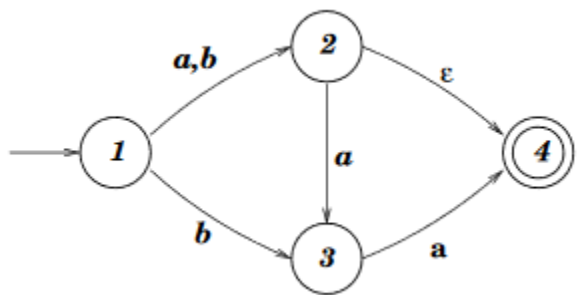




**Aufgaben aus
früheren
Compilerbau-
Klausuren**

Aufgabe 1

- a) Vereinfachen Sie den regulären Ausdruck $(a|\epsilon)^+bc | (aab)c | bc | bcbc$
- b) Betrachten Sie den regulären Ausdruck $r = b | ba | baa | (a|b) (a|\epsilon)$ und den durch das nachfolgende Zustandsübergangsdiagramm definierten NEA A:



Beweisen Sie, dass r und A nicht äquivalent sind.

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

Lösung Aufgabe 1

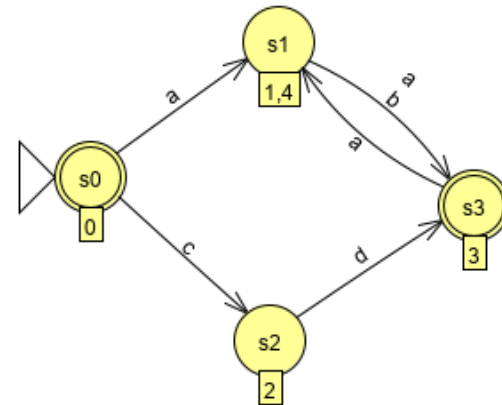
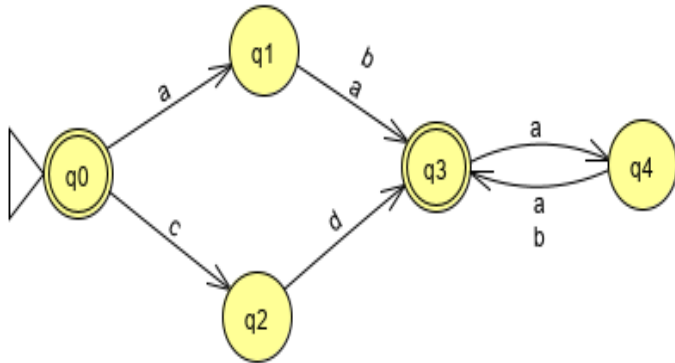
a) $(a^* \mid bc) bc$

b) r und A sind äquivalent genau dann, wenn $L(r)=L(A)$.

Das Wort aaa wird vom Automaten erkannt, passt aber zu keiner der Alternativen des regulären Ausdrucks.

Oder: Das Wort aa wird vom regulären Ausdruck erkannt, aber nicht vom Automaten.

c)



PS: Bei diesem Automaten kann man noch die Zustände $q1$ und $q4$ zusammenfassen. Dann entsteht der rechte (äquivalente) Automat.

