Lexikalisch-Funktionale Grammatik (LFG), Teil 2

Vorlesung “Grammatikformalismen”

Alexander Koller

18. Juni 2019

Grammatische Strukturen

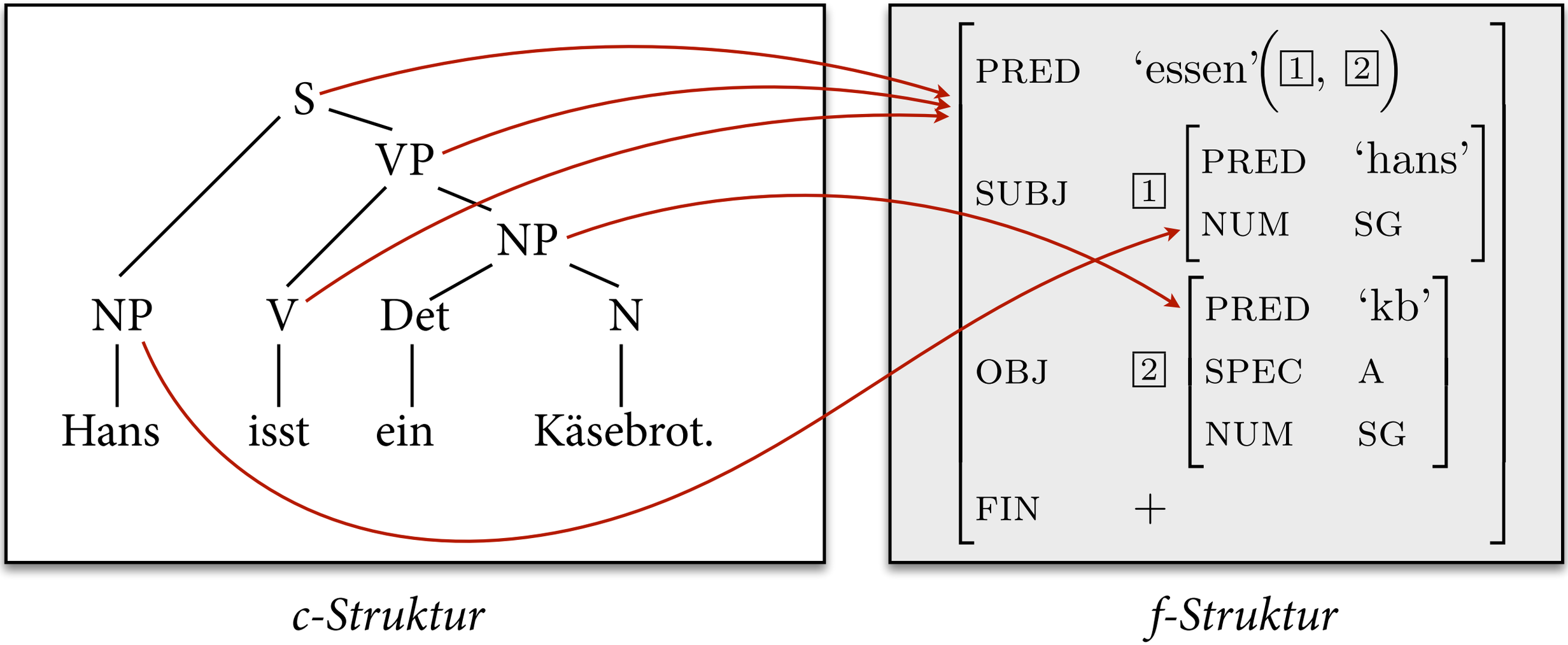
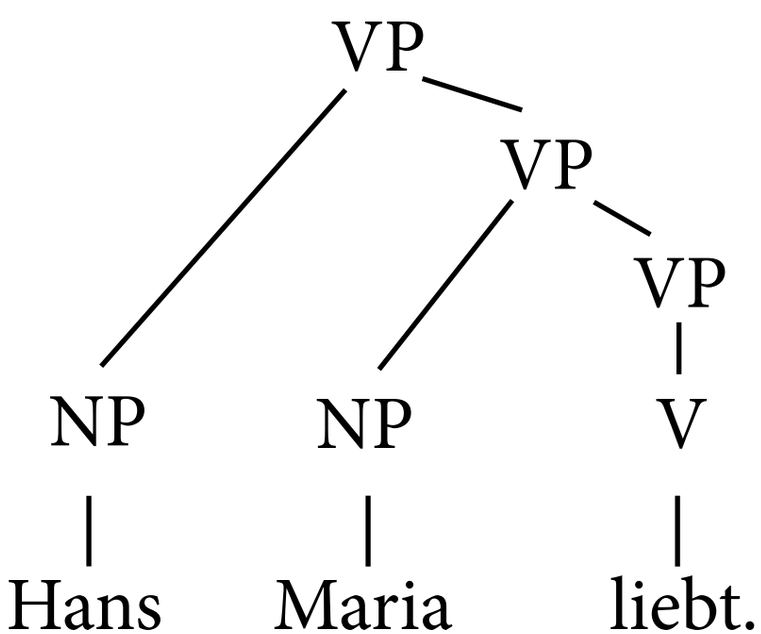


Abbildung ϕ

Verbletztstellung

VP	$\rightarrow$	V		Hans	NP	$(\uparrow \textsc{pred}) = \text{‘Hans’}$
		$\uparrow = \downarrow$				$(\uparrow \textsc{agr case}) = \textsc{nom}$
VP	$\rightarrow$	NP	VP	Maria	NP	$(\uparrow \textsc{pred}) = \text{‘Maria’}$
		$(\uparrow \textsc{subj}) = \downarrow$	$\uparrow = \downarrow$			$(\uparrow \textsc{agr case}) = \textsc{acc}$
VP	$\rightarrow$	NP	VP	liebt	V	$(\uparrow \textsc{pred}) = \text{‘lieben’}\langle(\uparrow \textsc{subj}), (\uparrow \textsc{obj})\rangle$
		$(\uparrow \textsc{obj}) = \downarrow$	$\uparrow = \downarrow$			$(\uparrow \textsc{subj agr case}) = \textsc{nom}$
						$(\uparrow \textsc{obj agr case}) = \textsc{acc}$



$$\begin{bmatrix} \textsc{pred} & \text{‘lieben’}\left\langle \left(\uparrow \textsc{subj}\right), \left(\uparrow \textsc{obj}\right) \right\rangle \\ \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{‘hans’} \\ \textsc{agr} & \begin{bmatrix} \textsc{case} & \textsc{nom} \end{bmatrix} \end{bmatrix} \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{‘maria’} \\ \textsc{agr} & \begin{bmatrix} \textsc{case} & \textsc{acc} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Präpositionalphrasen

- PPs können als Komplemente oder als Adjunkte auftreten.

- Allgemeine Regel für PPs:

$$\begin{array}{ccccc} \text{PP} & \rightarrow & \text{P} & & \text{NP} \\ & & \uparrow{=}\downarrow & & (\uparrow\textsc{obj}) = \downarrow \end{array}$$

- Lexikoneinträge für Präpositionen:

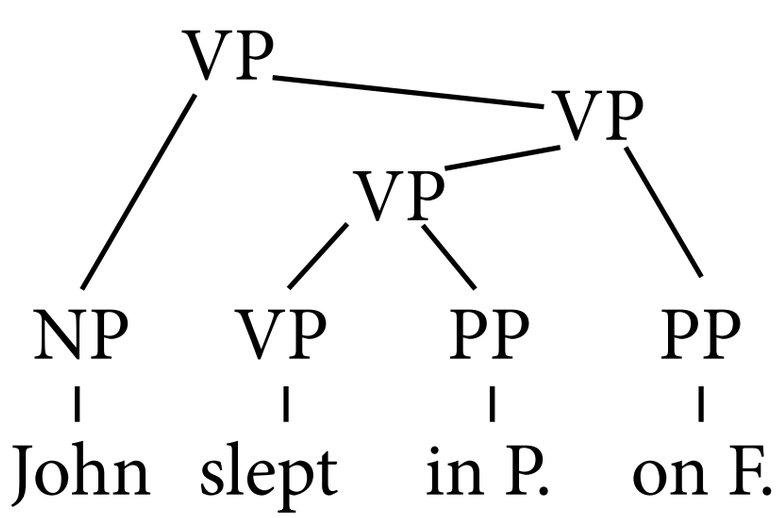
$$\begin{array}{lll} \text{to} & \text{P} & (\uparrow \textsc{pred}) = \text{‘to’}\langle(\uparrow\textsc{obj})\rangle \\ \text{in} & \text{P} & (\uparrow \textsc{pred}) = \text{‘loc’}\langle(\uparrow\textsc{obj})\rangle \end{array}$$

PPs als Adjunkte

- Adjunkte mit mengenwertigem ADJ-Feature. ADJ ist nicht regierbar, also nicht von Kohärenz und Vollständigkeit betroffen.

$$\begin{array}{ccc} \text{VP} & \rightarrow & \text{VP} \quad \text{PP} \\ & & \uparrow = \downarrow \quad \downarrow \in (\uparrow \text{ADJ}) \end{array}$$

"John slept in Potsdam on Friday"



$$\begin{bmatrix} \text{PRED} & \text{'sleep'}\langle(\uparrow\text{SUBJ})\rangle \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{'john'}\end{bmatrix} \\ \text{ADJ} & \left\{\begin{bmatrix}\text{PRED} & \text{'loc'}\langle(\uparrow\text{OBJ})\rangle \\ \text{OBJ} & \begin{bmatrix}\text{PRED} & \text{'potsdam'}\end{bmatrix}\end{bmatrix}, \begin{bmatrix}\text{PRED} & \text{'time'}\langle(\uparrow\text{OBJ})\rangle \\ \text{OBJ} & \begin{bmatrix}\text{PRED} & \text{'friday'}\end{bmatrix}\end{bmatrix}\right\} \end{bmatrix}$$

