Semistructured Data and Linguistics

Uwe Mönnich and Frank Morawietz

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

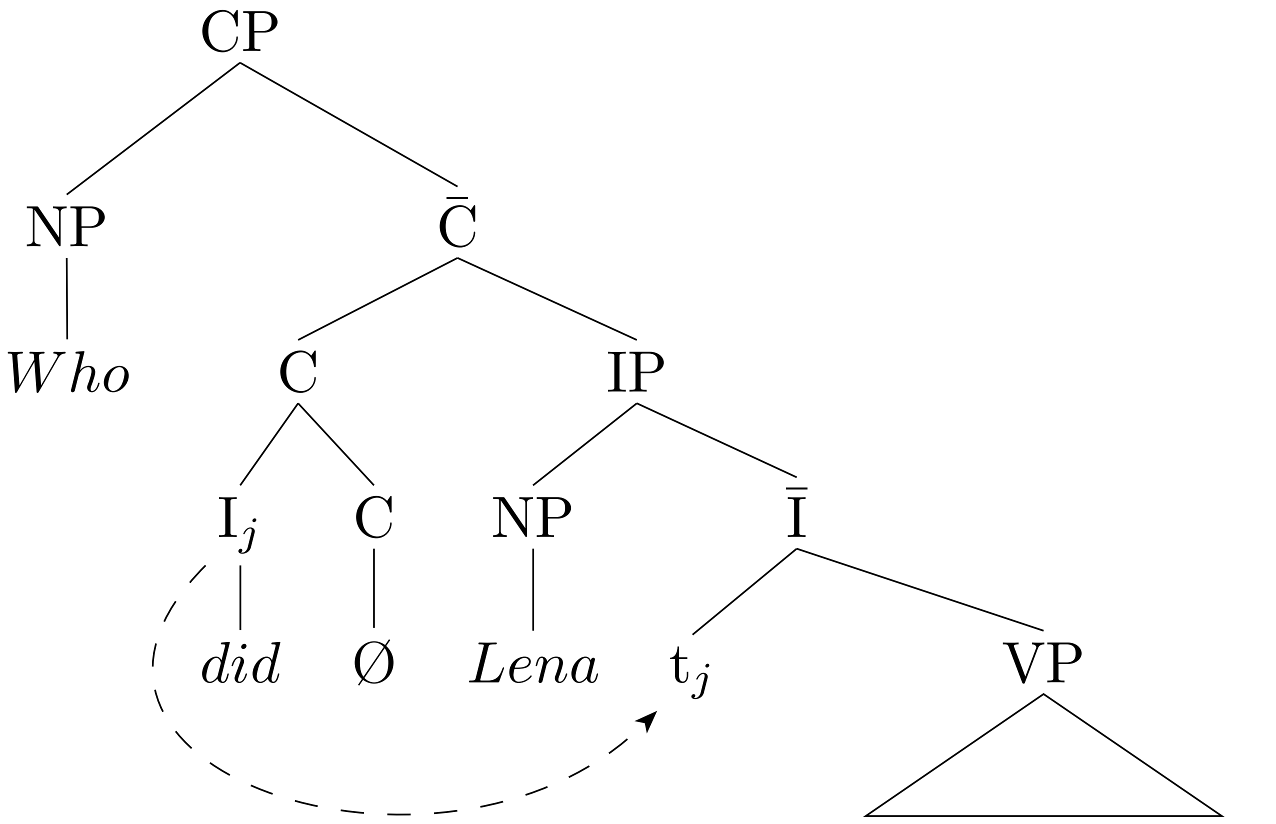
http://tcl.sfs.uni-tuebingen.de/

Seminar für Sprachwissenschaft
Arbeitsbereich Theoretische Computerlinguistik
Universität Tübingen

Outline

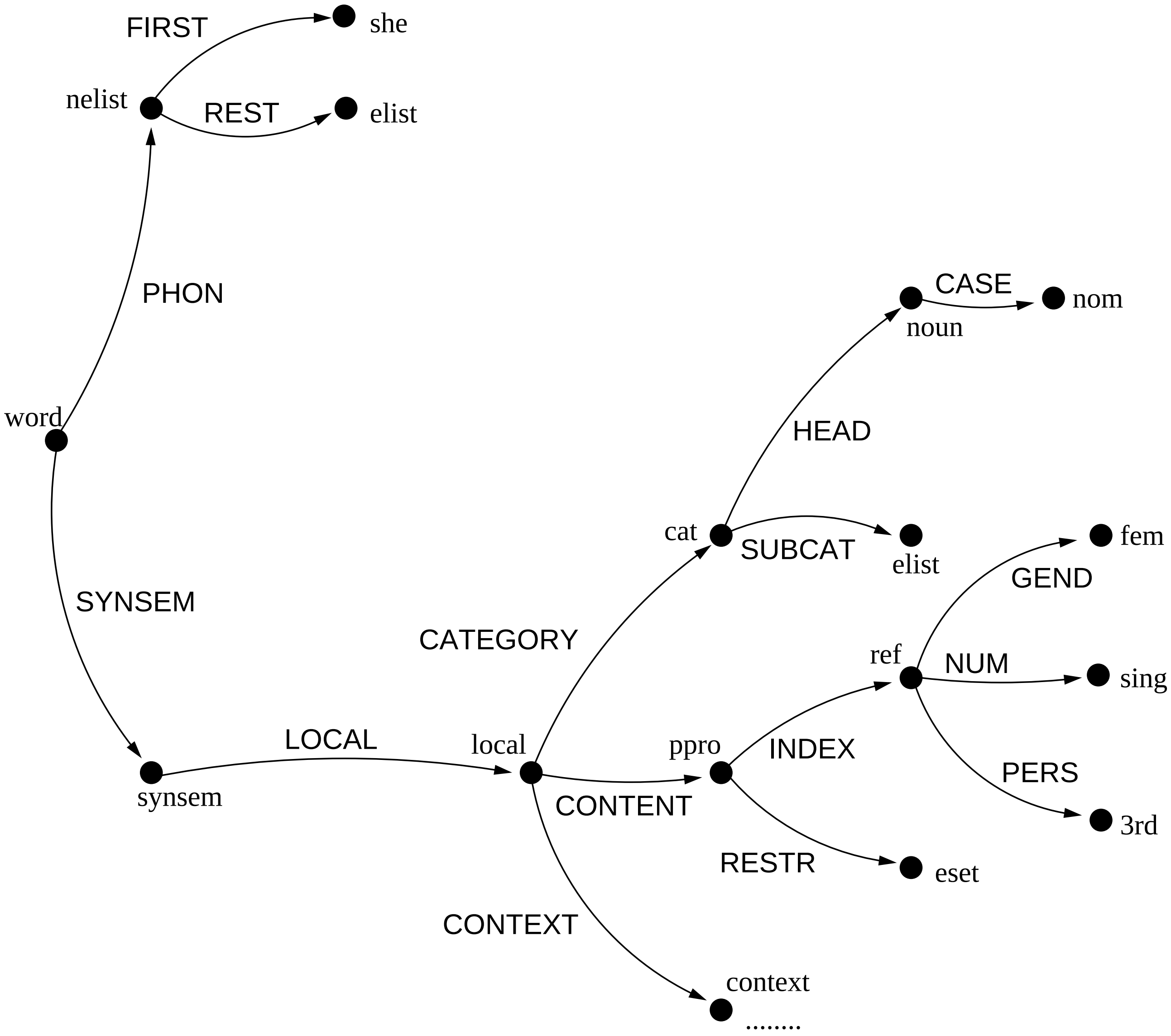
- Motivation and Theoretical Background
- TAGs
- (Monadic) Context-Free Tree Grammars
- From (M)CFTGs to Regular Tree Grammars
- Interpreting the Intended Structures

Semistructured Data in Linguistics



Who did Lena invite to her birthday party?

Semistructured Data in Linguistics



Semistructured Data in Linguistics

$$
\begin{bmatrix}
\textit{word} & \\
\text{PHON} & \begin{bmatrix} \textit{nelist} & \\ \text{FIRST} & \textit{she} \\ \text{REST} & \textit{elist} \end{bmatrix} \\
\text{SYNSEM} & \begin{bmatrix} \textit{synsem} & \\ \text{LOCAL} & \begin{bmatrix} \textit{local} & \\ \text{CATEGORY} & \begin{bmatrix} \textit{cat} & \\ \text{HEAD} & \begin{bmatrix}\textit{noun} & \\ \text{CASE} & \textit{nom}\end{bmatrix} \\ \text{SUBCAT} & \textit{elist} \end{bmatrix} \\ \text{CONTENT} & \begin{bmatrix} \textit{ppro} & \\ \text{INDEX} & \begin{bmatrix} \textit{ref} & \\ \text{GEND} & \textit{fem} \\ \text{NUM} & \textit{sing} \\ \text{PERS} & \textit{3rd} \end{bmatrix} \\ \text{RESTR} & \textit{eset} \end{bmatrix} \\ \text{CONTEXT} & \begin{bmatrix} \textit{context} \\ \ldots\;\ldots \end{bmatrix} \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

Semistructured Data in Linguistics

Penn Treebank II:

$(TOP\ (S\ (NP\text{-}SBJ$ my best friend$)$
$(VP$ gave
$(NP$ me$)$
$(NP$ chocolate$)$
$(NP\text{-}TMP$ yesterday$))$
.$))$

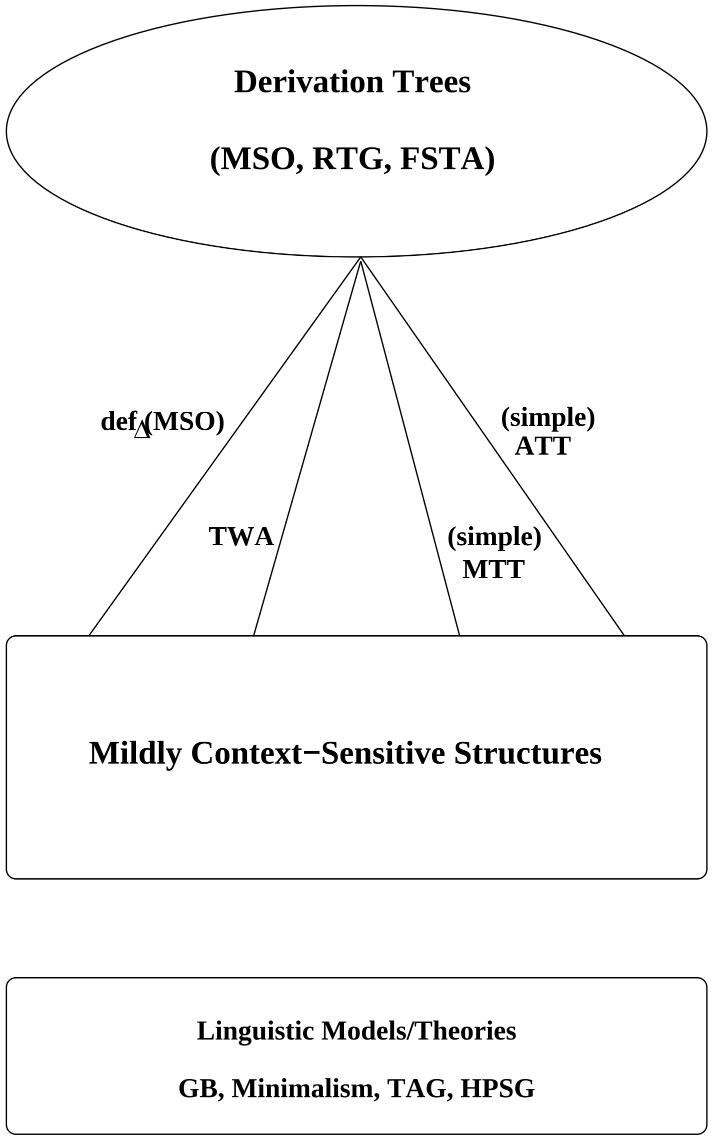
Semistructured Data in Linguistics

TGrep:

$$S <1 \ /\hat{\ }NP/ < (VP < (NP \ \$.. \ NP))$$

Get all Ss that start with an NP (not necessarily the subject) and that dominate a VP that in turn has two NP children – in other words, sentences with what might be double-object VPs.

Overview



Derivation Trees
(MSO, RTG, FSTA)
def_{Δ}(MSO)
TWA
(simple) MTT
(simple) ATT
Mildly Context−Sensitive Structures
Linguistic Models/Theories
GB, Minimalism, TAG, HPSG

General Structure of the Approach

Translating TAGs to Monadic CFTGs

$\Rightarrow$ Lifting of Monadic Context-Free Tree Languages

$\Rightarrow$ Logical Characterization of the lifted Monadic CFTG-Languages

$\Rightarrow$ Retrieval of the Intended Structures via MSO-Definable Transductions

General Structure of the Approach

Translating TAGs to Monadic CFTGs

$\Rightarrow$ Lifting of Monadic Context-Free Tree Languages

$\Rightarrow$ Logical Characterization of the lifted Monadic CFTG-Languages

$\Rightarrow$ Retrieval of the Intended Structures via MSO-Definable Transductions

- trees as relational structures

General Structure of the Approach

Translating TAGs to Monadic CFTGs

$\Rightarrow$ Lifting of Monadic Context-Free Tree Languages

$\Rightarrow$ Logical Characterization of the lifted Monadic CFTG-Languages

$\Rightarrow$ 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

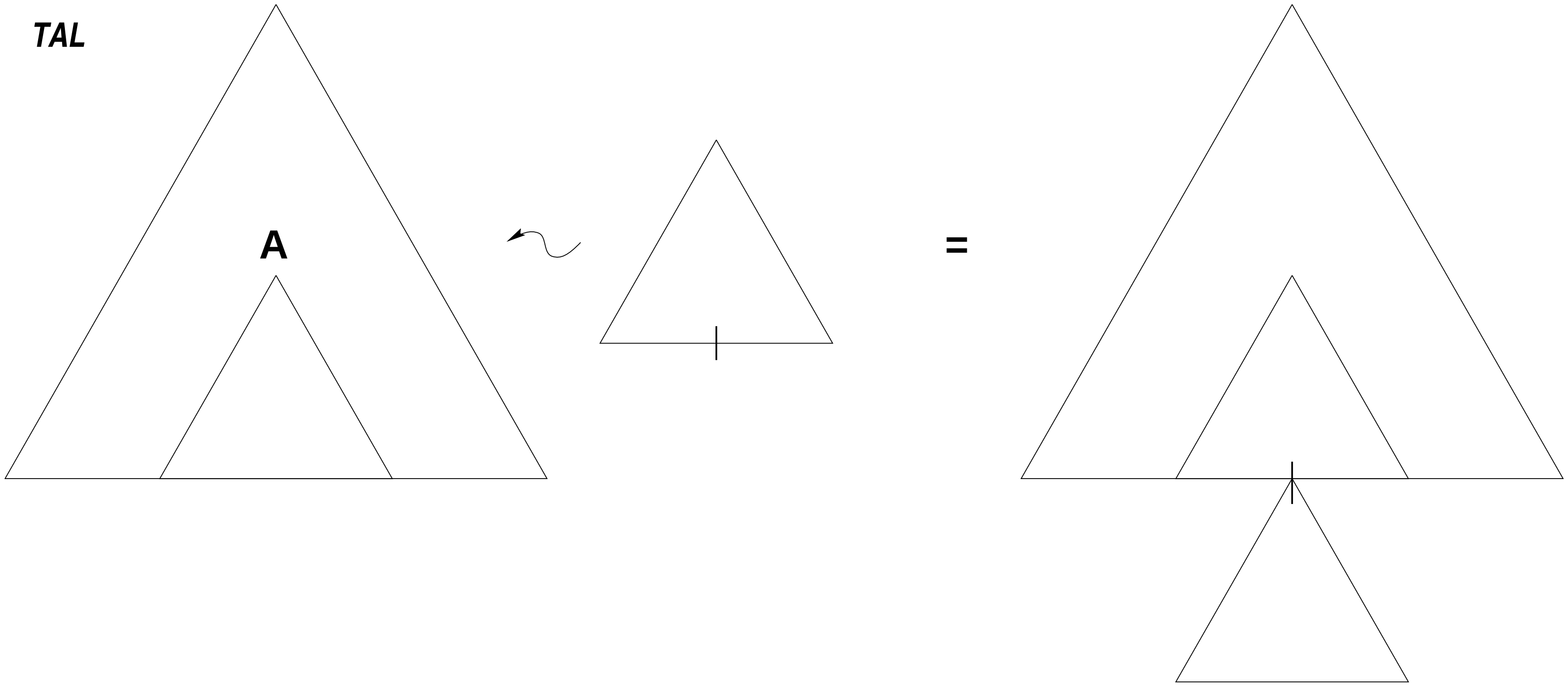
$\Rightarrow$ Lifting of Monadic Context-Free Tree Languages

