Equivalence to Minimalist Grammars: All boils down to overt phrasal movement

Jens Michaelis

Bielefeld University, Bielefeld, Germany

MCFG+2, Nara, Japan, September 9-10, 2011

Outline

- Minimalist grammars
  - The formalism
  - Trees in terms of a reduced tuple representation
  - Redefining the formalism (almost as MCFGs)
- Multiple context-free grammars
  - Normal forms
  - Representing normal forms as minimalist grammars
- Concluding remarks

Minimalist grammars

- Minimalist grammars (MGs) (Stabler 1997, 1999) provide an attempt at a rigorous algebraic formalization (of some) of the perspectives adopted in the minimalist branch of generative grammar.

- MGs in the above format constitute a mildly context-sensitive grammar formalism in the sense of Joshi 1985 (Michaelis 2001a,b)

  Two crucial features of MGs helped achieving this result:

Minimalist grammars

- Minimalist grammars (MGs) (Stabler 1997, 1999) provide an attempt at a rigorous algebraic formalization (of some) of the perspectives adopted in the minimalist branch of generative grammar.

- MGs in the above format constitute a mildly context-sensitive grammar formalism in the sense of Joshi 1985 (Michaelis 2001a,b)

  Two crucial features of MGs helped achieving this result:

  - the resource sensitivity (encoded in the checking mechanism)
  - the shortest move condition (as a locality constraint)

Minimalist grammars

- Work on MGs defined in this sense can, thus, be seen as having led to a realignment of "grammars found 'useful' by linguists" and formal complexity theory.

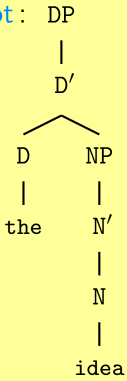
- In fact, MGs are capable of integrating (if needed) a variety of (arguably) "odd" items from the syntactician's toolbox, e.g.,

  - head movement (Stabler 1997, 2001)
  - (strict) remnant movement (Stabler 1997, 1999)
  - affix hopping (Stabler 2001)
  - adjunction and scrambling (Frey & Gärtner 2002)
  - late adjunction and extraposition (Gärtner & Michaelis 2008)
  - copy-movement (Kobele 2006)
  - relativized minimality (Stabler 2011)

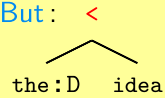
Minimalist expressions (labeled trees)

- The objects generated by an MG: minimalist expressions

Not:

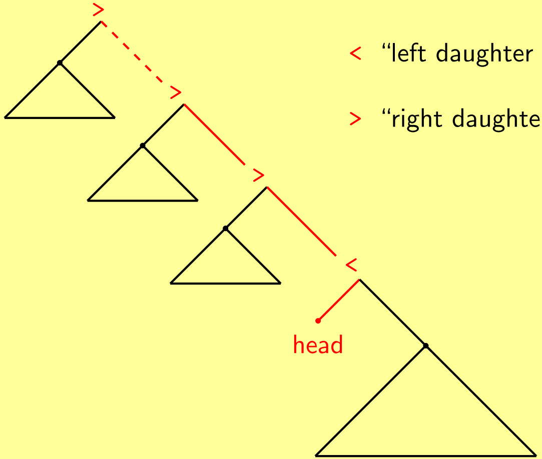


But:



< "points towards" the projecting daughter.

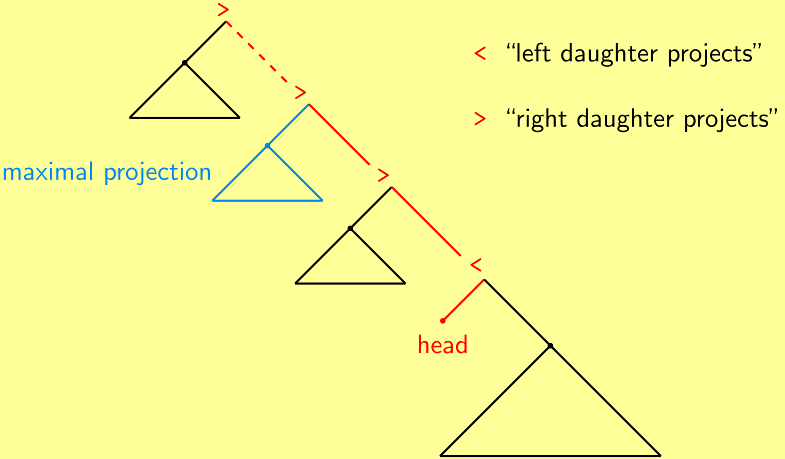
Minimalist expressions (labeled trees)



< "left daughter projects"

> "right daughter projects"

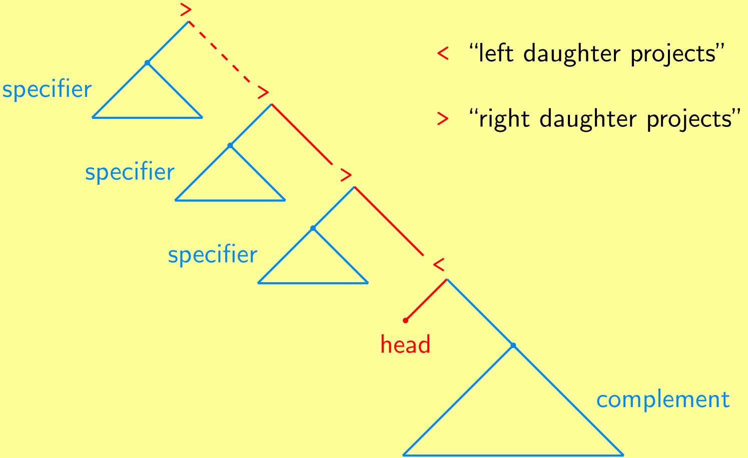
- non-leaf-labels [projection]



- non-leaf-labels [projection]

— maximal projections: each subtree whose root does not project —

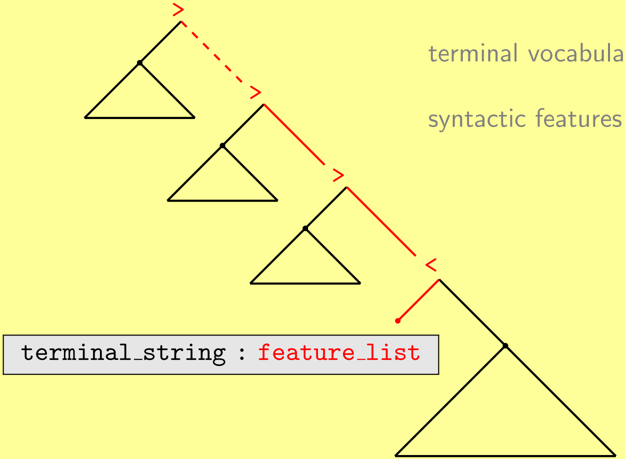
Minimalist expressions (labeled trees)



- non-leaf-labels [projection]

  — maximal projections: each subtree whose root does not project —

Minimalist expressions (labeled trees)



terminal vocabulary

syntactic features

- leaf-labels

Minimalist expressions (syntactic features)

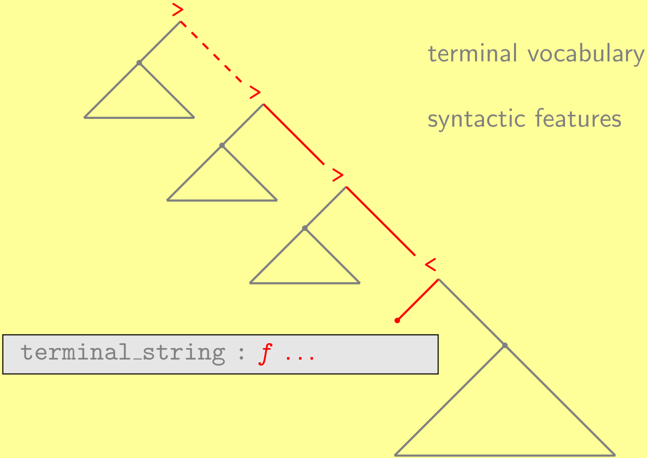
- There are different types of syntactic features.

selectees:	`x`
selectors:	`=x`
licensees:	`-x`
licensors:	`+x`
...	

- Starting from a lexicon, a finite set of simple expressions, minimalist expressions can be built up recursively by checking off instances of syntactic features "from left to right."

- Different types of syntactic features trigger different structure building functions.

Minimalist expressions

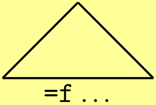


head-label is of the form `terminal_string : f ...`

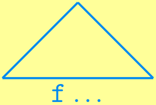
Structure building functions

$$\text{merge} : \text{Trees} \times \text{Trees} \xrightarrow{\text{part}} \text{Trees}$$

tree displays feature `=f`



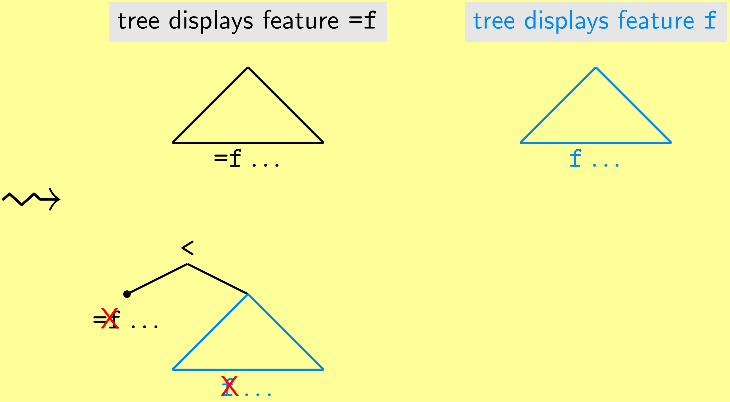
tree displays feature `f`



$\rightsquigarrow$

Structure building functions

$$\text{merge} : \text{Trees} \times \text{Trees} \xrightarrow{\text{part}} \text{Trees}$$



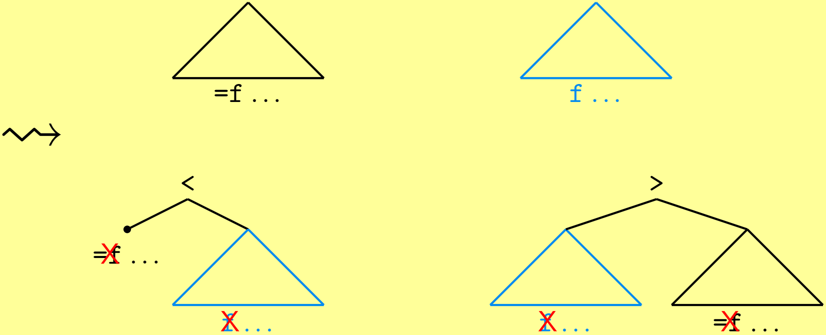
selecting tree a simple head

Structure building functions

$$\text{merge} : \text{Trees} \times \text{Trees} \xrightarrow{\text{part}} \text{Trees}$$

tree displays feature =f

tree displays feature f

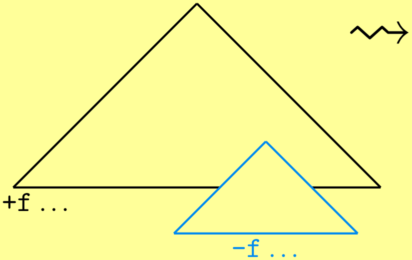


selecting tree a simple head

selecting tree complex

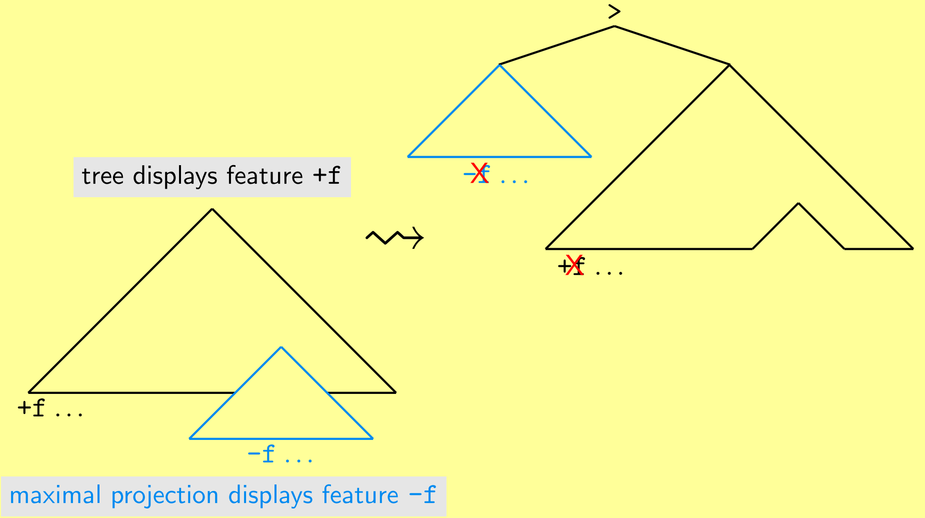
$$\text{move} : \text{Trees} \xrightarrow{\text{part}} 2^{\text{Trees}}$$

tree displays feature `+f`



maximal projection displays feature `−f`

$\text{move} : \text{Trees} \xrightarrow{\text{part}} 2^{\text{Trees}}$



Structure building functions

- A simple example of an embedded interrogative clause shows the different types of features at work in order to serve as a demonstration of the general cases.

  - merge :
    - right selection
    - left selection
  - move:
    - overt phrasal movement

Example of a lexicon

(a set of single noded labeled trees)

that :: =I C	*the* :: =N D -k
[] :: =I +wh C	*which* :: =N D -k -wh
does :: =v +k I	
sleep :: =D +k v	
[] :: =V =D v	*cat* :: N
bite :: =D +k V	*dog* :: N

merge

=f ⇝ right selection

which :: =N D -k -wh

merge

=f ⇝ right selection

which :: =N D -k -wh *dog* :: N

merge

`=f` ⇝ right selection

<

which : `=N` `D` `-k` `-wh` ^ *dog* : `N`

merge

=f ⇝ right selection

<

which : =N D -k -wh

dog : N

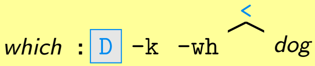
merge

$\texttt{f} \rightsquigarrow$ tree, selectable via merge

$<$

which : $\Rightarrow\texttt{N}$ $\boxed{\texttt{D}}$ $\texttt{-k}$ $\texttt{-wh}$ *dog* : $\texttt{N}$

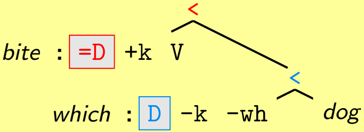
f $\rightsquigarrow$ tree, selectable via merge



