

## Einführung in die Theoretische Informatik

### Hausaufgabe 1 (5 Punkte)

Wir betrachten das folgende einfache Modell eines Aufzugs in einem Gebäude mit 3 Stockwerken, nummeriert mit 0, 1, 2. Der Aufzug kann folgende Aktionen durchführen:

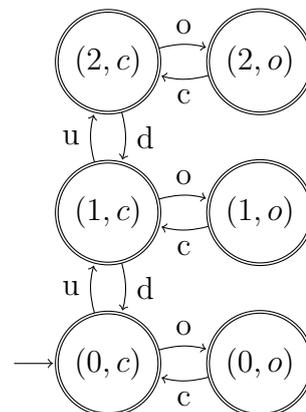
- (o) Tür öffnen (open)
- (c) Tür schließen (close)
- (u) Ein Stockwerk nach oben fahren (up)
- (d) Ein Stockwerk nach unten fahren (down)

Dabei gelten die Einschränkungen, dass der Aufzug niemals mit offener Tür fährt, und natürlich auch nicht in Stockwerke fährt, die nicht existieren.

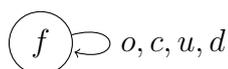
1. Modellieren Sie den Aufzug als NFA  $N$  über  $\Sigma = \{o, c, u, d\}$ , so dass  $L(N)$  genau die Wörter enthält, die eine zulässige Aktionsfolge des Aufzugs darstellen.
2. Was ändert sich in diesem Fall, wenn man den NFA in einen DFA umwandelt?

### Lösungsvorschlag

1. Der Zustandsraum des Aufzugs besteht aus den Stockwerken und dem Zustand der Tür (offen oder geschlossen):  $Q = \{0, 1, 2\} \times \{o, c\}$ . Wir modellieren die Aktionen als Zustandsübergänge in der Grafik rechts. In diesem Fall ist  $F = Q$ , da nach Aufgabenstellung eine Aktionsfolge jederzeit enden darf. Wird eine unzulässige Eingabe gelesen (z.B.  $ou$ ), gibt es an der entsprechenden Stelle keine Übergang.



2. Wir erhalten einen DFA, indem wir einen neuen Zustand



einführen, der als Fehlerzustand fungiert. Alle fehlenden Übergänge werden ergänzt und führen in diesen neuen Zustand. Genau dieses Resultat erhält man auch, wenn man die Potenzmengenkonstruktion auf den NFA anwendet. Der Fehlerzustand entspricht dann der leeren Zustandsmenge.

## Hausaufgabe 2 (5 Punkte)

Das Reißverschlussprodukt von zwei Sprachen  $L_1$  und  $L_2$  ist wie folgt definiert:

$$L_1 \% L_2 = \{a_1 b_1 a_2 b_2 \dots a_n b_n \mid a_i \in \Sigma, b_i \in \Sigma, a_1 \dots a_n \in L_1, b_1 \dots b_n \in L_2\}$$

1. Geben Sie einen regulären Ausdruck an, der  $L(aba^*) \% L(bc^*)$  beschreibt.
2. Vergleichen Sie das Reißverschlussprodukt mit dem Shuffle-Produkt aus TA2 von Blatt 2, und finden Sie zwei Sprachen  $A$  und  $B$  und ein Wort, das in  $(A \parallel B)$ , aber nicht in  $A \% B$  liegt.
3. Zeigen Sie durch eine geeignete Automatenkonstruktion: Wenn  $L_1$  und  $L_2$  regulär sind, dann ist auch  $L_1 \% L_2$  regulär.

Beschreiben Sie ihre Konstruktion zunächst informell.

