

Theoretische Informatik 1

Große Übung 3

René Maseli
Prof. Dr. Roland Meyer

TU Braunschweig
Wintersemester 2024/25

Potenzmengenkonstruktion nach Rabin & Scott

Es reichen die erreichbaren Zustände. Dazu starte mit $\{q_0\}$.

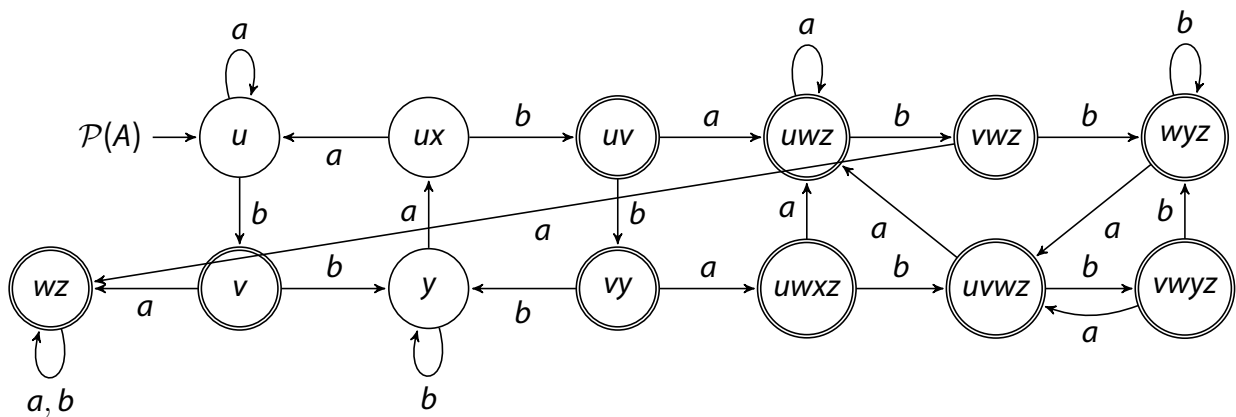
Für jedes Symbol in Σ , bestimme die Folgezustandsmenge und füge eine Kante hinzu.

Falls es nun keine Zustände gibt, denen ausgehende Kanten fehlen, ist man fertig.

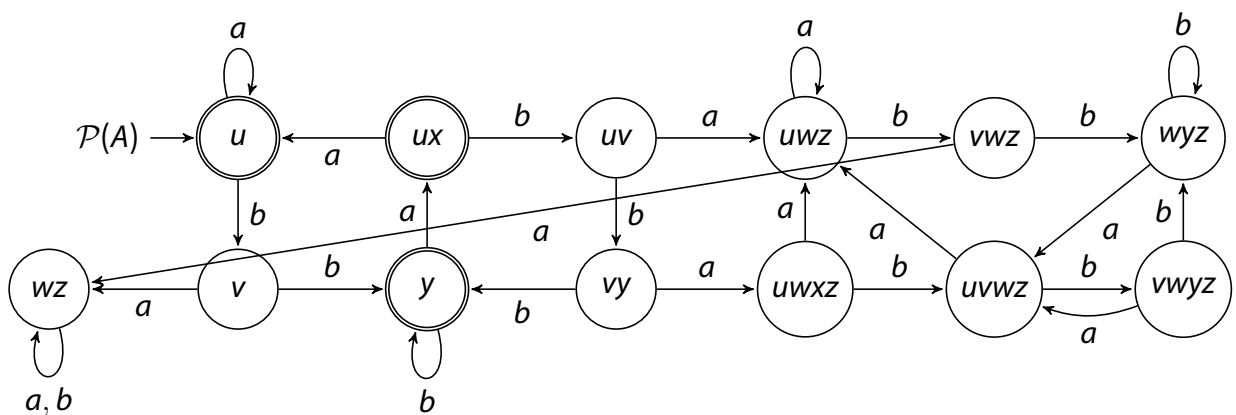
Sonst, wiederhole den Prozess mit einem solchen Zustand.

