Grundlagen der theoretischen Informatik

Kurt Sieber

Fakultät IV, Department ETI
Universität Siegen

SS 2013

Vorlesung vom 02.05.2013

Verallgemeinerung des Beipiels:

Sei Σ ein Alphabet und sei $u = a_1 \dots a_n \in \Sigma^*$.
Wie sieht ein (minimaler) DEA A_u aus, der die Sprache

$$L_u = \{w \in \Sigma^* \mid u \text{ ist Teilwort von } w\}$$

erkennt? Offensichtlich genügt es (wie im Beispiel $u = 0100$), dass sich A_u das "bisher erkannte Präfix" des Suchwortes u merkt.

Genauer:

Sei w das bisher gelesene Wort.

Wenn w bereits in L_u liegt, d.h. wenn w schon das ganze Suchwort u enthält, dann sollte sich A_u in einem Endzustand befinden und diesen nicht mehr verlassen.

Wenn w noch nicht in L_u liegt, dann sollte A_u das längste Präfix von u kennen, das zugleich Suffix von w ist.

Reguläre Sprachen

Damit bieten sich die Präfixe des Suchwortes u als Zustände des DEA A_u an: Sei $u_i = a_1 \dots a_i$ $(i = 0, \dots, n)$ das Präfix der Länge i von u. Dann definiert man einen DEA $A_u = (\Sigma, Q, s, F, \delta)$ mit

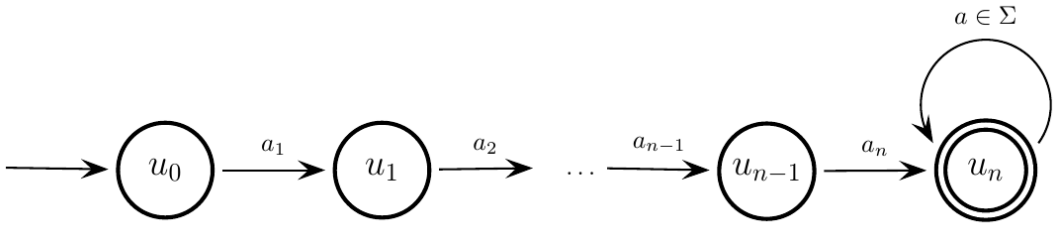
$$Q = \mathit{Pref}(u) = \{u_0, \dots, u_n\}$$

$$s = u_0 = \varepsilon$$

$$F = \{u_n\} = \{u\}$$

$$\delta(u_{i-1}, a_i) = u_i \text{ für } i = 0, \dots, n$$

Damit ist das "Rückgrat" (engl. *spine*) des Automaten A_u festgelegt



und es fehlen nur noch die Übergänge $\delta(u_{i-1}, a)$ mit $a \neq a_i$.

Man erhält sie durch folgende Überlegung: A_u soll sich das längste Präfix von u merken, das Suffix des bisher gelesenen Wortes w ist. Deshalb muss für alle $w \in \Sigma^*$ gelten: Wenn u_{i-1} das längste Wort in $\mathit{Pref}(u) \cap \mathit{Suff}(w)$ ist, dann ist $\delta(u_{i-1}, a) = u_j$ das längste Wort in $\mathit{Pref}(u) \cap \mathit{Suff}(wa)$.

Wählt man insbesondere $w = u_{i-1}$, so ergibt sich $\delta(u_{i-1}, a) = u_j$, wobei u_j das *längste Wort in* $\mathit{Pref}(u) \cap \mathit{Suff}(u_{i-1}a)$ ist. Damit ist die Übergangsfunktion δ des Automaten A_u vollständig festgelegt.

Durch Induktion über die Länge von w lässt sich zeigen, dass jedes Wort $w \in \Sigma^*$ tatsächlich zum gewünschten Zustand führt, d.h.

$$\delta^*(s, w) = \begin{cases} u_n & \text{falls } w \in L_u \\ \text{das längste } u_i \in \mathit{Pref}(u) \cap \mathit{Suff}(w) & \text{sonst} \end{cases}$$

Die Minimalität von A_u ergibt sich daraus, dass $u_i, u_j \in \mathit{Pref}(u)$ mit $i \neq j$ nicht zum gleichen Zustand führen dürfen.

