5 Vorlesung vom 20. April 2000

5.1 Über die Äquivalenz von DFA und NFA [1]

Da jeder DFA ein NFA ist, ist es klar[2], dass die Klasse der Sprachen, welche ein NFA akzeptiert die Sprachen enthält, die von einem DFA akzeptiert werden. Es stellt sich jedoch heraus, dass auch nur diese Sprachen von NFA akzeptiert werden. Der Beweis dafür baut darauf auf, dass gezeigt werden kann, dass ein DFA einen NFA simulieren kann. (Für jeden NFA kann ein equivalenter DFA konstruiert werden.) Der DFA simuliert einen NFA indem die einzelnen Zustände des DFA den Mengen von Zuständen des NFA entsprechen. Der so konstruierte DFA überwacht all die Zustände, die der NFA durch die selbe Eingabefolge erreichen (wie der DFA) könnte. Somit gelangt man zu dem folgendem Theorem:

5.1.1 Theorem

Wenn L eine akzeptierte Sprache für einen nichdeterministischen endlichen Automaten ist, dann existiert ein deterministischer endlicher Automat, der auch L akzeptiert. In anderen Worten: Sei $L \subseteq \Sigma^*$ eine Sprache, dann sind folgende Aussagen gleichwertig:

1. $L = T(M_1)$ für einen deterministischen endlichen Automaten M_1
2. $L = T(M_2)$ für einen nichtdeterministischen endlichen Automaten M_2

5.1.2 Beweis des Buches

Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein NFA, welcher die Sprache L akzeptiert. Ein DFA $M' = (Q', \Sigma, \delta', q'_0, F')$ sei dann wie folgt definiert: Die Zustände von M' sind alle Teilmengen der Menge der Zustände von M. Das heißt: $Q' = 2^Q$. M' wird alle Zustände behandeln, welche auch M behandeln kann. F' ist die Menge, der Zustände von Q', welche einen Endzustand von M enthalten. Ein Element von Q' wird dargestellt, als $[q_1, q_2, \ldots, q_i]$, wobei $q_1, q_2, \ldots, q_i$ aus Q stammen. Zu beachten ist, dass $[q_1, q_2, \ldots, q_i]$ ein einziger Zustand des DFA darstellt. Weiterhin ist $q_0 = [q_0]$.

Wir definieren nun:

$$\delta' = ([q_1, q_2, \ldots, q_i], a) = [p_1, p_2, \ldots, p_j] \tag{5.1}$$

dann und nur dann, wenn:

$$\delta = (\{q_1, q_2, \ldots, q_i\}, a) = \{p_1, p_2, \ldots, p_j\} \tag{5.2}$$

In anderen Worten, δ' angewendet auf $[q_1, q_2, \ldots, q_i]$ von Q' wird berechnet, durch anwenden von δ auf jeden Zustand von Q, welcher durch $[q_1, q_2, \ldots, q_i]$ repräsentiert wird. Durch das Anwenden von δ auf alle $q_1, q_2, \ldots, q_i$ und vereinigen der Ergebnisse, erhalten wir eine neue Menge von Zuständen, $p_1, p_2, \ldots, p_j$. Diese neue Menge hat einen Vertreter $[p_1, p_2, \ldots, p_j]$ in Q' und dieser ist der Wert von $\delta' = ([q_1, q_2, \ldots, q_i], a)$.

Es lässt sich nun durch Induktion über die Länge eines Eingabestrings x zeigen, dass

$$\delta'(q'_0, x) = [q_1, q_2, \ldots, q_i] \tag{5.3}$$

[1] Vgl.: INTRODUCTION TO AUTOMATA THEORY, LANGUAGES AND COMPUTATION by John E. HOPECROFT and Jeffrey D. ULLMAN pages 22f

[2] Ein DFA ist ein Spezialfall des NFA, bei dem es für jeden Zustand eine einzigartige Übergangsfunktion für jedes Symbol gibt.

dann und nur dann gilt, wenn:

$$\delta(q_0,x)=\{q_1,q_2,\dots,q_i\} \tag{5.4}$$

Induktionsverankerung:

Man sieht leicht, dass für $|x|=0$ die Behauptung gilt, da $q'_0=[q_0]$ und x gleich ε sein muss.

Induktionsschritt:

Angenommen, die Behauptung ist wahr für Eingaben der Länge $\leq m$. Weiterhin sei xa eine Kette der Länge $m+1$ mit a in Σ. Dann gilt:

$$\delta'=(q'_0,xa)=\delta'(\delta'(q'_0,x),a) \tag{5.5}$$

Aufgrund der Behauptung ist:

$$\delta'(q'_0,x)=[p_1,p_2,\dots,p_j] \tag{5.6}$$

dann und nur dann wahr, wenn:

$$\delta(q0,x)=\{p_1,p_2,\dots,p_j\} \tag{5.7}$$

gilt. Doch mit der Definition von δ' gilt:

$$\delta'([p_1,p_2,\dots,p_j],a)=[r_1,r_2,\dots,r_k] \tag{5.8}$$

dann und nur dann, wenn:

$$\delta(\{p_1,p_2,\dots,p_j\},a)=\{r_1,r_2,\dots,r_k\} \tag{5.9}$$

gilt. Somit ist:

$$\delta'(q'_0,xa)=[r_1,r_2,\dots,r_k] \tag{5.10}$$

dann und nur dann, wenn:

$$\delta(q_0,xa)=\{r_1,r_2,\dots,r_k\} \tag{5.11}$$

und das unterstützt die Behauptung. Um den Beweis zu vollenden, muss nur noch hinzugefügt werden, dass $\delta'(q'_0,x)$ genau dann zu F' führt, wenn $\delta(q_0,x)$ einen Zustand von Q enthält, der in F liegt. Somit ist $L(M)=L(M')$

Soweit der Beweis des Buches, in der Vorlesung haben wir den Beweis nur geringfügig anders gehört.

5.1.3 Beweis der Vorlesung

Zur Wiederholung hier noch einmal den zu beweisenden Satz: Sei $L\subseteq\Sigma^*$ eine Sprache, dann sind folgende Aussagen gleichwertig:

1. $L=T(M_1)$ für einen deterministischen endlichen Automaten M_1
2. $L=T(M_2)$ für einen nichtdeterministischen endlichen Automaten M_2

„Hinrichtung“:

Zuerst zeigen wir hier, mit einfachen Definitionen, aus 1 folgt 2. Gegeben sei ein DFA $M_1=(Q_1,\Sigma,\delta_1,q_0^{(1)},F_1)$ mit $\delta_1:Q_1\times\Sigma\to Q_1$. Dieser Automat akzeptiert die Sprache $L=T(M_1)$ nach Definition. Nun definieren wir uns dazu einen NFA $M_2=_{Df}(Q_1,\Sigma,\delta_2,q_0^{(1)},F_1)$ mit einer Übergangsfunktion $\delta_2:Q_1\times\Sigma\to P(Q_1)$.

$\overset{2}{\sim}$:

	a	b
1.	$[2]_{\overset{1}{\sim}}$	$[3]_{\overset{1}{\sim}}$
2.	$[6]_{\overset{1}{\sim}}$	$[1,4,5]_{\overset{1}{\sim}}$
3.	$[1,4,5]_{\overset{1}{\sim}}$	$[6]_{\overset{1}{\sim}}$
4.	$[2]_{\overset{1}{\sim}}$	$[3]_{\overset{1}{\sim}}$
5.	$[2]_{\overset{1}{\sim}}$	$[3]_{\overset{1}{\sim}}$
6.	$[6]_{\overset{1}{\sim}}$	$[6]_{\overset{1}{\sim}}$

Folglich sind die Klassen von $\overset{2}{\sim}$:

$[1,4,5],[2],[3],[6]$

An dieser Stelle sind wir dann auch schon fertig, da wir sehen, dass die Klassen $\overset{1}{\sim}$ und $\overset{2}{\sim}$ schon äquivalent sind, somit ist die ~ Relation definiert durch:

$$\sim \equiv \overset{1}{\sim} \tag{8.12}$$

Es bleibt nur noch den endgültigen Automaten $M'=\left(Q',\Sigma,\delta',[q_0]_\sim,F'\right)$ aufzustellen. Dieser sieht dann, so aus, wie in der folgenden Abbildung gezeigt.

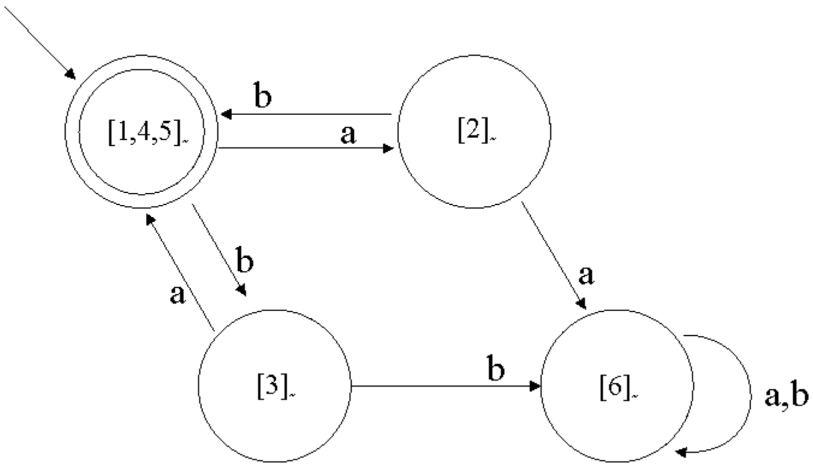

Abbildung 15 der minimierte Automat M'

Diese Übergangsfunktion ist wie folgt definiert:

$$\delta_2(q,\sigma) =_{Df} \{\delta_1(q,\sigma)\} \tag{5.12}$$

Aus dieser Definition folgt nun schon, dass M_2 ein nichtdeterministischer Automat ist. Der einzige Unterschied zum DFA besteht darin, dass die Übergangsfunktion so definiert ist, dass sie eine Menge von Folgezuständen liefert. In unserem Fall sind dies jedoch alles einelementige Mengen.

