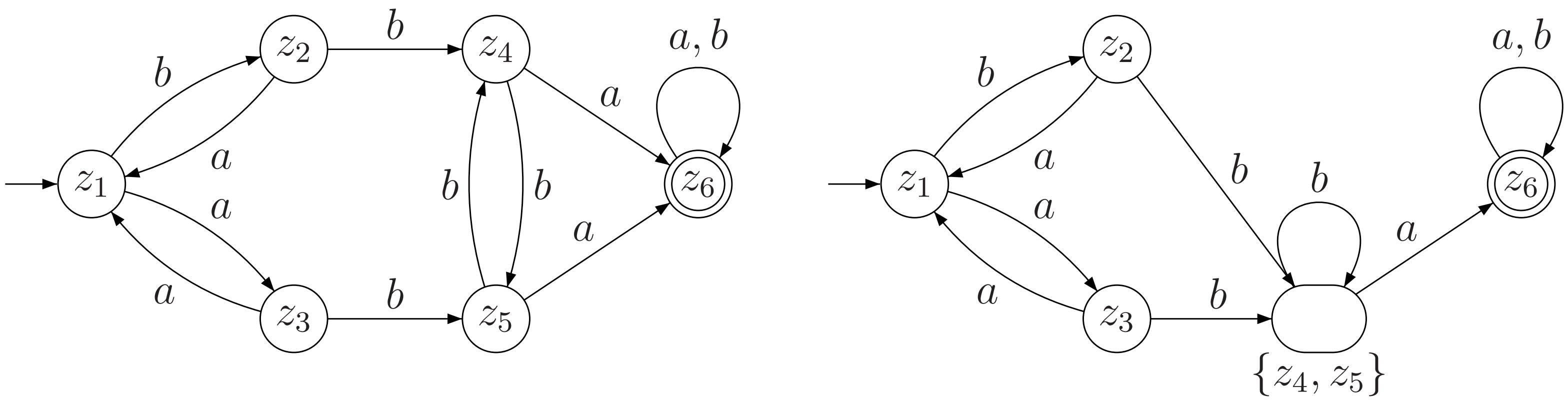
2.5 Minimale Automaten

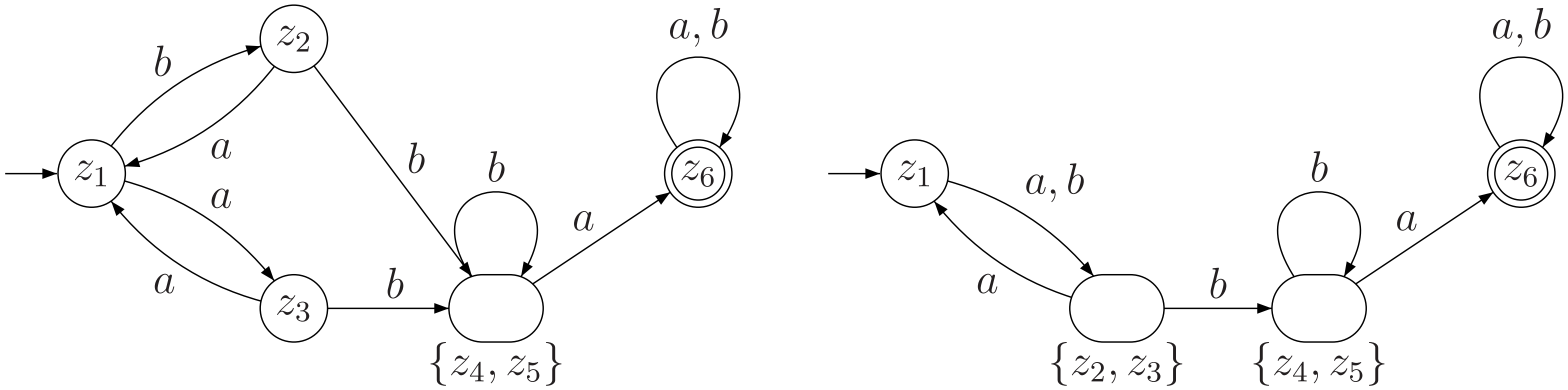
Beispiel 2.17.



