An Introduction to
Mildly Context-Sensitive Grammar Formalisms

— *Tree Adjoining Grammars* —

Gerhard Jäger & Jens Michaelis

Universität Potsdam

{jaeger,michael}@ling.uni-potsdam.de

Mild context-sensitivity (Joshi 1985)

- a concept motivated by the intention of characterizing a narrow class of formal grammars which are “only slightly more powerful than CFGs,” and which nevertheless allow for descriptions of natural languages in a linguistically significant way.

Mild context-sensitivity (Joshi 1985)

- a concept motivated by the intention of characterizing a narrow class of formal grammars which are "only slightly more powerful than CFGs," and which nevertheless allow for descriptions of natural languages in a linguistically significant way.

  According to Joshi (1985, p. 225) a *mildly context-sensitive language*, L, has to fulfil three criteria, to be understood as a "rough characterization." Somewhat paraphrased, these are:

Mild context-sensitivity (Joshi 1985)

- a concept motivated by the intention of characterizing a narrow class of formal grammars which are "only slightly more powerful than CFGs," and which nevertheless allow for descriptions of natural languages in a linguistically significant way.

  According to Joshi (1985, p. 225) a *mildly context-sensitive language*, L, has to fulfil three criteria, to be understood as a "rough characterization." Somewhat paraphrased, these are:

  (1) the parsing problem for L is solvable in polynomial time,

  (2) L has the constant growth property, and

  (3) there is a finite upper bound for L limiting the number of different instantiations of factorized cross-serial dependencies occurring in a sentence of L.

Mild context-sensitivity

- A collection of mildly context-sensitive grammar (MCSG) formalisms is presented in Joshi et al. 1991:

  - tree adjoining grammars (TAGs) (Joshi et al. 1975; Joshi 1985)
  - (restricted) combinatory categorial grammars (CCGs) (as formalized e.g. in Weir & Joshi 1988 in accordance with the CCG-version developed in Steedman 1987, 1990)
  - linear indexed grammars (LIGs) as they arise from Gazdar 1988
  - head grammars (HGs) (Pollard 1984)
  - multicomponent TAGs (MCTAGs) (Joshi 1987; Vijay-Shanker et al. 1987) as a generalization of TAGs
  - linear context-free rewriting systems (LCFRSs) (Vijay-Shanker et al. 1987) as a generalization of HGs, or, likewise, as a restriction of generalized CFGs (GCFGs) (Pollard 1984)

Mild context-sensitivity

- TAGs, CCGs, LIGs and HGs are weakly equivalent (see e.g. Vijay–Shanker & Weir 1994)
- MCTAGs and LCFRSs are weakly equivalent (Weir 1988)

Mild context-sensitivity

- TAGs, CCGs, LIGs and HGs are weakly equivalent (see e.g. Vijay–Shanker & Weir 1994) [a,b]
- MCTAGs and LCFRSs are weakly equivalent (Weir 1988) [c]

[a] The weak equivalence to LIGs, CCGs and HGs holds for *TAGs with local constraints (on tree adjoining)* as formally introduced e.g. in Vijay–Shanker & Joshi 1985, following a suggestion in Joshi et al. 1975, and capturing the intended use of *local constraints (on adjoining)* of the kind proposed in Joshi 1985. The class of TAGs with local constraints properly extends the strong as well as the weak generative capacity of the class of TAGs without such constraints.

[b] Note also that HGs as defined e.g. in Vijay–Shanker & Weir 1994 provide a modified version of HGs as originally defined in Pollard 1984. In terms of weak equivalence, HGs of this modified type subsume HGs of the original type, and vice versa. Corresponding proofs can be found in Vijay–Shanker et al. 1986 and Seki et al. 1991, respectively.

[c] More precisely, MCTAGs in their *set-local* variant, i.e. MCTAGs which, during the course of a derivation, allow the members of a derived sequence of auxiliary trees to be (simultaneously) adjoined at distinct nodes to the members of a single elementary tree sequence (cf. Definition 2.7.1 in Weir 1988).

Finite labeled trees

$$t = \langle N_t , \lhd_t^* , \prec_t , \mathrm{label}_t \rangle$$

Finite labeled trees

$$t = \langle N_t , \lhd_t^* , \prec_t , \mathrm{label}_t \rangle$$

- $\langle N_t , \lhd_t^* , \prec_t \rangle$ a finite ordered tree :

Finite labeled trees

$$t = \langle N_t , \triangleleft_t^* , \prec_t , \mathrm{label}_t \rangle$$

- $\langle N_t , \triangleleft_t^* , \prec_t \rangle$ a finite ordered tree :

  N_t the finite set nodes

Finite labeled trees

$$t = \langle N_t , \lhd_t^* , \prec_t , label_t \rangle$$

- $\langle N_t , \lhd_t^* , \prec_t \rangle$ a finite ordered tree :

  N_t the finite set nodes

  $\lhd_t^*$ and $\prec_t$ the binary relations of dominance and precedence on N_t , respectively

Finite labeled trees

$$t = \langle N_t , \triangleleft_t^* , \prec_t , \mathrm{label}_t \rangle$$

- $\langle N_t , \triangleleft_t^* , \prec_t \rangle$ a finite ordered tree :

  N_t the finite set nodes

  $\triangleleft_t^*$ and $\prec_t$ the binary relations of dominance and precedence on N_t , respectively

  i.e., $\triangleleft_t^*$ is the reflexive-transitive closure of $\triangleleft_t$, the binary relation of immediate dominance on N_t

Finite labeled trees

$$t = \langle N_t , \triangleleft_t^* , \prec_t , \mathrm{label}_t \rangle$$

- $\langle N_t , \triangleleft_t^* , \prec_t \rangle$ a finite ordered tree :

  N_t the finite set nodes

  $\triangleleft_t^*$ and $\prec_t$ the binary relations of dominance and precedence on N_t , respectively

  i.e., $\triangleleft_t^*$ is the reflexive-transitive closure of $\triangleleft_t$, the binary relation of immediate dominance on N_t

- label_t the labeling (function) , a function from N_t into a set of labels.

Objects specified by a tree adjoining grammar

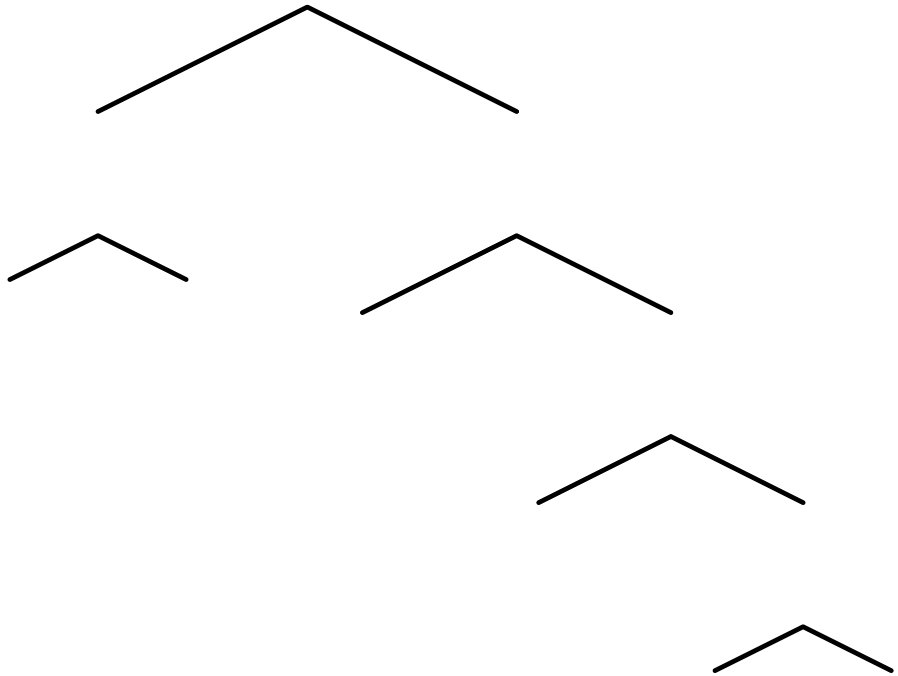
V_N a set of nonterminals

V_T a set of terminals

Objects specified by a tree adjoining grammar

V_N a set of nonterminals

V_T a set of terminals

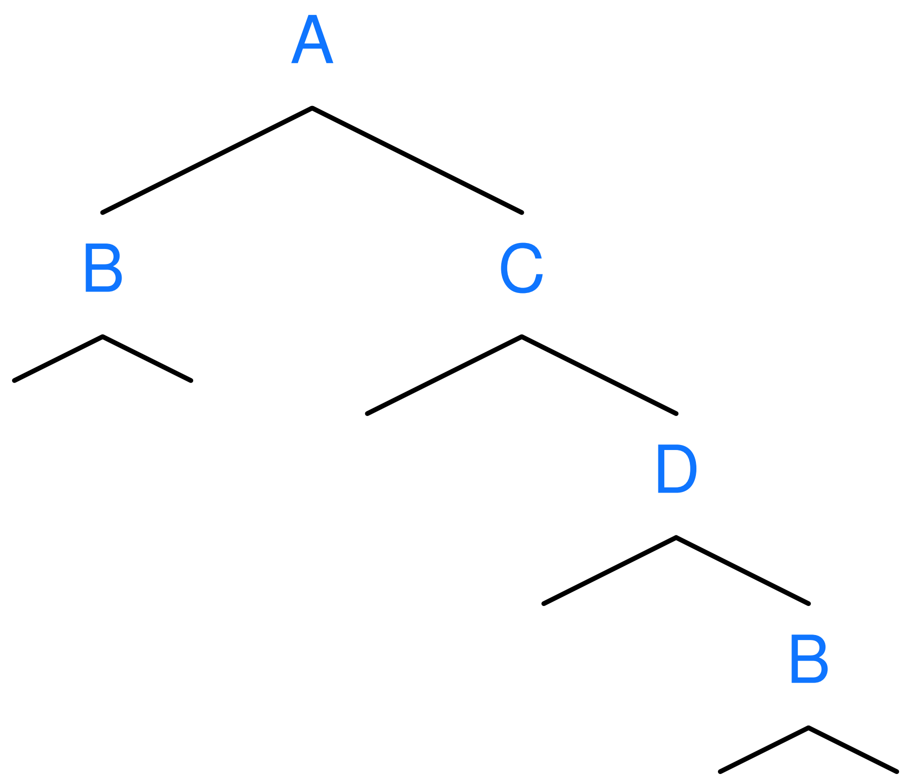


$$t = \langle N_t , \triangleleft_t^* , \prec_t , \text{label}_t \rangle$$

Objects specified by a tree adjoining grammar

V_N a set of nonterminals

V_T a set of terminals



$$t = \langle N_t , \triangleleft_t^* , \prec_t , \text{label}_t \rangle$$

$$\text{label}_t : \text{NonLeaves}_t \rightarrow V_N$$

Objects specified by a tree adjoining grammar

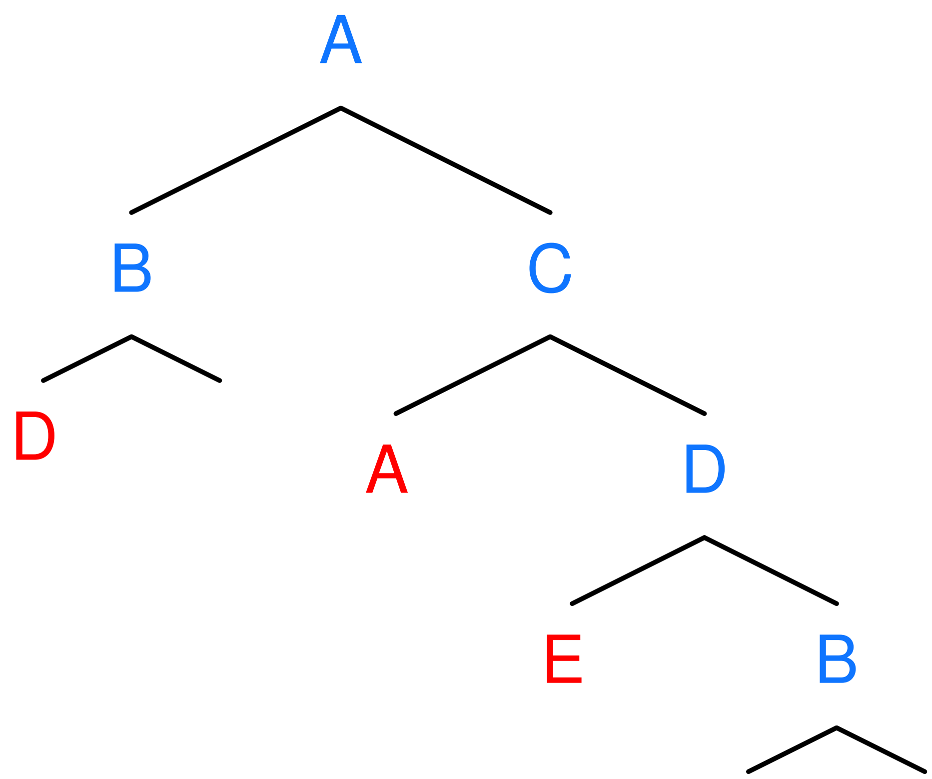
V_N a set of nonterminals

V_T a set of terminals

$$t = \langle N_t, \lhd_t^*, \prec_t, \mathrm{label}_t \rangle$$

$$\mathrm{label}_t : \mathrm{NonLeaves}_t \rightarrow V_N$$

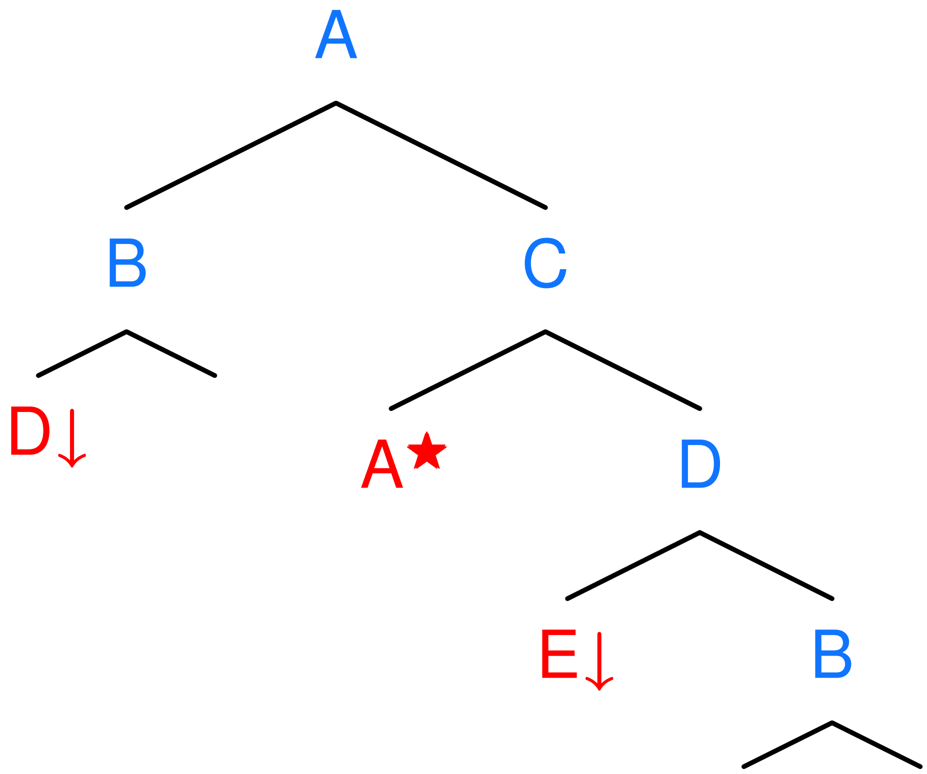
$$\mathrm{Leaves}_t \rightarrow V_N$$



Objects specified by a tree adjoining grammar

V_N a set of nonterminals

V_T a set of terminals



$$t = \langle N_t , \lhd_t^* , \prec_t , \text{label}_t \rangle$$

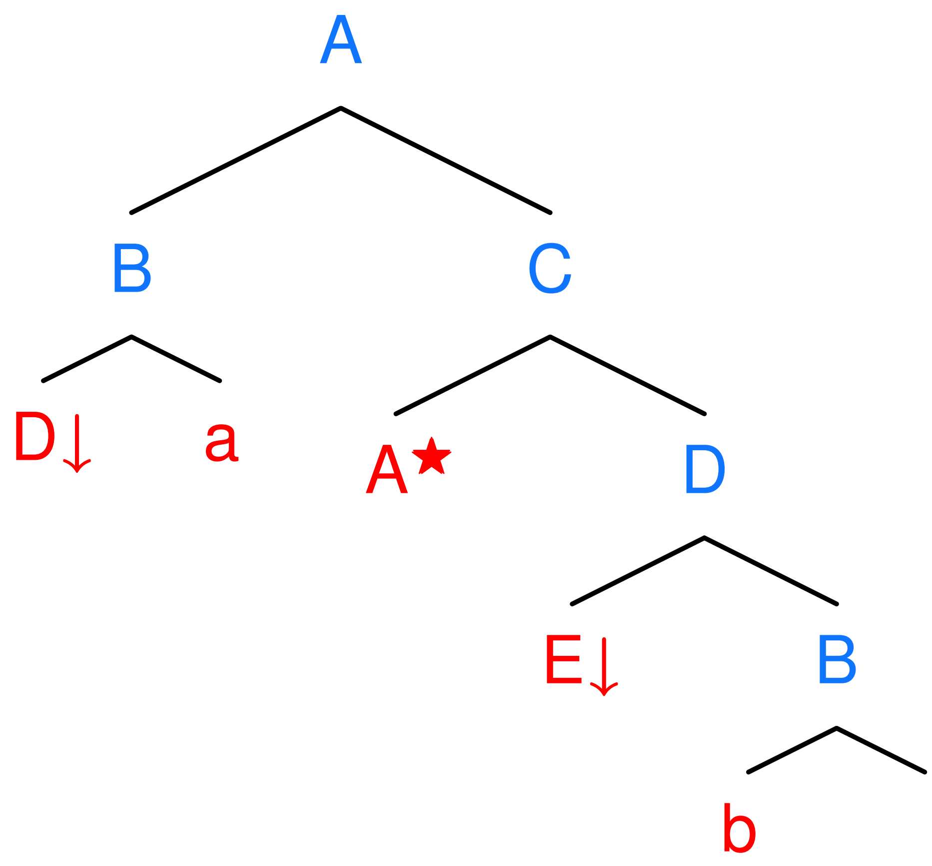
$$\text{label}_t : \text{NonLeaves}_t \longrightarrow V_N$$

$$\text{Leaves}_t \longrightarrow V_N \{\downarrow, \star\}$$

Objects specified by a tree adjoining grammar

V_N a set of nonterminals

V_T a set of terminals



$$t = \langle N_t , \lhd_t^* , \prec_t , \text{label}_t \rangle$$

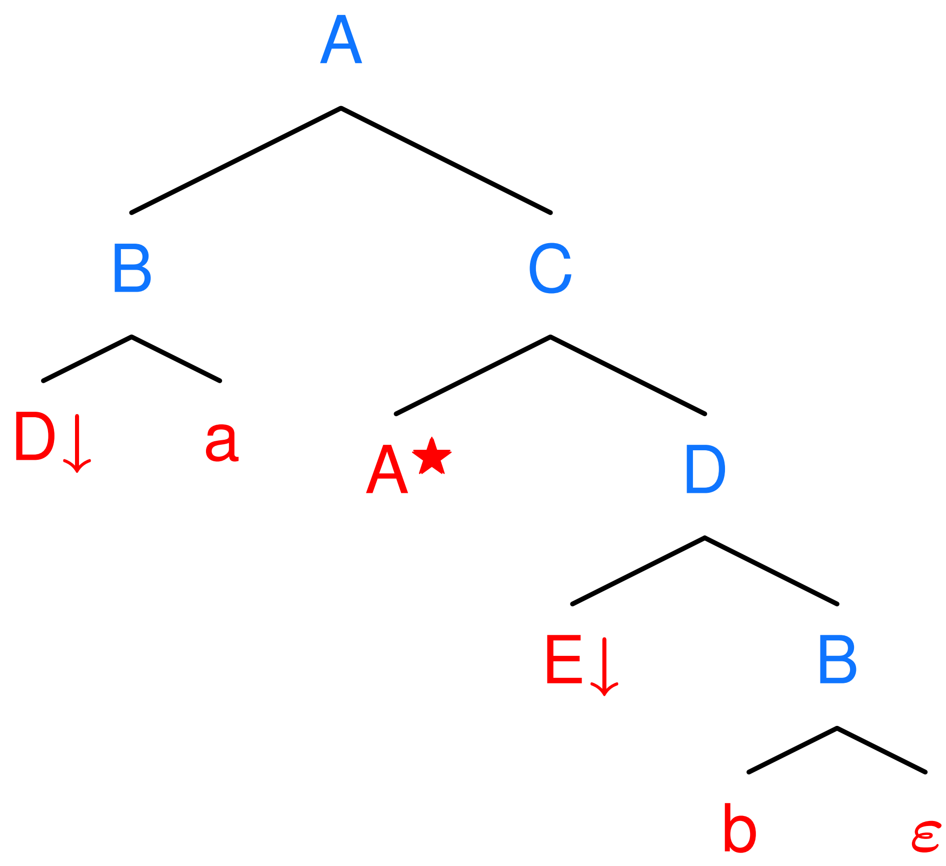
$$\text{label}_t : \text{NonLeaves}_t \longrightarrow V_N$$

$$\text{Leaves}_t \longrightarrow V_N \{\downarrow, ^\star\} \cup V_T$$

Objects specified by a tree adjoining grammar

V_N a set of nonterminals

V_T a set of terminals



$$t = \langle N_t , \triangleleft_t^* , \prec_t , \text{label}_t \rangle$$

$$\text{label}_t : \text{NonLeaves}_t \rightarrow V_N$$

$$\text{Leaves}_t \rightarrow V_N \{\downarrow, ^\star\} \cup V_T \cup \{\varepsilon\}$$

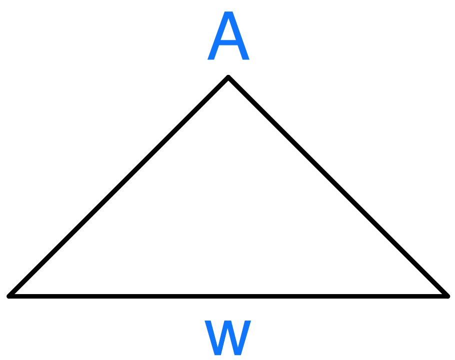
Deriving (labeled) trees by a tree adjoining grammar

- Labeled trees can be derived from others by applying the structure building operators, namely, substitution and adjoining.

Deriving (labeled) trees by a tree adjoining grammar

- Labeled trees can be derived from others by applying the structure building operators, namely, substitution and adjoining.

- Derivations start from initial trees.

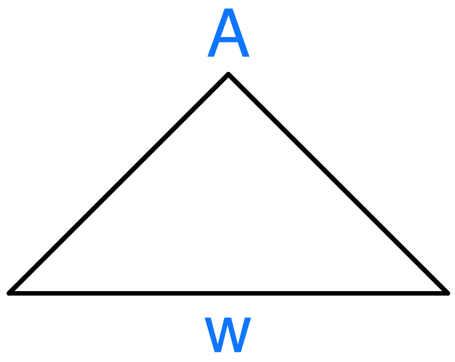


$A \in V_N$

$w \in \text{Strings}(V_N\{\downarrow\} \cup V_T)$

Deriving (labeled) trees by a tree adjoining grammar

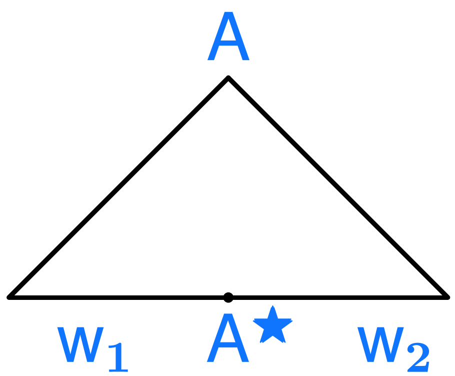
- Labeled trees can be derived from others by applying the structure building operators, namely, substitution and adjoining.
- Derivations start from initial trees.



$A \in V_N$

$w \in \text{Strings}(V_N\{\downarrow\} \cup V_T)$

- W.r.t. adjoining, auxiliary trees are of central importance.



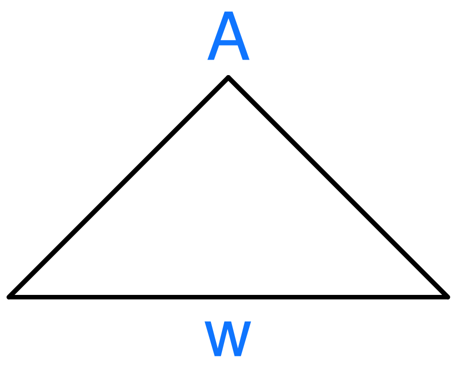
$A \in V_N$

$w_1, w_2 \in \text{Strings}(V_N\{\downarrow\} \cup V_T)$

Deriving (labeled) trees by a tree adjoining grammar

- Labeled trees can be derived from others by applying the structure building operators, namely, substitution and adjoining.

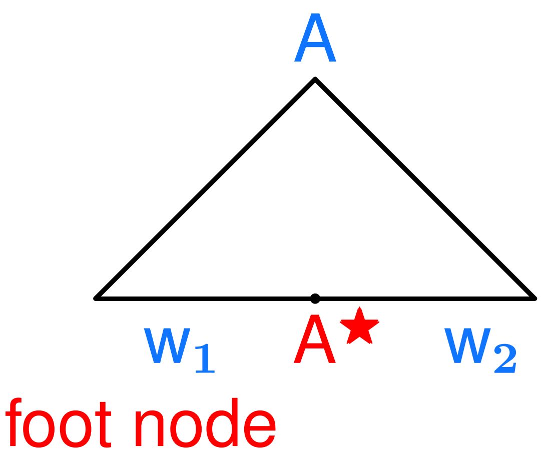
- Derivations start from initial trees.



$A \in V_N$

$w \in \text{Strings}(V_N\{\downarrow\} \cup V_T)$

- W.r.t. adjoining, auxiliary trees are of central importance.



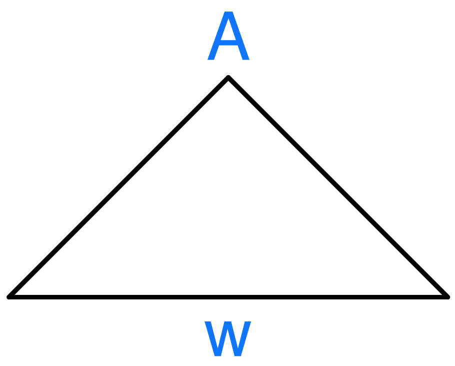
$A \in V_N$

$w_1, w_2 \in \text{Strings}(V_N\{\downarrow\} \cup V_T)$

Deriving (labeled) trees by a tree adjoining grammar

- Labeled trees can be derived from others by applying the structure building operators, namely, substitution and adjoining.

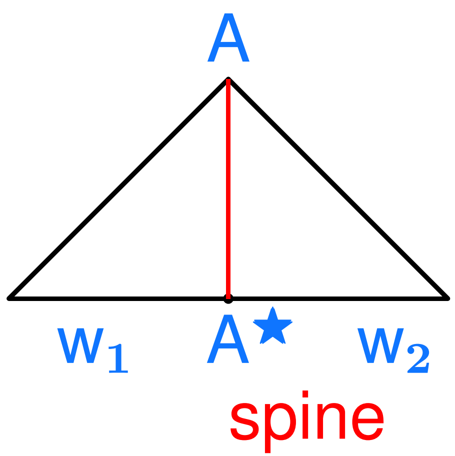
- Derivations start from initial trees.



$A \in V_N$

$w \in \text{Strings}(V_N\{\downarrow\} \cup V_T)$

- W.r.t. adjoining, auxiliary trees are of central importance.



$A \in V_N$

$w_1, w_2 \in \text{Strings}(V_N\{\downarrow\} \cup V_T)$

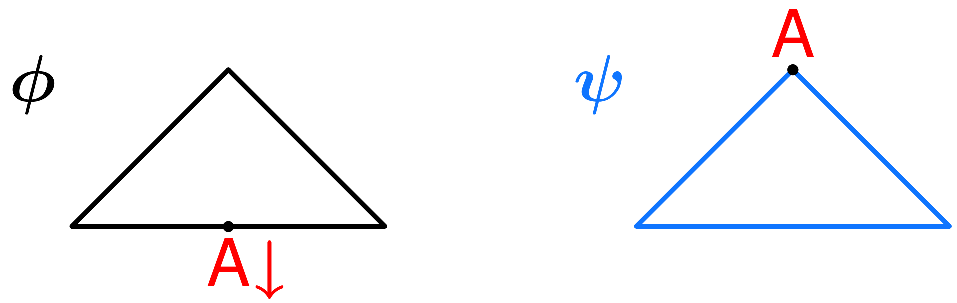
Structure building operators

$$\text{substitution} : \text{Trees}(V) \times \text{Trees}(V) \xrightarrow{\text{part}} 2^{\text{Trees}(V)}$$

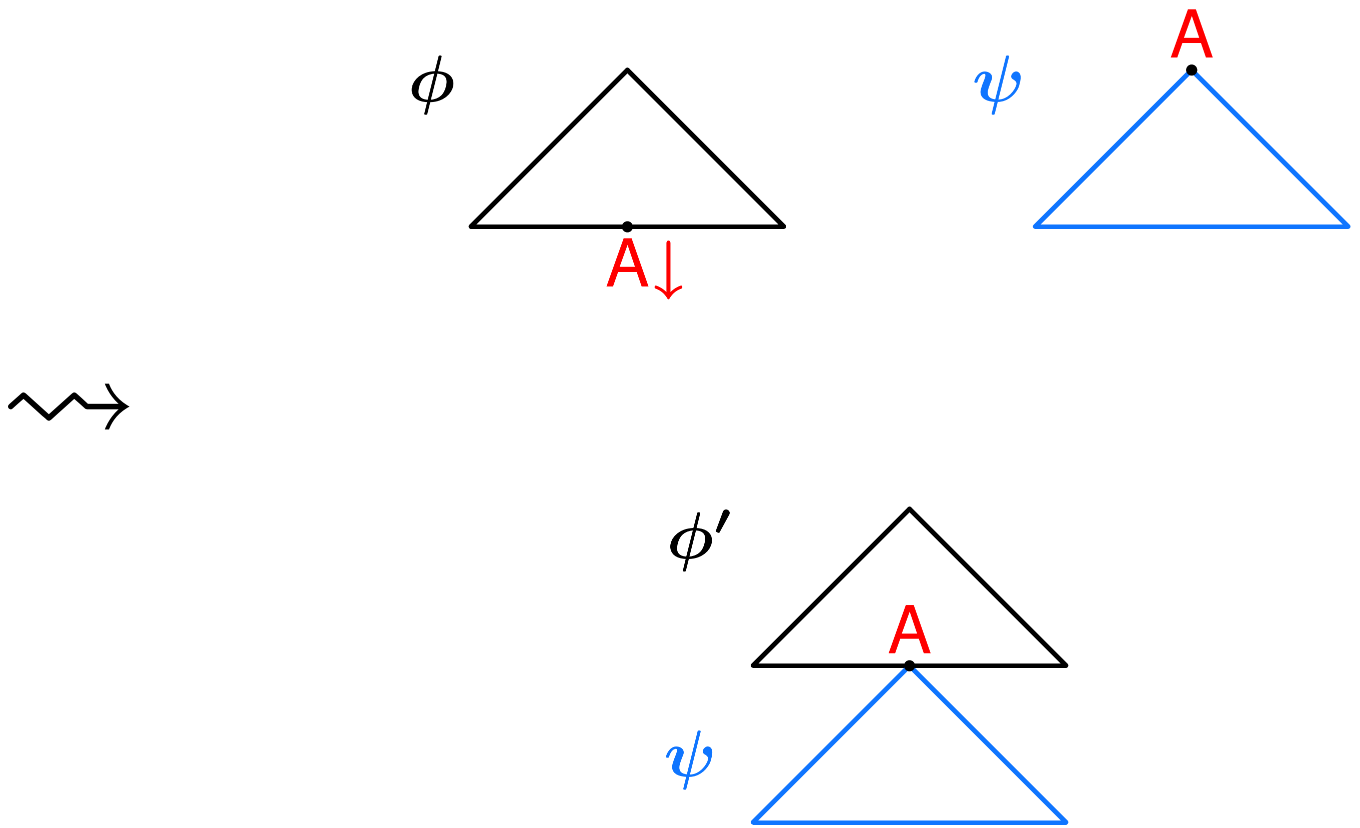
$\langle \phi, \psi \rangle \in \text{Domain(substitution)} :\Longleftrightarrow$

- ϕ has a leaf labeled $A\downarrow$ for some $A \in V_N$
- ψ's root is labeled A

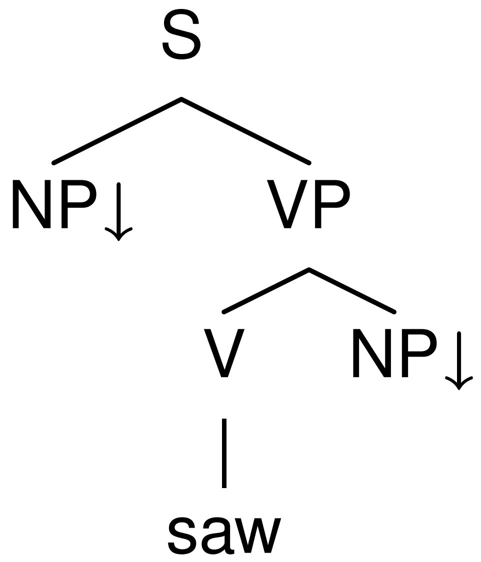
$$\text{substitution} : \text{Trees}(V) \times \text{Trees}(V) \xrightarrow{\text{part}} 2^{\text{Trees}(V)}$$

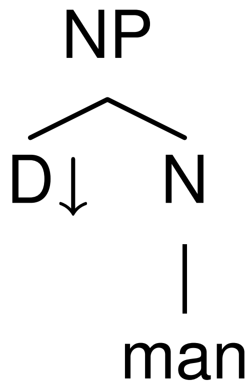


$$\text{substitution} : \text{Trees}(V) \times \text{Trees}(V) \xrightarrow{\text{part}} 2^{\text{Trees}(V)}$$