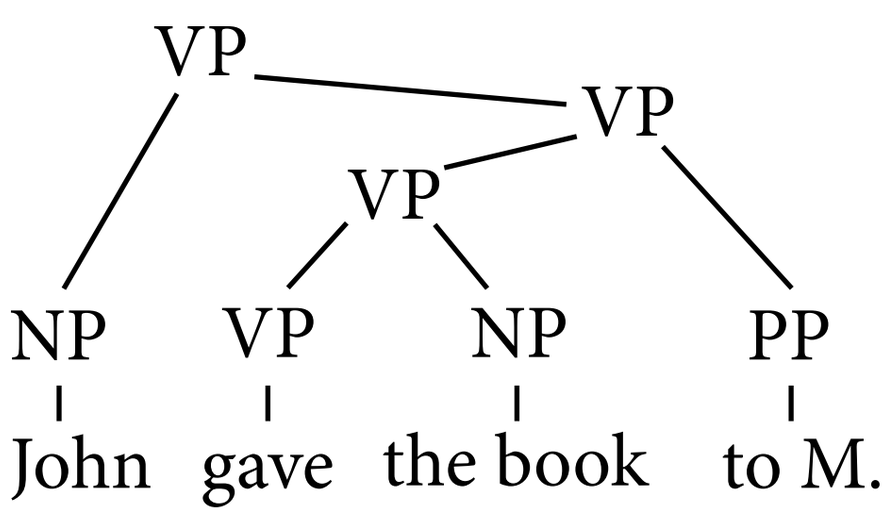
PPs als Komplemente

- PP-Komplement muss sagen, welche (regierbare) grammatische Rolle des Funktors es füllen möchte.
  - Verwende Schreibweise (↓ PCASE): Evaluiert zum Wert des PCASE-Features des Kindes.

$$\begin{array}{lcll} \text{VP} & \rightarrow & \text{VP} & \text{PP} \\ & & \uparrow = \downarrow & (\uparrow(\downarrow \text{PCASE})) = \downarrow \end{array}$$

$$\begin{array}{lll} \text{to} & \text{P} & (\uparrow \textsc{pred}) = \text{‘to’}\langle(\uparrow \textsc{obj})\rangle \\ & & (\uparrow \text{PCASE}) = \textsc{obl-dir} \end{array}$$

“John gave the book to Mary”



$$\begin{bmatrix} \textsc{pred} & \text{‘give’}\left\langle (\uparrow \textsc{subj}), (\uparrow \textsc{obj}), (\uparrow \textsc{obl-dir}) \right\rangle \\ \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{‘john’} \end{bmatrix} \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{‘book’} \end{bmatrix} \\ \textsc{obl-dir} & \begin{bmatrix} \textsc{pred} & \text{‘to’}\left\langle (\uparrow \textsc{obj}) \right\rangle \\ \textsc{obj} & \begin{bmatrix} \textsc{pred} & \text{‘mary’} \end{bmatrix} \\ \textsc{pcase} & \textsc{obl-dir} \end{bmatrix} \end{bmatrix}$$

PCASE und Passiv

- Passiv kann einfach als lexikalische Ambiguität des Verbs beschrieben werden:

$$\begin{array}{lll} \text{liebt} & \text{V} & (\uparrow \textsc{pred}) = \text{‘lieben’}\langle(\uparrow\textsc{subj}), (\uparrow\textsc{obj})\rangle \\ & & (\uparrow\textsc{subj agr case}) = \textsc{nom} \\ & & (\uparrow\textsc{obj agr case}) = \textsc{acc} \end{array}$$

$$\begin{array}{lll} \text{geliebt} & \text{V} & (\uparrow \textsc{pred}) = \text{‘lieben’}\langle(\uparrow\textsc{obl-ag}), (\uparrow\textsc{subj})\rangle \\ & & (\uparrow\textsc{subj agr case}) = \textsc{nom} \end{array}$$

- Nützt PP-Maschinerie in Kombination mit Kohärenz/Vollständigkeit aus.

Heute

- Fernabhängigkeiten / Functional Uncertainty.
- Expressivität von LFG.
- Parsing von LFG.
- Wahrscheinlichkeitsmodelle für LFG.

Fernabhängigkeiten in LFG

- Grammatikalität:
  - c-Struktur: korrekter Parsebaum der kfG.
  - f-Struktur: eindeutig, kohärent, vollständig und erfüllt alle f-Struktur-Constraints aus der Grammatik

- Präd-Arg-Struktur muss in der *f-Struktur* richtig herauskommen; c-Struktur eher Mittel zum Zweck.

- Fernabhängigkeiten: Unifiziere zwei f-Knoten, die in der c-Struktur weit auseinanderstehen.

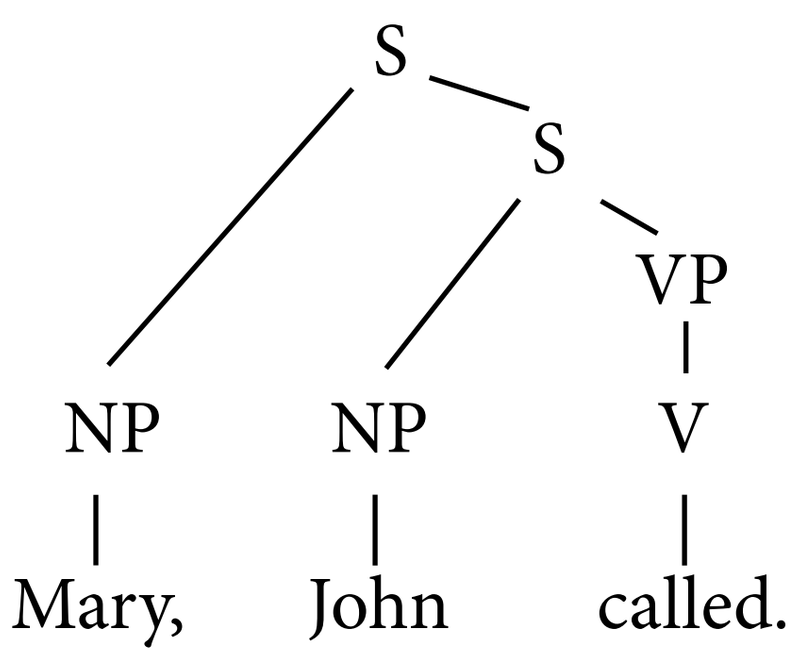
Fernabhängigkeiten

$$\mathrm{S} \rightarrow \underset{\substack{(\uparrow \textsc{topic}) = \downarrow \\ (\uparrow \textsc{topic}) = (\uparrow \textsc{obj})}}{\mathrm{NP}} \quad \underset{\uparrow = \downarrow}{\mathrm{S}}$$

$$\mathrm{S} \rightarrow \underset{(\uparrow \textsc{subj}) = \downarrow}{\mathrm{NP}} \quad \underset{\uparrow = \downarrow}{\mathrm{VP}}$$

$$\mathrm{VP} \rightarrow \underset{\uparrow = \downarrow}{\mathrm{V}} \quad \left(\underset{(\uparrow \textsc{obj}) = \downarrow)}{\mathrm{NP}} \right)$$

$$\text{called} \quad \mathrm{V} \quad (\uparrow \textsc{pred}) = \text{‘call’}\langle(\uparrow \textsc{subj}), (\uparrow \textsc{obj})\rangle$$



$$\begin{bmatrix} \textsc{pred} & \text{‘call’}\left\langle \left(\uparrow \textsc{subj}\right), \left(\uparrow \textsc{obj}\right) \right\rangle \\ \textsc{subj} & \left[\textsc{pred} \quad \text{‘john’} \right] \\ \textsc{obj} & \boxed{1} \\ \textsc{topic} & \boxed{1} \left[\textsc{pred} \quad \text{‘mary’} \right] \end{bmatrix}$$

Fernabhängigkeiten

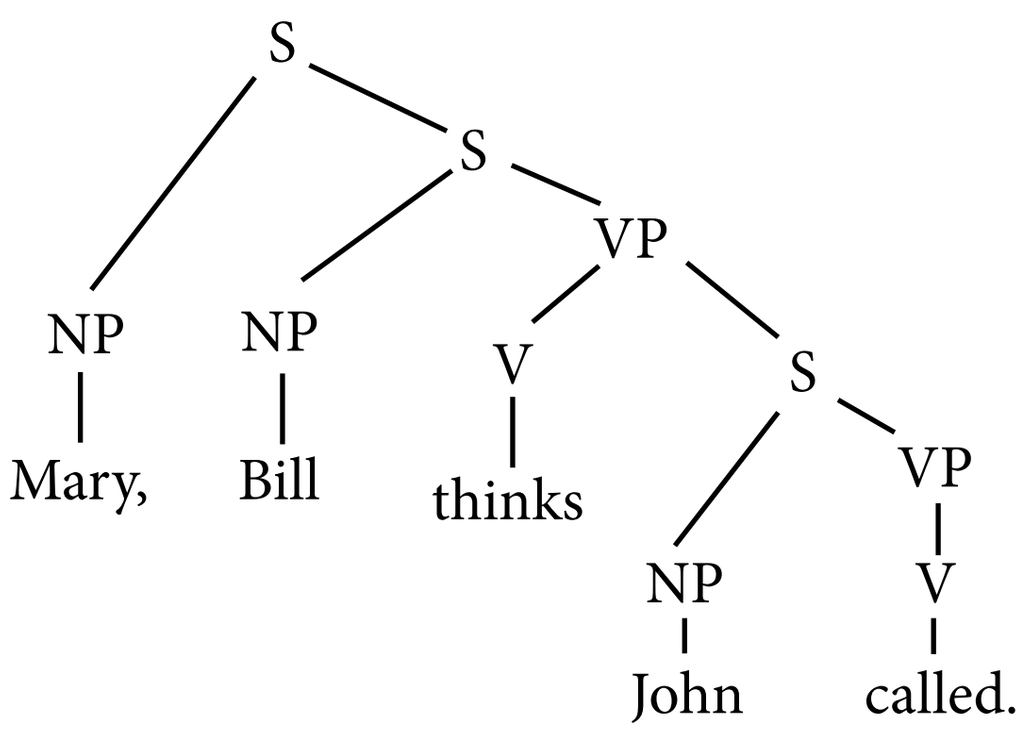
$$\text{S} \rightarrow \underset{\substack{(\uparrow \text{TOPIC}) = \downarrow \\ (\uparrow \text{TOPIC}) = (\uparrow \text{COMP OBJ})}}{\text{NP}} \quad \underset{\uparrow = \downarrow}{\text{S}}$$

