Automaten und Formale Sprachen

Teil I: Endliche Automaten

0. Einführung
1. Nichtdeterministische endliche Automaten
2. Deterministische endliche Automaten
3. Nachweis der Nichterkennbarkeit
4. Abschlußeigenschaften und Entscheidungsprobleme
5. Reguläre Ausdrücke und Sprachen

§ 3. Nachweis der Nichterkennbarkeit

Nachweis der Erkennbarkeit: konstruiere einen endlichen Automaten (NEA, DEA) für die Sprache.

Nachweis der Nichterkennbarkeit: ist schwieriger, da man zeigen muß, daß es keinen endlichen Automaten für die Sprache geben kann.

Lemma 3.1 (Pumping-Lemma: einfache Version)

Es sei L eine erkennbare Sprache. Dann gibt es eine natürliche Zahl $n_0 \geq 1$, so daß gilt:

Jedes Wort $w \in L$ mit $|w| \geq n_0$ läßt sich zerlegen in $w = xyz$ mit

- $y \neq \varepsilon$
- $xy^k z \in L$ für alle $k \geq 0$.

Dresden

Beispiel 3.2 (eine nicht erkennbare Sprache)

$L = \{a^n b^n \mid n \geq 0\}$ ist nicht erkennbar.

Beweis:

Angenommen, L ist erkennbar.

Dann gibt es eine Zahl n_0 mit den in Lemma 3.1 beschriebenen Eigenschaften.

Es sei nun n so, daß $2n \geq n_0$ ist. Da

$$a^n b^n \in L$$

ist, hat es eine Zerlegung

$$a^n b^n = xyz$$

mit $y \neq \varepsilon$ und $xy^k z \in L$ für alle $k \geq 0$.

Unterscheide drei Fälle je nach der Lage von y in $a^n b^n$:

aaaaaaaaaaaaaaaaaaaaaaaaaaabbbbbbbbbbbbbbbbbbbbbbb

y — 1. Fall | y — 3. Fall | y — 2. Fall

Dresden

Satz 3.3 (Leerheitsproblem entscheidbar)

Es sei L eine erkennbare Sprache (gegeben durch NEA oder DEA). Dann kann man effektiv entscheiden, ob $L = \emptyset$ oder nicht.

Beweis:

Es sei L erkennbar und n_0 die Zahl aus dem Pumping Lemma.
Dann gilt:

$$(*) \quad L \neq \emptyset \text{ gdw } \exists w \in L \text{ mit } |w| < n_0.$$

Entscheide nun $L = \emptyset$ wie folgt:

- Betrachte die endlich vielen Wörter $w \in \Sigma^*$ mit $|w| < n_0$.
- Für jedes solche w kann $w \in L$ leicht entschieden werden; z.B. durch Eingabe in DEA (siehe auch §4, Wortproblem).

Zu zeigen bleibt $(*)$.

Beispiel 3.4 (eine nicht erkennbare Sprache, die Lemma 3.1 erfüllt)

$$\boxed{L = \{a^n b^m \mid n \neq m\}}$$

Versucht man, Nichterkennbarkeit mit Lemma 3.1 zu zeigen, so scheitert man, da das Lemma für L zutrifft:

Wähle $n_0 := 3$.

Es sei nun $w \in L$ mit $|w| \geq 3$, d.h.

$$w = a^n b^m, \;\; n \neq m \;\text{ und }\; n + m \geq 3.$$

Zu zeigen: es gibt x, y, z mit $w = xyz$, $y \neq \varepsilon$ und

$$xy^k z \in L \;\text{ für alle }\; k \geq 0.$$

1. Fall: $n > m$, d.h. $n = m + i$ für $i \geq 1$.

2. Fall: $n < m$ kann entsprechend behandelt werden.

Wie zeigt man Nichterkennbarkeit von L?