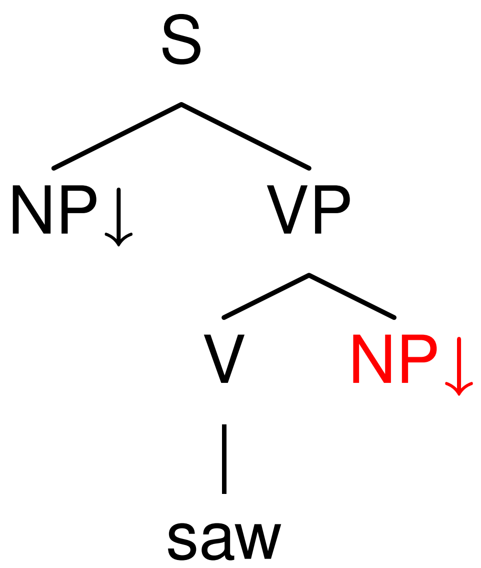


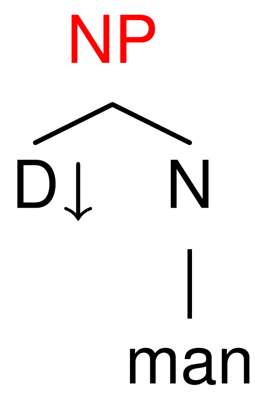
substitution



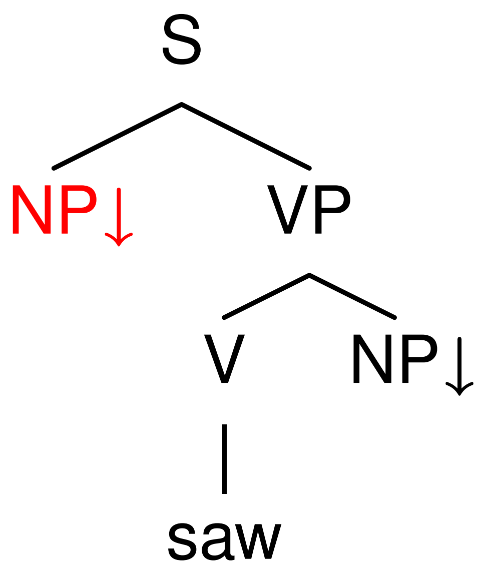


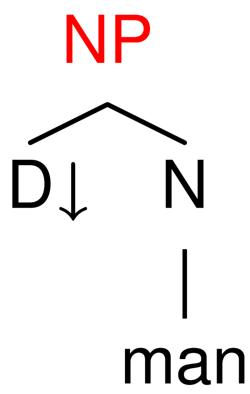
substitution



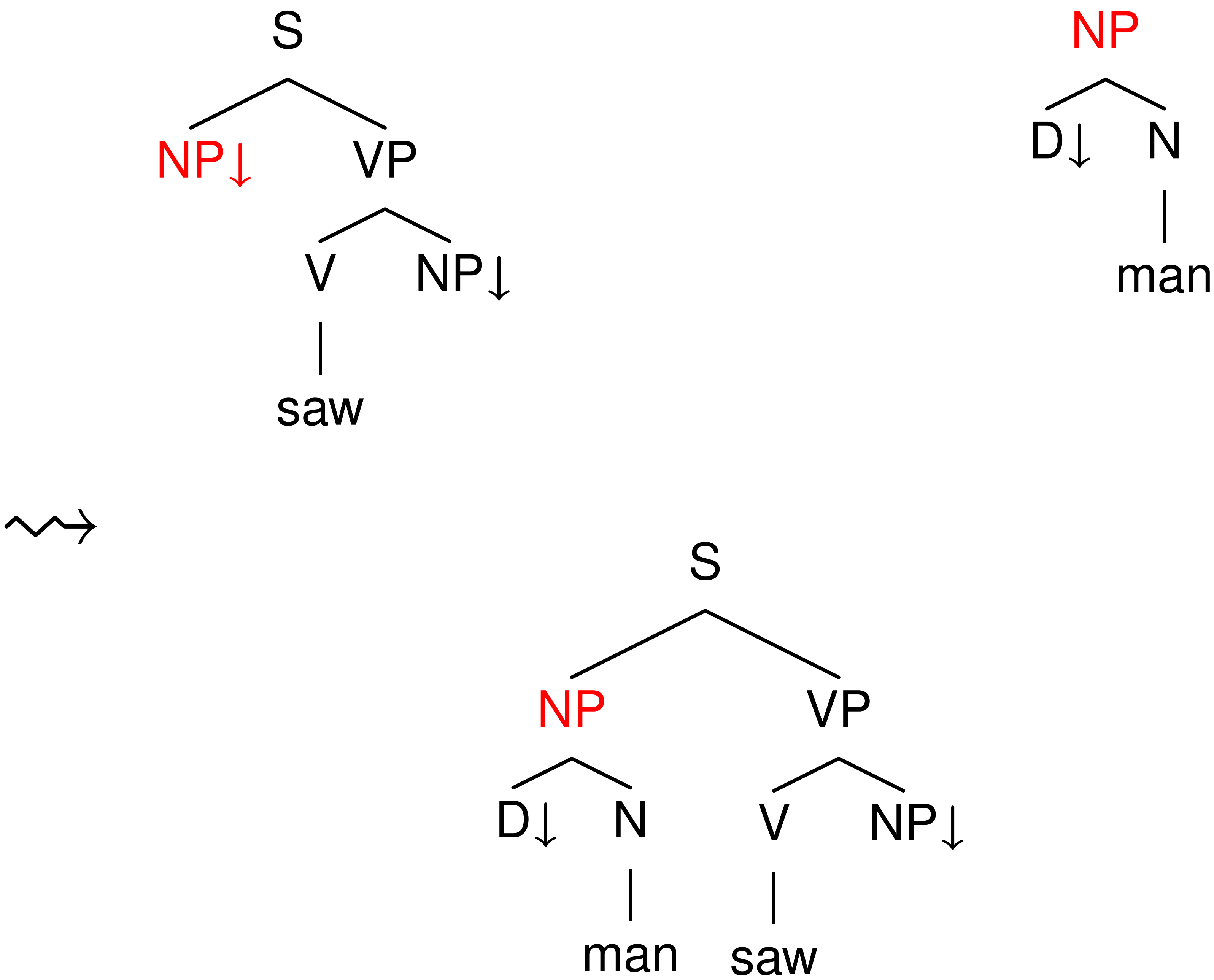


substitution





substitution



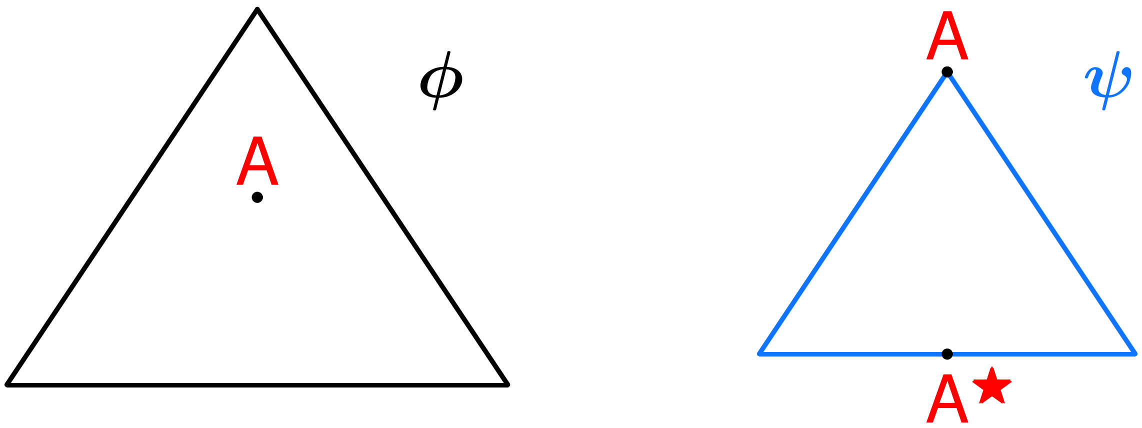
Structure building operators

$$\text{adjoining} : \text{Trees}(V) \times \text{Trees}(V) \xrightarrow{\text{part}} 2^{\text{Trees}(V)}$$

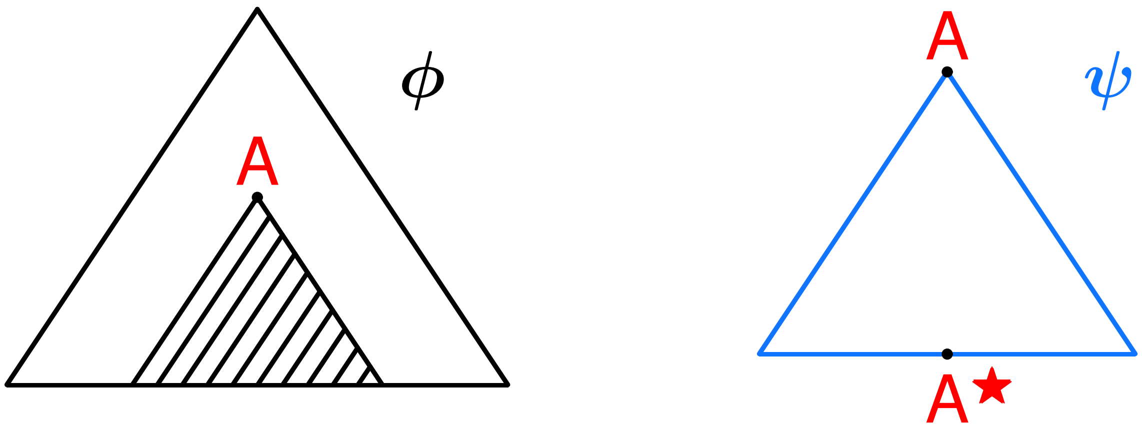
$$\langle \phi , \psi \rangle \in \text{Domain}(\text{adjoining}) \;:\Longleftrightarrow$$

- ϕ has a node labeled A for some $A \in V_N$
- ψ's root is labeled A and ψ has a leaf labeled $A^{\star}$

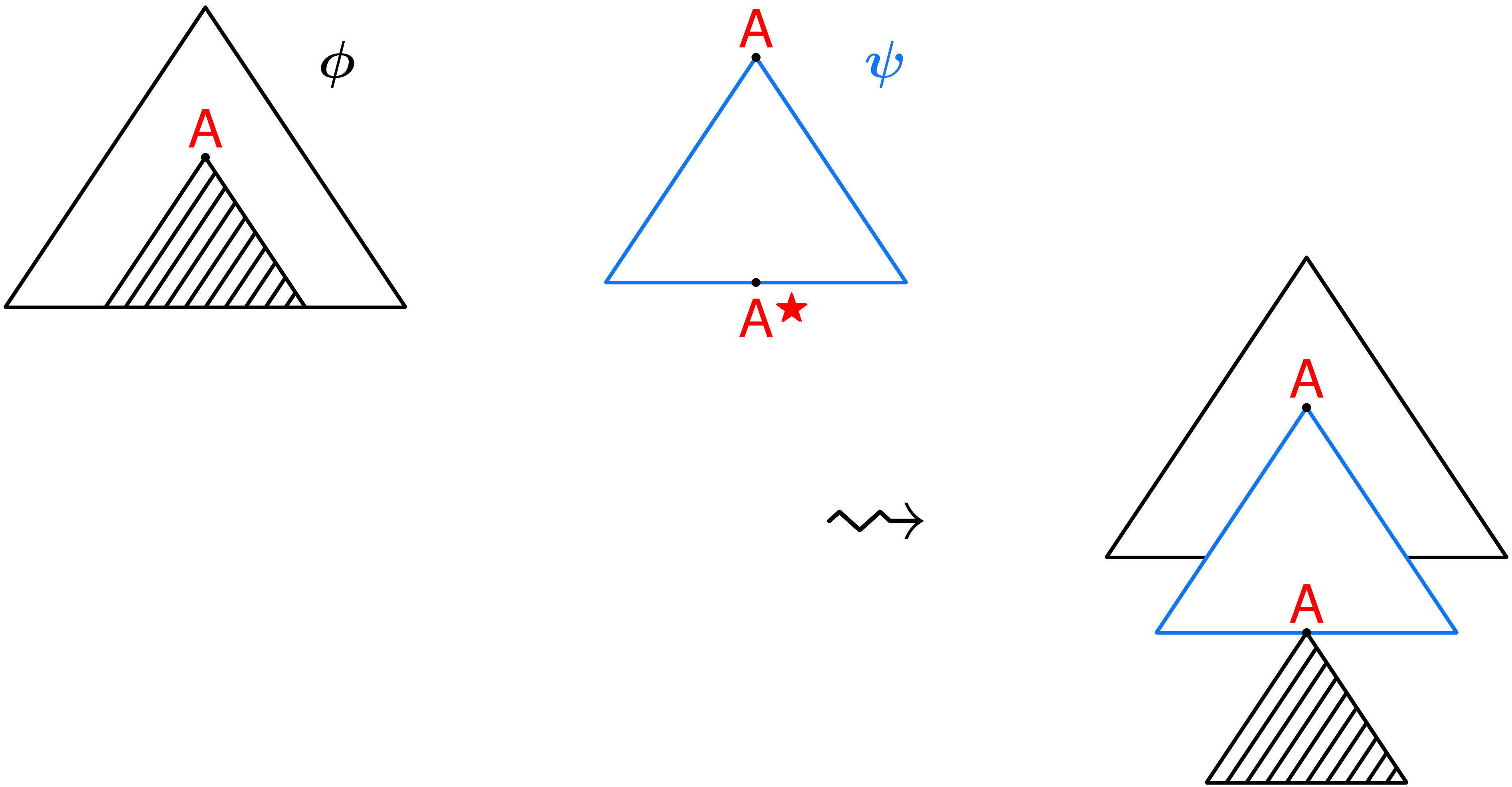
adjoining : $\text{Trees}(V) \times \text{Trees}(V) \xrightarrow{\text{part}} 2^{\text{Trees}(V)}$



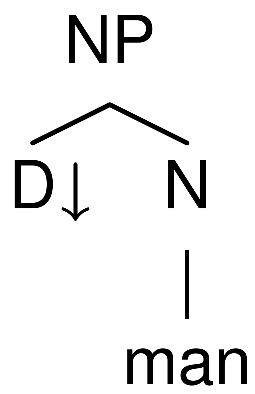
$$\text{adjoining} : \text{Trees}(V) \times \text{Trees}(V) \xrightarrow{\text{part}} 2^{\text{Trees}(V)}$$

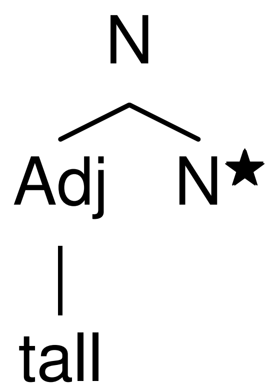


$$\text{adjoining} : \text{Trees}(V) \times \text{Trees}(V) \xrightarrow{\text{part}} 2^{\text{Trees}(V)}$$

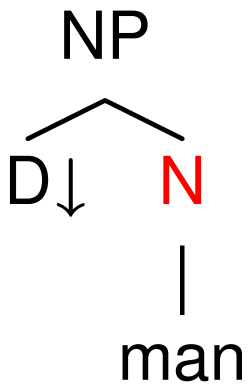


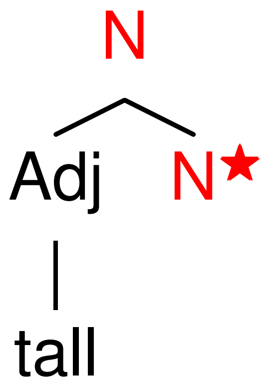
adjoining



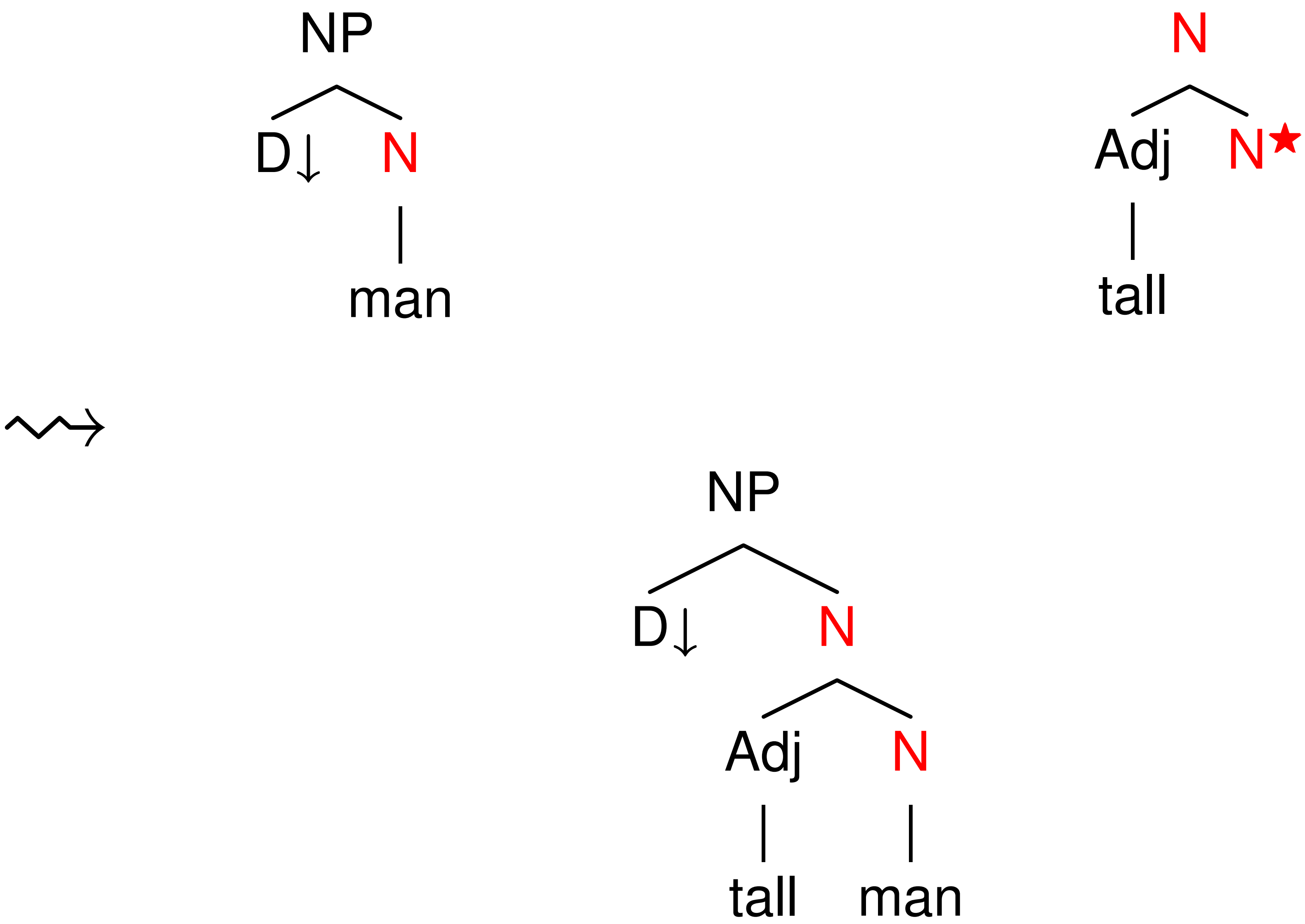


adjoining

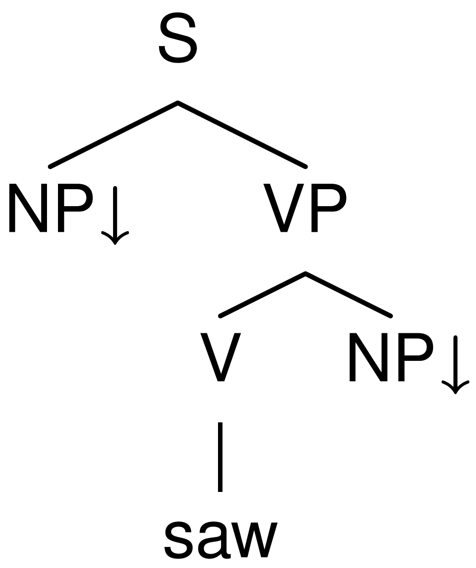


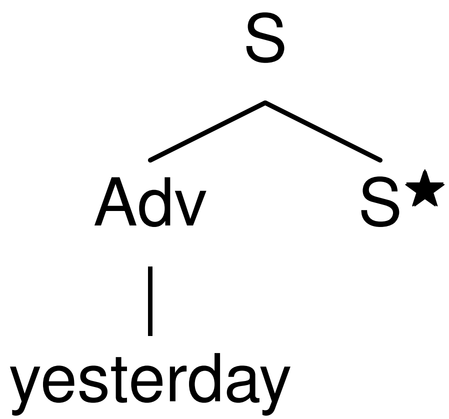


adjoining

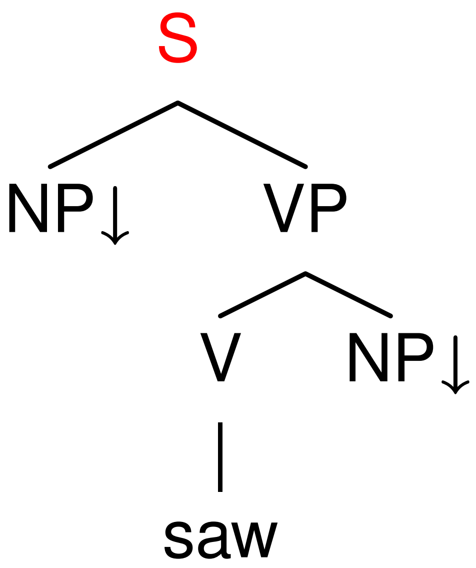


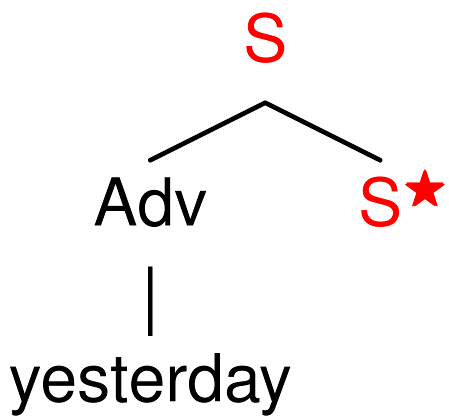
adjoining



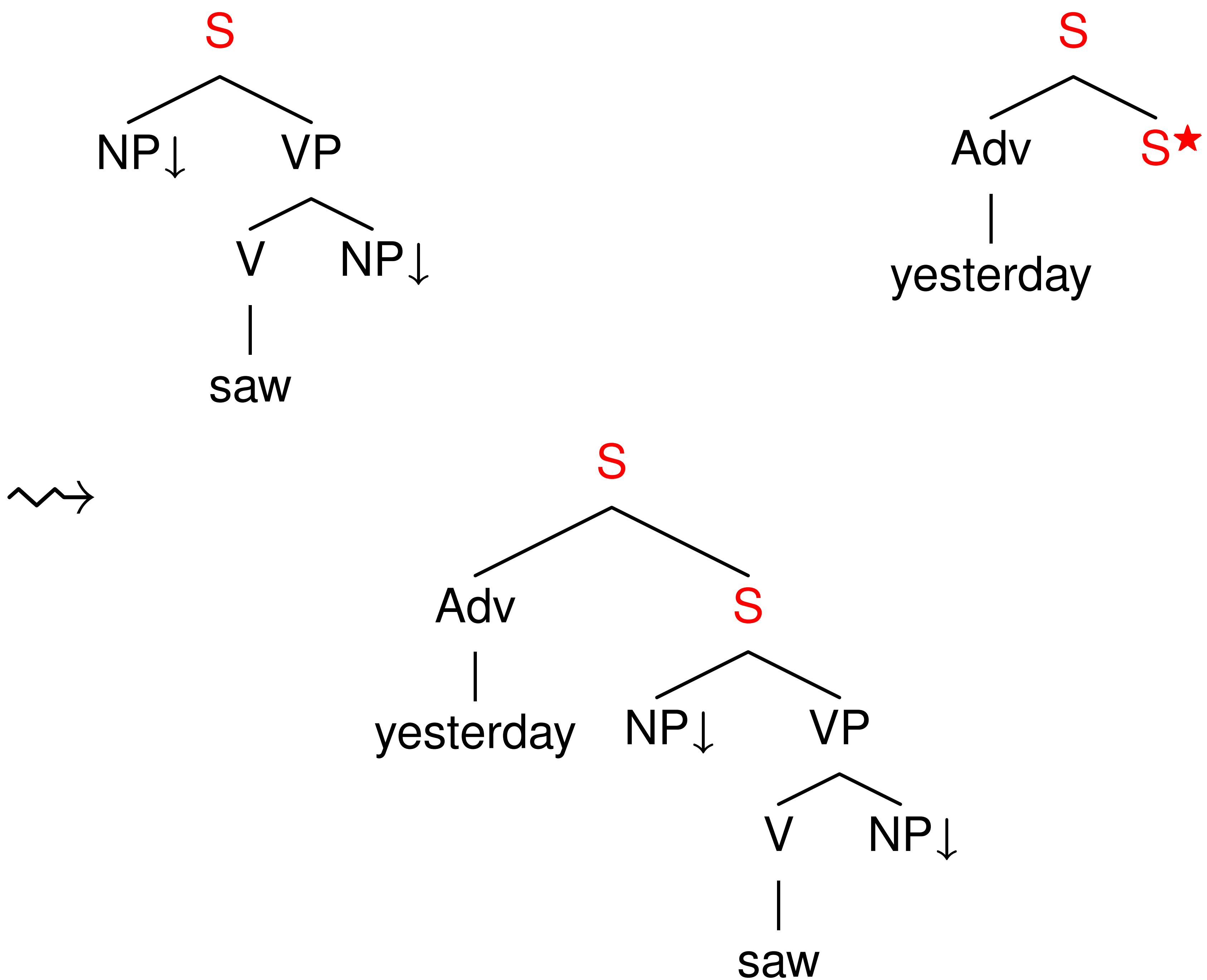


adjoining





adjoining



Tree adjoining grammars (TAGs)

$$G = \langle V_N , V_T , T_{Ini} , T_{Aux} , S \rangle$$

Tree adjoining grammars (TAGs)

$$G = \langle V_N , V_T , T_{Ini} , T_{Aux} , S \rangle$$

- V_N a set of nonterminals

$$G = \langle V_N , V_T , T_{Ini} , T_{Aux} , S \rangle$$

- V_N a set of nonterminals
- V_T a set of terminals

Tree adjoining grammars (TAGs)

$$G = \langle V_N , V_T , T_{Ini} , T_{Aux} , S \rangle$$

- V_N a set of nonterminals
- V_T a set of terminals
- T_{Ini} a finite set of initial trees

Tree adjoining grammars (TAGs)

$$G = \langle V_N , V_T , T_{Ini} , T_{Aux} , S \rangle$$

- V_N a set of nonterminals
- V_T a set of terminals
- T_{Ini} a finite set of initial trees
- T_{Aux} a finite set of auxiliary trees

Tree adjoining grammars (TAGs)

$$G = \langle V_N, V_T, T_{Ini}, T_{Aux}, S \rangle$$

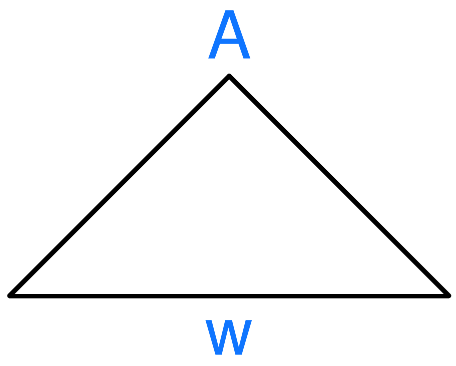
- V_N a set of nonterminals
- V_T a set of terminals
- T_{Ini} a finite set of initial trees
- T_{Aux} a finite set of auxiliary trees
- $S \in V_N$ a distinguished nonterminal (the start symbol)

Elementary trees: initial vs. auxiliary

Elementary trees: initial vs. auxiliary

$t \in T_{Ini}$

t is a finite labeled tree $\langle N_t , \lhd_t^* , \prec_t , label_t \rangle$ such that



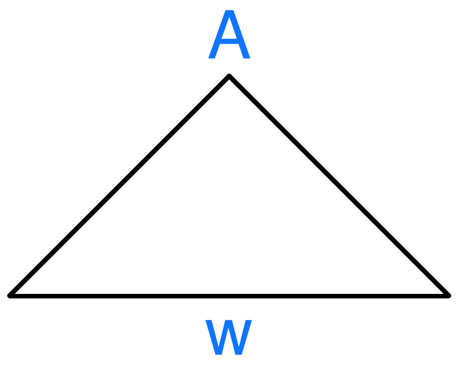
$A \in V_N$

$w \in Strings(V_N\{\downarrow\} \cup V_T)$

Elementary trees: initial vs. auxiliary

$t \in T_{Ini}$

t is a finite labeled tree $\langle N_t, \lhd_t^*, \prec_t, label_t \rangle$ such that

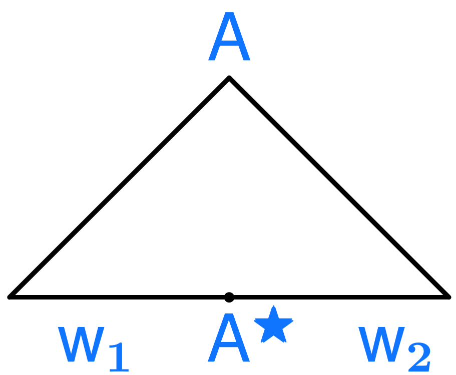


$A \in V_N$

$w \in Strings(V_N\{\downarrow\} \cup V_T)$

$t \in T_{Aux}$

t is a finite labeled tree $\langle N_t, \lhd_t^*, \prec_t, label_t \rangle$ such that

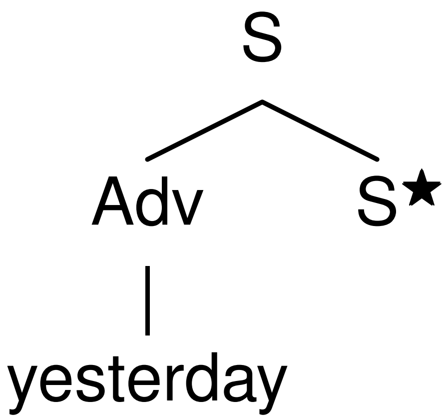


$A \in V_N$

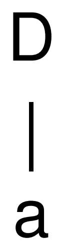
$w_1, w_2 \in Strings(V_N\{\downarrow\} \cup V_T)$

Elementary trees: examples

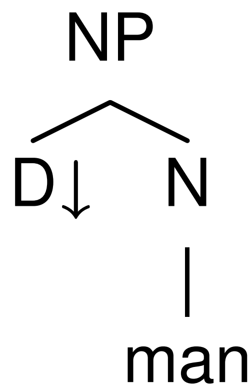
(β_{yest})



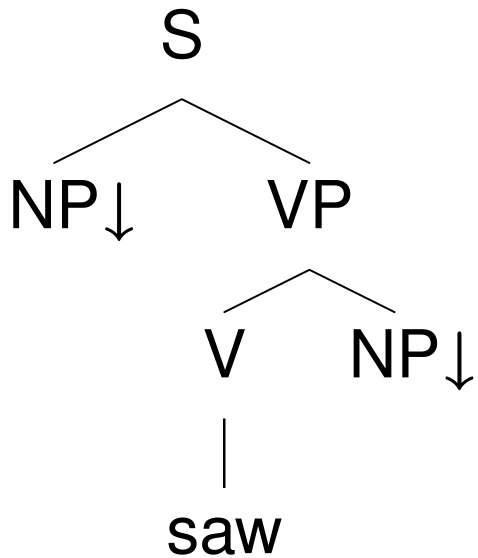
(α_{a})



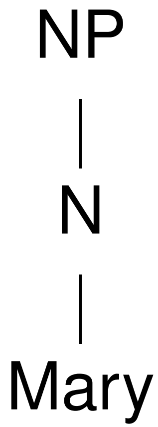
(α_{man})



(α_{saw})



(α_{Mary})



Closure(G) , the closure of a TAG $G = \langle V_N , V_T , T_{Ini} , T_{Aux} , S \rangle$,

is the closure of $T_{Ini} \cup T_{Aux}$ under finitely many applications of substitution and adjoining .

Tree adjoining languages

$\mathsf{Closure}(G)$, the closure of a TAG $G = \langle V_N, V_T, T_{Ini}, T_{Aux}, S \rangle$,

is the closure of $T_{Ini} \cup T_{Aux}$ under finitely many applications of substitution and adjoining .

$t \in \mathsf{Closure}(G)$ is complete $:\Longleftrightarrow$

t's root-label is S and $\mathsf{yield}(t) \in \mathsf{Strings}(V_T)$.

Tree adjoining languages

Closure(G), the closure of a TAG $G = \langle V_N, V_T, T_{Ini}, T_{Aux}, S \rangle$,

is the closure of $T_{Ini} \cup T_{Aux}$ under finitely many applications of substitution and adjoining.

$t \in$ Closure(G) is complete $:\Longleftrightarrow$

t's root-label is S and $\text{yield}(t) \in \text{Strings}(V_T)$.

The tree and string language generated by G

$$T(G) = \{\, t \mid t \in \text{Closure}(G) \text{ and complete} \,\}$$

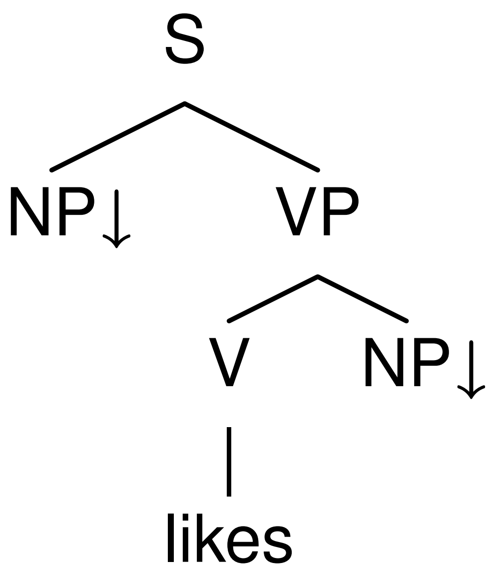
$$L(G) = \{\, \text{yield}(t) \mid t \in T(G) \,\}$$

Linguistic applications

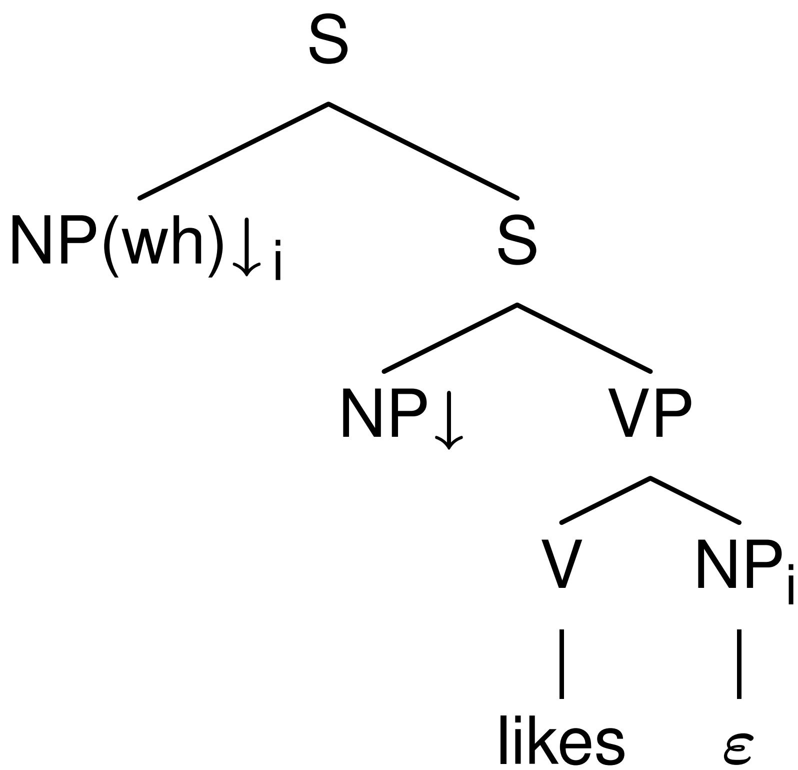
- Wh-movement
- Verbclusters

Linguistic applications: elementary trees for ‘likes’

(α_1)

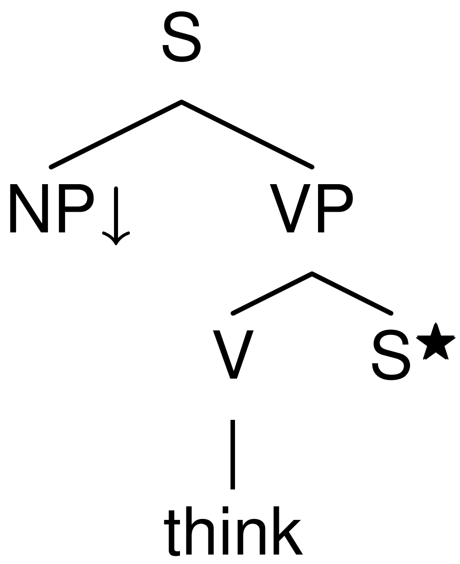


(α_2)

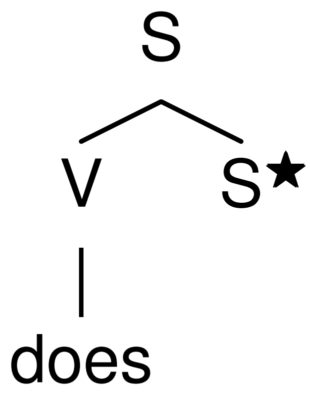


Linguistic applications: sample elementary trees

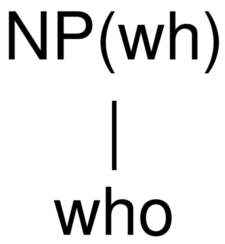
(β_1)



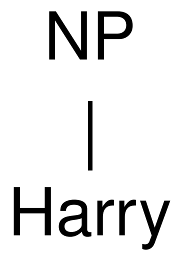
(β_2)



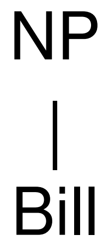
(α_3)



(α_4)

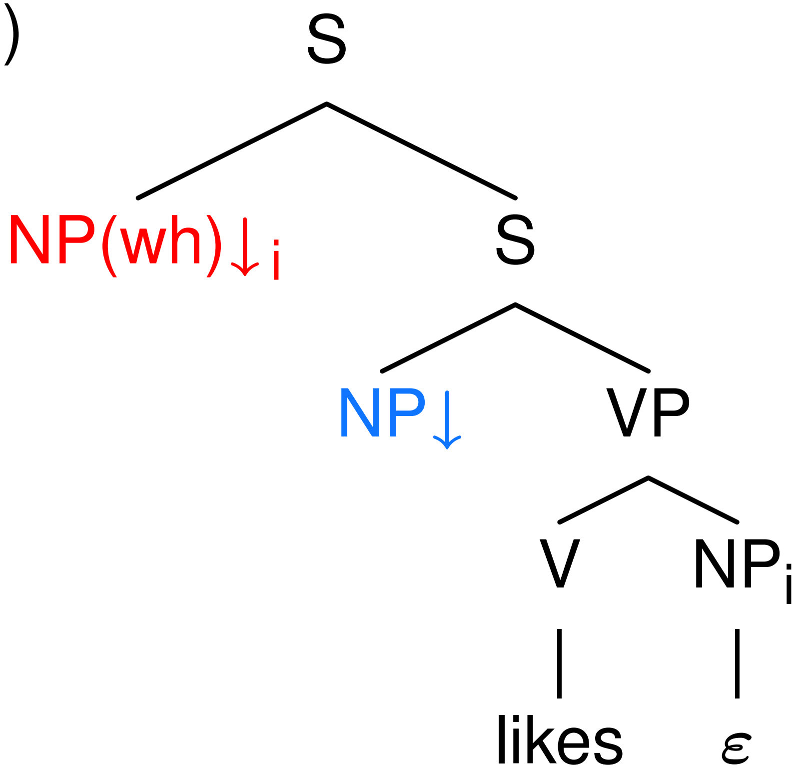


(α_5)

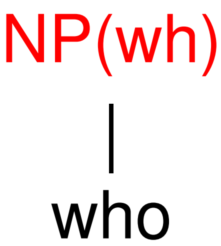


Linguistic applications: sample derivation

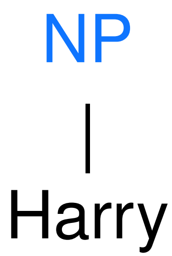
(α_2)



(α_3)