Durch Induktion über alle $n \geq 0$ kann man dies zeigen:

$$\begin{aligned}
&\delta_2(q_0,x_1\cdots x_n)=\{p\} \Leftrightarrow \delta_1(q_0,x_1\cdots x_n)=p\\
&\Rightarrow\\
&x\in T(M_2) \Leftrightarrow \quad \delta_2(q_0,x)\cap F_1 \neq 0\\
&\qquad \Leftrightarrow \exists p: \quad \delta_2(q_0,x)=\{p\}\subseteq F_1\\
&\qquad \Leftrightarrow \exists p: \quad \delta_1(q_0,x)=p\in F_1\\
&\qquad \Leftrightarrow x\in T(M_1)\\
&\Rightarrow T(M_2)=T(M_1)=L
\end{aligned} \tag{5.13}$$

In diesem Beweis ist ausgenutzt worden, dass ein Wort nur dann von einem NFA erkannt wird, wenn die Menge von $\{p\}$ eine Teilmenge von F_1 ist. Dies bedeutet für einen gleichwertigen DFA, dass dieses $p \in F_1$ sein muss, um akzeptiert zu werden. Dies war bei uns nach Definition der Fall.

Rückrichtung:

Nun bleibt noch zu zeigen, dass der umgekehrte Fall gilt. Sprich, wir haben einen NFA gegeben und wollen daraus einen DFA konstruieren. (Aus 2 folgt 1): Hier sei ein NFA $M_2=\left(Q_2,\Sigma,\delta_2,q_0^{(2)},F_2\right)$ mit einer Übergangsfunktion $\delta_2 : Q_2 x \times \Sigma \rightarrow P(Q_2)$ gegeben. Auch dieser Automat soll nach Definition die Sprache $L=T(M_2)$ akzeptieren.

Wir definieren nun einen DFA M_1 wie folgt:

$$\begin{aligned}
&M_1 =_{Df} \left(P(Q_2),\Sigma,\delta_1,\{q_0^{(2)}\},F_1\right)\\
&F_1 =_{Df} \left\{A \middle| A\subseteq Q_2 \wedge (A\cap F_2 \neq \varnothing)\right\} \Rightarrow F_1 \subseteq P(Q_2)\\
&\delta_1 : P(Q_2)\times\Sigma \rightarrow P(Q_2)\\
&\delta_1(A,\sigma)=\bigcup_{p\in A}\delta_2(p,\sigma),\sigma\in\Sigma,A\in P(Q_2)
\end{aligned} \tag{5.14}$$

Der so definierte Automat ist ein deterministischer endlicher Automat. Nun bleibt zu zeigen, dass jeder Zustand, den der DFA erreicht, der Menge von Zuständen entspricht, die der NFA erreicht hätte:

$$\delta_1\left(\{q_0^{(2)}\},x\right)=\delta_2\left(q_0^{(2)},x\right) \quad \forall x\in\Sigma^* \tag{5.15}$$

Dies wird wieder durch Induktion über die Länge der Eingaben bewiesen.

Induktionsbehauptung:

$$\delta_1\left(\{q_0^{(2)}\},x_1\cdots x_n\right)=\delta_2\left(q_0^{(2)},x_1\cdots x_n\right) \tag{5.16}$$

Induktionsverankerung (n=0):

$\delta_1\left(\{q_0^{(2)}\},\varepsilon\right)=\delta_2(q_0^{(2)},\varepsilon)$ Man sieht leicht, dass für leere Eingaben der Definition der δ -Funktion nach das selbe Ergebnis (das Bleiben im Startzustand $q_0^{(2)}$) erreicht wird.

Induktionsschritt ($n \rightarrow n+1$)**:**

$$\begin{aligned}
\delta_1\left(\left\{q_0^{(2)}\right\}, x_1 \cdots x_n x_{n+1}\right) &= \delta_1\left(\delta_1\left(\left\{q_0^{(2)}\right\}, x_1 \cdots x_n\right), x_{n+1}\right) \\
&= \delta_1\left(\delta_2\left(q_0^{(2)}, x_1 \cdots x_n\right), x_{n+1}\right) \\
&= \bigcup_{p \in \delta_2\left(q_0^{(2)}, x_1 \cdots x_n\right)} \delta_2\left(p, x_{n+1}\right) \qquad (5.17) \\
&= \delta_2\left(q_0^{(2)}, x_1 \cdots x_n x_{n+1}\right)
\end{aligned}$$

Bei diesem Beweis haben wir folgendes Wissen angewendet. Da es ein DFA ist, nimmt man den Zustand, der mit den ersten n Symbolen (Von diesen wissen wir aufgrund der Behauptung, dass für diese die Übergangsfunktion δ_2 gleichwertig mit δ_1 ist.) erreicht hat und vereinigt diesen mit dem Zustand, der durch das $n+1$. Symbol erreicht wird. Und die Vereinigung der Teilmengen der ersten $\{1 \ldots n\}$ Symbole zusammen mit dem Symbol x_{n+1} ist gleichbedeutend mit der Übergangsfunktion δ_2 angewendet auf die Symbole $\{1 \ldots n+1\}$. Nun ist noch zu zeigen, dass auch die selben Wörter akzeptiert werden.

$$\begin{aligned}
\forall x \in \Sigma^* \quad x \in T(M_2) &\Leftrightarrow \delta_2\left(q_0^{(2)}, x\right) \cap F_2 \neq \varnothing \\
&\Leftrightarrow \delta_2\left(q_0^{(2)}, x\right) \in F_1 \\
&\Leftrightarrow \delta_1\left(\left\{q_0^{(2)}\right\}, x\right) \in F_1 \qquad (5.18) \\
&\Leftrightarrow x \in T(M_1) \\
\Rightarrow L = T(M_2) &= T(M_1)
\end{aligned}$$

Auch dieser Beweis ist wieder mittels der Definitionen von δ_1 geführt worden.

Somit steht fest, dass es für alle nichtdeterministischen endlichen Automaten $M = (Q, \Sigma, \delta, q_0, F)$ ein deterministischer endlicher Automat $M' = (Q', \Sigma, \delta, q'_0, F')$ existiert für den die folgenden zwei Aussagen gelten:

1. $T(M)) = T(M')$
2. $|Q'| \leq 2^{|Q|}$

Die erste Aussage haben wir eben bewiesen. Die zweite Aussage folgt aus der Betrachtung der möglichen Potenzmenge. Hier der kurze Induktionsbeweis.

Induktionsbehauptung:

Für alle $n \geq 0$ mit $|S| = n$ gilt $|P(S)| = |2^S| = 2^{|S|}$.

Induktionsverankerung:

Für $n = 0$ gilt $P(S) = \{\varnothing\}$, $|P(S)| = 1 = 2^0$.

Induktionsschritt ($n \rightarrow n+1$)**:**

Bei der Betrachtung der Symbole $\{s_1, \ldots, s_n, s_{n+1}\}$ steht folgendes fest. Die Anzahl der Teilmengen ohne s_{n+1} beträgt 2^n. Die Anzahl der Teilmengen mit s_{n+1} beträgt 2^n. Zusammen gibt dies: $2^n + 2^n = 2 \cdot 2^n = 2^{n+1}$. Damit ist die Behauptung gezeigt.

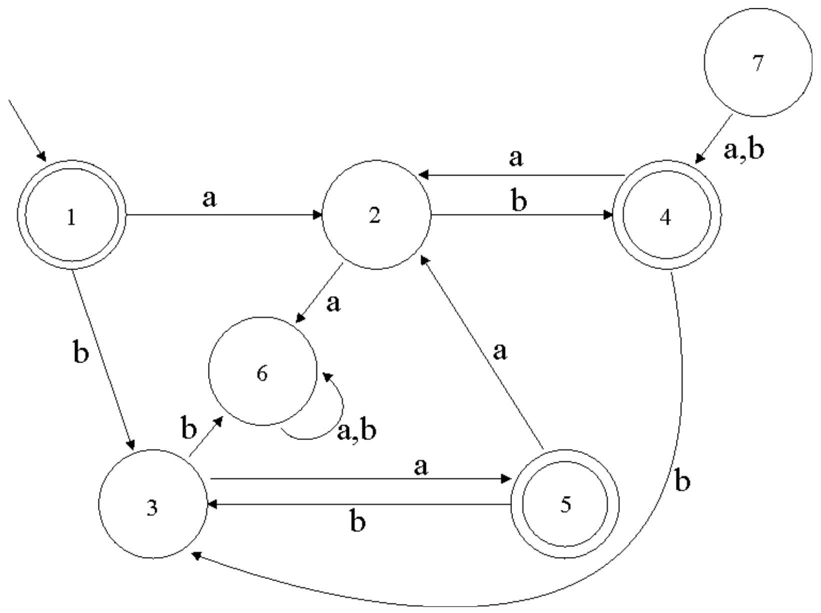

Abbildung 14: Beispiel eines nicht minimalen Automaten

Durch das Anwenden des oben beschriebenen Algorithmus gelangt man schnell zu folgender Tabelle:

$\overset{0}{\sim}$: $[2,3,6]$ **Fehler! Es ist nicht möglich, durch die Bearbeitung von Feldfunktionen Objekte zu erstellen.**

$\overset{1}{\sim}$:

	a	b
1.	$[2,3,6]_{\overset{0}{\sim}}$	$[2,3,6]_{\overset{0}{\sim}}$
2.	$[2,3,6]_{\overset{0}{\sim}}$	$[1,4,5]_{\overset{0}{\sim}}$
3.	$[1,4,5]_{\overset{0}{\sim}}$	$[2,3,6]_{\overset{0}{\sim}}$
4.	$[2,3,6]_{\overset{0}{\sim}}$	$[2,3,6]_{\overset{0}{\sim}}$
5.	$[2,3,6]_{\overset{0}{\sim}}$	$[2,3,6]_{\overset{0}{\sim}}$
6.	$[2,3,6]_{\overset{0}{\sim}}$	$[2,3,6]_{\overset{0}{\sim}}$

Folglich sind die Klassen von $\overset{1}{\sim}$:

$[1,4,5],[2],[3],[6]$

1. Man beseitige alle nicht-erreichbaren Zustände, d.h. alle Zustände q, für die gilt $q \notin \left\{\delta\left(q_0, x\right) \middle| x \in \Sigma^*\right\}$. (Es stellt sich natürlich die Frage, in wie weit dieses Problem algorithmisch lösbar ist. Ein möglicher Algorithmus könnte alle durch Wörter der Länge n erreichbaren Zustände betrachten. Damit kommt man durch alle erreichbaren Zustände.)
2. $\overset{0}{\sim}$ hat maximal zwei Äquivalenzklassen. Zum einen die akzeptierenden Zustände und dann noch die nicht akzeptierenden Zustände: $\left\{q \middle| q \in Q - F\right\}_{\overset{0}{\sim}}$, $\left\{q \middle| q \in F\right\}_{\overset{0}{\sim}}$
3. Angenommen, man hat die Äquivalenzklassen der Relation $\overset{k}{\sim}$ für $\left(k \geq 0\right)$, dann bestimmt man die Klassen der Relation $\overset{k+1}{\sim}$ wie folgt:
Zunächst wird nun die folgende Tabelle aufgestellt:

	a_1 a_2 …	a_j	… $a_{\lvert\Sigma\rvert}$
q_0 q_1 ⋮			
q_i		$\left[\delta\left(q_i, a_j\right)\right]_{\overset{k}{\sim}}$	
⋮ $q_{\lvert Q\rvert-1}$			