Lemma 3.5 (Pumping-Lemma: verschärfte Version)

Es sei L eine erkennbare Sprache. Dann gibt es eine natürliche Zahl $n_0 \geq 1$, so daß gilt:

Für alle Wörter $u, v, w \in \Sigma^*$ mit $uvw \in L$ und $|v| \geq n_0$ gibt es eine Zerlegung $v = xyz$ mit

- $y \neq \varepsilon$
- $uxy^kzw \in L$ für alle $k \geq 0$.

Vorteil dieses Lemmas beim Nachweis der Nichterkennbarkeit?

Man weiß, daß man beliebig positionierte Teilstücke zerlegen kann, wenn sie nur lang genug sind.

Dadurch kann man dafür sorgen, daß das y an einer geeigneten Stelle liegt (im Beispiel 3.4 im kürzeren Block).

Beispiel 3.4 (Fortsetzung)

$\boxed{L = \{a^n b^m \mid n \neq m\}}$ ist nicht erkennbar.

Beweis: Angenommen, doch; dann gibt es $n_0 \geq 1$, das die in Lemma 3.5 geforderten Eigenschaften hat.

Wir betrachten nun die Wörter

$$u := \varepsilon, \quad v := a^{n_0}, \quad w := b^{n_0!+n_0}$$

Offenbar ist $uvw = a^{n_0} b^{n_0!+n_0} \in L$.

Mit Lemma 3.5 gibt es eine Zerlegung $v = xyz$ mit $y \neq \varepsilon$ und $uxy^kzw \in L$ für alle $k \geq 0$.

Es sei

$$x = a^{n_1}, \quad y = a^{n_2}, \quad z = a^{n_3} \quad \text{wobei } n_1 + n_2 + n_3 = n_0 \text{ und } n_2 > 0$$

Offenbar existiert ein l mit $n_2 \cdot l = n_0!$ (da $0 < n_2 \leq n_0$).

Es ist nun aber $n_1 + (l+1)n_2 + n_3 = n_0 + n_0!$,

was $uxy^{l+1}zw \notin L$ liefert. $\lightning$

Dresden

Automaten und Formale Sprachen

Teil I: Endliche Automaten

0. Einführung
1. Nichtdeterministische endliche Automaten
2. Deterministische endliche Automaten
3. Nachweis der Nichterkennbarkeit
4. Abschlußeigenschaften und Entscheidungsprobleme
5. Reguläre Ausdrücke und Sprachen

Dresden

Satz 4.1 (Abschlußeigenschaften erkennbarer Sprachen)

Sind L_1, L_2 erkennbar, so auch

1. $L_1 \cup L_2$ (Vereinigung)
2. $\overline{L_1}$ (Komplement)
3. $L_1 \cap L_2$ (Durchschnitt)
4. $L_1 \setminus L_2$ (Differenz)
5. $L_1 \cdot L_2$ (Konkatenation)
6. L_1^* (Kleene-Stern)

Beweis von Satz 4.1:

Es sei $\mathcal{A}_i = (Q_i, \Sigma, q_{0i}, \Delta_i, F_i)$ ein NEA für L_i $(i = 1, 2)$.

O.B.d.A. sei $Q_1 \cap Q_2 = \emptyset$.

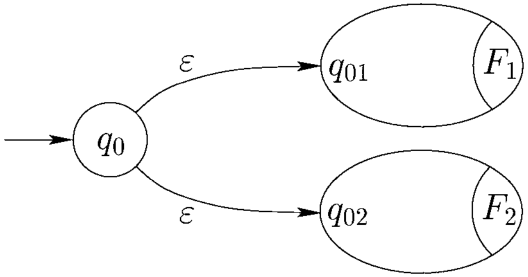
1. Abschluß unter Vereinigung:

Der folgende ε-NEA akzeptiert $L_1 \cup L_2$:

$$\mathcal{A} := (Q_1 \cup Q_2 \cup \{q_0\}, \Sigma, q_0, \Delta, F_1 \cup F_2),$$

wobei $q_0 \notin Q_1 \cup Q_2$ und

$$\Delta := \Delta_1 \cup \Delta_2 \cup \{(q_0, \varepsilon, q_{01}), (q_0, \varepsilon, q_{02})\}.$$



Mit Lemma 1.12 gibt es zu $\mathcal{A}$ einen äquivalenten NEA.