Betrachte linken DFA M_1. Dann gilt für alle Wörter $w \in \{a, b\}^*$:

$$\delta(z_4, w) \in F \iff |w|_a > 0 \iff \delta(z_5, w) \in F$$

Es ist also egal, ob der Automat sich im Zustand z_4 oder im Zustand z_5 befindet. Daher akzeptiert M_1 dieselbe Sprache wie der rechte DFA M_2, in dem wir die „äquivalenten" Zustände z_4 und z_5 „verschmolzen" haben.



In diesem DFA gilt für alle $w \in \{a, b\}^* \setminus \{\varepsilon\}$: $\delta(z_2, w) = \delta(z_3, w)$ und daher

$$\delta(z_2, w) \in F \iff \delta(z_3, w) \in F\,.$$

Es ist also egal, ob der Automat sich im Zustand z_2 oder im Zustand z_3 befindet. Daher akzeptiert M_2 dieselbe Sprache wie der rechte DFA M_3, in dem wir die „äquivalenten" Zustände z_2 und z_3 „verschmolzen" haben. In diesem DFA sind keine 2 Zustände mehr „äquivalent", er kann also nicht weiter verkleinert werden.

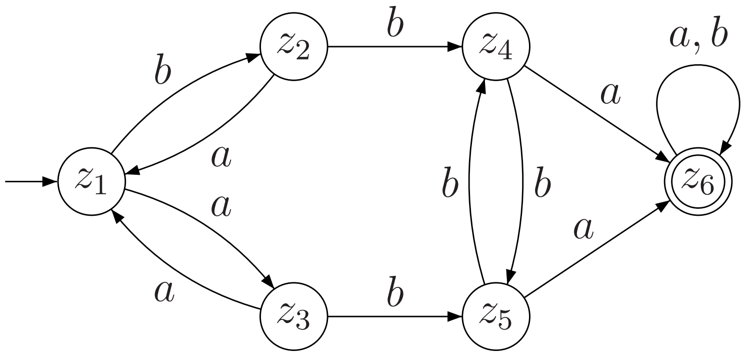
Systematisches Vorgehen Um festzustellen, daß 2 Zustände z und z' „äquivalent“ sind, muß man überprüfen, ob $\delta(z,w) \in F \iff \delta(z',w) \in F$ für alle Wörter w gilt. Da dies eine unendliche Bedingung ist, ist nicht klar, wie sie algorithmisch gelöst werden kann.
Wir gehen hierzu wie folgt vor:
Eingabe: DFA $M = (Z, \Sigma, \iota, \delta, F)$, in dem alle Zustände von ι aus erreichbar sind
Ausgabe: Menge der Paare (z, z') von Zuständen aus Z mit

$$\delta(z,w) \in F \iff \delta(z',w) \in F \text{ für alle } w \in \Sigma^*$$

1. Stelle eine Tabelle aller ungeordneten Zustandspaare $\{z, z'\}$ mit $z \neq z'$ auf.
2. Markiere alle Paare $\{z, z'\}$ mit $z \in F$ und $z' \notin F$.
3. Markiere ein unmarkiertes Paar $\{z, z'\}$, für das es ein $a \in \Sigma$ gibt, so daß das Paar $\{\delta(z,a), \delta(z',a)\}$ bereits markiert ist.
4. Wiederhole den vorherigen Schritt, bis keine neuen Markierungen entstehen.
5. Gib alle Paare $\{z, z'\}$ aus, die nicht markiert sind.

Der DFA M' wird jetzt berechnet, indem die ausgegebenen Zustandspaare „verschmolzen“ werden.

Beispiel 2.17 (Fortsetzung)



z_2	9				
z_3	8				
z_4	7	2	5		
z_5	6	4	3		
z_6	1	1	1	1	1
	z_1	z_2	z_3	z_4	z_5

Ausgegeben werden also die Paare $\{z_2, z_3\}$ und $\{z_4, z_5\}$. Das „Verschmelzen“ dieser Zustände im DFA M_1 liefert den DFA M_3 aus Beispiel 2.17.