bite :: =D +k V

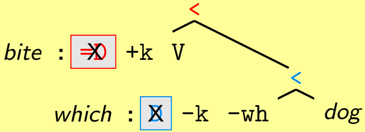
which : D -k -wh < *dog*

=f ⇝ right selection



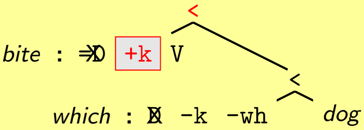
merge

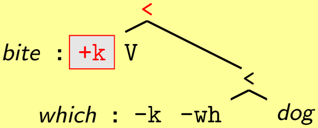
=f $\rightsquigarrow$ right selection

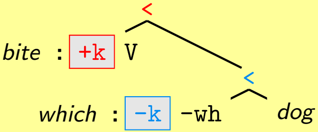


merge

+f $\rightsquigarrow$ tree, demanding movement

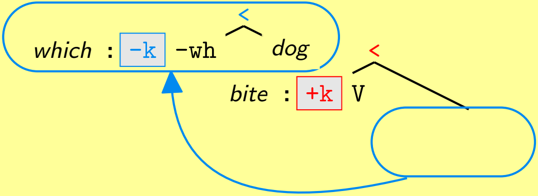



<
bite : +k V
<
which : -k -wh
dog


<
bite : +k V
<
which : -k -wh
dog

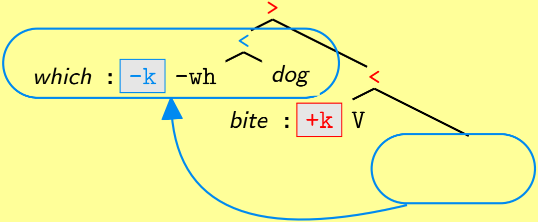
move

+f ⇝ overt phrasal movement



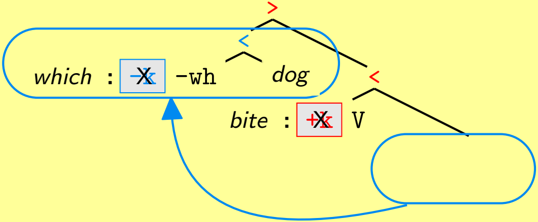
move

+f ⇝ overt phrasal movement

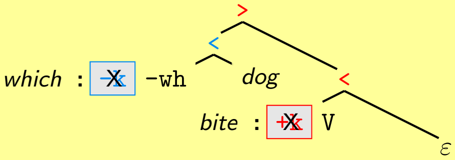


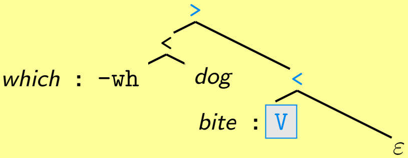
move

+f ⇝ overt phrasal movement



+f ⇝ overt phrasal movement

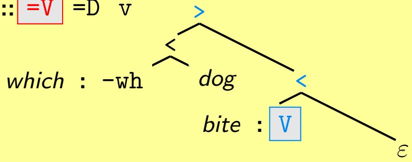


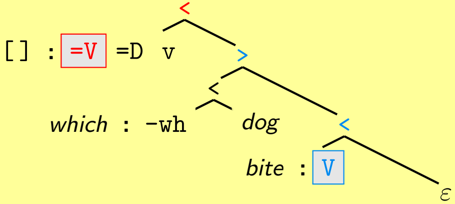


>
<
which : -wh
dog
<
bite : V
ε

merge

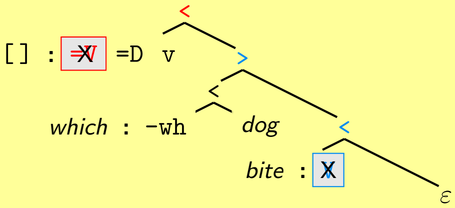
[] :: =V =D v

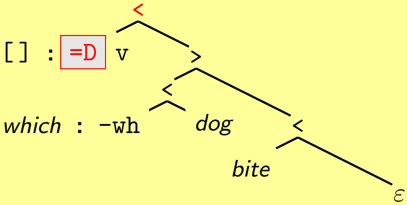



<
[] : =V =D v
>
<
which : -wh
dog
<
bite : V
ε

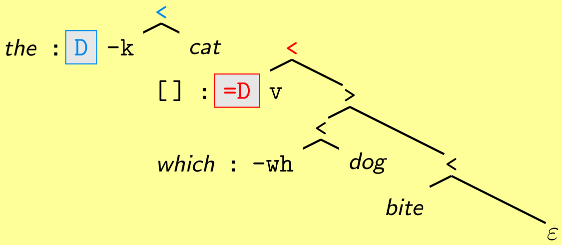
merge

=f ⇝ right selection



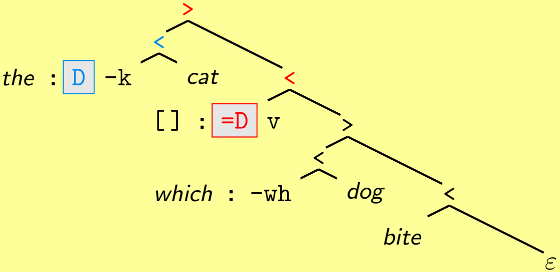


<
[] : =D v
>
<
which : -wh
dog
<
bite
ε


<
the : D -k
cat
<
[] : =D v
>
<
which : -wh
dog
<
bite
ε

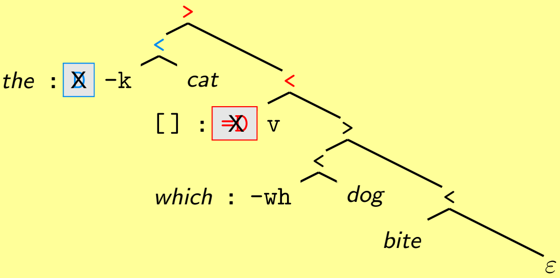
merge

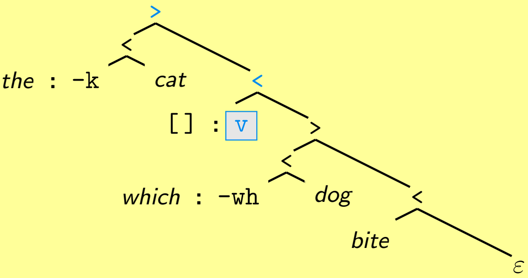
=f $\rightsquigarrow$ left selection

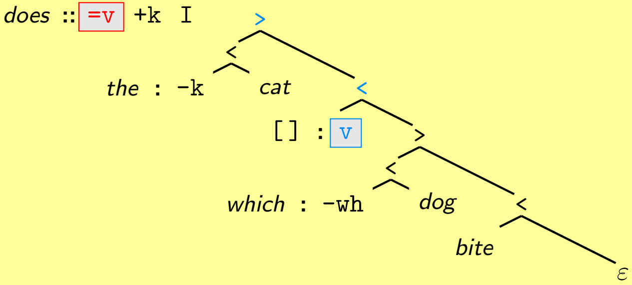


merge

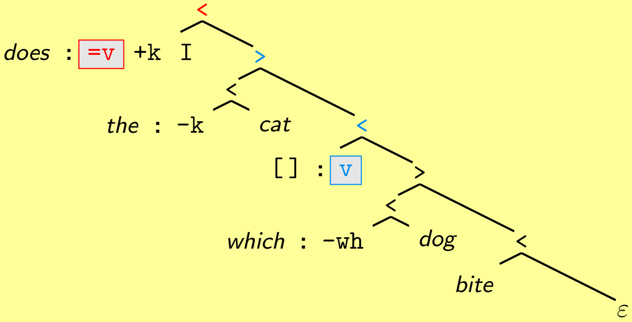
=f ⇝ left selection

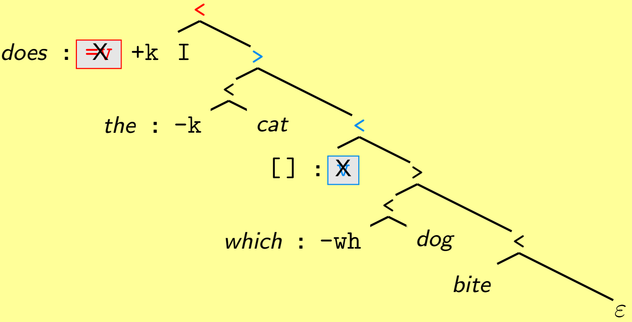


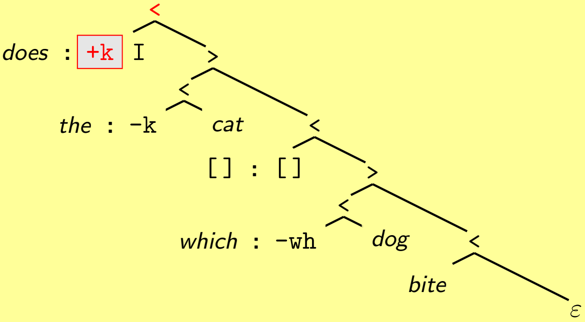

>
<
the : -k
cat
<
[] : v
>
<
which : -wh
dog
<
bite
ε


does :: =v +k I
>
<
the : -k
cat
<
[] : v
>
<
which : -wh
dog
<
bite
ε

=f ⇝ right selection

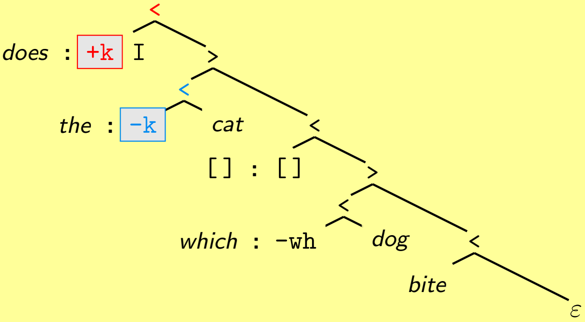


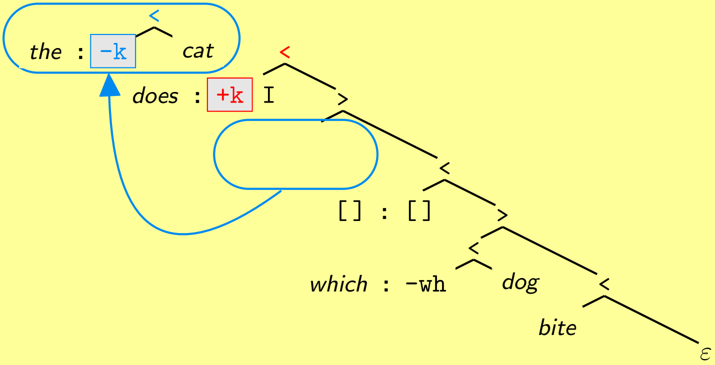

<
does : =v +k I
>
<
the : -k
cat
<
[] : v
>
<
which : -wh
dog
<
bite
ε


<
does : +k I
>
<
the : -k
cat
<
[] : []
>
<
which : -wh
dog
<
bite
ε

move

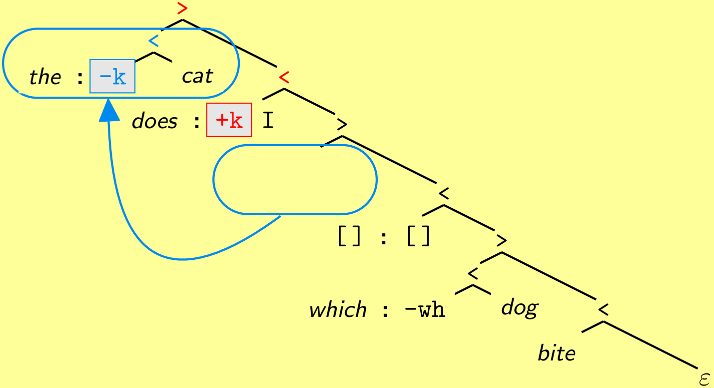
+f ⇝ overt phrasal movement

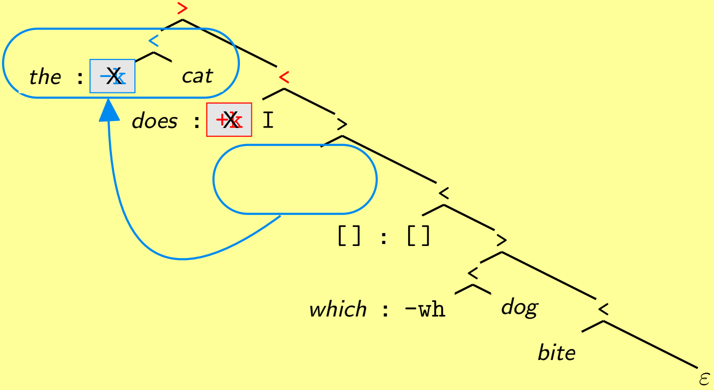


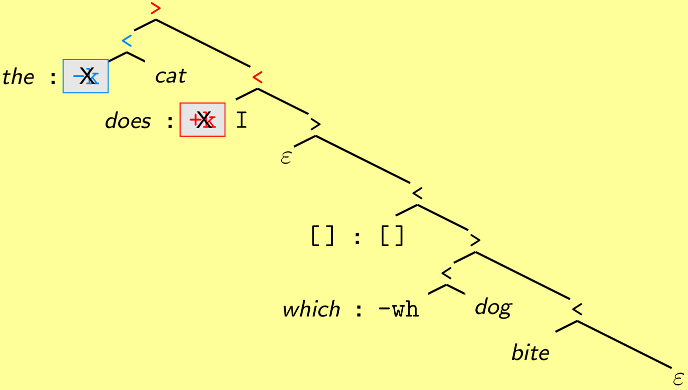

<
the : -k
cat
<
does : +k I
>
<
[] : []
>
<
which : -wh
dog
<
bite
ε

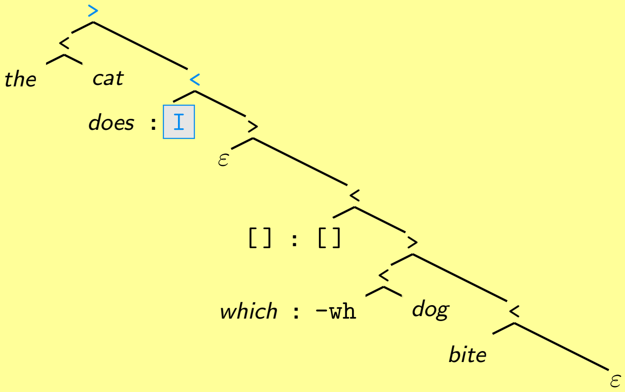
move

+f ⇝ overt phrasal movement




the : -k
cat
does : +k I
[] : []
which : -wh
dog
bite
ε

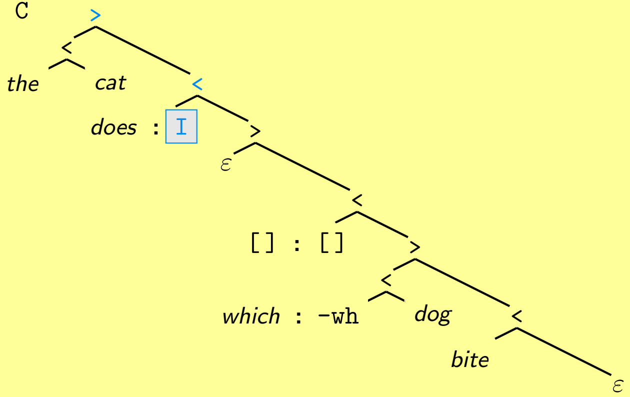

>
<
the : -k
cat
<
does : +k I
>
ε
<
[] : []
>
<
which : -wh
dog
<
bite
ε


>
<
the
cat
<
does : I
>
ε
<
[] : []
>
<
which : -wh
dog
<
bite
ε

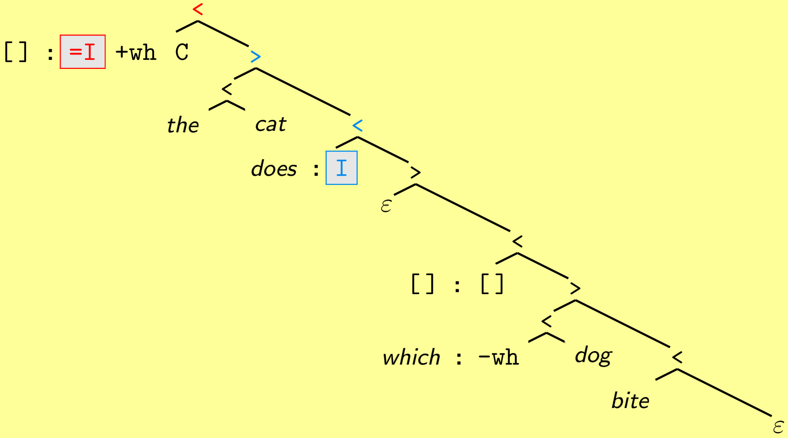
merge

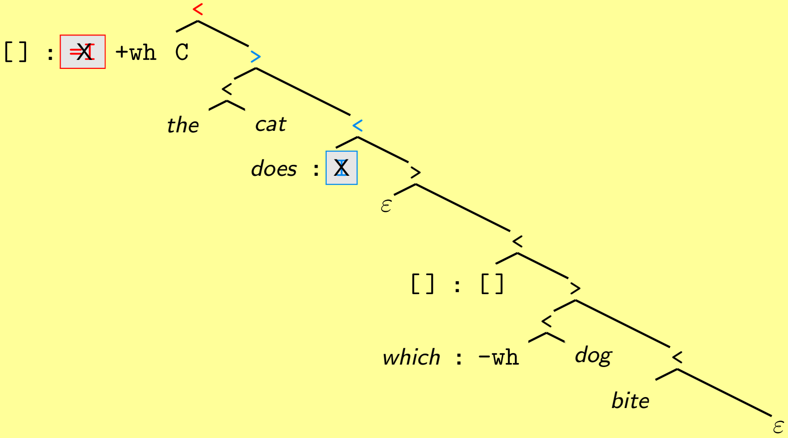
=f ⇝ right selection

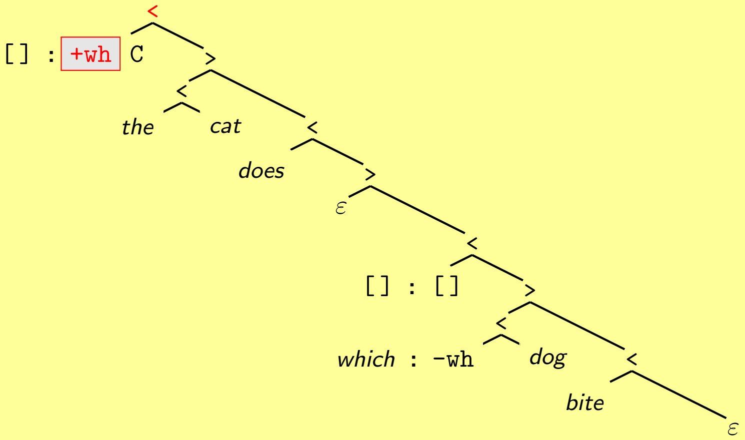
[] :: =I +wh C



merge

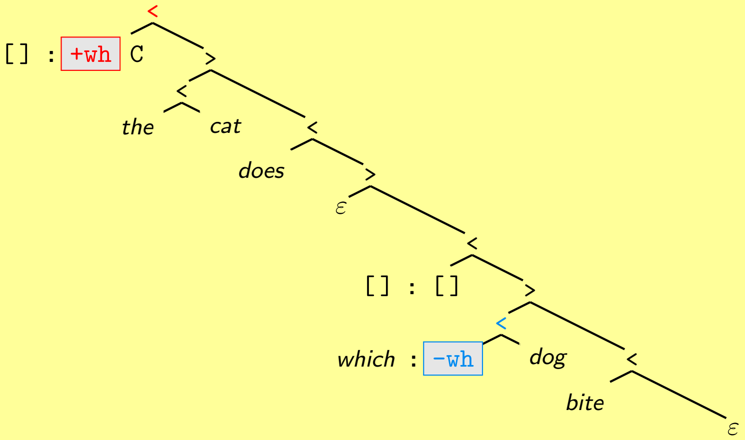


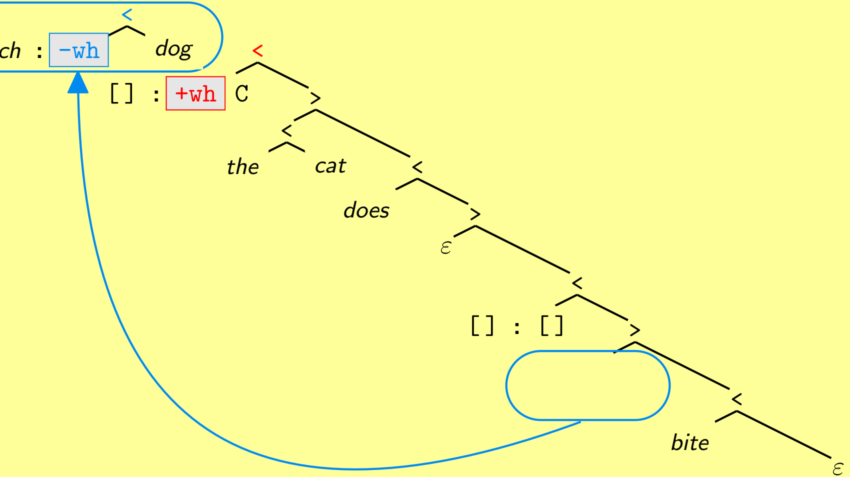

[] : X +wh C
the
cat
does : X
[] : []
which : -wh
dog
bite


[] : +wh C
the
cat
does
ε
[] : []
which : -wh
dog
bite
ε

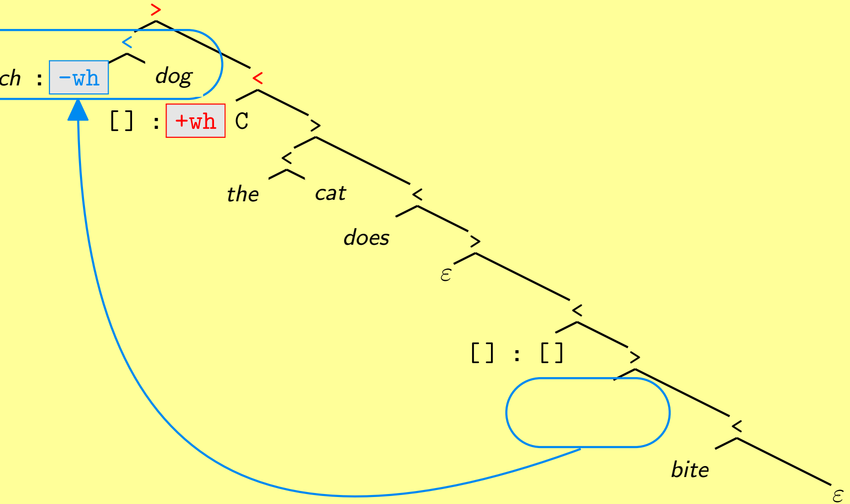
move

+f ⇝ overt phrasal movement



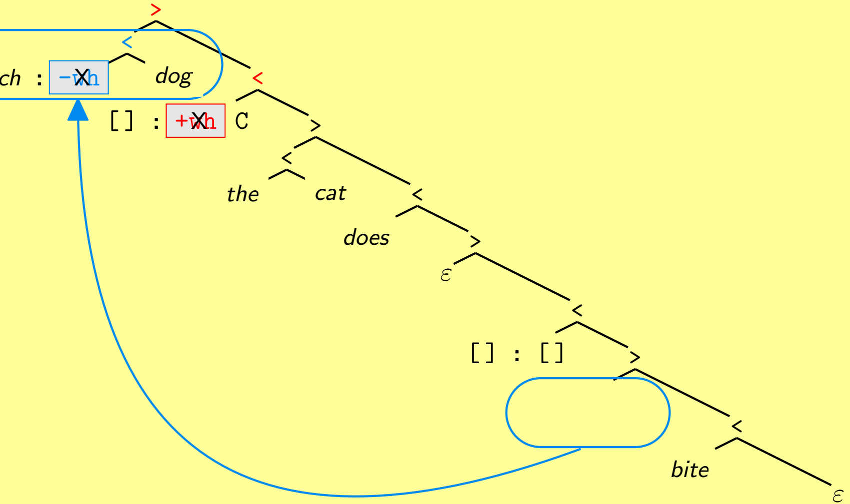

<
ch : -wh
dog
<
[] : +wh C
>
<
the
cat
<
does
>
ε
<
[] : []
>
<
bite
ε

+f ⇝ overt phrasal movement



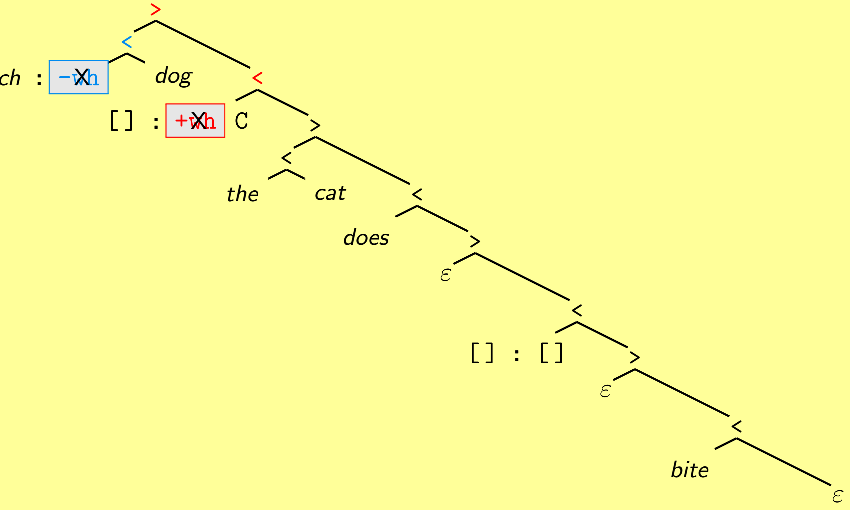
move

+f ⇝ overt phrasal movement



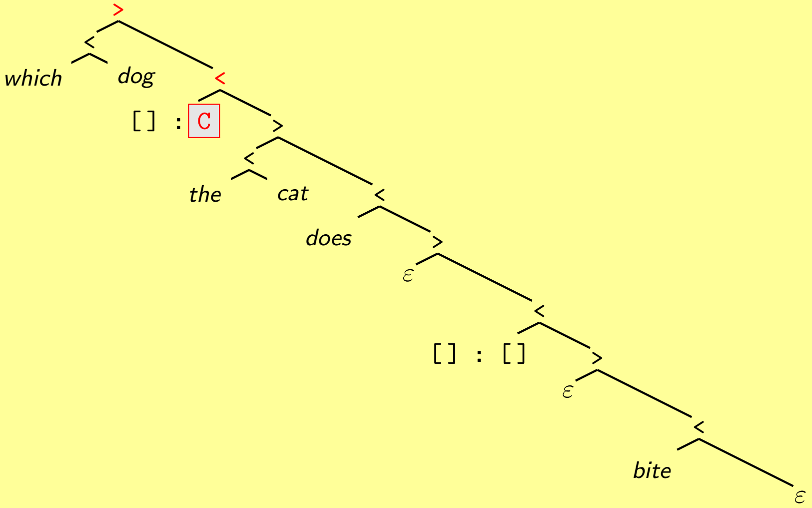
move

+f ⇝ overt phrasal movement

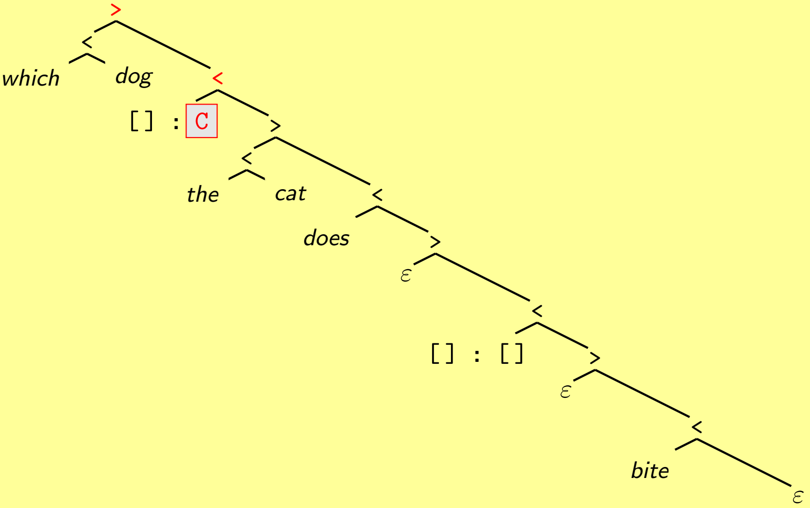


Complete tree

"accepted as category C"



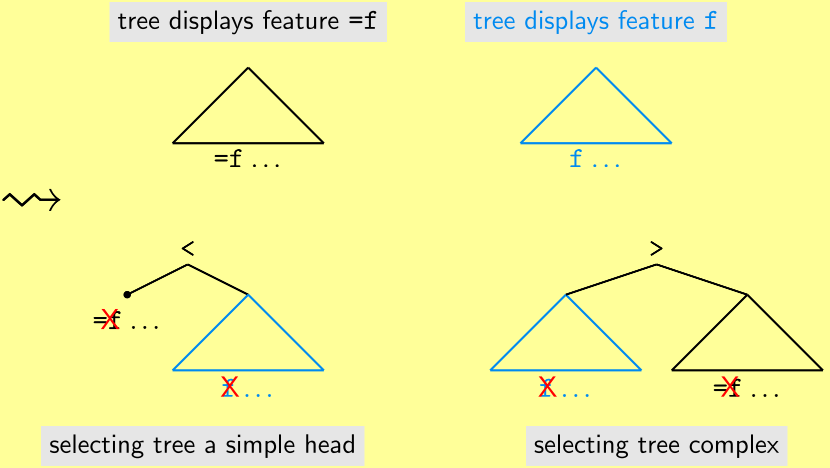
"accepted as category C"



- No unchecked syntactic features but one instance of C within the head-label.

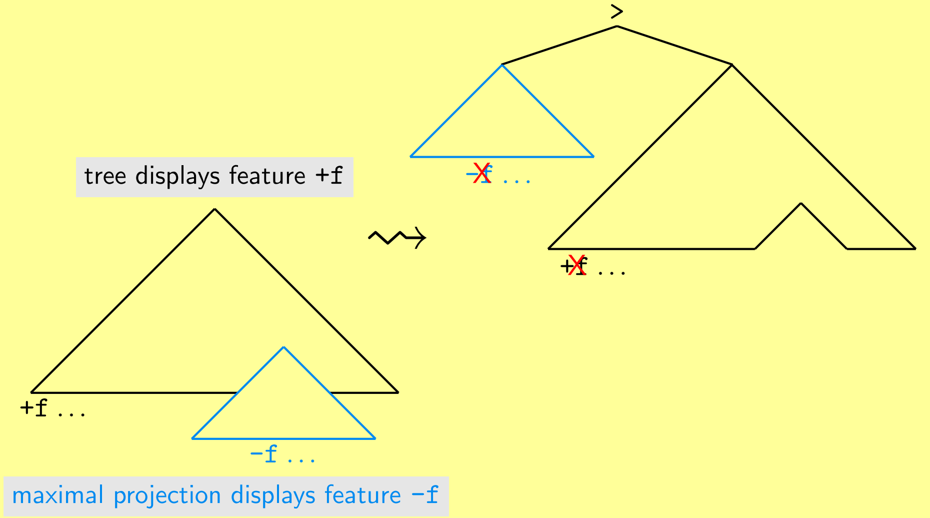
Structure building functions

$$\text{merge} : \text{Trees} \times \text{Trees} \xrightarrow{\text{part}} \text{Trees}$$



Structure building functions (overt phrasal movement)

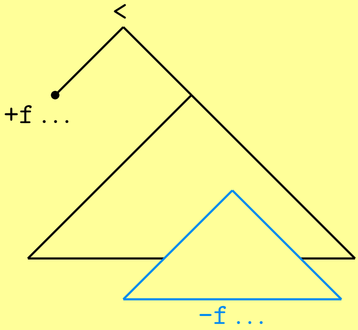
move : Trees $\xrightarrow{\text{part}}$ 2^{Trees}



Shortest movement condition (SMC) (Stabler 1997, 1999)

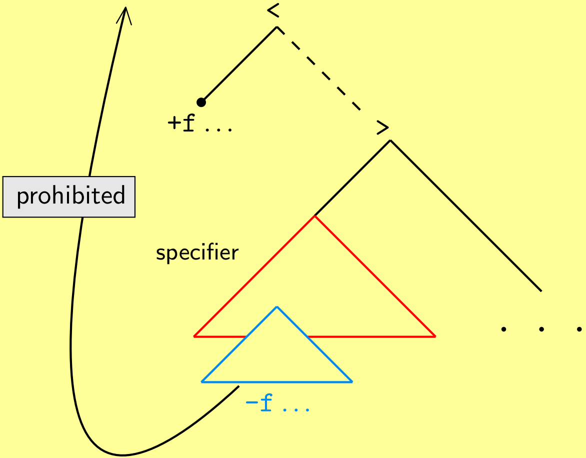
- The number of competing licensee features triggering a movement is (finitely) bounded by some number n.

  In the strictest version $n = 1$: move only applies, if there is at most one maximal projection displaying a matching licensee feature.



Specifier island condition (SPIC) (Stabler 1999)

- Proper "extraction" from specifiers is blocked.



Minimalist grammars

$G = \langle$ Vocabulary , SynFeat , Lex , Ω , c $\rangle$ an MG

- Vocabulary — a finite set — [terminal vocabulary]
- SynFeat — a finite set — [(syntactic) features]

Selectees	$\cup$	Selectors	$\cup$	Licensees	$\cup$	Licensors
x		=x		-x		+x

Minimalist grammars

$G = \langle \text{Vocabulary}, \text{SynFeat}, \text{Lex}, \Omega, \text{c} \rangle$ an MG

- Vocabulary — a finite set — [terminal vocabulary]
- SynFeat — a finite set — [(syntactic) features]

  Selectees ∪ Selectors ∪ Licensees ∪ Licensors

  x =x -x +x

- $\text{Lex} \subseteq \text{Vocabulary}^* \times \{::\} \times \text{SynFeat}^*$ [lexicon]

  — a finite set of single noded minimalist expressions —

Minimalist grammars

$G = \langle \text{Vocabulary}, \text{SynFeat}, \text{Lex}, \Omega, c \rangle$ an MG

- Vocabulary — a finite set — [terminal vocabulary]
- SynFeat — a finite set — [(syntactic) features]

  Selectees ∪ Selectors ∪ Licensees ∪ Licensors
  x =x -x +x

- $\text{Lex} \subseteq \text{Vocabulary}^* \times \{::\} \times \text{SynFeat}^*$ [lexicon]

  — a finite set of single noded minimalist expressions —

- $\Omega = \{ \text{merge}, \text{move} \}$ [structure building functions]
- $c \in \text{Selectees}$ [distinguished category]

The closure of G [Closure(G)] :$\Longleftrightarrow$

closure of the lexicon under finite applications of the functions in Ω.

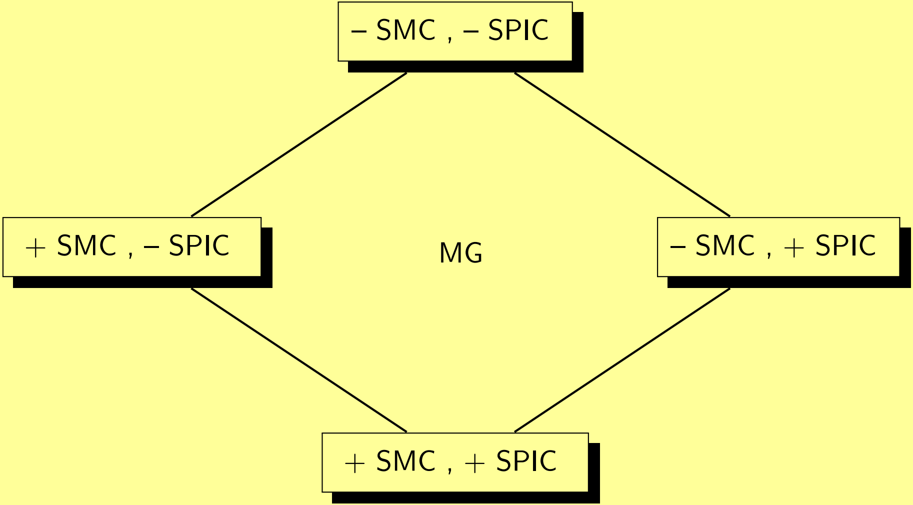
The tree language of G [Trees(G)] :$\Longleftrightarrow$

trees in the closure with essentially no unchecked syntactic features
— only head-label contains exactly one unchecked instance of c.

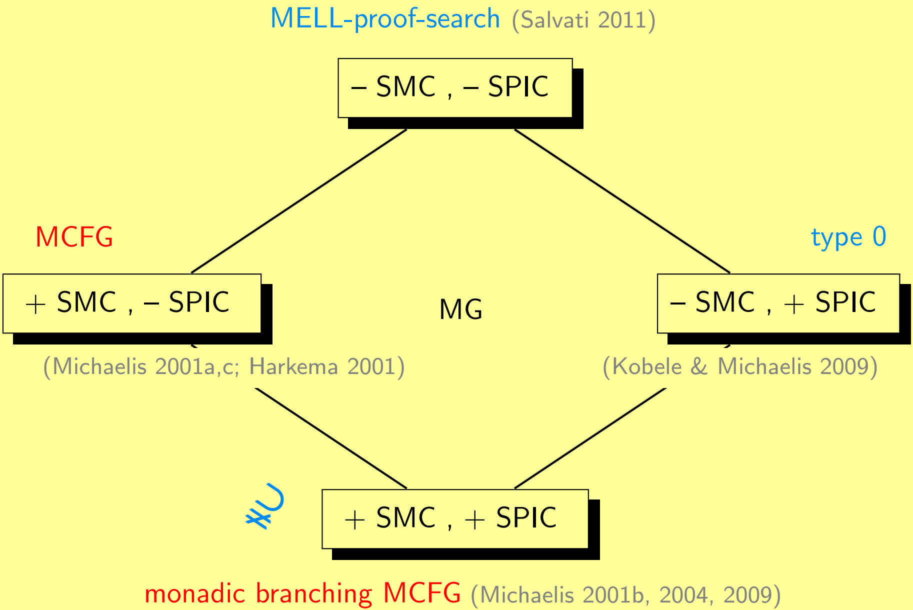
The string language of G [L(G)] :$\Longleftrightarrow$

(terminal) yields of the trees belonging to the tree language.

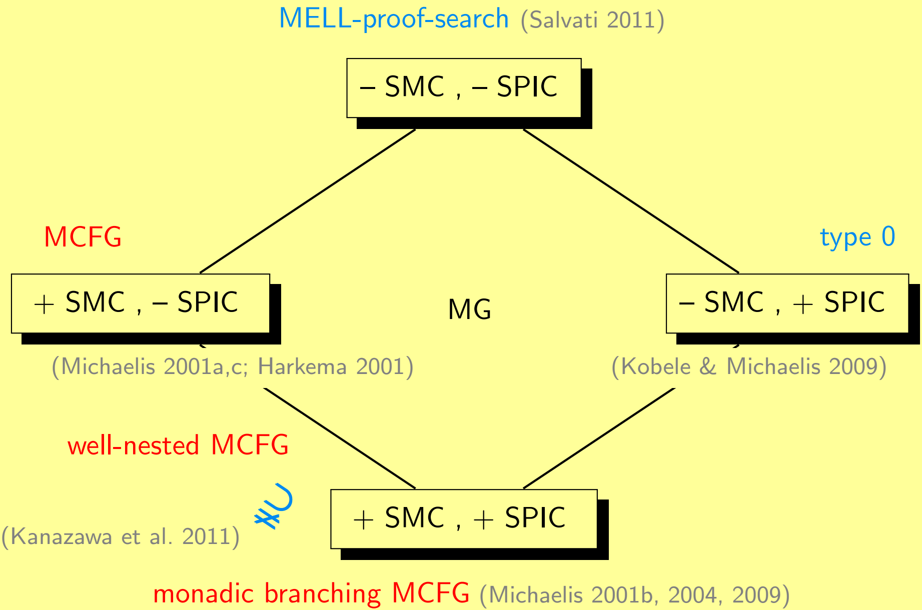
SMC and SPIC — restricting the move-operator domain



SMC and SPIC — restricting the move-operator domain



SMC and SPIC — restricting the move-operator domain



- The method

  essentially developed to prove that the MG-string languages provide a subclass of MCFLs in Michaelis 2001a, and

  leading to the succinct, chain-based MG-reformulation presented in Stabler & Keenan 2003, reducing "classical" MGs to their "bare essentials:"

- Defining a finite partition on the "relevant" MG-tree set,
  - giving rise to a finite set of nonterminals in MCFG-terms,
  - nevertheless deriving all possible "terminal yields."

Trees as tuples (in summary)

- General idea:

  compact tuple representation reducing minimalist trees to exactly the information which is relevant within a (proceeding) derivation.

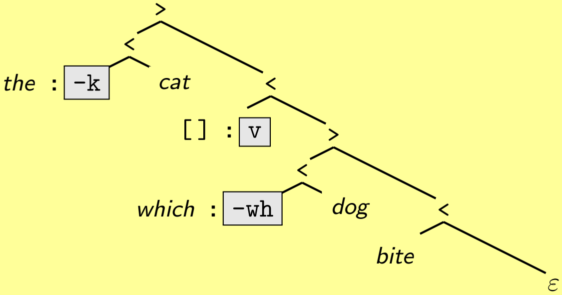
- Put differently:

  every part of a maximal projection not related to some unchecked feature, i.e. every part of a constituent being an "unextractable" part of a higher constituent the latter providing some unchecked feature, is compactly represented with this higher constituent.

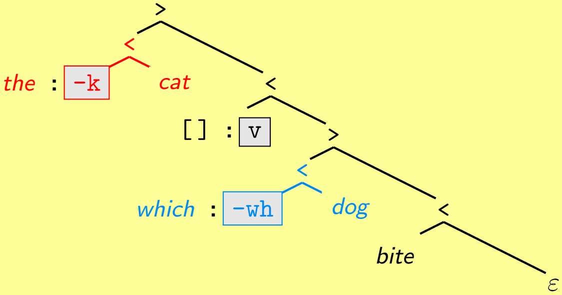
  Doing so, information about the tree structure and the relation between "still active" constituents can be ignored to a large extend.

- Examples . . .