2. Abschluß unter Komplement:

Einen DEA für $\overline{L_1}$ erhält man wie folgt: Zunächst verwendet man die Potenzmengenkonstruktion, um zu $\mathcal{A}_1$ einen äquivalenten DEA $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ zu konstruieren.

Der DEA für $\overline{L_1}$ ist nun

$$\overline{\mathcal{A}} := (Q, \Sigma, q_0, \delta, Q \setminus F).$$

Es gilt nämlich:

$$\begin{array}{lll} w \in \overline{L_1} & \text{gdw.} & w \notin L(\mathcal{A}_1) \\ & \text{gdw} & w \notin L(\mathcal{A}) \\ & \text{gdw} & \delta(q_0, w) \notin F \\ & \text{gdw} & \delta(q_0, w) \in Q \setminus F \\ & \text{gdw} & w \in L(\overline{\mathcal{A}}) \end{array}$$

Beachte: Man darf dies nicht direkt mit dem NEA machen!

Dresden

3. Abschluß unter Durchschnitt:

Folgt aus 1) und 2), da

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

Da aber Abschluß unter Komplement die Potenzmengenkonstruktion benötigt und diese exponentiell sein kann, ist es günstiger, direkt einen NEA für $L_1 \cap L_2$ zu konstruieren.

Der Produktautomat:

$$\mathcal{A} := (Q_1 \times Q_2, \Sigma, (q_{01}, q_{02}), \Delta, F_1 \times F_2)$$

mit

$$\Delta \;:=\; \{((q_1, q_2), a, (q_1', q_2')) \;\mid\; \begin{array}{l} (q_1, a, q_1') \in \Delta_1 \text{ und} \\ (q_2, a, q_2') \in \Delta_2 \end{array} \}.$$

Ein Übergang in $\mathcal{A}$ ist also genau dann möglich, wenn der entsprechende Übergang in $\mathcal{A}_1$ und $\mathcal{A}_2$ möglich ist. Daraus ergibt sich leicht

$$L(\mathcal{A}) = L_1 \cap L_2.$$

4. Abschluß unter Differenz:

Folgt aus 1) und 2), da

$$L_1 \setminus L_2 = L_1 \cap \overline{L_2}$$

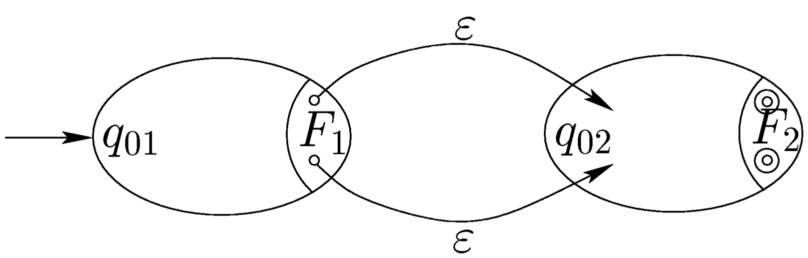
5. Abschluß unter Konkatentation:

Der folgende ε-NEA akzeptiert $L_1 \cdot L_2$:

$$\mathcal{A} := (Q_1 \cup Q_2, \Sigma, q_{01}, \Delta, F_2),$$

wobei

$$\Delta := \Delta_1 \cup \Delta_2 \cup \{(f, \varepsilon, q_{02}) \mid f \in F_1\}.$$



