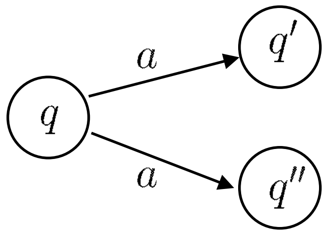
Automaten und Formale Sprachen

Teil I: Endliche Automaten

Wieso sind NEAs nichtdeterministisch?

Zu einem Paar $(q, a) \in Q \times \Sigma$ kann es mehr als einen Übergang geben!



Oder auch gar keinen!

Zu einem gegebenen Wort w kann es in einem NEA keinen, einen, oder mehrere Pfade vom Anfangszustand aus geben.

Das Wort wird akzeptiert wenn mindestens einer dieser Pfade zu einem Endzustand führt.

Bei deterministischen endlichen Automaten gibt es zu jedem Wort genau einen Pfad, der mit dem Anfangszustand beginnt.

Definition 2.1 (Deterministischer endlicher Automat)

Ein NEA $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ heißt deterministisch (DEA), falls es für alle $q \in Q, a \in \Sigma$ genau ein $q' \in Q$ gibt mit $(q, a, q') \in \Delta$.

Anstelle der Übergangsrelation Δ verwenden wir bei DEAs die Übergangsfunktion $\delta : Q \times \Sigma \to Q$ mit

$$\delta(q, a) = q' \text{ gdw } (q, a, q') \in \Delta.$$

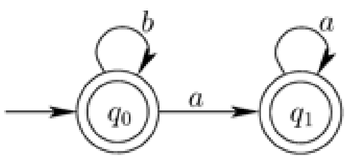
DEAs werden daher in der Form $(Q, \Sigma, q_0, \delta, F)$ geschrieben.

Beachte: neben der Eindeutigkeit des Übergangs wird hier auch die Existenz des Übergangs gefordert.

In der Literatur wird für DEAs manchmal nur Eindeutigkeit gefordert; dies ist aber für die akzeptierte Sprachklasse irrelevant.

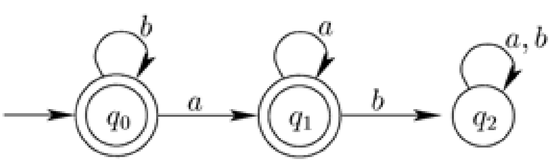
Beispiel 2.2

Der NEA



aus Beispiel 1.7 erfüllt zwar die Eindeutigkeitsforderung, er ist aber kein DEA in unserem Sinne, da für (q_1, b) kein Übergang existiert.

Einen äquivalenten DEA erhält man durch Hinzunahme eines Papierkorbzustands:



Definition 2.3 (kanonische Fortsetzung der Übergangsfunktion)

Die kanonische Fortsetzung von $\delta : Q \times \Sigma \to Q$ auf eine Funktion

$$\delta^* : Q \times \Sigma^* \to Q$$

wird induktiv (über die Wortlänge) definiert:

- $\delta^*(q, \varepsilon) := q$,
- $\delta^*(q, wa) := \delta(\delta^*(q, w), a)$.

Der Einfachheit halber schreiben wir in Zukunft auch für Wörter $\delta(q, w)$ anstelle von $\delta^*(q, w)$.

Beachte: für einen DEA $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ gilt:

(a) $\delta(q, w) = q'$ gdw q' ist der eindeutige (!) Zustand mit $q \xrightarrow{w}_{\mathcal{A}} q'$.

(b) $L(\mathcal{A}) = \{w \in \Sigma^* \mid \delta(q_0, w) \in F\}$.

(c) $\delta(q, uv) = \delta(\delta(q, u), v)$.

Satz 2.4 (Rabin/Scott)

Zu jedem NEA kann man effektiv einen äquivalenten DEA konstruieren.

Es folgt, daß NEAs und DEAs dieselbe Klasse von Sprachen akzeptieren.

Beweisidee: sei $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ eine NEA.

Wann gilt für $w \in \Sigma^*$, daß $w \in L(\mathcal{A})$ ist? Dazu betrachten wir

$$\{q \in Q \mid q_0 \xrightarrow{w}_{\mathcal{A}} q\}.$$

Offenbar gilt:

$w \in L(\mathcal{A})$ gdw diese Menge enthält mindestens einen Endzustand.

