Parsing-Repetition

Übersicht

- Falsifizierbarkeit, oder: Sind Grammatiken wissenschaftlich?
- Grammatik, Formalismus
- Kontextfreie Grammatiken
- Ableitungen
- DCGs
- linksrekursive Grammatiken

Ziel

- Kurz-Repetition der im letzten Semester eingeführten Konzepte

Sind Grammatiken wissenschaftlich?

Falsifizierbarkeit

- Bei einer wissenschaftlichen Theorie soll klar sein, durch welche Beobachtungen sie widerlegt *(falsifiziert)* werden könnte.

Womit wird eine Grammatik widerlegt?

- Bei vielen linguistischen Theorien unklar (!)
- Umdenken seit den 1950er Jahren
  - Theorie über die Struktur einer Sprache = *formales* System von Grammatikregeln
- Beobachtbar bei einer Grammatik:
  - Ein korrekter Satz entspricht nicht der Grammatik
  - Ein falscher Satz entspricht der Grammatik

    → klares, nachprüfbares Falsifikations-Kriterium

Grammatik, Formalismus

Grammatiken sollen so festgehalten werden, dass nachprüfbar ist, ob ein Satz der Grammatik entspricht.

- Das heisst auch: In einer Notation, deren Bedeutung klar ist
  - Natürliche Sprache ist jedoch unscharf und mehrdeutig
  - Daher eignet sich eine formale Schreibweise besser

Formalismus

- Formalisierung: Verwendung formaler Sprachen der Mathematik zur Beschreibung der Grammatik einer natürlichen Sprache
- Vorteile eines Formalismus mit mathematischer Grundlage:
  - klarere Terminologie, grössere Präzision — »es ist klar, worüber geredet wird«
  - leichtere und zuverlässigere Überprüfbarkeit der Argumente

Kontextfreie Grammatiken (CFG)

Kontextfreie Grammatiken *(Context-Free Grammars, CFG)* sind ein solcher Formalismus

- gut erforschte mathematische Eigenschaften
- Nachteile:
  - eine exotische Konstruktion des Zürichdeutschen sowie gewisse Ausdrücke von Bambara können mit CFG *nicht* beschrieben werden
  - äusserst umständlich (aber nicht unmöglich!), etwas komplexere sprachliche Phänomene mit CFG zu modellieren
- Es gibt andere, besser geeignete Formalismen als CFG
  - Lexikalisch-Funktionale Grammatik (LFG), Generalisierte Phrasenstruktur-Grammatik (GPSG), Kopf-getriebene (head-driven) Phrasenstruktur-Grammatik (HPSG), Kategorialgrammatik (CG), Baumadjunktionsgrammatik (TAG), etc. etc.

Kontextfreie Regeln

Bestandteile einer kontextfreien Regel:

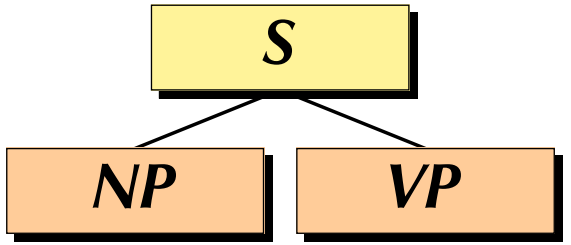
$$S \rightarrow NP\ VP$$

linke Seite
left-hand side

rechte Seite
right-hand side

Zwei gleichwertige Lesarten:

- Ein S besteht aus einer NP und einer VP
- Eine NP, gefolgt von einer VP, ergibt ein S



Kontextfreie Grammatiken

Beispiele für Regeln einer kontextfreien Grammatik:

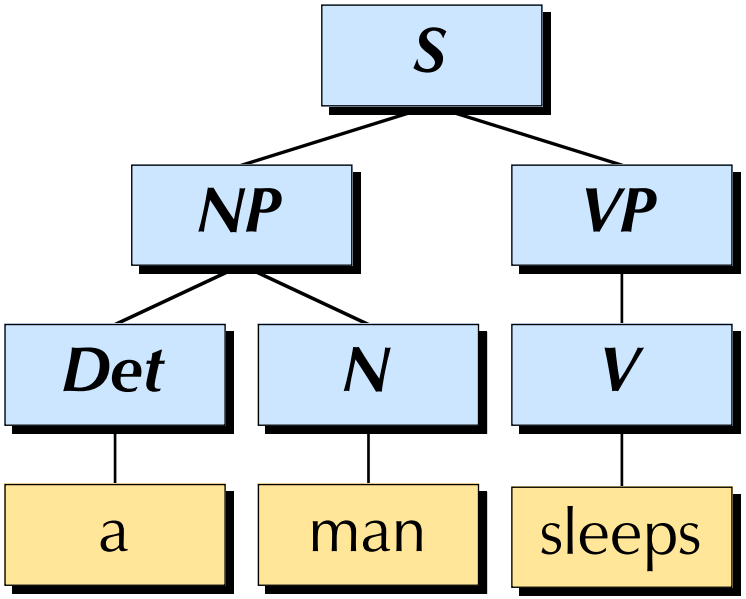
S → NP VP	*Det → the*	*V → loves*
NP → Det N	*Det → a*	*V → sleeps*
VP → V NP	*N → man*	*V → sees*
VP → V	*N → woman*	*V → thinks*

Gemäss diesen Grammatikregeln sind beispielsweise die folgenden Sätze (S) erlaubt:

- a man sleeps
- the woman sees a man
- a woman loves a woman

Kontextfreie Grammatiken

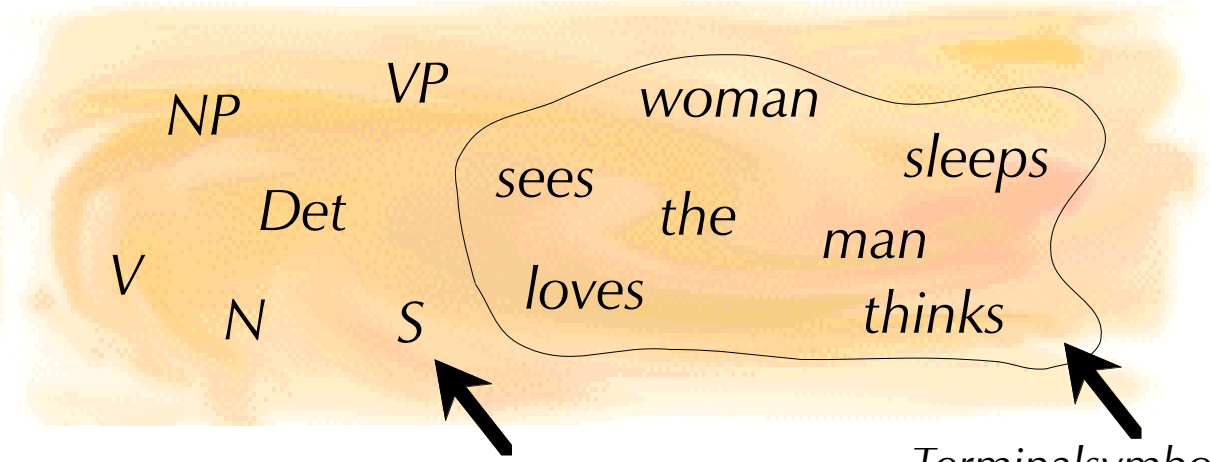
Struktur von »a man sleeps«:



$S \rightarrow NP\ VP$	$Det \rightarrow the$	$V \rightarrow loves$
$NP \rightarrow Det\ N$	$Det \rightarrow a$	$V \rightarrow sleeps$
$VP \rightarrow V\ NP$	$N \rightarrow man$	$V \rightarrow sees$
$VP \rightarrow V$	$N \rightarrow woman$	$V \rightarrow thinks$

Bestandteile einer CFG

Symbole — V



Startsymbol — S
(Element von V, aber nicht von Σ)

Terminalsymbole — Σ
(Teilmenge von V)

Regeln — R

S → NP VP	*Det → the*	*V → loves*
NP → Det N	*Det → a*	*V → sleeps*
VP → V NP	*N → man*	*V → sees*
VP → V	*N → woman*	*V → thinks*

Bestandteile einer CFG

Eine kontextfreie Grammatik ist ein 4-Tupel $\langle V, \Sigma, R, S \rangle$, wobei

