Einführung in die Computerlinguistik
Syntax & Parsing

Dozentin: Wiebke Petersen

19.12.2009

Syntax

- $\sigma\upsilon\nu\tau\alpha\xi\iota\zeta$ (syntaxis) Zusammenordnung, Zusammenstellung
- Syntax ist die Lehre von der Grammatikalität und Struktur der Sätze
  Colourless green ideas sleep furiously.
  (*) Furiously green sleep ideas colourless.
  - Konstituentenstruktur (NP,VP,PP,...)
  - Wortordnung
  - Dependenzstruktur (Subjekt, Objekt, ...)
- Das Ziel der Syntax ist es, das grammatische Wissen zu modellieren, über das die muttersprachlichen Sprecher einer Sprache unbewußt verfügen.
- Wichtige Anwendungsgebiete:
  - Grammatikprüfung
  - Question Answering
  - Informationsextraktion
  - maschinelle Übersetzung

Phrasen- / Konstituentenstruktur

- bestimmte Sequenzen von Wörtern bilden Phrasen oder Konstituenten
- dies sind die Struktureinheiten des Satzes.

Stellen Sie eine Liste aller möglicher Nominalphrasen auf, die in den folgenden Sätzen vorkommen:

- Jonathan Powell, the former Prime Minister's chief of staff, admitted that Mr Blair had made a mistake in an intelligence dossier on Iraq,s nuclear and chemical weapons but said its importance had been overstated. (The Times 18.1.2010, online)
- Baron August von Finck verkaufte die Bank der Familie und investierte bevorzugt in Schweizer Firmen. Für den erzkonservativen Schlossbesitzer gehört es zum guten Ton, ihm wohlgesinnte Parteien mit üppigen Spenden zu unterstützen - nicht nur, wenn es um Hotels geht. (SZ 18.1.2010, online)
- Mehr als sechzig Schlösser gibt es im Schweizer Kanton Thurgau. Zu den schönsten zählt das Schloss Weinfelden, das hoch über dem Ort thront und nur durch eine Zugbrücke zu erreichen ist. (SZ 18.1.2010, online)
- Seit fast vier Jahrzehnten ist der reichste Thurgauer, der Münchner Baron August von Finck, der Hausherr dieser liebevoll restaurierten Liegenschaft, die Kenner an das liechtensteinische Fürstenschloss erinnert: In Stein gehauene Geschichte. (SZ 18.1.2010, online)

Tests für Phrasengliederung / Konstituententests

- Substitutionstest:

Tests für Phrasengliederung / Konstituententests

- Substitutionstest:
  - ich sehe den Mann mit dem Fernglas.
  - ich sehe hundert tollwütige Hunde mit dem Fernglas.
  - (*) ich sehe lief
- Eliminierungstest:

Tests für Phrasengliederung / Konstituententests

- Substitutionstest:
  - ich sehe den Mann mit dem Fernglas.
  - ich sehe hundert tollwütige Hunde mit dem Fernglas.
  - (*) ich sehe lief
- Eliminierungstest:
  - ich sehe hundert tollwütige Hunde mit dem Fernglas
  - ich sehe mit dem Fernglas
  - (*) ich sehe hundert tollwütige
- Fragetest:

Tests für Phrasengliederung / Konstituententests

- Substitutionstest:
  - ich sehe den Mann mit dem Fernglas.
  - ich sehe hundert tollwütige Hunde mit dem Fernglas.
  - (*) ich sehe lief
- Eliminierungstest:
  - ich sehe hundert tollwütige Hunde mit dem Fernglas
  - ich sehe mit dem Fernglas
  - (*) ich sehe hundert tollwütige
- Fragetest:
  - Wen sehe ich?
  - Mit was sehe ich?
- Koordinationstest:

Tests für Phrasengliederung / Konstituententests

- Substitutionstest:
  - ich sehe den Mann mit dem Fernglas.
  - ich sehe hundert tollwütige Hunde mit dem Fernglas.
  - (*) ich sehe lief
- Eliminierungstest:
  - ich sehe hundert tollwütige Hunde mit dem Fernglas
  - ich sehe mit dem Fernglas
  - (*) ich sehe hundert tollwütige
- Fragetest:
  - Wen sehe ich?
  - Mit was sehe ich?
- Koordinationstest:
  - ich sehe den Mann und hundert tollwütige Hunde mit dem Fernglas
- Permutationstest:

Tests für Phrasengliederung / Konstituententests

- Substitutionstest:
  - ich sehe den Mann mit dem Fernglas.
  - ich sehe hundert tollwütige Hunde mit dem Fernglas.
  - (*) ich sehe lief
- Eliminierungstest:
  - ich sehe hundert tollwütige Hunde mit dem Fernglas
  - ich sehe mit dem Fernglas
  - (*) ich sehe hundert tollwütige
- Fragetest:
  - Wen sehe ich?
  - Mit was sehe ich?
- Koordinationstest:
  - ich sehe den Mann und hundert tollwütige Hunde mit dem Fernglas
- Permutationstest:
  - hundert tollwütige Hunde und den Mann sehe ich mit dem Fernglas.
  - mit dem Fernglas sehe ich hundert tollwütige Hunde und den Mann.
  - (*) dem Fernglas sehe ich hundert tollwütige Hunde und den Mann mit.
- ...

Analyse einer kontextfreien Phrasenstrukturregel

$S \rightarrow NP\ VP$

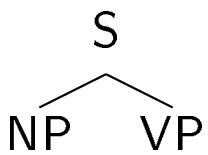
- Es gibt die Konstituenten S, NP, VP in der Sprache.
- Ein S kann aus einer NP gefolgt von einer VP aufgebaut sein.
- Es wird nicht ausgeschlossen, dass es weitere Konstituenten gibt.
- Es wird nicht behauptet, dass dies die einzig mögliche Struktur von S in dieser Sprache ist.
- Es wird nicht behauptet, dass NP und VP nur in dieser Position vorkommen können.

Analyse einer kontextfreien Phrasenstrukturregel

$S \rightarrow NP\ VP$

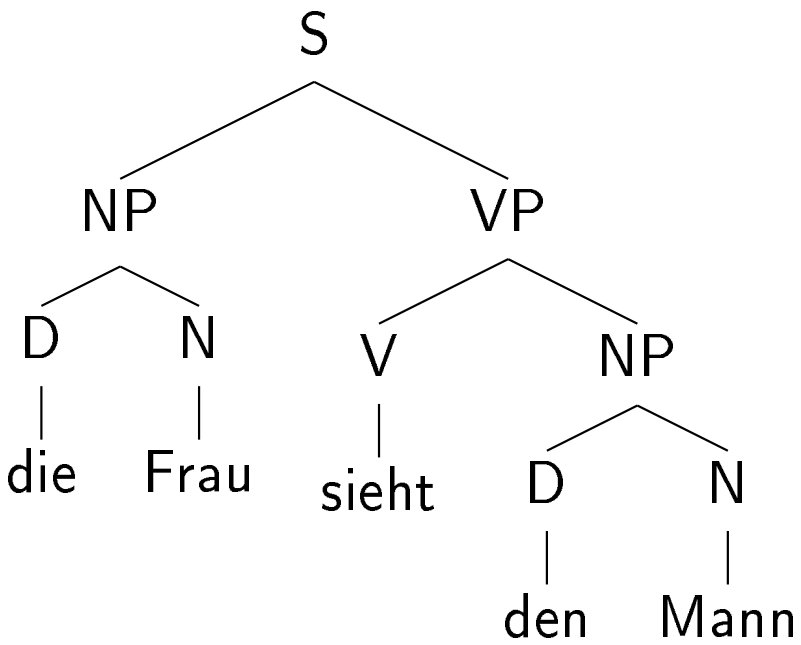
- Es gibt die Konstituenten S, NP, VP in der Sprache.
- Ein S kann aus einer NP gefolgt von einer VP aufgebaut sein.
- Es wird nicht ausgeschlossen, dass es weitere Konstituenten gibt.
- Es wird nicht behauptet, dass dies die einzig mögliche Struktur von S in dieser Sprache ist.
- Es wird nicht behauptet, dass NP und VP nur in dieser Position vorkommen können.

- Eine Phrasenstrukturregel legt fest, aus welchen Konstituenten eine Phrase aufgebaut ist (hierarchische Struktur, direkte Dominanz)
- und sie legt die Reihenfolge der Konstituenten fest (lineare Struktur, direkte Präzedenz)



Syntaxbäume

$$P = \left\{ \begin{array}{llllllllll} S & \rightarrow & NP\ VP & VP & \rightarrow & V\ NP & NP & \rightarrow & D\ N \\ D & \rightarrow & die & D & \rightarrow & den & & & \\ N & \rightarrow & Frau & N & \rightarrow & Mann & V & \rightarrow & sieht \end{array} \right\}$$



Klammerschreibweise:

$[_S[_{NP}[_D die][_N Frau]][_{VP}[_V sieht][_{NP}[_D den][_N Mann]]]]$

Über- und Untergenerierung

$$P = \left\{ \begin{array}{llllllll} S & \rightarrow & NP\ VP & VP & \rightarrow & V\ NP & NP & \rightarrow & D\ N \\ D & \rightarrow & die & D & \rightarrow & den & & & \\ N & \rightarrow & Frau & N & \rightarrow & Mann & V & \rightarrow & sieht \end{array} \right\}$$

- Eine Grammatik mit diesen Produktionen erzeugt Sätze, die im Deutschen nicht grammatikalisch sind. (Übergenerierung)
  Bsp.: Den Mann sieht den Mann, den Frau sieht die Mann
- Eine Grammatik mit diesen Produktionen erzeugt nicht alle grammatikalischen Sätze des Deutschen (Untergenerierung)
  Bsp.: Die Computerlinguistik ist eine Teildisziplin der Linguistik.

