

Lösungsvorschlag zum Aufgabenblatt 7

 Punkte von 11

Aufgabe 1 (2+4+2+3)

a) Was ist ein Compiler?

Ein Programm, das einen Quelltext aus einer höheren Programmiersprache in Maschinen- oder Assemblercode übersetzt.

b) Erläutern Sie an einem Beispiel aus der im Praktikum benutzten Programmiersprache SPL die nachfolgenden Begriffe.

- Typdefinition
 - Typbezeichner
 - Typausdruck
 - Typ
- Ein Typ ist ein Sprachkonzept einer typisierten Programmiersprache, das eine Menge von Werten, deren interne Repräsentation im Speicher und die darauf verfügbaren Operationen festlegt.
- Eine Typdefinition definiert einen neuen Typbezeichner und ordnet ihm einen Typausdruck zu.
Beispiel:

```
type paar = array [2] of int;
```

- Der Typbezeichner ist *paar*
- Der Typausdruck ist `array [2] of int`

c) Betrachten Sie den nachfolgenden Ausdruck mit den binären Operatoren \circ und \times :

$$a \circ b \times c \times d \circ e \circ f \times g$$

Wenn \circ rechtsassoziativ und \times linksassoziativ ist und \circ die höhere Präzedenz hat, wie sieht dann ein äquivalenter vollständig geklammerter Ausdruck aus? Ergänzen Sie oben die Klammern, so dass die Zuordnung der Operanden zu den Operatoren klar ersichtlich wird.

$$(((a \circ b) \times c) \times (d \circ (e \circ f))) \times g$$

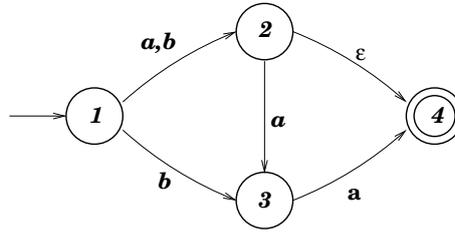
d) Was ist eine kontextfreie Grammatik? Geben Sie eine exakte Definition an.

siehe Skript

Aufgabe 2 (3+3+3 Punkte)

a) Vereinfachen Sie den regulären Ausdruck $(a \mid \varepsilon)^+bc \mid (aab)c \mid bc \mid bc bc$
 $(a^* \mid bc)bc$

b) Betrachten Sie den regulären Ausdruck $r = b \mid ba \mid baa \mid (a \mid b)(a \mid \varepsilon)$ und den durch das nachfolgende Zustandsübergangsdiagramm definierten NEA A



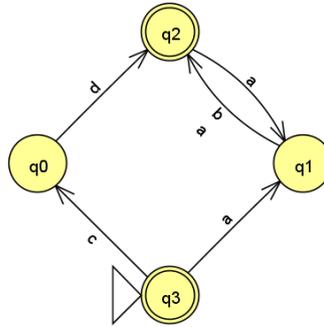
Beweisen Sie, dass r und A nicht äquivalent sind.

r und A sind äquivalent, g.d.w. $L(r) = L(A)$

A akzeptiert das Wort aaa mit der Berechnung $1 \xrightarrow{a} 2 \xrightarrow{a} 3 \xrightarrow{a} 4$, so dass $aaa \in L(A)$

Wie man leicht sieht, passt aaa aber zu keiner der 4 Varianten von r , so dass $aaa \notin L(r)$

c) Geben Sie zu dem regulären Ausdruck $r = (ab \mid cd \mid \varepsilon)(aa \mid ab)^*$ einen äquivalenten endlichen Automaten an. Die volle Punktzahl gibt es nur für einen deterministischen Automaten, Nichtdeterminismus führt zur Abwertung.



