

# Übungen 12: Shift-Reduce-Parsing

Programmiertechniken in der Computerlinguistik I · Wintersemester 2004/2005

Um keine Zeit mit Tippen und Tippfehlerkorrektur zu verschwenden, findest du Programmtexte dieser Übungen unter <http://www.cl.unizh.ch/siclemat/lehre/ws0405/pci1/uebung12.txt>

## 1. Shift-Reduce-Parsing auf Papier

Gegeben sei folgende Grammatik:

|                       |                        |
|-----------------------|------------------------|
| $S \rightarrow NP VP$ | $NP \rightarrow Det N$ |
| $VP \rightarrow V$    | $Det \rightarrow der$  |
| $N \rightarrow Magen$ | $V \rightarrow knurrt$ |

Zeichne die einzelnen Schritte auf, die ein Shift-Reduce-Parser vornimmt, um die Eingabe 'der Magen knurrt' vom Startsymbol  $S$  abzuleiten. Gib für jede Aktion an, wie der Stapel und die verbleibende Eingabekette aussehen. Nimm folgende Darstellungsform:

| Schritt | Aktion | Stapel | Eingabekette       |
|---------|--------|--------|--------------------|
| 0       | —      | []     | [der,Magen,knurrt] |
| ...     | ...    | ...    | ...                |

## 2. Shift-Reduce-Parsing-Algorithmus

Lade die Programmtexte von der Homepage herunter oder schreibe sie von den Vorlesungsfolien ab. (Hilft manchmal sogar beim Verständnis :-)

a) Schreibe die Grammatik und das Lexikon von Aufgabe 1 im Format des Shift-Reduce-Parsers hin. Stelle den Trace-Modus ein, parse den Satz "der Magen kurt" und überprüfe dabei deine Papiertabelle aus Aufgabe 1.

b) Lass den Shift-Reduce-Parser die Eingabekette [der, 'Magen', der, 'Magen'] analysieren. Beobachte das Verhalten des Shift-Reduce-Parsers und interpretiere das Resultat.

c) Der Standard-DCG-Prolog-Parser konnte mit `phrase/2` aufgerufen werden. Definiere ein Prädikat `shift_reduce_parse/2`, das dasselbe für den neuen Algorithmus macht.

```
?- shift_reduce_parse(s, [der, 'Magen', knurrt]).
```

d) Ergänze die Grammatik um die folgenden Adjektiv-Regeln:

|                            |                            |
|----------------------------|----------------------------|
| $NP \rightarrow Det Adj N$ | $Adj \rightarrow \epsilon$ |
|                            | $Adj \rightarrow leere$    |

Stelle den Trace-Modus von Prolog ein. Lass den Shift-Reduce-Parser die Eingabekette [der, leere, 'Magen'] analysieren. Beobachte das Verhalten des Shift-Reduce-Parsers und interpretiere das Resultat.

e) Entferne die in Aufgabe 2d) eingefügten Regeln wieder aus der Grammatik. Ergänze die Grammatik nun um die folgenden Regeln:

|                             |                        |
|-----------------------------|------------------------|
| $NP \rightarrow NP Conj NP$ | $Conj \rightarrow und$ |
|                             | $N \rightarrow Hund$   |

Lass den Shift-Reduce-Parser die Eingabekette [der, 'Magen', und, der, 'Hund'] analysieren. Wie geht er mit rekursiven Regeln um?

f) Mache die Grammatiksymbole komplex, so dass der Syntaxbaum als Argument zurückgeliefert wird.

```
?- shift_reduce_parse(s(Baum), [der, 'Magen', knurrt]).  
Baum = s(np(det(der), n('Magen')), vp(v(knurrt)))
```