Wir definieren daher einen DEA mit Zustandsmenge

$$2^Q := \{P \mid P \subseteq Q\}$$ Potenzmengenkonstruktion

für den gilt:

$$\delta(\{q_0\}, w) = \{q \in Q \mid q_0 \xrightarrow{w}_{\mathcal{A}} q\}.$$

Dresden

Beweis von Satz 2.4

Sei der NEA $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ gegeben. Der DEA $\mathcal{A}' = (2^Q, \Sigma, \{q_0\}, \delta, F')$ ist definiert durch:

- $\delta(P, a) = \bigcup_{p \in P} \{p' \mid (p, a, p') \in \Delta\}$ für alle $P \in 2^Q$ und $a \in \Sigma$
- $F' = \{P \in 2^Q \mid P \cap F \neq \emptyset\}$

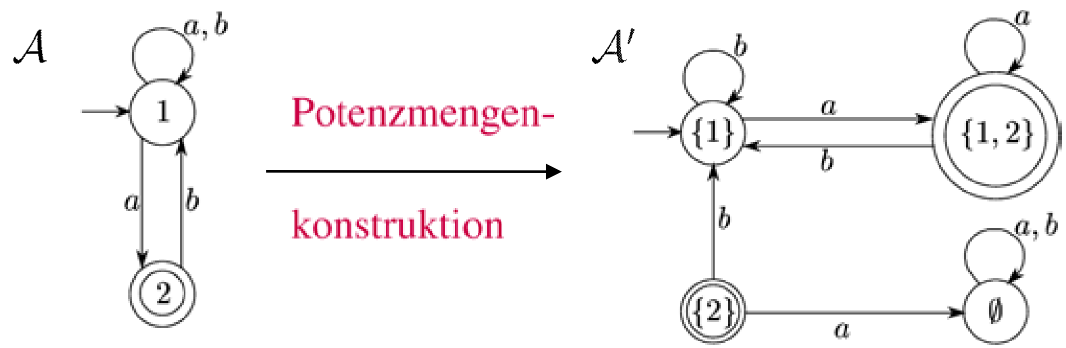
Behauptung: $q' \in \delta(\{q\}, w)$ gdw $q \xrightarrow{w}_{\mathcal{A}} q'$.

Daraus folgt $L(\mathcal{A}) = L(\mathcal{A}')$, da:

$$\begin{array}{rll} w \in L(\mathcal{A}) & \text{gdw} & \exists q \in F : q_0 \xrightarrow{w}_{\mathcal{A}} q \\ & \text{gdw} & \exists q \in F : q \in \delta(\{q_0\}, w) \\ & \text{gdw} & \delta(\{q_0\}, w) \cap F \neq \emptyset \\ & \text{gdw} & \delta(\{q_0\}, w) \in F' \\ & \text{gdw} & w \in L(\mathcal{A}') \end{array}$$

Dresden

Beispiel 2.5



Nachteil der Potenzmengenkonstruktion

Zustandsanzahl wird exponentiell vergrößert:

- im allgemeinen nicht vermeidbar (Übung);
- in vielen Fällen kommt man mit weniger Zuständen aus.

Wie findet man einen äquivalenten DEA mit minimaler Zustandsanzahl?

Minimierung endlicher deterministischer Automaten
erfolgt in zwei Schritten:

- Schritt 1: Eliminieren unerreichbarer Zustände.
- Schritt 2: Zusammenfassen äquivalenter Zustände.

Schritt 1: Eliminieren unerreichbarer Zustände

Definition 2.6 (Erreichbarkeit eines Zustandes)

Ein Zustand q des DEA $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ heißt erreichbar, falls es ein $w \in \Sigma^*$ gibt mit $\delta(q_0, w) = q$. Sonst heißt q unerreichbar.

