Universität Karlsruhe (TH)
Lehrstuhl Informatik für Ingenieure und Naturwissenschaftler
Dr. Thomas Worsch
Matthias Schulz

9. Juni 2008

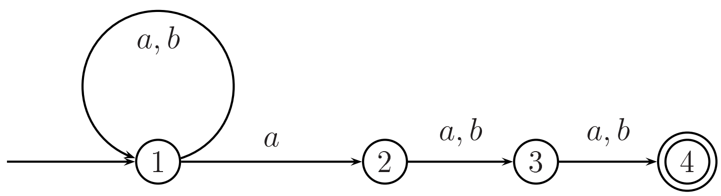
Einführung in die Informatik

Lösungen zu Übungsblatt 7

– Endliche Automaten –

Aufgabe 1:

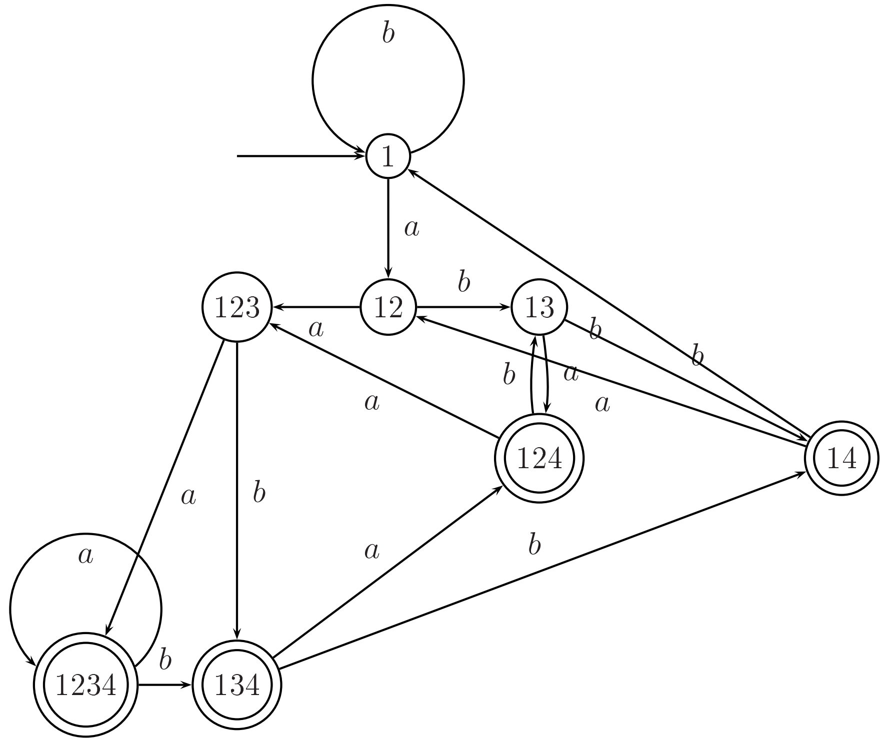
a) Ein nichtdeterministischer Automat A_3, der $\mathsf{L_3}$ erkennt, hat folgende Gestalt:



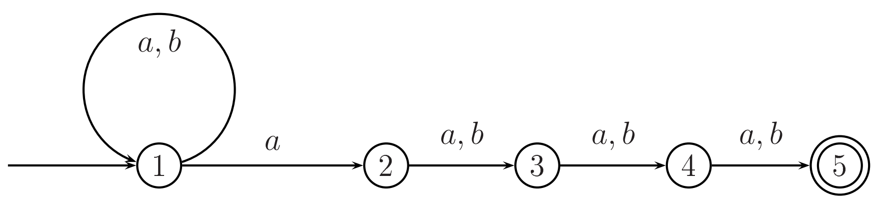
Bei der Konstruktion des deterministischen Automaten betrachten wir die nach einander auftretenden Teilmengen von Z, die wir beim Durchlaufen des nichtdeterministischen Automaten erhalten:

Teilmenge	a	b	Akzeptierender Zustand?
{1}	{1, 2}	{1}	Nein
{1, 2}	{1, 2, 3}	{1, 3}	Nein
{1,2,3}	{1, 2, 3, 4}	{1, 3, 4}	Nein
{1, 3}	{1, 2, 4}	{1, 4}	Nein
{1, 2, 3, 4}	{1, 2, 3, 4}	{1, 3, 4}	Ja
{1, 3, 4}	{1, 2, 4}	{1, 4}	Ja
{1, 2, 4}	{1, 2, 3}	{1, 3}	Ja
{1, 4}	{1, 2}	{1}	Ja