Parsing

- *to parse* (grammatisch zerlegen) abgeleitet von *pars* (lateinisch) Teil
- Ein Parser ist ein Automat, der einer Zeichenkette aufgrund einer Grammatik einen Derivationsbaum zuordnet.

$$\begin{array}{c} \text{Grammatik} \\ + \\ \text{Zeichenkette} \end{array} \longrightarrow \text{Derivationsbaum}$$

Parsing ist ein Suchproblem.

Unterschied: Recognizer – Parser

Beides sind Automaten

Recognizer: stellt ausschließlich fest, ob eine Zeichenfolge ein Wort der von der Grammatik generierten Sprache ist oder nicht (Kellerautomat).

Parser: erstellt den Derivationsbaum einer Zeichenfolge im Bezug auf die Grammatik.

Parsingstrategien

Parsingstrategien unterscheiden sich darin, in welcher Reihenfolge die Knoten eines Derivationsbaums erstellt werden.
Man unterscheidet zwei Hauptstrategien voneinander

- **inputgetriebenes** Parsing (**bottom up**): geleitet von der zu parsenden Zeichenkette
- **theoriegetriebenes** Parsing (**top down**): geleitet von der Grammatik

Zusätzlich charakterisiert man Parsingstrategien mit folgenden Begriffen:

depth-first $\leftrightarrow$ breadth-first

left-to-right $\leftrightarrow$ right-to-left

top-down-left-to-right-depth-first-Parser

top-down:

- Parser **beginnt beim Startsymbol** S und versucht, durch sukzessive Regelanwendung schließlich bei der Eingabekette zu landen.
- Regelanwendungen (von links nach rechts) nennt man **Expansion**.
- Das Einlesen eines Elements der Eingabekette nennt man **Scan**.

left-to-right: Der Parser versucht immer den am weitesten links stehenden Knoten des Ableitungsbaums zu expandieren oder mit diesem Knoten einen Scan durchzuführen.

depth-first: Der Parser versucht immer die am weitesten unten stehenden Knoten (das sind immer die zuletzt gebildeten) weiter zu expandieren oder hier einen Scan durchzuführen.

Beispiel: top-down-left-to-right-depth-first-Parser

$$P = \left\{ \begin{array}{lllllllll} S & \rightarrow & NP\ VP & VP & \rightarrow & V\ NP & NP & \rightarrow & D\ N \\ D & \rightarrow & die & D & \rightarrow & den & & & \\ N & \rightarrow & Frau & N & \rightarrow & Mann & V & \rightarrow & sieht \end{array} \right\}$$

Die Frau sieht den Mann

S

Beispiel: top-down-left-to-right-depth-first-Parser

$$P = \left\{ \begin{array}{lllllllll} S & \rightarrow & NP\ VP & VP & \rightarrow & V\ NP & NP & \rightarrow & D\ N \\ D & \rightarrow & die & D & \rightarrow & den & & & \\ N & \rightarrow & Frau & N & \rightarrow & Mann & V & \rightarrow & sieht \end{array} \right\}$$

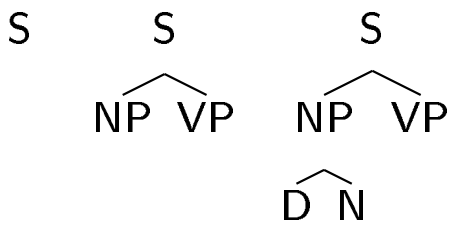
Die Frau sieht den Mann

S S

Beispiel: top-down-left-to-right-depth-first-Parser

$$P = \left\{ \begin{array}{llllllllll} S & \rightarrow & NP\ VP & VP & \rightarrow & V\ NP & NP & \rightarrow & D\ N \\ D & \rightarrow & die & D & \rightarrow & den & & & \\ N & \rightarrow & Frau & N & \rightarrow & Mann & V & \rightarrow & sieht \end{array} \right\}$$

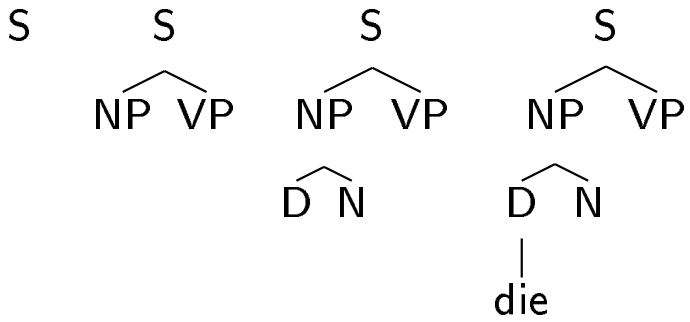
Die Frau sieht den Mann



Beispiel: top-down-left-to-right-depth-first-Parser

$$P = \left\{ \begin{array}{lllllllll} S & \rightarrow & NP\ VP & VP & \rightarrow & V\ NP & NP & \rightarrow & D\ N \\ D & \rightarrow & die & D & \rightarrow & den & & & \\ N & \rightarrow & Frau & N & \rightarrow & Mann & V & \rightarrow & sieht \end{array} \right\}$$

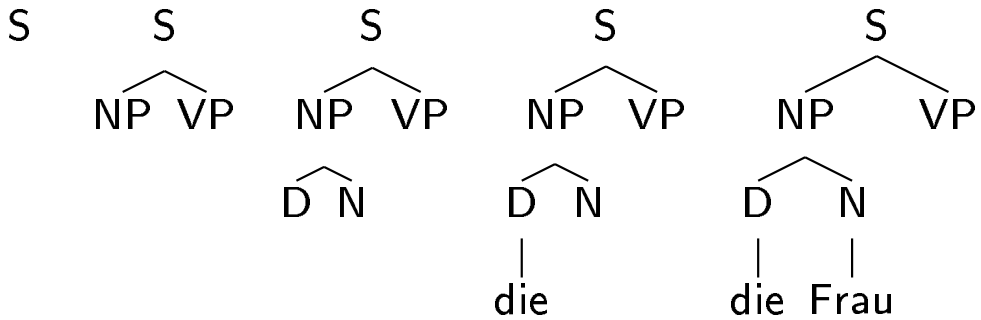
Die Frau sieht den Mann



Beispiel: top-down-left-to-right-depth-first-Parser

$$P = \left\{ \begin{array}{llllllllll} S & \rightarrow & NP\ VP & VP & \rightarrow & V\ NP & NP & \rightarrow & D\ N \\ D & \rightarrow & die & D & \rightarrow & den & & & \\ N & \rightarrow & Frau & N & \rightarrow & Mann & V & \rightarrow & sieht \end{array} \right\}$$

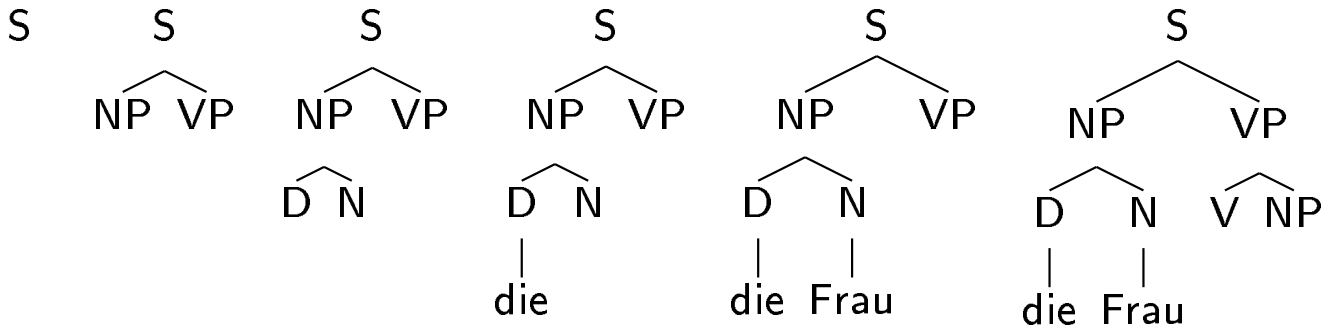
Die Frau sieht den Mann



Beispiel: top-down-left-to-right-depth-first-Parser

$$P = \left\{ \begin{array}{llllllllll} S & \rightarrow & NP\ VP & VP & \rightarrow & V\ NP & NP & \rightarrow & D\ N \\ D & \rightarrow & die & D & \rightarrow & den & & & \\ N & \rightarrow & Frau & N & \rightarrow & Mann & V & \rightarrow & sieht \end{array} \right\}$$

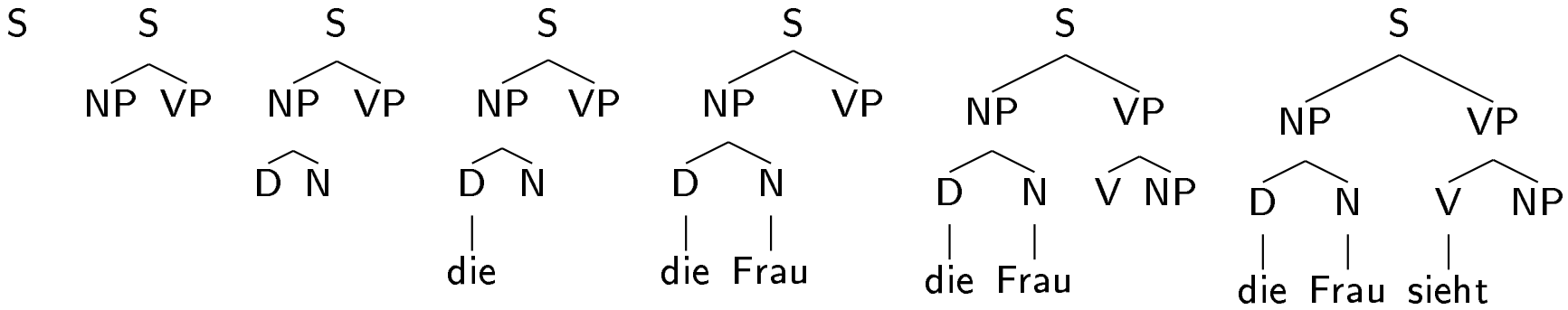
Die Frau sieht den Mann



Beispiel: top-down-left-to-right-depth-first-Parser

$$P = \left\{ \begin{array}{llllllllll} S & \rightarrow & NP\ VP & VP & \rightarrow & V\ NP & NP & \rightarrow & D\ N \\ D & \rightarrow & die & D & \rightarrow & den & & & \\ N & \rightarrow & Frau & N & \rightarrow & Mann & V & \rightarrow & sieht \end{array} \right\}$$