Da für die akzeptierte Sprache nur Zustände wichtig sind, welche von q_0 erreichbar sind, erhält man durch Weglassen unerreichbarer Zustände einen äquivalenten Automaten

$$\mathcal{A}_0 := (Q_0, \Sigma, q_0, \delta_0, F_0) \text{ mit}$$

$$\begin{aligned} Q_0 &:= \{q \in Q \mid q \text{ erreichbar}\} \\ \delta_0 &:= \delta\mid_{Q_0 \times \Sigma} \\ F_0 &:= F \cap Q_0 \end{aligned}$$

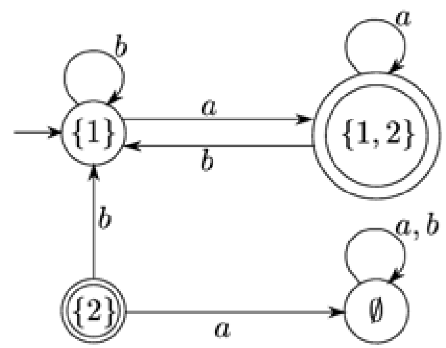
Beachte: Für $q \in Q_0$ und $a \in \Sigma$ ist auch $\delta(q, a) \in Q_0$!

Beachte: die Menge der erreichbaren Zustände kann mit linearem Aufwand (in der Zustandszahl) berechnet werden (Erreichbarkeitsproblem in Graphen).

Dresden

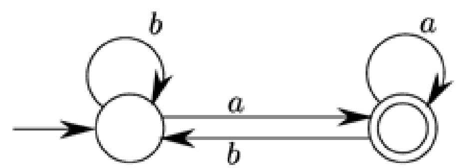
Beispiel:

Im Automaten $\mathcal{A}'$ von Beispiel 2.5



sind die Zustände $\{2\}$ und $\emptyset$ nicht erreichbar.

Durch Weglassen dieser Zustände erhält man $\mathcal{A}'_0$:



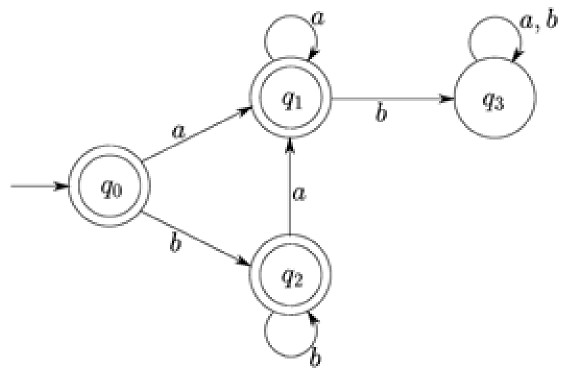
Bei der Potenzmengenkonstruktion kann man die Betrachtung unerreichbarer Elemente von 2^Q vermeiden, indem man mit $\{q_0\}$ beginnt und nur die davon erreichbaren Mengen sukzessive konstruiert.

Schritt 2: Zusammenfassen äquivalenter Zustände

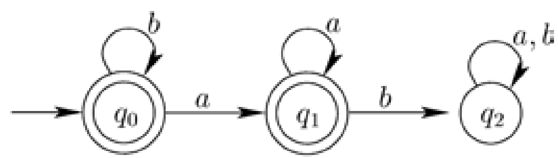
Ein DEA ohne unerreichbare Zustände muß noch nicht minimal sein, da er verschiedene Zustände enthalten kann, die sich „gleich“ verhalten.

Beispiel 2.7:

Im DEA



sind alle Zustände erreichbar,
er erkennt aber dieselbe Sprache wie der kleinere DEA



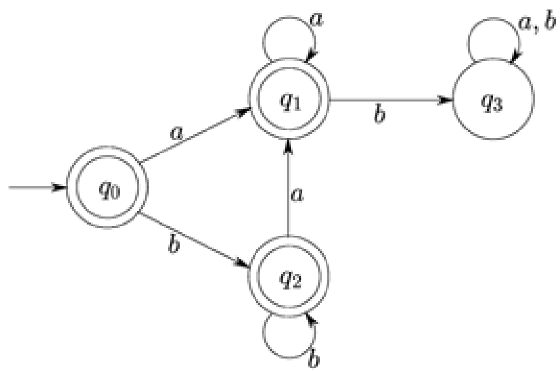
aus Beispiel 2.2.

Definition 2.8 (Äquivalenz von Zuständen)

Es sei $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ ein DEA. Für $q \in Q$ sei

$$\mathcal{A}_q = (Q, \Sigma, q, \delta, F).$$

Zwei Zustände $q, q' \in Q$ heißen $\mathcal{A}$-äquivalent $(q \sim_{\mathcal{A}} q')$ gdw $L(\mathcal{A}_q) = L(\mathcal{A}_{q'})$.