$$\text{S} \rightarrow \underset{(\uparrow \text{SUBJ}) = \downarrow}{\text{NP}} \quad \underset{\uparrow = \downarrow}{\text{VP}}$$

$$\text{VP} \rightarrow \underset{\uparrow = \downarrow}{\text{V}} \left(\underset{(\uparrow \text{OBJ}) = \downarrow)}{\text{NP}} \right) \left(\underset{(\uparrow \text{COMP}) = \downarrow}{\text{S}} \right)$$

$$\text{called} \quad \text{V} \quad (\uparrow \text{PRED}) = \text{‘call’}\langle(\uparrow \text{SUBJ}), (\uparrow \text{OBJ})\rangle$$

$$\text{thinks} \quad \text{V} \quad (\uparrow \text{PRED}) = \text{‘think’}\langle(\uparrow \text{SUBJ}), (\uparrow \text{COMP})\rangle$$



$$\begin{bmatrix} \text{PRED} & \text{‘think’}\left\langle \left(\uparrow \text{SUBJ}\right), \left(\uparrow \text{COMP}\right) \right\rangle \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{‘bill’} \end{bmatrix} \\ \text{COMP} & \begin{bmatrix} \text{PRED} & \text{‘call’}\left\langle \left(\uparrow \text{SUBJ}\right), \left(\uparrow \text{OBJ}\right) \right\rangle \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{‘john’} \end{bmatrix} \\ \text{OBJ} & \boxed{1} \end{bmatrix} \\ \text{TOPIC} & \boxed{1} \begin{bmatrix} \text{PRED} & \text{‘mary’} \end{bmatrix} \end{bmatrix}$$

Fernabhängigkeiten

- Problem: für jede Einbettungstiefe braucht man separate Regel.

$$(\uparrow \textsc{topic}) = (\uparrow \textsc{obj})$$
$$(\uparrow \textsc{topic}) = (\uparrow \textsc{comp obj})$$
$$(\uparrow \textsc{topic}) = (\uparrow \textsc{comp comp obj}) \ \ldots$$

- Lösung: *Functional uncertainty*
  = regulärer Ausdruck über Pfade
  = Disjunktion über alle möglichen Pfadlängen.

$$(\uparrow \textsc{topic}) = (\uparrow \textsc{comp}^* \ \textsc{obj})$$

Fernabhängigkeiten

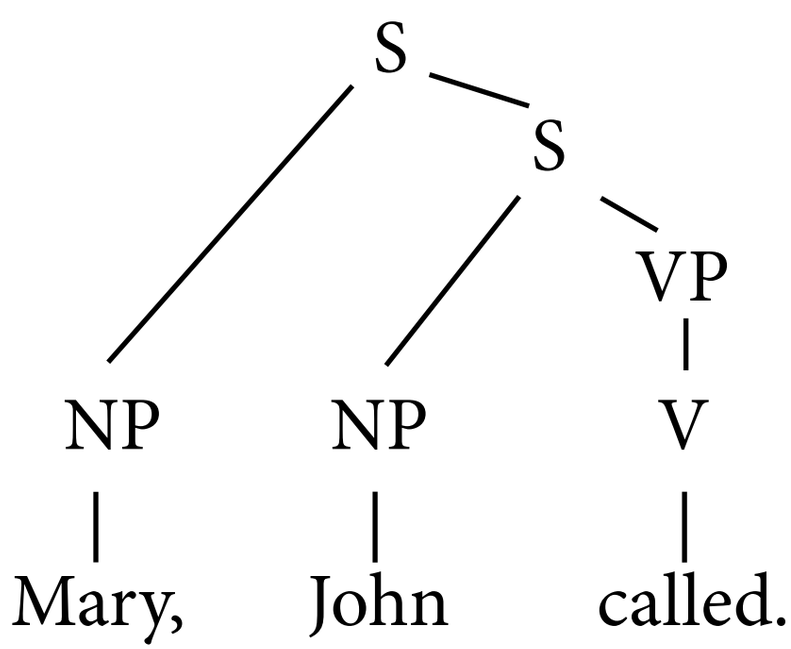
$$\text{S} \rightarrow \underset{\substack{(\uparrow \text{TOPIC}) = \downarrow \\ \boxed{(\uparrow \text{TOPIC}) = (\uparrow \text{COMP}^* \text{ OBJ})}}}{\text{NP}} \quad \underset{\uparrow = \downarrow}{\text{S}}$$

$$\text{S} \rightarrow \underset{(\uparrow \text{SUBJ}) = \downarrow}{\text{NP}} \quad \underset{\uparrow = \downarrow}{\text{VP}}$$

$$\text{VP} \rightarrow \underset{\uparrow = \downarrow}{\text{V}} \left(\underset{(\uparrow \text{OBJ}) = \downarrow)}{\text{NP}} \right) \left(\underset{(\uparrow \text{COMP}) = \downarrow}{\text{S}} \right)$$

called V $(\uparrow \text{PRED}) = \text{‘call’}\langle(\uparrow \text{SUBJ}), (\uparrow \text{OBJ})\rangle$

thinks V $(\uparrow \text{PRED}) = \text{‘think’}\langle(\uparrow \text{SUBJ}), (\uparrow \text{COMP})\rangle$



$$\begin{bmatrix} \text{PRED} & \text{‘call’}\left\langle \left(\uparrow \text{SUBJ}\right), \left(\uparrow \text{OBJ}\right) \right\rangle \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{‘john’} \end{bmatrix} \\ \text{OBJ} & \boxed{1} \\ \text{TOPIC} & \boxed{1} \begin{bmatrix} \text{PRED} & \text{‘mary’} \end{bmatrix} \end{bmatrix}$$

Fernabhängigkeiten

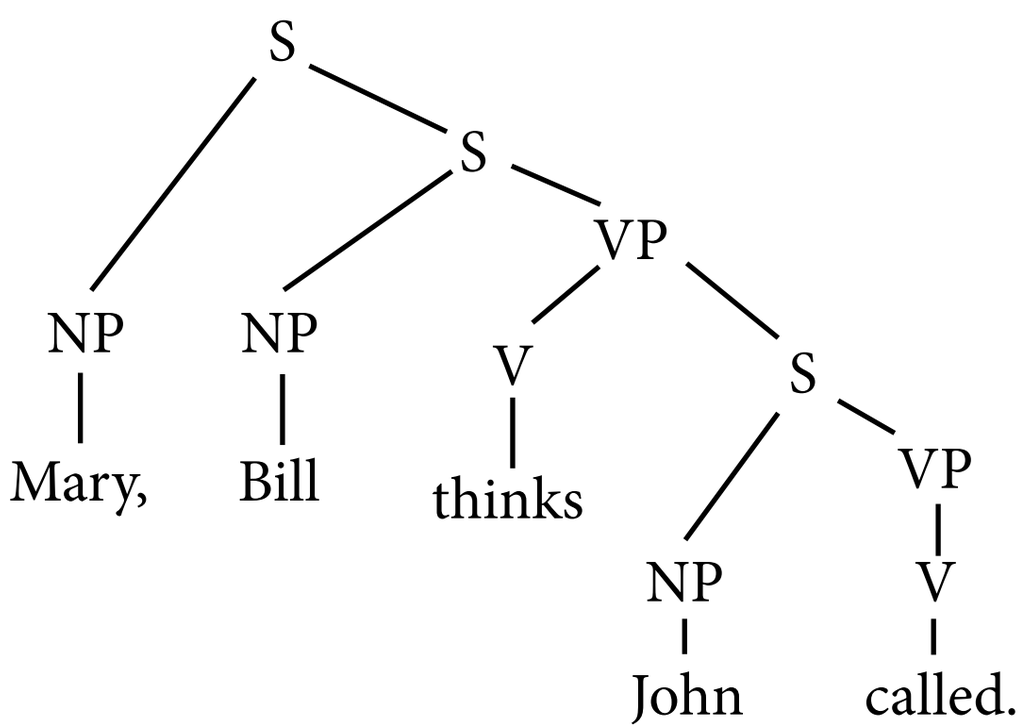
$$\text{S} \rightarrow \underset{\substack{(\uparrow \text{TOPIC}) = \downarrow \\ \boxed{(\uparrow \text{TOPIC}) = (\uparrow \text{COMP}^* \text{ OBJ})}}}{\text{NP}} \quad \underset{\uparrow = \downarrow}{\text{S}}$$

$$\text{S} \rightarrow \underset{(\uparrow \text{SUBJ}) = \downarrow}{\text{NP}} \quad \underset{\uparrow = \downarrow}{\text{VP}}$$

$$\text{VP} \rightarrow \underset{\uparrow = \downarrow}{\text{V}} \left(\underset{(\uparrow \text{OBJ}) = \downarrow)}{\text{NP}} \right) \left(\underset{(\uparrow \text{COMP}) = \downarrow}{\text{S}} \right)$$

called V $(\uparrow \text{PRED}) = \text{‘call’}\langle(\uparrow \text{SUBJ}), (\uparrow \text{OBJ})\rangle$

thinks V $(\uparrow \text{PRED}) = \text{‘think’}\langle(\uparrow \text{SUBJ}), (\uparrow \text{COMP})\rangle$



$$\begin{bmatrix} \text{PRED} & \text{‘think’}\left\langle \left(\uparrow \text{SUBJ}\right), \left(\uparrow \text{COMP}\right) \right\rangle \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{‘bill’} \end{bmatrix} \\ \text{COMP} & \begin{bmatrix} \text{PRED} & \text{‘call’}\left\langle \left(\uparrow \text{SUBJ}\right), \left(\uparrow \text{OBJ}\right) \right\rangle \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{‘john’} \end{bmatrix} \\ \text{OBJ} & \boxed{1} \end{bmatrix} \\ \text{TOPIC} & \boxed{1} \begin{bmatrix} \text{PRED} & \text{‘mary’} \end{bmatrix} \end{bmatrix}$$

Expressivität von LFG

- Ist LFG ein schwach kontextsensitiver Grammatikformalismus?

