

Lösungsvorschlag zur Übung 6 zur Vorlesung Formale Sprachen und Komplexität

Wenn Sie Automaten angeben, tun Sie dies immer in Form eines Zustandsgraphen. Andere Formen der Darstellung (z.B. als Liste von Übergängen) werden nicht gewertet, da sie sehr viel aufwändiger zu korrigieren sind. Vergessen Sie nicht, im Zustandsgraph Start- und Endzustände zu markieren.

FSK6-1 Kontextfreie Grammatiken und Kellerautomaten

Sei $L = \{a^{2n}\$a^n \mid n \in \mathbb{N}_{>0}\}$ eine Sprache über dem Alphabet $\Sigma = \{a, \$\}$.

- a) Geben Sie eine kontextfreie Grammatik an, die L erkennt.

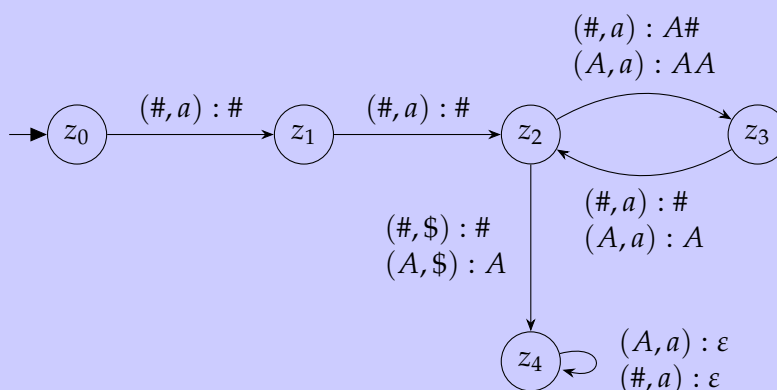
LÖSUNGSVORSCHLAG:

$(\{S\}, \Sigma, \{S \rightarrow aaSa, S \rightarrow aa\$a\}, S)$

- b) Geben Sie einen Kellerautomaten an, der L akzeptiert (mit leerem Keller oder mit Endzuständen). Erklären Sie kurz, warum Ihr Automat genau L akzeptiert.

LÖSUNGSVORSCHLAG:

Der folgende Kellerautomat akzeptiert L mit leerem Keller.



Von z_0 bis z_2 lesen wir 2 a 's, wobei der Keller unverändert nur $\#$ enthält. Die Schleife zwischen z_2 und z_3 liest $2m$ weitere a 's ein (für ein $m \in \mathbb{N}$) und legt m A 's auf den Keller. Insgesamt haben wir bisher also $2m + 2$ a 's gelesen. Von z_3 nach z_4 lesen wir ein $\$$ und lassen den Keller unverändert.

Zu diesem Zeitpunkt hat der Keller also die Form $A^m\#$. Die Schleife bei z_4 liest für jedes Symbol auf dem Keller ein a ein. Wenn sich dadurch der Keller leert und wir am Ende der Eingabe sind, haben wir weitere $m + 1$ a 's gelesen, also insgesamt das Wort $a^{2(m+1)}\$a^{m+1}$.

FSK6-2 CYK-Algorithmus

Sei G die Grammatik $(\{A_1, A_2, A_3, A_4, A_5\}, \{\$, \#\}, P, A_1)$ mit

$$P = \{A_1 \rightarrow A_3A_4 \mid A_3A_2, \\ A_2 \rightarrow A_2A_3 \mid A_4A_4, \\ A_3 \rightarrow \$, \\ A_4 \rightarrow \# \mid A_3A_4, \\ A_5 \rightarrow A_4A_4 \mid \#\}$$

- a) Prüfen Sie mit dem CYK-Algorithmus, ob $w_1 = \$\#\#\#$ in $L(G)$ ist. Erstellen Sie dazu die entsprechende Tabelle des Algorithmus und erklären Sie anhand der Tabelle, ob das Wort in $L(G)$ ist.

LÖSUNGSVORSCHLAG:

Wort:	\$	#	\$	#	#
j \ i	1	2	3	4	5
1	A_3	A_4, A_5	A_3	A_4, A_5	A_4, A_5
2	A_1, A_4		A_1, A_4	A_2, A_5	
3		A_2, A_5	A_1, A_2, A_5		
4	A_1, A_2, A_5				
5					

Da das Startsymbol A_1 nicht in Zeile 5, Spalte 1 enthalten ist, ist $\$\#\#\# \notin L(G)$.

- b) Geben Sie alle weiteren Wörter w an, für die sich aus der Tabelle ergibt, dass $w \in L(G)$ ist.

LÖSUNGSVORSCHLAG:

Alle Zellen, die das Startsymbol A_1 enthalten, stehen für ein Teilwort, das in $L(G)$ liegt. Aus der Tabelle kann man somit $\$, \$\#, \$\#\#$ und $\$\#\#\#$ ablesen.