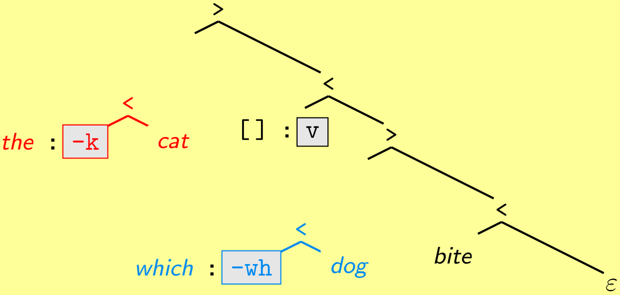
Trees as tuples



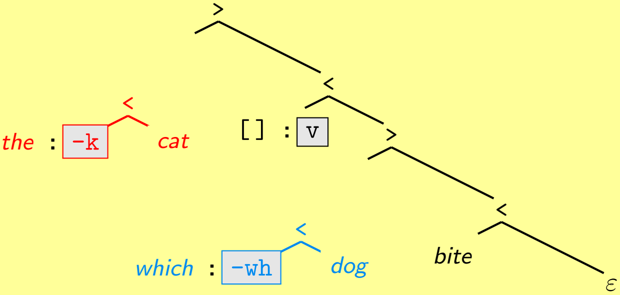
Trees as tuples



Trees as tuples

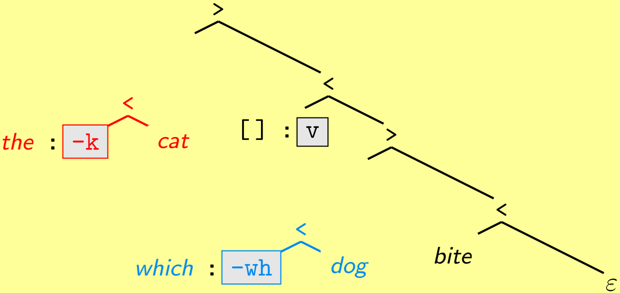


Trees as tuples



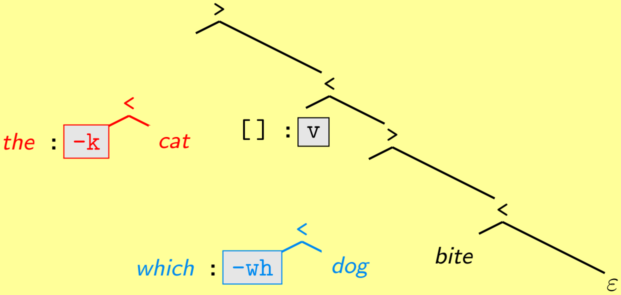
⟨ : v , : -wh , : -k ⟩

Trees as tuples



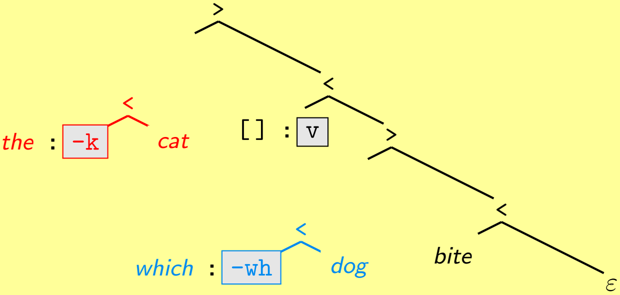
$$\Big\langle \quad : \boxed{\texttt{v}}\,, \qquad : \boxed{\texttt{-wh}}\,, \quad \textit{the}\frown\textit{cat} : \boxed{\texttt{-k}} \Big\rangle$$

Trees as tuples



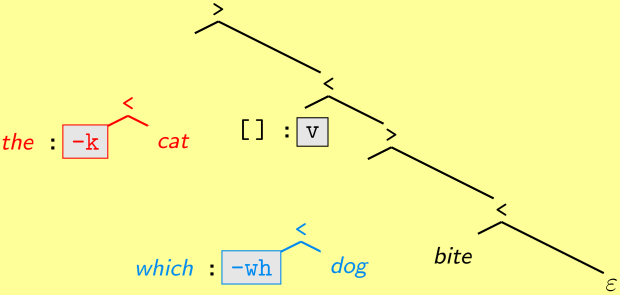
$\left\langle \quad : \boxed{\texttt{v}}\ ,\ which\frown dog : \boxed{\texttt{-wh}}\ ,\ the\frown cat : \boxed{\texttt{-k}} \right\rangle$

Trees as tuples



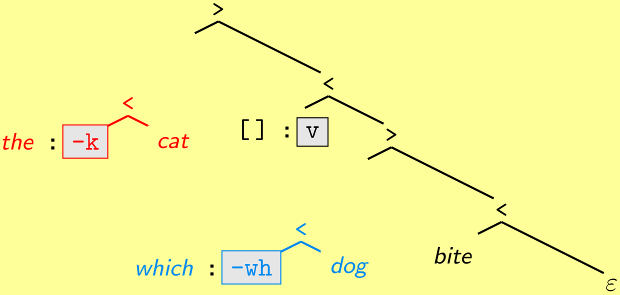
$\left\langle\ \textit{bite} : \texttt{v}\ ,\ \textit{which}\frown\textit{dog} : \texttt{-wh}\ ,\ \textit{the}\frown\textit{cat} : \texttt{-k}\ \right\rangle$

Trees as tuples



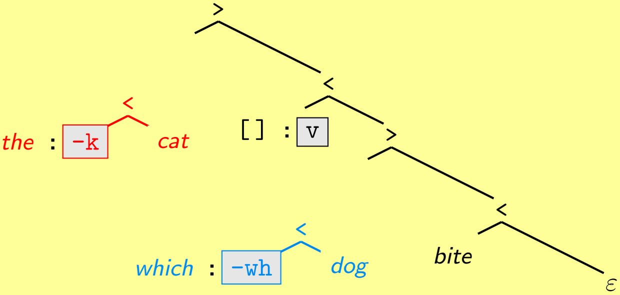
$\langle \quad : \texttt{v}, \qquad : \texttt{-wh}, \qquad : \texttt{-k} \rangle$

Trees as tuples



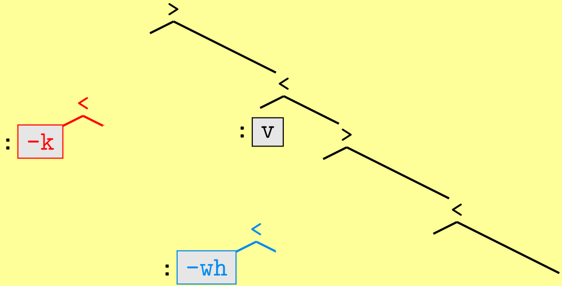
$\Big\langle$: , v , -wh , -k $\Big\rangle$

Trees as tuples



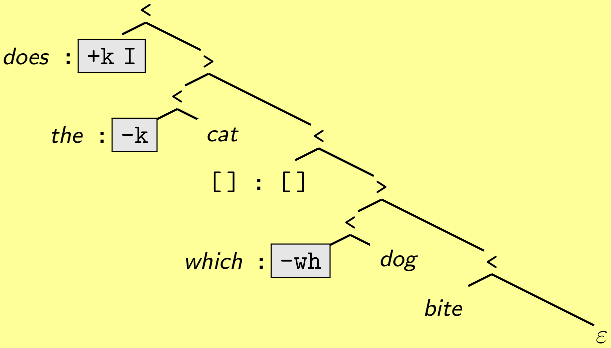
$$\left\langle \texttt{:}\ ,\ \boxed{\texttt{v}}\ ,\ \boxed{\texttt{-wh}}\ ,\ \boxed{\texttt{-k}} \right\rangle \left(\textit{bite}\ ,\ \textit{which}^\frown\textit{dog}\ ,\ \textit{the}^\frown\textit{cat} \right)$$

Trees as tuples

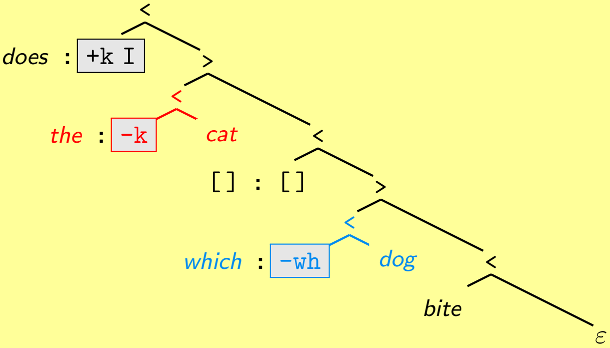


$$\left\langle\ :\ ,\ \boxed{\mathtt{v}}\ ,\ \boxed{\mathtt{-wh}}\ ,\ \boxed{\mathtt{-k}}\ \right\rangle \left(\ \_\_\_\ ,\ \_\_\_\_\_\_\ ,\ \_\_\_\_\_\ \right)$$

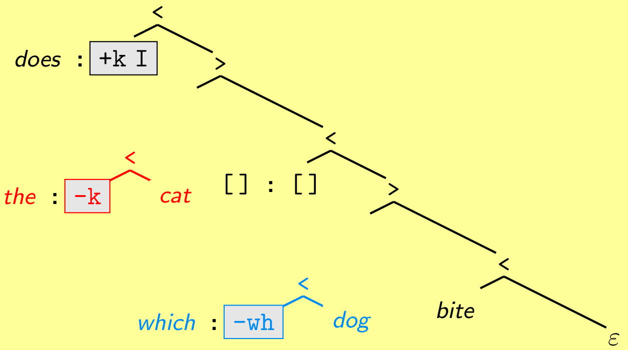
Trees as tuples



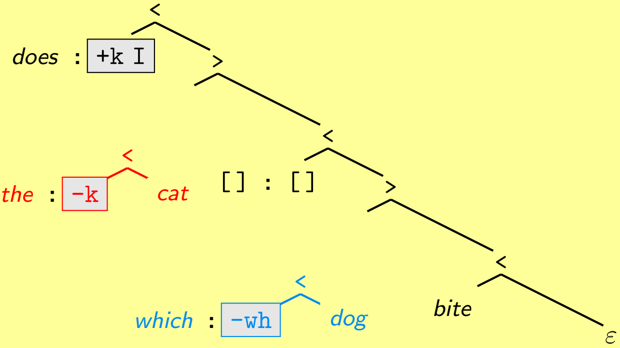
Trees as tuples



Trees as tuples

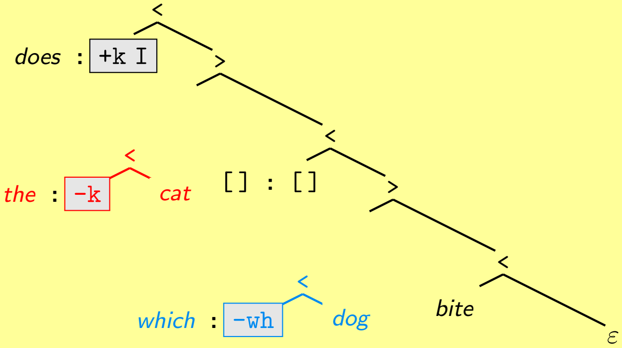


Trees as tuples



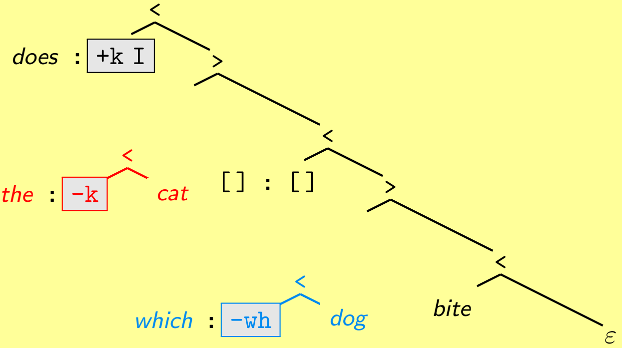
⟨ : +k I , : -wh , : -k ⟩

Trees as tuples



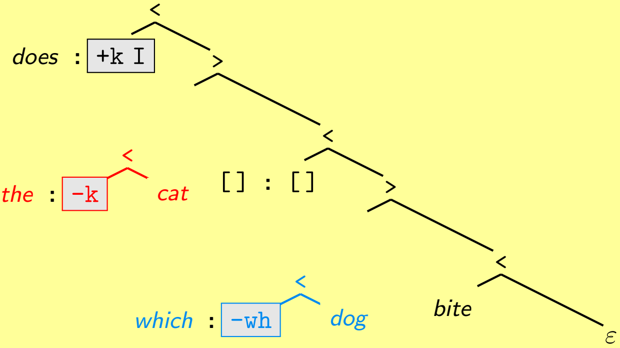
$\Big\langle$: +k I , *which⌢dog* : -wh , *the⌢cat* : -k $\Big\rangle$

Trees as tuples



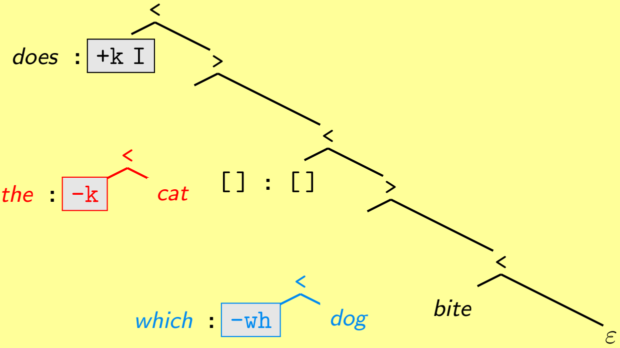
$$\left\langle \textit{does}\frown\textit{bite} : \boxed{\texttt{+k I}},\ \textit{which}\frown\textit{dog} : \boxed{\texttt{-wh}},\ \textit{the}\frown\textit{cat} : \boxed{\texttt{-k}} \right\rangle$$

Trees as tuples



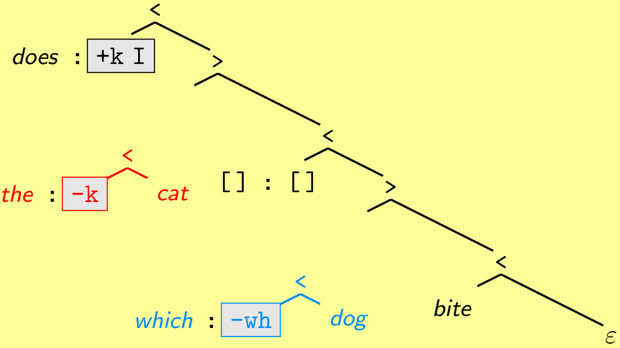
⟨ : +k I , : -wh , : -k ⟩

Trees as tuples



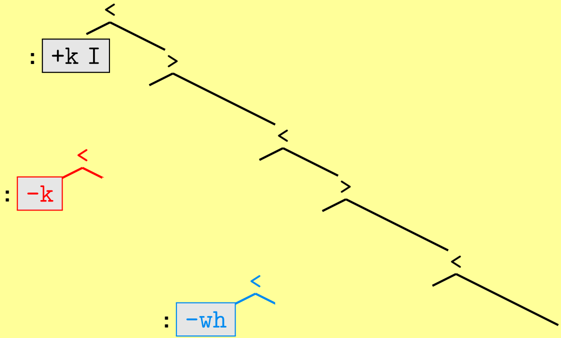
⟨ : , +k I , -wh , -k ⟩

Trees as tuples



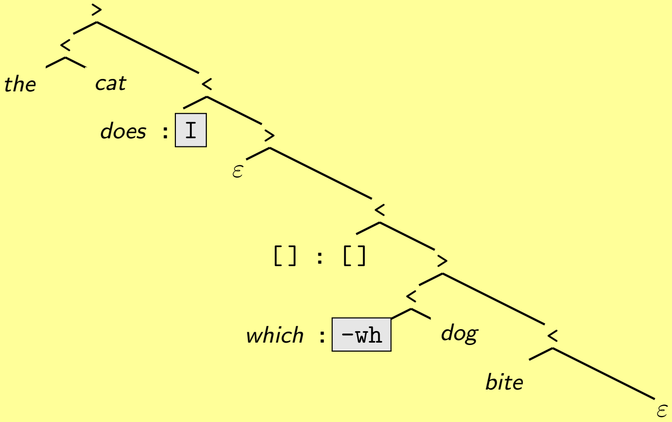
$\langle$: , +k I , -wh , -k $\rangle$ (*does⌢bite* , *which⌢dog* , *the⌢cat*)

Trees as tuples

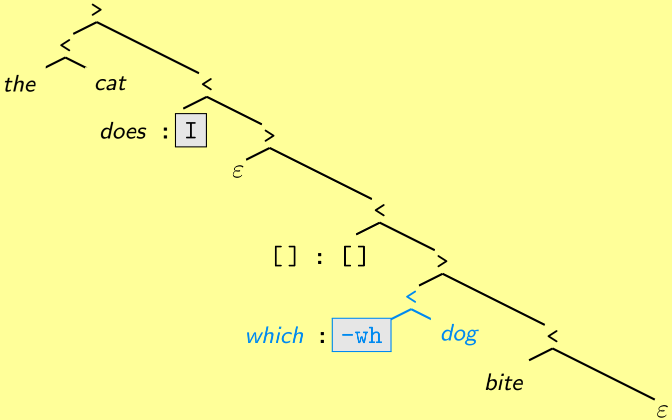


⟨ : , +k I , -wh , -k ⟩(________ , ________ , ________)

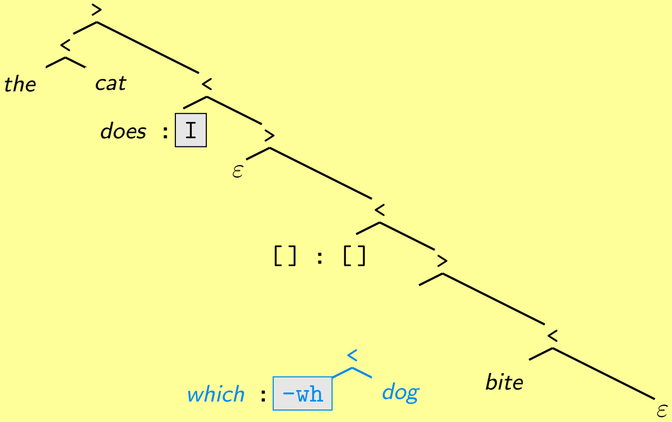
Trees as tuples



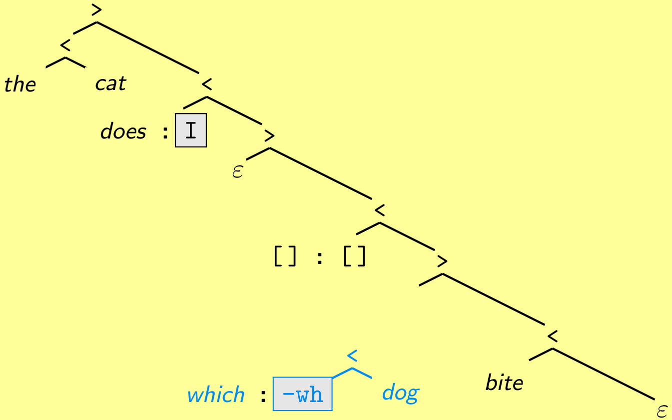
Trees as tuples



Trees as tuples

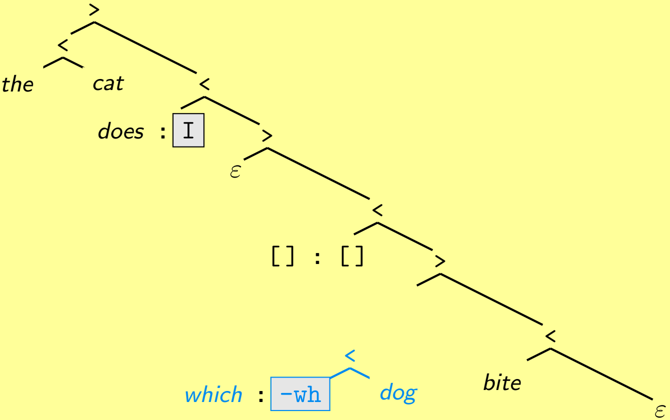


Trees as tuples



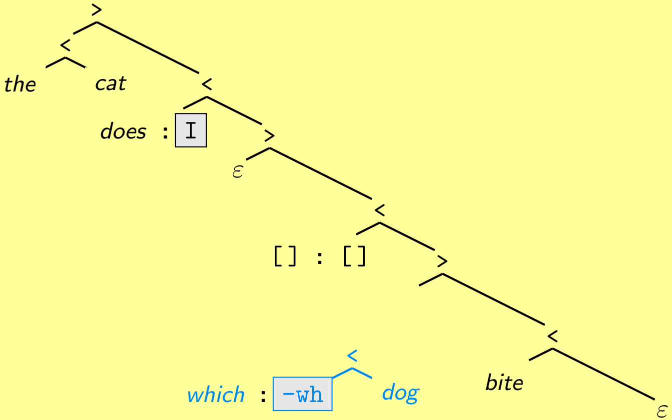
⟨ : I , : -wh ⟩

Trees as tuples



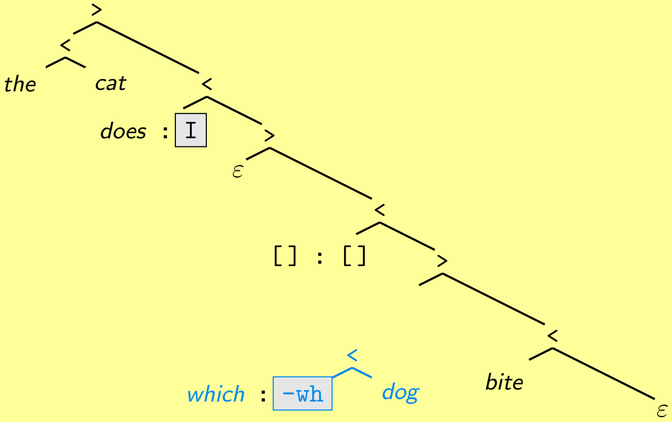
$\langle$: I , *which⁀dog* : -wh $\rangle$

Trees as tuples



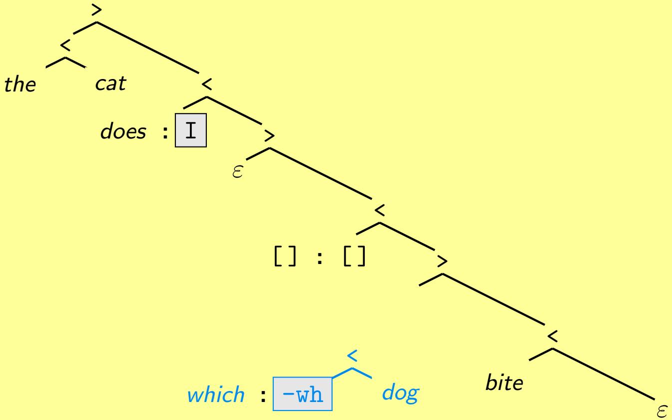
$\left\langle\ \textit{the}\frown\textit{cat}\frown\textit{does}\frown\textit{bite} : \texttt{I}\ ,\ \textit{which}\frown\textit{dog} : \texttt{-wh}\ \right\rangle$

Trees as tuples



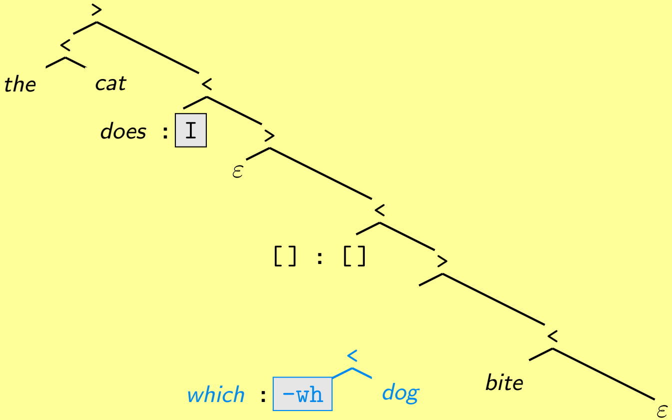
$\langle$: I , : -wh $\rangle$

Trees as tuples



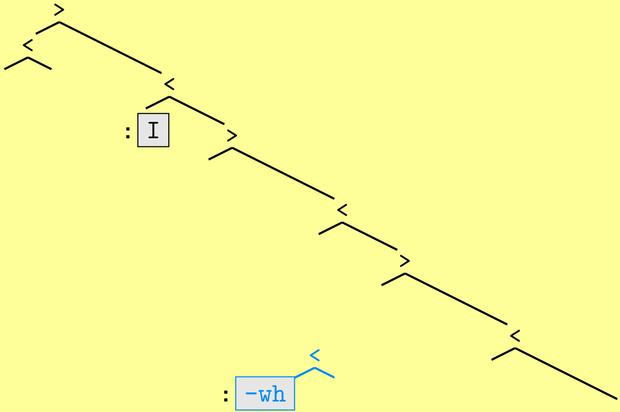
⟨ : , I , -wh ⟩

Trees as tuples



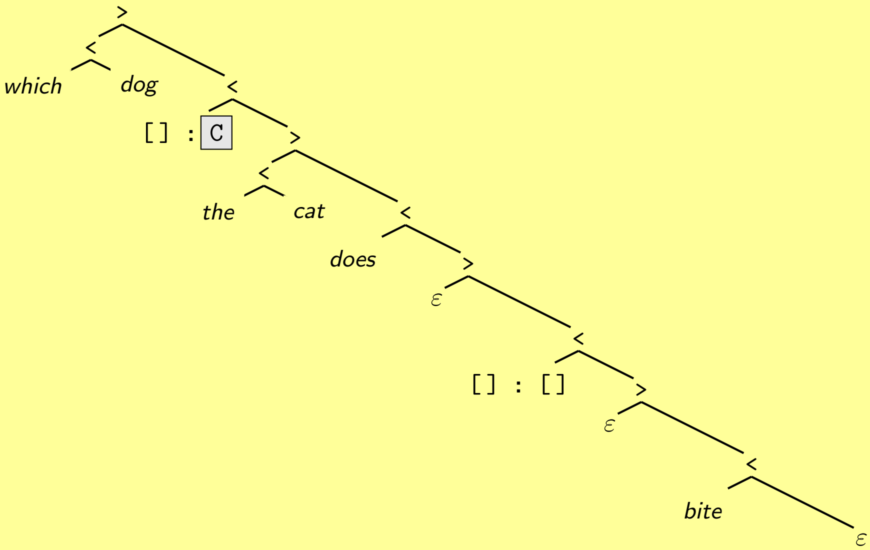
$\Big\langle$: , I , -wh $\Big\rangle$ (*the*⌢*cat*⌢*does*⌢*bite* , *which*⌢*dog*)

Trees as tuples

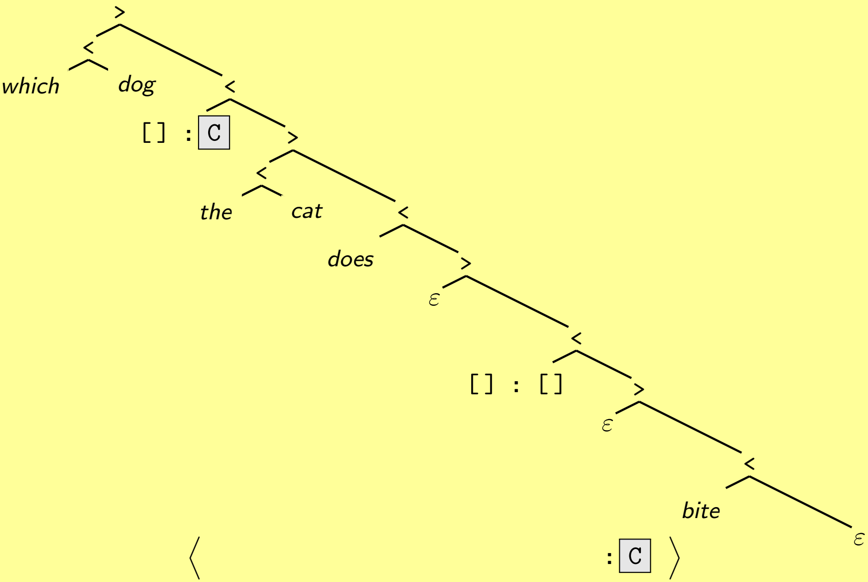


$$\left\langle\ :\ ,\ \boxed{\texttt{I}}\ ,\ \boxed{\texttt{-wh}}\ \right\rangle \left(\underline{\hspace{6cm}}\ ,\ \underline{\hspace{3cm}}\right)$$

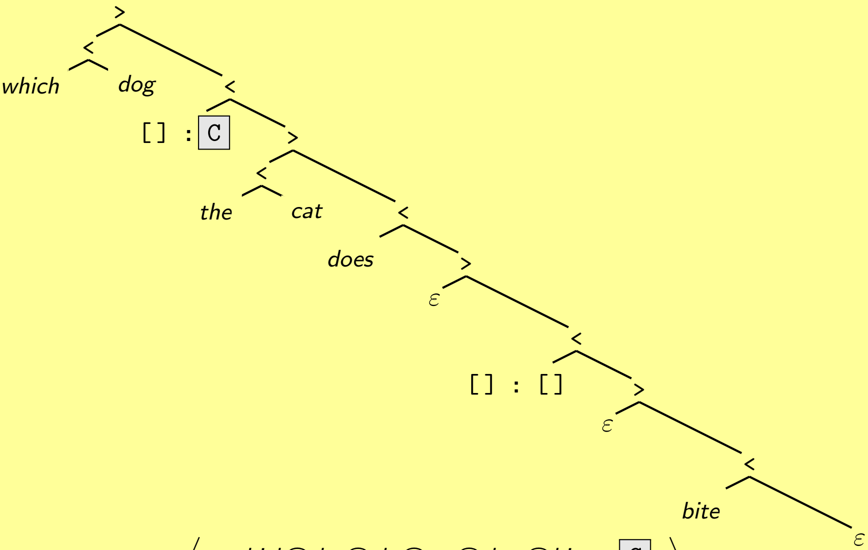
Trees as tuples



Trees as tuples

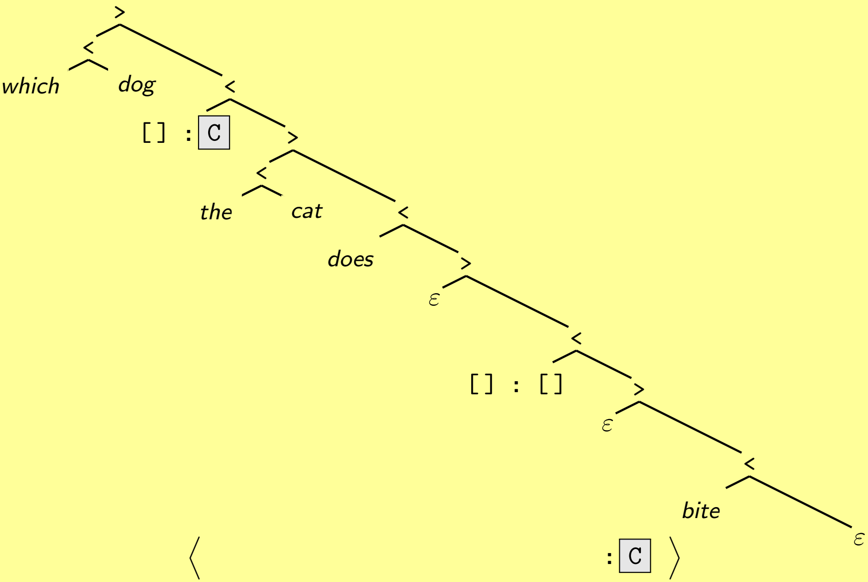


Trees as tuples

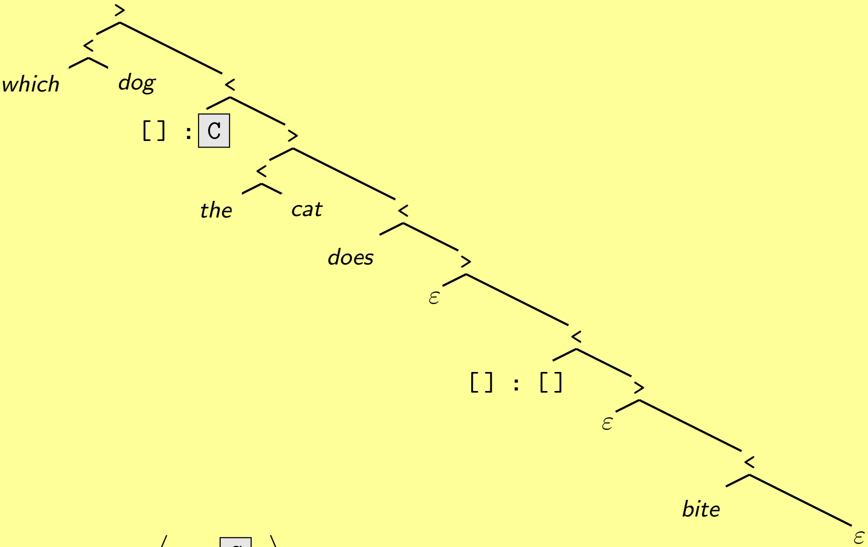


$\langle\ \textit{which}^\frown\textit{dog}^\frown\textit{the}^\frown\textit{cat}^\frown\textit{does}^\frown\textit{bite}$: C $\rangle$