Satz 2.18. *Sei M ein DFA und M' der nach obigem Algorithmus berechnete DFA.*

1. $L(M) = L(M')$
2. *Es gibt keinen DFA M'' mit weniger Zuständen als M' und $L(M) = L(M'')$.*
3. *Jeder DFA M'' mit $L(M) = L(M'')$ und genauso vielen Zuständen wie M' ist isomorph zu M', d.h. „gleich bis auf Umbenennung der Zustände“.*

Daher heißt der berechnete DFA M' auch *Minimalautomat*.

Um festzustellen, ob zwei DFAs M_1 und M_2 dieselbe Sprache erkennen, können wir also wie folgt vorgehen:

1. berechne die Minimalautomaton M_1' und M_2'
2. teste, ob diese isomorph sind, d.h. „gleich bis auf Umbenennung der Zustände“

2.6 Analyse regulärer Sprachen

Wir diskutieren Verfahren, die die folgenden Fragestellungen bzw. Probleme für reguläre Sprachen entscheiden. Dabei nehmen wir an, daß reguläre Sprachen als DFAs, NFAs oder reguläre Ausdrücke gegeben sind.

- *Wortproblem:* Gilt $w \in L$ für eine gegebene reguläre Sprache L und $w \in \Sigma^*$?
- *Leerheitsproblem:* Gilt $L = \emptyset$ für eine gegebene reguläre Sprache L?
- *Endlichkeitsproblem:* Ist eine gegebene reguläre Sprache L endlich?
- *Disjunktheitsproblem:* Gilt $L_1 \cap L_2 = \emptyset$ für gegebene reguläre L_1, L_2?
- *Inklusionsproblem:* Gilt $L_1 \subseteq L_2$ für gegebene reguläre L_1, L_2?
- *Äquivalenzproblem:* Gilt $L_1 = L_2$ für gegebene reguläre L_1, L_2?

<u>Wortproblem (DFA)</u>

Eingabe: DFA $M = (Z, \Sigma, \iota, \delta, F)$ und $w \in \Sigma^*$
Frage: $w \in L(M)$?

Verfahren:
Sei $w = a_1 a_2 \cdots a_n$ mit $a_i \in \Sigma$.
Verfolge die Zustandsübergänge von M, die durch die Symbole $a_1, \ldots, a_n$ vorgegeben sind:

$z := \iota$
for $i := 1$ **to** n **do**
    $z := \delta(z, a_i)$
endfor
if $z \in F$ **then return**(JA) **else return**(NEIN)

<u>Zeitbedarf:</u> polynomiell in M und w

Wortproblem (NFA)

Eingabe: NFA $M = (Z, \Sigma, I, \delta, F)$ und $w \in \Sigma^*$
Frage: $w \in L(M)$?

Verfahren:
man könnte aus dem NFA M einen äquivalenten DFA M' berechnen und dann obiges Verfahren anwenden. Der 1. Schritt verlange Zeit exponentiell in M.

alternatives Verfahren:
Sei $w = a_1 a_2 \cdots a_n$ mit $a_i \in \Sigma$.
Berechne induktiv die Menge S von Zuständen, in denen sich der NFA nach Lesen von $a_1 a_2 \cdots a_i$ befinden kann:

$S := I$
for $i := 1$ **to** n **do**
  $S := \{z' \in Z \mid \exists z \in S\colon z' \in \delta(z, a_i)\}$
endfor
if $S \cap F \neq \emptyset$ **then** **return**(JA) **else** **return**(NEIN)

Zeitbedarf: polynomiell in M und w

Leerheitsproblem

Eingabe: NFA $M = (Z, \Sigma, I, \delta, F)$.
Frage: $L(M) = \emptyset$?

Verfahren:
Sei $G = (Z, \to)$ der gerichtete Graph mit

$$z \to z' \iff \exists a \in \Sigma : z' \in \delta(z, a)\,.$$

Dann gilt: $L(M) \neq \emptyset$ genau dann, wenn es in dem Graphen G einen (evtl. leeren) Pfad von einem Knoten aus I zu einem Knoten aus F gibt.
Dies kann z.B. mit dem Algorithmus' von Dijkstra entschieden werden.