Der Automat hat somit folgende (etwas unübersichtliche) Form:



b) Ein nichtdeterministischer Automat A_4, der $\mathsf{L_4}$ erkennt, hat folgende Gestalt:



Konstruktion des deterministischen Automaten:

Teilmenge	a	b	Akzeptierender Zustand?
{1}	{1, 2}	{1}	Nein
{1, 2}	{1, 2, 3}	{1, 3}	Nein
{1,3}	{1, 2, 4}	{1, 4}	Nein
{1, 4}	{1, 2, 5}	{1, 5}	Nein
{1, 5}	{1, 2}	{1}	Ja
{1, 2, 3}	{1, 2, 3, 4}	{1, 3, 4}	Nein
{1, 2, 4}	{1, 2, 3, 5}	{1, 3, 5}	Nein
{1, 2, 5}	{1, 2, 3}	{1, 3}	Ja
{1, 3, 4}	{1, 2, 4, 5}	{1, 4, 5}	Nein
{1, 3, 5}	{1, 2, 4}	{1, 4}	Ja
{1, 4, 5}	{1, 2, 5}	{1, 5}	Ja
{1, 2, 3, 4}	{1, 2, 3, 4, 5}	{1, 3, 4, 5}	Nein
{1, 2, 3, 5}	{1, 2, 3, 4}	{1, 3, 4}	Ja
{1, 2, 4, 5}	{1, 2, 3, 5}	{1, 3, 5}	Ja
{1, 3, 4, 5}	{1, 2, 4, 5}	{1, 4, 5}	Ja
{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}	{1, 3, 4, 5}	Ja

Dadurch ist der deterministsiche Automat festgelegt. (Eine graphische Darstellung ist sehr kompliziert!)

c) $N(i)$: Man betrachte einen Durchlauf des NEA, der das Wort ab^{i-1} erkennt, also in einem akzeptierenden Zustand endet. Angenommen, ein Zustand s wird dabei zweimal besucht.

1. Fall: s ist nicht der Anfangszustand.
Seien ab^k und ab^l mit $0 \leq k < l \leq i-1$ die Präfixe von ab^{i-1}, für die der Automat jeweils den Zustand s annimmt. Dann könnte der Automat, indem er nach Einlesen von ab^l wie bei ab^k mit der Berechnung fortfährt, auch $ab^{l+((i-1)-k)} = ab^{i-1+(l-k)}$ erkennen, dessen i-letzes Zeichen jedoch b ist $\rightarrow$ Widerspruch.

2. Fall: s ist der Anfangszustand.
Sei ab^k ein Präfix von ab^{i-1}, für das der Automat den Zustand s annimmt. Ausgehend vom Startzustand kann dann somit das Wort $b^{(i-1)-k}$ erkannt werden, wenn man der weiteren Berechnung des Automaten bei der Erkennung von ab^{i-1} folgt. Dies ist ein Widerspruch, da $b^{i-1-k} \notin L_i$.

Somit müssen während der Erkennung von ab^{i-1} lauter verschiedene Zustände durchlaufen werden; dies bedeutet, dass der Automat mindestens $i+1$ Zustände haben muss.

Andererseits gilt für den NEA A_i mit den Zuständen $S, 1, \ldots i$, der Überführungsrelation Δ mit
$\Delta = \{(x, S, S) \mid x \in \{a, b\}\} \cup \{(a, S, 1)\} \cup \{x, k, k+1 \mid x \in \{a, b\}, 1 \leq k \leq i-1\}$,
sowie dem akzeptierenden Zustand i, dass $L(A_i) = \mathsf{L}_\mathsf{i}$ gilt.

$D(i)$: Man betrachte alle 2^i Wörter über $\{a, b\}$ der Länge i. Angenommen, der

Automat $D(i)$ würde für zwei dieser Wörter im gleichen Zustand s enden. Seien $w_1 = x_1 \cdots x_i \neq w_2 = y_1 \cdots y_i$ diese Wörter und $k \in \{1, \dots i\}$ so gewählt, dass $x_k \neq y_k$ gilt. (Solch ein k muss es geben, da ansonsten $w_1 = w_2$ gelten würde.)

