

Übungsblatt 2

Aufgabe 1. Sei Σ ein beliebiges Alphabet. Geben Sie eine Grammatik an, die die Sprache

$$L = \{w \in \Sigma^* \mid w = w^r\}$$

erzeugt. Dabei ist w^r das Wort w rückwärts gelesen, z.B. für $w = aabb$ ist $w^r = bbaa$. Die Sprache L ist damit die Menge aller Palindrome über dem Alphabet Σ .

Lösung zu Aufgabe 1. Wir geben zunächst ein einfaches Beispiel. Sei $\Sigma = \{a, b\}$ das Alphabet. Wir suchen eine Grammatik G mit $L = L(G)$. Eine Möglichkeit ist $G = (V, \Sigma, P, S)$ mit

- $V = \{S\}$
- $P = \{S \rightarrow aSa \mid bSb \mid a \mid b \mid \varepsilon\}$

Erklärung

Formal müsste man wieder beide Mengeninklusionen \subseteq und \supseteq zeigen.

Mit den Produktionen $S \rightarrow aSa$ und $S \rightarrow bSb$ verlängert man das Wort am Anfang und am Ende beliebig oft jeweils um den gleichen Buchstaben, was sicherstellt, dass Palindrome erzeugt werden. Bei Wörtern gerader Länge wird die Ableitung durch Anwendung der Produktion $S \rightarrow \varepsilon$ beendet. Bei Wörtern ungerader Länge wird am Ende durch Anwendung von $S \rightarrow a$ oder $S \rightarrow b$ noch ein Symbol in der Mitte hinzugefügt.

Anmerkung: Gesucht ist eine beliebige Grammatik, daher ist die ε -Sonderregel nicht von Bedeutung. Man könnte die Grammatik unter Verwendung der Sonderregel allerdings so umformen, dass sie kontextfrei ist.

Für ein beliebiges Alphabet Σ wählt man die Produktionen zum Beispiel wie folgt:

$$P = \{S \rightarrow \varepsilon\} \cup \bigcup_{x \in \Sigma} \{S \rightarrow xSx, S \rightarrow x\}$$

Die Begründung ist hierbei genau wie im Fall $\Sigma = \{a, b\}$.

Aufgabe 2. Geben Sie zu jeder der folgenden Sprachen eine Grammatik und einen endlichen Automaten an.

- (a) $L_1 = \{w \in \{a, b\}^* \mid \text{Das Wort } w \text{ enthält mindestens ein } b.\}$
- (b) $L_2 = \{w \in \{a, b\}^* \mid \text{Die Anzahl der } a\text{'s ist durch 3 teilbar.}\}$
- (c) $L_3 = \{w \in \{a, b\}^+ \mid \text{Der erste und letzte Buchstabe in } w \text{ stimmen überein.}\}$
- (d) $L_4 = \{a^n b^m c^\ell \mid n \geq 0, m \geq 1, \ell \geq 2\}$
- (e) $L_5 = \{w \in \{a, b\}^* \mid |w| \leq 3\}$

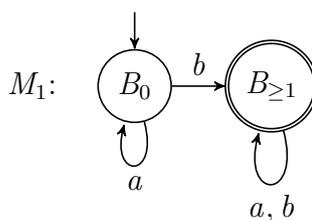
Lösung zu Aufgabe 2. Im Folgenden seien G_i die Grammatiken mit $L(G_i) = L_i$ und M_i die Automaten mit $T(M_i) = L_i$.

(a) $G_1 = (V, \Sigma, P, B_0)$ wobei

- $V = \{B_0, B_{\geq 1}\}$
- $\Sigma = \{a, b\}$
- $P = \{B_0 \rightarrow aB_0 \mid bB_{\geq 1}, B_{\geq 1} \rightarrow \varepsilon \mid aB_{\geq 1} \mid bB_{\geq 1}\}$

Alternativ, als reguläre Grammatik (ε eliminiert):

- $P = \{B_0 \rightarrow b \mid aB_0 \mid bB_{\geq 1}, B_{\geq 1} \rightarrow a \mid b \mid aB_{\geq 1} \mid bB_{\geq 1}\}$

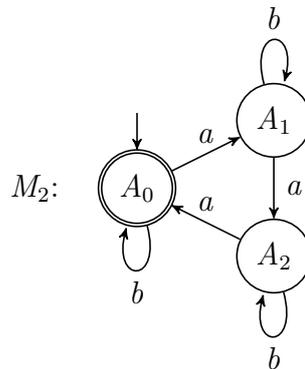


Erklärung

Wir verwenden zwei Zustände in diesem DFA um zu unterscheiden, ob bisher ausschließlich a 's gelesen wurden (Zustand B_0) oder mindestens ein b gelesen wurde (Zustand $B_{\geq 1}$). Die Idee für die Grammatik ist identisch.

(b) $G_2 = (V, \Sigma, P, A_0)$ wobei

- $V = \{A_0, A_1, A_2\}$
- $\Sigma = \{a, b\}$
- $P = \{A_0 \rightarrow \varepsilon | aA_1 | bA_0, A_1 \rightarrow aA_2 | bA_1, A_2 \rightarrow aA_0 | bA_2\}$

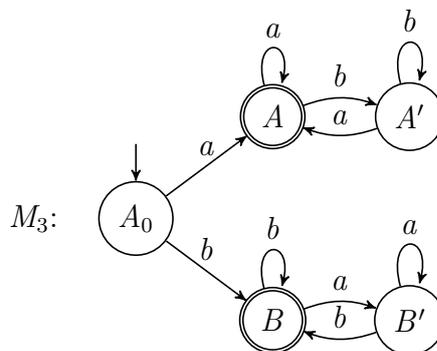


Erklärung

Wir verwenden drei Zustände in diesem DFA, um die Anzahl der a 's modulo 3 zu zählen. Der Zustand A_0 sagt aus, dass die gelesene Anzahl der a 's durch 3 teilbar ist. Zustand A_1 entspricht einem Rest von 1 und Zustand A_2 entspricht einem Rest von 2 bei Teilung durch 3. Die Grammatik ergibt sich wieder direkt aus dem Automaten.

