Computerlinguistik I

Vorlesung im WiSe 2017/2018
(M-GSW-09)

Prof. Dr. Udo Hahn

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

http://www.julielab.de

Grundbegriffe zur Syntaxanalyse von CFGs

- Das Wort- bzw. Erkennungsproblem für eine kontextfreie Grammatik $\mathcal{G}$:

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

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

Grundbegriffe zur Syntaxanalyse von CFGs

- Das Analyseproblem für eine kontextfreie Grammatik $\mathcal{G}$:

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

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

  Die Bestimmung der syntaktischen Struktur heißt Syntaxanalyse bzw. Parsing.

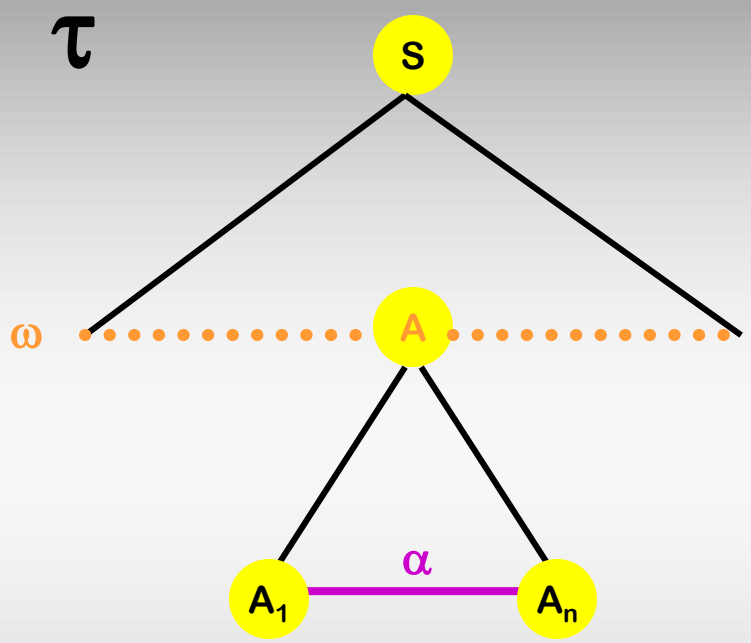
Bemerkungen zur Syntaxanalyse von CFGs

- 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).

Grundbegriffe zur Syntaxanalyse von CFGs

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

Gliederung eines Strukturbaums

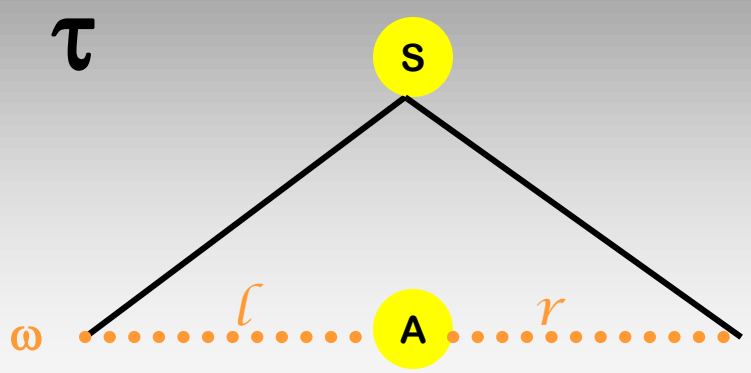


Grundbegriffe zur Syntaxanalyse von CFGs

- Durch die Fixierung einer festen Verzweigung und der zugehörigen Anwendung einer Produktion wird τ in Teilstrukturen zerlegt.

  Dazu betrachten wir die Klasse τ_A aller Strukturbäume zu $\mathcal{G}$, die Anfangsteilbäume von τ sind (d.h. die gleiche Wurzel besitzen) und den fest herausgegriffenen Knoten A als Endknoten haben. Diese haben Endschnittbilder der Form $\ell \mathrm{A} r$ mit $\ell, r \in \mathcal{V}^*$ und beschreiben die Ableitung: $S \overset{*}{\Rightarrow} \ell \mathrm{A} r$.