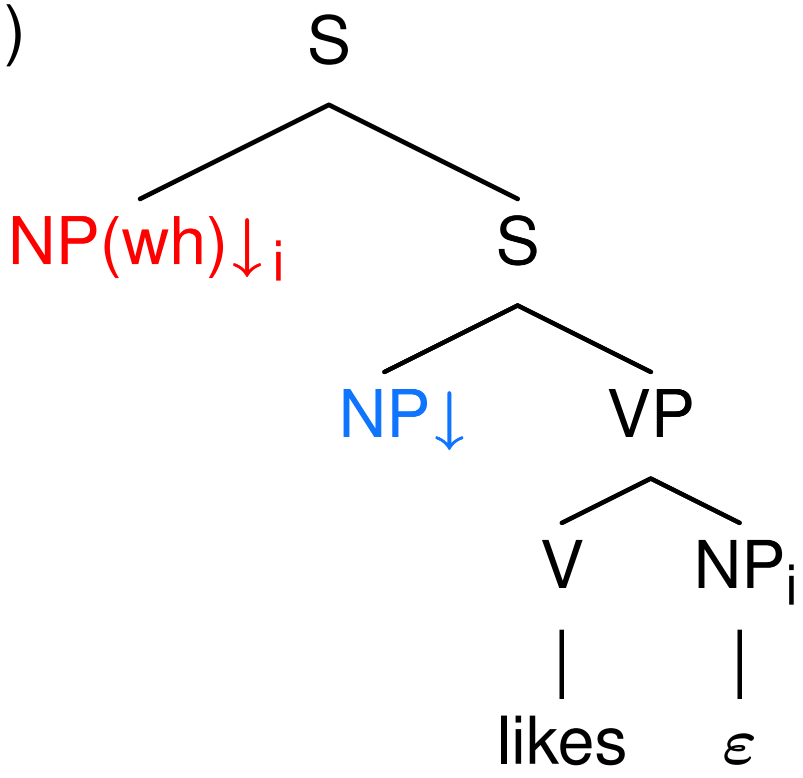


(α_4)

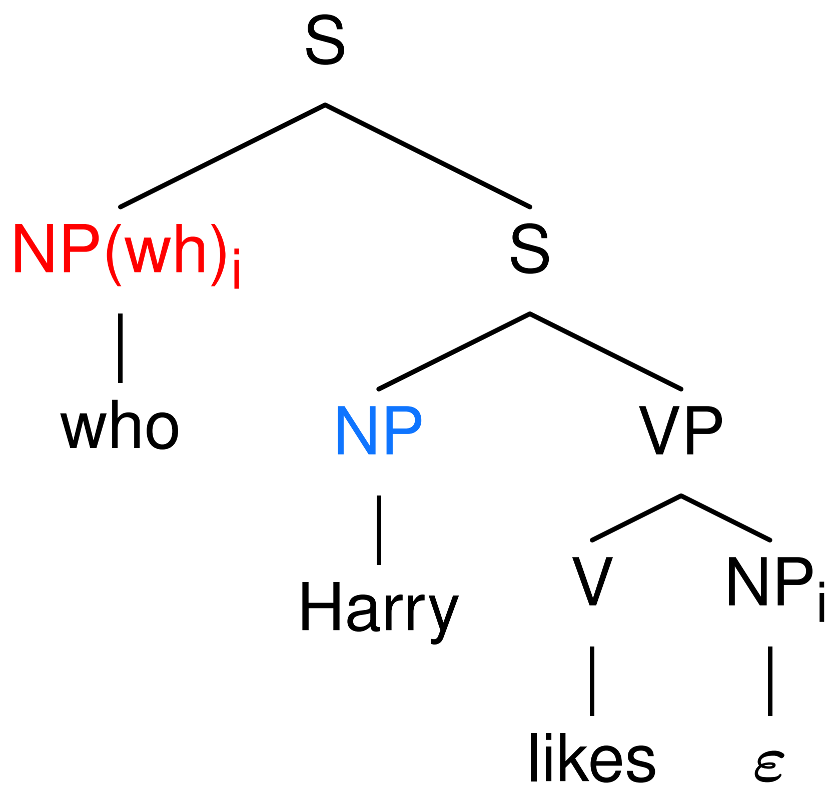


Linguistic applications: sample derivation

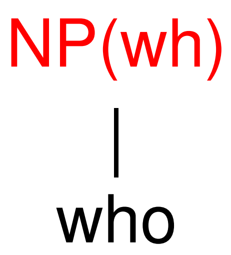
(α_2)



(γ_1)

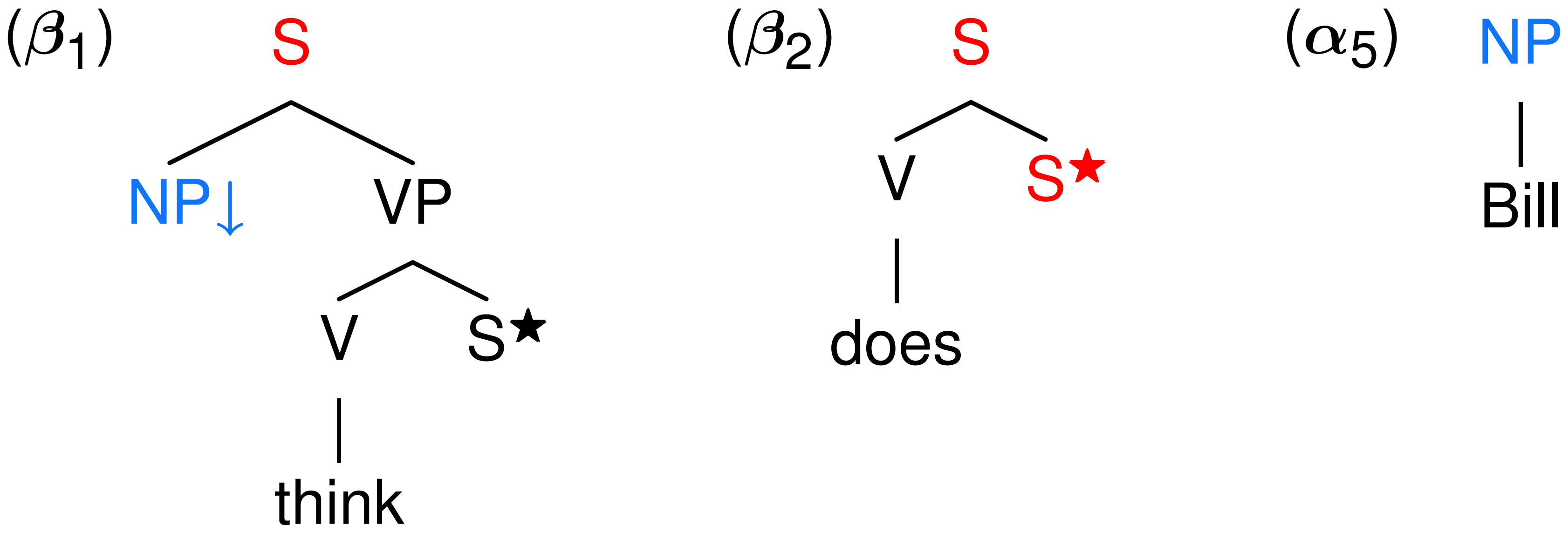


(α_3)

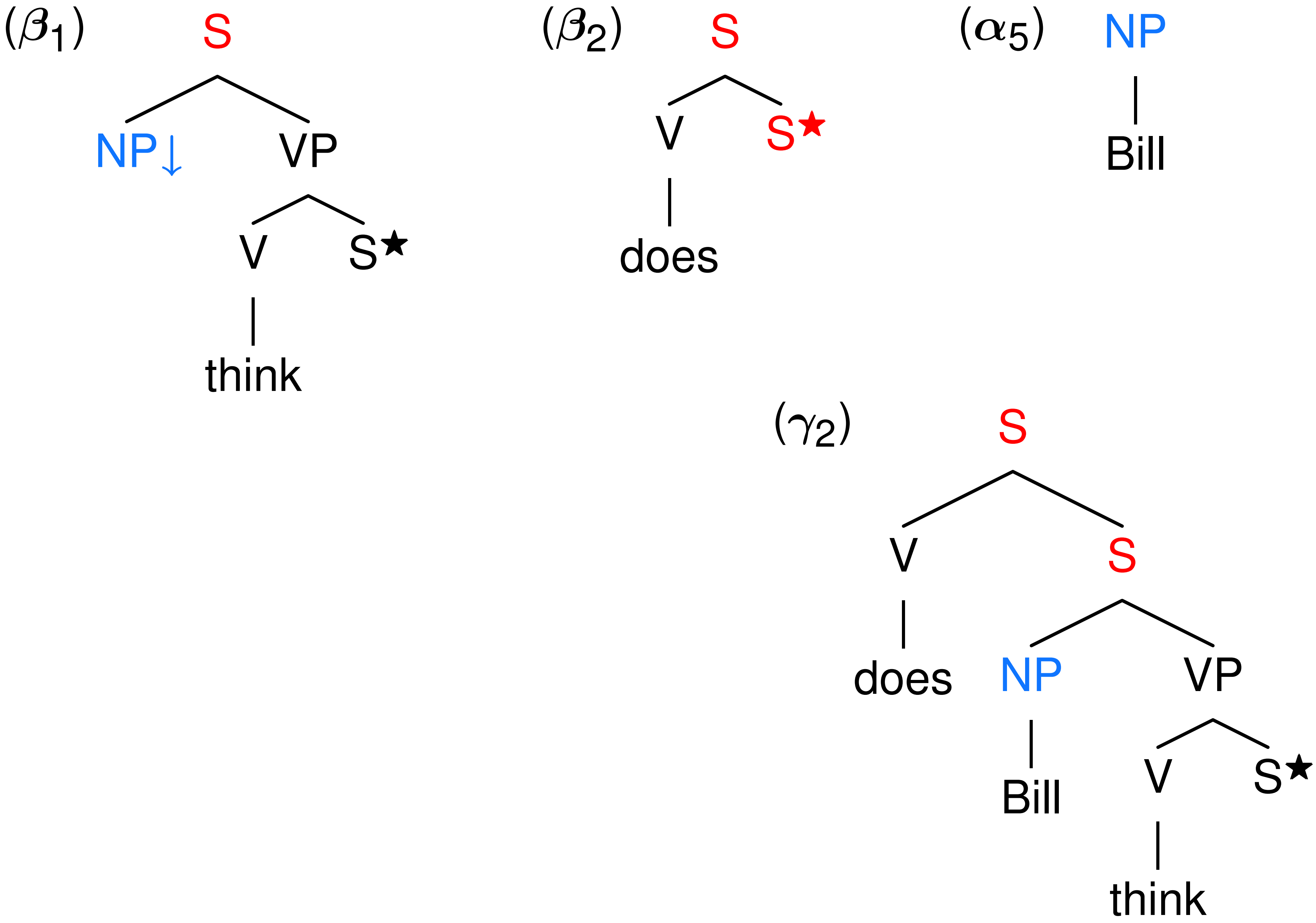


(α_4)

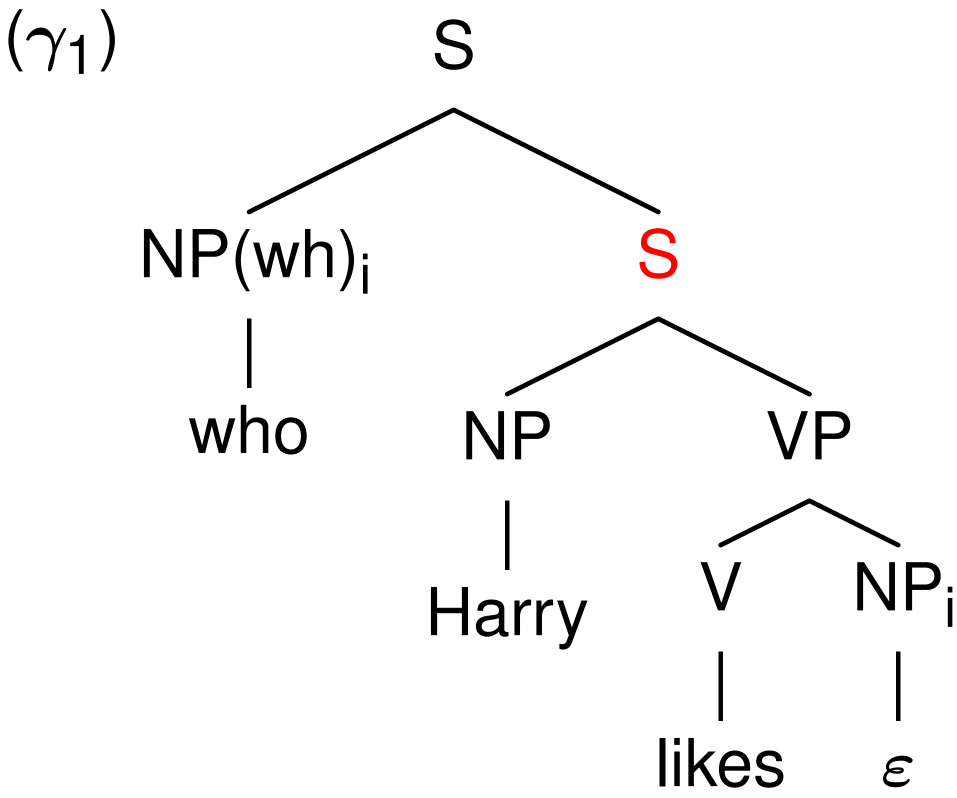
Linguistic applications: sample derivation

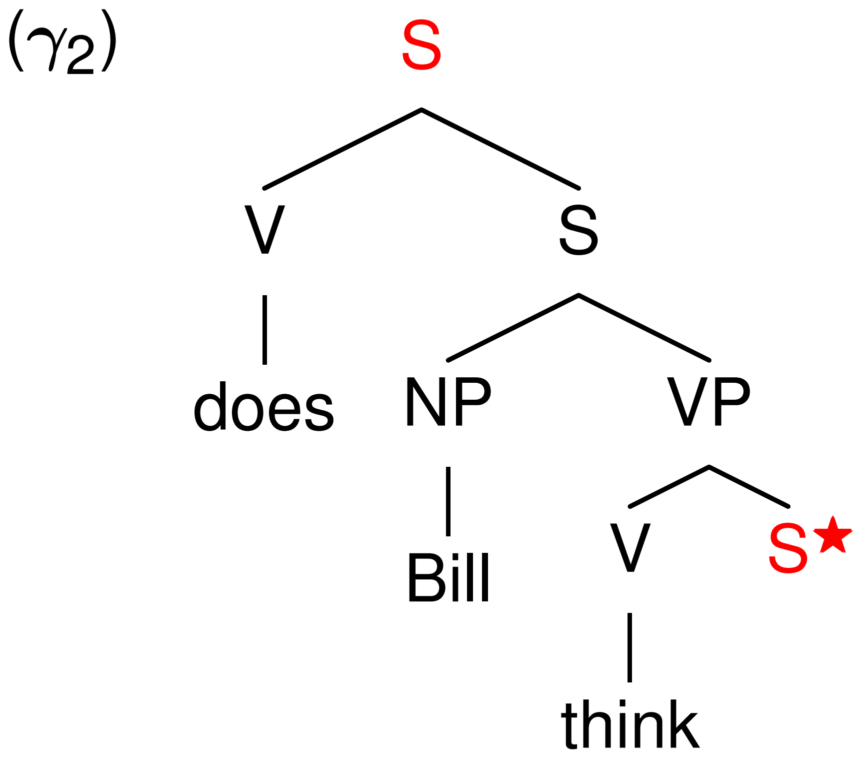


Linguistic applications: sample derivation



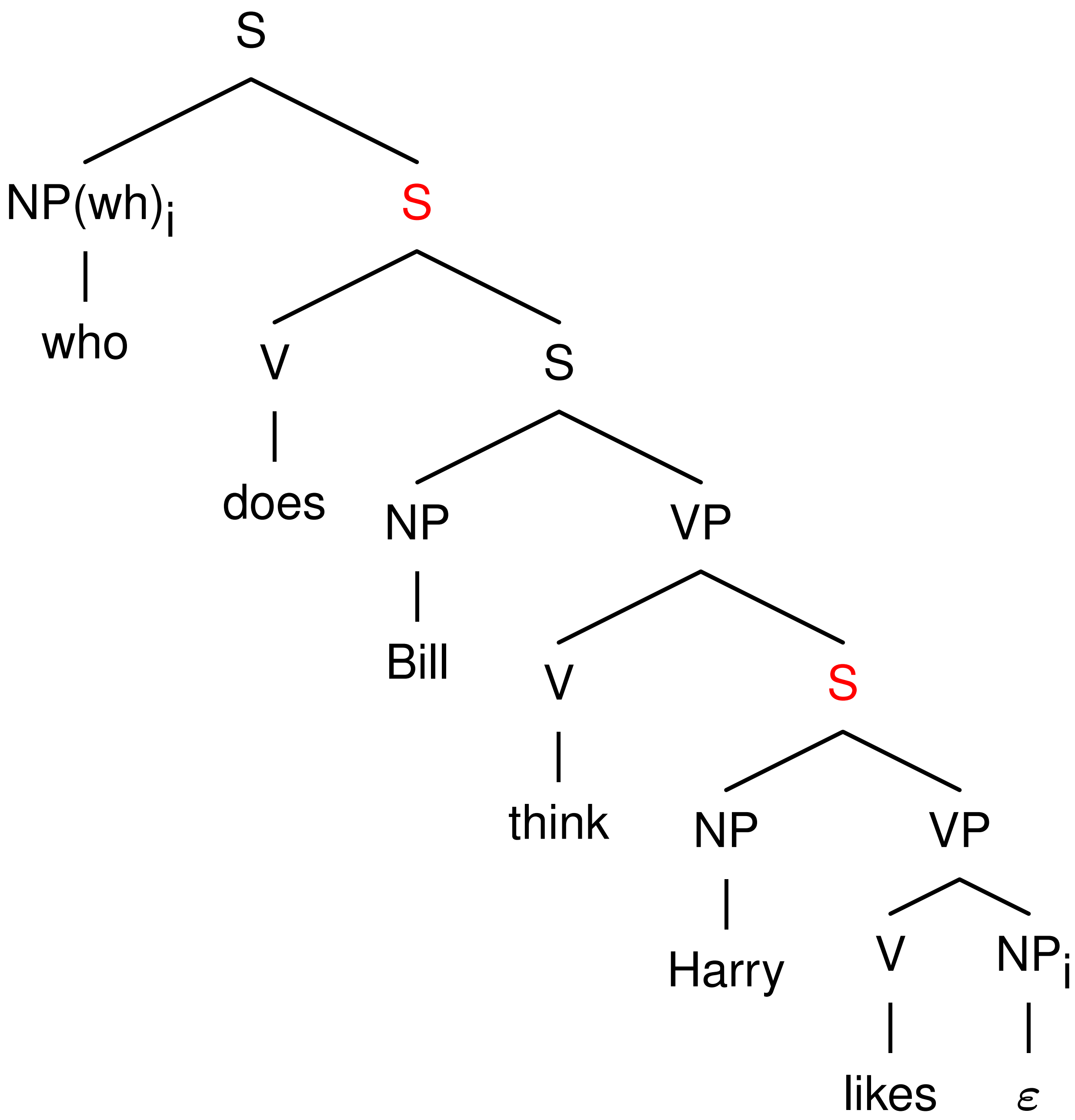
Linguistic applications: sample derivation



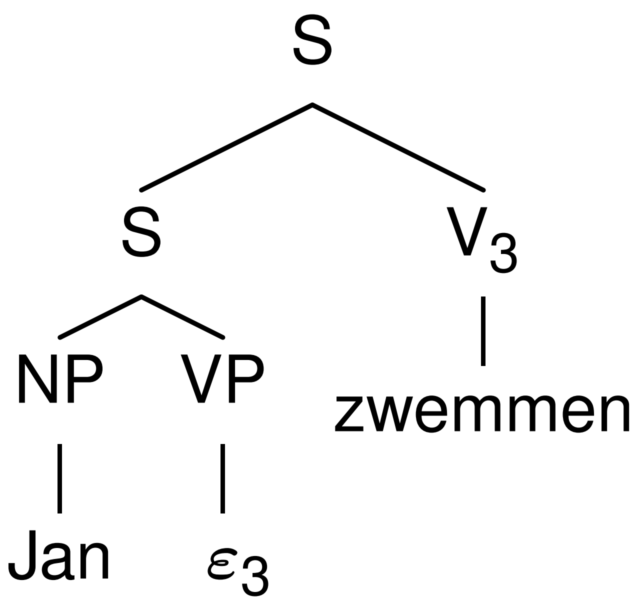


Linguistic applications: sample derivation

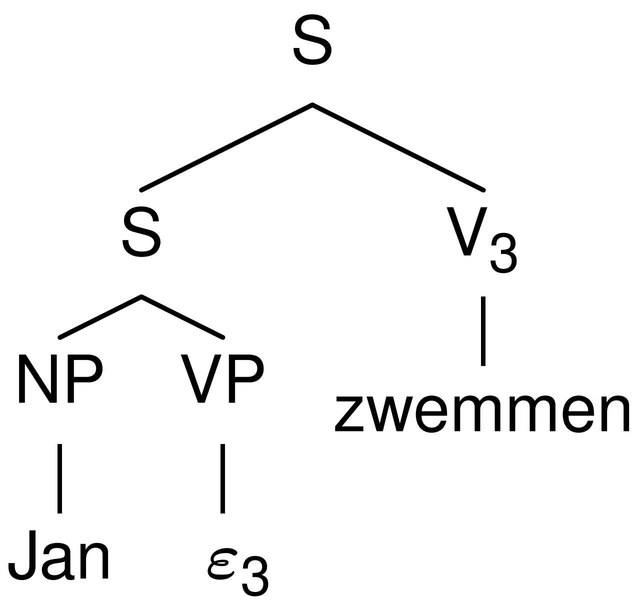
(γ_3)



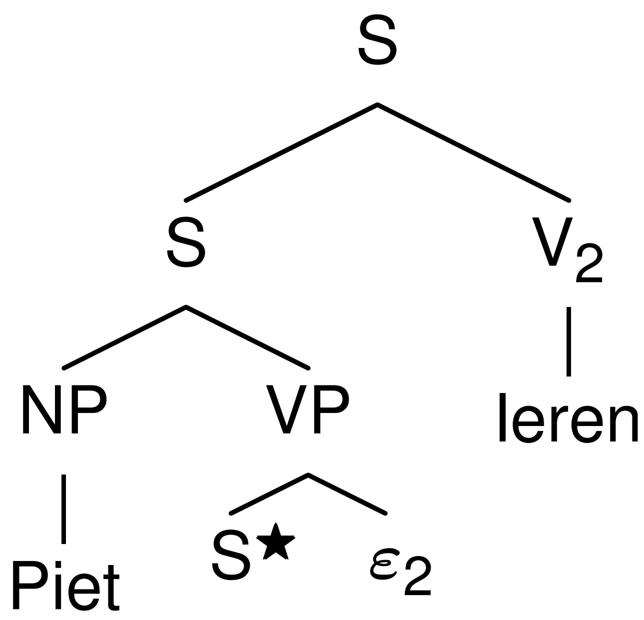
(α_7)



(α_7)

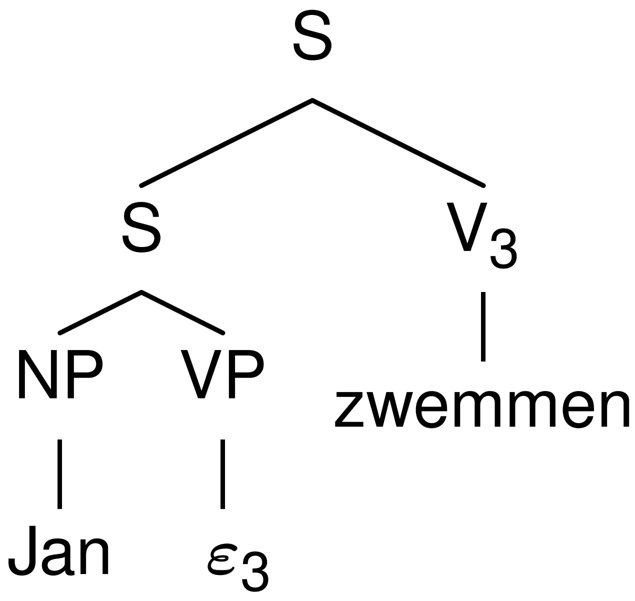


(β_4)

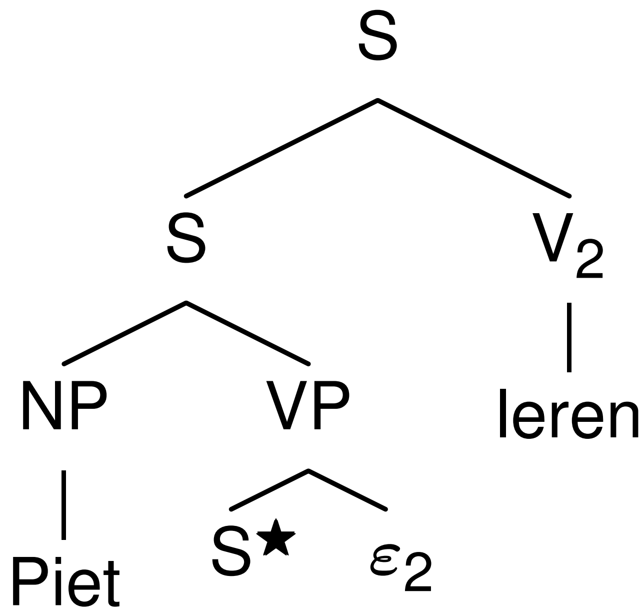


Linguistic applications: “verb clusters”

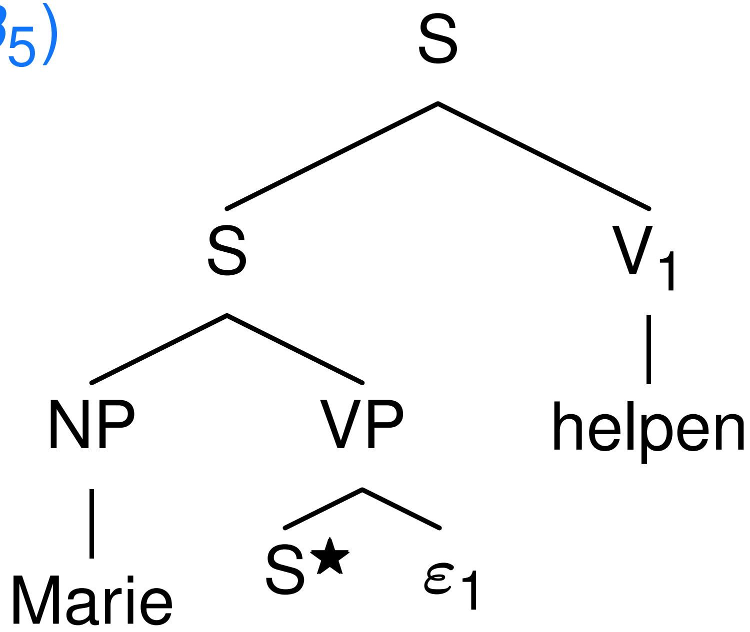
(α_7)



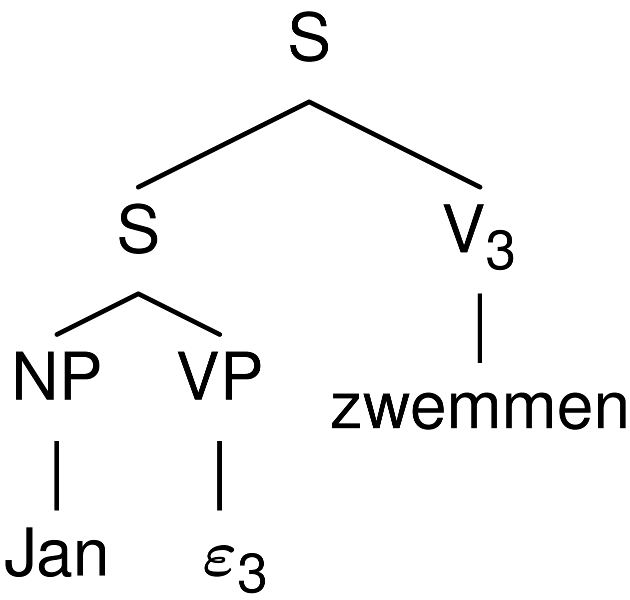
(β_4)



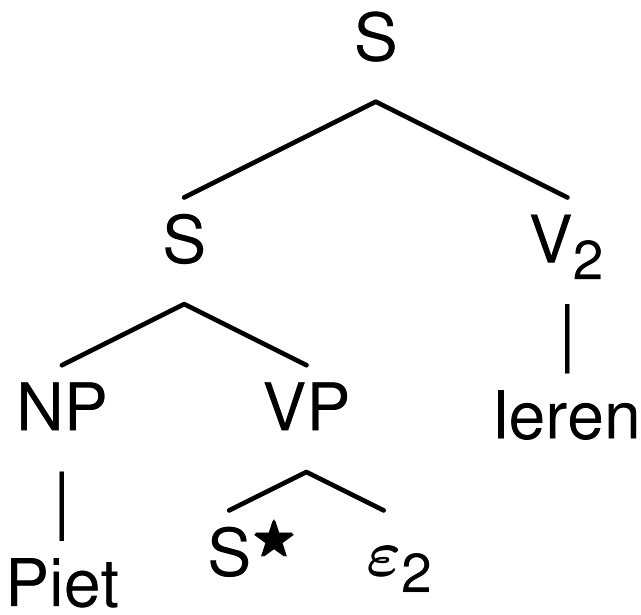
(β_5)



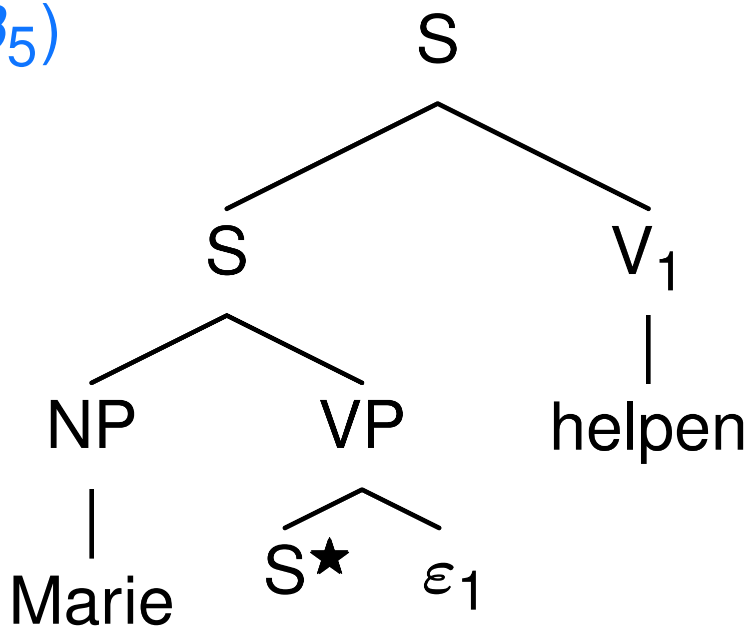
(α_7)



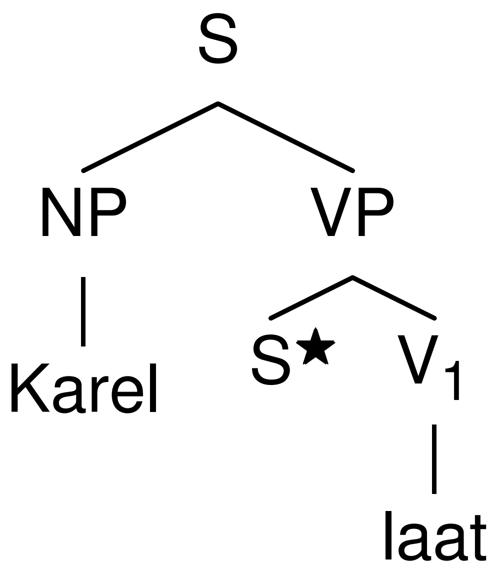
(β_4)



(β_5)



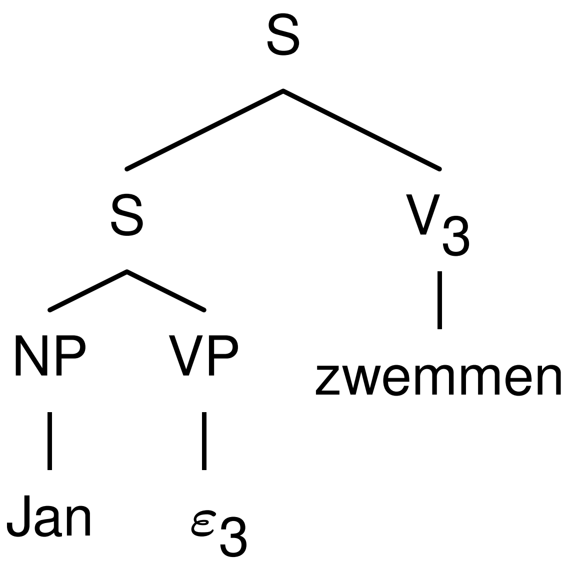
(β_6)



‘dat Karel Marie Piet Jan laat helpen leren zwemmen’

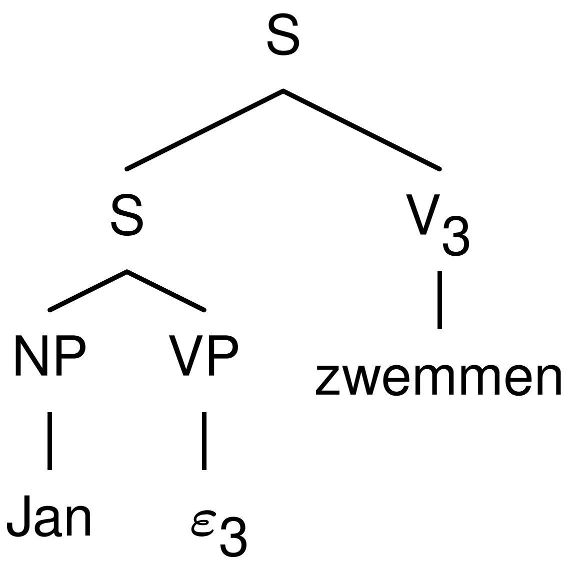
‘dat Karel Marie Piet Jan laat helpen leren zwemmen’

(α_7)

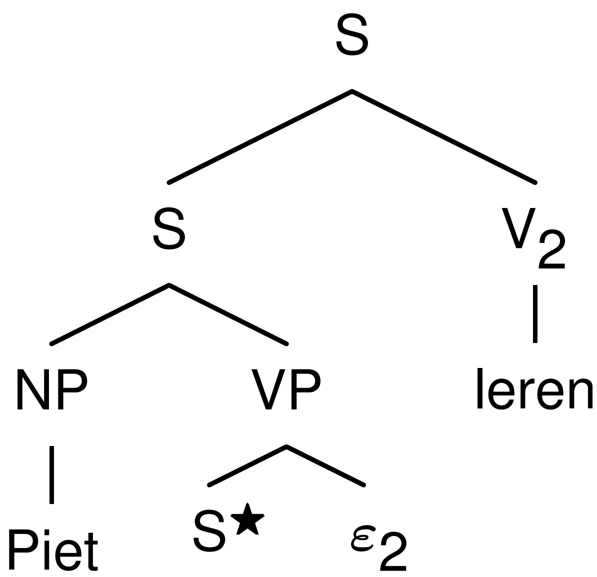


‘dat Karel Marie Piet Jan laat helpen leren zwemmen’

(α_7)

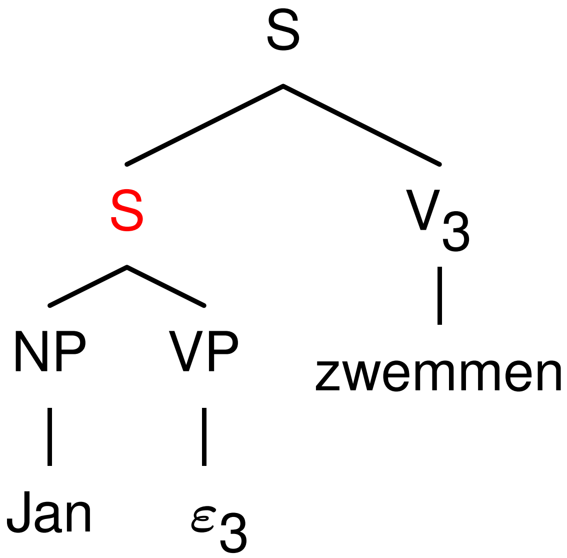


(β_4)

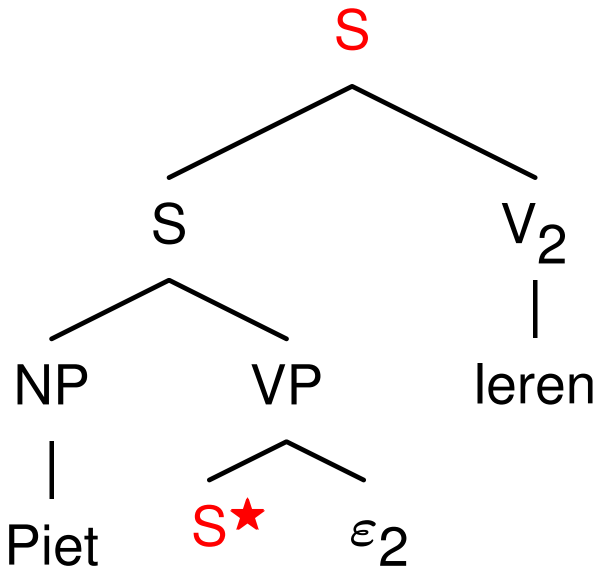


‘dat Karel Marie Piet Jan laat helpen leren zwemmen’