Aufgabe 3 (3+3+3+2+3+2 Punkte)

Gegeben sei folgende kontextfreie Grammatik G

- $S \rightarrow ABC$
- $A \rightarrow aaA \mid Ae \mid \varepsilon$
- $B \rightarrow bbBc \mid A$
- $C \rightarrow cC \mid d$

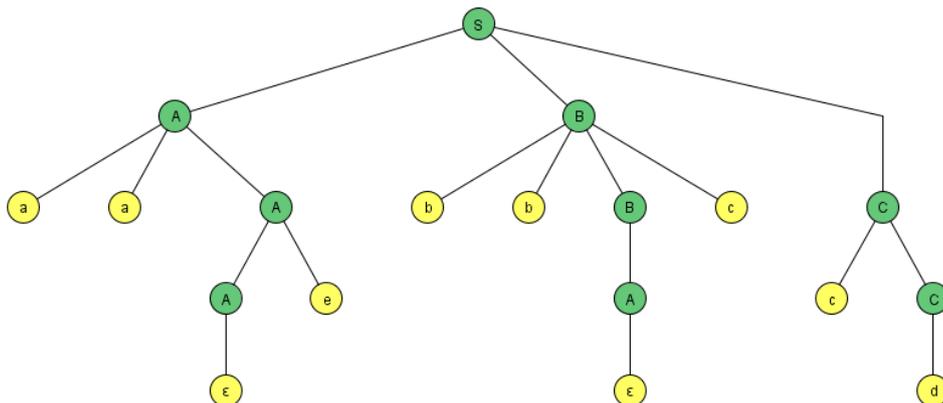
a) Geben Sie drei Wörter aus $L(G)$ mit jeweils maximal 2 Zeichen an: d, cd, ed

b) Geben Sie eine Rechtsableitung und den Ableitungsbaum zu $aeebbccd$ an.

Rechtsableitung:

$S \Rightarrow ABC \Rightarrow ABcC \Rightarrow ABcd \Rightarrow AbbBccd \Rightarrow AbbAccd \Rightarrow Abbccd \Rightarrow aaAbbccd \Rightarrow aaAebbccd \Rightarrow aeebbccd$

Ableitungsbaum:



c) Beweisen Sie, dass die Grammatik mehrdeutig ist.

G ist mehrdeutig, g.d.w. es für ein Wort $w \in L(G)$ zwei verschiedene Rechtsableitungen gibt. Für ed gibt es zwei Rechtsableitungen:

$$S \Rightarrow ABC \Rightarrow ABd \Rightarrow AAd \Rightarrow AAed \Rightarrow Aed \Rightarrow ed$$

$$S \Rightarrow ABC \Rightarrow ABd \Rightarrow AAd \Rightarrow Ad \Rightarrow Aed \Rightarrow ed$$

d) Bestimmen Sie $FIRST(S)$: $\{a, e, b, c, d\}$

e) Bestimmen Sie $FOLLOW(A)$: $\{e, b, a, c, d\}$

f) Was steht in der LL(1)-Parsertabelle in dem Eintrag zu A und e ? $A \rightarrow \epsilon, A \rightarrow Ae$

Punkte von 14

Aufgabe 4 (6+4+4 Punkte)

a) Bestimmen Sie zur nachfolgenden Grammatik die LR(0)-Elemente und die Übergänge im zugehörigen DEA.

b) Geben Sie die SLR(1)-Parsertabelle dazu an.

c) Geben Sie die Berechnung des SLR(1)-Parsers für die Eingabe aba an. Falls die Tabelle Shift/Reduce-Konflikte enthält, soll der Parser dabei immer die SHIFT-Aktion wählen.

- | | |
|------------------------------|------------------------------|
| (1) $S \rightarrow aBS$ | (0) $S' \rightarrow S\$$ |
| (2) $S \rightarrow a$ | (1) $S \rightarrow aBS$ |
| (3) $B \rightarrow bB$ | (2) $S \rightarrow a$ |
| (4) $B \rightarrow \epsilon$ | (3) $B \rightarrow bB$ |
| | (4) $B \rightarrow \epsilon$ |
- Erweiterte Grammatik:

Lösung zu a und b (mit JFLAP berechnet)

The screenshot shows the JFLAP interface for building an SLR(1) parser. On the left, the grammar is defined as:

- $S' \rightarrow S$
- $S \rightarrow aBS$
- $S \rightarrow a$
- $B \rightarrow bB$
- $B \rightarrow \epsilon$