Gliederung eines Strukturbaums

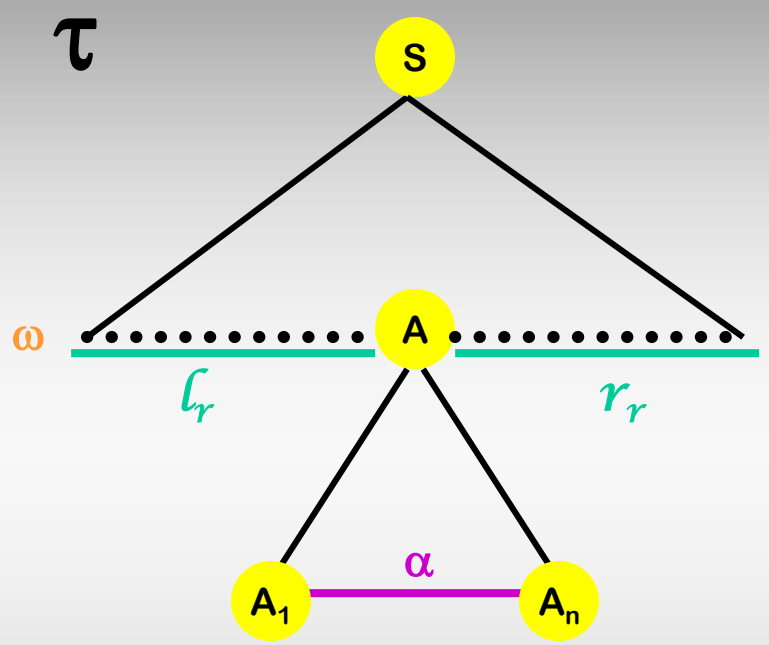


Grundbegriffe zur Syntaxanalyse von CFGs

- Sei τ_{min} der eindeutig fixierte Strukturbaum in τ_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 $A \rightarrow \alpha$.

