

Einführung in die Computerlinguistik

Parsing

WS 2011/2012

Manfred Pinkal

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

Beispiel: $L = a^n b^n$

- Der Automat liest zunächst nacheinander a's ein und „merkt sie sich“, indem er sie in den Speicher schreibt. Anschließend liest er nacheinander die b's und nimmt für jedes gelesene b ein a vom Speicher.
- Wenn Eingabewort und gespeicherte Folge von a's gleichzeitig aufgebraucht sind, wird das Eingabewort akzeptiert.
- Für die Verarbeitung kontextfreier Sprachen reicht es, den Speicher als Stack oder Stapel zu benutzen, auf den nach dem LastIn-FirstOut-Prinzip zugegriffen wird. Der Stack wird auch „Keller“ oder „Push-Down-Store“ genannt. Wir sprechen im Deutschen deshalb vom „Kellerautomaten“, im Englischen vom „Push-down Automaton“ (PDA).

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

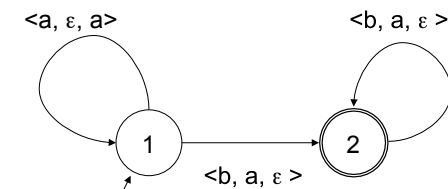
Endliches Gedächtnis

- Der endliche Automat kann nur beschränkte Information in seinen Zuständen kodieren.
- Das Gedächtnis eines endlichen Automaten mit n Zuständen reicht deshalb höchstens n-1 Zeichen zurück.
- Kontextfreie Grammatiken erzeugen aber beliebig tief geschachtelte Strukturen, in denen beliebig weit voneinander entfernte Elemente voneinander abhängen können.
- Wir können das Gedächtnisproblem durch einen zusätzlichen Speicher mit im Prinzip unbegrenzter Kapazität lösen.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

Kellerautomaten: Ein Beispiel

- Ein Kellerautomat, der $a^n b^n$ akzeptiert:



- $\langle u, v, w \rangle$ als Kanteninschrift steht für: Lies Eingabe u, lösche v vom Stack, schreibe w auf den Stack.
- Im Zustand 1 werden a's gelesen und in den Stack geschrieben.
- Beim ersten b wechselt der Automat in den Zustand 2 und löscht für jedes gelesene b ein a vom Stack.
- Wenn die Eingabe abgearbeitet, der Stack leer und ein Endzustand erreicht ist, wird das Eingabewort akzeptiert.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

Keller-Automaten und kontextfreie Grammatiken

- Kontextfreie Grammatiken (CFGs) erlauben die einfache und elegante Definition von kontextfreien Sprachen – kommen aber zunächst ohne ein sinnvolles Analyseverfahren, das die Zugehörigkeit eines Wortes w zu $L(G)$ entscheidet.
- Keller-Automaten stellen ein einfaches Verfahren zur Entscheidung von $w \in L(G)$ für bestimmte kontextfreie Sprachen zur Verfügung – aber keine intuitive Methode, um komplexe Sprachen direkt zu modellieren.
- Frage: Sind die Formalismen der CFG und des PDA gleich stark?
- Frage: Können wir eine CFG in einen äquivalenten PDA überführen?
- Die Antwort auf beide Fragen ist ja. Der Beweis erfolgt wie bei der NEA-DEA-Überführung konstruktiv, durch Spezifikation von Verfahren.

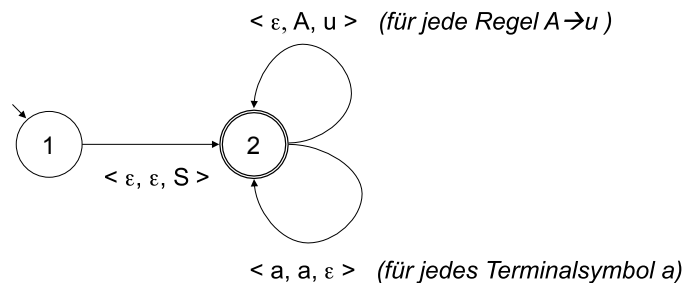
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

Keller-Automaten und kontextfreie Grammatiken

- Idee: Wir simulieren den Ableitungsprozess der CFG im Stack des PDA, und gleichen die auf dem Stack erzeugten Terminalsymbole mit der Eingabe ab.
- Die Abfolge der Regelanwendung gibt uns gleichzeitig Information über die syntaktische Struktur.
- Systeme, die für eine gegebene Grammatik und einen Eingabesatz die syntaktische Struktur bestimmen, nennen wir **Parser**.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

Ein Schema für CFG-Parsing



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

Ein Schema für CFG-Parsing

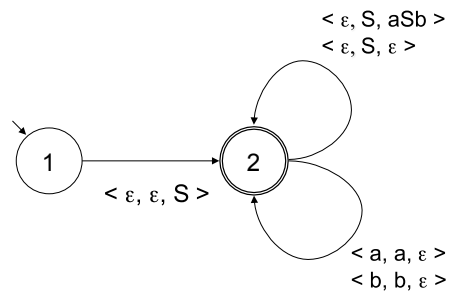
- Initialisiere den Stack des Automaten mit dem Startsymbol.
- Wenn sich das oberste Stack-Element ein nicht-termiales Symbol A ist, wähle eine Grammatikregel $A \rightarrow u$ und ersetze das Stack-Symbol A durch u .
- Wenn das oberste Stack-Element ein Terminalsymbol a und mit dem aktuelle Eingabesymbol identisch ist, lösche a vom Stack und rücke in der Eingabe vor.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

Beispiel: $a^n b^n$

$S \rightarrow aSb$

$S \rightarrow \epsilon$



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Beispiel: Kleine Grammatik fürs Deutsche

$S \rightarrow NP VP$

$NP \rightarrow Det N$

$NP \rightarrow PN$

$VP \rightarrow V NP$

$Det \rightarrow die$

$N \rightarrow Katze$

$V \rightarrow mag$

$PN \rightarrow Anna$

$\langle \epsilon, S, NP VP \rangle$

$\langle \epsilon, NP, Det N \rangle$

$\langle \epsilon, NP, PN \rangle$

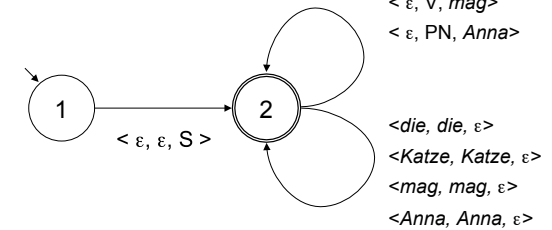
$\langle \epsilon, VP, V NP \rangle$

$\langle \epsilon, Det, die \rangle$

$\langle \epsilon, N, Katze \rangle$

$\langle \epsilon, V, mag \rangle$

$\langle \epsilon, PN, Anna \rangle$



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Kontextfreier Top-Down-Parser