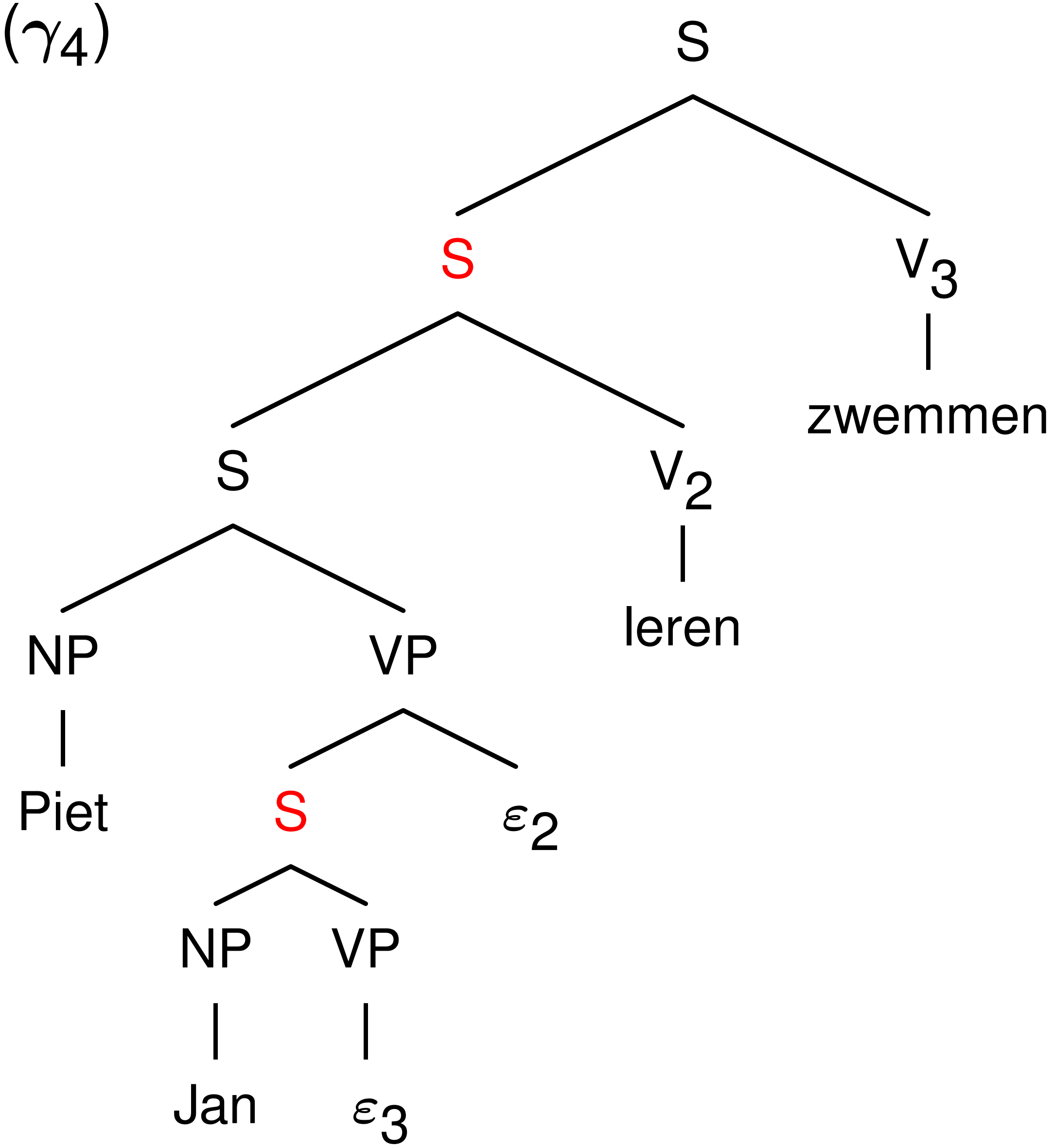
(α_7)



(β_4)

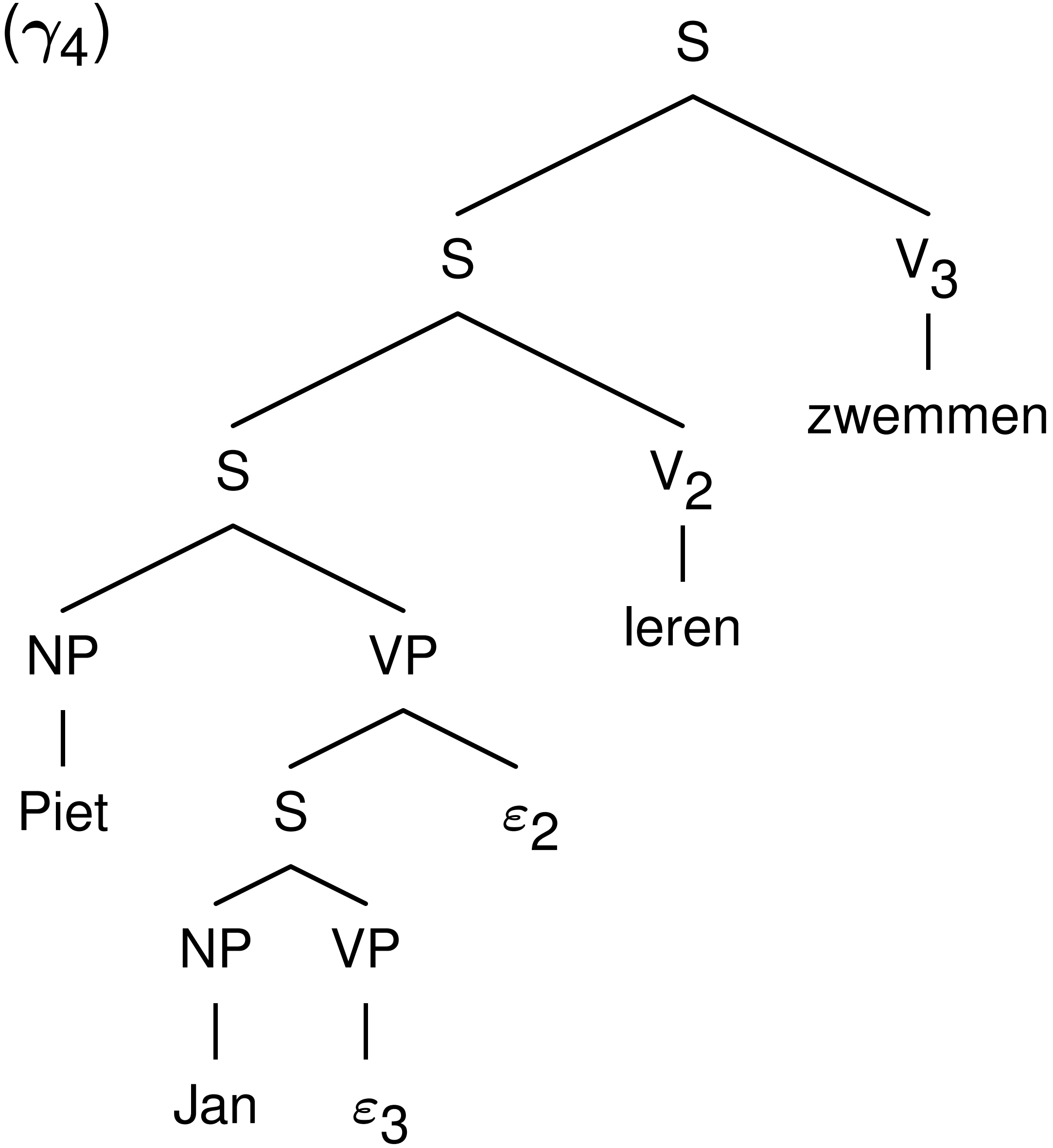


(γ_4)



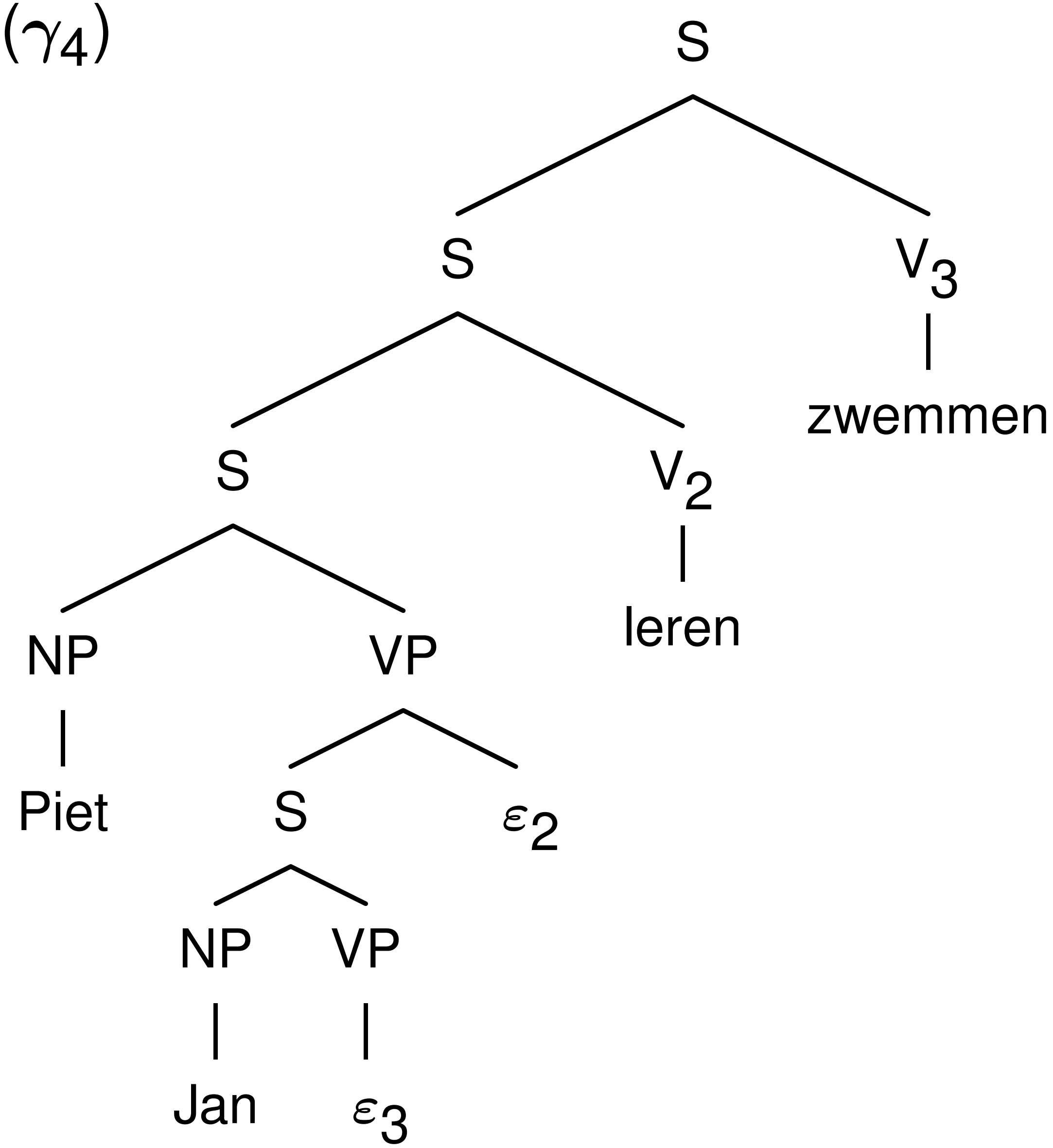
‘dat Karel Marie Piet Jan laat helpen leren zwemmen’

(γ_4)

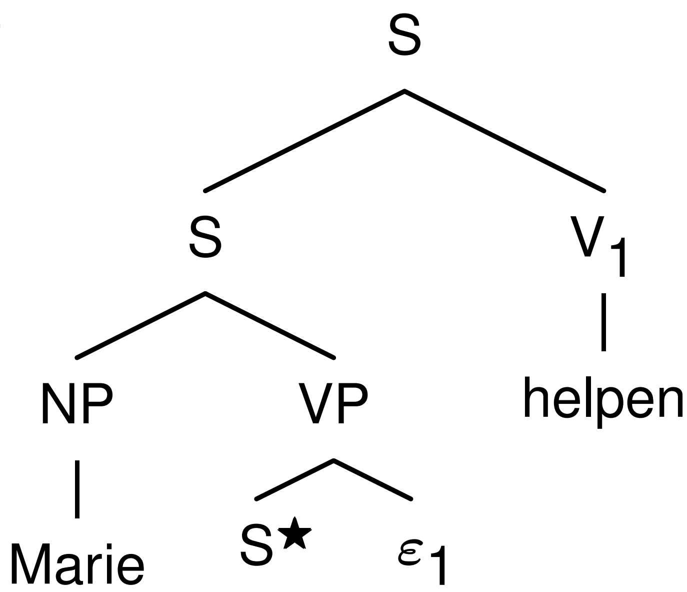


‘dat Karel Marie Piet Jan laat helpen leren zwemmen’

(γ_4)

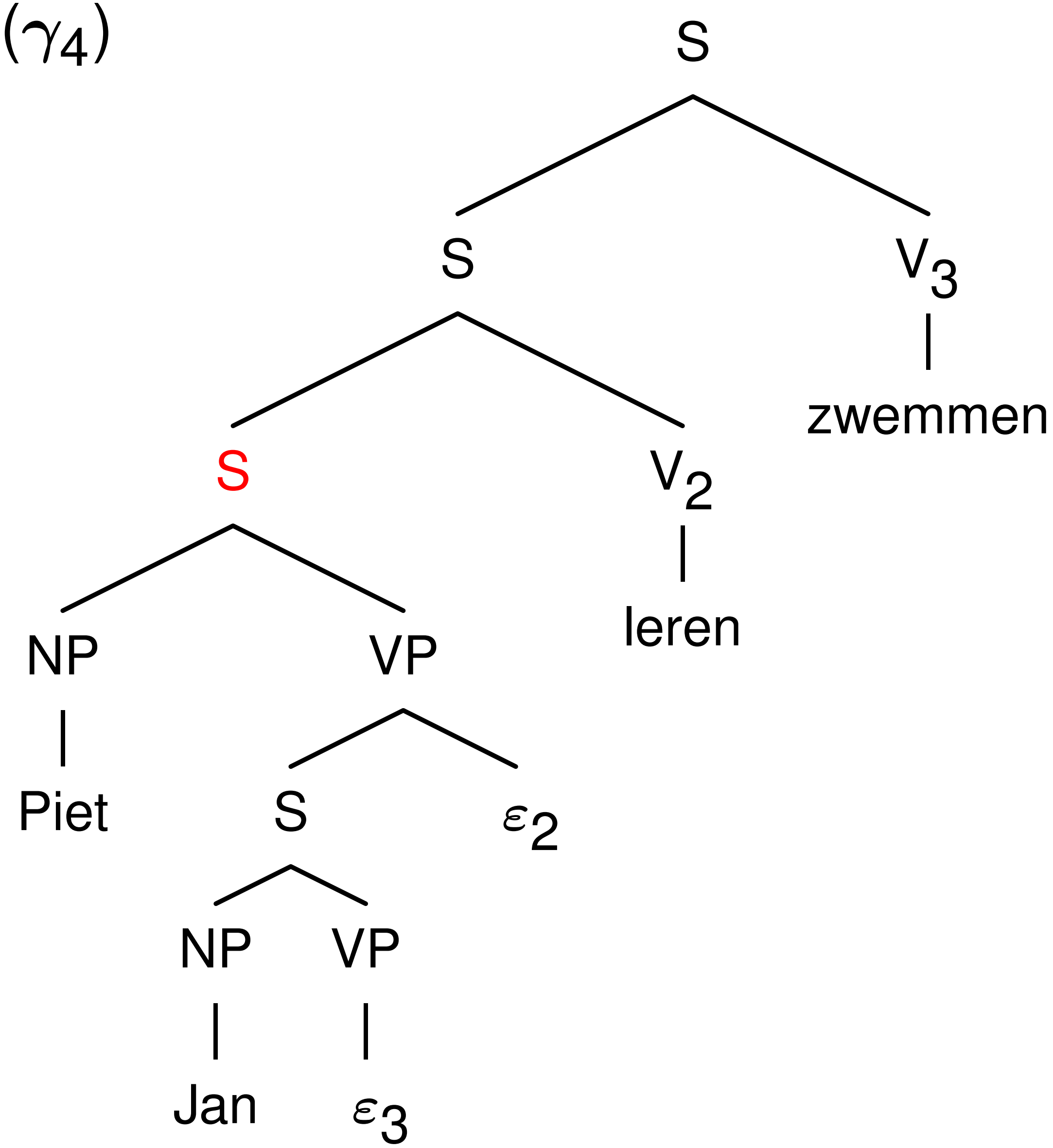


(β_5)

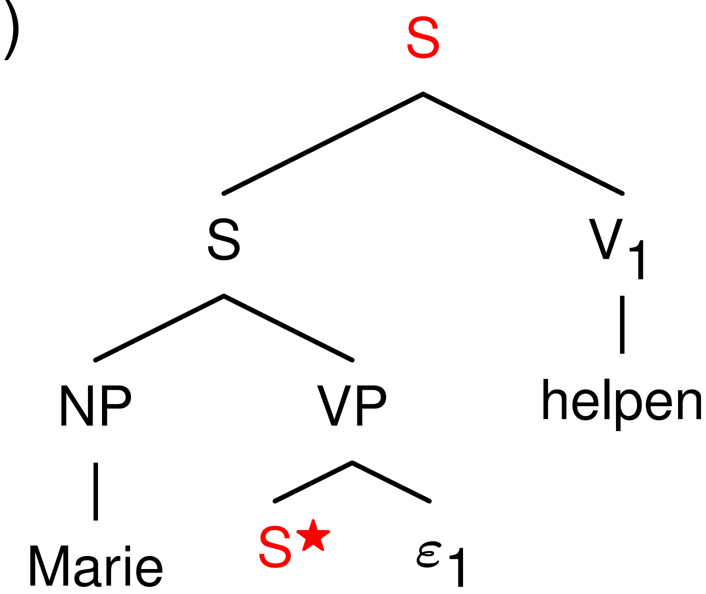


‘dat Karel Marie Piet Jan laat helpen leren zwemmen’

(γ_4)

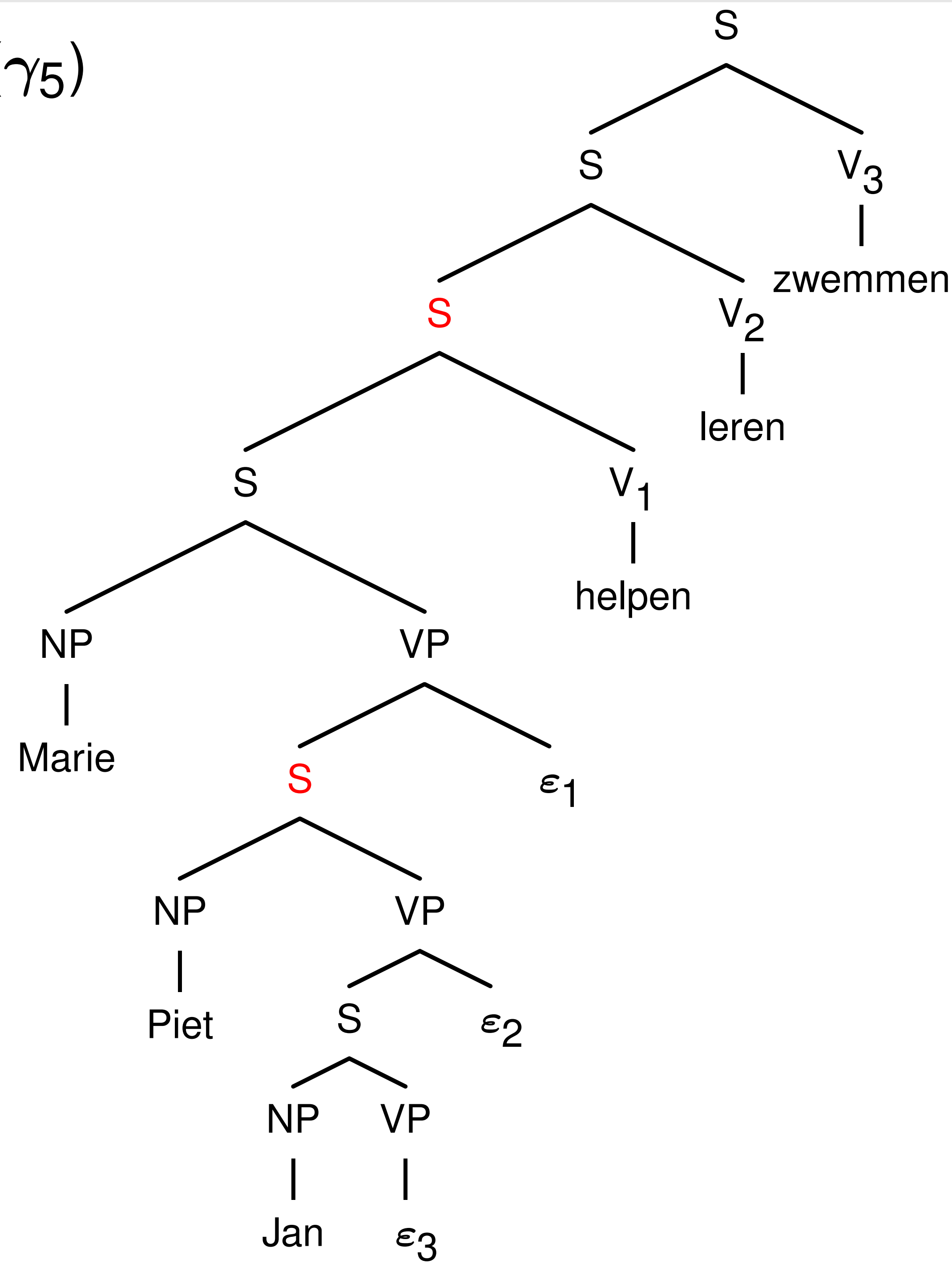


(β_5)



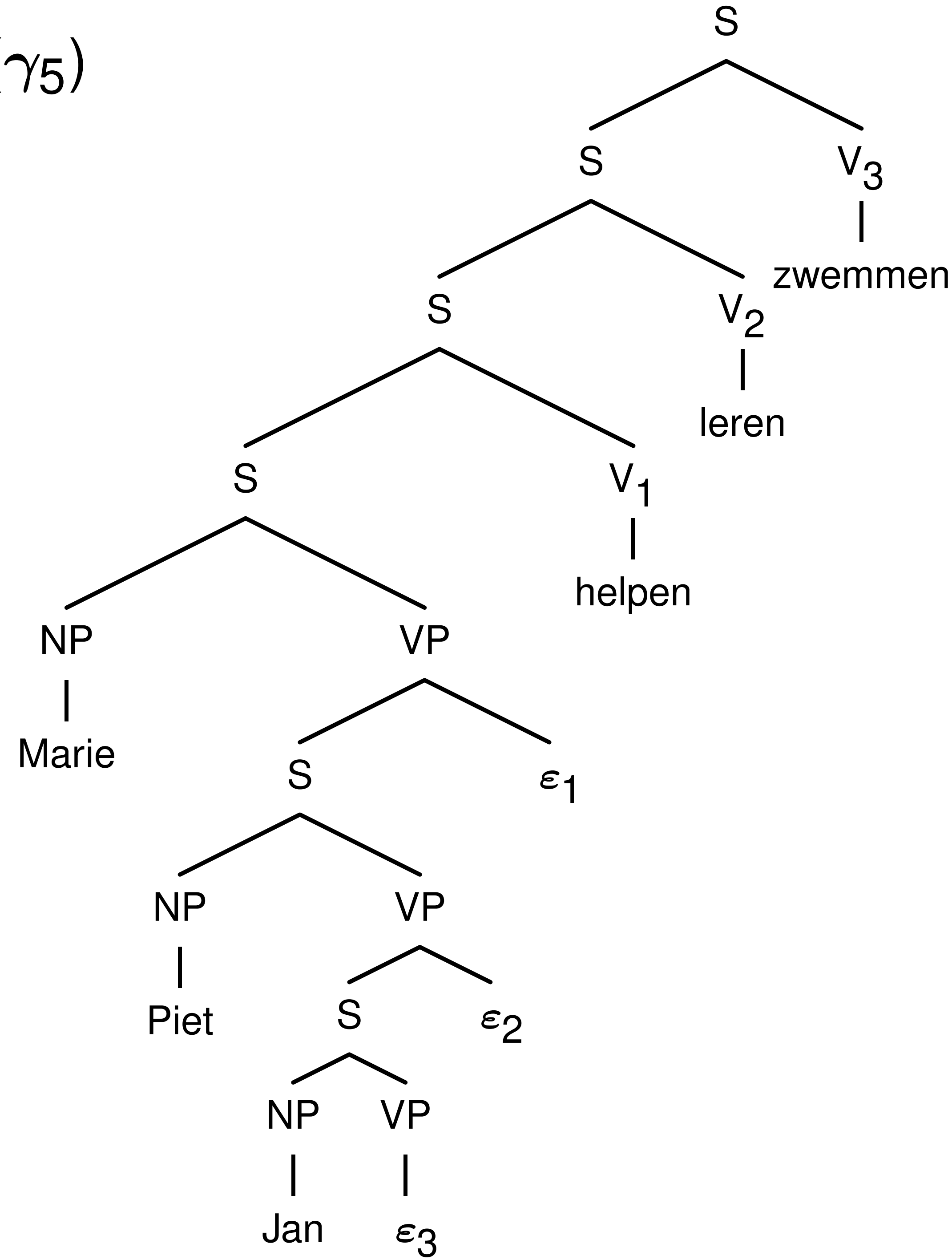
‘dat Karel Marie Piet Jan laat helpen leren zwemmen’

(γ_5)



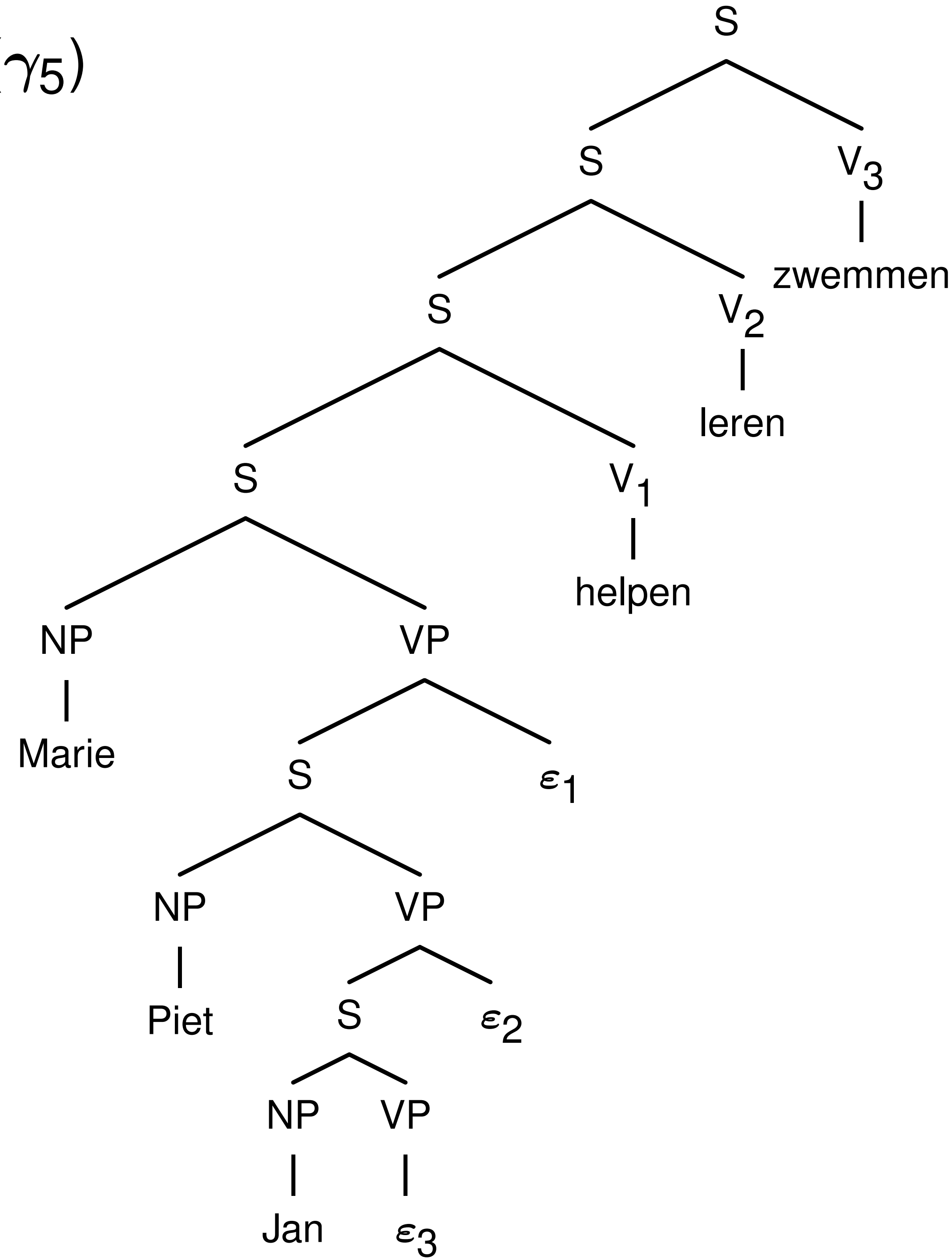
‘dat Karel Marie Piet Jan laat helpen leren zwemmen’

(γ_5)

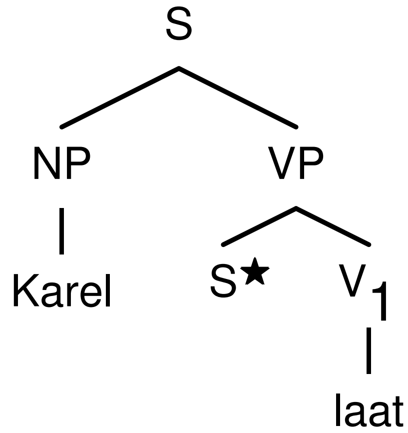


‘dat Karel Marie Piet Jan laat helpen leren zwemmen’

(γ_5)

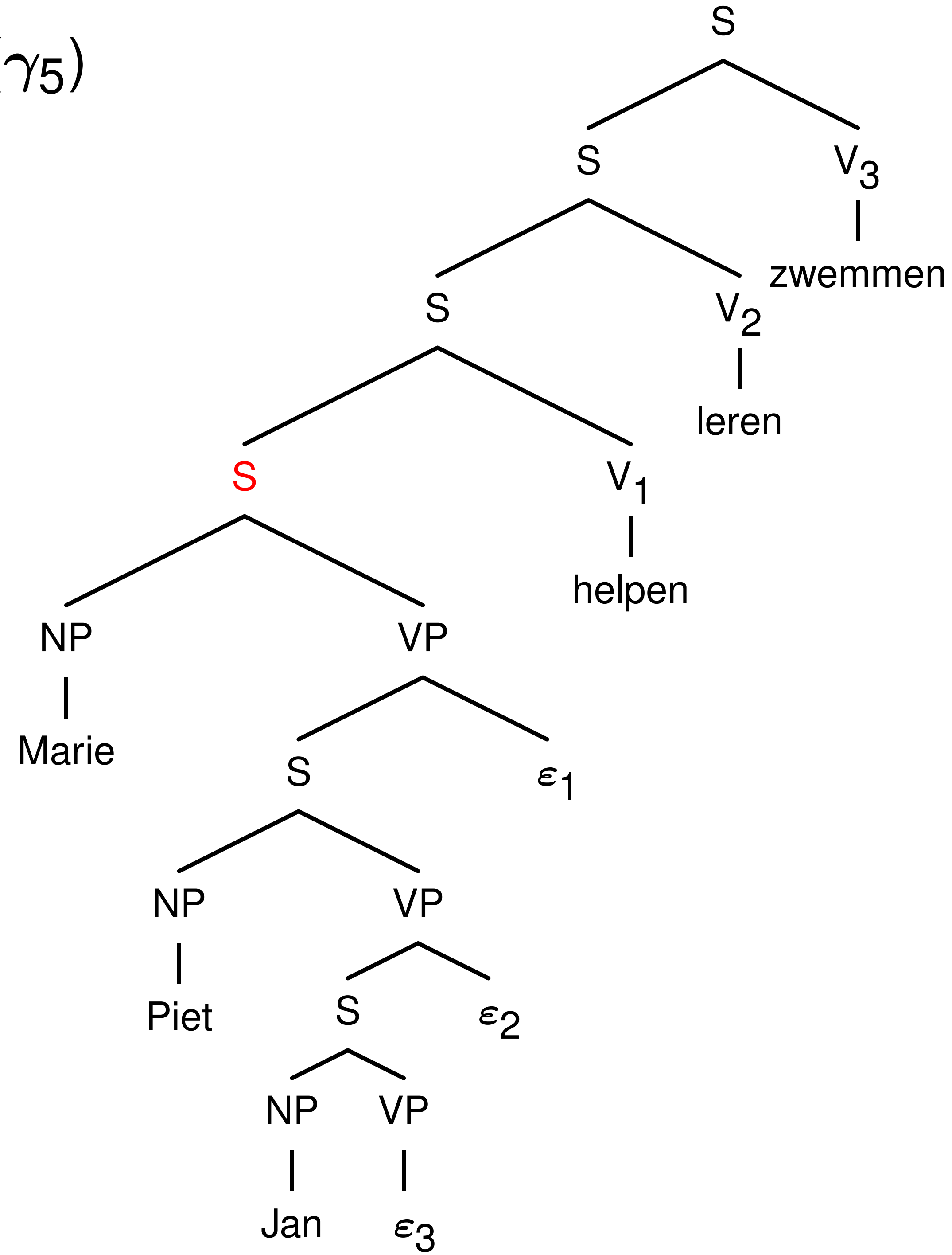


(β_6)

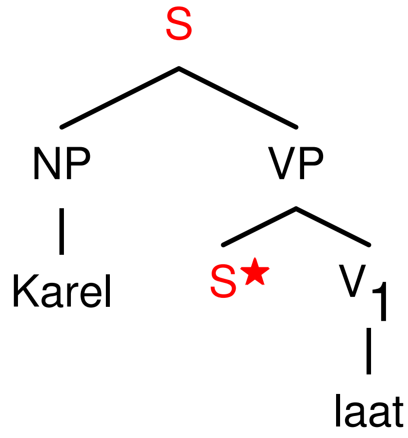


‘dat Karel Marie Piet Jan laat helpen leren zwemmen’

(γ_5)



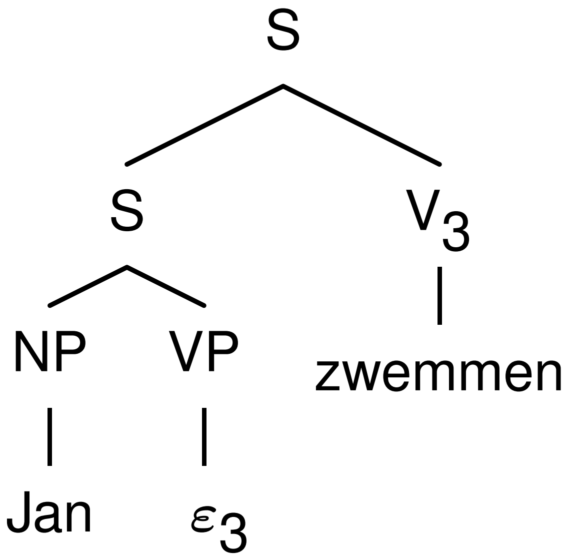
(β_6)



‘dat Karel Piet Jan laat leren zwemmen’

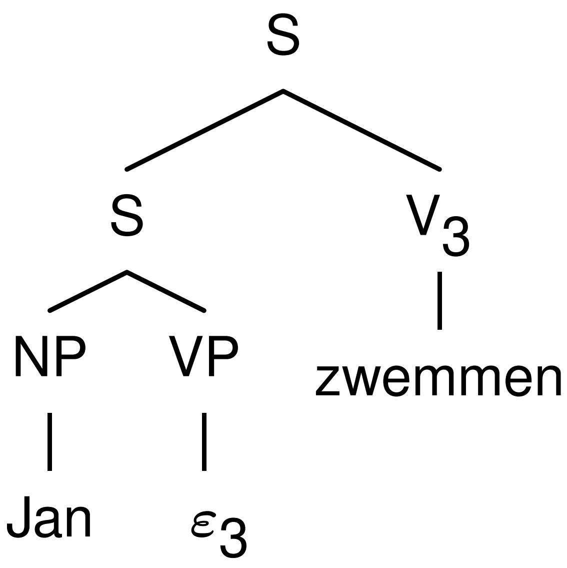
‘dat Karel Piet Jan laat leren zwemmen’

(α_7)

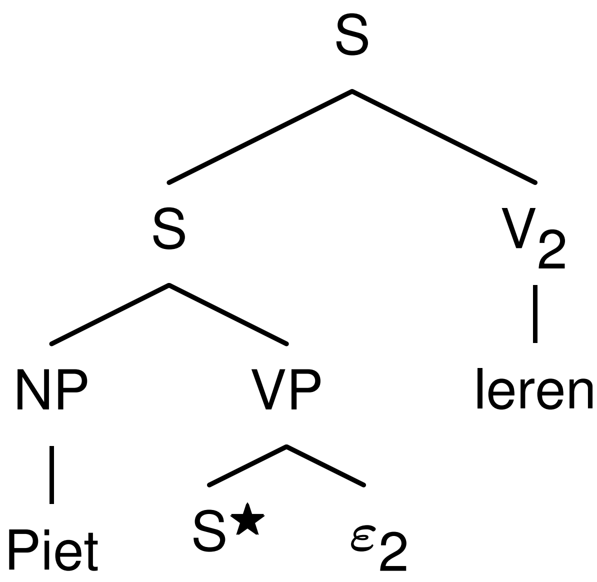


‘dat Karel Piet Jan laat leren zwemmen’

(α_7)

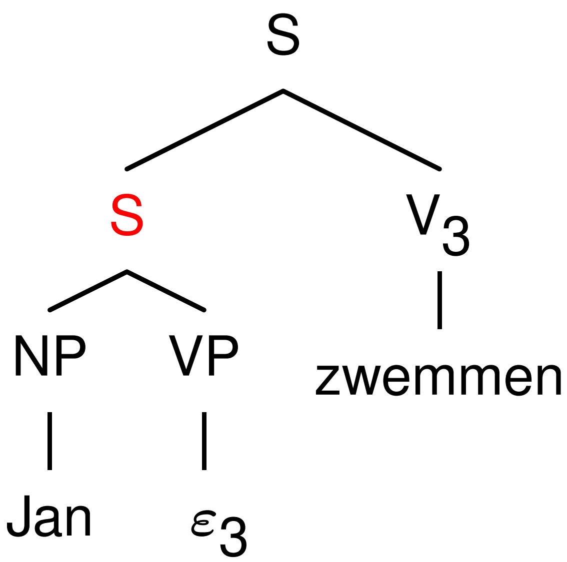


(β_4)

