Technische Universität Chemnitz Sommersemester 2002
Professur Theoretische Informatik
Prof. Dr. A. Goerdt, Dipl.-Inf. F. Schädlich, Dipl.-Inf. A. Lanka

5. Übung „Theoretische Informatik II“

Lösungsvorschläge

Aufgabe 1

DEA $M = (Z,\ \Sigma,\ \delta,\ z_1,\ E)$ mit

$$Z = \{z_1, z_2, \ldots, z_n\}, \quad |Z| = n,$$
$$E = \{z_{e_1}, z_{e_2}, \ldots, z_{e_m}\} \subseteq Z, \quad |E| = m$$

$$R^k_{i,j} = \{\omega\colon \omega \in \Sigma^*, \hat{\delta}(z_i, \omega) = z_j \text{ und } \text{für jedes echte Präfix } \alpha \subset \omega \text{ gilt } \hat{\delta}(z_i, \alpha) = z_l \text{ mit } l \le k\}$$

Stellt man $R^k_{i,j}$ als regulären Ausdruck dar, so findet man folgende rekursive Beschreibung:

$$R^0_{i,j} = \begin{cases} \{\epsilon\} \cup \{x\colon x \in \Sigma \text{ und } \delta(z_i, x) = z_i\} & \text{für } i = j \\ \{x\colon x \in \Sigma \text{ und } \delta(z_i, x) = z_j\} & \text{sonst} \end{cases}.$$
$$R^{k+1}_{i,j} = R^k_{i,j} \mid R^k_{i,k+1} \left(R^k_{k+1,k+1}\right)^* R^k_{k+1,j}$$

Die vom Automaten M akzeptierte Sprache L läßt sich nun als

$$L = R^n_{1,e_1} \mid R^n_{1,e_2} \mid \ldots \mid R^n_{1,e_m}$$

darstellen.

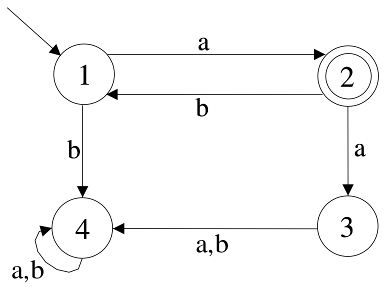
Für den Automaten der Aufgabe findet man nun:

	$k=0$	$k=1$	$k=2$	$k=3$
$R^k_{1,1}$	ϵ	ϵ	$(aa)^*$	
$R^k_{1,2}$	a	a	$a(aa)^*$	$a(aa)^* \mid a^*b\big((a\|b)a^*b\big)^*(a\|b)(aa)^*$
$R^k_{1,3}$	b	b	a^*b	$a^*b\big((a\|b)a^*b\big)^*$
$R^k_{2,1}$	a	a	$a(aa)^*$	
$R^k_{2,2}$	ϵ	$\epsilon \mid aa$	$(aa)^*$	
$R^k_{2,3}$	b	$b \mid a$	a^*b	
$R^k_{3,1}$	$\emptyset$	$\emptyset$	$(a \mid b)(aa)^*a$	
$R^k_{3,2}$	$a \mid b$	$a \mid b$	$(a \mid b)(aa)^*$	
$R^k_{3,3}$	ϵ	ϵ	$\epsilon \mid (a\|b)a^*b$	

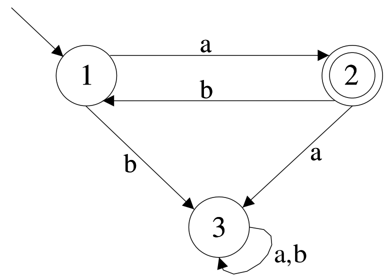
$$L = a(aa)^* \mid a^*b\big((a|b)a^*b\big)^*(a|b)(aa)^* \mid a^*b\big((a|b)a^*b\big)^*$$

Aufgabe 2

a) Zustandsgraph:



b) Minimalautomat:



Markierungsalgorithmus:

1. Schritt: Endzustand und ein Nichtendzustand gehören nicht zusammen, also markieren
2. Schritt: Alle Zustandspaare (a, b) markieren, die wir nicht verschmelzen können.
Das Paar (a, b) kann nicht „verschmolzen" werden, wenn es ein Eingabewort w gibt, so dass bei der Eingabe von w einmal ein Endzustand erreicht wird und einmal nicht. Dies wird dynamisch aufgebaut:
Gibt es ein Wort w, so gibt es bei einem weiteren Durchlauf einen Buchstaben x des Alphabets, so dass der Tabellenplatz $(\delta(a, x), \delta(b, x))$ bereits markiert ist.
$\Rightarrow$ FOR EACH $x \in \Sigma$ DO
IF $(\delta(a, x), \delta(b, x))$ markiert THEN markiere (a, b)
3. Schritt: Wiederhole Schritt 2, bis sich in der Tabelle nichts mehr ändert

Zwei Zustände können zusammengefaßt werden, wenn das Feld in der Tabelle nicht markiert ist.

	1	2	3	4
1	-	-	-	-
2	$\times_0$	-	-	-
3	$\times_1$	$\times_0$	-	-
4	$\times_2$	$\times_0$	o	-

Bemerkung: Die Markierungen „$\times$" in der Tabelle sind zur besseren Übersicht in der Reihenfolge ihres Eintragens indiziert.