- Der vorgestellte Parser erzeugt ausgehend vom Startsymbol Ableitungen und damit implizit einen Ableitungsbaum. Terminalsymbole werden von links nach rechts mit der Eingabe abgeglichen.
- Wir sprechen bei diesem Vorgehen von „Top-Down“-Parsing: Der Ableitungsbaum bzw. **Parsebaum** wird von oben nach unten, im „rekursiven Abstieg“ durch die Struktur, aufgebaut.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Kontextfreier Top-Down Parser

$S \rightarrow NP VP$

$NP \rightarrow Det N$

$NP \rightarrow PN$

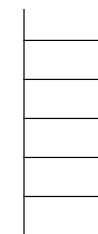
$VP \rightarrow V NP$

$Det \rightarrow die$

$N \rightarrow Katze$

$V \rightarrow mag$

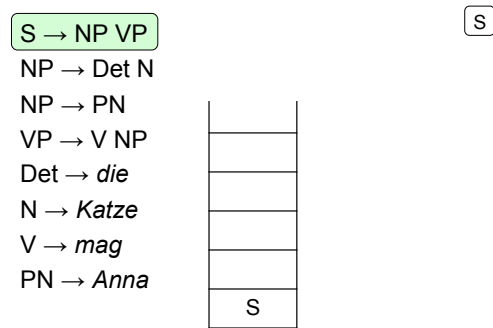
$PN \rightarrow Anna$



Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

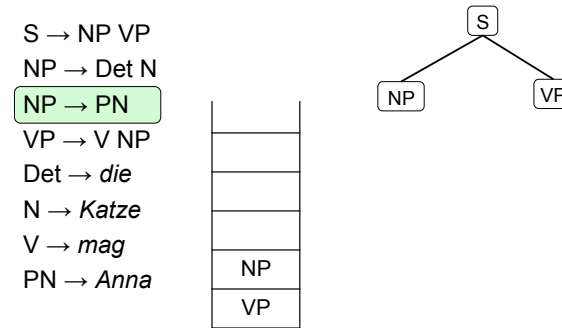
Kontextfreier Top-Down Parser



Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

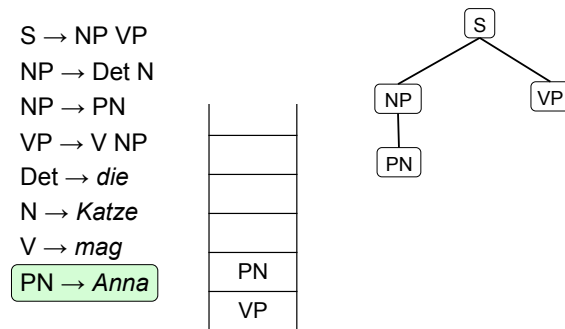
Kontextfreier Top-Down Parser



Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

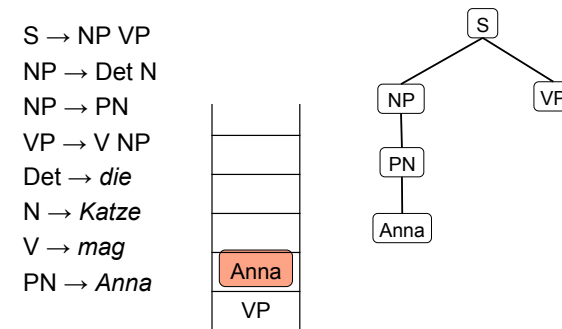
Kontextfreier Top-Down Parser



Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

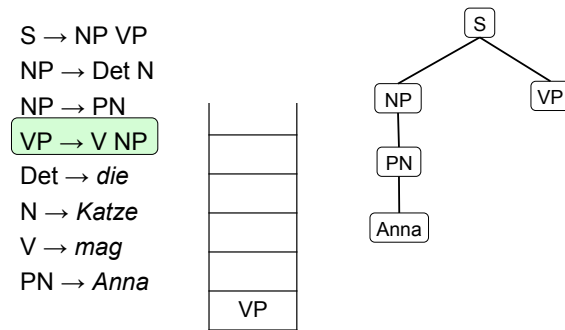
Kontextfreier Top-Down Parser



Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

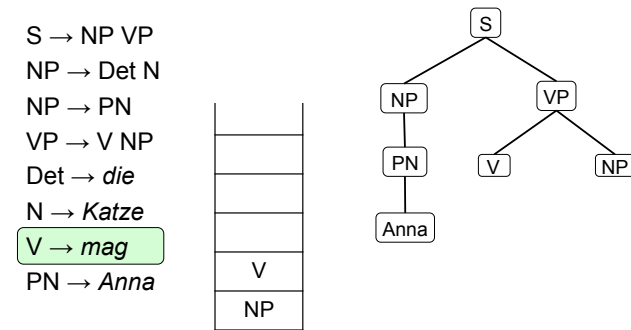
Kontextfreier Top-Down Parser



Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

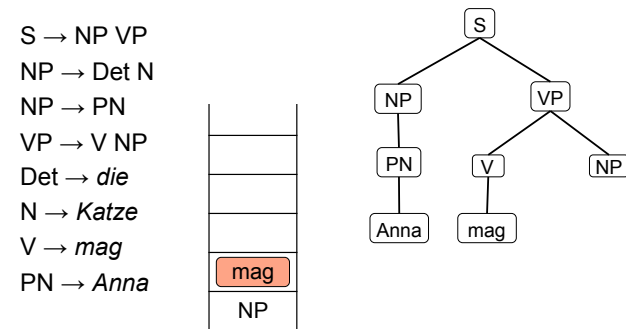
Kontextfreier Top-Down Parser



Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

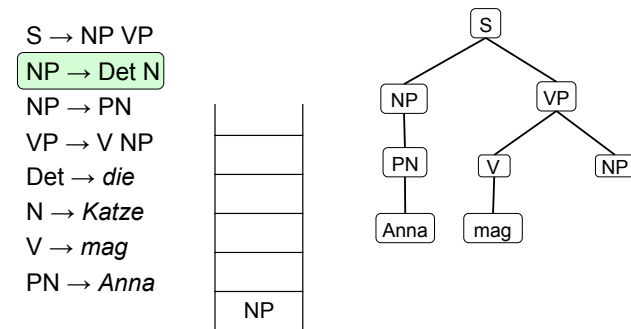
Kontextfreier Top-Down Parser



Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Kontextfreier Top-Down Parser

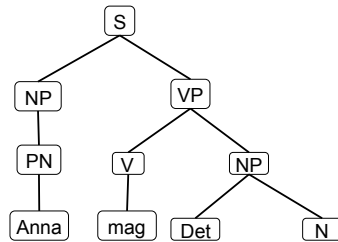
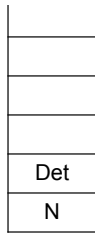


Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Kontextfreier Top-Down Parser

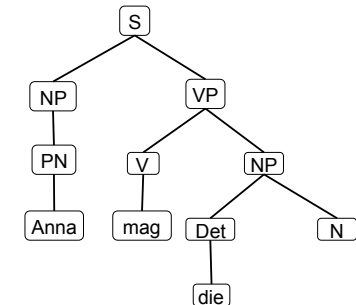
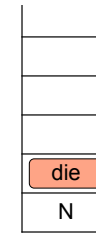
$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$



Anna mag die Katze

Kontextfreier Top-Down Parser

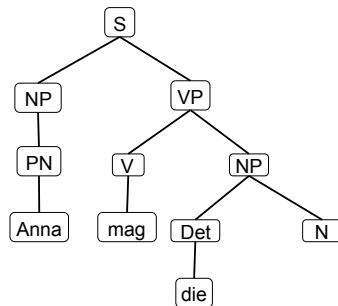
$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$



Anna mag die Katze

Kontextfreier Top-Down Parser

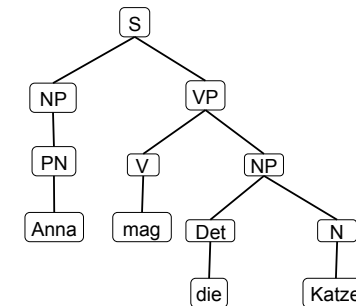
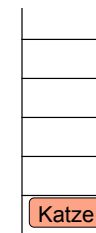
$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$



Anna mag die Katze

Kontextfreier Top-Down Parser

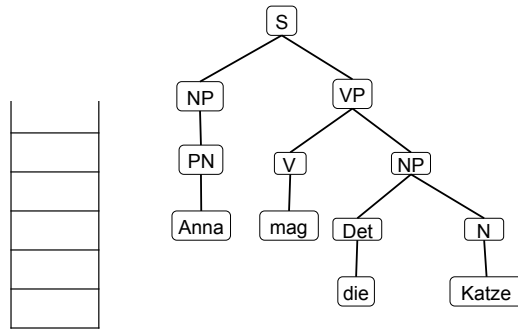
$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$



Anna mag die Katze

Kontextfreier Top-Down Parser

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

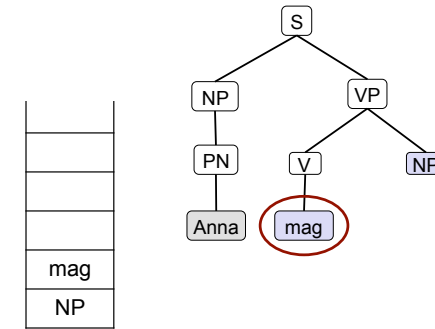


Anna mag die Katze_

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Verbrauchte, aktive, aktueller Knoten

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$



Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Kontextfreier Top-Down-Parser

- Die Operation des Top-Down-Parsers kann direkt auf Strukturbäumen dargestellt werden.
- Aktive Knoten** sind alle Blattknoten eines Baums, die noch nicht **verbraucht** sind, d.h., entweder nicht-terminale Knoten oder terminale Knoten, die noch nicht mit der Eingabe abgeglichen wurden. – Aktive Knoten entsprechen genau dem Speicherinhalt.
- Initialisierung mit S. – Der **aktuelle Knoten**, auf dem der Parser operiert, ist der am weitesten links stehende aktive Knoten.
 - Nicht-Terminalsymbol A: Wähle eine Ersetzungsregel mit linker Seite A und expandiere den aktiven Knoten entsprechend.
 - Terminalsymbol a: Gleiche mit dem aktuellen Eingabesymbol ab, markiere den Knoten im Erfolgsfall als verbraucht, und rücke in der Eingabe vor.
- Wenn die Eingabe verbraucht ist und keine aktiven Knoten mehr vorhanden sind, wird die Eingabe akzeptiert und der erzeugte Parsebaum wird als Analyse ausgegeben.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Kontextfreier Top-Down Parser

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Kontextfreier Top-Down Parser

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

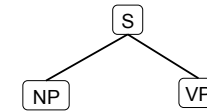
S

Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Kontextfreier Top-Down Parser

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

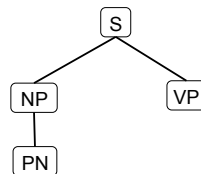


Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Kontextfreier Top-Down Parser

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

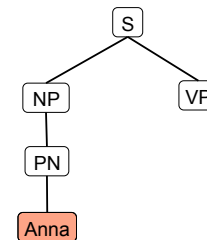


Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Kontextfreier Top-Down Parser

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

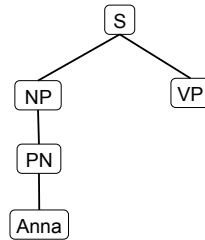


Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Kontextfreier Top-Down Parser

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

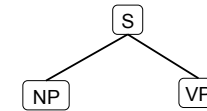


Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Nicht-Determinismus in der Regelanwendung

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

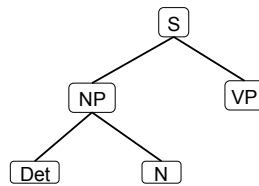


Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Nicht-Determinismus in der Regelanwendung

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

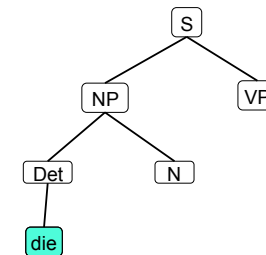


Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Nicht-Determinismus in der Regelanwendung

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$



Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Top-Down-Parser, Eigenschaften

- Der Top-Down-Parser ist im Allgemeinen nicht-deterministisch: Für dasselbe Nichtterminal gibt es mehrere, eventuell sehr viele Ersetzungsregeln.
- Eine technische Lösung: Arbeiten mit einer Agenda, Last-In – First-Out, Tiefensuche mit Backtracking.
- Problem: Es werden viele Teilstrukturen erzeugt, die nie erfolgreich sein können, weil die Eingabekette keine passenden Wörter enthält.
 - Beispiel: Grammatik versucht, Subjekts-NP abzuleiten, der Satz fängt mit einer PP an.
- Im Falle „links-rekursiver“ Grammatiken geht der Parser in eine Endlosschleife.
 - Beispiel: $VP \rightarrow VP PP$
 - Links-rekursive Regeln kann man vermeiden, aber dann sind bestimmte Strukturen nicht mehr natürlich darstellbar.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

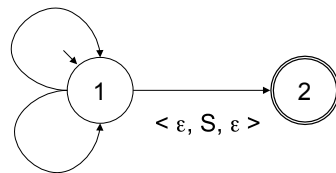
Alternative: Bottom-Up Parser

- Lies Symbole der Eingabekette und lege sie in den Stack.
- Wenn die obersten n Symbole im Stack umgekehrt gelesen die rechte Seite einer Grammatikregel $A \rightarrow u$ bilden, ersetze sie im Stack durch A .
- Wenn die Eingabe abgearbeitet ist und der Stack nur noch das Startsymbol S enthält, akzeptiere die Eingabe.
- Der Parser heißt auch „Shift-Reduce“-Parser: Aktionen bestehen darin, dass Eingabesymbole in den Stack verschoben ("shift") und dann zu Nicht-Terminalsymbolen reduziert werden ("reduce").

