

# Computerlinguistik II / Sprachtechnologie

Vorlesung im SS 2010  
(M-GSW-10)

Prof. Dr. Udo Hahn

Lehrstuhl für Computerlinguistik  
Institut für Germanistische Sprachwissenschaft  
Friedrich-Schiller-Universität Jena

## Grundbegriffe zur Syntaxanalyse von CFGs

- Das **Wort-** bzw. **Erkennungsproblem** für eine kontextfreie Grammatik  $G$  :

Zeige für  $G = ( N, T, P, S )$  und  $\omega \in T^*$ ,  
dass  $\omega$  von  $G$  (nicht) erzeugt werden  
kann (d.h.:  $\omega \in \mathcal{L}(G)$  oder  $\omega \notin \mathcal{L}(G)$  ).

Ein Algorithmus, der dieses Problem löst,  
heißt **Erkennungsalgorithmus** (oder  
**Recognizer**).

2

## Grundbegriffe zur Syntaxanalyse von CFGs

- Das **Analyseproblem** für eine kontextfreie Grammatik  $G$  :

Bestimme für  $G = ( N, T, P, S )$  und  $\omega \in T^*$   
entweder eine syntaktische Struktur von  $\omega$   
bezüglich  $G$  oder zeige, dass  $\omega \notin \mathcal{L}(G)$  .

Ein Algorithmus, der dieses Problem löst,  
heißt **Analysealgorithmus** (oder **Parser**).

Die Bestimmung der syntaktischen Struktur  
heißt **Syntaxanalyse** bzw. **Parsing**.

3

## Bemerkungen zur Syntaxanalyse von CFGs

- Ein Analysealgorithmus löst mit der (fehlschlagenden) Bestimmung einer syntaktischen Struktur stets auch das Wortproblem.
- Für Typ-0-Grammatiken ist das Wortproblem unlösbar.
- Für Typ-1-Grammatiken, die bestimmten Beschränkungen unterliegen, und generell für Typ-2-Grammatiken ist das Wortproblem lösbar – wenn auch (für Typ-1) mit z.T. beträchtlicher, aber noch polynomialer Berechnungskomplexität.
- Für Typ-3-Grammatiken ist das Wort- und Analyseproblem einfach lösbar (linear).

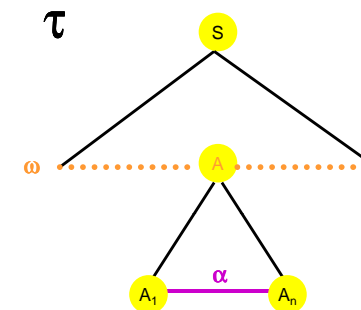
4

## Grundbegriffe zur Syntaxanalyse von CFGs

- Sei  $\omega$  ein von der kontextfreien Grammatik  $G$  erzeugtes Wort und  $\tau$  ein zugehöriger Strukturbaum, der eine feste (beliebig wählbare, aber dann gegebene) Verzweigung besitzt, die aus einem Knoten und seinen direkten Nachfolgern besteht. Diese Verzweigung beschreibt die Anwendung einer Produktion, etwa  $A \rightarrow \alpha$  mit  $\alpha = A_1 \dots A_n$  und  $A_i \in V$  für  $1 \leq i \leq n$ .

5

## Gliederung eines Strukturbaums



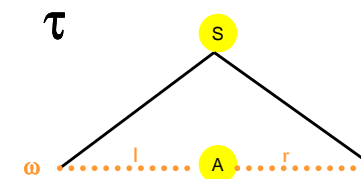
6

## Grundbegriffe zur Syntaxanalyse von CFGs

- Durch die Fixierung einer festen Verzweigung und der zugehörigen Anwendung einer Produktion wird  $\tau$  in **Teilstrukturen** zerlegt. Dazu betrachten wir die Klasse  $\tau_A$  aller Strukturbäume zu  $G$ , die Anfangsteilbäume von  $\tau$  sind und den fest herausgegriffenen Knoten  $A$  als Endknoten haben. Diese haben Endschnittbilder der Form  $lAr$  mit  $l, r \in V^*$  und beschreiben die Ableitung:  $S^* \Rightarrow lAr$ .