Sei nun ohne Beschränkung der Allgemeinheit $x_k = a$ und $y_k = b$. Da das Wort $x_1 \cdots x_i b^{k-1}$ nach Voraussetzung von $D(i)$ erkannt wird (i-letztes Zeichen ist $x_k = a$), wird auch $y_1 \cdots y_i b^{k-1}$ von $D(i)$ erkannt, da jeweils nach Lesen von w_1 und w_2 die gleichen Zustände durchlaufen werden.

Da das i-letzte Zeichen von $y_1 \cdots y_i b^{k-1}$ jedoch $y_k = b$ ist, ist dies ein Widerspruch $\Rightarrow D(i)$ kann bei keinen zwei Wörtern mit i Zeichen in den gleichen Zustand geraten $\Rightarrow D(i)$ besitzt mindestens 2^i Zustände.

Andererseits erkennt der Automat $D(i)$ mit Zustandsmenge $Z = \{a, b\}^i$, Startzustand b^i, Überführungsfunktion δ mit $\delta(y, xw) = wy$, wobei $x, y \in \{a, b\}$ und $w \in \{a, b\}^{i-1}$, dessen Zustände genau dann akzeptierend sind, wenn sie mit a beginnen, genau die Wörter in $\mathsf{L_i}$ und besitzt 2^i Zustände.

Aufgabe 2: Es gelte $A = (X, Z_1, \Delta_1, S_1, F_1), B = (X, Z_2, \Delta_2, S_2, F_2)$, wobei $Z_1 \cap Z_2 = \emptyset$ gilt.

a) Idee: Einer der beiden Automaten wird anfangs "gewählt" und durchlaufen; dazu "schreibt" man beide Automaten neben einander und nimmt jeden Startzustand der beiden Automaten als möglichen Startzustand.

   Formal: $C = (X, Z_1 \cup Z_2, \Delta_1 \cup \Delta_2, S_1 \cup S_2, F_1 \cup F_2)$.

b) Idee: Verbinde alle akzeptierenden Zustände von A so, wie die Startzustände von B verbunden sind (wenn Startzustand s von B beim Lesen von x in z übergeht, so geht im neuen Automaten jeder akzeptierende Zustand f von A beim Lesen von x in z über); falls ein Startzustand von B akzeptierender Zustand ist, so sind die akzeptierenden Zustände von A auch akzeptierende Zustände des neuen Automaten, sonst nicht. Die akzeptierenden Zustände von B sind akzeptierende Zustände von C. (Erst wird A durchlaufen, danach B).

   Formal: $C = (X, Z_1 \cup Z_2, \Delta', S_1, F')$ mit $\Delta' = \Delta_1 \cup \Delta_2 \cup$ $\{(x, f, z) \mid x \in X, f \in F_1, \exists s \in S_2 : (x, s, z) \in \Delta_2\}$ und

   $$F' = \begin{cases} F_2 \cup F_1 & \text{falls } F_2 \cap S_2 \neq \emptyset \\ F_2 & \text{falls } F_2 \cap S_2 = \emptyset \end{cases}$$

   Die Startzustände des zweiten Automaten lassen sich im Allgemeinen nicht löschen, sondern müssen beibehalten werden, wenn sie von anderen Zuständen des zweiten Automaten aus wieder erreichbar sind.

c) Idee: Worte "rückwärts" einlesen, beginnend mit den akzeptierenden Zuständen und endend mit den Startzuständen. Wenn der Automat im Zustand s beim Lesen von

x in Zustand s' übergehen kann, so kann der neue Automaten im Zustand s' beim Lesen von x in s übergehen.

Formal: $C = (X, Z_1, \Delta', F_1, S_1)$ mit $(x, s, s') \in \Delta' \Leftrightarrow (x, s', s) \in \Delta_1$.

d) Idee: Man lässt beide Automaten gleichzeitig laufen und akzeptiert ein Wort, wenn beide Automaten in einem akzeptierenden Zustand enden. Dazu muss man sich immer das Paar der aktuellen Zustände "merken".

   Formal: $C = (X, Z_1 \times Z_2, \Delta', S_1 \times S_2, F_1 \times F_2)$ mit $(x, (z_1, z_2), (z_1', z_2')) \in \Delta' \Leftrightarrow (x, z_1, z_1') \in \Delta_1, (x, z_2, z_2') \in \Delta_2$.

e) Idee: Man vertauscht akzeptierende und nicht akzeptierende Zustände.

   Aber: Bei einem nichtdeterministischen Automaten muss der Automat bei einem zur erkannten Sprache gehörenden Wort nicht unbedingt in einem akzeptierenden Zustand enden.