Alle Felder $\left(q_i, a_j\right)$ erhalten also als Eintrag $\left[\delta\left(q_i, a_j\right)\right]_{\overset{k}{\sim}}$. Nun formt man die Klassen von $\overset{k+1}{\sim}$, indem man alle Zustände, die bereits bei $\overset{k}{\sim}$ in einer Klasse waren *und* die in obiger Liste gleiche „Zeilen" haben in jeweils eine Klasse der Relation $\overset{k+1}{\sim}$ schreibt.
4. Wenn nun $\overset{k}{\sim}$ äquivalent zu $\overset{k+1}{\sim}$ ist (was, wie schon gezeigt, spätestens bei $k = n-1$ eintreten muss), dann bricht das Verfahren ab. Wir bilden dann die ~ Relation durch Setzen, von $\sim \equiv \overset{k}{\sim}$.
5. Der minimale Automat $M' = \left(Q', \Sigma, \delta', \left[q_0\right]_{\sim}, F'\right)$ wird letztlich wie oben schon beschrieben konstruiert:

$$\begin{aligned} Q' &= \left\{\left[q\right]_{\sim} \middle| q \in Q\right\} \\ F' &= \left\{\left[q\right]_{\sim} \middle| q \in F\right\} \\ \delta'\left(\left[q\right]_{\sim}, a\right) &= \left[\delta\left(q, a\right)\right]_{\sim} \end{aligned} \tag{8.11}$$

8.2.2 Beispiel

Wir betrachten einen Automaten $M = \left(Q, \Sigma, \delta, q_0, F\right)$, welcher die Sprache $T\left(M\right) = \left\{ab, ba\right\}^*$ akzeptieren soll. Ein Automat, der dies löst ist in folgender Grafik aufgelistet. Dabei ist zu beachten, dass der Zustand 7 nicht erreicht werden kann.

5.1.4 Beispiel 1 (Konstruktion eines DFA aus einem NFA)

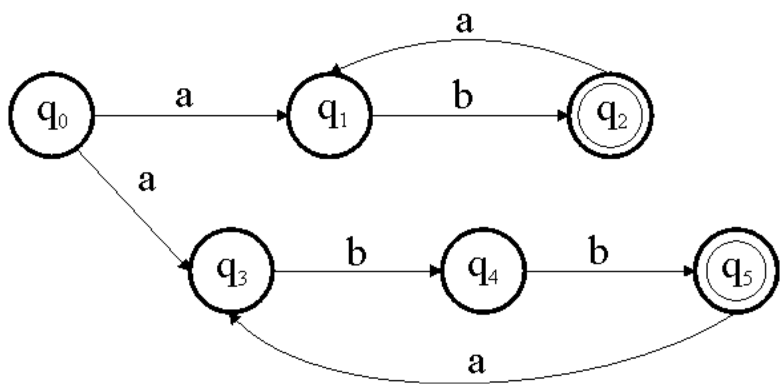

Abbildung 7 – ein nichtdeterministischer, endlicher Automat M

Dieser Automat akzeptiert offensichtlich die Sprache $T\left(M\right) = \left\{\left(ab\right)^n \middle| n \geq 1\right\} \cup \left\{\left(abb\right)^n \middle| n \geq 1\right\}$.

Ein deterministischer endlicher Automat könnte so aussehen:

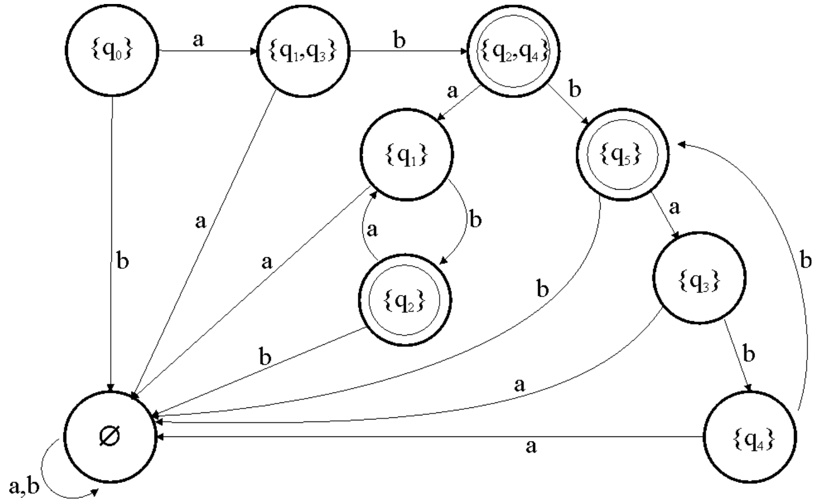

Abbildung 8 - Der DFA, der zu' M äquivalent ist

In diesem Beispiel hat der DFA M' 9 Zustände, während der NFA M 6 Zustände hatte. Und das liegt noch einiges unter den maximal 2^6 möglichen Zuständen.

5.1.5 Beispiel 2 (Konstruktion eines DFA aus einem NFA)

Hier nun ein etwas komplexeres Beispiel. Es soll ein NFA umgewandelt werden, der die Sprache $L_4 = \{0,1\}^* \cdot \{1\} \cdot \{0,1\}^3$ akzeptiert. Ein solcher NFA kann so aussehen:

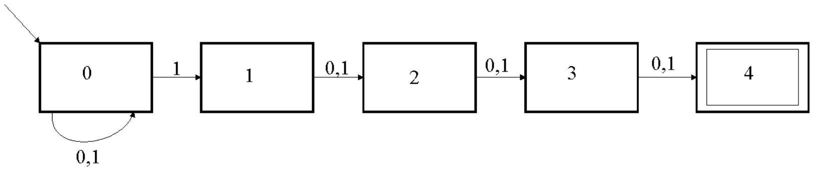

Abbildung 9: NFA zur Sprache L_4

Einen DFA für die Sprache bildet dann:

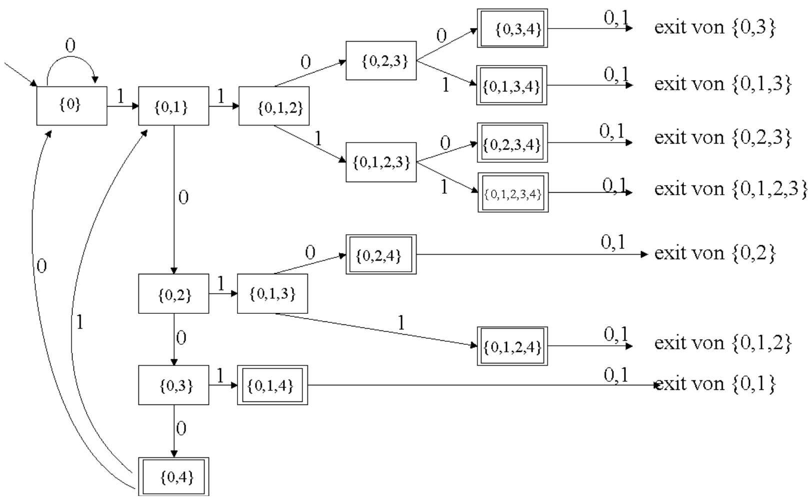

Abbildung 10: der aus dem NFA erzeugte DFA für die Sprache L_4

Bei diesem Beispiel benötigt der NFA 5 Zustände, der DFA jedoch 16 Zustände. Es stellt sich nun die Frage, ob diese Konstruktion auch wirklich minimal ist.

5.2 Satz über die Minimierung von Zuständen

Für alle $n \geq 1$ gilt: Die Sprache $L_n = \{0,1\}^* \cdot \{1\} \cdot \{0,1\}^{n-1}$ kann von einem nichtdeterministischen Automaten mit $(n+1)$ Zuständen akzeptiert werden. Dann folgt für jeden deterministischen Automaten $M = (Q, \Sigma, \delta, q_0, F)$ mit der Sprache $L_n = T(M) \Rightarrow |Q| \geq 2^n$.

Beweis:

Sei $L_n = \{0,1\}^* \cdot \{1\} \cdot \{0,1\}^{n-1}$ gegeben. Weiterhin auch ein NFA M der diese Sprache akzeptiert, also $L_n = T(M)$. Ein Graph dieses Automaten kann wie folgt aussehen:

δ' ist mit :

$$\forall x \in \Sigma^* : \delta'([q], x) = [\delta(q, x)] \tag{8.4}$$

wohldefiniert, aus folgendem Grund: Wenn q und q' in der selben Äquivalenzklasse liegen und die Relation definiert sind, dann ist auch $\delta'([q], a)$ mit $\delta'([q'], a)$ in Relation.

Mit der Definition von F' gilt dann:

$$\begin{aligned} x \in T(M) \quad &\Leftrightarrow \quad \delta(q_0, x) \in F \Leftrightarrow [\delta(q_0, x)] \in F' \\ &\Leftrightarrow \quad \delta'([q_0], x) \in F' \\ &\Leftrightarrow \quad x \in T(M') \end{aligned} \tag{8.5}$$

Somit akzeptieren also beide Automaten die selbe Sprache. Die Frage ist nun, ob dieser Automat auch minimal ist.