Gliederung eines Strukturbaums



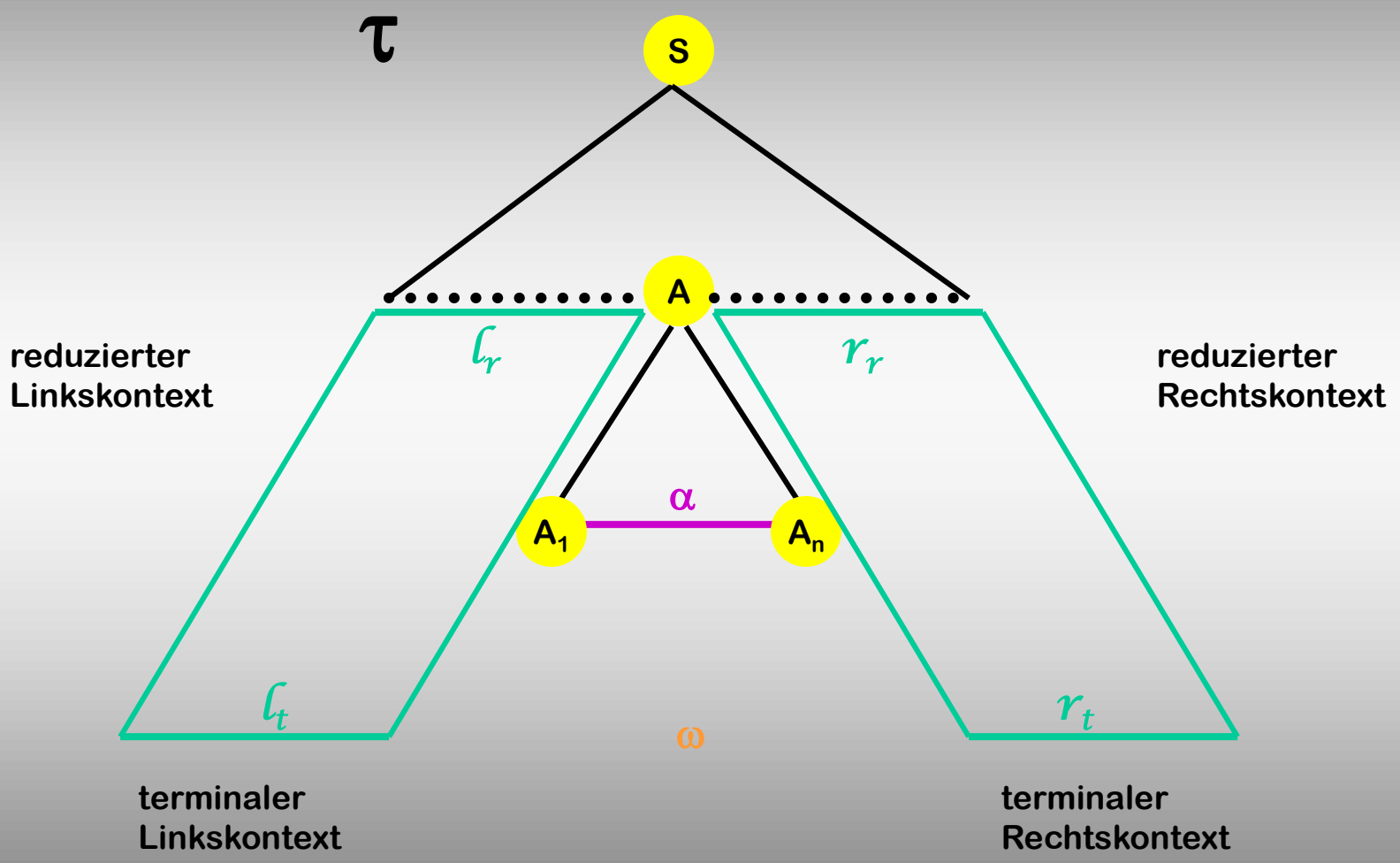
reduzierter Linkskontext

reduzierter Rechtskontext

Grundbegriffe zur Syntaxanalyse von CFGs

- Sei τ_{max} der eindeutig fixierte Strukturbaum in τ_A mit maximaler Knotenzahl und $\ell_t \, A \, r_t$ sein Endschnittbild. Dann sind $\ell_t , r_t \in T^*$.
  ℓ_t heißt dann terminaler Linkskontext und r_t terminaler Rechtskontext zur betrachteten Anwendung der Produktion $A \rightarrow \alpha$.