Reguläre Sprachen

Wir haben also einen *Algorithmus*, der der zu jedem Wort $u \in \Sigma^*$ einen minimalen 'Suchautomaten' für u liefert, d.h. einen minimalen DEA A_u, der genau die Texte akzeptiert, die das Wort u enthalten.

Auf diesem Algorithmus basieren *Suchprogramme*:

Für jedes Suchwort u wird ein DEA konstruiert, der L_u erkennt.
Dieser DEA wird auf den zu durchsuchenden Text angesetzt.

Lohnt sich der Aufwand?

Ja, weil die Länge m des zu durchsuchenden Textes meist viel größer ist als die Länge n des Suchwortes.
Der Zeitaufwand für die Konstruktion des DEA ist linear in n.
Die Laufzeit des DEA beträgt m.
Also ist die Gesamtlaufzeit nur wenig größer als m.

Zum Vergleich:
Beim 'naiven' Suchalgorithmus testet man für $i = 1, \ldots, m-n+1$, ob das Suchwort u an der Stelle i des Textes w beginnt.

Das erfordert im schlimmsten Fall Laufzeit $n\,(m-n+1)$,
z.B. wenn $u = 0^{n-1}1$ und $w = 0^m$:
An jeder Stelle i von w muss man bis zum letzten Zeichen von u laufen, um zu sehen, dass u nicht passt.

In der Praxis ist man natürlich nicht nur daran interessiert, *ob* das Suchwort im Text vorkommt, sondern auch *wo* es vorkommt. Dazu konstruiert man sich einen DEA A'_u, der die Sprache $L'_u = \{w \in \Sigma^* \mid u$ ist *Suffix* von $w\}$ erkennt, und der immer dann eine Markierung im Suchtext setzt, wenn er gerade im Endzustand ist, d.h. wenn er das Suchwort gerade gefunden hat.
A'_u unterscheidet sich von A_u nur darin, dass er nicht unbedingt im Endzustand bleibt, wenn er das Suchwort u schon gefunden hat, sondern von dort mit jedem Zeichen a in den Zustand $[u_j]$ mit $j = \max\{u_i \in \{u_0, \ldots, u_n\} \mid u_i$ ist Suffix von $ua\}$ übergeht.

Minimierung

Bisher:
Konstruktion eines minimalen DEA für L aus den L-Äquivalenzklassen.

Nachteil:
Zur Bestimmung der L-Äquivalenzklassen gibt es keinen Algorithmus, sondern es ist im allgemeinen 'mathematische Argumentation' nötig (die in Spezialfällen wie bei der Sprache L_u einen Algorithmus zur Konstruktion des DEA liefern kann).

Deshalb stellt sich die Frage:
Kann man aus einem beliebigen DEA einen äquivalenten minimalen DEA erhalten?.

Lösungsansatz:
Wenn ein DEA unnötige Unterscheidungen trifft, d.h. wenn zwei L-äquivalente Wörter in unterschiedlichen Zuständen landen, dann kann man diese Zustände miteinander 'verschmelzen'.

'Verschmelzen' bedeutet aus mathematischer Sicht:
Man fasst die Zustände zu *Äquivalenzklassen* zusammen.