7

## Gliederung eines Strukturbaums



8

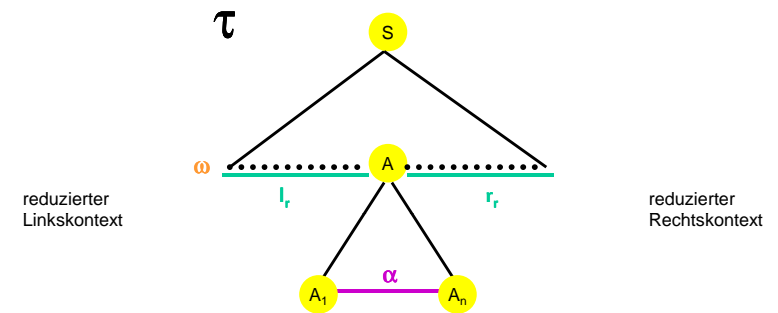
## Grundbegriffe zur Syntaxanalyse von CFGs

- Sei  $\tau_{\min}$  der eindeutig fixierte Strukturbaum in  $\tau_A$  mit minimaler Knotenzahl und  $l_r A r_r$  sein Endschnittbild.

$l_r$  ist der **reduzierte Linkskontext** und  $r_r$  der **reduzierte Rechtskontext** zur betrachteten Anwendung der Produktion  $A \rightarrow \alpha$ .

9

## Gliederung eines Strukturbaums



10

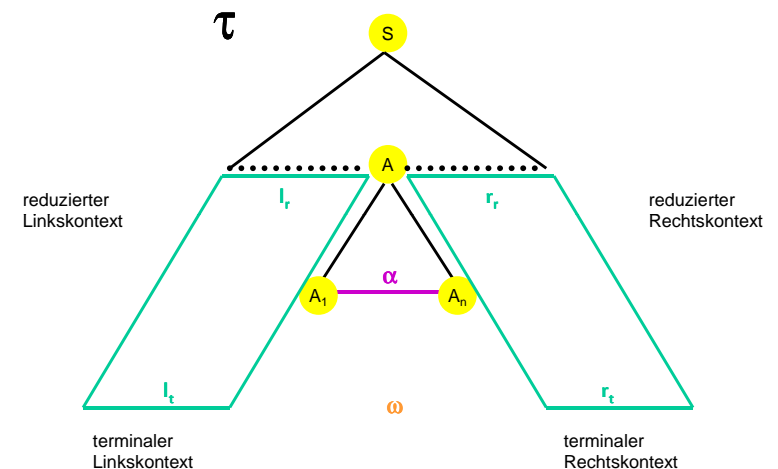
## Grundbegriffe zur Syntaxanalyse von CFGs

- Sei  $\tau_{\max}$  der eindeutig fixierte Strukturbaum in  $\tau_A$  mit maximaler Knotenzahl und  $l_t A r_t$  sein Endschnittbild. Dann sind  $l_t, r_t \in T^*$ .

$l_t$  heißt dann **terminaler Linkskontext** und  $r_t$  **terminaler Rechtskontext** zur betrachteten Anwendung der Produktion  $A \rightarrow \alpha$ .

11

## Gliederung eines Strukturbaums



12

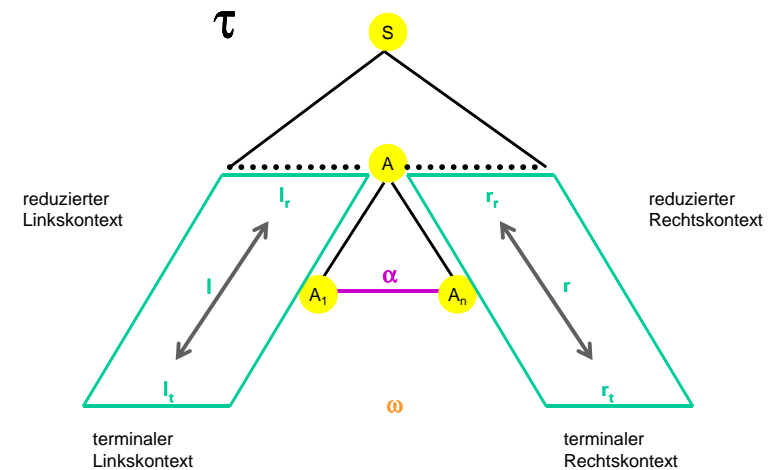
## Grundbegriffe zur Syntaxanalyse von CFGs

- Sei  $\tau_{bel}$  ein beliebig herausgegriffener Strukturbaum in  $\tau_A$  und sei  $l A r$  sein Endschnittbild. Dann gilt:

$$l_r^* \Rightarrow l^* \Rightarrow l_t \text{ und } r_r^* \Rightarrow r^* \Rightarrow r_t.$$

13

## Gliederung eines Strukturbaums



14

## Grundbegriffe zur Syntaxanalyse von CFGs

- Die fest herausgegriffene Anwendung der Produktion  $A \rightarrow \alpha$  bestimmt in der betrachteten syntaktischen Struktur von  $\omega$  somit vier Teilstrukturen, die Ableitungen für