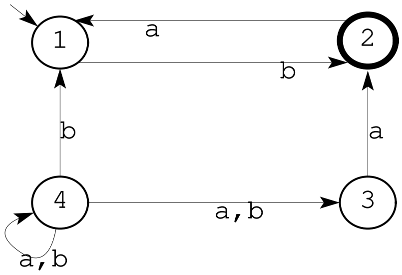
Übersicht:

Zustandspaar (p, q) in Tabelle	$x \in \Sigma$	$(\delta(p), \delta(q))$	$(\delta(p, x), \delta(q, x))$ in der Tabelle markiert?	(p, q) markieren im Schritt
1,3	a	2,4	ja	1
1,4	a	2,4	ja	2
3,4	a	4,4	nein	-
3,4	b	4,4	nein	-
3,4	a	4,4	nein	-
3,4	b	4,4	nein	-

keine Änderung in der Tabelle mehr $\rightarrow$ Abbruch. $\Rightarrow$ Zustände 3 und 4 können „verschmolzen" werden.

c) In (b) wird für jedes unmarkierte Paar in jeder Runde getestet, ob es markiert werden kann. Diese Vorgehensweise ist offensichtlich nicht optimal. Sinnvoller ist es, bei der Markierung eines Paares zu untersuchen, welche Zustandspaare auf Grund dieser Markierung ebenfalls markiert werden müssen:

Wir drehen den Automatengraphen um, d. h. wir tauschen die Kantenrichtung. Dies führt zu folgendem Automaten:



Jetzt markieren wir alle Paare, die einen Endzustand und einen Nicht-Endzustand enthalten. Ausgehend von diesen Paaren markieren wir alle Paare, die von diesen erreicht werden können. Dies wiederholen wir solange, bis kein neues Paar markiert werden kann. Das besondere bei diesem Verfahren ist, daß jedes Knotenpaar nur einmal „expandiert“ werden muß.

In unserem neuen Graphen gibt es genau $|\Sigma| \cdot n$ viele Kanten. Da wir bei der Suche nach neuen Paaren jeweils zwei Kanten untersuchen. Bricht die Suche nach $O((|\Sigma| \cdot n)^2)$ vielen Schritten ab.

Wir bilden nun einen neuen ungerichteten Graphen G, der als Knoten genau die Zustände enthält. Für jedes markierte Paar (i, j) tragen wir in G die Kante $\{i, j\}$ ein. Dafür brauchen wir nicht mehr Aufwand als $O(n^2)$. Wenn wir damit fertig sind, invertieren wir den Graphen. Wir machen also aus Kanten Nicht-Kanten und aus Nicht-Kanten werden Kanten.

Jede Zusammenhangskomponente in G^I entspricht einer Menge von Zuständen, die verschmolzen werden können.

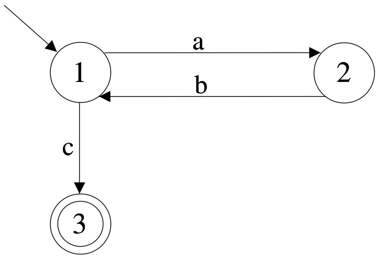
Insgesamt haben wir einen Aufwand von $O((|\Sigma| \cdot n)^2)$. Wenn wir davon ausgehen, daß das Alphabet eine konstante Größe hat, verschwindet das $|\Sigma|^2$ im O und der Aufwand beträgt $O(n^2)$.

d)

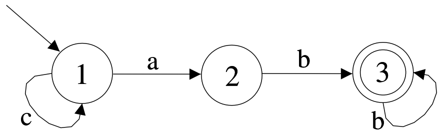
$$
\begin{aligned}
L &= \{a(ba)^n : n \geq 0\} \\
[1] &= \{\varepsilon, ab, abab, ...\} = \{(ab)^i : i \geq 0\} \\
[2] &= \{a, aba, ababa, ...] = \{a(ba)^i : i >= 0\} \\
[3] &= \{\text{alle Wörter, die mit } b \text{ anfangen oder } aa \text{ bzw. } bb \text{ enthalten}\} \\
&= \{bx, xaay, xbby : x, y \in \Sigma^*\}
\end{aligned}
$$

Aufgabe 3

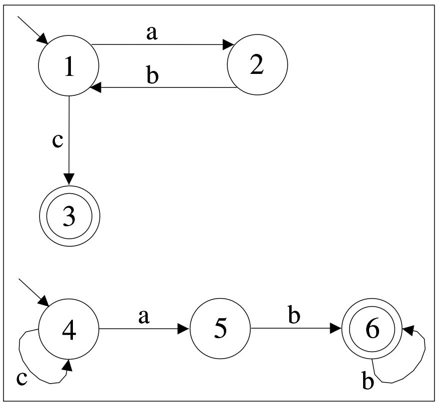
A1: $L(A_1) = L_1$



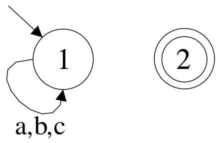
A2: $L(A_2) = L_2$



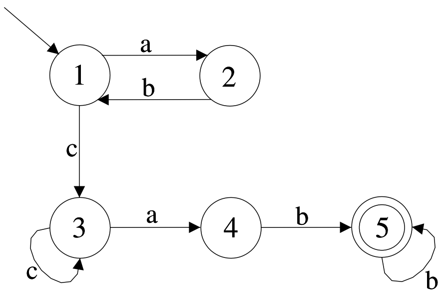
A3: $L(A_3) = L_1 \cup L_2$
Der Automat A_3 bildet die Vereinigung von A_1 und A_2



A4: $L(A_4) = L_1 \cap L_2$
Der Automat A_4 bildet die Schnittmenge von A_1 und A_2. Da von A_1 und A_2 keine gleichen Wörter erkannt werden, ist $L(A_4) = \oslash$



A5: $L(A_5) = L_1 L_2$



A6: $L(A_6) = L_1^*$