(Akzeptierend sind die Zustandsmengen, die einen akzeptierenden Zustand enthalten.)

Tipp: Bevorzugt zuerst kleinere Teilmengen vor Größeren.



Komplementierung eines DFA: Akzeptierend sind nun die Zustände, die keinen akzeptierenden Zustand enthalten. $Q_F'' := \{Q' \subseteq Q \mid \text{nicht } \exists q \in Q' : q \in Q_F\}$



Abgeschlossenheit unter Komplement

In der Vorlesung wurde gesagt, dass aus der Existenz von DFAs und der Idee des „Flippens“ akzeptierender Zustände automatisch folgt, dass das Komplement regulärer Sprachen immer auch regulär sein muss. Ein formaler Beweis könnte so aussehen:

Sei L eine reguläre Sprache.

Demnach gibt es einen NFA A mit $\mathcal{L}(A) = L$.

Betrachte den Potenzmengen-Automaten $\mathcal{P}(A)$ mit $\mathcal{L}(\mathcal{P}(A)) = \mathcal{L}(A)$ (Rabin & Scott).

Betrachte weiterhin den DFA $\overline{\mathcal{P}(A)}$, bei welchem die Finalzustände „geflippt“ sind.

Da $\mathcal{P}(A)$ deterministisch ist, gilt $\mathcal{L}(\overline{\mathcal{P}(A)}) = \overline{\mathcal{L}(\mathcal{P}(A))}$.

Zusammen erhalten wir, dass $\mathcal{L}(\overline{\mathcal{P}(A)}) = \overline{L}$ gilt.

Damit ist \overline{L} regulär.

Da dies für alle L gilt, sind reguläre Sprachen unter Komplement abgeschlossen.

Produktautomat

Theorem

Seien Σ ein endliches Alphabet und $L, K \subseteq \Sigma^*$ regulär. Der Durchschnitt $L \cap K$ ist regulär.

Beweis

Klar mit deMorgan'schen Regeln und der bereits bekannten Abgeschlossenheit unter Komplement: $L \cap K = \overline{\overline{L} \cup \overline{K}}$. □

Dennoch gibt es einen praktischeren Ansatz. Seien $A = \langle Q_A, q_{0A}, \rightarrow_A, Q_{FA} \rangle$ und $B = \langle Q_B, q_{0B}, \rightarrow_B, Q_{FB} \rangle$ endliche Automaten mit $\mathcal{L}(A) = L$ und $\mathcal{L}(B) = K$. Der Produktautomat $A \cap B = \langle Q_A \times Q_B, \langle q_{0A}, q_{0B} \rangle, \rightarrow', Q_{FA} \times Q_{FB} \rangle$ enthält genau dann eine Transition $\langle p_A, p_B \rangle \xrightarrow{s} \langle q_A, q_B \rangle$, wenn gleichzeitig $p_A \xrightarrow{s} q_A$ und $p_B \xrightarrow{s} q_B$ existieren. Man sagt, $A \cap B$ simuliert gleichzeitig A und B und akzeptiert genau dann, wenn beide Simulationen akzeptieren.

Es gilt $\mathcal{L}(A \cap B) = \mathcal{L}(A) \cap \mathcal{L}(B)$.