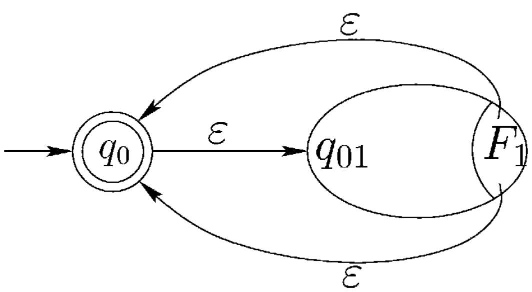
6. Abschluß unter Kleene-Stern:

Der folgende ε-NEA akzeptiert L_1^*:

$$\mathcal{A} := (Q_1 \cup \{q_0\}, \Sigma, q_0, \Delta, \{q_0\}),$$

wobei $q_0 \notin Q_1$ und

$$\Delta := \Delta_1 \cup \{(f, \varepsilon, q_0) \mid f \in F_1\} \cup \{(q_0, \varepsilon, q_{01})\}.$$



Beachte: Alle angegebenen Konstruktionen sind effektiv.

Die Automaten für $L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 \cdot L_2$ und L_1^* sind polynomiell in der Größe der Automaten für L_1, L_2.

Beim Komplement kann die Konstruktion exponentiell sein, wenn man von einem NEA ausgeht.

Dresden

Beispiel 4.2 (Abschlußeigenschaften zum Nachweis der Nichterkennbarkeit)

$\boxed{L = \{a^n b^m \mid n \neq m\}}$ ist nicht erkennbar.

Beweis:

Anstatt dies direkt mit Lemma 3.5 zu zeigen, kann man auch verwenden, daß bereits bekannt ist, daß

$$L' := \{a^n b^n \mid n \geq 0\}$$

nicht erkennbar ist.

Wäre nämlich L erkennbar, so mit Satz 4.1 auch

$$L' = \overline{L} \cap \{a\}^* \cdot \{b\}^*.$$

Da wir schon wissen, daß L' nicht erkennbar ist, kann auch L nicht erkennbar sein.

Dresden

Entscheidungsprobleme für erkennbare Sprachen:

Wir betrachten

- das Leerheitsproblem: Ist $L \neq \emptyset$?
- das Wortproblem: Ist $w \in L$?
- das Äquivalenzproblem: Ist $L_1 = L_2$?

Dabei gehen wir davon aus, daß die erkennbare Sprache durch einen DEA oder NEA gegeben ist.

Bzgl. Entscheidbarkeit macht es keinen Unterschied, ob man einen DEA oder NEA gegeben hat:

Aus NEA kann man effektiv äquivalenten DEA konstruieren (Satz 2.4)!

Bzgl. Komplexität kann der Unterschied aber relevant sein:

Übergang NEA $\longrightarrow$ DEA kann exponentiell sein (Potenzmengenkonstruktion)!

Das Leerheitsproblem

Gegeben:	erkennbare Sprache L (als DEA oder NEA)
Frage:	Ist $L \neq \emptyset$?

Wir wissen bereits (Satz 3.3), daß das Problem entscheidbar ist:

Es sei $\mathcal{A}$ eine NEA mit n_0 Zuständen. Dann gilt:

$$L(\mathcal{A}) \neq \emptyset \quad \text{gdw} \quad \exists w \in L(\mathcal{A}) \text{ mit } |w| < n_0.$$

Allerdings ist das so erhaltene Entscheidungsverfahren viel zu aufwendig (exponentiell viele Wörter der Länge $< n_0$, falls $|\Sigma| > 1$).

Dresden

Satz 4.3

Es sei $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ ein NEA. Dann kann man „$L(\mathcal{A}) \neq \emptyset$?“ in der Zeit $O(|Q| + |\Delta|)$ entscheiden.

Beweis:

Man kann $\mathcal{A}$ als gerichteten Graphen $G = (Q, E)$ auffassen mit

$$E := \{(q_1, q_2) \mid (q_1, a, q_2) \in \Delta \text{ für ein } a \in \Sigma\}.$$

Dann gilt:

$L(\mathcal{A}) \neq \emptyset$ gdw in der von q_0 aus erreichbaren Knotenmenge befindet sich ein Endzustand

Die von q_0 aus erreichbaren Knoten kann man mit Aufwand $O(|Q| + |E|)$ berechnen (vgl. Vorlesung „Algorithmen und Datenstrukturen“).