(c) $G_3 = (V, \Sigma, P, S)$ wobei

- $V = \{S, A, B\}$
- $\Sigma = \{a, b\}$
- $P = \{S \rightarrow a|b|aA|bB, A \rightarrow a|aA|bA, B \rightarrow b|aB|bB\}$



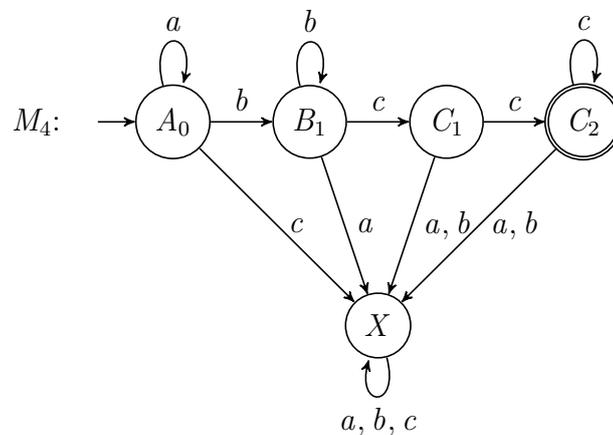
Erklärung

Die Wörter a und b beginnen und enden mit dem selben Buchstaben (daher die Regeln $S \rightarrow a$ und $S \rightarrow b$). Für längere Wörter merkt die Grammatik sich mit Hilfe von dem Nichtterminal A , dass der erste Buchstabe ein a ist und das Wort somit auch mit a enden muss. Analog werden mit Hilfe von B Wörter erzeugt, die mindestens Länge 2 haben und mit b beginnen und enden.

Der DFA arbeitet nach dem gleichen Prinzip, d.h. vom Startzustand aus erreicht man mit a einen Zustand A , so dass das Wort auch mit a enden muss und mit b erreicht man vom Startzustand einen Zustand B , so dass hier das Wort auch mit b enden muss.

(d) $G_4 = (V, \Sigma, P, A)$ wobei

- $V = \{A, B, C\}$
- $\Sigma = \{a, b\}$
- $P = \{A \rightarrow aA|bB, B \rightarrow bB|cC, C \rightarrow c|cC\}$



Erklärung

Der DFA und die Grammatik lesen bzw. erzeugen zuerst beliebig viele a 's (Zustand A_0 bzw. Nichtterminal A), wobei es auch möglich ist kein a zu erzeugen und direkt in den anschließenden b -Block überzugehen. Anschließend werden beliebig viele b 's gelesen bzw. erzeugt (Zustand B_1 und Nichtterminal B) und dabei wird sichergestellt, dass mindestens ein b wirklich gelesen bzw. erzeugt wird. Abschließend werden beliebig viele

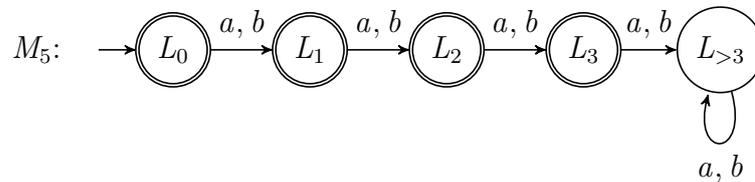
c 's gelesen bzw. erzeugt (Zustände C_1 , C_2 und Nichtterminal C) und dabei wird sichergestellt, dass die Anzahl der gelesenen c 's mindestens zwei ist. Der Fangzustand X im DFA wird erreicht, falls ein Wort gelesen wird, welches nicht aus a 's gefolgt von b 's gefolgt von c 's besteht.

(e) $G_5 = (V, \Sigma, P, S)$ wobei

- $V = \{S\}$
- $\Sigma = \{a, b\}$
- $P = \{S \rightarrow \varepsilon | a|b|aa|ab|ba|bb|aaa|aab|aba|abb|baa|bab|bba|bbb\}$

Alternativ, als reguläre Grammatik:

- $V = \{S, L_1, L_2\}$
- $P = \{S \rightarrow \varepsilon | a|b|aL_1|bL_1, L_1 \rightarrow a|b|aL_2|bL_2, L_2 \rightarrow a|b\}$



Erklärung

Für den DFA gilt: Der Startzustand L_0 ist gleichzeitig ein Endzustand, sodass das leere Wort akzeptiert wird, während L_1 mit Wörtern der Länge 1, L_2 mit Wörtern der Länge 2 und L_3 mit Wörtern der Länge 3 erreicht wird. Alle Wörter der Länge mindestens 4 landen im Fangzustand $L_{>3}$. Für die Grammatik können wir für jedes Wort w der Länge $|w| \leq 3$ eine Produktion $S \rightarrow w$ angeben, allerdings ist diese Grammatik nicht regulär. Für eine reguläre Grammatik muss man für verschiedene Wortlängen andere Nichtterminale verwenden.

Aufgabe 3. Geben Sie zu der folgenden Sprache einen DFA an:

$$L = \{w \in \{a, b, \dots, y, z\}^* \mid w \text{ enthält das Teilwort } \textit{essen}\}$$

Lösung zu Aufgabe 3. Sei $\Sigma = \{a, b, \dots, z\}$. Der Automat sieht wie folgt aus:

