Syntaktische Analyse

- Satzbildung durch Wortkomposition
- Ausgangspunkt für Bedeutungsanalyse
  (1) *Der Hund fraß den Knochen*
  (2) *Der Knochen wurde vom Hund gefressen*

- Struktur aus Syntaxregeln hilft der Bedeutungsfindung im Gegensatz zu:
  – Es ist immer das 2. Substantiv, das gefressen wird.
- Zerlegung der Bedeutungsfindung auf Basis von syntaktischen Zerlegungen (etwa von Phrasen)
  (3) *sp[Der Hase mit den langen Ohren] erfreute sich an* sp[*einem großen grünen Salatblatt*]

- Bedeutungsfindung durch Komposition der Teilbedeutungen
- Komposition abhängig von gewählter Clusterung
  (4) *Tim sah Maria mit dem Fernglas*
  (5) *Ich sah den Kölner Dom auf dem Flug nach Frankfurt*

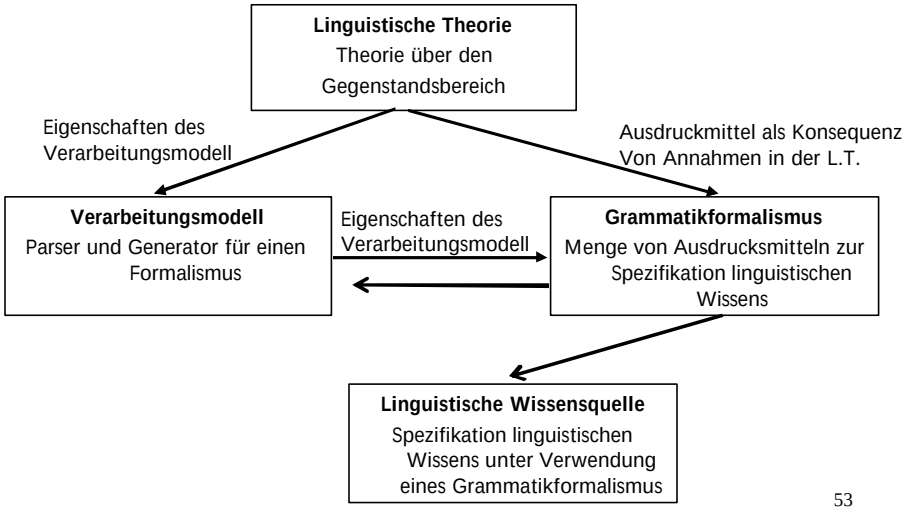
Beschreibungsformalismus

Definition: Ein Grammatikformalismus ist ein Formalismus zur Repräsentation sprachlichen Wissens auf den verschiedenen linguistischen Beschreibungsebenen.

- Kriterien zur Ausgestaltung eines Grammatikformalismus unter Berücksichtigung der Beziehung von Grammatikformalismus zu linguistischer Theorie:
  – **Ausdrucksstärke:** Welche Phänomenbereiche können beschrieben werden?
  – **Algorithmisierbarkeit:** Welche Verarbeitungsmodelle werden unterstützt? Welche Algorithmisierbarkeit ist möglich?
  – **Linguistische Motiviertheit:** Umfang direkt zu verwendender linguistisch abgesicherter Ausdrucksmittel? Inwieweit sind Ausdruckmittel direkt an spezifische linguistische Theorie gebunden?

University of Bielefeld

Beschreibungsformalismus



University of Bielefeld

Grammatik

- Grammatik als formales Beschreibungssystem für Syntaxregeln zur Definition der Wortorganisation
- Menge von Ersetzungsregeln für eine Kette von Symbolen (=Satz)
- Symbole eingeordnet in syntaktische Kategorien
  - Nomen, Verb, Adjektiv, …
  - Nominalphrase, Verbalphrase
- Satz ist regulär, wenn ausgehend von einem Startsymbol der Satz mit Hilfe der Regeln erzeugt werden kann (Parsen)
- Was soll die Grammatik tun?
  - Unterscheidung zwischen „korrekten“ und „inkorrekten“ Sätzen
    - Der Junge wirft den Ball. (+)
    - Der wirft Junge Ball den. (-)
  - Zuweisung einer „sinnvollen“ Struktur
    - (Der Junge) (wirft den Ball)
    - (Der) (Junge wirft) (den Ball)
- Eigenschaften der Grammatik: Kompakt, modular
- Syntaxanalysierer (Parser) überprüft Satz auf Grammatikkonformität