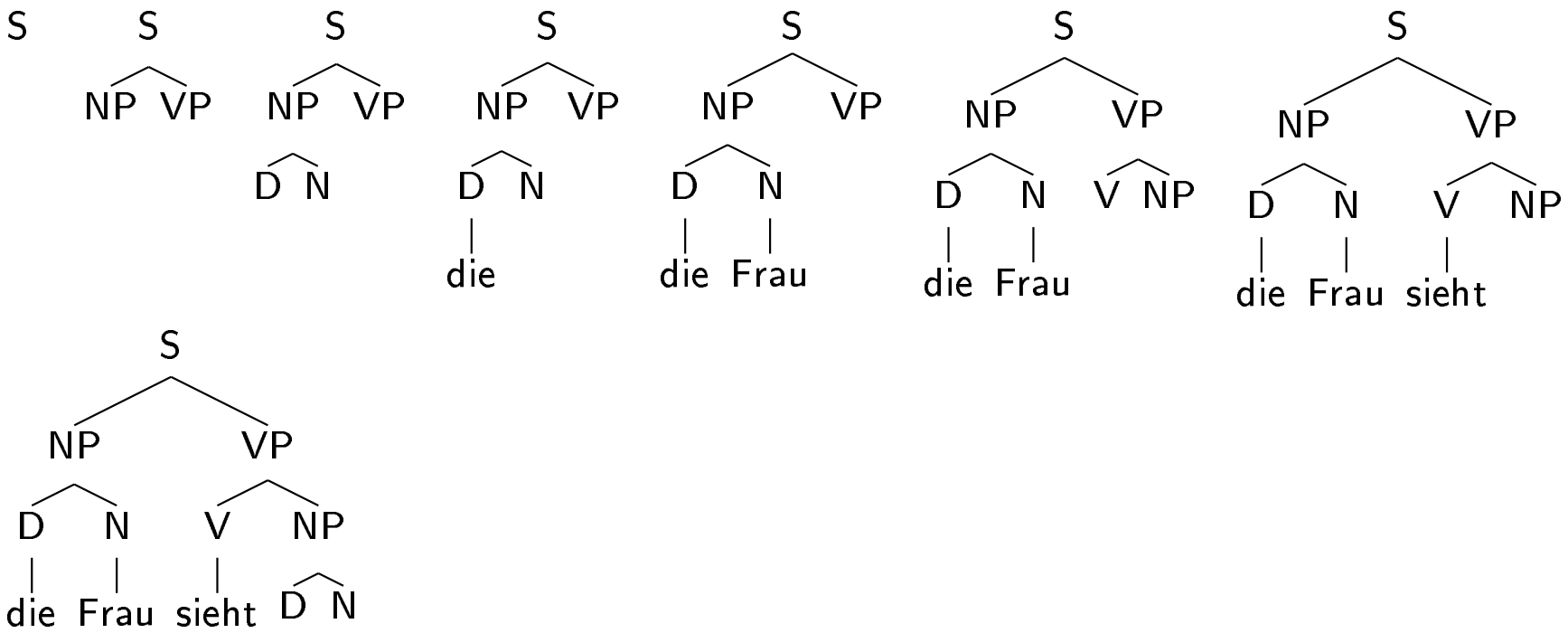
Die Frau sieht den Mann



Beispiel: top-down-left-to-right-depth-first-Parser

$$P = \left\{ \begin{array}{llllllllll} S & \rightarrow & NP\ VP & VP & \rightarrow & V\ NP & NP & \rightarrow & D\ N \\ D & \rightarrow & die & D & \rightarrow & den & & & \\ N & \rightarrow & Frau & N & \rightarrow & Mann & V & \rightarrow & sieht \end{array} \right\}$$

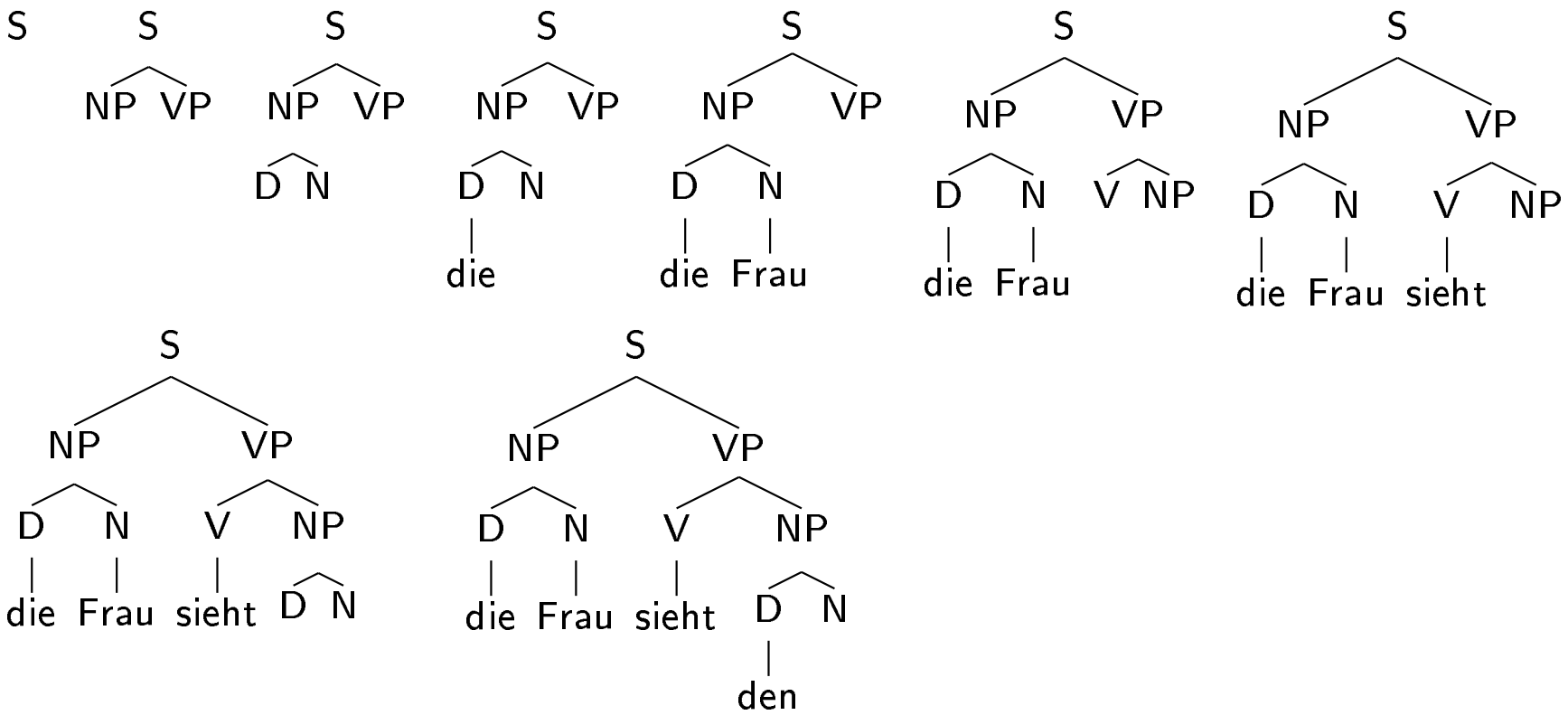
Die Frau sieht den Mann



Beispiel: top-down-left-to-right-depth-first-Parser

$$P = \left\{ \begin{array}{llllllllll} S & \rightarrow & NP\ VP & VP & \rightarrow & V\ NP & NP & \rightarrow & D\ N \\ D & \rightarrow & die & D & \rightarrow & den & & & \\ N & \rightarrow & Frau & N & \rightarrow & Mann & V & \rightarrow & sieht \end{array} \right\}$$