$\Rightarrow$ Logical Characterization of the lifted Monadic CFTG-Languages

$\Rightarrow$ 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)

Derivation Steps in TAGs



Tree Grammars (CFTG & RTG)

A *context-free tree grammar* Γ 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

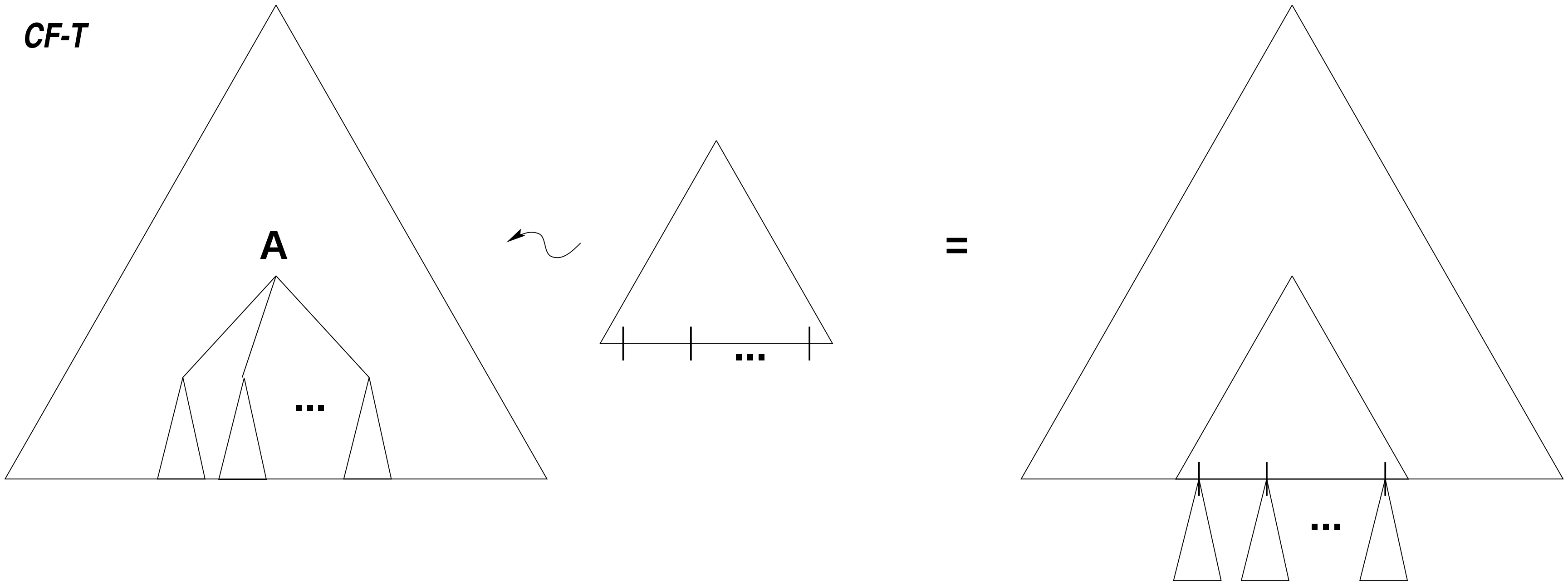
$$F(x_1, \cdots, x_n) \longrightarrow t$$

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

An application of a rule $F(x_1, \cdots, x_n) \longrightarrow t$ “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



Monadic CFTGs

$$A \longrightarrow a$$

$$A \longrightarrow B(C)$$

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

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

Fact

Theorem (Mönnich 1997, Fujiyoshi & Kasai 2000)

TAGs are equivalent to monadic CFTGs.

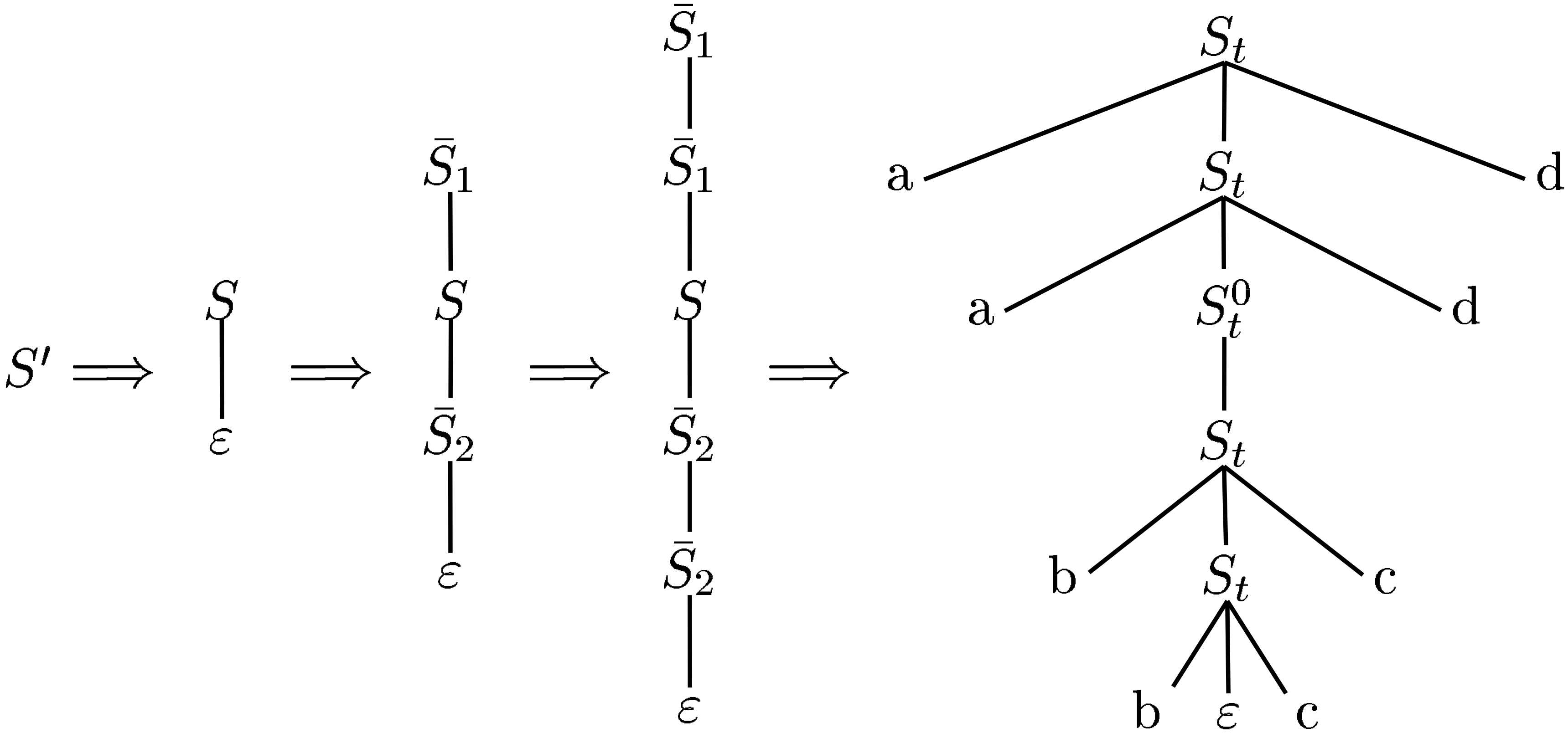
Monadic CFTG $\mathcal{G}$ for $a^n b^n c^n d^n$

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

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

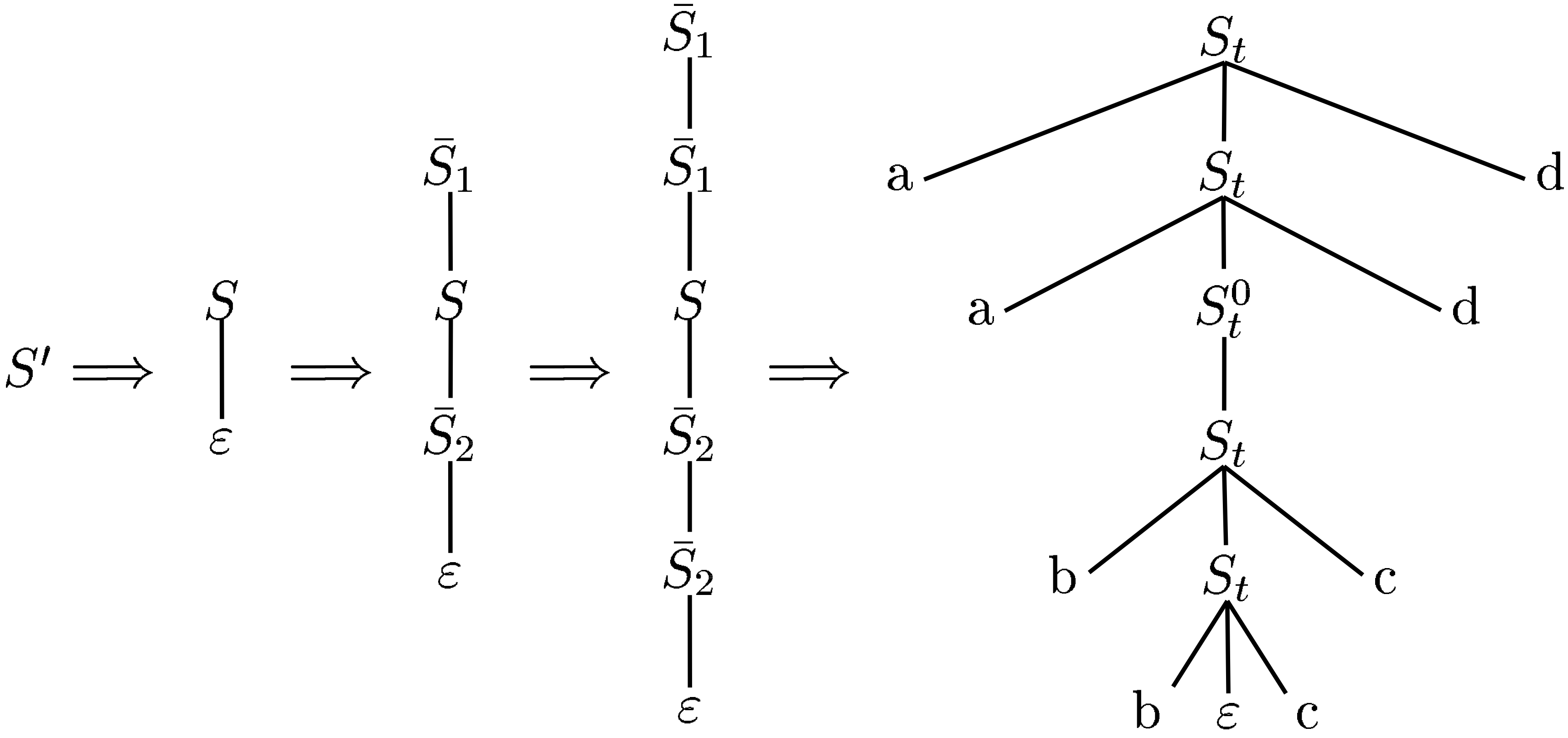
$$\begin{array}{rcl} \overline{a} & \longrightarrow & a \\ \overline{b} & \longrightarrow & b \\ \overline{c} & \longrightarrow & c \\ \overline{d} & \longrightarrow & d \end{array}$$

An example derivation



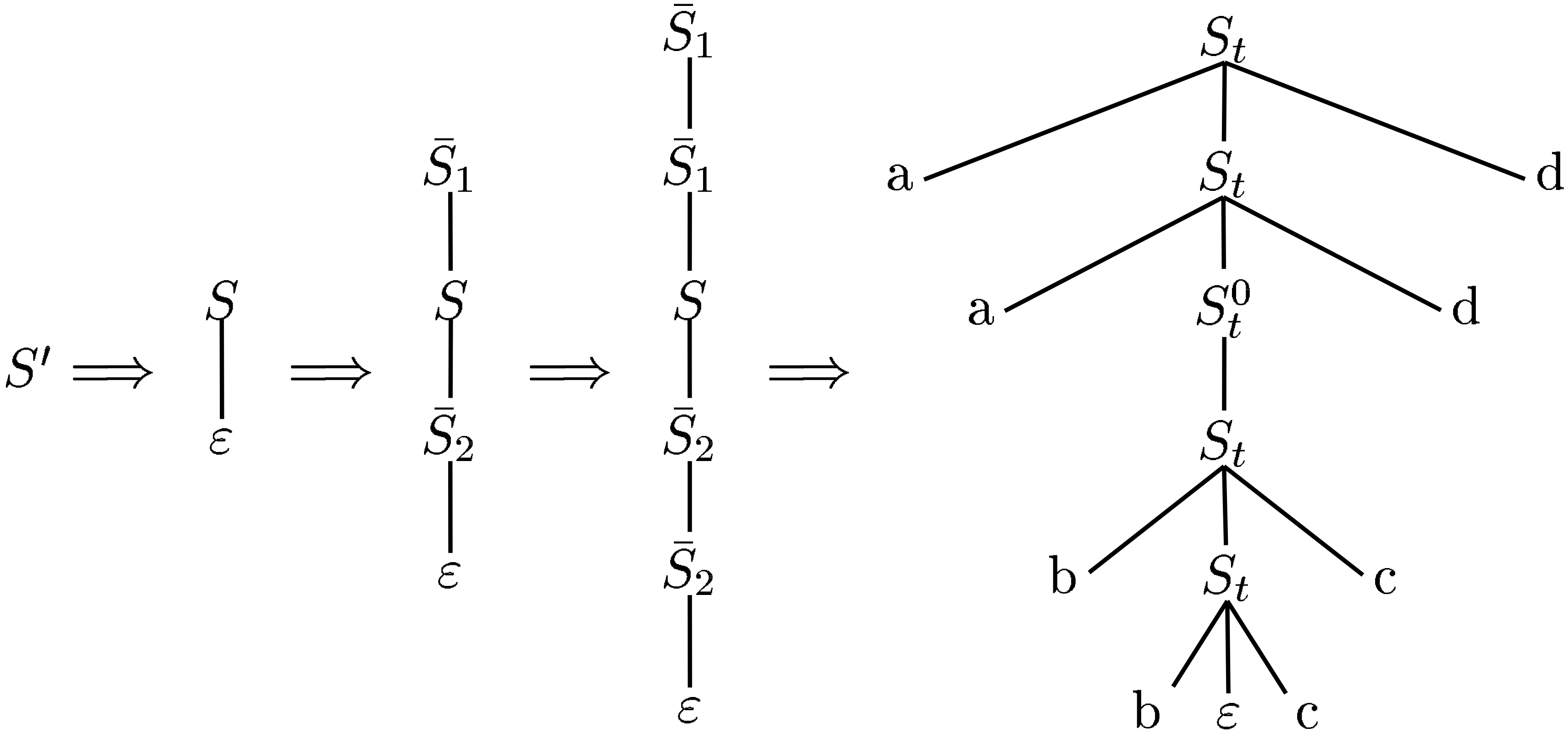
$S' \longrightarrow S(\varepsilon)$

An example derivation



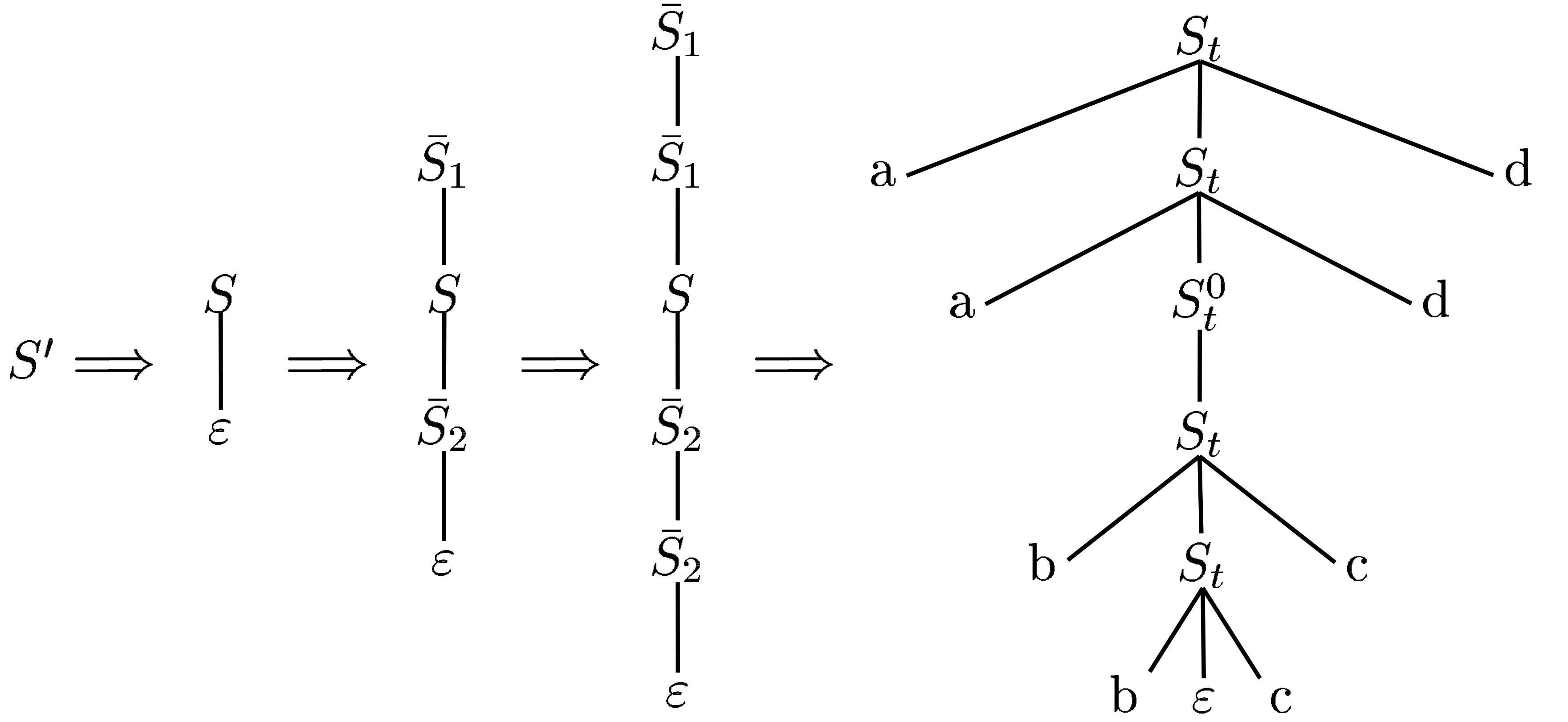
$$S(x) \longrightarrow \overline{S}_1(S(\overline{S}_2(x)))$$