Gliederung eines Strukturbaums

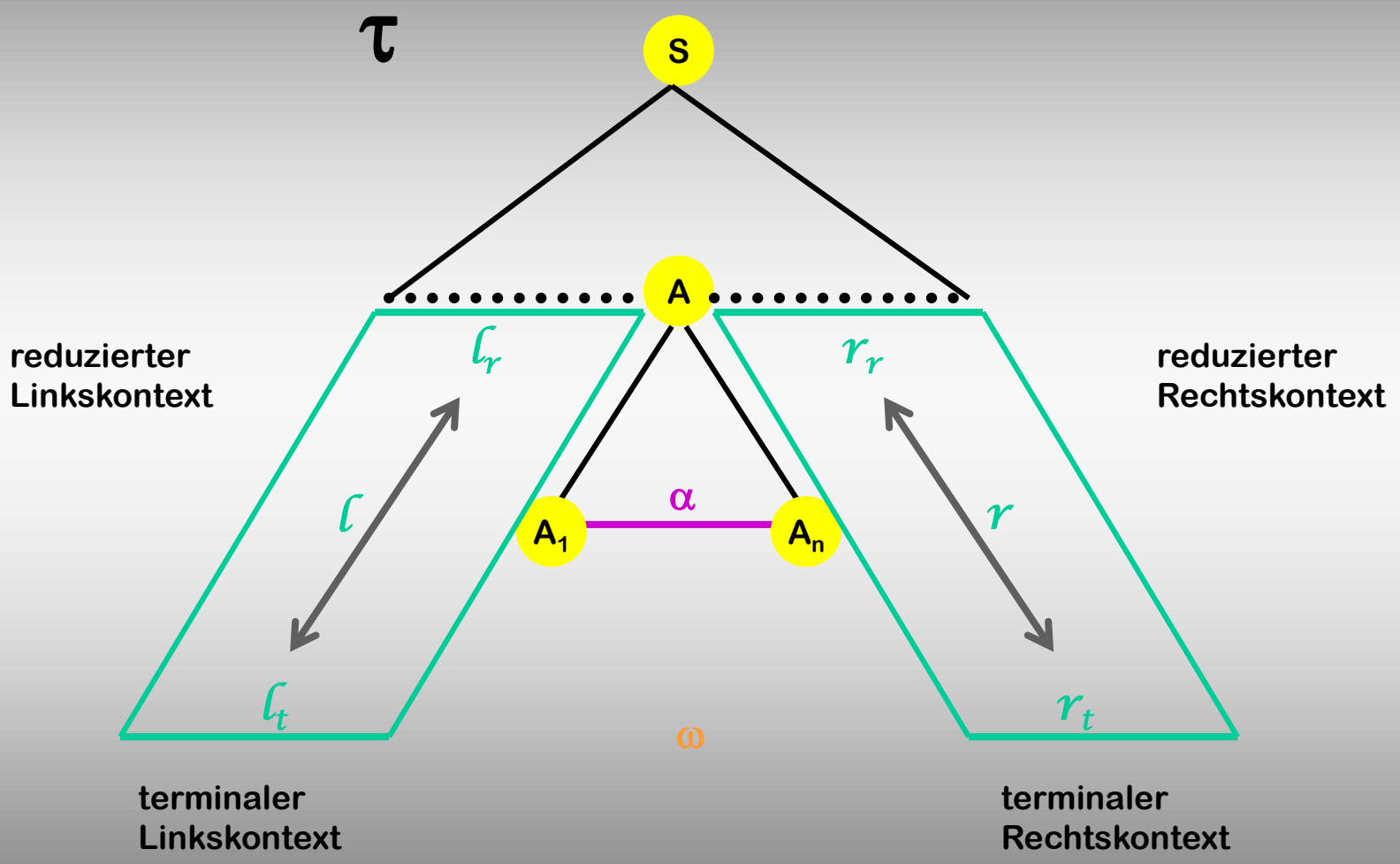


Grundbegriffe zur Syntaxanalyse von CFGs

- Sei τ_{bel} ein beliebig herausgegriffener Strukturbaum in τ_A und sei $\ell \mathrm{A}\, r$ sein Endschnittbild. Dann gilt:

$$\ell_r \overset{*}{\Rightarrow} \ell \overset{*}{\Rightarrow} \ell_t \text{ und } r_r \overset{*}{\Rightarrow} r \overset{*}{\Rightarrow} r_t .$$

Gliederung eines Strukturbaums



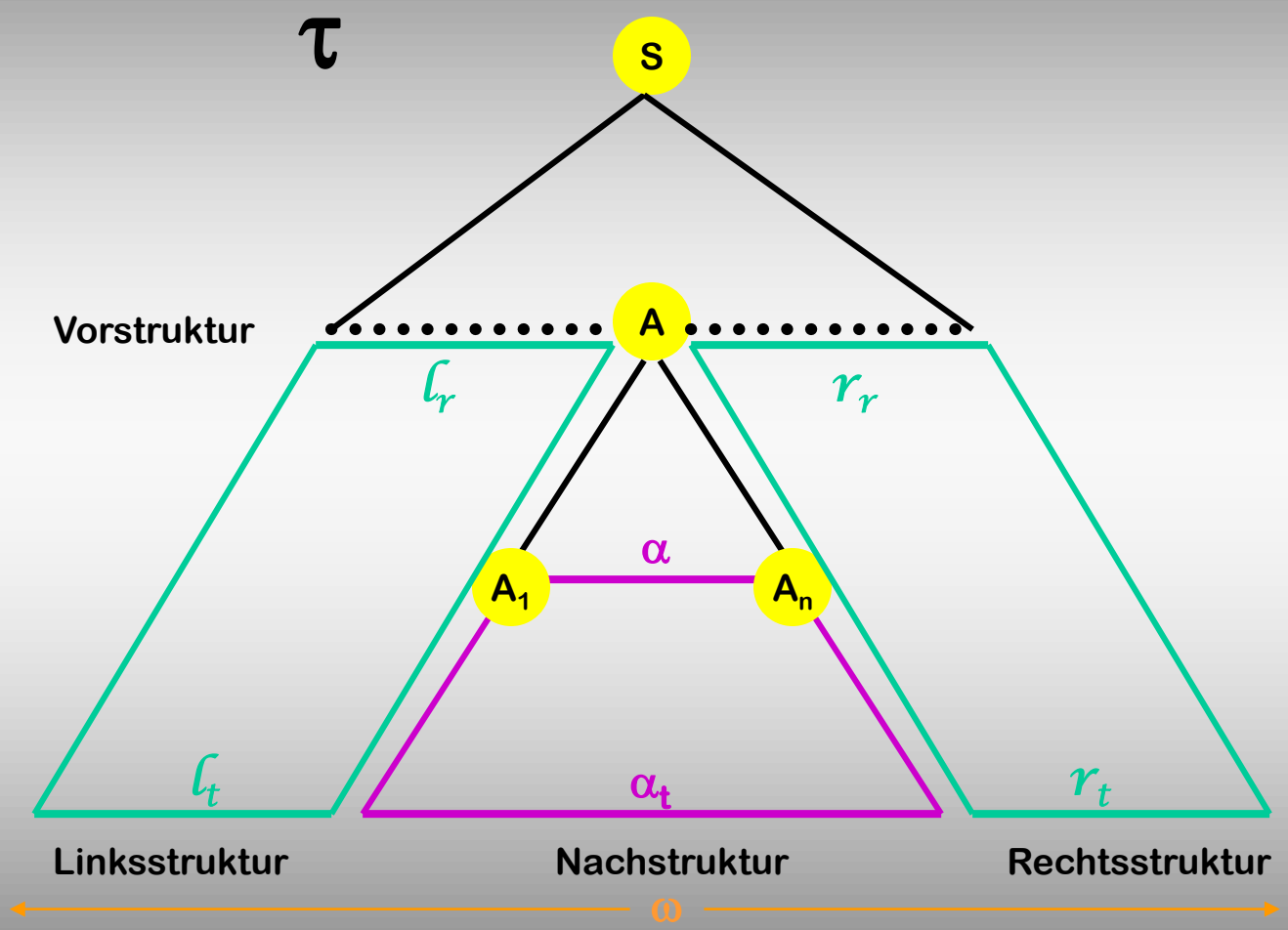
Grundbegriffe zur Syntaxanalyse von CFGs