In Beispiel 2.7 sind q_0 und q_2 äquivalent:

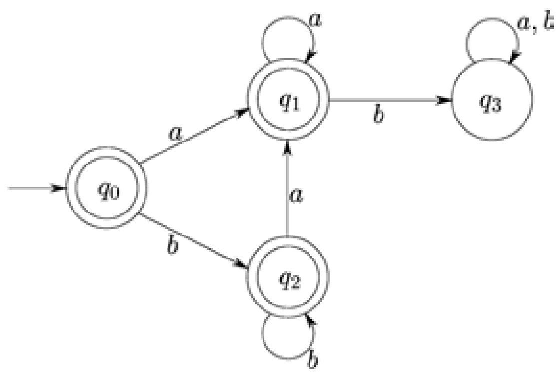
$$L(\mathcal{A}_{q_0}) = \{b\}^* \cdot \{a\}^* = L(\mathcal{A}_{q_2}).$$

Dresden

Lemma 2.9 (Eigenschaften von $\sim_{\mathcal{A}}$)

1. $\sim_{\mathcal{A}}$ ist eine Äquivalenzrelation auf Q,
   d.h. reflexiv, transitiv und symmetrisch.
2. $\sim_{\mathcal{A}}$ ist verträglich mit der Übergangsfunktion,
   d.h. $q \sim_{\mathcal{A}} q' \Rightarrow \forall a \in \Sigma : \delta(q,a) \sim_{\mathcal{A}} \delta(q',a)$.
3. $\sim_{\mathcal{A}}$ kann effektiv bestimmt werden.

Die $\sim_{\mathcal{A}}$-Klasse eines Zustands $q \in Q$ schreiben wir $\widetilde{q} := \{q' \in Q \mid q \sim_{\mathcal{A}} q'\}$.



Beispiel: für den Automaten aus Beispiel 2.7 gilt

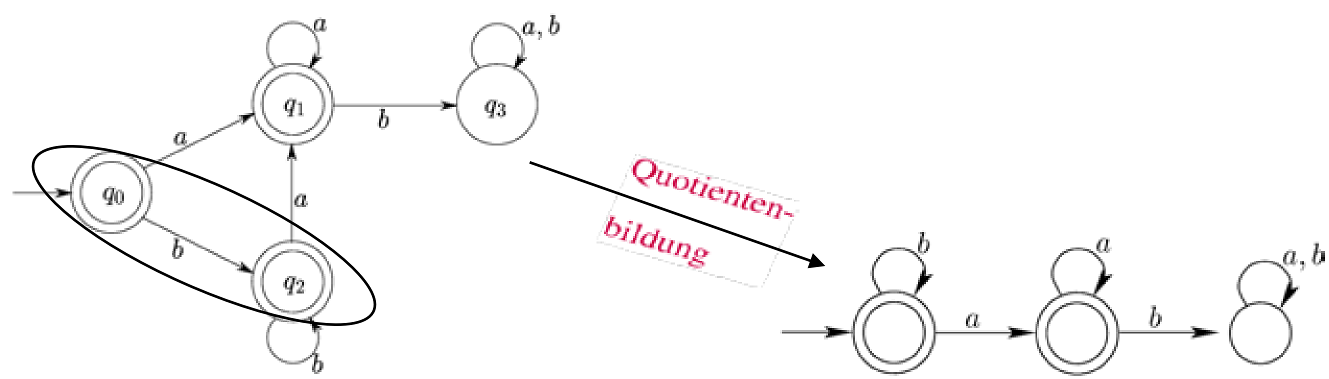
- $\sim_0$ hat die Klassen $F = \{q_0, q_1, q_2\}$ und $Q \setminus F = \{q_3\}$.
- $\sim_1$ hat die Klassen $\{q_1\}, \{q_0, q_2\}, \{q_3\}$.
  Zum Beispiel ist $\delta(q_0, b) = \delta(q_2, b) \in F$ und $\delta(q_1, b) \notin F$.
- $\sim_2 = \sim_1 = \sim_{\mathcal{A}}$.