Beweis

Sei $w = w_1 w_2 \dots w_n \in \Sigma^*$. Wir zeigen beide Teilmengen-Beziehungen gleichzeitig.

$$w \in \mathcal{L}(A) \cap \mathcal{L}(B)$$

$$\text{gdw. } q_{0A} \xrightarrow{w}_A q_{nA} \in Q_{FA} \text{ und } q_{0B} \xrightarrow{w}_B q_{nB} \in Q_{FB}$$

$$\text{gdw. es } q_{0A} \xrightarrow{w_1}_A q_{1A} \xrightarrow{w_2}_A \dots \xrightarrow{w_n}_A q_{nA} \in Q_{FA} \text{ und } q_{0B} \xrightarrow{w_1}_B q_{1B} \xrightarrow{w_2}_B \dots \xrightarrow{w_n}_B q_{nB} \in Q_{FB} \text{ gibt}$$

$$\text{gdw. es } \langle q_{0A}, q_{0B} \rangle \xrightarrow{w_1} \langle q_{1A}, q_{1B} \rangle \xrightarrow{w_2} \dots \xrightarrow{w_n} \langle q_{nA}, q_{nB} \rangle \in Q_{FA} \times Q_{FB} \text{ gibt}$$

$$\text{gdw. } \langle q_{0A}, q_{0B} \rangle \xrightarrow{w} \langle q_{nA}, q_{nB} \rangle \in Q_{FA} \times Q_{FB}$$

$$\text{gdw. } w \in \mathcal{L}(A \cap B)$$

Shuffle

Definition

Seien Σ ein Alphabet und $v, w \in \Sigma^*$. Der Shuffle $v \sqcup w$ ist die Menge der Worte, die durch eine beliebige abwechslung der Buchstaben von v und w hervorgehen. Dabei soll die Reihenfolge der Buchstaben aus demselben Wort erhalten bleiben. Der Shuffle zweier Sprachen $L \subseteq \Sigma^*$ und $M \subseteq \Sigma^*$ ist die Sprache

$$L \sqcup M := \begin{cases} \{v_1 \cdot w_1\} & \text{falls } L = \{v_1\} \text{ und } M = \{w_1\} \text{ und } (v_1 = \varepsilon \text{ oder } w_1 = \varepsilon) \\ a \cdot (x \sqcup w_2) \cup b \cdot (v_2 \sqcup y) & \text{falls } L = \{v_2\} \text{ und } M = \{w_2\} \text{ und } v_2 = a \cdot x \text{ und } w_2 = b \cdot y \\ \bigcup_{v_3 \in L, w_3 \in M} \{v_3\} \sqcup \{w_3\} & \text{sonst} \end{cases}$$

Beispiel

$aab \sqcup ba = \{\underline{aabb}, \underline{aabb}, \underline{aabab}, \underline{ababa}, \underline{abaab}, \underline{abaab}, \underline{baaba}, \underline{baaab}, \underline{baaab}, \underline{baaab}\}$

Beispiele für reguläre Sprachen sind

$$(aa)^* \sqcup b^* = b^*(ab^*ab^*)^*$$

$$a^* \sqcup b^* = \{a, b\}^*$$

$$(ab)^* \sqcup (ba)^* = (ab \cup ba)^*$$

$$a \sqcup a^* \cdot b \cdot a^* = a^* \cdot (ab \cup ba) \cdot a^*$$

Proposition

Falls L und M regulär sind, dann ist es auch $L \sqcup M$.

Seien $A = \langle Q_A, q_{0A}, \rightarrow_A, Q_{FA} \rangle$ und $B = \langle Q_B, q_{0B}, \rightarrow_B, Q_{FB} \rangle$ NFAs auf dem Alphabet Σ . Wir konstruieren einen neuen NFA $A \sqcup B = \langle Q, q_0, \rightarrow, Q_F \rangle$ mit $\mathcal{L}(A \sqcup B) = \mathcal{L}(A) \sqcup \mathcal{L}(B)$.

$$Q = Q_A \times Q_B$$

$$q_0 = \langle q_{0A}, q_{0B} \rangle$$

$$Q_F = Q_{FA} \times Q_{FB}$$

$$\langle p_A, p_B \rangle \xrightarrow{s} \langle q_A, q_B \rangle \quad \text{gdw.} \quad \left(p_A \xrightarrow{s} p_A \wedge p_B = q_B \right) \vee \left(p_A = q_A \wedge p_B \xrightarrow{s} q_B \right)$$

Nun gilt es, die Korrektheit dieser Konstruktion zu zeigen. Das Folgende wurde gegen Ende nur kurz beschrieben.