Insbesondere ist also das Leerheitsproblem für erkennbare Sprachen mit polynomiellem Aufwand lösbar.

Das Wortproblem

Gegeben: erkennbare Sprache $L \subseteq \Sigma^*$, Wort $w \in \Sigma^*$

Frage: Gilt $w \in L$?

Ist $L = L(\mathcal{A})$ für einen DEA $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$, so kann man einfach, beginnend mit q_0, durch Anwendung von δ berechnen, zu welchem Zustand in $\mathcal{A}$ man mit w kommt und prüfen, ob dies ein Endzustand ist.

Nimmt man Σ als konstant an, so benötigt eine Anwendung von δ nur konstante Zeit.

Dies liefert:

Satz 4.4

Es sei $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ ein DEA und $w \in \Sigma^*$.
Dann kann man „$w \in L(\mathcal{A})$?“ in der Zeit $O(|w|)$ entscheiden.

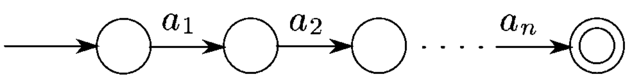
Für einen NEA ist dies nicht so einfach, da man ja verschiedene mit w beschriftete Pfade haben kann und man diese (im schlimmsten Fall) alle betrachten muß, um festzustellen, ob einer davon mit einem Endzustand aufhört.

Satz 4.5

Es sei $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ ein NEA und $w \in \Sigma^*$.
Dann kann man „$w \in L(\mathcal{A})$?“ in der Zeit $O(|w| \cdot (|Q| + |\Delta|))$ entscheiden.

Beweis:

Konstruiere zunächst einen Automaten $\mathcal{A}_w$, der genau $w = a_1 \ldots a_n$ akzeptiert, d.h. $L(\mathcal{A}_w) = \{a_1 \ldots a_n\}$:



Dieser Automat hat $|w| + 1$ Zustände.

Offenbar ist

$$w \in L(\mathcal{A}) \text{ gdw } L(\mathcal{A}) \cap L(\mathcal{A}_w) \neq \emptyset.$$

Dresden

$L(\mathcal{A}) \cap L(\mathcal{A}_w)$ wird vom Produktautomaten zu $\mathcal{A}$ und $\mathcal{A}_w$ akzeptiert.

Für diesen können wir das Resultat für das Leerheitsproblem (Satz 4.3) anwenden (Aufwand linear in Zustandszahl plus Größe der Übergangsrelation).

Wie groß ist nun dieser Produktautomat?

Zustände:

$|Q| \cdot (|w| + 1)$

Übergänge:

Für jeden Übergang $\bigcirc \xrightarrow{a_i} \bigcirc$ in $\mathcal{A}_w$ gibt es maximal $|\Delta|$ mögliche Übergänge in $\mathcal{A}$. Also insgesamt maximal $|w| \cdot |\Delta|$ viele Übergänge im Produktautomaten.

Nach Satz 4.3 ist daher der Aufwand zum Testen von $L(\mathcal{A}) \cap L(\mathcal{A}_w) \neq \emptyset$:

$$O(|Q| \cdot (|w| + 1) + |w| \cdot |\Delta|) = O(|w| \cdot (|Q| + |\Delta|))$$

Das Äquivalenzproblem

Gegeben: erkennbare Sprachen L_1, L_2.

Frage: Gilt $L_1 = L_2$?

Wie bereits erwähnt, kann man das Äquivalenzproblem auf das Leerheitsproblem reduzieren:

$$L_1 = L_2 \text{ gdw } (L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2) = \emptyset$$

Satz 4.6

Das Äquivalenzproblem ist für erkennbare Sprachen entscheidbar.