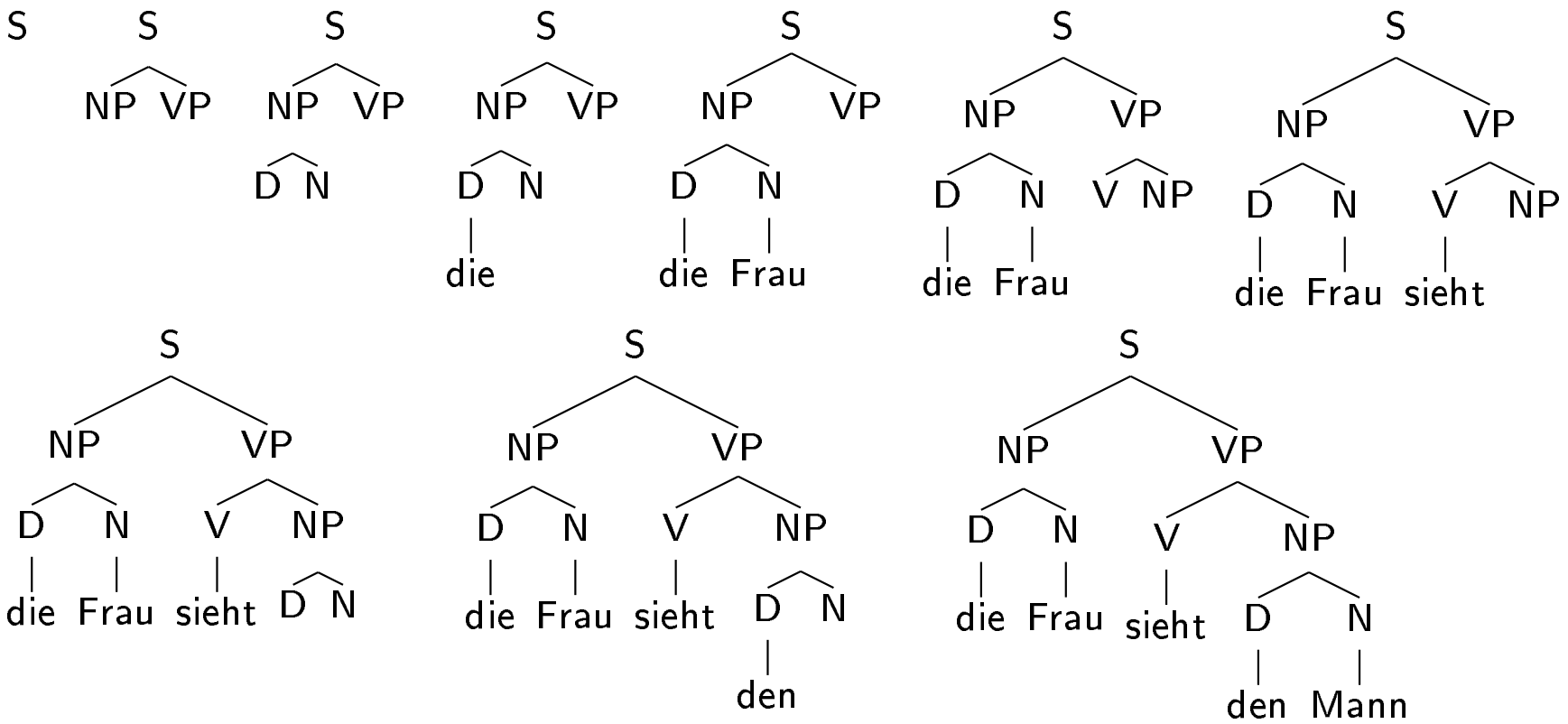
Die Frau sieht den Mann



Beispiel: top-down-left-to-right-depth-first-Parser

$$P = \left\{ \begin{array}{llllllllll} S & \rightarrow & NP\ VP & VP & \rightarrow & V\ NP & NP & \rightarrow & D\ N \\ D & \rightarrow & die & D & \rightarrow & den & & & \\ N & \rightarrow & Frau & N & \rightarrow & Mann & V & \rightarrow & sieht \end{array} \right\}$$

Die Frau sieht den Mann



Linksrekursion

Top-down-left-to-right-Parser terminieren nicht bei Grammatiken, die linksrekursive Regeln beinhalten!

S $\rightarrow$ S und S

NP $\rightarrow$ NP PP

s

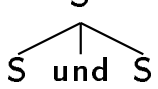
Linksrekursion

Top-down-left-to-right-Parser terminieren nicht bei Grammatiken, die linksrekursive Regeln beinhalten!

S → S und S

NP → NP PP

S

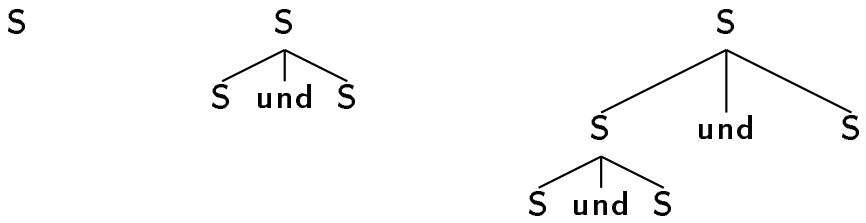


Linksrekursion

Top-down-left-to-right-Parser terminieren nicht bei Grammatiken, die linksrekursive Regeln beinhalten!

S → S und S

NP → NP PP

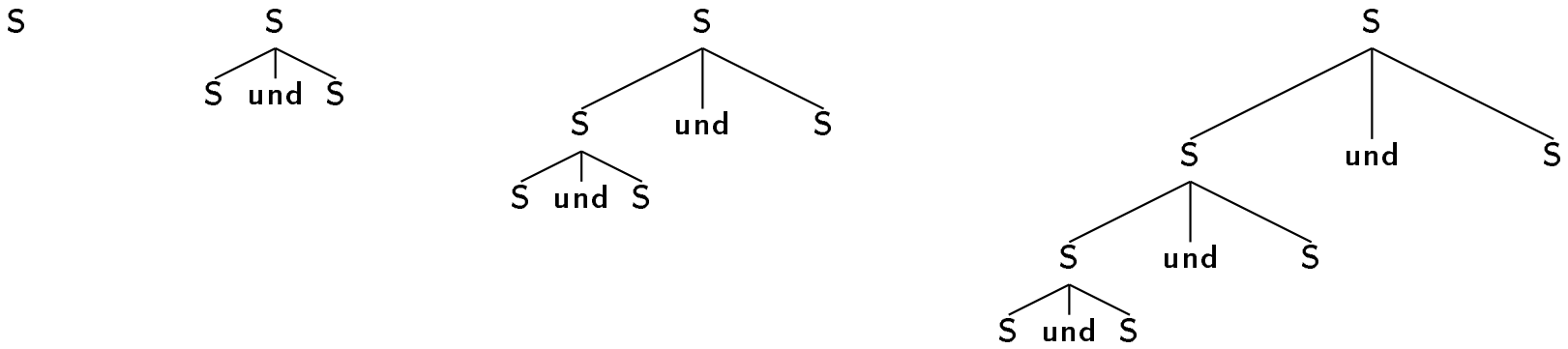


Linksrekursion

Top-down-left-to-right-Parser terminieren nicht bei Grammatiken, die linksrekursive Regeln beinhalten!

S → S und S

NP → NP PP

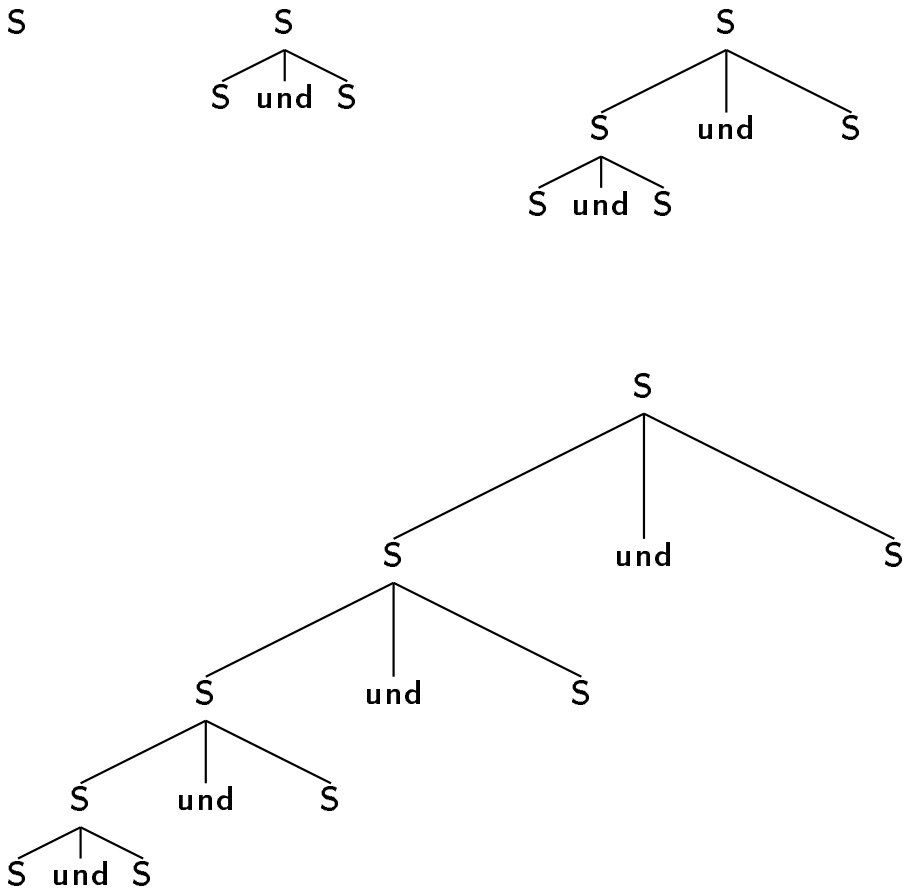


Linksrekursion

Top-down-left-to-right-Parser terminieren nicht bei Grammatiken, die linksrekursive Regeln beinhalten!