- V ist eine endliche Menge von **Symbolen**
- Σ ist eine endliche Menge von **Terminalsymbolen** $\subset V$
- R — endliche Menge von **Regeln** $\subseteq (V - \Sigma) \times V^*$
- S — **Startsymbol** $\in V - \Sigma$

Nicht-Terminale

- $V - \Sigma$ (die Menge der Elemente von V, die nicht in Σ sind)

Wir schreiben für die Regeln $A \rightarrow u$

- statt $\langle A, u \rangle$, um die Regeln lesbarer zu machen

Ableitung

Für $u, v \in V^*$ schreibt man $u \Rightarrow v$ genau dann wenn

- es gibt Ketten $x, y, v' \in V^*$ und $A \in V - \Sigma$ so dass
  - $u = x\,A\,y$
  - $v = x\,v'\,y$
  - $A \rightarrow v'$

Beispiele

- *a man V a woman* ⇒ *a man sees a woman*
- *Det man V a woman* ⇒ *a man V a woman*
- *a man VP* ⇒ *a man V NP*
- *S* ⇒ *NP VP*

S → NP VP	*Det → the*	*V → loves*
NP → Det N	*Det → a*	*V → sleeps*
VP → V NP	*N → man*	*V → sees*
VP → V	*N → woman*	

Ableitung

Eine Folge von durch ⇒ verbundenen Ketten heisst Ableitung *(derivation)*.

Da in jedem Schritt *irgendein* Nichtterminal ersetzt wird, kann dieselbe Kette auf verschiedene Weisen vom Startsymbol abgeleitet werden:

- *S ⇒ NP VP ⇒ Det N VP ⇒ a N VP ⇒ a man VP ⇒ a man V ⇒ a man sleeps*
- *S ⇒ NP VP ⇒ NP V ⇒ Det N V ⇒ Det man V ⇒ Det man sleeps ⇒ a man sleeps*

Parser, Akzeptor

Ein Programm zur syntaktischen Analyse (»Parser«)

- nimmt eine Kette von Wörtern entgegen
- beurteilt, ob die Eingabe gemäss den Regeln einer Grammatik zulässig ist
  - Wenn ja, geben richtige Parser die Struktur der Eingabekette aus
  - Akzeptoren antworten nur mit »akzeptabel« bzw. »nicht akzeptabel«

Für denselben Grammatik-Formalismus kann es verschiedene Parsingverfahren geben

- Für kontextfreie Grammatiken: Dutzende Verfahren

Top-Down-Parsing

Ein Top-Down-Parser für eine kontextfreie Grammatik

- fängt mit dem Startsymbol an
- führt wiederholt Ableitungsschritte durch
- Ziel: Ableiten der zu analysierenden Kette

$S \Rightarrow NP\ VP \Rightarrow Det\ N\ VP \Rightarrow a\ N\ VP \Rightarrow a\ man\ VP \Rightarrow a\ man\ V \Rightarrow a\ man\ sleeps$

$S \rightarrow NP\ VP$	$Det \rightarrow the$	$V \rightarrow loves$
$NP \rightarrow Det\ N$	$Det \rightarrow a$	$V \rightarrow sleeps$
$VP \rightarrow V\ NP$	$N \rightarrow man$	$V \rightarrow sees$
$VP \rightarrow V$	$N \rightarrow woman$	$V \rightarrow thinks$

Bottom-Up-Parsing

Ein Bottom-Up-Parser für eine kontextfreie Grammatik

- fängt mit der zu analysierenden Kette an
- führt wiederholt Ableitungsschritte *»rückwärts«* durch
- Ziel: Erreichen des Startsymbols

$S \Rightarrow NP\ VP \Rightarrow NP\ V \Rightarrow NP\ sleeps \Rightarrow Det\ N\ sleeps \Rightarrow Det\ man\ sleeps \Rightarrow$ *a man sleeps*

$S \rightarrow NP\ VP$	$Det \rightarrow the$	$V \rightarrow loves$
$NP \rightarrow Det\ N$	$Det \rightarrow a$	$V \rightarrow sleeps$
$VP \rightarrow V\ NP$	$N \rightarrow man$	$V \rightarrow sees$
$VP \rightarrow V$	$N \rightarrow woman$	$V \rightarrow thinks$

Definit-Klausel-Grammatiken (DCGs)

In Prolog ist ein einfacher Top-Down-Parser bereits eingebaut.