- Definition schwach kontextsensitiv:
  - enthält kontextfreie Sprachen
  - cross-serial dependencies / Copy-Sprache
  - constant growth
  - polynomielles Parsingproblem

Expressivität von LFG

- Ist LFG ein schwach kontextsensitiver Grammatikformalismus?

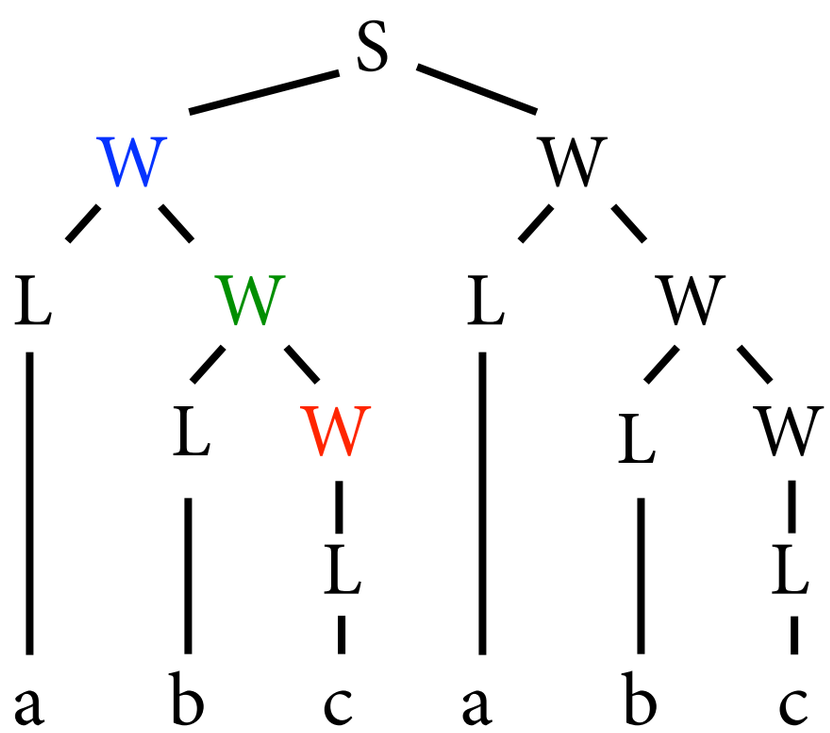
- Definition schwach kontextsensitiv:
  - enthält kontextfreie Sprachen ✔
  - cross-serial dependencies / Copy-Sprache
  - constant growth
  - polynomielles Parsingproblem

Copy-Sprache in LFG

$$\mathrm{W} \rightarrow \underset{\uparrow=\downarrow}{\mathrm{L}} \left(\begin{array}{c} \mathrm{W} \\ (\uparrow \mathrm{W}) = \downarrow \end{array} \right)$$

$$\mathrm{S} \rightarrow \underset{\uparrow=\downarrow}{\mathrm{W}} \quad \underset{\uparrow=\downarrow}{\mathrm{W}}$$

$$\begin{array}{lll} \mathrm{a} & \mathrm{L} & (\uparrow L) = \mathrm{a} \\ \mathrm{b} & \mathrm{L} & (\uparrow L) = \mathrm{b} \\ \mathrm{c} & \mathrm{L} & (\uparrow L) = \mathrm{c} \end{array}$$



$$\begin{bmatrix} \mathrm{L} & \mathrm{a} \\ \mathrm{W} & \begin{bmatrix} \mathrm{L} & \mathrm{b} \\ \mathrm{W} & \begin{bmatrix} \mathrm{L} & \mathrm{c} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

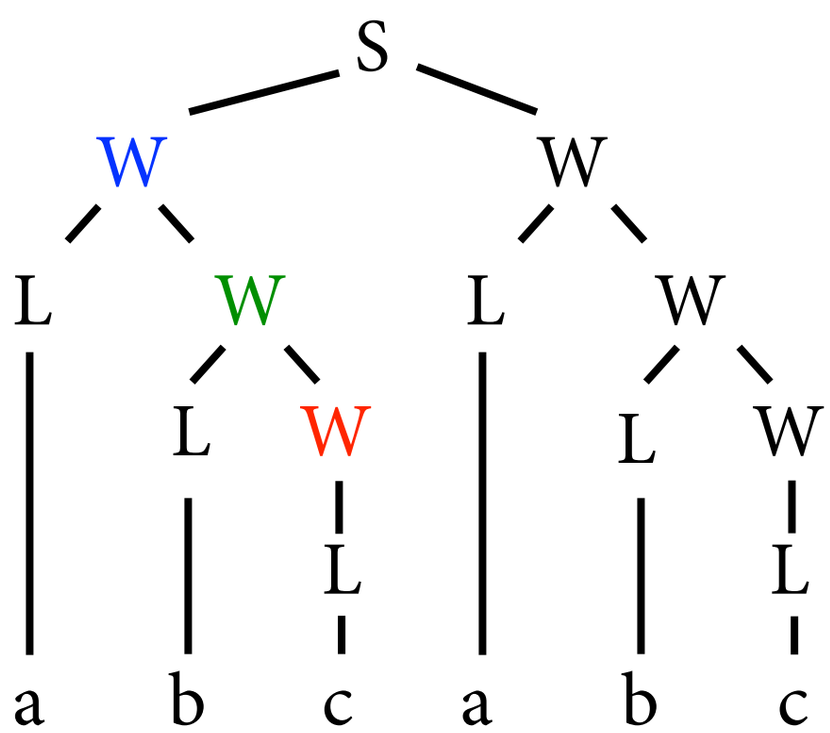
(Kaplan & Bresnan 82)

Copy-Sprache in LFG

$$\mathrm{W} \rightarrow \underset{\uparrow=\downarrow}{\mathrm{L}} \left(\begin{array}{c} \mathrm{W} \\ (\uparrow \mathrm{W}) = \downarrow \end{array} \right)$$

$$\mathrm{S} \rightarrow \underset{\uparrow=\downarrow}{\mathrm{W}} \quad \underset{\uparrow=\downarrow}{\mathrm{W}}$$

$$\begin{array}{lll} \mathrm{a} & \mathrm{L} & (\uparrow L) = \mathrm{a} \\ \mathrm{b} & \mathrm{L} & (\uparrow L) = \mathrm{b} \\ \mathrm{c} & \mathrm{L} & (\uparrow L) = \mathrm{c} \end{array}$$



$$\begin{bmatrix} \mathrm{L} & \mathrm{a} \\ \mathrm{W} & \begin{bmatrix} \mathrm{L} & \mathrm{b} \\ \mathrm{W} & \boxed{\begin{bmatrix} \mathrm{L} & \mathrm{c} \end{bmatrix}} \end{bmatrix} \end{bmatrix}$$

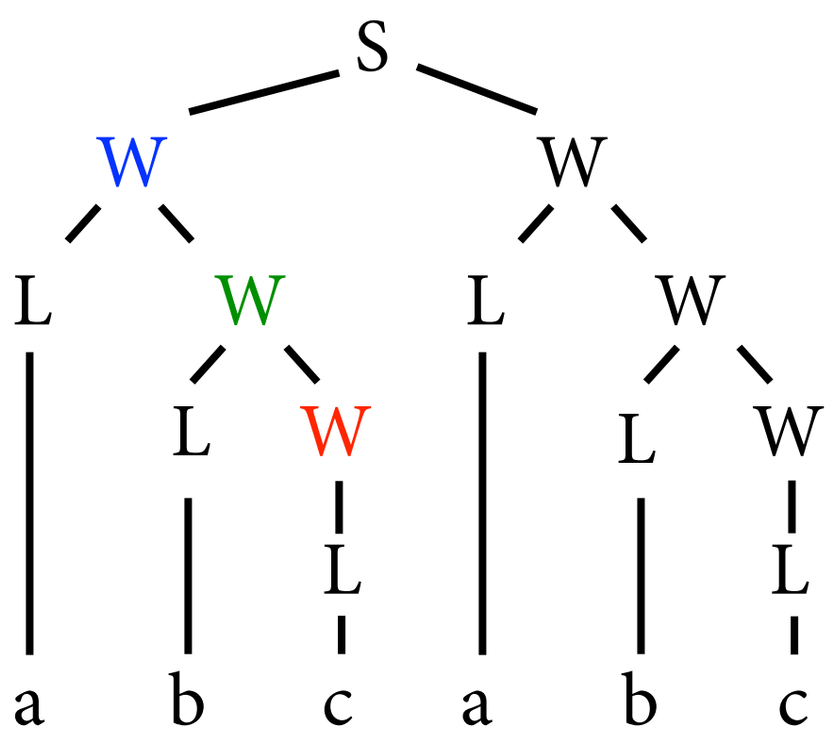
(Kaplan & Bresnan 82)