An example derivation



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

An example derivation



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

Regular Tree Grammars

$\Gamma = \langle \Sigma, \mathrm{F}_0, S, \mathrm{P} \rangle$ where

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

$$\mathrm{F}_0 = \langle \mathrm{F}_{\varepsilon,s} \,|\, s \in \mathcal{S} \rangle$$

$S \in \mathrm{F}_0$ is the starting symbol

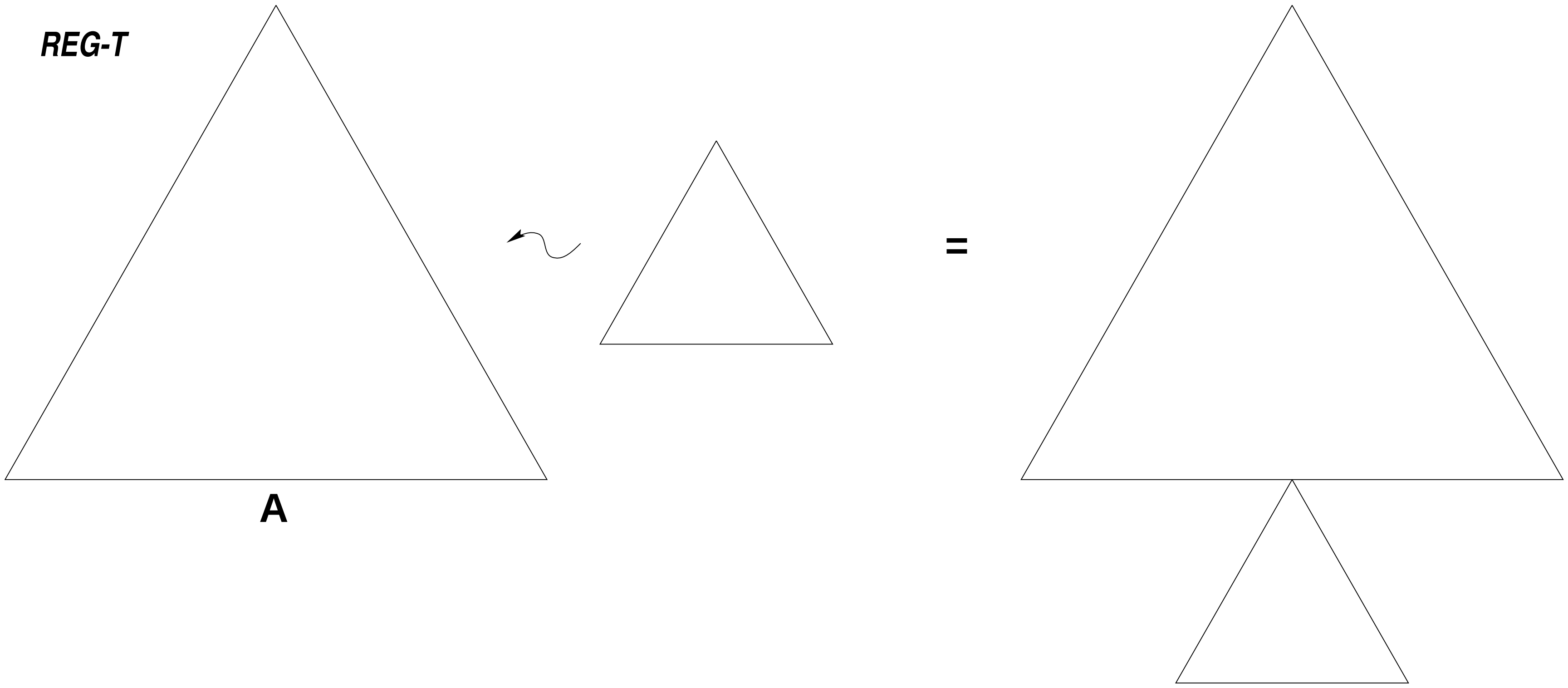
P is a set of productions

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

$$F \longrightarrow t$$

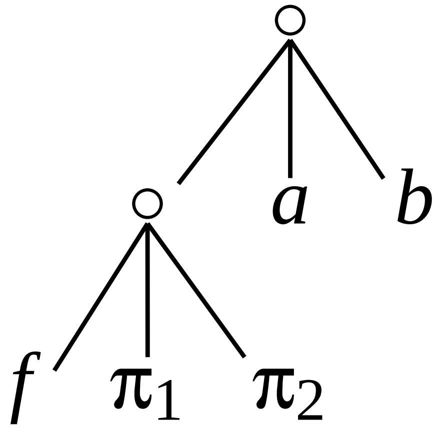
with $F \in \mathrm{F}_{\varepsilon,s}$, and t a term over $T(\Sigma \cup \mathrm{F}_0)$.

Derivation Steps in RTGs



From (M)CFTGs to RTGs: Intuition

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



Lifting

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

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

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

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

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

$$\text{for } f \in \Sigma_n, n \geq 1$$

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

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

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

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

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

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

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

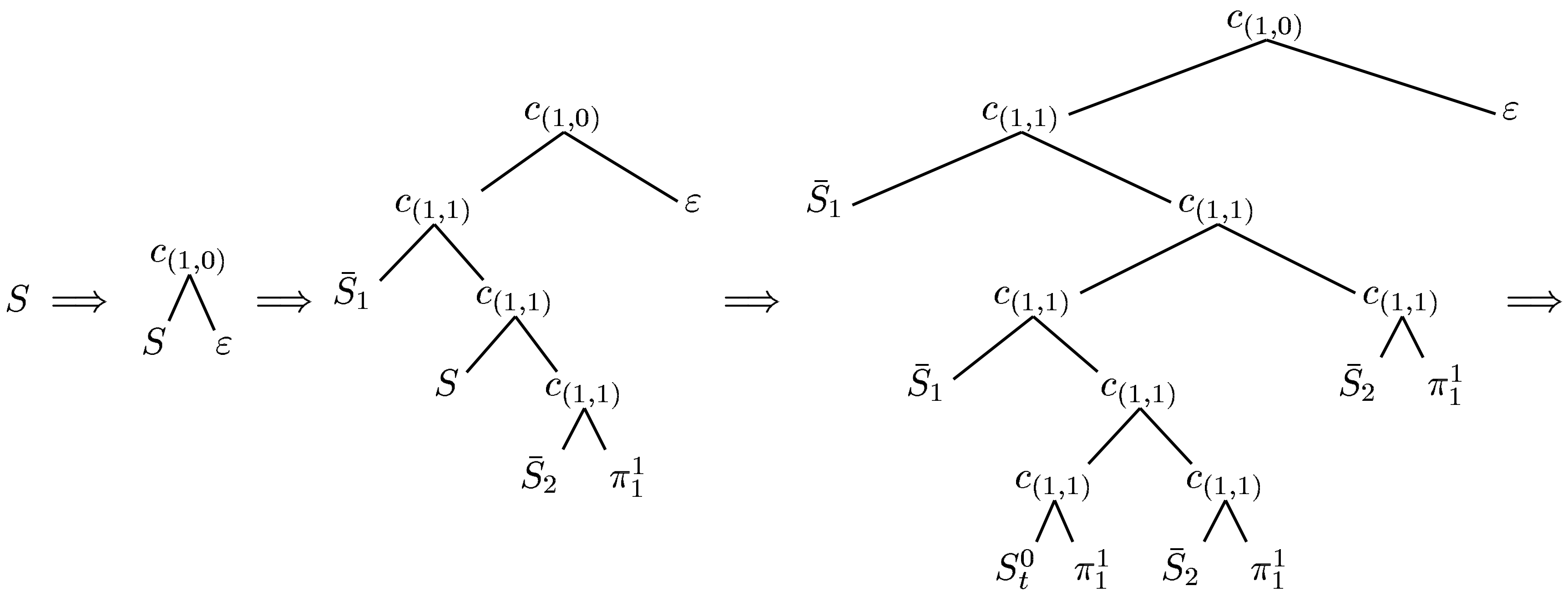
$$S(x) \longrightarrow \overline{S}_1(S(\overline{S}_2(x)))$$

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

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

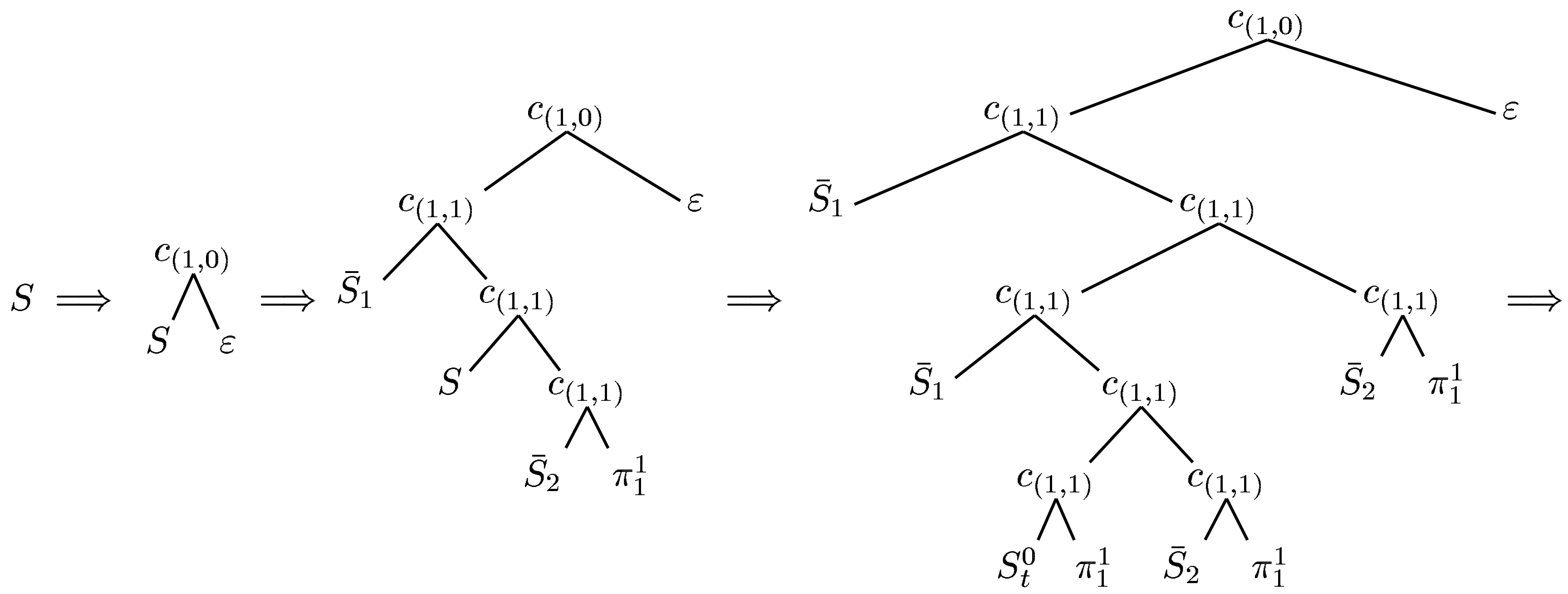
$$\overline{S}_2(x) \longrightarrow S_t(\overline{b}, x, \overline{c})$$

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



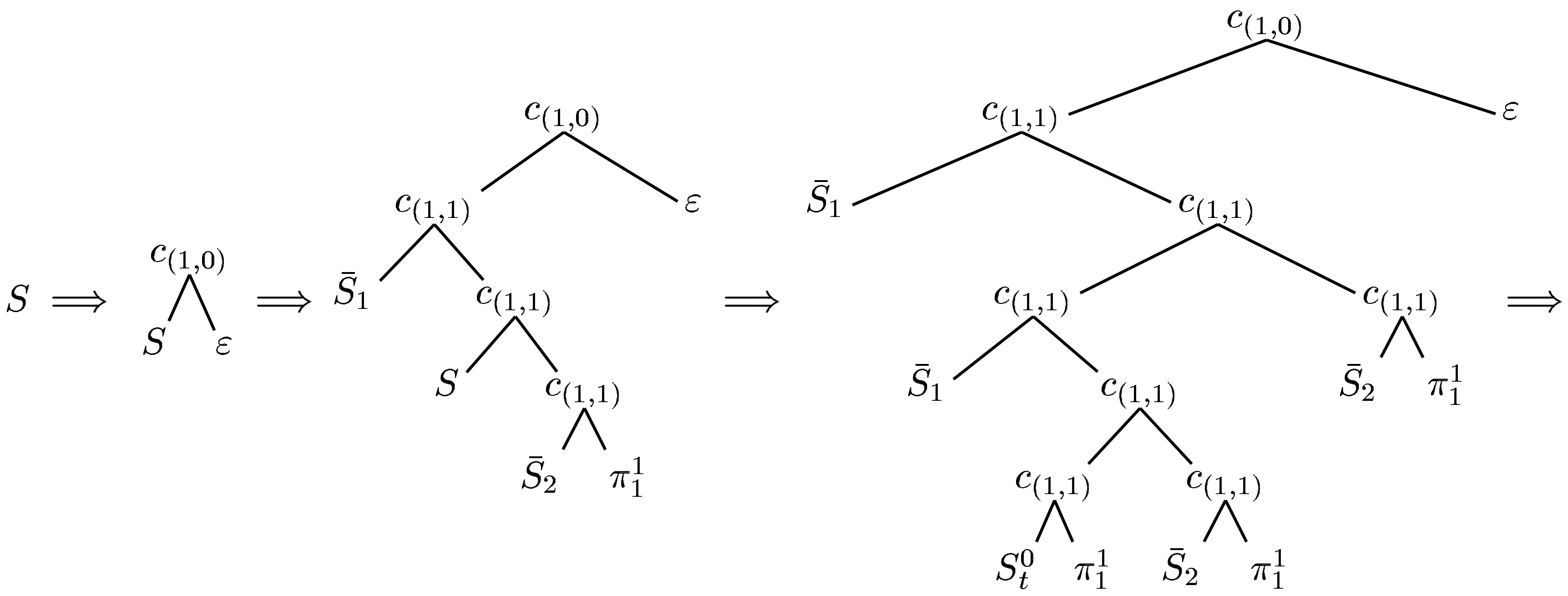
$$S' \longrightarrow S(\varepsilon)$$

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



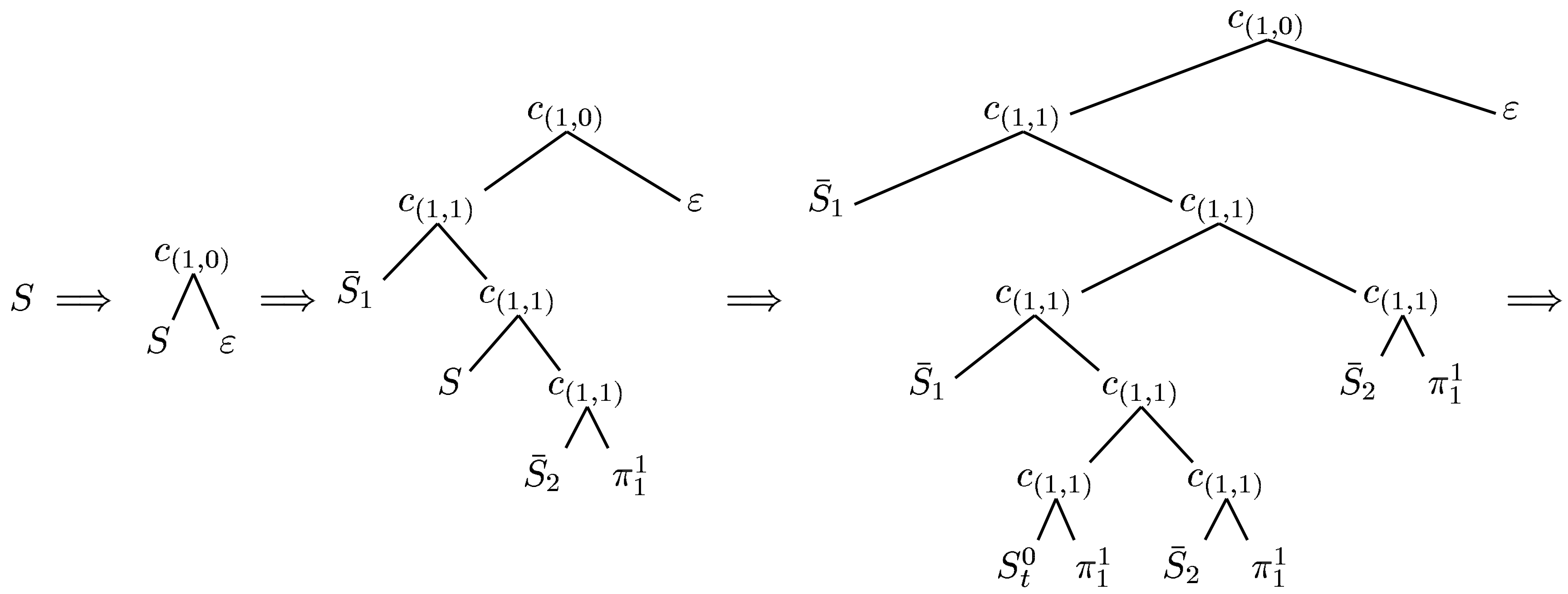
$$S(x) \longrightarrow \overline{S}_1(S(\overline{S}_2(x)))$$

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



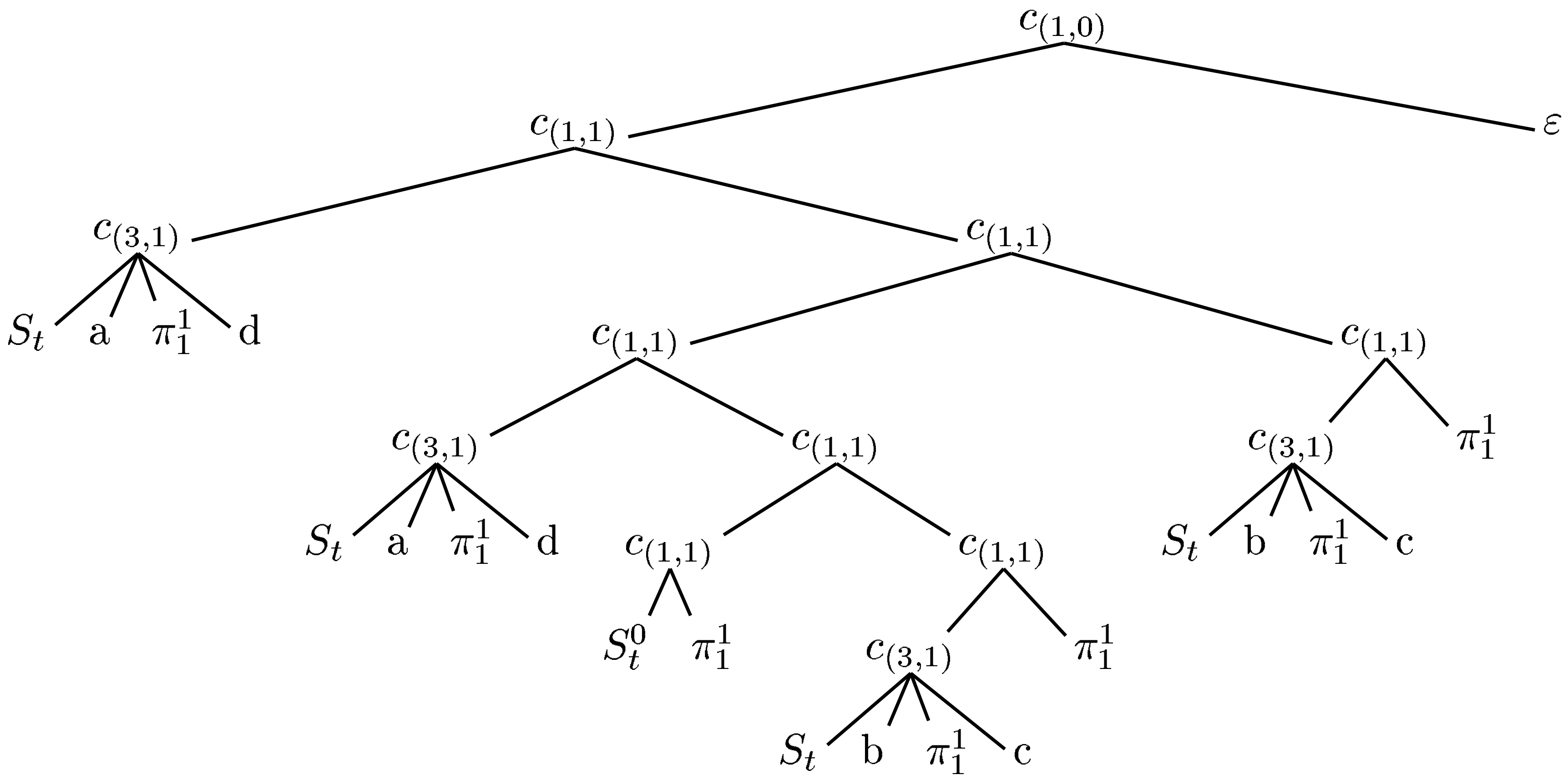
$$S(x) \longrightarrow \overline{S}_1(S(\overline{S}_2(x)))$$

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



$$\bar{S}_1(x) \longrightarrow S_t(\bar{a}, x, \bar{d})$$
$$\bar{S}_2(x) \longrightarrow S_t(\bar{b}, x, \bar{c})$$

An example derivation of $\mathcal{G}'$ (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 $\varphi_{\mathfrak{A}_{\mathcal{G}'}}$

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

$$(\forall x)[(\neg \exists y)[x \lhd y] \rightarrow x \in X_0] \wedge$$ % Initial State

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

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

Eberhard Karls Universität Tübingen

Goal

Define a set of relations

$$R^I = \{\blacktriangleleft^\star, \blacktriangleleft, \blacktriangleleft^+, \triangleleft, \textit{c-command}, \ldots\}$$

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 $\triangleleft \in R^I$ as the precedence relation on the resulting structure,

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

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

MSO definable transductions

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

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

φ the domain of the transduction

ψ the resulting domain of Q

θ_q the new relations

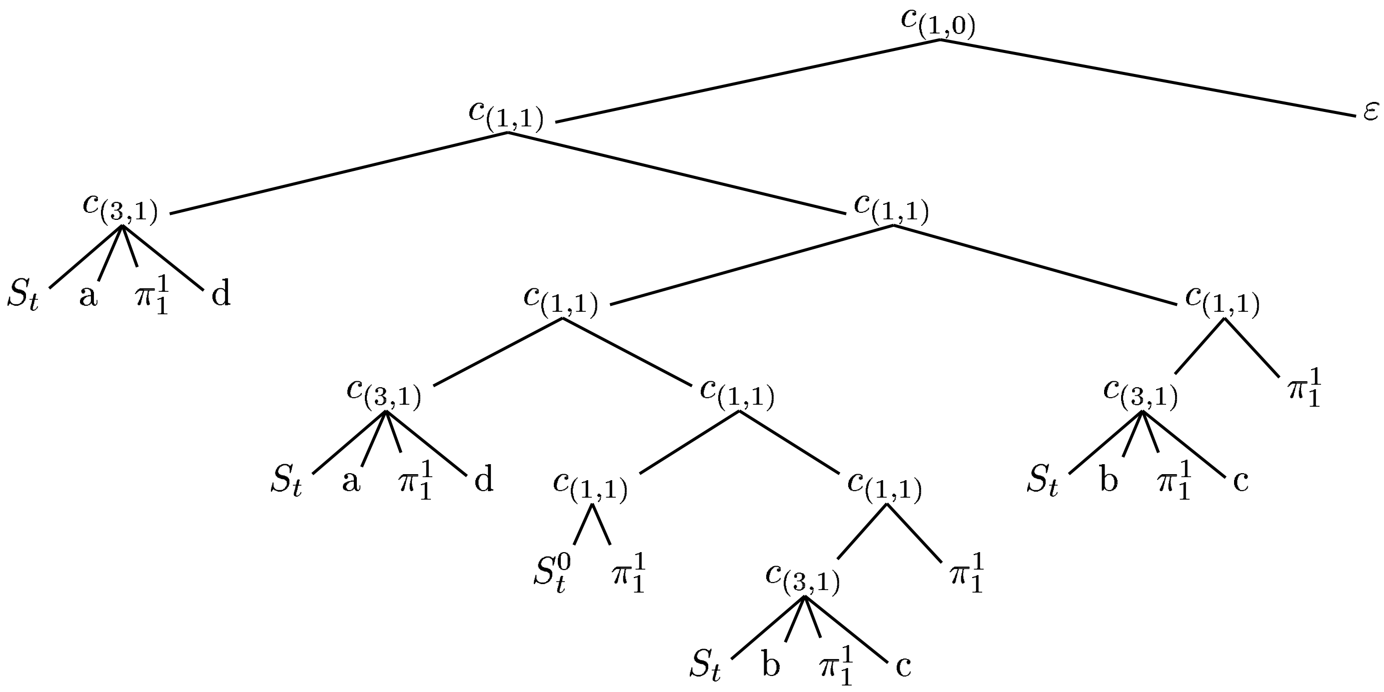
Def'n from Courcelle 1997, Basic Idea (Rabin 1965)

Reconstructing the Intended Structures

Homomorphism:

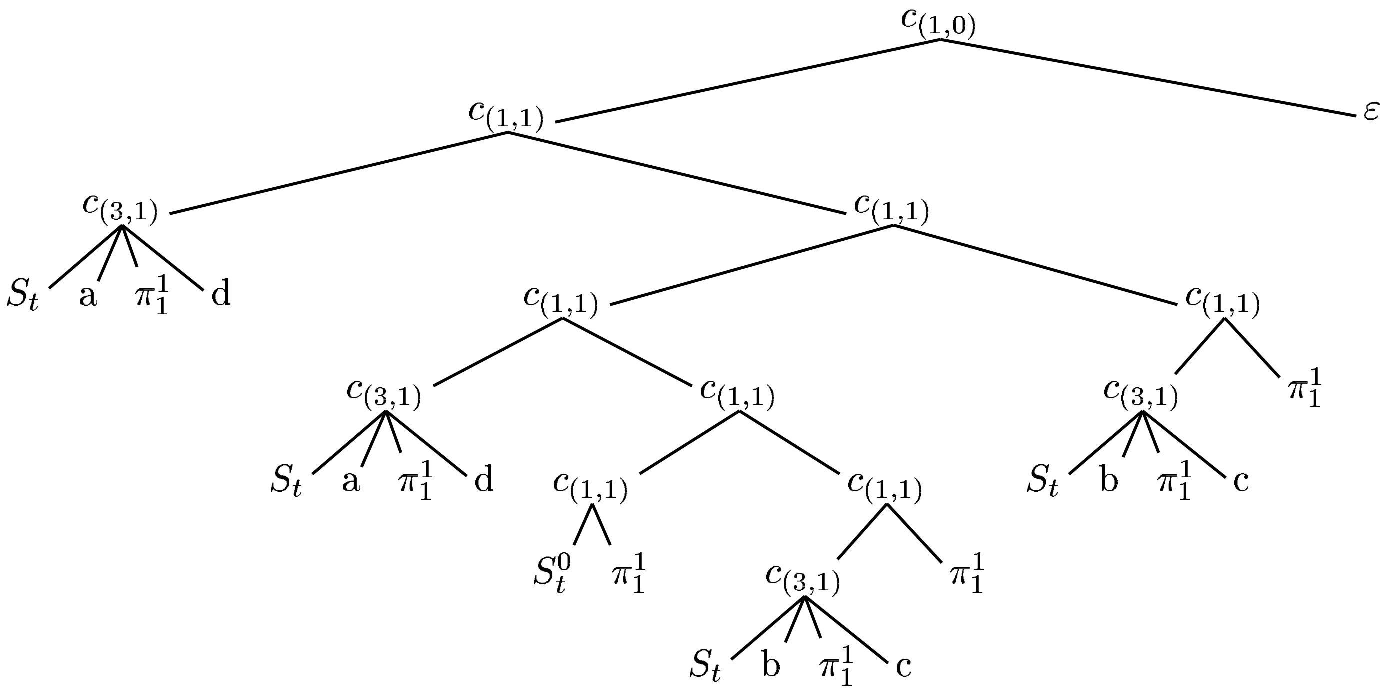
$$h(f') = f(x_1, \ldots, x_n) \text{ for } f \in \Sigma_n$$
$$h(\pi_i^n) = x_i$$
$$h(c_{(n,k)}(t, t_1, \ldots, t_n)) = h(t)[h(t_1), \ldots, h(t_n)]$$

A few general facts



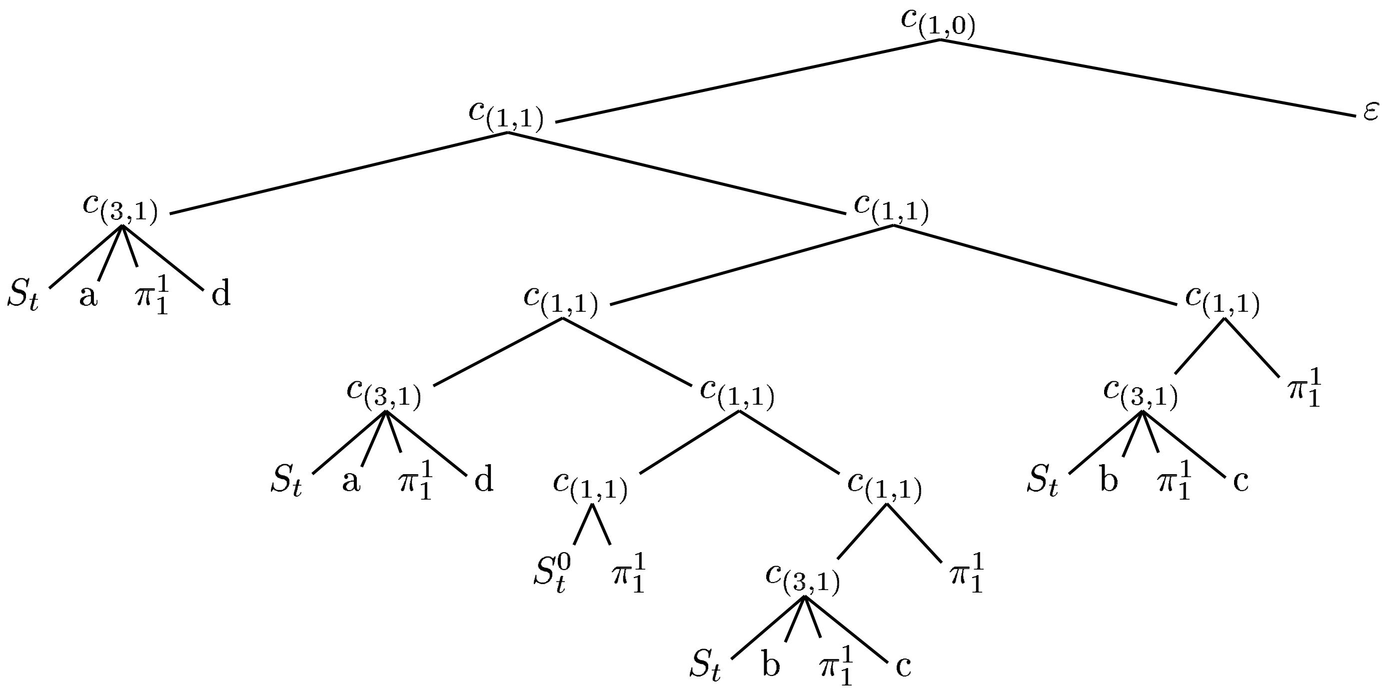
1. Our trees feature three families of labels: the "linguistic" symbols $\mathrm{L} = \mathcal{T}^L \cup \mathcal{N}^L$, where $\mathcal{T} = \Sigma_0$ of the underlying (M)CFTG and $\mathcal{N} = \bigcup_{n\geq 1} \Sigma_n$; the "composition" symbols $\mathrm{C} = \bigcup_{n,k\geq 0} c_{n,k}$; and the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