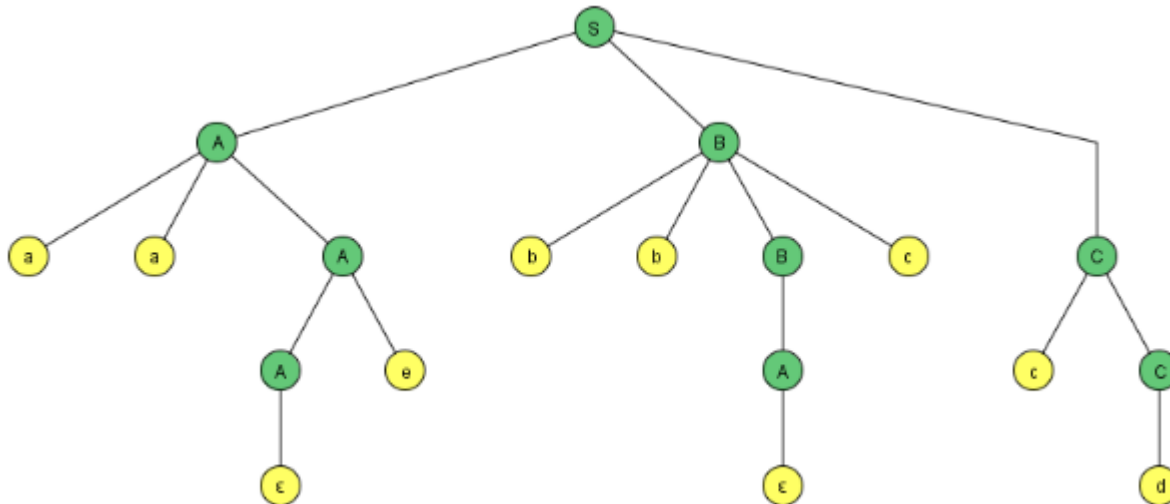
Aufgabe 2

- a) Gegeben sei folgende kontextfreie Grammatik G
- $$S \rightarrow ABC$$
- $$A \rightarrow aaA \mid Ae \mid \varepsilon$$
- $$B \rightarrow bbBc \mid A$$
- $$C \rightarrow cC \mid d$$
- Geben Sie drei Wörter aus $L(G)$ mit jeweils maximal 2 Zeichen an.
- b) Geben Sie eine Rechtsableitung und den Ableitungsbaum zu $aaebbccd$ an.
- c) Beweisen Sie, dass die Grammatik mehrdeutig ist.
- d) Bestimmen Sie $FIRST(S)$.
- e) Bestimmen Sie $Follow(A)$.
- f) Was steht in der $LL(1)$ -Parsertabelle in dem Eintrag zu A und e ?

Lösung Aufgabe 2 a-c)

a) d, cd, ed

b) $S \rightarrow ABC \rightarrow ABcC \rightarrow Abcd \rightarrow AbbBccd \rightarrow AbbAccd \rightarrow Abbcccd \rightarrow aaAbbcccd \rightarrow aaAebbccd \rightarrow aaebbccd$



c) Eine Grammatik G ist mehrdeutig, wenn es ein Wort aus $L(G)$ gibt, zu dem es zwei verschiedene Ableitungsbäume gibt. Dies ist für $w=ed$ der Fall:
 $S \rightarrow ABC \rightarrow ABd \rightarrow AAd \rightarrow AAed \rightarrow Aed \rightarrow ed$, bzw.
 $S \rightarrow ABC \rightarrow ABd \rightarrow AAd \rightarrow Ad \rightarrow Aed \rightarrow ed$

Lösung Aufgabe 2d-f)

d) $\text{First}(A) = \{a, e, \epsilon\}$

$$\text{First}(B) = \{a, b, e, \epsilon\}$$

$$\text{First}(C) = \{c, d\}$$

$$\Rightarrow \text{First}(S) = \{a, b, c, d, e\}$$

e) $\text{Follow}(C) = \{\$ \}$

$$\text{Follow}(B) = \{c, d\}$$

$$\Rightarrow \text{Follow}(A) = \{a, b, c, d, e\}$$

f) In Frage kommen die drei Regeln für das Nichtterminalsymbol A:

Regel $A \rightarrow Ae$: Da $e \in \text{First}(Ae)$, muss diese Regel in $\text{TAB}[A, e]$ sein.

Regel $A \rightarrow aaE$: e ist nicht in $\text{First}(aaE)$, daher ist diese Regel irrelevant.

Regel $A \rightarrow \epsilon$: Da $e \in \text{Follow}(A)$, wird auch $A \rightarrow \epsilon$ in $\text{TAB}[A, e]$ eingetragen.