## Lösungsvorschlag

1.  $abbc(ac)^*$
2. Ein einfaches Beispiel ist  $A = \{a, aa\}$  und  $B = \{b\}$ . Dann ist  $A \parallel B = \{ba, ab, baa, aba, aab\}$ , dagegen ist  $A \% B = \{ab\}$ . Im Allgemeinen gilt stets  $A \% B \subseteq A \parallel B$ .
3. Für die Konstruktion bildet man das Produkt der beiden Automaten mit einem zusätzlichen Flag, welches angibt, welcher der beiden Automaten gerade "an der Reihe" ist. Wir nehmen also DFA's  $A_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$  und  $A_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$  an und konstruieren daraus den DFA

$$A' = (Q_1 \times Q_2 \times \{1, 2\}, \Sigma, \delta', (q_{01}, q_{02}, 1), F_1 \times F_2 \times \{1\})$$

Die Übergangsfunktion  $\delta'$  ist definiert mit

$$\begin{aligned} \delta'((q, q', 1), a) &= (\delta(q, a), q', 2) & \forall q \in Q_1, q' \in Q_2, a \in \Sigma \\ \delta'((q, q', 2), a) &= (q, \delta(q', a), 1) & \forall q \in Q_1, q' \in Q_2, a \in \Sigma \end{aligned}$$

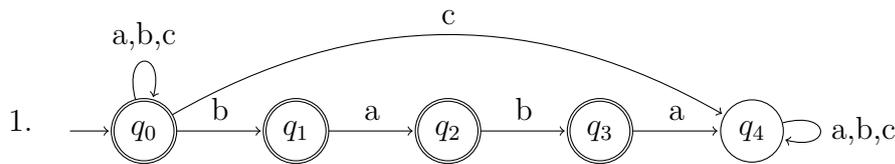
## Hausaufgabe 3 (5 Punkte)

Ein  $\forall$ -NFA ("all-NFA") ist wie ein NFA als 5-Tupel  $N = (Q, \Sigma, \delta, q_0, F)$  definiert. Allerdings ist die akzeptierte Sprache anders definiert:

$$L_{\forall}(N) = \{w \in \Sigma^* \mid \hat{\delta}(\{q_0\}, w) \subseteq F\}$$

1. Sei  $\Sigma = \{a, b, c\}$ . Konstruieren Sie einen  $\forall$ -NFA  $N$ , so dass  $L_{\forall}(N)$  genau die Menge aller Wörter aus  $\Sigma^*$  ist, die weder ein  $c$  noch das Teilwort  $baba$  enthalten.
2. Zeigen Sie, dass man aus jedem  $\forall$ -NFA  $N$  einen NFA  $N'$  konstruieren kann, so dass  $L(N') = \Sigma^* \setminus L_{\forall}(N)$ .  
Vorgehensweise: Beschreiben Sie die Konstruktion zunächst informell, dann formal als Tupel, und beweisen Sie dann die Korrektheit Ihrer Konstruktion.
3. Modifizieren Sie die Potenzmengenkonstruktion geeignet, so dass man damit einen  $\forall$ -NFA in einen äquivalenten DFA umwandeln kann.

## Lösungsvorschlag



2. Es genügt hier, die Endzustände zu Nicht-Endzuständen zu machen und umgekehrt. Sei also  $N' = (Q, \Sigma, \delta, q_0, Q \setminus F)$ . Nun gilt für beliebiges  $w \in \Sigma^*$ :

$$w \in L(N') \iff \hat{\delta}(\{q_0\}, w) \cap (Q \setminus F) \neq \emptyset \iff \hat{\delta}(\{q_0\}, w) \not\subseteq F \iff w \notin L_{\forall}(N)$$

Damit ist  $L(N') = \Sigma^* \setminus L_{\forall}(N)$ .

3. Der zu einem  $\forall$ -NFA  $N = (Q, \Sigma, \delta, q_0, F)$  äquivalente DFA ist  $M = (\mathcal{P}(Q), \Sigma, \bar{\delta}, q_0, \mathcal{P}(F))$ . Auch hier verändert sich im Vergleich zur normalen Potenzmengenkonstruktion nur die Endzustandsmenge. Analog zum Beweis von Satz 2.8 kann man auch hier die Korrektheit leicht zeigen:

$$w \in L_{\forall}(N) \iff \hat{\delta}(\{q_0\}, w) \subseteq F \iff \hat{\delta}(\{q_0\}, w) \in \mathcal{P}(F) \iff w \in L(M)$$

