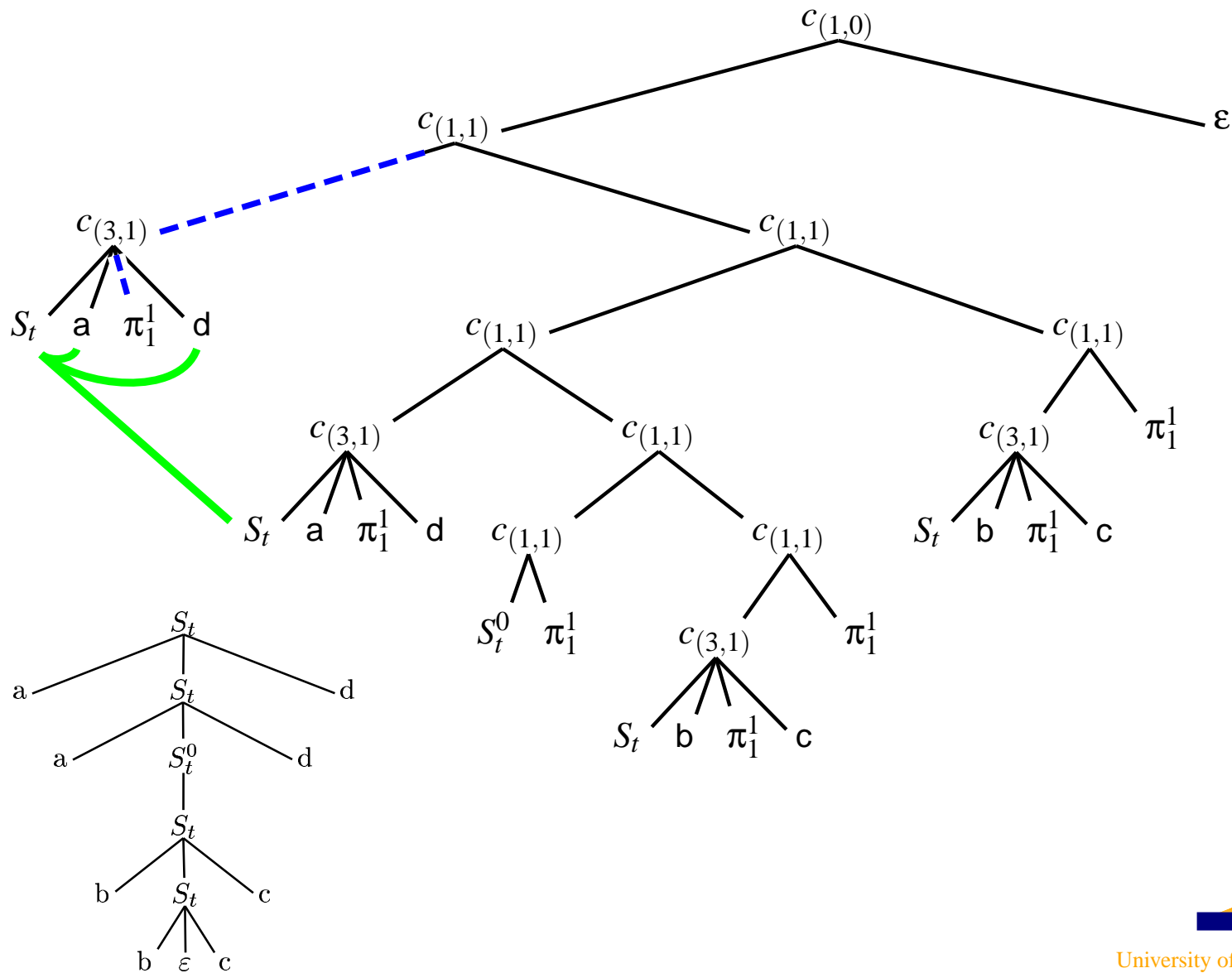
EBERHARD KARLS UNIVERSITÄT TÜBINGEN



EBERHARD KARLS UNIVERSITÄT TÜBINGEN



# Reconstructing the Intended Structures



EBERHARD KARLS UNIVERSITÄT TÜBINGEN



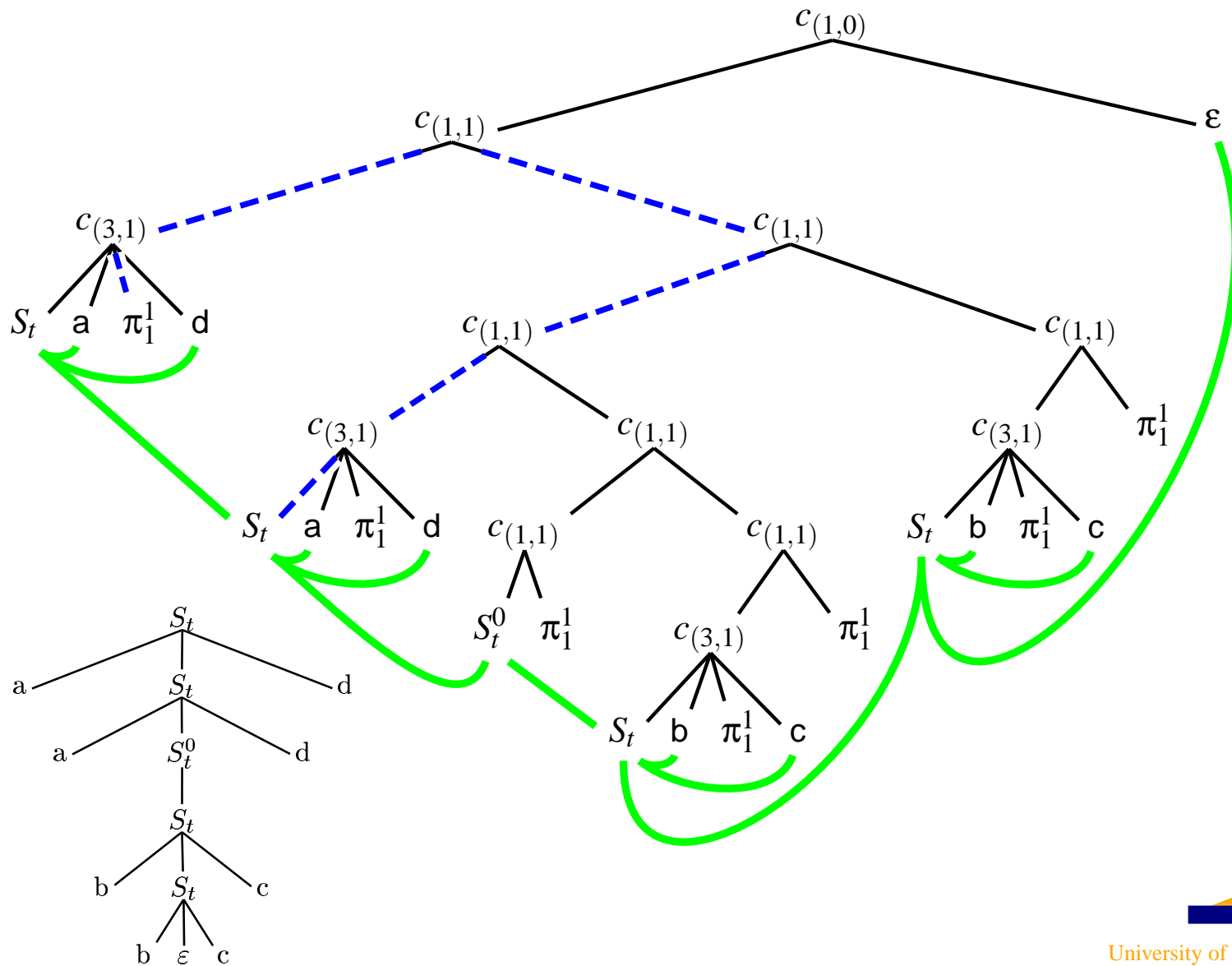
EBERHARD KARLS UNIVERSITÄT TÜBINGEN



