Grundzüge der Theoretischen Informatik

20. Oktober 2021

Markus Bläser

Universität des Saarlandes

Modalitäten

CMS:

- `cms.sic.saarland/ti2122`
- Anmeldung bis Freitag, 22.10.2021, 23:59 Uhr

Vorlesung:

- Mittwochs 14:15–16:00, Freitags 8:30–10:00
- Vorlesungen Hybrid (GHH + Zoom)
- Aufnahmen werden einige Tage später zur Verfügung gestellt.
- Skript via CMS (in Englisch)

Modalitäten (2)

Übungen:

- Jede Woche Freitags ein Übungsblatt
- Abgabe Freitags 8:00 elektronisch via CMS
- Gruppenabgabe bis zu 3 Personen möglich, Grupenbildung bis zum 27.10 im CMS
- CMS-Forum/Discord: Abgabepartner suchen
- Wöchentliche Tutorien Dienstags in Präsenz oder auf Zoom
- Office Hour wöchentlich Donnerstags 14:15-15:00 via Discord oder in Präsenz

Klausur:

- Zulassung: 50% der Punkte in den Übungsblätter
- Vorläufige Termine: 7. März 2022 und 7. April 2022

Wörter (Kapitel A.3)

$\Sigma = \{a, b, c, \dots, z\}$

$\Sigma = \{a, b\}$

$\Sigma = \{0, 1\}$

- Endliche Menge Σ (*Alphabet*)
- Elemente in Σ heißen *Zeichen* oder Symbole
- Ein (endliches) *Wort* über Σ ist eine Funktion

  f / $w : \{1, \dots, \ell\} \to \Sigma$
- ℓ ist die Länge von w
- Es gibt ein Wort der Länge 0, das *leere Wort* ε

abab

$f(1) = a = f(3)$

$f(2) = b = f(4)$

$\emptyset \to \Sigma$

Wörter (2)

w	1	2	3	4
	a	b	a	b

$abab$

Wir schreiben ein Wort $w : \{1, \ldots, \ell\} \to \Sigma$ als

- Tabelle,
- kompakt als $w(1)w(2)\ldots w(\ell)$,
- kompakter als $w_1 w_2 \ldots w_\ell$

Wörter (3)

$(a, b, a, b) \in \Sigma^4$

n-Tupel

- Σ^n bezeichnet alle Wörter der Länge n
- $\Sigma^* := \underline{\bigcup_{n=0}^{\infty} \Sigma^n}$ bezeichnet alle endlichen Wörter
- $\Sigma^0 = \{\varepsilon\}$
- Wir identifizieren Σ mit Σ^1

$abab \in \Sigma^*$

$aaa \ldots\ldots \notin \Sigma^*$

KEIN Wort

$\Sigma^1 \cup \Sigma^2 \cup \ldots\ldots$

Wörter (4)

Konkatenation:

- $w : \{1, \ldots, \ell\} \to \Sigma$, $x : \{1, \ldots, k\} \to \Sigma$.
- Die Konkatenation von w und x ist die Funktion

$$
\begin{aligned}
&\{1, \ldots, \ell + k\} \to \Sigma \\
&\qquad i \mapsto \begin{cases} w(i) & \text{if } 1 \leq i \leq \ell \\ x(i \underline{- \ell}) & \text{if } \ell + 1 \leq i \leq \ell + k. \end{cases}
\end{aligned}
$$

- Wir bezeichnen die Konkatenation mit wx.
- w^i bezeichnet die i-fache Konkatenation von w mit sich selbst
- $\varepsilon w = w \varepsilon = w$. $\quad w^0 = \varepsilon \quad w^3 = (w\,w)\,w = w\,(w\,w)$

$\varepsilon\varepsilon = \varepsilon$ $\qquad = w\,w\,w$

Wörter (5)

$x = bb$
$y = abba$ $\Rightarrow$ $u = a$, $v = a$

Teilwörter:

- x heißt *Teilwort* von y falls es Wörter u und v gibt mit $y = uxv$.
- Falls $u = \varepsilon$, dann heißt x *Präfix*. $y = xv$
- Falls $v = \varepsilon$, dann heißt x *Suffix* $y = wx$

Kapitel 1: Endliche Automaten

Endliche Automaten

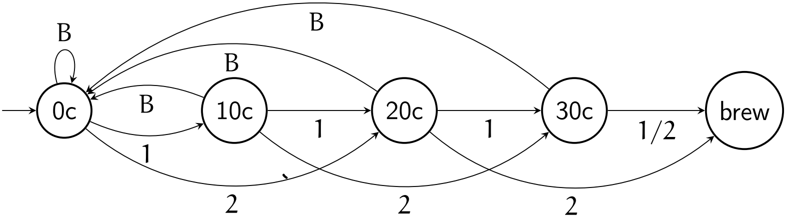
Beispiel: Kaffeeautomat

- Einwurf von 10 Cent- und 20 Cent-Münzen.
- Wurden 40 Cent oder mehr eingeworfen, so gibt die Maschine einen Kaffee aus.
- Zuviel bezahltes Geld wird einbehalten.
- Solange weniger als 40 Cent eingeworfen wurden, kann die Geld-zurück-Taste gedrückt werden.