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

  $S \overset{*}{\Rightarrow} \ell_r A r_r$, $\ell_r \overset{*}{\Rightarrow} \ell_t$, $\alpha \overset{*}{\Rightarrow} \alpha_t$ und $r_r \overset{*}{\Rightarrow} r_t$

  mit $\omega = \ell_t \alpha_t r_t$ entsprechen.
- Man nennt diese Teilstrukturen die zur herausgegriffenen Anwendung der Produktion $A \rightarrow \alpha$ gehörige Vorstruktur, Linksstruktur, Nachstruktur und Rechtsstruktur.

Gliederung eines Strukturbaums



Grundbegriffe zur Syntaxanalyse von CFGs

Die im Folgenden betrachteten Analyse-strategien 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 Links-struktur total bekannt ist, spricht man von einer Analyse von links nach rechts (analog: Analyse von rechts nach links).

Grundbegriffe zur Syntaxanalyse 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.

Grundbegriffe zur Syntaxanalyse 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.

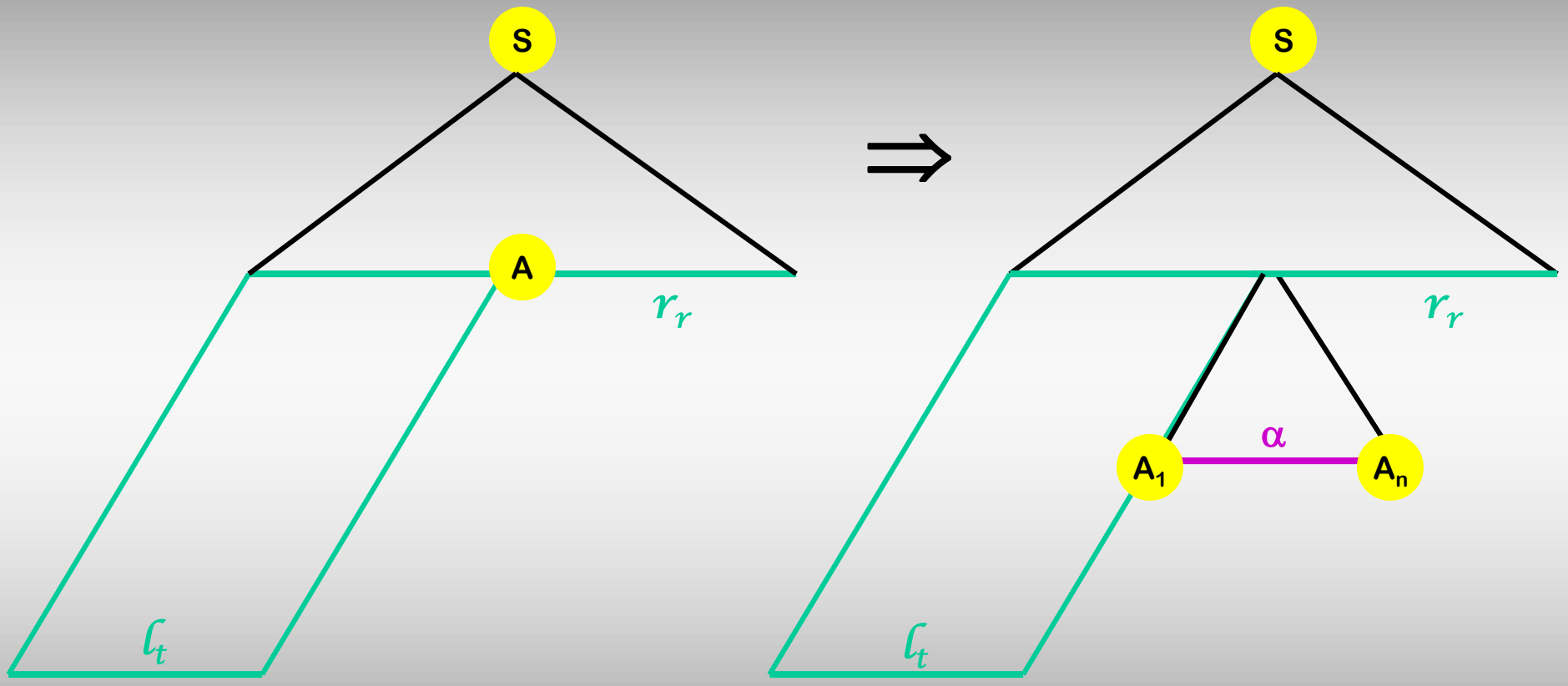
Grundbegriffe zur Syntaxanalyse von CFGs