Definition 2.32 *Sei $A = (\Sigma, Q, s, F, \delta)$ ein DEA. Zwei Zustände $p, q \in Q$ heißen A-äquivalent (Schreibweise: $p \sim_A q$), wenn für alle $w \in \Sigma^*$ gilt:*

$$\delta^*(p, w) \in F \Leftrightarrow \delta^*(q, w) \in F$$

Mit anderen Worten:

$$\begin{aligned} p \sim_A q \quad \Leftrightarrow \quad & \text{für alle } w \in \Sigma^* \text{ gilt:} \\ & \text{entweder } \delta^*(p, w), \delta^*(q, w) \in F \\ & \text{oder } \delta^*(p, w), \delta^*(q, w) \notin F \end{aligned}$$

A-Äquivalenz ist eine Äquivalenzrelation auf der Menge Q. Die A-Äquivalenzklasse eines Zustands q bezeichnen wir mit $[q]_A$, also

$$[q]_A = \{p \in Q \mid p \sim_A q\}$$

Lemma 2.33

1. *Wenn* $p \sim_A q$*, dann gilt auch* $\delta^*(p, v) \sim_A \delta^*(q, v)$ *für alle* $v \in \Sigma^*$.
2. *Für jeden Zustand* $q \in Q$ *gilt:* $q \in F \Leftrightarrow [q]_A \subseteq F$
   (Also $F = \bigcup_{q \in F} [q]_A$*.)*

Beweis:

1. Sei $p \sim_A q$ und $v \in \Sigma^*$.

   Dann gilt für alle $v' \in \Sigma^*$:

   $\delta^*(p, vv') \in F \Leftrightarrow \delta^*(q, vv') \in F$,

   also $\delta^*(\delta^*(p, v), v') \in F \Leftrightarrow \delta^*(\delta^*(q, v), v') \in F$,

   und das bedeutet $\delta^*(p, v) \sim_A \delta^*(q, v)$.

2. '$\Leftarrow$' ist klar.

   '$\Rightarrow$': Sei $q \in F$ und $p \sim_A q$.

   Dann ist wegen $\delta(q, \varepsilon) \in F$ auch $p = \delta(p, \varepsilon) \in F$. □

Satz 2.34 *Zu jedem DEA A lässt sich ein äquivalenter minimaler DEA $\bar{A}$ konstruieren. $\bar{A}$ ist isomorph zum Myhill-Nerode-Automaten für die Sprache $L(A)$, d.h. er unterscheidet sich von ihm nur durch eine Umbenennung der Zustände.*

Beweis:

Sei $A = (\Sigma, Q, s, F, \delta)$.

Wir dürfen annehmen, dass A nur erreichbare Zustände besitzt.

Wir definieren $\bar{A} = (\Sigma, \bar{Q}, \bar{s}, \bar{F}, \bar{\delta})$ durch

$\bar{Q} = \{[q]_A \mid q \in Q\}$

$\bar{s} = [s]_A$

$\bar{F} = \{[q]_A \mid q \in F\} \quad (= \{[q]_A \mid [q]_A \subseteq F\})$

$\bar{\delta} : \bar{Q} \times \Sigma \to \bar{Q}$

$\bar{\delta}([q]_A, a) = [\delta(q, a)]_A$

$\bar{\delta}$ ist wohldefiniert, denn aus $p \sim_A q$ folgt nach Lemma 2.33 stets $\delta(p,a) \sim_A \delta(q,a)$, und durch Induktion über die Länge von w zeigt man leicht, dass für alle $q \in Q$ und alle $w \in \Sigma^*$ gilt:

$$\bar{\delta}^*([q]_A, w) = [\delta^*(q,w)]_A \qquad (*)$$

Damit folgt:

$$\begin{array}{lcll} w \in L(\bar{A}) & \Leftrightarrow & \bar{\delta}^*([s]_A, w) \in \bar{F} & \text{per Definition von } L(\bar{A}) \\ & \Leftrightarrow & [\delta^*(s,w)]_A \in \bar{F} & \text{wegen } (*) \\ & \Leftrightarrow & [\delta^*(s,w)]_A \subseteq F & \text{per Definition von } \bar{F} \\ & \Leftrightarrow & \delta^*(s,w) \in F & \text{nach Lemma 2.33} \\ & \Leftrightarrow & w \in L(A) & \text{per Definition von } L(A) \end{array}$$

Also ist $L(\bar{A}) = L(A)$, d.h. $\bar{A}$ ist äquivalent zu A.