University of Bielefeld

Grammatik

- Linguistische Strukturen und Prozesse
  - Kompositionalität
    - Interpretation einer Struktur lässt sich aus Teilen und ihrer Zusammensetzung ableiten.
    - Interpretation jedes Teils kann lokal erfolgen.
    - Bedingt analytische (paradigmatische Betrachtung).
  - Deklarativität
    - Trennung zwischen Wissen und Verarbeitung.
    - Verarbeitung kann nach rein strukturellen Gesichtspunkten erfolgen
    - Unterstützung von analytischen und generativen Prozessen.
- Konkatenation
  - Lineare Aneinanderreihung einzelner Elemente.
  - Häufigste Operation über linguistischen Daten.
  - Bei natürlichen Sprachen finden sich auch Tilgung und Ersetzung.

University of Bielefeld

Chomskys Hierarchie der Grammatiken

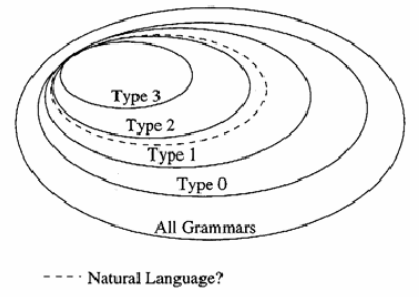
Regeln zum Ausdruck der korrekten grammatikalischen Struktur einer Sprache für Analyse und Synthese
Format der hierarchischen Chomsky Grammatiken

A B C … -> E F G …

Typ 0: Transformation Grammar
Regeln der Form: *beliebig -> beliebig*
Mächtigkeit: allgemeine Turing Maschine

Typ 1: kontextsensitive Grammatik
Regeln der Form : *A B C -> A D C*
Mächtigkeit : Linear Bound Automata

Typ 2: kontextfreie Grammatik
Regeln der Form: *A -> B C D...*
Mächtigkeit : Push Down Stack Automata



Typ 3: Regulär or Rechtslinear
Regeln der Form : *A -> x B* und *A -> x* wobei x terminal
Mächtigkeit : Finite State Automata

University of Bielefeld

Grammatikformalismen

Phrasenstrukturgrammatiken Beispiel:

(7) NP -> Det Adj N

- (7) beschreibt einfache Sätze wie
  - „der alte Mann“
  - „ein junger Mann“
  - „dem großen Haus“
- Menge von Regeln wie (7) bilden eine Phrasenstrukturgrammatik (PSG).
- PSG sind kontextfreie Grammatiken.

PSG:

Grammatik	$G = (\Sigma, \Phi, P, S)$
Terminale	$\Sigma = \{alte, dem, der, ein, großen, Haus, junger, Mann, Vater, \ldots\}$
Nonterminale	$\Phi = \{Adj, Det, N, NP, \ldots\}$
Regeln	$P = \left\{ \begin{array}{l} S \rightarrow NP\ VP \mid \ldots, NP \rightarrow Det\ N \mid Det\ Adj\ N \mid \ldots \\ Det \rightarrow dem \mid der \mid ein \mid \ldots, Adj \rightarrow alte \mid junger \mid \ldots, \\ N \rightarrow Haus \mid Mann \mid Vater \mid \ldots \end{array} \right\}$
Startsymbol	S