- Dem Erkennen der Anwendung der Produktion $A \rightarrow \alpha$ entspricht der Übergang von der Satzform s = $\ell_t\, A\, r_r$ zur Satzform s‘ = $\ell_t\, \alpha\, r_r$, 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 Voraussetzung immer in T^*.

  Somit ergeben die bei der Top-Down-Analyse nacheinander gefundenen Ableitungsschritte eine Linksableitung, also:

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

Links-Rechts-Top-Down-Analyse



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.

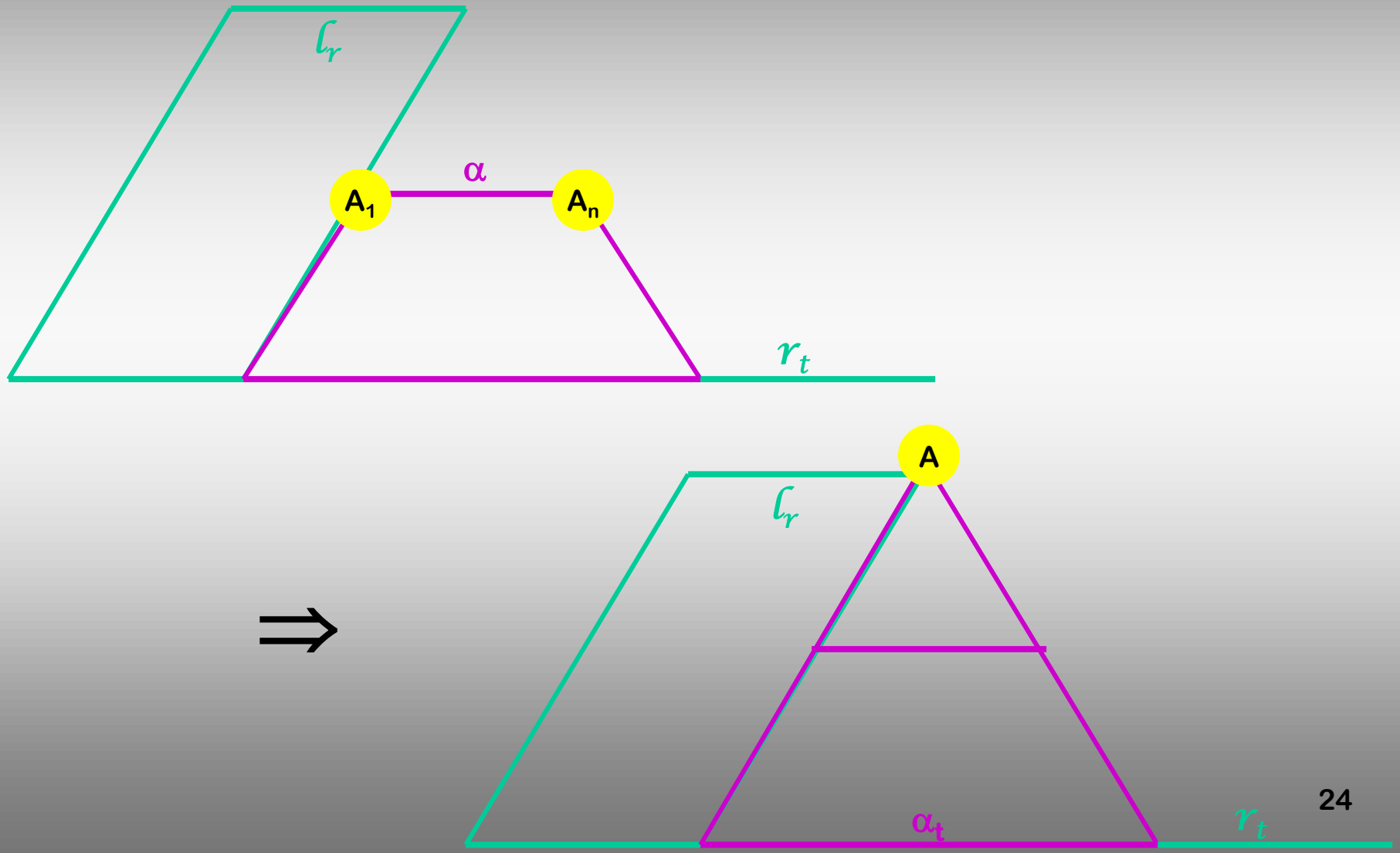
Grundbegriffe zur Syntaxanalyse von CFGs