## Hausaufgabe 4 (5 Punkte)

Die Zustandsübergangsfunktion eines NFA  $A = (Q, \Sigma, \delta, q_0, \{q_1\})$  sei gegeben durch

$q_i$	$\delta(q_i, 0)$	$\delta(q_i, 1)$
$q_0$	$\emptyset$	$\{q_1, q_2\}$
$q_1$	$\{q_2\}$	$\{q_1, q_3\}$
$q_2$	$\{q_2, q_3\}$	$\{q_2\}$
$q_3$	$\{q_0\}$	$\{q_1\}$

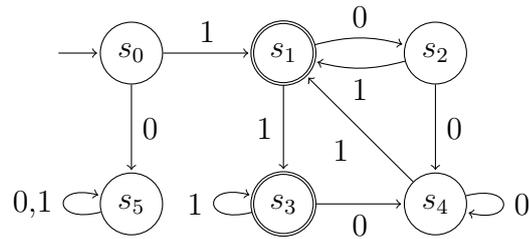
- Konstruieren Sie mit dem Potenzmengenverfahren einen DFA  $B$ , der  $L(A)$  akzeptiert.
- Geben Sie nun einen DFA  $C$  an, der  $\overline{L(A)} = \Sigma^* \setminus L(A)$  akzeptiert.

## Lösungsvorschlag

- Anwendung des Potenzmengenverfahrens. Wir schreiben kurz  $q_i q_j$  für  $\{q_i, q_j\}$  etc.

$q_i$	$\delta(q_i, 0)$	$\delta(q_i, 1)$	neuer Name
$q_0$	$\emptyset$	$q_1 q_2$	$s_0$
$q_1 q_2$	$q_2 q_3$	$q_1 q_2 q_3$	$s_1$
$q_2 q_3$	$q_0 q_2 q_3$	$q_1 q_2$	$s_2$
$q_1 q_2 q_3$	$q_0 q_2 q_3$	$q_1 q_2 q_3$	$s_3$
$q_0 q_2 q_3$	$q_0 q_2 q_3$	$q_1 q_2$	$s_4$
$\emptyset$	$\emptyset$	$\emptyset$	$s_5$

Damit ergibt sich  $B = (\{s_0, s_1, s_2, s_3, s_4, s_5\}, \{0, 1\}, \delta_B, s_0, \{s_1, s_3\})$  mit



2.  $C$  unterscheidet sich von  $B$  nur durch die Menge der Endzustände, die invertiert wird. Damit ist  $C = (\{s_0, s_1, s_2, s_3, s_4, s_5\}, \{0, 1\}, \delta_B, s_0, \{s_0, s_2, s_4, s_5\})$ .

Man beachte, dass die Komplementbildung durch invertieren der Endzustände nur beim DFA funktioniert, nicht aber beim NFA.

## Vorbereitung 1

Was lässt sich über die Menge der Pumping-Lemma-Zahlen einer endlichen Sprache  $L \subseteq \Sigma^*$  aussagen, wenn nur  $|\Sigma| = n$  bekannt ist?

### Lösungsvorschlag

Die Menge der Pumping-Lemma-Zahlen einer regulären Sprache ist stets unendlich und, wenn  $n$  eine Pumping-Lemma-Zahl ist, dann ist  $n + 1$  ebenfalls eine Pumping-Lemma-Zahl. Demnach lässt sich sicherlich die Menge der natürlichen Zahlen in zwei Abschnitte  $[n]$  und  $\mathbb{N} \setminus [n]$  so zerlegen, dass  $\mathbb{N} \setminus [n]$  genau die Menge aller Pumping-Lemma-Zahlen einer regulären Sprache ist.

Man sieht leicht, dass bei einer endlichen Sprache  $L$  die Pumping-Lemma-Zahl eine obere Schranke an die Wortlänge ist: Jedes längere Wort könnte man mit der üblichen Methode “aufpumpen” und daraus unendlich viele Wörter in  $L$  erhalten – ein Widerspruch.

Bei festem Alphabet  $\Sigma$  erhält man daraus auch eine obere Schranke für  $|L|$ , da es maximal  $\sum_{i=0}^{n-1} |\Sigma|^i$  Wörter gibt, die kürzer sind als  $n$ .