Außerdem gilt für alle $u_1, u_2 \in \Sigma^*$:

$$
\begin{aligned}
u_1 \sim_{L(A)} u_2 \quad &\Leftrightarrow \quad \text{für alle } w \in \Sigma^* \text{ gilt} \\
&\qquad u_1 w \in L(A) \Leftrightarrow u_2 w \in L(A) \\
&\Leftrightarrow \quad \text{für alle } w \in \Sigma^* \text{ gilt} \\
&\qquad \delta^*(s, u_1 w) \in F \Leftrightarrow \delta^*(s, u_2 w) \in F \\
&\Leftrightarrow \quad \text{für alle } w \in \Sigma^* \text{ gilt} \\
&\qquad \delta^*(\delta^*(s, u_1), w) \in F \Leftrightarrow \delta^*(\delta^*(s, u_2), w) \in F \\
&\Leftrightarrow \quad \delta^*(s, u_1) \sim_A \delta^*(s, u_2) \\
&\Leftrightarrow \quad [\delta^*(s, u_1)]_A = \delta^*(s, u_2)]_A \\
&\Leftrightarrow \quad \bar{\delta}^*([s]_A, u_1) = \bar{\delta}^*([s]_A, u_2)
\end{aligned}
$$

wobei die letzte Umformung wieder wegen (*) gilt.

Zwei Wörter u_1, u_2 sind also genau dann $L(A)$-äquivalent, wenn sie in $\bar{A}$ zum gleichen Zustand führen.

Das bedeutet, dass die Anzahl der erreichbaren Zustände von $\bar{A}$ nicht größer ist als die Anzahl der $L(A)$-Äquivalenzklassen, d.h. die Anzahl der Zustände des Myhill-Nerode-Automaten.

Und weil $\bar{A}$—wie der ursprüngliche Automat A—nur erreichbare Zustände hat, folgt daraus, dass er die kleinstmögliche Anzahl von Zuständen hat, d.h. $\bar{A}$ ist minimal.

Die folgende genauere Argumentation zeigt, dass er sogar zum Myhill-Nerode-Automaten isomorph ist.

Sei $\tilde{A} = (\Sigma, \tilde{Q}, \tilde{s}, \tilde{F}, \tilde{\delta})$ der Myhill-Nerode-Automat für $L(A)$.

Wir definieren eine Abbildung

$$\Phi : \tilde{Q} \rightarrow \bar{Q}$$
$$\Phi([w]_{L(A)}) = \bar{\delta}^*([s]_A, w)$$

Nach den obigen Betrachtungen über die $L(A)$-Äquivalenzklassen ist Φ wohldefiniert und injektiv, und weil $\bar{A}$ nur erreichbare Zustände hat, ist Φ auch surjektiv.

Darüber hinaus gilt:

- $\Phi(\tilde{s}) = \Phi([\varepsilon]_{L(A)}) = \bar{\delta}^*([s]_A, \varepsilon) = [s]_A$
- $\Phi([w]_{L(A)}) \in \bar{F} \Leftrightarrow \bar{\delta}^*([s]_A, w) \in \bar{F} \Leftrightarrow w \in L(\bar{A})$
  $\Leftrightarrow w \in L(A) \Leftrightarrow [w]_{L(A)} \in \tilde{F}$
- $\Phi(\tilde{\delta}([w]_{L(A)}, a)) = \Phi([wa]_{L(A)}) = \bar{\delta}^*([s]_A, wa)$
  $= \bar{\delta}(\bar{\delta}^*([s]_A, w), a) = \bar{\delta}(\Phi([w]_{L(A)}), a)$

Also ist Φ eine bijektive Abbildung, unter der sich die Start- und Endzustände und die Zustandsübergänge in den beiden Automaten entsprechen.

Das bedeutet, dass Φ ein Isomorphismus ist, d.h. eine reine Umbenennung der Zustände.

Reguläre Sprachen

Es bleibt zu zeigen, dass sich der DEA $\bar{A}$ aus dem DEA A *konstruieren* lässt, d.h. dass sich die A-Äquivalenzklassen bestimmen lassen (alles andere ist klar).

