Parsing of Natural Languages

Heiko Vogler / Dresden, Germany

joint work with

Frank Drewes / Umeå, Sweden

Mark-Jan Nederhof / St Andrews, Scotland

Kilian Gebhardt / Dresden

Markus Teichmann / Dresden

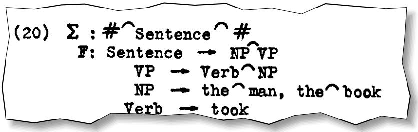
[Chomsky 56]: Three Models for the Description of Language.

"There are **two central problems** *in the descriptive study of languages. One primary concern of the linguist is* **to discover simple and *revealing* grammars for natural languages***. At the same time,* **by studying the properties of such successful grammars ...,** *he hopes to arrive at a general theory of linguistic structure."*

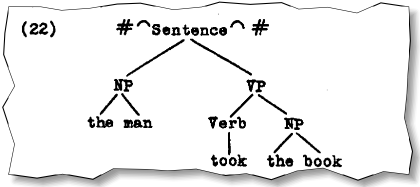
[Chomsky 56]: Three Models for the Description of Language.

> *"There are* **two central problems** *in the descriptive study of languages. One primary concern of the linguist is* **to discover simple and *revealing* grammars for natural languages***. At the same time,* **by studying the properties of such successful grammars ...,** *he hopes to arrive at a general theory of linguistic structure."*

context-free grammar:



phrase structure tree:



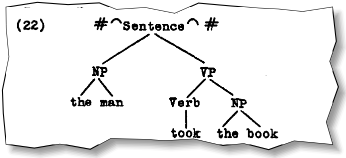
E: natural language $\quad$ $\mathcal{H}$: set of syntactic structures

$$\boxed{\text{parsing} : E \to \mathcal{H}}$$

E: natural language $\mathcal{H}$: set of syntactic structures

$$\boxed{\text{parsing} : E \rightarrow \mathcal{H}}$$

parsing with context-free grammar G:

parsing(the man took the book) $=$ 

Outline

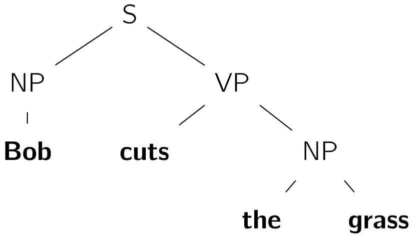
- Syntax of natural language sentences
- Ambiguity and parsing
- A particular "simple and revealing grammar" model
- Probabilistic hybrid grammars

Outline

- Syntax of natural language sentences
- Ambiguity and parsing
- A particular "simple and revealing grammar" model
- Probabilistic hybrid grammars

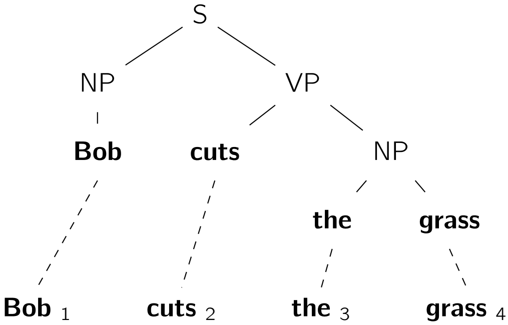
Bob $_1$ **cuts** $_2$ **the** $_3$ **grass** $_4$

phrase structure tree:

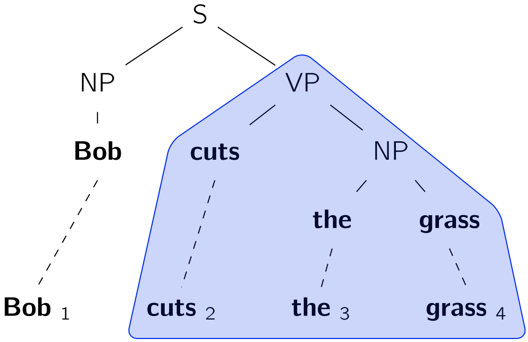


Bob $_1$ **cuts** $_2$ **the** $_3$ **grass** $_4$

phrase structure tree:



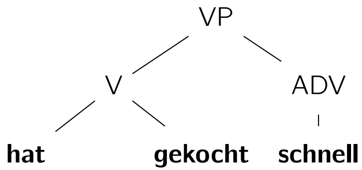
phrase structure tree: (continuous)



continuous:
each tree position covers an interval of sentence positions

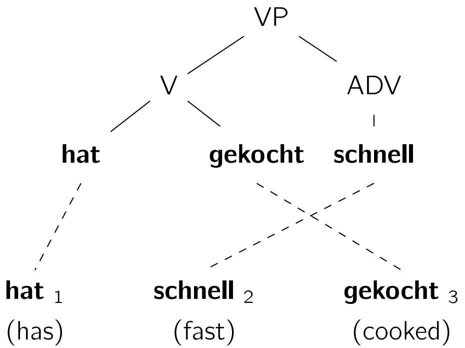
hat $_1$	**schnell** $_2$	**gekocht** $_3$
(has)	(fast)	(cooked)

phrase structure tree:

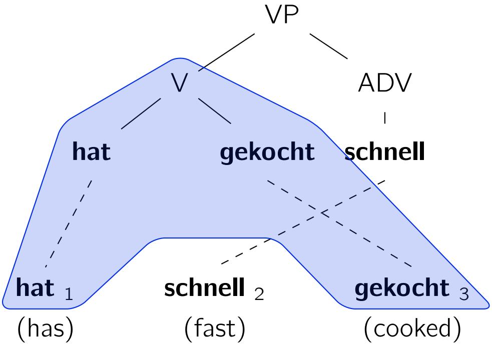


hat $_1$	**schnell** $_2$	**gekocht** $_3$
(has)	(fast)	(cooked)

phrase structure tree:



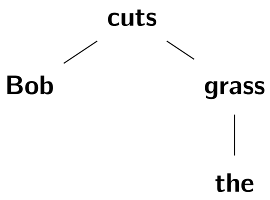
phrase structure tree: (discontinuous)



[Tesnière 59]

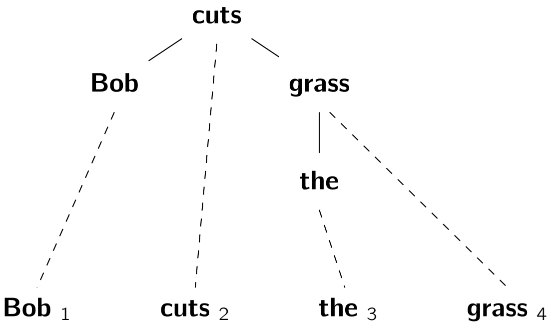
Bob $_1$ **cuts** $_2$ **the** $_3$ **grass** $_4$

dependency tree: [Tesnière 59]

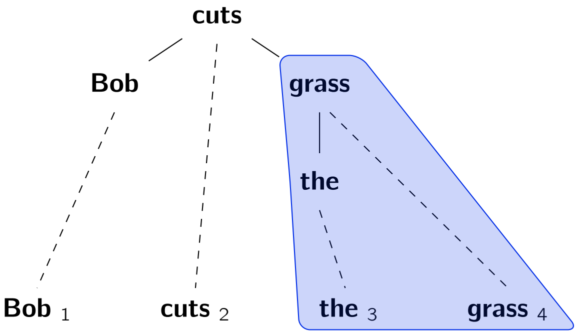


Bob $_1$ **cuts** $_2$ **the** $_3$ **grass** $_4$

dependency tree: [Tesnière 59]



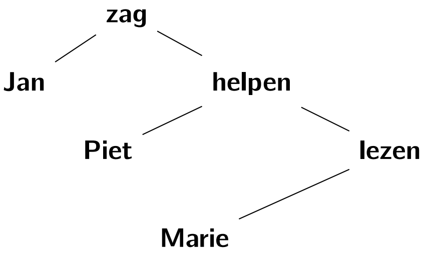
dependency tree: (projective) [Tesnière 59]



projective:
each tree position covers an interval of sentence positions

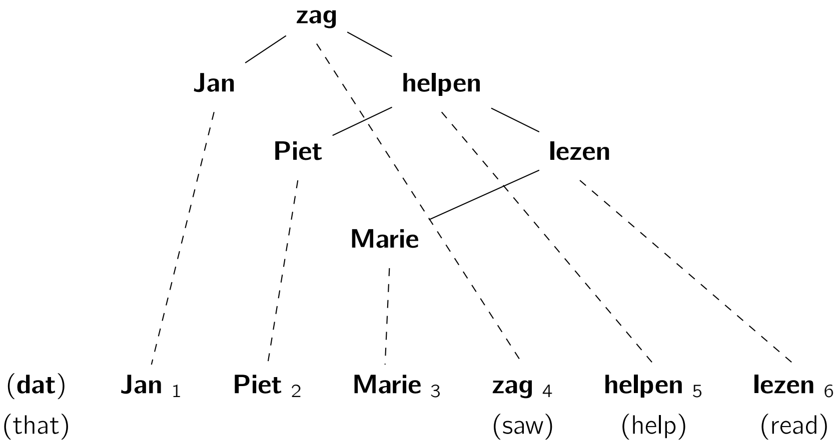
(**dat**)	**Jan** $_1$	**Piet** $_2$	**Marie** $_3$	**zag** $_4$	**helpen** $_5$	**lezen** $_6$
(that)				(saw)	(help)	(read)

dependency tree:

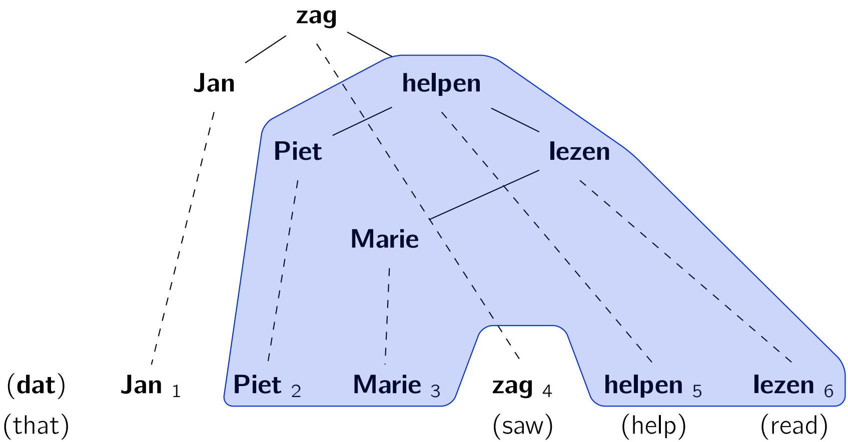


(**dat**)	**Jan** $_1$	**Piet** $_2$	**Marie** $_3$	**zag** $_4$	**helpen** $_5$	**lezen** $_6$
(that)				(saw)	(help)	(read)

dependency tree:



dependency tree: (non-projective)



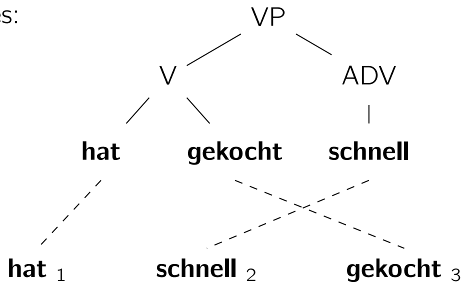
Let Σ be a ranked alphabet. A hybrid tree is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \mathrm{pos}(\xi)$
- $\preceq$ linear order on U

Let Σ be a ranked alphabet. A hybrid tree is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \text{pos}(\xi)$
- $\preceq$ linear order on U

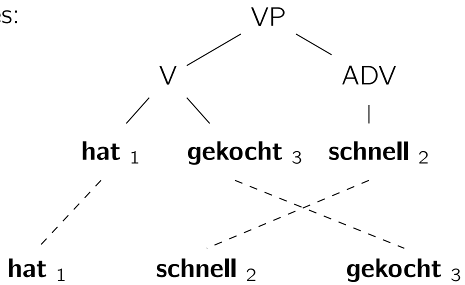
phrase structure trees:
$U = \text{leaves}(\xi)$



Let Σ be a ranked alphabet. A hybrid tree is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \mathrm{pos}(\xi)$
- $\preceq$ linear order on U

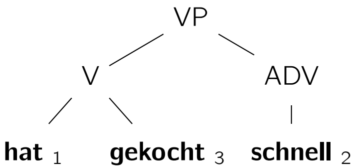
phrase structure trees:
$U = \mathrm{leaves}(\xi)$



Let Σ be a ranked alphabet. A hybrid tree is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \mathsf{pos}(\xi)$
- $\preceq$ linear order on U

phrase structure trees:
$U = \mathsf{leaves}(\xi)$



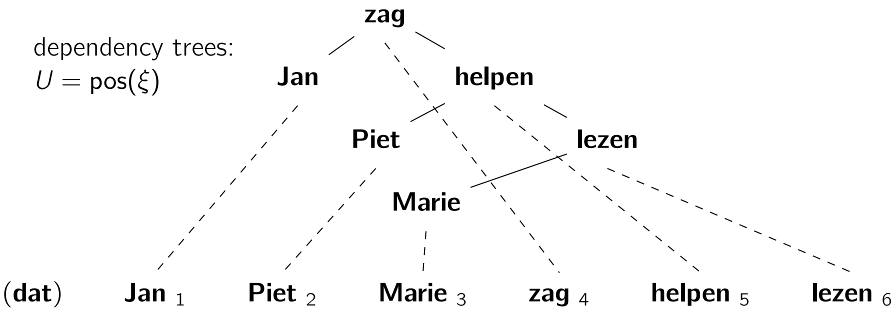
Let Σ be a ranked alphabet. A <u>hybrid tree</u> is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \mathsf{pos}(\xi)$
- $\preceq$ linear order on U

Let Σ be a ranked alphabet. A hybrid tree is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \mathrm{pos}(\xi)$
- $\preceq$ linear order on U

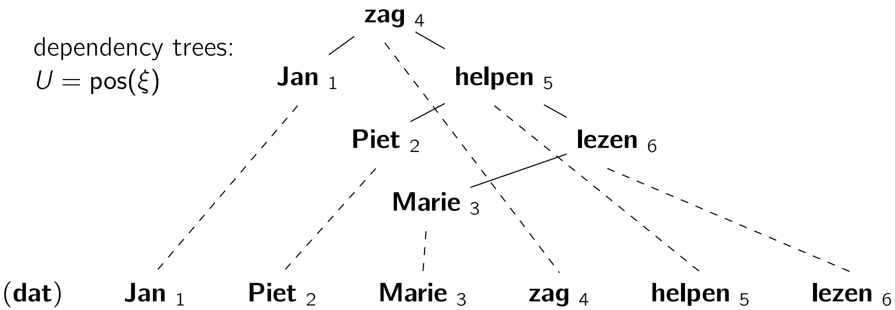
dependency trees:
$U = \mathrm{pos}(\xi)$



Let Σ be a ranked alphabet. A hybrid tree is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \mathsf{pos}(\xi)$
- $\preceq$ linear order on U

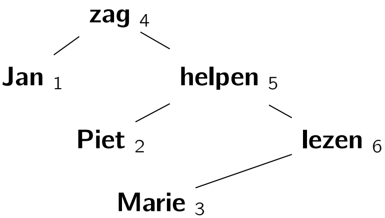
dependency trees:
$U = \mathsf{pos}(\xi)$



Let Σ be a ranked alphabet. A hybrid tree is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \mathrm{pos}(\xi)$
- $\preceq$ linear order on U

dependency trees:
$U = \mathrm{pos}(\xi)$



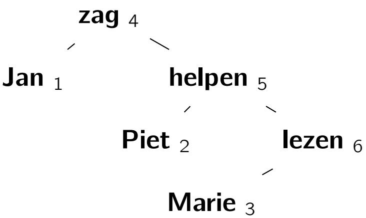
Let Σ be a ranked alphabet. A hybrid tree is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \mathrm{pos}(\xi)$
- $\preceq$ linear order on U

Let Σ be a ranked alphabet. A <u>hybrid tree</u> is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \text{pos}(\xi)$
- $\preceq$ linear order on U

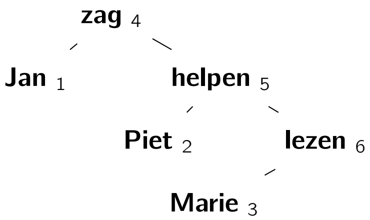
h:



Let Σ be a ranked alphabet. A hybrid tree is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \mathrm{pos}(\xi)$
- $\preceq$ linear order on U

h:

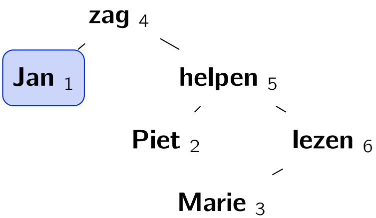


$\mathrm{str}(h) =$

Let Σ be a ranked alphabet. A hybrid tree is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \mathrm{pos}(\xi)$
- $\preceq$ linear order on U

h:

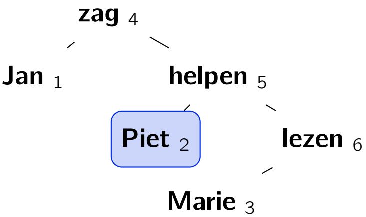


$\mathrm{str}(h) = \boxed{\textbf{Jan}\ _1}$

Let Σ be a ranked alphabet. A hybrid tree is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \mathrm{pos}(\xi)$
- $\preceq$ linear order on U

h:

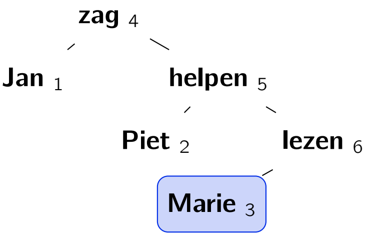


$\mathrm{str}(h) = \textbf{Jan}_1 \ \boxed{\textbf{Piet}_2}$

Let Σ be a ranked alphabet. A hybrid tree is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \mathrm{pos}(\xi)$
- $\preceq$ linear order on U

h:

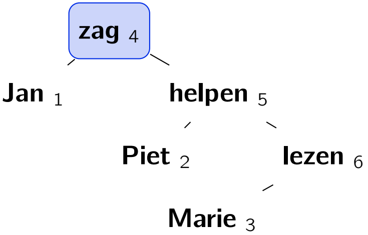


$\mathrm{str}(h) =$ **Jan** $_1$ **Piet** $_2$ $\boxed{\textbf{Marie}\ _3}$

Let Σ be a ranked alphabet. A <u>hybrid tree</u> is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \mathrm{pos}(\xi)$
- $\preceq$ linear order on U

h:

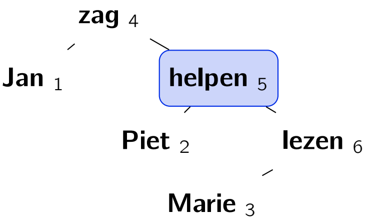


$\mathrm{str}(h) = $ **Jan** $_1$ **Piet** $_2$ **Marie** $_3$ $\boxed{\textbf{zag}\ _4}$

Let Σ be a ranked alphabet. A hybrid tree is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \mathrm{pos}(\xi)$
- $\preceq$ linear order on U

h:

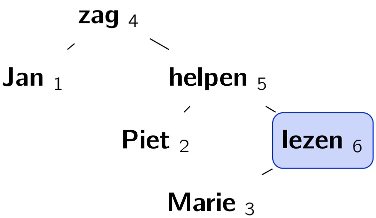


$\mathrm{str}(h) =$ **Jan** $_1$ **Piet** $_2$ **Marie** $_3$ **zag** $_4$ $\boxed{\textbf{helpen}\ _5}$

Let Σ be a ranked alphabet. A hybrid tree is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \mathrm{pos}(\xi)$
- $\preceq$ linear order on U

h:

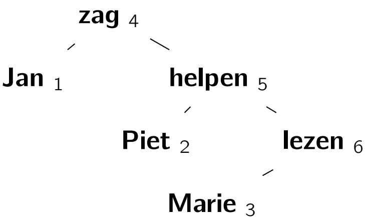


$\mathrm{str}(h) =$ **Jan** $_1$ **Piet** $_2$ **Marie** $_3$ **zag** $_4$ **helpen** $_5$ $\boxed{\textbf{lezen}\ _6}$

Let Σ be a ranked alphabet. A <u>hybrid tree</u> is a tuple $h = (\xi, U, \preceq)$ where

- ξ is a tree over Σ
- $U \subseteq \text{pos}(\xi)$
- $\preceq$ linear order on U

h:



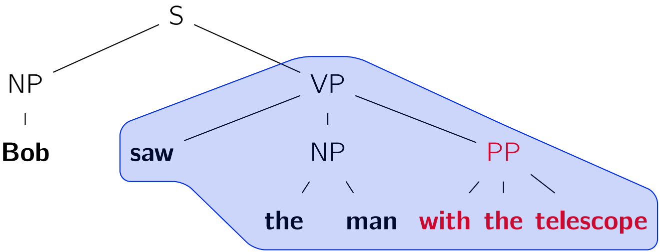
$\mathrm{str}(h) =$ **Jan** $_1$ **Piet** $_2$ **Marie** $_3$ **zag** $_4$ **helpen** $_5$ **lezen** $_6$

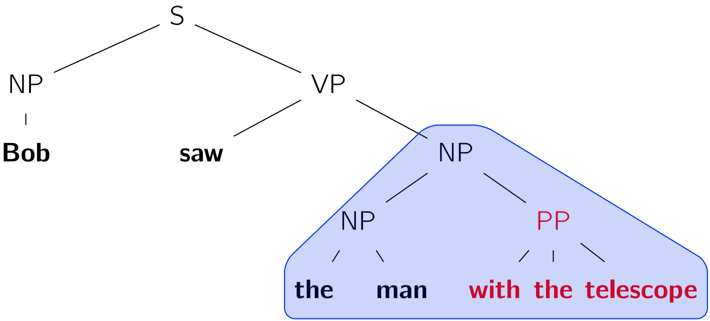
Outline

- Syntax of natural language sentences
  - hybrid trees: phrase structure trees, dependency trees
- Ambiguity and parsing
- A particular "simple and revealing grammar" model
- Probabilistic hybrid grammars

Outline

- Syntax of natural language sentences
  - hybrid trees: phrase structure trees, dependency trees
- $\boxed{\text{Ambiguity and parsing}}$
- A particular "simple and revealing grammar" model
- Probabilistic hybrid grammars


S
NP
VP
Bob
saw
NP
PP
the
man
with
the
telescope


S
NP
VP
Bob
saw
NP
NP
PP
the
man
with
the
telescope

E: natural language $\quad \mathcal{H}$: set of syntactic structures

parsing : $E \rightarrow \mathcal{H}$

$$\text{parsing}(e) = \operatorname*{argmax}_{\substack{h \in \mathcal{H}: \\ \mathrm{str}(h) = e}} P(h \mid \mathcal{M})$$

where

- $\mathcal{M}$: probabilistic language model
- $P(h \mid \mathcal{M})$: probability of h given $\mathcal{M}$

Outline

- Syntax of natural language sentences
  - hybrid trees: phrase structure trees, dependency trees
- Ambiguity and parsing
  - probabilistic language models
- A particular "simple and revealing grammar" model
- Probabilistic hybrid grammars

Outline

- Syntax of natural language sentences
  - hybrid trees: phrase structure trees, dependency trees
- Ambiguity and parsing
  - probabilistic language models
- A particular "simple and revealing grammar" model
- Probabilistic hybrid grammars

probabilistic linear context-free rewriting systems (prob. LCFRS) (G, p)
[Vijay-Shanker, Weir, Joshi 87]
[Seki, Matsumura, Fujii, Kasami 91]

probabilistic linear context-free rewriting systems (prob. LCFRS) (G, p)
[Vijay-Shanker, Weir, Joshi 87]
[Seki, Matsumura, Fujii, Kasami 91]

LCFRS G:

$$
\begin{array}{lll}
S & \to & N \quad V \\
V & \to & N \quad V \\
V & \to & N \\
N & \to & \varepsilon \\
N & \to & \varepsilon \\
N & \to & \varepsilon
\end{array}
$$

probabilistic linear context-free rewriting systems (prob. LCFRS) (G, p)
[Vijay-Shanker, Weir, Joshi 87]
[Seki, Matsumura, Fujii, Kasami 91]

LCFRS G:

$$
\begin{array}{lcl}
S & \rightarrow & N(x)\ V(y_1, y_2) \\
V & \rightarrow & N(x)\ V(y_1, y_2) \\
V & \rightarrow & N(x) \\
N & \rightarrow & \varepsilon \\
N & \rightarrow & \varepsilon \\
N & \rightarrow & \varepsilon
\end{array}
$$

$\text{fanout}(S) = \text{fanout}(N) = 1$
$\text{fanout}(V) = 2$

probabilistic linear context-free rewriting systems (prob. LCFRS) (G, p)
[Vijay-Shanker, Weir, Joshi 87]
[Seki, Matsumura, Fujii, Kasami 91]

LCFRS G:

$$
\begin{array}{lcl}
S(x\ y_1\ \textbf{zag}\ y_2) & \rightarrow & N(x)\ V(y_1, y_2) \\
V(x\ y_1,\ \textbf{helpen}\ y_2) & \rightarrow & N(x)\ V(y_1, y_2) \\
V(x,\ \textbf{lezen}) & \rightarrow & N(x) \\
N(\textbf{Jan}) & \rightarrow & \varepsilon \\
N(\textbf{Piet}) & \rightarrow & \varepsilon \\
N(\textbf{Marie}) & \rightarrow & \varepsilon
\end{array}
$$

$\text{fanout}(S) = \text{fanout}(N) = 1$
$\text{fanout}(V) = 2$

no copying/deletion of variables

probabilistic linear context-free rewriting systems (prob. LCFRS) (G, p)
[Vijay-Shanker, Weir, Joshi 87]
[Seki, Matsumura, Fujii, Kasami 91]

LCFRS G:			probability assignment p:
$S(x\ y_1\ \textbf{zag}\ y_2)$	$\rightarrow$	$N(x)\ V(y_1, y_2)$	1
$V(x\ y_1,\ \textbf{helpen}\ y_2)$	$\rightarrow$	$N(x)\ V(y_1, y_2)$	0.5
$V(x,\ \textbf{lezen})$	$\rightarrow$	$N(x)$	0.5
$N(\textbf{Jan})$	$\rightarrow$	ε	0.33
$N(\textbf{Piet})$	$\rightarrow$	ε	0.33
$N(\textbf{Marie})$	$\rightarrow$	ε	0.33

$\text{fanout}(S) = \text{fanout}(N) = 1$
$\text{fanout}(V) = 2$

no copying/deletion of variables

probabilistic linear context-free rewriting systems (prob. LCFRS) (G, p)
[Vijay-Shanker, Weir, Joshi 87]
[Seki, Matsumura, Fujii, Kasami 91]

LCFRS G:			probability assignment p:
$S(x\ y_1\ \mathbf{zag}\ y_2)$	$\rightarrow$	$N(x)\ V(y_1, y_2)$	1
$V(x\ y_1,\ \mathbf{helpen}\ y_2)$	$\rightarrow$	$N(x)\ V(y_1, y_2)$	0.5
$V(x,\ \mathbf{lezen})$	$\rightarrow$	$N(x)$	0.5
$N(\mathbf{Jan})$	$\rightarrow$	ε	0.33
$N(\mathbf{Piet})$	$\rightarrow$	ε	0.33
$N(\mathbf{Marie})$	$\rightarrow$	ε	0.33

$\text{fanout}(S) = \text{fanout}(N) = 1$
$\text{fanout}(V) = 2$

no copying/deletion of variables
lexicalized LCFRS ("anchor")

probabilistic linear context-free rewriting systems (prob. LCFRS) (G, p)
[Vijay-Shanker, Weir, Joshi 87]
[Seki, Matsumura, Fujii, Kasami 91]