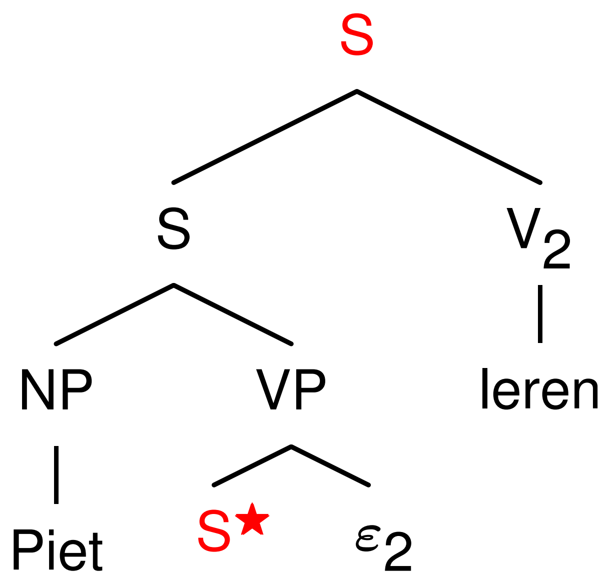


‘dat Karel Piet Jan laat leren zwemmen’

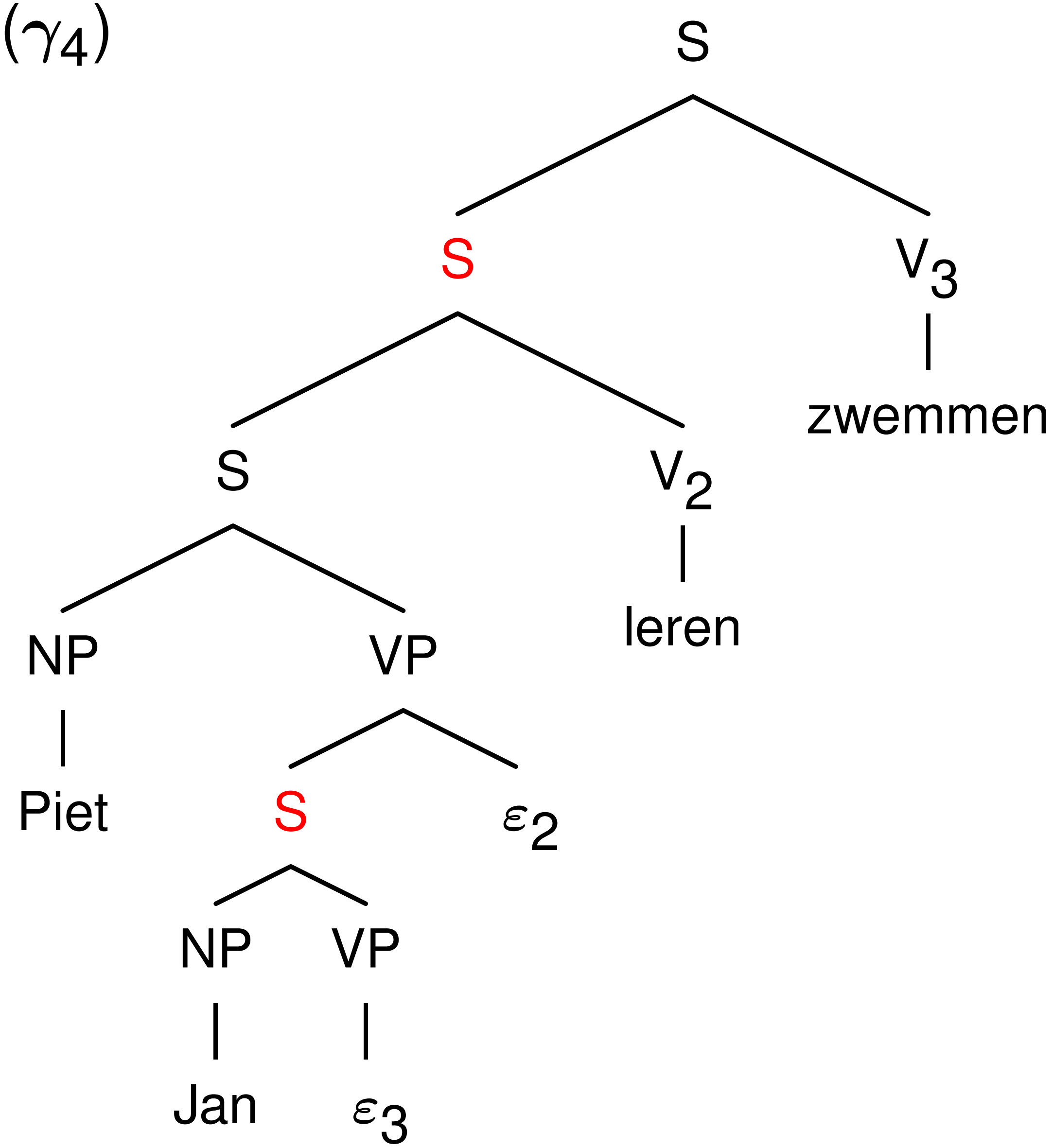
(α_7)



(β_4)

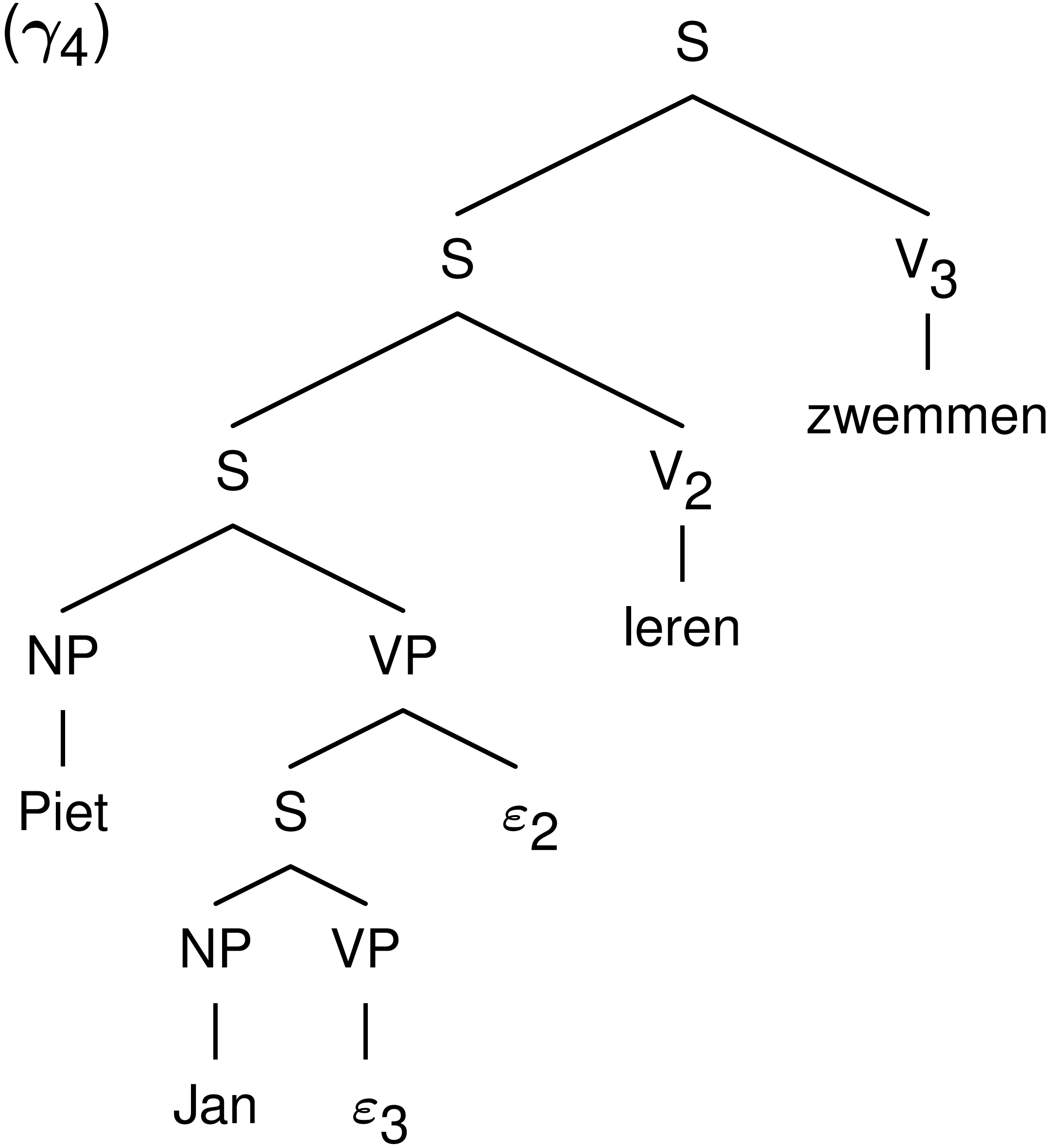


(γ_4)

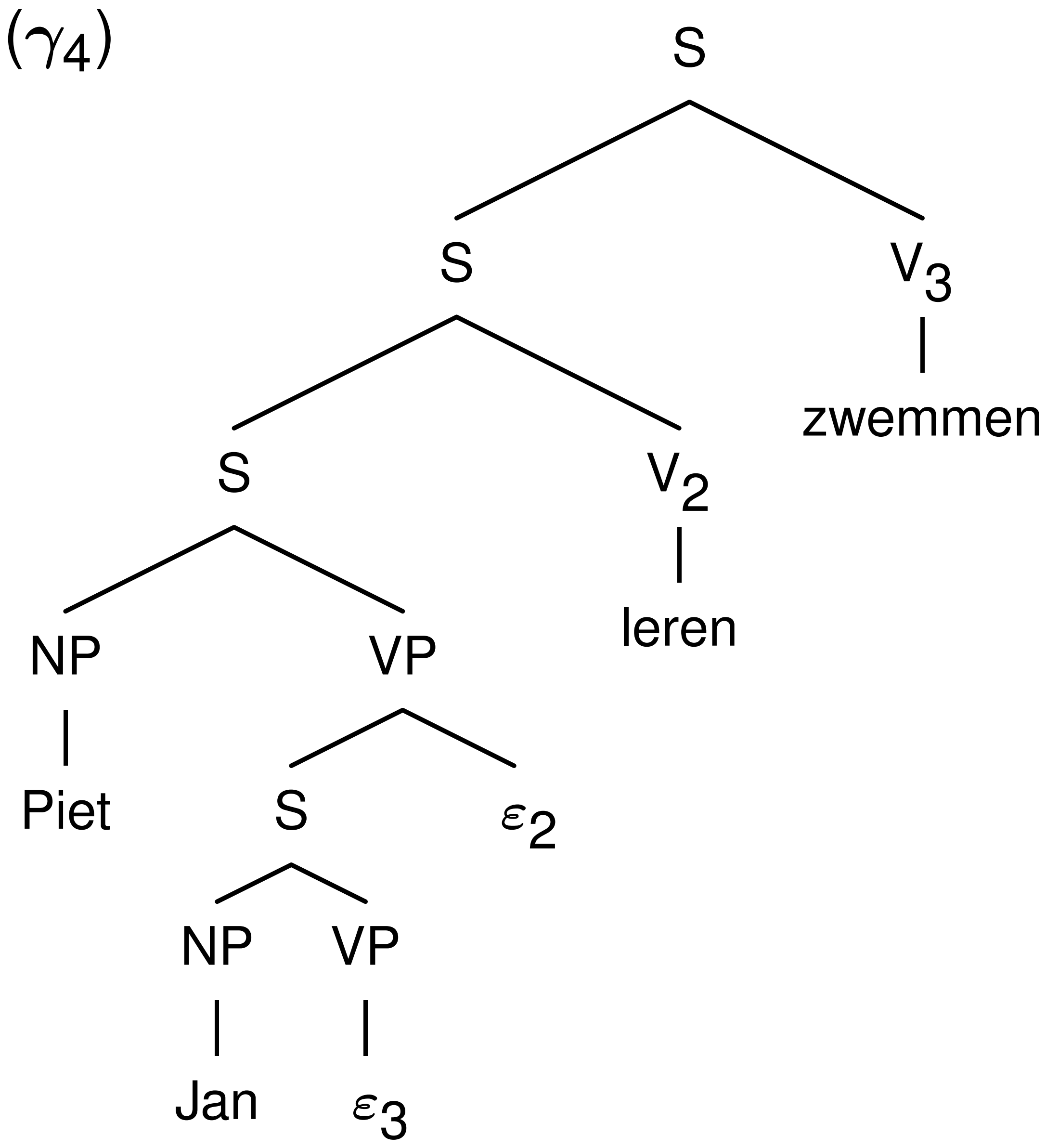


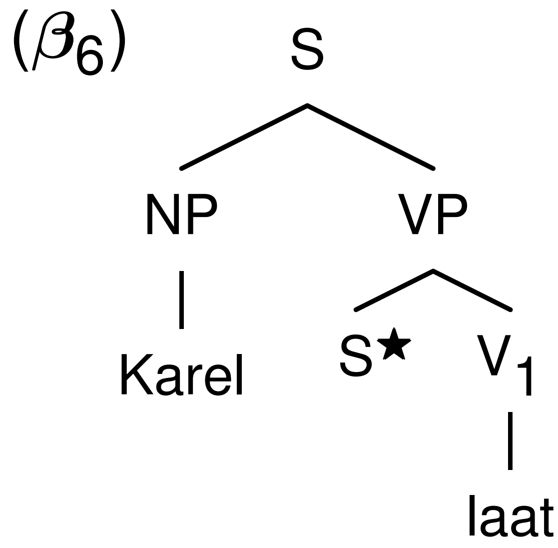
‘dat Karel Piet Jan laat helpen leren zwemmen’

(γ_4)

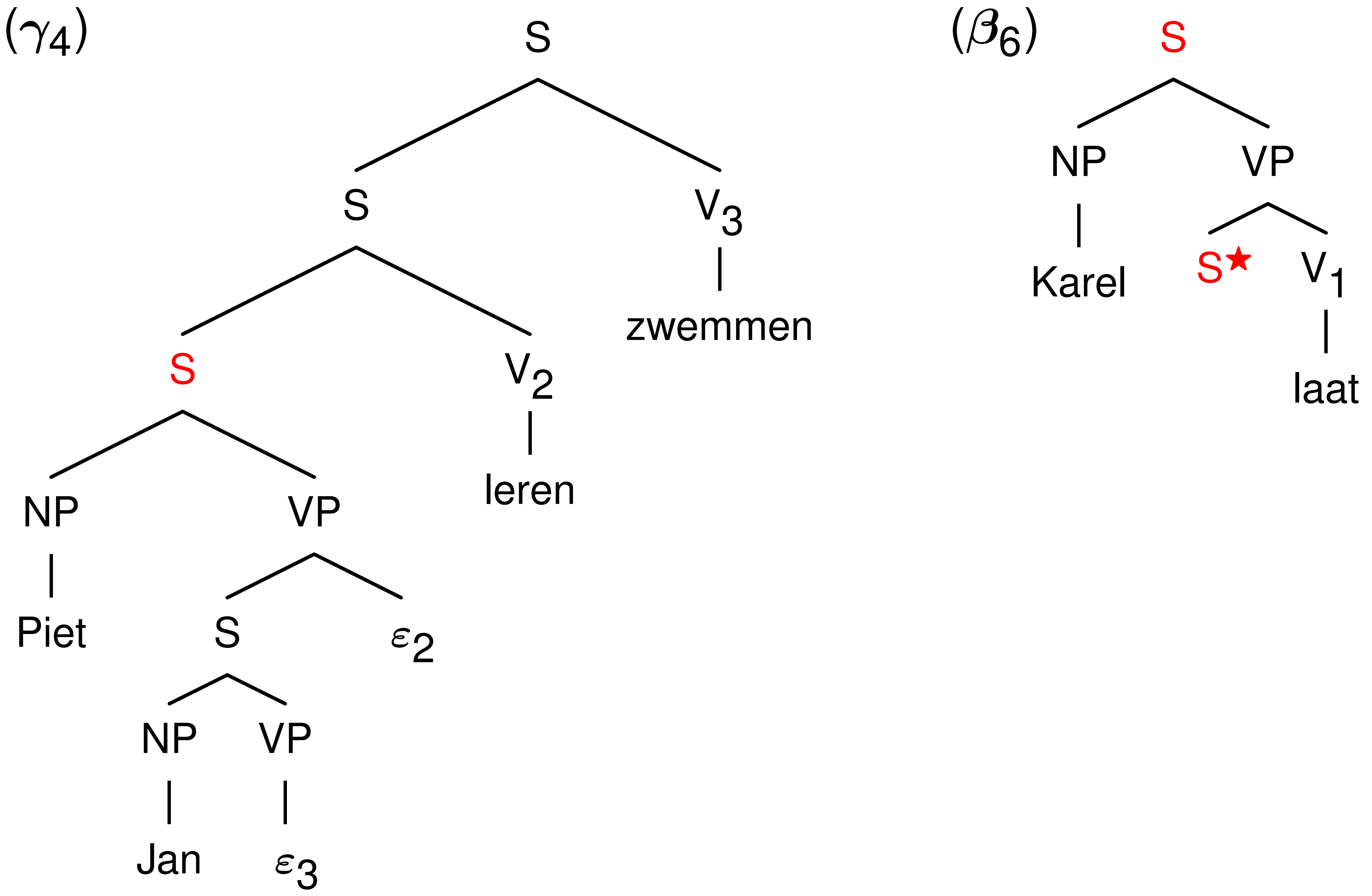


‘dat Karel Piet Jan laat helpen leren zwemmen’



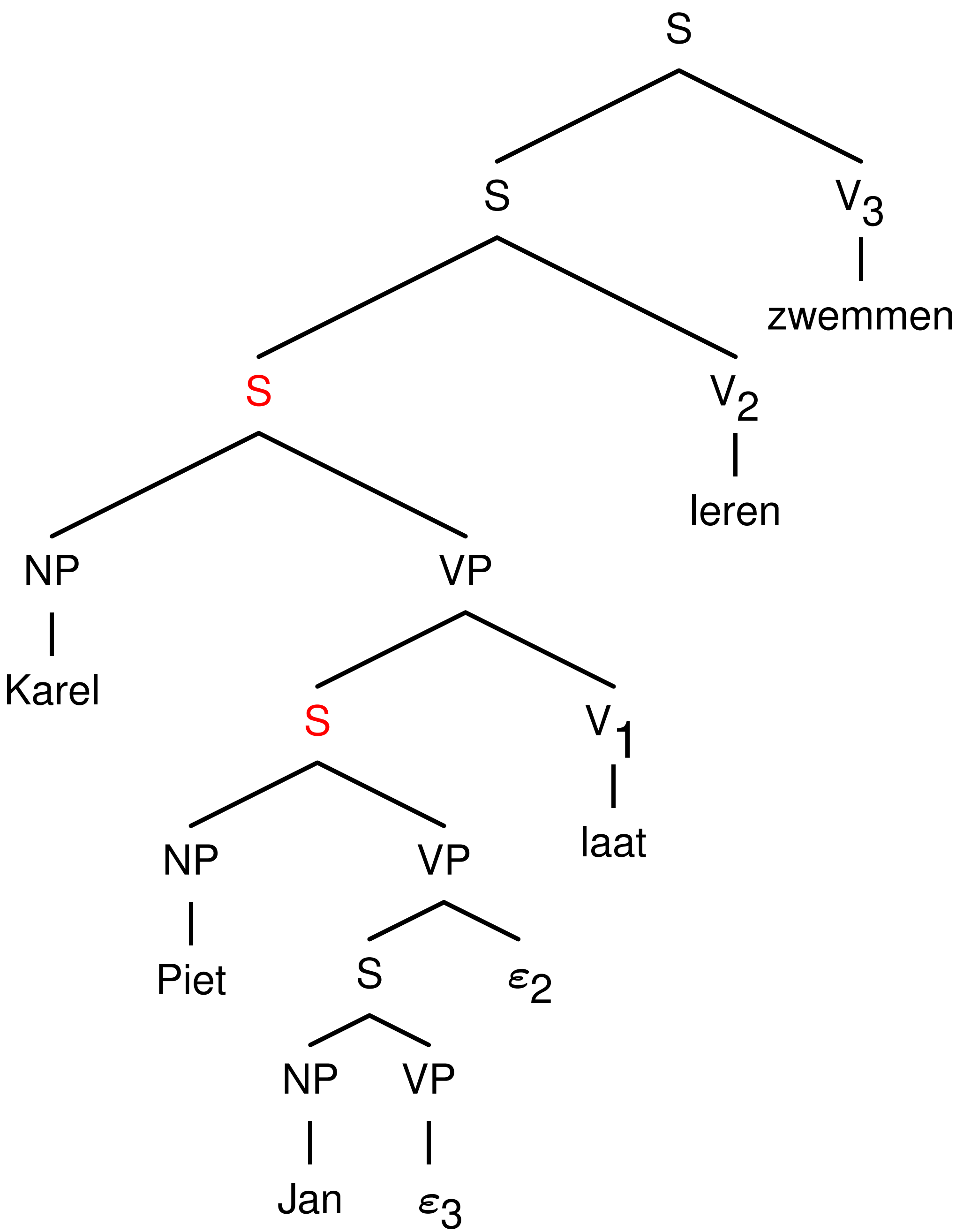


‘dat Karel Piet Jan laat helpen leren zwemmen’



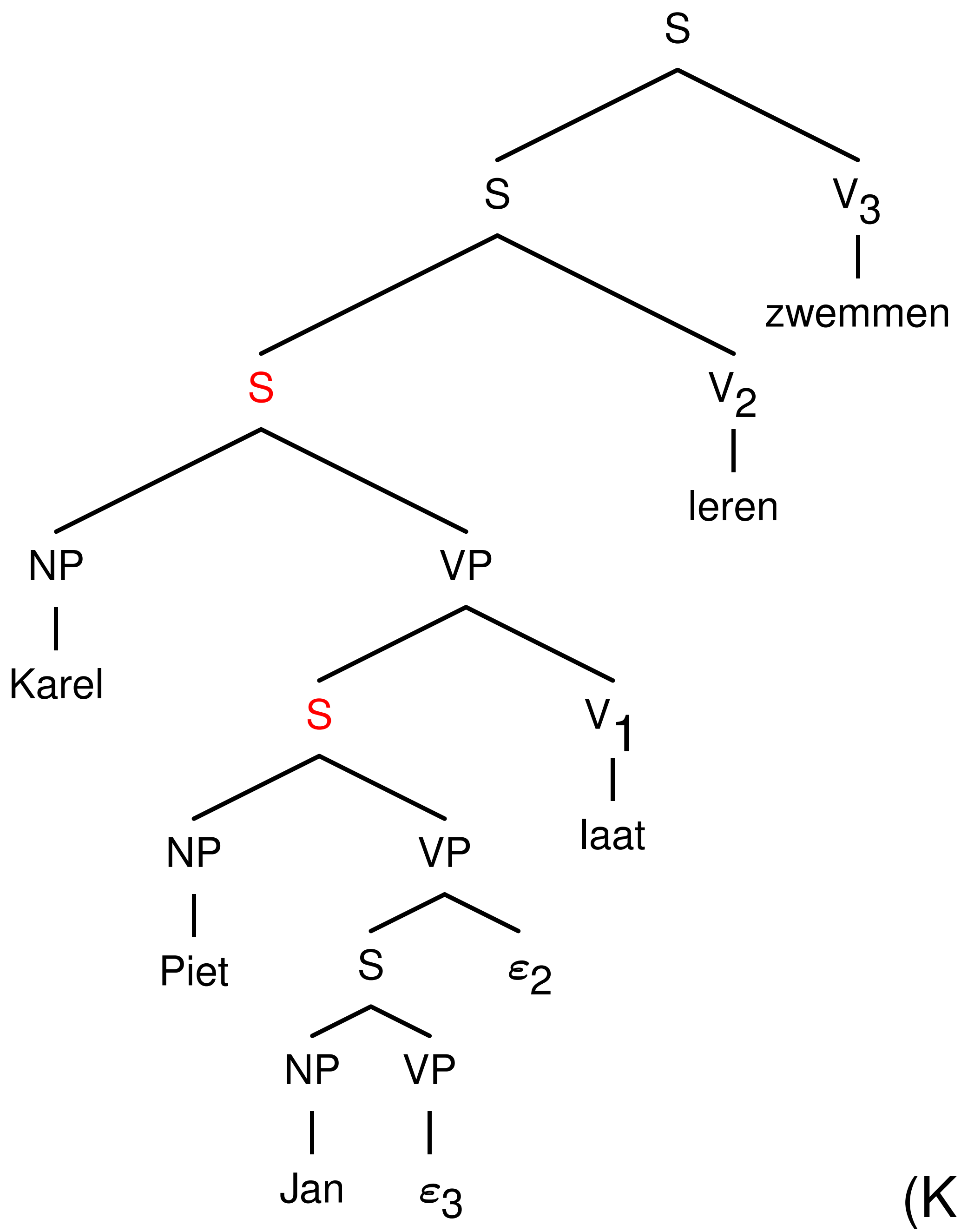
‘dat Karel Piet Jan laat leren zwemmen’

(γ_6)



‘dat Karel Piet Jan laat leren zwemmen’

(γ_6)



(Kroch & Santorini 1991)

Some formal properties

- TAGs (even TIGs) strongly lexicalize CFGs

Some formal properties

- TAGs (even TIGs) strongly lexicalize CFGs
- Lexicalized tree adjoining grammars (LTAGs):
  each elementary tree has at least one lexical anchor

Some formal properties

- $\mathcal{CFL} \subsetneq \mathcal{TAL} \subsetneq \mathcal{CSL}$

Some formal properties

- $\mathcal{CFL} \subsetneq \mathcal{TAL} \subsetneq \mathcal{CSL}$

  But

Some formal properties

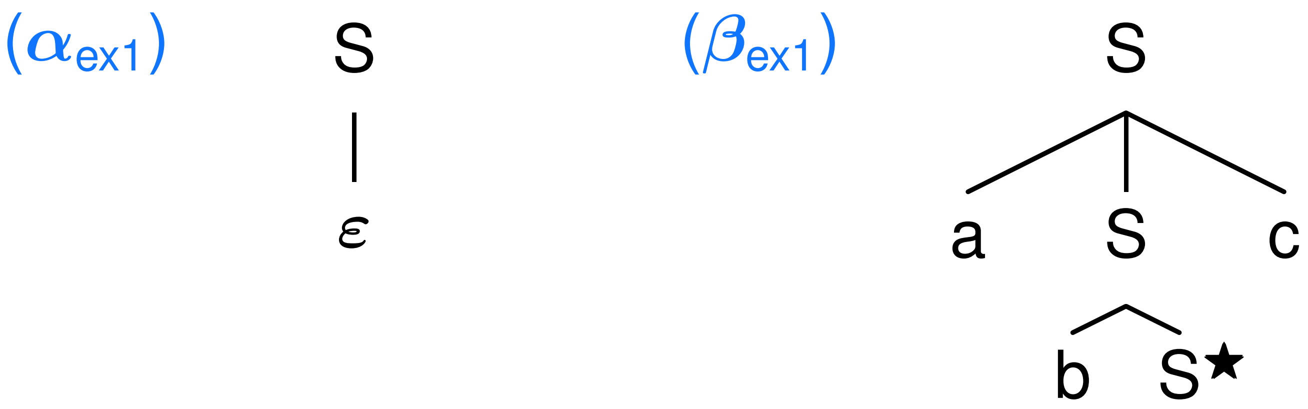
- $\mathcal{CFL} \subsetneq \mathcal{TAL} \subsetneq \mathcal{CSL}$

  But so far, $\mathcal{TAL}$ is not an AFL, because in particular, it is not closed under intersection with regular languages.

- $\mathcal{CFL} \subsetneq \mathcal{TAL} \subsetneq \mathcal{CSL}$

  But so far, $\mathcal{TAL}$ is not an AFL, because in particular, it is not closed under intersection with regular languages.

- Consider the TAG G_{ex1} whose elementary trees are the following :



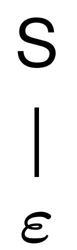
Some formal properties

- $\mathcal{CFL} \subsetneq \mathcal{TAL} \subsetneq \mathcal{CSL}$

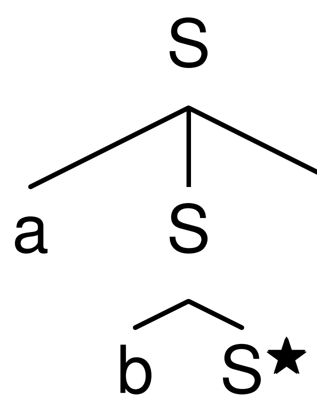
  But so far, $\mathcal{TAL}$ is not an AFL, because in particular, it is not closed under intersection with regular languages.

- Consider the TAG G_{ex1} whose elementary trees are the following :

(α_{ex1})

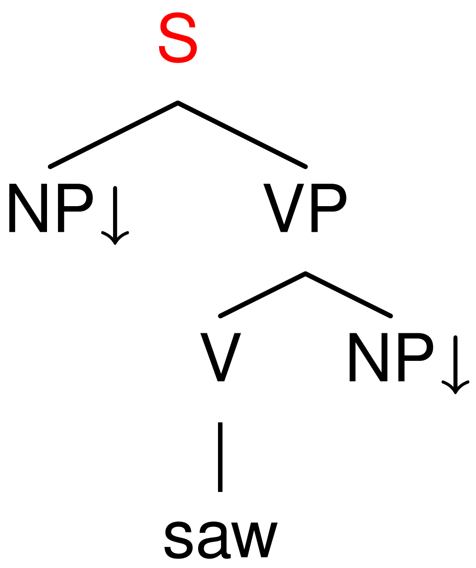


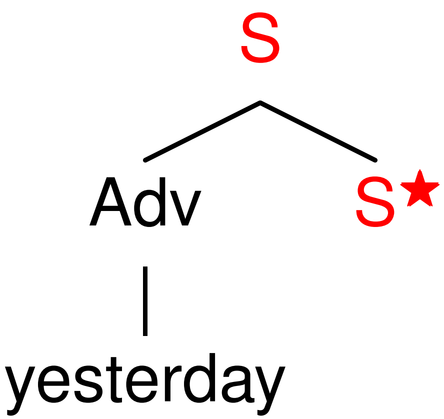
(β_{ex1})



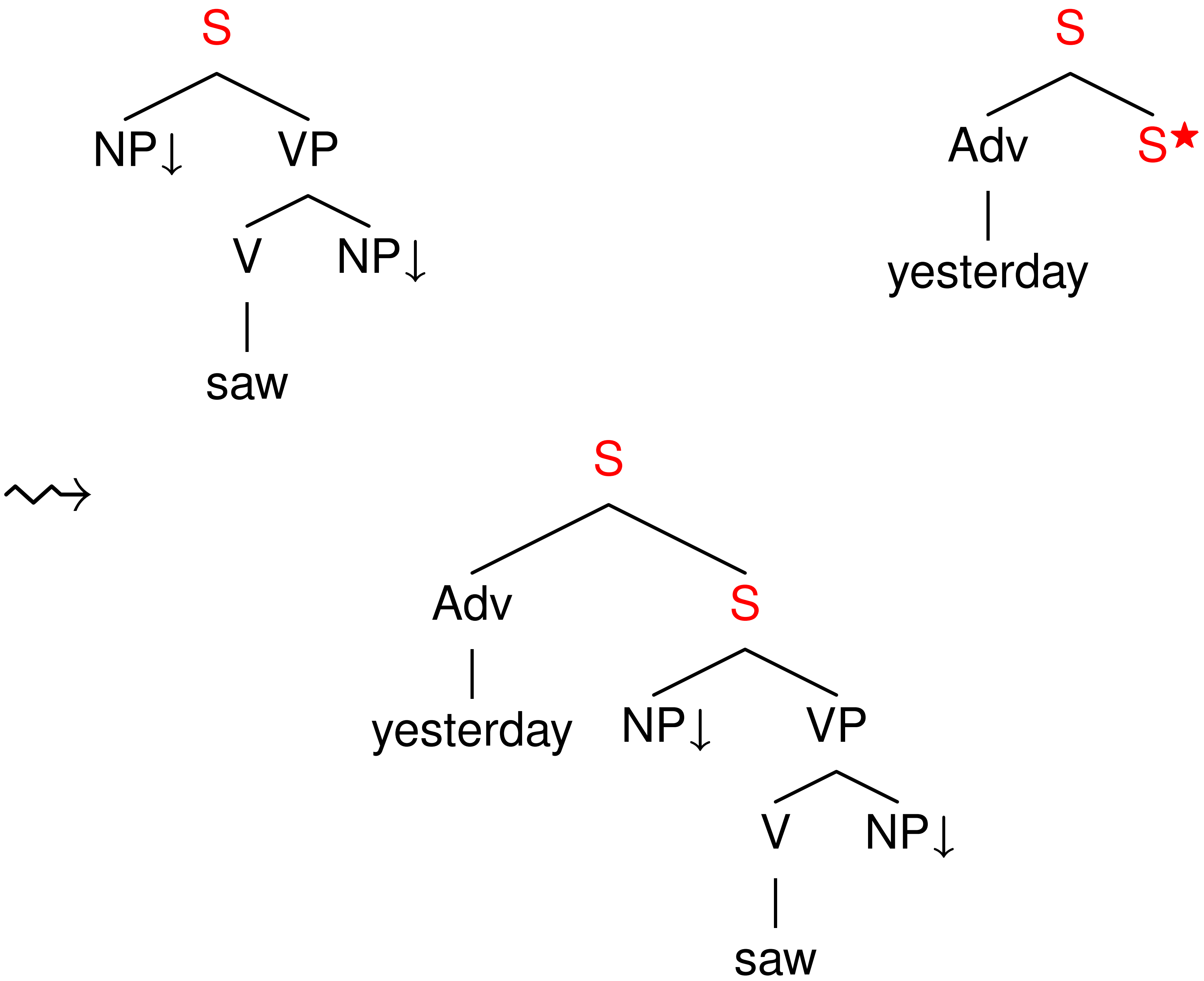
$$L(G_{ex1}) \cap \{a^k b^l c^m \mid k, l, m \geq 0\} = \{a^n b^n c^n \mid n \geq 0\}$$