Ist das Alphabet nicht fest vorgegeben, dann ist die Anzahl der Elemente einer nichtleeren regulären Sprache fast völlig unabhängig von ihrer Pumping-Lemma-Zahl, denn es gibt zu jedem Paar natürlicher Zahlen  $(n, m)$  mit  $n > 1$  und  $m > 1$  eine reguläre Sprache mit  $m$  Elementen, deren kleinste Pumping-Lemma-Zahl gleich  $n$  ist.

Dies zeigt man durch Konstruktion von endlichen Sprachen mit unterschiedlichen Wortlängen. Es gibt zu jedem  $n$  eine Sprache  $L$  mit  $|L| = n$ , in denen alle Wortlängen gleich 1 sind, demzufolge 2 eine Pumping-Lemma-Zahl zu  $L$  ist. Man benötigt dazu  $|\Sigma| = n$ . Umgekehrt gibt es einelementige Sprachen mit Wortlänge gleich  $n$  und folglich Pumping-Lemma-Zahl  $n + 1$ .

## Vorbereitung 2

Seien  $A, B$  und  $X$  Sprachen mit  $\epsilon \notin A$ . Beweisen oder widerlegen Sie die Umkehrung von Arden’s Lemma, d. h.  $X = A^*B \implies X = AX \cup B$ .

### Lösungsvorschlag

Wir beweisen die Umkehrung von Arden’s Lemma.

Sei  $X = A^*B$ . Im Folgenden benutzen wir bereits bewiesene bzw. elementare Eigenschaften von Mengenoperationen.

$$\begin{aligned} AX \cup B &= A(A^*B) \cup B \\ &= (AA^*)B \cup \{\epsilon\}B \\ &= (AA^* \cup \{\epsilon\})B \\ &= A^*B \\ &= X. \end{aligned}$$

## Tutoraufgabe 1

Sei  $\Sigma$  ein Alphabet,  $a \in \Sigma$ , und  $L \subseteq \Sigma^*$  eine Sprache über  $\Sigma$ . Wir definieren die Sprachen  $\min(L)$  und  $\text{quot}(a, L)$  wie folgt:

$$\begin{aligned} \min(L) &= \{w \in L \mid \text{kein echtes Präfix von } w \text{ liegt in } L\} \\ \text{quot}(a, L) &= \{w \in \Sigma^* \mid aw \in L\} \end{aligned}$$

Zeigen Sie, dass  $\min(L)$  und  $\text{quot}(a, L)$  regulär sind, wenn  $L$  regulär ist.

## Lösungsvorschlag

1. Wir nehmen an, dass  $M = (Q, \Sigma, \delta, q_0, F)$  ein DFA ist, der die Sprache  $L$  akzeptiert und konstruieren daraus einen DFA  $M'$ , der  $\min(L)$  akzeptiert. Dabei ist die Idee, dass man die Wörter aus der akzeptierten Sprache entfernt, von denen bereits ein echtes Präfix in  $L$  liegt. In den Automaten übersetzt bedeutet das, dass aus einem Endzustand kein weiterer Endzustand mehr erreichbar sein darf. Das erreicht man am einfachsten, indem man einen Fehlerzustand  $f$  einführt, und von jedem Endzustand alle Übergänge in den neuen Fehlerzustand wechseln lässt.

Formal ist  $M' = (Q \cup \{f\}, \Sigma, \delta', q_0, F)$  mit

$$\delta'(q, a) = \begin{cases} \delta(q, a) & \text{falls } q \in Q \setminus F, \\ f & \text{falls } q \in F \cup \{f\}. \end{cases}$$

2. Wir nehmen wieder einen DFA  $M = (Q, \Sigma, \delta, q_0, F)$  mit  $L(M) = L$  an. Um einen DFA für  $\text{quot}(a, L)$ , zu erhalten, genügt es, den Startzustand zu ändern:

$$M' = (Q, \Sigma, \delta, \delta(q_0, a), F).$$

Die Korrektheit der Konstruktion ist leicht zu zeigen: Für alle  $w \in \Sigma^*$  gilt.

$$w \in L(M') \iff \hat{\delta}(\delta(q_0, a), w) \in F \iff \hat{\delta}(q_0, aw) \in F \iff aw \in L(M).$$

Damit ist  $L(M') = \{w \in \Sigma^* \mid aw \in L\} = \text{quot}(a, L)$ .