A few general facts



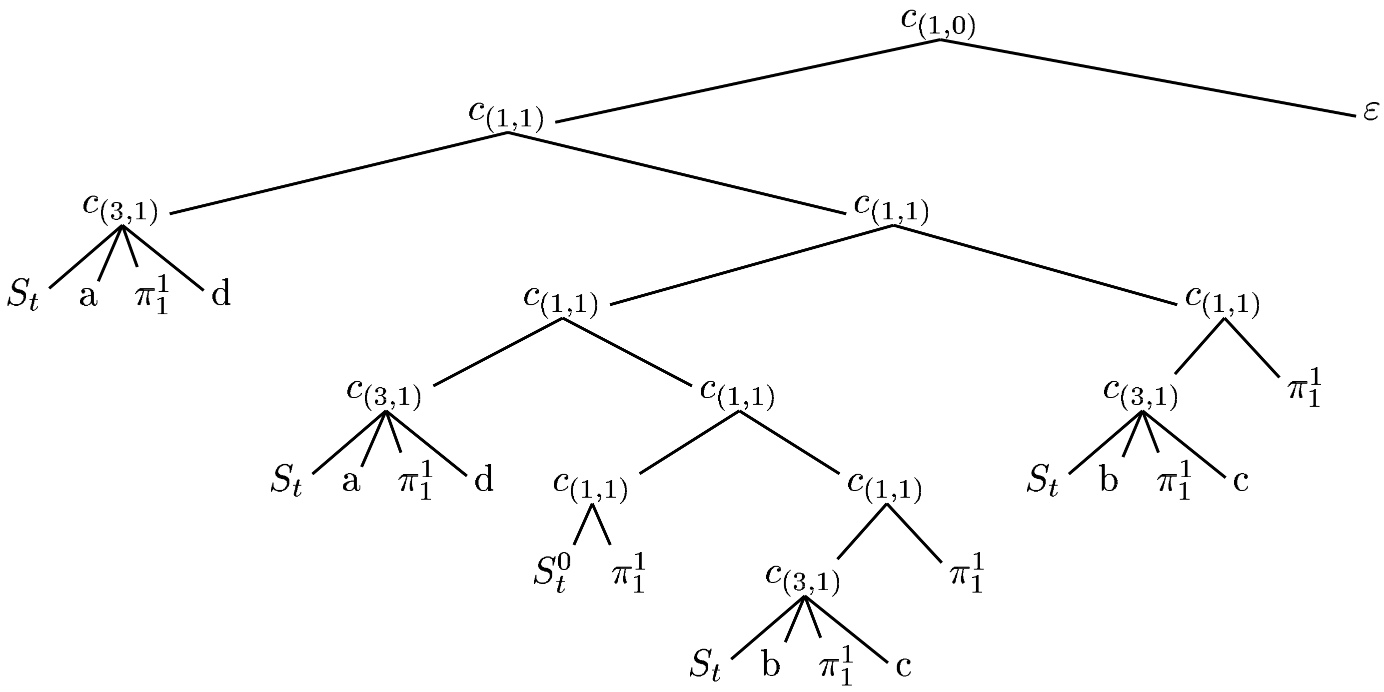
2. All non-terminal nodes in T^L are labeled by some $c \in \mathsf{C}$. No terminal node is labeled by some c.

A few general facts



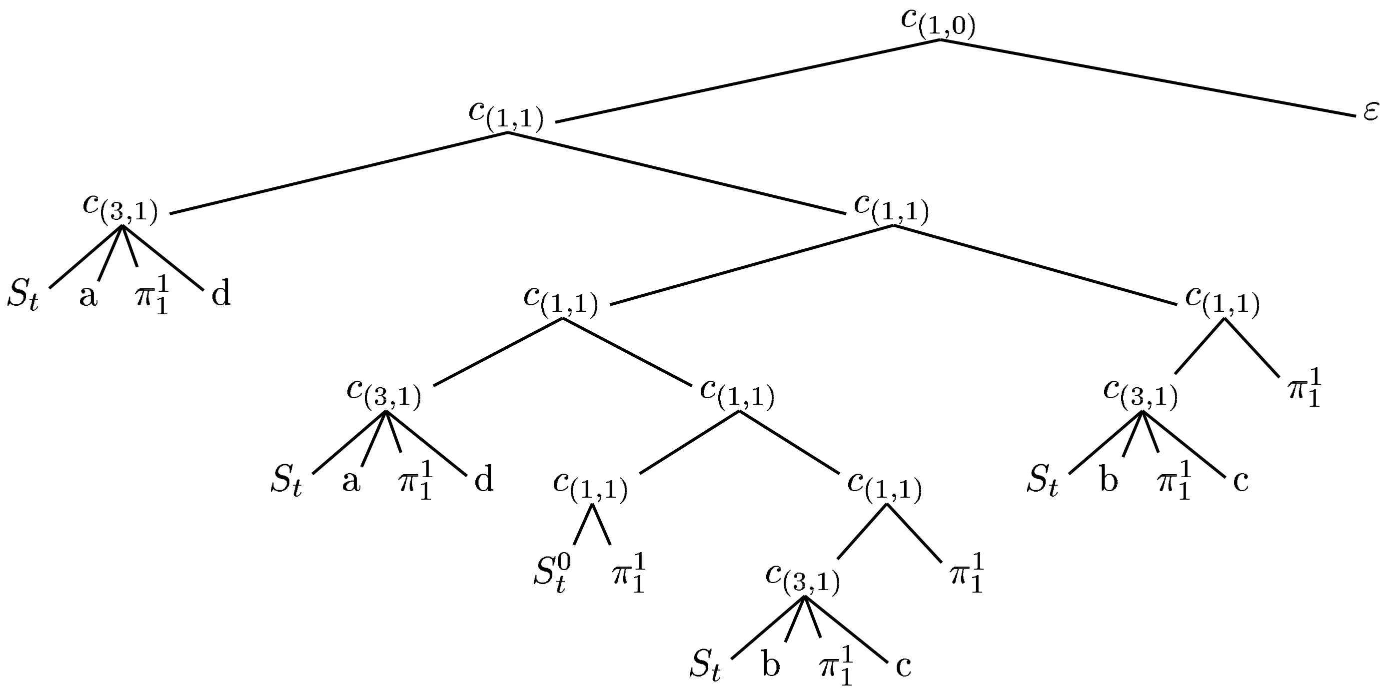
3. The terminal nodes in T^L are either labeled by some "linguistic" symbol or by some "projection" symbol $\pi \in \Pi_i$.

A few general facts



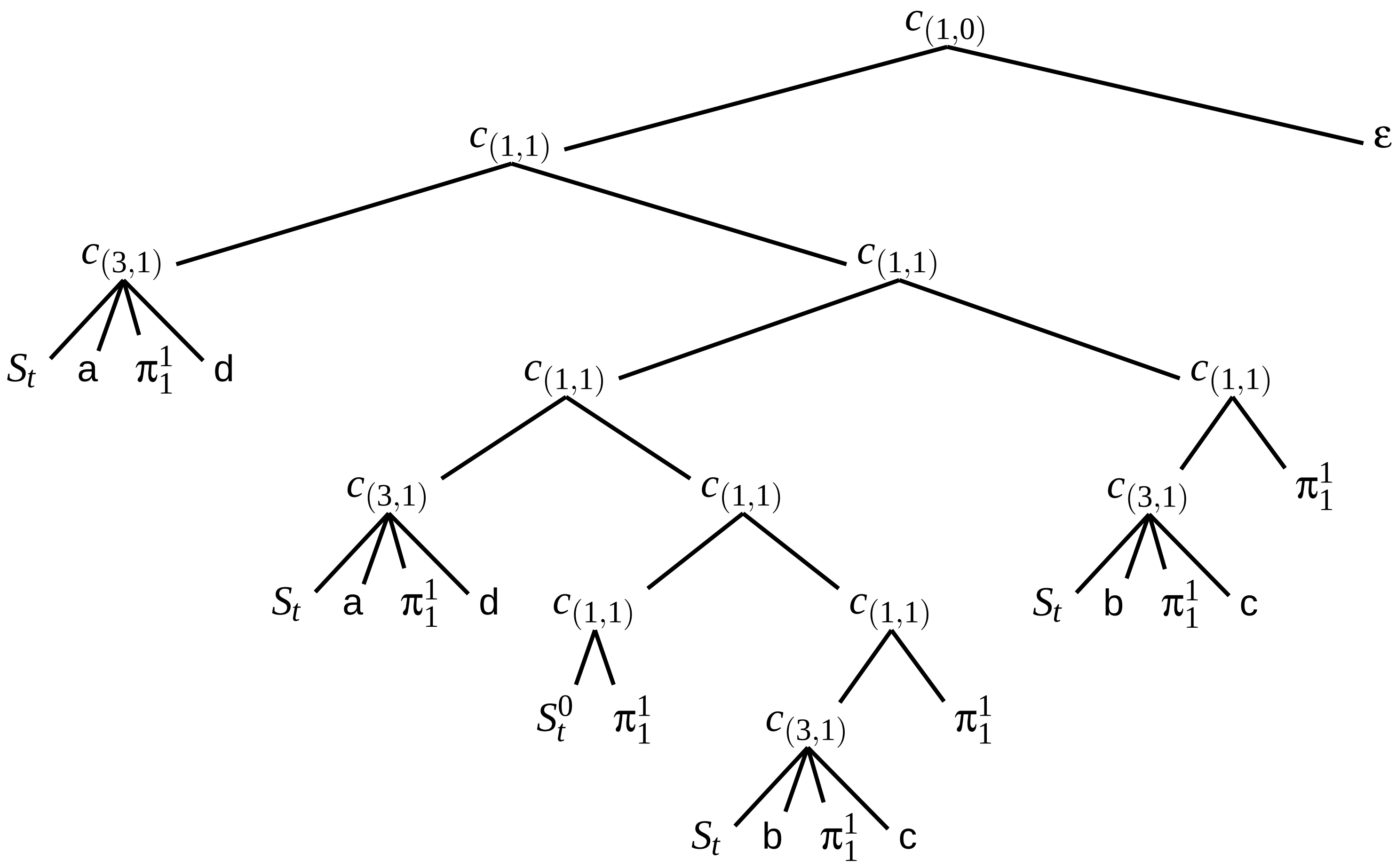
4. Any “linguistic” node dominating anything in the intended tree is on some left branch in T^L, i.e., it is the left-most daughter of some $c \in \mathrm{C}$.

A few general facts

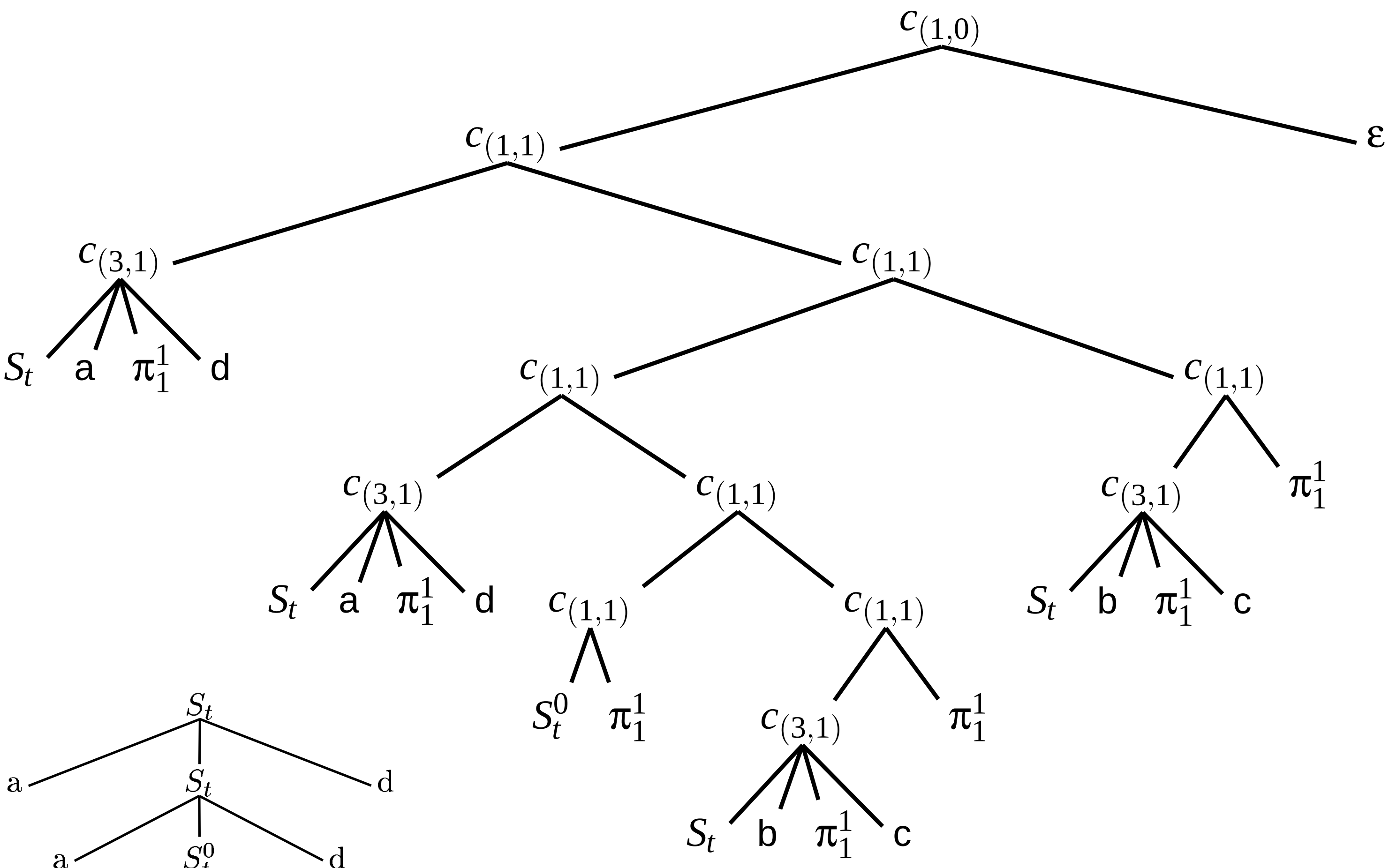


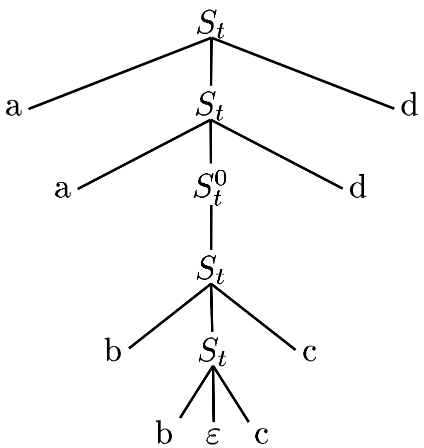
5. For any node p labeled with some “projection” symbol $\pi \in \Pi_i$ in T^L there is a unique node n (labeled with some $c \in \mathrm{C}$ by (2.)) which properly dominates p and whose i-th sister (to the right) will eventually evaluate to the value of π. Moreover, n will be the first node properly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