Dresden

Definition 2.10 (Quotientenautomat)

Der Quotientenautomat $\widetilde{\mathcal{A}} = (\widetilde{Q}, \Sigma, \widetilde{q}_0, \widetilde{\delta}, \widetilde{F})$ zum DEA $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ ist definiert durch:

- $\widetilde{Q} := \{\widetilde{q} \mid q \in Q\}$
- $\widetilde{\delta}(\widetilde{q}, a) := \widetilde{\delta(q, a)}$ (repräsentantenunabhängig wegen Lemma 2.9)
- $\widetilde{F} := \{\widetilde{q} \mid q \in F\}$ (repräsentantenunabhängig nach Definition von $\sim_{\mathcal{A}}$)



Dresden

Lemma 2.11

$\widetilde{\mathcal{A}}$ ist äquivalent zu $\mathcal{A}$.

Beweis:

$$
\begin{aligned}
w \in L(\mathcal{A}) \quad & \text{gdw} \quad \delta(q_0, w) \in F \\
& \text{gdw} \quad \widetilde{\delta(q_0, w)} \in \widetilde{F} \\
& \text{gdw} \quad \widetilde{\delta}(\widetilde{q}_0, w) \in \widetilde{F} \\
& \text{gdw} \quad w \in L(\widetilde{\mathcal{A}}).
\end{aligned}
$$

Induktion über $|w|$

Dresden

Definition 2.12 (reduzierter Automat zu einem DEA)

Für einen DEA $\mathcal{A}$ bezeichnet

$$\mathcal{A}_{red} := \widetilde{\mathcal{A}}_0$$

den reduzierten Automaten, den man aus $\mathcal{A}$ durch Eliminieren unerreichbarer Zustände und Zusammenfassen äquivalenter Zustände erhält.

Wir werden im folgenden zeigen:

- $\mathcal{A}_{red}$ kann nicht weiter vereinfacht werden, d.h. er ist der kleinste DEA, mit dem man $L(\mathcal{A})$ akzeptieren kann.
- $\mathcal{A}_{red}$ hängt nur von $L(\mathcal{A})$ und nicht von $\mathcal{A}$ ab, d.h. gilt $L(\mathcal{A}) = L(\mathcal{B})$, so auch $\mathcal{A}_{red} \simeq \mathcal{B}_{red}$.

gleich bis auf Zustandsumbenennung (isomorph)

Dazu konstruieren wir zu jeder erkennbaren Sprache L einen „kanonischen“ Automaten $\mathcal{A}_L$ und zeigen, daß jeder reduzierte Automat, der L akzeptiert, isomorph zu $\mathcal{A}_L$ ist.

Definition 2.13 (Nerode-Rechtskongruenz)

Es sei $L \subseteq \Sigma^*$ eine beliebige formale Sprache. Für $u, v \in \Sigma^*$ definieren wir:

$$u \simeq_L v \;\text{ gdw }\; \forall w \in \Sigma^* : uw \in L \Leftrightarrow vw \in L.$$

Beispiel 2.14 (vgl. Beispiele 1.7, 2.2, 2.7)

$$L = \{w \in \{a,b\}^* \mid ab \text{ ist nicht Infix von } w\} = \{b\}^* \cdot \{a\}^*.$$

- Es gilt $\varepsilon \simeq_L b$:

$$\begin{array}{lll} \forall w : \varepsilon w \in L & \text{gdw} & w \in L \\ & \text{gdw} & w \text{ enthält } ab \text{ nicht} \\ & \text{gdw} & bw \text{ enthält } ab \text{ nicht} \\ & \text{gdw} & bw \in L \end{array}$$

- Außerdem gilt $\varepsilon \not\simeq_L a$: $\quad \varepsilon b \in L$, aber $a \cdot b \notin L$

Lemma 2.15 (Eigenschaften von $\simeq_L$)

1. $\simeq_L$ ist eine Äquivalenzrelation.

2. $\simeq_L$ ist Rechtskongruenz, d.h. zusätzlich zu 1. gilt:

$$u \simeq_L v \Rightarrow \forall w \in \Sigma^* : uw \simeq_L vw.$$

3. L ist Vereinigung von $\simeq_L$-Klassen:

$$L = \bigcup_{u \in L} [u],$$

wobei $[u] := \{v \mid u \simeq_L v\}$.

