

Theoretische Informatik

Übungsstunde 2

Emil Vinu

Aufbau

- Feedback Blatt 1
- Recap
- Kahoot
- Aufgaben

Feedback Blatt 1

Feedback Blatt 1

- Aufgabe 2 lief sehr gut
- In Aufgabe 1 gingen die meisten Punkte verloren
 - fast nie wegen eines falschen Beispiels
 - sondern weil die **Begründung fehlte**

→ *Beispiel + eine Zeile Rechnung reicht*

Aufgabe 1(a)

Gesucht: Σ, L_1, L_2 mit $|L_1| = k$ und $|L_1 L_2| = k + 1$.

Aufgabe 1(a)

Gesucht: Σ, L_1, L_2 mit $|L_1| = k$ und $|L_1 L_2| = k + 1$.

$\Sigma = \{a\}$, $L_1 = \{a^i \mid 1 \leq i \leq k\}$, $L_2 = \{\lambda, a\}$. Dann ist

$$L_1 L_2 = \{a^i \mid 1 \leq i \leq k\} \cup \{a^i \mid 2 \leq i \leq k + 1\} = \{a^i \mid 1 \leq i \leq k + 1\},$$

also $|L_1 L_2| = k + 1$.

Aufgabe 1(a)

Gesucht: Σ, L_1, L_2 mit $|L_1| = k$ und $|L_1 L_2| = k + 1$.

$\Sigma = \{a\}$, $L_1 = \{a^i \mid 1 \leq i \leq k\}$, $L_2 = \{\lambda, a\}$. Dann ist

$$L_1 L_2 = \{a^i \mid 1 \leq i \leq k\} \cup \{a^i \mid 2 \leq i \leq k + 1\} = \{a^i \mid 1 \leq i \leq k + 1\},$$

also $|L_1 L_2| = k + 1$.

„Die Rechnung ist die Begründung.“

! muss für alle $k \geq 1$ klappen, also $k = 1$ testen

Aufgabe 1(d)

$$\Sigma = \{a\}, \quad L_1 = \{a^{li} \mid 0 \leq i \leq k-1\}, \quad L_2 = \{a^j \mid 0 \leq j \leq l-1\}$$

Warum hat $L_1 L_2 = \{a^{li+j}\}$ genau $k \cdot l$ Wörter?

Aufgabe 1(d)

$$\Sigma = \{a\}, L_1 = \{a^{li} \mid 0 \leq i \leq k-1\}, L_2 = \{a^j \mid 0 \leq j \leq l-1\}$$

Warum hat $L_1 L_2 = \{a^{li+j}\}$ genau $k \cdot l$ Wörter?

Weil $li + j$ mit $0 \leq j < l$ eine Division mit Rest durch l ist: i und j sind durch die Länge eindeutig bestimmt.

„Die Wörter aus L_2 sind zu kurz, um das nächste Wort aus L_1 zu erreichen.“

Kleene-Stern

Definition 2.9. $L^0 = \{\lambda\}$, $L^{i+1} = L^i \cdot L$, $L^* = \bigcup_{i \in \mathbb{N}} L^i$ und $L^+ = \bigcup_{i \geq 1} L^i = L \cdot L^*$.

! $\emptyset^* = \{\lambda\}$, nicht $\emptyset$

! $\lambda \in L^*$ für jede Sprache L

Kleene-Stern

Definition 2.9. $L^0 = \{\lambda\}$, $L^{i+1} = L^i \cdot L$, $L^* = \bigcup_{i \in \mathbb{N}} L^i$ und $L^+ = \bigcup_{i \geq 1} L^i = L \cdot L^*$.

! $\emptyset^* = \{\lambda\}$, nicht $\emptyset$

! $\lambda \in L^*$ für jede Sprache L

Damit ist 2(c) schnell erledigt: $\lambda \in (L^c)^*$, aber $\lambda \notin (L^*)^c$.

Recap

Kanonische Ordnung

Definition 2.8. Sei $\Sigma = \{s_1, \dots, s_m\}$ mit $s_1 < \dots < s_m$. Für $u, v \in \Sigma^*$ ist $u < v \iff |u| < |v| \vee (|u| = |v| \wedge u = xs_iu' \wedge v = xs_jv' \text{ mit } i < j)$.