Copy-Sprache in LFG

$$\mathrm{W} \rightarrow \underset{\uparrow=\downarrow}{\mathrm{L}} \begin{pmatrix} \mathrm{W} \\ (\uparrow \mathrm{W}) = \downarrow \end{pmatrix}$$

$$\mathrm{S} \rightarrow \underset{\uparrow=\downarrow}{\mathrm{W}} \quad \underset{\uparrow=\downarrow}{\mathrm{W}}$$

$$\begin{array}{lll} \mathrm{a} & \mathrm{L} & (\uparrow L) = \mathrm{a} \\ \mathrm{b} & \mathrm{L} & (\uparrow L) = \mathrm{b} \\ \mathrm{c} & \mathrm{L} & (\uparrow L) = \mathrm{c} \end{array}$$



$$\begin{bmatrix} \mathrm{L} & \mathrm{a} \\ \mathrm{W} & \begin{bmatrix} \mathrm{L} & \mathrm{b} \\ \mathrm{W} & \begin{bmatrix} \mathrm{L} & \mathrm{c} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

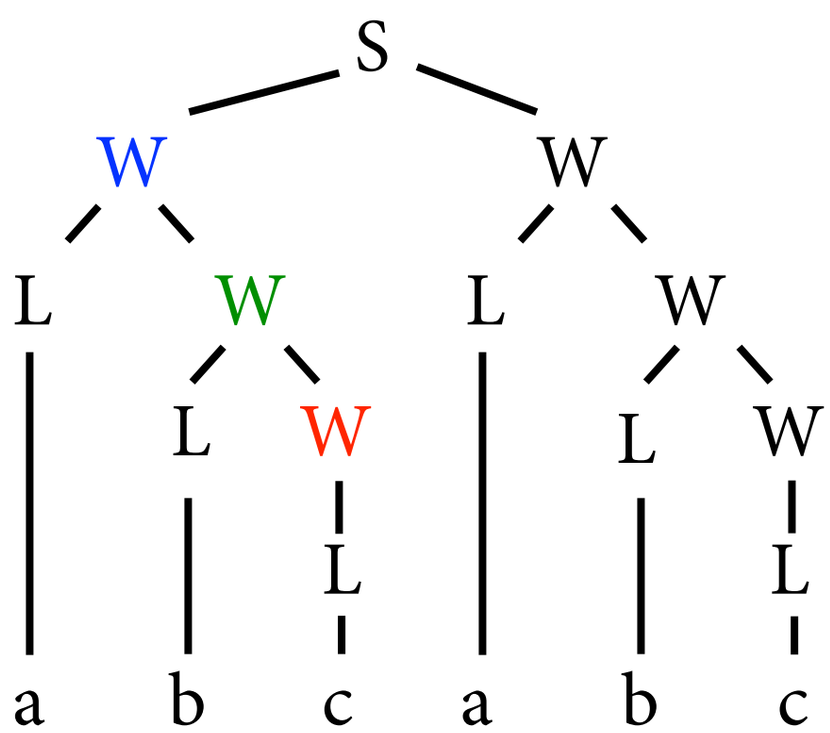
(Kaplan & Bresnan 82)

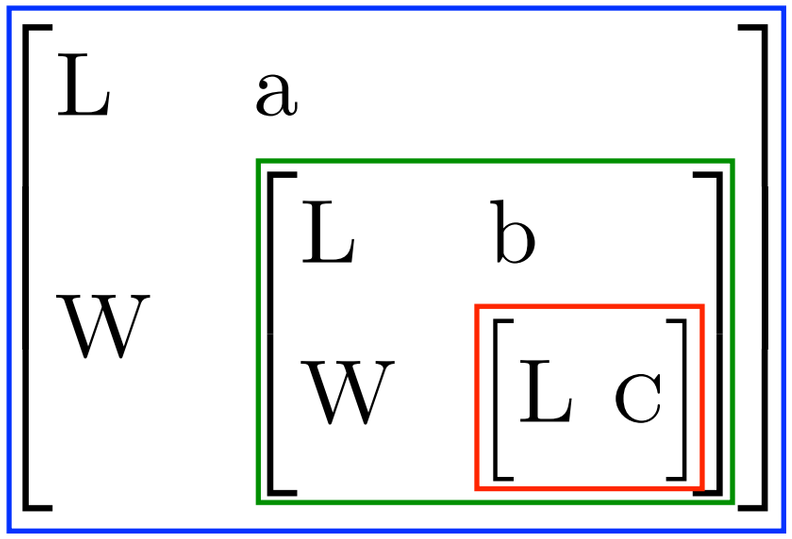
Copy-Sprache in LFG

$$\mathrm{W} \rightarrow \underset{\uparrow=\downarrow}{\mathrm{L}} \left(\begin{array}{c} \mathrm{W} \\ (\uparrow \mathrm{W}) = \downarrow \end{array} \right)$$

$$\mathrm{S} \rightarrow \underset{\uparrow=\downarrow}{\mathrm{W}} \quad \underset{\uparrow=\downarrow}{\mathrm{W}}$$

$$\begin{array}{lll} \mathrm{a} & \mathrm{L} & (\uparrow L) = \mathrm{a} \\ \mathrm{b} & \mathrm{L} & (\uparrow L) = \mathrm{b} \\ \mathrm{c} & \mathrm{L} & (\uparrow L) = \mathrm{c} \end{array}$$





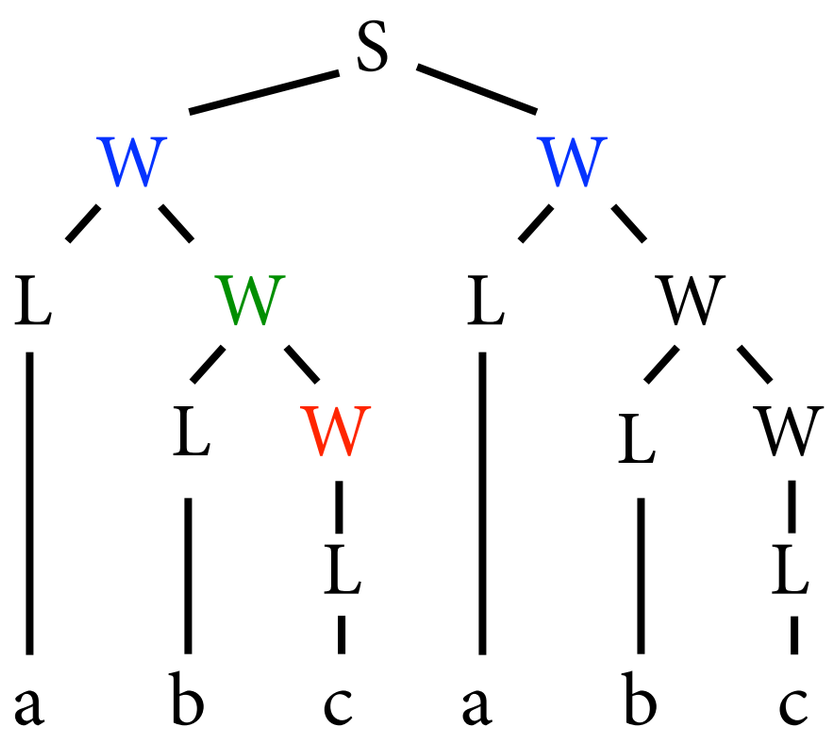
(Kaplan & Bresnan 82)

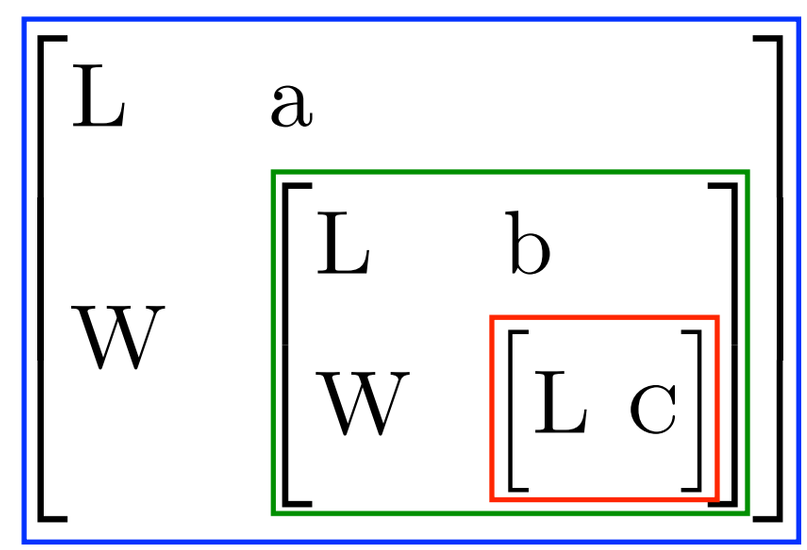
Copy-Sprache in LFG

$$\mathrm{W} \rightarrow \underset{\uparrow=\downarrow}{\mathrm{L}} \begin{pmatrix} \mathrm{W} \\ (\uparrow \mathrm{W}) = \downarrow \end{pmatrix}$$

$$\mathrm{S} \rightarrow \underset{\uparrow=\downarrow}{\mathrm{W}} \quad \underset{\uparrow=\downarrow}{\mathrm{W}}$$

$$\begin{array}{lll} \mathrm{a} & \mathrm{L} & (\uparrow L) = \mathrm{a} \\ \mathrm{b} & \mathrm{L} & (\uparrow L) = \mathrm{b} \\ \mathrm{c} & \mathrm{L} & (\uparrow L) = \mathrm{c} \end{array}$$





