Universität Karlsruhe (TH)

Institut für Innovatives Rechnen und Programmstrukturen (IPD)

Übersetzerbau WS 2003/04 — http://www.info.uni-karlsruhe.de/

Dozent: Prof. Dr.rer.nat. G. Goos — goos@ipd.info.uni-karlsruhe.de

Übungsleiter: Rubino Geiß — rubino@ipd.info.uni-karlsruhe.de

Übungsblatt 3 | Ausgabe: 17.11.2003 | Besprechung: 20.11.2003

Aufgabe 1: *Mehrdeutige Grammatiken*

Gegeben sei folgende Grammatik:

$$
\begin{array}{lcl}
\mathit{stmt} & \longrightarrow & \textbf{if } \mathit{expr} \textbf{ then } \mathit{stmt} \\
& | & \textbf{if } \mathit{expr} \textbf{ then } \mathit{stmt} \textbf{ else } \mathit{stmt} \\
& | & \mathit{other\_stmt}
\end{array}
$$

Zeigen Sie, dass diese Grammatik mehrdeutig ist. Welches Problem ergibt sich dadurch?

Aufgabe 2: *LL(0)-Eigenschaft*

2.1 $|L(G)|$

Wieviele Wörter enthält eine Sprache, die durch eine LL(0)-Grammatik beschrieben werden kann?

2.2 $|P|$

Wieviele Produktionen gibt es in einer LL(0)-Grammatik für jedes Nichtterminal?

Aufgabe 3: *LR(k)-Bedingung*

Untersuchen Sie, ob und für welches k die folgenden Grammatiken die $LR(k)$-Bedingung erfüllen:

(a)
$$
\begin{array}{l}
S \to C \mid D \\
C \to aC \mid b \\
D \to aD \mid c
\end{array}
$$

(b)
$$
\begin{array}{l}
S \to Ab \mid Bc \\
A \to Aa \mid e \\
B \to Ba \mid e
\end{array}
$$

(c)
$$
\begin{array}{l}
S \to E \\
E \to aAd \mid aBc \mid bAc \mid bBd \\
A \to eA \mid e \\
B \to eB \mid e
\end{array}
$$

Aufgabe 4: *LALR-Generatoren*

```
%%
e : "id" |       /* eine Variable */
    m            /* ein Methodenaufruf */
  ;

m : "id" |       /* Methode ohne Parameter */
    m "(" a ")"  /* mit Parameter und Deprozedurierung */
  ;

a : "id" |       /* Argumentliste */
    a "," "id"
  ;
```

Wenn man obige Grammatik mit 'bison' oder 'yacc' verarbeiten will, bekommt man die Meldung:

```
rrc.y contains 1 reduce/reduce conflict.
```

4.1 *Konflikte in der LALR-Tabelle*

Wie erklären sie das?

4.2 *LALR Eigenschaft herstellen*

Wie kann man das beheben, ohne die Sprache zu verändern?

4.3 *shift/reduce conflict*

Geben Sie eine Grammatik an die einen 'shift/reduce conflict' enthält!

Aufgabe 5: *LR(0), SLR(1), LALR(1) und LR(1)*

5.1 *SLR(1), ¬LR(0)*

Geben Sie eine SLR(1)-Grammatik an, die nicht LR(0) ist.

5.2 *LALR(1), ¬SLR(1)*

Geben Sie eine LALR(1)-Grammatik an, die nicht SLR(1) ist.

5.3 *LR(1), ¬LALR(1)*

Geben Sie eine LR(1)-Grammatik an, die nicht LALR(1) ist.

5.4 *¬LR(k)*

Geben Sie eine eindeutige Grammatik an, die nicht LR(k) ist (für alle k).

5.5 *LR(k), ¬LL(k)*

Geben Sie eine LR(k)-Grammatik an, die nicht LL(k) ist (für alle k).

Aufgabe 6: *LALR(1)-Zerteiler, Shift-Reduce-Übergänge*

Gegeben sei die Grammatik

$Z \to S$	(1)	$L \to *R$	(4)
$S \to L = R$	(2)	$L \to a$	(5)
$S \to R$	(3)	$R \to L$	(6)

6.1 *Konstruktion*

Zeigen Sie, daß die Grammatik LALR(1) ist durch Bestimmung der Zerteilertabelle/Übergangsmatrix. Ist sie auch SLR(1)? Wie kann man dies anhand der Zerteilertabelle begründen?

6.2 *Kettenproduktionen*

Eliminieren Sie die Kettenproduktionen (vgl. 7.3.4 in [Waite/Goos]) und stellen Sie die Übergangsmatrix auf.

6.3 *Shift-Reduce-Übergänge*

Führen Sie Shift-Reduce-Übergänge in der Übergangsmatrix ein. Diskutieren Sie die Auswirkungen der verschiedenen Methoden zur Bestimmung/Optimierung der Übergangsmatrix auf die Zerteilung eines Programms.

Aufgabe 7: *Fehlerbehandlung*

7.1 *LL*

Gegeben sei die Grammatik

$$\begin{aligned} Z &\to E\# \\ E &\to FE' \\ E' &\to +FE' \mid \varepsilon \\ F &\to i \mid (E) \end{aligned}$$

Welche Produnktionen zeichnet man aus? Wie sieht der entsprechende LL(1)-Zerteiler mit Error-Recovery aus? Akzeptieren Sie $(i(i+i\#$.

7.2 *LR*

Gegeben sei die Grammatik

(1) $Z \to E\#$
(2) $E \to E + F$
(3) $E \to F$
(4) $F \to i$
(5) $F \to (E)$

Erzeugen Sie einen LR-Zerteiler mit Error-Recovery. Akzeptieren Sie wiederum den String $(i(i+i\#$. Was fällt auf?