Zuerst definieren wir die ρ Relation als:

$$x \;\rho\; y \Leftrightarrow_{Df} \delta'([q_0], x) = \delta'([q_0], y) \tag{8.6}$$

Weiterhin wissen wir über diesen Automaten, dass der $Index(\rho) = |Q'|$, da jeder Zustand erreichbar ist. Somit wissen wir, dass die ρ Relation die Nerode-Automat Relation R_L verfeinert:

$$x \;\rho\; y \Rightarrow x \; R_L \; y \tag{8.7}$$

Gilt nun auch, die Umkehrung? Folgt aus $x \;\rho\; y \Leftarrow x \; R_L \; y$? Der Beweis erfolgt durch Kontraposition.

$$x \;\not\rho\; y \Rightarrow x \; \not R_L \; y \tag{8.8}$$

Die Klassen können nicht gleich sein, da es die Elemente nicht in der ~ Relation gibt. Es gibt daher ein Element z, so dass:

$$\begin{aligned} & x \;\not\rho\; y \Rightarrow [\delta(q_0, x)] = \delta'([q_0], x) \neq \delta'([q_0], y) = [\delta(q_0, y)] \\ & \Rightarrow \exists \; z : (\delta(q_0, xz) \in F \wedge \delta(q_0, yz) \notin F) \vee (\delta(q_0, xz) \notin F \wedge \delta(q_0, yz) \in F) \\ & \Rightarrow x \; \not R_L \; y \end{aligned} \tag{8.9}$$

Die beiden Relationen sind also gleichwertig: $x \;\rho\; y \Leftrightarrow x \; R_L \; y$. Und die Minimalzahl der Zustände ist: $|Q'| = index(\rho) = index(R_L)$ Und damit ist gezeigt, dass der Automat M' minimal ist.

8.2 „Praktische“ Minimierung von deterministischen endlichen Automaten

Nachdem wir wissen, dass die Bestimmung der Äquivalenzklassen $q \sim q'$ definiert ist für: $\forall x \in \Sigma^* : \delta(q, x) \in F \Leftrightarrow \delta(q', x) \in F$. Eine Konstruktion, die den (bis auf Isomorphie) eindeutig bestimmten minimalen endlichen Automaten ($M' = (Q', \Sigma, \delta, q_0, F)$ mit $Q' = \{[q]_\sim \,|\, q \in Q\}$) liefert benötigt die folgende Eigenschaft:

$$q \overset{k}{\sim} q' \Leftrightarrow q \overset{k-1}{\sim} q' \wedge \left(\forall a \in \Sigma : \delta(q, a) \overset{k-1}{\sim} \delta(q', a) \right) \tag{8.10}$$

8.2.1 Der „Algorithmus“

Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein endlicher Automat. Dann gelangt man durch die folgenden Schritte zu dem minimalen Automaten:

8 Vorlesung vom 2. Mai 2000

8.1 Minimierung mittels Nerode Automaten

8.1.1 Satz über die Gleichwertigkeit von Relationen

Sei $M=(Q,\Sigma,\delta,q_0,F)$ ein endlicher deterministischer Automat mit der Mächtigkeit $|Q|=n$. Dann gilt: $q \sim q' \Leftrightarrow q \overset{n-1}{\sim} q'$.

Der Beweis der „Hinrichtung". ist trivial und folgt auch schon aus dem Satz der letzten Vorlesung. Die „Rückrichtung" ist da schon komplizierter. Der Beweis erfolgt indirekt. Angenommen die beiden Relationen währen nicht identisch:

$$q \overset{n-1}{\sim} q' \wedge q \nsim q' \Rightarrow \sim \not\equiv \overset{n-1}{\sim} \tag{8.1}$$

Grundsätzlich gilt, dass ~ eine Verfeinerung von $\overset{n}{\sim}$ ist. Damit muss $\overset{n}{\sim}$ eine echte Verfeinerung von $\overset{n-1}{\sim}$ sein, denn sonst wäre ja $\sim \equiv \overset{n-1}{\sim}$, was ein Widerspruch zur Annahme wäre.

Durch eine Betrachtung der Anzahl der Klassen der Zustände ($\overset{0}{\sim} \geq \overset{1}{\sim} \geq \overset{2}{\sim} \geq \ldots \geq \overset{n-1}{\sim} \geq \overset{n}{\sim}$) erkennt man schnell, dass es mindestens eine Klasse von Zuständen geben muss, und zwar die der akzeptierenden Zustände. Da weiterhin gilt, dass $\overset{n}{\sim}$ eine echte Verfeinerung von $\overset{n-1}{\sim}$ ist, sind alle anderen Relationen echte Verfeinerungen (also gilt $\overset{0}{\sim} > \overset{1}{\sim} > \overset{2}{\sim} > \ldots > \overset{n-1}{\sim} > \overset{n}{\sim}$). Die Anzahl der Klassen wächst durch jeden Index um eins. Führt man diesen Gedankengang fort und beachtet, dass es alles echte Verfeinerungen sind, so gelangt man für den $Index\left(\overset{n}{\sim}\right)$ zu dem Schluss, dass dieser mindestens $n+1$ Klassen enthalten muss.

$$\begin{aligned} &1 \leq Index\left(\overset{0}{\sim}\right) < Index\left(\overset{1}{\sim}\right) < Index\left(\overset{2}{\sim}\right) < \ldots < Index\left(\overset{n-1}{\sim}\right) < Index\left(\overset{n}{\sim}\right) \\ &\Rightarrow Index\left(\overset{n}{\sim}\right) \geq n+1 \end{aligned} \tag{8.2}$$

Dies ist allerdings ein Widerspruch, da es höchstens n (mit $n=|Q|$) Äquivalenzklassen auftreten können.

8.1.2 Konstruktion eines minimalen Automaten

Sei nun ~ weiterhin wie bisher definiert. Dann gibt es zu dem endlichen Automaten $M=(Q,\Sigma,\delta,q_0,F)$ einen äquivalenten minimalen Automaten $M'=(Q',\Sigma,\delta',[q_0]_\sim,F')$. Dabei ist:

$$\begin{aligned} Q' &=_{Df} \{[q]_\sim \mid q \in Q\} \\ F' &=_{Df} \{[f]_\sim \mid f \in F\} \\ \delta'([q]_\sim,a) &=_{Df} [\delta(q,a)]_\sim \end{aligned} \tag{8.3}$$

Dieser Automat wird wie folgt konstruiert, als Zustände nehme man die Äquivalenzklassen der Zustände. Als akzeptierte Zustände gelten die Äquivalenzklassen der Tilderelation von F.

Damit ist $M'=(Q',\Sigma,\delta',[q_0]_\sim,F')$ ein minimaler Automat mit $T(M)=T(M')$.

Aber ist dieser Automat auch wirklich minimal, oder nur ein reduzierter Automat? Gezeigt wird dies durch den folgenden Beweis:

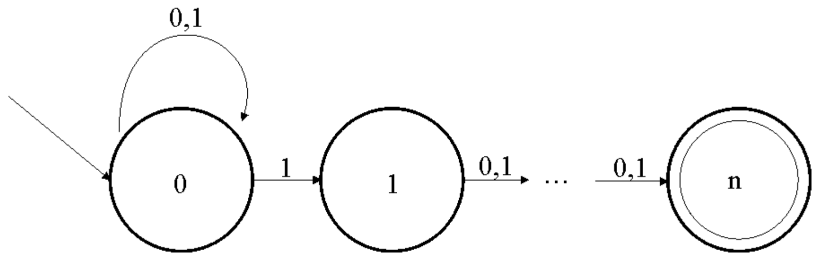

Abbildung 11: NFA für die Sprache Ln

Sei nun weiterhin $M=(Q,\Sigma,\delta,q_0,F)$ ein DFA, der die selbe Sprache akzeptiert, also $L_n=T(M)$.

Behauptung: Jedes Wort des möglichen Eingabealphabetes (von 2^n möglichen Wörtern) muss von dem Automaten unterschieden werden können.

$$\forall x,y \in \{0,1\}^n : x \neq y \Rightarrow \delta(q_0,x) \neq \delta(q_0,y) \tag{5.19}$$

Angenommen zwei gleichlange Wörter aus $\{0,1\}^n$, die unterschiedlich sind (d.h. sie unterscheiden sich nach einigen gleichen Zeichen an einer Stelle, danach folgt ein Wort z) hätten dieselben Nachfolgezustände.

$$\begin{aligned} &x,y \in \{0,1\}^n \wedge x \neq y \\ &\Rightarrow \quad x = x_1\ 1\ z \\ &\qquad\quad y = y_1\ 0\ z \\ &\qquad\quad |x_1| = |y_1| \\ &angenommen: \quad \delta(q_0,x) = \delta(q_0,y) \end{aligned} \tag{5.20}$$

Dann werden entweder beide Wörter akzeptiert, oder aber keines. In beiden Fällen macht der Automat einen Fehler.

$$\begin{aligned} \Rightarrow\quad & \delta\left(q_0,x_1\ 1\ z\ 0^{n-|z|-1}\right) &=\quad& \delta\left(q_0,x\ 0^{n-1-|z|}\right) \\ =\quad & \delta\left(\delta(q_0,x),0^{n-1-|z|}\right) &=\quad& \delta\left(\delta(q_0,y),0^{n-1-|z|}\right) \\ =\quad & \delta\left(q_0,y\ 0^{n-1-|z|}\right) &=\quad& \delta\left(q_0,y_1\ z\ 0^{n-1-|z|}\right) \\ \Rightarrow\quad & x_1\ 1\ \underbrace{z\ 0^{n-1-|z|}}_{n-1} \in T(M) &\Leftrightarrow\quad& y_1\ 0\ \underbrace{z\ 0^{n-1|z|}}_{n-1} \in T(M) \end{aligned} \tag{5.21}$$

Der Widerspruch liegt darin, dass $x_1\ 1\ z\ 0^{n-1-|z|}$ Element der Sprache L_n ist, $y_1\ 0\ z\ 0^{n-1-|z|}$ jedoch nicht, trotzdem wird für beide Wörter die selbe Entscheidung getroffen, folglich benötigt man für unterschiedliche Eingabewörter unterschiedliche Zustände. Die maximale Anzahl der Zustände beträgt, wie schon vorher gezeigt 2^n.

$$\begin{aligned} &x_1\ 1\ z\ 0^{n-1-|z|} \in L_n \\ &y_1\ 0\ z\ 0^{n-1-|z|} \notin L_n \\ &\Rightarrow \forall x,y \in \{0,1\}^n : x \neq y \Rightarrow \delta(q_0,x) \neq \delta(q_0,y) \\ &\left|\{0,1\}^n\right| = 2^n \Rightarrow |Q| \geq 2^n \end{aligned} \tag{5.22}$$

6 Vorlesung vom 25. April 2000

6.1 Nachtrag zur vorherigen Vorlesung

Gegeben ist die folgende (reguläre) Sprache:

$$T(M) = \left\{(ab)^n \mid n \geq 1\right\} \cup \left\{(abb)^n \mid n \geq 1\right\} \tag{6.1}$$