- Grammatik besteht aus Prolog-Klauseln (Definiten Klauseln)
- nützlich zum schnellen Ausprobieren einer Mini-Grammatik

Allerdings:

- eher ineffizientes Verfahren
  - bei mehrdeutigen Grammatiken werden unter Umständen Teile des Satzes mehrmals analysiert
- »Aufhängen« bei bestimmten Grammatiken
- ⇒ für richtige Sprachverarbeitungsprojekte so gut wie unbrauchbar

DCGs: Formalismus

Regeln mit Nichtterminalen auf der rechten Seite:

```
s --> np, vp.
np --> det, n.
```

```
vp --> v.
vp --> v, np.
```

Regeln mit Terminal-Symbolen auf der rechten Seite:

```
det --> [the].
det --> [a].
```

```
n --> [cat].
v --> [sees].
v --> [sings].
```

DCGs: Benutzung

Ein eingebautes Prädikat phrase/2 überprüft, ob von einem Nichtterminal eine Kette von Terminalsymbolen abgeleitet werden kann.

```
?- phrase(s, [the, cat, sings]).
yes
```

```
?- phrase(np, [a, cat]).
yes
```

```
?- phrase(s, [a, cat]).
no
```

Aufgaben: Parsing-Repetition

Programmiertechniken der Computerlinguistik 2 · Sommersemester 1999

1. Das Fohlen schnaubt

Gegeben sei eine kontextfreie Grammatik mit Startsymbol *S* und folgenden Regeln:

S → *NP VP*	*NP* → *Det N*
VP → *V*	*Det* → *das*
N → *Fohlen*	*V* → *schnaubt*

Nenne eine der möglichen Ableitungen für die Kette *»das Fohlen schnaubt«* vom Startsymbol.

2. Grammatik für deutsche Sätze

Im letzten Semester war es eine der Aufgaben zum Thema »Definit-Klausel-Grammatiken«, eine Grammatik für deutsche Sätze zu formulieren, welche auch Kongruenzerscheinungen berücksichtigt. Die damals als Musterlösung ausgeteilte Grammatik findest Du auf einem separaten Blatt.

Probiere die Grammatik aus, indem Du einige Sätze analysieren lässt. Repetiere die Lektion zu Definit-Klausel-Grammatiken aus dem letzten Semester, wenn Dir nicht mehr klar ist, welche Funktion den Argumenten in Klammern zukommt.

3. Alle ableitbaren Ketten

Schreibe ein Prolog-Prädikat, welches in einem *Failure-Driven Loop* wiederholt `phrase/2` aufruft. Hierbei soll es alle Terminalketten auf den Bildschirm ausgeben, die vom Startsymbol *s* ableitbar sind. Teste Dein Prädikat mit der Grammatik aus Aufgabe 2.

4. Adjektive

Erweitere die Grammatik von Aufgabe 2 um Adjektive wie *wahrheitsliebend, ernährungsbewusst* oder *regenbogenfarbig*. Du darfst selbstverständlich auch kürzere Adjektive verwenden.

Es sollte nun möglich sein, Ketten wie *die wahrheitsliebende Katze geht* vom Startsymbol *s* abzuleiten, nicht aber Ketten wie **die wahrheitsliebendes Katze geht*.[1]

5. Bestimmte und unbestimmte Artikel

Die Grammatik enthält bisher nur bestimmte Artikel *(der, die, das),* nicht aber unbestimmte Artikel *(ein, eine)*.

a) Erweitere die Grammatik entsprechend.

b) Ist die Grammatik nun übergenerierend? Stelle dies mit dem Prädikat aus Aufgabe 3 fest: Werden Ketten ausgegeben, die für Dein Sprachempfinden ungrammatisch sind? *Tip:* Achte insbesondere auf den Gegensatz zwischen *der regenbogenfarbige Hund* und *ein regenbogenfarbiger Hund*.

c) Schlage gegebenenfalls in einer Grammatik des Deutschen nach, welche Eigenschaften zusätzlich zu *Kasus, Numerus* und *Genus* in deutschen Nominalphrasen eine Rolle spielen.

d) Verändere die Grammatik so, dass Ketten wie **ein regenbogenfarbige Hund geht* oder **der ernährungsbewusster Hund geht* nicht mehr vom Startsymbol *s* abgeleitet werden können.