.

1 = 10ct einwerfen   2 = 20ct einwerfen

B = Geld zurück

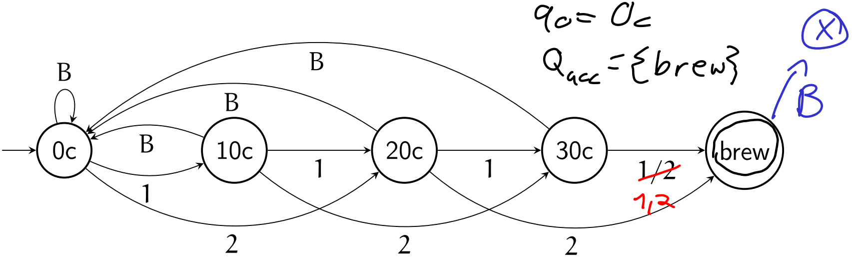


1 B 1 B 1 B 1 B 1 B 2 1 B 2 2

$Q = \{0c, 10c, 20c, 30c, brew\}$ $\Sigma = \{1, 2, B\}$

$q_0 = 0c$

$Q_{acc} = \{brew\}$



1 1 2 $0c \xrightarrow{1} 10c \xrightarrow{1} 20c \xrightarrow{2} brew$

δ	1	2	B
0c	10c	20c	0c
10c	20c	30c	0c
20c			0c
30c			0c
brew	—	—	—

112B

Endliche Automaten

Definition (1.1)

Ein endlicher Automat wird durch ein 5-Tupel $(Q, \Sigma, \delta, q_0, Q_{acc})$ beschrieben:

1. Q ist eine endliche Menge, die *Zustandsmenge*.
2. Σ ist eine endliche Menge, das *Eingabealphabet*
3. $\delta : Q \times \Sigma \to Q$ ist die *Übergangsfunktion*. → partielle Fkt. ⇕ totale Fkt ← Standard
4. $q_0 \in Q$ ist der *Startzustand*.
5. $Q_{acc} \subseteq Q$ ist die Menge der *akzeptierenden Zustände*.

Berechnungen

Definition (1.2)

Sei $M = (Q, \Sigma, \delta, q_0, Q_{\mathrm{acc}})$ ein endlicher Automat und $w \in \Sigma^*$, $|w| = n$.

1. Eine Folge $s_0, \dots, s_n \in Q$ heißt eine *Berechnung* von M auf w, falls
   1.1 $s_0 = q_0$,
   1.2 für alle $0 \leq \nu < n$: $\delta(s_\nu, w_{\nu+1}) = s_{\nu+1}$,
2. Die Berechnung heißt *akzeptierend*, falls zusätzlich $s_n \in Q_{\mathrm{acc}}$ gilt. Sonst heißt die Berechnung *verwerfend*.

Reguläre Sprachen

Kaffeeaut. akz. 22
verw. 1
22B

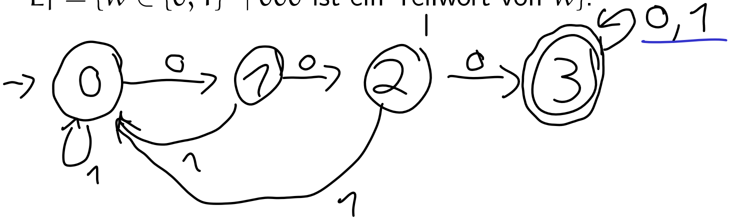
Definition (1.4)

1. Ein endlicher Automat $M = (Q, \Sigma, \delta, q_0, Q_{acc})$ *akzeptiert* ein Wort $w \in \Sigma^*$, wenn es eine akzeptierende Berechnung von M auf w gibt. Sonst verwirft M w.
2. $L(M) = \{w \in \Sigma^* \mid M \text{ akzeptiert } w\}$ ist die von M *erkannte Sprache*.
3. $A \subseteq \Sigma^*$ ist eine *reguläre Sprache*, falls es ein M gibt mit $A = L(M)$.
4. REG bezeichnet die Menge aller regulären Sprachen.

über beliebigen endl. Alphabeten.

Wie entwirft man einen endlichen Automat?