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Schema für den Shift-Reduce-Parser

$\langle a, \varepsilon, a \rangle$ (für jedes Terminalsymbol a)



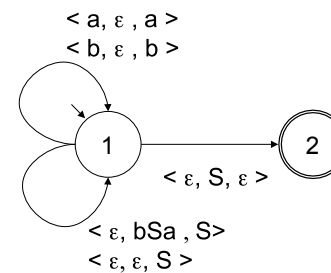
$\langle \varepsilon, u^R, A \rangle$ (für jede Regel $A \rightarrow u$)

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Beispiel: $a^n b^n$

$S \rightarrow aSb$

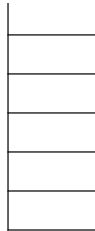
$S \rightarrow \varepsilon$



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Shift-Reduce-Parser

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

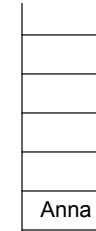


Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Shift-Reduce-Parser

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$



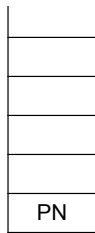
Anna

Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Shift-Reduce-Parser

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

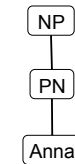
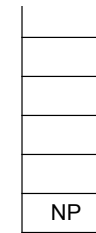


Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Shift-Reduce-Parser

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

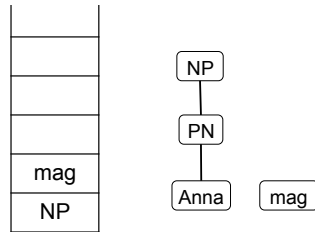


Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Shift-Reduce-Parser

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

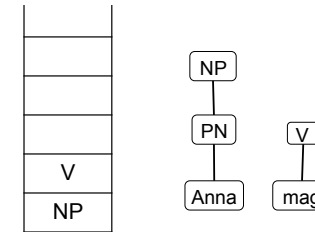


Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Shift-Reduce-Parser

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

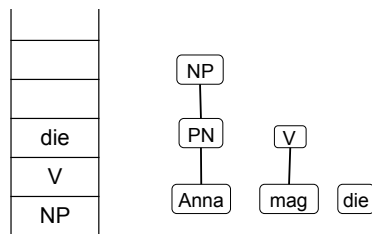


Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Shift-Reduce-Parser

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

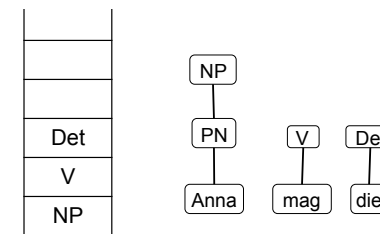


Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Shift-Reduce-Parser

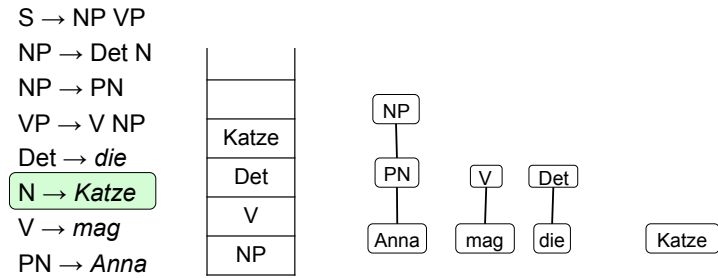
$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$



Anna mag die Katze

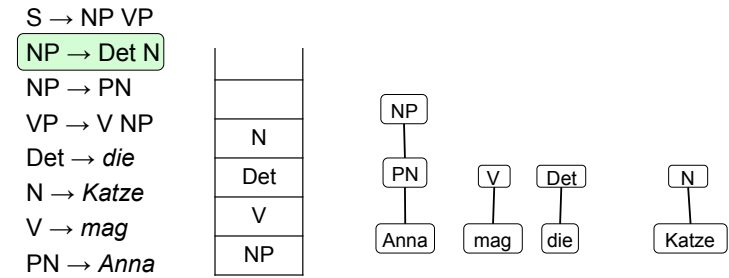
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Shift-Reduce-Parser



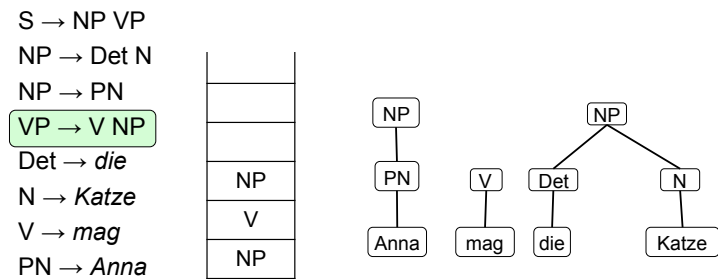
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Shift-Reduce-Parser



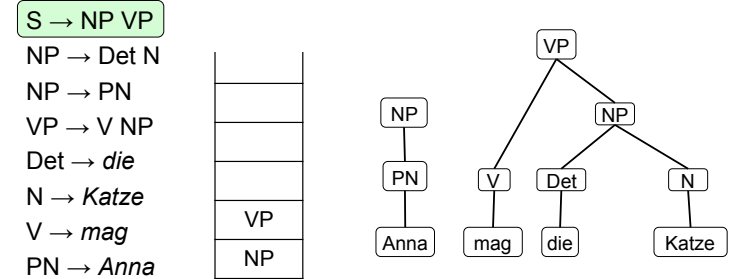
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Shift-Reduce-Parser



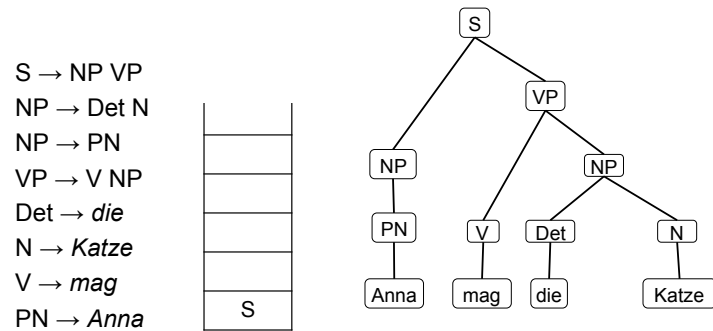
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Shift-Reduce-Parser