Reconstructing the Intended Structures

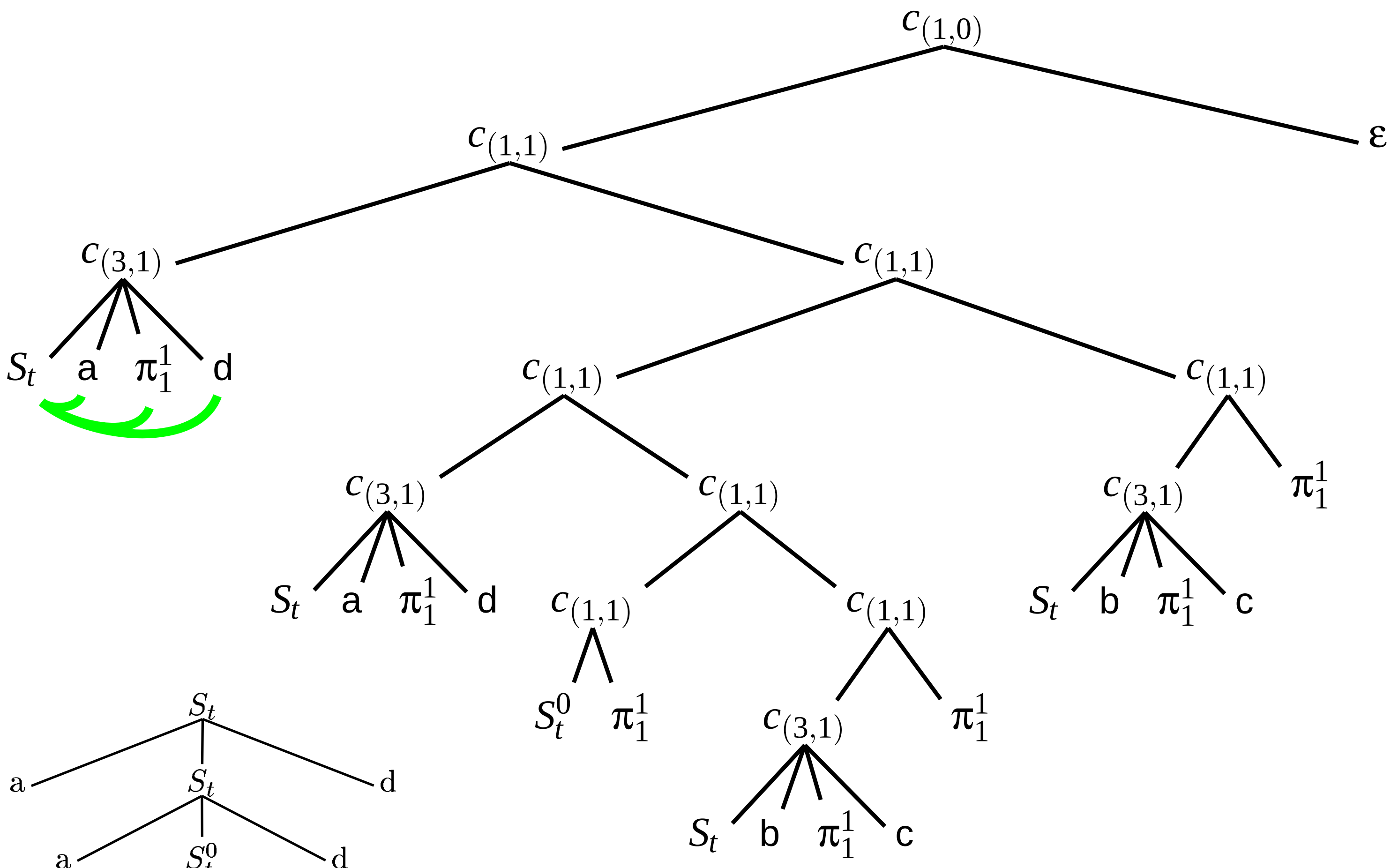


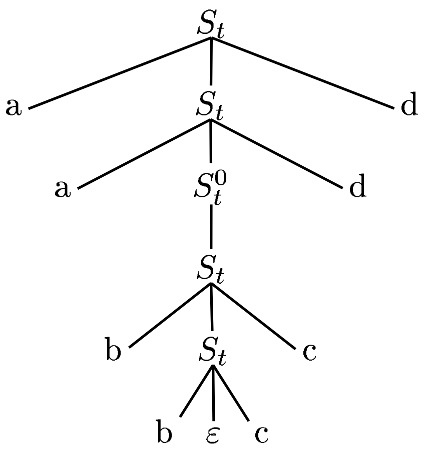
Reconstructing the Intended Structures



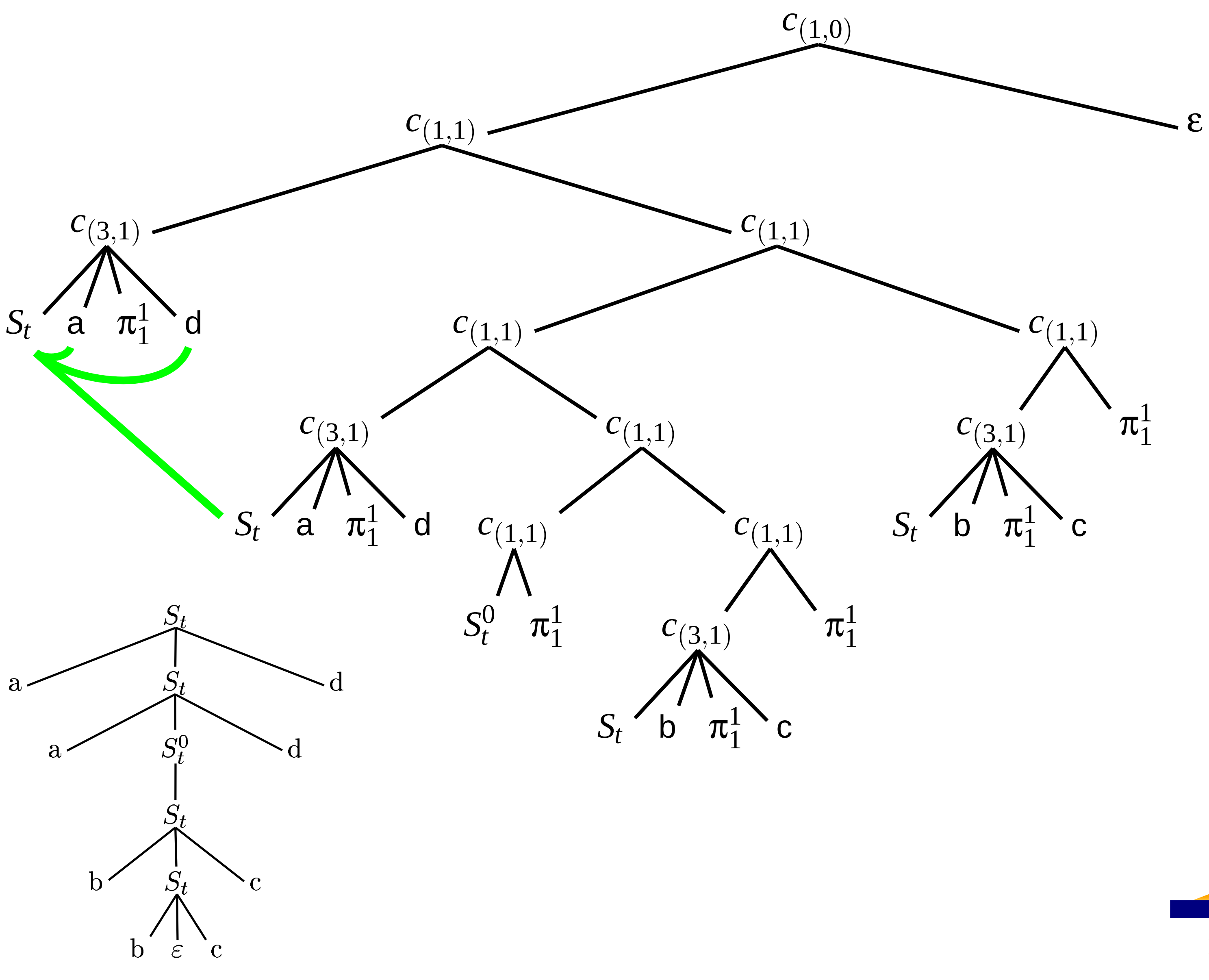


Reconstructing the Intended Structures



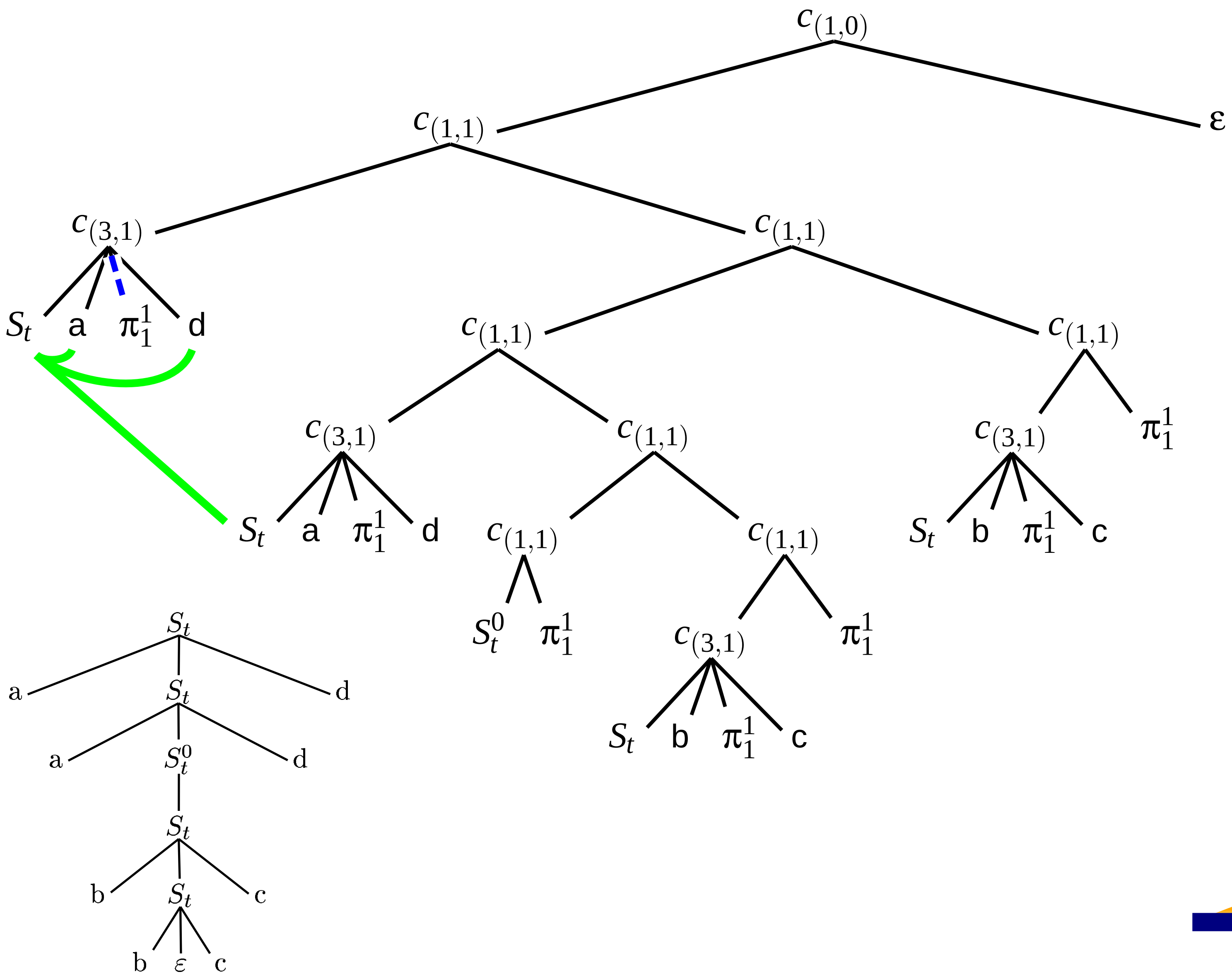


Reconstructing the Intended Structures

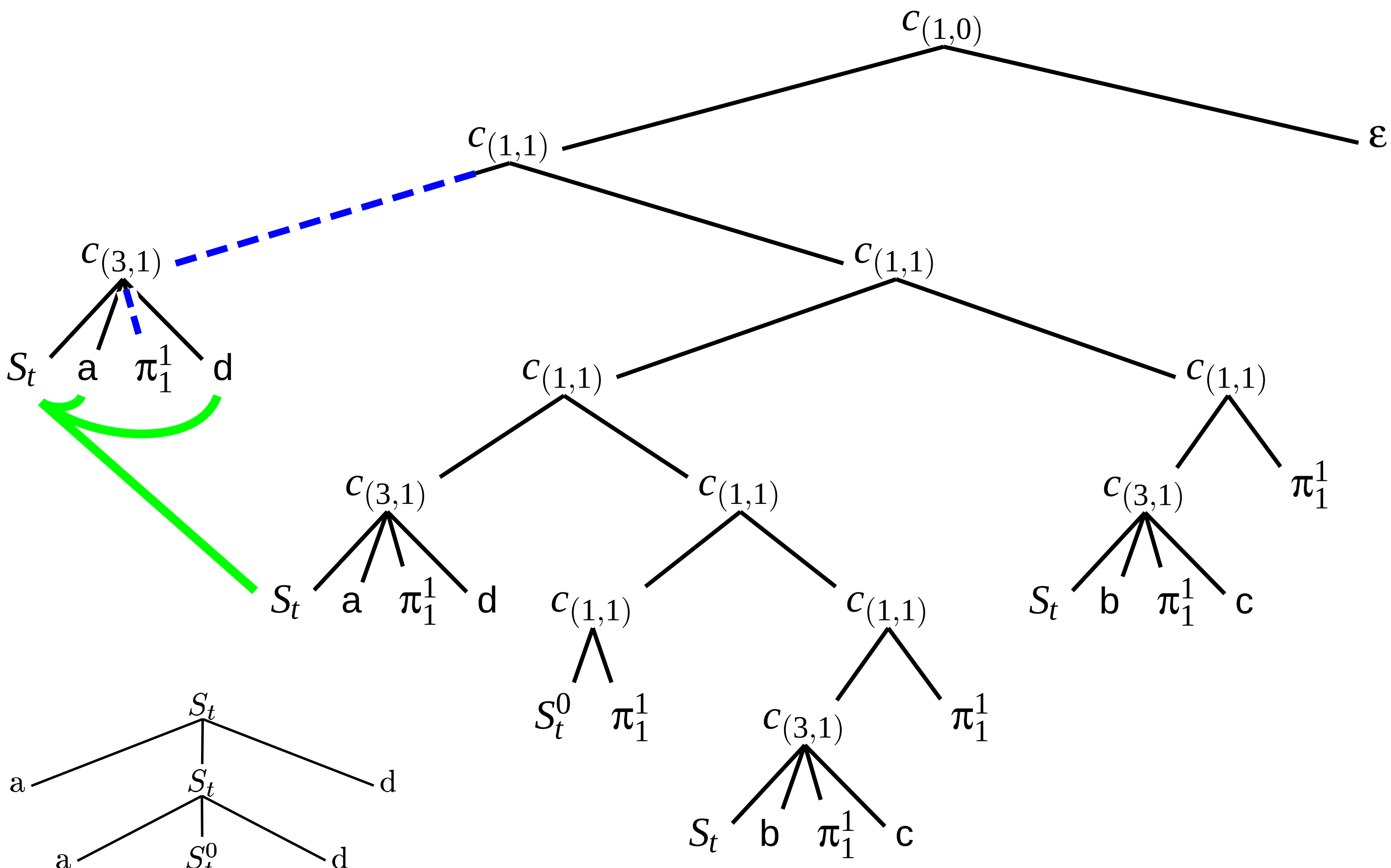


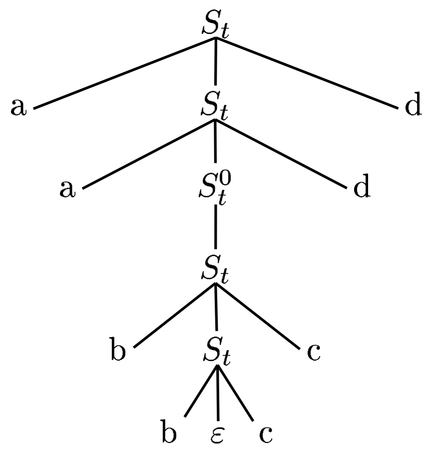
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Reconstructing the Intended Structures

