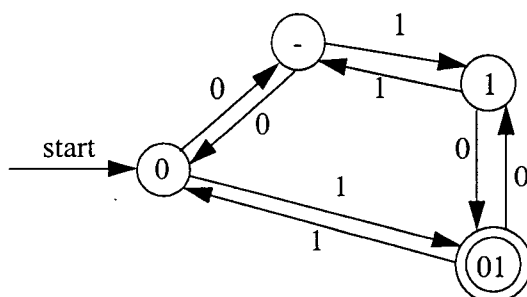


**Lösungsvorschläge
zur Klausur
„1810 Übersetzerbau“
21. Februar 2004**

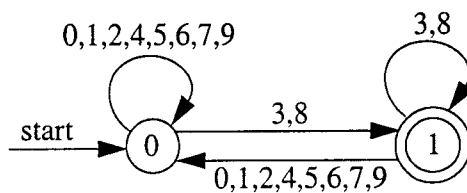
Aufgabe 1

(a)

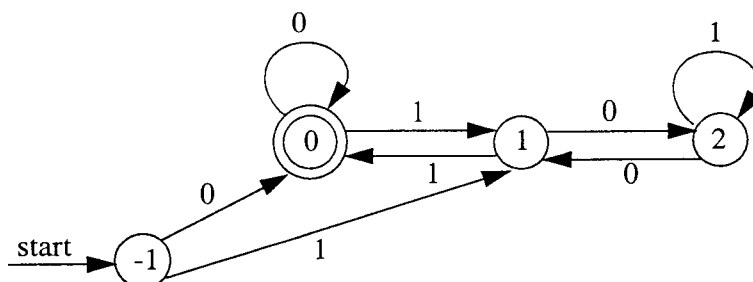
Ein die Sprache L_1 akzeptierender, endlicher Automat ist: $M_1 = (\{-, 0, 1, 01\}, \{0, 1\}, \delta_1, 0, \{01\})$, mit $\delta_1 =$ 

Der Name eines Zustandes gibt an, für welche Zeichen die durch die Sprache gegebene Bedingung erfüllt ist.

(b)

Die Sprache L_2 wird von dem endlichen Automaten M_2 erkannt: $M_2 = (\{0, 1\}, \{0, \dots, 9\}, \delta_2, 0, \{1\})$, mit $\delta_2 =$ 

(c)

Die Sprache L_3 wird durch den Automaten M_3 erkannt: $M_3 = (\{-1, 0, 1, 2\}, \{0, 1\}, \delta_3, -1, \{0\})$, mit $\delta_3 =$ 

Beschreibung: Der Zustand -1 dient dazu, das leere Wort nicht zuzulassen. Die anderen drei Zustände sind jeweils mit dem Wert der bisher gelesenen Zahl Modulo 3 gekennzeichnet. Die Übergänge sind durch folgende Tabelle ersichtlich:

Zustand	Wert von w	Übergang 0	Übergang 1
0	$x*3$	$w'=(x*3)*2 = y*3$	$w'=(x*3)*2+1 = y*3+1$
1	$x*3+1$	$w'=(x*3+1)*2 = y*3+2$	$w'=(x*3+1)*2+1 = z*3$
2	$x*3+2$	$w'=(x*3+2)*2 = z*3+1$	$w'=(x*3+2)*2+1 = z*3+2$

Aufgabe 2

(a)

In den Produktionen für die Nichtterminale A, B, C tritt direkte Linksrekursion auf. Die Beseitigung ergibt eine Grammatik mit den folgenden Produktionen:

$$\begin{aligned}
 S &\rightarrow ABA \\
 A &\rightarrow \mathbf{a}AdA' \mid \mathbf{g}A' \\
 A' &\rightarrow \mathbf{ac}A' \mid \mathbf{aa}A' \mid \varepsilon \\
 B &\rightarrow \mathbf{fc}B' \mid \mathbf{CB}' \\
 B' &\rightarrow \mathbf{h}B' \mid \varepsilon \\
 C &\rightarrow \mathbf{c}C' \\
 C' &\rightarrow \mathbf{CC}' \mid \varepsilon
 \end{aligned}$$

Die Grammatik G' erhält man durch Linksfaktorisierung für das neue Nichtterminal A' .