Sind die Sprachen durch DEAs gegeben, so ist dies in polynomieller Zeit möglich.

Beweis von Satz 4.6:

Die Automatenkonstruktionen, welche Abschluß unter Vereinigung, Durchschnitt und Komplement zeigen, sind effektiv.

Daraus ergibt sich Entscheidbarkeit (auch bei NEAs).

Die Konstruktionen für Vereinigung und Durchschnitt sind polynomiell.
Beim Komplement ist dies nur dann der Fall, wenn bereits DEAs vorliegen.

Wir werden später sehen, daß sich der exponentielle Zeitaufwand für die Potenzmengenkonstruktion (wahrscheinlich) bei NEAs nicht vermeiden läßt (Abschnitt zu Komplexitätstheorie der Vorlesung GThI 2).

Automaten und Formale Sprachen

Teil I: Endliche Automaten

§ 5. Reguläre Ausdrücke und Sprachen

Verschiedene äquivalente Charakterisierungen der Klasse der erkennbaren Sprachen mit Hilfe von Transitionssystemen und Rechtskongruenzen:

Eine Sprache $L \subseteq \Sigma^*$ ist erkennbar gdw

1. $L = L(\mathcal{A})$ für einen NEA $\mathcal{A}$.
2. $L = L(\mathcal{A})$ für einen ε-NEA $\mathcal{A}$.
3. $L = L(\mathcal{A})$ für einen NEA mit Wortübergängen $\mathcal{A}$.
4. $L = L(\mathcal{A})$ für ein endliches Transitionssystem $\mathcal{A}$.
5. $L = L(\mathcal{A})$ für einen DEA $\mathcal{A}$.
6. Die Nerode-Rechtskongruenz $\simeq_L$ hat endlichen Index.

Im folgenden betrachten wir eine weitere Charakterisierung mit Hilfe regulärer Ausdrücke.

Definition 5.1 (Syntax regulärer Ausdrücke)

Es sei Σ ein endliches Alphabet. Die Menge Reg_Σ der regulären Ausdrücke über Σ ist induktiv definiert:

- $\emptyset$, ε, a (für $a \in \Sigma$) sind Elemente von Reg_Σ.
- Sind $r, s \in Reg_\Sigma$, so auch $(r + s)$, $(r \cdot s)$, $r^* \in Reg_\Sigma$.

Beispiel 5.2

$((a \cdot b^*) + \emptyset^*)^* \in Reg_\Sigma$ für $\Sigma = \{a, b\}$.

Notation: wir lassen Außenklammern weg und vereinbaren:

- * bindet stärker als $\cdot$,
- $\cdot$ bindet stärker bindet als $+$,
- $\cdot$ wird meist ganz weggelassen.

$(ab^* + \emptyset^*)^*$

Dresden

Definition 5.3 (Semantik regulärer Ausdrücke)

Die durch den regulären Ausdruck r definierte Sprache $L(r)$ ist induktiv definiert:

- $L(\emptyset) := \emptyset, \quad L(\varepsilon) := \{\varepsilon\}, \quad L(a) := \{a\},$
- $L(r+s) := L(r) \cup L(s), \quad L(r \cdot s) := L(r) \cdot L(s), \quad L(r^*) := L(r)^*.$

Eine Sprache $L \subseteq \Sigma^*$ heißt regulär, falls es ein $r \in Reg_\Sigma$ gibt mit $L = L(r)$.

Beispiel 5.4

- $(a+b)^*ab(a+b)^*$ definiert die Sprache aller Wörter über $\{a, b\}$, die Infix ab haben.
- $L(ab^* + b) = \{ab^i \mid i \geq 0\} \cup \{b\}$

Bemerkung: Statt $L(r)$ schreiben wir im folgenden häufig einfach r.