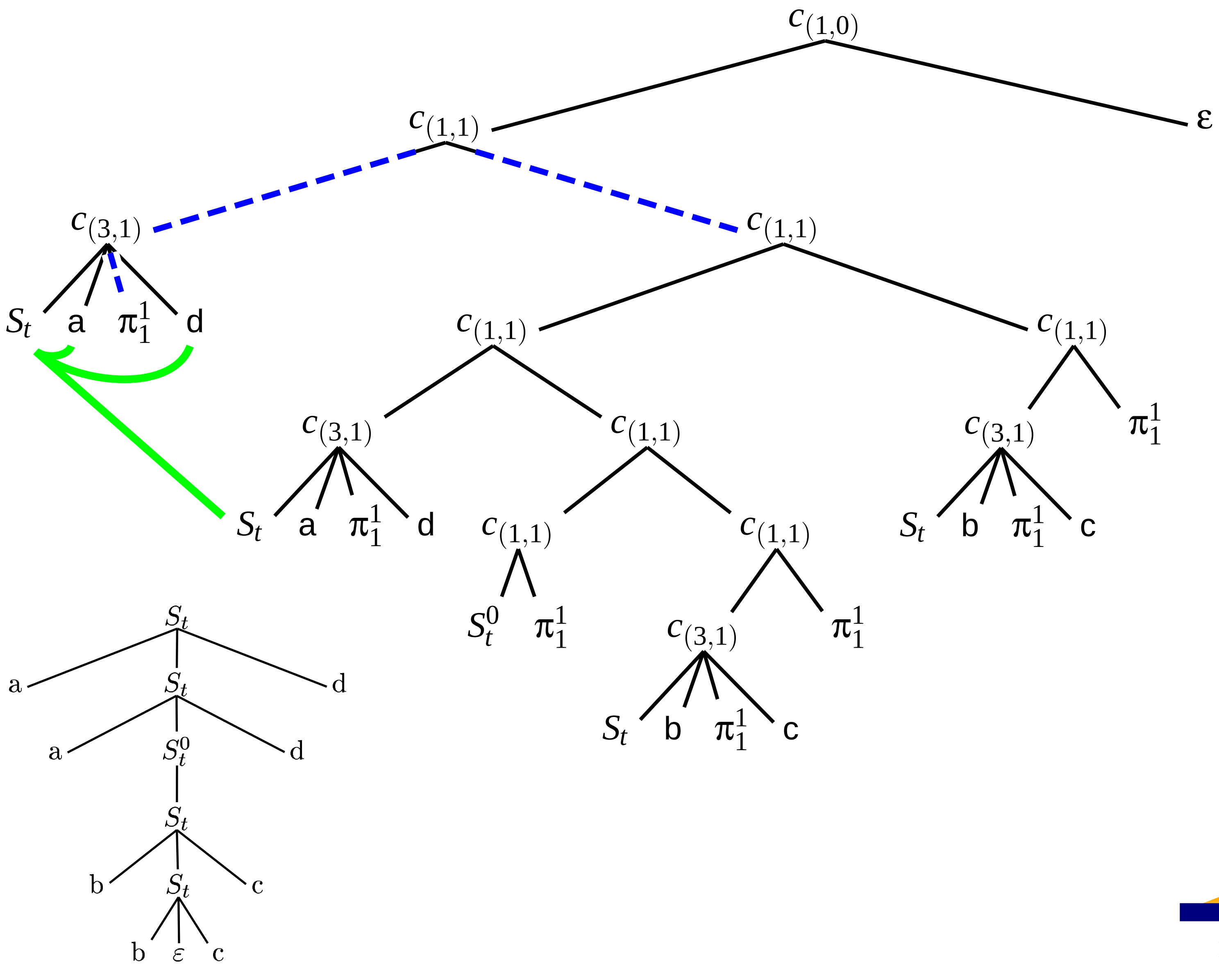


Reconstructing the Intended Structures

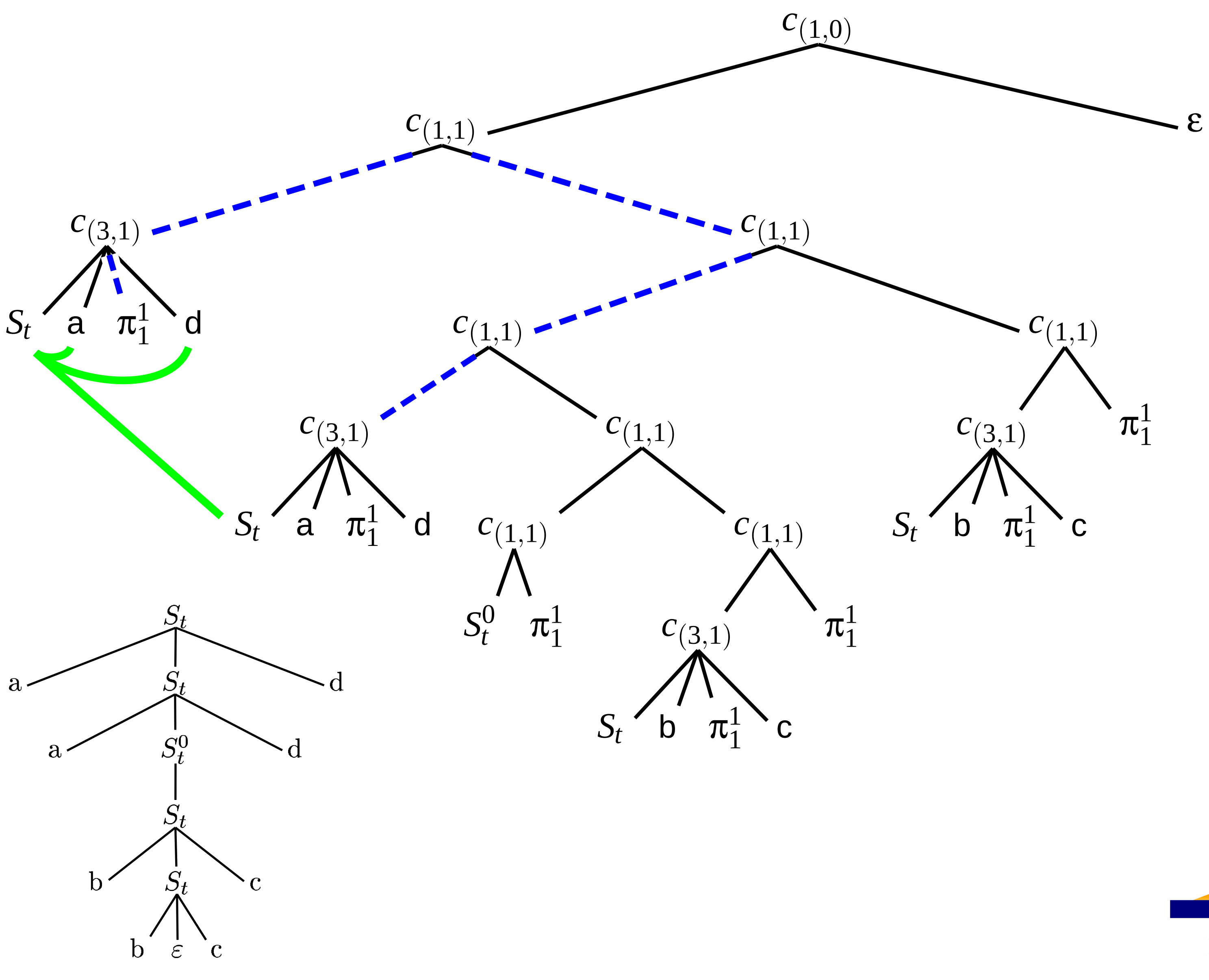




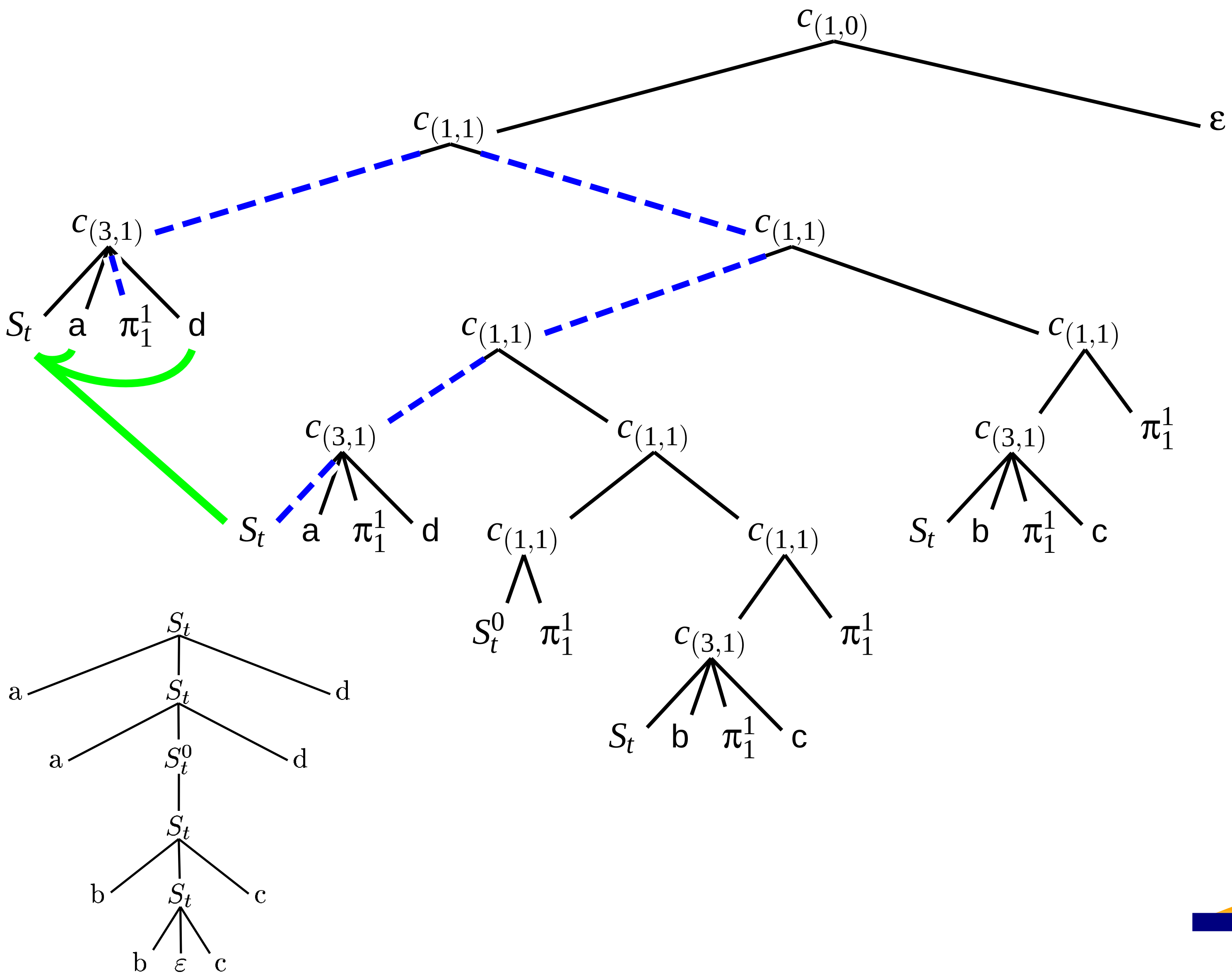
Reconstructing the Intended Structures



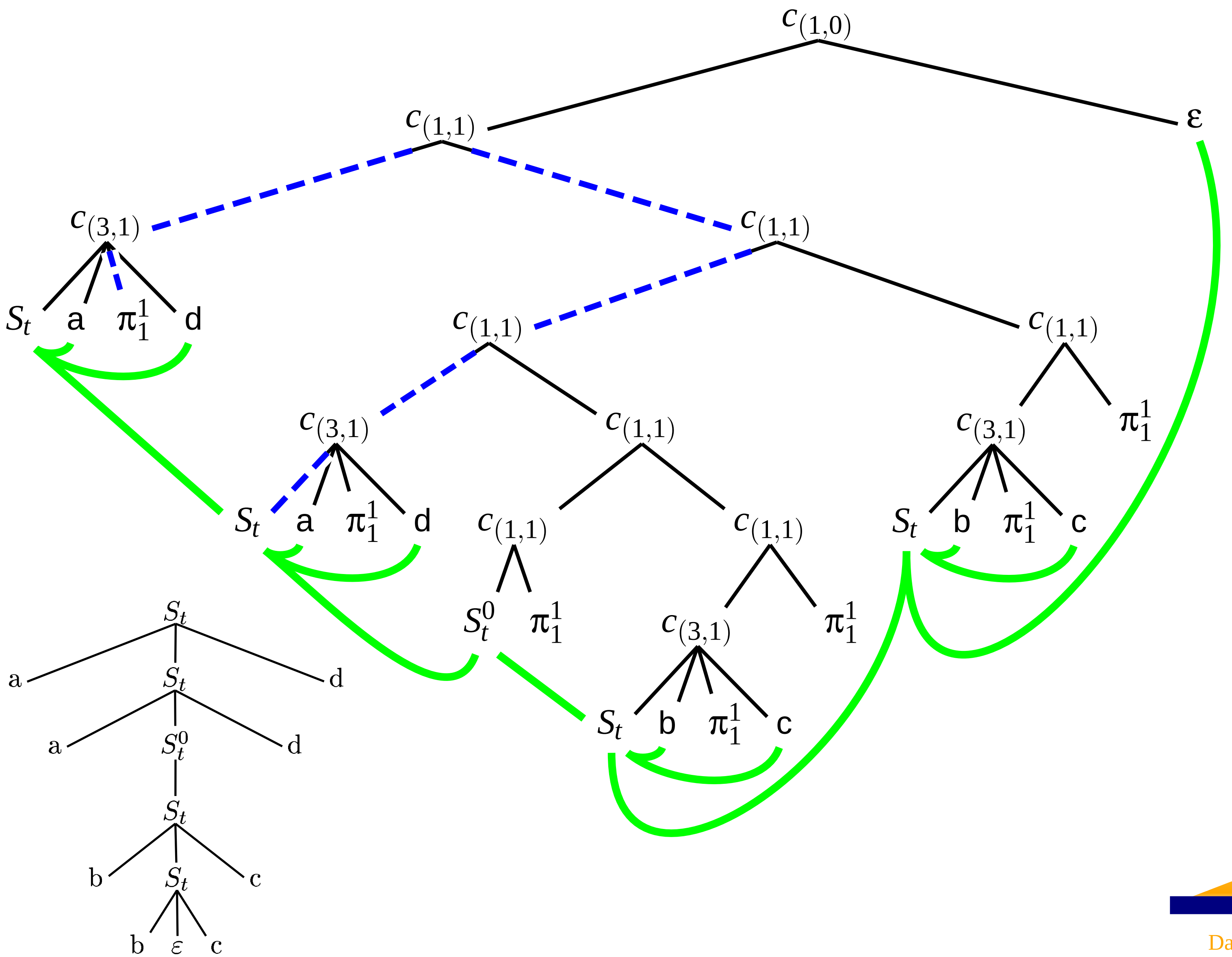
Reconstructing the Intended Structures



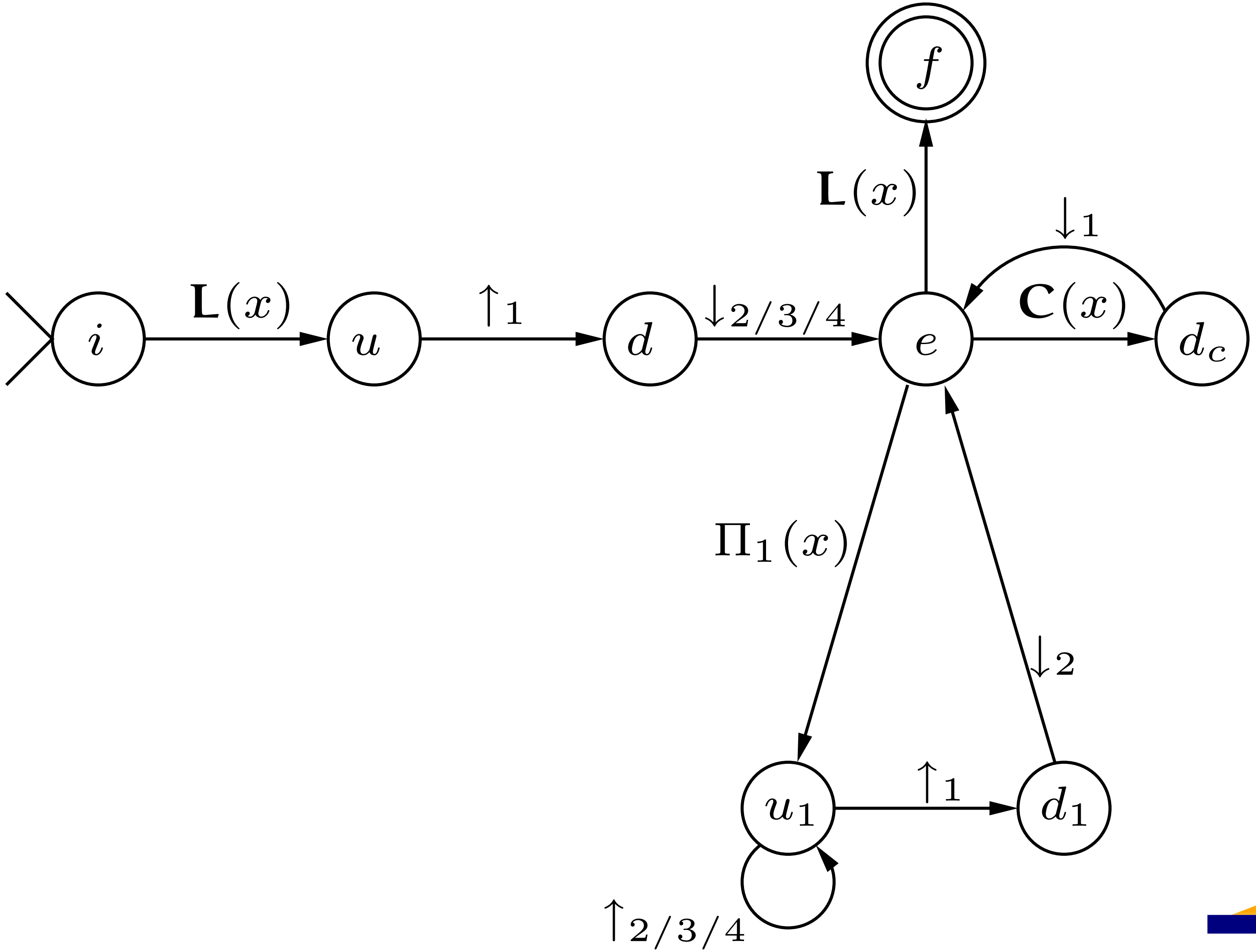
Reconstructing the Intended Structures



Reconstructing the Intended Structures



The tree-walking automaton A



The walking language $W_{\blacktriangleleft}$

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

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

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

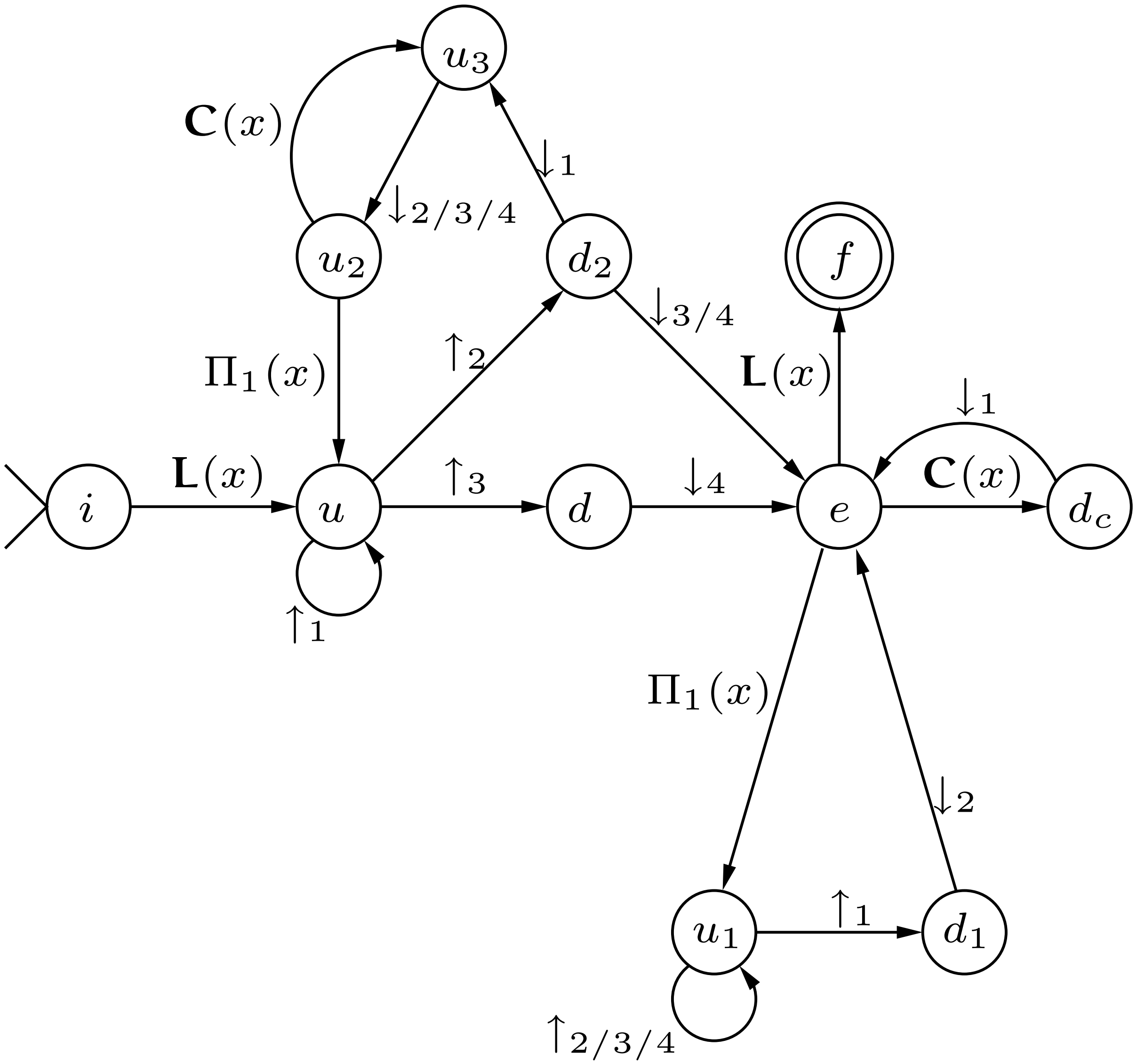
$W_{\blacktriangleleft}$ can be inductively translated into an MSO-formula *trans*$_{W_{\blacktriangleleft}}(x,y)$.

Logical Reconstruction of Precedence: $x \lhd y$

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

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

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

$$\varphi = \varphi_{\mathfrak{A}_{\mathcal{G}'}}$$

$$\psi = \mathsf{L}(x)$$

$$\theta_{\blacktriangleleft} = \mathsf{trans}_{W_{\blacktriangleleft}}(x, y)$$

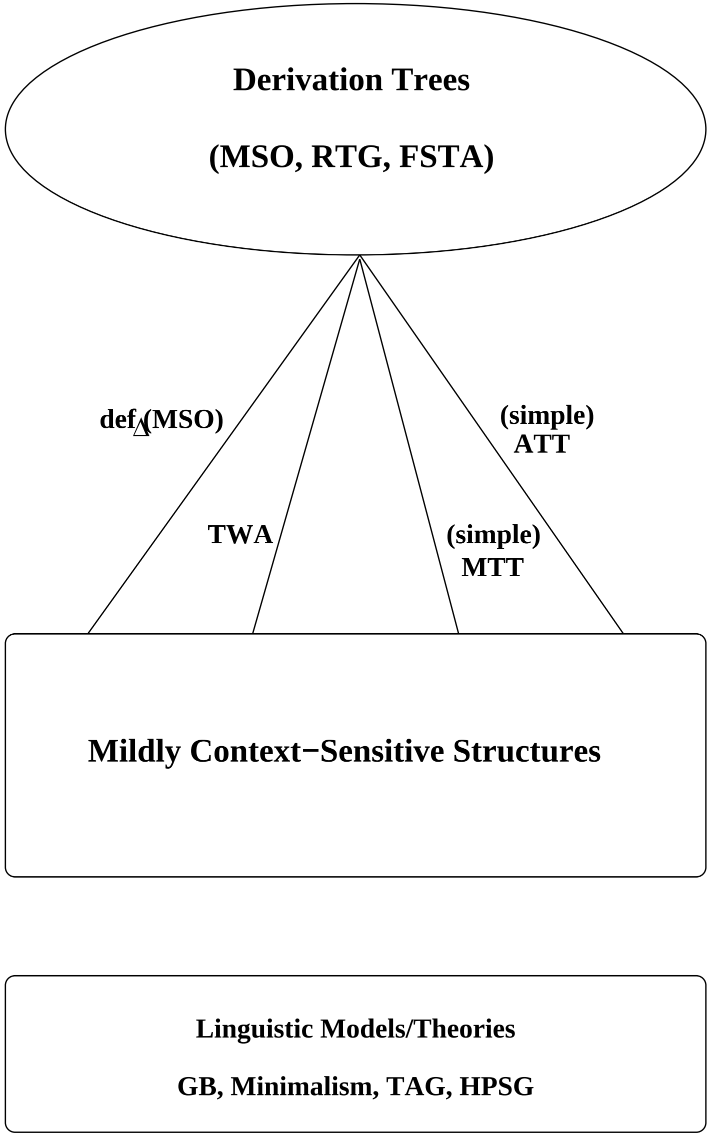
$$\theta_{\blacktriangleleft^\star} = (\forall X)[\blacktriangleleft\textit{-closed}(X) \vee x \in X \ \rightarrow \ y \in X]$$

$$\theta_{\blacktriangleleft^+} = x \blacktriangleleft^\star y \vee x \not\approx y$$

$$\theta_{\vartriangleleft} = (\exists u, v)[u \blacktriangleleft^\star x \wedge v \blacktriangleleft^\star y \wedge \mathsf{trans}_{W_{\vartriangleleft}}(u, v)]$$

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

Conclusion


Derivation Trees
(MSO, RTG, FSTA)
defΔ(MSO)
TWA
(simple)
MTT
(simple)
ATT
Mildly Context−Sensitive Structures
Linguistic Models/Theories
GB, Minimalism, TAG, HPSG

Outlook

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, ...

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

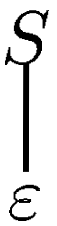
$\mathcal{A}$ is a finite set of auxiliary trees

Def'n from Joshi and Schabes 1997

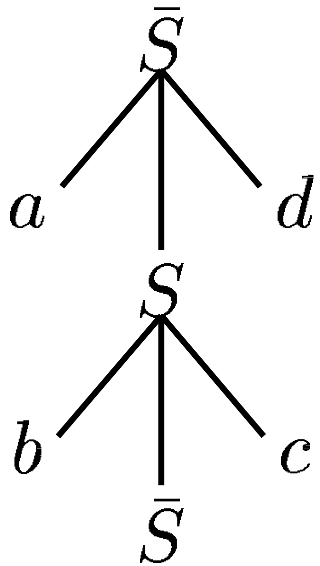
TAG for $a^n b^n c^n d^n$