University of Bielefeld

Beispiel für eine einfache deutsche Grammatik

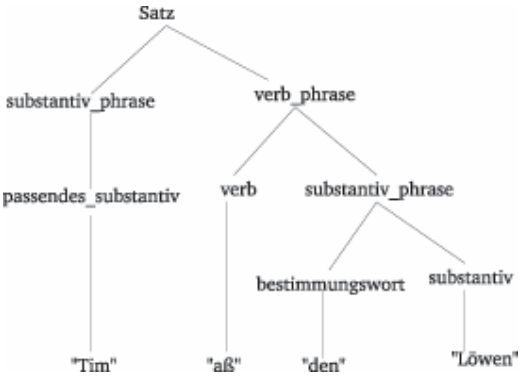
Grammatik in DCG (definite clause grammar oder definiten Klauselgrammatik):

```
satz                --> substantiv_phrase, verb_phrase
substantiv_phrase   --> passendes_substantiv
substantiv_phrase   --> bestimmungswort, substantiv
verb_phrase         --> verb, substantiv_phrase
passendes_substantiv --> [Maria]
passendes_substantiv --> [Tim]
substantiv          --> [Löwen]
substantiv          --> [Keks]
verb                --> [aß]
verb                --> [küsste]
bestimmungswort     --> [den]
```

University of Bielefeld

Ableitungsbaum

Ableitungsbaum für „*Tim aß den Löwen*“



University of Bielefeld

Grammatikformalismen

Eine einfache Grammatik mit zahlenmäßiger Übereinstimmung zwischen Subjekt und Verb:

```
satz                    -->  substantiv_phrase(Num),
                                        verb_phrase(Num)
substantiv_phrase(Num)  -->  passendes_substantiv(Num)
substantiv_phrase(Num)  -->  bestimmungswort (Num),
                                        substantiv(Num)
verb_phrase(Num)        -->  verb(Num), substantiv_phrase (_)
passendes_substantiv(s) -->  [maria]
substantiv(s)           -->  [löwe]
substantiv(p)           -->  [löwen]
best(s)                 -->  [den]
best(p)                 -->  [die]
verb(s)                 -->  [isst]
verb(p)                 -->  [essen]
```

University of Bielefeld

Grammatikformalismen

- Natürliche Sprache hat großes Vokabular.
- Syntaktisch sich gleich verhaltende Wörter werden zu prälexikalischen Kategorien (Präterminale) zusammengefasst.
- Zuordnung von Wörtern zu Kategorien durch ein Lexikon.

(8) NP -> N | Det N | AP N | Det AP N
AP -> Adj | Adj AP

(9) NP -> (Det) Adj* N

- Ausdrucksfähigkeit wird nicht erhöht (schwache Äquivalenz zur PSG ohne Lexikon, Rekursion (8) wird durch Iteration (9) ersetzt.)

PSG mit Lexikon:

Grammatik	$G' = (\Sigma', \Phi', P', S')$
Präterminale	$\Sigma' = \{Det, Adj, N, \ldots\}$
Nonterminale	$\Phi' = \{S, VP, NP, \ldots\}$
Regeln	$P' = \{S \rightarrow NP\ VP \mid \ldots, NP \rightarrow Det\ N \mid Det\ Adj\ N \mid \ldots\}$
Startsymbol	S
Lexikon	$L = \begin{Bmatrix} dem[Det], der[Det], ein[Det], alte[Adj], großen[Adj], \\ junger[Adj], Haus[N], Mann[N], Vater[N], \ldots \end{Bmatrix}$

University of Bielefeld

Grammatikformalismen

- Regel (7) NP -> Det Adj N enthält zwei Arten von Informationen:
  - Dominazinformation: NP besteht aus Det, Adj und N.
  - Präzedenzinformation: Det kommt vor Adj, Adj vor N.
- Dadurch erhält man zwei disjunkte Mengen Beschreibungsregeln:
  - Dominanzregeln: Aus welchen Subphrasen bestehen Phrasen.
  - Präzedenzregeln: Partielle Ordnung der Phrasen; Reihenfolge der Phrasen, die innerhalb der Dominanzregeln nicht verletzt werden dürfen.