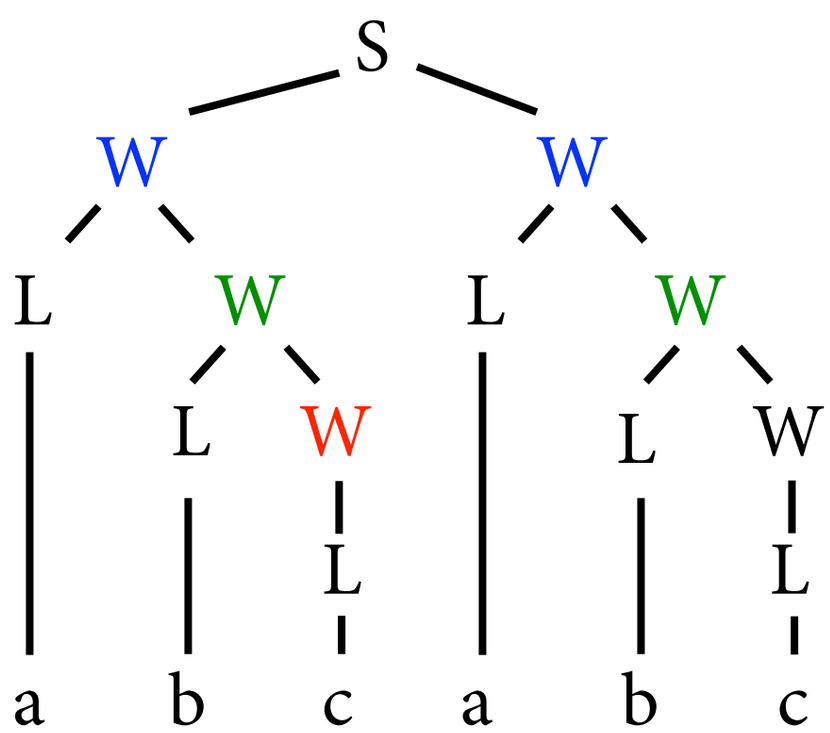
(Kaplan & Bresnan 82)

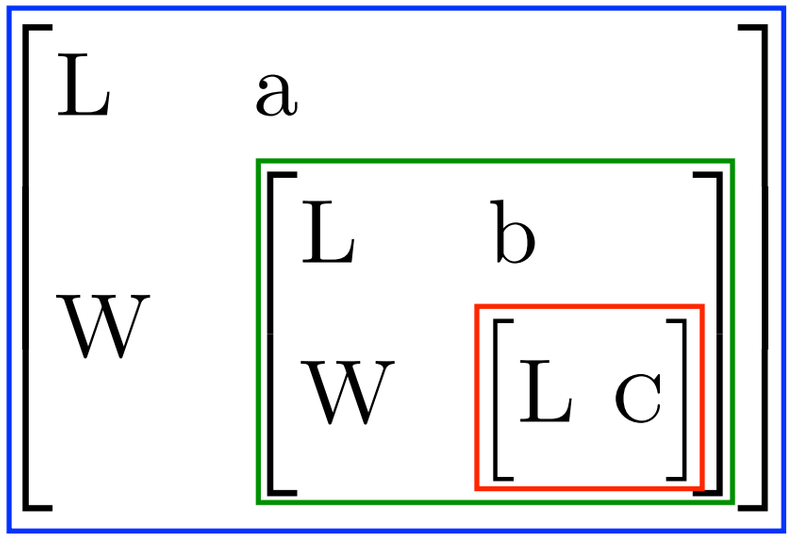
Copy-Sprache in LFG

$$\mathrm{W} \rightarrow \underset{\uparrow=\downarrow}{\mathrm{L}} \begin{pmatrix} \mathrm{W} \\ (\uparrow \mathrm{W}) = \downarrow \end{pmatrix}$$

$$\mathrm{S} \rightarrow \underset{\uparrow=\downarrow}{\mathrm{W}} \quad \underset{\uparrow=\downarrow}{\mathrm{W}}$$

$$\begin{array}{lll} \mathrm{a} & \mathrm{L} & (\uparrow L) = \mathrm{a} \\ \mathrm{b} & \mathrm{L} & (\uparrow L) = \mathrm{b} \\ \mathrm{c} & \mathrm{L} & (\uparrow L) = \mathrm{c} \end{array}$$





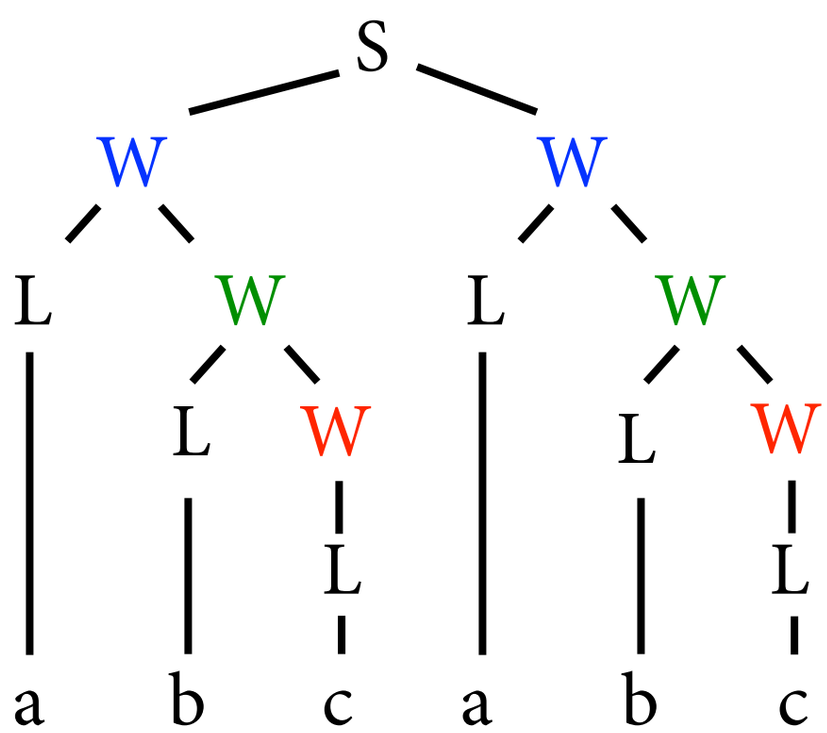
(Kaplan & Bresnan 82)

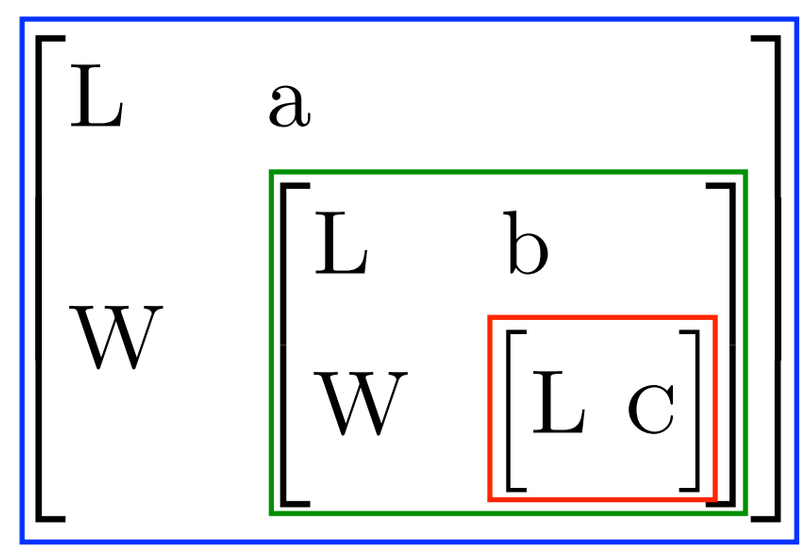
Copy-Sprache in LFG

$$\mathrm{W} \rightarrow \underset{\uparrow=\downarrow}{\mathrm{L}} \left(\begin{array}{c} \mathrm{W} \\ (\uparrow \mathrm{W}) = \downarrow \end{array} \right)$$

$$\mathrm{S} \rightarrow \underset{\uparrow=\downarrow}{\mathrm{W}} \quad \underset{\uparrow=\downarrow}{\mathrm{W}}$$

$$\begin{array}{lll} \mathrm{a} & \mathrm{L} & (\uparrow L) = \mathrm{a} \\ \mathrm{b} & \mathrm{L} & (\uparrow L) = \mathrm{b} \\ \mathrm{c} & \mathrm{L} & (\uparrow L) = \mathrm{c} \end{array}$$





(Kaplan & Bresnan 82)

Expressivität von LFG

- Ist LFG ein schwach kontextsensitiver Grammatikformalismus?

- Definition schwach kontextsensitiv:
  - enthält kontextfreie Sprachen ✔
  - cross-serial dependencies / Copy-Sprache
  - constant growth
  - polynomielles Parsingproblem

Expressivität von LFG

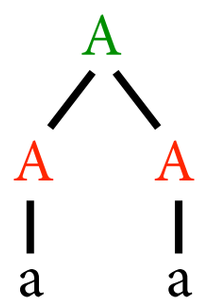
- Ist LFG ein schwach kontextsensitiver Grammatikformalismus?