- $abb \in ab^* + b \quad$ (eigentlich $abb \in L(ab^* + b)$)
- $(ab)^*a = a(ba)^* \quad$ (eigentlich $L((ab)^*a) = L(a(ba)^*)$)

Wir zeigen nun, daß man mit regulären Ausdrücken genau die erkennbaren Sprachen beschreiben kann.

Satz 5.5 (Satz von Kleene)

Für eine Sprache $L \subseteq \Sigma^*$ sind äquivalent:

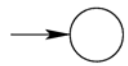
1) L ist regulär.

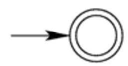
2) L ist erkennbar.

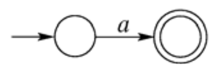
Beweis:

„1 $\rightarrow$ 2“: Induktion über den Aufbau regulärer Ausdrücke

Verankerung:

- $L(\emptyset) = \emptyset$ erkennbar:  ist NEA für $\emptyset$ (kein Endzustand).

- $L(\varepsilon) = \{\varepsilon\}$ erkennbar:  ist NEA für $\{\varepsilon\}$.

- $L(a) = \{a\}$ erkennbar: 

ist NEA für $\{a\}$.

Schritt:

Weiß man bereits, daß $L(r)$ und $L(s)$ erkennbar sind,
so folgt mit Satz 4.1 (Abschlußeigenschaften), daß auch

- $L(r + s) = L(r) \cup L(s)$,

- $L(r \cdot s) = L(r) \cdot L(s)$ und

- $L(r^*) = L(r)^*$

erkennbar sind.

„2 $\rightarrow$ 1“:

Sei $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ ein NEA mit $L = L(\mathcal{A})$.

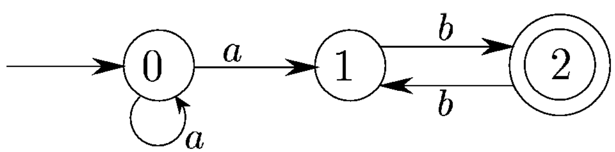
Wie in Definition 2.8 sei für $q \in Q$

$$\mathcal{A}_q := (Q, \Sigma, q, \Delta, F) \quad \text{und} \quad L_q := L(\mathcal{A}_q),$$

d.h. insbesondere $L = L_{q_0}$.

Der Zusammenhang zwischen den L_qs kann nun als Gleichungssystem beschrieben werden, dessen Lösungen eindeutig bestimmte reguläre Sprachen sind.

Beispiel 5.6



$$\begin{aligned} L_0 &= \{a\} \cdot L_0 \cup \{a\} \cdot L_1 \\ L_1 &= \{b\} \cdot L_2 \\ L_2 &= \{b\} \cdot L_1 \underline{\cup \{\varepsilon\}} \end{aligned}$$

2 ist Endzustand

Allgemeine Form des Gleichungssystems:

Es sei $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ ein NEA und die Sprachen L_p für $p \in Q$ wie oben definiert.

Für $p, q \in Q$ sei

$$A_{p,q} = \{a \in \Sigma \mid (p, a, q) \in \Delta\}$$

und für $p \in Q$ sei

$$B_p = \begin{cases} \{\varepsilon\} & \text{falls} \quad p \in F \\ \emptyset & \text{falls} \quad p \notin F \end{cases}$$

Damit erfüllen die Sprachen L_p die folgenden Gleichungen:

Für alle $p \in Q$ gilt:

$$L_p = (\bigcup_{q \in Q} A_{p,q} \cdot L_q) \cup B_p$$

Allgemeine Form des Gleichungssystems (Fortsetzung):

Behauptung:

Das Gleichungssystem

$$(*) \qquad X_p = (\bigcup_{q \in Q} A_{p,q} \cdot X_q) \cup B_p \quad (p \in Q)$$

hat genau eine Lösung und diese besteht aus regulären Sprachen.

Aus der Behauptung folgt, daß die Sprachen L_p regulär sind, da sie ja das System $(*)$ lösen.