EBERHARD KARLS UNIVERSITÄT TÜBINGEN

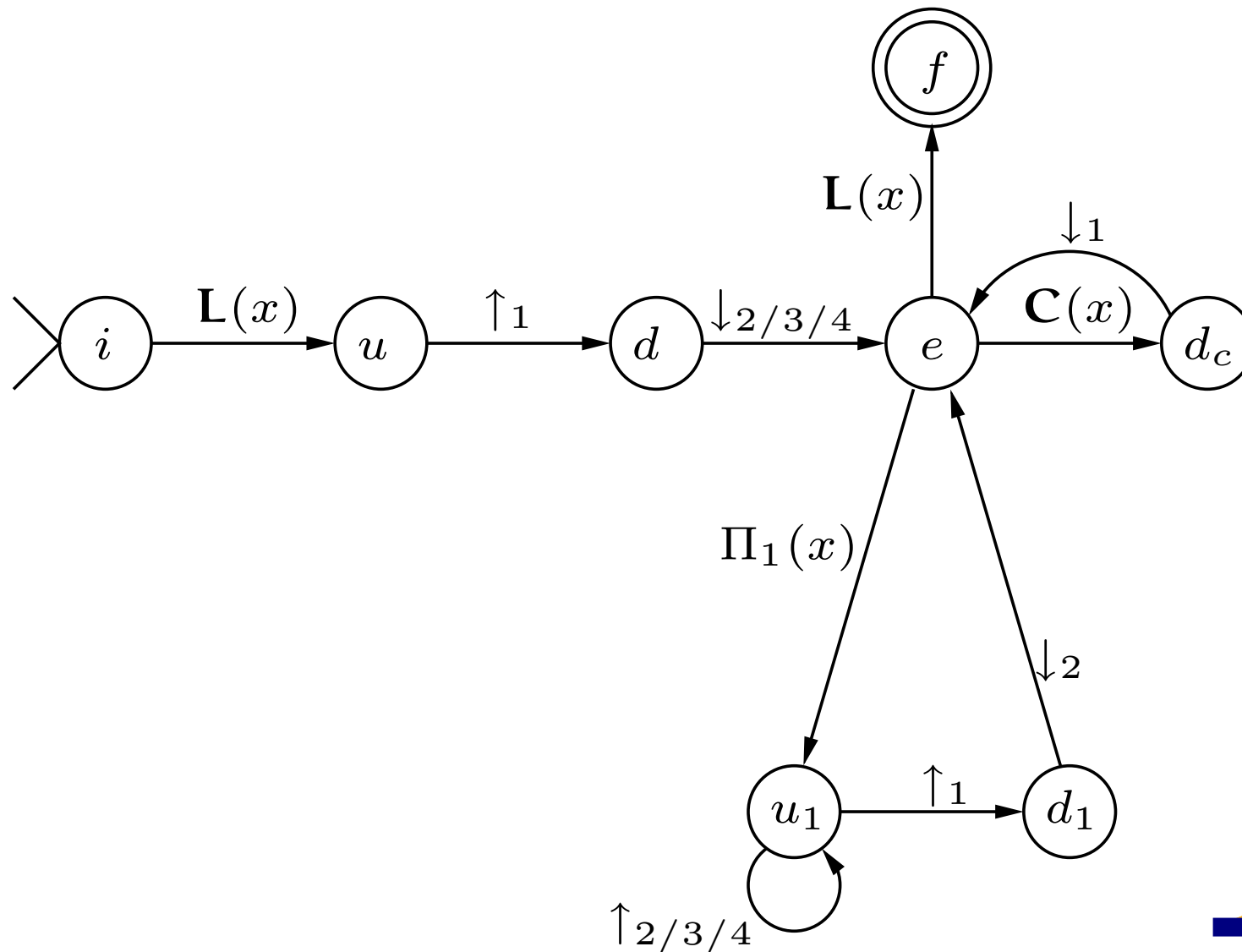


# Reconstructing the Intended Structures





# The tree-walking automaton $A_{\triangleleft}$



# The walking language $W_{\triangleleft}$

$$W_{\triangleleft} = \mathbf{L}(x) \cdot \uparrow_1 \cdot (\downarrow_2 \cup \downarrow_3 \cup \downarrow_4) \cdot \left( \bigcup_{1 \leq i \leq k} W_{\Pi_i} \cup W_{\mathbf{C}} \right)^* \cdot \mathbf{L}(x)$$

$$W_{\mathbf{C}} = \mathbf{C}(x) \cdot \downarrow_1$$

$$W_{\Pi_i} = \Pi_i(x) \cdot (\uparrow_2 \cup \uparrow_3 \cup \uparrow_4)^* \cdot \uparrow_1 \cdot \downarrow_{i+1}$$