Unrestricted adjoining

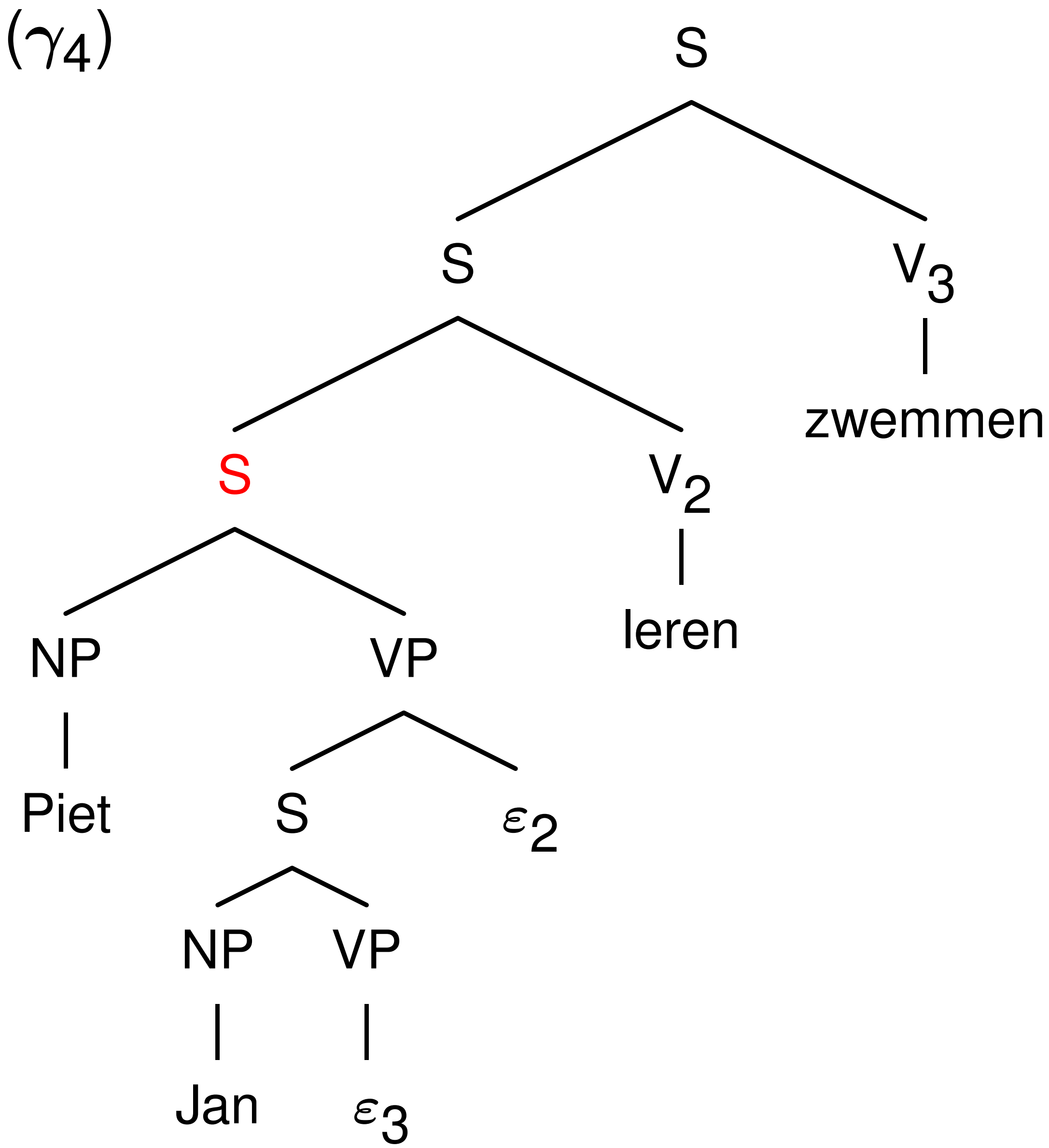


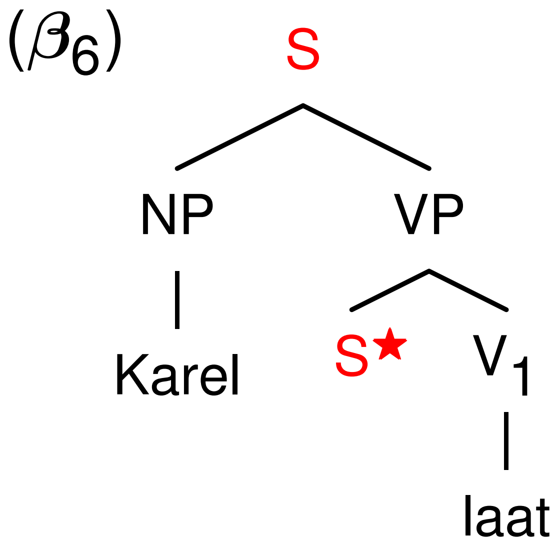


Unrestricted adjoining



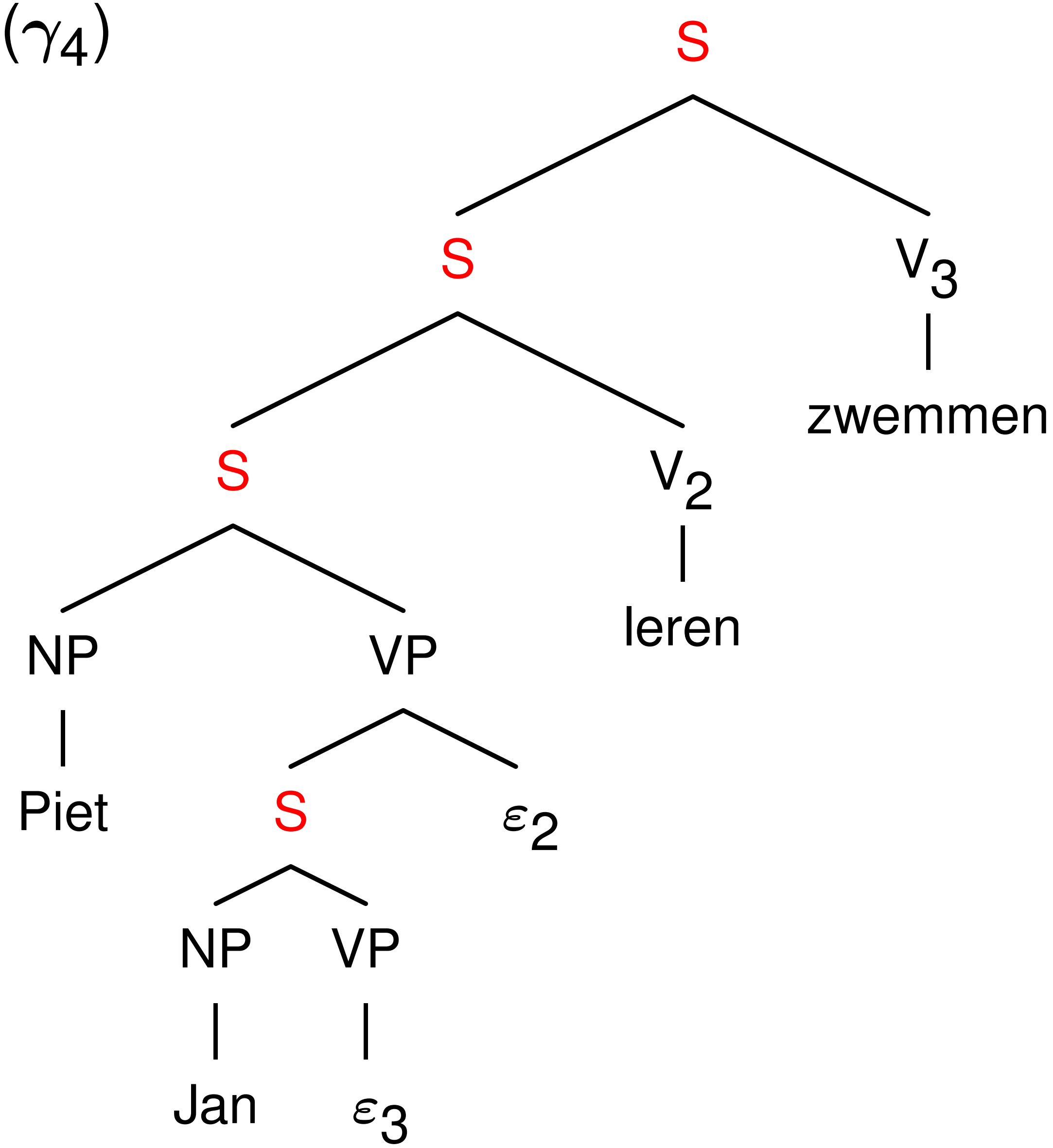
Unrestricted adjoining



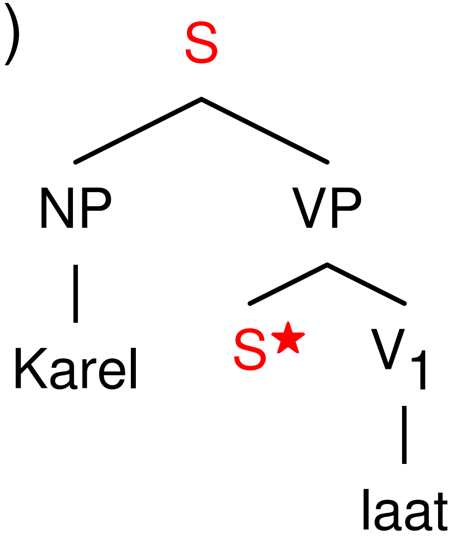


Unrestricted adjoining

(γ_4)

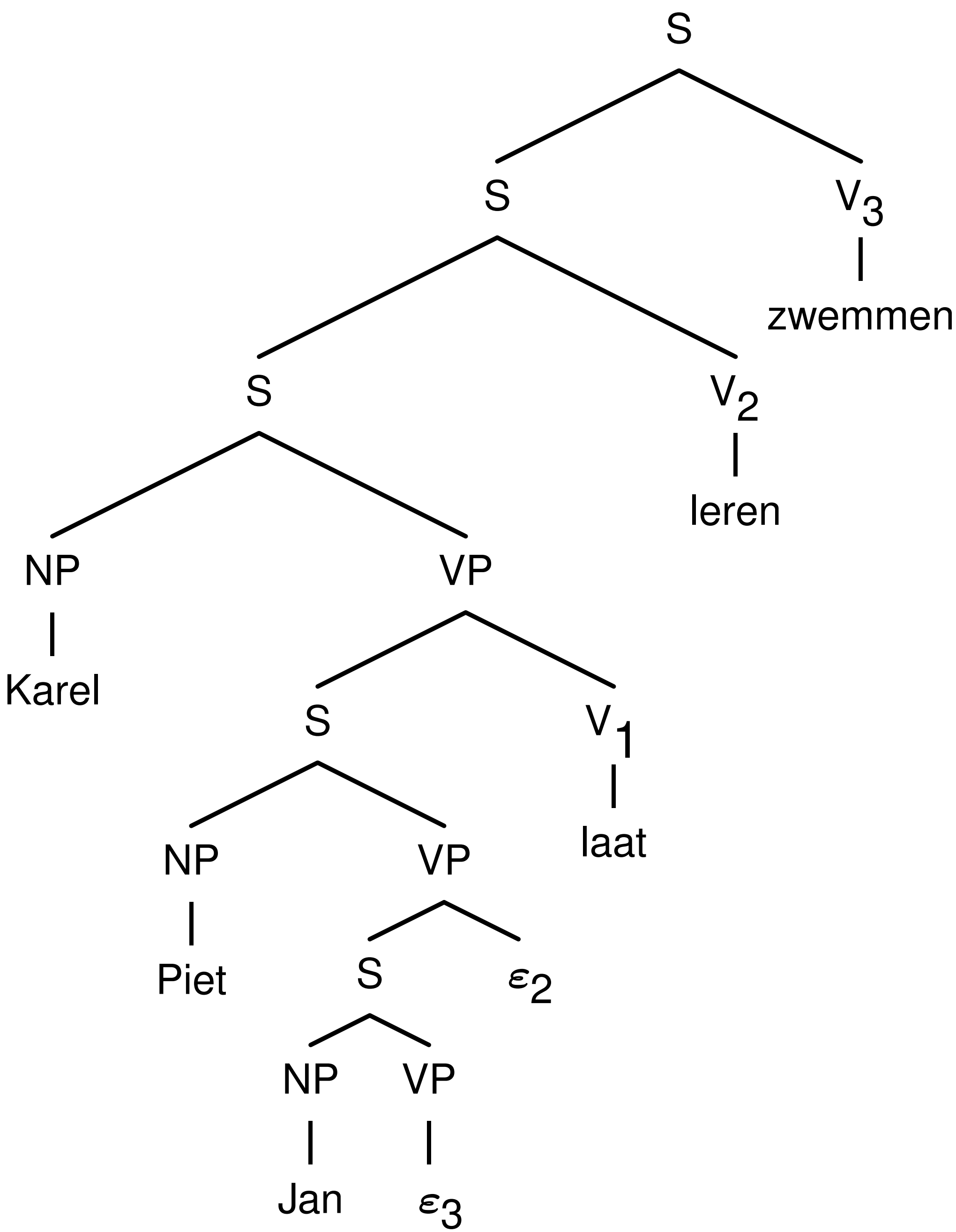


(β_6)



‘dat Karel Piet Jan laat leren zwemmen’

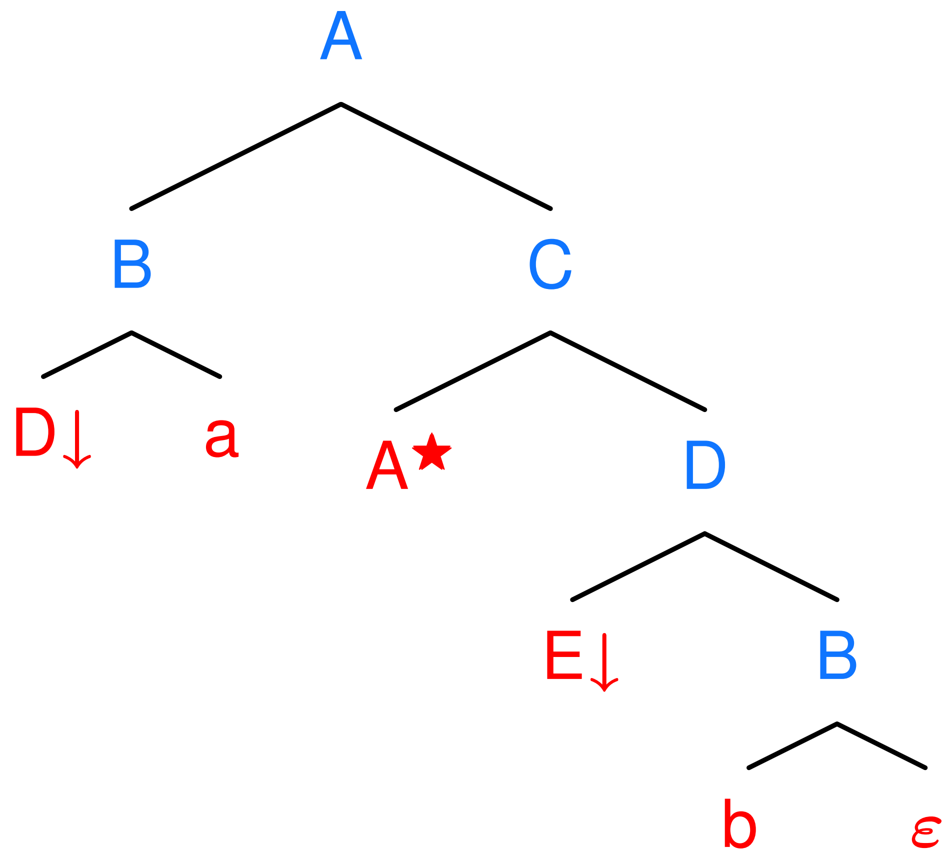
(γ_6)



Adding adjoining constraints

V_N a set of nonterminals

V_T a set of terminals



$$t = \langle N_t , \lhd_t^* , \prec_t , label_t \rangle$$

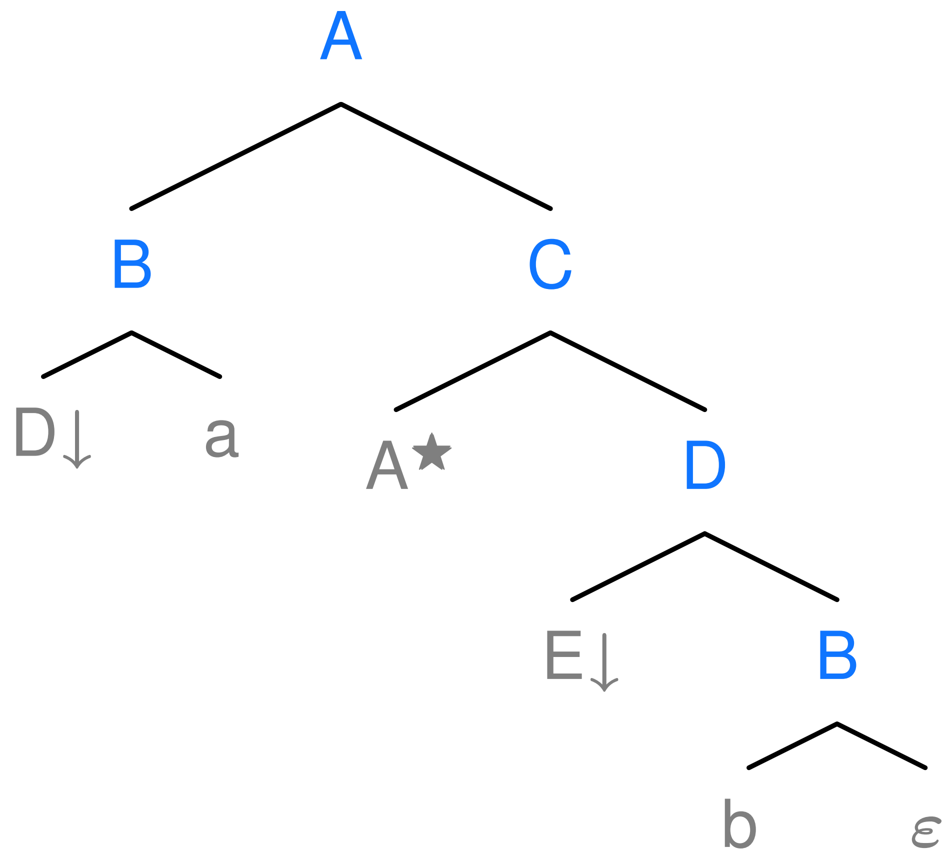
$$label_t : NonLeaves_t \rightarrow V_N$$

$$Leaves_t \rightarrow V_N\{\downarrow, ^\star\} \cup V_T \cup \{\varepsilon\}$$

Adding adjoining constraints

V_N a set of nonterminals

V_T a set of terminals



$$t = \langle N_t , \lhd_t^* , \prec_t , \text{label}_t \rangle$$

$$\text{label}_t : \text{NonLeaves}_t \rightarrow V_N$$

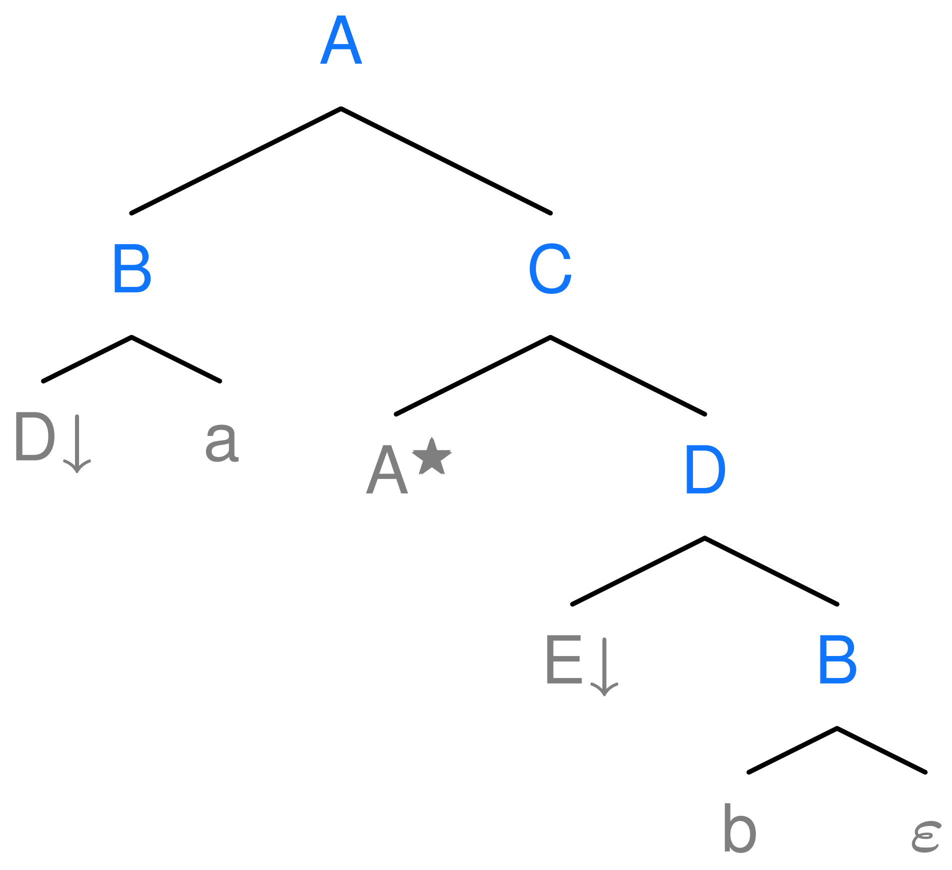
$$\text{Leaves}_t \rightarrow V_N\{\downarrow, \star\} \cup V_T \cup \{\varepsilon\}$$

Adding adjoining constraints

V_N a set of nonterminals

V_T a set of terminals

T_{Aux} a set of auxiliary trees



$$t = \langle N_t , \triangleleft_t^* , \prec_t , label_t \rangle$$

$$label_t : NonLeaves_t \rightarrow V_N$$

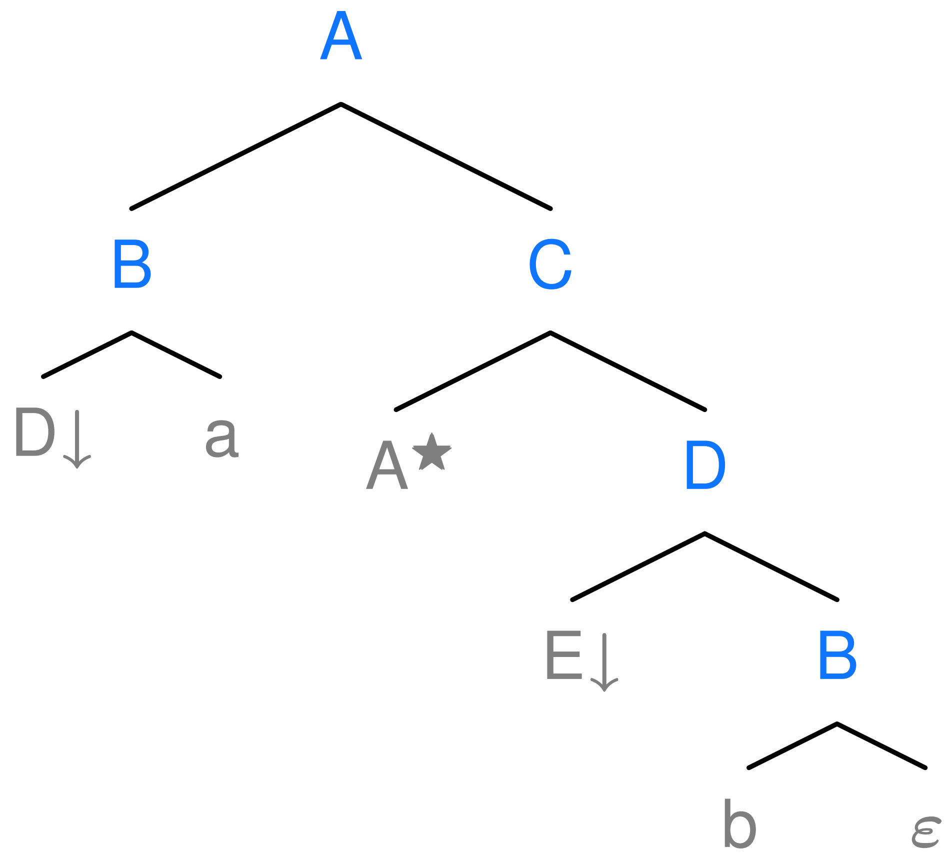
$$Leaves_t \rightarrow V_N\{\downarrow ,^\star\} \cup V_T \cup \{\varepsilon\}$$

Adding adjoining constraints

V_N a set of nonterminals

V_T a set of terminals

T_{Aux} a set of auxiliary trees



$$t = \langle N_t , \triangleleft_t^* , \prec_t , \text{label}_t \rangle$$

$$\text{label}_t : \text{NonLeaves}_t \rightarrow V_N \times 2^{T_{Aux}} \times \{+,-\}$$

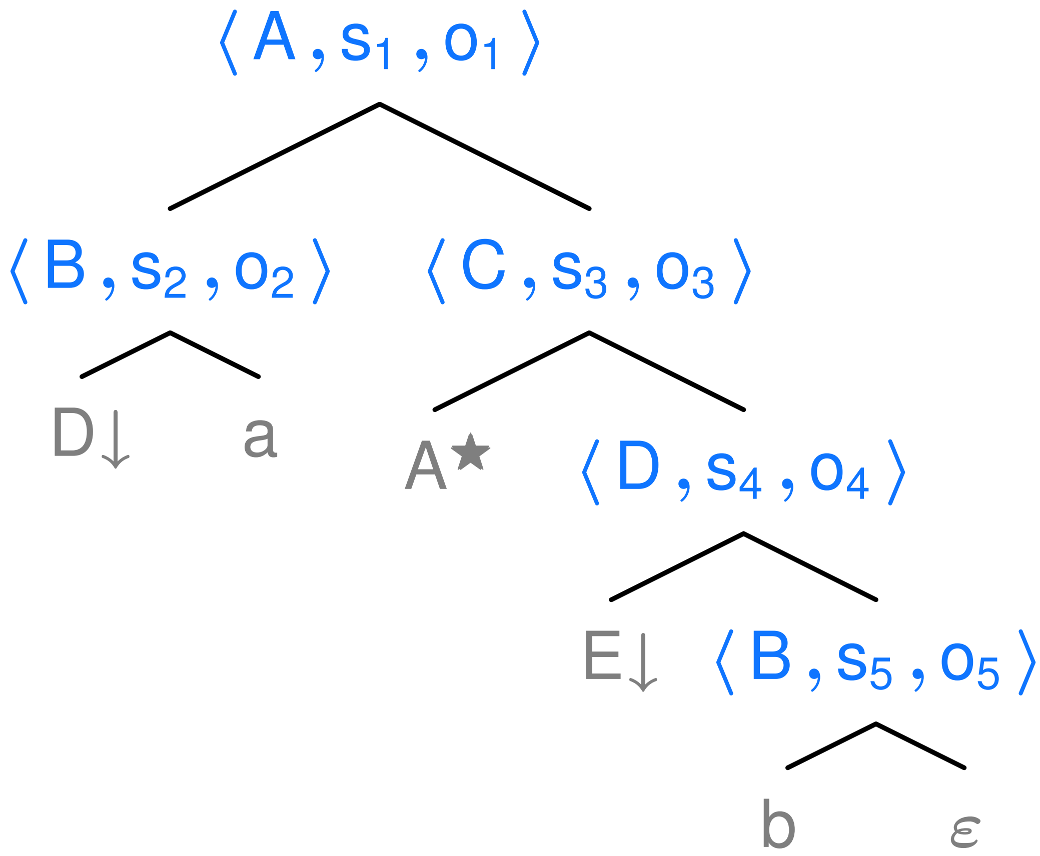
$$\text{Leaves}_t \rightarrow V_N\{\downarrow,\star\} \cup V_T \cup \{\varepsilon\}$$

Adding adjoining constraints

V_N a set of nonterminals

V_T a set of terminals

T_{Aux} a set of auxiliary trees



$$t = \langle N_t , \lhd_t^* , \prec_t , \mathrm{label}_t \rangle$$

$$\mathrm{label}_t : \mathrm{NonLeaves}_t \rightarrow V_N \times 2^{T_{Aux}} \times \{+,-\}$$

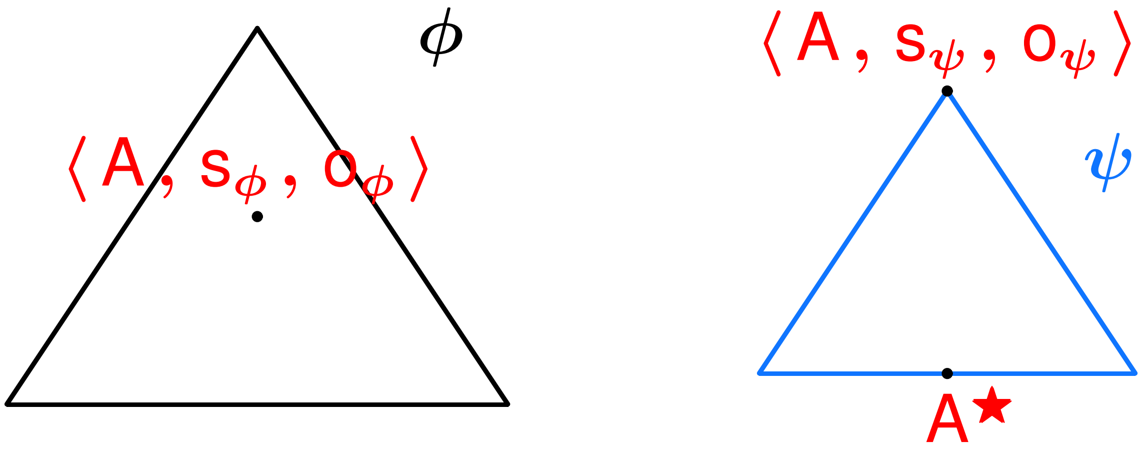
$$\mathrm{Leaves}_t \rightarrow V_N\{\downarrow, \star\} \cup V_T \cup \{\varepsilon\}$$

$$\text{adjoining} : \text{Trees}(V) \times \text{Trees}(V) \xrightarrow{\text{part}} 2^{\text{Trees}(V)}$$

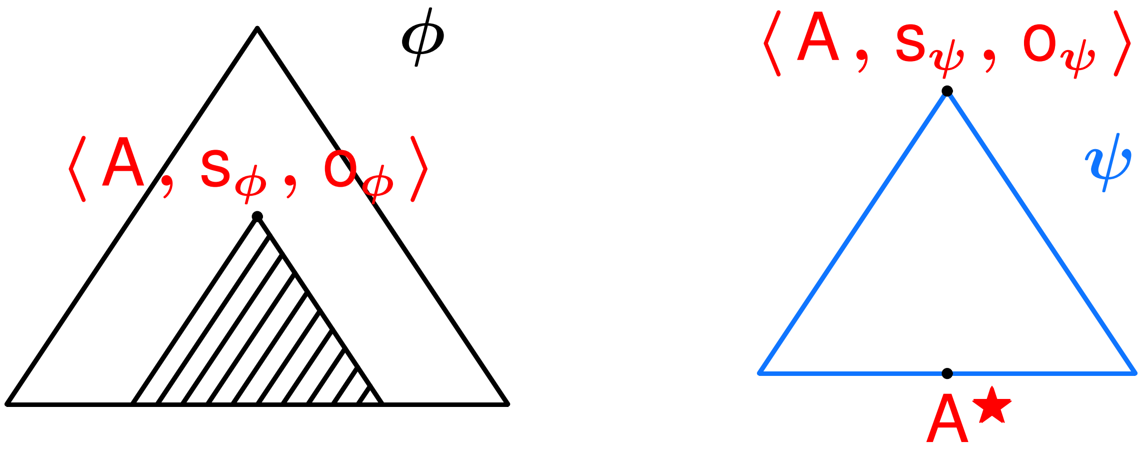
$\langle \phi , \psi \rangle \in \text{Domain}(\text{adjoining}) \;:\Longleftrightarrow$

- ψ's root is labeled $\langle A , s_\psi , o_\psi \rangle$ for some $s_\psi \subseteq T_{Aux}$ and $o_\psi \in \{+,-\}$, and ψ has a leaf labeled $A^\star$
- ϕ has a node labeled $\langle A , s_\phi , o_\phi \rangle$ for some $A \in V_N$, $s_\phi \subseteq T_{Aux}$ and $o_\phi \in \{+,-\}$ such that $\psi \in s_\phi$

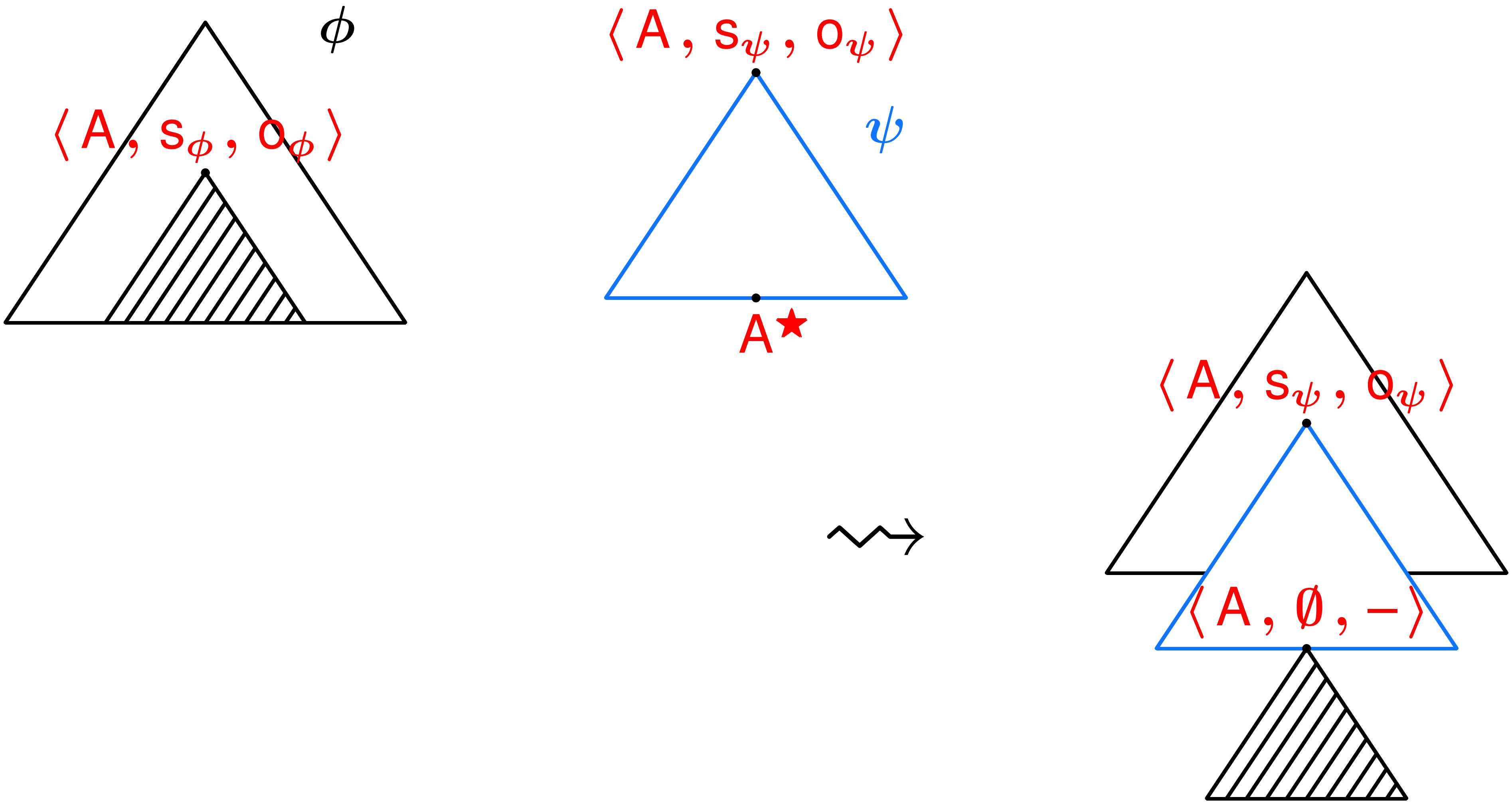
$$\text{adjoining} : \text{Trees}(V) \times \text{Trees}(V) \xrightarrow{\text{part}} 2^{\text{Trees}(V)}$$



$$\text{adjoining} : \text{Trees}(V) \times \text{Trees}(V) \xrightarrow{\text{part}} 2^{\text{Trees}(V)}$$



$$\text{adjoining} : \text{Trees}(V) \times \text{Trees}(V) \xrightarrow{\text{part}} 2^{\text{Trees}(V)}$$



$\mathsf{Closure}(G)$, the closure of a TAG $G = \langle V_N, V_T, T_{Ini}, T_{Aux}, S \rangle$,

is the closure of $T_{Ini} \cup T_{Aux}$ under substitution and adjoining.

$\mathsf{Closure}(G)$, the closure of a TAG $G = \langle V_N, V_T, T_{Ini}, T_{Aux}, S \rangle$,

is the closure of $T_{Ini} \cup T_{Aux}$ under substitution and adjoining.

$t \in \mathsf{Closure}(G)$ is complete $:\Longleftrightarrow$

each non-leaf-label is of the form $\langle A, s, - \rangle$ for some $A \in V_N$ and $s \subseteq T_{Aux}$, $A = S$ in case of the root, and $\mathsf{yield}(t) \in \mathsf{Strings}(V_T)$.

Closure(G), the closure of a TAG $G = \langle V_N, V_T, T_{Ini}, T_{Aux}, S \rangle$,

is the closure of $T_{Ini} \cup T_{Aux}$ under substitution and adjoining.

$t \in$ Closure(G) is complete $:\Longleftrightarrow$

each non-leaf-label is of the form $\langle A, s, - \rangle$ for some $A \in V_N$ and $s \subseteq T_{Aux}$, $A = S$ in case of the root, and $\text{yield}(t) \in \text{Strings}(V_T)$.

The tree and string language generated by G

$$T(G) = \{ t \mid t \in \text{Closure}(G) \text{ and complete} \}$$

$$L(G) = \{ \text{yield}(t) \mid t \in T(G) \}$$

Some formal properties

- $\mathcal{CFL} \subsetneq \mathcal{TAL} \subsetneq \mathcal{CSL}$

Some formal properties

- $\mathcal{CFL} \subsetneq \mathcal{TAL} \subsetneq \mathcal{CSL}$

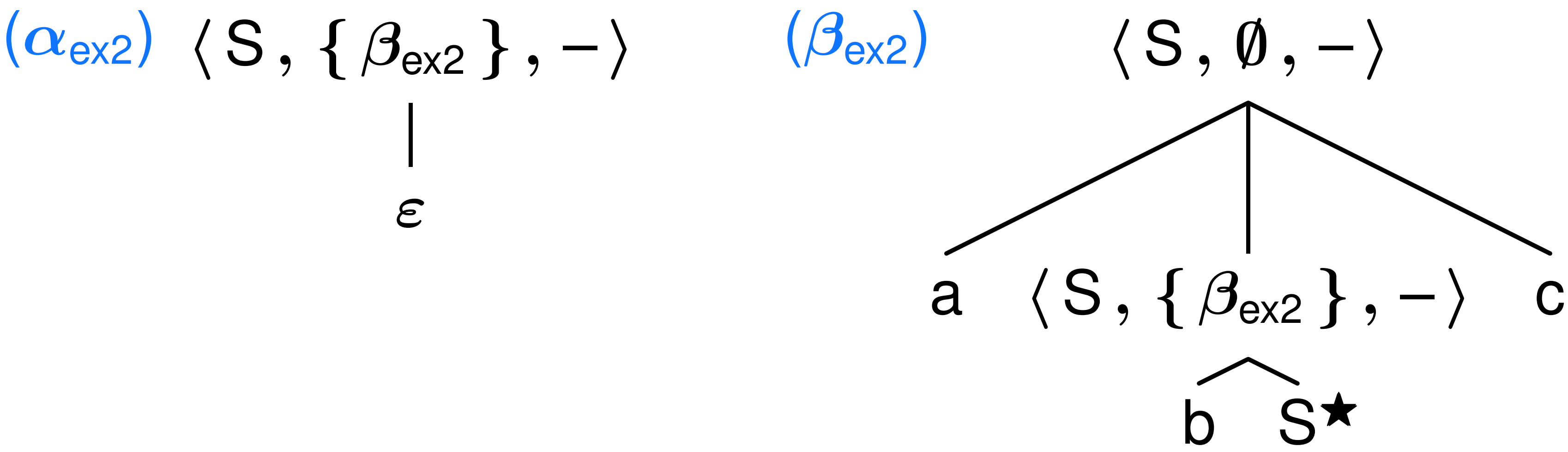
  Now $\mathcal{TAL}$ is a substitution-closed full AFL.

Some formal properties

- $\mathcal{CFL} \subsetneq \mathcal{TAL} \subsetneq \mathcal{CSL}$

  Now $\mathcal{TAL}$ is a substitution-closed full AFL.

- Consider the TAG G_{ex2} whose elementary trees are the following :

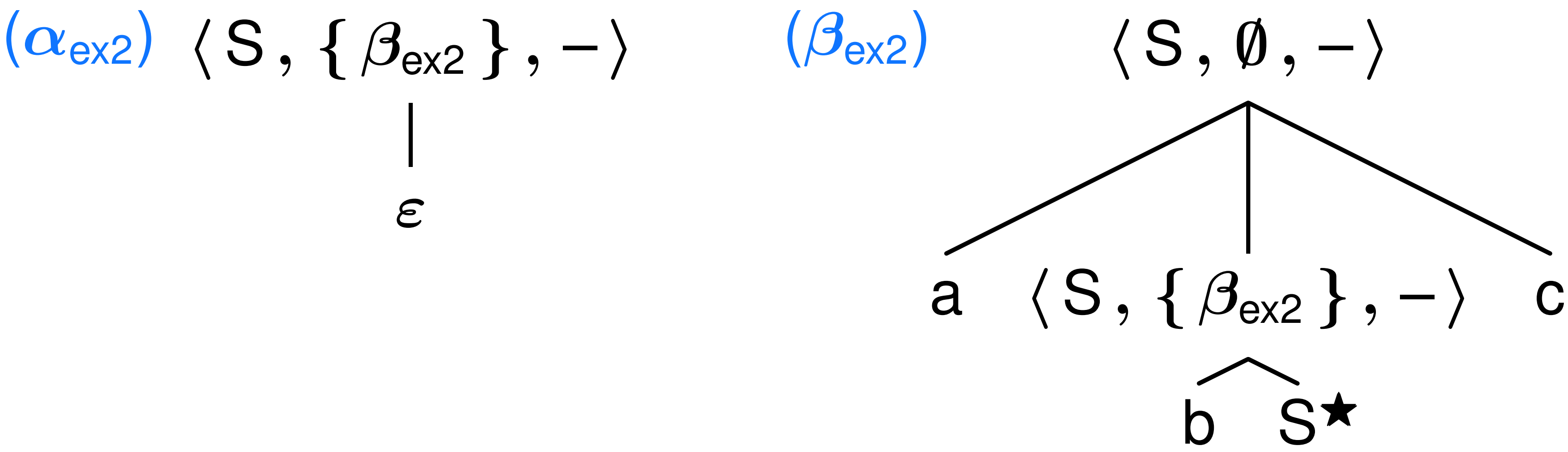


Some formal properties

- $\mathcal{CFL} \subsetneq \mathcal{TAL} \subsetneq \mathcal{CSL}$

  Now $\mathcal{TAL}$ is a substitution-closed full AFL.

- Consider the TAG G_{ex2} whose elementary trees are the following :



$L(G_{ex2}) = \{a^n b^n c^n \mid n \geq 0\}$

Extensions of TAGs: multicomponent TAGs

$$G = \langle V_N , V_T , T_{Ini} , Tuples_{Aux} , S \rangle$$

$$G = \langle V_N, V_T, T_{Ini}, Tuples_{Aux}, S \rangle$$

- V_N a set of nonterminals

Extensions of TAGs: multicomponent TAGs

$$G = \langle V_N , V_T , T_{Ini} , Tuples_{Aux} , S \rangle$$

- V_N a set of nonterminals
- V_T a set of terminals

Extensions of TAGs: multicomponent TAGs

$$G = \langle V_N , V_T , T_{Ini} , Tuples_{Aux} , S \rangle$$

- V_N a set of nonterminals
- V_T a set of terminals
- T_{Ini} a finite set of initial trees

Extensions of TAGs: multicomponent TAGs

$$G = \langle V_N, V_T, T_{Ini}, Tuples_{Aux}, S \rangle$$

- V_N a set of nonterminals
- V_T a set of terminals
- T_{Ini} a finite set of initial trees
- $Tuples_{Aux}$ a finite set of finite tuples of auxiliary trees

Extensions of TAGs: multicomponent TAGs

$$G = \langle V_N, V_T, T_{Ini}, Tuples_{Aux}, S \rangle$$

- V_N a set of nonterminals
- V_T a set of terminals
- T_{Ini} a finite set of initial trees
- $Tuples_{Aux}$ a finite set of finite tuples of auxiliary trees
- $S \in V_N$ a distinguished nonterminal (the start symbol)

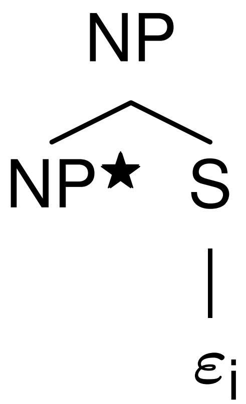
Extensions of TAGs: multicomponent TAGs

- Reasons for tree-local MCTAGs: “syntactic sugar”
- Reasons for set-local MCTAGs (?): “real” verb clusters

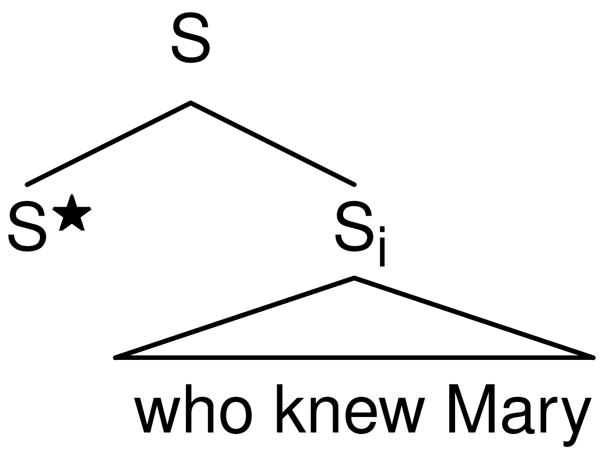
Elementary tree set: relative clause extraposed from subject

$\beta_3 = \langle \beta_{31} , \beta_{32} \rangle$

(β_{31})

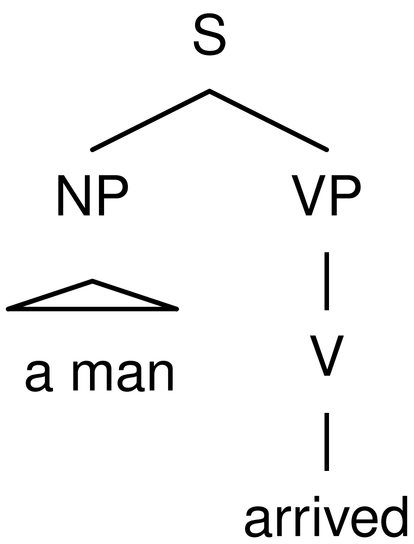


(β_{32})



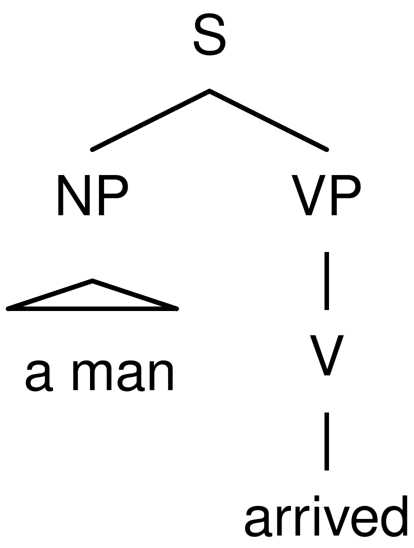
Relative clause extraposed from subject NP

(α_6)

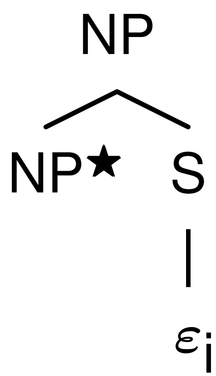


Relative clause extraposed from subject NP

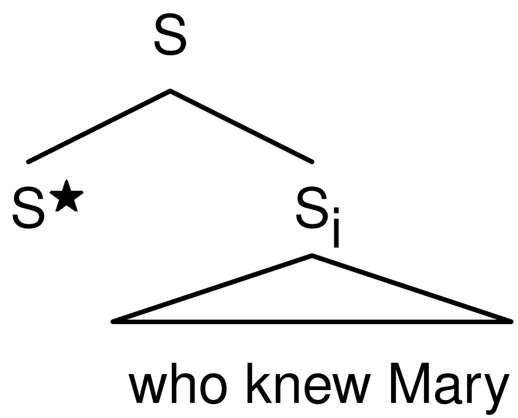
(α_6)



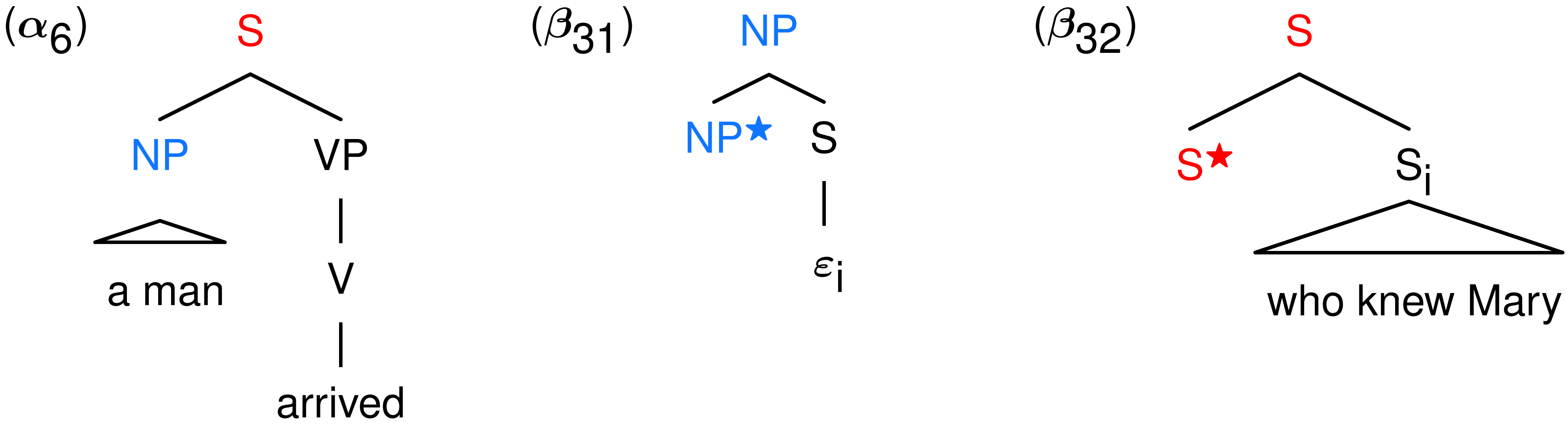
(β_{31})



(β_{32})

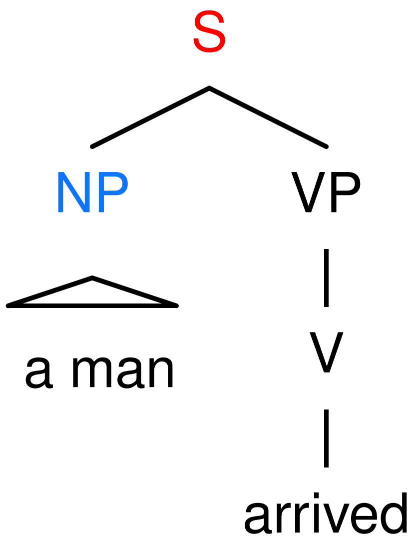


Relative clause extraposed from subject NP

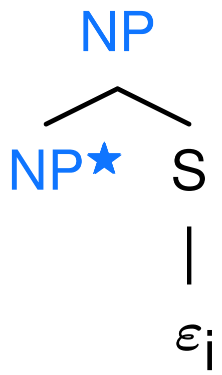


Relative clause extraposed from subject NP

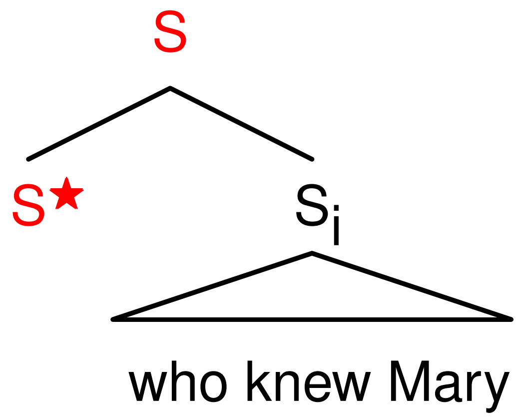
(α_6)



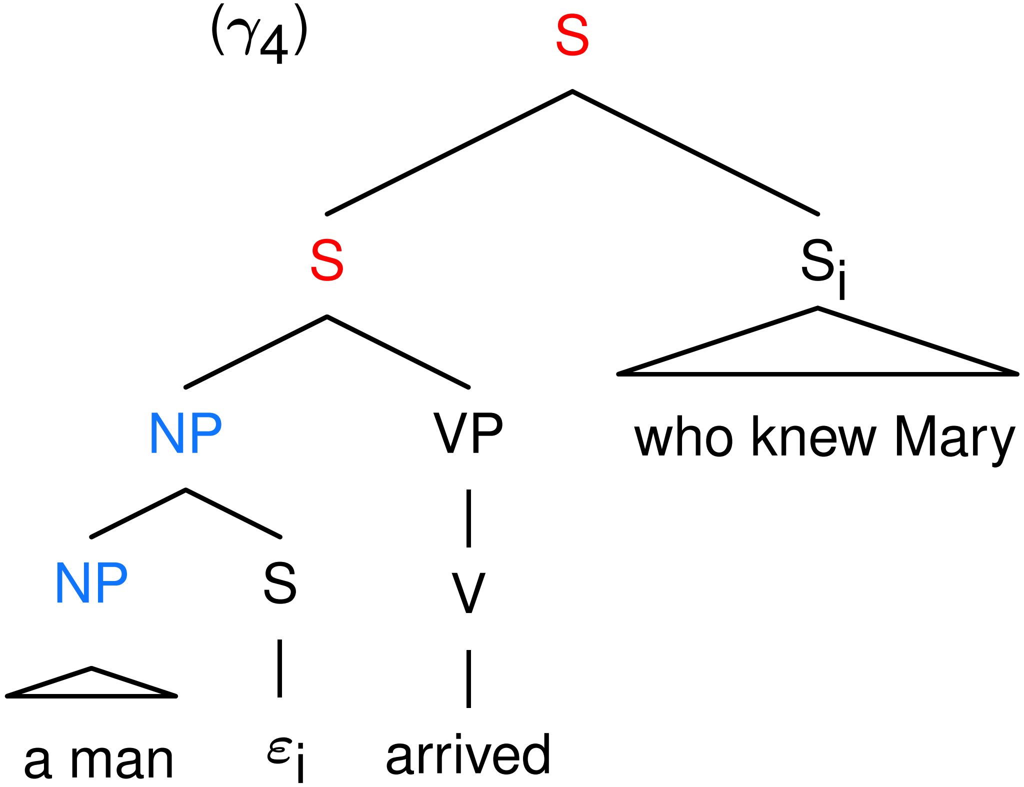
(β_{31})



(β_{32})



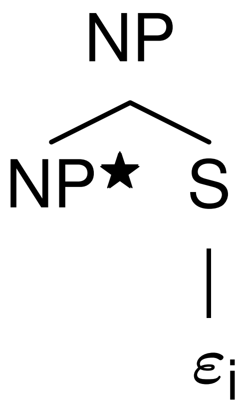
(γ_4)



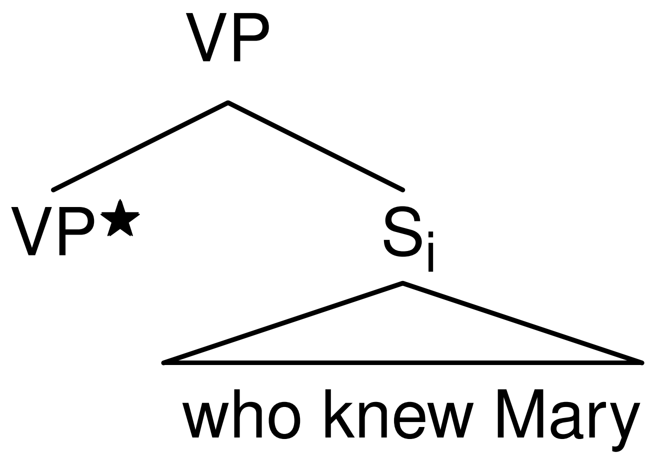
Elementary tree set: relative clause extraposed from object

$\beta_4 = \langle\, \beta_{41}\,, \beta_{42}\,\rangle$

(β_{41})



(β_{42})

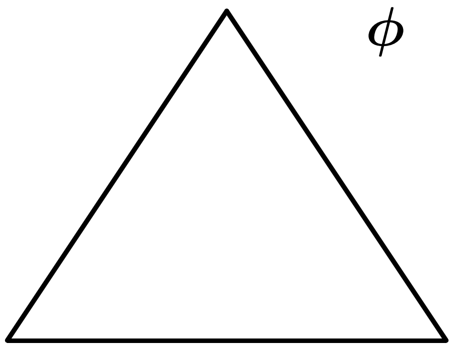


Tree-local multicomponent TAGs (MCTAGs)

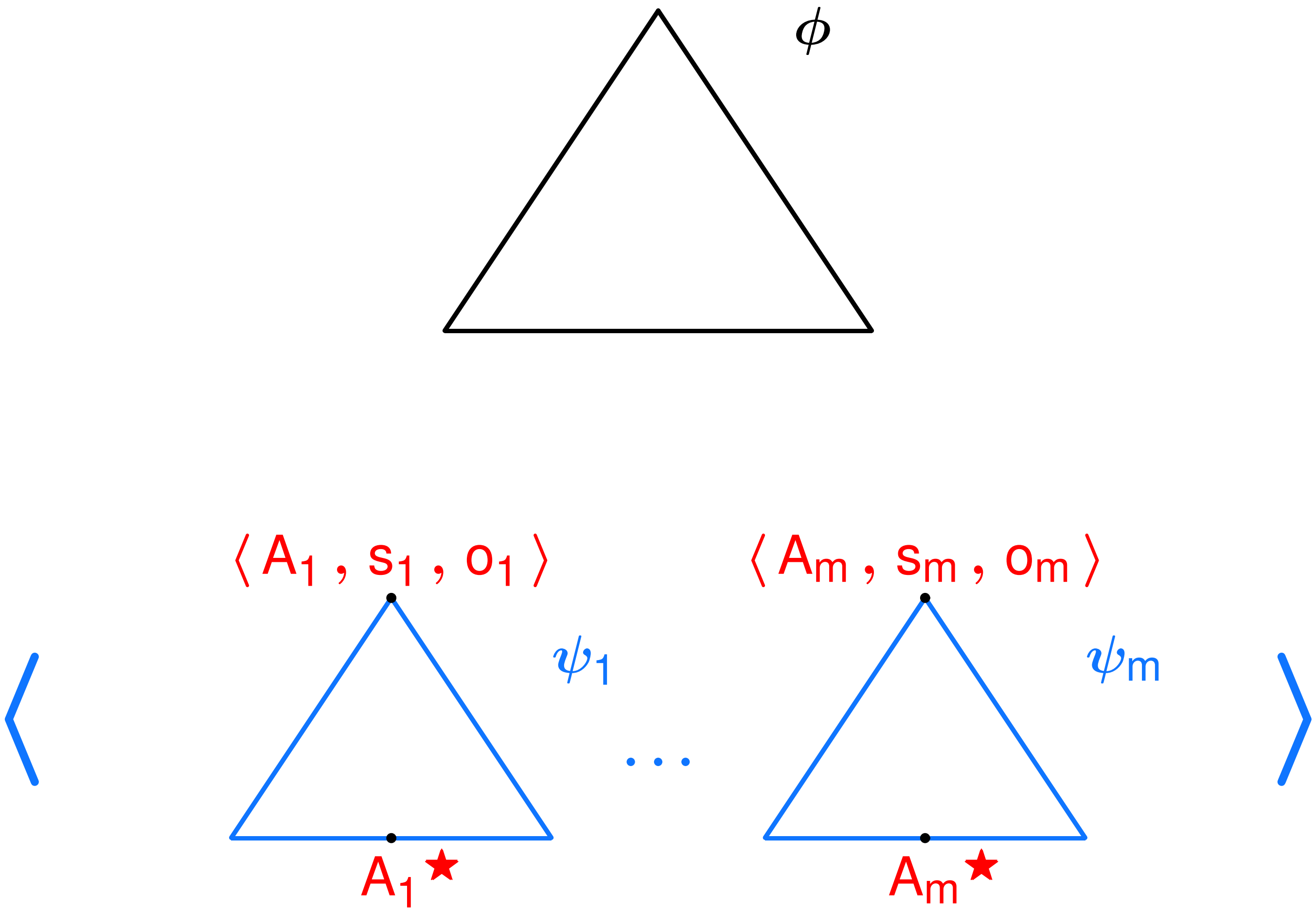
$$G = \langle V_N, V_T, T_{Ini}, Tuples_{Aux}, S \rangle$$

- V_N a set of nonterminals
- V_T a set of terminals
- T_{Ini} a finite set of initial trees
- $Tuples_{Aux}$ a finite set of finite tuples of auxiliary trees
- $S \in V_N$ a distinguished nonterminal (the start symbol)

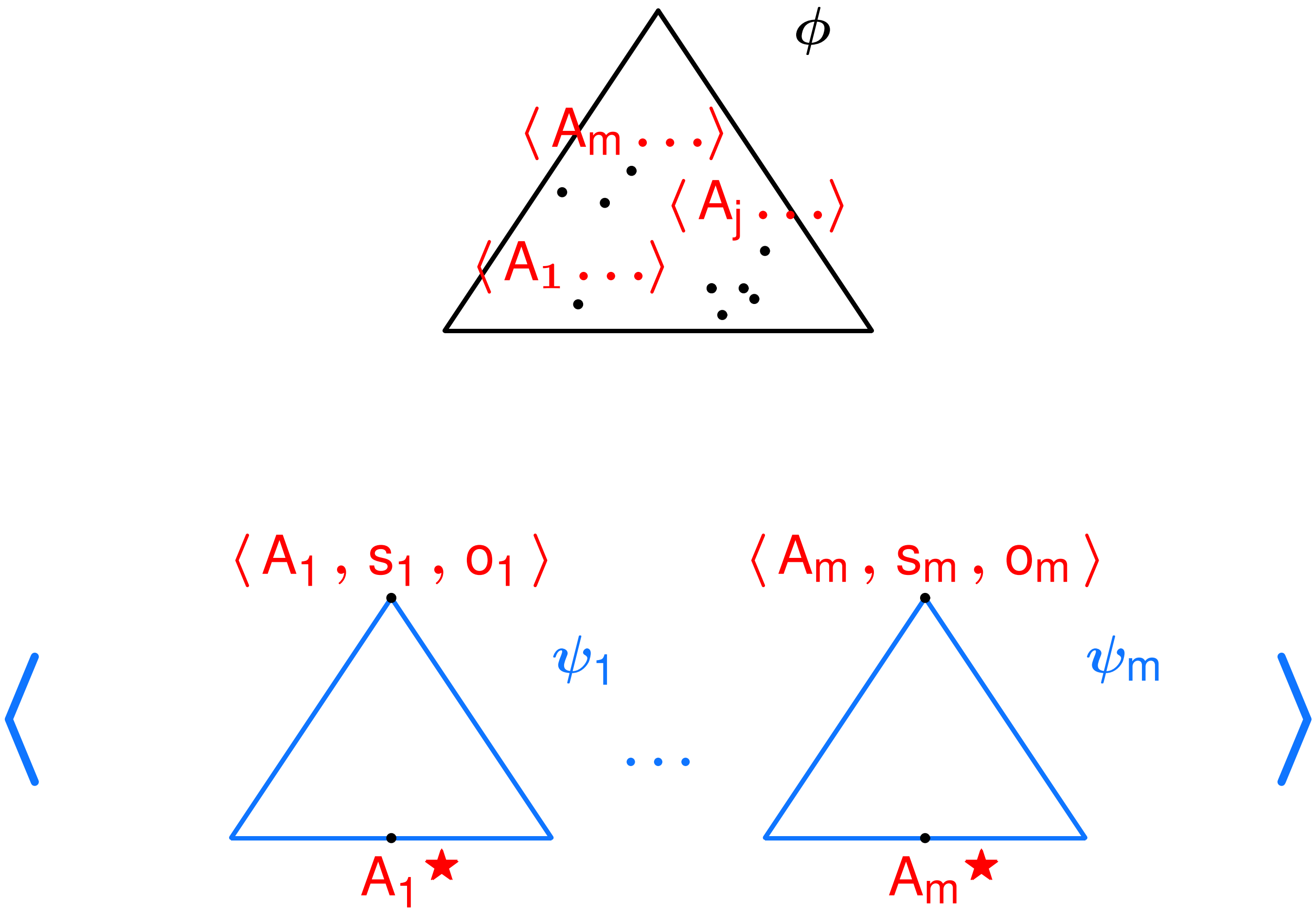
$$\text{adjoining} : \left(T_{Ini} \cup \text{Tuples}_{Aux}\right) \times \text{Treetuples}(V) \xrightarrow{\text{part}} 2^{\text{Treetuples}(V)}$$



$$\text{adjoining}: \left(T_{Ini} \cup \text{Tuples}_{Aux}\right) \times \text{Treetuples}(V) \xrightarrow{\text{part}} 2^{\text{Treetuples}(V)}$$



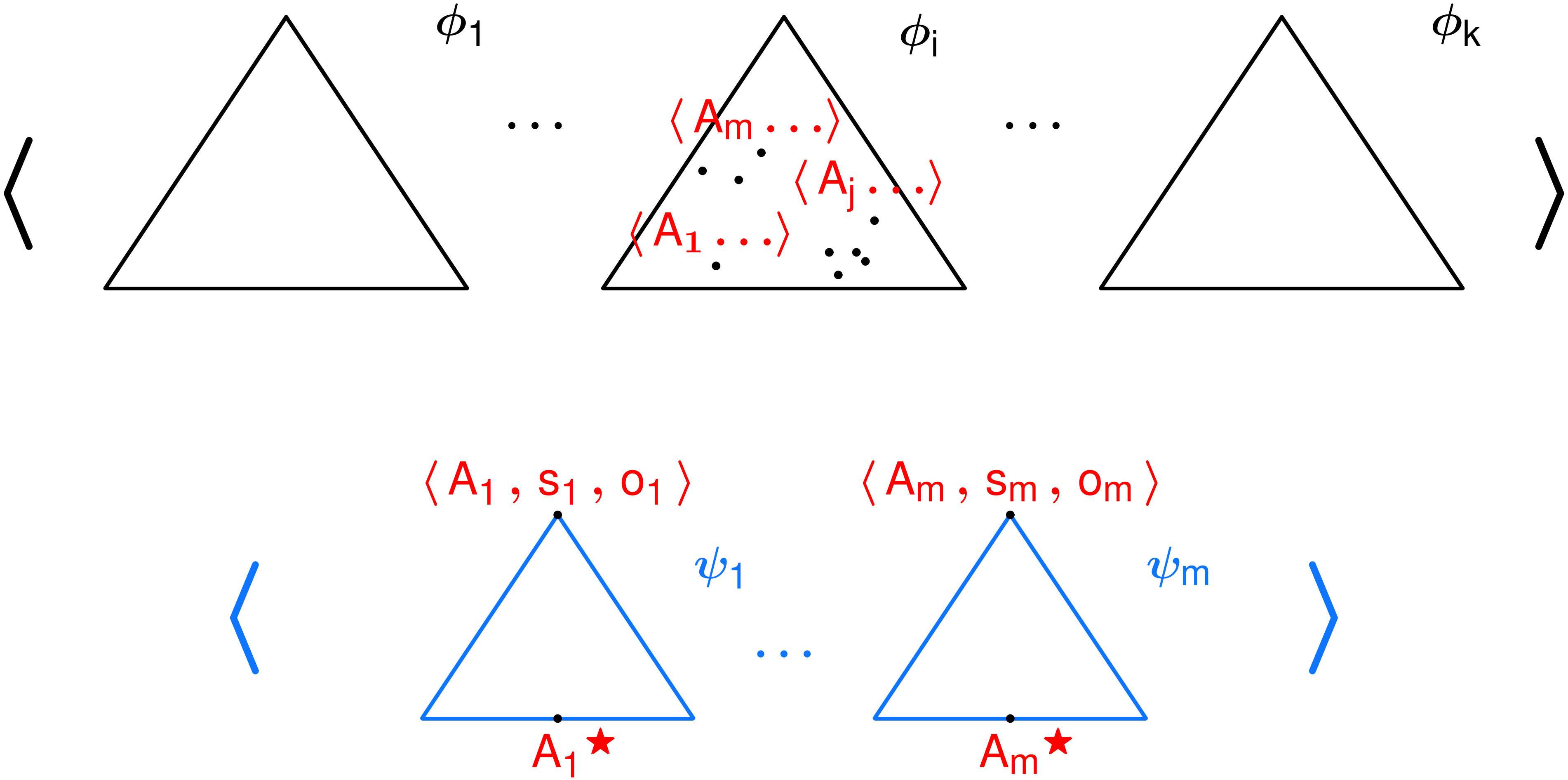
$$\text{adjoining} : \left(T_{Ini} \cup \text{Tuples}_{Aux} \right) \times \text{Treetuples}(V) \xrightarrow{\text{part}} 2^{\text{Treetuples}(V)}$$



Structure building operators

$V = V_N \cup V_T$

$$\text{adjoining}: \left(T_{Ini} \cup \text{Tuples}_{Aux}\right) \times \text{Treetuples}(V) \xrightarrow{\text{part}} 2^{\text{Treetuples}(V)}$$



Some formal properties

TAGs and tree-local MCTAGs are strongly equivalent !

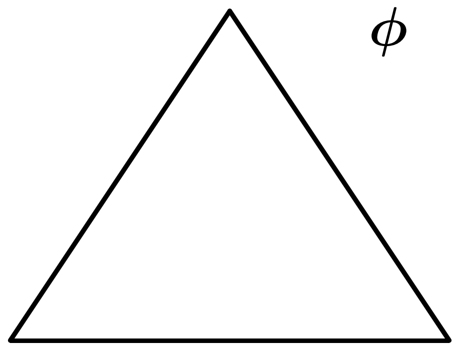
This does not hold for TAGs and set-local MCTAGs !

Set-local multicomponent TAGs (MCTAGs)

$G = \langle V_N, V_T, T_{Ini}, Tuples_{Aux}, S \rangle$

- V_N a set of nonterminals
- V_T a set of terminals
- T_{Ini} a finite set of initial trees
- $Tuples_{Aux}$ a finite set of finite tuples of auxiliary trees
- $S \in V_N$ a distinguished nonterminal (the start symbol)

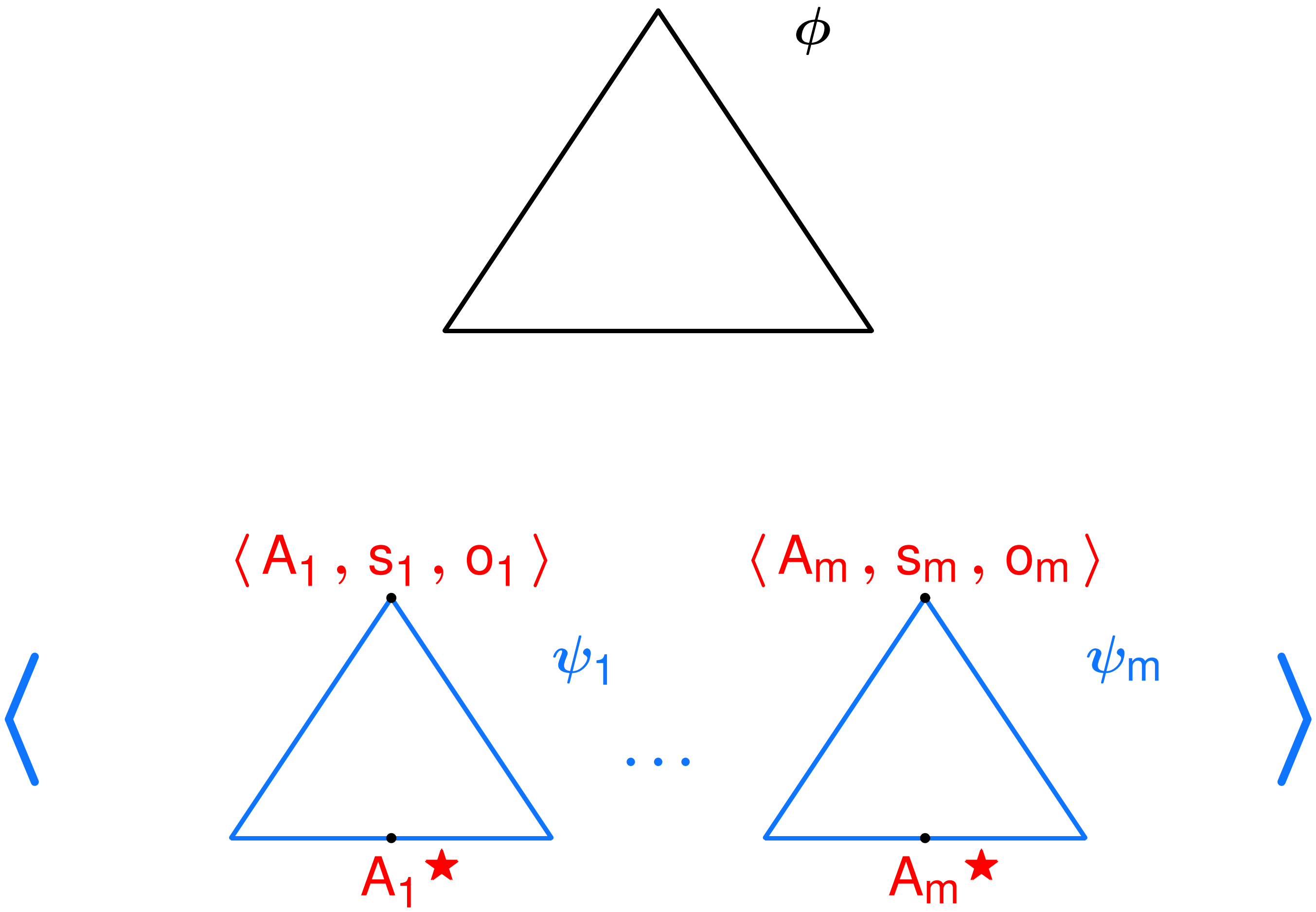
$$\text{adjoining}: \left(T_{Ini} \cup \text{Tuples}_{Aux}\right) \times \text{Treetuples}(V) \xrightarrow{\text{part}} 2^{\text{Treetuples}(V)}$$