Korollar:

Es gibt für jede dieser Sprachen einen nichtdeterministischen, endlichen Automaten, welcher mit n+1 Zuständen auskommt:

$$\begin{aligned}&\forall n \geq 1: \\ &\exists \text{ NFA } M=(Q,\Sigma,\delta,q_0,F) \quad \text{mit} \quad L_n = T(M) \wedge |Q| = n+1\end{aligned} \tag{6.2}$$

Es gibt für jede dieser Sprachen einen deterministischen, endlichen Automaten, welcher mit weniger als 2^{n+1} Zuständen auskommt:

$$\begin{aligned}&\forall n \geq 1: \\ &\exists \text{ DFA } M=(Q,\Sigma,\delta,q_0,F) \quad \text{mit} \quad L_n = T(M) \wedge |Q| < 2^{n+1}\end{aligned} \tag{6.3}$$

Jeder deterministische, endliche Automaten, der eine dieser Sprachen beschreibt, besitzt mindestens 2^n Zustände.

$$\begin{aligned}&\forall n \geq 1: \\ &\forall \text{ DFA } M=(Q,\Sigma,\delta,q_0,F): \quad L_n = T(M) \Rightarrow |Q| \geq 2^n\end{aligned} \tag{6.4}$$

Satz:

Es gibt für jede dieser Sprachen einen nichtdeterministischen, endlichen Automaten, welcher mit n Zuständen auskommt:

$$\begin{aligned}&\forall n \geq 1: \\ &\exists \text{ NFA } M=(Q,\Sigma,\delta,q_0,F) \quad \text{mit} \quad L_n = T(M) \wedge |Q| = n\end{aligned} \tag{6.5}$$

Jeder deterministische, endliche Automaten, der eine dieser Sprachen beschreibt, besitzt mindestens 2^n Zustände.

$$\begin{aligned}&\forall n \geq 1: \\ &\forall \text{ DFA } M=(Q,\Sigma,\delta,q_0,F): \quad L_n = T(M) \Rightarrow |Q| \geq 2^n\end{aligned} \tag{6.6}$$

Beweis:

Der Beweis gestaltet sich umfangreich, deswegen wird an dieser Stelle darauf verzichtet.

6.2 Mathematischer Hintergrund

6.2.1 Erweiterung des Begriffes „Äquivalenzrelation"

Man spricht von einer „*Äquivalenzrelation*", wenn eine Relation reflexiv, transitiv und symmetrisch ist, und von einer Ordnung, wenn eine Relation reflexiv, transitiv und antisymmetrisch ist. Zum Beispiel ist die normale Vergleichsoperation auf der Menge der reellen Zahlen („=") eine Äquivalenzrelation, und die kleiner-gleich-Operation („≤") eine Ordnung.

Unter dem „*Index*" einer Äquivalenzrelation versteht man die Maximalzahl paarweise verschiedener Äquivalenzklassen.

$$\overset{k}{\sim} \equiv \overset{k+1}{\sim} \Rightarrow \overset{k}{\sim} \equiv \overset{k+s}{\sim} \quad \forall s \geq 0 \tag{7.25}$$

Beweis:

Die Behauptung kann man durch Induktion beweisen. Für s = 0 ist die Aussage klar, da es sich um dieselbe Relation k handelt. Für alle weiteren nehmen wir an, dass

$$\overset{k}{\sim} \equiv \overset{k+s}{\sim} \tag{7.26}$$

bereits gelte. Dann kann man auf k+s+1 gehen

$$\begin{aligned}&q \overset{k+s+1}{\sim} q' \Leftrightarrow q \overset{k+s}{\sim} q' \wedge \left(\forall a \in \Sigma : \delta(q,a) \overset{k+s}{\sim} \delta(q',a)\right) \\ &q \overset{k+1}{\sim} q' \Leftrightarrow q \overset{k}{\sim} q' \wedge \left(\forall a \in \Sigma : \delta(q,a) \overset{k}{\sim} \delta(q',a)\right) \\ &q \overset{k+2}{\sim} q' \Leftrightarrow q \overset{k+1}{\sim} q' \wedge \left(\forall a \in \Sigma : \delta(q,a) \overset{k+1}{\sim} \delta(q',a)\right) \\ &\quad \ldots \\ &\quad \Leftrightarrow q \overset{k}{\sim} q'\end{aligned} \tag{7.27}$$

Korollar:

Der Index der k-1 Relation ist kleiner als der der k Relation

$$Index(\overset{k-1}{\sim}) \leq Index(\overset{k}{\sim}) \tag{7.28}$$

Damit haben wir als Ausgangspunkt für weitere Betrachtungen folgende Aussagen zusammengestellt

$$\overset{k}{\sim} \leq \overset{k-1}{\sim} \tag{7.29}$$

$$Index(\overset{k}{\sim}) \geq Index(\overset{k-1}{\sim}) \tag{7.30}$$

$$\overset{k}{\sim} \equiv \overset{k+1}{\sim} \Rightarrow \forall s \geq 0 : \overset{k}{\sim} \equiv \overset{k+s}{\sim} \tag{7.31}$$

Beweis:

Die genau-dann-wenn Beziehung der obigen Behauptung beweisen wir in beide Richtungen. Wenn die k Relation gilt, dann muß auch die k-1 Relation gelten, da die k Relation alle Wörter enthält, die auch in k-1 enthalten sind.

Seien ein Wort $z \in \Sigma^*$ mit $|z| \le k-1$ und ein Zeichen $a \in \Sigma$ beliebig gewählt, dann gilt

$$\delta(\delta(q,a),z) \in F \Leftrightarrow \delta(q,az) \in F \Leftrightarrow \delta(q',az) \in F \Leftrightarrow \delta(\delta(q',a),z) \in F \tag{7.16}$$

und die Zustände q und q' gehören zur k-1 Relation

$$\delta(q,a) \overset{k-1}{\sim} \delta(q',a) \tag{7.17}$$

Damit ist gezeigt

$$q \overset{k}{\sim} q' \Rightarrow q \overset{k-1}{\sim} q' \tag{7.18}$$

Gilt bereits die k-1 Relation, so zeigen wir, dass die Behauptung in der anderen Richtung gilt. Für $\forall z \in \Sigma^*$ mit $|z| = k$ gilt

$$\delta(q,z) \in F \Leftrightarrow \delta(q',z) \in F \tag{7.19}$$

Sei $z \in \Sigma^*$ mit $|z| = k$ und $z = aw$ mit $a \in \Sigma \wedge |w| = k-1$. Dann läßt sich zeigen

$$\begin{aligned}
&\delta(q,a) \overset{k-1}{\sim} \delta(q',a) \\
&\Rightarrow (\delta(\delta(q,a),w) \in F \Leftrightarrow \delta(\delta(q',a),w) \in F) \\
&\Rightarrow (\delta(q,aw) = \delta(q,z) \in F \Leftrightarrow \delta(q',aw) = \delta(q',z) \in F) \\
&\Rightarrow q \overset{k}{\sim} q'
\end{aligned} \tag{7.20}$$

Somit gilt die Behauptung auch in der anderen Richtung

$$q \overset{k}{\sim} q' \Leftarrow q \overset{k-1}{\sim} q' \tag{7.21}$$

7.2.2 Verfeinerung von Äquivalenzrelationen

Zwischen Äquivalenzrelationen lassen sich Beziehungen ausdrücken, d. h. sie können kongruent oder verschieden sein. Nehmen wir die beiden Äquivalenzrelationen ρ_1 und ρ_2 die auf der Menge S liegen. Sind die ρ_1 und ρ_2 kongruent, dann

$$\begin{aligned}
&\rho_1 \equiv \rho_2 \Leftrightarrow \forall [x]_{\rho_1} \ \exists [y]_{\rho_2} : [x]_{\rho_1} = [y]_{\rho_2} \\
&\forall x, y \in S : x\rho_1 y \Leftrightarrow x\rho_2 y
\end{aligned} \tag{7.22}$$

Andernfalls ist eine Relation kleiner oder echt kleiner als die andere. Man spricht von *Verfeinerung*

$$\begin{aligned}
&\rho_1 \le \rho_2 \Leftrightarrow (x\rho_1 y \Rightarrow x\rho_2 y) \\
&\rho_1 < \rho_2 \Leftrightarrow (\rho_1 \le \rho_2 \wedge \rho_1 \not\equiv \rho_2)
\end{aligned} \tag{7.23}$$

7.2.3 Anwendung auf k Relation zur Zustandsreduktion

Betrachtet man die k Relationen, dann enthält die k Relation höchstens gleich viele Zustände wie die k-1 Relation. k ist daher eine Verfeinerung von k-1

$$\overset{k}{\sim} \le \overset{k-1}{\sim} \tag{7.24}$$

Wir behaupten zudem

6.2.2 Rechts-Invarianz

Eine Äquivalenzrelation $R \subseteq S \times S$ auf einer algebraischen Struktur $\langle S, \circ \rangle$ mit einer binären Verknüpfung $\circ : S \times S \to S$ heißt „*rechts-invariant*"

$$\underset{\text{Df}}{\Leftrightarrow} xRy \Rightarrow \forall z \in S : \underbrace{(x \circ z)}_{\in S} R \underbrace{(y \circ z)}_{\in S} \tag{6.7}$$

6.2.3 Links-Invarianz

Eine Äquivalenzrelation $R \subseteq S \times S$ auf einer algebraischen Struktur $\langle S, \circ \rangle$ mit einer binären Verknüpfung $\circ : S \times S \to S$ heißt „links-invariant"

$$\underset{\text{Df}}{\Leftrightarrow} xRy \Rightarrow \forall z \in S : \underbrace{(z \circ x)}_{\in S} R \underbrace{(z \circ y)}_{\in S} \tag{6.8}$$

6.2.4 Kongruenzrelation

Eine Äquivalenzrelation $R \subseteq S \times S$ auf einer algebraischen Struktur $\langle S, \circ \rangle$ mit einer binären Verknüpfung $\circ : S \times S \to S$ heißt „Kongruenzrelation":

$$\begin{aligned}
R \text{ Kongruenzrelation} \quad &\underset{\text{Df}}{\Leftrightarrow} xRy \Rightarrow \forall w, z \in S : \underbrace{(w \circ x \circ z)}_{\in S} R \underbrace{(w \circ y \circ z)}_{\in S} \Leftrightarrow \\
&\Leftrightarrow (R \text{ ist rechts-invariant}) \wedge (R \text{ ist links-invariant})
\end{aligned} \tag{6.9}$$

Eine Kongruenzrelation ist eine mit der Struktur verträgliche Äquivalenzrelation. Zum Beispiel ist auf dem Körper der reellen Zahlen die Gleichheitsoperation („=") bezüglich der Operationen „Addition" („+") und „Multiplikation" („·") eine Kongruenzrelation:

$$a = b \Rightarrow \forall x, y \in \square : (x \cdot a \cdot y) = (x \cdot b \cdot y), (x + a + y) = (x + b + y) \tag{6.10}$$

6.2.5 Quotientenmenge

Eine „*Quotientenmenge*" wird auf Basis einer Menge, und einer Kongruenzrelation definiert. Man versteht dann unter der „Quotientenmenge" die Menge aller Äquivalenzklassen, welche (bezüglich der gegebenen Kongruenzrelation) innerhalb der gegebenen Menge existieren.