LCFRS G:			probability assignment p:
$S(x\ y_1\ \textbf{zag}\ y_2)$	$\rightarrow$	$N(x)\ V(y_1, y_2)$	1
$V(x\ y_1,\ \textbf{helpen}\ y_2)$	$\rightarrow$	$N(x)\ V(y_1, y_2)$	0.5
$V(x,\ \textbf{lezen})$	$\rightarrow$	$N(x)$	0.5
$N(\textbf{Jan})$	$\rightarrow$	ε	0.33
$N(\textbf{Piet})$	$\rightarrow$	ε	0.33
$N(\textbf{Marie})$	$\rightarrow$	ε	0.33

$\text{fanout}(S) = \text{fanout}(N) = 1$
$\text{fanout}(V) = 2$

no copying/deletion of variables
lexicalized LCFRS ("anchor")

CFG = LCFRS(fanout=1):
$A \rightarrow aBbC \quad \rightsquigarrow \quad A(a\ x_1\ b\ x_2) \rightarrow B(x_1)\ C(x_2)$

lexicalized LCFRS G:

$$
\begin{array}{lcl}
S(x \;\; y_1 \;\; \textbf{zag} \;\; y_2) & \rightarrow & N(x)\; V(y_1, y_2) \\
V(x \;\; y_1,\;\; \textbf{helpen} \;\; y_2) & \rightarrow & N(x)\; V(y_1, y_2) \\
V(x,\;\; \textbf{lezen}) & \rightarrow & N(x)
\end{array}
$$

$$
N(\textbf{Jan}) \rightarrow \varepsilon \qquad N(\textbf{Piet}) \rightarrow \varepsilon \qquad N(\textbf{Marie}) \rightarrow \varepsilon
$$

lexicalized LCFRS G:

$$
\begin{array}{lcl}
S(x\ \ y_1\ \ \textbf{zag}\ \ y_2) & \rightarrow & N(x)\ V(y_1, y_2) \\
V(x\ \ y_1,\ \ \textbf{helpen}\ \ y_2) & \rightarrow & N(x)\ V(y_1, y_2) \\
V(x,\ \ \textbf{lezen}) & \rightarrow & N(x)
\end{array}
$$

$$N(\textbf{Jan}) \rightarrow \varepsilon \quad N(\textbf{Piet}) \rightarrow \varepsilon \quad N(\textbf{Marie}) \rightarrow \varepsilon$$

$$S\,(\,\mathbf{J}_1\ \ \mathbf{P}_2\ \ \mathbf{M}_3\ \ \mathbf{z}_4\ \ \mathbf{h}_5\ \ \boldsymbol{\ell}_6\,)$$

lexicalized LCFRS G:

$$\boxed{S(x\ y_1\ \mathbf{zag}\ y_2) \rightarrow N(x)\ V(y_1, y_2)}$$
$$V(x\ y_1, \mathbf{helpen}\ y_2) \rightarrow N(x)\ V(y_1, y_2)$$
$$V(x, \mathbf{lezen}) \rightarrow N(x)$$
$$N(\mathbf{Jan}) \rightarrow \varepsilon \quad N(\mathbf{Piet}) \rightarrow \varepsilon \quad N(\mathbf{Marie}) \rightarrow \varepsilon$$

$$S(\mathbf{J}_1\ \mathbf{P}_2\ \mathbf{M}_3\ \mathbf{z}_4\ \mathbf{h}_5\ \boldsymbol{\ell}_6)$$

lexicalized LCFRS G:

$$\boxed{S(\boxed{x}\ \boxed{y_1}\ \boxed{\textbf{zag}}\ \boxed{y_2}) \quad\rightarrow\quad N(x)\ V(y_1, y_2)}$$

$$V(x\ \ y_1,\ \textbf{helpen}\ \ y_2) \quad\rightarrow\quad N(x)\ V(y_1, y_2)$$

$$V(x,\ \textbf{lezen}) \quad\rightarrow\quad N(x)$$

$$N(\textbf{Jan}) \rightarrow \varepsilon \qquad N(\textbf{Piet}) \rightarrow \varepsilon \qquad N(\textbf{Marie}) \rightarrow \varepsilon$$

$$S\ (\boxed{\mathbf{J}_1}\ \boxed{\mathbf{P}_2\ \ \mathbf{M}_3}\ \boxed{\mathbf{z}_4}\ \boxed{\mathbf{h}_5\ \ \boldsymbol{\ell}_6})$$

lexicalized LCFRS G:

$$S(\boxed{x}\ \boxed{y_1}\ \boxed{\textbf{zag}}\ \boxed{y_2}) \rightarrow N(\boxed{x})\ V(\boxed{y_1}, \boxed{y_2})$$

$$V(x\ \ y_1,\ \textbf{helpen}\ \ y_2) \rightarrow N(x)\ V(y_1, y_2)$$

$$V(x,\ \textbf{lezen}) \rightarrow N(x)$$

$$N(\textbf{Jan}) \rightarrow \varepsilon \quad N(\textbf{Piet}) \rightarrow \varepsilon \quad N(\textbf{Marie}) \rightarrow \varepsilon$$

$$S\,(\boxed{\mathbf{J}_1}\,\boxed{\mathbf{P}_2\ \ \mathbf{M}_3}\,\boxed{\mathbf{z}_4}\,\boxed{\mathbf{h}_5\ \ \boldsymbol{\ell}_6})$$

lexicalized LCFRS G:

$$\boxed{S(\boxed{x}\ \boxed{y_1}\ \boxed{\textbf{zag}}\ \boxed{y_2}) \quad \rightarrow \quad N(\boxed{x})\ V(\boxed{y_1}, \boxed{y_2})}$$

$$V(x\ \ y_1,\ \textbf{helpen}\ \ y_2) \quad \rightarrow \quad N(x)\ V(y_1, y_2)$$

$$V(x,\ \textbf{lezen}) \quad \rightarrow \quad N(x)$$

$$N(\textbf{Jan}) \rightarrow \varepsilon \qquad N(\textbf{Piet}) \rightarrow \varepsilon \qquad N(\textbf{Marie}) \rightarrow \varepsilon$$

$$S\,(\boxed{\mathbf{J}_1}\,\boxed{\mathbf{P}_2\ \ \mathbf{M}_3}\,\boxed{\mathbf{z}_4}\,\boxed{\mathbf{h}_5\ \ \boldsymbol{\ell}_6})$$

$$N\,(\boxed{\mathbf{J}_1}) \qquad\qquad V\,(\boxed{\mathbf{P}_2\ \ \mathbf{M}_3}, \boxed{\mathbf{h}_5\ \ \boldsymbol{\ell}_6})$$

lexicalized LCFRS G:

$$\boxed{S(\boxed{x}\ \boxed{y_1}\ \boxed{\textbf{zag}}\ \boxed{y_2}) \quad\to\quad N(\boxed{x})\ V(\boxed{y_1}, \boxed{y_2})}$$

$$V(x\ \ y_1,\ \textbf{helpen}\ \ y_2) \quad\to\quad N(x)\ V(y_1, y_2)$$

$$V(x,\ \textbf{lezen}) \quad\to\quad N(x)$$

$$N(\textbf{Jan}) \to \varepsilon \qquad N(\textbf{Piet}) \to \varepsilon \qquad N(\textbf{Marie}) \to \varepsilon$$

$$S\,(\boxed{\mathbf{J}_1}\,\boxed{\mathbf{P}_2\ \ \mathbf{M}_3}\,\boxed{\mathbf{z}_4}\,\boxed{\mathbf{h}_5\ \ \boldsymbol{\ell}_6})$$

$$N\,(\boxed{\mathbf{J}_1}) \qquad\qquad V\,(\boxed{\mathbf{P}_2\ \ \mathbf{M}_3}, \boxed{\mathbf{h}_5\ \ \boldsymbol{\ell}_6})$$

$$\boxed{\textbf{zag}}$$

(**dat**)

lexicalized LCFRS G:

$$S(x\ y_1\ \textbf{zag}\ y_2) \rightarrow N(x)\ V(y_1, y_2)$$

$$V(x\ y_1,\ \textbf{helpen}\ y_2) \rightarrow N(x)\ V(y_1, y_2)$$

$$V(x,\ \textbf{lezen}) \rightarrow N(x)$$

$$N(\textbf{Jan}) \rightarrow \varepsilon \quad N(\textbf{Piet}) \rightarrow \varepsilon \quad N(\textbf{Marie}) \rightarrow \varepsilon$$

$$S(\mathbf{J}_1\ \mathbf{P}_2\ \mathbf{M}_3\ \mathbf{z}_4\ \mathbf{h}_5\ \boldsymbol{\ell}_6)$$

$$N(\mathbf{J}_1) \qquad V(\mathbf{P}_2\ \mathbf{M}_3,\ \mathbf{h}_5\ \boldsymbol{\ell}_6)$$

(**dat**) **zag**

lexicalized LCFRS G:

$$
\begin{aligned}
S(x\ y_1\ \textbf{zag}\ y_2) &\rightarrow N(x)\ V(y_1, y_2)\\
V(x\ y_1, \textbf{helpen}\ y_2) &\rightarrow N(x)\ V(y_1, y_2)\\
V(x, \textbf{lezen}) &\rightarrow N(x)
\end{aligned}
$$

$\boxed{N(\textbf{Jan}) \rightarrow \varepsilon}$ $\quad N(\textbf{Piet}) \rightarrow \varepsilon \quad N(\textbf{Marie}) \rightarrow \varepsilon$

$S(\mathbf{J}_1\ \mathbf{P}_2\ \mathbf{M}_3\ \mathbf{z}_4\ \mathbf{h}_5\ \boldsymbol{\ell}_6)$

$N(\mathbf{J}_1)$ $\qquad$ $V(\mathbf{P}_2\ \mathbf{M}_3, \mathbf{h}_5\ \boldsymbol{\ell}_6)$

(**dat**) $\qquad$ **zag**

lexicalized LCFRS G:

$$S(x\ y_1\ \mathbf{zag}\ y_2) \rightarrow N(x)\ V(y_1, y_2)$$

$$V(x\ y_1,\ \mathbf{helpen}\ y_2) \rightarrow N(x)\ V(y_1, y_2)$$

$$V(x,\ \mathbf{lezen}) \rightarrow N(x)$$

$$\boxed{N(\boxed{\mathbf{Jan}}) \rightarrow \varepsilon} \quad N(\mathbf{Piet}) \rightarrow \varepsilon \quad N(\mathbf{Marie}) \rightarrow \varepsilon$$

$$S(\mathbf{J}_1\ \mathbf{P}_2\ \mathbf{M}_3\ \mathbf{z}_4\ \mathbf{h}_5\ \boldsymbol{\ell}_6)$$

$$N(\boxed{\mathbf{J}_1}) \qquad V(\mathbf{P}_2\ \mathbf{M}_3,\ \mathbf{h}_5\ \boldsymbol{\ell}_6)$$

(**dat**) $\boxed{\textbf{Jan}}$ **zag**

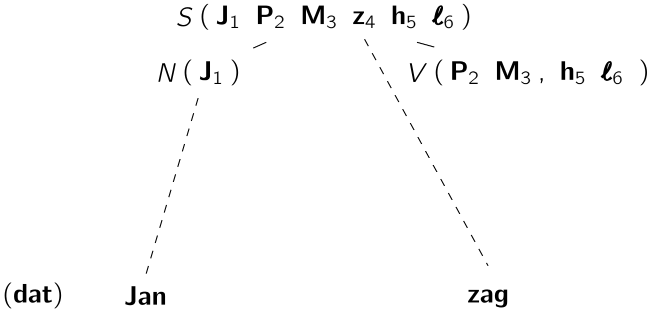
lexicalized LCFRS G:

$$S(x\ y_1\ \textbf{zag}\ y_2) \rightarrow N(x)\ V(y_1, y_2)$$

$$V(x\ y_1,\ \textbf{helpen}\ y_2) \rightarrow N(x)\ V(y_1, y_2)$$

$$V(x,\ \textbf{lezen}) \rightarrow N(x)$$

$$N(\textbf{Jan}) \rightarrow \varepsilon \quad N(\textbf{Piet}) \rightarrow \varepsilon \quad N(\textbf{Marie}) \rightarrow \varepsilon$$



lexicalized LCFRS G:

$$S(x\ \ y_1\ \ \textbf{zag}\ \ y_2) \rightarrow N(x)\ V(y_1, y_2)$$

$$\boxed{V(x\ \ y_1,\ \ \textbf{helpen}\ \ y_2) \rightarrow N(x)\ V(y_1, y_2)}$$

$$V(x,\ \ \textbf{lezen}) \rightarrow N(x)$$

$$N(\textbf{Jan}) \rightarrow \varepsilon \quad N(\textbf{Piet}) \rightarrow \varepsilon \quad N(\textbf{Marie}) \rightarrow \varepsilon$$

$$S(\mathbf{J}_1\ \ \mathbf{P}_2\ \ \mathbf{M}_3\ \ \mathbf{z}_4\ \ \mathbf{h}_5\ \ \boldsymbol{\ell}_6)$$

$$N(\mathbf{J}_1) \qquad V(\mathbf{P}_2\ \ \mathbf{M}_3,\ \ \mathbf{h}_5\ \ \boldsymbol{\ell}_6)$$

(**dat**) **Jan** **zag**

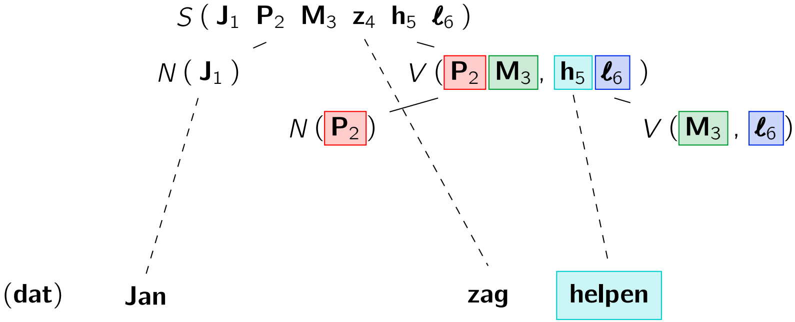
lexicalized LCFRS G:

$$S(x\ y_1\ \textbf{zag}\ y_2) \rightarrow N(x)\ V(y_1, y_2)$$

$$V(\boxed{x}\ \boxed{y_1},\ \boxed{\textbf{helpen}}\ \boxed{y_2}) \rightarrow N(\boxed{x})\ V(\boxed{y_1}, \boxed{y_2})$$

$$V(x,\ \textbf{lezen}) \rightarrow N(x)$$

$$N(\textbf{Jan}) \rightarrow \varepsilon \qquad N(\textbf{Piet}) \rightarrow \varepsilon \qquad N(\textbf{Marie}) \rightarrow \varepsilon$$



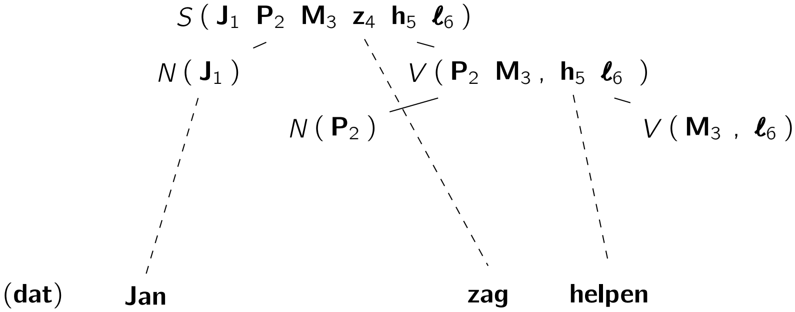
lexicalized LCFRS G:

$$S(x\ \ y_1\ \ \textbf{zag}\ \ y_2) \quad \rightarrow \quad N(x)\ V(y_1, y_2)$$

$$V(x\ \ y_1,\ \ \textbf{helpen}\ \ y_2) \quad \rightarrow \quad N(x)\ V(y_1, y_2)$$

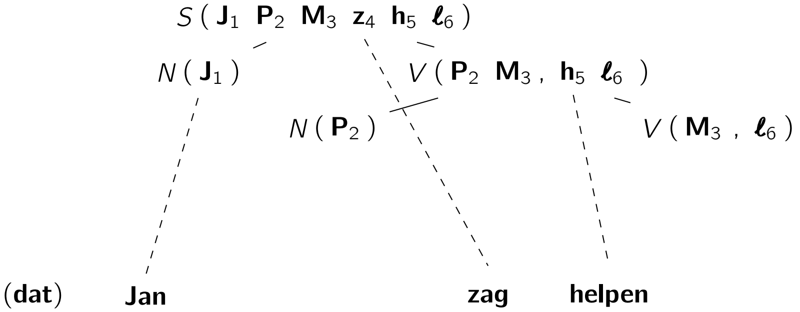
$$V(x,\ \ \textbf{lezen}) \quad \rightarrow \quad N(x)$$

$$N(\textbf{Jan}) \rightarrow \varepsilon \qquad N(\textbf{Piet}) \rightarrow \varepsilon \qquad N(\textbf{Marie}) \rightarrow \varepsilon$$



lexicalized LCFRS G:

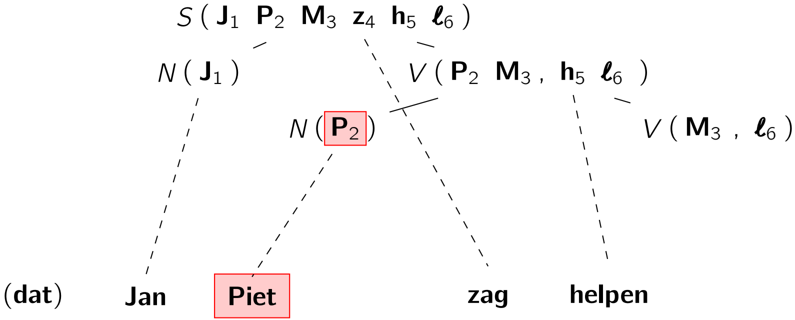
$$S(x\ y_1\ \textbf{zag}\ y_2) \rightarrow N(x)\ V(y_1, y_2)$$
$$V(x\ y_1,\ \textbf{helpen}\ y_2) \rightarrow N(x)\ V(y_1, y_2)$$
$$V(x,\ \textbf{lezen}) \rightarrow N(x)$$
$$N(\textbf{Jan}) \rightarrow \varepsilon \quad \boxed{N(\textbf{Piet}) \rightarrow \varepsilon} \quad N(\textbf{Marie}) \rightarrow \varepsilon$$



lexicalized LCFRS G:

$$
\begin{aligned}
S(x\ y_1\ \textbf{zag}\ y_2) &\rightarrow N(x)\ V(y_1, y_2)\\
V(x\ y_1,\ \textbf{helpen}\ y_2) &\rightarrow N(x)\ V(y_1, y_2)\\
V(x,\ \textbf{lezen}) &\rightarrow N(x)
\end{aligned}
$$

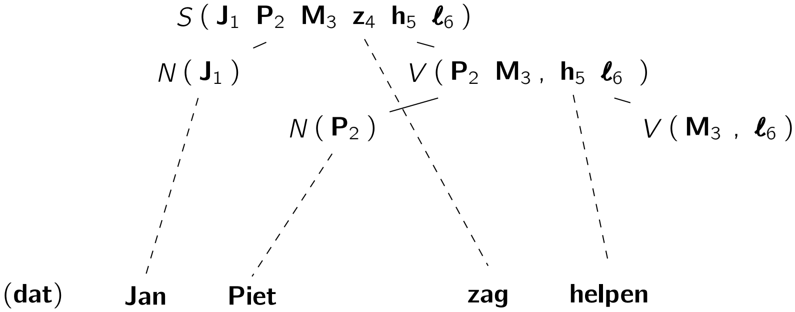
$N(\textbf{Jan}) \rightarrow \varepsilon \quad \boxed{N(\boxed{\textbf{Piet}}) \rightarrow \varepsilon} \quad N(\textbf{Marie}) \rightarrow \varepsilon$



lexicalized LCFRS G:

$$
\begin{array}{lcl}
S(x\ \ y_1\ \ \textbf{zag}\ \ y_2) & \rightarrow & N(x)\ V(y_1, y_2) \\
V(x\ \ y_1,\ \ \textbf{helpen}\ \ y_2) & \rightarrow & N(x)\ V(y_1, y_2) \\
V(x,\ \ \textbf{lezen}) & \rightarrow & N(x)
\end{array}
$$

$$N(\textbf{Jan}) \rightarrow \varepsilon \quad N(\textbf{Piet}) \rightarrow \varepsilon \quad N(\textbf{Marie}) \rightarrow \varepsilon$$



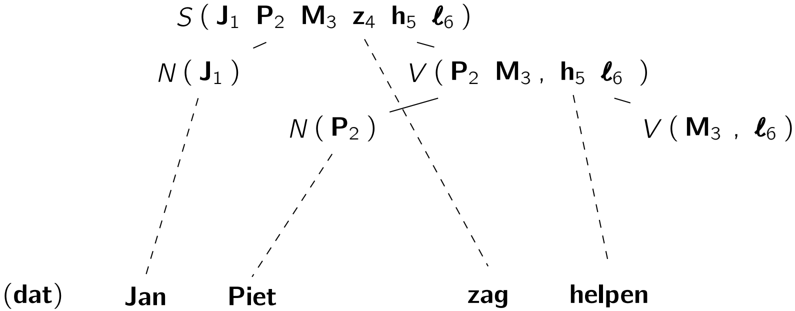
lexicalized LCFRS G:

$$S(x\ y_1\ \textbf{zag}\ y_2) \rightarrow N(x)\ V(y_1, y_2)$$

$$V(x\ y_1, \textbf{helpen}\ y_2) \rightarrow N(x)\ V(y_1, y_2)$$

$$\boxed{V(x, \textbf{lezen}) \rightarrow N(x)}$$

$$N(\textbf{Jan}) \rightarrow \varepsilon \quad N(\textbf{Piet}) \rightarrow \varepsilon \quad N(\textbf{Marie}) \rightarrow \varepsilon$$



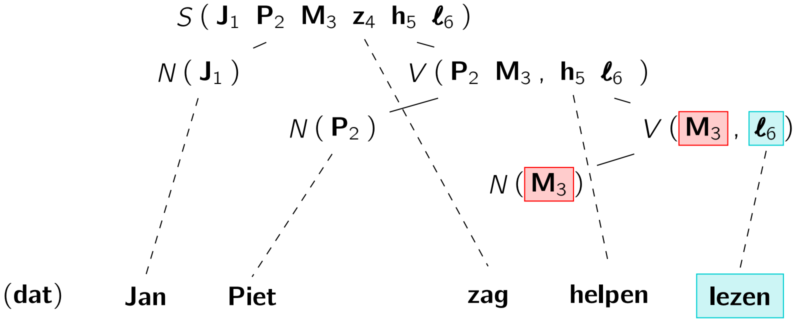
lexicalized LCFRS G:

$$S(x\ y_1\ \mathbf{zag}\ y_2) \rightarrow N(x)\ V(y_1, y_2)$$

$$V(x\ y_1,\ \mathbf{helpen}\ y_2) \rightarrow N(x)\ V(y_1, y_2)$$

$$\boxed{V(\boxed{x},\ \boxed{\mathbf{lezen}}) \rightarrow N(\boxed{x})}$$

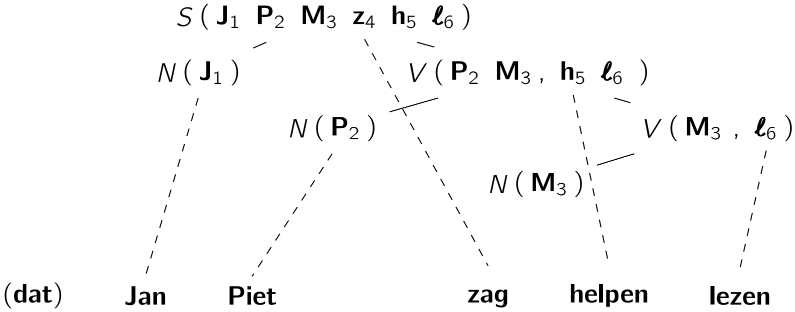
$$N(\mathbf{Jan}) \rightarrow \varepsilon \quad N(\mathbf{Piet}) \rightarrow \varepsilon \quad N(\mathbf{Marie}) \rightarrow \varepsilon$$



lexicalized LCFRS G:

$$
\begin{array}{lcl}
S(x\ \ y_1\ \ \textbf{zag}\ \ y_2) & \rightarrow & N(x)\ V(y_1, y_2) \\
V(x\ \ y_1,\ \ \textbf{helpen}\ \ y_2) & \rightarrow & N(x)\ V(y_1, y_2) \\
V(x,\ \ \textbf{lezen}) & \rightarrow & N(x)
\end{array}
$$

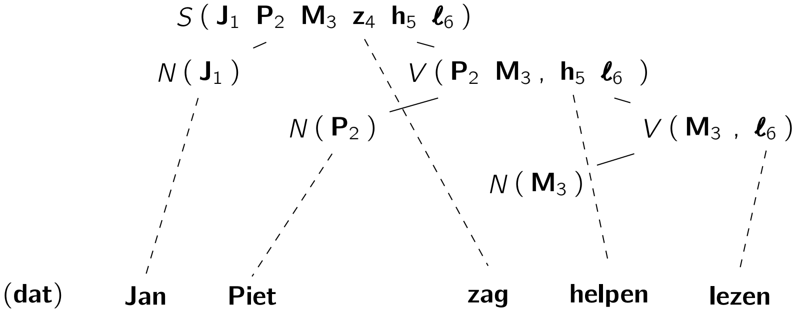
$$
N(\textbf{Jan}) \rightarrow \varepsilon \quad N(\textbf{Piet}) \rightarrow \varepsilon \quad N(\textbf{Marie}) \rightarrow \varepsilon
$$



lexicalized LCFRS G:

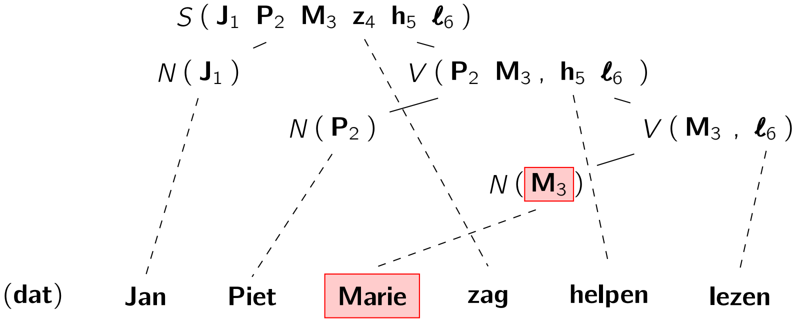
$$
\begin{array}{lcl}
S(x\ \ y_1\ \ \mathbf{zag}\ \ y_2) & \rightarrow & N(x)\ V(y_1, y_2) \\
V(x\ \ y_1,\ \ \mathbf{helpen}\ \ y_2) & \rightarrow & N(x)\ V(y_1, y_2) \\
V(x,\ \ \mathbf{lezen}) & \rightarrow & N(x)
\end{array}
$$

$$
N(\mathbf{Jan}) \rightarrow \varepsilon \qquad N(\mathbf{Piet}) \rightarrow \varepsilon \qquad \boxed{N(\mathbf{Marie}) \rightarrow \varepsilon}
$$



lexicalized LCFRS G:

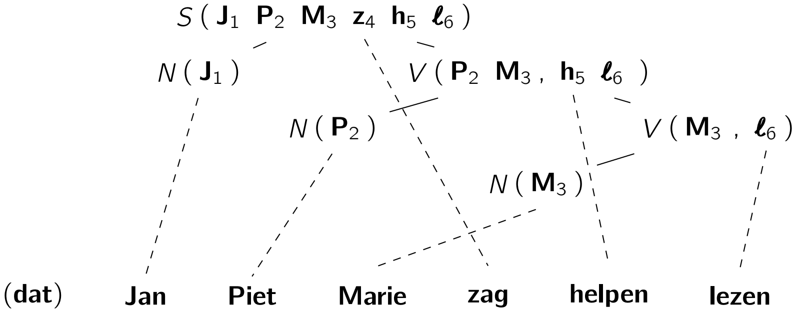
$$
\begin{array}{lcl}
S(x\ \ y_1\ \ \textbf{zag}\ \ y_2) & \rightarrow & N(x)\ V(y_1, y_2) \\
V(x\ \ y_1,\ \ \textbf{helpen}\ \ y_2) & \rightarrow & N(x)\ V(y_1, y_2) \\
V(x,\ \ \textbf{lezen}) & \rightarrow & N(x) \\
N(\textbf{Jan}) \rightarrow \varepsilon \quad N(\textbf{Piet}) \rightarrow \varepsilon & & \boxed{N(\boxed{\textbf{Marie}}) \rightarrow \varepsilon}
\end{array}
$$



lexicalized LCFRS G:

$$
\begin{array}{lcl}
S(x\ \ y_1\ \ \textbf{zag}\ \ y_2) & \rightarrow & N(x)\ V(y_1, y_2) \\
V(x\ \ y_1,\ \ \textbf{helpen}\ \ y_2) & \rightarrow & N(x)\ V(y_1, y_2) \\
V(x,\ \ \textbf{lezen}) & \rightarrow & N(x)
\end{array}
$$

$$N(\textbf{Jan}) \rightarrow \varepsilon \quad N(\textbf{Piet}) \rightarrow \varepsilon \quad N(\textbf{Marie}) \rightarrow \varepsilon$$



lexicalized LCFRS G:

$$S(x\ \ y_1\ \ \textbf{zag}\ \ y_2) \quad \rightarrow \quad N(x)\ V(y_1, y_2)$$

$$V(x\ \ y_1,\ \ \textbf{helpen}\ \ y_2) \quad \rightarrow \quad N(x)\ V(y_1, y_2)$$

$$V(x,\ \ \textbf{lezen}) \quad \rightarrow \quad N(x)$$

$$N(\textbf{Jan}) \rightarrow \varepsilon \qquad N(\textbf{Piet}) \rightarrow \varepsilon \qquad N(\textbf{Marie}) \rightarrow \varepsilon$$

parsing($\textbf{Jan}_1$ $\textbf{Piet}_2$ $\textbf{Marie}_3$ $\textbf{zag}_4$ $\textbf{helpen}_5$ $\textbf{lezen}_6$) =

$$S(\mathbf{J}_1\ \ \mathbf{P}_2\ \ \mathbf{M}_3\ \ \mathbf{z}_4\ \ \mathbf{h}_5\ \ \boldsymbol{\ell}_6)$$

$$N(\mathbf{J}_1) \qquad V(\mathbf{P}_2\ \ \mathbf{M}_3,\ \ \mathbf{h}_5\ \ \boldsymbol{\ell}_6)$$

$$N(\mathbf{P}_2) \qquad V(\mathbf{M}_3,\ \ \boldsymbol{\ell}_6)$$

$$N(\mathbf{M}_3)$$

projection to hybrid tree

E: natural language $\mathcal{H}$: set of dependency trees

parsing : $E \to \mathcal{H}$

$$\text{parsing}(e) = \mathrm{proj}\Big(\mathrm{argmax}_{\substack{d \in D_G: \\ \mathrm{str}(\mathrm{proj}(d)) = e}} P(d \mid (G, p))\Big)$$

where

- (G, p): probabilistic lexicalized LCFRS
  (D_G: set of proof trees of G)
- $\mathrm{proj} : D_G \to \mathcal{H}$
- $P(r_1 \ldots r_n \mid (G, p)) = p(r_1) \cdot \ldots \cdot p(r_n)$

Outline

- Syntax of natural language sentences
  - hybrid trees: phrase structure trees, dependency trees
- Ambiguity and parsing
  - probabilistic language models
- A particular "simple and revealing grammar" model
  - probabilistic LCFRS
- Probabilistic hybrid grammars

grammar formalisms	phrase structure trees		dependency trees		parsing compl.
	cont.	discont.	proj.	non-proj.	

grammar formalisms	phrase structure trees		dependency trees		parsing compl.
	cont.	discont.	proj.	non-proj.	
REG/HMM	-	-	-	-	$\mathcal{O}(n)$

grammar formalisms	phrase structure trees		dependency trees		parsing compl.
	cont.	discont.	proj.	non-proj.	
REG/HMM	-	-	-	-	$\mathcal{O}(n)$
CFG	**x**	-	**x**	-	$\mathcal{O}(n^3)$ (CNF)

grammar formalisms	phrase structure trees		dependency trees		parsing compl.
	cont.	discont.	proj.	non-proj.	
REG/HMM	-	-	-	-	$\mathcal{O}(n)$
CFG	x	-	x	-	$\mathcal{O}(n^3)$ (CNF)
LCFRS	x	x	x	x	$\mathcal{O}(n^{3 \cdot \mathrm{fanout}(G)})$ (binar.)

grammar formalisms	phrase structure trees		dependency trees		parsing compl.
	cont.	discont.	proj.	non-proj.	
REG/HMM	-	-	-	-	$\mathcal{O}(n)$
CFG	x	-	x	-	$\mathcal{O}(n^3)$ (CNF)
LCFRS	x	x	x	x	$\mathcal{O}(n^{3 \cdot \mathrm{fanout}(G)})$ (binar.)

trade-off: "richness" of syntactic structures versus parsing complexity

grammar formalisms	phrase structure trees		dependency trees		parsing compl.
	cont.	discont.	proj.	non-proj.	
REG/HMM	-	-	-	-	$\mathcal{O}(n)$
CFG	x	-	x	-	$\mathcal{O}(n^3)$ (CNF)
LCFRS	x	x	x	x	$\mathcal{O}(n^{3 \cdot \text{fanout}(G)})$ (binar.)

trade-off: "richness" of syntactic structures versus parsing complexity

Idea:
split generation of syntactic structures
from generation of strings

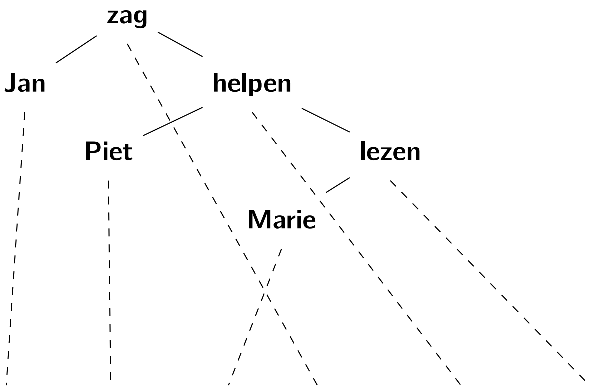
Outline

- ...
- Probabilistic hybrid grammars

[Nederhof, HV 14]
Hybrid Grammars for Discontinuous Parsing.
COLING 2014

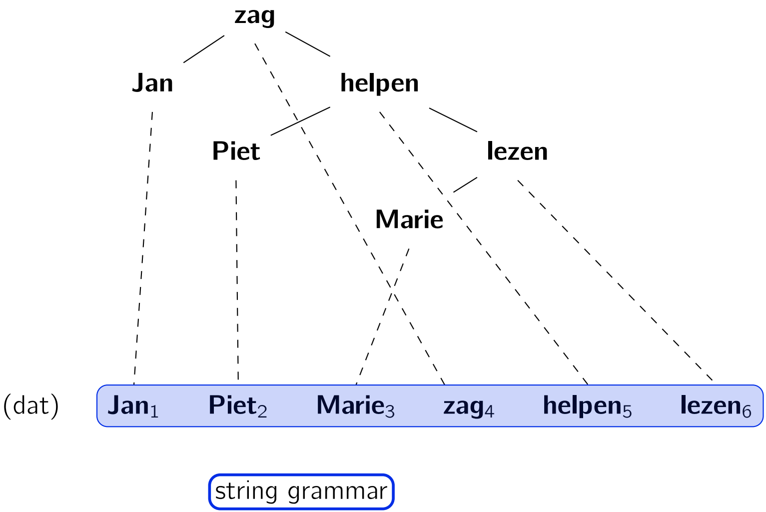
[Gebhardt, Nederhof, HV 17]
Hybrid grammars for parsing of discontinuous phrase structures and non-projective dependency structures,
accepted for publication 2016, Computational Linguistics.

hybrid tree:

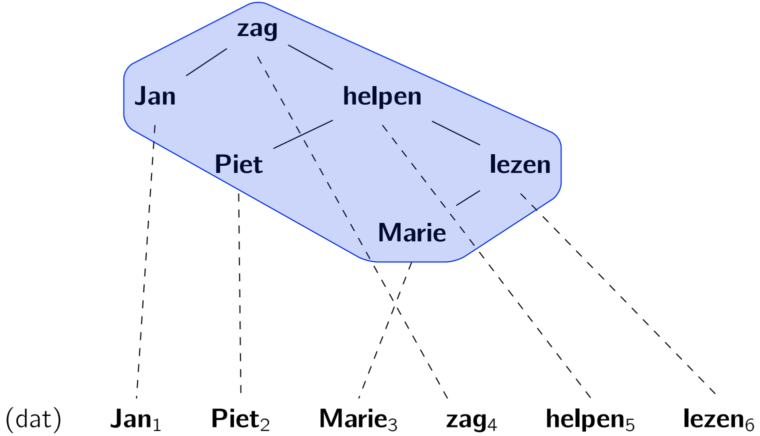


(dat) **Jan**$_1$ **Piet**$_2$ **Marie**$_3$ **zag**$_4$ **helpen**$_5$ **lezen**$_6$

hybrid tree:


zag
Jan
helpen
Piet
lezen
Marie
(dat)
Jan1
Piet2
Marie3
zag4
helpen5
lezen6
string grammar

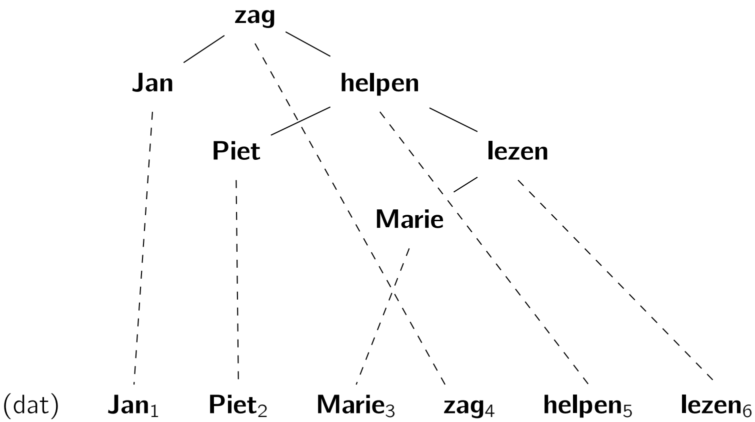
hybrid tree:



string grammar

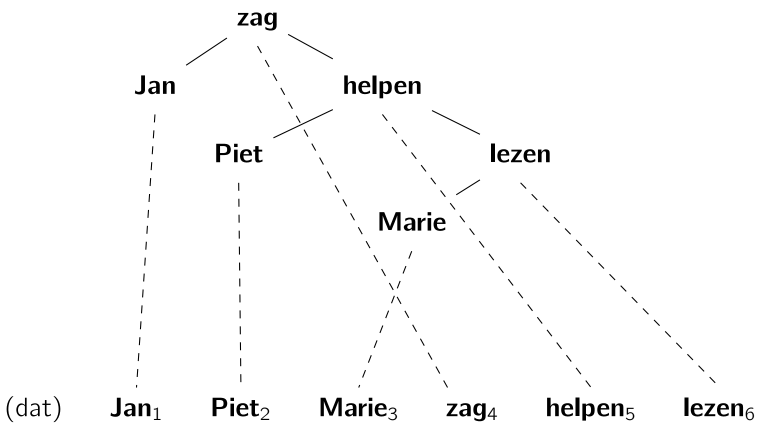
+ tree grammar

hybrid tree:



string grammar + synchronization + tree grammar

hybrid tree:



hybrid grammar = string grammar + synchronization + tree grammar

hybrid grammar = string grammar + synchronization + tree grammar

string grammar:

- regular grammars / finite-state automata
- context-free grammars
- macro grammars [Fischer 68]
- linear context-free rewriting systems (LCFRS)

tree grammar:

- regular tree grammars [Brainerd 69]
- context-free tree grammars [Rounds 70; Engelfriet, Schmidt 77]
- simple definite clause programs (sDCP)
  [Deransart, Maluszynski 85]

hybrid grammar = string grammar + synchronization + tree grammar

string grammar:

- regular grammars / finite-state automata
- context-free grammars
- macro grammars [Fischer 68]
- linear context-free rewriting systems (LCFRS)

tree grammar:

- regular tree grammars [Brainerd 69]
- context-free tree grammars [Rounds 70; Engelfriet, Schmidt 77]
- simple definite clause programs (sDCP) [Deransart, Maluszynski 85]

(LCFRS,sDCP)-hybrid grammars $\rightsquigarrow$ hybrid grammars

Outline

- ...
- Probabilistic hybrid grammars
  - simple definite clause programs (sDCP)
  - (LCFRS,sDCP)-hybrid grammars
  - parsing with hybrid grammars
  - how to obtain a hybrid grammar for E?
  - experiments

simple definite clause programs (sDCPs): $\approx$ attribute grammars [Knuth 68]

$$S \quad \rightarrow \quad A \quad C$$

simple definite clause programs (sDCPs): ≈ attribute grammars [Knuth 68]

$$S\ \boxed{z_1} \quad \rightarrow \quad A\ \boxed{z_1}\quad C\ \boxed{z_2}$$

synthesized attribute

simple definite clause programs (sDCPs): $\approx$ attribute grammars [Knuth 68]

$$S\,\boxed{z_1} \quad\to\quad \boxed{}\,A\,\boxed{z_1}\quad C\,\boxed{z_2}$$

inherited attribute

synthesized attribute

simple definite clause programs (sDCPs): $\approx$ attribute grammars [Knuth 68]

$$S\ \boxed{z_1} \quad \rightarrow \quad \boxed{z_2}\ A\ \boxed{z_1} \quad C\ \boxed{z_2}$$

inherited attribute

synthesized attribute

simple definite clause programs (sDCPs): ≈ attribute grammars [Knuth 68]

$$S\ \boxed{z_1} \quad \rightarrow \quad \boxed{z_2}\ A\ \boxed{z_1} \quad C\ \boxed{z_2}$$

simple definite clause programs (sDCPs): $\approx$ attribute grammars [Knuth 68]

$$S\ \boxed{z_1} \quad\rightarrow\quad \boxed{z_2}\ A\ \boxed{z_1} \quad C\ \boxed{z_2}$$

$$\boxed{z_1}\ A\ \boxed{\mathbf{h}(z_2, z_1)} \quad\rightarrow\quad B\ \boxed{z_2}$$

simple definite clause programs (sDCPs): ≈ attribute grammars [Knuth 68]

$$
\begin{array}{lcl}
S\ \boxed{z_1} & \rightarrow & \boxed{z_2}\ A\ \boxed{z_1}\quad C\ \boxed{z_2} \\
\boxed{z_1}\ A\ \boxed{\mathbf{h}(z_2, z_1)} & \rightarrow & B\ \boxed{z_2} \\
C\ \boxed{\boldsymbol{\ell}(z_1)} & \rightarrow & D\ \boxed{z_1}
\end{array}
\qquad
\begin{array}{lcl}
B\ \boxed{\mathbf{P}} & \rightarrow & \varepsilon \\
D\ \boxed{\mathbf{M}} & \rightarrow & \varepsilon
\end{array}
$$

simple definite clause programs (sDCPs): ≈ attribute grammars [Knuth 68]

$$S\ \boxed{z_1} \;\rightarrow\; \boxed{z_2}\ A\ \boxed{z_1} \quad C\ \boxed{z_2}$$

$$\boxed{z_1}\ A\ \boxed{\mathbf{h}(z_2, z_1)} \;\rightarrow\; B\ \boxed{z_2} \qquad\qquad B\ \boxed{\mathbf{P}} \;\rightarrow\; \varepsilon$$

$$C\ \boxed{\boldsymbol{\ell}(z_1)} \;\rightarrow\; D\ \boxed{z_1} \qquad\qquad D\ \boxed{\mathbf{M}} \;\rightarrow\; \varepsilon$$

$$S\ \boxed{\mathbf{h}(\mathbf{P}, \boldsymbol{\ell}(\mathbf{M}))}$$

simple definite clause programs (sDCPs): $\approx$ attribute grammars [Knuth 68]

$$S\ \boxed{z_1} \rightarrow \boxed{z_2}\ A\ \boxed{z_1}\quad C\ \boxed{z_2}$$

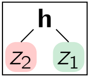
$$\boxed{z_1}\ A\ \boxed{\mathbf{h}(z_2, z_1)} \rightarrow B\ \boxed{z_2} \qquad\qquad B\ \boxed{\mathbf{P}} \rightarrow \varepsilon$$

$$C\ \boxed{\boldsymbol{\ell}(z_1)} \rightarrow D\ \boxed{z_1} \qquad\qquad D\ \boxed{\mathbf{M}} \rightarrow \varepsilon$$

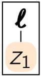
$$S\ \boxed{\mathbf{h}(\mathbf{P}, \boldsymbol{\ell}(\mathbf{M}))} \Rightarrow \boxed{}\ A\ \boxed{}\quad C\ \boxed{}$$

simple definite clause programs (sDCPs): $\approx$ attribute grammars [Knuth 68]

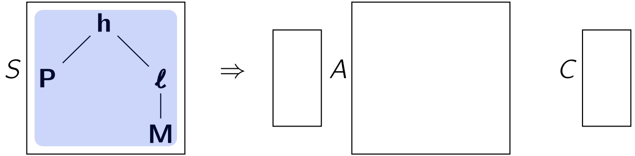
$$S\ \boxed{z_1} \rightarrow \boxed{z_2}\ A\ \boxed{z_1}\quad C\ \boxed{z_2}$$

$\boxed{z_1}\ A$  $\rightarrow B\ \boxed{z_2}$

$B\ \boxed{\mathbf{P}} \rightarrow \varepsilon$

C  $\rightarrow D\ \boxed{z_1}$

$D\ \boxed{\mathbf{M}} \rightarrow \varepsilon$

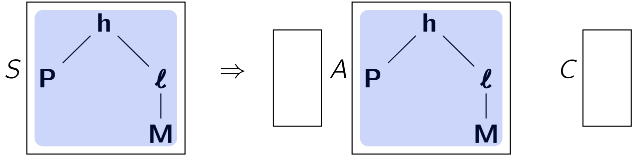


simple definite clause programs (sDCPs): $\approx$ attribute grammars [Knuth 68]

$$S\ \boxed{z_1} \rightarrow \boxed{z_2}\ A\ \boxed{z_1}\quad C\ \boxed{z_2}$$

$$\boxed{z_1}\ A\ \boxed{\mathbf{h}(z_2, z_1)} \rightarrow B\ \boxed{z_2} \qquad\qquad B\ \boxed{\mathbf{P}} \rightarrow \varepsilon$$

$$C\ \boxed{\boldsymbol{\ell}(z_1)} \rightarrow D\ \boxed{z_1} \qquad\qquad D\ \boxed{\mathbf{M}} \rightarrow \varepsilon$$



simple definite clause programs (sDCPs): ≈ attribute grammars [Knuth 68]

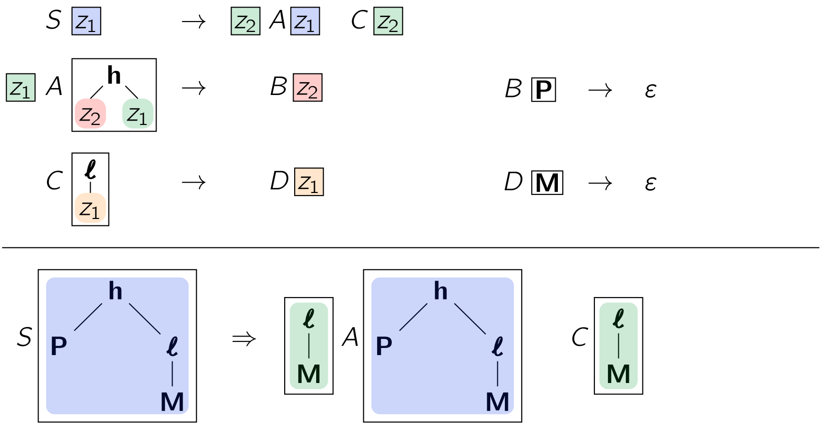
$$S\,\boxed{z_1} \rightarrow \boxed{z_2}\,A\,\boxed{z_1} \quad C\,\boxed{z_2}$$

$$\boxed{z_1}\,A\,\boxed{\mathbf{h}(z_2, z_1)} \rightarrow B\,\boxed{z_2} \qquad\qquad B\,\boxed{\mathbf{P}} \rightarrow \varepsilon$$

$$C\,\boxed{\boldsymbol{\ell}(z_1)} \rightarrow D\,\boxed{z_1} \qquad\qquad D\,\boxed{\mathbf{M}} \rightarrow \varepsilon$$

$$S\,\boxed{\mathbf{h}(\mathbf{P}, \boldsymbol{\ell}(\mathbf{M}))} \Rightarrow \boxed{\boldsymbol{\ell}(\mathbf{M})}\,A\,\boxed{\mathbf{h}(\mathbf{P}, \boldsymbol{\ell}(\mathbf{M}))} \quad C\,\boxed{\boldsymbol{\ell}(\mathbf{M})}$$

simple definite clause programs (sDCPs): ≈ attribute grammars [Knuth 68]

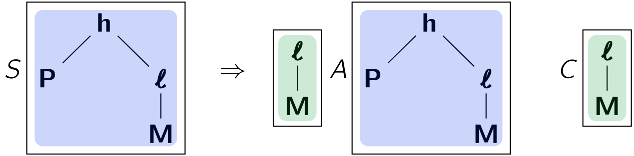


simple definite clause programs (sDCPs): $\approx$ attribute grammars [Knuth 68]

$$S\ \boxed{z_1} \rightarrow \boxed{z_2}\ A\ \boxed{z_1}\ \ C\ \boxed{z_2}$$

$$\boxed{z_1}\ A\ \boxed{\mathbf{h}(z_2, z_1)} \rightarrow B\ \boxed{z_2} \qquad\qquad B\ \boxed{\mathbf{P}} \rightarrow \varepsilon$$

$$C\ \boxed{\boldsymbol{\ell}(z_1)} \rightarrow D\ \boxed{z_1} \qquad\qquad D\ \boxed{\mathbf{M}} \rightarrow \varepsilon$$



simple definite clause programs (sDCPs): ≈ attribute grammars [Knuth 68]

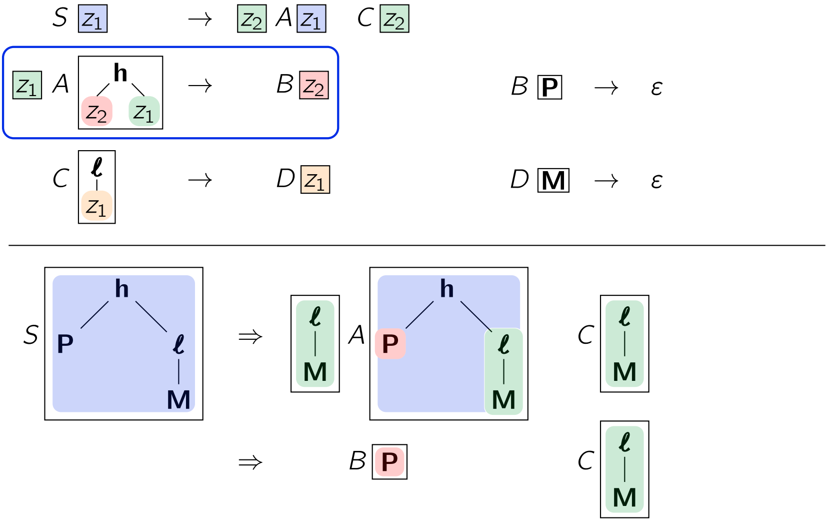
$$S\ \boxed{z_1} \rightarrow \boxed{z_2}\ A\ \boxed{z_1}\quad C\ \boxed{z_2}$$

$$\boxed{z_1}\ A\ \boxed{\mathbf{h}(z_2, z_1)} \rightarrow B\ \boxed{z_2} \qquad\qquad B\ \boxed{\mathbf{P}} \rightarrow \varepsilon$$

$$C\ \boxed{\boldsymbol{\ell}(z_1)} \rightarrow D\ \boxed{z_1} \qquad\qquad D\ \boxed{\mathbf{M}} \rightarrow \varepsilon$$

$$S\ \boxed{\mathbf{h}(\mathbf{P}, \boldsymbol{\ell}(\mathbf{M}))} \Rightarrow \boxed{\boldsymbol{\ell}(\mathbf{M})}\ A\ \boxed{\mathbf{h}(\mathbf{P}, \boldsymbol{\ell}(\mathbf{M}))}\quad C\ \boxed{\boldsymbol{\ell}(\mathbf{M})}$$

simple definite clause programs (sDCPs): ≈ attribute grammars [Knuth 68]



simple definite clause programs (sDCPs): ≈ attribute grammars [Knuth 68]

$$S\ \boxed{z_1} \to \boxed{z_2}\ A\ \boxed{z_1}\quad C\ \boxed{z_2}$$

$$\boxed{z_1}\ A\ \boxed{h(z_2, z_1)} \to B\ \boxed{z_2} \qquad\qquad B\ \boxed{\mathbf{P}} \to \varepsilon$$

$$C\ \boxed{\ell(z_1)} \to D\ \boxed{z_1} \qquad\qquad D\ \boxed{\mathbf{M}} \to \varepsilon$$

$$S\ \boxed{h(\mathbf{P}, \ell(\mathbf{M}))} \Rightarrow \boxed{\ell(\mathbf{M})}\ A\ \boxed{h(\mathbf{P}, \ell(\mathbf{M}))}\quad C\ \boxed{\ell(\mathbf{M})}$$

$$\Rightarrow B\ \boxed{\mathbf{P}}\quad C\ \boxed{\ell(\mathbf{M})}$$

simple definite clause programs (sDCPs): $\approx$ attribute grammars [Knuth 68]

$$S\ \boxed{z_1} \quad \rightarrow \quad \boxed{z_2}\ A\ \boxed{z_1} \quad C\ \boxed{z_2}$$

$$\boxed{z_1}\ A\ \boxed{\mathbf{h}(z_2, z_1)} \quad \rightarrow \quad B\ \boxed{z_2} \qquad\qquad B\ \boxed{\mathbf{P}} \quad \rightarrow \quad \varepsilon$$

$$C\ \boxed{\boldsymbol{\ell}(z_1)} \quad \rightarrow \quad D\ \boxed{z_1} \qquad\qquad D\ \boxed{\mathbf{M}} \quad \rightarrow \quad \varepsilon$$

$$S\ \boxed{\mathbf{h}(\mathbf{P}, \boldsymbol{\ell}(\mathbf{M}))} \quad \Rightarrow^2 \quad B\ \boxed{\mathbf{P}} \qquad C\ \boxed{\boldsymbol{\ell}(\mathbf{M})}$$

simple definite clause programs (sDCPs): ≈ attribute grammars [Knuth 68]

$$S\ \boxed{z_1} \rightarrow \boxed{z_2}\ A\ \boxed{z_1}\quad C\ \boxed{z_2}$$

$$\boxed{z_1}\ A\ \boxed{\mathbf{h}(z_2, z_1)} \rightarrow B\ \boxed{z_2} \qquad\qquad \boxed{B\ \boxed{\mathbf{P}} \rightarrow \varepsilon}$$

$$C\ \boxed{\boldsymbol{\ell}(z_1)} \rightarrow D\ \boxed{z_1} \qquad\qquad D\ \boxed{\mathbf{M}} \rightarrow \varepsilon$$

$$S\ \boxed{\mathbf{h}(\mathbf{P}, \boldsymbol{\ell}(\mathbf{M}))} \Rightarrow^2 B\ \boxed{\mathbf{P}} \qquad C\ \boxed{\boldsymbol{\ell}(\mathbf{M})}$$

simple definite clause programs (sDCPs): ≈ attribute grammars [Knuth 68]

$$S\,\boxed{z_1} \;\rightarrow\; \boxed{z_2}\,A\,\boxed{z_1}\quad C\,\boxed{z_2}$$

$$\boxed{z_1}\,A\,\boxed{\mathbf{h}(z_2, z_1)} \;\rightarrow\; B\,\boxed{z_2} \qquad\qquad B\,\boxed{\mathbf{P}} \;\rightarrow\; \varepsilon$$

$$C\,\boxed{\boldsymbol{\ell}(z_1)} \;\rightarrow\; D\,\boxed{z_1} \qquad\qquad D\,\boxed{\mathbf{M}} \;\rightarrow\; \varepsilon$$

$$S\,\boxed{\mathbf{h}(\mathbf{P}, \boldsymbol{\ell}(\mathbf{M}))} \;\Rightarrow^3\; C\,\boxed{\boldsymbol{\ell}(\mathbf{M})}$$

simple definite clause programs (sDCPs): ≈ attribute grammars [Knuth 68]