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

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

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

Links-Rechts-BottomUp-Analyse



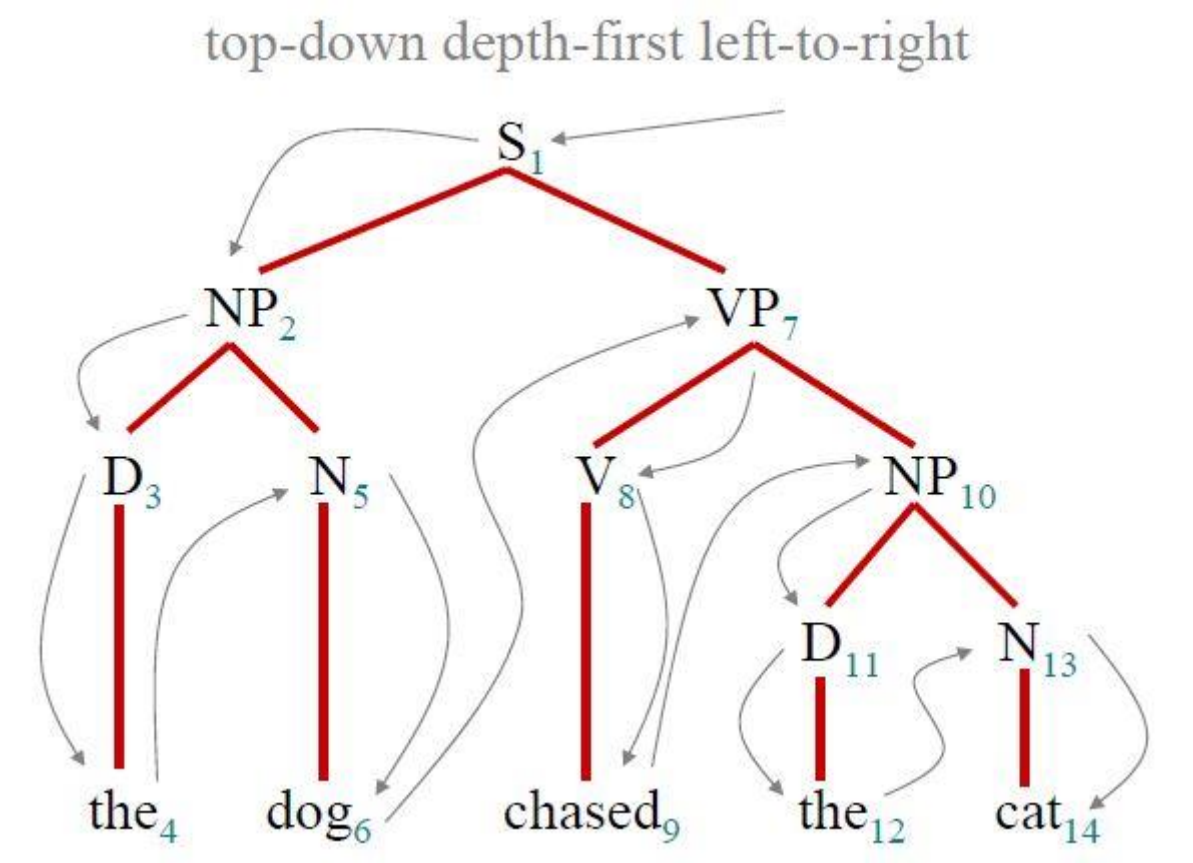
Beispielgrammatik (CFG)

Syntax	Lexikon
S → NP VP	Det → the
NP → Det N	Conj → and
NP → NP Conj NP[1]	N → dog
VP → V NP	N → cat
Det → ε[2]	V → chases