$$G' = (\{S, A, A', A'', B, B', C, C'\}, \{\mathbf{a}, \mathbf{c}, \mathbf{f}, \mathbf{g}, \mathbf{h}\}, P, S),$$

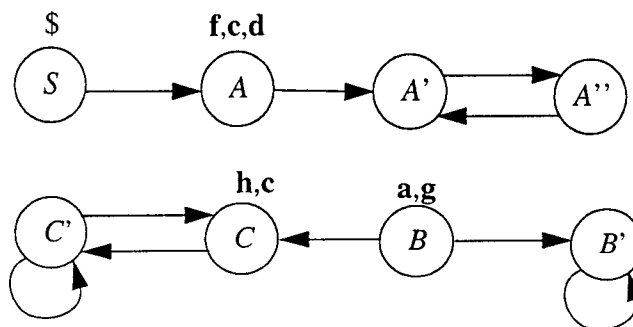
$$\text{mit } P = \left\{ \begin{array}{l}
 S \rightarrow ABA \\
 A \rightarrow \mathbf{a}AdA' \mid \mathbf{g}A' \\
 A' \rightarrow \mathbf{a}A'' \mid \varepsilon \\
 A'' \rightarrow \mathbf{c}A' \mid \mathbf{a}A' \\
 B \rightarrow \mathbf{fc}B' \mid \mathbf{CB}' \\
 B' \rightarrow \mathbf{h}B' \mid \varepsilon \\
 C \rightarrow \mathbf{c}C' \\
 C' \rightarrow \mathbf{CC}' \mid \varepsilon
 \end{array} \right\}.$$

(b)

Die Berechnung der FIRST-Mengen und der initialen Steuermengen ergibt:

$\{a, g\}$	$S \rightarrow$	ABA	$\{a, g\}$	(1)
$\{a, g\}$	$A \rightarrow$	$aAdA' \mid$	$\{a\}$	(2)
		gA'	$\{g\}$	(3)
$\{a, \varepsilon\}$	$A' \rightarrow$	$aA'' \mid$	$\{a\}$	(4)
		ε	$\{\varepsilon\}$	(5)
$\{a, c\}$	$A'' \rightarrow$	$cA' \mid$	$\{c\}$	(6)
		aA'	$\{a\}$	(7)
$\{f, c\}$	$B \rightarrow$	$fcB' \mid$	$\{f\}$	(8)
		CB'	$\{c\}$	(9)
$\{h, \varepsilon\}$	$B' \rightarrow$	$hB' \mid$	$\{h\}$	(10)
		ε	$\{\varepsilon\}$	(11)
$\{c\}$	$C \rightarrow$	cC'	$\{c\}$	(12)
$\{c, \varepsilon\}$	$C' \rightarrow$	$CC' \mid$	$\{c\}$	(13)
		ε	$\{\varepsilon\}$	(14)

Die FOLLOW-Mengen lassen sich aus dem folgenden Graphen ablesen:



$$\text{FOLLOW}(S) = \{\$ \}$$

$$\text{FOLLOW}(A) = \text{FOLLOW}(A') = \text{FOLLOW}(A'') = \{\$, f, c, d\}$$

$$\text{FOLLOW}(B') = \text{FOLLOW}(B) = \{a, g\}$$

$$\text{FOLLOW}(C) = \text{FOLLOW}(C') = \{a, g, h, c\}$$

(c)

Zur Angabe der Steuermengen verwenden wir die Numerierung der Produktionen aus dem Aufgabenteil (b).

$$\begin{array}{llll}
 D(1) = \{a, g\} & D(2) = \{a\} & D(3) = \{g\} & D(4) = \{a\} \\
 D(5) = \{f, c, d\} & D(6) = \{c\} & D(7) = \{a\} & D(8) = \{f\} \\
 D(9) = \{c\} & D(10) = \{h\} & D(11) = \{a, g\} & D(12) = \{c\} \\
 D(13) = \{c\} & D(14) = \{a, g, h, c\} & &
 \end{array}$$

Die Grammatik ist nicht vom Typ LL(1), da die Steuermengen der Produktionen 13 und 14 nicht disjunkt sind.