$\lambda < a < b < aa < ab < ba < bb < aaa < \dots$

„Erst nach Länge, dann wie im Lexikon.“

Endlicher Automat

Definition 3.1. Ein (deterministischer) endlicher Automat (EA) ist ein Quintupel $M = (Q, \Sigma, \delta, q_0, F)$, wobei Q eine endliche Menge von Zuständen, Σ ein Alphabet, $q_0 \in Q$ der Anfangszustand, $F \subseteq Q$ die Menge der akzeptierenden Zustände und $\delta: Q \times \Sigma \rightarrow Q$ die Übergangsfunktion ist.

„Ein Programm ohne Variablen. Der Zustand ist das einzige Gedächtnis.“

! δ ist eine Funktion: genau ein Pfeil pro Zustand und Buchstabe

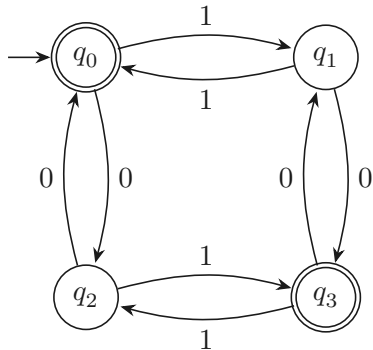
Konfiguration und Berechnung

- Konfiguration: $(q, w) \in Q \times \Sigma^*$
- Schritt: $(q, aw) \vdash_M (p, w)$, falls $\delta(q, a) = p$
- Berechnung auf x : $(q_0, x) \vdash_M \cdots \vdash_M (q, \lambda)$
- M akzeptiert x , falls $q \in F$

„Konfiguration = wo bin ich, und was muss ich noch lesen?“

→ auf jeder Eingabe gibt es genau eine Berechnung

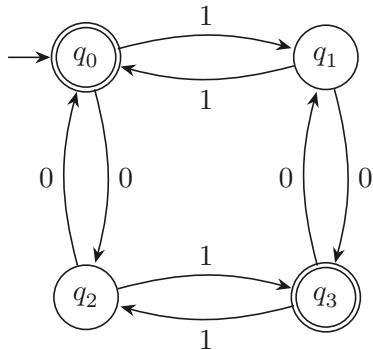
Beispiel



Berechnung auf 1011:

$$\begin{aligned} (q_0, 1011) &\vdash (q_1, 011) \vdash (q_3, 11) \\ &\vdash (q_2, 1) \vdash (q_3, \lambda) \end{aligned}$$

Beispiel



Berechnung auf 1011:

$(q_0, 1011) \vdash (q_1, 011) \vdash (q_3, 11)$
 $\vdash (q_2, 1) \vdash (q_3, \lambda)$

$q_3 \in F$, also $1011 \in L(M)$.

Klassen

Definition 3.2. $\hat{\delta}(q, \lambda) = q$ und $\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$.

$$\text{Kl}[p] = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) = p\}$$

- die Klassen zerlegen Σ^*
- $L(M) = \bigcup_{p \in F} \text{Kl}[p]$

„Jeder Zustand ist eine Schublade für die Wörter, die dort landen.“

Zwischenfrage

Welche Sprache akzeptiert der Automat aus dem Beispiel? Geht es mit weniger Zuständen?

Lösung

$\text{Kl}[q_0]: \quad |w|_0 \text{ gerade, } |w|_1 \text{ gerade}$

$\text{Kl}[q_1]: \quad |w|_0 \text{ gerade, } |w|_1 \text{ ungerade}$

$\text{Kl}[q_2]: \quad |w|_0 \text{ ungerade, } |w|_1 \text{ gerade}$

$\text{Kl}[q_3]: \quad |w|_0 \text{ ungerade, } |w|_1 \text{ ungerade}$

$$L(M) = \text{Kl}[q_0] \cup \text{Kl}[q_3] = \{w \mid |w| \text{ gerade}\}$$

Lösung

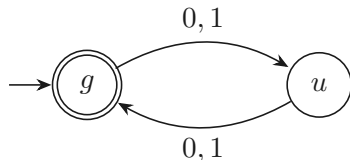
$\text{Kl}[q_0]: |w|_0 \text{ gerade, } |w|_1 \text{ gerade}$

$\text{Kl}[q_1]: |w|_0 \text{ gerade, } |w|_1 \text{ ungerade}$

$\text{Kl}[q_2]: |w|_0 \text{ ungerade, } |w|_1 \text{ gerade}$

$\text{Kl}[q_3]: |w|_0 \text{ ungerade, } |w|_1 \text{ ungerade}$

$$L(M) = \text{Kl}[q_0] \cup \text{Kl}[q_3] = \{w \mid |w| \text{ gerade}\}$$



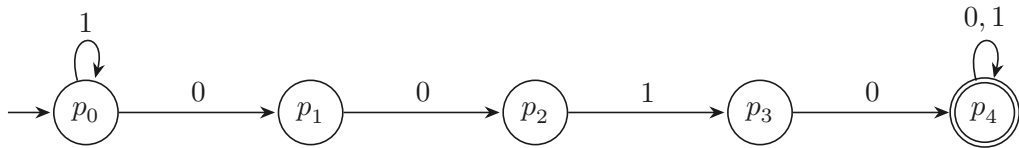
„Zwei Zustände reichen.“

Wie entwirft man einen EA?