[1] Der Stern wird in der Sprachwissenschaft üblicherweise dazu verwendet, ungrammatische Sätze kennzuzeichnen.

dcg.pl

Programmiertechniken der Computerlinguistik 2 · Sommersemester 1999
Online unter http://www.coli.uni-sb.de/~brawer/prolog/dcg/

```
% ------------------------------------------------------------
%  det(Kasus, Numerus, Genus) - Artikel
% ------------------------------------------------------------

det(nom, sg, m) --> [der].  % der Hund ist wichtig
det(gen, sg, m) --> [des].  % Max leidet wegen des Hundes
det(dat, sg, m) --> [dem].  % mit dem Hund ist das Leben schön
det(akk, sg, m) --> [den].  % Daniel dämonisierte den Hund

det(nom, pl, m) --> [die].  % die Hunde sind wichtig
det(gen, pl, m) --> [der].  % Max leidet wegen der Hunde
det(dat, pl, m) --> [den].  % mit den Hunden ist das Leben schön
det(akk, pl, m) --> [die].  % Daniel dämonisierte die Hunde

det(nom, sg, f) --> [die].  % die Tasse ist wichtig
det(gen, sg, f) --> [der].  % Max leidet wegen der Tasse
det(dat, sg, f) --> [der].  % mit der Tasse ist das Leben schön
det(akk, sg, f) --> [die].  % Daniel dämonisierte die Tasse

det(nom, pl, f) --> [die].  % die Tassen sind wichtig
det(gen, pl, f) --> [der].  % Max leidet wegen der Tassen
det(dat, pl, f) --> [den].  % mit den Tassen ist das Leben schön
det(akk, pl, f) --> [die].  % Daniel dämonisierte die Tassen

det(nom, sg, n) --> [das].  % das Haus ist wichtig
det(gen, sg, n) --> [des].  % Max leidet wegen des Hauses
det(dat, sg, n) --> [dem].  % mit dem Haus ist das Leben schön
det(akk, sg, n) --> [das].  % Daniel dämonisierte das Haus

det(nom, pl, n) --> [die].  % die Häuser sind wichtig
det(gen, pl, n) --> [der].  % Max leidet wegen der Häuser
det(dat, pl, n) --> [den].  % mit den Häusern ist das Leben schön
det(akk, pl, n) --> [die].  % Daniel dämonisierte die Häuser


% ------------------------------------------------------------
%  n(Kasus, Numerus, Genus) - Nomina
% ------------------------------------------------------------

n(nom, sg, m) --> [hund].   % der Hund geht
n(gen, sg, m) --> [hundes]. % Max leidet wegen des Hundes
n(dat, sg, m) --> [hund].   % mit dem Hund ist das Leben schön
n(dat, sg, m) --> [hunde].  % mit dem Hunde ist das Leben schön
n(akk, sg, m) --> [hund].   % Daniel dämonisierte den Hund
n(nom, pl, m) --> [hunde].  % die Hunde gehen
n(gen, pl, m) --> [hunde].  % Max leidet wegen der Hunde
n(dat, pl, m) --> [hunden]. % mit den Hunden ist das Leben schön
n(akk, pl, m) --> [hunde].  % Daniel dämonisierte die Hunde

n(nom, sg, f) --> [katze].  % der Hund geht
n(gen, sg, f) --> [katze].  % Max leidet wegen des Hundes
n(dat, sg, f) --> [katze].  % mit dem Hunde ist das Leben schön
n(akk, sg, f) --> [katze].  % Daniel dämonisierte den Hund
n(nom, pl, f) --> [katzen]. % die Katzen gehen
n(gen, pl, f) --> [katzen]. % Max leidet wegen der Katzen
n(dat, pl, f) --> [katzen]. % mit den Katzen ist das Leben schön
n(akk, pl, f) --> [katzen]. % Daniel dämonisierte die Katzen


% ------------------------------------------------------------
%  pron(Kasus, Person, Numerus) - Pronomina
% ------------------------------------------------------------
% Auch Wörter wie "mein" würden üblicherweise als Pronomina
% bezeichnet. Es sind auch nicht alle Kasus verzeichnet.

pron(nom, 1, sg) --> [ich].
pron(gen, 1, sg) --> [meiner].
pron(dat, 1, sg) --> [mir].
pron(akk, 1, sg) --> [mich].
pron(nom, 2, sg) --> [du].
pron(nom, 3, sg) --> [er].
pron(nom, 3, sg) --> [sie].
pron(nom, 3, sg) --> [es].
pron(nom, 1, pl) --> [wir].
pron(nom, 2, pl) --> [ihr].
pron(nom, 3, pl) --> [sie].


% ------------------------------------------------------------
%  v(Person, Numerus, Transitivität) - Verben
% ------------------------------------------------------------

v(1, sg, trans) --> [sehe].    % ich sehe den Hund
v(2, sg, trans) --> [siehst].  % du siehst den Hund
v(3, sg, trans) --> [sieht].   % Anna sieht den Hund
v(1, pl, trans) --> [sehen].   % wir sehen den Hund
v(2, pl, trans) --> [seht].    % ihr seht den Hund
v(3, pl, trans) --> [sehen].   % sie sehen den Hund

v(1, sg, intrans) --> [gehe].  % ich gehe
v(2, sg, intrans) --> [gehst]. % du gehst
v(3, sg, intrans) --> [geht].  % Anna geht
v(1, pl, intrans) --> [gehen]. % wir gehen
v(2, pl, intrans) --> [gehen]. % ihr geht
v(3, pl, intrans) --> [gehen]. % sie gehen
```

```
% ------------------------------------------------------------
%  s - Sätze
% ------------------------------------------------------------

s -->
  np(nom, Person, Numerus, _), % das Genus interessiert nicht
  vp(Person, Numerus).

% ------------------------------------------------------------
%  np(Kasus, Person, Numerus, Genus) - Nominalphrasen
% ------------------------------------------------------------

np(Kasus, 3, Numerus, Genus) -->
  det(Kasus, Numerus, Genus),  % der
  n(Kasus, Numerus, Genus).    % Katzen

np(Kasus, Person, Numerus, _) -->
  pron(Kasus, Person, Numerus).

% ------------------------------------------------------------
%  vp(Person, Numerus) - Verbalphrasen
% ------------------------------------------------------------

vp(Person, Numerus) -->
  v(Person, Numerus, intrans).

vp(Person, Numerus) -->      % [sie] sahen die Katze: Keine
  v(Person, Numerus, trans), % Kongruenz zwischen Verb und
  np(akk, _, _, _).          % Akkusativobjekt!
```