➢ Diese Art der Grammatik nennt sich ID/LP-Grammatik [Gazdar und Pullum, 1981;Gazdar et al., 1985] (immediate dominance/linear-precedence)

ID/LP-Grammatik:

Grammatik	$G = (\Sigma', \Phi', D, P, S)$
Präterminale	$\Sigma' = \{Det, Adj, N, \ldots\}$
Nonterminale	$\Phi' = \{S, VP, NP, \ldots\}$
Dominanzregeln	$D = \{S \rightarrow NP, VP \mid \ldots; NP \rightarrow Det, N \mid Det, Adj, N; \ldots\}$
Präzendenz regeln	$P = \{NP < VP, Det < Adj, Det < N, Adj < N, \ldots\}$
Startsymbol	S
Lexikon	$L = \begin{Bmatrix} dem[Det], der[Det], ein[Det], alte[Adj], großen[Adj], \\ junger[Adj], Haus[N], Mann[N], Vater[N], \ldots \end{Bmatrix}$

University of Bielefeld

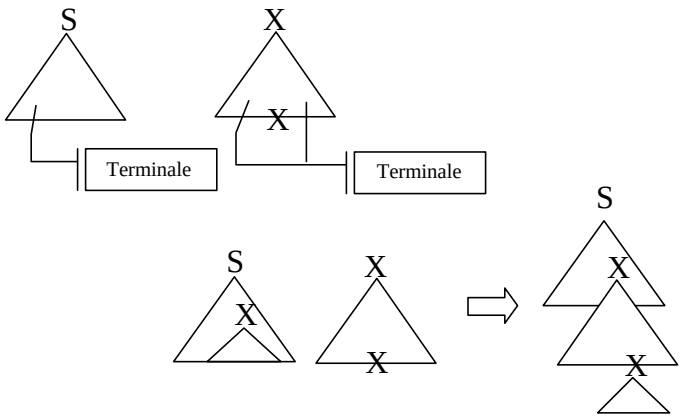
Grammatikformalismen

- Baumadjunktionsgrammatiken (Tree Adjoining Grammar – TAG) [Joshi, 1985]
  - Konkatenation als zweidimensionale Baumstruktur.
  - Statt Produktionsregeln gibt es zwei Baumarten:
    - Initialbäume: Wurzel Startsymbol S und Blätter ausschließlich Terminalsymbole
    - Auxiliarbäume: Blätter sind beliebige Anzahl von Terminalsymbolen und genau ein Nichtterminal, dieses ist das gleiche wie die Wurzel des AB.
    - Adjunktion als Standardoperation: AB kann einen beliebigen Teilbaum ersetzen wenn er die Wurzeln matchen
    - Jeder Initialbaum repräsentiert einen gültigen Satz der Sprache.
    - Adjunktion erzeugt ebenfalls gültige Sätze.

University of Bielefeld

Grammatikformalismen

- Baumadjunktionsgrammatiken (TAG)



University of Bielefeld

Grammatikformalismen

- **Baumadjunktionsgrammatiken (TAG):**

G = (I, A)

I = Menge der Initialbäume

A = Menge der Auxiliarbäume

Initialbäume sind Bäume, deren Wurzel das Startsymbol und deren Blätter alle Terminalsymbole sind.

Die Blätter der **Auxiliarbäume** sind Terminalsymbole bis auf genau ein Nichtterminal, das mit dem Wurzelknoten gleich ist.

Mit Hilfe der **Adjunktion** können Bäume zu komplexeren Bäumen zusammengesetzt werden.

- Mit TAGs können alle kontextfreien Sprachen dargestellt werden.
- TAGs können aber Bäume zuweisen, welche durch eine PSG nicht erzeugbar wären (schwache Äquivalenz).
- TAGs können aber so genannte mild kontextabhängige Sprachen akzeptieren.

University of Bielefeld

Probleme mit PSG

- Unzureichende schwache generative Kapazität:
  - Natürliche Sprachen enthalten kontextsensitive Konstruktion. Zumindest scheinen einige Sprachen nicht kontextfreie Konstruktionen zu erlauben (-> TAGs).
  - Kontextfreie Grammatik wird immer übergenerieren.
- Unzureichende starke generative Kapazität:
  - Manche linguistische Daten sind mit PS-Regeln schwer darzustellen (Fernabhängigkeit, Übereinstimmung, …)
  - Durch PS-Grammatik erzeugbare Strukturbeschreibungen sind linguistisch nicht adäquat (s. auch ID/LP und PSG).