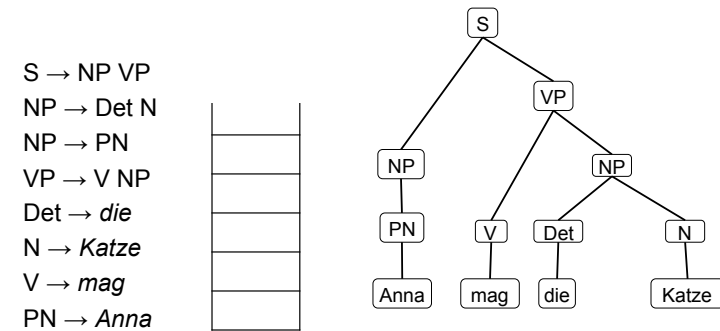
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Shift-Reduce-Parser



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Shift-Reduce-Parser



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Bottom-Up-Parser: Eigenschaften

- Auch der Bottom-Up-Parser ist nicht-deterministisch.
- Er vermeidet Analysen, die durch lexikalisches Material nicht abgedeckt sind, und ist insofern effizienter als der Top-Down-Parser.
- Er erzeugt jedoch Teilstrukturen auch dann, wenn absehbar ist, dass sie nie zu vollständigen Satzstrukturen führen.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

CFG-Parsing und Determinismus

- Top-Down- und Bottom-Up-Parser sind nicht-deterministisch – auch für die einfache Grammatik, die $a^n b^n$ erkennt.
- Der Nicht-Determinismus ist aber nicht unbedingt essenziell: $a^n b^n$ kann durch einen deterministischen PDA erkannt werden.
- Frage: Gibt es für jede kontextfreie Sprache L einen PDA, der L erkennt?
- Können Sätze natürlicher Sprachen also in linearer Zeit syntaktisch analysiert werden?
- Die Antwort ist leider Nein.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

CFG-Parsing und Determinismus

- Es gibt kontextfreie Sprachen, die deterministisch geparkt werden können, und Sprachen, bei denen das nicht möglich ist.
- Wir unterscheiden „deterministisch kontextfreie Sprachen“ und „nicht-deterministisch kontextfreie Sprachen“.
- Beispiel:
 - $L1 = \{ wcw^R \mid w \in \{a,b\}^* \}$
 - $L2 = \{ ww^R \mid w \in \{a,b\}^* \}$
 - L1 ist deterministisch, L2 nicht.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

CFG-Parsing und Determinismus

- Tendenziell sind alle interessanten formalen Sprachen („Klammersprache“, Arithmetik, Programmiersprachen) deterministisch kontextfrei.
- Natürliche Sprachen sind nicht-deterministisch.
- Sie müssen schon deswegen nicht-deterministisch sein, damit sie syntaktische Mehrdeutigkeit modellieren können.
- Es gibt allerdings Parsing-Methoden, die im Gegensatz zu Top-Down- und Bottom-Up-Parser effiziente Analyse ermöglichen.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

CFG-Parsing und Determinismus

Peter sieht den Mann mit dem Teleskop

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

CFG-Parsing und Determinismus

*Peter sieht den Mann mit dem Teleskop
durch ein Fernglas.*

*Peter sieht den an computerlinguistischen Frage-
stellungen interessierten Studenten mit dem Teleskop
durch ein Fernglas.*

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Chart-Parsing

- Das Problem: Teilstrukturen werden unnötigerweise immer wieder analysiert.
- Die Lösung: Der Parser speichert Zwischenresultate in einer Datenstruktur, auf die er im weiteren Verlauf der Verarbeitung zurückgreifen kann.
- Wir nennen die Datenstruktur „Chart“ und sprechen deshalb von „Chart-Parsing“.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Chart, graphische Darstellung

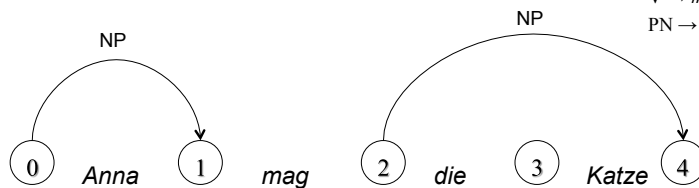
$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Chart, graphische Darstellung

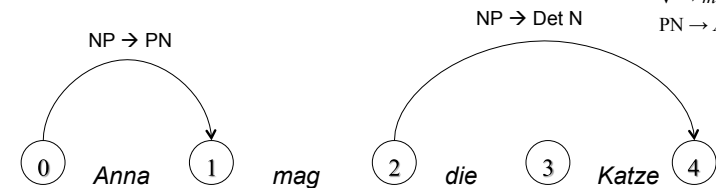
$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

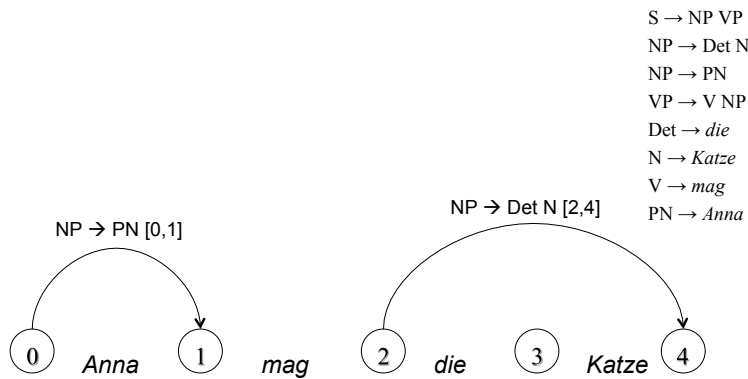
Chart, graphische Darstellung

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

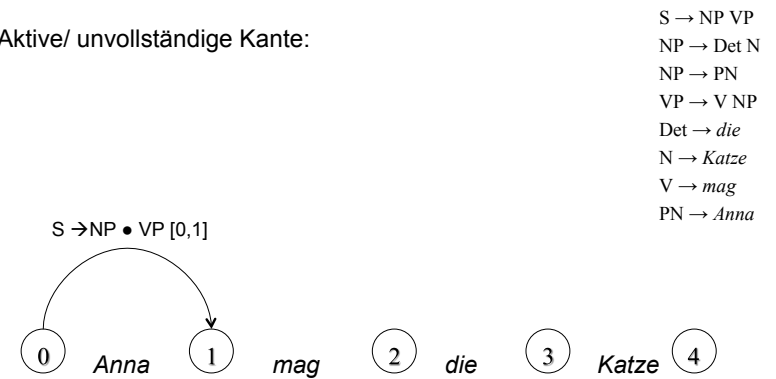
Chart, graphische Darstellung



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Chart, graphische Darstellung