S → S und S

NP → NP PP



bottom-up-breadth-first-left-to-right-Parser

bottom-up:

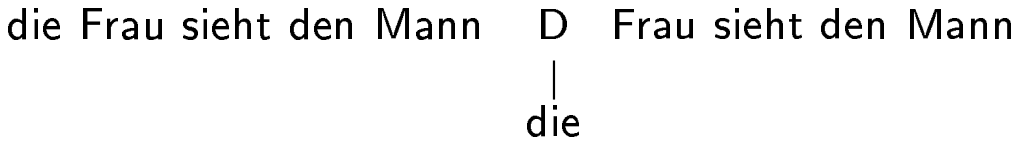
- Parser **beginnt bei der Eingabekette** und versucht, durch sukzessives **rückwärtiges** Anwenden der Regeln (von rechts nach links) schließlich bei dem Startsymbol S zu landen.

breadth-first: Die Symbole werden in der Reihenfolge ihrer Erzeugung abgearbeitet

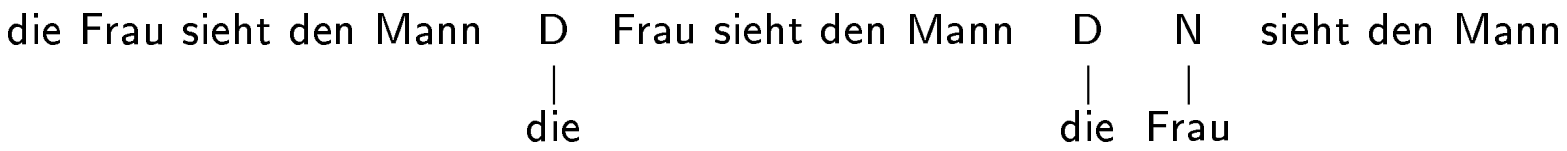
Beispiel: bottom-up-breadth-first-left-to-right-Parser

die Frau sieht den Mann

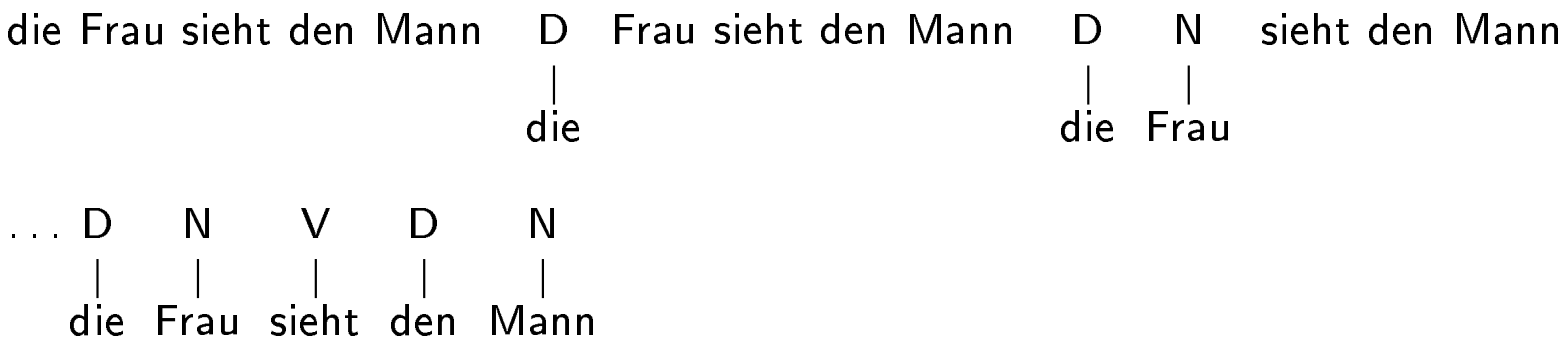
Beispiel: bottom-up-breadth-first-left-to-right-Parser



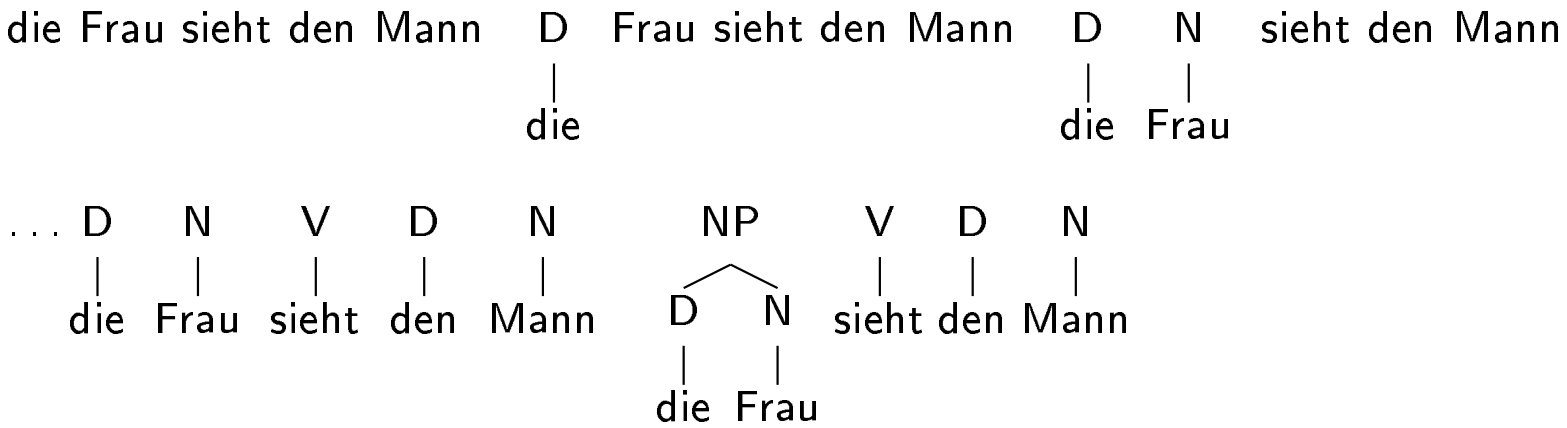
Beispiel: bottom-up-breadth-first-left-to-right-Parser



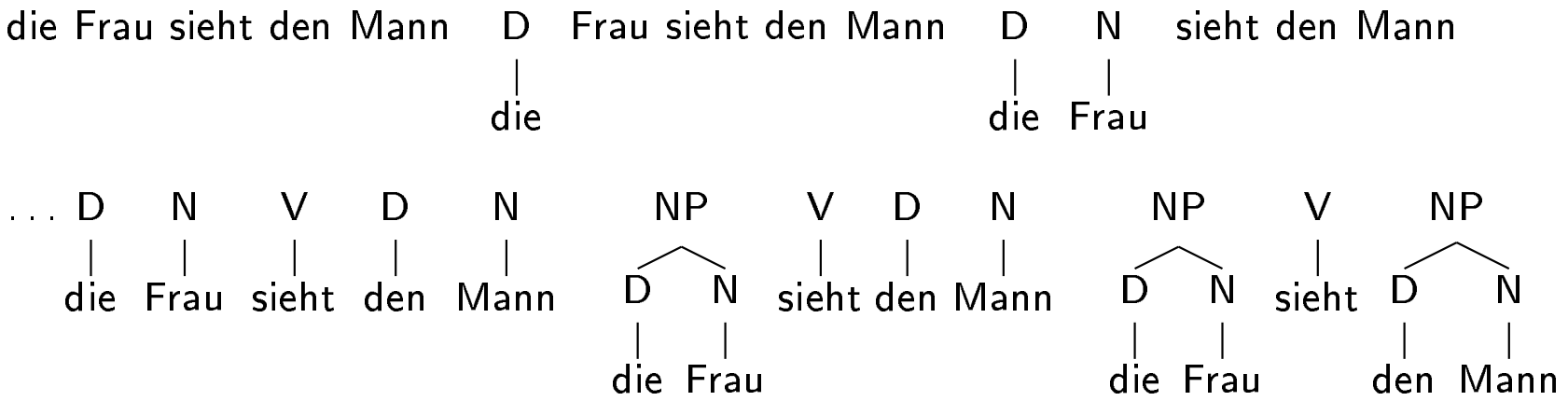
Beispiel: bottom-up-breadth-first-left-to-right-Parser



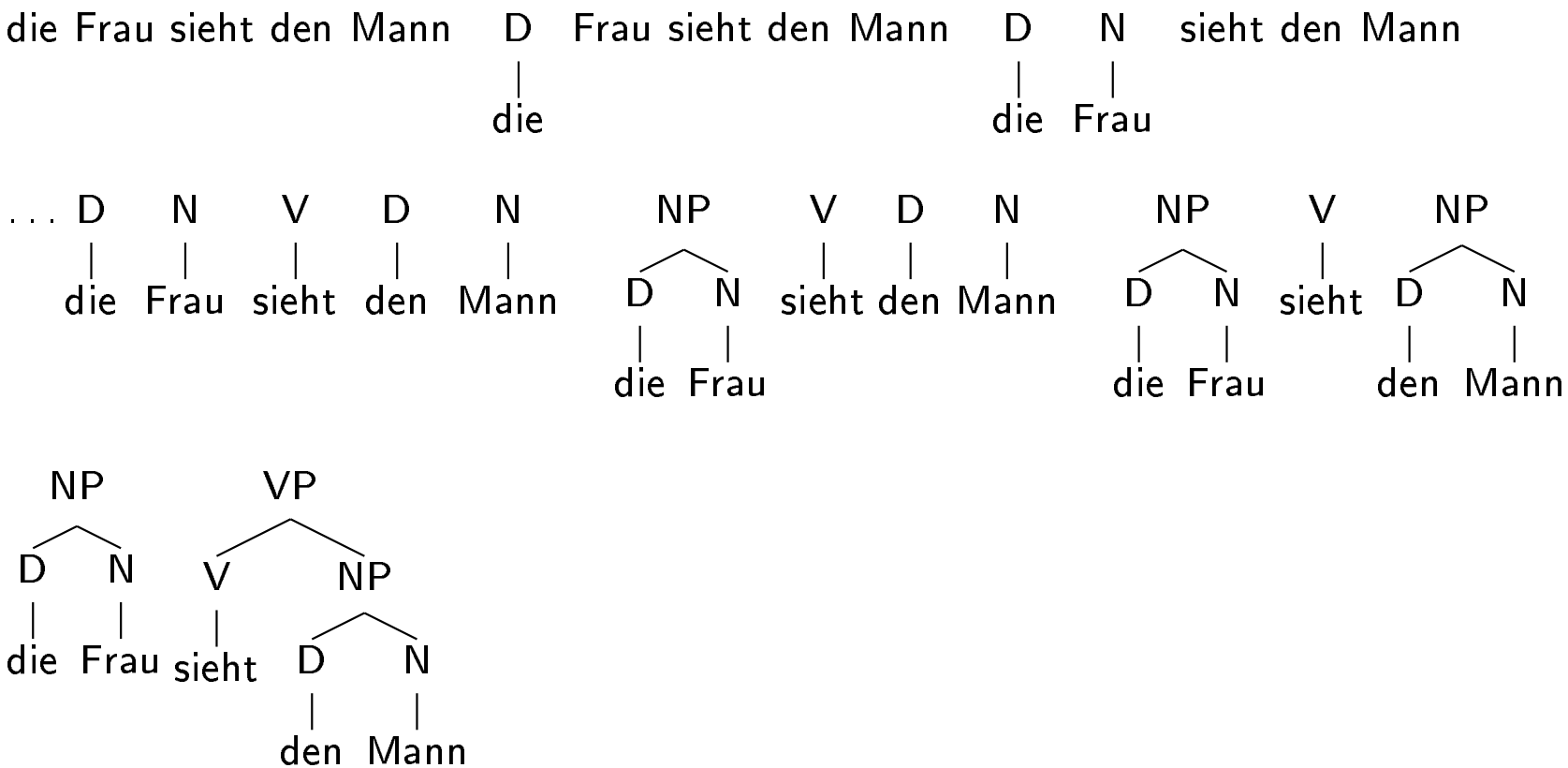
Beispiel: bottom-up-breadth-first-left-to-right-Parser