4. Ist $L = L(\mathcal{A})$ für einen DEA $\mathcal{A}$, so ist die Anzahl der $\simeq_L$-Klassen kleiner oder gleich der Zustandszahl von $\mathcal{A}$.

Index von $\simeq_L$

Dresden

Beispiel 2.14 (Fortsetzung)

$$L = \{w \in \{a,b\}^* \mid ab \text{ ist nicht Infix von } w\} = \{b\}^* \cdot \{a\}^*.$$

$\simeq_L$ hat drei Klassen:

- $[\varepsilon] = \{b\}^*$,
- $[a] = \{b\}^* \cdot \{a\}^+$,
- $[ab] = \Sigma^* \cdot \{ab\} \cdot \Sigma^*$.

Es ist $L = [\varepsilon] \cup [a]$.

Dresden

Man kann die $\simeq_L$-Klassen nun als Zustände eines Automaten für L verwenden.

Definition 2.16 (Transitionssystem $\mathcal{A}_L$ zu einer Sprache L)

Zu $L \subseteq \Sigma^*$ ist das Transitionssystem $\mathcal{A}_L := (Q', \Sigma, q_0', \delta', F')$ definiert durch:

- $Q' := \{[u] \mid u \in \Sigma^*\}$
- $q_0' := [\varepsilon]$
- $\delta'([u], a) := [ua]$
- $F' := \{[u] \mid u \in L\}$

$\left.\begin{array}{l} \delta'([u], a) := [ua] \\ F' := \{[u] \mid u \in L\} \end{array}\right\}$ repräsentantenunabhängig wegen Lemma 2.15

Dresden

Lemma 2.17

Hat $\simeq_L$ endlichen Index, so ist $\mathcal{A}_L$ ein DEA mit $L = L(\mathcal{A}_L)$.

Beweis:

Hat $\simeq_L$ endlich viele Klassen, so hat $\mathcal{A}_L$ endlich viele Zustände und ist daher ein DEA. Außerdem gilt:

$$\begin{aligned}
L(\mathcal{A}_L) &= \{u \mid \delta'(q'_0, u) \in F'\} \\
&= \{u \mid \delta'([\varepsilon], u) \in F'\} \\
&= \{u \mid [u] \in F'\} \qquad (\text{wegen } \delta'([u], v) = [uv]) \\
&= \{u \mid u \in L\} \\
&= L
\end{aligned}$$

Induktion über $|v|$

Dresden

Satz 2.18 (Satz von Nerode)

Ein formale Sprache L ist erkennbar gdw $\simeq_L$ endlichen Index hat.

Beweis:

„$\Rightarrow$“: Ergibt sich unmittelbar aus Lemma 2.15.

„$\Leftarrow$“: Ergibt sich unmittelbar aus Lemma 2.17,
da $\mathcal{A}_L$ DEA ist, der L akzeptiert.

Beispiel 2.18 (eine nicht-erkennbare Sprache)

$L := \{a^n b^n \mid n \geq 0\}$ ist nicht erkennbar, da für $n \neq m$ gilt:

$$a^n \not\simeq_L a^m.$$

Grund: es ist $a^n b^n \in L$, aber $a^m b^n \notin L$.

Die $\simeq_L$-Klassen $[a^0], [a^1], [a^2], \ldots$ sind also alle verschieden.
Daher hat $\simeq_L$ unendlichen Index.

Weitere Methoden zum Nachweis der Nichterkennbarkeit später (§3).

Zunächst: Zusammenhang zwischen reduziertem Automaten und Neroderechtskongruenz.

Definition 2.20 (Isomorphie von Automaten)

Zwei DEAs $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ und $\mathcal{A}' = (Q', \Sigma, q_0', \delta', F')$ heißen isomorph ($\mathcal{A} \simeq \mathcal{A}'$) gdw es eine Bijektion $\mathrm{f} : Q \to Q'$ gibt mit:

- $\mathrm{f}(q_0) = q_0'$
- $\mathrm{f}(F) = F'$, wobei $\mathrm{f}(F) := \{\mathrm{f}(q) \mid q \in F\}$
- $\mathrm{f}(\delta(q, a)) = \delta'(\mathrm{f}(q), a)$ für alle $q \in Q$, $a \in \Sigma$

Isomorph heißt: identisch bis auf die Benennung der Zustände.