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



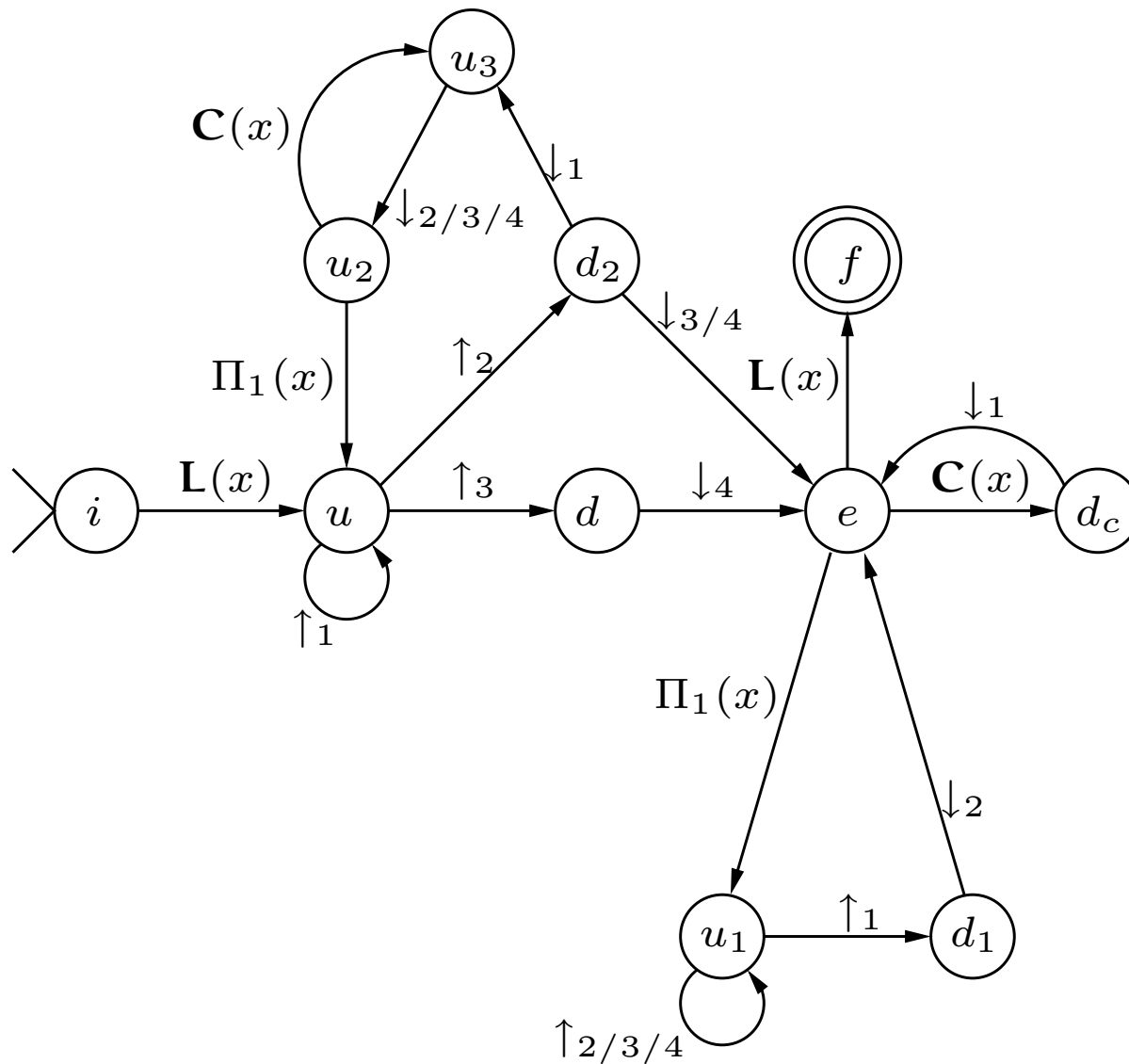
# Logical Reconstruction of Precedence: $x \triangleleft y$

**Terminal Nodes:**  $x$  and  $y$  are terminal nodes and for some internal node  $z$  dominating  $x$  and  $y$  and for the paths  $X$  and  $Y$  used by the TWA to connect  $z$  with  $x$  and  $y$ , respectively, the first leaf node on  $X$  precedes the first leaf node on  $Y$ .