Trees as tuples

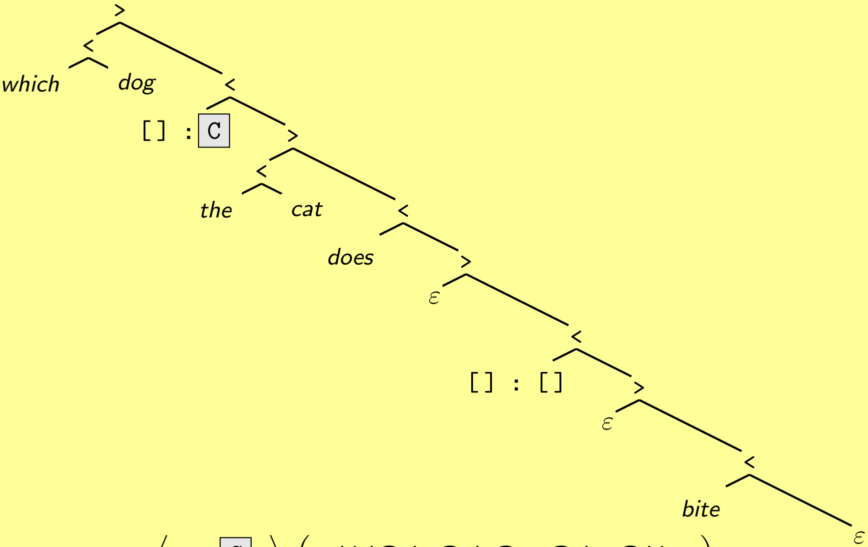


Trees as tuples



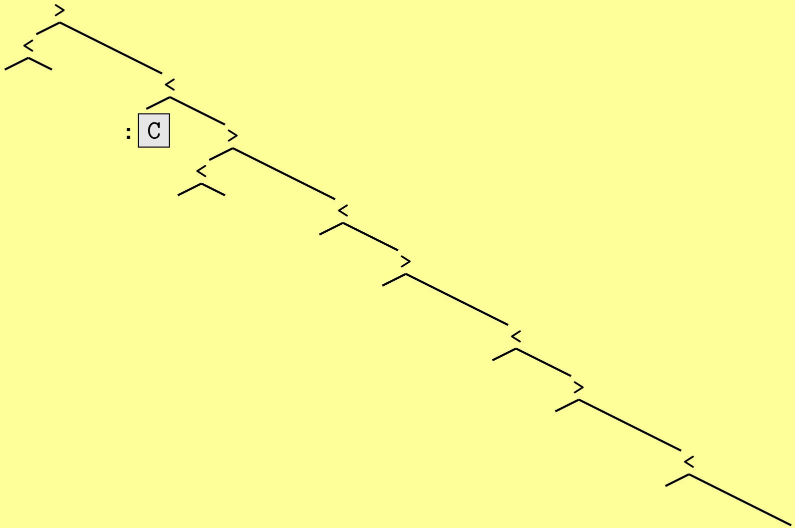
$\langle\ :\ ,\ \boxed{\mathtt{C}}\ \rangle$

Trees as tuples



$$\left\langle\ \mathtt{:}\ ,\ \boxed{\mathtt{C}}\ \right\rangle \left(\ \textit{which}^\frown\textit{dog}^\frown\textit{the}^\frown\textit{cat}^\frown\textit{does}^\frown\textit{bite}\ \right)$$

Trees as tuples



$$\left\langle\ \mathbf{:}\ ,\ \boxed{\mathtt{C}}\ \right\rangle \left(\ \underline{\hspace{8cm}}\ \right)$$

Trees as tuples (general form)

$G = \langle$ Vocabulary , SynFeat , Lex , Ω , c $\rangle$ an MG.

tuple representation of some $\tau \in$ Closure(G)

$$\langle \bullet , \gamma_0 , \gamma_1 , \ldots , \gamma_k \rangle (x_0 , x_1 , \ldots , x_k)$$

$\bullet \in \{ : , :: \}$ $\gamma_i \in$ SynFeat* $x_i \in$ Vocabulary*

$\bullet = ::$ iff $\tau \in$ Lex

Trees as tuples (general form)

$G = \langle \text{Vocabulary}, \text{SynFeat}, \text{Lex}, \Omega, c \rangle$ an MG.

tuple representation of some $\tau \in \text{Closure}(G)$

$$\langle \bullet, \gamma_0, \gamma_1, \ldots, \gamma_k \rangle (x_0, x_1, \ldots, x_k)$$

$\bullet \in \{ :, :: \}$ $\gamma_i \in \text{SynFeat}^*$ $x_i \in \text{Vocabulary}^*$

$\bullet = ::$ iff $\tau \in \text{Lex}$

general form defines a partition on Closure(G)

Trees as tuples (general form)

$G = \langle$ Vocabulary , SynFeat , Lex , Ω , c $\rangle$ an MG.

tuple representation of some $\boldsymbol{\tau} \in$ Closure(G)

$$\langle \bullet , \gamma_0 , \gamma_1 , \ldots , \gamma_k \rangle$$

$\bullet \in \{ : , :: \}$ $\gamma_i \in \text{SynFeat}^*$

$\bullet = ::$ iff $\tau \in$ Lex

general form defines a partition on Closure(G)

Trees as tuples (general form)

$G = \langle \text{Vocabulary}, \text{SynFeat}, \text{Lex}, \Omega, c \rangle$ an MG.

tuple representation of some $\boldsymbol{\tau} \in \text{Closure}(G)$

$$\langle \bullet, \gamma_0, \gamma_1, \ldots, \gamma_k \rangle$$

$\bullet \in \{ :, :: \}$ $\gamma_i \in \text{SynFeat}^*$

$\bullet = ::$ iff $\tau \in \text{Lex}$

general form defines a partition on Closure(G)

- There are only finitely many possibilities for γ_i.

  "structure building by feature checking" and $\boldsymbol{\tau} \in \text{Closure}(G)$ implies:
  γ_i is the suffix of the syntactic feature part of the label of a lexical item

$$x :: \lambda \gamma_i \in \text{Lex}$$

Trees as tuples

- The tuple representation is compatible with the structure building operators, that is to say “merge” and “move,” can be canonically reformulated.

- The tuple representation is exactly what can be employed to define an equivalent MCFG. The only things missing are

  - the replacement of the terminal strings by variables as far as “merge” and “move” are concerned,
  - the introduction of terminating rules simulating “lexical insertion,” and
  - a reduction to a finite number of nonterminals and rules.

merge (reformulated)

merge 1:

$$\langle \boxed{\odot, \texttt{=f}\ \gamma}, \gamma_1, \ldots, \gamma_k \rangle (\boxed{x_0}, x_1, \ldots, x_k) \qquad \langle \boxed{\bullet, \texttt{f}\ \delta}, \delta_1, \ldots, \delta_l \rangle (\boxed{y_0}, y_1, \ldots, y_l)$$

merge (reformulated)

merge 1:

$\delta \neq \varepsilon$

$\langle \boxed{\odot\,,\texttt{=f}\,\gamma}\,, \gamma_1\,, \ldots\,, \gamma_k \rangle\, (\boxed{x_0}\,, x_1\,, \ldots\,, x_k\,)$ $\quad$ $\langle \boxed{\bullet\,,\texttt{f}\,\delta}\,, \delta_1\,, \ldots\,, \delta_l \rangle\, (\boxed{y_0}\,, y_1\,, \ldots\,, y_l\,)$

merge 1:

$\delta \neq \varepsilon$

$$\frac{\langle \boxed{\odot, \texttt{=f}\ \gamma}, \gamma_1, \ldots, \gamma_k \rangle (\boxed{x_0}, x_1, \ldots, x_k) \qquad \langle \boxed{\bullet, \texttt{f}\ \delta}, \delta_1, \ldots, \delta_l \rangle (\boxed{y_0}, y_1, \ldots, y_l)}{\langle \boxed{:, \texttt{=f}\ \gamma}, \gamma_1, \ldots, \gamma_k, \qquad \rangle (\boxed{x_0}, x_1, \ldots, x_k, \quad , \qquad)}$$

merge 1: $\delta \neq \varepsilon$

$$\frac{\langle \boxed{\odot\,, \texttt{=f}\ \gamma}, \gamma_1, \ldots, \gamma_k \rangle (\boxed{x_0}, x_1, \ldots, x_k) \qquad \langle \boxed{\bullet\,, \texttt{f}\ \delta}, \delta_1, \ldots, \delta_l \rangle (\boxed{y_0}, y_1, \ldots, y_l)}{\langle \boxed{:\,, \texttt{=f}\ \gamma}, \gamma_1, \ldots, \gamma_k, \boxed{\texttt{f}\ \delta}, \delta_1, \ldots, \delta_l \rangle (\boxed{x_0}, x_1, \ldots, x_k, \boxed{y_0}, y_1, \ldots, y_l)}$$

merge (reformulated)

merge 1: $\delta \neq \varepsilon$

$$\frac{\langle \boxed{\odot, \texttt{=f}\,\gamma}, \gamma_1, \ldots, \gamma_k \rangle (\boxed{x_0}, x_1, \ldots, x_k) \qquad \langle \boxed{\bullet, \texttt{f}\,\delta}, \delta_1, \ldots, \delta_l \rangle (\boxed{y_0}, y_1, \ldots, y_l)}{\langle \boxed{:, \not{\texttt{=}}\not{\texttt{f}}\,\gamma}, \gamma_1, \ldots, \gamma_k, \boxed{\not{\texttt{f}}\,\delta}, \delta_1, \ldots, \delta_l \rangle (\boxed{x_0}, x_1, \ldots, x_k, \boxed{y_0}, y_1, \ldots, y_l)}$$

merge (reformulated)

merge 1: $\delta \neq \varepsilon$

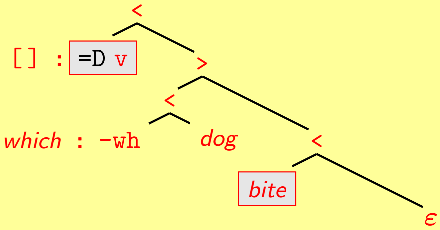
$$\frac{\langle \odot, \mathtt{=f}\ \gamma, \gamma_1, \ldots, \gamma_k \rangle (x_0, x_1, \ldots, x_k) \qquad \langle \bullet, \mathtt{f}\ \delta, \delta_1, \ldots, \delta_l \rangle (y_0, y_1, \ldots, y_l)}{\langle :, \cancel{\mathtt{=f}}\ \gamma, \gamma_1, \ldots, \gamma_k, \cancel{\mathtt{f}}\ \delta, \delta_1, \ldots, \delta_l \rangle (x_0, x_1, \ldots, x_k, y_0, y_1, \ldots, y_l)}$$

merge (reformulated)

merge 1:

$\delta \neq \varepsilon$

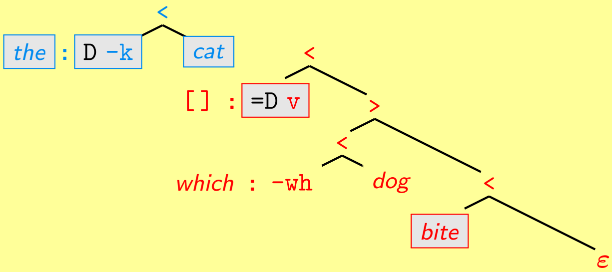
$$\frac{\langle \boxed{\odot, =\mathtt{f}\,\gamma}, \gamma_1, \ldots, \gamma_k \rangle (\boxed{x_0}, x_1, \ldots, x_k) \qquad \langle \boxed{\bullet, \mathtt{f}\,\delta}, \delta_1, \ldots, \delta_l \rangle (\boxed{y_0}, y_1, \ldots, y_l)}{\langle \boxed{:, \cancel{=\mathtt{f}}\,\gamma}, \gamma_1, \ldots, \gamma_k, \boxed{\cancel{\mathtt{f}}\,\delta}, \delta_1, \ldots, \delta_l \rangle (\boxed{x_0}, x_1, \ldots, x_k, \boxed{y_0}, y_1, \ldots, y_l)}$$



merge 1:

$\delta \neq \varepsilon$

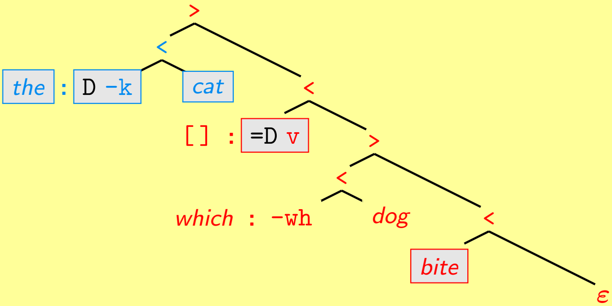
$$\frac{\langle\ \odot, \texttt{=f}\ \gamma, \gamma_1, \ldots, \gamma_k \rangle\ (x_0, x_1, \ldots, x_k) \qquad \langle\ \bullet, \texttt{f}\ \delta, \delta_1, \ldots, \delta_l \rangle\ (y_0, y_1, \ldots, y_l)}{\langle\ :, \not{\texttt{=f}}\ \gamma, \gamma_1, \ldots, \gamma_k, \not{\texttt{f}}\ \delta, \delta_1, \ldots, \delta_l \rangle\ (x_0, x_1, \ldots, x_k, y_0, y_1, \ldots, y_l)}$$



merge 1:

$\delta \neq \varepsilon$

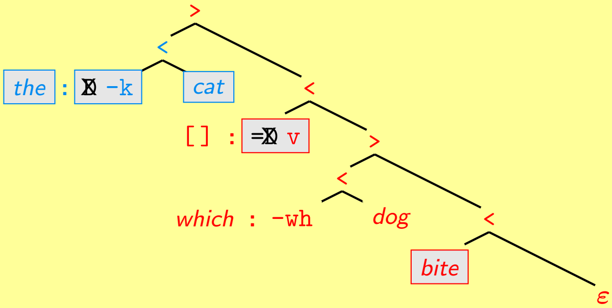
$$\frac{\left\langle \boxed{\odot, =\mathtt{f}\ \gamma}, \gamma_1, \ldots, \gamma_k \right\rangle \left(\boxed{x_0}, x_1, \ldots, x_k \right) \qquad \left\langle \boxed{\bullet, \mathtt{f}\ \delta}, \delta_1, \ldots, \delta_l \right\rangle \left(\boxed{y_0}, y_1, \ldots, y_l \right)}{\left\langle \boxed{:, \not{=}\!\not{\mathtt{f}}\ \gamma}, \gamma_1, \ldots, \gamma_k, \boxed{\not{\mathtt{f}}\ \delta}, \delta_1, \ldots, \delta_l \right\rangle \left(\boxed{x_0}, x_1, \ldots, x_k, \boxed{y_0}, y_1, \ldots, y_l \right)}$$



merge 1:

$\delta \neq \varepsilon$

$$\frac{\langle \odot, \texttt{=f}\,\gamma, \gamma_1, \ldots, \gamma_k \rangle (x_0, x_1, \ldots, x_k) \qquad \langle \bullet, \texttt{f}\,\delta, \delta_1, \ldots, \delta_l \rangle (y_0, y_1, \ldots, y_l)}{\langle :, \cancel{\texttt{=f}}\,\gamma, \gamma_1, \ldots, \gamma_k, \cancel{\texttt{f}}\,\delta, \delta_1, \ldots, \delta_l \rangle (x_0, x_1, \ldots, x_k, y_0, y_1, \ldots, y_l)}$$



merge (reformulated)

merge 1: $\delta \neq \varepsilon$

$$\langle \boxed{\odot, \texttt{=f}\ \gamma}, \gamma_1, \ldots, \gamma_k \rangle (\boxed{x_0}, x_1, \ldots, x_k) \qquad \langle \boxed{\bullet, \texttt{f}\ \delta}, \delta_1, \ldots, \delta_l \rangle (\boxed{y_0}, y_1, \ldots, y_l)$$

merge (reformulated)

merge 2:

$\langle \boxed{\odot\, , \texttt{=f}\ \gamma}\, , \gamma_1 , \ldots , \gamma_k \rangle (\boxed{x_0} , x_1 , \ldots , x_k) \qquad \langle \boxed{\bullet\, , \texttt{f}}\, , \delta_1 , \ldots , \delta_l \rangle (\boxed{y_0} , y_1 , \ldots , y_l)$

merge 2:

$\langle$ :: , =f γ , γ_1 , . . . , γ_k $\rangle$ (x_0 , x_1 , . . . , x_k) $\langle$ • , f , δ_1 , . . . , δ_l $\rangle$ (y_0 , y_1 , . . . , y_l)

merge 2:

$$\frac{\langle \boxed{::, =\mathtt{f}\,\gamma}, \gamma_1, \ldots, \gamma_k \rangle (\boxed{x_0}, x_1, \ldots, x_k) \qquad \langle \boxed{\bullet, \mathtt{f}\quad}, \delta_1, \ldots, \delta_l \rangle (\boxed{y_0}, y_1, \ldots, y_l)}{\langle \boxed{:, =\mathtt{f}\,\gamma}, \gamma_1, \ldots, \gamma_k, \qquad \rangle (\boxed{x_0} \qquad , x_1, \ldots, x_k, \qquad)}$$

merge 2:

$$\frac{\langle\ \boxed{::\ ,\ \mathtt{=f}\ \gamma}\ ,\gamma_1\ ,\ldots,\gamma_k\ \rangle\ (\ \boxed{x_0}\ ,x_1\ ,\ldots,x_k\ ) \qquad \langle\ \boxed{\bullet\ ,\ \mathtt{f}}\ ,\delta_1\ ,\ldots,\delta_l\ \rangle\ (\ \boxed{y_0}\ ,y_1\ ,\ldots,y_l\ )}{\langle\ \boxed{:\ ,\ \mathtt{=f}\ \gamma}\ ,\gamma_1\ ,\ldots,\gamma_k\ ,\delta_1\ ,\ldots,\delta_l\ \rangle\ (\ \boxed{x_0}\ \quad ,x_1\ ,\ldots,x_k\ ,y_1\ ,\ldots,y_l\ )}$$

merge 2:

$$\frac{\langle\, \boxed{\texttt{::}\,,\texttt{=f}\;\gamma}\,,\gamma_1,\ldots,\gamma_k\,\rangle\,(\,\boxed{x_0}\,,x_1,\ldots,x_k\,) \qquad \langle\,\boxed{\bullet\,,\texttt{f}}\,,\delta_1,\ldots,\delta_l\,\rangle\,(\,\boxed{y_0}\,,y_1,\ldots,y_l\,)}{\langle\,\boxed{\texttt{:}\,,\texttt{=f}\;\gamma}\,,\gamma_1,\ldots,\gamma_k,\delta_1,\ldots,\delta_l\,\rangle\,(\,\boxed{x_0}\,\boxed{y_0}\,,x_1,\ldots,x_k,y_1,\ldots,y_l\,)}$$

merge 2:

$$\frac{\langle \boxed{::, =\mathtt{f}\ \gamma}, \gamma_1, \ldots, \gamma_k \rangle\ (\boxed{x_0}, x_1, \ldots, x_k) \qquad \langle \boxed{\bullet, \mathtt{f}}, \delta_1, \ldots, \delta_l \rangle\ (\boxed{y_0}, y_1, \ldots, y_l)}{\langle \boxed{:, \not{=}\mathtt{f}\ \gamma}, \gamma_1, \ldots, \gamma_k, \delta_1, \ldots, \delta_l \rangle\ (\boxed{x_0}\ \boxed{y_0}, x_1, \ldots, x_k, y_1, \ldots, y_l)}$$

merge (reformulated)

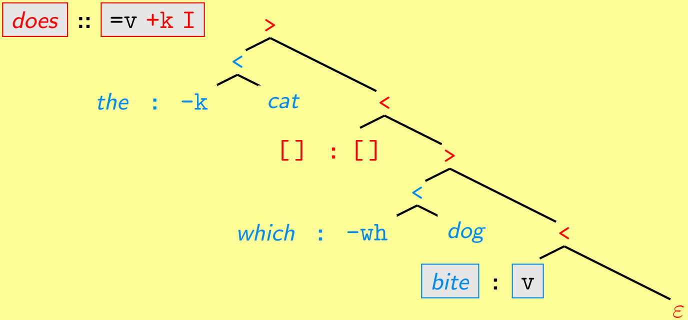
merge 2:

$$\frac{\langle \boxed{::,\ \mathtt{=f}\ \gamma}, \gamma_1, \ldots, \gamma_k \rangle (\boxed{x_0}, x_1, \ldots, x_k) \qquad \langle \boxed{\bullet,\ \mathtt{f}}, \delta_1, \ldots, \delta_l \rangle (\boxed{y_0}, y_1, \ldots, y_l)}{\langle \boxed{:,\ \not{\mathtt{=f}}\ \gamma}, \gamma_1, \ldots, \gamma_k, \delta_1, \ldots, \delta_l \rangle (\boxed{x_0}\boxed{y_0}, x_1, \ldots, x_k, y_1, \ldots, y_l)}$$

does :: =v +k I

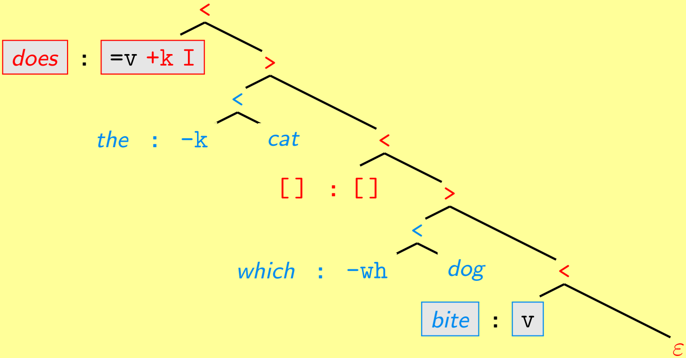
merge 2:

$$\frac{\langle\, \boxed{::\,,\, =\mathtt{f}\ \gamma}\,,\, \gamma_1 ,\, \ldots ,\, \gamma_k \,\rangle\ (\, \boxed{x_0}\,,\, x_1 ,\, \ldots ,\, x_k \,) \qquad \langle\, \boxed{\bullet\,,\, \mathtt{f}}\,,\, \delta_1 ,\, \ldots ,\, \delta_l \,\rangle\ (\, \boxed{y_0}\,,\, y_1 ,\, \ldots ,\, y_l \,)}{\langle\, \boxed{:\,,\, \cancel{=\mathtt{f}}\ \gamma}\,,\, \gamma_1 ,\, \ldots ,\, \gamma_k ,\, \delta_1 ,\, \ldots ,\, \delta_l \,\rangle\ (\, \boxed{x_0}\, \boxed{y_0}\,,\, x_1 ,\, \ldots ,\, x_k ,\, y_1 ,\, \ldots ,\, y_l \,)}$$



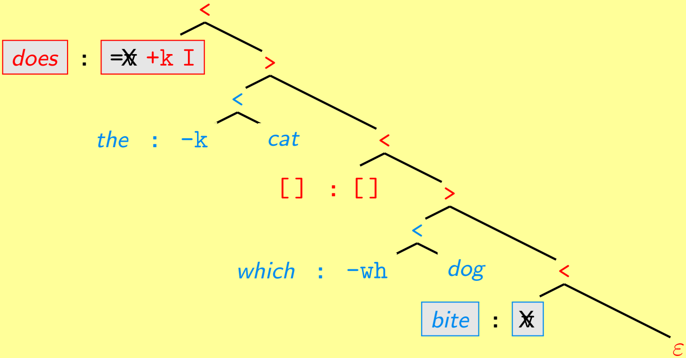
merge 2:

$$\frac{\langle \boxed{::\,,\ \mathtt{=f}\ \gamma}, \gamma_1, \ldots, \gamma_k \rangle\, (\boxed{x_0}, x_1, \ldots, x_k) \qquad \langle \boxed{\bullet\,,\ \mathtt{f}}, \delta_1, \ldots, \delta_l \rangle\, (\boxed{y_0}, y_1, \ldots, y_l)}{\langle \boxed{:\,,\ \not{=}\mathtt{f}\ \gamma}, \gamma_1, \ldots, \gamma_k, \delta_1, \ldots, \delta_l \rangle\, (\boxed{x_0}\boxed{y_0}, x_1, \ldots, x_k, y_1, \ldots, y_l)}$$



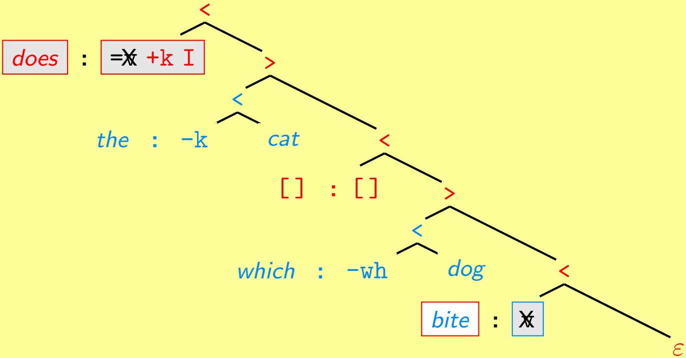
merge 2:

$$\frac{\langle\, \boxed{::\,,\, =\mathtt{f}\ \gamma}\,,\gamma_1,\ldots,\gamma_k \,\rangle\, (\,\boxed{x_0}\,, x_1, \ldots, x_k\,) \qquad \langle\, \boxed{\bullet\,,\,\mathtt{f}}\,,\delta_1,\ldots,\delta_l \,\rangle\, (\,\boxed{y_0}\,, y_1, \ldots, y_l\,)}{\langle\, \boxed{:\,,\, \not{=}\!\mathtt{f}\ \gamma}\,,\gamma_1,\ldots,\gamma_k,\delta_1,\ldots,\delta_l \,\rangle\, (\,\boxed{x_0}\,\boxed{y_0}\,, x_1, \ldots, x_k, y_1, \ldots, y_l\,)}$$



merge 2:

$$\frac{\left\langle \boxed{::, \texttt{=f}\ \gamma}, \gamma_1, \ldots, \gamma_k \right\rangle \left(\boxed{x_0}, x_1, \ldots, x_k \right) \qquad \left\langle \boxed{\bullet, \texttt{f}}, \delta_1, \ldots, \delta_l \right\rangle \left(\boxed{y_0}, y_1, \ldots, y_l \right)}{\left\langle \boxed{:, \cancel{\texttt{=f}}\ \gamma}, \gamma_1, \ldots, \gamma_k, \delta_1, \ldots, \delta_l \right\rangle \left(\boxed{x_0} \boxed{y_0}, x_1, \ldots, x_k, y_1, \ldots, y_l \right)}$$



merge (reformulated)

merge 2:

$\langle\, \boxed{::, \texttt{=f}\ \gamma}, \gamma_1, \ldots, \gamma_k \,\rangle\, (\boxed{x_0}, x_1, \ldots, x_k) \qquad \langle\, \boxed{\bullet, \texttt{f}}, \delta_1, \ldots, \delta_l \,\rangle\, (\boxed{y_0}, y_1, \ldots, y_l)$

merge (reformulated)

merge 3:

$\langle$ `: , =f` γ , γ_1 , . . . , γ_k $\rangle$ (x_0 , x_1 , . . . , x_k) $\langle$ ● , `f` , δ_1 , . . . , δ_l $\rangle$ (y_0 , y_1 , . . . , y_l)

merge 3:

$$\frac{\langle\ \boxed{\texttt{:}\ ,\ \texttt{=f}\ \gamma}\ ,\gamma_1,\ldots,\gamma_k\ \rangle\ (\ \boxed{x_0}\ ,x_1,\ldots,x_k\ ) \qquad \langle\ \boxed{\bullet\ ,\ \texttt{f}}\ ,\delta_1,\ldots,\delta_l\ \rangle\ (\ \boxed{y_0}\ ,y_1,\ldots,y_l\ )}{\langle\ \boxed{\texttt{:}\ ,\ \not{\texttt{=f}}\ \gamma}\ ,\gamma_1,\ldots,\gamma_k,\delta_1,\ldots,\delta_l\ \rangle\ (\ \boxed{x_0}\ \boxed{y_0}\ ,x_1,\ldots,x_k,y_1,\ldots,y_l\ )}$$

merge 3:

$$\frac{\left\langle \boxed{\mathtt{:}\,,\,\mathtt{=f}\;\gamma}\,,\gamma_1,\ldots,\gamma_k \right\rangle \left(\boxed{x_0}\,,x_1,\ldots,x_k \right) \qquad \left\langle \boxed{\bullet\,,\,\mathtt{f}}\,,\delta_1,\ldots,\delta_l \right\rangle \left(\boxed{y_0}\,,y_1,\ldots,y_l \right)}{\left\langle \boxed{\mathtt{:}\,,\,\cancel{\mathtt{=f}}\;\gamma}\,,\gamma_1,\ldots,\gamma_k,\delta_1,\ldots,\delta_l \right\rangle \left(\boxed{y_0}\,\boxed{x_0}\,,x_1,\ldots,x_k,y_1,\ldots,y_l \right)}$$

move 1:

$$\left\langle \boxed{\texttt{:}\,,\texttt{+f}\;\gamma}\,,\gamma_1\,,\ldots,\gamma_{j-1}\,,\boxed{\texttt{-f}\;\delta}\,,\gamma_{j+1}\,,\ldots,\gamma_k \right\rangle \left(\boxed{x_0}\,,x_1\,,\ldots,x_{j-1}\,,\boxed{x_j}\,,x_{j+1}\,,\ldots,x_k \right)$$

move

move 1:

$$\langle \boxed{\texttt{:}\,,\texttt{+f}\,\gamma}\,, \gamma_1, \ldots, \gamma_{j-1}, \boxed{\texttt{-f}\,\delta}\,, \gamma_{j+1}, \ldots, \gamma_k \rangle \,(\, \boxed{x_0}\,, x_1, \ldots, x_{j-1}, \boxed{x_j}\,, x_{j+1}, \ldots, x_k \,)$$

move 1:

$$\langle\ \boxed{\texttt{:}\ ,\texttt{+f}\ \gamma}\ ,\gamma_1\ ,\ldots,\gamma_{j-1}\ ,\ \boxed{\texttt{-f}\ \delta}\ ,\gamma_{j+1}\ ,\ldots,\gamma_k\ \rangle\ (\ \boxed{x_0}\ ,x_1\ ,\ldots,x_{j-1}\ ,\ \boxed{x_j}\ ,x_{j+1}\ ,\ldots,x_k\ )$$

move 1: $\delta \neq \varepsilon$

$$\langle \texttt{:}, \texttt{+f}\,\gamma, \gamma_1, \ldots, \gamma_{j-1}, \texttt{-f}\,\delta, \gamma_{j+1}, \ldots, \gamma_k \rangle (x_0, x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_k)$$

move

(reformulated)

move 1: $\delta \neq \varepsilon$

$$\frac{\langle\ \texttt{:}\ ,\texttt{+f}\ \gamma, \gamma_1, \ldots, \gamma_{j-1}, \texttt{-f}\ \delta, \gamma_{j+1}, \ldots, \gamma_k \rangle\ (x_0, x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_k)}{\langle\ \texttt{:}\ ,\texttt{+f}\ \gamma, \gamma_1, \ldots, \gamma_{j-1}, \qquad \gamma_{j+1}, \ldots, \gamma_k \rangle\ (x_0, x_1, \ldots, x_{j-1}, \qquad x_{j+1}, \ldots, x_k)}$$

move (reformulated)

move 1: $\delta \neq \varepsilon$

$$\frac{\langle\, \texttt{:}\,,\texttt{+f}\,\gamma\,, \gamma_1, \ldots, \gamma_{j-1}, \texttt{-f}\,\delta\,, \gamma_{j+1}, \ldots, \gamma_k \,\rangle\,(\, x_0, x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_k \,)}{\langle\, \texttt{:}\,,\texttt{+f}\,\gamma\,, \gamma_1, \ldots, \gamma_{j-1}, \texttt{-f}\,\delta\,, \gamma_{j+1}, \ldots, \gamma_k \,\rangle\,(\, x_0, x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_k \,)}$$

move (reformulated)

move 1: $\delta \neq \varepsilon$

$$\frac{\langle \boxed{\texttt{:}\,,\texttt{+f}\,\gamma}, \gamma_1, \dots, \gamma_{j-1}, \boxed{\texttt{-f}\,\delta}, \gamma_{j+1}, \dots, \gamma_k \rangle \left(\boxed{x_0}, x_1, \dots, x_{j-1}, \boxed{x_j}, x_{j+1}, \dots, x_k \right)}{\langle \boxed{\texttt{:}\,,\texttt{+}\cancel{\texttt{f}}\,\gamma}, \gamma_1, \dots, \gamma_{j-1}, \boxed{\texttt{-}\cancel{\texttt{f}}\,\delta}, \gamma_{j+1}, \dots, \gamma_k \rangle \left(\boxed{x_0}, x_1, \dots, x_{j-1}, \boxed{x_j}, x_{j+1}, \dots, x_k \right)}$$

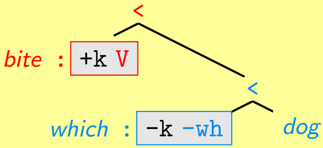
move (reformulated)

move 1: $\delta \neq \varepsilon$

$$\frac{\langle \boxed{\texttt{:}\,,\texttt{+f}\,\gamma}\,, \gamma_1, \ldots, \gamma_{j-1}, \boxed{\texttt{-f}\,\delta}\,, \gamma_{j+1}, \ldots, \gamma_k \rangle\ (\boxed{x_0}\,, x_1, \ldots, x_{j-1}, \boxed{x_j}\,, x_{j+1}, \ldots, x_k)}{\langle \boxed{\texttt{:}\,,\texttt{+}\cancel{\texttt{f}}\,\gamma}\,, \gamma_1, \ldots, \gamma_{j-1}, \boxed{\texttt{-}\cancel{\texttt{f}}\,\delta}\,, \gamma_{j+1}, \ldots, \gamma_k \rangle\ (\boxed{x_0}\,, x_1, \ldots, x_{j-1}, \boxed{x_j}\,, x_{j+1}, \ldots, x_k)}$$

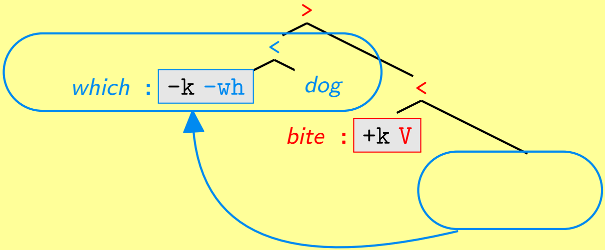
move 1: $\delta \neq \varepsilon$

$$\frac{\langle\, : , \texttt{+f}\ \gamma , \gamma_1 , \ldots , \gamma_{j-1} , \texttt{-f}\ \delta , \gamma_{j+1} , \ldots , \gamma_k \,\rangle \left(x_0 , x_1 , \ldots , x_{j-1} , x_j , x_{j+1} , \ldots , x_k \right)}{\langle\, : , \not{\texttt{+f}}\ \gamma , \gamma_1 , \ldots , \gamma_{j-1} , \not{\texttt{-f}}\ \delta , \gamma_{j+1} , \ldots , \gamma_k \,\rangle \left(x_0 , x_1 , \ldots , x_{j-1} , x_j , x_{j+1} , \ldots , x_k \right)}$$



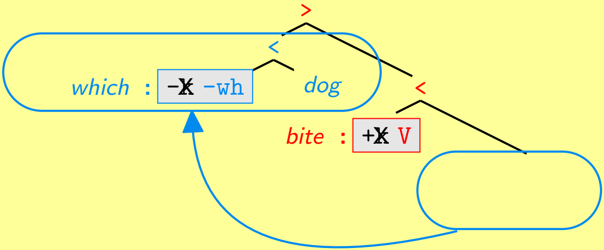
move 1: $\delta \neq \varepsilon$

$$\frac{\langle\ :\ ,\mathtt{+f}\ \gamma\ ,\gamma_1,\ldots,\gamma_{j-1},\ \mathtt{-f}\ \delta\ ,\gamma_{j+1},\ldots,\gamma_k\ \rangle\ (\ x_0\ ,x_1,\ldots,x_{j-1},\ x_j\ ,x_{j+1},\ldots,x_k\ )}{\langle\ :\ ,\cancel{\mathtt{+f}}\ \gamma\ ,\gamma_1,\ldots,\gamma_{j-1},\ \cancel{\mathtt{-f}}\ \delta\ ,\gamma_{j+1},\ldots,\gamma_k\ \rangle\ (\ x_0\ ,x_1,\ldots,x_{j-1},\ x_j\ ,x_{j+1},\ldots,x_k\ )}$$



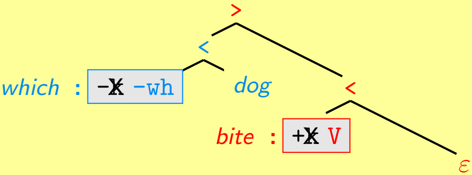
move 1: $\delta \neq \varepsilon$

$$\frac{\langle\ \boxed{:\ ,\ +\mathtt{f}\ \gamma}\ , \gamma_1 , \ldots , \gamma_{j-1} , \boxed{-\mathtt{f}\ \delta}\ , \gamma_{j+1} , \ldots , \gamma_k \ \rangle \ (\ \boxed{x_0}\ , x_1 , \ldots , x_{j-1} , \boxed{x_j}\ , x_{j+1} , \ldots , x_k\ )}{\langle\ \boxed{:\ ,\ +\not{\mathtt{f}}\ \gamma}\ , \gamma_1 , \ldots , \gamma_{j-1} , \boxed{-\not{\mathtt{f}}\ \delta}\ , \gamma_{j+1} , \ldots , \gamma_k \ \rangle \ (\ \boxed{x_0}\ , x_1 , \ldots , x_{j-1} , \boxed{x_j}\ , x_{j+1} , \ldots , x_k\ )}$$



move 1: $\delta \neq \varepsilon$

$$\frac{\langle\, \boxed{:\,, +f\ \gamma}\,, \gamma_1, \ldots, \gamma_{j-1}, \boxed{-f\ \delta}\,, \gamma_{j+1}, \ldots, \gamma_k \,\rangle\ (\boxed{x_0}\,, x_1, \ldots, x_{j-1}, \boxed{x_j}\,, x_{j+1}, \ldots, x_k)}{\langle\, \boxed{:\,, +\not{f}\ \gamma}\,, \gamma_1, \ldots, \gamma_{j-1}, \boxed{-\not{f}\ \delta}\,, \gamma_{j+1}, \ldots, \gamma_k \,\rangle\ (\boxed{x_0}\,, x_1, \ldots, x_{j-1}, \boxed{x_j}\,, x_{j+1}, \ldots, x_k)}$$



move (reformulated)

move 1: $\delta \neq \varepsilon$

$$\langle\, \texttt{:}\,,\texttt{+f}\,\gamma\,, \gamma_1\,, \ldots\,, \gamma_{j-1}\,, \texttt{-f}\,\delta\,, \gamma_{j+1}\,, \ldots\,, \gamma_k \,\rangle\, (\, x_0\,, x_1\,, \ldots\,, x_{j-1}\,, x_j\,, x_{j+1}\,, \ldots\,, x_k\,)$$

move

move 2:

$$\langle \boxed{\texttt{:}\,,\texttt{+f}\,\gamma}\,,\gamma_1\,,\ldots,\gamma_{j-1}\,,\boxed{\texttt{-f}}\,,\gamma_{j+1}\,,\ldots,\gamma_k\,\rangle\,(\,\boxed{x_0}\,,x_1\,,\ldots,x_{j-1}\,,\boxed{x_j}\,,x_{j+1}\,,\ldots,x_k\,)$$

move 2:

$$\frac{\left\langle \boxed{\texttt{:}\,,\texttt{+f}\;\gamma}\,,\gamma_1,\ldots,\gamma_{j-1},\boxed{\texttt{-f}}\,,\gamma_{j+1},\ldots,\gamma_k\right\rangle\left(\boxed{x_0}\,,x_1,\ldots,x_{j-1},\boxed{x_j}\,,x_{j+1},\ldots,x_k\right)}{\left\langle \boxed{\texttt{:}\,,\texttt{+f}\;\gamma}\,,\gamma_1,\ldots,\gamma_{j-1}\quad,\quad\gamma_{j+1},\ldots,\gamma_k\right\rangle\left(\boxed{x_j}\,\boxed{x_0}\,,x_1,\ldots,x_{j-1}\quad,\quad x_{j+1},\ldots,x_k\right)}$$

move (reformulated)

move 2:

$$\frac{\langle\ \boxed{\mathtt{:}\,,\,\mathtt{+f}\ \gamma}\,,\gamma_1\,,\ldots,\gamma_{j-1}\,,\boxed{\mathtt{-f}}\,,\gamma_{j+1}\,,\ldots,\gamma_k\ \rangle\ (\ \boxed{x_0}\,,x_1\,,\ldots,x_{j-1}\,,\boxed{x_j}\,,x_{j+1}\,,\ldots,x_k\ )}{\langle\ \boxed{\mathtt{:}\,,\,\cancel{\mathtt{+f}}\ \gamma}\,,\gamma_1\,,\ldots,\gamma_{j-1}\quad,\quad\gamma_{j+1}\,,\ldots,\gamma_k\ \rangle\ (\ \boxed{x_j}\,\boxed{x_0}\,,x_1\,,\ldots,x_{j-1}\quad,\quad x_{j+1}\,,\ldots,x_k\ )}$$

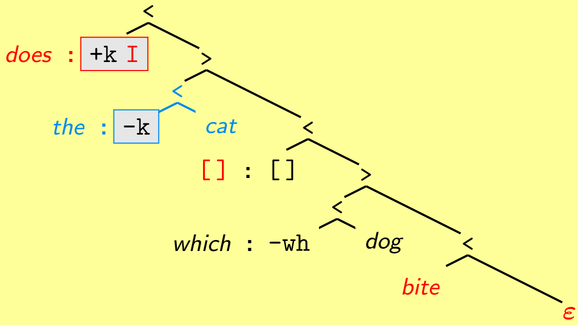
move (reformulated)

move 2:

$$\frac{\left\langle \boxed{\texttt{:}\,,\texttt{+f}\,\gamma}\,,\gamma_1,\ldots,\gamma_{j-1},\boxed{\texttt{-f}}\,,\gamma_{j+1},\ldots,\gamma_k \right\rangle \left(\boxed{x_0}\,,x_1,\ldots,x_{j-1},\boxed{x_j}\,,x_{j+1},\ldots,x_k \right)}{\left\langle \boxed{\texttt{:}\,,\cancel{\texttt{+f}}\,\gamma}\,,\gamma_1,\ldots,\gamma_{j-1}\quad,\quad\gamma_{j+1},\ldots,\gamma_k \right\rangle \left(\boxed{x_j}\,\boxed{x_0}\,,x_1,\ldots,x_{j-1}\quad,\quad x_{j+1},\ldots,x_k \right)}$$

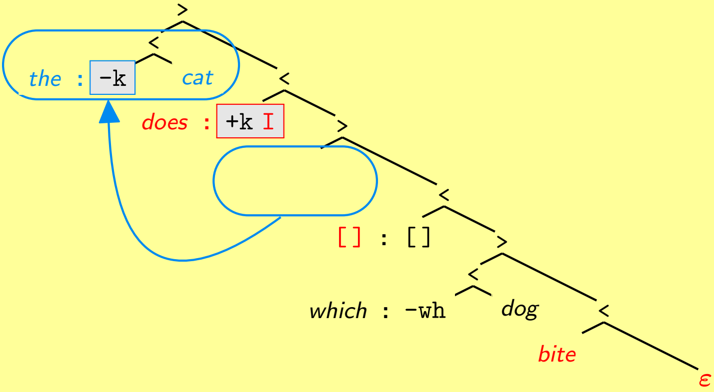
move 2:

$$\frac{\left\langle \boxed{\mathtt{:}\,,\mathtt{+f}\,\gamma}, \gamma_1, \ldots, \gamma_{j-1}, \boxed{\mathtt{-f}}, \gamma_{j+1}, \ldots, \gamma_k \right\rangle \left(\boxed{x_0}, x_1, \ldots, x_{j-1}, \boxed{x_j}, x_{j+1}, \ldots, x_k \right)}{\left\langle \boxed{\mathtt{:}\,,\not{\mathtt{+f}}\,\gamma}, \gamma_1, \ldots, \gamma_{j-1} \quad , \quad \gamma_{j+1}, \ldots, \gamma_k \right\rangle \left(\boxed{x_j}\boxed{x_0}, x_1, \ldots, x_{j-1} \quad , \quad x_{j+1}, \ldots, x_k \right)}$$



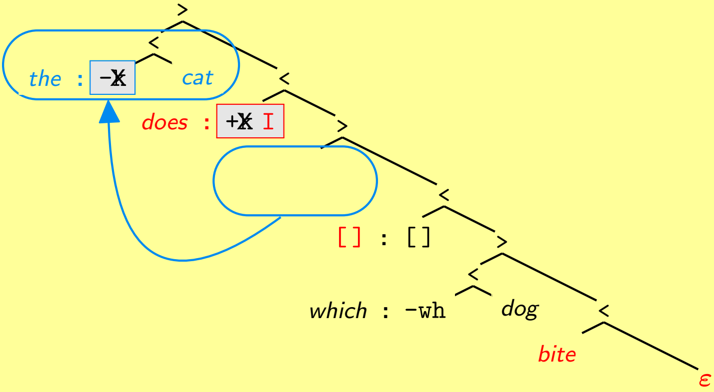
move 2:

$$\frac{\langle\ \boxed{:\ ,\ \texttt{+f}\ \gamma}\ ,\gamma_1,\ldots,\gamma_{j-1},\ \boxed{\texttt{-f}}\ ,\gamma_{j+1},\ldots,\gamma_k\ \rangle\ (\ \boxed{x_0}\ ,x_1,\ldots,x_{j-1},\ \boxed{x_j}\ ,x_{j+1},\ldots,x_k\ )}{\langle\ \boxed{:\ ,\ \not{\texttt{+}}\texttt{f}\ \gamma}\ ,\gamma_1,\ldots,\gamma_{j-1}\ ,\ \gamma_{j+1},\ldots,\gamma_k\ \rangle\ (\ \boxed{x_j}\ \boxed{x_0}\ ,x_1,\ldots,x_{j-1}\ ,\ x_{j+1},\ldots,x_k\ )}$$



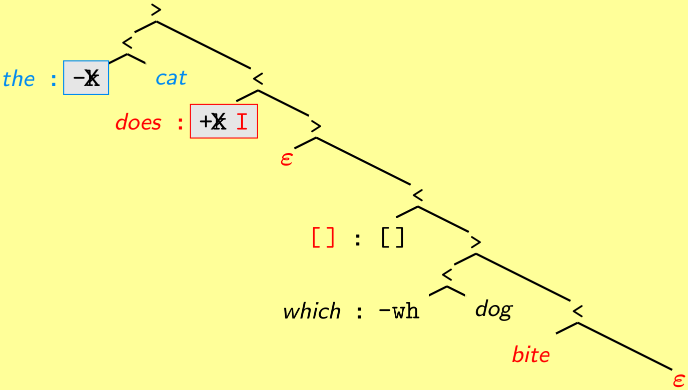
move 2:

$$\frac{\langle\, \texttt{:}\, , \texttt{+f}\ \gamma\, , \gamma_1 , \ldots , \gamma_{j-1} , \texttt{-f}\, , \gamma_{j+1} , \ldots , \gamma_k \,\rangle\, (\, x_0 , x_1 , \ldots , x_{j-1} , x_j , x_{j+1} , \ldots , x_k \,)}{\langle\, \texttt{:}\, , \not{\texttt{+f}}\ \gamma\, , \gamma_1 , \ldots , \gamma_{j-1} \quad , \quad \gamma_{j+1} , \ldots , \gamma_k \,\rangle\, (\, x_j\ x_0 , x_1 , \ldots , x_{j-1} \quad , \quad x_{j+1} , \ldots , x_k \,)}$$



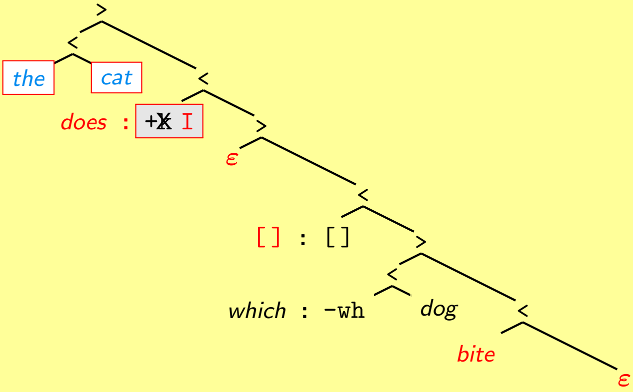
move 2:

$$\frac{\left\langle \boxed{\mathtt{:}\,,\mathtt{+f}\,\gamma}\,,\gamma_1\,,\ldots,\gamma_{j-1}\,,\boxed{\mathtt{-f}}\,,\gamma_{j+1}\,,\ldots,\gamma_k \right\rangle \left(\boxed{x_0}\,,x_1\,,\ldots,x_{j-1}\,,\boxed{x_j}\,,x_{j+1}\,,\ldots,x_k \right)}{\left\langle \boxed{\mathtt{:}\,,\not{\mathtt{+}}\mathtt{f}\,\gamma}\,,\gamma_1\,,\ldots,\gamma_{j-1}\quad,\quad\gamma_{j+1}\,,\ldots,\gamma_k \right\rangle \left(\boxed{x_j}\,\boxed{x_0}\,,x_1\,,\ldots,x_{j-1}\quad,\quad x_{j+1}\,,\ldots,x_k \right)}$$



move 2:

$$\frac{\langle\, :, +\mathtt{f}\ \gamma, \gamma_1, \ldots, \gamma_{j-1}, -\mathtt{f}, \gamma_{j+1}, \ldots, \gamma_k \,\rangle\,(x_0, x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_k)}{\langle\, :, \cancel{+\mathtt{f}}\ \gamma, \gamma_1, \ldots, \gamma_{j-1}, \gamma_{j+1}, \ldots, \gamma_k \,\rangle\,(x_j\ x_0, x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_k)}$$



$G = \langle$ Vocabulary, SynFeat, Lex, $\mathbf{\Omega}$, c $\rangle$ an MG

A minimal expression $\boldsymbol{\tau} \in$ Closure(G) is relevant $:\Longleftrightarrow$

for each $-\mathtt{x} \in$ Licensees,

there is at most one maximal projection in $\boldsymbol{\tau}$ that displays $-\mathtt{x}$.

Trees as tuples (towards a finite partition)

$G = \langle \text{Vocabulary}, \text{SynFeat}, \text{Lex}, \boldsymbol{\Omega}, c \rangle$ an MG

A minimal expression $\boldsymbol{\tau} \in \text{Closure}(G)$ is relevant $:\Longleftrightarrow$

for each $-\mathtt{x} \in$ Licensees,

there is at most one maximal projection in $\boldsymbol{\tau}$ that displays $-\mathtt{x}$.

- In fact — due to the SMC — this kind of structure is characteristic of each expression in Closure(G) involved in creating a complete expression.

Trees as tuples (towards a finite partition)

For relevant $\tau \in \text{Closure}(G)$ consider its tuple representation

$$\langle \odot , \gamma_0 , \gamma_1 , \ldots , \gamma_k \rangle (x_0 , x_1 , \ldots , x_k)$$

Trees as tuples (towards a finite partition)

For relevant $\tau \in \text{Closure}(G)$ consider its tuple representation

$$\langle \odot , \gamma_0 , \gamma_1 , \ldots , \gamma_k \rangle (x_0 , x_1 , \ldots , x_k)$$

- Recall : there are only finitely many possibilities for γ_i . $x :: \lambda \gamma_i \in \text{Lex}$

Trees as tuples (towards a finite partition)

For relevant $\tau \in \text{Closure}(G)$ consider its tuple representation

$$\langle \odot , \gamma_0 , \gamma_1 , \ldots , \gamma_k \rangle (x_0 , x_1 , \ldots , x_k)$$

- Recall : there are only finitely many possibilities for γ_i . $x :: \lambda \gamma_i \in \text{Lex}$

- The relevance of τ additionally implies $k \leq |\text{Licensees}|$.

Trees as tuples (towards a finite partition)

For relevant $\tau \in \text{Closure}(G)$ consider its tuple representation

$$\langle \odot, \gamma_0, \gamma_1, \ldots, \gamma_k \rangle (x_0, x_1, \ldots, x_k)$$

- Recall : there are only finitely many possibilities for γ_i . $x :: \lambda \gamma_i \in \text{Lex}$

- The relevance of τ additionally implies $k \leq |\text{Licensees}|$. Thus, for

$$\langle \odot, \gamma_0, \gamma_1, \ldots, \gamma_k \rangle$$

there are only finitely many possibilities since τ is relevant

Trees as tuples (towards a finite partition)

For relevant $\tau \in \mathrm{Closure}(G)$ consider its tuple representation

$$\langle \odot , \gamma_0 , \gamma_1 , \ldots , \gamma_k \rangle (x_0 , x_1 , \ldots , x_k)$$

- Recall : there are only finitely many possibilities for γ_i . $x :: \lambda \gamma_i \in \mathrm{Lex}$

- The relevance of τ additionally implies $k \leq |\,\mathrm{Licensees}\,|$. Thus, for

$$\langle \odot , \gamma_0 , \gamma_1 , \ldots , \gamma_k \rangle$$

there are only finitely many possibilities since τ is relevant

- Each such k+2-tuple constitutes a nonterminal of the equivalent MCFG .

Equivalent MCFG (in terms of its rules)

merge 1:

$$\frac{\langle \odot, \texttt{=f}\,\gamma, \gamma_1, \ldots, \gamma_k \rangle\ (x_0, x_1, \ldots, x_k) \qquad \langle \bullet, \texttt{f}\,\delta, \delta_1, \ldots, \delta_l \rangle\ (y_0, y_1, \ldots, y_l)}{\langle :, \gamma, \gamma_1, \ldots, \gamma_k, \delta, \delta_1, \ldots, \delta_l \rangle\ (x_0, x_1, \ldots, x_k, y_0, y_1, \ldots, y_l)}$$

merge 2:

$$\frac{\langle ::, \texttt{=f}\,\gamma, \gamma_1, \ldots, \gamma_k \rangle\ (x_0, x_1, \ldots, x_k) \qquad \langle \bullet, \texttt{f}\ , \delta_1, \ldots, \delta_l \rangle\ (y_0, y_1, \ldots, y_l)}{\langle :, \gamma, \gamma_1, \ldots, \gamma_k, \delta_1, \ldots, \delta_l \rangle\ (x_0\, y_0, , x_1, \ldots, x_k, y_1, \ldots, y_l)}$$

merge 3:

$$\frac{\langle :, \texttt{=f}\,\gamma, \gamma_1, \ldots, \gamma_k \rangle\ (x_0, x_1, \ldots, x_k) \qquad \langle \bullet, \texttt{f}\ , \delta_1, \ldots, \delta_l \rangle\ (y_0, y_1, \ldots, y_l)}{\langle :, \gamma, \gamma_1, \ldots, \gamma_k, \delta_1, \ldots, \delta_l \rangle\ (y_0\, x_0, , x_1, \ldots, x_k, y_1, \ldots, y_l)}$$

Equivalent MCFG (in terms of its rules)

merge 1:

$$\frac{\langle \odot, \texttt{=f}\,\gamma, \gamma_1, \ldots, \gamma_k \rangle (x_0, x_1, \ldots, x_k) \qquad \langle \bullet, \texttt{f}\,\delta, \delta_1, \ldots, \delta_l \rangle (y_0, y_1, \ldots, y_l)}{\langle :, \gamma, \gamma_1, \ldots, \gamma_k, \delta, \delta_1, \ldots, \delta_l \rangle (x_0, x_1, \ldots, x_k, y_0, y_1, \ldots, y_l)}$$

merge 2:

$$\frac{\langle ::, \texttt{=f}\,\gamma, \gamma_1, \ldots, \gamma_k \rangle (x_0, x_1, \ldots, x_k) \qquad \langle \bullet, \texttt{f}\;, \delta_1, \ldots, \delta_l \rangle (y_0, y_1, \ldots, y_l)}{\langle :, \gamma, \gamma_1, \ldots, \gamma_k, \delta_1, \ldots, \delta_l \rangle (x_0\, y_0, , x_1, \ldots, x_k, y_1, \ldots, y_l)}$$

merge 3:

$$\frac{\langle :, \texttt{=f}\,\gamma, \gamma_1, \ldots, \gamma_k \rangle (x_0, x_1, \ldots, x_k) \qquad \langle \bullet, \texttt{f}\;, \delta_1, \ldots, \delta_l \rangle (y_0, y_1, \ldots, y_l)}{\langle :, \gamma, \gamma_1, \ldots, \gamma_k, \delta_1, \ldots, \delta_l \rangle (y_0\, x_0, , x_1, \ldots, x_k, y_1, \ldots, y_l)}$$

Equivalent MCFG

(in terms of its rules)

merge 1:

$$\frac{B(x_0, x_1, \ldots, x_k) \qquad C(y_0, y_1, \ldots, y_l)}{A(x_0, x_1, \ldots, x_k, y_0, y_1, \ldots, y_l)}$$

merge 2:

$$\frac{B(x_0, x_1, \ldots, x_k) \qquad C(y_0, y_1, \ldots, y_l)}{A(x_0 y_0, , x_1, \ldots, x_k, y_1, \ldots, y_l)}$$

merge 3:

$$\frac{B(x_0, x_1, \ldots, x_k) \qquad C(y_0, y_1, \ldots, y_l)}{A(y_0 x_0, , x_1, \ldots, x_k, y_1, \ldots, y_l)}$$

Equivalent MCFG

(in terms of its rules)

move 1:

$$\frac{\langle\ \mathtt{:}\ , \mathtt{+f}\ \gamma, \gamma_1, \ldots, \gamma_{j-1}, \mathtt{-f}\ \delta, \gamma_{j+1}, \ldots, \gamma_k \rangle\ (x_0, x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_k)}{\langle\ \mathtt{:}\ , \gamma, \gamma_1, \ldots, \gamma_{j-1}, \delta, \gamma_{j+1}, \ldots, \gamma_k \rangle\ (x_0, x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_k)}$$

move 2:

$$\frac{\langle\ \mathtt{:}\ , \mathtt{+f}\ \gamma, \gamma_1, \ldots, \gamma_{j-1}, \mathtt{-f}\ , \gamma_{j+1}, \ldots, \gamma_k \rangle\ (x_0, x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_k)}{\langle\ \mathtt{:}\ , \gamma, \gamma_1, \ldots, \gamma_{j-1}, \gamma_{j+1}, \ldots, \gamma_k \rangle\ (x_j\, x_0, x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_k)}$$

Equivalent MCFG (in terms of its rules)

move 1:

$$\frac{\langle \, \mathbf{:} \, , \texttt{+f} \, \gamma \, , \gamma_1 \, , \ldots \, , \gamma_{j-1} \, , \texttt{-f} \, \delta \, , \gamma_{j+1} \, , \ldots \, , \gamma_k \, \rangle \; (x_0 \, , x_1 \, , \ldots \, , x_{j-1} \, , x_j \, , x_{j+1} \, , \ldots \, , x_k \,)}{\langle \, \mathbf{:} \, , \gamma \, , \gamma_1 \, , \ldots \, , \gamma_{j-1} \, , \delta \, , \gamma_{j+1} \, , \ldots \, , \gamma_k \, \rangle \; (x_0 \, , x_1 \, , \ldots \, , x_{j-1} \, , x_j \, , x_{j+1} \, , \ldots \, , x_k \,)}$$

move 2:

$$\frac{\langle \, \mathbf{:} \, , \texttt{+f} \, \gamma \, , \gamma_1 \, , \ldots \, , \gamma_{j-1} \, , \texttt{-f} \quad , \gamma_{j+1} \, , \ldots \, , \gamma_k \, \rangle \; (x_0 \, , x_1 \, , \ldots \, , x_{j-1} \, , x_j \, , x_{j+1} \, , \ldots \, , x_k \,)}{\langle \, \mathbf{:} \, , \gamma \, , \gamma_1 \, , \ldots \, , \gamma_{j-1} \, , \gamma_{j+1} \, , \ldots \, , \gamma_k \, \rangle \; (x_j \, x_0 \, , x_1 \, , \ldots \, , x_{j-1} \, , x_{j+1} \, , \ldots \, , x_k \,)}$$

Equivalent MCFG

move 1:

$$\frac{B(x_0, x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_k)}{A(x_0, x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_k)}$$

move 2:

$$\frac{B(x_0, x_1, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_k)}{A(x_j\, x_0, x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_k)}$$

Equivalent MCFG (in terms of its rules)

lexical insertion:

$$\frac{}{\langle \mathbf{::}, \boldsymbol{\alpha} \rangle \;\; (\boldsymbol{\pi})}$$

for $\boldsymbol{\pi} \mathbf{::} \boldsymbol{\alpha} \in \mathsf{Lex}$

Equivalent MCFG

(in terms of its rules)

lexical insertion:

$$\frac{}{\boxed{\langle\, ::\,, \alpha \,\rangle}\,(\,\pi\,)}$$

for $\pi :: \alpha \in \text{Lex}$

Equivalent MCFG (in terms of its rules)

lexical insertion:

$$\frac{}{\boxed{\mathsf{A}}\ (\pi)}$$

for $\pi :: \alpha \in$ Lex

Remarks: MG –> MCFG

- Feature consumption plus SMC are the crucial ingredients.
- Proof is more than a proof of just an embedding of string language classes.
- Adaption is possible, when head movement, left complement selection, rightward movement/extraposition and/or covert movement/agree is incorporated into the MG-formalism.
- Adaption is also possible, when late adjunction together with adjunct island condition is incorporated into the MG-formalism. This, in fact, is "more strictly" about string language equivalence.
- Adding SPIC, yields monadic branching MCFGs as output. Note that the set of relevant trees can be reduced in this case.