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

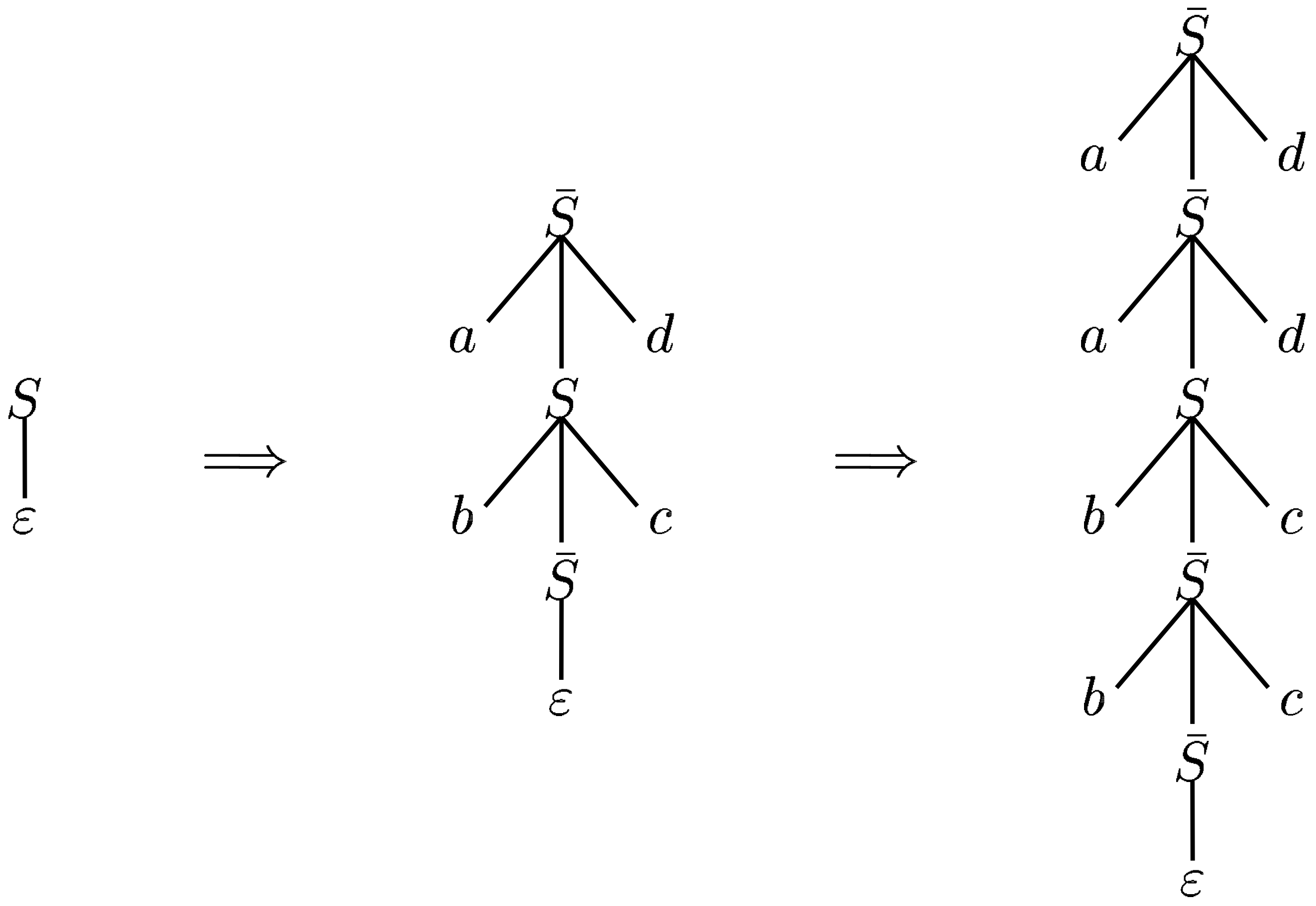
$\alpha =$



$\beta =$



An example derivation



back to TAGs

Lifting

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;

back to Lifting

Lifting

Suppose that Σ is a ranked alphabet. The *derived* $\mathbb{N}$-sorted alphabet Σ^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 ith projection symbol of sort n;

back to Lifting

Lifting

Suppose that Σ is a ranked alphabet. The *derived* $\mathbb{N}$-sorted alphabet Σ^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

Lifting

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

$$\Sigma_0^L = \Sigma_0'$$

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

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

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

back to Lifting

Model-Theoretic Interpretation

Basic Idea (Rabin 1965)

Obtaining a structure $\mathbb{B} = \langle B, \mathcal{Q} \rangle$ from a structure $\mathbb{A} = \langle A, \mathcal{R} \rangle$ where $\mathcal{R}$ and $\mathcal{Q}$ 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 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".

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Regular Tree-Node Relations

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

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

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.

back to A

Reflexive transitive closure in *MSO*

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

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

Walking language ⇒ *MSO*-formula

$$trans_{\emptyset}(x,y) = \bot$$

$$trans_{\downarrow_i}(x,y) = edge_i(x,y)$$

$$trans_{\uparrow_i}(x,y) = edge_i(y,x)$$

$$trans_{\varphi(x)}(x,y) = \varphi(x) \wedge x = y$$

$$trans_{W_1 \cup W_2}(x,y) = trans_{W_1}(x,y) \vee trans_{W_2}(x,y)$$

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

$$trans_{W^*}(x,y) = trans^*_W(x,y)$$

$$trans^*_W(x,y) = \forall X\,(\forall v,w\,(v \in X \wedge trans_W(v,w) \;\rightarrow\; w \in X) \wedge x \in X \;\rightarrow\; y \in X)$$

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

back to W