**Internal Nodes:**  $x$  and  $y$  are internal nodes of the intended tree and every terminal node which  $x$  dominates precedes (in the lifted tree) every terminal node that  $y$  dominates.



# The tree-walking automaton $A_{\triangleleft}$



# Intended Structures via MSO transduction

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

$$Q = \{\triangleleft, \triangleleft^*, \triangleleft^+, \triangleleft, \dots\}$$

$$\varphi = \phi_{\mathfrak{A}_{G'}}$$

$$\psi = L(x)$$

$$\theta_{\triangleleft} = \text{trans}_{W_{\triangleleft}}(x, y)$$

$$\theta_{\triangleleft^*} = (\forall X)[\triangleleft\text{-closed}(X) \vee x \in X \rightarrow y \in X]$$

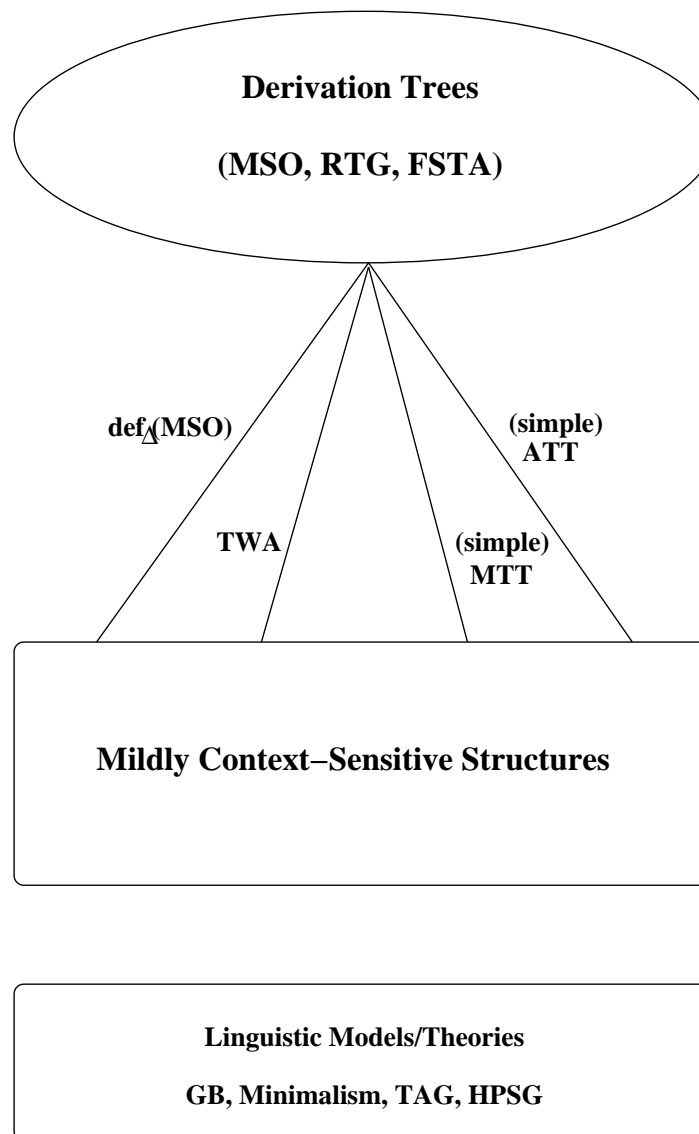
$$\theta_{\triangleleft^+} = x \triangleleft^* y \vee x \not\approx y$$

$$\theta_{\triangleleft} = (\exists u, v)[u \triangleleft^* x \wedge v \triangleleft^* y \wedge \text{trans}_{W_{\triangleleft}}(u, v)]$$

$$\theta_{\text{labels}} = \text{as in the old structure } R$$



# Conclusion



- Motivation and Theoretical Background
- TAGs
- (Monadic) Context-Free Tree Grammars
- From (M)CFTGs to Regular Tree Grammars
- Interpreting the Intended Structures



## Courcelle/Makowsky (Draft of July 2000)

Every MS definable transduction has a natural contravariant counterpart called their *backwards translation* mapping, an MS formula expressing a property of the object structure into an MS formula expressing the same property on the input structure.

...