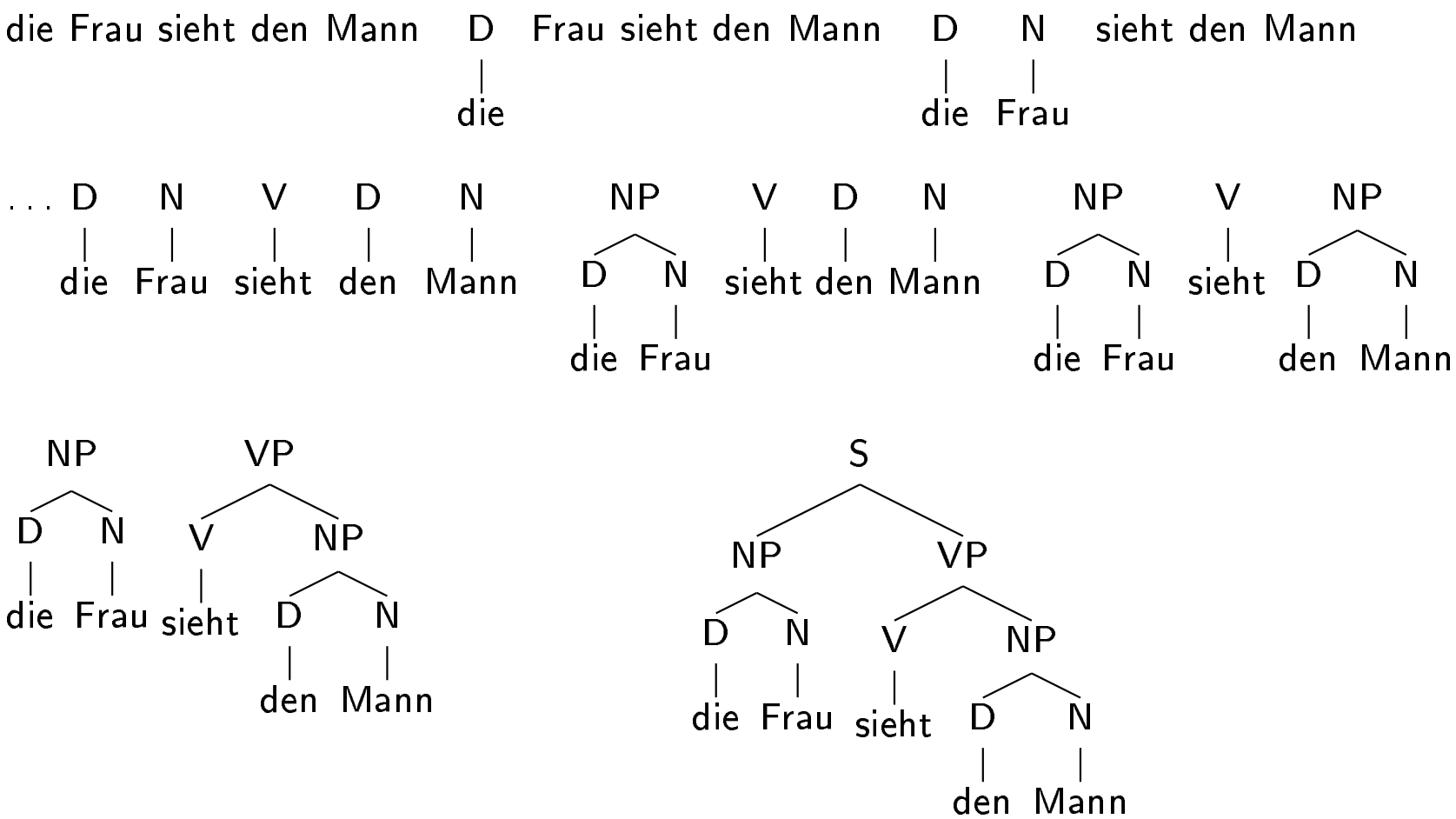
Beispiel: bottom-up-breadth-first-left-to-right-Parser



Beispiel: bottom-up-breadth-first-left-to-right-Parser



Beispiel: bottom-up-breadth-first-left-to-right-Parser



ϵ-Regeln

Bottom-up Parser terminieren nicht bei Grammatiken, die ϵ-Regeln beinhalten, da eine solche Regel jederzeit anwendbar ist!

$$S \rightarrow \epsilon$$

Vergleich: Bottom-up- und Top-down-Parser

top-down

- Sucht nur nach Derivationsbäumen, die echte Bäume sind.
- Aber verfolgt Bäume, die nicht zu der Eingabekette passen.
- Problem mit Linksrekursion.

bottom-up

- Formt nur Teilbäume, die zur Eingabekette passen.
- Aber verfolgt Teilbäume, die nie zu einem Derivationsbaum werden können.
- Problem mit ϵ-Expansion.

Top-down Parsing mit Backtracking

Falls ein Terminalsymbol mit der Eingabekette inkonsistent ist, werden schrittweise die vorherigen Schritte bis zur letzten Wahlmöglichkeit rückgängig gemacht

$$P = \left\{ \begin{array}{lcllcllcll} S & \rightarrow & NP\ VP & VP & \rightarrow & V & VP & \rightarrow & V\ NP \\ V & \rightarrow & flies & V & \rightarrow & calls & & & \\ NP & \rightarrow & Peter & NP & \rightarrow & Mary & & & \end{array} \right\}$$

Mary calls Peter

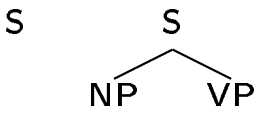
S

Top-down Parsing mit Backtracking

Falls ein Terminalsymbol mit der Eingabekette inkonsistent ist, werden schrittweise die vorherigen Schritte bis zur letzten Wahlmöglichkeit rückgängig gemacht

$$P = \left\{ \begin{array}{lllllllll} \text{S} & \rightarrow & \text{NP VP} & \text{VP} & \rightarrow & \text{V} & \text{VP} & \rightarrow & \text{V NP} \\ \text{V} & \rightarrow & \text{flies} & \text{V} & \rightarrow & \text{calls} & & & \\ \text{NP} & \rightarrow & \text{Peter} & \text{NP} & \rightarrow & \text{Mary} & & & \end{array} \right\}$$

Mary calls Peter

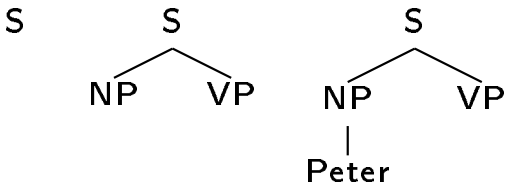


Top-down Parsing mit Backtracking

Falls ein Terminalsymbol mit der Eingabekette inkonsistent ist, werden schrittweise die vorherigen Schritte bis zur letzten Wahlmöglichkeit rückgängig gemacht

$$P = \left\{ \begin{array}{lllllllll} \text{S} & \rightarrow & \text{NP VP} & \text{VP} & \rightarrow & \text{V} & \text{VP} & \rightarrow & \text{V NP} \\ \text{V} & \rightarrow & \text{flies} & \text{V} & \rightarrow & \text{calls} & & & \\ \text{NP} & \rightarrow & \text{Peter} & \text{NP} & \rightarrow & \text{Mary} & & & \end{array} \right\}$$

Mary calls Peter

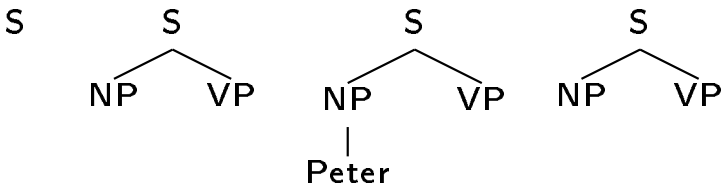


Top-down Parsing mit Backtracking

Falls ein Terminalsymbol mit der Eingabekette inkonsistent ist, werden schrittweise die vorherigen Schritte bis zur letzten Wahlmöglichkeit rückgängig gemacht

$$P = \left\{ \begin{array}{lllllllll} \text{S} & \rightarrow & \text{NP VP} & \text{VP} & \rightarrow & \text{V} & \text{VP} & \rightarrow & \text{V NP} \\ \text{V} & \rightarrow & \text{flies} & \text{V} & \rightarrow & \text{calls} & & & \\ \text{NP} & \rightarrow & \text{Peter} & \text{NP} & \rightarrow & \text{Mary} & & & \end{array} \right\}$$