The parse table is complete and shows the following entries:

		FIRST	FOLLOW
B		{e, b}	{a}
S		{a}	{\\$}

The DFA diagram shows states q_0 through q_6 with transitions on terminals a, b and non-terminals S, B . State q_0 is the start state. The transitions are:

- $q_0 \xrightarrow{a} q_2$
- $q_0 \xrightarrow{S} q_1$
- $q_1 \xrightarrow{S} q_0$
- $q_2 \xrightarrow{a} q_3$
- $q_2 \xrightarrow{B} q_3$
- $q_3 \xrightarrow{a} q_2$
- $q_3 \xrightarrow{S} q_5$
- $q_5 \xrightarrow{S} q_3$
- $q_2 \xrightarrow{b} q_4$
- $q_4 \xrightarrow{b} q_4$
- $q_4 \xrightarrow{B} q_6$
- $q_6 \xrightarrow{B} q_4$

The shift/reduce table at the bottom is:

	a	b	\$	B	S
0	s2				1
1			acc		
2	r4	s4	r2	3	
3	s2				5
4	r4	s4		6	
5			r1		
6	r3				

Lösung zu c:

Nr.	Stack	Resteingabe	Aktion
1	0	aba	s2
2	0a2	ba	s4
3	0a2b4	a	r4
4	0a2b4B6	a	r3
5	0a2B3	a	s2
6	0a2B3a2		r2
7	0a2B3S5		r1
8	0S1		accept

Punkte von 10

Aufgabe 5 (3 + 7 Punkte)

- Bestimmen Sie zur SPL-Prozedur p das Frame-Layout für den Aktivierungsrahmen: Bestandteile in der richtigen Reihenfolge mit Offsets zum Framepointer und Größen in Bytes
- Bestimmen Sie den ECO32-Assemblercode zu p. Die Prozedur *printi* erwartet einen Wertparameter vom Typ int. (SP=\$29, FP=\$25, RET=\$31, verfügbare Register: \$8-\$15).

```

proc p (i:int, ref j:int) {
    var k: array[2] of int;
    k[1] := i+j;
    printi(j);
}

```

Frame-Layout

Adresse	Größe	Inhalt
FP-8	8	k
FP-12	4	FP alt
FP-16	4	RETURN alt
FP-20	4	Arg.1 für printi

Puck-Assemblercode:

```

p:
SUBC $31 $31 20
ADDC $8 $31 8
STW $29 $8
ADDC $29 $31 20
ADDC $8 $29 -16
STW $30 $8
ADDC $8 $29 -8
SETW $9 1
SETW $10 2
LTU $10 $9 $10
BRF $10 indexError
SETW $10 4
MULU $9 $9 $10
ADD $8 $8 $9
ADDC $9 $29 0
LDW $9 $9
ADDC $10 $29 4
LDW $10 $10
LDW $10 $10
ADD $9 $9 $10
STW $9 $8
ADDC $8 $29 4
LDW $8 $8
LDW $8 $8
ADDC $9 $31 0
STW $8 $9
CALL $30 printi
ADDC $8 $29 -16
LDW $30 $8
ADDC $8 $31 8
LDW $29 $8
ADDC $31 $31 20
JMPR $30

```

ECO32-Assemblercode:

```

.export p
p:
sub $29,$29,20 ; allocate frame
stw $25,$29,8 ; save old frame pointer
add $25,$29,20 ; setup new frame pointer
stw $31,$25,-16 ; save return register
add $8,$25,-8
add $9,$0,1
add $10,$0,2
bgeu $9,$10,_indexError
mul $9,$9,4
add $8,$8,$9
add $9,$25,0
ldw $9,$9,0
add $10,$25,4
ldw $10,$10,0
ldw $10,$10,0
add $9,$9,$10
stw $9,$8,0
add $8,$25,4
ldw $8,$8,0
ldw $8,$8,0
stw $8,$29,0 ; store arg #0
jal printi
ldw $31,$25,-16 ; restore return register
ldw $25,$29,8 ; restore old frame pointer
add $29,$29,20 ; release frame
jr $31 ; return

```