Beweis

Sei $x = x_1 \dots x_k \in \mathcal{L}(A \sqcup B)$.

Dann gibt es einen akzeptierenden Lauf $q_0 \xrightarrow{x_1} \langle a_1, b_1 \rangle \xrightarrow{x_2} \dots \xrightarrow{x_k} \langle a_k, b_k \rangle \in Q_F$.

Dabei simuliert gemäß der obigen Konstruktion jede dieser Transitionen entweder eine A -Transition oder eine B -Transition, während der simulierte Zustand des jeweils anderen Automaten unverändert bleibt.

Die Aneinanderreihung der A -Transitionen liefert nun ein Subwort $v \in \Sigma^*$ mit einem akzeptierenden Lauf $q_{0A} \xrightarrow{v}^* a_k \in Q_{FA}$, der aufgrund der Konstruktion von $A \sqcup B$ in A existieren musste.

Analog erhalten wir über den B -Transitionen das andere Subwort $w \in \Sigma^*$ und einen akzeptierenden Lauf $q_{0B} \xrightarrow{w}^* b_k \in Q_{FB}$.

Jetzt beobachten wir $x \in v \sqcup w \subseteq \mathcal{L}(A) \sqcup \mathcal{L}(B)$.

Da dies für alle x gilt, folgt $\mathcal{L}(A \sqcup B) \subseteq \mathcal{L}(A) \sqcup \mathcal{L}(B)$.

Sei $x = x_1 \dots x_k \in \mathcal{L}(A) \sqcup \mathcal{L}(B)$, insbesondere $x \in v \sqcup w$ mit $v = v_1 \dots v_n \in \mathcal{L}(A)$ und $w = w_1 \dots w_m \in \mathcal{L}(B)$ mit $n + m = k$. Genauer betrachte eine beliebige Buchstabenzuweisung $f: \{1, \dots, k\} \rightarrow \{v, w\}$ mit $v = \langle x_i \rangle_{f(i)=v}$ und $w = \langle x_i \rangle_{f(i)=w}$, falls mehrere existieren.

Dann gibt es akzeptierenden Läufe $q_{0A} = a_0 \xrightarrow{v_1}^* \dots \xrightarrow{v_n}^* a_n \in Q_{FA}$ und $q_{0B} = b_0 \xrightarrow{w_1}^* \dots \xrightarrow{w_m}^* b_m \in Q_{FB}$.

In der gleichen Anordnung wie im Wort, lässt sich nun der folgende Lauf konstruieren:

$$q_0 = \langle a'_0, b'_0 \rangle \xrightarrow{x_1} \langle a'_1, b'_1 \rangle \xrightarrow{x_2} \dots \xrightarrow{x_k} \langle a'_k, b'_k \rangle$$

$$\langle a'_{i+1}, b'_{i+1} \rangle := \begin{cases} \langle a_{j+1}, b'_i \rangle & \text{falls } a'_i = a_j \text{ und } f(i+1) = v \\ \langle a'_i, b_{j+1} \rangle & \text{falls } b'_i = b_j \text{ und } f(i+1) = w. \end{cases}$$

Per Konstruktion existieren alle erforderlichen Transitionen tatsächlich und es gilt $\langle a'_k, b'_k \rangle = \langle a_n, b_m \rangle \in Q_F$. Also wird x von $A \sqcup B$ akzeptiert.

Da dies für alle x gilt, folgt $\mathcal{L}(A) \sqcup \mathcal{L}(B) \subseteq \mathcal{L}(A \sqcup B)$.

Zusammen folgt $\mathcal{L}(A \sqcup B) = \mathcal{L}(A) \sqcup \mathcal{L}(B)$. □