Gegeben sei die algebraische Struktur $\langle \Sigma^*, \circ \rangle$ und die Kongruenzrelation „~". Unter der „Quotientenmenge" von Σ^* und ~ versteht man:

$$\Sigma^* /_{\square} \underset{\text{Df}}{=} \{ [x]_{\square} \mid x \in \Sigma^* \} \tag{6.11}$$

6.2.6 Quotientenmonoid

Unter dem „*Quotientenmonoid*" versteht man die Zusammenfassung der Quotientenmenge mit einer Operation auf dieser Menge. Die Abbildung $f : \Sigma^* \to \Sigma^* /_{\square}$ (der Elemente der ursprünglichen Menge auf die der Quotientenmenge) definiert man sinnvollerweise als einen Homomorphismus[3]. Die neue Operation für das Quotientenmonoid sei „•".

Das „Quotientenmonoid" ergibt sich dann als:

$$\langle \Sigma^* /_{\square}, \bullet \rangle \text{ bzw. } \langle \{ [x]_{\square} \mid x \in \Sigma^* \}, \bullet \rangle \tag{6.12}$$

[3] Bei der homomorphen Eigenschaft handelt es sich nicht um eine zwangsläufige Konsequenz, oder um eine durch das Monoid definierten Eigenschaft. Vielmehr wird der Homomorphismus (willkürlich) gefordert, um die neue, algebraische Struktur als „Ring" zu erhalten

Die Rechenregeln für das Quotientenmonoid leiten sich direkt aus der homomorphen Eigenschaft der Abbildung her:

$$[x]\bullet[y]\underset{\text{Df}}{=}[x\circ y] \tag{6.13}$$

$$\begin{aligned}&[x]=[x']\\&[y]=[y']\\&\Downarrow\\&x\sim x'\\&y\sim y'\\&\Downarrow \text{Kongruenz von } \sim\\&x\circ y\sim x'\circ y\\&x'\circ y\sim x'\circ y'\\&\Downarrow \text{Transitivität von } \sim\\&x\circ y\sim x'\circ y'\\&\Downarrow\\&[x\circ y]=[x'\circ y']\end{aligned} \tag{6.14}$$

Man sieht an der obigen Herleitung, wie bedeutsam die Eigenschaften „Kongruenz“ und „Transitivität“ der Kongruenzrelation „~“ für die Quotientenmenge sind.

Beispiel 1:

Gegeben sei die Menge der natürlichen Zahlen $\mathbb{N}$ mit der Verknüpfung „·“ (Multiplikation), und ein Modul *m*. Zwei Zahlen gelten als äquivalent bezüglich „~“, wenn sie den gleichen Rest bei einer Division durch *m* abwerfen:

$$\begin{aligned}&a,b\in\mathbb{N}\\&a\sim b\Leftrightarrow a \bmod m=b \bmod m\end{aligned} \tag{6.15}$$

Gemäß dieser Definition ist z.B. bei m=5 die Zahl 11 äquivalent zu 31: 11~31. Es handelt sich bei „~“ um eine Äquivalenzrelation, da die Bedingungen Reflexitivität, Transitivität und Symmetrie erfüllt sind (der Nachweis dieser Eigenschaften sollte keine Probleme bereiten).

Es handelt sich bei „~“ aber nicht nur um eine Äquivalenzrelation, sondern sogar um eine Kongruenzrelation. Der Nachweis hierfür ist etwas aufwendiger:

Nach (6.9) gilt:

$\sim$ Kongruenzrelation $\underset{\text{Df}}{\Leftrightarrow} a\sim b\Rightarrow \forall x,y\in\mathbb{N}:\underbrace{(x\cdot a\cdot y)}_{\in\mathbb{N}}\sim\underbrace{(x\cdot b\cdot y)}_{\in\mathbb{N}}\Leftrightarrow$

$$\Leftrightarrow a \bmod m=b \bmod m\Rightarrow \forall x,y\in\mathbb{N}:\underbrace{(x\cdot a\cdot y)}_{\in\mathbb{N}} \bmod m=\underbrace{(x\cdot b\cdot y)}_{\in\mathbb{N}} \bmod m$$

Die abschließende Implikation sollte nach kurzem Nachdenken für jedermann ersichtlicherweise korrekt sein; ein Beweis an dieser Stelle ist nicht nötig.

Daher gelten gemäß (6.13) und (6.14) die folgenden Rechengesetze:

$$\begin{aligned}&\forall a,b\in\mathbb{N}\\&a \bmod m\bullet b \bmod m=a\cdot b \bmod m\\&a \bmod m=a' \bmod m, b \bmod m=b' \bmod m\\&\Downarrow\\&a\cdot b \bmod m=a'\cdot b' \bmod m\end{aligned} \tag{6.16}$$

$$\begin{aligned}&h(q)=[x]_{R_L} \text{ mit } \delta(q_0,x)=q\\&h(p)=[y]_{R_L} \text{ mit } \delta(q_0,y)=p\end{aligned} \tag{7.11}$$

Dann sind x und y nicht äquivalent zueinander (bezüglich der ~ Relation) und damit gilt Injektivität von h ($x\sim y\Rightarrow xR_Ly$ gilt immer).

Isomorphismus von h

Da h injektiv und surjektiv, ist die Abbildung bijektiv. h ist daher ein Isomorphismus. Für einen Zustand q existiert ein x, dass den Automaten M in diesen Zustand überführt.

$$q\in Q\Rightarrow \exists x \text{ mit } \delta(q_0,x)=q \tag{7.12}$$

$$\begin{aligned}\delta_N(h(q),a)&=\delta_N(h(\delta(q_0,x)),a)\\&=\delta_N([x],a)\\&=[xa]\\&=h(\delta(q_0,xa))\\&=h(\delta(\delta(q_0,x),a))\\&=h(\delta(q,a))\end{aligned} \tag{7.13}$$

Die Herleitung zeigt, dass es egal ist, ob die Konkatenation von x und a zunächst in der Urmenge (die Zustände von Q) stattfindet und dann nach Q_N abgebildet wird oder umgekehrt.

Damit ist h isomorph und als Korollar kann festgestellt werden, dass alle minimalen Automaten (DFA) isomorph sind. Anders ausgedrückt sind alle minimalen Automaten bis auf Isomorphien eindeutig bestimmt. Die Verdrahtung der Zustände ist gleich, lediglich die Zustandsbezeichnung kann variieren.

7.2 Minimierung von endlichen Automaten

7.2.1 Rekursion der k Relationen

Endliche Automaten können sich in einem nicht minimalen Zustanden befinden. Mit einiger Logik und Transformation lassen sie sich in minimale Automaten überführen, die isomorph zum Nerode Automaten sind (wenn es sich denn um DFAs handelt - bei NFAs ist die Anzahl der nicht isomorphen minimalen Automaten u. U. beliebig groß).

Für die Minimierung - auch Zustandsreduktion genannt - führen wir zunächst eine neue Äquivalenzrelation k zwischen zwei Zuständen q und q' ein. Zustände sind bezüglich k äquivalent, wenn sie Wörter x, deren Länge kleiner gleich k ist, in einen Endzustand überführen oder beide Zustände die Wörter nicht in einen Endzustand überführen.

Der Betrachtung zugrunde liegt ein endlicher Automat $M=(Q,\Sigma,\delta,q_0,F)$ mit q und q' Zuständen aus Q. Dann seien folgende Äquivalenzrelationen definiert

$$\begin{aligned}&q\overset{k}{\sim}q'\Leftrightarrow\left(\forall x\in\Sigma^*,\ |x|\le k:\delta(q,x)\in F\Leftrightarrow\delta(q',x)\in F\right)\\&q\sim q'\Leftrightarrow\left(\forall x\in\Sigma^*,\ \delta(q,x)\in F\Leftrightarrow\delta(q',x)\in F\right)\end{aligned} \tag{7.14}$$

Scheinbar besteht zwischen der auf eine Wortlänge k beschränkten und der allgemeinen ~ Relation ein Zusammenhang. Daher behaupten wir

Behauptung:

$$q\overset{k}{\sim}q'\Leftrightarrow q\overset{k-1}{\sim}q'\wedge\left(\forall a\in\Sigma,\ \delta(q,a)\overset{k-1}{\sim}\delta(q',a)\right) \tag{7.15}$$

$$\begin{tikzcd} Q_1 \times \Sigma_1 \arrow[r, "h"] \arrow[d, "\delta_1"] & Q_2 \times \Sigma_2 \arrow[d, "\delta_2"] \\ Q_1 \arrow[r, "h"] & Q_2 \end{tikzcd}$$

Abbildung 13 - Homomorphismus zwischen Automat M und Nerode Automat

Es existiert eine Abbildung h, die alle Zustände von M in Zustände in M_N (Äquivalenzklassen) überführt. Formal ausgedrückt

$$h(q) = [x]_{R_L} \Leftrightarrow \exists x : \delta(q_0, x) = q \tag{7.4}$$

Hier stellt sich die Frage, ob h eindeutig definiert ist. Alle Wörter x und y, die den Automaten M in den gleichen Zustand überführen, müssen auch M_N in die gleiche Äquivalenzklasse überführen. Zu Zeigen ist, dass x und y der gleichen Äquivalenzklasse von R_L angehören.

$$\begin{gathered} x, y \in \Sigma^* : \delta(q_0, x) = q = \delta(q_0, y) \\ \overset{?}{\Leftrightarrow} \\ [x]_{R_L} = [y]_{R_L} \end{gathered} \tag{7.5}$$

Beweis:

Zum Beweis führen wir eine neue Äquivalenzrelation ~ ein

$$x \sim y \Leftrightarrow x, y \in \Sigma^* : \delta(q_0, x) = \delta(q_0, y) \tag{7.6}$$

Überführen zwei Wörter x und y den Automaten in den gleichen Zustand, so sind sie äquivalent. Da die Anzahl der Zustände der oben betrachteten Automaten minimal und gleich ist, gilt für den Index (die Anzahl der maximal paarweise verschiedenen Äquivalenzklassen) der ~ Relation

$$index(\sim) \leq |Q| = |Q_N| \tag{7.7}$$

Weiterhin gelte

$$x, y \in \Sigma^* : \delta(q_0, x) = \delta(q_0, y) \Rightarrow x \sim y \Rightarrow x R_L y \Rightarrow [x]_{R_L} = [y]_{R_L} \tag{7.8}$$

Somit ist h als eindeutige Abbildung von Zuständen Q auf Äquivalenzklassen aus Q_N definiert.

