

Übungen 12: Shift-Reduce-Parsing

Programmiertechniken in der Computerlinguistik I · Wintersemester 2000/2001

Um keine Zeit mit Tippen und Tippfehlerkorrektur zu verschwenden, findest du die Programmtexte dieser Übung unter <http://www.ifi.unizh.ch/cl/sicemat/lehre/ws0001/pcl1/uebung12.txt>

1. Shift-Reduce-Parsing

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) 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_phrase(s, [a,man,sleeps]).
```

- b) Lass den Shift-Reduce-Parser die Eingabekette `[a,man,a,man]` analysieren. Beobachte das Verhalten des Shift-Reduce-Parsers und interpretiere das Resultat.

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

$NP \rightarrow Det A N$	$A \rightarrow \epsilon$
	$A \rightarrow nice$

Stelle den Trace-Modus von Prolog ein. Lass den Shift-Reduce-Parser die Eingabekette `[a,nice,man]` analysieren. Beobachte das Verhalten des Shift-Reduce-Parsers und interpretiere das Resultat.

- d) Mache die Grammatiksymbole komplex, so dass der Syntaxbaum zurückgeliefert wird.

```
?- shift_reduce_parse(s(Baum), [a,man,sleeps]).  
Baum = s(np(det(a),n(man)),vp(v(sleeps)))
```