FSK6-3 Begrenzter Keller

- a) Ein Kellerautomat mit k -begrenztem Keller ist ein Kellerautomat, bei dem maximal k Symbole auf dem Keller liegen können. Wenn der Keller voll ist, können Übergänge, die ein Symbol auf den Keller legen würden, nicht gewählt werden.

Zeigen Sie, dass für jeden Kellerautomaten K und $k \in \mathbb{N}$ gilt:

$$L(K \text{ mit } k\text{-begrenztem Keller}) \subseteq L(K)$$

Hierbei nehmen wir an, dass K mit leerem Keller akzeptiert.

LÖSUNGSVORSCHLAG:

Wenn ein Wort $w \in \Sigma^*$ von K mit k -begrenztem Keller akzeptiert wird, dann gibt es eine Folge von Übergängen in K , die den Keller leert und bei der der Keller nie größer als k wird. Dieselbe Konfigurationsfolge ist auch in K mit unbegrenztem Keller gültig und leert den Keller, also wird w auch von K akzeptiert.

- b) Zeigen Sie, dass für alle $k \in \mathbb{N}_{>0}$ die Kellerautomaten mit k -begrenztem Keller genau die regulären Sprachen beschreiben.

LÖSUNGSVORSCHLAG:

Man kann jeden endlichen Automaten als Kellerautomaten mit Endzuständen auffassen, der immer das Symbol $\#$ vom Keller nimmt und wieder auf den Keller legt. Dieser akzeptiert die gleiche Sprache. Damit ist jede reguläre Sprache durch einen 1-begrenzten Keller (und damit auch einen k -begrenzten Keller) darstellbar. Aus der Vorlesung wissen wir, dass Akzeptanz durch leeren Keller äquivalent zu Akzeptanz mit Endzuständen ist. Daher gilt die Aussage für Kellerautomaten, die mit leerem Keller akzeptieren, ebenfalls.

Umgekehrt kann ein Kellerautomat mit k -begrenztem Keller nur endlich viele verschiedene Kellerkonfigurationen (damit ist nur der Teil $W \in \Gamma^*$ aus der Konfiguration des Kellerautomaten gemeint) haben, da es nur endlich viele verschiedene Kellersymbole gibt und es eine feste obere Grenze für die Kellertiefe gibt.

Sei nun C die Menge dieser möglichen Kellerkonfigurationen und Z die Zustandsmenge des Kellerautomaten. Dann können wir einen NFA mit ε -Übergängen konstruieren, dessen Zustandsmenge $C \times Z$ ist. Der Automat hat einen Übergang von Zustand (c, z) zu (c', z') mit Symbol a (oder mit ε) genau dann wenn der Kellerautomat mit a (oder mit ε) in der Kellerkonfiguration c von z nach z' mit Kellerkonfiguration c' übergehen würde. Die Endzustände des NFA sind Zustände (c, z) , bei denen z ein Endzustand des Kellerautomaten ist. (Hier nehmen wir o.B.d.A. an, dass der Kellerautomat

mit Endzuständen akzeptiert.)

FSK6-4 Homomorphismen

Gegeben Alphabete Σ und Δ bezeichnen wir eine Abbildung $h: \Sigma^* \rightarrow \Delta^*$ als (Monoid-) Homomorphismus, wenn sie strukturerhaltend ist, d.h. wenn gilt:

$$\begin{aligned} h(\varepsilon) &= \varepsilon \\ h(u \circ v) &= h(u) \circ h(v) \quad \forall u, v \in \Sigma^* \end{aligned}$$

Für $L \subseteq \Sigma^*$ sei $h(L) = \{h(u) \mid u \in L\}$.

Man kann leicht zeigen, dass für alle Sprachen $L_1, L_2 \subseteq \Sigma^*$ und Homomorphismen h gilt:

$$\begin{aligned} h(\emptyset) &= \emptyset \\ h(L_1 \cup L_2) &= h(L_1) \cup h(L_2) \end{aligned}$$

a) Beweisen Sie für alle Sprachen $L_1, L_2 \subseteq \Sigma^*$ und Homomorphismen h :

$$\begin{aligned} h(L_1 \circ L_2) &= h(L_1) \circ h(L_2) \\ h(L_1^*) &= (h(L_1))^* \end{aligned}$$