   Daher: Erst deterministisch machen, dann vertauschen. Damit sind genau die Teilmengen von $Z_1 \setminus F_1$ akzeptierende Zustände.

   Formal: $C = (X, 2^{Z_1}, \delta, \{S_1\}, 2^{Z_1 \setminus F_1})$ mit $\delta(x, T) = \{z \in Z_1 \mid \exists z' \in T : (x, z', z) \in \Delta_1\}$.

Die Automaten in Teilaufgaben d) und e) sind deterministisch, wenn A und B deterministisch sind, für die anderen Automaten ist dies im Allgemeinen nicht der Fall:

a) C besitzt zwei Startzustände nach Konstruktion, ist also nicht deterministisch.

b) Von jedem akzeptierenden Zustand von A gehen Verbindungen zu Zuständen aus A und auch zu Zuständen aus B, möglicherweise auch bei gleicher Eingabe.

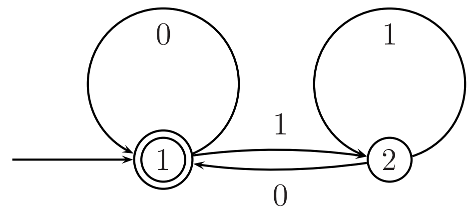
c) In A kann ein Zustand z von zwei Zuständen z_1 und z_2 aus erreicht werden, wenn ein bestimmtes Zeichen x eingelesen wird; in C ginge dann von z je ein Übergang zu z_1 und z_2, wenn x eingelesen wird. Somit ist C nicht deterministisch.

d) Wenn A und B deterministisch sind, so kann von jedem Paar (z_1, z_2) bei Eingabe von x nur das Paar $(\delta(x, z_1), \delta(x, z_2))$ erreicht werden. C ist somit deterministisch.

e) Die Konstrukion sorgt dafür, dass stets ein deterministischer Automat vorliegt.

Aufgabe 3:

a) Eine Zahl in Binärdarstellung ist genau dann durch 2 teilbar, wenn ihre letzte Stelle 0 ist. Der endliche Automat hat folgende Gestalt:



Ein regulärer Ausdruck, der T_2 beschreibt, ist $(1^*0)^*$.

b) Die Idee ist, sich immer zu merken, welchen Rest die Zahl, die durch die ersten k Ziffern beschrieben wird, bei der Division durch 3 hat. Bei den Übergängen betrachtet man nun, welchen Rest r bei Division durch 3 die Zahl hat, die sich durch Multiplikation mit 2 und Addition des $k+1$-ten Zeichen ergibt (Beispiel: $z(111) = 7$; $z(1110) = 14 = 2 \cdot 7 + 0$, $z(1111) = 15 = 2 \cdot 7 + 1$); dieser Rest lässt sich alleine durch Betrachtung von r bestimmen:

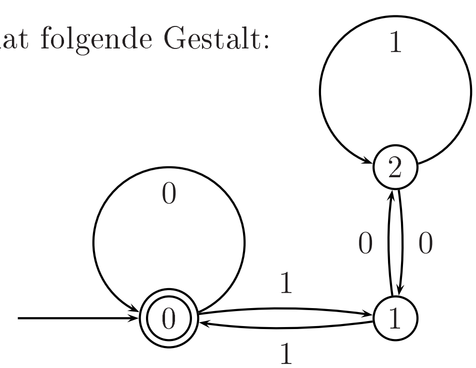
Sei x die Zahl, die durch die ersten k Zeichen dargestellt wird.

$r = 0 \Rightarrow \exists n \in \mathbb{N}_0 : x = 3n$
$\Rightarrow 2x + 0 = 6n \Rightarrow$ Rest 0,
$\Rightarrow 2x + 1 = 6n + 1 \Rightarrow$ Rest 1.

$r = 1 \Rightarrow \exists n \in \mathbb{N}_0 : x = 3n + 1$
$\Rightarrow 2x + 0 = 6n + 2 \Rightarrow$ Rest 2,
$\Rightarrow 2x + 1 = 6n + 3 \Rightarrow$ Rest 0.

$r = 2 \Rightarrow \exists n \in \mathbb{N}_0 : x = 3n + 2$
$\Rightarrow 2x + 0 = 6n + 4 \Rightarrow$ Rest 1,
$\Rightarrow 2x + 1 = 6n + 5 \Rightarrow$ Rest 2.