Dazu definieren wir Relationen $\sim_n$ auf Q durch Induktion über n:

$$\begin{aligned} p \sim_0 q &\Leftrightarrow (p \in F \Leftrightarrow q \in F) \\ &\Leftrightarrow \text{entweder } p, q \in F \text{ oder } p, q \notin F \\ p \sim_{n+1} q &\Leftrightarrow p \sim_n q \text{ und für alle } a \in \Sigma \text{ gilt } \delta(p, a) \sim_n \delta(q, a) \end{aligned}$$

Durch Induktion über n folgt leicht, dass für jedes n gilt:

$$\begin{aligned} p \sim_n q \Leftrightarrow &\text{ für } \textit{alle } w \in \Sigma^* \text{ mit } |w| \leq n \text{ gilt} \\ &\delta^*(p, w) \in F \Leftrightarrow \delta^*(q, w) \in F \end{aligned}$$

Damit ist klar, dass jedes $\sim_n$ eine Äquivalenzrelation ist, und dass $\sim_A$ der Durchschnitt aller $\sim_n$ ist.

Also bleibt nur noch zu zeigen, dass sich $\bigcap_{n\geq 0} \sim_n$ berechnen lässt.

Zunächst beachte man, dass (per Definition) $\sim_{n+1} \subseteq \sim_n$ für alle $n \geq 0$ gilt, d.h. die $\sim_n$ bilden eine absteigende Folge $\sim_0 \supseteq \sim_1 \supseteq \ldots$ von Relationen auf Q, also eine absteigende Folge von Teilmengen von $Q \times Q$.

Da $Q \times Q$ endlich ist, kann eine solche Folge *nicht echt absteigend* sein, also gibt es eine Zahl m mit $\sim_m = \sim_{m+1}$.

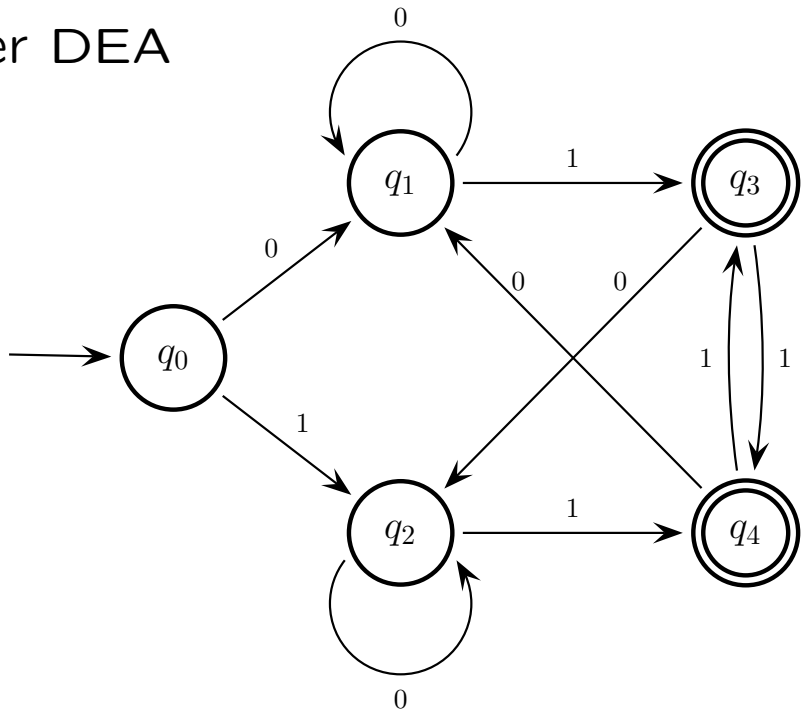
Weil sich $\sim_{m+2}$ wieder auf die gleiche Art aus $\sim_{m+1}$ ergibt wie $\sim_{m+1}$ aus $\sim_m$, folgt dann $\sim_m = \sim_n$ für alle $n \geq m$,

und das bedeutet, dass $\sim_m = \bigcap_{n\geq 0} \sim_n = \sim_A$.