## Tutoraufgabe 2

1. Sei  $\Sigma = \{a, b, c, d, e\}$ .  
Zeigen Sie, dass die Sprache  $L = \{ab^{2^i}cd^ie \mid i \in \mathbb{N}\}$  nicht regulär ist.
2. Sei  $\Sigma = \{1\}$ . Zeigen Sie, dass die Sprache  $P = \{1^p \mid p \text{ prim}\}$  nicht regulär ist.

## Lösungsvorschlag

1. Wir nehmen an, dass  $L$  regulär ist, und leiten wie folgt einen Widerspruch her.

Zunächst folgt für eine Pumping-Lemma Zahl  $n \in \mathbb{N}$  mit  $n > 0$  für alle  $z \in L$  mit  $|z| \geq n$  die Eigenschaft

$$\exists u, v, w \in \Sigma^* : \begin{array}{l} 1. \quad z = uvw, \\ 2. \quad v \neq \epsilon, \\ 3. \quad |uv| \leq n, \\ 4. \quad uv^i w \in L, \forall i \geq 0. \end{array}$$

Dass diese Eigenschaft nicht für alle  $z \in L$  mit  $|z| \geq n$  gelten kann, zeigt man z. B. mit dem Wort  $z = ab^{2^n}cd^ne$ .

Für dieses Wort  $z$  ist zunächst klar, dass  $|z| \geq n$ . Und es ist klar, dass  $z$  in  $L$  enthalten ist. Deshalb kann man eine Zerlegbarkeit von  $z$  annehmen, so dass Eigenschaften 1. bis 4. gelten.

Sei also  $z = uvw$  mit  $v \neq \epsilon$ ,  $|uv| \leq n$  und  $uv^i w \in L, \forall i \geq 0$ .

Wegen  $|uv| \leq n$  kann  $v$  kein  $c$ ,  $d$  oder  $e$  enthalten. Dann muss also  $v$  aus  $a$ 's und/oder  $b$ 's bestehen, und, da  $v$  nicht leer ist, folglich mit  $a$  oder  $b$  enden. Beide Fälle führen wir wie folgt zum Widerspruch.

Im Folgenden bezeichnen wir mit  $|w|_x$  die Anzahl der Buchstaben  $x$ , die in  $w$  enthalten sind.

Falls  $v$  mit  $a$  endet, dann gilt  $|uv^2 w|_a > 1$ . Daraus folgt  $uv^2 w \notin L$  im Widerspruch zu Eigenschaft 4. für  $i = 2$ .

Falls  $v$  mit  $b$  endet, dann gilt  $|uv^0 w|_b < 2n$  und  $|uv^0 w|_d = n$ .

Daraus folgt ebenfalls  $uv^0 w \notin L$  im Widerspruch zu Eigenschaft 4. für  $i = 0$ .

Man beachte, dass im Beweis alle Eigenschaften 1. bis 4. verwendet wurden.

2. Wir nehmen an, dass  $L = \{1^p \mid p \text{ prim}\}$  regulär ist. Sei  $n$  eine Pumping-Lemma-Zahl.

Da die Menge der Primzahlen bekanntlich unendlich ist, gibt es eine Primzahl  $p$  bzw. ein  $z = 1^p \in L$ , so dass  $z$  mindestens  $n$  Zeichen enthält, also  $|z| \geq n$ .

Sei also  $z \in L$  mit  $|z| \geq n$  und sei  $z = uvw$  mit  $|uv| \leq n$ ,  $v$  nicht leer und  $uv^i w \in L$  für alle  $i \geq 0$ .

Wegen  $uw = uv^0 w \in L$  ist  $|uw|$  eine Primzahl.

Wir setzen  $x = uv^{|uw|} w$ . Es gilt  $x \in L$ , d. h.  $|x|$  ist eine Primzahl.