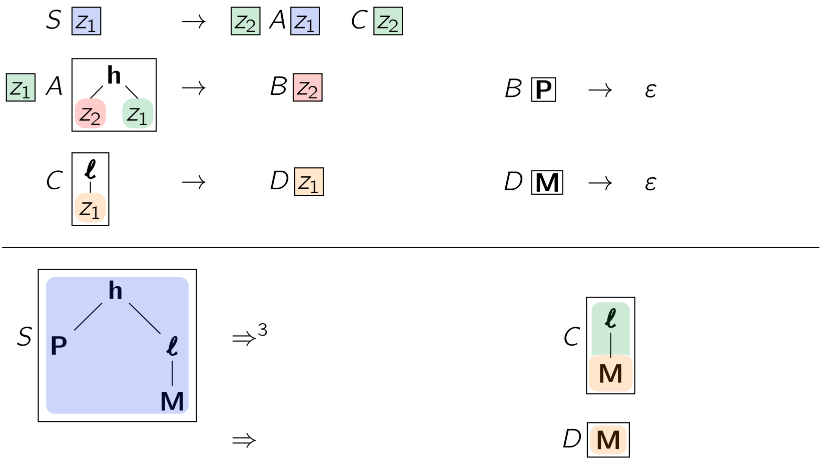
$$S\ \boxed{z_1} \rightarrow \boxed{z_2}\ A\ \boxed{z_1}\quad C\ \boxed{z_2}$$

$$\boxed{z_1}\ A\ \boxed{\mathbf{h}(z_2, z_1)} \rightarrow B\ \boxed{z_2} \qquad\qquad B\ \boxed{\mathbf{P}} \rightarrow \varepsilon$$

$$C\ \boxed{\boldsymbol{\ell}(z_1)} \rightarrow D\ \boxed{z_1} \qquad\qquad D\ \boxed{\mathbf{M}} \rightarrow \varepsilon$$

$$S\ \boxed{\mathbf{h}(\mathbf{P}, \boldsymbol{\ell}(\mathbf{M}))} \Rightarrow^3 \qquad C\ \boxed{\boldsymbol{\ell}(\mathbf{M})}$$

simple definite clause programs (sDCPs): $\approx$ attribute grammars [Knuth 68]



simple definite clause programs (sDCPs): $\approx$ attribute grammars [Knuth 68]

$$S\,\boxed{z_1} \;\rightarrow\; \boxed{z_2}\,A\,\boxed{z_1}\quad C\,\boxed{z_2}$$

$$\boxed{z_1}\,A\,\boxed{\mathbf{h}(z_2, z_1)} \;\rightarrow\; B\,\boxed{z_2} \qquad\qquad B\,\boxed{\mathbf{P}} \;\rightarrow\; \varepsilon$$

$$C\,\boxed{\boldsymbol{\ell}(z_1)} \;\rightarrow\; D\,\boxed{z_1} \qquad\qquad D\,\boxed{\mathbf{M}} \;\rightarrow\; \varepsilon$$

$$S\,\boxed{\mathbf{h}(\mathbf{P}, \boldsymbol{\ell}(\mathbf{M}))} \;\Rightarrow^4\; D\,\boxed{\mathbf{M}}$$

simple definite clause programs (sDCPs): ≈ attribute grammars [Knuth 68]

$$S\ \boxed{z_1} \rightarrow \boxed{z_2}\ A\ \boxed{z_1}\quad C\ \boxed{z_2}$$

$$\boxed{z_1}\ A\ \boxed{\mathbf{h}(z_2, z_1)} \rightarrow B\ \boxed{z_2} \qquad\qquad B\ \boxed{\mathbf{P}} \rightarrow \varepsilon$$

$$C\ \boxed{\boldsymbol{\ell}(z_1)} \rightarrow D\ \boxed{z_1} \qquad\qquad \boxed{D\ \boxed{\mathbf{M}} \rightarrow \varepsilon}$$

$$S\ \boxed{\mathbf{h}(\mathbf{P}, \boldsymbol{\ell}(\mathbf{M}))} \Rightarrow^4 D\ \boxed{\mathbf{M}}$$

$$\Rightarrow \varepsilon$$

(LCFRS,sDCP)-hybrid grammar:

$$S\,\boxed{z_1} \rightarrow \boxed{z_2}\,A\,\boxed{z_1}\;C\,\boxed{z_2}$$
$$S(x_1\,x_3\,x_2\,x_4) \rightarrow A(x_1, x_2)\;C(x_3, x_4)$$

$$\boxed{z_1}\,A\,\boxed{\mathbf{h}(z_2\ z_1)} \rightarrow B\,\boxed{z_2}$$
$$A(x_1, \mathbf{h}) \rightarrow B(x_1)$$

$$C\,\boxed{\boldsymbol{\ell}(z_1)} \rightarrow D\,\boxed{z_1}$$
$$C(x_1, \boldsymbol{\ell}) \rightarrow D(x_1)$$

$$B\,\boxed{\mathbf{P}} \rightarrow \varepsilon$$
$$B(\mathbf{P}) \rightarrow \varepsilon$$

$$D\,\boxed{\mathbf{M}} \rightarrow \varepsilon$$
$$D(\mathbf{M}) \rightarrow \varepsilon$$

(LCFRS,sDCP)-hybrid grammar:

$$S\,\boxed{z_1} \to \boxed{z_2}\,A\,\boxed{z_1}\;\;C\,\boxed{z_2}$$
$$S(x_1\;x_3\;x_2\;x_4) \to A(x_1, x_2)\;C(x_3, x_4)$$

$$\boxed{z_1}\,A\,\boxed{\mathbf{h}(z_2\;z_1)} \to B\,\boxed{z_2}$$
$$A(x_1, \mathbf{h}) \to B(x_1)$$

$$C\,\boxed{\boldsymbol{\ell}(z_1)} \to D\,\boxed{z_1}$$
$$C(x_1, \boldsymbol{\ell}) \to D(x_1)$$

$$B\,\boxed{\mathbf{P}} \to \varepsilon$$
$$B(\mathbf{P}) \to \varepsilon$$

$$D\,\boxed{\mathbf{M}} \to \varepsilon$$
$$D(\mathbf{M}) \to \varepsilon$$

(LCFRS,sDCP)-hybrid grammar:

$$S\ \boxed{z_1} \rightarrow \boxed{z_2}\ A\ \boxed{z_1}\quad C\ \boxed{z_2}$$
$$S\,(x_1\,x_3\,x_2\,x_4) \rightarrow A\,(x_1, x_2)\ C\,(x_3, x_4)$$

$$\boxed{z_1}\ A\ \boxed{\mathbf{h}(z_2\ z_1)} \rightarrow B\ \boxed{z_2}$$
$$A\,(x_1, \mathbf{h}) \rightarrow B\,(x_1)$$

$$C\ \boxed{\boldsymbol{\ell}(z_1)} \rightarrow D\ \boxed{z_1}$$
$$C\,(x_1, \boldsymbol{\ell}) \rightarrow D\,(x_1)$$

$$B\ \boxed{\mathbf{P}} \rightarrow \varepsilon$$
$$B\,(\mathbf{P}) \rightarrow \varepsilon$$

$$D\ \boxed{\mathbf{M}} \rightarrow \varepsilon$$
$$D\,(\mathbf{M}) \rightarrow \varepsilon$$

(LCFRS,sDCP)-hybrid grammar:

$$S\ \boxed{z_1} \rightarrow \boxed{z_2}\ A\ \boxed{z_1}\quad C\ \boxed{z_2}$$

$$S\,(x_1\ x_3\ x_2\ x_4) \rightarrow A\,(x_1, x_2)\ C\,(x_3, x_4)$$

$$\boxed{z_1}\ A\ \boxed{\mathbf{h}(z_2\ z_1)} \rightarrow B\ \boxed{z_2}$$

$$A\,(x_1, \mathbf{h}) \rightarrow B\,(x_1)$$

$$C\ \boxed{\ell(z_1)} \rightarrow D\ \boxed{z_1}$$

$$C\,(x_1, \ell) \rightarrow D\,(x_1)$$

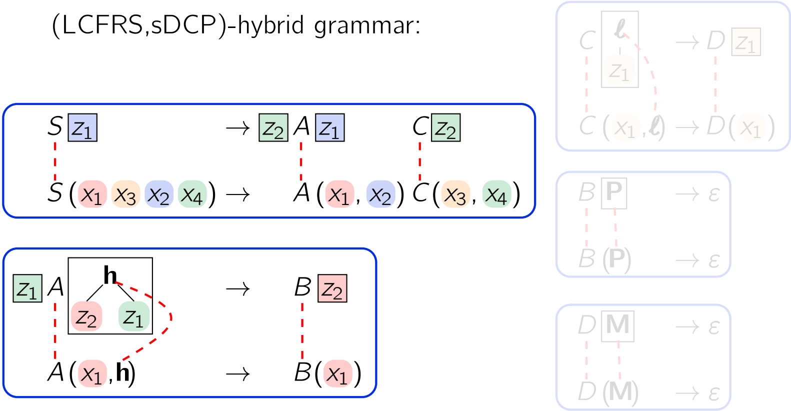
$$B\ \boxed{\mathbf{P}} \rightarrow \varepsilon$$

$$B\,(\mathbf{P}) \rightarrow \varepsilon$$

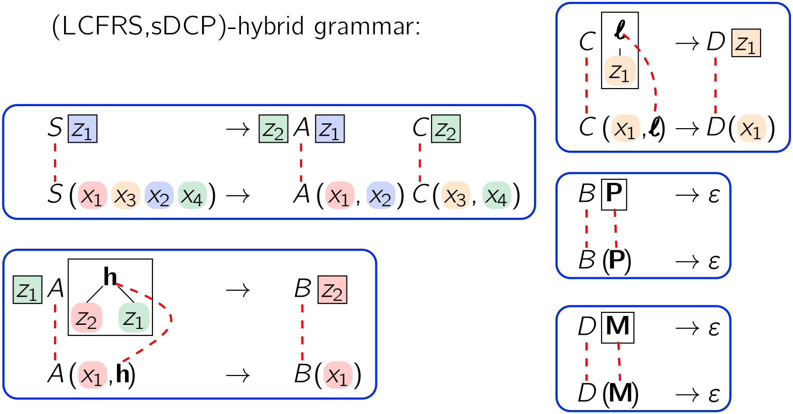
$$D\ \boxed{\mathbf{M}} \rightarrow \varepsilon$$

$$D\,(\mathbf{M}) \rightarrow \varepsilon$$

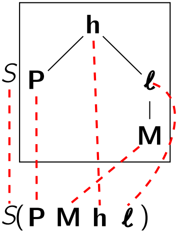
(LCFRS,sDCP)-hybrid grammar:



(LCFRS,sDCP)-hybrid grammar:



(LCFRS,sDCP)-hybrid grammar:



(LCFRS,sDCP)-hybrid grammar:

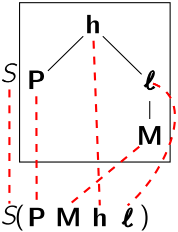
$$S\ \boxed{z_1} \rightarrow \boxed{z_2}\ A\ \boxed{z_1}\quad C\ \boxed{z_2}$$
$$S\,(x_1\ x_3\ x_2\ x_4) \rightarrow A\,(x_1, x_2)\ C\,(x_3, x_4)$$

$$\boxed{z_1}\ A\ \boxed{h(z_2\ z_1)} \rightarrow B\ \boxed{z_2}$$
$$A\,(x_1, \mathbf{h}) \rightarrow B\,(x_1)$$

$$C\ \boxed{\boldsymbol{\ell}(z_1)} \rightarrow D\ \boxed{z_1}$$
$$C\,(x_1, \boldsymbol{\ell}) \rightarrow D\,(x_1)$$

$$B\ \boxed{\mathbf{P}} \rightarrow \varepsilon$$
$$B\,(\mathbf{P}) \rightarrow \varepsilon$$

$$D\ \boxed{\mathbf{M}} \rightarrow \varepsilon$$
$$D\,(\mathbf{M}) \rightarrow \varepsilon$$



(LCFRS,sDCP)-hybrid grammar:

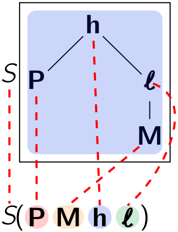
$$S\,\boxed{z_1} \rightarrow \boxed{z_2}\,A\,\boxed{z_1}\ \ C\,\boxed{z_2}$$
$$S(x_1\,x_3\,x_2\,x_4) \rightarrow A(x_1, x_2)\ C(x_3, x_4)$$

$$\boxed{z_1}\,A\,\boxed{\mathbf{h}(z_2\ z_1)} \rightarrow B\,\boxed{z_2}$$
$$A(x_1, \mathbf{h}) \rightarrow B(x_1)$$

$$C\,\boxed{\boldsymbol{\ell}(z_1)} \rightarrow D\,\boxed{z_1}$$
$$C(x_1, \boldsymbol{\ell}) \rightarrow D(x_1)$$

$$B\,\boxed{\mathbf{P}} \rightarrow \varepsilon$$
$$B(\mathbf{P}) \rightarrow \varepsilon$$

$$D\,\boxed{\mathbf{M}} \rightarrow \varepsilon$$
$$D(\mathbf{M}) \rightarrow \varepsilon$$



(LCFRS,sDCP)-hybrid grammar:

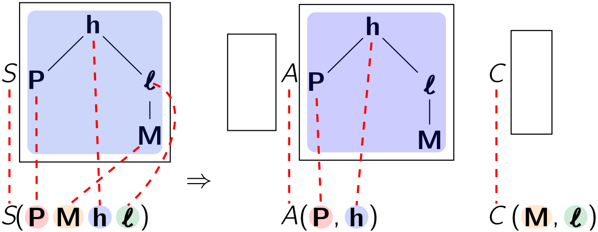
$$S\,\boxed{z_1} \rightarrow \boxed{z_2}\,A\,\boxed{z_1}\quad C\,\boxed{z_2}$$
$$S\,(x_1\, x_3\, x_2\, x_4) \rightarrow A\,(x_1, x_2)\; C\,(x_3, x_4)$$

$$\boxed{z_1}\,A\,\boxed{h(z_2, z_1)} \rightarrow B\,\boxed{z_2}$$
$$A\,(x_1, \mathbf{h}) \rightarrow B\,(x_1)$$

$$C\,\boxed{\ell(z_1)} \rightarrow D\,\boxed{z_1}$$
$$C\,(x_1, \ell) \rightarrow D\,(x_1)$$

$$B\,\boxed{\mathbf{P}} \rightarrow \varepsilon$$
$$B\,(\mathbf{P}) \rightarrow \varepsilon$$

$$D\,\boxed{\mathbf{M}} \rightarrow \varepsilon$$
$$D\,(\mathbf{M}) \rightarrow \varepsilon$$

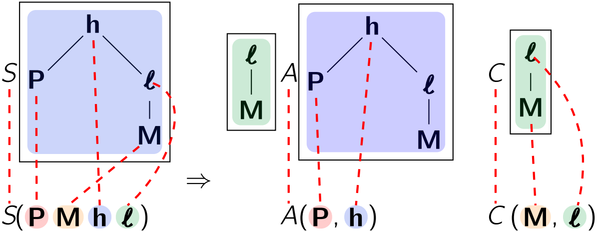


(LCFRS,sDCP)-hybrid grammar:

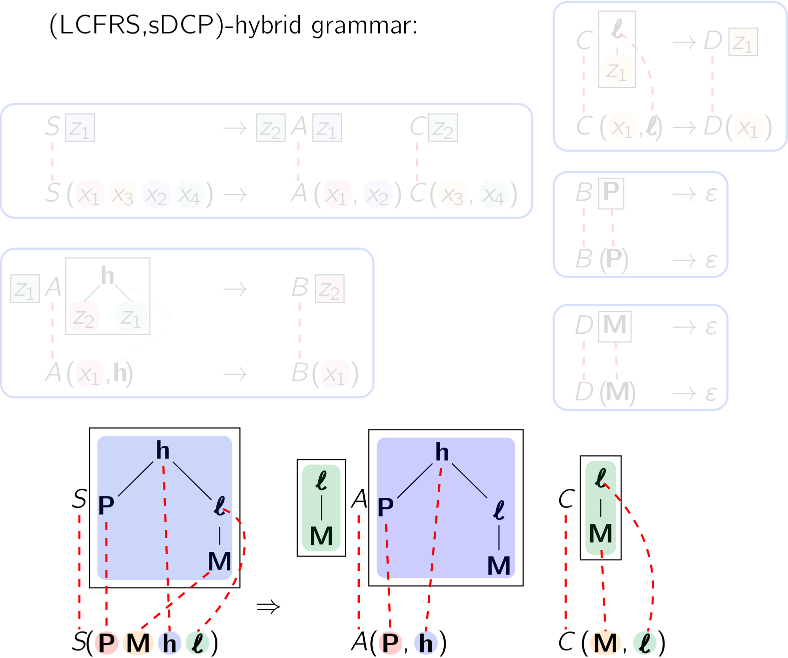
$$S\,\boxed{z_1} \to \boxed{z_2}\,A\,\boxed{z_1}\quad C\,\boxed{z_2}$$
$$S\,(x_1\,x_3\,x_2\,x_4) \to A\,(x_1, x_2)\ C\,(x_3, x_4)$$

$$B\,\boxed{\mathbf{P}} \to \varepsilon$$
$$B\,(\mathbf{P}) \to \varepsilon$$

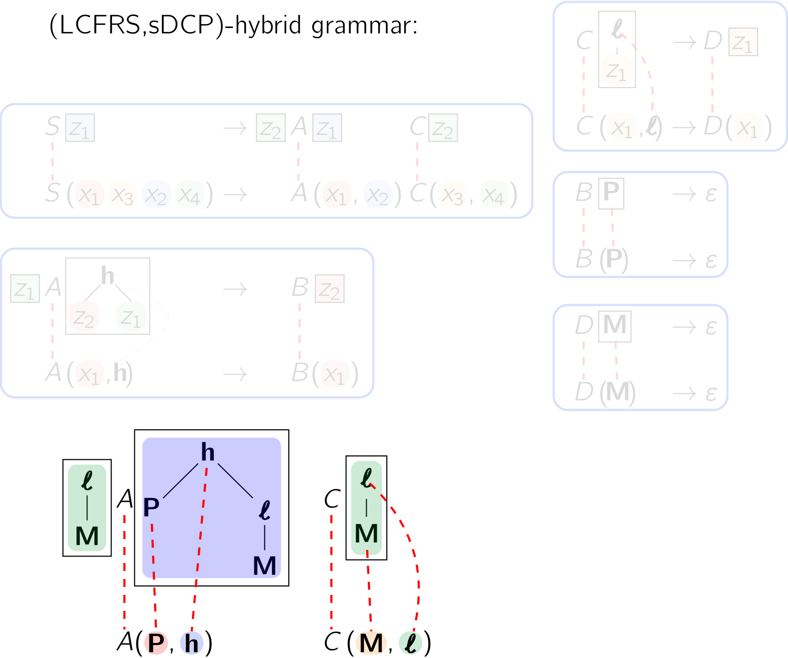
$$D\,\boxed{\mathbf{M}} \to \varepsilon$$
$$D\,(\mathbf{M}) \to \varepsilon$$



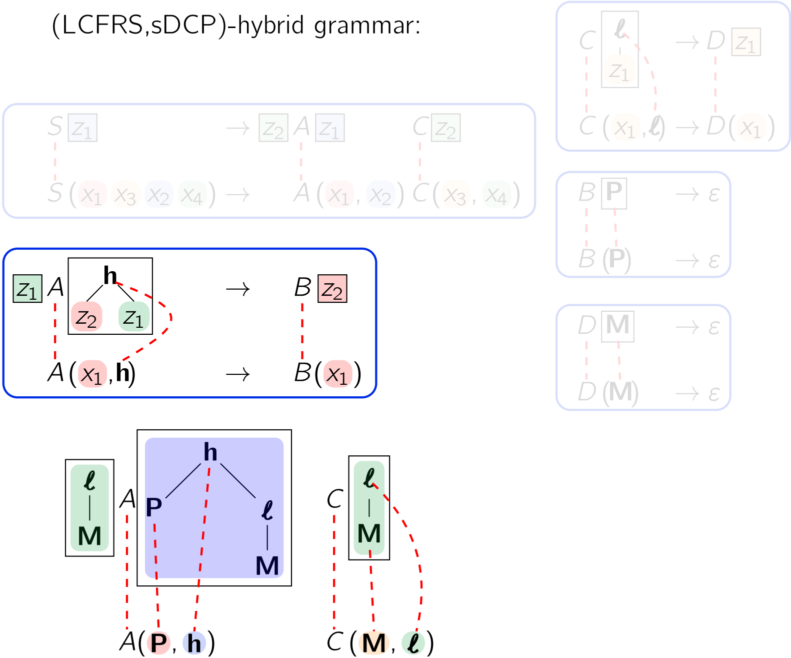
(LCFRS,sDCP)-hybrid grammar:



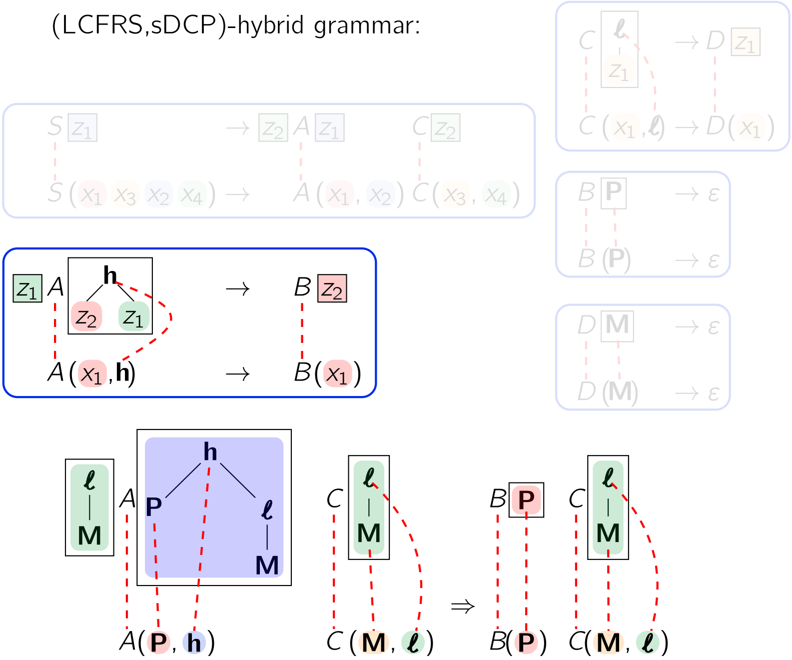
(LCFRS,sDCP)-hybrid grammar:



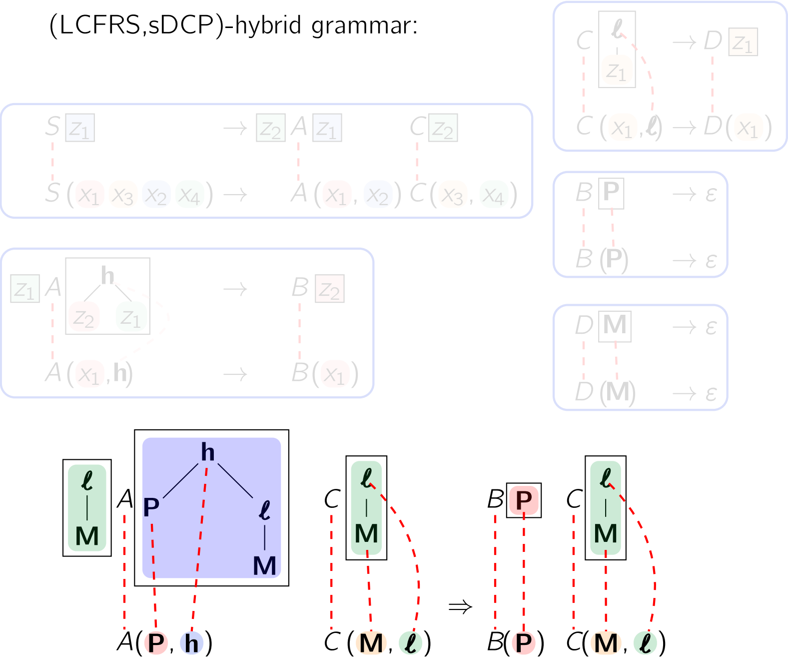
(LCFRS,sDCP)-hybrid grammar:



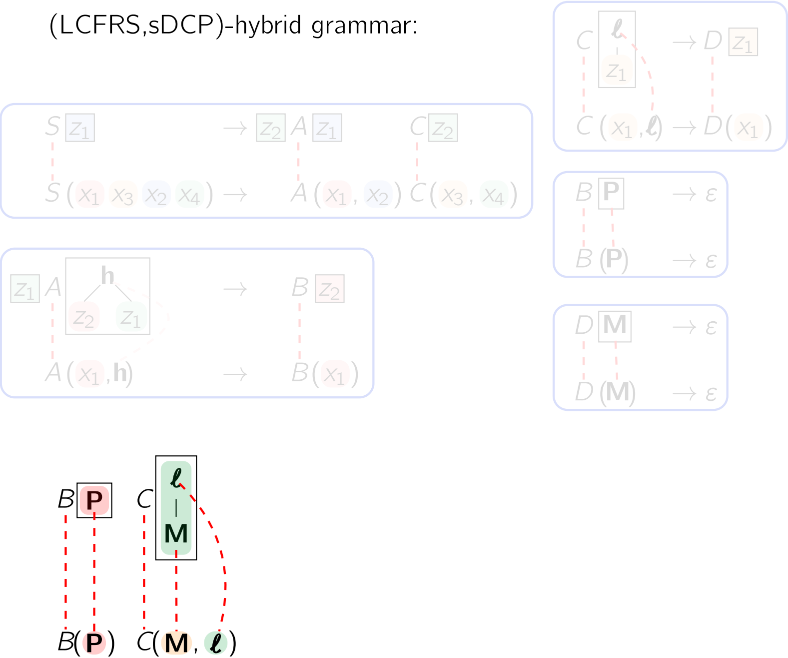
(LCFRS,sDCP)-hybrid grammar:



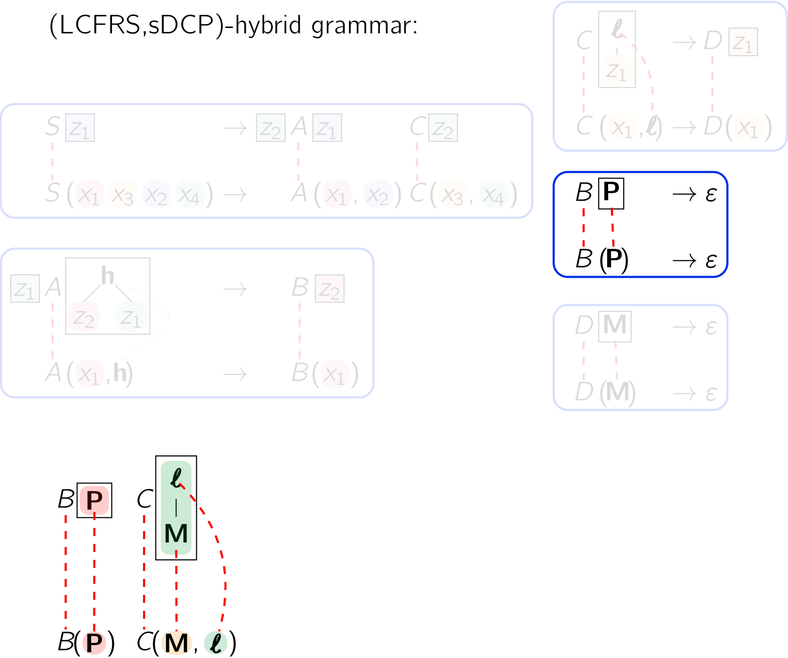
(LCFRS,sDCP)-hybrid grammar:



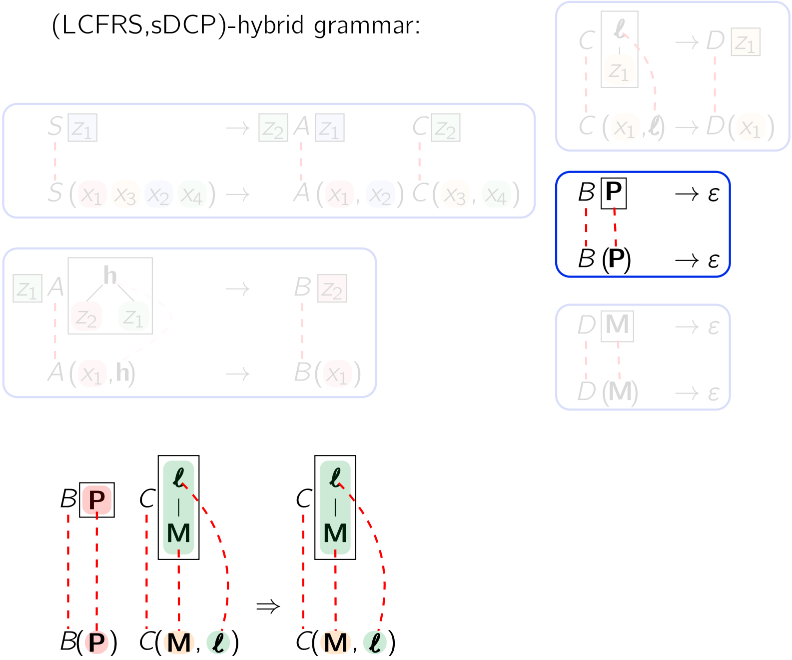
(LCFRS,sDCP)-hybrid grammar:



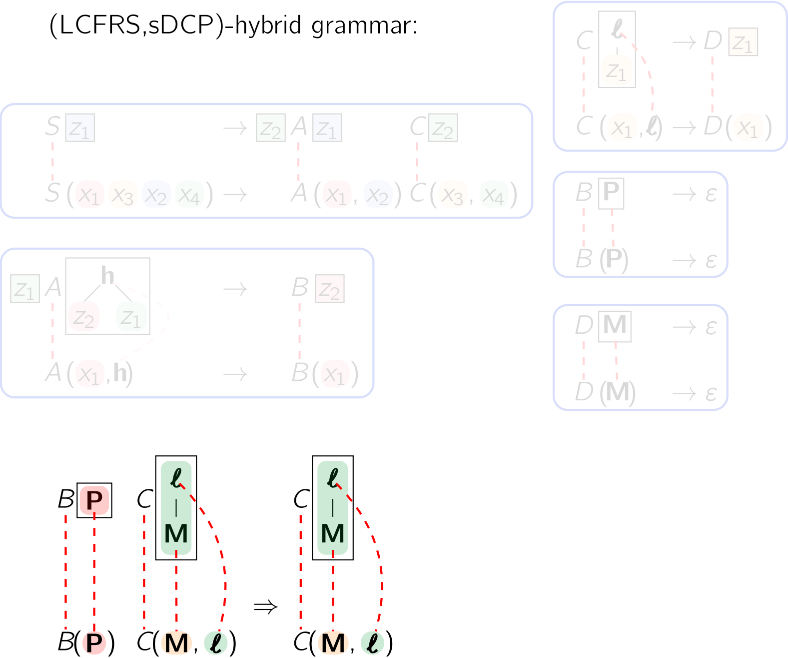
(LCFRS,sDCP)-hybrid grammar:



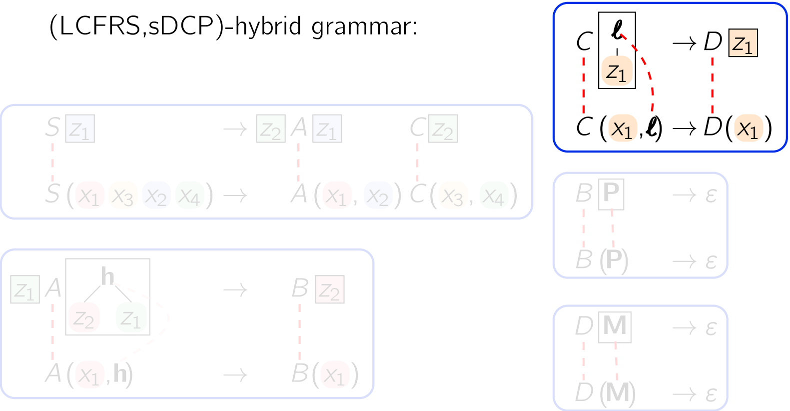
(LCFRS,sDCP)-hybrid grammar:

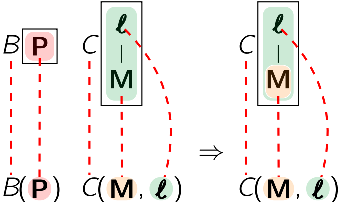


(LCFRS,sDCP)-hybrid grammar:

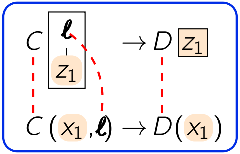


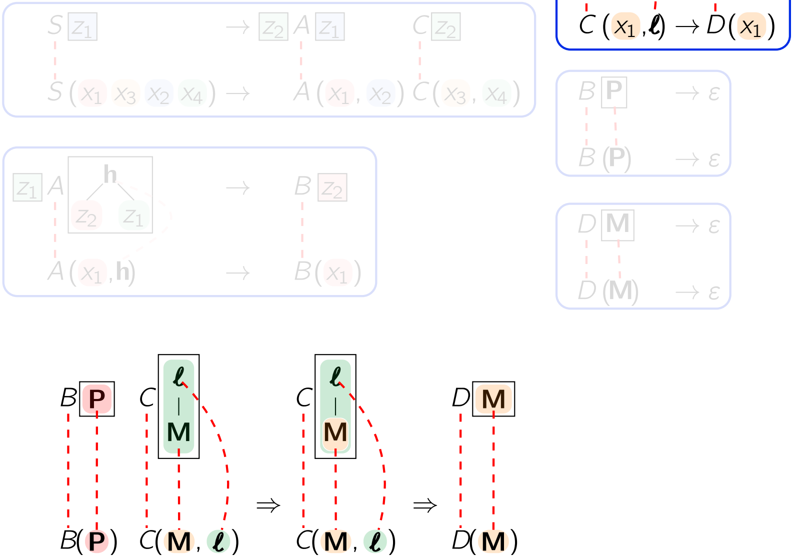
(LCFRS,sDCP)-hybrid grammar:



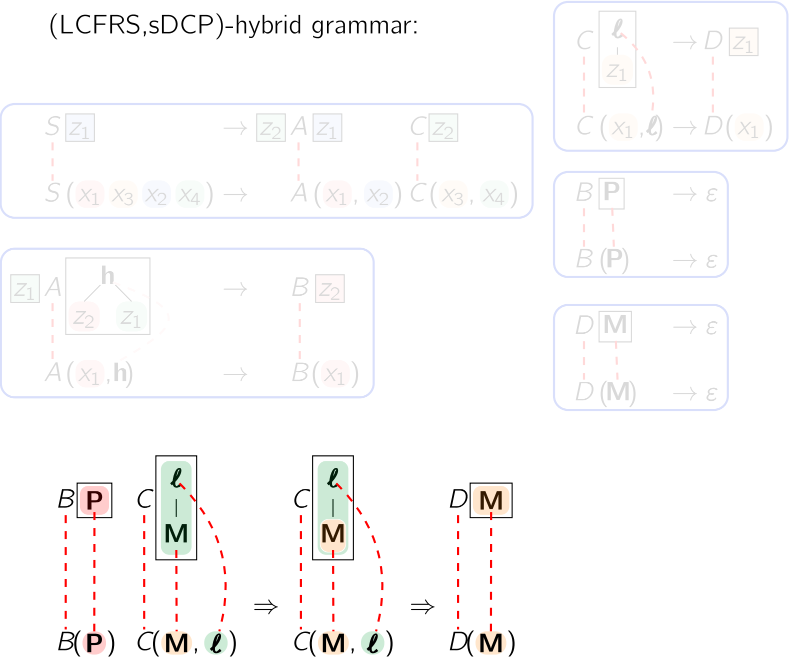


(LCFRS,sDCP)-hybrid grammar:

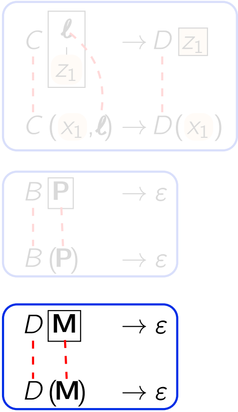


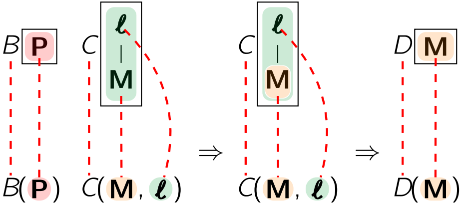


(LCFRS,sDCP)-hybrid grammar:

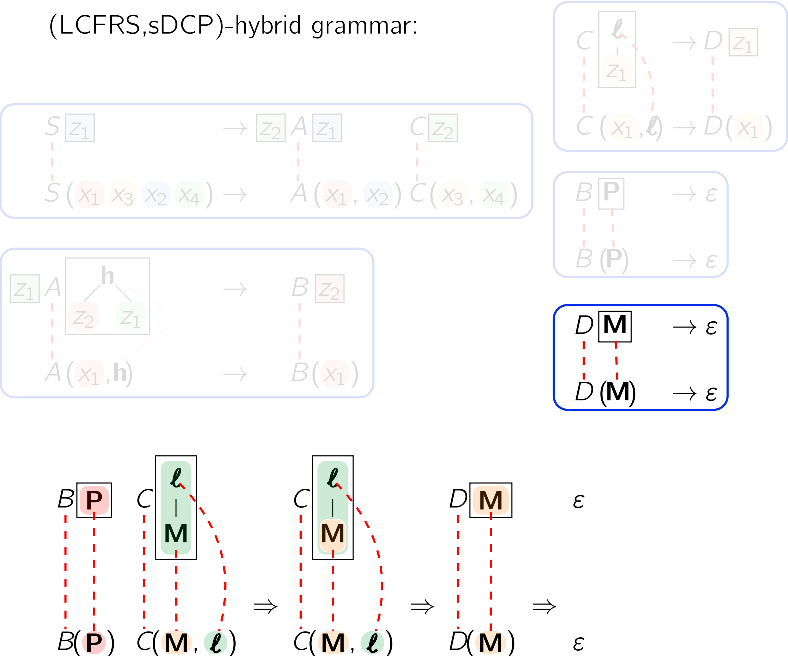


(LCFRS,sDCP)-hybrid grammar:

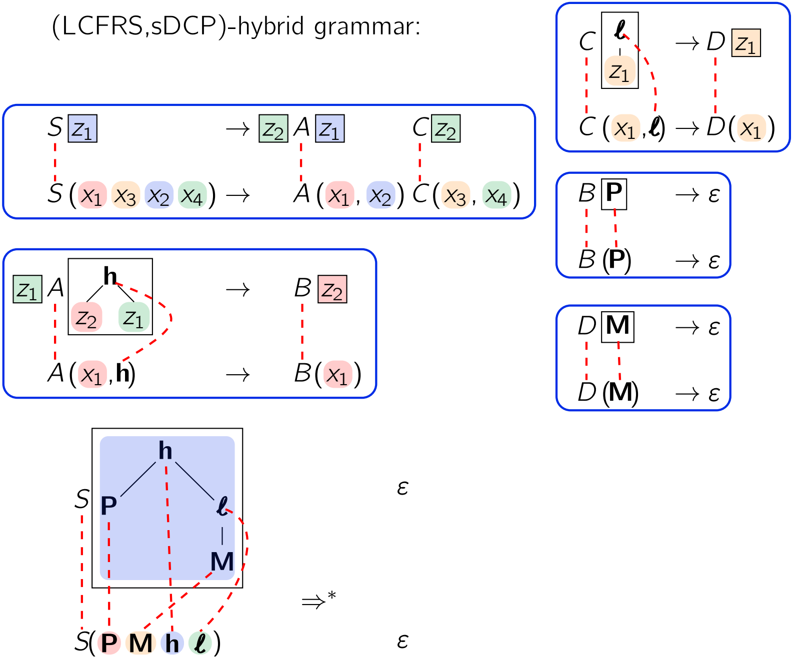




(LCFRS,sDCP)-hybrid grammar:



(LCFRS,sDCP)-hybrid grammar:



E: natural language $\quad\mathcal{H}$: set of hybrid trees

parsing : $E \to \mathcal{H}$

$$\text{parsing}(e) = \operatorname{proj}\Big(\operatorname{argmax}_{\substack{d \in D_G:\\ \operatorname{str}(\operatorname{proj}(d))=e}} P(d \mid (G,p))\Big)$$

where

- (G,p): probabilistic hybrid grammar
  (D_G: set of proof trees of G)
- $\operatorname{proj} : D_G \to \mathcal{H}$
- $P(r_1 \dots r_n \mid (G,p)) = p(r_1) \cdot \ldots \cdot p(r_n)$

parsing with hybrid grammar G:

sentence $w =$ **Jan**$_1$ **Piet**$_2$ **Marie**$_3$ **zag**$_4$ **helpen**$_5$ **lezen**$_6$

1. consider LCFRS $\mathrm{proj}_{\mathrm{LCFRS}}(G)$
2. parsing of w with $\mathrm{proj}_{\mathrm{LCFRS}}(G)$
   $\Rightarrow$ proof tree d for "$w \in L(\mathrm{proj}_{\mathrm{LCFRS}}(G))$"
3. enrich proof tree d by sDCP-part of used rules
   $\Rightarrow$ proof tree d' for "$h \in L(G)$ with $\mathrm{str}(h) = w$"
4. project proof tree d' to dependency tree

parsing with hybrid grammar G:

sentence $w =$ **Jan**$_1$ **Piet**$_2$ **Marie**$_3$ **zag**$_4$ **helpen**$_5$ **lezen**$_6$

1. consider LCFRS $\mathrm{proj}_{\mathrm{LCFRS}}(G)$
2. parsing of w with $\mathrm{proj}_{\mathrm{LCFRS}}(G)$
   $\Rightarrow$ proof tree d for "$w \in L(\mathrm{proj}_{\mathrm{LCFRS}}(G))$"
3. enrich proof tree d by sDCP-part of used rules
   $\Rightarrow$ proof tree d' for "$h \in L(G)$ with $\mathrm{str}(h) = w$"
4. project proof tree d' to dependency tree

$$\rho: \quad \begin{array}{l} \boxed{x_1} \; \{4\} \boxed{\begin{array}{c} \boldsymbol{\ell} \\ | \\ x_1 \end{array}} \to \; \varepsilon \\ \qquad \{4\}(\, \boldsymbol{\ell} \,) \; \to \; \varepsilon \end{array} \qquad \xrightarrow[\mathrm{proj}_{\mathrm{LCFRS}}]{} \qquad \{4\}(\, \boldsymbol{\ell} \,) \xrightarrow{\rho} \varepsilon$$

parsing with hybrid grammar G:

sentence $w =$ **Jan**$_1$ **Piet**$_2$ **Marie**$_3$ **zag**$_4$ **helpen**$_5$ **lezen**$_6$

1. consider LCFRS $\mathrm{proj}_{\mathrm{LCFRS}}(G)$
2. parsing of w with $\mathrm{proj}_{\mathrm{LCFRS}}(G)$
   $\Rightarrow$ proof tree d for "$w \in L(\mathrm{proj}_{\mathrm{LCFRS}}(G))$"
3. enrich proof tree d by sDCP-part of used rules
   $\Rightarrow$ proof tree d' for "$h \in L(G)$ with $\mathrm{str}(h) = w$"
4. project proof tree d' to dependency tree

$$\rho: \quad \begin{array}{l} \boxed{x_1} \ \{4\} \ \boxed{\begin{array}{c} \boldsymbol{\ell} \\ | \\ x_1 \end{array}} \to \ \varepsilon \\ \{4\}(\,\boldsymbol{\ell}\,) \ \to \ \varepsilon \end{array} \qquad \xrightarrow[\mathrm{proj}_{\mathrm{LCFRS}}]{} \qquad \{4\}(\,\boldsymbol{\ell}\,) \xrightarrow{\rho} \varepsilon$$

parsing with hybrid grammar G:

sentence $w = \textbf{Jan}_1\ \textbf{Piet}_2\ \textbf{Marie}_3\ \textbf{zag}_4\ \textbf{helpen}_5\ \textbf{lezen}_6$

1. consider LCFRS $\mathrm{proj}_{\mathrm{LCFRS}}(G)$
2. parsing of w with $\mathrm{proj}_{\mathrm{LCFRS}}(G)$
   $\Rightarrow$ proof tree d for "$w \in L(\mathrm{proj}_{\mathrm{LCFRS}}(G))$"
3. enrich proof tree d by sDCP-part of used rules
   $\Rightarrow$ proof tree d' for "$h \in L(G)$ with $\mathrm{str}(h) = w$"
4. project proof tree d' to dependency tree

$$\rho: \quad \boxed{x_1}\ \{4\}\ \boxed{\begin{array}{c}\boldsymbol{\ell}\\ |\\ x_1\end{array}} \to \varepsilon \qquad \xleftarrow{\text{enrich}}$$

$$\{4\}(\boldsymbol{\ell}) \to \varepsilon \qquad \xrightarrow[\mathrm{proj}_{\mathrm{LCFRS}}]{} \qquad \{4\}(\boldsymbol{\ell}) \xrightarrow{\rho} \varepsilon$$

parsing with hybrid grammar G:

sentence $w = \textbf{Jan}_1\ \textbf{Piet}_2\ \textbf{Marie}_3\ \textbf{zag}_4\ \textbf{helpen}_5\ \textbf{lezen}_6$

1. consider LCFRS $\mathrm{proj}_{\mathrm{LCFRS}}(G)$

2. parsing of w with $\mathrm{proj}_{\mathrm{LCFRS}}(G)$
   $\Rightarrow$ proof tree d for "$w \in L(\mathrm{proj}_{\mathrm{LCFRS}}(G))$"

3. enrich proof tree d by sDCP-part of used rules
   $\Rightarrow$ proof tree d' for "$h \in L(G)$ with $\mathrm{str}(h) = w$"

4. project proof tree d' to dependency tree

$$\rho:\quad \begin{array}{l} \boxed{x_1}\ \{4\}\ \boxed{\begin{array}{c}\boldsymbol{\ell}\\ |\\ x_1\end{array}} \to\ \varepsilon \\ \{4\}(\,\boldsymbol{\ell}\,)\ \to\ \varepsilon \end{array} \qquad \xleftarrow{\text{enrich}} \quad \xrightarrow[\mathrm{proj}_{\mathrm{LCFRS}}]{} \qquad \{4\}(\,\boldsymbol{\ell}\,) \xrightarrow{\rho} \varepsilon$$

How to obtain a hybrid grammar for E?

- grammar induction from corpus of hybrid trees
  [Nederhof, HV 14], [Gebhardt, Nederhof, HV 17]

- training of probabilities
  [Drewes, Gebhardt, HV 16]
  EM-training for weighted aligned hypergraph bimorphisms.
  ACL Workshop StatFSM 2016

How to obtain a hybrid grammar for E?

- grammar induction from corpus of hybrid trees
  [Nederhof, HV 14], [Gebhardt, Nederhof, HV 17]

- training of probabilities
  [Drewes, Gebhardt, HV 16]
  EM-training for weighted aligned hypergraph bimorphisms.
  ACL Workshop StatFSM 2016

How to obtain a hybrid grammar for E?

- grammar induction from corpus of hybrid trees
  [Nederhof, HV 14], [Gebhardt, Nederhof, HV 17]

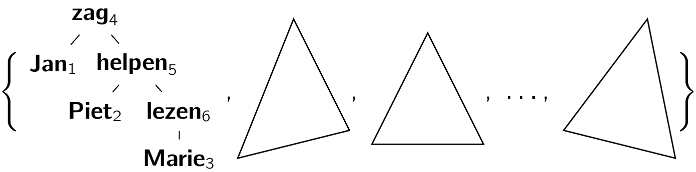
- training of probabilities
  [Drewes, Gebhardt, HV 16]
  EM-training for weighted aligned hypergraph bimorphisms.
  ACL Workshop StatFSM 2016

How to obtain a hybrid grammar for E?

- grammar induction from corpus of hybrid trees
  [Nederhof, HV 14], [Gebhardt, Nederhof, HV 17]

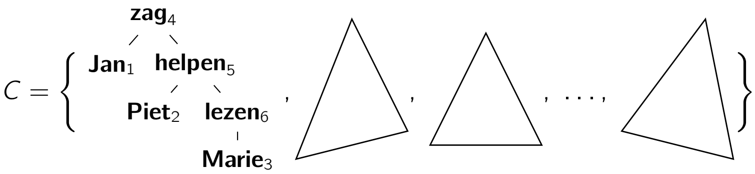
- training of probabilities
  [Drewes, Gebhardt, HV 16]
  EM-training for weighted aligned hypergraph bimorphisms.
  ACL Workshop StatFSM 2016

corpus of dependency trees for language E:

$C =$ 

(corpus = large, finite set)

corpus of dependency trees for language E:

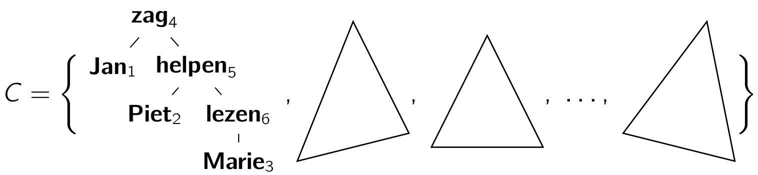


(corpus = large, finite set)

corpora for dependency trees

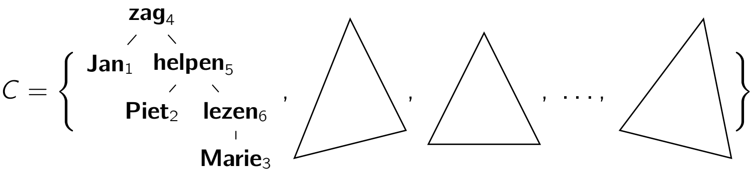
- Prague Dependency Treebank (Czech, 26k sentences)
- Slovene Dependency Treebank (Slovene, 30k words)
- Danish Dependency Treebank (Danish, 5k sentence)
- Talbanken05 (Swedish, 30k words)
- Metu-Sabancı treebank (Turkish, 7k sentences)
- ...

corpus of dependency trees for language E:



(corpus = large, finite set)

corpus of dependency trees for language E:



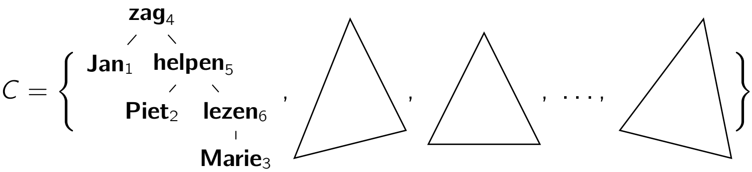
(corpus = large, finite set)

goal:

construct hybrid grammar G such that $C \subsetneq L(G) \subsetneq \mathcal{H}$

"G generalizes C"

corpus of dependency trees for language E:



(corpus = large, finite set)

goal:

construct hybrid grammar G such that $C \subsetneq L(G) \subsetneq \mathcal{H}$

"G generalizes C"

transformation $C \rightsquigarrow G$ is called grammar induction

grammar induction from <u>one</u> hybrid tree:

hybrid tree
$h = (\xi, U, \preceq)$

grammar induction from one hybrid tree:

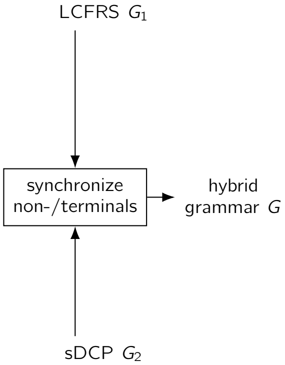
hybrid tree
$h = (\xi, U, \preceq)$

hybrid
grammar G

$L(G) = \{h\}$

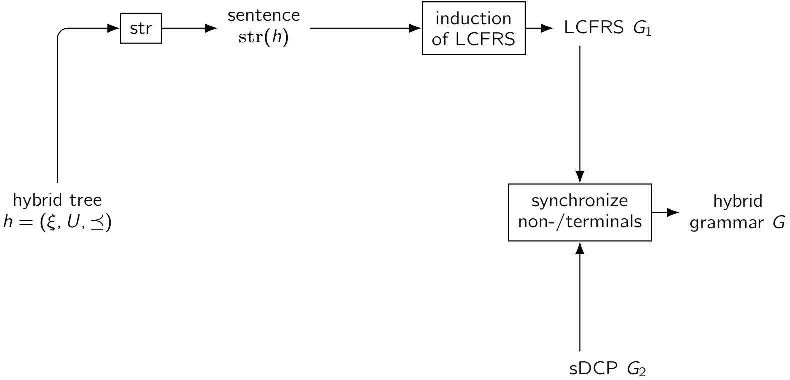
grammar induction from <u>one</u> hybrid tree:

hybrid tree
$h = (\xi, U, \preceq)$



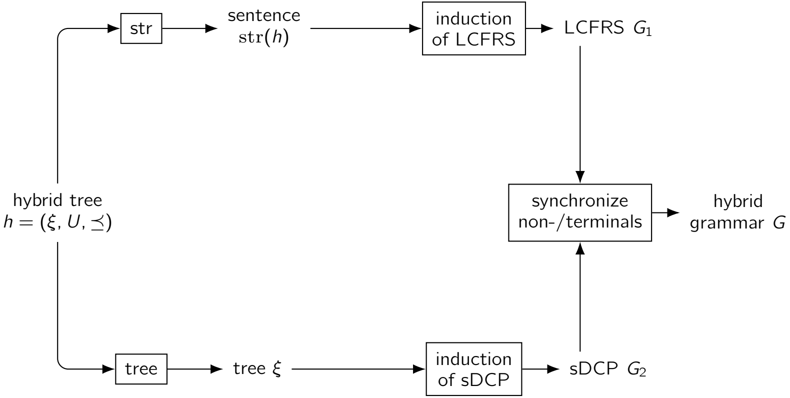
$L(G) = \{h\}$

grammar induction from one hybrid tree:



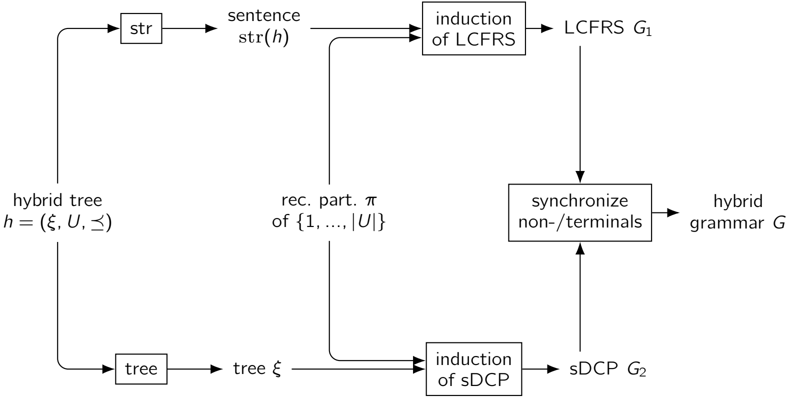
$$L(G) = \{h\}$$

grammar induction from <u>one</u> hybrid tree:



$L(G) = \{h\}$

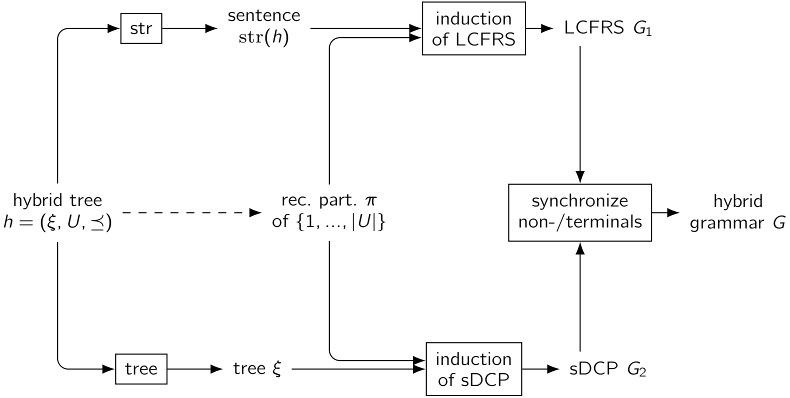
grammar induction from one hybrid tree:



$L(G) = \{h\}$

parsing of $\mathrm{str}(h)$ according to π

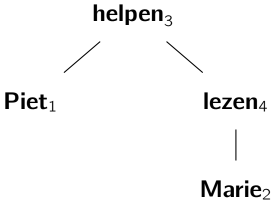
grammar induction from one hybrid tree:



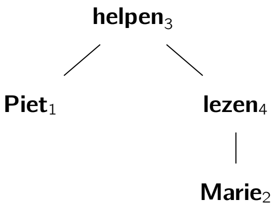
$L(G) = \{h\}$

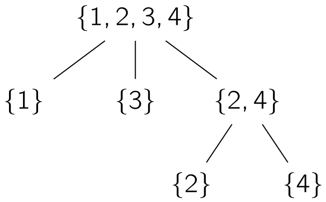
parsing of $\mathrm{str}(h)$ according to π

hybrid tree $h = (\xi, U, \preceq)$



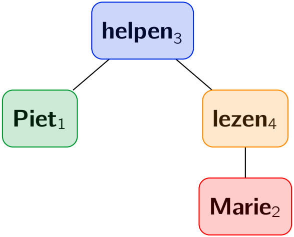
hybrid tree $h = (\xi, U, \preceq)$ recursive partitioning π of $\{1, \ldots, |U|\}$



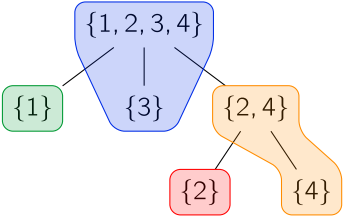


$$\mathrm{fanout}(\pi) = \max_{p \in \mathrm{pos}(\pi)} \Big(\text{number of intervals in label of } \pi \text{ at } p\Big)$$

hybrid tree $h = (\xi, U, \preceq)$



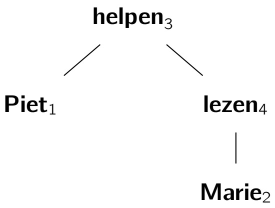
recursive partitioning π of $\{1, \ldots, |U|\}$



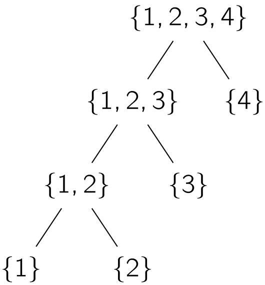
- extracted from h

$$\text{fanout}(\pi) = \max_{p \in \mathrm{pos}(\pi)} \Big(\text{number of intervals in label of } \pi \text{ at } p\Big)$$

hybrid tree $h = (\xi, U, \preceq)$



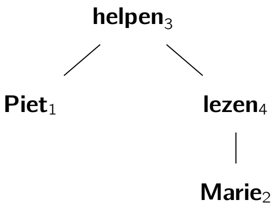
recursive partitioning π of $\{1, \ldots, |U|\}$



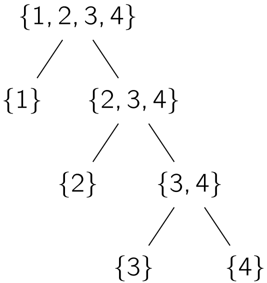
- extracted from h
- left-/right-branching

$$\text{fanout}(\pi) = \max_{p \in \text{pos}(\pi)} \Big(\text{number of intervals in label of } \pi \text{ at } p\Big)$$

hybrid tree $h = (\xi, U, \preceq)$



recursive partitioning π of $\{1, \ldots, |U|\}$

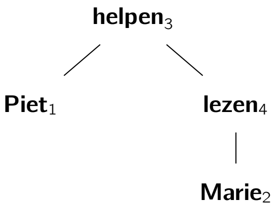


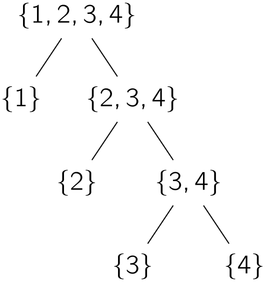
- extracted from h
- left-/right-branching

$$\text{fanout}(\pi) = \max_{p \in \mathrm{pos}(\pi)} \Big(\text{number of intervals in label of } \pi \text{ at } p\Big)$$

hybrid tree $h = (\xi, U, \preceq)$

recursive partitioning π of $\{1, \ldots, |U|\}$

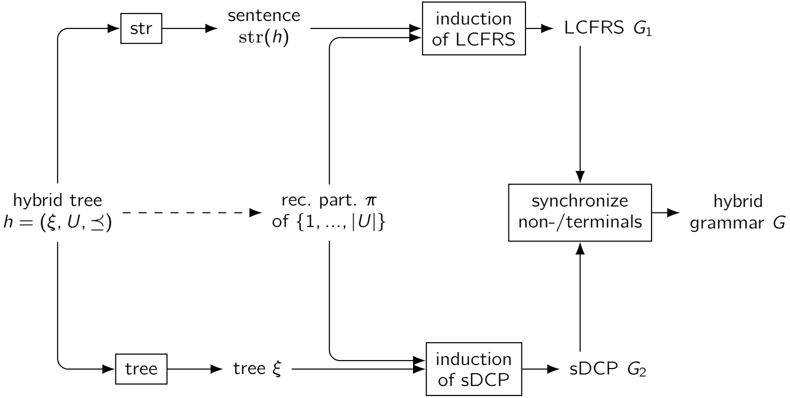




- extracted from h
- left-/right-branching
- ...

$$\text{fanout}(\pi) = \max_{p \in \text{pos}(\pi)} \Big(\text{number of intervals in label of } \pi \text{ at } p\Big)$$

grammar induction from <u>one</u> hybrid tree:

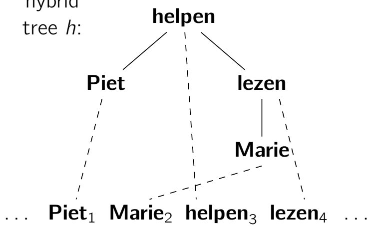


$L(G) = \{h\}$

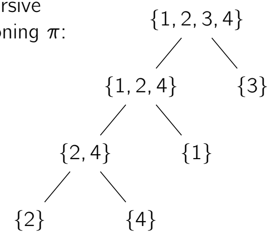
parsing of $\mathrm{str}(h)$ according to π

induction of LCFRS G_1:

hybrid tree h:

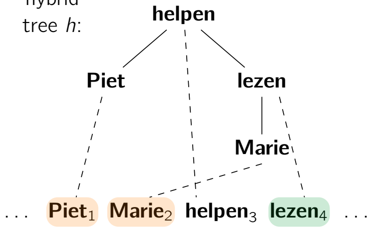


recursive partitioning π:

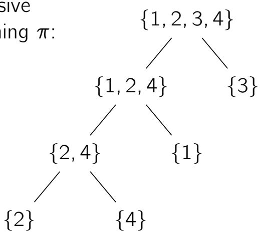


induction of LCFRS G_1:

hybrid tree h:

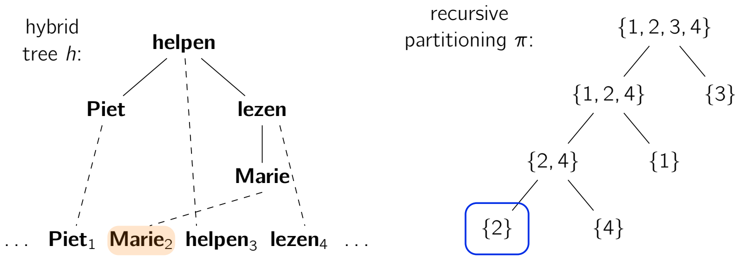


recursive partitioning π:

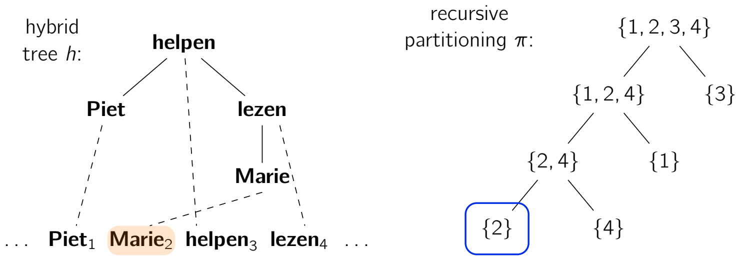


$\{1, 2, 4\}$(**Piet Marie**, **lezen**) $\Rightarrow^*$ ε

induction of LCFRS G_1:



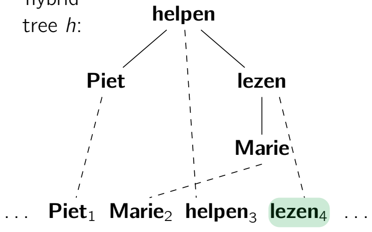
induction of LCFRS G_1:



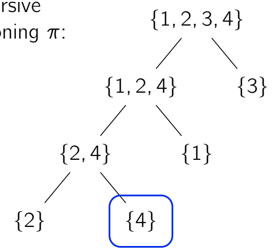
$\{2\}\,(\textbf{Marie}) \rightarrow \varepsilon$

induction of LCFRS G_1:

hybrid tree h:

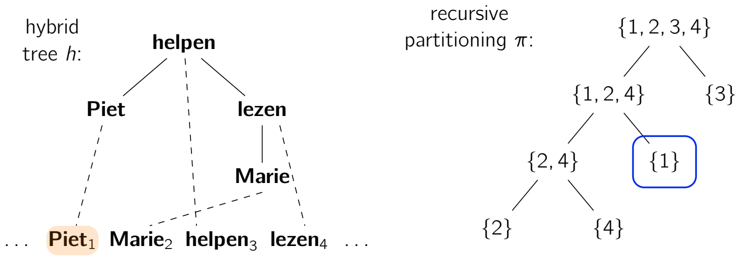


recursive partitioning π:



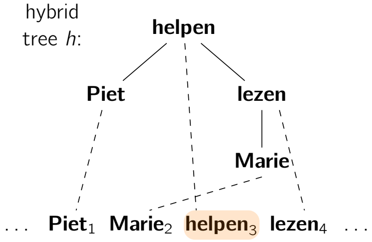
$$\{2\}\,(\textbf{Marie}) \rightarrow \varepsilon \quad \{4\}\,(\textbf{lezen}) \rightarrow \varepsilon$$

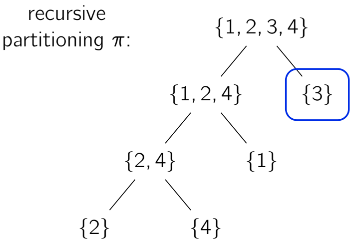
induction of LCFRS G_1:



$\{2\}\,(\textbf{Marie}) \to \varepsilon \quad \{4\}\,(\textbf{lezen}) \to \varepsilon \quad \{1\}\,(\textbf{Piet}) \to \varepsilon$

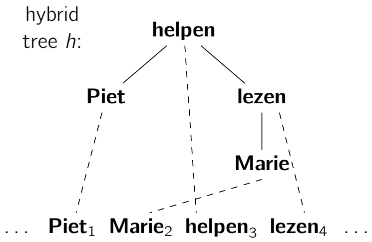
induction of LCFRS G_1:

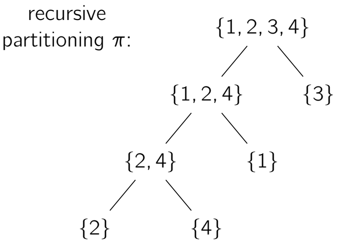




$\{2\}\,(\textbf{Marie}) \to \varepsilon \quad \{4\}\,(\textbf{lezen}) \to \varepsilon \quad \{1\}\,(\textbf{Piet}) \to \varepsilon \quad \{3\}\,(\textbf{helpen}) \to \varepsilon$

induction of LCFRS G_1:

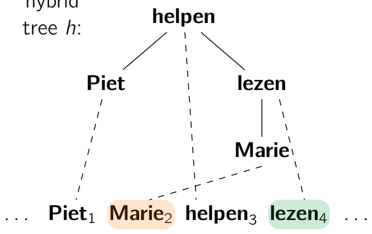




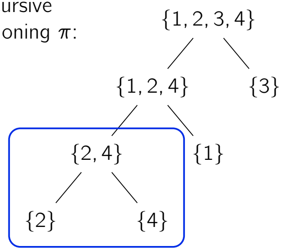
$\{\mathit{2}\}\,(\textbf{Marie}) \to \varepsilon \quad \{\mathit{4}\}\,(\textbf{lezen}) \to \varepsilon \quad \{\mathit{1}\}\,(\textbf{Piet}) \to \varepsilon \quad \{\mathit{3}\}\,(\textbf{helpen}) \to \varepsilon$

induction of LCFRS G_1:

hybrid tree h:



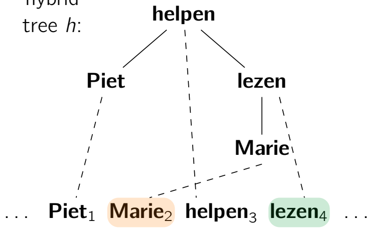
recursive partitioning π:



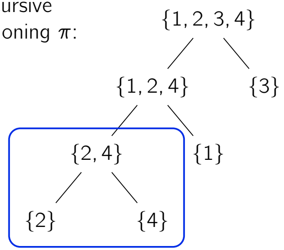
$\{2\}\,(\textbf{Marie}) \to \varepsilon \quad \{4\}\,(\textbf{lezen}) \to \varepsilon \quad \{1\}\,(\textbf{Piet}) \to \varepsilon \quad \{3\}\,(\textbf{helpen}) \to \varepsilon$

induction of LCFRS G_1:

hybrid tree h:



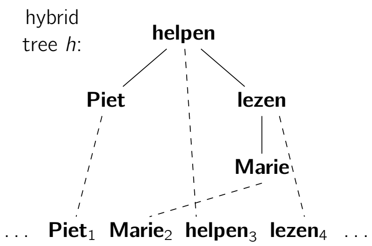
recursive partitioning π:

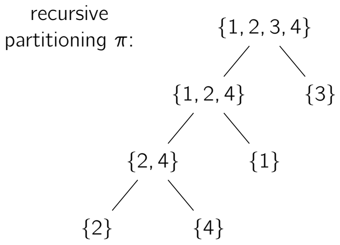


$\{2\}\,(\textbf{Marie}) \to \varepsilon \quad \{4\}\,(\textbf{lezen}) \to \varepsilon \quad \{1\}\,(\textbf{Piet}) \to \varepsilon \quad \{3\}\,(\textbf{helpen}) \to \varepsilon$

$\{2,4\}(x_1, x_2) \to \{2\}(x_1) \;\; \{4\}(x_2)$

induction of LCFRS G_1:



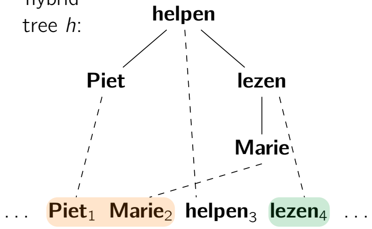


$\{2\}\,(\textbf{Marie}) \to \varepsilon \quad \{4\}\,(\textbf{lezen}) \to \varepsilon \quad \{1\}\,(\textbf{Piet}) \to \varepsilon \quad \{3\}\,(\textbf{helpen}) \to \varepsilon$

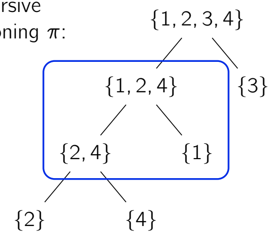
$\{2,4\}(x_1, x_2) \to \{2\}(x_1) \;\; \{4\}(x_2)$

induction of LCFRS G_1:

hybrid tree h:



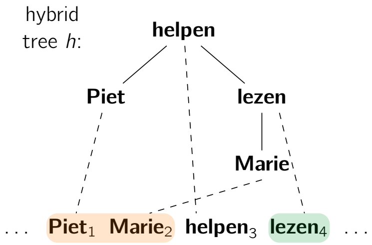
recursive partitioning π:

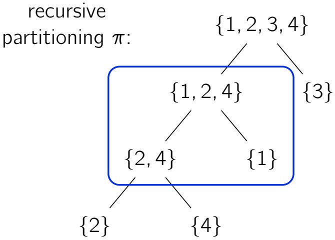


$$\{2\}\,(\textbf{Marie}) \to \varepsilon \quad \{4\}\,(\textbf{lezen}) \to \varepsilon \quad \{1\}\,(\textbf{Piet}) \to \varepsilon \quad \{3\}\,(\textbf{helpen}) \to \varepsilon$$

$$\{2,4\}(x_1, x_2) \to \{2\}(x_1) \;\; \{4\}(x_2)$$

induction of LCFRS G_1:





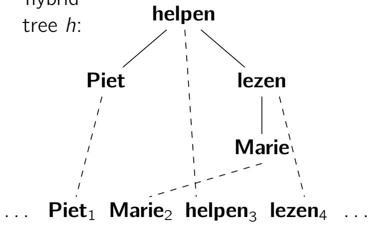
$\{2\}\,(\textbf{Marie}) \to \varepsilon \quad \{4\}\,(\textbf{lezen}) \to \varepsilon \quad \{1\}\,(\textbf{Piet}) \to \varepsilon \quad \{3\}\,(\textbf{helpen}) \to \varepsilon$

$\{2,4\}(x_1, x_2) \to \{2\}(x_1) \;\; \{4\}(x_2)$

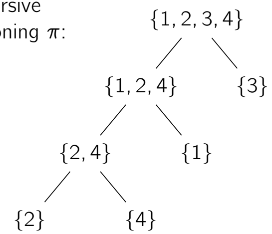
$\{1,2,4\}(x_3\, x_1, x_2) \to \{2,4\}(x_1, x_2) \;\; \{1\}(x_3)$

induction of LCFRS G_1:

hybrid tree h:



recursive partitioning π:



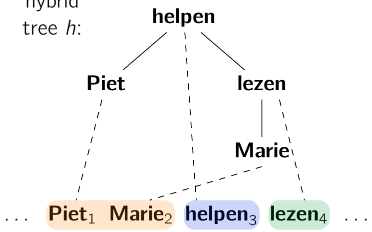
$$\{2\}\,(\textbf{Marie}) \to \varepsilon \quad \{4\}\,(\textbf{lezen}) \to \varepsilon \quad \{1\}\,(\textbf{Piet}) \to \varepsilon \quad \{3\}\,(\textbf{helpen}) \to \varepsilon$$

$$\{2,4\}(x_1, x_2) \to \{2\}(x_1) \;\; \{4\}(x_2)$$

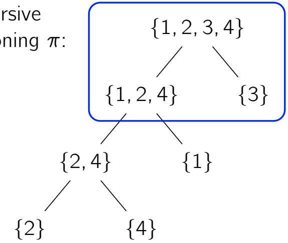
$$\{1,2,4\}(x_3\, x_1, x_2) \to \{2,4\}(x_1, x_2) \;\; \{1\}(x_3)$$

induction of LCFRS G_1:

hybrid tree h:



recursive partitioning π:



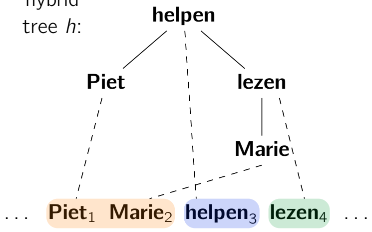
$\{2\}(\textbf{Marie}) \to \varepsilon \quad \{4\}(\textbf{lezen}) \to \varepsilon \quad \{1\}(\textbf{Piet}) \to \varepsilon \quad \{3\}(\textbf{helpen}) \to \varepsilon$

$\{2,4\}(x_1, x_2) \to \{2\}(x_1) \;\; \{4\}(x_2)$

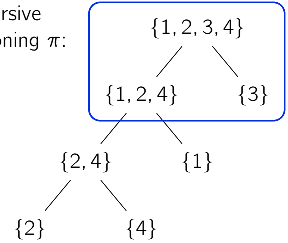
$\{1,2,4\}(x_3\, x_1, x_2) \to \{2,4\}(x_1, x_2) \;\; \{1\}(x_3)$

induction of LCFRS G_1:

hybrid tree h:



recursive partitioning π:



$$\{2\}\,(\textbf{Marie}) \to \varepsilon \quad \{4\}\,(\textbf{lezen}) \to \varepsilon \quad \{1\}\,(\textbf{Piet}) \to \varepsilon \quad \{3\}\,(\textbf{helpen}) \to \varepsilon$$

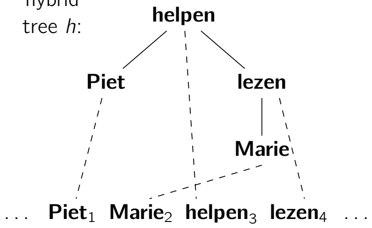
$$\{2,4\}(x_1, x_2) \to \{2\}(x_1) \;\; \{4\}(x_2)$$

$$\{1,2,4\}(x_3\, x_1, x_2) \to \{2,4\}(x_1, x_2) \;\; \{1\}(x_3)$$

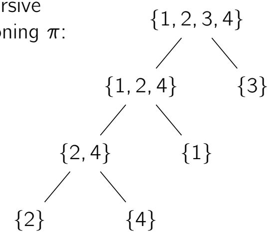
$$\{1,2,3,4\}\,(x_1\, x_3\, x_2) \to \{1,2,4\}\,(x_1, x_2) \;\; \{3\}\,(x_3)$$

induction of LCFRS G_1:

hybrid tree h:



recursive partitioning π:



$$\{2\}\,(\textbf{Marie}) \to \varepsilon \quad \{4\}\,(\textbf{lezen}) \to \varepsilon \quad \{1\}\,(\textbf{Piet}) \to \varepsilon \quad \{3\}\,(\textbf{helpen}) \to \varepsilon$$

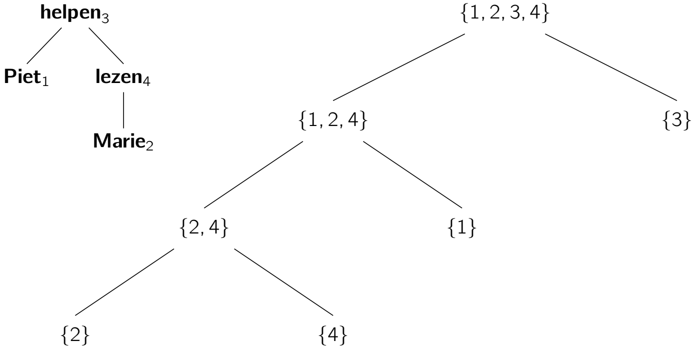
$$\{2,4\}(x_1, x_2) \to \{2\}(x_1) \;\; \{4\}(x_2)$$

$$\{1,2,4\}(x_3\, x_1, x_2) \to \{2,4\}(x_1, x_2) \;\; \{1\}(x_3)$$

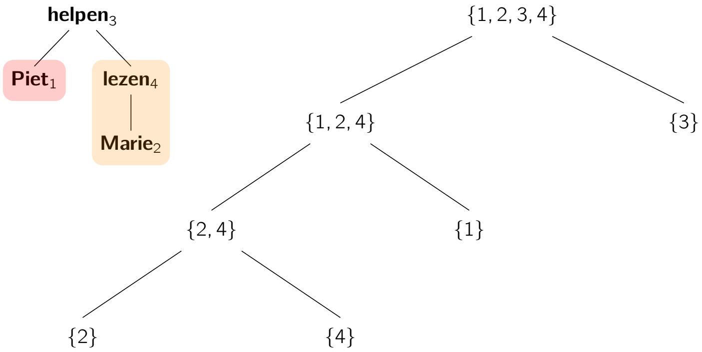
$$\{1,2,3,4\}\,(x_1\, x_3\, x_2) \to \{1,2,4\}\,(x_1, x_2) \;\; \{3\}\,(x_3)$$

$$\text{fanout}(G_1) = \text{fanout}(\pi)$$

induction of sDCP G_2:

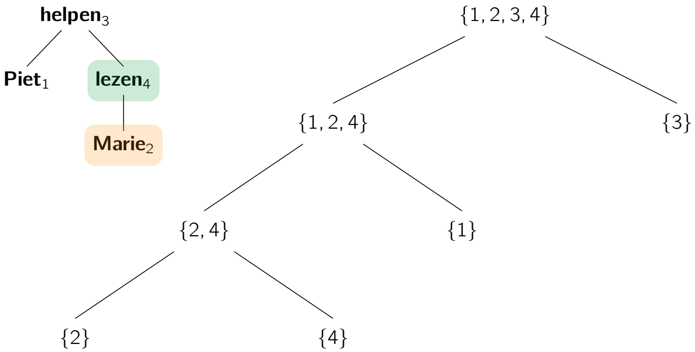


induction of sDCP G_2:



$$\{1,2,4\}\ \boxed{\mathbf{P}}\ \boxed{\boldsymbol{\ell}(\mathbf{M})} \quad \Rightarrow^* \quad \varepsilon$$

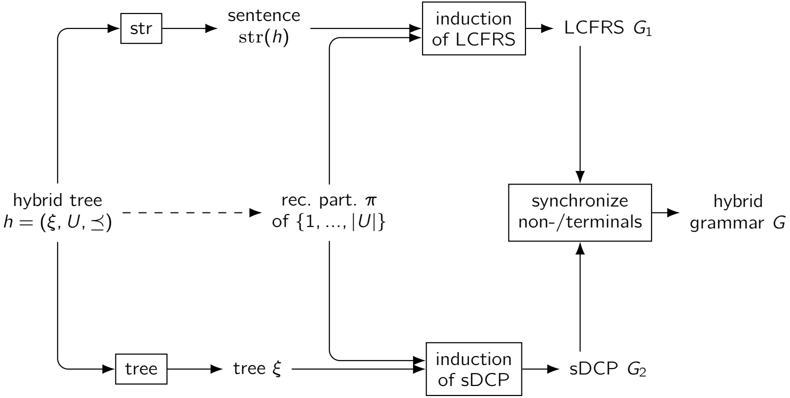
induction of sDCP G_2:



$$\{1,2,4\}\ \boxed{\mathbf{P}}\ \boxed{\boldsymbol{\ell}(\mathbf{M})} \quad \Rightarrow^* \quad \varepsilon$$

$$\boxed{\mathbf{M}}\ \{4\}\ \boxed{\boldsymbol{\ell}(\,\mathbf{M}\,)} \quad \Rightarrow^* \quad \varepsilon$$

grammar induction from one hybrid tree:



$L(G) = \{h\}$
parsing of $\mathrm{str}(h)$ according to π

synchronization of nonterminals and terminals:

sDCP-rule: $\{2\}\ \boxed{\mathbf{M}} \rightarrow \varepsilon$

LCFRS-rule: $\{2\}\ (\mathbf{M}) \rightarrow \varepsilon$

synchronization of nonterminals and terminals:

sDCP-rule: $\{2\}\ \boxed{\mathbf{M}} \to \varepsilon$

LCFRS-rule: $\{2\}\ (\mathbf{M}) \to \varepsilon$

synchronization of nonterminals and terminals:

sDCP-rule: $\{2\}\ \boxed{\mathbf{M}} \rightarrow \varepsilon$

LCFRS-rule: $\{2\}\ (\mathbf{M}) \rightarrow \varepsilon$

synchronization of nonterminals and terminals:

sDCP-rule: $\{2\}\ \boxed{\mathbf{M}} \rightarrow \varepsilon$ $\qquad$ $\boxed{x_1}\ \{4\}\ \boxed{\boldsymbol{\ell}(x_1)} \rightarrow \varepsilon$

LCFRS-rule: $\{2\}\ (\mathbf{M}) \rightarrow \varepsilon$ $\qquad$ $\{4\}(\boldsymbol{\ell}) \rightarrow \varepsilon$

synchronization of nonterminals and terminals:

sDCP-rule: $\{2\}\ \boxed{\mathbf{M}} \rightarrow \varepsilon$ $\quad$ $\boxed{x_1}\ \{4\}\ \boxed{\begin{array}{c}\boldsymbol{\ell}\\ | \\ x_1\end{array}} \rightarrow \varepsilon$

LCFRS-rule: $\{2\}\ (\mathbf{M}) \rightarrow \varepsilon$ $\quad$ $\{4\}(\boldsymbol{\ell}) \rightarrow \varepsilon$

synchronization of nonterminals and terminals:

sDCP-rule: $\{2\}\ \boxed{\mathbf{M}} \rightarrow \varepsilon$ $\qquad$ $\boxed{x_1}\ \{4\}\ \boxed{\begin{array}{c}\boldsymbol{\ell} \\ | \\ x_1\end{array}} \rightarrow \varepsilon$

LCFRS-rule: $\{2\}\ (\mathbf{M}) \rightarrow \varepsilon$ $\qquad$ $\{4\}(\boldsymbol{\ell}) \rightarrow \varepsilon$

synchronization of nonterminals and terminals:

sDCP-rule: $\{2\}\ \boxed{\mathbf{M}} \rightarrow \varepsilon$ $\qquad$ $\boxed{x_1}\ \{4\}\ \boxed{\begin{array}{c}\boldsymbol{\ell}\\ |\\ x_1\end{array}} \rightarrow \varepsilon$

LCFRS-rule: $\{2\}\ (\mathbf{M}) \rightarrow \varepsilon$ $\qquad$ $\{4\}(\boldsymbol{\ell}) \rightarrow \varepsilon$

sDCP-rule: $\{2,4\}\ \boxed{x_2} \rightarrow \{2\}\boxed{x_1}\quad \boxed{x_1}\ \{4\}\ \boxed{x_2}$

LCFRS-rule: $\{2,4\}\ (x_1, x_2) \rightarrow \{2\}(x_1) \quad \{4\}\ (x_2)$

synchronization of nonterminals and terminals:

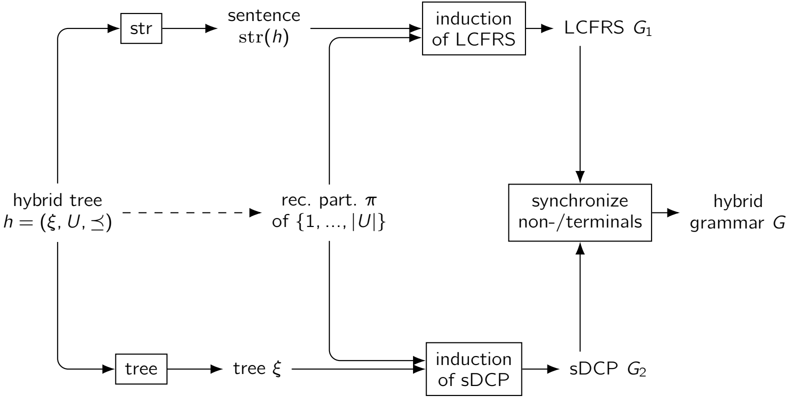
sDCP-rule: $\{2\}\ \boxed{\mathbf{M}} \to \varepsilon$ $\qquad$ $\boxed{x_1}\ \{4\}\ \boxed{\begin{array}{c}\boldsymbol{\ell}\\ | \\ x_1\end{array}} \to \varepsilon$

LCFRS-rule: $\{2\}\ (\mathbf{M}) \to \varepsilon$ $\qquad$ $\{4\}(\boldsymbol{\ell}) \to \varepsilon$

sDCP-rule: $\{2,4\}\ \boxed{x_2} \to \{2\}\boxed{x_1}\ \ \boxed{x_1}\ \{4\}\ \boxed{x_2}$

LCFRS-rule: $\{2,4\}\ (x_1, x_2) \to \{2\}(x_1)\ \ \{4\}\ (x_2)$

grammar induction from <u>one</u> hybrid tree:

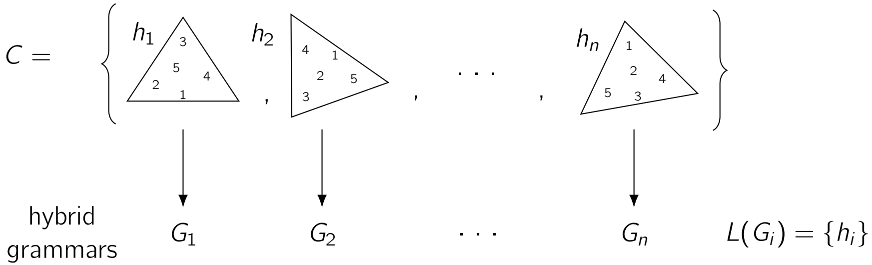


$L(G) = \{h\}$

parsing of $\mathrm{str}(h)$ according to π

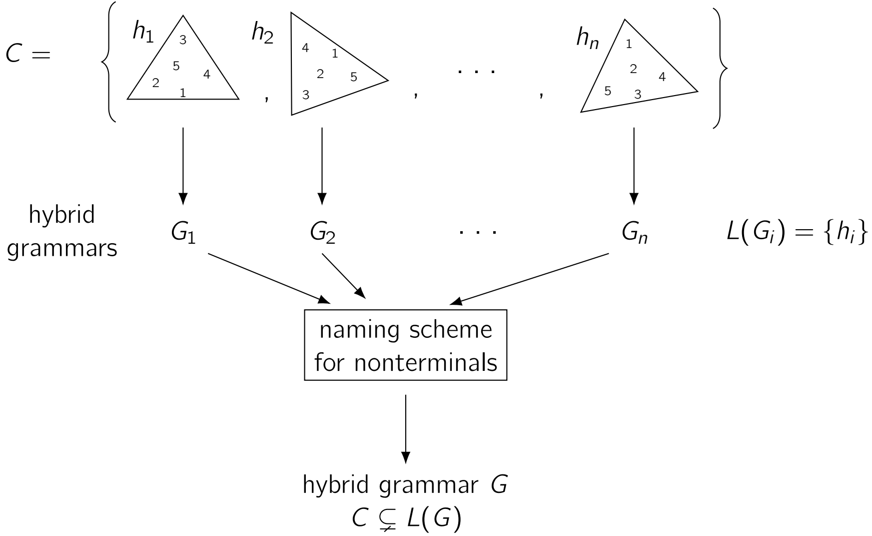
grammar induction from a corpus of hybrid trees:

corpus of hybrid trees $+$ choice of recursive partitioning



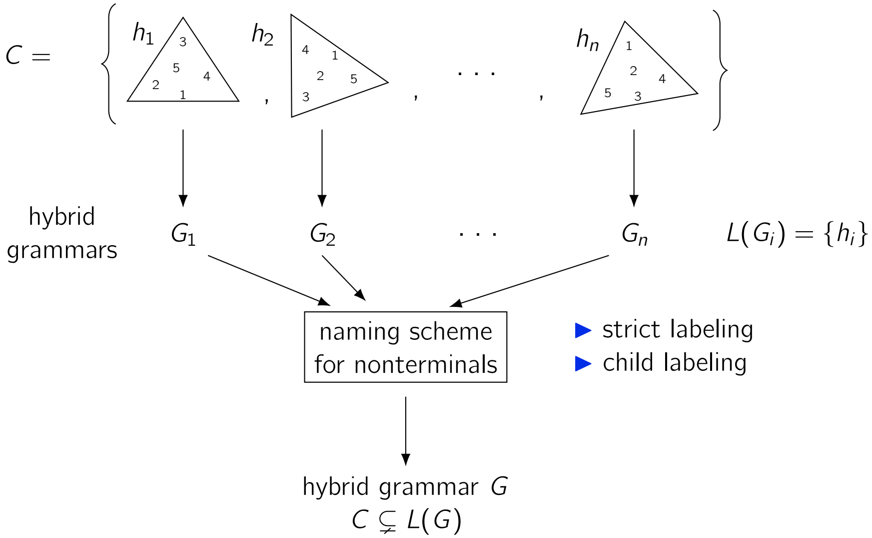
grammar induction from a corpus of hybrid trees:

corpus of hybrid trees $+$ choice of recursive partitioning



grammar induction from a corpus of hybrid trees:

corpus of hybrid trees $+$ choice of recursive partitioning



grammar formalisms	phrase structure trees		dependency trees		parsing compl.
	cont.	discont.	proj.	non-proj.	
REG/HMM	-	-	-	-	$\mathcal{O}(n)$
CFG	x	-	x	-	$\mathcal{O}(n^3)$ (CNF)
LCFRS	x	x	x	x	$\mathcal{O}(n^{3 \cdot \mathrm{fanout}(G)})$ (binar.)

(LCFRS,sDCP)-hybrid grammar:

	x	x	x	x	

choice of recursive partitioning:

- directly extracted and transformed to fanout $\leq k$	$\mathcal{O}(n^{3 \cdot k})$
- left-/right-branching	$\mathcal{O}(n)$

experiments for dependency parsing:

- NEGRA corpus; 16.509 German language sentences of length ≤ 25

experiments for dependency parsing:

- NEGRA corpus; 16.509 German language sentences of length ≤ 25
- split into: training corpus (14.858) and test corpus (1.651)

training corpus $C_1 \longrightarrow$ probabilistic hybrid grammar (G, p)

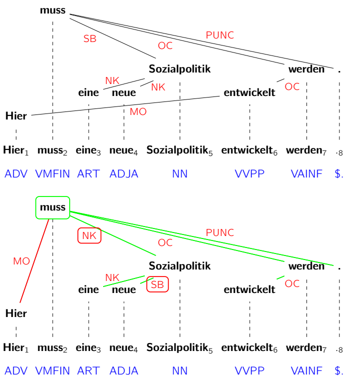
test corpus C_2: $\forall h \in C_2 : \; h \stackrel{?}{=} \mathrm{parsing}_{(G,p)}(\mathrm{str}(h))$

experiments for dependency parsing:

- NEGRA corpus; 16.509 German language sentences of length ≤ 25
- split into: training corpus (14.858) and test corpus (1.651)
- evaluation metrics:
  unlabeled attachement score (UAS), labeled attachment score (LAS), label accuracy (LA)

(a)

(b)



UAS: 7 / 8

LAS: 5 / 8

LA: 6 / 8

experiments for dependency parsing:

- NEGRA corpus; 16.509 German language sentences of length ≤ 25
- split into: training corpus (14.858) and test corpus (1.651)
- evaluation metrics:
  unlabeled attachement score (UAS), labeled attachment score (LAS), label accuracy (LA)
- parameters of our grammar induction:
  - naming scheme (strict labeling, child labeling)
  - argument label (POS+DEPREL, POS, DEPREL)
  - recursive partitioning (directly extracted, transformed to fanout k, left-/right-branching)

rec. part.	argument label	nont.	rules	$f_{\max}$	f_{avg}	fail	UAS	LAS	LA	time
child labeling										
direct	POS+DEPREL	4,739	27,042	7	1.10	693	52.2	40.8	42.6	253
$k = 1$	POS+DEPREL	13,178	35,071	1	1.00	202	77.0	69.1	73.3	288
$k = 2$	POS+DEPREL	11,156	32,231	2	1.17	195	77.7	70.1	74.2	355
r-branch	POS+DEPREL	42,577	79,648	1	1.00	775	45.7	32.8	34.7	49
l-branch	POS+DEPREL	40,100	75,321	1	1.00	768	46.0	33.2	35.0	45
direct	POS	675	19,276	7	1.24	303	69.3	50.0	55.4	300
$k = 1$	POS	3,464	15,826	1	1.00	30	**82.5**	63.5	70.6	244
$k = 2$	POS	2,099	13,347	2	1.40	35	82.4	63.3	70.5	410
r-branch	POS	19,804	51,733	1	1.00	372	62.7	44.7	50.6	222
l-branch	POS	17,240	45,883	1	1.00	342	63.9	45.6	51.4	197
direct	DEPREL	2,505	19,511	7	1.13	3	78.9	71.6	78.6	484
$k = 1$	DEPREL	8,059	22,613	1	1.00	1	79.5	71.7	79.0	608
$k = 2$	DEPREL	6,651	20,314	2	1.20	1	79.8	**72.0**	**79.2**	971
$k = 3$	DEPREL	6,438	19,962	3	1.25	1	79.5	71.9	79.1	1,013
r-branch	DEPREL	27,653	54,360	1	1.00	2	76.3	67.5	76.1	216
l-branch	DEPREL	25,699	50,418	1	1.00	1	76.2	67.6	76.1	198
cascade: child labeling, $k = 1$, POS+DEPREL/POS/DEPREL						1	**84.3**	**76.1**	**81.6**	325
LCFRS [Maier, Kallmeyer 10]:										
rparse simple		920	18,587	7	1.37	56	77.3	70.0	76.2	350
rparse ($v = 1$, $h = 3$)		40,141	61,450	7	1.10	13	78.5	71.4	79.0	778
MaltParser, unlexicalized, stacklazy [Nivre, Hall, Nilsson 06]						0	85.6	80.0	85.0	24

rec. part.	argument label	nont.	rules	$f_{\max}$	f_{avg}	fail	UAS	LAS	LA	time
child labeling										
direct	POS+DEPREL	4,739	27,042	7	1.10	693	52.2	40.8	42.6	253
$k=1$	POS+DEPREL	13,178	35,071	1	1.00	202	77.0	69.1	73.3	288
$k=2$	POS+DEPREL	11,156	32,231	2	1.17	195	77.7	70.1	74.2	355
r-branch	POS+DEPREL	42,577	79,648	1	1.00	775	45.7	32.8	34.7	49
l-branch	POS+DEPREL	40,100	75,321	1	1.00	768	46.0	33.2	35.0	45
direct	POS	675	19,276	7	1.24	303	69.3	50.0	55.4	300
$k=1$	POS	3,464	15,826	1	1.00	30	**82.5**	63.5	70.6	244
$k=2$	POS	2,099	13,347	2	1.40	35	82.4	63.3	70.5	410
r-branch	POS	19,804	51,733	1	1.00	372	62.7	44.7	50.6	222
l-branch	POS	17,240	45,883	1	1.00	342	63.9	45.6	51.4	197
direct	DEPREL	2,505	19,511	7	1.13	3	78.9	71.6	78.6	484
$k=1$	DEPREL	8,059	22,613	1	1.00	1	79.5	71.7	79.0	608
$k=2$	DEPREL	6,651	20,314	2	1.20	1	79.8	**72.0**	**79.2**	971
$k=3$	DEPREL	6,438	19,962	3	1.25	1	79.5	71.9	79.1	1,013
r-branch	DEPREL	27,653	54,360	1	1.00	2	76.3	67.5	76.1	216
l-branch	DEPREL	25,699	50,418	1	1.00	1	76.2	67.6	76.1	198
cascade: child labeling, $k=1$, POS+DEPREL/POS/DEPREL						1	**84.3**	**76.1**	**81.6**	325
LCFRS [Maier, Kallmeyer 10]:										
rparse simple		920	18,587	7	1.37	56	77.3	70.0	76.2	350
rparse ($v=1$, $h=3$)		40,141	61,450	7	1.10	13	78.5	71.4	79.0	778
MaltParser, unlexicalized, stacklazy [Nivre, Hall, Nilsson 06]						0	85.6	80.0	85.0	24

[Chomsky 56]: Three Models for the Description of Language.

> *"There are* **two central problems** *in the descriptive study of languages. One primary concern of the linguist is* **to discover simple and *revealing* grammars for natural languages**. *At the same time,* **by studying the properties of such successful grammars ...,** *he hopes to arrive at a general theory of linguistic structure."*

parsing of natural languages with ...

- probabilistic LCFRS
- probabilistic hybrid grammars
  - induction of probabilistic (LCFRS,sDCP)-hybrid grammars
  - experiments

[Chomsky 56]: Three Models for the Description of Language.

> *"There are* **two central problems** *in the descriptive study of languages. One primary concern of the linguist is* **to discover simple and *revealing* grammars for natural languages**. *At the same time,* **by studying the properties of such successful grammars ...,** *he hopes to arrive at a general theory of linguistic structure."*

parsing of natural languages with ...

- probabilistic LCFRS
- probabilistic hybrid grammars
  - induction of probabilistic (LCFRS,sDCP)-hybrid grammars
  - experiments

thanks!

References:

- [Chomsky 56] Three Models for the Description of Language.
- [Deransart, Maluszynski 85] Relating logic programs and attribute grammars.
- [Drewes, Gebhardt, HV 16] EM-training for weighted aligned hypergraph bimorphisms.
- [Gebhardt, Nederhof, HV 17] Hybrid grammars for parsing of discontinuous phrase structures and non-projective dependency structures.
- [Kallmeyer, Maier 10] Data-driven parsing with probabilistic linear context-free rewriting systems.
- [Kuhlmann, Satta 09] Treebank grammar techniques for non-projective dependency parsing.
- [Maier, Kallmeyer 10] Discontinuity and non-projectivity: Using mildly context-sensitive formalisms for data-driven parsing.
- [Nivre, Hall, Nilsson 06] Maltparser: A data-driven parser-generator for dependency parsing.
- [Rounds 70] Tree-oriented proofs of some theorems on context-free and indexed languages.
- [Salomaa, Soittola 78] Automata-Theoretic Aspects of Formal Power Series.
- [Seki, Matsumura, Fujii, Kasami 91] On multiple context-free grammars.
- [Tesniére 59] Éleménts de syntaxe structurale.
- [Vijay-Shanker, Weir, Joshi 87] Characterizing structural descriptions produced by various grammatical formalisms.

... a bit of theory of LCFRS:

[Vijay-Shanker, Weir, Joshi 87]
[Seki, Matsura, Fujii, Kasami 91]
[Seki, Kato 91]
[Denkinger 15] and others

m-LCFRS ($m \in \mathbb{N}$): m is maximal number of arguments of nonterminals (fan-out)

... a bit of theory of LCFRS:

[Vijay-Shanker, Weir, Joshi 87]
[Seki, Matsura, Fujii, Kasami 91]
[Seki, Kato 91]
[Denkinger 15] and others

m-LCFRS ($m \in \mathbb{N}$): m is maximal number of arguments of nonterminals (fan-out)

- 1-LCFRS = context-free languages

... a bit of theory of LCFRS:

[Vijay-Shanker, Weir, Joshi 87]
[Seki, Matsura, Fujii, Kasami 91]
[Seki, Kato 91]
[Denkinger 15] and others

m-LCFRS ($m \in \mathbb{N}$): m is maximal number of arguments of nonterminals (fan-out)

- 1-LCFRS = context-free languages
- $\{a_1^n a_2^n \ldots a_{2m}^n \mid n \geq 0\} \in m\text{-LCFRS} \setminus (m-1)\text{-LCFRS}$ for each $m \in \mathbb{N}$

... a bit of theory of LCFRS:

[Vijay-Shanker, Weir, Joshi 87]
[Seki, Matsura, Fujii, Kasami 91]
[Seki, Kato 91]
[Denkinger 15] and others

m-LCFRS ($m \in \mathbb{N}$): m is maximal number of arguments of nonterminals (fan-out)

- 1-LCFRS = context-free languages
- $\{a_1^n a_2^n \dots a_{2m}^n \mid n \geq 0\} \in m\text{-LCFRS} \setminus (m-1)\text{-LCFRS}$ for each $m \in \mathbb{N}$
- $\{(a^m b^m)^n \mid m, n \geq 1\}$ is not LCFRS.

... a bit of theory of LCFRS:

[Vijay-Shanker, Weir, Joshi 87]
[Seki, Matsura, Fujii, Kasami 91]
[Seki, Kato 91]
[Denkinger 15] and others

m-LCFRS ($m \in \mathbb{N}$): m is maximal number of arguments of nonterminals (fan-out)

- 1-LCFRS = context-free languages
- $\{a_1^n a_2^n \dots a_{2m}^n \mid n \geq 0\} \in m\text{-LCFRS} \setminus (m-1)\text{-LCFRS}$
  for each $m \in \mathbb{N}$
- $\{(a^m b^m)^n \mid m, n \geq 1\}$ is not LCFRS.
- m-LCFRS is a substitution-closed full AFL.

... a bit of theory of LCFRS:

[Vijay-Shanker, Weir, Joshi 87]
[Seki, Matsura, Fujii, Kasami 91]
[Seki, Kato 91]
[Denkinger 15] and others

m-LCFRS ($m \in \mathbb{N}$): m is maximal number of arguments of nonterminals (fan-out)

- 1-LCFRS = context-free languages
- $\{a_1^n a_2^n \dots a_{2m}^n \mid n \geq 0\} \in m$-LCFRS $\setminus (m-1)$-LCFRS for each $m \in \mathbb{N}$
- $\{(a^m b^m)^n \mid m, n \geq 1\}$ is not LCFRS.
- m-LCFRS is a substitution-closed full AFL.
- well-nested LCFRS are linear macro languages.

... a bit of theory of LCFRS:

[Vijay-Shanker, Weir, Joshi 87]
[Seki, Matsura, Fujii, Kasami 91]
[Seki, Kato 91]
[Denkinger 15] and others

m-LCFRS ($m \in \mathbb{N}$): m is maximal number of arguments of nonterminals (fan-out)

- 1-LCFRS = context-free languages
- $\{a_1^n a_2^n \dots a_{2m}^n \mid n \geq 0\} \in m\text{-LCFRS} \setminus (m-1)\text{-LCFRS}$ for each $m \in \mathbb{N}$
- $\{(a^m b^m)^n \mid m, n \geq 1\}$ is not LCFRS.
- m-LCFRS is a substitution-closed full AFL.
- well-nested LCFRS are linear macro languages.
- Let G binarized m-LCFRS and $w \in \Sigma^*$. It can decided in $\mathcal{O}(|w|^{3 \cdot \mathrm{fanout}(G)})$ whether $w \in L(G)$.

... a bit of theory of LCFRS:

[Vijay-Shanker, Weir, Joshi 87]
[Seki, Matsura, Fujii, Kasami 91]
[Seki, Kato 91]
[Denkinger 15] and others

m-LCFRS ($m \in \mathbb{N}$): m is maximal number of arguments of nonterminals (fan-out)

- 1-LCFRS = context-free languages
- $\{a_1^n a_2^n \ldots a_{2m}^n \mid n \geq 0\} \in m\text{-LCFRS} \setminus (m-1)\text{-LCFRS}$ for each $m \in \mathbb{N}$
- $\{(a^m b^m)^n \mid m, n \geq 1\}$ is not LCFRS.
- m-LCFRS is a substitution-closed full AFL.
- well-nested LCFRS are linear macro languages.
- Let G binarized m-LCFRS and $w \in \Sigma^*$. It can decided in $\mathcal{O}(|w|^{3 \cdot \mathrm{fanout}(G)})$ whether $w \in L(G)$.
- Chomsky-Schützenberger theorem for weighted LCFRS

synchronous grammars:
(two grammars of the same type; synchronization via nonterminals)

- transduction grammars [Lewis, Stearns 68]
  = synchronous context-free grammars [Chiang 07]
- generalized syntax-directed translation schemes [Aho, Ullman 71]
- synchronous tree-substitution grammars [Schabes 90]
- synchronous tree-adjoining grammars [Abeillé, Schabes, Joshi 90; Shieber, Schabes 90]
- synchronous context-free tree grammars [Nederhof, V. 12]
- synchronous linear context-free rewriting systems [Kaeshammer 13]

corpora for phrase structure trees

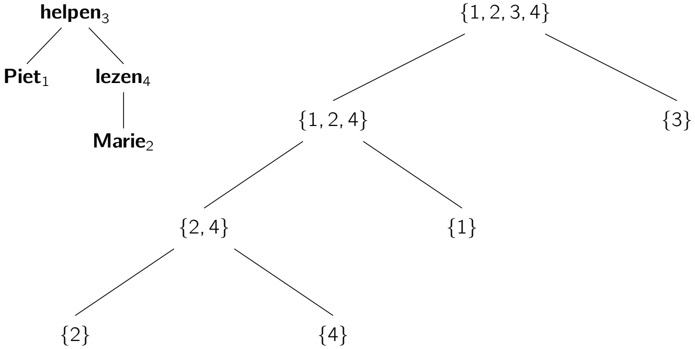
corpus = large, finite set

- TIGER treebank (German, 50k sentences)
- NEGRA corpus (German, 20k sentences)
- Japanese Verbmobil treebank (Japanese, 20k sentences)
- The Bosque part of the Floresta sintà(c)tica (Portuguese, 41k sentences)
- Alpino treebank (Dutch, 150k words)
- Sinica treebank (Chinese, 361k words)
- ...