Wir zeigen zunächst, wie man eine einzige Gleichung der Form

$$(**) \qquad X = A \cdot X \cup B$$

eindeutig lösen kann.

Daraus ergibt sich dann die Lösung von Gleichungssystemen der Form $(*)$ durch Induktion über die Anzahl der Variablen.

Dresden

Lemma 5.7 (Arden)

Es seien $A, B \subseteq \Sigma^*$ und $\varepsilon \notin A$.
Die Gleichung

$$(**) \quad X = A \cdot X \cup B$$

hat als eindeutige Lösung $X = A^* \cdot B$.

Beachte: Sind A, B regulär, so auch $A^* B$.

Dresden

Beweis von „$2 \to 1$“ des Satzes von Kleene mit Ardens Lemma:

Aus Ardens Lemma folgt für die Gleichung $(*)$, die wir als Gleichung der Form $(**)$ für ein fixiertes $p \in Q$ auffassen:

$$X_p = A_{p,p} \cdot X_p \cup \left(\left(\bigcup_{p \neq q} A_{p,q} \cdot X_q \right) \cup B_p \right)$$

hat als eindeutige Lösung

$$X_p = A^*_{p,p} \cdot \left(\left(\bigcup_{q \neq p} A_{p,q} \cdot X_q \right) \cup B_p \right)$$

Setzt man diese Lösung in $(*)$ ein, so erhält man ein System mit einer Variablen weniger.

Nach Induktion hat dieses eine eindeutige Lösung, die aus regulären Sprachen besteht.

Da in obiger Lösung für X_p nur reguläre Operationen $(\cup, \cdot, ^*)$ verwendet werden, ist auch diese Lösung regulär (und eindeutig nach Arden).

Beispiel 5.6 (Fortsetzung)

$$\begin{aligned} X_0 &= \{a\} \cdot X_0 \cup \{a\} \cdot X_1 \\ X_1 &= \{b\} \cdot X_2 \\ X_2 &= \{b\} \cdot X_1 \cup \{\varepsilon\} \end{aligned}$$

Auflösen nach X_0: $X_0 = \{a\}^* \cdot \{a\} \cdot X_1$

Einsetzen ändert die anderen Gleichungen nicht.

Auflösen nach X_1: $X_1 = \emptyset^* \cdot \{b\} \cdot X_2 = \{b\} \cdot X_2$

Einsetzen liefert: $X_2 = \{b\} \cdot \{b\} \cdot X_2 \cup \{\varepsilon\}$

Auflösen nach X_2: $X_2 = (\{b\} \cdot \{b\})^*$

Damit ist $X_1 = \{b\} \cdot (\{b\} \cdot \{b\})^*$ und

$X_0 = \{a\}^* \cdot \{a\} \cdot \{b\} \cdot (\{b\} \cdot \{b\})^* = L(\mathcal{A})$.

Als regulärer Ausdruck: $L(\mathcal{A}) = L(a^*ab(bb)^*)$.

Beweis von Ardens Lemma:

(1) A^*B ist Lösung:

$$A \cdot A^* \cdot B \cup B = (A \cdot A^* \cup \{\varepsilon\}) \cdot B = A^* \cdot B$$

(2) Eindeutigkeit:

Es sei L eine Lösung, d.h. $L = AL \cup B$.

(2.1) $A^*B \subseteq L$:

Wir zeigen durch Induktion über n: $A^nB \subseteq L$.

- $A^0B = B \subseteq AL \cup B = L$
- Gelte $A^nB \subseteq L$.

  Es folgt: $A^{n+1}B = AA^nB \subseteq AL \subseteq AL \cup B = L$

Wegen $A^*B = \bigcup_{n \geq 0} A^nB$ folgt daraus $A^*B \subseteq L$.

(2.2) $L \subseteq A^*B$: Angenommen, dies gilt nicht.

Es sei $w \in L$ ein Wort minimaler Länge mit $w \notin A^*B$.