Es folgt aber auch  $|x| = |uw| + |uw||v| = |uw|(1 + |v|)$ , d. h.  $|x|$  ist wegen  $|v| \neq 0$  keine Primzahl. Widerspruch!

### Tutoraufgabe 3

Wir betrachten den DFA  $A = (Q, \Sigma, \delta, q_0, \{q_1, q_2\})$ , dessen Zustandsübergangsfunktion  $\delta$  gegeben ist durch

$q_i$	$\delta(q_i, 0)$	$\delta(q_i, 1)$
$q_0$	$q_1$	$q_2$
$q_1$	$q_0$	$q_1$
$q_2$	$q_1$	$q_0$

Stellen Sie zur Berechnung der Sprachen  $L_i = \{w \in \Sigma^* \mid \hat{\delta}(q_i, w) \in \{q_1, q_2\}\}$  ein Gleichungssystem mit entsprechenden Unbekannten  $X_i$  für die Sprachen  $L_i$  auf und berechnen Sie mit Hilfe von Ardens Lemma alle  $X_i$  als reguläre Ausdrücke.

## Lösungsvorschlag

Aufstellen des Gleichungssystems:

$$X_0 \equiv 0X_1 \mid 1X_2 \quad (1)$$

$$X_1 \equiv 0X_0 \mid 1X_1 \mid \epsilon \quad (2)$$

$$X_2 \equiv 0X_1 \mid 1X_0 \mid \epsilon \quad (3)$$

Einsetzen von Gleichung (1) in (2, 3):

$$\begin{aligned} X_1 &\equiv 0(0X_1 \mid 1X_2) \mid 1X_1 \mid \epsilon \\ &\equiv (00 \mid 1)X_1 \mid 01X_2 \mid \epsilon \end{aligned} \quad (4)$$

$$\begin{aligned} X_2 &\equiv 0X_1 \mid 1(0X_1 \mid 1X_2) \mid \epsilon \\ &\equiv (0 \mid 10)X_1 \mid 11X_2 \mid \epsilon \end{aligned} \quad (5)$$

Auflösen von Gleichung (4) nach  $X_1$  (Ardens Lemma):

$$X_1 \equiv (00 \mid 1)^*(01X_2 \mid \epsilon) \quad (6)$$

Einsetzen in Gleichung (5):

$$\begin{aligned} X_2 &\equiv \underbrace{(0 \mid 10)(00 \mid 1)^*(01X_2 \mid \epsilon)}_{\alpha} \mid 11X_2 \mid \epsilon \\ &\equiv (\alpha 01 \mid 11)X_2 \mid \alpha \mid \epsilon \end{aligned}$$

Auflösen nach  $X_2$  (Ardens Lemma):

$$X_2 \equiv (\alpha 01 \mid 11)^*(\alpha \mid \epsilon)$$

Durch Einsetzen in die Gleichungen (6) und (1) erhalten wir schließlich  $X_1$  und  $X_0$ :

$$\begin{aligned} X_1 &\equiv (00 \mid 1)^*(01(\alpha 01 \mid 11)^*(\alpha \mid \epsilon) \mid \epsilon) \\ X_0 &\equiv 0(00 \mid 1)^*(01(\alpha 01 \mid 11)^*(\alpha \mid \epsilon) \mid \epsilon) \mid 1(\alpha 01 \mid 11)^*(\alpha \mid \epsilon) \end{aligned}$$

*Strategische Hinweise:* Um bei größer werdenden Ausdrücken nicht den Überblick zu verlieren, empfiehlt es sich, Abkürzungen für variablenfreie Teilausdrücke einzuführen, wie oben das  $\alpha$ . Das verhindert auch viele Abschreibefehler. Natürlich ist  $\alpha$  keine neue Variable, sondern nur eine Abkürzung für einen festen Ausdruck.

Gleichungen der Form  $X_i \equiv R$ , wobei  $X_i$  nicht in  $R$  vorkommt, setzt man am besten sofort in die restlichen Gleichungen ein, womit man sofort eine Variable (oben:  $X_0$ ) aus dem System eliminiert hat.