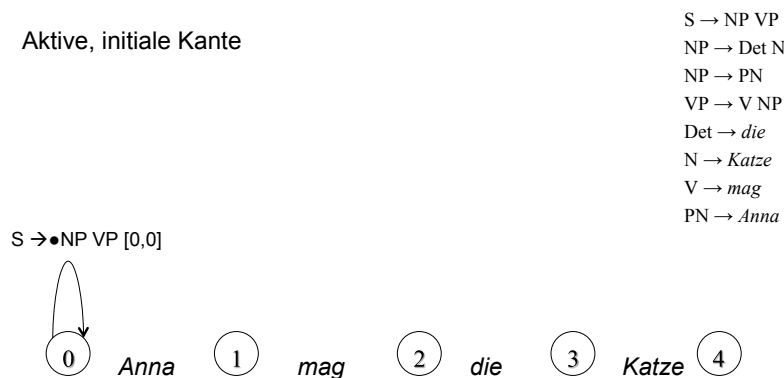
Aktive/ unvollständige Kante:



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Beispiel-Chart

Aktive, initiale Kante



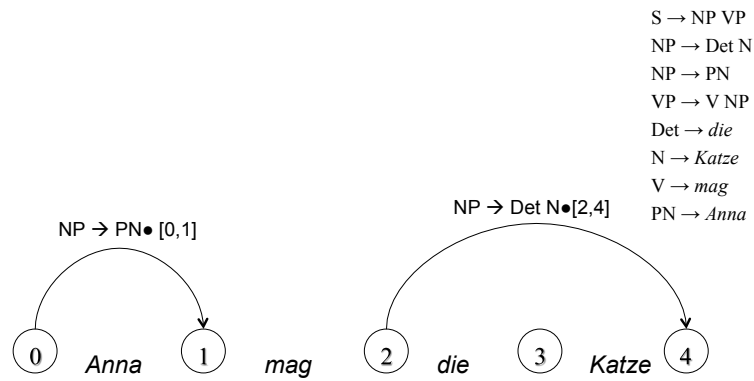
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Chart: Formale Darstellung

- Für einen Eingabesatz der Länge n wird eine Chart mit $n+1$ Spalten eingerichtet.
- Einträge bestehen aus zwei Informationen, die zusammen ein (potentiell unvollständiges) Parseresultat kodieren:
 - „Punktierte Regel“ der Form $A \rightarrow u \bullet v$: $A \rightarrow w$ mit $w=uv$ ist Regel der Grammatik, der Punkt teilt die rechte Regelseite in den abgearbeiteten und den noch offenen Teil.
 - Paar $[i, j]$ bezeichnet den Teilstring der Eingabekette, auf dem dies Ergebnis erzielt wurde.
- Beispiel: $\langle S \rightarrow NP \bullet VP, [0, 2] \rangle$
 - Regel $S \rightarrow NP VP$ wurde in Position 0 angesetzt, an Position 2 ist eine NP erkannt, von Position 2 an muss noch eine VP erkannt werden.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Chart, graphische Darstellung



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

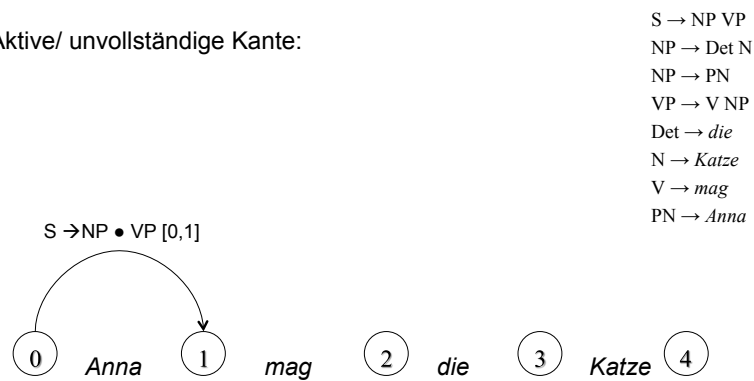
Chart: Tabellen-Darstellung

Position 0	Position 1	Position 2	Position 3	Position 4
	NP \rightarrow PN • [0,1]			NP \rightarrow Det N • [2,4]

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Chart, graphische Darstellung

Aktive/ unvollständige Kante:



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Chart: Tabellen-Darstellung

Position 0	Position 1	Position 2	Position 3	Position 4
	S \rightarrow NP • VP, [0,1]			

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Earley-Algorithmus

- Ein Standard-Algorithmus für das Chart-Parsing ist der **Earley-Algorithmus**, der von links nach rechts durch den Eingabesatz geht und in einer Top-Down-Strategie alle Analyse-Alternativen parallel verfolgt.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Earley-Algorithmus – Vorgehen

- Es seien $u, v \in V^*$: (möglicherweise leere) Ketten von (Terminal- oder Nicht-Terminal-)Symbolen; P punktierte Erzeugungsregeln.
- Initialisiere die erste Spalte der Chart mit $\langle S \rightarrow \bullet u, [0, 0] \rangle$ für jede Regel $S \rightarrow u$ der Grammatik.
- Gehe schrittweise von links nach rechts durch die Chart. In jedem Schritt j : Wende auf jeden Eintrag $\langle P, [i, j] \rangle$ die Operationen **Scan**, **Predict**, und **Complete** so lange an, bis keine neue Anwendung mehr möglich ist. Dann gehe zu $j+1$.
- Operationen des Earley-Algorithmus:
 - Wenn P unvollständig und von der Form $A \rightarrow u \bullet av$ ist (a Terminalsymbol): **Scan**
 - Wenn P unvollständig und von der Form $A \rightarrow u \bullet Bv$ ist (B Nicht-Terminalsymbol): **Predict**
 - Wenn P vollständig (von der Form $A \rightarrow u \bullet$) ist: **Complete**.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Predictor

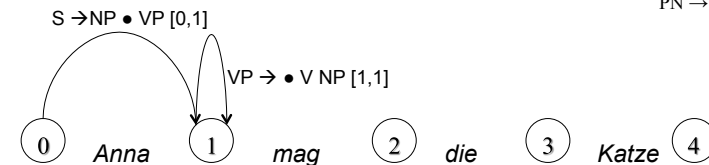
$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

