

Übungsblatt 9

Präsenzaufgaben

Aufgabe 1 Parsing-Algorithmen

Gegeben sei folgende kontextfreie Grammatik:

```
1 | S → NP VP
2 |
3 | NP → DET NOM
4 | NOM → ADJ NOM | N
5 |
6 | ADJ → "schöne" | "kluge" | "dicke"
7 | DET → "der" | "die" | "das"
8 | N → "Mann" | "Frau" | "Buch"
9 |
10 | VP → V NP NP | V NP | V
11 | V → "gibt" | "schenkt" | "schläft" | "gefällt" | "kennt"
```

Parsen Sie jeweils alle unten angegebenen Sätze mit jedem der folgenden Parsing-Algorithmen. Bei der Auswahl von Regeln soll immer von oben nach unten und (bei Alternativen) von links nach rechts vorgegangen werden.

- (a) Recursive Descent Parser
- (b) Shift Reduce Parser
- (c) Earley Algorithmus

- (1) Der Mann gibt der Frau das Buch.
- (2) Die kluge Frau kennt das dicke Buch.
- (3) Der Mann schläft.

Hausaufgaben

Aufgabe 2 Parsing-Algorithmen Reloaded

Kombinationen von Algorithmus und Satz in Aufgabe 1, die während der Präsenzaufgaben nicht behandelt wurden, stellen zusätzliches Übungsmaterial dar.

Aufgabe 3 Imperativ ohne Subjekt

Betrachten Sie die folgende kontextfreie Merkmalsgrammatik:

```
1 | S → NP VP
2 | S → VP
3 |
4 | NP → DET N
5 | VP → V[SUBCAT=tr] NP
6 | VP → V[SUBCAT=intr]
7 |
8 | DET → "das"
9 | N → "Kind" | "Buch"
10 |
11 | V[SUBCAT=tr] → "lies"
12 | V[SUBCAT=tr] → "liest"
13 | V[SUBCAT=intr] → "schlaf"
14 | V[SUBCAT=intr] → "schläft"
```

Hier wurde versucht, Aufforderungssätze zu modellieren. Allerdings akzeptiert diese Grammatik immer noch viele ungrammatische Sätze.

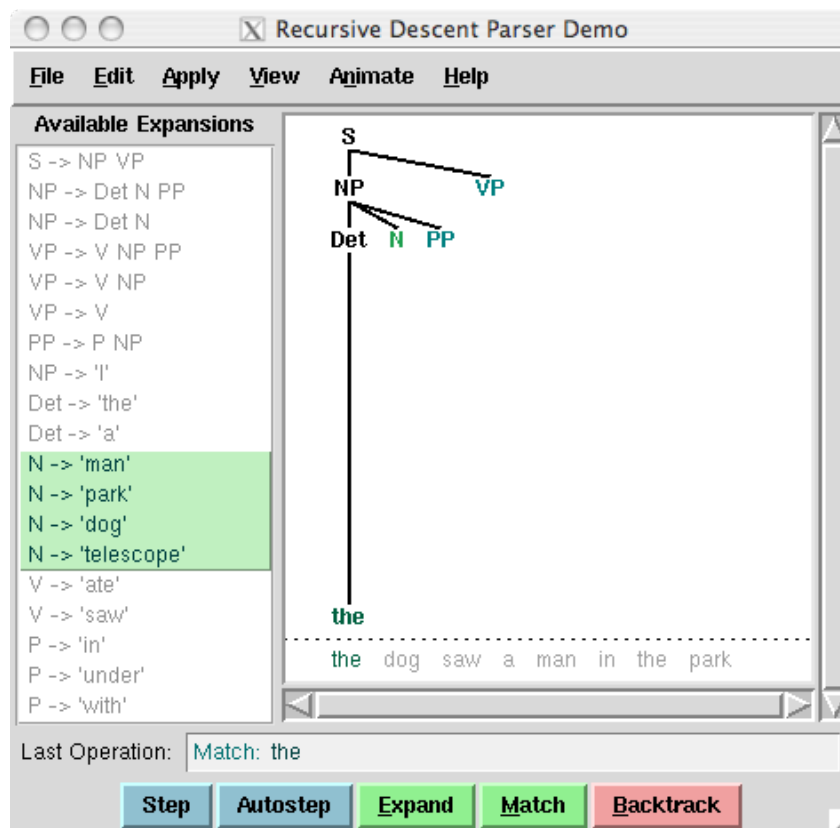
- (a) Fügen Sie ein Merkmal MOOD mit den möglichen Werten *imp* und *ind* für Imperativ und Indikativ hinzu und deklarieren Sie die entsprechenden Werte im Lexikon (den lexikalischen Regeln).
- (b) Formulieren Sie dann Bedingungen in den syntaktischen Regeln, sodass folgende grammatische Sätze (ohne *) weiterhin akzeptiert und folgende ungrammatische Sätze (mit *) abgelehnt werden.

- (4) das Kind schläft
- (5) das Kind liest das Buch
- (6) lies das Buch
- (7) schlaf
- (8) * das Kind schlaf

- (9) * das Kind lies das Buch
- (10) * liest das Buch
- (11) * schläft

*Aufgabe 4 Recursive Descent Parser Demo

Die Demo-Applikation `nltk.app.rdparger()` veranschaulicht die Funktionsweise eines Recursive Descent Parsers. Spielen Sie ein wenig mit der Demo, nehmen Sie Änderungen an der Grammatik vor und stellen Sie sicher, dass Sie verstehen wie jeder einzelne Schritt zustandekommt.



*Aufgabe 5 NLTK-Kapitel zum Parsing (8.4 und 8.2-extras)

In folgenden NLTK-Kapiteln zum Parsing mit kontextfreien Grammatiken können Sie sich weiterführend über top-down-, bottom-up- sowie Chart-Parsing informieren:

- **Teilkapitel 8.4** ('Parsing With Context Free Grammar'): <http://www.nltk.org/book/ch08.html>
- **Teilkapitel 2.1-2.8** ('Chart Parsing') des **Zusatzkapitels zu Kapitel 8**: <http://www.nltk.org/book/ch08-extras.html>