University of Bielefeld

Probleme mit PSG

Übereinstimmung (agreement) am Beispiel (7) NP -> Det Adj N:

- (7) beschreibt einfache Sätze wie
  - „der alte Mann“
  - „ein junger Mann“
  - „dem großen Haus“
- Aber auch
  - Der alter Vater
  - Ein junges Mann
  - Dem großen Häusern
- Regel (7) generiert über!
- Deutsch erfordert Übereinstimmung von Kasus, Genus und Numerus innerhalb einer NP. Adjektiva flektieren abhängig vom Determiner stark oder schwach (Paradigma).
- Dennoch können deutsche NPs kontextfrei beschrieben werden! Aufspaltung der Nichtterminalen (nach den agreement-Merkmalen) und Ersetzung von (7) durch Menge neuer Regeln.
  - Große Menge an Regeln.
  - Generalisierung geht verloren.

University of Bielefeld

Probleme mit PSG

- Annotation als Einschränkungsmöglichkeit:

Regel: NP -> Det Adj N

Test:
$$Num_{Det} \cap Num_{Adj} \cap Num_{N} \neq \{\}$$
$$Genus_{Det} \cap Genus_{Adj} \cap Genus_{N} \neq \{\}$$
$$Casus_{Det} \cap Casus_{Adj} \cap Casus_{N} \neq \{\}$$
$$Parad_{Det} \cap Parad_{Adj} \neq \{\}$$

Aktionen:
$$Num_{NP} := Num_{Det} \cap Num_{Adj} \cap Num_{N}$$
$$Genus_{NP} := Genus_{Det} \cap Genus_{Adj} \cap Genus_{N}$$
$$Casus_{NP} := Casus_{Det} \cap Casus_{Adj} \cap Casus_{N}$$

- Registerbelegung folgt aus Lexikoneinträgen:

Der: Num=Sing, Genus=mask, Casus=Nom, Parad.=stark
Num=Sing, Genus=fem, Casus=Gen, Parad.=stark
…

- Lexikonzugriff initialisiert die Register der prälexikalischen Kategorien.
- Aktionen propagieren die Werte zu den Nichtterminalen.

University of Bielefeld

Spezielle Themen der KI

NLP
Natural Language Processing
Parsing

University of Bielefeld

Parsing

Beispiellexikon und -grammatik für eine Sprache E0:

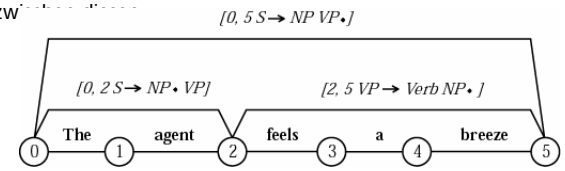
$$
\begin{array}{rcl}
\textit{Noun} & \rightarrow & \textbf{\textit{stench}} \mid \textbf{\textit{breeze}} \mid \textbf{\textit{glitter}} \mid \textbf{\textit{nothing}} \\
 & & \mid \textbf{\textit{wumpus}} \mid \textbf{\textit{pit}} \mid \textbf{\textit{pits}} \mid \textbf{\textit{gold}} \mid \textbf{\textit{east}} \mid \ldots \\
\textit{Verb} & \rightarrow & \textbf{\textit{is}} \mid \textbf{\textit{see}} \mid \textbf{\textit{smell}} \mid \textbf{\textit{shoot}} \mid \textbf{\textit{feel}} \mid \textbf{\textit{stinks}} \\
 & & \mid \textbf{\textit{go}} \mid \textbf{\textit{grab}} \mid \textbf{\textit{carry}} \mid \textbf{\textit{kill}} \mid \textbf{\textit{turn}} \mid \ldots \\
\textit{Adjective} & \rightarrow & \textbf{\textit{right}} \mid \textbf{\textit{left}} \mid \textbf{\textit{east}} \mid \textbf{\textit{south}} \mid \textbf{\textit{back}} \mid \textbf{\textit{smelly}} \mid \ldots \\
\textit{Adverb} & \rightarrow & \textbf{\textit{here}} \mid \textbf{\textit{there}} \mid \textbf{\textit{nearby}} \mid \textbf{\textit{ahead}} \\
 & & \mid \textbf{\textit{right}} \mid \textbf{\textit{left}} \mid \textbf{\textit{east}} \mid \textbf{\textit{south}} \mid \textbf{\textit{back}} \mid \ldots \\
\textit{Pronoun} & \rightarrow & \textbf{\textit{me}} \mid \textbf{\textit{you}} \mid \textbf{\textit{I}} \mid \textbf{\textit{it}} \mid \textbf{\textit{S/HE}} \mid \textbf{\textit{Y'ALL}} \ldots \\
\textit{Name} & \rightarrow & \textbf{\textit{John}} \mid \textbf{\textit{Mary}} \mid \textbf{\textit{Boston}} \mid \textbf{\textit{UCB}} \mid \textbf{\textit{PAJC}} \mid \ldots \\
\textit{Article} & \rightarrow & \textbf{\textit{the}} \mid \textbf{\textit{a}} \mid \textbf{\textit{an}} \mid \ldots \\
\textit{Preposition} & \rightarrow & \textbf{\textit{to}} \mid \textbf{\textit{in}} \mid \textbf{\textit{on}} \mid \textbf{\textit{near}} \mid \ldots \\
\textit{Conjunction} & \rightarrow & \textbf{\textit{and}} \mid \textbf{\textit{or}} \mid \textbf{\textit{but}} \mid \ldots \\
\textit{Digit} & \rightarrow & \textbf{0} \mid \textbf{1} \mid \textbf{2} \mid \textbf{3} \mid \textbf{4} \mid \textbf{5} \mid \textbf{6} \mid \textbf{7} \mid \textbf{8} \mid \textbf{9}
\end{array}
$$