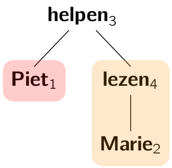
corpora for dependency trees

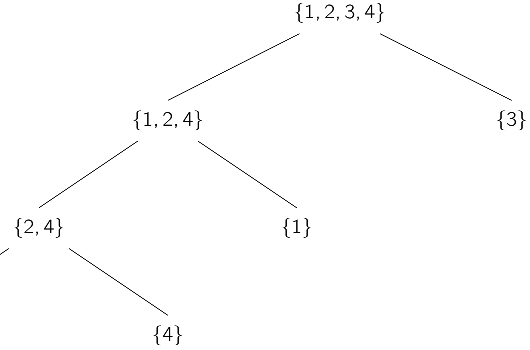
- Prague Dependency Treebank (Czech, 26k sentences)
- Slovene Dependency Treebank (Slovene, 30k words)
- Danish Dependency Treebank (Danish, 5k sentence)
- Talbanken05 (Swedish, 30k words)
- Metu-Sabancı treebank (Turkish, 7k sentences)
- ...

induction of sDCP G_2:



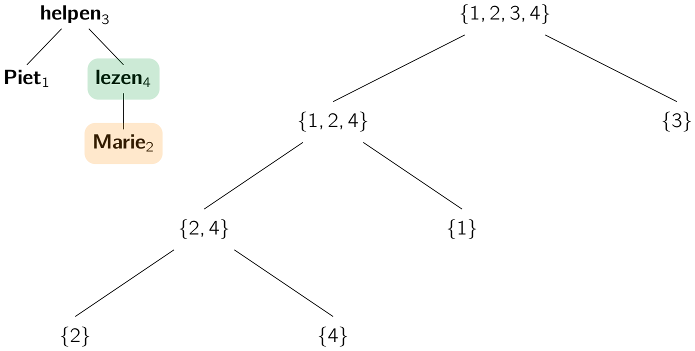
induction of sDCP G_2:





$$\{1,2,4\}\ \boxed{\mathbf{P}}\ \boxed{\boldsymbol{\ell}(\mathbf{M})} \Rightarrow^* \varepsilon$$

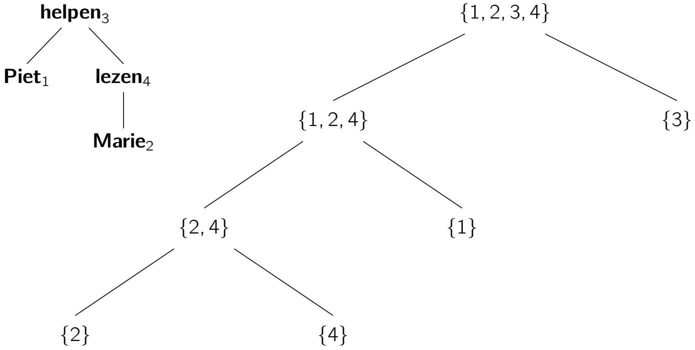
induction of sDCP G_2:



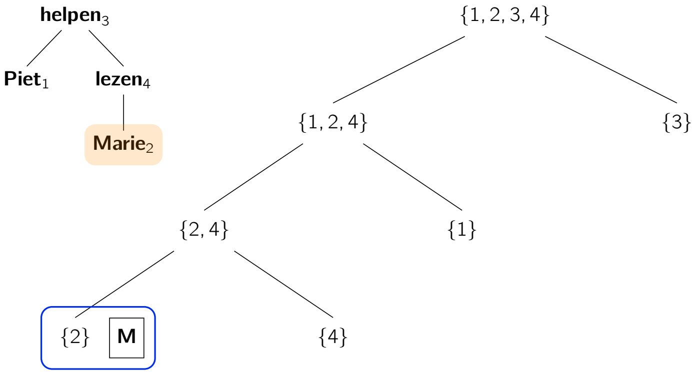
$$\{1,2,4\}\ \boxed{\mathbf{P}}\ \boxed{\boldsymbol{\ell}(\mathbf{M})} \quad \Rightarrow^* \quad \varepsilon$$

$$\boxed{\mathbf{M}}\ \{4\}\ \boxed{\boldsymbol{\ell}(\mathbf{M})} \quad \Rightarrow^* \quad \varepsilon$$

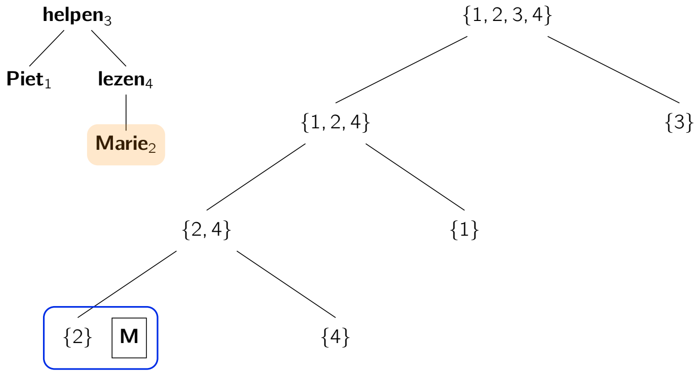
induction of sDCP G_2:



induction of sDCP G_2:

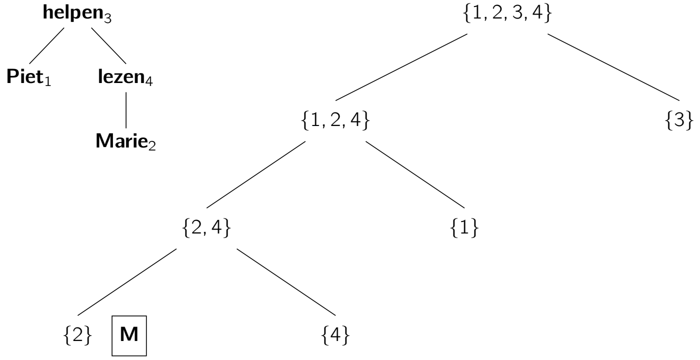


induction of sDCP G_2:



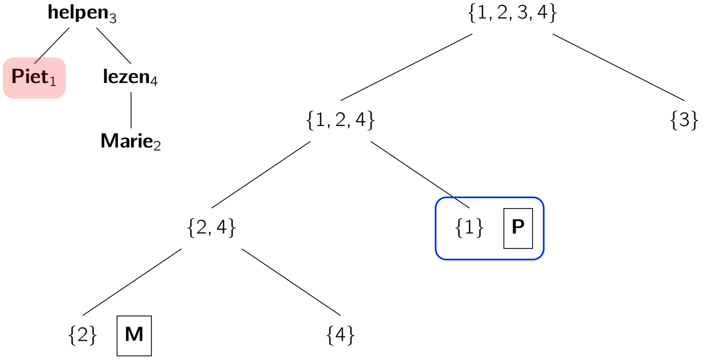
$\{2\}$ $\boxed{\textbf{Marie}}$ $\rightarrow$ ε

induction of sDCP G_2:



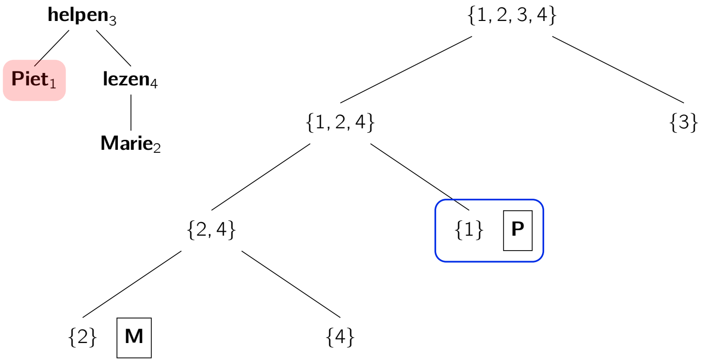
$$\{2\} \ \boxed{\textbf{Marie}} \ \rightarrow \ \varepsilon$$

induction of sDCP G_2:



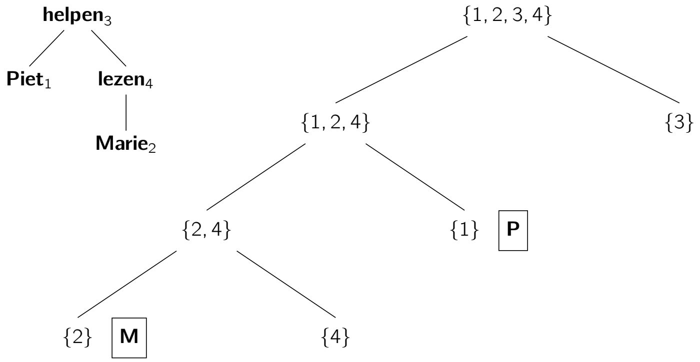
$$\{2\}\ \boxed{\textbf{Marie}} \rightarrow \varepsilon$$

induction of sDCP G_2:



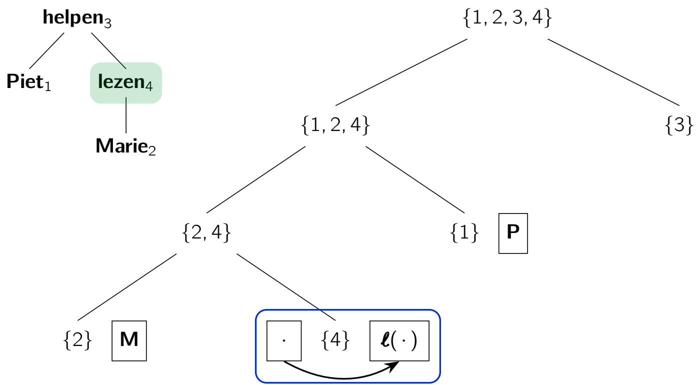
$\{2\} \; \boxed{\textbf{Marie}} \rightarrow \varepsilon$ $\qquad$ $\{1\} \; \boxed{\textbf{Piet}} \rightarrow \varepsilon$

induction of sDCP G_2:



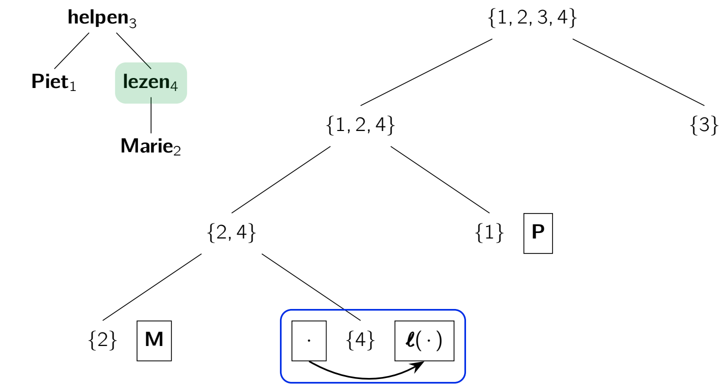
$\{2\}\quad \boxed{\textbf{Marie}} \;\rightarrow\; \varepsilon \qquad\qquad \{1\}\quad \boxed{\textbf{Piet}} \;\rightarrow\; \varepsilon$

induction of sDCP G_2:



$\{2\}\ \boxed{\textbf{Marie}} \rightarrow \varepsilon$ $\{1\}\ \boxed{\textbf{Piet}} \rightarrow \varepsilon$

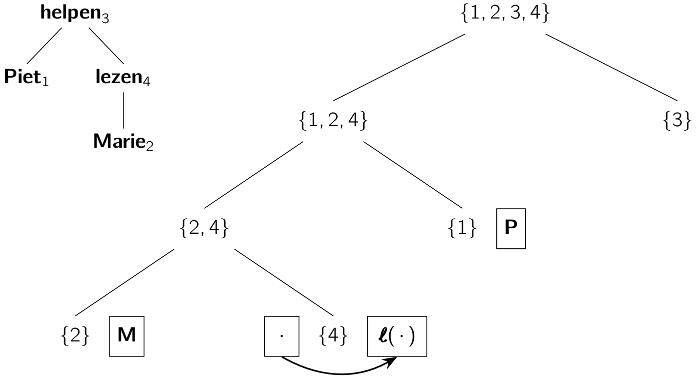
induction of sDCP G_2:



$$\{2\}\ \boxed{\textbf{Marie}} \rightarrow \varepsilon \qquad\qquad \{1\}\ \boxed{\textbf{Piet}} \rightarrow \varepsilon$$

$$\boxed{z_1}\ \{4\}\ \boxed{\begin{array}{c}\textbf{lezen}\\ | \\ z_1\end{array}} \rightarrow \varepsilon$$

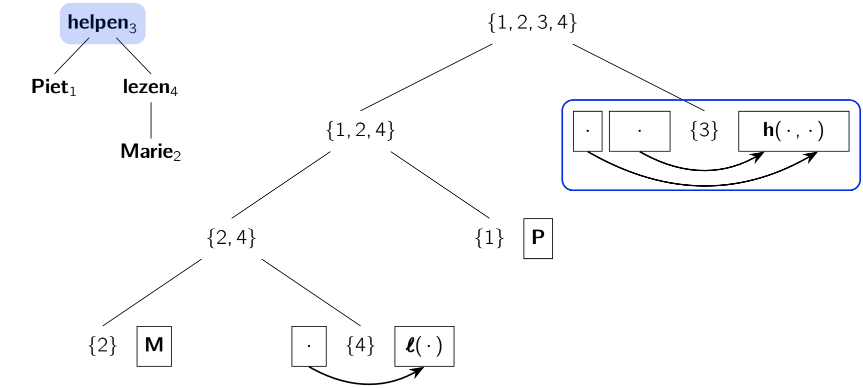
induction of sDCP G_2:



$$\{2\}\ \boxed{\textbf{Marie}} \rightarrow \varepsilon \qquad\qquad \{1\}\ \boxed{\textbf{Piet}} \rightarrow \varepsilon$$

$$\boxed{z_1}\ \{4\}\ \boxed{\begin{array}{c}\textbf{lezen}\\ | \\ z_1\end{array}} \rightarrow \varepsilon$$

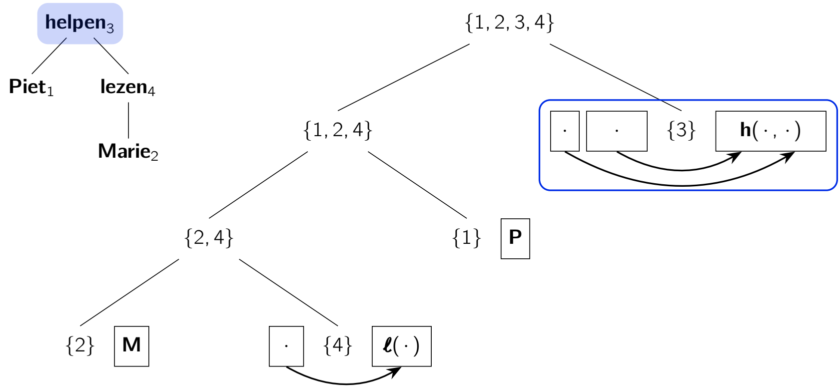
induction of sDCP G_2:



$\{2\}\ \boxed{\textbf{Marie}} \to \varepsilon$ $\qquad$ $\{1\}\ \boxed{\textbf{Piet}} \to \varepsilon$

$\boxed{z_1}\ \{4\}\ \boxed{\begin{array}{c}\textbf{lezen}\\ | \\ z_1\end{array}} \to \varepsilon$

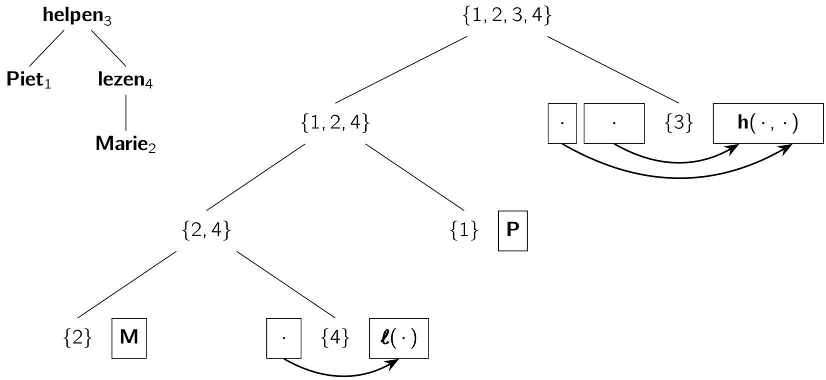
induction of sDCP G_2:



$\{2\}$ $\boxed{\textbf{Marie}} \rightarrow \varepsilon$ $\{1\}$ $\boxed{\textbf{Piet}} \rightarrow \varepsilon$

$\boxed{z_1}$ $\{4\}$ $\boxed{\textbf{lezen}(z_1)} \rightarrow \varepsilon$ $\boxed{z_1}\ \boxed{z_2}$ $\{3\}$ $\boxed{\textbf{helpen}(z_1, z_2)} \rightarrow \varepsilon$

induction of sDCP G_2:



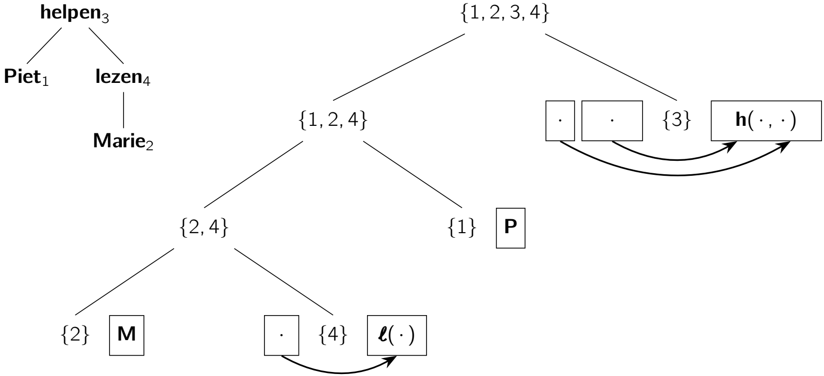
$\{2\}\ \boxed{\textbf{Marie}} \ \to\ \varepsilon$

$\{1\}\ \boxed{\textbf{Piet}} \ \to\ \varepsilon$

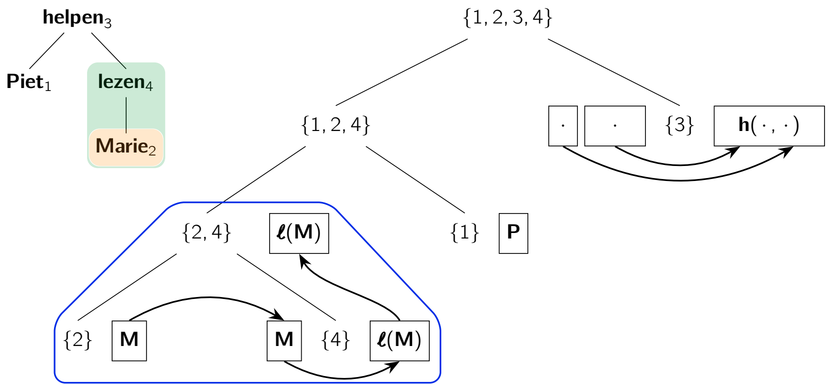
$\boxed{z_1}\ \ \{4\}\ \boxed{\begin{array}{c}\textbf{lezen}\\ | \\ z_1\end{array}} \ \to\ \varepsilon$

$\boxed{z_1}\ \boxed{z_2}\ \ \{3\}\ \boxed{\begin{array}{c}\textbf{helpen}\\ / \quad \backslash \\ z_1 \quad\ z_2\end{array}} \ \to\ \varepsilon$

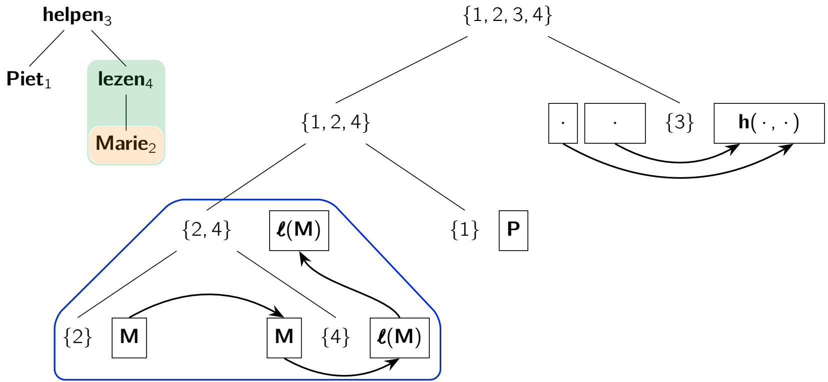
induction of sDCP G_2:



induction of sDCP G_2:

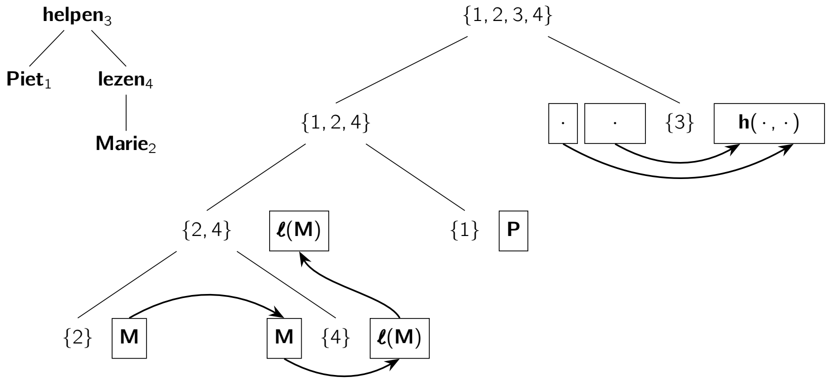


induction of sDCP G_2:



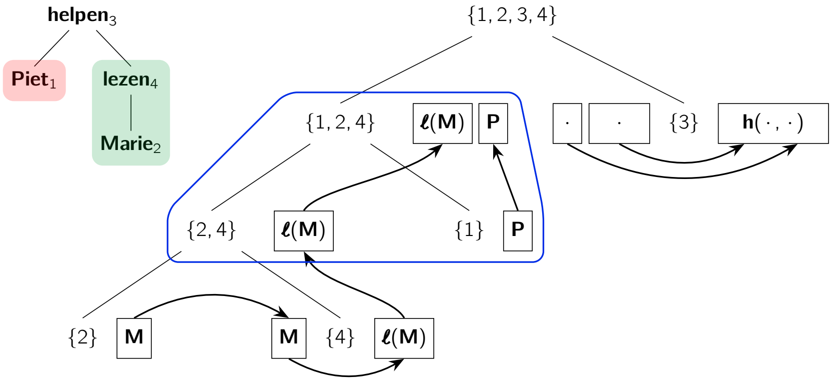
$\{2,4\}\ \boxed{z_2} \quad \rightarrow \quad \{2\}\boxed{z_1} \qquad \boxed{z_1}\ \{4\}\ \boxed{z_2}$

induction of sDCP G_2:



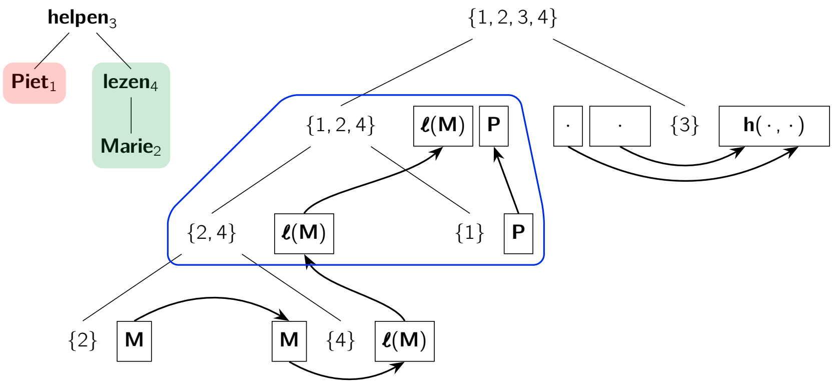
$$\{2,4\}\,\boxed{z_2} \;\rightarrow\; \{2\}\boxed{z_1} \quad \boxed{z_1}\,\{4\}\,\boxed{z_2}$$

induction of sDCP G_2:



$$\{2,4\}\,\boxed{z_2} \quad \to \quad \{2\}\boxed{z_1} \quad \boxed{z_1}\,\{4\}\,\boxed{z_2}$$

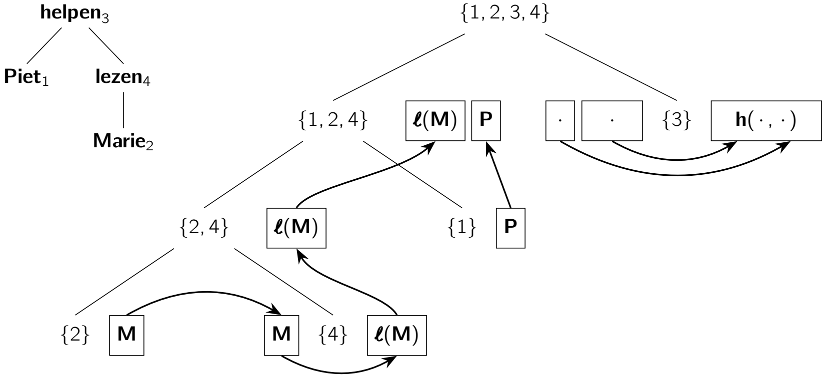
induction of sDCP G_2:



$$\{2,4\}\,\boxed{z_2} \;\rightarrow\; \{2\}\boxed{z_1} \quad \boxed{z_1}\,\{4\}\,\boxed{z_2}$$

$$\{1,2,4\}\,\boxed{z_1}\,\boxed{z_2} \;\rightarrow\; \{2,4\}\,\boxed{z_1} \quad \{1\}\,\boxed{z_2}$$

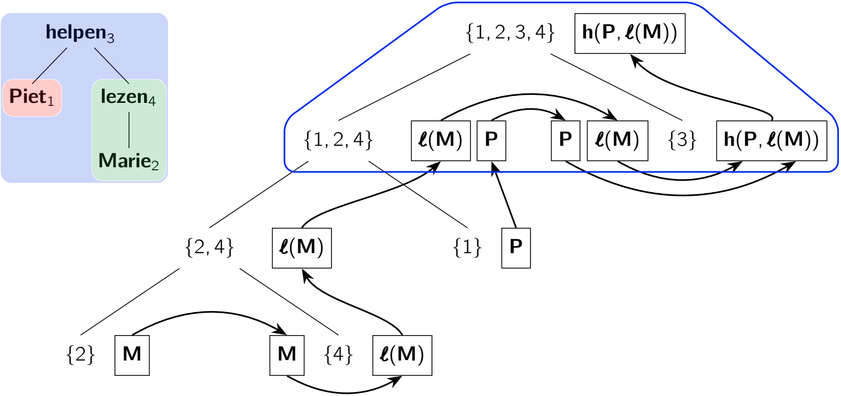
induction of sDCP G_2:



$$\{2,4\}\,\boxed{z_2} \;\rightarrow\; \{2\}\,\boxed{z_1} \quad \boxed{z_1}\,\{4\}\,\boxed{z_2}$$

$$\{1,2,4\}\,\boxed{z_1}\,\boxed{z_2} \;\rightarrow\; \{2,4\}\,\boxed{z_1} \quad \{1\}\,\boxed{z_2}$$

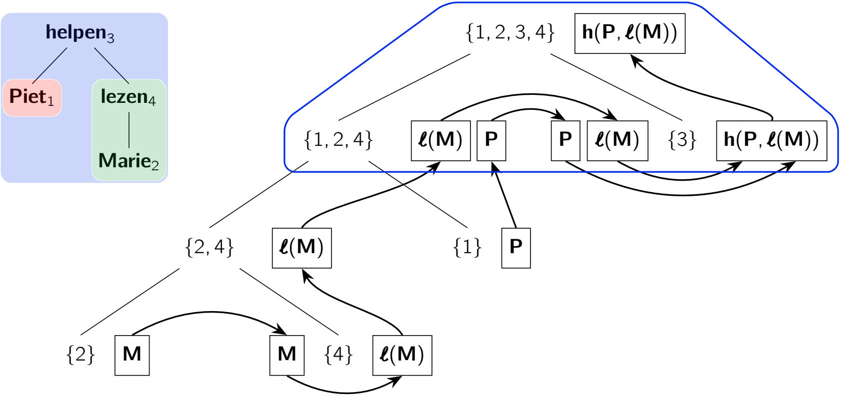
induction of sDCP G_2:



$$\{2,4\}\ \boxed{z_2} \quad \rightarrow \quad \{2\}\boxed{z_1} \quad \boxed{z_1}\ \{4\}\ \boxed{z_2}$$

$$\{1,2,4\}\ \boxed{z_1}\ \boxed{z_2} \quad \rightarrow \quad \{2,4\}\ \boxed{z_1} \quad \{1\}\ \boxed{z_2}$$

induction of sDCP G_2:



$$\{2,4\}\,\boxed{z_2} \;\rightarrow\; \{2\}\boxed{z_1} \quad \boxed{z_1}\,\{4\}\,\boxed{z_2}$$

$$\{1,2,4\}\,\boxed{z_1}\,\boxed{z_2} \;\rightarrow\; \{2,4\}\,\boxed{z_1} \quad \{1\}\,\boxed{z_2}$$

$$\{1,2,3,4\}\,\boxed{z_3} \;\rightarrow\; \{1,2,4\}\,\boxed{z_1}\,\boxed{z_2} \quad \boxed{z_2}\,\boxed{z_1}\,\{3\}\,\boxed{z_3}$$