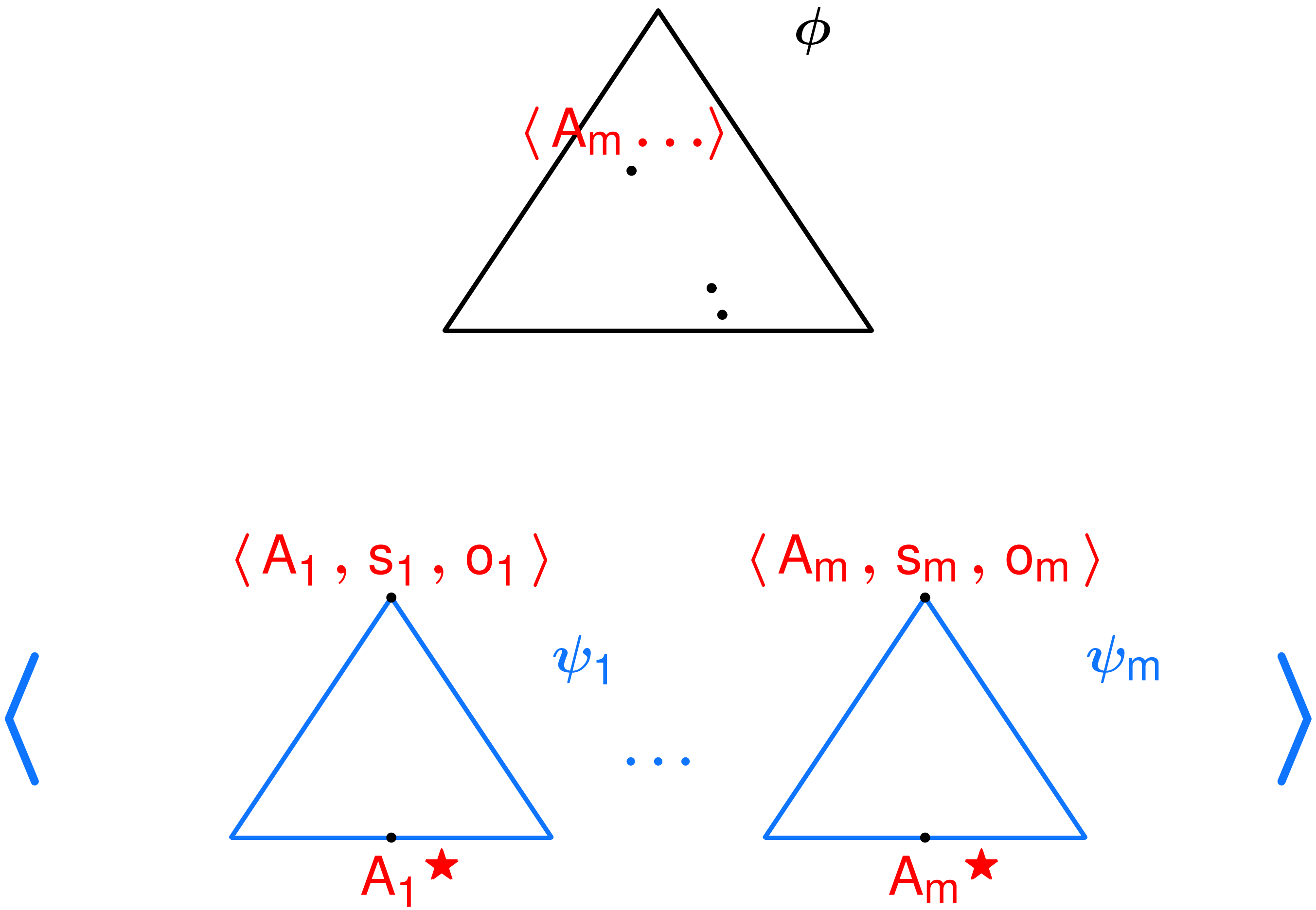
Structure building operators

$V = V_N \cup V_T$

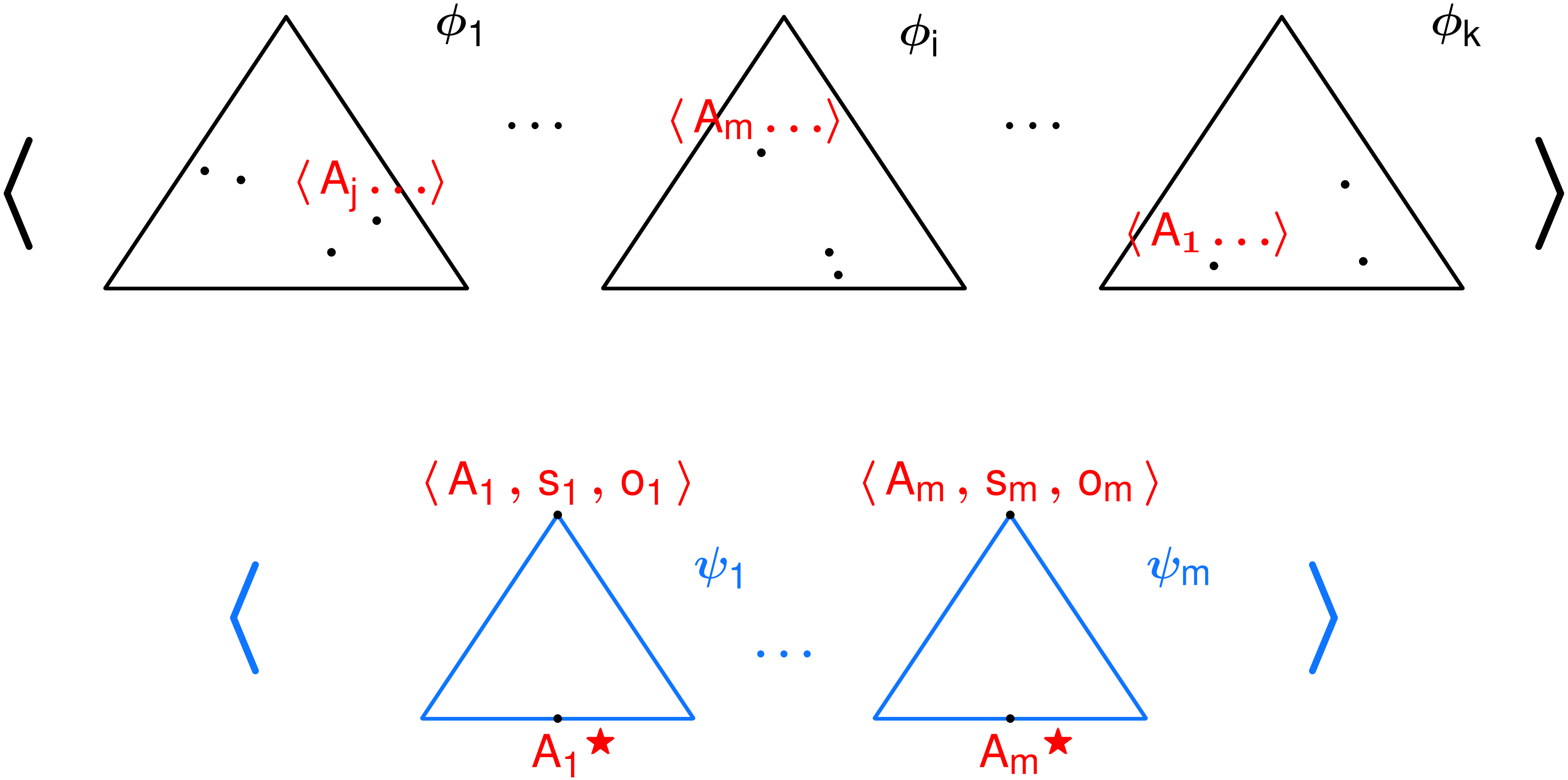
$$\text{adjoining}: \left(T_{Ini} \cup \text{Tuples}_{Aux}\right) \times \text{Treetuples}(V) \xrightarrow{\text{part}} 2^{\text{Treetuples}(V)}$$



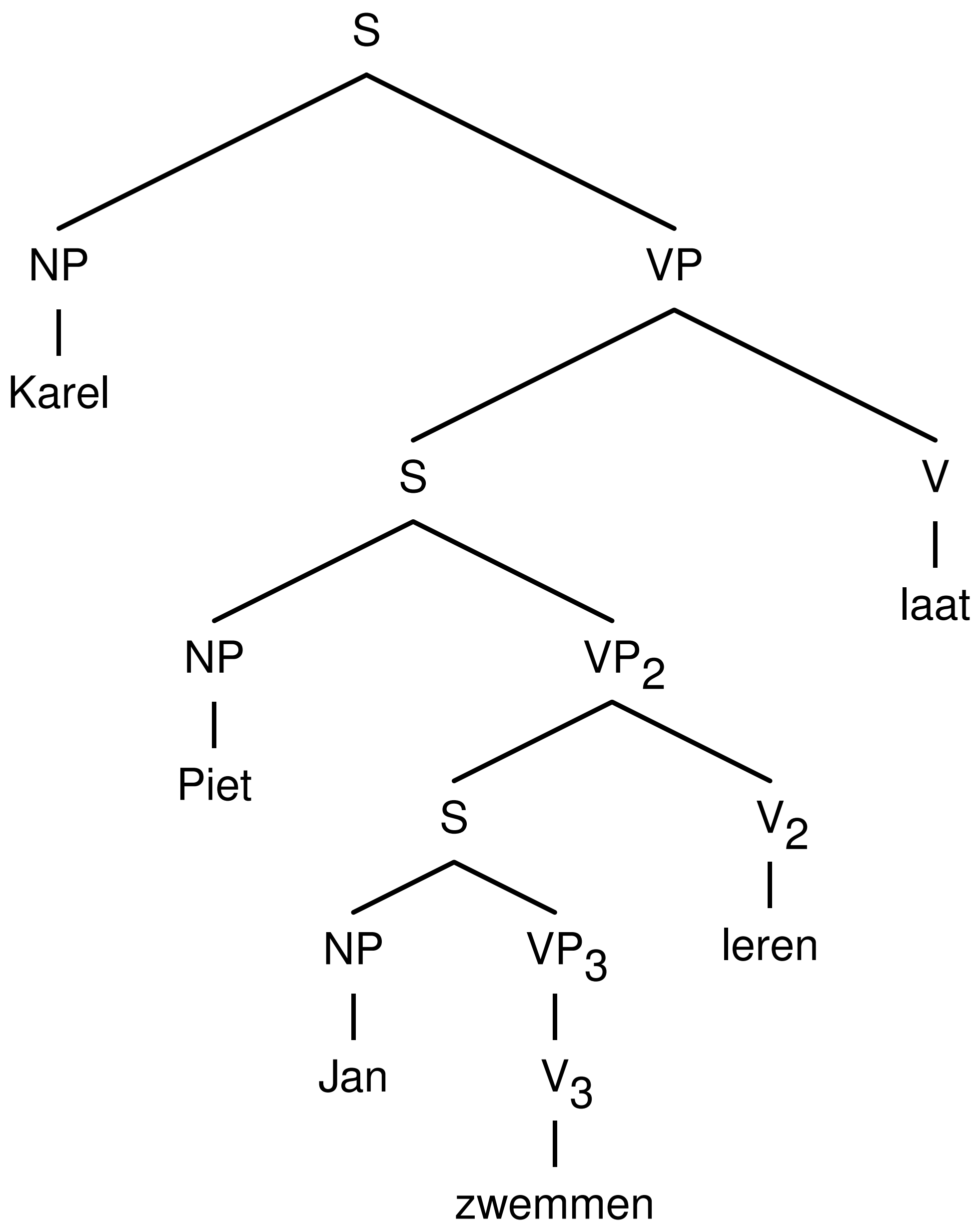
$$\text{adjoining}: \left(T_{Ini} \cup \text{Tuples}_{Aux}\right) \times \text{Treetuples}(V) \xrightarrow{\text{part}} 2^{\text{Treetuples}(V)}$$



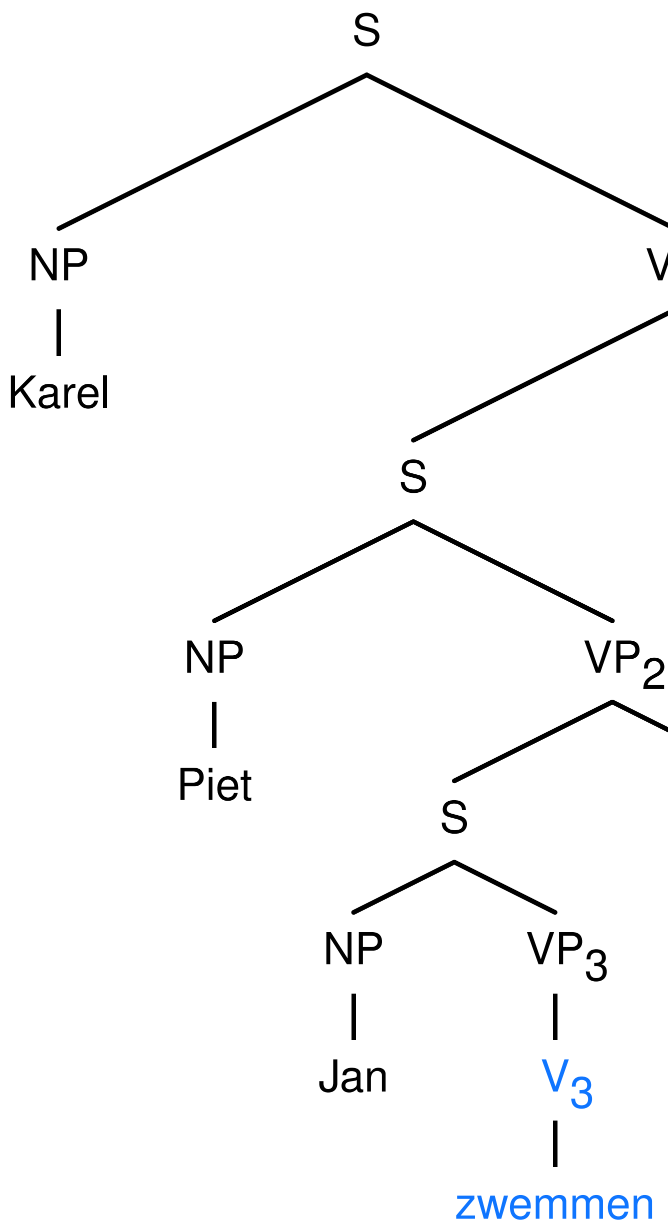
$$\text{adjoining}: \left(T_{Ini} \cup \text{Tuples}_{Aux}\right) \times \text{Treetuples}(V) \xrightarrow{\text{part}} 2^{\text{Treetuples}(V)}$$



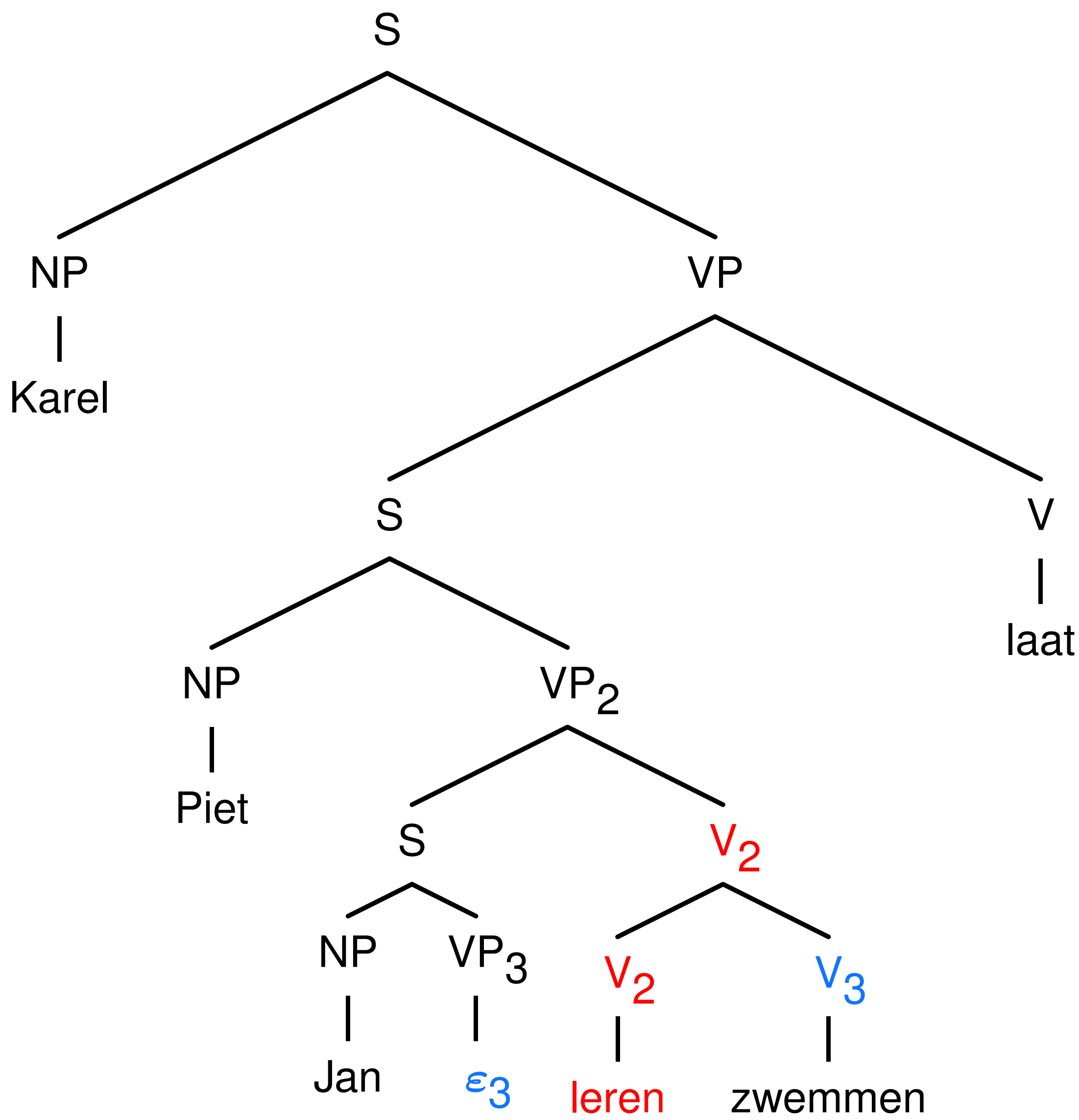
Basic structure: ‘dat Karel Piet Jan laat leren zwemmen’



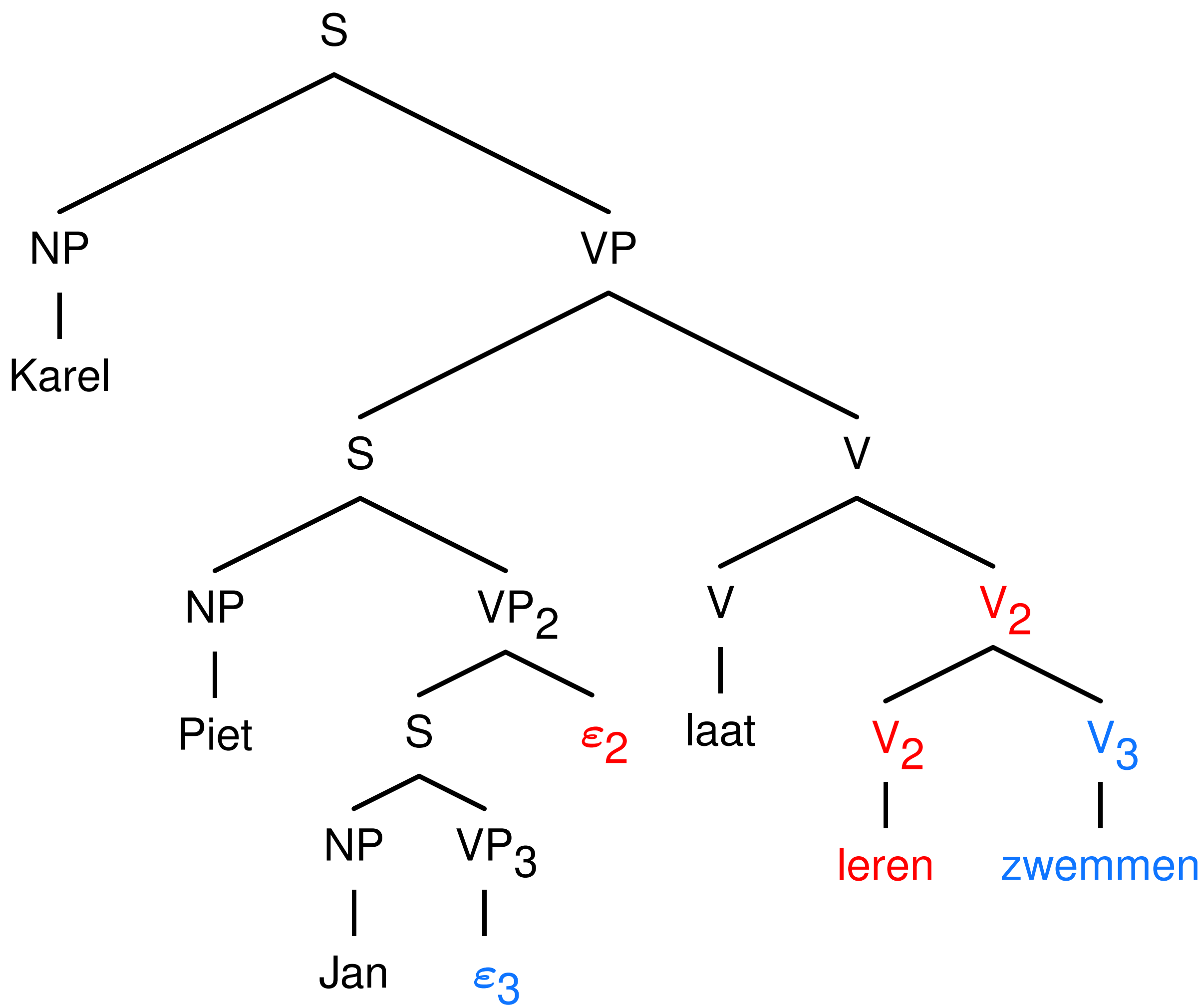
Basic structure: ‘dat Karel Piet Jan laat leren zwemmen’



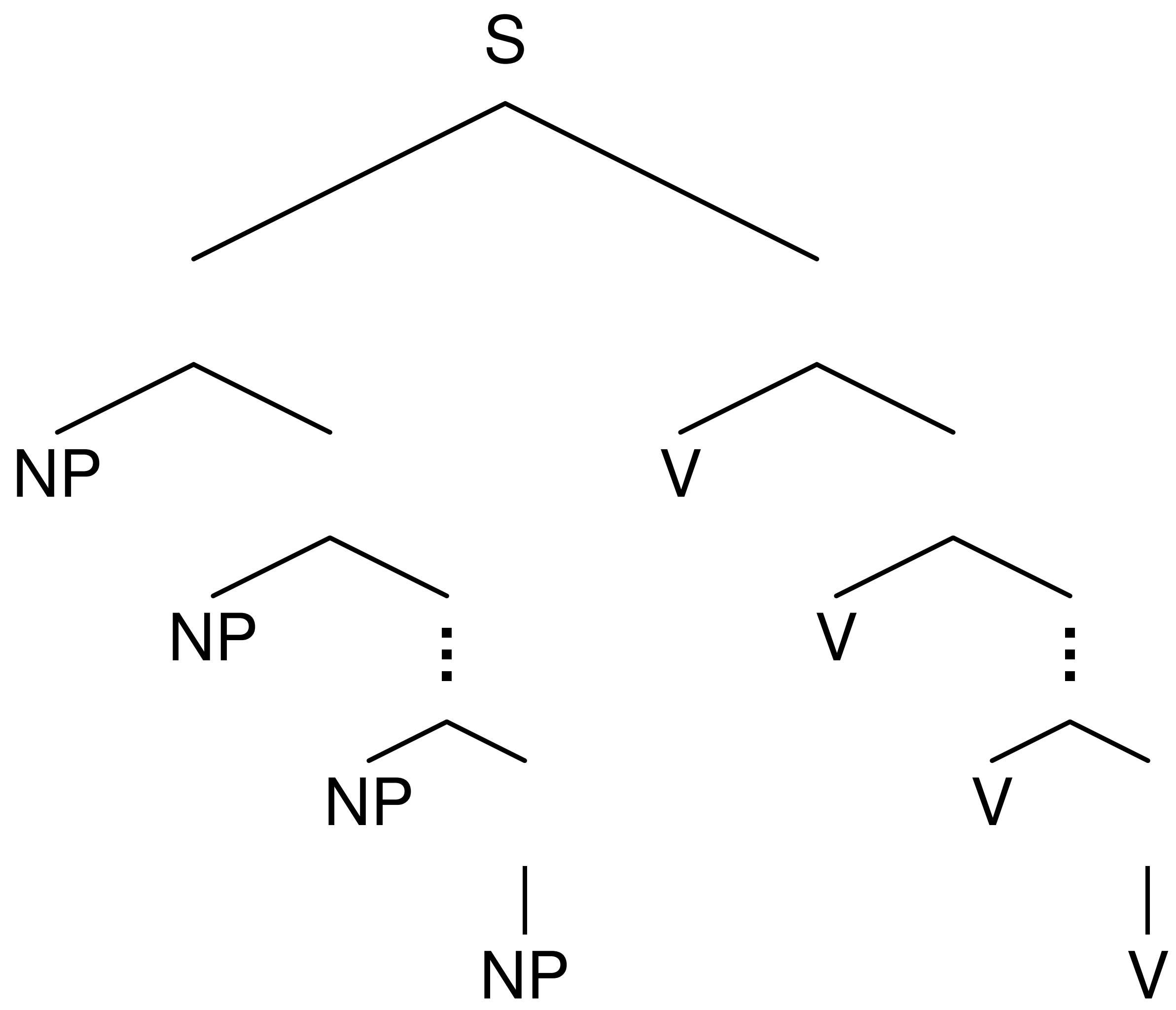
Succesive cyclic head movement: ‘dat Karel Piet Jan laat leren zwemmen



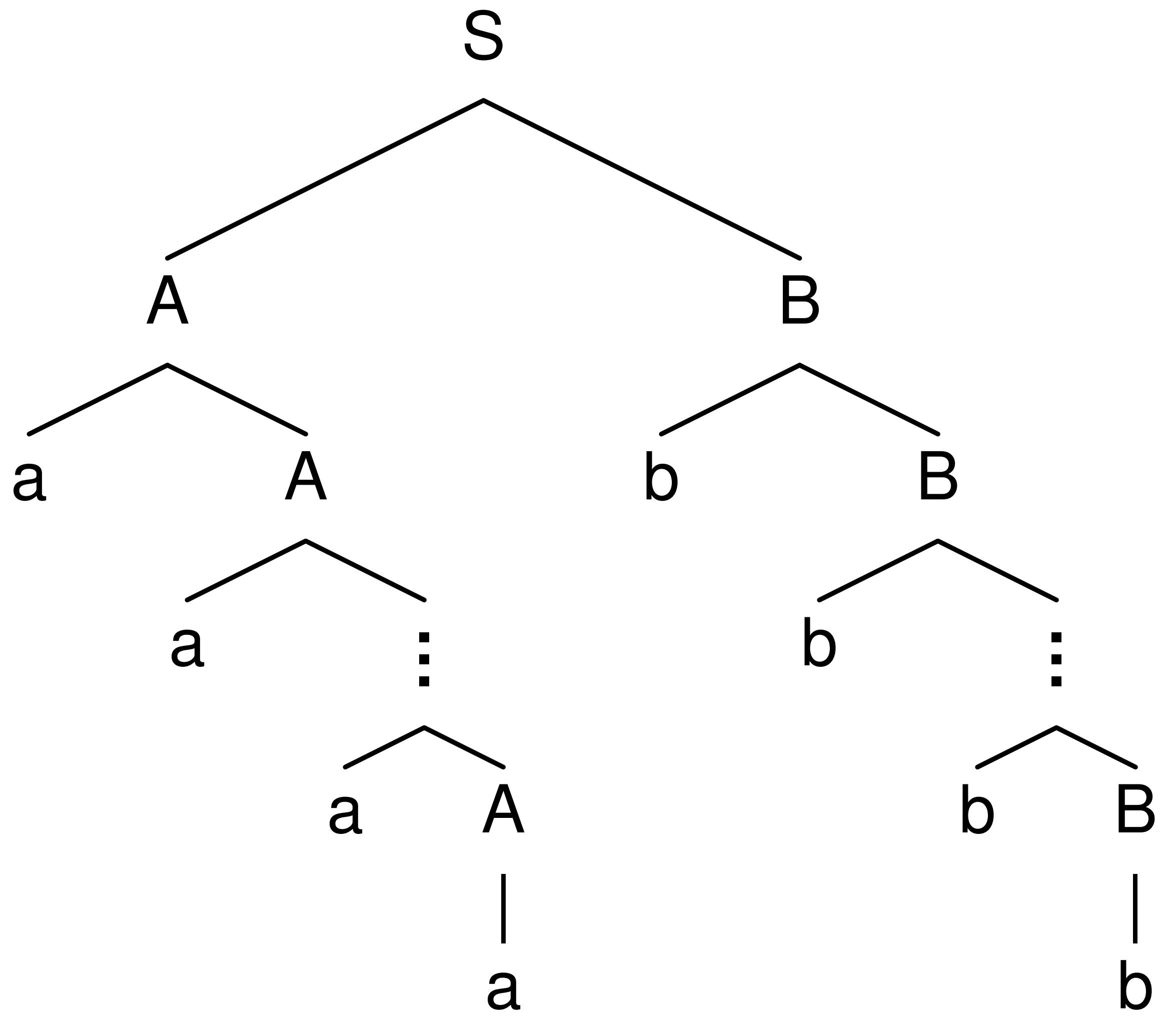
“Real” verb cluster analysis: ‘dat Karel Piet Jan laat leren zwemmen’

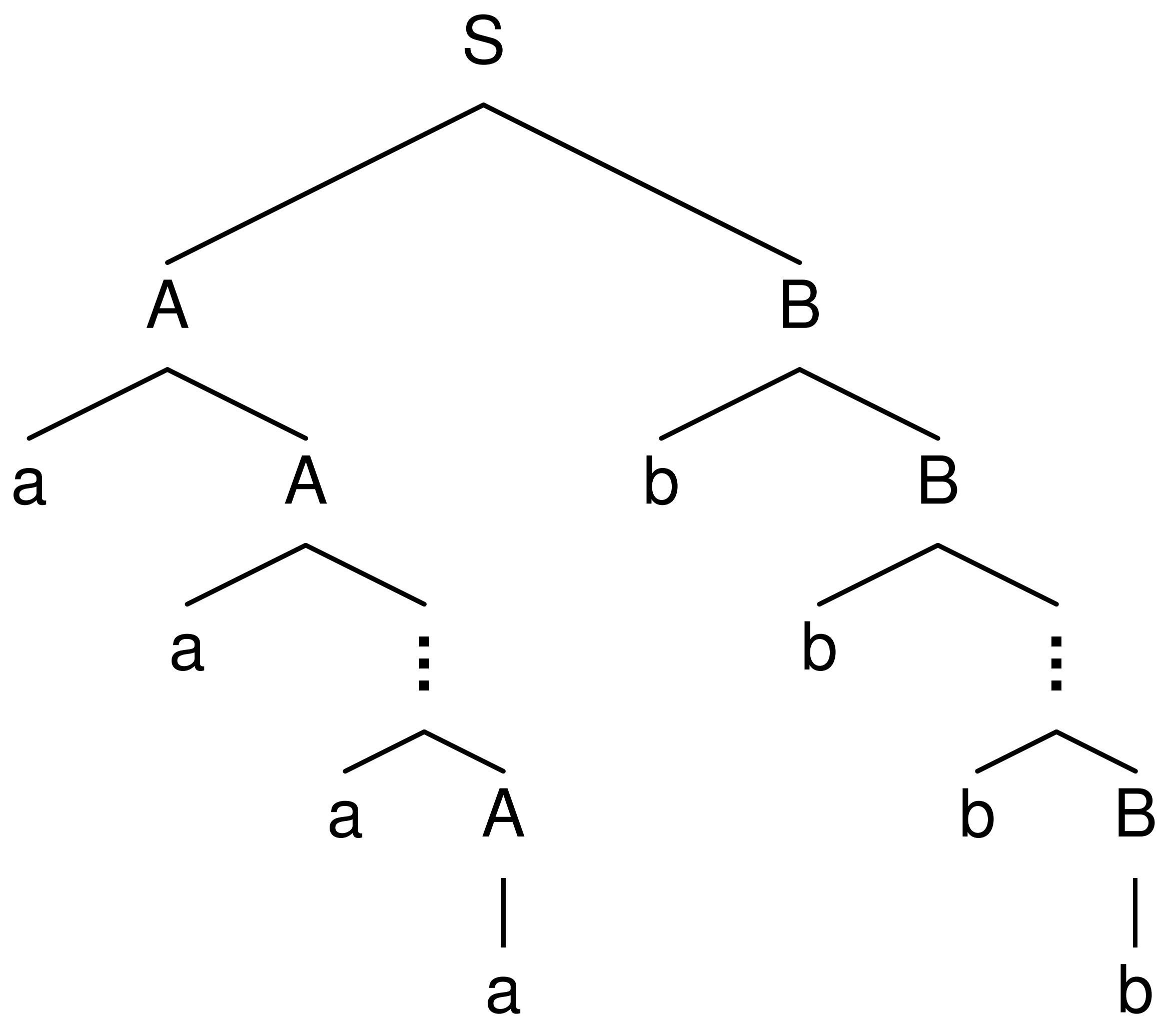


“Real” verb cluster analysis: essentially reduces to . . .



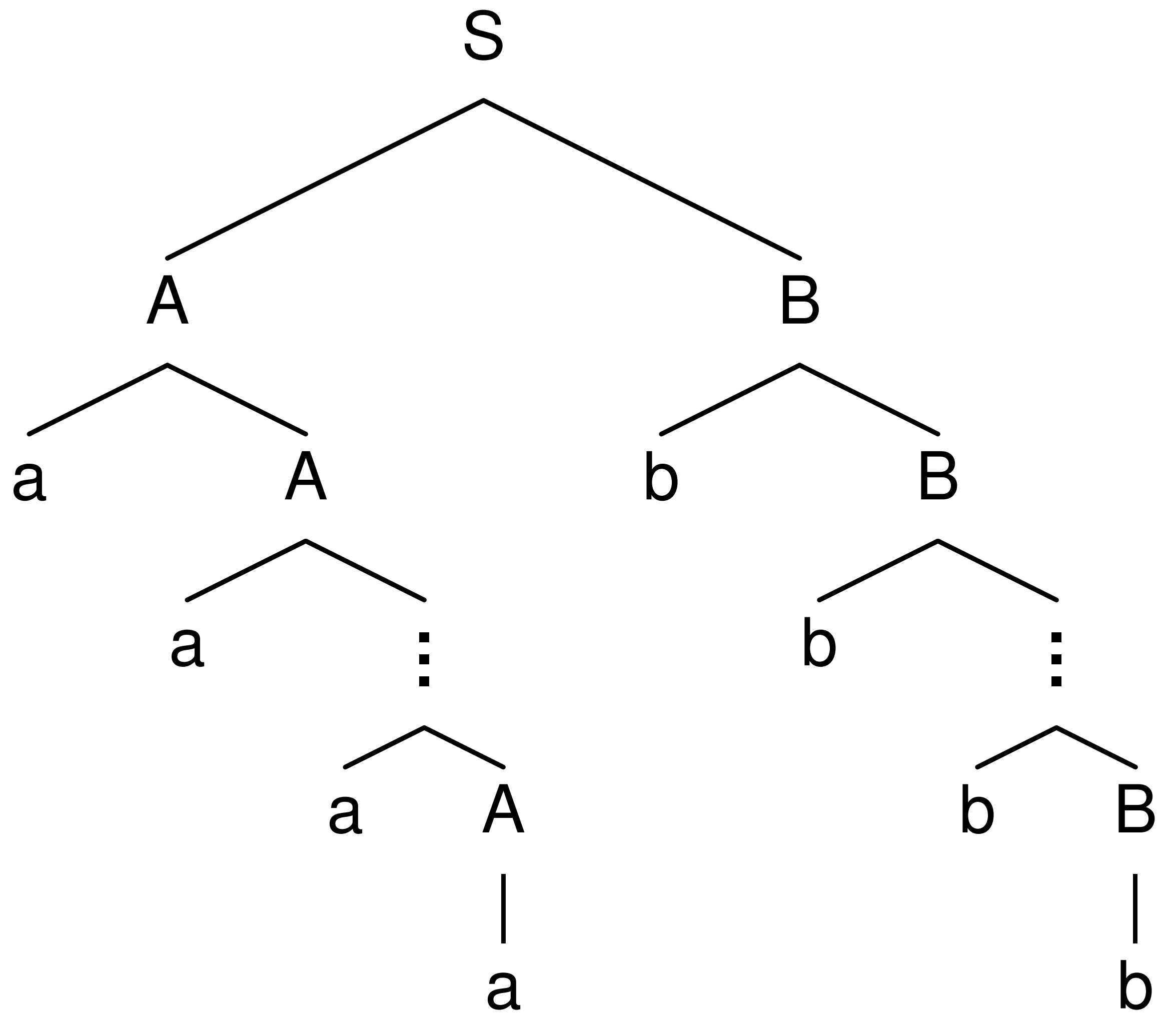
“Real” verb cluster analysis: essentially reduces to . . .





. . . a non-recognizable tree set.

“Real” verb cluster analysis: essentially reduces to . . .



. . . a non-recognizable tree set. (See also Bresnan et al. 1982)

References

Joan W. Bresnan, Ronald M. Kaplan, Stanley Peters and Annie Zaenen. 1982. Cross–serial dependencies in Dutch. *Linguistic Inquiry*, 13:613–635.

Gerald Gazdar. 1988. Applicability of indexed grammars to natural languages. In: U. Reyle and C. Rohrer (eds.), *Natural Language Parsing and Linguistic Theories*, pp. 69–94. D. Reidel, Dordrecht.

Aravind K. Joshi. 1985. Tree adjoining grammars: How much context–sensitivity is required to provide reasonable structural descriptions? In: D. R. Dowty, L. Karttunen and A. M. Zwicky (eds.), *Natural Language Parsing. Psychological, Computational, and Theoretical Perspectives*, pp. 206–250. Cambridge University Press, New York, NY.

Aravind K. Joshi. 1987. An introduction to tree adjoining grammars. In: A. Manaster–Ramer (ed.), *Mathematics of Language*, pp. 87–114. John Benjamins, Amsterdam.

Aravind K. Joshi, Leon S. Levy and Masako Takahashi. 1975. Tree adjunct grammars. *Journal of Computer and System Sciences*, 10:136–163.

Aravind K. Joshi, K. Vijay–Shanker and David J. Weir. 1991. The convergence of mildly context–sensitive grammar formalisms. In: P. Sells, S. M. Shieber and T. Wasow (eds.), *Foundational Issues in Natural Language Processing*, pp. 31–81. MIT Press, Cambridge, MA.

Anthony S. Kroch and Beatrice Santorini. 1991. The derived constituent structure of the West Germanic verb–raising construction. In: R. Freidin (ed.), *Principles and Parameters in Comparative Grammar*, pp. 269–338. MIT Press, Cambridge, MA.

Carl J. Pollard. 1984. *Generalized Phrase Structure Grammars, Head Grammars, and Natural Language*. Dissertation, Stanford University, Stanford, CA.

Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii and Tadao Kasami. 1991. On multiple context–free grammars. *Theoretical Computer Science*, 88:191–229.

Mark Steedman. 1987. Combinatory grammars and parasitic gaps. *Natural Language and Linguistic Theory*, 5:403–439.

Mark Steedman. 1990. Gapping as constituent coordination. *Linguistics and Philosophy*, 13:207–263.

K. Vijay–Shanker and Aravind K. Joshi. 1985. Some computational properties of tree adjoining grammars. In: *23th Annual Meeting of the Association for Computational Linguistics (ACL '85)*, Chicago, IL, pp. 82–93. ACL.

K. Vijay–Shanker and David J. Weir. 1994. The equivalence of four extensions of context–free grammars. *Mathematical Systems Theory*, 27:511–546.

K. Vijay–Shanker, David J. Weir and Aravind K. Joshi. 1986. Tree adjoining and head wrapping. In: *Proceedings of the 11th International Conference on Computational Linguistics (COLING '86)*, Bonn, pp. 202–207.

K. Vijay–Shanker, David J. Weir and Aravind K. Joshi. 1987. Characterizing structural descriptions produced by various grammatical formalisms. In: *25th Annual Meeting of the Association for Computational Linguistics (ACL '87)*, Stanford, CA, pp. 104–111. ACL.

David J. Weir. 1988. *Characterizing Mildly Context–Sensitive Grammar Formalisms*. Dissertation, University of Pennsylvania, Philadelphia, PA.

David J. Weir and Aravind K. Joshi. 1988. Combinatory categorial grammars: Generative power and relationship to linear context–free rewriting systems. In: *26th Annual Meeting of the Association for Computational Linguistics (ACL '88)*, Buffalo, NY, pp. 278–285. ACL.