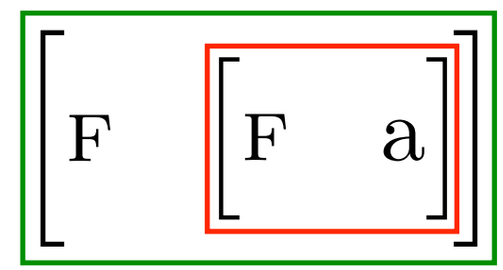
- Definition schwach kontextsensitiv:
  - enthält kontextfreie Sprachen ✔
  - cross-serial dependencies / Copy-Sprache ✔
  - constant growth
  - polynomielles Parsingproblem

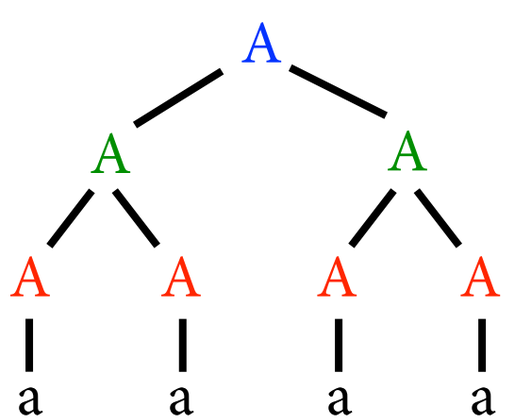
L_2 in LFG

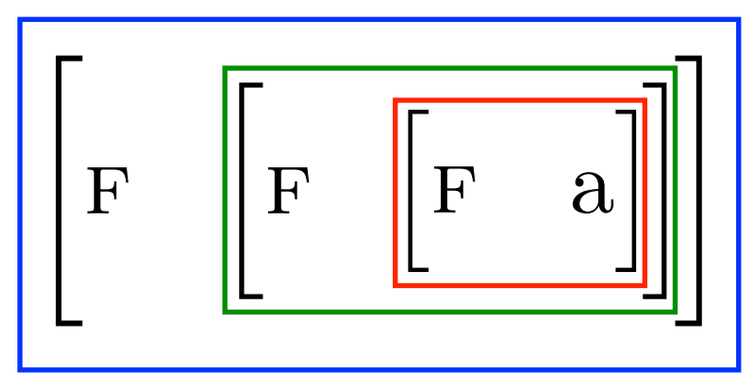
$$L_2 = \{a^{2^n} \mid n \geq 1\}$$

$$\begin{array}{cccccc} \mathrm{A} & \rightarrow & \mathrm{A} & \mathrm{A} & \mathrm{a} & \mathrm{A} \quad (\uparrow f) = \mathrm{a} \\ & & (\uparrow f) = \downarrow & (\uparrow f) = \downarrow & & \end{array}$$









(Berwick 84)

Status von LFG

- LFG ist *nicht* schwach kontextsensitiv:
  hat die constant-growth-Eigenschaft nicht. ✘

- Immerhin sind alle LFG-Sprachen kontextsensitiv.

- Parsingkomplexität?

Parsing von LFG

- Naiver Algorithmus:
  - Normales Chart-Parsing mit c-Struktur-Grammatik.
  - Zähle alle möglichen c-Strukturen auf.
  - Berechne für jede einzelne c-Struktur die möglichen f-Strukturen.
- Schritte 1 und 3 sind relativ klar:
  - Schritt 1 = Chartparsing
  - Schritt 3 = Lösen von Feature-Constraints.
- Wie ist es mit Schritt 2?

Aufzählen von c-Strukturen

- Problem: String kann unendlich viele c-Strukturen haben.
  - von denen jede eine andere f-Struktur hat und deshalb vielleicht die kleinste grammatisch korrekte ist
  - insbesondere mit *Kettenregeln,* d.h. A → B und B → A
    - Kettenregeln können beliebig oft angewendet werden, ohne im Parsing Fortschritt zu machen.

- Lösung: Beim Berechnen von c-Strukturen wiederholte Anwendung von Kettenregeln verbieten.

Aufzählen von c-Strukturen

- Problem: Jetzt terminiert c-Struktur-Parser, aber wir müssen immer noch alle Bäume aufzählen.

- String kann exponentiell viele kf. Parses haben.
  - A → A A: Parsebäume = alle binären Bäume der Länge n
  - Anzahl ist n-te Catalan-Zahl $C_n = \frac{1}{n+1}\binom{2n}{n}$

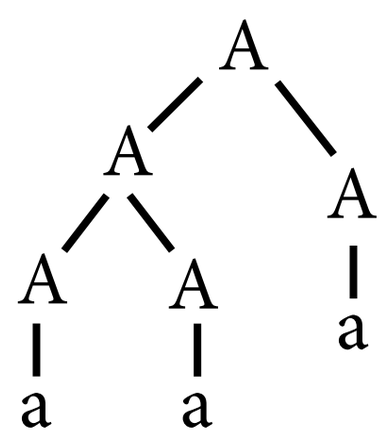
- Grundprinzip: Aufzählen von Parsebäumen ist fast immer eine ganz schlechte Idee.

Unifikation auf der Chart

- f-Struktur-Constraints schon beim Berechnen der c-Struktur-Chart auswerten.
- Manchmal schlägt Unifikation fehl. Dann Item gar nicht erst in Chart eintragen.
- Parsing immer noch worst-case exponentiell, aber Chart ist viel kleiner als Menge der c-Strukturen.

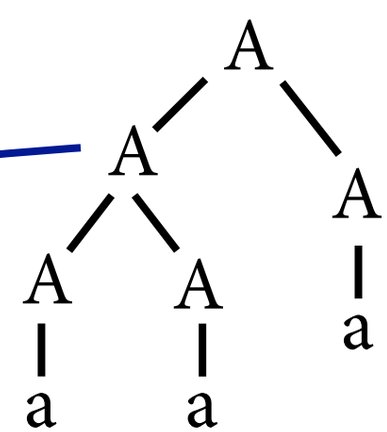
Unifikation auf der Chart

$$\begin{array}{cccccc} \mathrm{A} & \rightarrow & \mathrm{A} & \mathrm{A} & \mathrm{a} & \mathrm{A} \quad (\uparrow f) = \mathrm{a} \\ & & (\uparrow f) = \downarrow & (\uparrow f) = \downarrow & & \end{array}$$



Unifikation auf der Chart

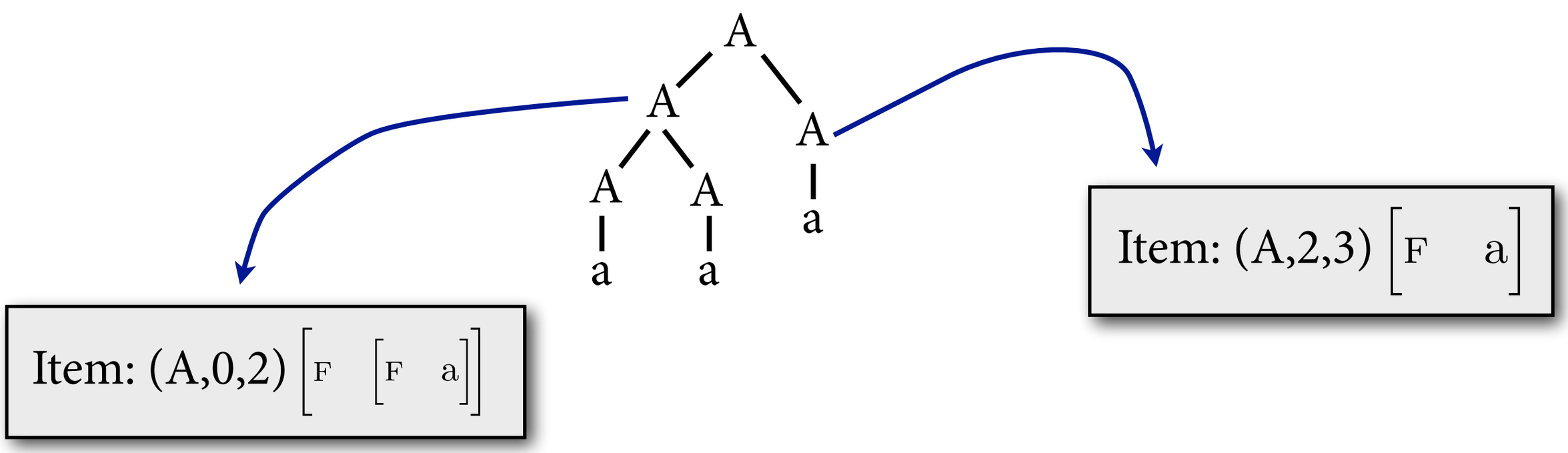
$$\begin{array}{cccccc} \mathrm{A} & \rightarrow & \mathrm{A} & \mathrm{A} & \mathrm{a} & \mathrm{A} \quad (\uparrow f) = \mathrm{a} \\ & & (\uparrow f) = \downarrow & (\uparrow f) = \downarrow & & \end{array}$$



Item: (A,0,2) $\left[\mathrm{F} \quad \left[\mathrm{F} \quad \mathrm{a}\right]\right]$

Unifikation auf der Chart

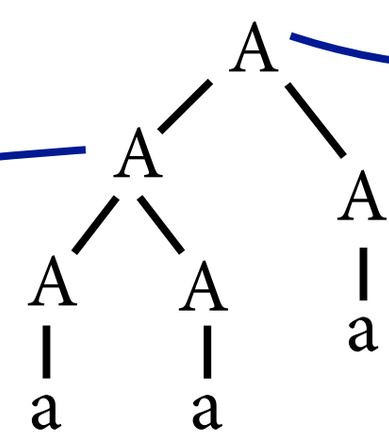
$$\begin{array}{cccccc} \mathrm{A} & \rightarrow & \mathrm{A} & \mathrm{A} & \mathrm{a} & \mathrm{A} \quad (\uparrow f) = \mathrm{a} \\ & & (\uparrow f) = \downarrow & (\uparrow f) = \downarrow & & \end{array}$$



Unifikation auf der Chart

$$\begin{array}{ccccc} \mathrm{A} & \rightarrow & \mathrm{A} & \mathrm{A} & \mathrm{a} \quad \mathrm{A} \quad (\uparrow f) = \mathrm{a} \\ & & (\uparrow f) = \downarrow & (\uparrow f) = \downarrow & \end{array}$$

Unifikation schlägt fehl:
keine f-Struktur für (A,0,3) abgeleitet.



Item: (A,2,3) $\begin{bmatrix} \mathrm{F} & \mathrm{a} \end{bmatrix}$

Item: (A,0,2) $\begin{bmatrix} \mathrm{F} & \begin{bmatrix} \mathrm{F} & \mathrm{a} \end{bmatrix} \end{bmatrix}$

LFG-Parsing

- LFG-spezifische Optimierung: Aus c-Chart einen f-Struktur-Constraint mit *Disjunktionen* berechnen und diesen lösen (Maxwell & Kaplan 95).