1. Was muss ich mir merken? Jede Information wird ein Zustand.
2. $Kl[q]$ für jeden Zustand aufschreiben.
3. Übergänge folgen aus den Klassen.
4. Akzeptierend sind die Zustände, deren Klasse in L liegt.

→ *die Klassen sind fast schon die Begründung*

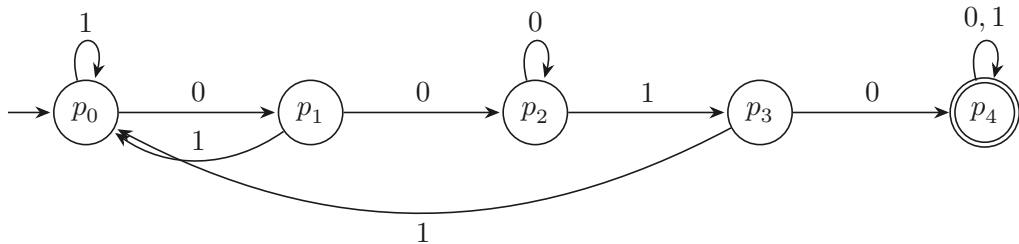
Beispiel: Teilwort 0010



„ p_i : die letzten i Zeichen sind der Anfang von 0010.“

! die Rückwärtspfeile sind die Falle

Beispiel: Teilwort 0010



„ p_i : die letzten i Zeichen sind der Anfang von 0010.“

! die Rückwärtspfeile sind die Falle

Produktautomat

Lemma 3.2. *Seien M_1, M_2 zwei EA über Σ . Für jedes $\odot \in \{\cup, \cap, -\}$ existiert ein EA M mit $L(M) = L(M_1) \odot L(M_2)$.*

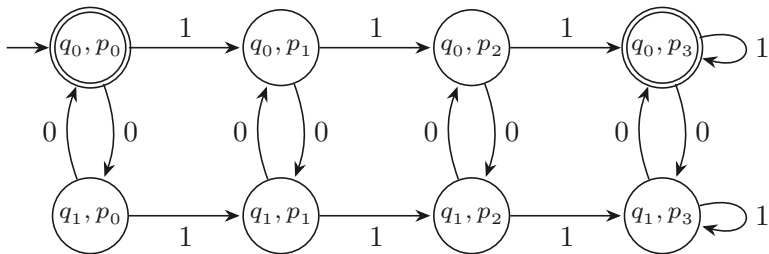
- $Q = Q_1 \times Q_2, q_0 = (q_{01}, q_{02})$
- $\delta((q, p), a) = (\delta_1(q, a), \delta_2(p, a))$
- $F = F_1 \times Q_2 \cup Q_1 \times F_2$ bzw. $F_1 \times F_2$ bzw. $F_1 \times (Q_2 - F_2)$

„Beide Automaten laufen gleichzeitig.“

Beispiel 3.3

L_1 : $|x|_0$ gerade (q_0, q_1)

L_2 : $|x|_1 = 0$ oder ≥ 3 (p_j : j Einsen)



Für $L_1 \cap L_2$: $F = \{(q_0, p_0), (q_0, p_3)\}$

Zwischenfrage

Welche Zustände sind akzeptierend für $L_1 - L_2$? Und für $L_2 - L_1$?

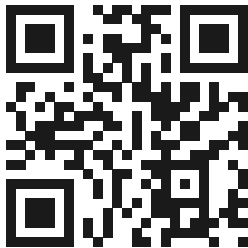
Lösung

- $L_1 - L_2$: $F_1 \times (Q_2 - F_2) = \{(q_0, p_1), (q_0, p_2)\}$
- $L_2 - L_1$: $(Q_1 - F_1) \times F_2 = \{(q_1, p_0), (q_1, p_3)\}$

→ der Automat bleibt gleich, nur F ändert sich

Kahoot

Kahoot!



<https://kahoot.it>

Aufgaben

Aufgaben



Aufgabe 1:

Entwerfen Sie EA über $\Sigma = \{a, b\}$ für $\emptyset$, $\{\lambda\}$ und Σ^+ .