Aufgabe 3

(a)

Der Ausdruck „**if a then if b then c else d**“ kann durch zwei verschiedene Ableitungen erzeugt werden:

$$\begin{aligned}
 S &\rightarrow \text{if } E \text{ then } S \text{ else } S \rightarrow \text{if } E \text{ then } S \text{ else } d \rightarrow \text{if } E \text{ then if } E \text{ then } S \text{ else } d \\
 &\rightarrow \text{if } E \text{ then if } E \text{ then } c \text{ else } d \rightarrow \text{if } E \text{ then if } b \text{ then } c \text{ else } d \\
 &\rightarrow \text{if } a \text{ then if } b \text{ then } c \text{ else } d
 \end{aligned}$$

oder

$$\begin{aligned}
 S &\rightarrow \text{if } E \text{ then } S \rightarrow \text{if } E \text{ then if } E \text{ then } S \text{ else } S \rightarrow \text{if } E \text{ then if } E \text{ then } S \text{ else } d \\
 &\rightarrow \text{if } E \text{ then if } E \text{ then } c \text{ else } d \rightarrow \text{if } E \text{ then if } b \text{ then } c \text{ else } d \\
 &\rightarrow \text{if } a \text{ then if } b \text{ then } c \text{ else } d
 \end{aligned}$$

(b)

Die LR(0)-Kollektion sieht folgendermaßen aus:

	$0 \xrightarrow{s} 1$
0: $S' \rightarrow \cdot S$ $S \rightarrow \cdot \text{if } E \text{ then } S$ $S \rightarrow \cdot \text{if } E \text{ then } S \text{ else } S$ $S \rightarrow \cdot c$ $S \rightarrow \cdot d$	1: $S' \rightarrow S \cdot$
$0 \xrightarrow{c} 2$	$0 \xrightarrow{d} 3$
2: $S \rightarrow c \cdot$	3: $S \rightarrow d \cdot$
4: $E \rightarrow a \cdot$	5: $E \rightarrow b \cdot$
$0 \xrightarrow{\text{if}} 6$	$6 \xrightarrow{E} 7$
6: $S \rightarrow \text{if } \cdot E \text{ then } S$ $S \rightarrow \text{if } \cdot E \text{ then } S \text{ else } S$ $E \rightarrow \cdot a$ $6 \xrightarrow{a} 4$ $E \rightarrow \cdot b$ $6 \xrightarrow{b} 5$	7: $S \rightarrow \text{if } E \cdot \text{ then } S$ $S \rightarrow \text{if } E \cdot \text{ then } S \text{ else } S$
$7 \xrightarrow{\text{then}} 8$	$8 \xrightarrow{s} 9$
8: $S \rightarrow \text{if } E \text{ then } \cdot S$ $S \rightarrow \text{if } E \text{ then } \cdot S \text{ else } S$ $S \rightarrow \cdot \text{if } E \text{ then } S$ $S \rightarrow \cdot \text{if } E \text{ then } S \text{ else } S$ $S \rightarrow \cdot c$ $S \rightarrow \cdot d$ $8 \xrightarrow{\text{if}} 6$ $8 \xrightarrow{c} 2$ $8 \xrightarrow{d} 3$	9: $S \rightarrow \text{if } E \text{ then } S \cdot$ $S \rightarrow \text{if } E \text{ then } S \cdot \text{ else } S$
$9 \xrightarrow{\text{else}} 10$	$10 \xrightarrow{s} 11$
10: $S \rightarrow \text{if } E \text{ then } S \text{ else } \cdot S$ $S \rightarrow \dots$ $10 \xrightarrow{\text{if}} 6$ $10 \xrightarrow{c} 2$ $10 \xrightarrow{d} 3$	11: $S \rightarrow \text{if } E \text{ then } S \text{ else } S \cdot$

(c)

Wie man leicht sieht, ist $FOLLOW(S) = \{\text{else}, \$\}$ und $FOLLOW(E) = \{\text{then}\}$. Numeriert man die Produktionen wie in der Aufgabenstellung, so erhält man die folgende Tabelle:

Zustand	Action								Goto	
	if	then	else	a	b	c	d	\$	S	E
0	s6					s2	s3		1	
1								acc		
2			r3					r3		
3			r4					r4		
4		r5								
5		r6								
6				s4	s5					7
7		s8								
8	s6					s2	s3		9	
9			s10/r1					r1		
10	s6					s2	s3		11	
11			r2					r2		

Im Zustand 9 tritt ein *shift/reduce*-Konflikt auf, der durch ein *shift* aufgelöst wird, um einem kompletten „if ... then ... else“-Ausdruck den Vorrang gegenüber einem „if ... then ...“-Ausdruck zu geben.