Multiple Context-Free Grammars (Seki et al. 1991)

$G = \langle N, \Sigma, P, S\rangle$ an MCFG

- N a finite set of nonterminals , a ranked alphabet
- S the start symbol , nonterminal of rank 1
- Σ a finite set of terminals
- P a finite set of rules :

$$A \leftarrow B_1 \quad , \ldots , B_m$$

$G = \langle N, \Sigma, P, S \rangle$ an MCFG

- N a finite set of nonterminals , a ranked alphabet
- S the start symbol , nonterminal of rank 1
- Σ a finite set of terminals
- P a finite set of rules :

$$A(t_1, \ldots, t_k) \leftarrow B_1(x_{1,1}, \ldots, x_{1,k_1}), \ldots, B_m(x_{m,1}, \ldots, x_{m,k_m})$$

Multiple Context-Free Grammars (Seki et al. 1991)

$G = \langle N, \Sigma, P, S \rangle$ an MCFG

- N a finite set of nonterminals , a ranked alphabet
- S the start symbol , nonterminal of rank 1
- Σ a finite set of terminals
- P a finite set of rules :

$$A(t_1, \ldots, t_k) \leftarrow B_1(x_{1,1}, \ldots, x_{1,k_1}), \ldots, B_m(x_{m,1}, \ldots, x_{m,k_m})$$

- $\text{rank}(A) = k$, $\text{rank}(B_i) = k_i$
- $x_{1,1}, \ldots, x_{m,k_m}$ variables
- $t_j \in (\Sigma \cup \{x_{1,1}, \ldots, x_{m,k_m}\})^*$
- $x_{i,j}$ occurs at most once in $t_1 \cdots t_k$

Multiple Context-Free Grammars (Seki et al. 1991)

$G = \langle N, \Sigma, P, S \rangle$ an MCFG

- N a finite set of nonterminals , a ranked alphabet
- S the start symbol , nonterminal of rank 1
- Σ a finite set of terminals
- P a finite set of rules :

$A(t_1, \ldots, t_k) \leftarrow$ terminating rule if $m=0$

- $\text{rank}(A) = k$, $\text{rank}(B_i) = k_i$
- $x_{1,1}, \ldots, x_{m,k_m}$ variables
- $t_j \in (\Sigma \cup \{x_{1,1}, \ldots, x_{m,k_m}\})^*$
- $x_{i,j}$ occurs at most once in $t_1 \cdots t_k$

Multiple Context-Free Grammars (Seki et al. 1991)

$G = \langle N, \Sigma, P, S \rangle$ an MCFG

- N a finite set of nonterminals , a ranked alphabet
- S the start symbol , nonterminal of rank 1
- Σ a finite set of terminals
- P a finite set of rules :

Example:

$$S(x_2\, y_2\, x_1\, y_1) \leftarrow B(x_1, x_2), C(y_1, y_2)$$
$$B(a\, x_1\, b, c\, x_2\, d) \leftarrow B(x_1, x_2)$$
$$C(a\, y_1\, b, c\, y_2\, d) \leftarrow C(y_1, y_2)$$
$$B(a\, b, c\, d) \leftarrow$$
$$C(a\, b, c\, d) \leftarrow$$

Non-permuting MCFGs

$G = \langle N, \Sigma, P, S \rangle$ an MCFG$_{\text{non-perm}}$

- N a finite set of nonterminals , a ranked alphabet
- S the start symbol , nonterminal of rank 1
- Σ a finite set of terminals
- P a finite set of rules :

$$A(t_1, \ldots, t_k) \leftarrow B_1(x_{1,1}, \ldots, x_{1,k_1}), \ldots, B_m(x_{m,1}, \ldots, x_{m,k_m})$$

Non-permuting MCFGs

$G = \langle N, \Sigma, P, S \rangle$ an $\mathrm{MCFG}_{\text{non-perm}}$

- N a finite set of nonterminals , a ranked alphabet
- S the start symbol , nonterminal of rank 1
- Σ a finite set of terminals
- P a finite set of rules :

$$A(t_1, \ldots, t_k) \leftarrow B_1(x_{1,1}, \ldots, x_{1,k_1}), \ldots, B_m(x_{m,1}, \ldots, x_{m,k_m})$$

+ do not permute variables from one nonterminal within $t_1 \cdots t_k$:
  the order of variables from one nonterminal component is preserved.

Non-permuting MCFGs

$G = \langle N, \Sigma, P, S \rangle$ an MCFG$_{\text{non-perm}}$

- N a finite set of nonterminals , a ranked alphabet
- S the start symbol , nonterminal of rank 1
- Σ a finite set of terminals
- P a finite set of rules :

Non-example: $S(x_2\, y_2\, x_1\, y_1) \leftarrow B(x_1, x_2), C(y_1, y_2)$

Non-permuting MCFGs

$G = \langle N, \Sigma, P, S \rangle$ an $\text{MCFG}_{\text{non-perm}}$

- N a finite set of nonterminals , a ranked alphabet
- S the start symbol , nonterminal of rank 1
- Σ a finite set of terminals
- P a finite set of rules :

Example: $S(x_1\, y_1\, y_2\, x_2) \leftarrow B(x_1, x_2), C(y_1, y_2)$

Dimension and rank of MCFGs

$G = \langle N, \Sigma, P, S \rangle$ an MCFG

- rank of G: maximal number of nonterminal instances on the righthand side of some rule
- G has rank f $\rightsquigarrow$ G is an MCFG(f)
- The language derived by G is an MCFL, resp., an MCFL(f)

MCFG-normal form

- MCFG(2) constitutes a normal form for MCFG, since we have

Proposition: MCFL = MCFL(2)

MCFG-normal form

- MCFG(2) constitutes a normal form for MCFG, since we have

Proposition: MCFL = MCFL(2)

- A restricted MCFG-normal form is, thus, the following:

An MCFG, G, is an MCFG_{mb}, or, monadic branching if

- G is of rank 2, and
- each binary rule is of the form:

  $A(t_1, \ldots, t_k) \leftarrow B(x), C(y_1, \ldots, y_n)$

$$A(t_1, \ldots, t_k) \leftarrow B_1(x_{1,1}, \ldots, x_{1,k_1}), \ldots, B_m(x_{m,1}, \ldots, x_{m,k_m})$$

- $t_j \in (\Sigma \cup \{x_{1,1}, \ldots, x_{m,k_m}\})^*$
- $x_{i,j}$ occurs at most once in $t_1 \cdots t_k$

In addition

$A(t_1, \ldots, t_k) \leftarrow B_1(x_{1,1}, \ldots, x_{1,k_1}), \ldots, B_m(x_{m,1}, \ldots, x_{m,k_m})$

- $t_j \in (\Sigma \cup \{x_{1,1}, \ldots, x_{m,k_m}\})^*$
- $x_{i,j}$ occurs at most once in $t_1 \cdots t_k$

In addition

- non-deleting: $x_{i,j}$ does occur in $t_1 \cdots t_k$

MCFG-normal form

$A(t_1, \ldots, t_k) \leftarrow B_1(x_{1,1}, \ldots, x_{1,k_1}), \ldots, B_m(x_{m,1}, \ldots, x_{m,k_m})$

- $t_j \in (\Sigma \cup \{x_{1,1}, \ldots, x_{m,k_m}\})^*$
- $x_{i,j}$ occurs at most once in $t_1 \cdots t_k$

In addition

- non-deleting: $x_{i,j}$ does occur in $t_1 \cdots t_k$
- non-permuting: $x_{i,j}$ precedes $x_{i,j'}$ in $t_1 \cdots t_k$ for $j < j'$

  i.e., order of variables from one nonterminal component is preserved

$A(t_1, \ldots, t_k) \leftarrow B_1(x_{1,1}, \ldots, x_{1,k_1}), \ldots, B_m(x_{m,1}, \ldots, x_{m,k_m})$

- $t_j \in (\Sigma \cup \{x_{1,1}, \ldots, x_{m,k_m}\})^*$
- $x_{i,j}$ occurs at most once in $t_1 \cdots t_k$

In addition

- non-deleting: $x_{i,j}$ does occur in $t_1 \cdots t_k$
- non-permuting: $x_{i,j}$ precedes $x_{i,j'}$ in $t_1 \cdots t_k$ for $j < j'$

  i.e., order of variables from one nonterminal component is preserved
- strictly non-terminating: $m \geq 1 \rightsquigarrow t_j \in \{x_{1,1}, \ldots, x_{m,k_m}\}^*$

MCFG-normal form

$A(t_1, \ldots, t_k) \leftarrow B_1(x_{1,1}, \ldots, x_{1,k_1}), \ldots, B_m(x_{m,1}, \ldots, x_{m,k_m})$

- $t_j \in (\Sigma \cup \{x_{1,1}, \ldots, x_{m,k_m}\})^*$
- $x_{i,j}$ occurs at most once in $t_1 \cdots t_k$

In addition

- non-deleting: $x_{i,j}$ does occur in $t_1 \cdots t_k$
- non-permuting: $x_{i,j}$ precedes $x_{i,j'}$ in $t_1 \cdots t_k$ for $j < j'$

  i.e., order of variables from one nonterminal component is preserved

- strictly non-terminating: $m \geq 1 \rightsquigarrow t_j \in \{x_{1,1}, \ldots, x_{m,k_m}\}^*$

  or simple terminating: $m = 0 \rightsquigarrow \text{rank}(A) = 1$ and $t_1 \in \Sigma \cup \{\varepsilon\}$

$$A(t_1, \ldots, t_k) \leftarrow B_1(x_{1,1}, \ldots, x_{1,k_1}), \ldots, B_m(x_{m,1}, \ldots, x_{m,k_m})$$

- $t_j \in (\Sigma \cup \{x_{1,1}, \ldots, x_{m,k_m}\})^*$
- $x_{i,j}$ occurs at most once in $t_1 \cdots t_k$

In addition

- non-deleting: $x_{i,j}$ does occur in $t_1 \cdots t_k$
- non-permuting: $x_{i,j}$ precedes $x_{i,j'}$ in $t_1 \cdots t_k$ for $j < j'$
  i.e., order of variables from one nonterminal component is preserved
- strictly non-terminating: $m \geq 1 \rightsquigarrow t_j \in \{x_{1,1}, \ldots, x_{m,k_m}\}^*$
  or simple terminating: $m = 0 \rightsquigarrow \text{rank}(A) = 1$ and $t_1 \in \Sigma \cup \{\varepsilon\}$
- doublet-free: $A, B_1, \ldots, B_m$ are pairwise distinct

MCFG(2)-normal form –> MG-normal form

- cf. Harkema 2001, Michaelis 2001c, Michaelis 2004

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{\ \mathtt{A}_i \mid A \in N, 1 \leq i \leq \text{rank}(A)+1 \} \cup \{\mathtt{c}\}$
- Licensees $= \{\mathtt{-A}_i \mid A \in N, 1 \leq i \leq \text{rank}(A) \}$
- Vocabulary $= \Sigma$

Equivalent MG

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{\ \mathtt{A}_i \mid A \in N, 1 \leq i \leq \text{rank}(A)+1\} \cup \{\mathtt{c}\}$
- Licensees $= \{\mathtt{-A}_i \mid A \in N, 1 \leq i \leq \text{rank}(A)\}$
- Vocabulary $= \Sigma$ $\qquad$ $\mathtt{c} \in$ Selectees is the distinguished category

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{\ \mathtt{A}_i \,|\, A \in N, 1 \leq i \leq \mathrm{rank}(A)+1\} \cup \{\mathtt{c}\}$
- Licensees $= \{\mathtt{-A}_i \,|\, A \in N, 1 \leq i \leq \mathrm{rank}(A)\}$
- Vocabulary $= \Sigma$ $\quad$ $\mathtt{c} \in$ Selectees is the distinguished category
- Defining the MG-lexicon:

Equivalent MG

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \ \mathtt{A}_i \,|\, A \in N, 1 \leq i \leq \mathrm{rank}(A)+1 \} \cup \{ \mathrm{c} \}$
- Licensees $= \{ \mathtt{-A}_i \,|\, A \in N, 1 \leq i \leq \mathrm{rank}(A) \}$
- Vocabulary $= \Sigma$ $\quad$ $\mathrm{c} \in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$$\boxed{A(t_1, \ldots, t_i, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)}$$

Equivalent MG

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \ \mathtt{A}_i \,|\, A \in N, 1 \leq i \leq \text{rank}(A)+1 \} \cup \{ c \}$
- Licensees $= \{ \mathtt{-A}_i \,|\, A \in N, 1 \leq i \leq \text{rank}(A) \}$
- Vocabulary $= \Sigma$ $\quad$ $c \in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$$\boxed{A(t_1, \ldots, t_i, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)}$$

$t_i = z_{i,1}, z_{i,1} \cdots z_{i,n(i)}$ with $z_{i,j} \in \{x_1, \ldots, x_l, y_1, \ldots, y_m\}$

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{\ \mathtt{A}_i \mid A \in N, 1 \leq i \leq \mathrm{rank}(A)+1 \} \cup \{ \mathtt{c} \}$
- Licensees $= \{ \mathtt{-A}_i \mid A \in N, 1 \leq i \leq \mathrm{rank}(A) \}$
- Vocabulary $= \Sigma$ $\quad$ $\mathtt{c} \in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$$A(t_1, \ldots, t_i, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)$$

$t_i = z_{i,1}, z_{i,1} \cdots z_{i,n(i)}$ with $z_{i,j} \in \{x_1, \ldots, x_l, y_1, \ldots, y_m\}$

$\varepsilon :: \mathtt{=C}_1\ \mathtt{=B}_1\ \mathtt{A}_{k+1}$ $\quad$ start calculating A , select B and C

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \ \mathtt{A_i} \mid A \in N, 1 \leq i \leq \text{rank}(A)+1 \} \cup \{ c \}$
- Licensees $= \{ \mathtt{-A_i} \mid A \in N, 1 \leq i \leq \text{rank}(A) \}$
- Vocabulary $= \Sigma$ $\quad$ $c \in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$$A(t_1, \ldots, t_i, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)$$

$t_i = z_{i,1}, z_{i,1} \cdots z_{i,n(i)}$ with $z_{i,j} \in \{x_1, \ldots, x_l, y_1, \ldots, y_m\}$

ε :: $\mathtt{=C_1 \ =B_1 \ A_{k+1}}$ $\quad$ start calculating A , select B and C

ε :: $\mathtt{=A_{i+1}} \quad \mathtt{A_i \ -A_i}$ $\quad$ i-th component of A , $i = k, \ldots, 1$

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \; \mathtt{A}_i \,|\, A \in N, 1 \leq i \leq \text{rank}(A)+1 \} \cup \{ c \}$
- Licensees $= \{ \mathtt{-A}_i \,|\, A \in N, 1 \leq i \leq \text{rank}(A) \}$
- Vocabulary $= \Sigma$ $\quad$ $c \in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$$A(t_1, \ldots, t_i, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)$$

$t_i = z_{i,1}, z_{i,1} \cdots z_{i,n(i)}$ with $z_{i,j} \in \{x_1, \ldots, x_l, y_1, \ldots, y_m\}$

$\varepsilon :: \mathtt{=C}_1 \ \mathtt{=B}_1 \ \mathtt{A}_{k+1}$ start calculating A , select B and C

$\varepsilon :: \mathtt{=A}_{i+1} \ \mathtt{+L}_{i,n(i)} \ \ldots \ \mathtt{+L}_{i,2} \ \mathtt{+L}_{i,1} \ \mathtt{A}_i \ \mathtt{-A}_i$ i-th component of A , $i = k, \ldots, 1$

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \ \mathtt{A}_i \mid A \in N, 1 \leq i \leq \mathrm{rank}(A)+1 \} \cup \{ c \}$
- Licensees $= \{ \mathtt{-A}_i \mid A \in N, 1 \leq i \leq \mathrm{rank}(A) \}$
- Vocabulary $= \Sigma$ $\quad$ $c \in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$$A(t_1, \ldots, t_i, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)$$

$t_i =$ $\quad$ with $z_{i,j} \in \{x_1, \ldots, x_l, y_1, \ldots, y_m\}$

$\varepsilon :: \mathtt{=C}_1\ \mathtt{=B}_1\ \mathtt{A}_{k+1}$ $\quad$ start calculating A, select B and C

$\varepsilon :: \mathtt{=A}_{i+1}\ \mathtt{+L}_{i,n(i)}\ \ldots\ \mathtt{+L}_{i,2}\ \mathtt{+L}_{i,1}\ \mathtt{A}_i\ \mathtt{-A}_i$ $\quad$ i-th component of A, $i = k, \ldots, 1$

Equivalent MG

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \mathtt{A}_i \mid A \in N, 1 \leq i \leq \text{rank}(A)+1 \} \cup \{ c \}$
- Licensees $= \{ \mathtt{-A}_i \mid A \in N, 1 \leq i \leq \text{rank}(A) \}$
- Vocabulary $= \Sigma$ $c \in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$$A(t_1, \ldots, t_i, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)$$

$t_i = z_{i,1}, z_{i,1} \cdots z_{i,n(i)}$ with $z_{i,j} \in \{x_1, \ldots, x_l, y_1, \ldots, y_m\}$

$\varepsilon :: \mathtt{=C}_1\ \mathtt{=B}_1\ \mathtt{A}_{k+1}$ start calculating A , select B and C

$\varepsilon :: \mathtt{=A}_{i+1}\ \mathtt{+L}_{i,n(i)} \ \ldots\ \mathtt{+L}_{i,2}\ \mathtt{+L}_{i,1}\ \mathtt{A}_i\ \mathtt{-A}_i$ i-th component of A , $i = k, \ldots, 1$

Equivalent MG

$G = \langle N , \Sigma , P , S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \ \mathtt{A}_i \,|\, A \in N , 1 \leq i \leq \mathrm{rank}(A)+1 \} \cup \{ c \}$
- Licensees $= \{ \mathtt{-A}_i \,|\, A \in N , 1 \leq i \leq \mathrm{rank}(A) \}$
- Vocabulary $= \Sigma$ $\quad$ $c \in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$$A(t_1 , \ldots , t_i , \ldots , t_k) \leftarrow B(x_1 , \ldots , x_l) , C(y_1 , \ldots , y_m)$$

$t_i = z_{i,1} , z_{i,1} \cdots z_{i,n(i)}$ with $z_{i,j} \in \{x_1 , \ldots , x_l , y_1 , \ldots , y_m\}$

$\varepsilon :: \mathtt{=C}_1 \ \mathtt{=B}_1 \ \mathtt{A}_{k+1}$ start calculating A , select B and C

$\varepsilon :: \mathtt{=A}_{i+1} \ \mathtt{+L}_{i,n(i)} \ \ldots \ \mathtt{+L}_{i,2} \ \mathtt{+L}_{i,1} \ \mathtt{A}_i \ \mathtt{-A}_i$ i-th component of A , $i = k , \ldots , 1$

$\mathtt{+L}_{i,j} = \mathtt{+B}_p$ iff $z_{i,j} = x_p$ $\quad$ $\mathtt{+L}_{i,j} = \mathtt{+C}_p$ iff $z_{i,j} = y_p$

Equivalent MG

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \; \mathtt{A}_i \,|\, A \in N, 1 \leq i \leq \text{rank}(A)+1 \} \cup \{ \mathtt{c} \}$
- Licensees $= \{ \mathtt{-A}_i \,|\, A \in N, 1 \leq i \leq \text{rank}(A) \}$
- Vocabulary $= \Sigma$ $\quad$ $\mathtt{c} \in$ Selectees is the distinguished category
- Defining the MG-lexicon:

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \ \mathtt{A}_i \,|\, A \in N, 1 \leq i \leq \text{rank}(A)+1 \} \cup \{ c \}$
- Licensees $= \{ \mathtt{-A}_i \,|\, A \in N, 1 \leq i \leq \text{rank}(A) \}$
- Vocabulary $= \Sigma$ $\quad$ $c \in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$$A(t_1, \ldots, t_i, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l)$$

$t_i = z_{i,1}, z_{i,1} \cdots z_{i,n(i)}$ $\quad$ with $\quad z_{i,j} \in \{x_1, \ldots, x_l, y_1, \ldots, y_m\}$

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{\ \mathtt{A}_i \mid A \in N,\ 1 \leq i \leq \mathrm{rank}(A)+1 \} \cup \{ c \}$
- Licensees $= \{ \mathtt{-A}_i \mid A \in N,\ 1 \leq i \leq \mathrm{rank}(A) \}$
- Vocabulary $= \Sigma$ $\quad$ $c \in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$$A(t_1, \ldots, t_i, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l)$$

$t_i = z_{i,1}, z_{i,1} \cdots z_{i,n(i)}$ with $z_{i,j} \in \{x_1, \ldots, x_l, y_1, \ldots, y_m\}$

$\varepsilon :: \mathtt{=B}_1\ \mathtt{A}_{k+1}$ — start calculating A, select B

$\varepsilon :: \mathtt{=A}_{i+1}\quad \mathtt{A}_i\ \mathtt{-A}_i$ — i-th component of A, $i = k, \ldots, 1$

Equivalent MG

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \mathtt{A}_i \mid A \in N, 1 \leq i \leq \text{rank}(A)+1 \} \cup \{ \mathtt{c} \}$
- Licensees $= \{ \mathtt{-A}_i \mid A \in N, 1 \leq i \leq \text{rank}(A) \}$
- Vocabulary $= \Sigma$ $\quad$ $\mathtt{c} \in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$$A(t_1, \ldots, t_i, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l)$$

$t_i = z_{i,1}, z_{i,1} \cdots z_{i,n(i)}$ with $z_{i,j} \in \{x_1, \ldots, x_l, y_1, \ldots, y_m\}$

$\varepsilon :: \mathtt{=B}_1\ \mathtt{A}_{k+1}$ $\quad$ start calculating A, select B

$\varepsilon :: \mathtt{=A}_{i+1}\ \mathtt{+L}_{i,n(i)} \ldots \mathtt{+L}_{i,2}\ \mathtt{+L}_{i,1}\ \mathtt{A}_i\ \mathtt{-A}_i$ $\quad$ i-th component of A, $i = k, \ldots, 1$

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \ \mathtt{A}_i \,|\, A \in N, 1 \leq i \leq \text{rank}(A)+1 \} \cup \{ \mathtt{c} \}$
- Licensees $= \{ \mathtt{-A}_i \,|\, A \in N, 1 \leq i \leq \text{rank}(A) \}$
- Vocabulary $= \Sigma$ c $\in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$$A(t_1, \ldots, t_i, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l)$$

$t_i = z_{i,1}, z_{i,1} \cdots z_{i,n(i)}$ with $z_{i,j} \in \{x_1, \ldots, x_l, y_1, \ldots, y_m\}$

ε :: $\mathtt{=B}_1 \, \mathtt{A}_{k+1}$ start calculating A , select B

ε :: $\mathtt{=A}_{i+1}$ $\mathtt{A}_i \ \mathtt{-A}_i$ i-th component of A , $i = k, \ldots, 1$

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \mathtt{A_i} \,|\, A \in N, 1 \leq i \leq \text{rank}(A)+1 \} \cup \{ \mathtt{c} \}$
- Licensees $= \{ \mathtt{-A_i} \,|\, A \in N, 1 \leq i \leq \text{rank}(A) \}$
- Vocabulary $= \Sigma$ c $\in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$$A(t_1, \ldots, t_i, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l)$$

$t_i =$ with $z_{i,j} \in \{x_1, \ldots, x_l, y_1, \ldots, y_m\}$

ε :: $\mathtt{=B_1\ A_{k+1}}$ start calculating A , select B

ε :: $\mathtt{=A_{i+1}}$ $+L_{i,n(i)} \ldots +L_{i,2}\ +L_{i,1}$ $\mathtt{A_i\ -A_i}$ i-th component of A , $i = k, \ldots, 1$

Equivalent MG

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \ \mathtt{A}_i \mid A \in N, 1 \leq i \leq \text{rank}(A)+1 \} \cup \{ \mathtt{c} \}$
- Licensees $= \{ \mathtt{-A}_i \mid A \in N, 1 \leq i \leq \text{rank}(A) \}$
- Vocabulary $= \Sigma$ $\quad$ $\mathtt{c} \in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$$A(t_1, \ldots, t_i, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l)$$

$t_i = z_{i,1}, z_{i,1} \cdots z_{i,n(i)}$ with $z_{i,j} \in \{x_1, \ldots, x_l, y_1, \ldots, y_m\}$

$\varepsilon :: \mathtt{=B}_1\ \mathtt{A}_{k+1}$ $\quad$ start calculating A, select B

$\varepsilon :: \mathtt{=A}_{i+1}\ \mathtt{+L}_{i,n(i)} \ \ldots \ \mathtt{+L}_{i,2}\ \mathtt{+L}_{i,1}\ \mathtt{A}_i\ \mathtt{-A}_i$ $\quad$ i-th component of A, $i = k, \ldots, 1$

Equivalent MG

$G = \langle N , \Sigma , P , S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \ \mathtt{A}_i \,|\, A \in N , 1 \leq i \leq \mathrm{rank}(A){+}1 \} \cup \{ \mathtt{c} \}$
- Licensees $= \{ \mathtt{-A}_i \,|\, A \in N , 1 \leq i \leq \mathrm{rank}(A) \}$
- Vocabulary $= \Sigma$ $\quad$ $\mathtt{c} \in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$$A(t_1 , \ldots , t_i , \ldots , t_k) \leftarrow B(x_1 , \ldots , x_l)$$

$t_i = z_{i,1} , z_{i,1} \cdots z_{i,n(i)}$ with $z_{i,j} \in \{x_1 , \ldots , x_l , y_1 , \ldots , y_m\}$

$\varepsilon :: \mathtt{=B}_1 \; \mathtt{A}_{k+1}$ $\quad$ start calculating A , select B

$\varepsilon :: \mathtt{=A}_{i+1} \; \mathtt{+L}_{i,n(i)} \; \ldots \; \mathtt{+L}_{i,2} \; \mathtt{+L}_{i,1} \; \mathtt{A}_i \; \mathtt{-A}_i$ $\quad$ i-th component of A , $i = k , \ldots , 1$

$\mathtt{+L}_{i,j} = \mathtt{+B}_p$ iff $z_{i,j} = x_p$

$G = \langle N, \Sigma, P, S\rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{\ \mathtt{A_i} \mid A \in N, 1 \leq i \leq \mathrm{rank}(A)+1\} \cup \{\mathtt{c}\}$
- Licensees $= \{\mathtt{-A_i} \mid A \in N, 1 \leq i \leq \mathrm{rank}(A)\}$
- Vocabulary $= \Sigma$ $\mathtt{c} \in$ Selectees is the distinguished category
- Defining the MG-lexicon:

Equivalent MG

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \; \mathtt{A}_i \,|\, A \in N, 1 \le i \le \mathrm{rank}(A){+}1 \} \cup \{ c \}$
- Licensees $= \{ \mathtt{-A}_i \,|\, A \in N, 1 \le i \le \mathrm{rank}(A) \}$
- Vocabulary $= \Sigma$ $\quad$ $c \in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$A(w) \leftarrow$ $\qquad$ for some $w \in \Sigma \cup \{\varepsilon\}$

Equivalent MG

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \ \mathtt{A_i} \,|\, A \in N, 1 \leq i \leq \text{rank}(A)+1 \} \cup \{ \mathsf{c} \}$
- Licensees $= \{ \mathtt{-A_i} \,|\, A \in N, 1 \leq i \leq \text{rank}(A) \}$
- Vocabulary $= \Sigma$ $\quad$ $\mathsf{c} \in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$A(w) \leftarrow$ $\quad$ for some $w \in \Sigma \cup \{\varepsilon\}$

$w :: \mathtt{A_1} \ \mathtt{-A_1}$

"lexical insertion"

$G = \langle N, \Sigma, P, S\rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{\ \mathtt{A}_i \,|\, A \in N,\, 1 \leq i \leq \text{rank}(A)+1\,\} \cup \{\,c\,\}$
- Licensees $= \{\,\mathtt{-A}_i \,|\, A \in N,\, 1 \leq i \leq \text{rank}(A)\,\}$
- Vocabulary $= \Sigma$ c $\in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$A(w) \leftarrow$ for some $w \in \Sigma \cup \{\varepsilon\}$

$w :: \mathtt{A}_1\ \mathtt{-A}_1$ "lexical insertion"

- Defining the MG-lexicon: Additionally

Equivalent MG

$G = \langle N, \Sigma, P, S \rangle$ an MCFG(2) in corresponding normal form

- Selectees $= \{ \ \mathtt{A}_i \,|\, A \in N,\ 1 \leq i \leq \text{rank}(A)+1 \} \cup \{ c \}$
- Licensees $= \{ \mathtt{-A}_i \,|\, A \in N,\ 1 \leq i \leq \text{rank}(A) \}$
- Vocabulary $= \Sigma$ $\quad$ $c \in$ Selectees is the distinguished category
- Defining the MG-lexicon: Consider

$A(w) \leftarrow$ $\quad$ for some $w \in \Sigma \cup \{ \varepsilon \}$

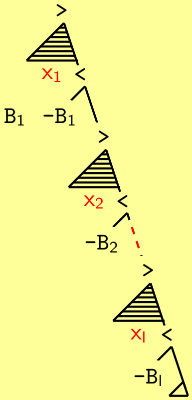
$w :: \mathtt{A}_1\ \mathtt{-A}_1$ $\quad$ "lexical insertion"

- Defining the MG-lexicon: Additionally

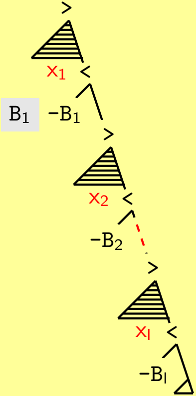
$\varepsilon :: \mathtt{=S}_1\ \mathtt{+S}_1\ c$ $\quad$ "completor", MCFG-rule independent

$A(t_1, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)$

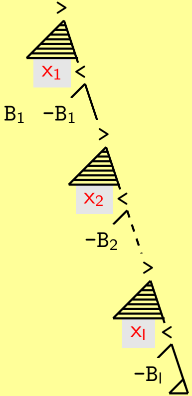
$A(t_1, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)$



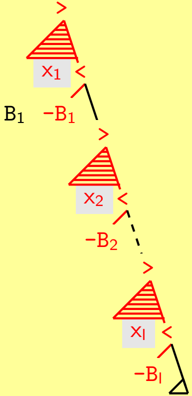
$A(t_1, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)$



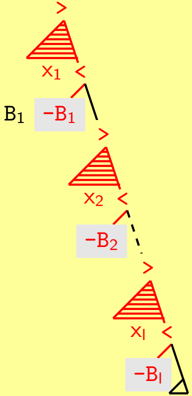
$A(t_1, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)$



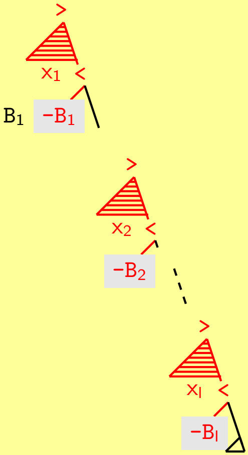
$$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l), C(y_1,\ldots,y_m)$$



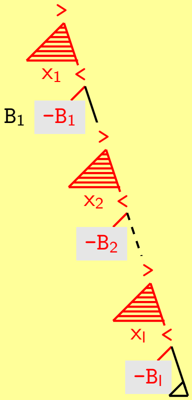
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l), C(y_1,\ldots,y_m)$



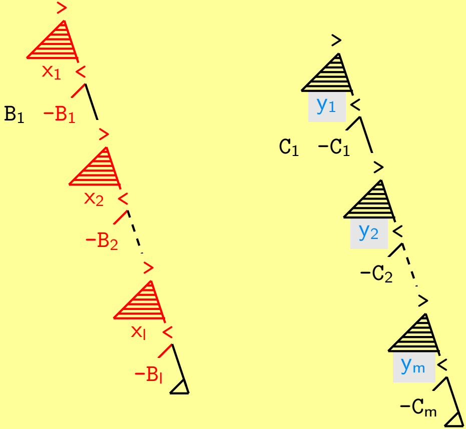
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l), C(y_1,\ldots,y_m)$



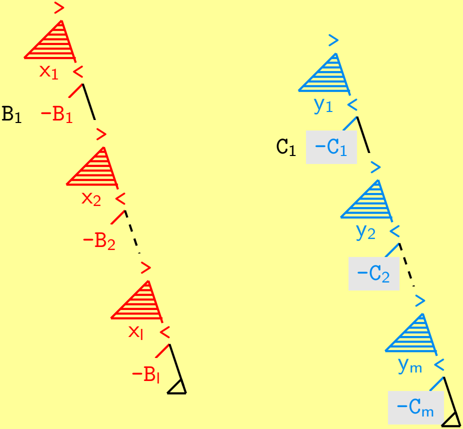
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l), C(y_1,\ldots,y_m)$



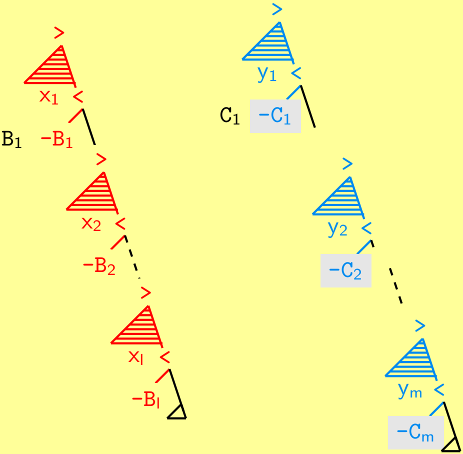
$$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l),\ C(y_1,\ldots,y_m)$$



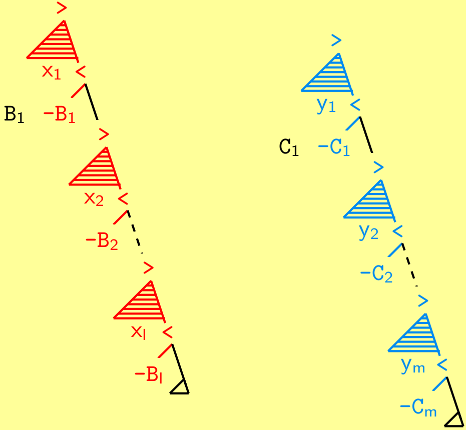
$$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l),\ C(y_1,\ldots,y_m)$$



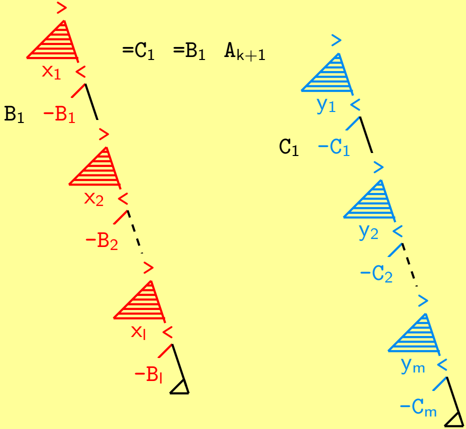
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l), C(y_1,\ldots,y_m)$



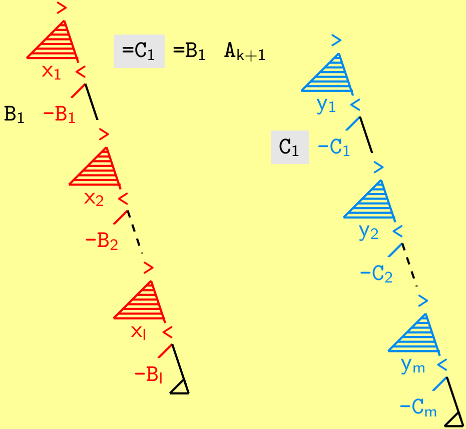
$A(t_1, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)$



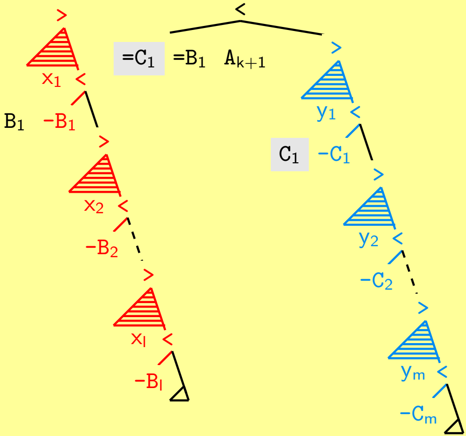
$A(t_1, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)$



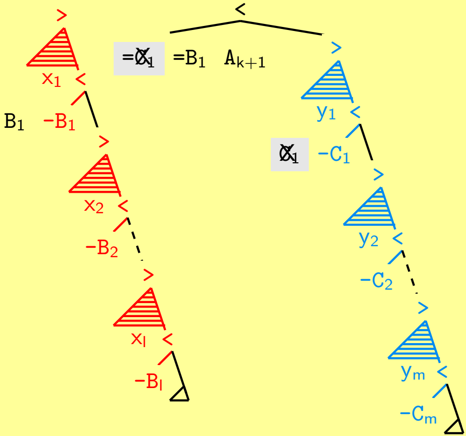
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l)\ ,\ C(y_1,\ldots,y_m)$



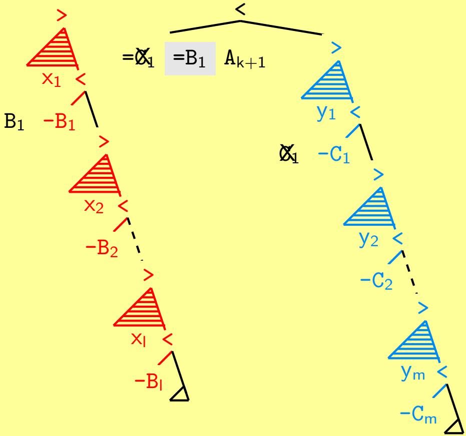
$A(t_1,\dots,t_k) \leftarrow B(x_1,\dots,x_l)\,,\; C(y_1,\dots,y_m)$



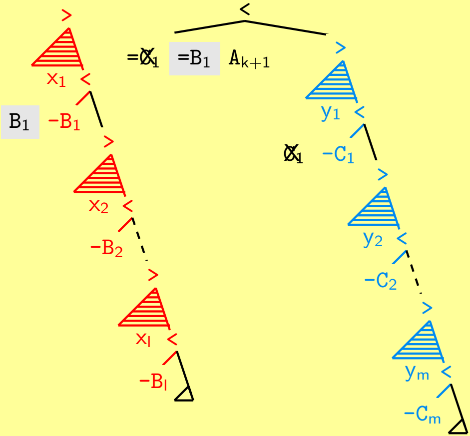
$A(t_1, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)$



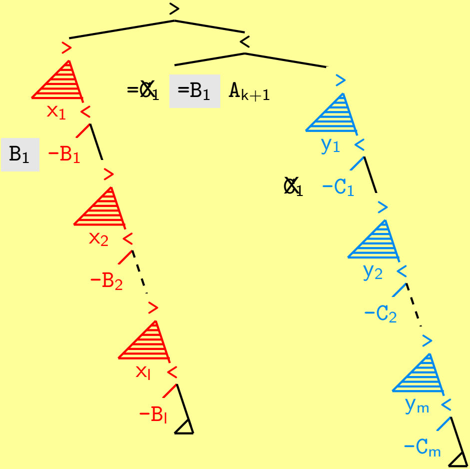
$A(t_1, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)$



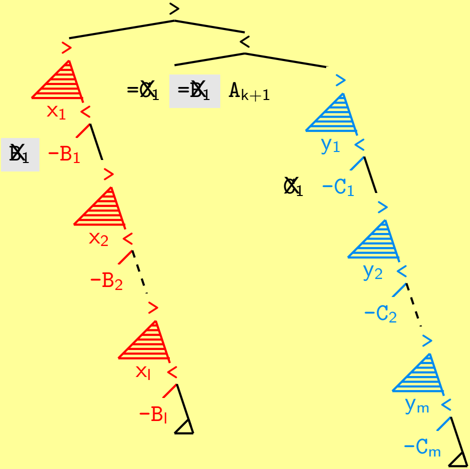
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l), C(y_1,\ldots,y_m)$



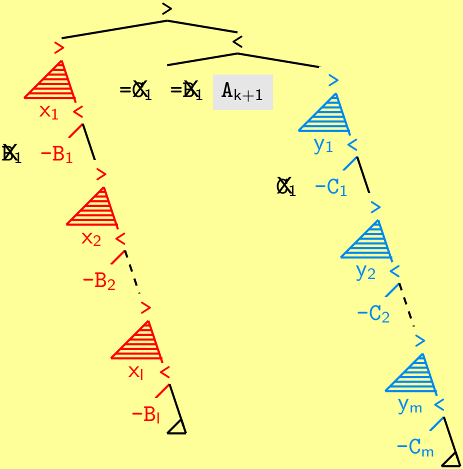
$A(t_1, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)$



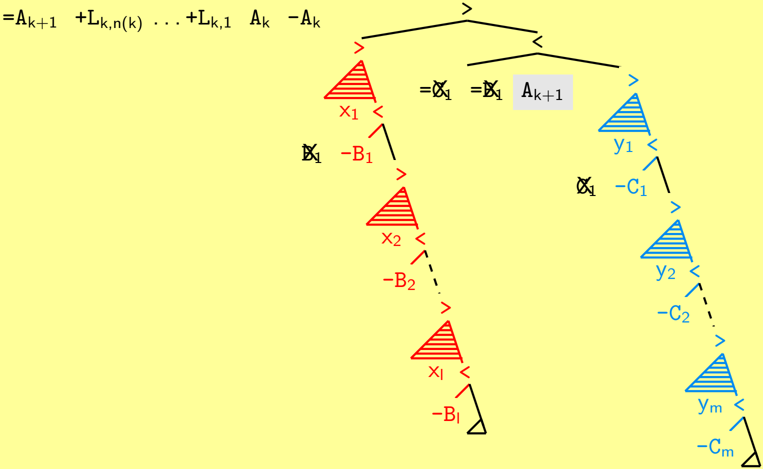
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l)\,,\, C(y_1,\ldots,y_m)$



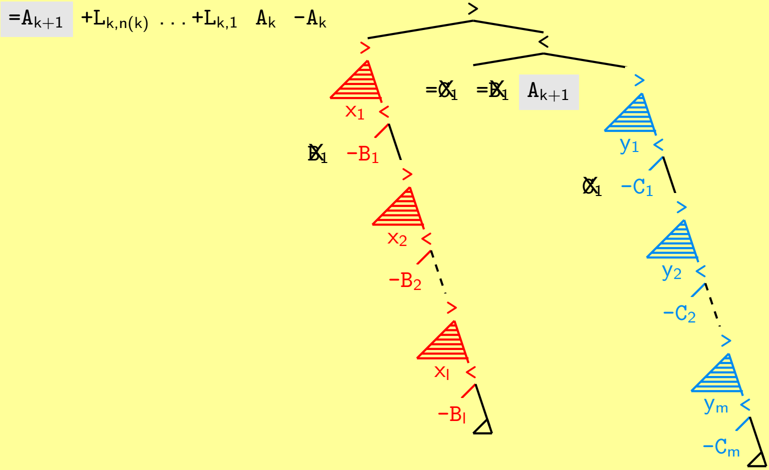
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l)\,,\ C(y_1,\ldots,y_m)$



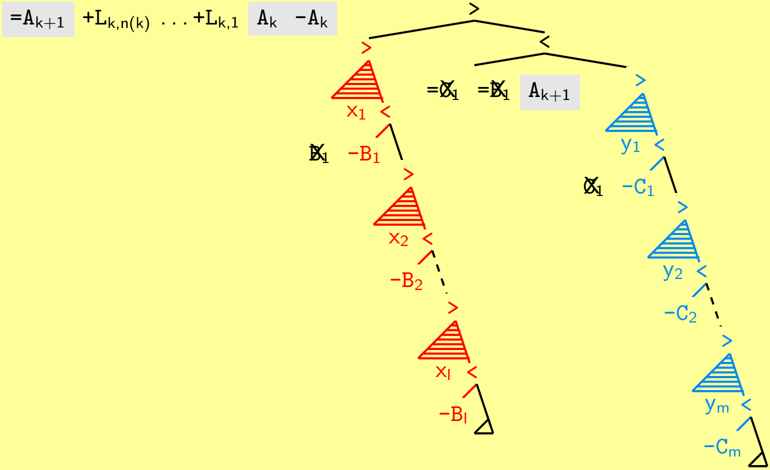
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l)\,,\ C(y_1,\ldots,y_m)$



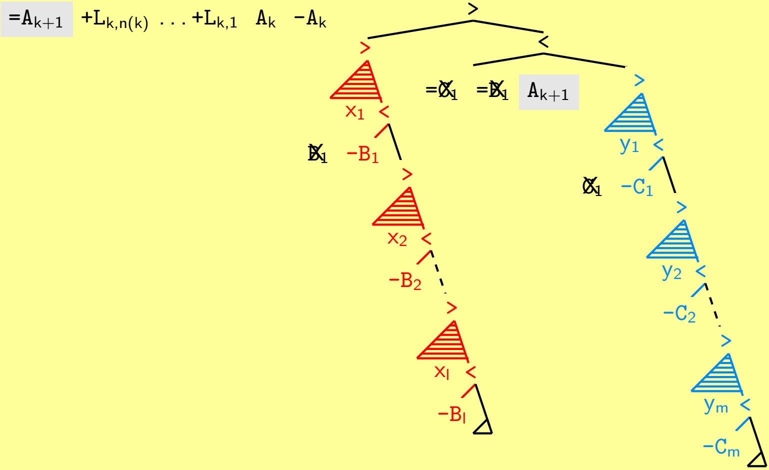
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l)\,,\ C(y_1,\ldots,y_m)$



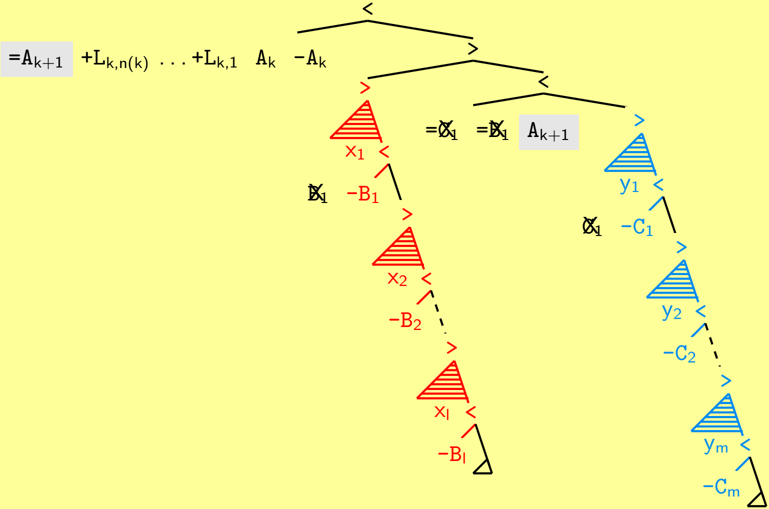
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l),\ C(y_1,\ldots,y_m)$



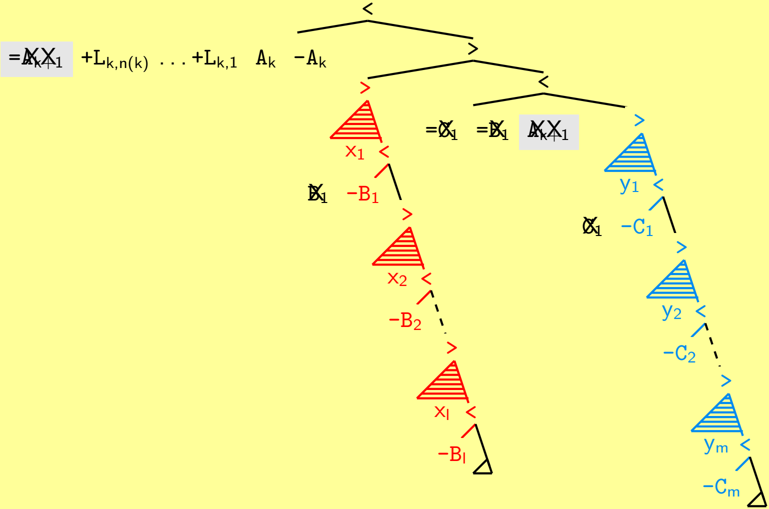
$A(t_1, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)$



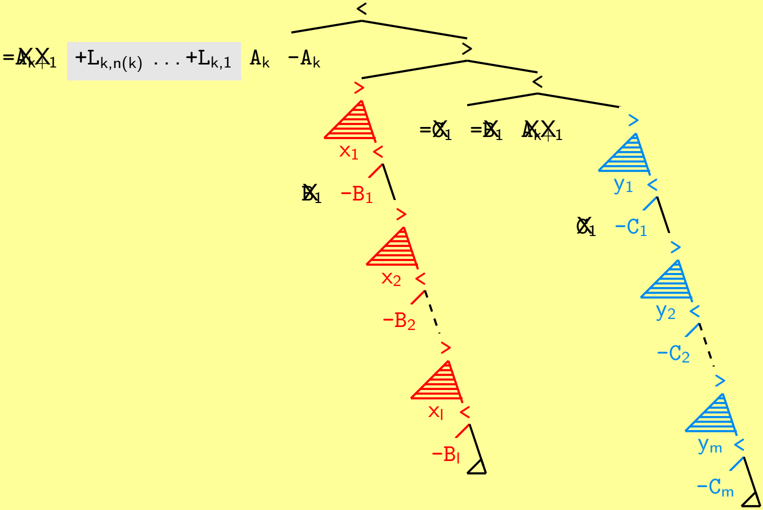
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l),\ C(y_1,\ldots,y_m)$



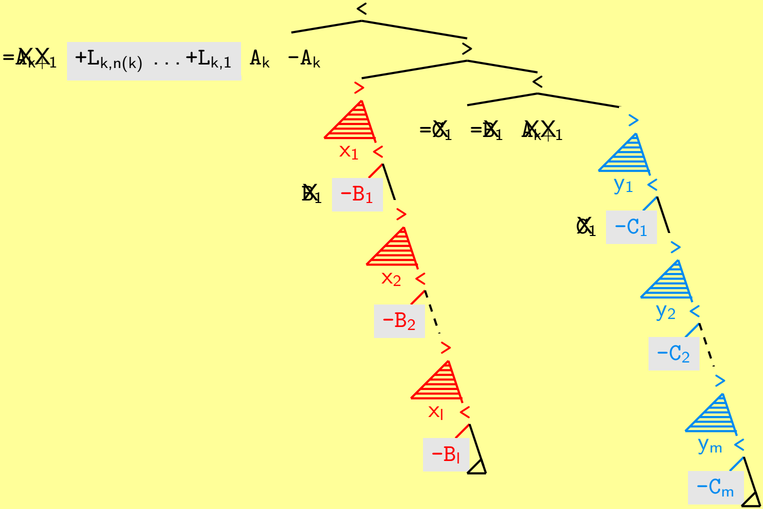
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l)\,,\, C(y_1,\ldots,y_m)$



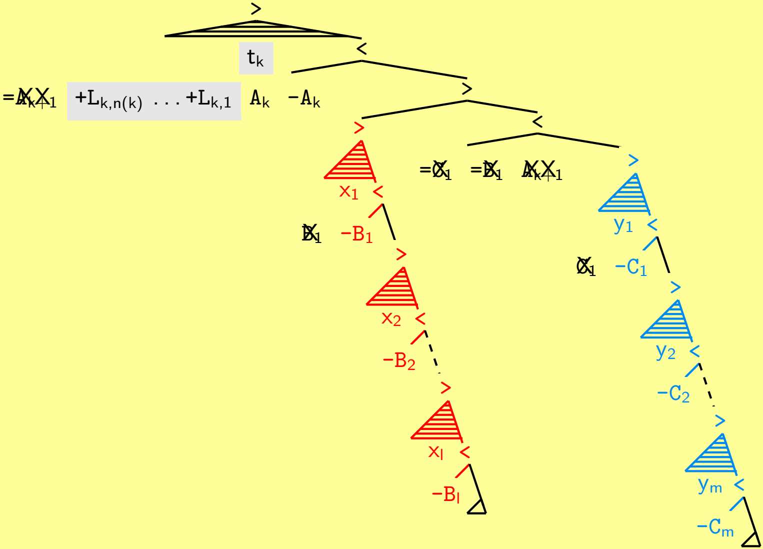
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l), C(y_1,\ldots,y_m)$



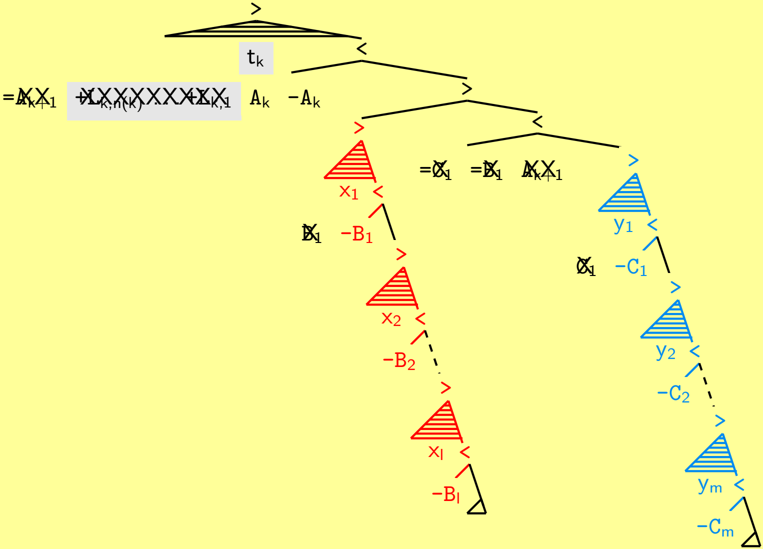
$A(t_1, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l)\, , C(y_1, \ldots, y_m)$



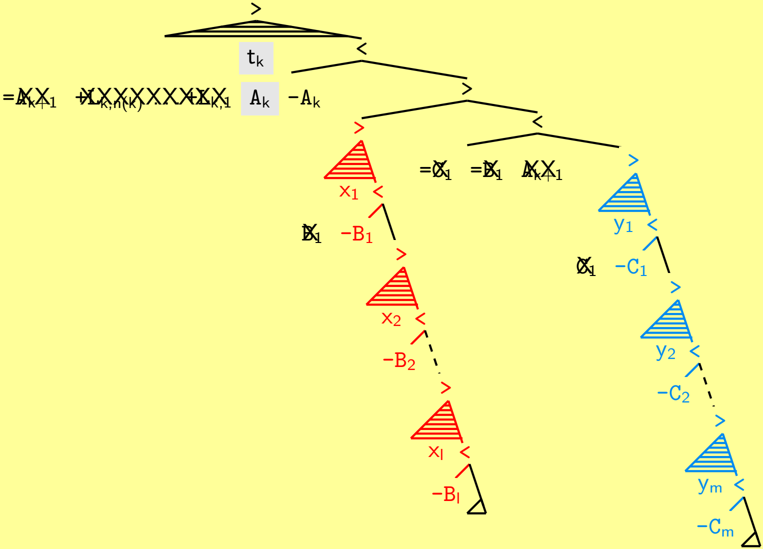
$A(t_1, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l),\ C(y_1, \ldots, y_m)$



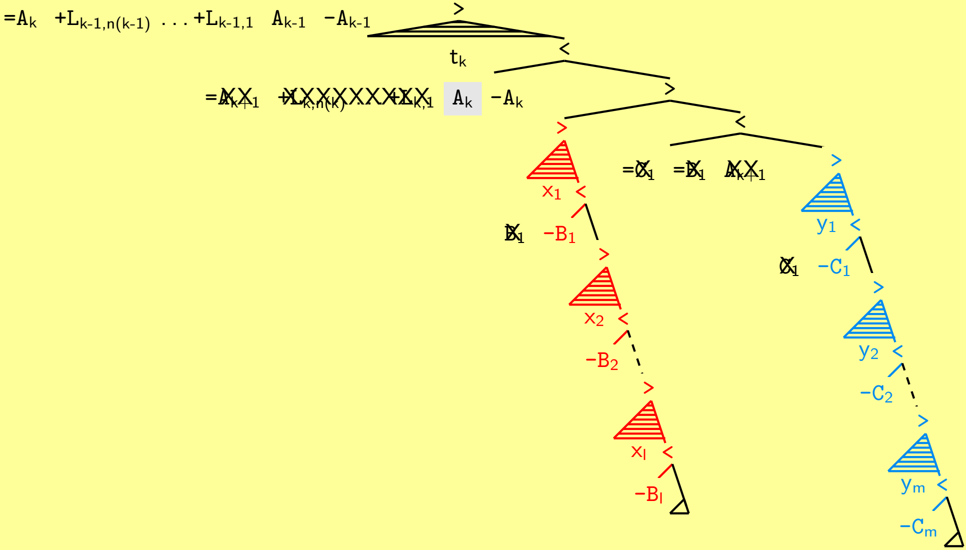
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l)\,,\; C(y_1,\ldots,y_m)$