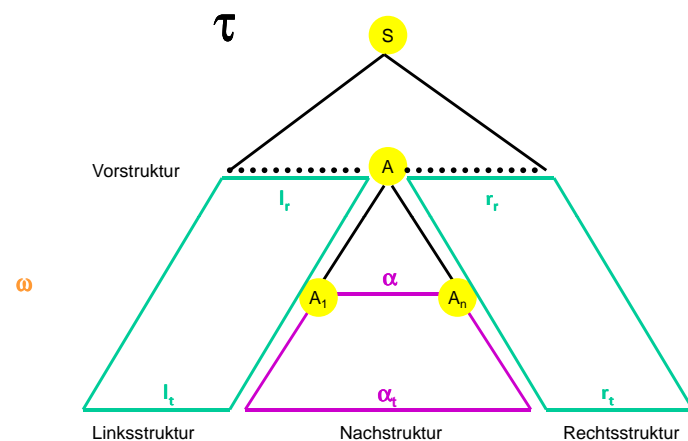
$$S^* \Rightarrow l_r A r_r, l_r^* \Rightarrow l_t, \alpha^* \Rightarrow \alpha_t \text{ und } r_r^* \Rightarrow r_t$$

mit  $\omega = l_t \alpha_t r_t$  entsprechen.

- Man nennt diese Teilstrukturen die zur herausgegriffenen Anwendung der Produktion  $A \rightarrow \alpha$  gehörige **Vorstruktur**, **Linksstruktur**, **Nachstruktur** und **Rechtsstruktur**.

15

## Gliederung eines Strukturbaums



16

## Grundbegriffe zur Syntexanalyse von CFGs

Die im Folgenden betrachteten **Analysestrategien** sind dadurch charakterisiert, dass bestimmte, zu einer festen Anwendung einer Produktion gehörige Teilstrukturen für das Erkennen der Anwendung dieser Produktion jeweils bekannt sein müssen.

- Erfolgt das Erkennen der Anwendung einer Produktion, sobald die zugehörige Linksstruktur total bekannt ist, spricht man von einer **Analyse von links nach rechts** (analog: **Analyse von rechts nach links**).

17

## Grundbegriffe zur Syntexanalyse von CFGs

In beiden Fällen ist die Reihenfolge des Auffindens der zugehörigen Vor- und Nachstruktur noch nicht festgelegt.

- Ist jeweils die Vorstruktur total bekannt ist, während die Nachstruktur noch unbekannt ist, spricht man von einer **abwärts gerichteten** (oder **Top-Down**-) **Analyse**. Im umgekehrten Fall – bei einer total erkannten Nachstruktur und noch unbekannter Vorstruktur – von einer **aufwärts gerichteten** (oder **Bottom-Up**-) **Analyse**.

18

## Grundbegriffe zur Syntexanalyse von CFGs

- Bei einer **Top-Down-Analyse** sind beim Erkennen der Anwendung einer Produktion  $A \rightarrow \alpha$  die zugehörige Vorstruktur und Linksstruktur (und damit insbesondere auch der zugehörige terminale Linkskontext  $l_t$  und der zugehörige reduzierte Rechtskontext  $r_r$ ) bekannt.

19

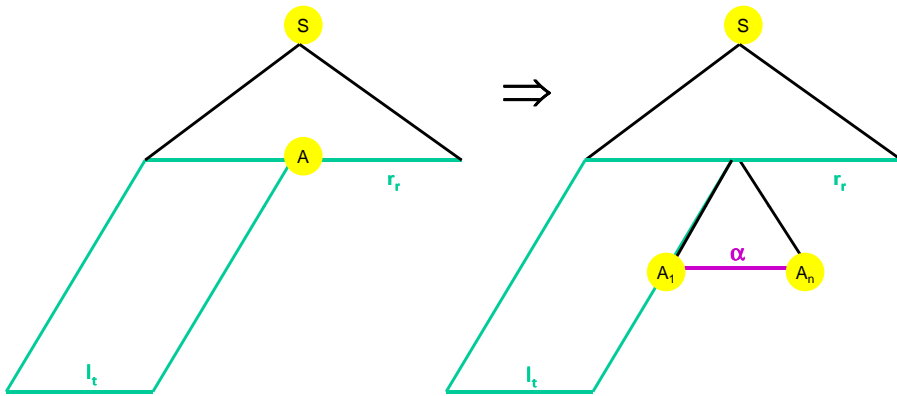
## Grundbegriffe zur Syntexanalyse von CFGs