(d)

Stack	Eingabe	Aktion
0	if a then if b then c else d\$	s6
0 if 6	a then if b then c else d\$	s4
0 if 6 a 4	then if b then c else d\$	r5 E → a
0 if 6 E 7	then if b then c else d\$	s8
0 if 6 E 7 then 8	if b then c else d\$	s6
0 if 6 E 7 then 8 if 6	b then c else d\$	s5
0 if 6 E 7 then 8 if 6 b 5	then c else d\$	r6 E → b
0 if 6 E 7 then 8 if 6 E 7	then c else d\$	s8
0 if 6 E 7 then 8 if 6 E 7 then 8	c else d\$	s2

Stack	Eingabe	Aktion
0 if 6 E 7 then 8 if 6 E 7 then 8 c 2		else d\$ r3 S → c
0 if 6 E 7 then 8 if 6 E 7 then 8 S 9		else d\$ s10
0 if 6 E 7 then 8 if 6 E 7 then 8 S 9 else 10		d\$ s3
0 if 6 E 7 then 8 if 6 E 7 then 8 S 9 else 10 d 3		\$ r4 S → d
0 if 6 E 7 then 8 if 6 E 7 then 8 S 9 else 10 S 11		\$ r2 S → ...
0 if 6 E 7 then 8 S 9		\$ r1 S → ...
0 S 1		\$ acc

Aufgabe 4

1. Elimination toten Codes

Aus (2) folgt, daß die Zeilen (9) - (14) und (24) - (27) direkt gelöscht werden können. Selbiges gilt für die Zeilen (15) und (31). Da Zeile (2) nun nicht mehr gebraucht wird, wird diese Zeile ebenfalls gelöscht.

2. Konstantenpropagation und Konstantenfaltung

Aufgrund von Zeile (1) werden folgende Zeilen wie gezeigt ersetzt:

```
(32) for i := 0 to 100 do begin
(33)   for j := i to 100 do begin
(34)     z := z+n*i+10-x;
```

Zeile (1) kann nun gelöscht werden. Aus (3) folgt:

```
(4)   y := 2;
(6)   for i := 0 to 2 do begin
(7)     z := z+1
```

Die Zeilen (3) und (4) können nun gelöscht werden.

3. Schleifenfaltung

Die Schleife in (6) bis (8) kann aufgelöst werden. Wir führen zudem sofort eine Konstantenfaltung durch, so daß die Zeilen (6) bis (8) ersetzt werden durch

```
(6)   z := 3
```

Zeile (5) kann nun ebenfalls gelöscht werden.

4. Verlagerung von Invarianten

Die Variablenberechnung in (34) ist nur teilweise von den Schleifen in (32) und (34) abhängig. Daher werden die Zeilen (32) bis (36) ersetzt durch

```
(32) r := 10-x;
(33) for i := 0 to 100 do begin
(34)   s := n*i;
(35)   for j := i to 100 do begin
(36)     z := z+s+r
(37)   end
(38) end;
```

5. Code-Hoisting

Der Ausdruck $k+x$ kommt in den verschiedenen `if`-Zweigen mehrfach vor. Durch Faktorisierung werden die Zeilen (16) bis (30) durch die folgenden Zeilen ersetzt:

```
(16) w := k+x;
(17) if x < k then
(18)   x := w+1
(19) else
(20)   if x > k then
(21)     x := w-1
(22)   else
(23)     if x = k then
(24)       x := sqrt(w)
(25)     end
(26)   end
(27) end;
```

Als Ergebnis aller Optimierungsschritte erhalten wir schließlich das nachstehende Programmstück.

```
(1) z := 3;
(2) w := k+x;
(3) if x < k then
(4)   x := w+1
(5) else
(6)   if x > k then
(7)     x := w-1
(8)   else
(9)     if x = k then
(10)      x := sqrt(w)
(11)    end
(12)  end
(13) end;
(14) r := 10-x;
(15) for i := 0 to 100 do begin
(16)   s := n*i;
(17)   for j := i to 100 do begin
(18)     z := z+s+r
(19)   end
(20) end;
```