Damit haben wir einen Algorithmus zur Berechnung von $\sim_A$ gefunden: Wir berechnen die Relationen $\sim_0, \sim_1, \ldots$ bis wir das erste m finden mit $\sim_m = \sim_{m+1}$.

Für dieses m gilt dann $\sim_m = \sim_A$. □

Beispiel: Sei A der DEA



$\sim_0$ hat die Äquivalenzklassen $Q \setminus F = \{q_0, q_1, q_2\}$ und $F = \{q_3, q_4\}$.

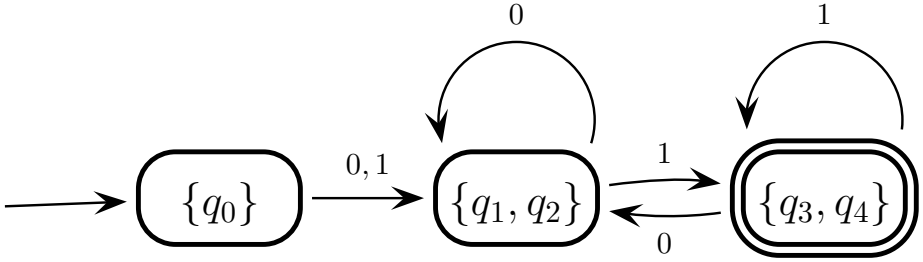
1. Verfeinerungsschritt:

$\delta(q_0, 0) = \delta(q_1, 0) = q_1$ und $\delta(q_2, 0) = q_2$ liegen in $Q \setminus F$,
aber $\delta(q_0, 1) = q_2 \in Q \setminus F$ und $\delta(q_1, 1) = q_3, \delta(q_2, 1) = q_4$ liegen in F.
Also zerfällt $\{q_0, q_1, q_2\}$ in zwei $\sim_1$-Klassen $\{q_0\}$ und $\{q_1, q_2\}$.

$\delta(q_3, 0) = q_2$ und $\delta(q_4, 0) = q_1$ liegen in $Q \setminus F$,
$\delta(q_3, 1) = q_4$ und $\delta(q_4, 1) = q_3$ liegen in F.
Also bleibt die $\sim_0$-Klasse $\{q_3, q_4\}$ als $\sim_1$-Klasse erhalten.
Damit besitzt $\sim_1$ die drei Äquivalenzklassen $\{q_0\}, \{q_1, q_2\}$ und $\{q_3, q_4\}$.

2. Verfeinerungsschritt:

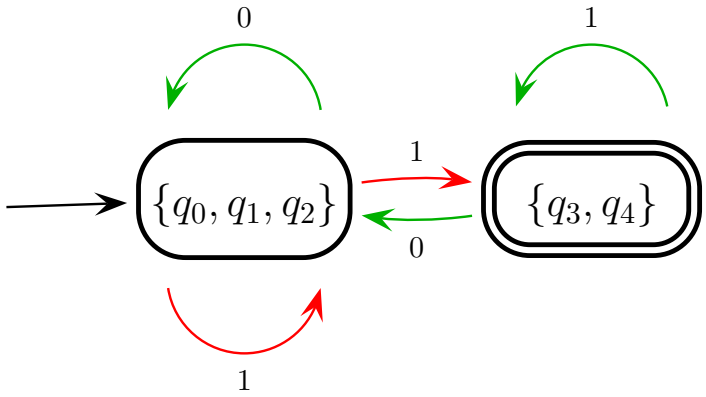
Die $\sim_1$-Klasse $\{q_0\}$ ist nicht mehr weiter aufteilbar.
$\delta(q_1, 0), \delta(q_2, 0) \in \{q_1, q_2\}$ und $\delta(q_1, 1), \delta(q_2, 1) \in \{q_3, q_4\}$.
Also bleibt die $\sim_1$-Klasse $\{q_1, q_2\}$ als $\sim_2$-Klasse erhalten.
$\delta(q_3, 0), \delta(q_4, 0) \in \{q_1, q_2\}$ und $\delta(q_3, 1), \delta(q_4, 1) \in \{q_3, q_4\}$.
Also bleibt auch die $\sim_1$-Klasse $\{q_3, q_4\}$ als $\sim_2$-Klasse erhalten.
Damit ist $\sim_1 = \sim_2$ gezeigt, also ist $\sim_2 = \sim_A$,
und man erhält den folgenden zu A äquivalenten minimalen DEA:



Eine anschauliche Erklärung für die Relationen $\sim_n$:

Man kann jede dieser Äquivalenzrelationen als 'Versuch' auffassen, den minimalen DEA zu konstruieren:
Man nimmt die Äquivalenzklassen von $\sim_n$ als Zustände, und versucht die Zustandsübergänge zwischen ihnen richtig festzulegen.
Scheitert der Versuch, so sieht man daran, wie die Äquivalenzklassen weiter aufgeteilt werden müssen. Gelingt der Versuch, so hat man den minimalen DEA gefunden (weil man mit $\sim_0$, d.h. mit der kleinstmöglichen Anzahl von Zuständen begonnen und immer nur die notwendigen Verfeinerungen vorgenommen hat).

An unserem Beispiel sieht das so aus:
Man versucht zunächst, einen DEA zu konstruieren, der nur die beiden Äquivalenzklassen $\{q_0, q_1, q_2\}$ und $\{q_3, q_4\}$ von $\sim_0$ als Zustände hat:



Das geht gut für die Übergänge

$\delta(q_0, 0), \delta(q_1, 0), \delta(q_2, 0) \in \{q_0, q_1, q_2\}$,

$\delta(q_3, 0), \delta(q_4, 0) \in \{q_0, q_1, q_2\}$,

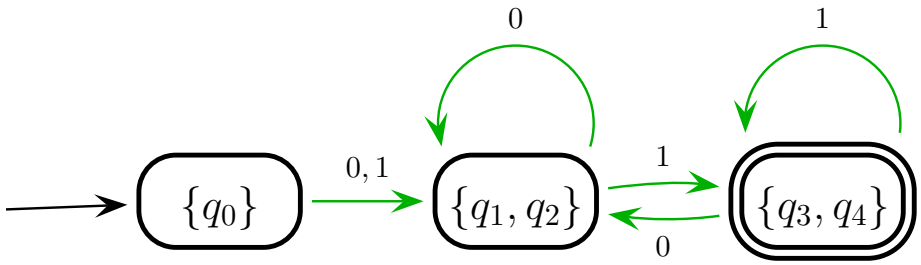
und $\delta(q_3, 1), \delta(q_4, 1) \in \{q_3, q_4\}$,

scheitert aber an den sich widersprechenden Übergängen

$\delta(q_0, 1) \in \{q_0, q_1, q_2\}$ und $\delta(q_1, 1), \delta(q_2, 1) \in \{q_3, q_4\}$.

An den gescheiterten Übergängen sieht man, dass $\{q_0, q_1, q_2\}$ in $\{q_0\}$ und $\{q_1, q_2\}$ aufgeteilt werden muss.

Also versucht man jetzt, einen DEA zu konstruieren, der die drei Äquivalenzklassen $\{q_0\}$, $\{q_1, q_2\}$ und $\{q_3, q_4\}$ von $\sim_1$ als Zustände hat:



Jetzt geht mit den Übergängen alles gut:

$\delta(q_0, 0) \in \{q_1, q_2\}$ $\quad$ $\delta(q_0, 1) \in \{q_1, q_2\}$

$\delta(q_1, 0), \delta(q_2, 0) \in \{q_1, q_2\}$ $\quad$ $\delta(q_1, 1), \delta(q_2, 1) \in \{q_3, q_4\}$

$\delta(q_3, 0), \delta(q_4, 0) \in \{q_1, q_2\}$ $\quad$ $\delta(q_3, 1), \delta(q_4, 1) \in \{q_3, q_4\}$

Also ist der minimale DEA gefunden.