Mary calls Peter

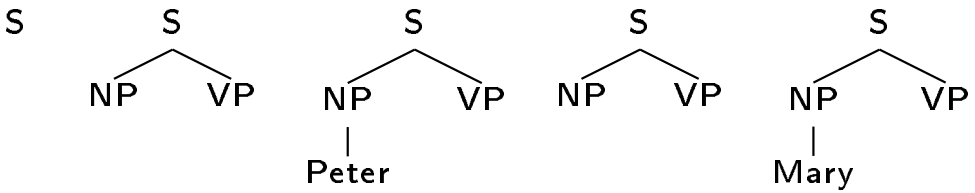


Top-down Parsing mit Backtracking

Falls ein Terminalsymbol mit der Eingabekette inkonsistent ist, werden schrittweise die vorherigen Schritte bis zur letzten Wahlmöglichkeit rückgängig gemacht

$$P = \left\{ \begin{array}{lllllllll} S & \rightarrow & NP\ VP & VP & \rightarrow & V & VP & \rightarrow & V\ NP \\ V & \rightarrow & flies & V & \rightarrow & calls & & & \\ NP & \rightarrow & Peter & NP & \rightarrow & Mary & & & \end{array} \right\}$$

Mary calls Peter

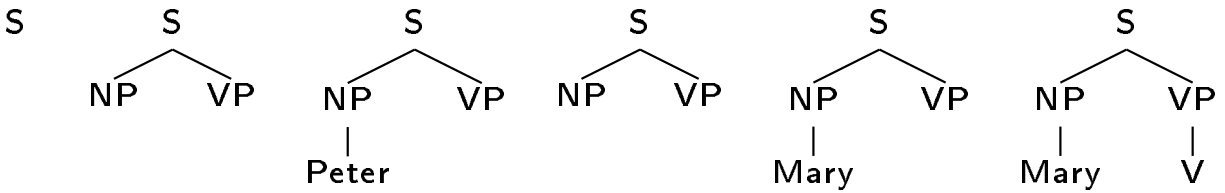


Top-down Parsing mit Backtracking

Falls ein Terminalsymbol mit der Eingabekette inkonsistent ist, werden schrittweise die vorherigen Schritte bis zur letzten Wahlmöglichkeit rückgängig gemacht

$$P = \left\{ \begin{array}{llllllll} \text{S} & \rightarrow & \text{NP VP} & \text{VP} & \rightarrow & \text{V} & \text{VP} & \rightarrow & \text{V NP} \\ \text{V} & \rightarrow & \text{flies} & \text{V} & \rightarrow & \text{calls} & & & \\ \text{NP} & \rightarrow & \text{Peter} & \text{NP} & \rightarrow & \text{Mary} & & & \end{array} \right\}$$

Mary calls Peter

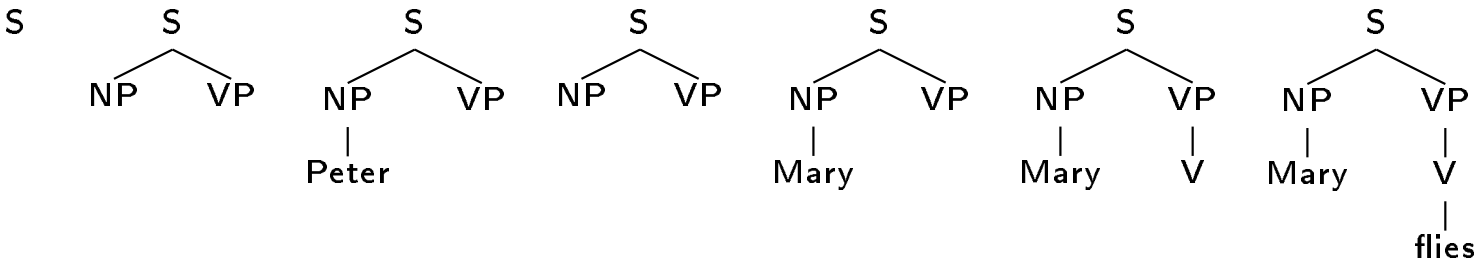


Top-down Parsing mit Backtracking

Falls ein Terminalsymbol mit der Eingabekette inkonsistent ist, werden schrittweise die vorherigen Schritte bis zur letzten Wahlmöglichkeit rückgängig gemacht

$$P = \left\{ \begin{array}{llllllllll} \text{S} & \rightarrow & \text{NP VP} & \text{VP} & \rightarrow & \text{V} & \text{VP} & \rightarrow & \text{V NP} \\ \text{V} & \rightarrow & \text{flies} & \text{V} & \rightarrow & \text{calls} & & & \\ \text{NP} & \rightarrow & \text{Peter} & \text{NP} & \rightarrow & \text{Mary} & & & \end{array} \right\}$$

Mary calls Peter

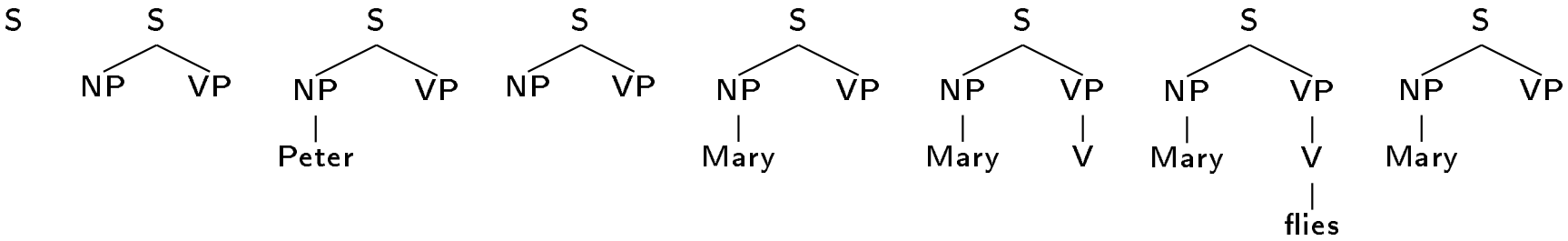


Top-down Parsing mit Backtracking

Falls ein Terminalsymbol mit der Eingabekette inkonsistent ist, werden schrittweise die vorherigen Schritte bis zur letzten Wahlmöglichkeit rückgängig gemacht

$$P = \left\{ \begin{array}{llllllllll} \text{S} & \rightarrow & \text{NP VP} & \text{VP} & \rightarrow & \text{V} & \text{VP} & \rightarrow & \text{V NP} \\ \text{V} & \rightarrow & \text{flies} & \text{V} & \rightarrow & \text{calls} & & & \\ \text{NP} & \rightarrow & \text{Peter} & \text{NP} & \rightarrow & \text{Mary} & & & \end{array} \right\}$$

Mary calls Peter

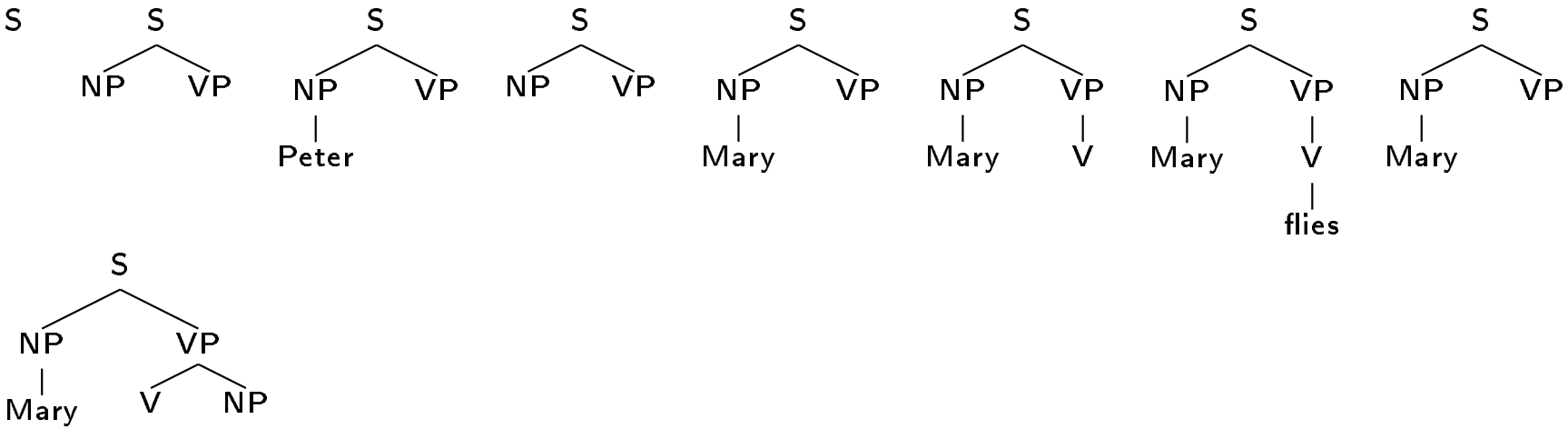


Top-down Parsing mit Backtracking

Falls ein Terminalsymbol mit der Eingabekette inkonsistent ist, werden schrittweise die vorherigen Schritte bis zur letzten Wahlmöglichkeit rückgängig gemacht

$$P = \left\{ \begin{array}{llllllllll} \text{S} & \rightarrow & \text{NP VP} & \text{VP} & \rightarrow & \text{V} & \text{VP} & \rightarrow & \text{V NP} \\ \text{V} & \rightarrow & \text{flies} & \text{V} & \rightarrow & \text{calls} & & & \\ \text{NP} & \rightarrow & \text{Peter} & \text{NP} & \rightarrow & \text{Mary} & & & \end{array} \right\}$$