- Beste bekannte Algorithmen für LFG-Parsing sind worst-case exponentiell. Optimierungen reduzieren *typische* Laufzeit mit echten Grammatiken.

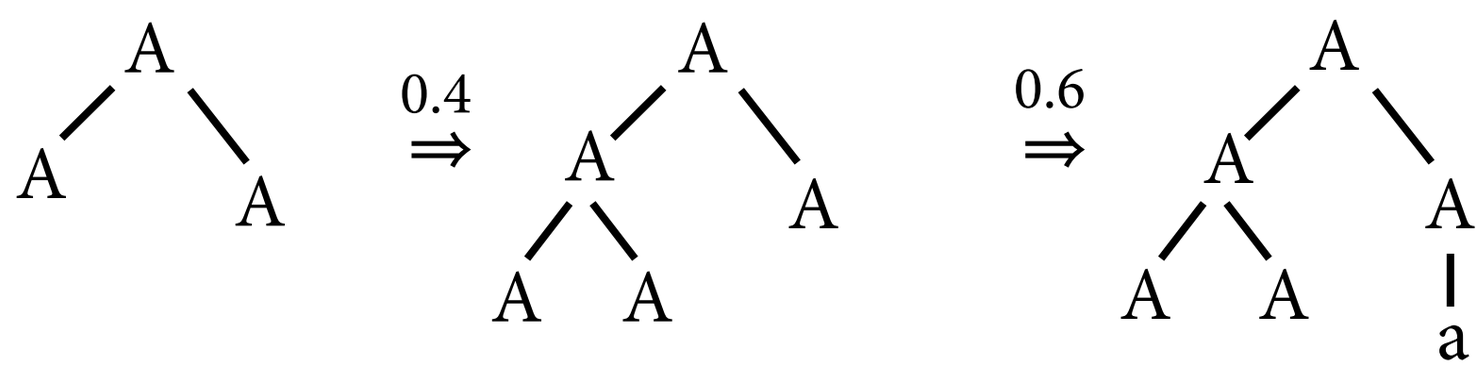
- Praxis: Parser im XLE-System sehr effizient.

Wahrscheinlichkeitsmodell

- LFG-Grammatiken enthalten kfGen. Naheliegender Gedanke: PCFGen auf LFG verallgemeinern.

- PCFG: Expansion verschiedener Knoten sind statistisch unabhängige Ereignisse.

A → A A	[0.4]
A → a	[0.6]

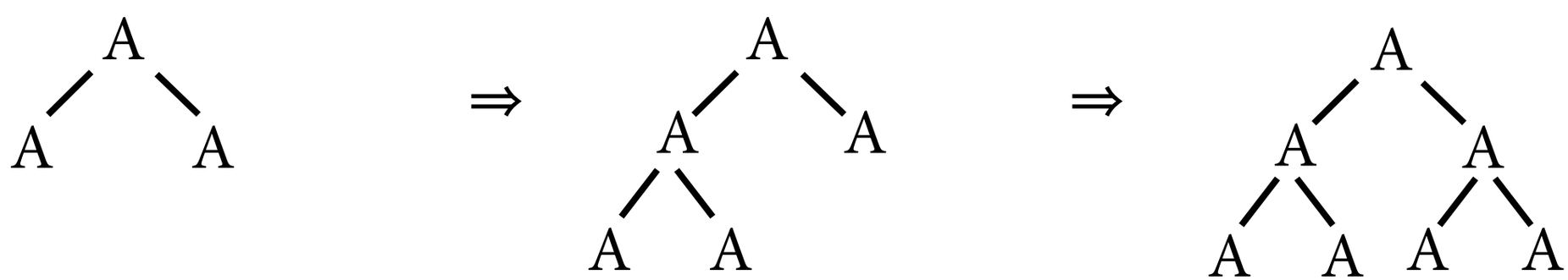


Statistische Abhängigkeiten

- Problem: In LFG schränken f-Constraints ein, wie verschiedene c-Knoten expandiert werden dürfen.

- Expansionen *nicht statistisch unabhängig*.

$$\mathrm{A} \quad \rightarrow \quad \underset{(\uparrow f) = \downarrow}{\mathrm{A}} \quad \underset{(\uparrow f) = \downarrow}{\mathrm{A}} \qquad\qquad \mathrm{a} \quad \mathrm{A} \quad (\uparrow f) = \mathrm{a}$$



Maximum-Entropy-Modell

- Typische Lösung: Verwende ein Maximum-Entropy-Modell für LFG-Ableitungen.

$$P(t|w) = \frac{e^{\theta \cdot f(t,w)}}{\sum_{t'} e^{\theta \cdot f(t',w)}}$$

- Feature-Funktionen $f_i(t,w)$; Gewichte θ_i.

- MaxEnt-Modelle sind robust gegenüber statistischen Abhängigkeiten der Features.

- Maximum Entropy = log-linear (siehe CCG).

(Riezler et al. 02)

MaxEnt für LFG: Features

- Features über c-Struktur
  - wie oft wurde jede c-Regel verwendet?
- Features über f-Struktur
  - wie oft wird jede grammatische Funktion verwendet?
  - von wem wird sie gefüllt?
- Lexikon-Features:
  - wie oft wird Wort x mit Lexikoneintrag L verwendet?
- Und noch viele mehr.

(Riezler et al. 02)

Evaluation

- Penn Treebank hat keine LFG-Annotationen, aber kann PTB in LFG-Baumbank “konvertieren”:
  - Originaltext mit großer LFG-Grammatik parsen
  - nur LFG-Ableitungen behalten, deren c-Struktur mit Annotation übereinstimmt

- F-Score
  (aus Precision & Recall von f-Struktur-Einträgen):
  - lower bound (zufällige Auswahl aus Parses): 75.5
  - upper bound (Parse mit bestem Match): 84.1
  - trainiertes MaxEnt-Modell: 78.6

(Riezler et al. 02)

Zusammenfassung

- Fernabhängigkeiten gehen prima,
  mit Functional Uncertainty.

- Expressivität und Parsing:
  - kontextsensitiv, aber nicht schwach kontextsensitiv
  - worst-case-Komplexität ist exponentiell
  - durch geschickten Umgang mit Unifikationen
    in der Praxis effizient parsbar.

- Standard-W.modell: Maximum Entropy.

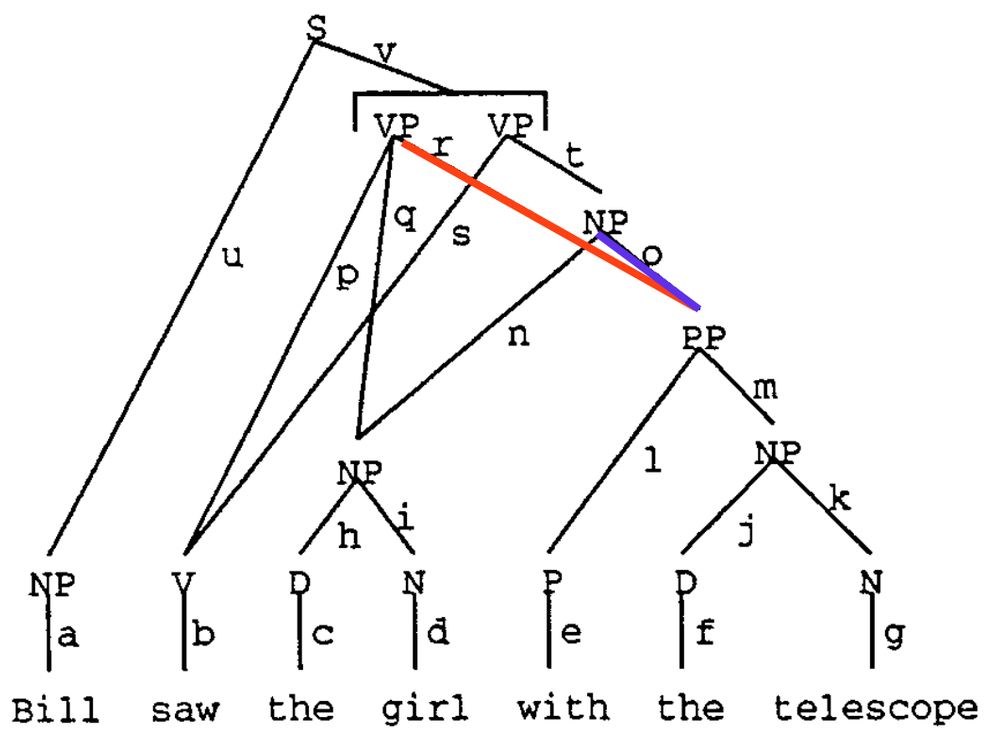
Globale f-Constraints

- Jede Regel in der Chart trägt f-Constraints bei. Wir sammeln alle auf und lösen sie am Schluss alle auf einmal.

- Eintrag (A, i, k) in Chart mit Regeln
  $A \rightarrow r_1, \ldots, A \rightarrow r_n$ expandiert:
  Constraint für (A,i,k) ist *Disjunktion* der Constraints für $r_1, \ldots, r_n$.

(Maxwell & Kaplan 95)

Globale f-Constraints



$$a \wedge u \wedge b \wedge c \wedge h \wedge d \wedge i \wedge e \wedge l \wedge f \wedge j \wedge g \wedge k \wedge m \wedge p \wedge q \wedge (r \vee o) \wedge v$$

f-Struktur 1: $a \wedge u \wedge b \wedge c \wedge h \wedge d \wedge i \wedge e \wedge l \wedge f \wedge j \wedge g \wedge k \wedge m \wedge p \wedge q \wedge r \wedge v$

f-Struktur 2: $a \wedge u \wedge b \wedge c \wedge h \wedge d \wedge i \wedge e \wedge l \wedge f \wedge j \wedge g \wedge k \wedge m \wedge p \wedge q \wedge o \wedge v$