Lemma 2.21 (Isomorphie erhält die Sprache)

$$\boxed{\mathcal{A} \simeq \mathcal{A}' \;\Rightarrow\; L(\mathcal{A}) = L(\mathcal{A}')}$$

Beweis:

Es sei $\mathrm{f} : Q \to Q'$ der Isomorphismus. Durch Induktion über $|w|$ zeigt man leicht, daß

$$\mathrm{f}(\delta(q, w)) = \delta'(\mathrm{f}(q), w).$$

Daher gilt:

$$\begin{array}{lll} w \in L(\mathcal{A}) & \text{gdw} & \delta(q_0, w) \in F \\ & \text{gdw} & \mathrm{f}(\delta(q_0, w)) \in F' \qquad (\text{wegen } F' = \mathrm{f}(F)) \\ & \text{gdw} & \delta'(\mathrm{f}(q_0), w) \in F' \\ & \text{gdw} & \delta'(q_0', w) \in F' \qquad (\text{wegen } q_0' = \mathrm{f}(q_0)) \\ & \text{gdw} & w \in L(\mathcal{A}') \end{array}$$

Satz 2.22 (der reduzierte Automat ist minimal und eindeutig)

Es sei L eine erkennbare Sprache. Dann gilt:

1) $\mathcal{A}_L$ hat minimale Zustandszahl unter allen DEAs, welche L erkennen.

2) Ist $\mathcal{A}$ ein DEA mit $L(\mathcal{A}) = L$, so ist der reduzierte Automat $\mathcal{A}_{red} := \tilde{\mathcal{A}}_0$ isomorph zu $\mathcal{A}_L$.

Beweis:

1. Mit Satz 2.18 hat $\simeq_L$ endlichen Index. Daher ist mit Lemma 2.17 $\mathcal{A}_L$ ein DEA für L. Mit Lemma 2.15 hat jeder DEA, der L akzeptiert, mindestens soviele Zustände, wie $\simeq_L$ Klassen (d.h. wie $\mathcal{A}_L$ Zustände) hat.

2. Es sei $\mathcal{A}_{red} = (Q, \Sigma, q_0, \delta, F)$ und $\mathcal{A}_L = (Q', \Sigma, q'_0, \delta', F')$.

Dresden

Zur Erinnerung:

Der Automat $\mathcal{A}_L := (Q', \Sigma, q_0', \delta', F')$ ist definiert durch:

- $Q' := \{[u] \mid u \in \Sigma^*\}$
- $q_0' := [\varepsilon]$
- $\delta'([u], a) := [ua]$
- $F' := \{[u] \mid u \in L\}$

$\mathcal{A}_{red} = \tilde{\mathcal{A}}_0 = (Q, \Sigma, q_0, \delta, F)$ erhält man aus $\mathcal{A}$ durch

- Entfernen unerreichbarer Zustände
- Identifizieren äquivalenter Zustände

Wir definieren eine Funktion

$$\mathrm{f} : Q \to Q'$$

und zeigen, daß sie ein Isomorphismus ist.

Für $q \in Q$ existiert (mindestens) ein Wort $w_q \in \Sigma^*$ mit $\delta(q_0, w_q) = q$, da in $\mathcal{A}_{red}$ alle Zustände erreichbar sind. O.B.d.A. sei $w_{q_0} = \varepsilon$.

Wir definieren $\mathrm{f}(q) := [w_q]$.

Dresden

Korollar 2.23

Es seien $\mathcal{A}$ und $\mathcal{A}'$ DEAs. Dann gilt:
$L(\mathcal{A}) = L(\mathcal{A}')$ gdw $\mathcal{A}_{red} \simeq \mathcal{A}'_{red}$.

Man erhält somit eine (allerdings ineffiziente) Methode, um von zwei DEAs zu entscheiden, ob sie dieselbe Sprache akzeptieren (d.h. das Äquivalenzproblem zu entscheiden):

- konstruiere die reduzierten Automaten wie beschrieben
- stelle fest, ob die so erhaltenen reduzierten DEAs isomorph sind (teste alle Bijektionen)

Hat man NEAs gegeben, so kann man diese zuerst deterministisch machen (Potenzmengenkonstruktion) und dann das Verfahren für DEAs anwenden.