$L_1 = \{w \in \{0,1\}^* \mid 000 \text{ ist ein Teilwort von } w\}$.



$$\delta(3,0) = \delta(3,1) = 3$$

Erweiterte Übergangsfunktion

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

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

$$\delta^*(q, \varepsilon) = q$$

$$\delta^*(q, v\sigma) = \begin{cases} \delta(\underbrace{\delta^*(q, v)}, \sigma) & \text{falls } \delta^*(q, v) \text{ definiert ist} \\ \text{undefiniert} & \text{sonst} \end{cases}$$

$\sigma \in \Sigma$

Beobachtung; M akzeptiert $w \iff \delta^*(q_0, w) \in Q_{acc}$

Lemma (1.5)

Für alle $q \in Q$ *und* $x, y \in \Sigma^*$ *gilt*

$$\delta^*(q, xy) = \delta^*(\delta^*(q, x), y),$$

sofern $\delta^*(q, xy)$ *definiert ist.* $\Rightarrow \delta^*(q, x_1 x_2 \ldots x_e) = \delta(\delta \cdots \delta(q, x_1), x_2) \ldots x_e)$

$$\delta^*(q, xy) = \delta^*(\delta^*(q, x), y)$$

Induktion über $|y|$:

IA: $y = \varepsilon$ $\quad \delta^*(q, x\varepsilon) = \delta^*(q, x) = \delta^*(\delta^*(q, x), \varepsilon)$

IS $y = y'\sigma$ $\quad \delta^*(q, xy'\sigma) = \delta(\delta^*(q, xy'), \sigma)$

$$= \delta(\delta^*(\delta^*(q, x), y'), \sigma)$$

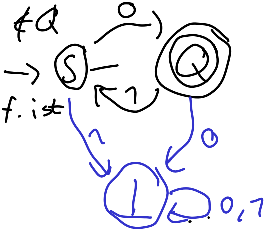
$$= \delta^*(\delta^*(q, x), \underbrace{y'\sigma}_{=y}) \qquad \square$$

Totale Übergangsfunktionen

Lemma (1.6)

Sei $M = (Q, \Sigma, \delta, q_0, Q_{acc})$ *ein endlicher Automat. Dann gibt es einen Automaten* $M' = (Q', \Sigma, \delta', q_0, Q_{acc})$, *so dass* δ' *total ist und* $L(M) = L(M')$.

$Q' = Q \cup \{\text{dead-lock}\}$, $\text{dead-lock} \notin Q$



$q \in Q$, $\sigma \in \Sigma$: $\delta'(q,\sigma) = \begin{cases} \delta(q,\sigma) & \text{falls } \delta(q,\sigma) \text{ def. ist} \\ \text{dead-lock} & \text{sonst} \end{cases}$

$\delta'(\text{dead-lock}, \sigma) = \text{dead lock}.$

akz. Ber. in M → akz. Ber. in M'

verwerf. Ber. in M → verwerf. Ber in M'

kein Ber in M → verwerf. Ber. im M' (durch dead-lock)

$L(M) = L(M')$ $\square$

Produktautomat

$x = 00000$ $\to A$ $\to S'$

Lemma (1.7)

Seien $M_1 = (Q_1, \Sigma, \delta_1, q_{0,1}, Q_{\mathrm{acc},1})$ *und* $M_2 = (Q_2, \Sigma, \delta_2, q_{0,2}, Q_{\mathrm{acc},2})$ *zwei endliche Automaten, so dass* δ_1 *und* δ_2 *total sind. Die Funktion* Δ *definiert durch*

$Q' = Q_1 \times Q_2$

$$\begin{array}{rrcl} \Delta : & \underline{(Q_1 \times Q_2)} \times \Sigma & \to & \underline{Q_1 \times Q_2} \\ & ((q_1, q_2), \sigma) & \mapsto & (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma)) \end{array}$$

erfüllt

$$\Delta^*((q_1, q_2), w) = (\delta_1^*(q_1, w), \delta_2^*(q_2, w))$$

für alle $q_1 \in Q_1$, $q_2 \in Q_2$ *und* $w \in \Sigma^*$.

$$\Delta : ((q_1, q_2), \sigma) \longmapsto (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma))$$

$$(*) \quad \Delta^*((q_1, q_2), w) = (\delta_1^*(q_1, w), \delta_2^*(q_2, w))$$

Beweis per Ind. über $|w|$

IA: $w = \varepsilon$ $\quad \Delta^*((q_1, q_2), \varepsilon) = (q_1, q_2) = (\delta_1^*(q_1, \varepsilon), \delta_2^*(q_2, \varepsilon))$

IS: $w = w'\sigma$ $\quad \Delta^*((q_1, q_2), w) = \Delta(\Delta^*((q_1, q_2), w'), \sigma)$

$$\overset{(*)}{=} \Delta((\delta_1^*(q_1, w'), \delta_2^*(q_2, w')), \sigma)$$

$$= (\underbrace{\delta_1(\delta_1^*(q_1, w'), \sigma)}_{\delta_1^*(q_1, w)}, \underbrace{\delta_2(\delta_2^*(q_2, w'), \sigma)}_{\delta_2^*(q_2, w)}) \quad \square$$

Abschlusseigenschaften

Theorem (1.8)

REG *ist abgeschlossen unter Schnitt, Vereinigung und Mengendifferenz, d.h. sind* $A, B \subseteq \Sigma^*$ *regulär, so auch* $A \cap B$, $A \cup B$ *und* $A \setminus B$.