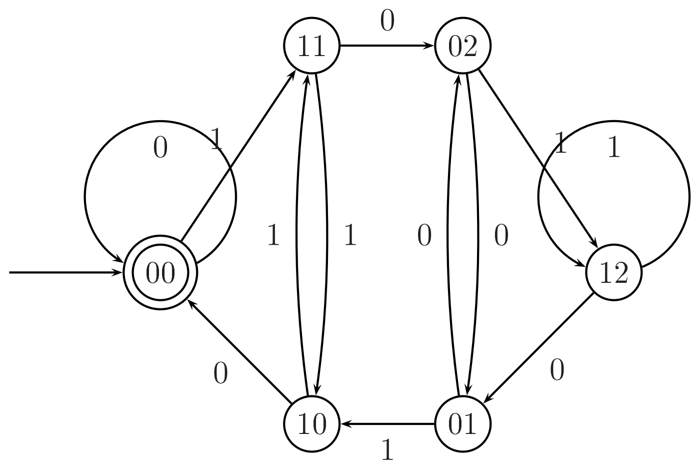
Der endliche Automat hat folgende Gestalt:



Ein regulärer Ausdruck, der T_3 beschreibt, sieht folgendermaßen aus: $(0^*(1(01^*0)^*1)^*)^*$

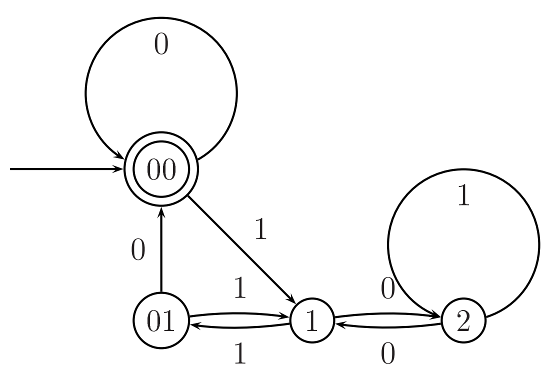
c) Eine Zahl ist genau dann durch 6 teilbar, wenn sie durch 3 teilbar ist und mit einer 0 endet.

Möglichkeit 1: Schneiden der Automaten in a) und b) wie in Aufgabe 2:



Möglichkeit 2:

"Verdoppeln" des Zustandes 0 beim Automaten in b): Zustand 01 wird angenommen, wenn das letzte Zeichen eine 1 war, Zustand 00 wird angenommen, wenn das letzte Zeichen eine 0 war:



d) Für m ungerade konstruieren wir einen Automaten mit m Zuständen, der T_m erkennt:

$A = (\{0,1\}, \{\mathbf{0}, \ldots, \mathbf{m-1}\}, \delta, \mathbf{0}, \mathbf{0})$, $\delta(x,s)$ sei immer der Rest von $2s+x$ bei Division durch m für alle $x \in \{0,1\}, s \in \{\mathbf{0}, \ldots, \mathbf{m-1}\}$.

Für jedes Wort $w \in \{0,1\}^*$ endet der Automat A im Zustand $\mathbf{r_{z(w)}}$, wobei $r_{z(w)}$ der Rest von $z(w)$ bei Division durch m ist.

Die Aussage stimmt für ϵ und damit für alle Wörter der Länge 0.

Annahme: Die Aussage gilt für alle Wörter der Länge n.

Sei wx ein Wort der Länge $n+1$ und $x \in \{0,1\}$. Dann gilt $z(wx) = 2z(w) + x$. Nach Annahme befindet sich A nach Einlesen von w im Zustand $\mathbf{r}$, der den Rest von $z(w)$ bei der Division durch m bezeichnet. $z(w)$ lässt sich also darstellen als $m \cdot k + r$ für ein $k \in \mathbb{N}$, und $z(wx) = m \cdot 2k + 2r + x$ folgt.

$2r + x$ sei $m \cdot k' + r'$ für ein $k' \in \mathbb{N}$ und $0 \leq r' \leq m-1$. Dann geht A beim Lesen von x, wenn A vorher im Zustand $\mathbf{r}$ war, nach Konstruktion in den Zustand $\mathbf{r'}$ über.

Es gilt $z(wx) = m \cdot 2k + 2r + x = m \cdot (2k + k') + r'$ und $0 \leq r' \leq m-1$; r' ist somit der Rest von $z(wx)$ bei Division durch m, wie behauptet.

$z(w)$ ist somit genau dann durch m teilbar, wenn A beim Lesen von w in dem einzigen akzeptierenden Zustand $\mathbf{0}$ endet, was genau dann der Fall ist, wenn A w akzeptiert. Damit gilt $L(A) = T_m$.