LÖSUNGSVORSCHLAG:

$$\begin{aligned} h(L_1 \circ L_2) &= h(\{u \circ v \mid u \in L_1, v \in L_2\}) = \{h(u \circ v) \mid u \in L_1, v \in L_2\} \\ &= \{h(u) \circ h(v) \mid u \in L_1, v \in L_2\} = \{h(u) \mid u \in L_1\} \circ \{h(v) \mid v \in L_2\} \\ &= h(\{u \mid u \in L_1\}) \circ h(\{v \mid v \in L_2\}) = h(L_1) \circ h(L_2) \end{aligned}$$

$$\begin{aligned} w &\in h(L_1^*) \\ \Leftrightarrow \exists u \in L_1^*. w &= h(u) \\ \Leftrightarrow \exists n \in \mathbb{N}. \exists u_1, \dots, u_n \in L_1. w &= h(u_1 \circ \dots \circ u_n) \\ \Leftrightarrow \exists n \in \mathbb{N}. \exists u_1, \dots, u_n \in L_1. w &= h(u_1) \circ \dots \circ h(u_n) \\ \Leftrightarrow \exists n \in \mathbb{N}. \exists u'_1, \dots, u'_n \in h(L_1). w &= u'_1 \circ \dots \circ u'_n \\ \Leftrightarrow \exists u' \in (h(L_1))^*. w &= u' \\ \Leftrightarrow w \in (h(L_1))^* \end{aligned}$$

b) Beweisen Sie für alle regulären Sprachen L und Homomorphismen h : Wenn L regulär ist, dann ist auch $h(L)$ regulär.

LÖSUNGSVORSCHLAG:

Da L regulär ist, gibt es einen regulären Ausdruck α mit $L = L(\alpha)$. Wir konstruieren einen regulären Ausdruck $h(\alpha)$ mit $L(h(\alpha)) = h(L(\alpha))$. Damit folgt, dass $h(L(\alpha)) = h(L)$ regulär ist. Die Konstruktion erfolgt durch Induktion über die Länge n von α :

- $n = 0$: ist nicht möglich, weil reguläre Ausdrücke mindestens Länge 1 haben.
- $n > 0$: Es gibt folgende Möglichkeiten.

- $\alpha = \emptyset$. Wähle $h(\alpha) = \emptyset$. Dann ist $L(\alpha) = \emptyset = L(h(\alpha))$.
- $\alpha = \varepsilon$. Wähle $h(\varepsilon) = \varepsilon$. Dann ist $L(\alpha) = \{\varepsilon\} = L(h(\alpha))$.
- $\alpha = a$ (mit $a \in \Sigma$). Wähle $h(\alpha) = h(a)$. Dann ist $h(L(\alpha)) = h(\{a\}) = \{h(a)\} = L(h(\alpha))$.
- $\alpha = \alpha_1\alpha_2$. Wähle $h(\alpha) = h(\alpha_1)h(\alpha_2)$. Dann ist

$$\begin{aligned} h(L(\alpha)) &= h(L(\alpha_1\alpha_2)) = h(L(\alpha_1)L(\alpha_2)) = h(L(\alpha_1))h(L(\alpha_2)) \\ &\stackrel{\text{IH}}{=} L(h(\alpha_1))L(h(\alpha_2)) = L(h(\alpha_1)h(\alpha_2)) = L(h(\alpha)) \end{aligned}$$

- $\alpha = (\alpha_1|\alpha_2)$. Wähle $h(\alpha) = (h(\alpha_1)|h(\alpha_2))$. Dann ist

$$\begin{aligned} h(L(\alpha)) &= h(L(\alpha_1) \cup L(\alpha_2)) = h(L(\alpha_1)) \cup h(L(\alpha_2)) \\ &\stackrel{\text{IH}}{=} L(h(\alpha_1)) \cup L(h(\alpha_2)) = L(h(\alpha_1)|h(\alpha_2)) = L(h(\alpha)) \end{aligned}$$

- $\alpha = \alpha_1^*$. Wähle $h(\alpha) = (h(\alpha_1))^*$. Dann ist

$$\begin{aligned} h(L(\alpha)) &= h(L(\alpha_1^*)) = h((L(\alpha_1))^*) \stackrel{\text{IH}}{=} (h(L(\alpha_1)))^* \\ &= (L(h(\alpha_1)))^* = L((h(\alpha_1))^*) = L(h(\alpha)) \end{aligned}$$

- c) Zeigen Sie, dass über dem Alphabet $\Sigma = \{a, b, c, d\}$ die Sprache

$$L = \{a^n b^i c^n d^j \mid n, i, j \in \mathbb{N}\}$$

nicht regulär ist. Geben Sie zu diesem Zweck einen Homomorphismus h an, so dass $h(L) = \{a^n b^n \mid n \in \mathbb{N}\}$ ist. Da $h(L)$ bekanntlich nicht regulär ist, kann dann auch L nicht regulär sein.

LÖSUNGSVORSCHLAG:

Wir definieren:

$$h(a) = a$$

$$h(b) = \varepsilon$$

$$h(c) = b$$

$$h(d) = \varepsilon$$

$$h(x_1 \dots x_n) = h(x_1) \dots h(x_n) \quad \text{für } i \in \mathbb{N} \text{ und } x_i \in \Sigma$$

Offensichtlich ist $h(L) = \{a^n b^n \mid n \in \mathbb{N}\}$. Außerdem ist h auch ein Homomorphismus: per Definition gilt $h(\varepsilon) = \varepsilon$ und $h(uv) = h(u)h(v)$.