Könnte es Zustände q in Automat M geben, für die h(q) nicht definiert ist?

$$\exists q \in Q \text{ mit } h(q) \text{ nicht definiert} \Rightarrow \neg\exists x \text{ mit } \delta(q_0, x) = q \tag{7.9}$$

Wenn ein solcher Zustand existieren würde, dann gebe es kein Wort x, dass den Automaten in diesen Zustand überführt und der Zustand somit überflüssig. Damit wäre M nicht mehr minimal und ein Widerspruch läge vor. h ist für alle Zustände q aus Q definiert.

Surjektivität von h

$$[x] \in Q_N, \exists q \in Q \text{ mit } \delta(q_0, x) = q \wedge h(q) = x \tag{7.10}$$

Injektivität von h

Wenn $q \neq p$ zwei verschiedene Zustände aus Q sind und [x] und [y] ihre Abbildungen in Q_N, dann gilt

6.3 Reguläre Ausdrücke

6.3.1 Kleenesche Hülle

Sei Σ eine endliche Menge von Symbolen und seien *L*, L_1 und L_2 Mengen von Zeichenketten aus Σ*. Die Konkatenation von L_1 und L_2 – geschrieben als L_1L_2 – ist die Menge $\{xy \mid x \text{ ist aus } L_1 \text{ und } y \text{ ist aus } L_2\}$. Die Zeichenketten in L_1L_2 werden gebildet, indem man an eine aus L_1 gewählte Zeichenkette eine Zeichenkette aus L_2 anhängt und das in allen möglichen Kombinationen. Wir definieren nun:

$$\begin{aligned} &L^0 \underset{\text{Df}}{=} \{\in\} \\ &L^i \underset{\text{Df}}{=} LL^{i-1} \quad \forall i \geq 1 \\ &L^* = \bigcup_{i=0}^{\infty} L^i \qquad \text{(Hülle, Kleenesche Hülle)} \\ &L^+ = \bigcup_{i=1}^{\infty} L^i \qquad \text{(positive Hülle)} \end{aligned} \tag{6.17}$$

*L** bezeichnet also alle Wörter, die durch die Konkatenation einer beliebigen Anzahl von Wörtern aus *L* entstehen.

Beispiel:

Sei $L_1 = \{10,1\}$ und $L_2 = \{011,11\}$. Dann ist $L_1L_2 = \{10011,1011,111\}$. Ebenso gilt $\{10,11\}^* = \{\in,10,11,1010,1011,1110,1111,...\}$.

6.3.2 Reguläre Ausdrücke

Die von endlichen Automaten akzeptierten Sprachen lassen sich durch einfache Ausdrücke beschreiben, die als „*reguläre Ausdrücke*“ bezeichnet werden. Auf Grund dieser Äquivalenz werden die Sprachen der endlichen Automaten auch „*reguläre Mengen*“ genannt.

Oder umgekehrt: *Reguläre Ausdrücke* bilden einen Formalismus zur Beschreibung *formaler Sprachen*.

Zwei reguläre Ausdrücke heißen „*äquivalent*“, wenn sie die gleiche formale Sprache beschreiben (z.B. [ab]*a und a[ba]*). Die Äquivalenz regulärer Ausdrücke lässt sich über algebraische Gesetze zeigen.

6.3.3 Reguläre Ausdrücke vs. endliche Automaten

Satz:

Die folgenden drei Aussagen sind äquivalent:

3. *L* ist reguläre Sprache/reguläre Menge
   ⇕
4. *L* ist die Vereinigung von einigen Äquivalenzklassen einer rechts-invarianten Äquivalenzrelation von endlichem Index.
   ⇕
5. Sei R_L eine Relation, die wie folgt definiert ist:

   $$x\, R_L\, y \Leftrightarrow (\forall z \in \Sigma^* : xz \in L \Leftrightarrow yz \in L) \tag{6.18}$$

   Dann ist R_L von endlichem Index.

Beweis:

$1 \Rightarrow 2$:

Wenn L eine reguläre Menge ist, dann existiert ein endlicher Automat (DFA/NFA), der exakt diese Menge beschreibt:

$$L \text{ ist regulär} \Leftrightarrow \exists M : M = (Q, \Sigma, \delta, q_0, F) : L = T(M) \tag{6.19}$$

Sei ρ eine Äquivalenzrelation, welche alle Wörter aus Σ*, die zu dem selben Zustand des Automaten führen, in einer Äquivalenzklasse zusammenfasst:

$$x, y \in \Sigma^* : x \; \rho \; y \underset{\mathrm{Df}}{\Leftrightarrow} \delta(q_0, x) = \delta(q_0, y) \tag{6.20}$$

Die Äquivalenzrelation ρ ist rechts-invariant:

$$\forall x, y, z \in \Sigma^* : \delta(q_0, x) = \delta(q_0, y) \Rightarrow \delta(q_0, xz) = \delta(q_0, yz) \tag{6.21}$$

Dann lässt sich hieraus die Behauptung folgern:

$$\operatorname{index}(\rho) \le |Q| \tag{6.22}$$

Wieso lässt sich diese Behauptung folgern? Nun, gehen wir von dem Gegenteil aus, nämlich dass $\operatorname{index}(\rho) > |Q|$. Das würde bedeuten, dass es mehr Wörter gibt, als es Zustände gibt, die aber dennoch alle zu verschiedenen Zuständen führen. Der Widerspruch dieser Annahme liegt auf der Hand.

Damit lässt sich die reguläre Menge L darstellen als Menge der Eingabewörter, welche zu einem der akzeptierenden Zustände des Automaten führt. Jeder akzeptierende Zustand des Automaten kann bezüglich der Äquivalenzrelation (definitionsgemäß) nur von einer Klasse von Eingabewörtern erreicht werden.

Die reguläre Menge L können wir darstellen als Vereinigung aller akzeptierenden Zustände des Automaten, und damit als Vereinigung einiger Äquivalenzklassen einer rechts-invarianten Äquivalenzrelation von endlichem Index:

$$L = \bigcup_{\delta(q_0, x) \in F} [x]_\rho \tag{6.23}$$

$2 \Rightarrow 3$:

Im vorangegangenen Teil wurde die Äquivalenzrelation ρ eingeführt, diese benutzen wir weiterhin. Darüber hinaus definieren eine (zweite) Äquivalenzrelation R_L wie folgt:

$$\begin{aligned} \forall x, y \in \Sigma^* : x \; R_L \; y \quad & \Leftrightarrow \left(\forall z \in \Sigma^* : xz \in L \Leftrightarrow yz \in L\right) \Leftrightarrow \\ & \Leftrightarrow \left(\forall z \in \Sigma^* : \left(xz \in L \wedge yz \in L\right) \vee \left(xz \notin L \wedge yz \notin L\right)\right) \end{aligned} \tag{6.24}$$

Die Definition der Äquivalenzrelation R_L ist etwas kompliziert. Textuell formuliert bedeutet (6.24), dass alle jenen Wörter aus Σ* in Relation stehen, die a) entweder beide in der Sprache enthalten sind, oder beide nicht in der Sprache enthalten sind; und b) die Bedingung a) auch dann noch erfüllen, wenn ihnen ein beliebiges Wort „angehängt wird".

Die Definition dieser Äquivalenzrelation erinnert an die Zustandsminimierung via Implikationstafel. Man bezeichnet dabei alle Zustände als „äquivalent", die bei den selben Eingabesymbolen in die selben Nachfolgezustände, oder in äquivalente Nachfolgezustände überführt werden.

Behauptung:

$$\forall x, y \in \Sigma^* : x \; \rho \; y \Rightarrow x \; R_L \; y \tag{6.25}$$

7 Vorlesung vom 27. April 2000

7.1 Nerode Automat

7.1.1 Homomorphismus und Isomorphismus

Gegeben seien zwei Gruppen S_1 und S_2 mit $\langle S_1, \cdot \rangle$ und $\langle S_2, \circ \rangle$. Eine Abbildung $h : S_1 \to S_2$ heißt Homomorphismus, wenn für alle $x, y \in S_1$ gilt

$$\begin{aligned} & h(x \cdot y) = h(x) \circ h(y) \\ & \text{bzw.} \\ & h(\cdot(x,y)) = \circ(h(x), h(y)) \end{aligned} \tag{7.1}$$

Eine Illustration verdeutlicht das

$$\begin{tikzcd} S_1 \times S_1 \arrow[r, "h"] \arrow[d, "\cdot"] & S_2 \times S_2 \arrow[d, "\circ"] \\ S_1 \arrow[r, "h"] & S_2 \end{tikzcd}$$

Abbildung 12 - Homomorphismus

Ist eine homomorphe Abbildung bijektiv, so wird sie isomorph genannt. Man sagt, die Struktur beider Gruppen ist gleich, sie unterscheiden sich nur in den Bezeichnungen der Elemente.

7.1.2 Minimalität und Eindeutigkeit des Nerode Automaten

Nachdem in den vorangegangenen Abschnitten gezeigt wurde, dass der Nerode Automat minimal ist, vergleichen wir ihn jetzt mit anderen minimalen Automaten. Dabei werden wir feststellen, dass alle minimalen Automaten (DFA) einer Sprache L die gleiche "Verdrahtung" ihrer Zustände haben und sich lediglich in der Bezeichnung der Zustände unterscheiden, sie also isomorph sind.