S	$\rightarrow$	$NP\ VP$	I + feel a breeze
	$\mid$	$S\ Conjunction\ S$	I feel a breeze + and + I smell a wumpus
NP	$\rightarrow$	$Pronoun$	I
	$\mid$	$Noun$	pits
	$\mid$	$Article\ Noun$	the + wumpus
	$\mid$	$Digit\ Digit$	3 4
	$\mid$	$NP\ PP$	the wumpus + to the east
	$\mid$	$NP\ RelClause$	the wumpus + that is smelly
VP	$\rightarrow$	$Verb$	stinks
	$\mid$	$VP\ NP$	feel + a breeze
	$\mid$	$VP\ Adjective$	is + smelly
	$\mid$	$VP\ PP$	turn + to the east
	$\mid$	$VP\ Adverb$	go + ahead
PP	$\rightarrow$	$Preposition\ NP$	to + the east
$RelClause$	$\rightarrow$	$\textbf{\textit{that}}\ VP$	that + is smelly

University of Bielefeld

Parsing

- Parsing von links nach rechts kann backtracking erfordern:
  - „Have the students in section 4 of course 201 take the exam.“
  - „Have the students in section 4 of course 201 taken the exam?“
- Backtracking mit Reanalyse ist kostenintensiv.
- ➢ Resultate müssen zwischengespeichert werden
  - Im Beispiel kann die gelungene Analyse der NP „the students in section 4 of course 201“ in einen Zwischenspeicher - **chart** - abgelegt werden.
  - Parsing Algorithmen mit charts werden **Chartparser** genannt.
  - Alle Phrasen im Kontext eines Zweiges können auch in einem anderen Zweig des Suchraums verwendet werden.
  - Ein chart für einen n-Wort Satz besteht aus n+1 Knoten und Kanten zwischen diesen