Zeitbedarf: polynomiell in M

Endlichkeitsproblem

Eingabe: NFA $M = (Z, \Sigma, I, \delta, F)$.
Frage: Ist $L(M)$ endlich?

Verfahren:
Sei $G = (Z, \rightarrow)$ wieder der gerichtete Graph mit

$$z \rightarrow z' \iff \exists a \in \Sigma : z' \in \delta(z, a)\,.$$

Dann gilt: $L(M)$ ist genau dann unendlich, wenn es $z \in Z$, $z_0 \in I$ und $z_1 \in F$ gibt mit $z_0 \rightarrow^* z \rightarrow^+ z \rightarrow^* z_1$, d.h. z liegt auf einem Zyklus, ist von einem Startzustand aus erreichbar und von z kann ein Endzustand erreicht werden.
Dies kann wieder mit dem Algorithmus von Dijkstra entschieden werden.

Zeitbedarf: polynomiell in M

Disjunktheitsproblem

Eingabe: NFAs M_1 und M_2
Frage: Gilt $L(M_1) \cap L(M_2) = \emptyset$?

Verfahren:
Konstruiere aus M_1 und M_2 einen NFA M mit $L(M) = L(M_1) \cap L(M_2)$ (mgl. nach Folgerung 2.6 in polynomieller Zeit mittels Kreuzproduktkonstruktion).
Teste, ob $L(M) = \emptyset$ gilt.

Zeitbedarf: polynomiell in M_1 und M_2, da M nur polynomiell groß sein kann

Inklusionsproblem (NFAs)

Korrekturen vom 18.5.2020

Eingabe: NFAs M_1 und M_2.
Frage: Gilt $L(M_1) \subseteq L(M_2)$.

Verfahren:

	Zeitbedarf
1. berechne DFA M_2' mit $L(M_2') = L(M_2)$	exponentiell in M_2
2. berechne DFA M_2'' mit $L(M_2'') = \Sigma^* \setminus L(M_2')$	polynomiell in M_2'
3. berechne NFA M mit $L(M) = L(M_2'') \cap L(M_1)$	polynomiell in M_2'' und M_1
4. entscheide, ob $L(M) = \emptyset$	polynomiell in M

Dieses Verfahren ist korrekt, denn es gelten

$$L(M_1) \subseteq L(M_2) \iff L(M_1) \setminus L(M_2) = \emptyset$$

und

$$L(M_1) \setminus L(M_2) = \cancel{\Sigma^* \setminus} L(M_1) \cap \Sigma^* \setminus L(M_2) = L(M_2'') \cap L(M_1) = L(M)$$

Zeitbedarf: exponentiell in M_2 und polynomiell in M_1

Inklusionsproblem (NFA, DFA) Korrekturen vom 18.5.2020

Eingabe: NFA M_1 und DFA M_2.
Frage: Gilt $L(M_1) \subseteq L(M_2)$.

Verfahren:
verwende obiges Verfahren, wobei wir $M_2{}' = M_2$ setzen können

Zeitbedarf: polynomiell in M_1 und M_2

Äquivalenzproblem (NFAs)

Eingabe: NFAs M_1 und M_2.
Frage: Gilt $L(M_1) = L(M_2)$?

Verfahren 1:
Es gilt: $L(M_1) = L(M_2)$ genau dann, wenn $L(M_1) \subseteq L(M_2)$ und $L(M_2) \subseteq L(M_1)$.

Verfahren 2:

1. berechne DFAs mit $L(M_1') = L(M_1)$ und $L(M_2') = L(M_2)$
2. berechne minimale DFAs mit $L(M_1'') = L(M_1')$ und $L(M_2'') = L(M_2')$
3. teste, ob M_1'' und M_2'' isomorph sind, d.h. durch Umbenennung der Zustände ineinander überführt werden können

Dieses Verfahren ist korrekt nach Satz 2.18.

Zeitbedarf für beide Verfahren: exponentiell in M_1 und in M_2
polynomiell, wenn M_1 und M_2 DFAs sind