In particular, MS decidability results and existence of linear algorithms are easily obtained once a class of structures is recognized to be the image by a transduction of a class of trees, ...





# Appendix: Some more Details

# Lifting

Suppose that  $\Sigma$  is a ranked alphabet. The *derived*  $\mathbb{N}$ -sorted alphabet  $\Sigma^L$  is defined as follows:

For each  $n \geq 0$ ,

$$\Sigma'_n = \{f' \mid f \in \Sigma_n\}$$

is a new set of symbols of sort  $n$ ;

back to Lifting



# Lifting

Suppose that  $\Sigma$  is a ranked alphabet. The *derived*  $\mathbb{N}$ -sorted alphabet  $\Sigma^L$  is defined as follows:

for each  $n \geq 1$  and each  $i, 1 \leq i \leq n$ ,

$$\pi_i^n$$

is a new symbol, the  $i$ th projection symbol of sort  $n$ ;

back to Lifting



# Lifting

Suppose that  $\Sigma$  is a ranked alphabet. The *derived*  $\mathbb{N}$ -sorted alphabet  $\Sigma^L$  is defined as follows:

for each  $n \geq 0, k \geq 0$  the new symbol

$$c_{(n,k)}$$

is the  $(n, k)$ th composition symbol.

[back](#) to Lifting



Suppose that  $\Sigma$  is a ranked alphabet. The *derived*  $\mathbb{N}$ -sorted alphabet  $\Sigma^L$  is defined as follows:

$$\Sigma_0^L = \Sigma'_0$$

$$\Sigma_n^L = \Sigma'_n \cup \{\pi_i^n \mid 1 \leq i \leq n\} \text{ for } n \geq 1$$

$$\Sigma_{n,k}^L = \{c_{(n,k)}\} \text{ for } n, k \geq 0$$

$$\Sigma_i^L = \text{otherwise}$$

[back to Lifting](#)



# Model-Theoretic Interpretation

Basic Idea (Rabin 1965)

Obtaining a structure  $\mathbb{B} = \langle B, Q \rangle$  from a structure  $\mathbb{A} = \langle A, \mathcal{R} \rangle$  where  $\mathcal{R}$  and  $Q$  are families of relation symbols.

[back](#) to MSO transductions



# Tree-Walking Automaton

A *tree-walking automaton (with tests)* (Bloem and Engelfriet 1997) over some ranked alphabet  $\Sigma$  is a finite automaton  $\mathfrak{A} = (Q, \Delta, \delta, I, F)$  with states  $Q$ , directives  $\Delta$ , transitions  $\delta : Q \times \Delta \rightarrow Q$  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)”,
- $\phi(x)$  “verify that  $\phi$  holds at the current node”.



# Regular Tree-Node Relations

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

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

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  $\phi(x)$  of  $\mathfrak{A}$  are *MSO* definable,  $\mathfrak{A}$  specifies a *regular tree-node relation*, which is itself *MSO* definable.





# Reflexive transitive closure in *MSO*

$$R\text{-closed}(X) \stackrel{def}{\iff} (\forall x, y)[x \in X \wedge R(x, y) \rightarrow y \in X]$$

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



# Walking language $\Rightarrow$ *MSO*-formula

$$\mathit{trans}(x, y) = \perp$$

$$\mathit{trans}_{\downarrow_i}(x, y) = \mathit{edge}_i(x, y)$$

$$\mathit{trans}_{\uparrow_i}(x, y) = \mathit{edge}_i(y, x)$$

$$\mathit{trans}_{\phi(x)}(x, y) = \phi(x) \wedge x = y$$

$$\mathit{trans}_{W_1 \cup W_2}(x, y) = \mathit{trans}_{W_1}(x, y) \vee \mathit{trans}_{W_2}(x, y)$$

$$\mathit{trans}_{W_1 \cdot W_2}(x, y) = \exists z (\mathit{trans}_{W_1}(x, z) \wedge \mathit{trans}_{W_2}(z, y))$$

$$\mathit{trans}_{W^*}(x, y) = \mathit{trans}_W^*(x, y)$$

$$\mathit{trans}_W^*(x, y) = \forall X (\forall v, w (v \in X \wedge \mathit{trans}_W(v, w) \rightarrow w \in X) \\ \wedge x \in X \rightarrow y \in X)$$

back to  $W_{\triangleleft}$