Aufgabe 2:

Entwerfen Sie einen EA für $L = \{w \in \Sigma_{\text{bool}}^* \mid \text{Nummer}(w) \equiv 0 \pmod{3}\}$.

Tipp: $\text{Nummer}(wb) = 2 \cdot \text{Nummer}(w) + b$.



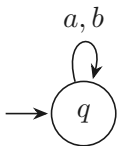
Aufgabe 3:

Entwerfen Sie modular einen EA für

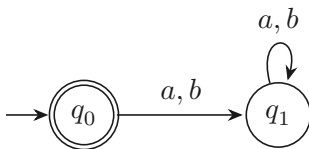
$\{w \in \{a, b\}^* \mid |w|_a \text{ gerade und } w \text{ endet mit } b\}$.

Lösung Aufgabe 1

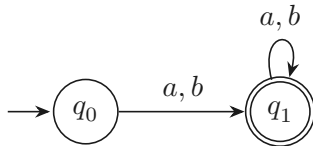
$\emptyset$:



$\{\lambda\}$:

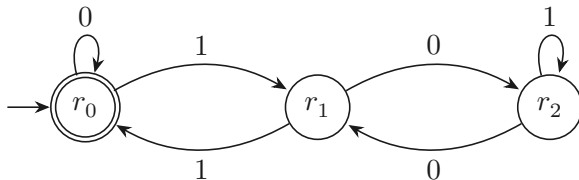


Σ^+ :



! auch $\emptyset$ braucht einen Zustand, denn $q_0 \in Q$

Lösung Aufgabe 2



$$\text{Kl}[r_i] = \{w \mid \text{Nummer}(w) \bmod 3 = i\}, \quad \delta(r_i, b) = r_{(2i+b) \bmod 3}$$

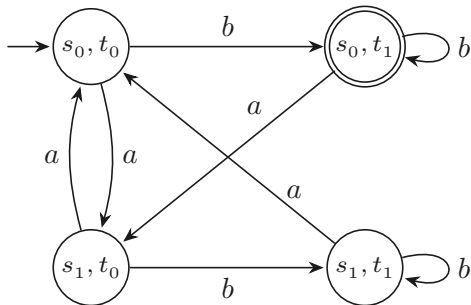
„Man merkt sich nur den Rest.“

Lösung Aufgabe 3

M_1 : $s_0 = |w|_a$ gerade, $s_1 =$ ungerade

M_2 : $t_1 =$ endet mit b , $t_0 =$ sonst

$$F = \{(s_0, t_1)\}$$



Noch zwei Aufgaben



Aufgabe 4:

Beweisen Sie für den EA aus Aufgabe 2 per Induktion über $|w|$, dass

$$\hat{\delta}(r_0, w) = r_{\text{Nummer}(w) \bmod 3}.$$



Aufgabe 5:

Wie viele Zustände braucht man für $\text{Nummer}(w) \equiv 0 \pmod{4}$?

Lösung Aufgabe 4

Es gilt $\text{Nummer}(zb) = 2 \cdot \text{Nummer}(z) + b$. Sei $N(w) = \text{Nummer}(w) \bmod 3$.

Induktionsanfang. $\hat{\delta}(r_0, \lambda) = r_0$ und $\text{Nummer}(\lambda) = 0$.

Lösung Aufgabe 4

Es gilt $\text{Nummer}(zb) = 2 \cdot \text{Nummer}(z) + b$. Sei $N(w) = \text{Nummer}(w) \bmod 3$.

Induktionsanfang. $\hat{\delta}(r_0, \lambda) = r_0$ und $\text{Nummer}(\lambda) = 0$.

Induktionsschritt. Sei $w = zb$. Dann ist

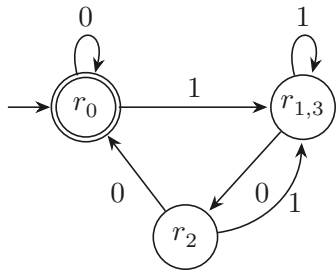
$$\hat{\delta}(r_0, zb) = \delta(\hat{\delta}(r_0, z), b) = \delta(r_{N(z)}, b) = r_{(2N(z)+b) \bmod 3} = r_{N(zb)}. \quad \square$$

Lösung Aufgabe 5

Mit Resten mod 4 bräuchte man 4 Zustände.

Aber r_1 und r_3 haben dieselben Nachfolger und sind beide nicht akzeptierend. Also reichen 3.

„Nur die letzten zwei Bits zählen.“



Bis nächste Woche!