University of Bielefeld

Chartparser

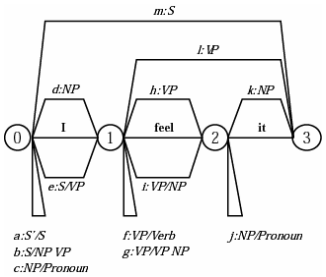
- Chartparser vereinen top-down und bottom-up:
  - Scanner (bottom-up) startet von den Worten aus und benutzt das Wort nur, um einen existierenden Kanteneintrag zu ergänzen.
  - Predictor (top-down) trägt neue Kanten in chart ein und bestimmt welche Symbole an welchen Positionen erfordert werden.
  - Completer (bottom-up) arbeitet wie Scanner und baut Konstituenten zur Ergänzung vorhandener Kanten auf.

```
function CHART-PARSE(string, grammar) returns chart

  chart[0] ← {[0, 0, S′ → • S]}
  for v ← from 1 to LENGTH(string) do
    SCANNER(v, string[v])
  end
  return chart

  procedure ADD-EDGE(edge)
      if edge in chart[END(edge)] then do nothing
      else
          push edge on chart[END(edge)]
          if COMPLETE?(edge) then COMPLETER(edge)
          else PREDICTOR(edge)

  procedure SCANNER(j, word)
      for each [i, j, A → α • B β] in chart[j] do
          if word is of category B then
              ADD-EDGE([i, j+1, A → α B • β])
      end

  procedure PREDICTOR([i, j, A → α • B β])
      for each (B → γ) in REWRITES-FOR(B, grammar) do
          ADD-EDGE([j, j, B → • γ])
      end

  procedure COMPLETER([j, k, B → γ •])
      for each [i, j, A → α • B′ β] in chart[j] do
          if B = B′ then
              ADD-EDGE([i, k, A → α B′ • β])
      end
```

University of Bielefeld

Chartparser

Edge	Procedure	Derivation
a	INITIALIZER	[0, 0, $S' \rightarrow \bullet S$]
b	PREDICTOR(a)	[0, 0, $S \rightarrow \bullet NP\ VP$]
c	PREDICTOR(b)	[0, 0, $NP \rightarrow \bullet Pronoun$]
d	SCANNER(c)	[0, 1, $NP \rightarrow Pronoun \bullet$]
e	COMPLETER(b,d)	[0, 1, $S \rightarrow NP \bullet VP$]
f	PREDICTOR(e)	[1, 1, $VP \rightarrow \bullet Verb$]
g	PREDICTOR(e)	[1, 1, $VP \rightarrow \bullet VP\ NP$]
h	SCANNER(f)	[1, 2, $VP \rightarrow Verb \bullet$]
i	COMPLETER(g,h)	[1, 2, $VP \rightarrow VP \bullet NP$]
j	PREDICTOR(g)	[2, 2, $NP \rightarrow \bullet Pronoun$]
k	SCANNER(j)	[2, 3, $NP \rightarrow Pronoun \bullet$]
l	COMPLETER(i,k)	[1, 3, $VP \rightarrow VP\ NP \bullet$]
m	COMPLETER(e,l)	[0, 3, $S \rightarrow NP\ VP \bullet$]

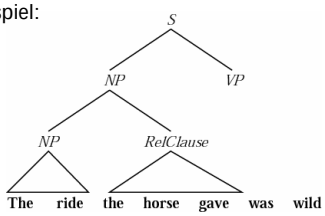


- Im Beispiel werden 13 Kanten in chart eingetragen, 8 sind unvollständig und 5 komplett.
- Predictor-Scanner-Completer Zyklen, etwa:
  - Kante (a) sucht nach S, um eine NP (b) mit folgenden Pronomen (c) vorherzusehen.
  - Scanner findet Pronomen (d) am richtigen Platz.
  - Completer kombiniert die unvollständige Kante (b) mit der vollständigen Kante (d) für eine neue Kante (e).

University of Bielefeld

Chartparser

- Chartparser vermeiden die Erzeugung vieler Kanten bei einfachem bottom-up Vorgehen.
- Beispiel:



  - bottom-up würde „ride the horse“ als VP labeln und später verwerfen.
  - In der Beispielgrammatik E0 ist aber keine Konkatenation von VP zu the; der Chartparser verhindert die unnötige VP Erzeugung.
  - Algorithmen welche von links nach rechts arbeiten und die Erzeugung unnötiger Konstituenten erzeugen heißen **left-corner Parser**.
- Zeit- und Speicherbedarf bei Chartparsern ist polynomial.
  - $O(kn^2)$ Speicher für die Kanten, mit n: Anzahl der Worte und k: Grammatikkonstante.
  - Der Algorithmus terminiert, wenn keine Kanten mehr erzeugt werden können.
  - $O(n^3)$ Zeitaufwand im worst-case (Bestes Ergebnis für kontextfreie Sprachen).