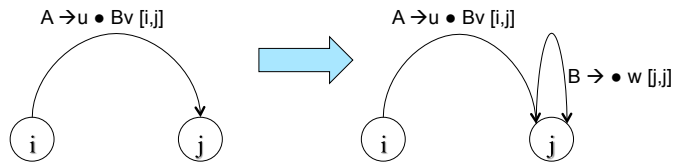
Predictor

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Predictor, Schema



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Earley-Algorithmus – Predictor

- Für Position j , Eintrag $\langle P, [i, j] \rangle$:
 - Wenn P unvollständig und von der Form $A \rightarrow u \bullet Bv$ ist (B Nicht-Terminalsymbol):
 - Füge für jede Regel $B \rightarrow w$ der Grammatik $\langle B \rightarrow \bullet w, [j, j] \rangle$ zur Position j hinzu.
- Die punktierte Regel $A \rightarrow u \bullet Bv$ im Eintrag drückt aus, dass als nächstes eine Konstituente der Kategorie B kommen könnte. Der Predictor schreibt alle Regeln in die Chart, die Ausdrücke der Kategorie B erzeugen.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Scanner: Beispiel

$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

$PN \rightarrow \bullet Anna [0,0]$

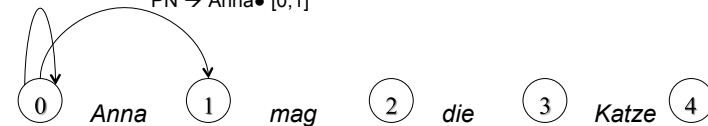


Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Scanner: Beispiel

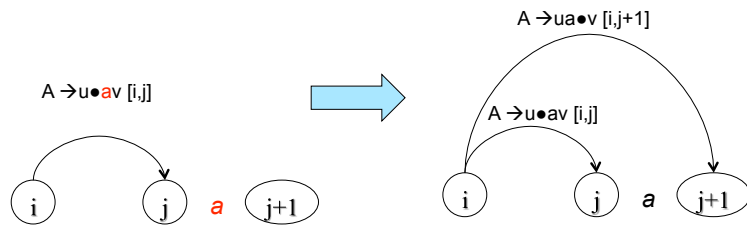
$S \rightarrow NP VP$
 $NP \rightarrow Det N$
 $NP \rightarrow PN$
 $VP \rightarrow V NP$
 $Det \rightarrow die$
 $N \rightarrow Katze$
 $V \rightarrow mag$
 $PN \rightarrow Anna$

$PN \rightarrow \bullet Anna [0,0]$ $PN \rightarrow Anna \bullet [0,1]$



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Scanner, Schema



81

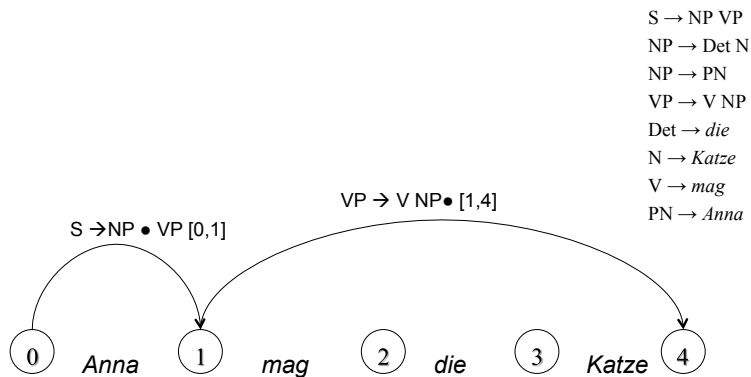
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Earley-Algorithmus – Scanner

- Für Position j , Eintrag $\langle P, [i, j] \rangle$:
 - Wenn P unvollständig und von der Form $A \rightarrow u \bullet av$ ist (a Terminalsymbol), und die Position $[j, j+1]$ der Eingabe mit a gefüllt ist:
 - Füge $\langle A \rightarrow ua \bullet v, [i, j+1] \rangle$ zur Spalte $j+1$ der Chart hinzu.
- Die punktierte Regel $A \rightarrow u \bullet av$ drückt aus, dass als nächstes das Eingabesymbol a erwartet wird. Der Scanner liest das Eingabewort und gleicht es mit der Regel ab. Im Erfolgsfall wird der Punkt über a hinweg geschoben und ein neuer Eintrag in die nächste Spalte der Chart gemacht.

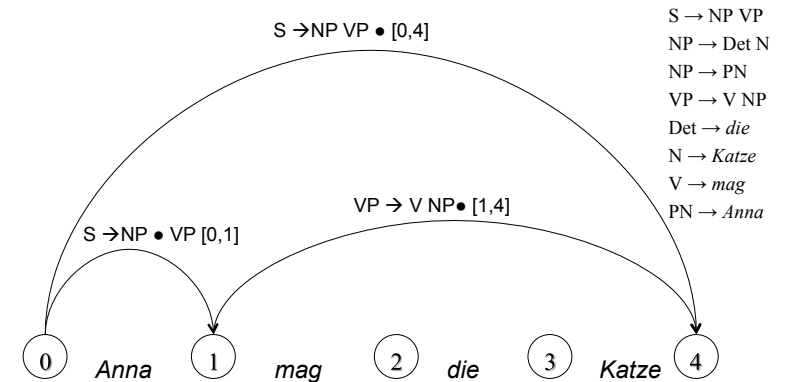
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Completer, Beispiel



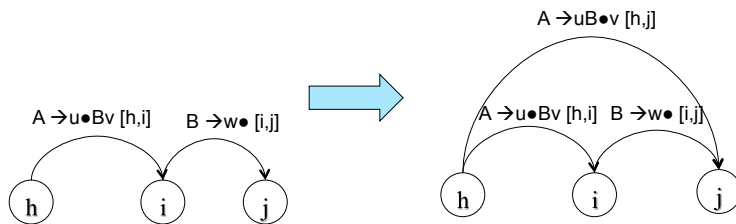
Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Complete, Beispiel



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Completer, Schema



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

Earley-Algorithmus – Completer

- Für Position j , Eintrag $\langle P, [i, j] \rangle$:
 - Wenn P vollständig und von der Form $B \rightarrow w$:
 - Füge für jeden bestehenden Eintrag $\langle A \rightarrow u \bullet Bv, [h, i] \rangle$ einen neuen Eintrag $\langle A \rightarrow uB \bullet v, [h, j] \rangle$ hinzu.
- $B \rightarrow u \bullet$ besagt, dass die rechte Seite der Regel vollständig abgearbeitet wurde. Das Resultat kann verwendet werden, um eine bereits bestehende partielle Analyse, die als nächstes einen Ausdruck der Kategorie B erwartet, zu komplettieren.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