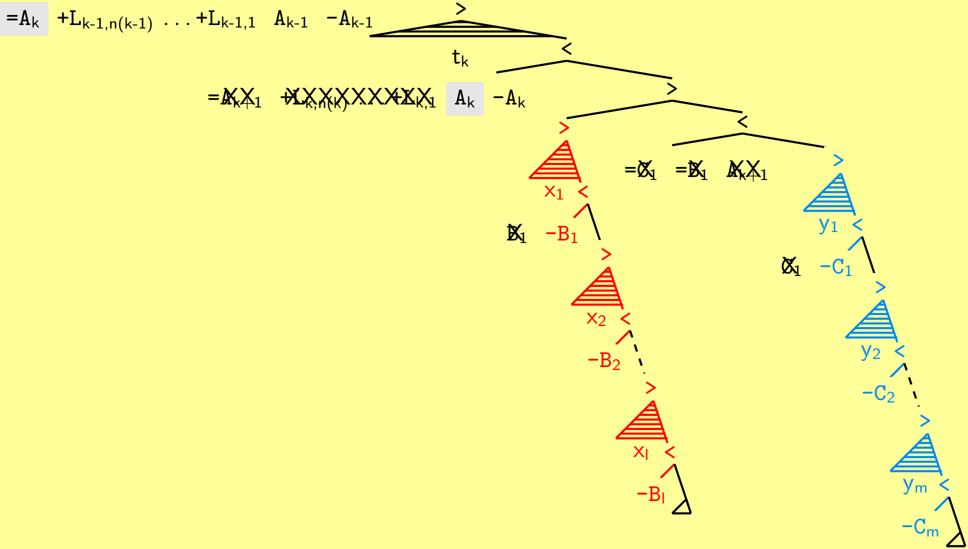
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l)\,,\, C(y_1,\ldots,y_m)$



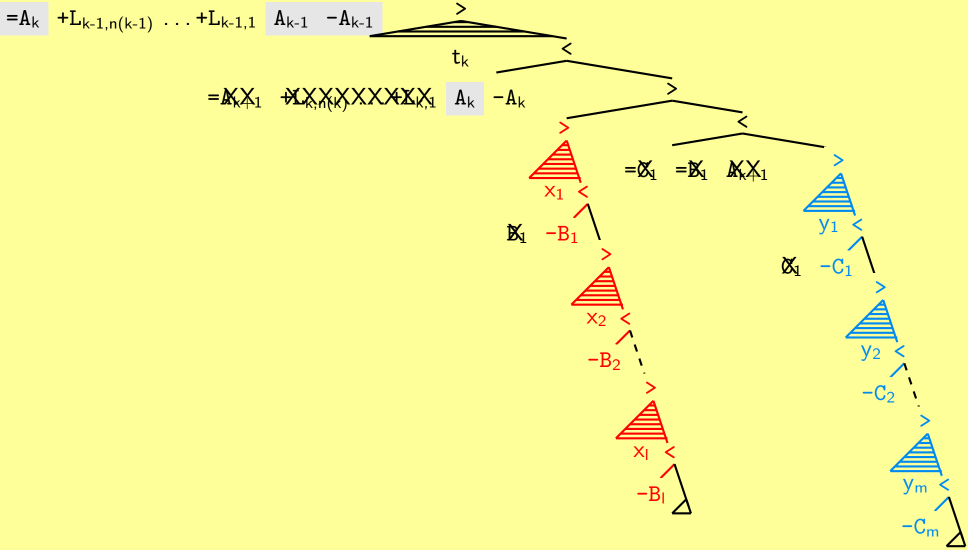
$$A(t_1,\dots,t_k) \leftarrow B(x_1,\dots,x_l),\ C(y_1,\dots,y_m)$$



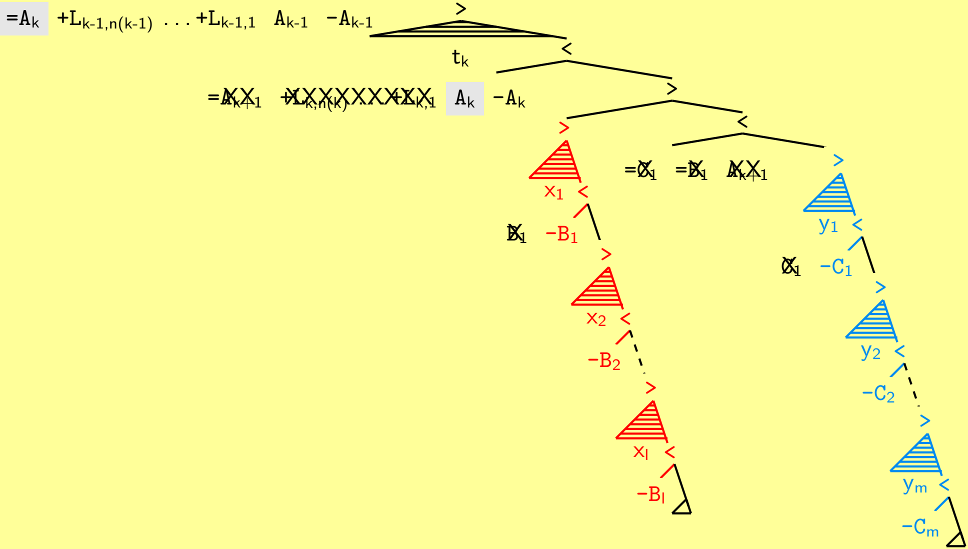
$A(t_1, \ldots, t_k) \leftarrow B(x_1, \ldots, x_l), C(y_1, \ldots, y_m)$



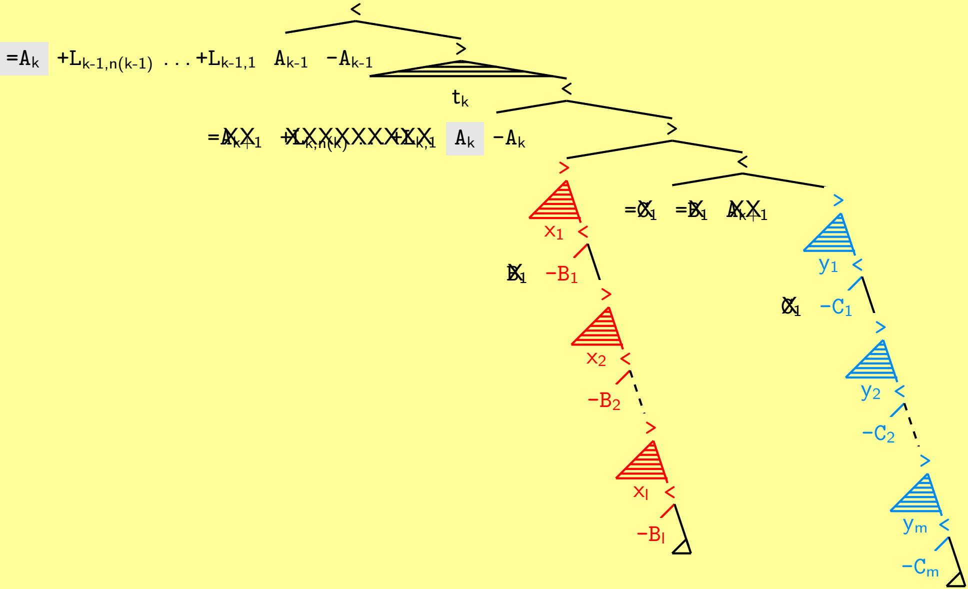
$$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l),\ C(y_1,\ldots,y_m)$$



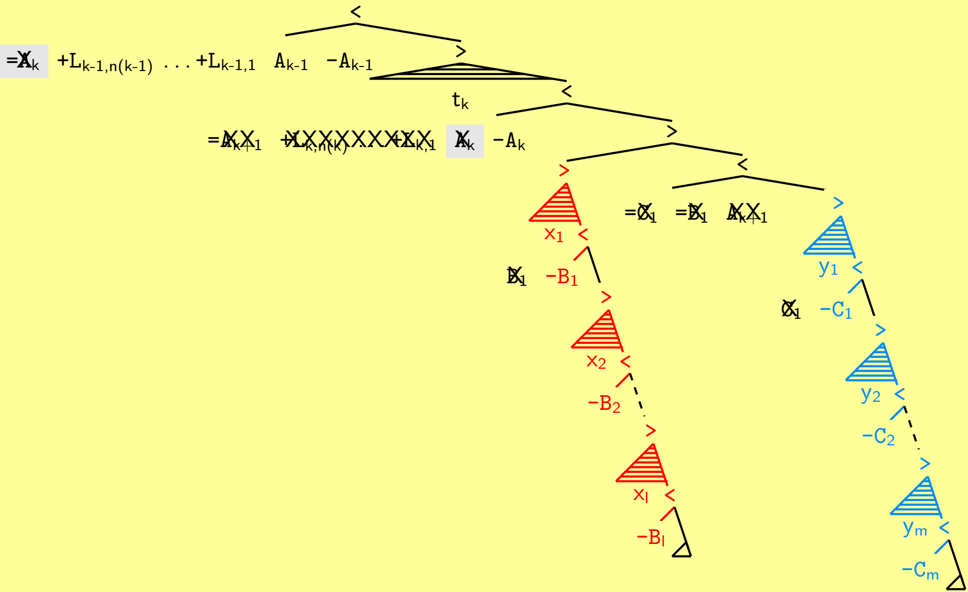
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l),\ C(y_1,\ldots,y_m)$



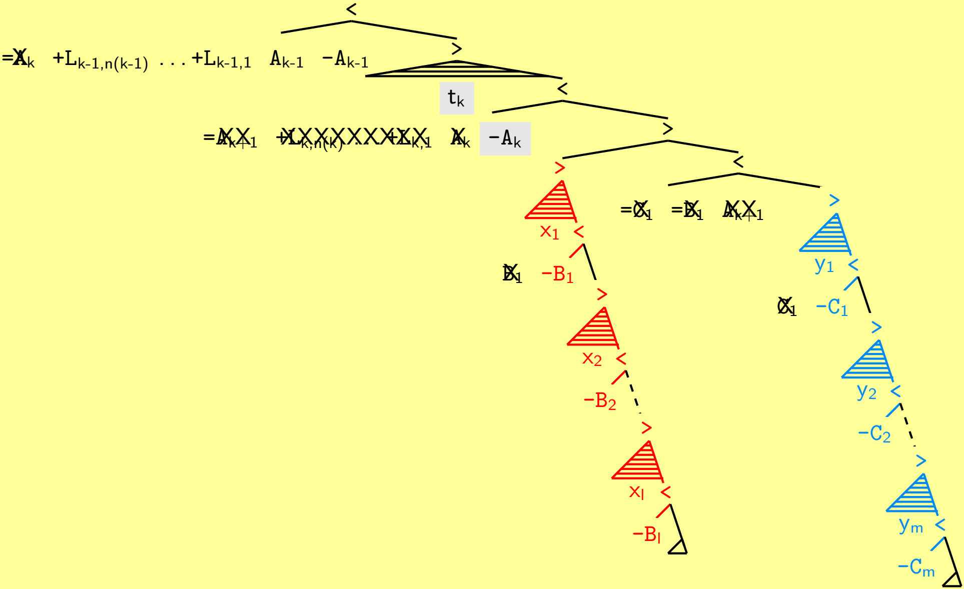
$$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l),\ C(y_1,\ldots,y_m)$$



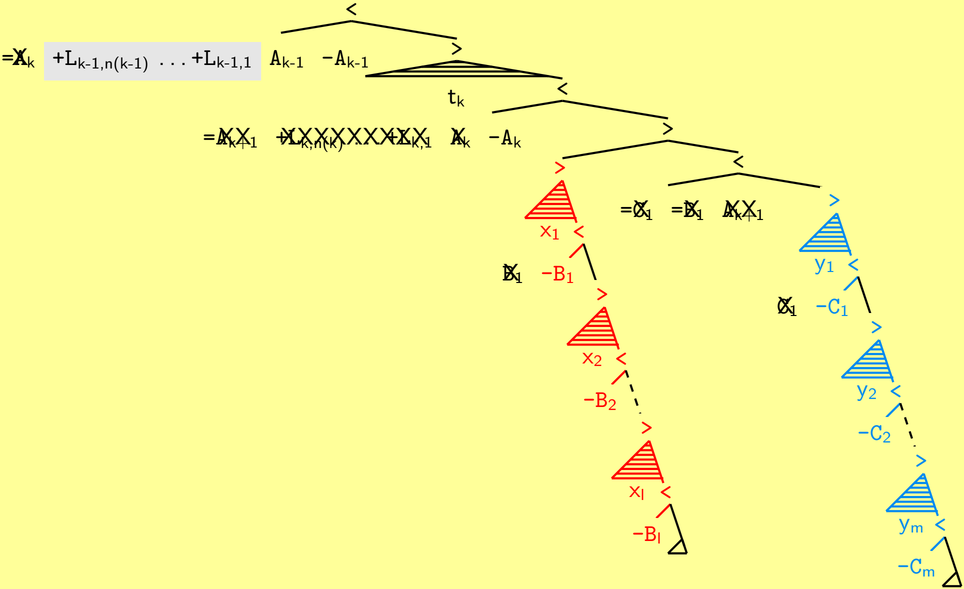
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l)\,,\,C(y_1,\ldots,y_m)$



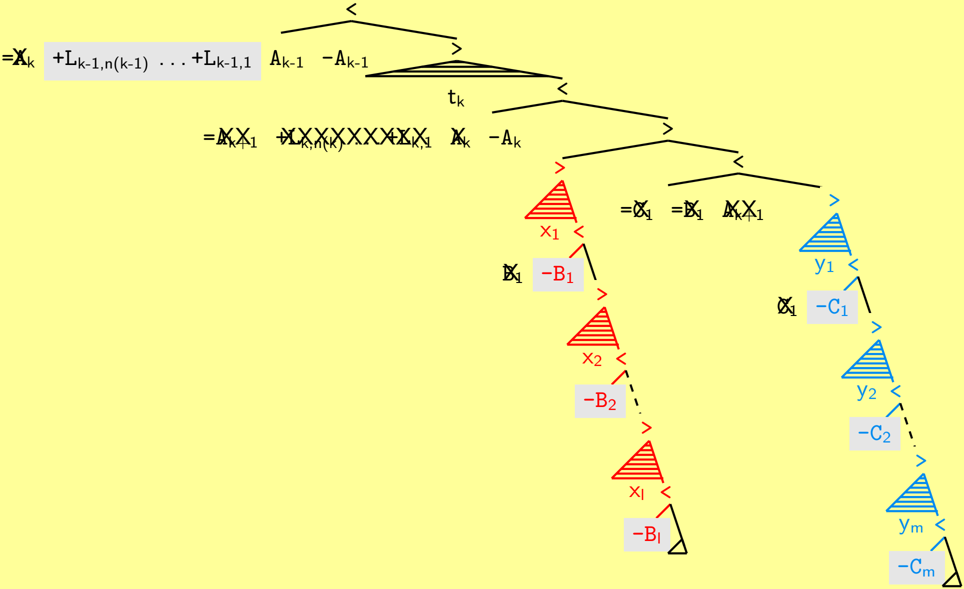
$$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l),\ C(y_1,\ldots,y_m)$$



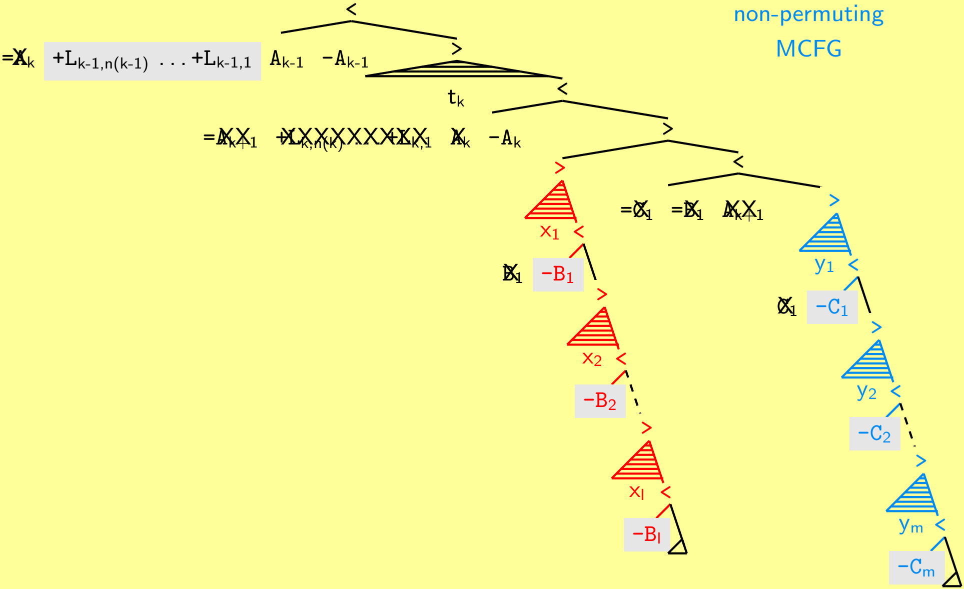
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l),\ C(y_1,\ldots,y_m)$



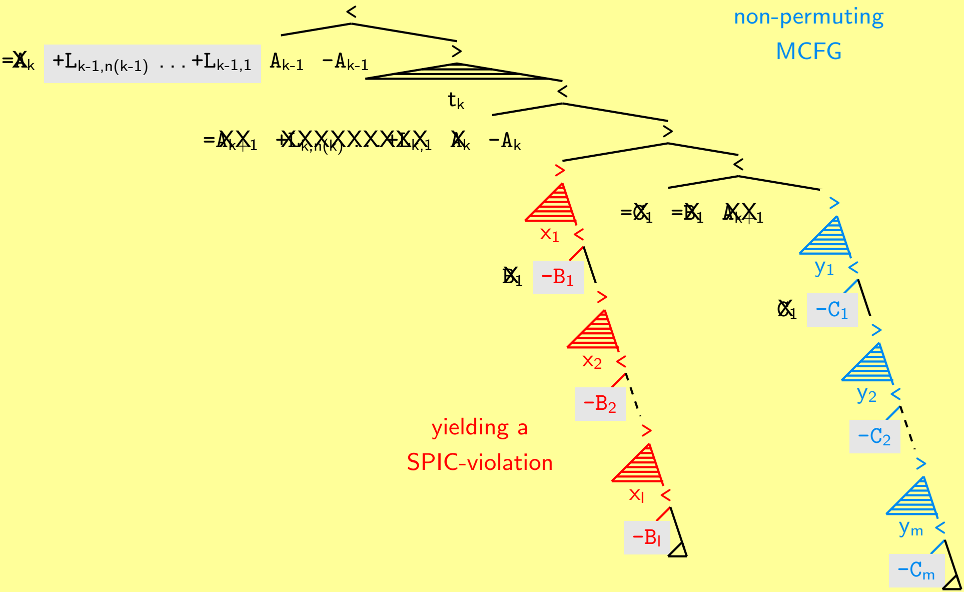
$$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l), C(y_1,\ldots,y_m)$$



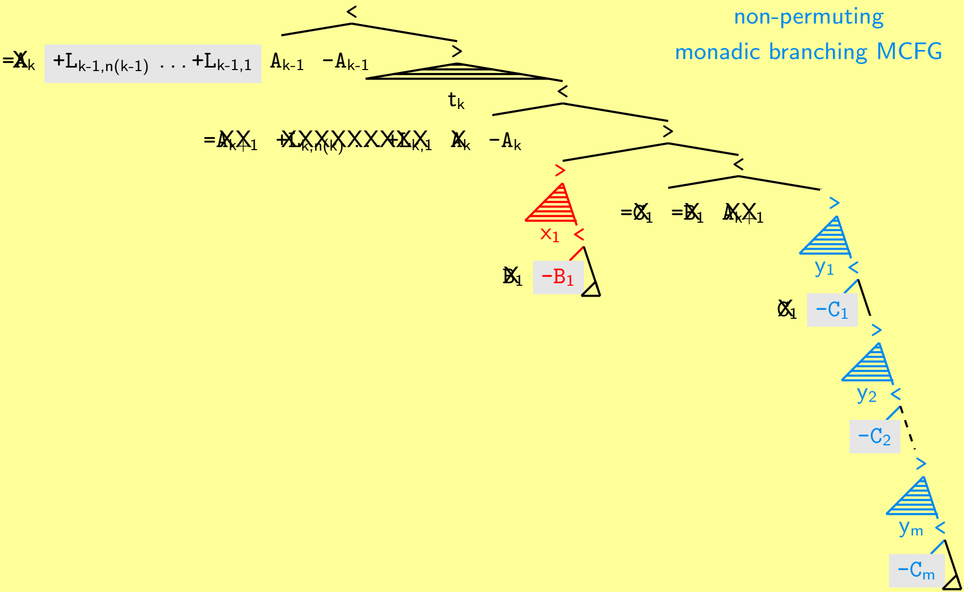
$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l),\, C(y_1,\ldots,y_m)$



$$A(t_1,\ldots,t_k) \leftarrow B(x_1,\ldots,x_l),\ C(y_1,\ldots,y_m)$$



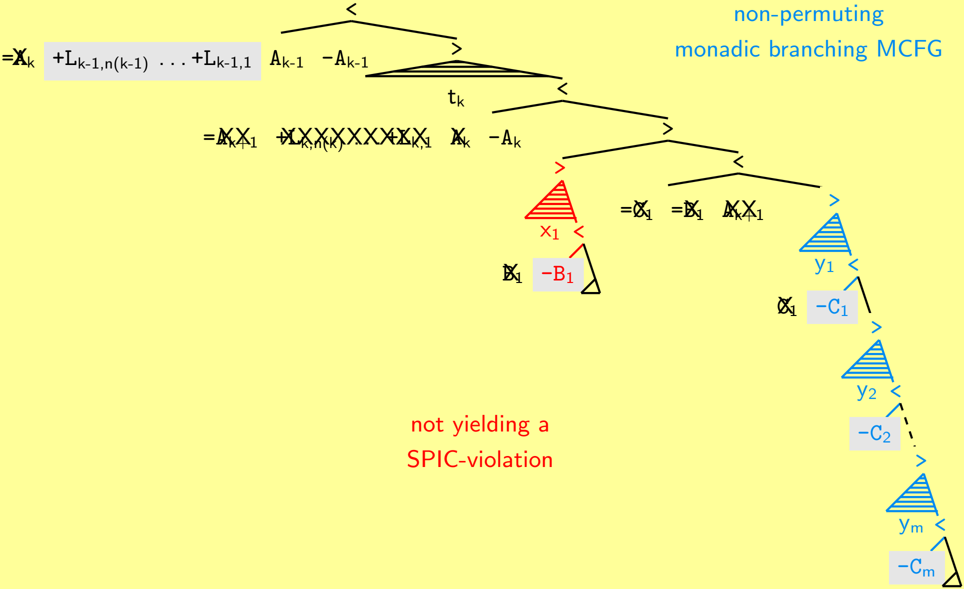
$A(t_1, \ldots, t_k) \leftarrow B(x_1), C(y_1, \ldots, y_m)$



non-permuting

monadic branching MCFG

$A(t_1, \dots, t_k) \leftarrow B(x_1), C(y_1, \dots, y_m)$



non-permuting monadic branching MCFG

not yielding a SPIC-violation

Concluding remarks: MCFG –> MG

- non-deletion condition on transformed MCFG cannot be dropped
- every other condition imposed on transformed MCFG could be generally dropped taking into account a necessary adaption of the transformation procedure
- non-permuting condition in the general case yields an MG obeying $\mathsf{SPIC}_{\mathsf{move}}$
- non-permuting condition necessary to show an MCFGmb results in an MG(+SMC,+SPIC)
- if there is no doublet-freeness, implementation of an additional "move-cycle" is necessary to arrive in an equivalent MG

Concluding remarks: MCFG –> MG

- if syncategorematic material appears in non-terminating rules, additional selectors in the defined lexical items are necessary, as well as additional "non-movable" lexical items representing the syncategorematic material
- terminating rules in general MCFG-form need both: "licensees and selectors," that is to say, those rules need more than one laxical MG-entry.
- it is also possible to construct the resulting MG such that there is only one specifier per head, and such that specifiers are additionally non-movable in the $\mathrm{MCFG}_{\mathrm{mb}}$-case, the latter leading to a strict MG in the sense of Stabler 1999.

Concluding remarks: MG –> MCFG (repeated)

- Feature consumption plus SMC are the crucial ingredients.
- Proof is more than a proof of just an embedding of string language classes.
- Adaption is possible, when head movement, left complement selection, rightward movement/extraposition and/or covert movement/agree is incorporated into the MG-formalism.
- Adaption is also possible, when late adjunction together with adjunct island condition is incorporated into the MG-formalism. This, in fact, is "more strictly" about string language equivalence.
- Adding SPIC yields monadic branching MCFGs as output. Note that the set of relevant trees can be reduced in this case.

de Groote, Philippe, Glyn Morrill, and Christian Retoré, eds. 2001. *Logical Aspects of Computational Linguistics (LACL 2001), Lecture Notes in Artificial Intelligence Vol. 2099*. Berlin, Heidelberg: Springer.

Frey, Werner and Hans-Martin Gärtner. 2002. On the treatment of scrambling and adjunction in minimalist grammars. In *Proceedings of the Conference on Formal Grammar (FGTrento), Trento*, pages 41–52.

Gärtner, Hans-Martin and Jens Michaelis. 2008. A note on countercyclicity and minimalist grammars. In G. Penn, ed., *Proceedings of FGVienna: The 8th Conference on Formal Grammar*, pages 95–109. Stanford, CA: CSLI Publications. `http://cslipublications.stanford.edu/FG/2003/index.html`.

Harkema, Henk. 2001. A characterization of minimalist languages. In de Groote et al. (2001), pages 193–211.

Joshi, Aravind K. 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*, pages 206–250. New York, NY: Cambridge University Press.

Kanazawa, Makoto, Jens Michaelis, Sylvain Salvati, and Ryo Yoshinaka. 2011. Well-nestedness properly subsumes strict derivational minimalism. In S. Pogodalla and J.-P. Prost, eds., *Logical Aspects of Computational Linguistics (LACL 2011), Lecture Notes in Artificial Intelligence Vol. 6736*, pages 112–128. Berlin, Heidelberg: Springer.

Kanazawa, Makoto and Sylvain Salvati. 2010. The copying power of well-nested multiple context-free grammars. In A.-H. Dediu, H. Fernau, and C. Martín-Vide, eds., *Language and Automata Theory and Applications (LATA 2010), Lecture Notes in Computer Science Vol. 6031*, pages 344–355. Berlin, Heidelberg: Springer.

Kobele, Gregory M. 2006. *Generating Copies. An investigation into structural identity in language and grammar*. Ph.D. thesis, University of California, Los Angeles, CA.

Kobele, Gregory M. and Jens Michaelis. 2009. Two type-0 variants of minimalist grammars. In Rogers (2009), pages 81–91. `http://cslipublications.stanford.edu/FG/2005/index.html`.

Michaelis, Jens. 2001a. Derivational minimalism is mildly context-sensitive. In M. Moortgat, ed., *Logical Aspects of Computational Linguistics (LACL '98), Lecture Notes in Artificial Intelligence Vol. 2014*, pages 179–198. Berlin, Heidelberg: Springer.

Michaelis, Jens. 2001b. *On Formal Properties of Minimalist Grammars*. Linguistics in Potsdam 13. Potsdam: Universitätsbibliothek, Publikationsstelle. Ph.D. thesis.

Michaelis, Jens. 2001c. Transforming linear context-free rewriting systems into minimalist grammars. In de Groote et al. (2001), pages 228–244.

Michaelis, Jens. 2004. Observations on strict derivational minimalism. *Electronic Notes in Theoretical Computer Science* 53:192–209. Available at `http://www.sciencedirect.com/science/journal/15710661`.

Michaelis, Jens. 2009. An additional observation on strict derivational minimalism. In Rogers (2009), pages 101–111. `http://cslipublications.stanford.edu/FG/2005/index.html`.

Rogers, James, ed. 2009. *Proceedings of FG-MoL 2005: The 10th Conference on Formal Grammar and The 9th Meeting on Mathematics of Language*. Stanford, CA: CSLI Publications.

Salvati, Sylvain. 2011. Minimalist grammars in the light of logic. In S. Pogodalla, M. Quatrini, and C. Retoré, eds., *Logic and Grammar. Essays Dedicated to Alain Lecomte on the Occasion of His 60th Birthday, Lecture Notes in Artificial Intelligence Vol. 6700*, pages 81–117. Berlin, Heidelberg: Springer.

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

Stabler, Edward P. 1997. Derivational minimalism. In C. Retoré, ed., *Logical Aspects of Computational Linguistics (LACL '96), Lecture Notes in Artificial Intelligence Vol. 1328*, pages 68–95. Berlin, Heidelberg: Springer.

Stabler, Edward P. 1999. Remnant movement and complexity. In G. Bouma, G.-J. M. Kruijff, E. Hinrichs, and R. T. Oehrle, eds., *Constraints and Resources in Natural Language Syntax and Semantics*, pages 299–326. Stanford, CA: CSLI Publications.

Stabler, Edward P. 2001. Recognizing head movement. In de Groote et al. (2001), pages 245–260.

Stabler, Edward P. 2011. Computational perspectives on minimalism. In C. Boeckx, ed., *Oxford Handbook of Linguistic Minimalism*, pages 616–641. New York, NY: Oxford University Press.

Stabler, Edward P. and Edward L. Keenan. 2003. Structural similarity within and among languages. *Theoretical Computer Science* 293:345–363.