In $\text{TAB}[A, e]$ müssen daher $A \rightarrow Ae$ und $A \rightarrow \epsilon$ eingetragen werden.

[Anmerkung: Die Grammatik ist somit keine LL(1)-Grammatik.]

Aufgabe 3 (Klausur SS18)

- a) Bestimmen Sie zur nachfolgenden Grammatik G 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) $S \rightarrow aBS$

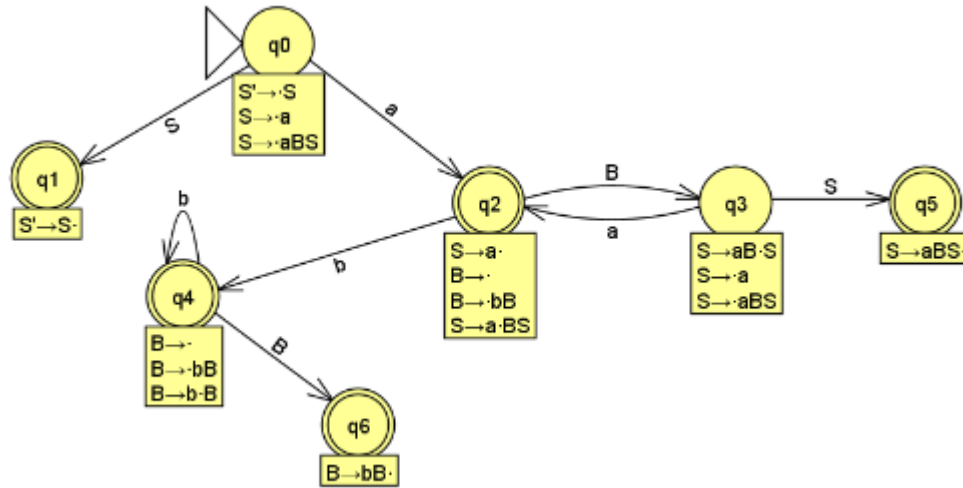
(2) $S \rightarrow a$

(3) $B \rightarrow bB$

(4) $B \rightarrow \varepsilon$

Lösung Aufgabe 3a-b)

a)



b)

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

Lösung Aufgabe 3c)

Zustandsstack	Symbolstack	Eingabewort	Aktion
0		aba	s2
02	a	ba	s4
024	ab	a	r4; goto 6
0246	abB	a	r3 ; goto 3
023	aB	a	s2
0232	aBa	\$	r2; goto 5
0235	aBS	\$	r1; goto 1
01	S	\$	accept

Aufgabe 4)

- a) 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
- b) 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);  
}
```

Lösung Aufgabe 4)

Adresse	Größe	Inhalt
FP-8	8	k
FP-12	4	Old FP
FP-16	4	ReturnAdr
FP-20	4	Argument 1 für printi

```

.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

```

Aufgabe 5)

Übersetzen Sie den folgenden regulären Ausdruck in eine rechtsrekursive Grammatik: $((xy^*x) \mid (yx^*y))$?

Lösung Aufgabe 5)

Zerlegen des Ausdrucks in drei Teile:

1. xy^*x wird erkannt durch:
 $A \rightarrow xB, B \rightarrow yB, B \rightarrow xC, C \rightarrow \varepsilon$
2. yx^*y wird erkannt durch:
 $K \rightarrow yL, L \rightarrow xL, L \rightarrow yM, M \rightarrow \varepsilon$
3. Das Fragezeichen entspricht:
 $T \rightarrow \varepsilon$

Insgesamt:

- $S \rightarrow A, S \rightarrow K, S \rightarrow T,$
 $A \rightarrow xB, B \rightarrow yB, B \rightarrow xC, C \rightarrow \varepsilon$
 $K \rightarrow yL, L \rightarrow xL, L \rightarrow yM, M \rightarrow \varepsilon$
 $T \rightarrow \varepsilon$