1 nicht für top-down-Parser
2 nicht für bottom-up-Parser

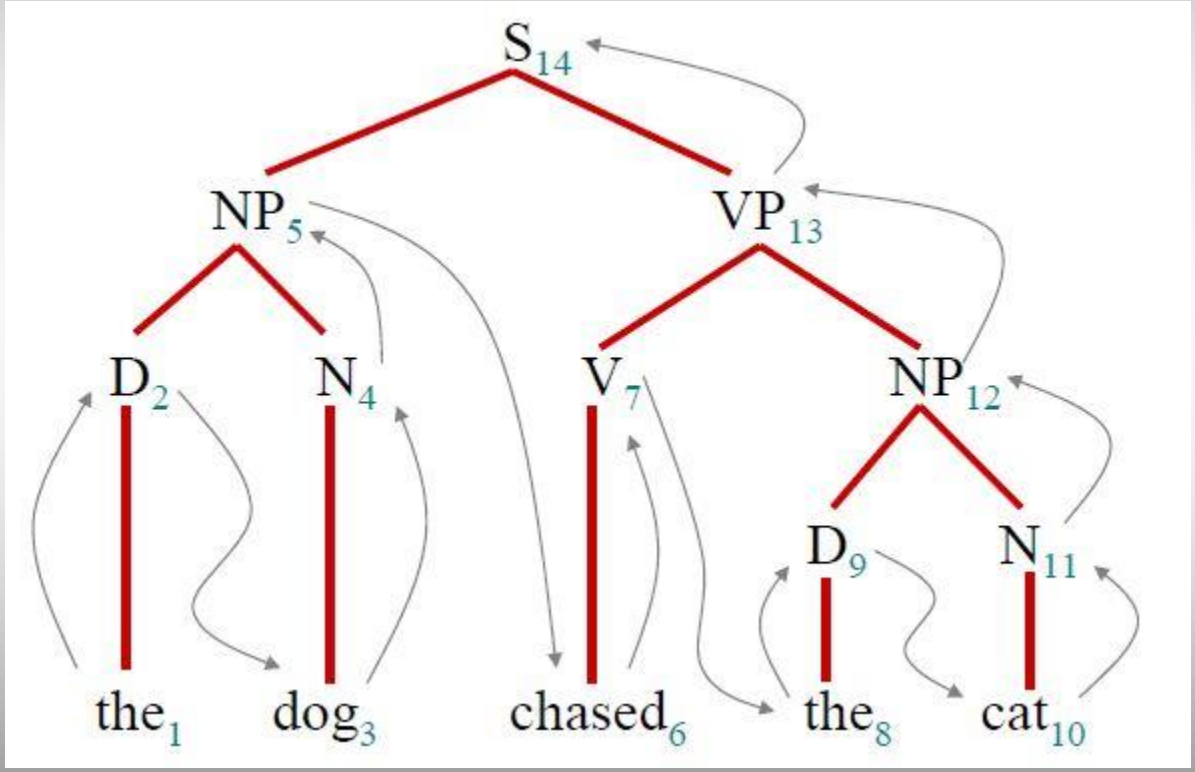
Christof Rumpf (2008): http://www.phil-fak.uni-duesseldorf.de/~rumpf/talks/clintro2.pdf

Top-Down-Traversierung



Christof Rumpf (2008): http://www.phil-fak.uni-duesseldorf.de/~rumpf/talks/clintro2.pdf

Bottom-Up-Traversierung



Christof Rumpf (2008): http://www.phil-fak.uni-duesseldorf.de/~rumpf/talks/clintro2.pdf

Shift-Reduce-Parsing

Der Shift-Reduce-Algorithmus verfolgt eine Bottom-Up-Strategie.

1. Shift: lege ein Wort aus der Eingabekette auf einen Stapel.
2. Reduce: reduziere den Stapel mit Hilfe der Grammatik soweit wie möglich.
3. Falls die Eingabekette noch Wörter enthält, gehe zu Shift, sonst halte.

Christof Rumpf (2008): http://www.phil-fak.uni-duesseldorf.de/~rumpf/talks/clintro2.pdf

Shift-Reduce-Parsing-Beispiel

Schritt	Aktion	Stack	Input String
	Start		the dog barked
1	Shift	the	dog barked
2	Reduce	D	dog barked
3	Shift	D dog	barked
4	Reduce	D N	barked
5	Reduce	NP	barked
6	Shift	NP barked	
7	Reduce	NP V	
8	Reduce	NP VP	
9	Reduce	S	

Christof Rumpf (2008): http://www.phil-fak.uni-duesseldorf.de/~rumpf/talks/clintro2.pdf

Grundbegriffe zur Syntaxanalyse von CFGs

- Sind die gerade betrachteten Übergänge der Satzform s = $\ell_t \, A \, r_r$ auf s‘ = $\ell_t \, \alpha \, r_r$ bei der Top-Down-Analyse bzw. s = $\ell_r \, \alpha \, r_t$ auf s‘ = $\ell_r \, A \, r_t$ 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.