Earley-Algorithmus: Abschluss

- Der Aufbau der Chart ist abgeschlossen, wenn der $n+1$ -te Schritt beendet ist.
- Die Eingabe wird akzeptiert, wenn die Chart $\langle S \rightarrow u \bullet, [0, n] \rangle$ enthält.
- Auf den folgenden Folien wird die Chart aufgebaut für die obige Beispielgrammatik und den Beispielsatz

Anna mag die Katze

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

Beispiel-Chart: Initialisierung

Position 0 Position 1 Position 2 Position 3 Position 4
 $S \rightarrow \bullet NP VP, [0,0]$

$S \rightarrow NP VP$
$NP \rightarrow Det N$
$NP \rightarrow PN$
$VP \rightarrow V NP$
$Det \rightarrow die$
$N \rightarrow Katze$
$V \rightarrow mag$
$PN \rightarrow Anna$

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UoS Computerlinguistik

Beispiel-Chart: Operationen auf Pos. 0

Position 0	Position 1	Position 2	Position 3	Position 4
S → •NP VP, [0,0]	Scanner:			
Predictor: NP → •Det N, [0,0]	PN → Anna•, [0,1]			
NP → •PN, [0,0]				
Det → •die, [0,0]				
PN → •Anna, [0,0]				

S → NP VP
 NP → Det N
 NP → PN
 VP → V NP
 Det → die
 N → Katze
 V → mag
 PN → Anna

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Beispiel-Chart: Operationen auf Pos. 1

Position 0	Position 1	Position 2	Position 3	Position 4
S → •NP VP, [0,0]	Scanner:	Scanner:		
Predictor: NP → •Det N, [0,0]	PN → Anna•, [0,1]	V → mag•, [1,2]		
NP → •PN, [0,0]	Completer:			
Det → •die, [0,0]	NP → PN•, [0,1]			
PN → •Anna, [0,0]	S → NP•VP, [0,1]			

Predictor:
 VP → •V NP, [1,1]
 V → •mag, [1,1]

S → NP VP
 NP → Det N
 NP → PN
 VP → V NP
 Det → die
 N → Katze
 V → mag
 PN → Anna

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Beispiel-Chart: Operationen auf Pos. 2

Position 0	Position 1	Position 2	Position 3	Position 4
S → •NP VP, [0,0]	Scanner:	Scanner:	Scanner:	
Predictor: NP → •Det N, [0,0]	PN → Anna•, [0,1]	V → mag•, [1,2]	Det → die•, [2,3]	
NP → •PN, [0,0]	Completer:	Completer:		
Det → •die, [0,0]	NP → PN•, [0,1]	VP → V•NP, [1,2]		
PN → •Anna, [0,0]	S → NP•VP, [0,1]	Predictor:		

Predictor:
 NP → •Det N, [2,2]
 VP → •V NP, [1,1]
 V → •mag, [1,1]

S → NP VP
 NP → Det N
 NP → PN
 VP → V NP
 Det → die
 N → Katze
 V → mag
 PN → Anna

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Beispiel-Chart: Operationen auf Pos. 4

Position 0	Position 1	Position 2	Position 3	Position 4
S → •NP VP, [0,0]	Scanner:	Scanner:	Scanner:	Scanner:
Predictor: NP → •Det N, [0,0]	PN → Anna•, [0,1]	V → mag•, [1,2]	Det → die•, [2,3]	N → Katze•, [3,4]
NP → •PN, [0,0]	Completer:	Completer:	Completer:	Completer:
Det → •die, [0,0]	NP → PN•, [0,1]	VP → V•NP, [1,2]	NP → Det•N, [2,3]	NP → Det N•, [2,4]
PN → •Anna, [0,0]	S → NP•VP, [0,1]	Predictor:	Predictor:	VP → V NP•, [1,4]

Predictor:
 NP → •Det N, [2,2]
 VP → •V NP, [1,1]
 V → •mag, [1,1]

S → NP VP
 NP → Det N
 NP → PN
 VP → V NP
 Det → die
 N → Katze
 V → mag
 PN → Anna

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Beispiel-Chart: Operationen auf Pos. 4

Position 0	Position 1	Position 2	Position 3	Position 4
S → •NP VP, [0,0]	Scanner:	Scanner:	Scanner:	Scanner:
Predictor:	PN → Anna•, [0,1]	V → mag•, [1,2]	Det → die•, [2,3]	N → Katze•, [3,4]
NP → •Det N, [0,0]	Completer:	Completer:	Completer:	Completer:
NP → •PN, [0,0]	NP → PN•, [0,1]	VP → V•NP, [1,2]	NP → Det•N, [2,3]	NP → Det N•, [2,4]
Det → •die, [0,0]	S → NP•VP, [0,1]	Predictor:	Predictor:	VP → V NP•, [1,4]
PN → •Anna, [0,0]	Predictor:	NP → •Det N, [2,2]	N → •Katze, [3,3]	S → NP VP•, [0,4]
	VP → •V NP, [1,1]	NP → •PN, [2,2]		
	V → •mag, [1,1]	Det → •die, [2,2]		
		PN → •Anna, [2,2]		

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Earley-Algorithmus

- Der Eintrag $S \rightarrow NP VP \bullet$, [0, 4] zeigt, dass die analysierte Wortkette ein in unserer Beispielgrammatik grammatischer Satz ist.
- Wenn wir die vollständigen Regeleinträge der Chart so miteinander verlinken, dass ein Eintrag auf die Einträge verweist, auf denen er aufbaut, können wir außerdem die syntaktische Struktur / den Parsebaum ablesen.

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Beispiel-Chart

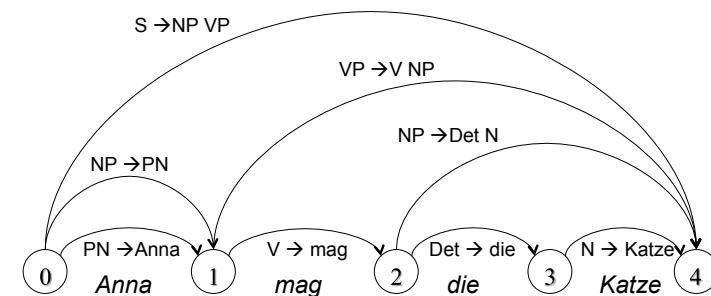
- Auf vollständige Einträge beschränkt, Abhängigkeiten sind markiert

Position 0	Position 1	Position 2	Position 3	Position 4
	PN → Anna•, [0,1]	V → mag•, [1,2]	Det → die•, [2,3]	N → Katze•, [3,4]
	NP → PN•, [0,1]			NP → Det N•, [2,4]
				VP → V NP•, [1,4]
				S → NP VP•, [0,4]

Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik

Beispiel-Chart

- Alternative grafische Repräsentation der vollständigen Einträge



Vorlesung "Einführung in die CL" 2011/2012 © M. Pinkal UdS Computerlinguistik