Von *s* ableitbare Terminalketten

der hund geht · der hund sieht den hund · der hund sieht die hunde · der hund sieht die katze · der hund sieht die katzen · der hund sieht mich · die hunde gehen · die hunde sehen den hund · die hunde sehen die hunde · die hunde sehen die katze · die hunde sehen die katzen · die hunde sehen mich · die katze geht · die katze sieht den hund · die katze sieht die hunde · die katze sieht die katze · die katze sieht die katzen · die katze sieht mich · die katzen gehen · die katzen sehen den hund · die katzen sehen die hunde · die katzen sehen die katze · die katzen sehen die katzen · die katzen sehen mich · ich gehe · ich sehe den hund · ich sehe die hunde · ich sehe die katze · ich sehe die katzen · ich sehe mich · du gehst · du siehst den hund · du siehst die hunde · du siehst die katze · du siehst die katzen · du siehst mich · er geht · er sieht den hund · er sieht die hunde · er sieht die katze · er sieht die katzen · er sieht mich · sie geht · sie sieht den hund · sie sieht die hunde · sie sieht die katze · sie sieht die katzen · sie sieht mich · es geht · es sieht den hund · es sieht die hunde · es sieht die katze · es sieht die katzen · es sieht mich · wir gehen · wir sehen den hund · wir sehen die hunde · wir sehen die katze · wir sehen die katzen · wir sehen mich · ihr gehen · ihr seht den hund · ihr seht die hunde · ihr seht die katze · ihr seht die katzen · ihr seht mich · sie gehen · sie sehen den hund · sie sehen die hunde · sie sehen die katze · sie sehen die katzen · sie sehen mich