Mary calls Peter

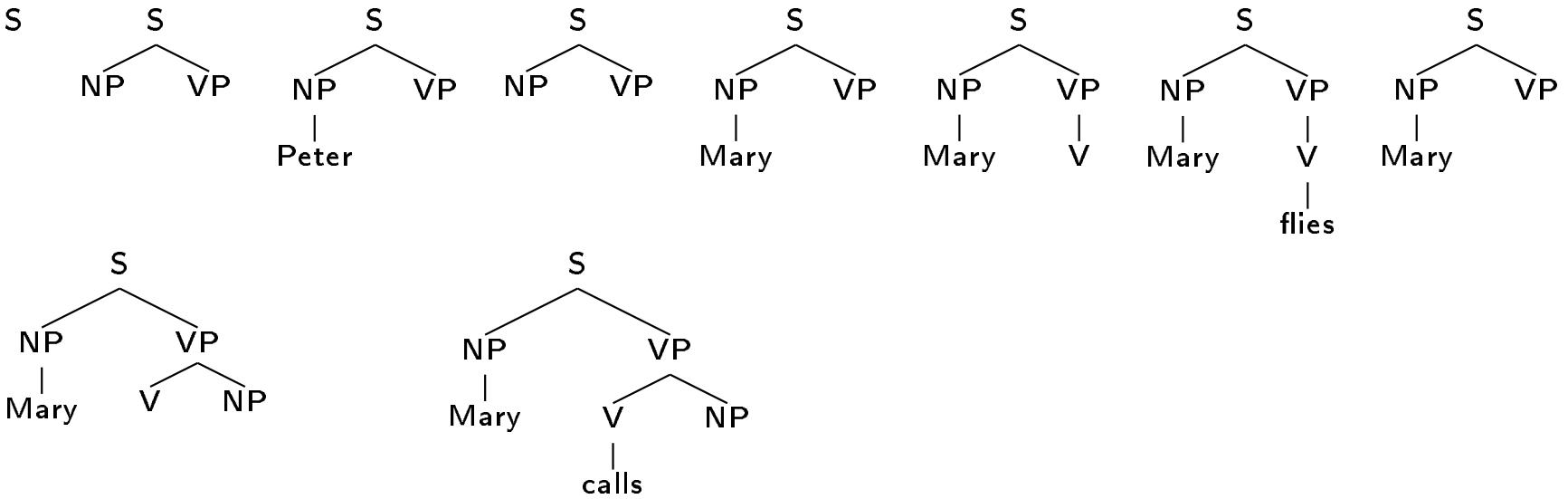


Top-down Parsing mit Backtracking

Falls ein Terminalsymbol mit der Eingabekette inkonsistent ist, werden schrittweise die vorherigen Schritte bis zur letzten Wahlmöglichkeit rückgängig gemacht

$$P = \left\{ \begin{array}{llllllllll} \text{S} & \rightarrow & \text{NP VP} & \text{VP} & \rightarrow & \text{V} & \text{VP} & \rightarrow & \text{V NP} \\ \text{V} & \rightarrow & \text{flies} & \text{V} & \rightarrow & \text{calls} & & & \\ \text{NP} & \rightarrow & \text{Peter} & \text{NP} & \rightarrow & \text{Mary} & & & \end{array} \right\}$$

Mary calls Peter

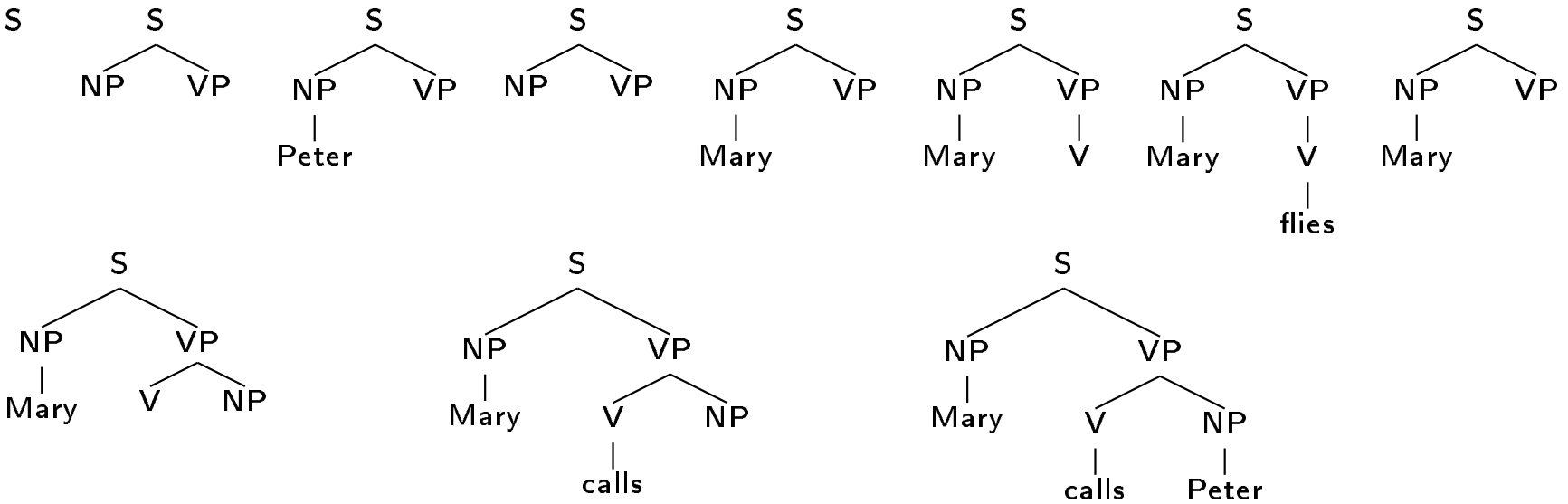


Top-down Parsing mit Backtracking

Falls ein Terminalsymbol mit der Eingabekette inkonsistent ist, werden schrittweise die vorherigen Schritte bis zur letzten Wahlmöglichkeit rückgängig gemacht

$$P = \left\{ \begin{array}{llllllllll} \text{S} & \rightarrow & \text{NP VP} & \text{VP} & \rightarrow & \text{V} & \text{VP} & \rightarrow & \text{V NP} \\ \text{V} & \rightarrow & \text{flies} & \text{V} & \rightarrow & \text{calls} & & & \\ \text{NP} & \rightarrow & \text{Peter} & \text{NP} & \rightarrow & \text{Mary} & & & \end{array} \right\}$$

Mary calls Peter



Top-down Parsing mit Backtracking (Beispiel aus Carstensen et. al.)

Nr.	Linkssatzform	Eingabe	Schritt
1	S	*der Hund sieht die Katze*	
2	NP VP	*der Hund sieht die Katze*	EXPAND
3	Det N VP	*der Hund sieht die Katze*	EXPAND
4	*der* N VP	*der Hund sieht die Katze*	EXPAND
5	N VP	*Hund sieht die Katze*	SCAN
6	*Hund* VP	*Hund sieht die Katze*	EXPAND
7	VP	*sieht die Katze*	SCAN
8	V	*sieht die Katze*	EXPAND
9	*bellt*	*sieht die Katze*	EXPAND
8′	V	*sieht die Katze*	BACKTRACK nach 8
9′	*sieht*	*sieht die Katze*	EXPAND
7″	VP	*sieht die Katze*	BACKTRACK nach 7
8″	V NP	*sieht die Katze*	EXPAND
9″	*bellt* NP	*sieht die Katze*	EXPAND
8‴	V NP	*sieht die Katze*	BACKTRACK nach 8″
9‴	*sieht* NP	*sieht die Katze*	EXPAND
10‴	NP	*die Katze*	SCAN
11‴	Det N	*die Katze*	EXPAND
12‴	*der* N	*die Katze*	EXPAND
11⁗	Det N	*die Katze*	BACKTRACK nach 11‴
12⁗	*die* N	*die Katze*	EXPAND
13⁗	N	*Katze*	SCAN
14⁗	*Hund*	*Katze*	EXPAND
13⁗⁗	N	*Katze*	BACKTRACK nach 13⁗
14⁗⁗	*Katze*	*Katze*	EXPAND
15⁗⁗			SCAN, akzeptiere.

Parsing: Probleme

- hohe Ambiguität (Bsp.: 'time flies like an arrow', weiteres Bsp. in den HA)
- Abdeckung
- Effizienz

Hausaufgaben (Abgabe bis zum 26.1.2010) (BN: 2 Aufgaben)

1. Geben Sie zu mindestens zwei der Tests für Phrasengliederung ein Beispiel an, warum dieser Test allein zur Bestimmung der Phrasenstruktur eines Satzes nicht ausreicht.
2. Geben Sie eine kontextfreie Grammatik an, die Sätze der folgenden Art generiert: He likes the meal on the flight to New York on Monday.
   Zu den Regeln sollten $NP \rightarrow NP\ PP$ und $VP \rightarrow VP\ PP$ gehören.
3. Wieviele Derivationsbäume ergeben sich aus Ihrer Grammatik zu dem genannten Satz? Zeichnen Sie mindestens 5 verschiedene.
4. Welche Parsingstrategie (top-down oder bottom-up) ist für Ihre Grammatik geeignet? Begründen Sie Ihre Entscheidung und führen Sie einen Parse durch. Beachten Sie, daß Sie unter Umständen Backtracking einsetzen müssen.
5. Passen Sie die Grammatik von Folie 7 so an, daß Übergeneralisierungen vermieden werden.
6. Welcher Parsingstrategie folgen die Kellerautomaten, die nach dem in der letzten Sitzung vorgestellten Verfahren aus einer kontextfreien Grammatik gewonnen werden?

Literatur

- Carstensen et. al. (2004), Kapitel 3.4
- Jurafsky & Martin (2008), Kapitel 13