Angenommen, es gibt einen Automaten A mit weniger als m Zuständen, der T_m erkennt. Dann muss es zwei verschiedene Wörter w_1 und w_2 geben mit $0 \leq z(w_1) < z(w_2) \leq m-1$, für die A beim Einlesen den gleichen Zustand annimmt. (Die Binärdarstellungen der Zahlen von 0 bis $m-1$ sind m verschiedene Wörter!)

Idee: Wenn wir an w_1 und w_2 jeweils die gleichen Zeichenketten anhängen, müssen auch die so erhaltenen Wörter beim Einlesen von A zum gleichen Zustand führen. Wir hängen nun eine solche Zeichenkette v an, dass $z(w_1 v)$ durch m teilbar ist, was für $z(w_2 v)$ nicht möglich sein wird.

Sei k die Länge der Binärdarstellung von m und u die Binärdarstellung der kleinsten Zahl, die man zu $z(w_1 0^k)$ addieren muss, um eine durch m teilbare Zahl zu erhalten. Diese Zahl muss kleiner als m sein, was bedeutet, dass u höchstens k Zeichen lang ist.

Sei l die Länge von u.

Dann gilt $z(w_1 0^{k-l} u) = z(w_1 0^k) + z(u)$ ist nach Konstruktion von u durch m teilbar.

Da $w_1 0^{k-l} u$ von A erkannt wird, wird auch $w_2 0^{k-l} u$ von A erkannt, da in beiden Fällen A nach Einlesen von w_1 beziehungsweise w_2 im gleichen Zustand ist und die gleichen weiteren Zeichen einliest.

Somit muss $z(w_2 0^{k-l} u)$ durch m teilbar sein.

Es gilt also: $z(w_1 0^k) + z(u) = k_1 \cdot m$, $z(w_2 0^k) + z(u) = k_2 \cdot m$ für $k_1, k_2 \in \mathbb{N}$ und damit:

$z(w_2 0^k) - z(w_1 0^k) = (k_2 - k_1) \cdot m$
$\Rightarrow 2^k \cdot (z(w_2) - z(w_1)) = (k_2 - k_1) \cdot m$.

Da m ungerade ist, muss $k_2 - k_1$ durch 2^k teilbar sein, und es bleibt für $k' = (\frac{k_2-k_1}{2^k}$:

$z(w_2) - z(w_1) = k' \cdot m$.

Sowohl $z(w_1)$ als auch $z(w_2)$ sind jedoch kleiner als m, so dass die Gleichung nur erfüllt sein kann, wenn $z(w_1) = z(w_2)$ gilt. Dies war jedoch ausgeschlossen.

Somit kann es keinen Automaten mit weniger als m Zuständen geben, der T_m erkennt.

e) Eine Zahl ist genau dann durch 2^n teilbar, wenn die letzten n Ziffern ihrer Binärdarstellung 0 sind. Ein endlicher Automat mit $n+1$ Zuständen, der diese Binärdarstellungen erkennt, kann einfach speichern, wie viele Nullen zuletzt gelesen wurden (maximal n):

$Z = \{0, 1, \ldots, n\}, S = 0, F = \{n\}$
$\forall s \in \{0, \ldots, n\} : \delta(1, s) = 0, \delta(0, s) = \min\{n, s+1\}$.

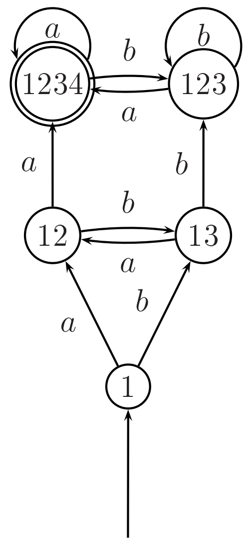
Aufgabe 4:

a) $(a \cup b)^*(a \cup bb)(ab)^*a$

b) Wir stellen wieder die Tabelle auf:

Teilmenge	a	b	Akzeptierender Zustand?
{1}	{1, 2}	{1, 3}	Nein
{1, 2}	{1, 2, 3, 4}	{1, 3}	Nein
{1, 3}	{1, 2}	{1, 2, 3}	Nein
{1, 2, 3}	{1, 2, 3, 4}	{1, 2, 3}	Nein
{1, 2, 3, 4}	{1, 2, 3, 4}	{1, 2, 3}	Ja

Der Automat hat folgende Gestalt:


a
b
b
1234
123
a
a
b
b
12
13
a
a
b
1