Sei L eine Sprache, die von einem minimalen Automaten M und dem Nerode Automaten M_N erkannt wird

$$\begin{aligned} M &= (Q, \Sigma, \delta, q_0, F) & L &= T(M) \\ M_N &= (Q_N, \Sigma, \delta_N, q_{0N}, F_N) & L &= T(M_N) \end{aligned} \tag{7.2}$$

dann seien M und M_N wie oben beschrieben isomorph, d. h. es existiert ein Homomorphismus

$$\begin{aligned} & h : Q \to Q_N \\ & h(\delta(q, a)) = \delta_N(h(q), a) \end{aligned} \tag{7.3}$$

Das wird in folgendem Schaubild illustriert

Beweis:

Weil sowohl M als auch M_N minimal sind, müssen sie über die gleiche Anzahl an Zuständen verfügen:

$$|Q| = |Q_N| \tag{6.46}$$

Sei *h()* so definiert, dass es jeden Zustand q aus der Menge der Zustände Q auf eine Äquivalenzklasse bezüglich der Äquivalenzrelation R_L abbilde (wir erinnern uns: die Äquivalenzklassen bezüglich der Äquivalenzrelation R_L liegen dem konstruierten Nerode Automaten zugrunde).:

$$h(q) \underset{\text{Df}}{=} [x]_{R_L} \Leftrightarrow \exists x : \delta(q_0, x) = q \tag{6.47}$$

Beweis zu (6.25):

Bei dem Beweis machen wir uns die Tatsache zunutze, dass die Äquivalenzrelation ρ rechts-invariant ist:

$$\begin{aligned}
&\forall x, y \in \Sigma^* : x\ \rho\ y \Rightarrow \forall z \in \Sigma^* : xz\ \rho\ yz \\
&\Downarrow \\
&\forall x, y \in \Sigma^* : x\ \rho\ y \Rightarrow \left(\forall z \in \Sigma^* : xz \in L \Leftrightarrow yz \in L\right) \\
&\Downarrow \\
&\forall x, y \in \Sigma^* : x\ \rho\ y \Rightarrow \left(\forall z \in \Sigma^* : (xz \in L \wedge yz \in L) \vee (xz \notin L \wedge yz \notin L)\right) \\
&\Updownarrow \text{ siehe oben} \\
&\forall x, y \in \Sigma^* : x\ \rho\ y \Rightarrow x\ R_L\ y
\end{aligned} \tag{6.26}$$

Aus der Tatsache, dass jedes Paar von Eingabewörtern aus Σ*, dass in der Äquivalenzrelation ρ enthalten ist, automatisch auch in der Äquivalenzrelation R_L enthalten ist, resultiert, dass die Äquivalenzklasse eines beliebigen Eingabewortes bezüglich der Äquivalenzrelation ρ eine Teilmenge der Äquivalenzklasse bezüglich der Äquivalenzrelation R_L ist:

$$\forall x \in \Sigma^* : [x]_\rho \subseteq [x]_{R_L} \tag{6.27}$$

Daraus folgt, dass es bezüglich der Äquivalenzrelation ρ mindestens so viele (disjunkte) Äquivalenzklassen geben muss, wie bezüglich der Äquivalenzrelation R_L:

$$\text{index}(R_L) \leq \text{index}(\rho) \leq \text{"endlich"} \tag{6.28}$$

Daraus wiederum folgt, dass auch der Index der Äquivalenzrelation R_L endlich sein muß:

$$\text{index}(\mathrm{R_L}) \text{ ist endlich} \tag{6.29}$$

$2 \Rightarrow 3$:

Behauptung:

Die Äquivalenzrelation R_L ist rechts-invariant.

Beweis:

$$\begin{aligned}
&\forall x, y, w, z \in \Sigma^* : x\ R_L\ y \\
&\Updownarrow \text{ Definition } \mathrm{R_L} \\
&(x \in L \Leftrightarrow y \in L) \\
&\Updownarrow \text{ Determinismus des Automaten} \\
&\left(x(zw) \in L \Leftrightarrow y(zw) \in L\right) \\
&\Updownarrow \text{ Assoziativität der Konkatenation} \\
&\left((xz)w \in L \Leftrightarrow (yz)w \in L\right) \\
&\Updownarrow \\
&\forall x, y, z \in \Sigma^* : x\ R_L\ y \Leftrightarrow xz\ R_L\ yz \\
&\Downarrow \\
&R_L \text{ ist rechts-invariant}
\end{aligned} \tag{6.30}$$

6.4 Nerode Automat

6.4.1 Definition des Nerode Automaten

Mit dem bisher erlangten Wissen, ist es möglich, einen minimalen Automaten zu erstellen, der eine reguläre Sprache repräsentiert. Dieser Automat heißt „*Nerode Automat*".

Zur Konstruktion des Nerode Automaten wird erneut die Äquivalenzrelation R_L herangezogen, die wie folgt definiert ist:

$$\begin{aligned}\forall x,y\in\Sigma^*: x\,R_L\,y \quad &\Leftrightarrow \left(\forall z\in\Sigma^*: xz\in L \Leftrightarrow yz\in L\right)\Leftrightarrow\\ &\Leftrightarrow \left(\forall z\in\Sigma^*: \left(xz\in L\wedge yz\in L\right)\vee\left(xz\notin L\wedge yz\notin L\right)\right)\end{aligned} \tag{6.31}$$

Die Menge der unterschiedlichen Äquivalenzklassen, die durch R_L definiert werden, wird im Folgenden mit „Q_N" bezeichnet:

$$Q_N \underset{\text{Df}}{=} \left\{[x]_{R_L} \,\middle|\, x\in\Sigma^*\right\} \tag{6.32}$$

Der Nerode Automat ist definiert als:

$$M_N \underset{\text{Df}}{=} \left\{Q_N,\Sigma,\delta_N,[e]_{R_L},F_N\right\} \qquad \text{(Nerode Automat)} \tag{6.33}$$

mit

$$F_N = \left\{[x]_{R_L} \,\middle|\, x\in L\right\} \tag{6.34}$$

$$\delta_N\left([x],a\right) \underset{\text{Df}}{=} [xa] \tag{6.35}$$

δ_N ist damit wohldefiniert.

6.4.2 Repräsentantenunabhängigkeit des Nerode Automaten:

Im Folgenden wird gezeigt, dass die Übergangsfunktion δ_N wohldefiniert ist, d.h. ihre Definition repräsentantenunabhängig ist. Dazu ist es nötig, zu zeigen, dass die Übergangsfunktion für alle Eingabewörter, die äquivalent sind, den selben Nachfolgezustand liefert.

Zu zeigen:

$$[x]_{R_L} = [y]_{R_L} \Rightarrow \delta_N\left([x]_{R_L},a\right) = \delta_N\left([y]_{R_L},a\right) \tag{6.36}$$

Beweis:

$$\begin{gathered}[x]_{R_L} = [y]_{R_L}\\ \Downarrow\\ x\,R_L\,y\\ \Downarrow\\ xa\,R_L\,ya\\ \Downarrow\\ [xa]_{R_L} = [ya]_{R_L}\\ \Downarrow\\ \delta_N\left([x]_{R_L},a\right) = \delta_N\left([y]_{R_L},a\right)\\ \Downarrow\\ M_N \text{ ist endlicher Automat}\end{gathered} \tag{6.37}$$

6.4.3 Regularität des Nerode Automaten:

Im Folgenden wird bewiesen, dass die Sprache, die der Nerode Automat erzeugt/definiert, eine reguläre Sprache ist.

$$\begin{aligned} x\in T\left(M_N\right) \quad &\Leftrightarrow \delta_N\left([e],x\right)\in F_N\\ &\Leftrightarrow [x]\in F_N\\ &\Leftrightarrow x\in L \end{aligned} \tag{6.38}$$

$$\Rightarrow T\left(M_N\right) = L \Rightarrow$$

$$\Rightarrow L \text{ ist regulär}$$

6.4.4 Minimalität des Nerode Automaten

Im Folgenden wird bewiesen, dass der Nerode Automat minimal ist, d.h. dass jeder endliche Automat, der die selbe Sprache L definiert, mindestens genauso viele Zustände haben muß wie M_N.

Satz:

Der Nerode Automat ist minimal.

Beweis:

Der Beweis ist ein Widerspruchsbeweis. Es wird angenommen, dass sehr wohl ein Automat existiert, der die selbe Sprache L definiert, der aber mit weniger Zuständen auskommt:

$$\exists M: M = \left(Q,\Sigma,\delta,q_0,F\right) \quad T\left(M\right) = L \quad |Q| < |Q_N| \tag{6.39}$$

Die Definition der Äquivalenzrelation ρ wird aus (6.20), (6.22) und (6.28) übernommen:

$$x,y\in\Sigma^*: x\,\rho\,y \underset{\text{Df}}{\Leftrightarrow} \delta\left(q_0,x\right) = \delta\left(q_0,y\right) \tag{6.40}$$

$$\operatorname{index}\left(\rho\right) \le |Q| \tag{6.41}$$

$$\operatorname{index}\left(R_L\right) \le \operatorname{index}\left(\rho\right) \le \text{"endlich"} \tag{6.42}$$

Weil die Menge Q_N der Menge der Äquivalenzklassen entspricht, die durch die Äquivalenzrelation R_L definiert werden (siehe (6.32)), gilt:

$$|Q_N| = \operatorname{index}\left(R_L\right) \tag{6.43}$$

Aus (6.39), (6.41), (6.42) und (6.43) folgt:

$$\begin{gathered}\operatorname{index}\left(\rho\right) \le |Q| < |Q_N| = \operatorname{index}\left(R_L\right) \le \operatorname{index}\left(\rho\right)\\ \Downarrow\\ \operatorname{index}\left(\rho\right) < \operatorname{index}\left(\rho\right)\\ \Downarrow\\ \text{Widerspruch}\end{gathered} \tag{6.44}$$

6.4.5 Eindeutigkeit des Nerode Automaten

Satz:

Sei $M = \left(Q,\Sigma,\delta,q_0,F\right)$ ein minimaler Automat für eine Sprache L mit $T\left(M\right) = L$.

Sei $M_N = \left\{Q_N,\Sigma,\delta_N,q_{0N},F_N\right\}$ der Nerode Automat für L mit $T\left(M_N\right) = L$. Dann ist M isomorph zu M_N. D.h. es existiert ein Isomorphismus

$$h: Q \rightarrow Q_N \quad \text{mit} \quad h\left(\delta\left(q,a\right)\right) = \delta_N\left(h\left(q\right),a\right) \tag{6.45}$$