- Dem Erkennen der Anwendung der Produktion  $A \rightarrow \alpha$  entspricht der Übergang von der Satzform  $s = l_t A r_r$  zur Satzform  $s' = l_t \alpha r_r$ , wobei  $s$  und  $s'$  die Endschnittbilder der jeweils erkannten Teile der Struktur sind. Dieser Übergang von  $s$  nach  $s'$  entspricht dem Ableitungsschritt  $[l_t, A \rightarrow \alpha, r_r]$ . Dabei liegt  $l_t$  nach Vor-aussetzung immer in  $T^*$ . Somit ergeben die bei der **Top-Down-Analyse** nacheinander gefundenen Ableitungsschritte eine **Linksableitung**, also:

$$\Delta = \{ \delta_i \}_{i=1}^n \text{ mit } \delta_i = [l_i, A_i \rightarrow \gamma_i, r_i], l_i \in T^* \text{ für } 1 \leq i \leq n.$$

20

## Links-Rechts-Top-Down-Analyse



21

## Grundbegriffe zur Syntaxanalyse von CFGs

- Bei einer **Bottom-Up-Analyse** sind beim Erkennen der Anwendung einer Produktion  $A \rightarrow \alpha$  die zugehörige Nachstruktur und Linksstruktur (und damit insbesondere auch der zugehörige reduzierte Linkskontext  $l_r$ , die rechte Seite der Produktion  $A \rightarrow \alpha$  und der zugehörige terminale Rechtskontext  $r_t$ ) bekannt.

22

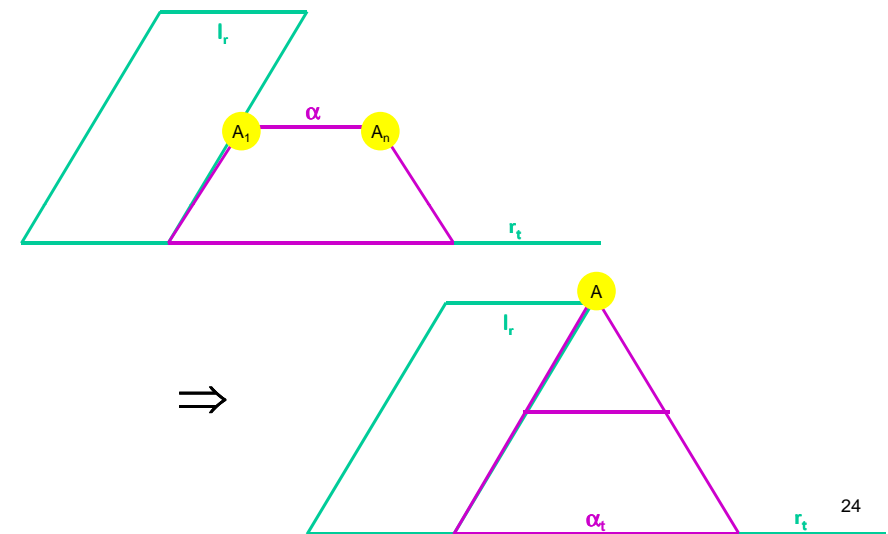
## Grundbegriffe zur Syntaxanalyse von CFGs

- Dem Erkennen der Anwendung der Produktion  $A \rightarrow \alpha$  entspricht die Reduktion der Satzform  $s = l_r \alpha r_t$  auf die Satzform  $s' = l_r A r_t$ , wobei  $s$  und  $s'$  als Endschnittbilder der jeweils noch unbekannt Teile zugleich Anfangsschnittbilder der jeweils erkannten Teile der Struktur sind. Dieser Übergang von  $s$  nach  $s'$  entspricht dem Reduktionsschritt  $[l_r, A \rightarrow \alpha, r_t]$ . Dabei liegt  $r_t$  nach Voraussetzung immer in  $T^*$ .

Somit ergeben die bei der **Bottom-Up-Analyse** nacheinander gefundenen Ableitungsschritte eine **Rechtsreduktion**, also:

$$\Delta = \{ \delta_i \}_{i=1}^n \text{ mit } \delta_i = [l_i, A_i \rightarrow \gamma_i, r_i], r_i \in T^* \text{ für } 1 \leq i \leq n. \quad 23$$

## Links-Rechts-BottomUp-Analyse



24

# Grundbegriffe zur Syntaxanalyse von CFGs

- Sind die gerade betrachteten Übergänge der Satzform  $s = l_t \mathbf{A} r_r$  auf  $s' = l_t \alpha r_r$  bei der Top-Down-Analyse bzw.  $s = l_r \alpha r_t$  auf  $s' = l_r \mathbf{A} r_t$  bei der Bottom-Up-Analyse für  $l_t, r_t \in T^*$  nicht erfüllt (d.h., es liegt keine Linksableitung bzw. Rechtsreduktion vor), ist die Ableitung bzw. Reduktion in Bezug auf ihre Analysestrategie nicht mehr eindeutig determiniert (d.h., es existieren mehrere Kontrollwörter / Parses für eine Ableitung);  
sie heißt dann **nicht-deterministisch**.