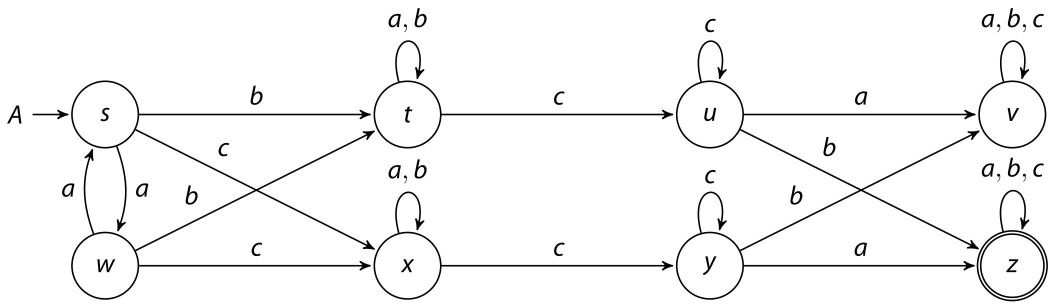
Übungsaufgabe 7.7: Table-Filling-Algorithmus

Gegeben ist der folgende DFA. Ziel ist es, einen minimalen DFA zu erzeugen.



Bemerkung: In der Übung waren die Zustände von 0 bis 7 benannt.

Die folgende Tabelle betrachtet ungeordnete Paare aus Zuständen von A. Die Nummern stehen für die Iteration des Algorithmus, in welcher das entsprechende Paar von Zuständen als ungleich festgestellt wurden.

	s	t	u	v	w	x	y	z
s		3	1	2		2	1	0
t			1	2	2	2	1	0
u				1	1	1	1	0
v					3	2	1	0
w						2	1	0
x							1	0
y								0
z								

Vorgehensweise:

Initial (0) alle Q_F von $Q \setminus Q_F$ trennen.

0: z anders als der Rest

Iterativ (1,2,...) die Vorgänger frisch getrennter Zustände untersuchen. Als Hilfestellung stehen hier die für die jeweilige Iteration interessanten Transitionen, Self-Loops ausgenommen.

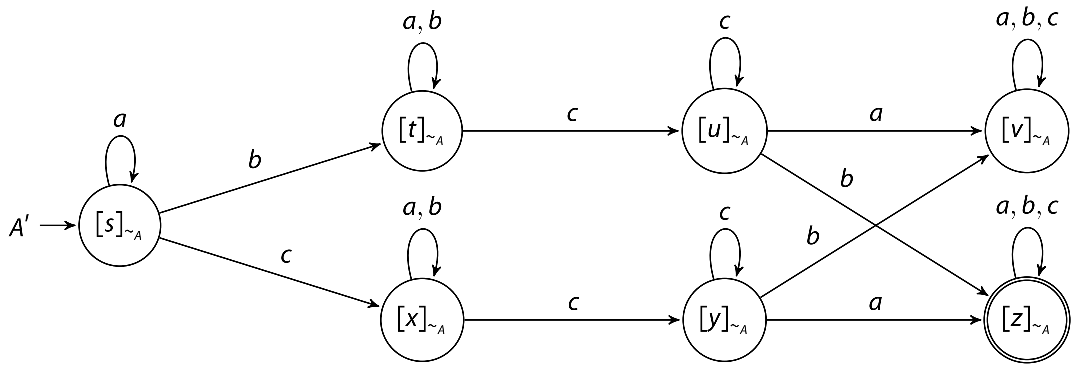
1: $u \xrightarrow{b} z, y \xrightarrow{a} z$ (⇒ 11 neue Trennungen)

2: $t \xrightarrow{c} u, x \xrightarrow{c} y$ (⇒ 7 neu)

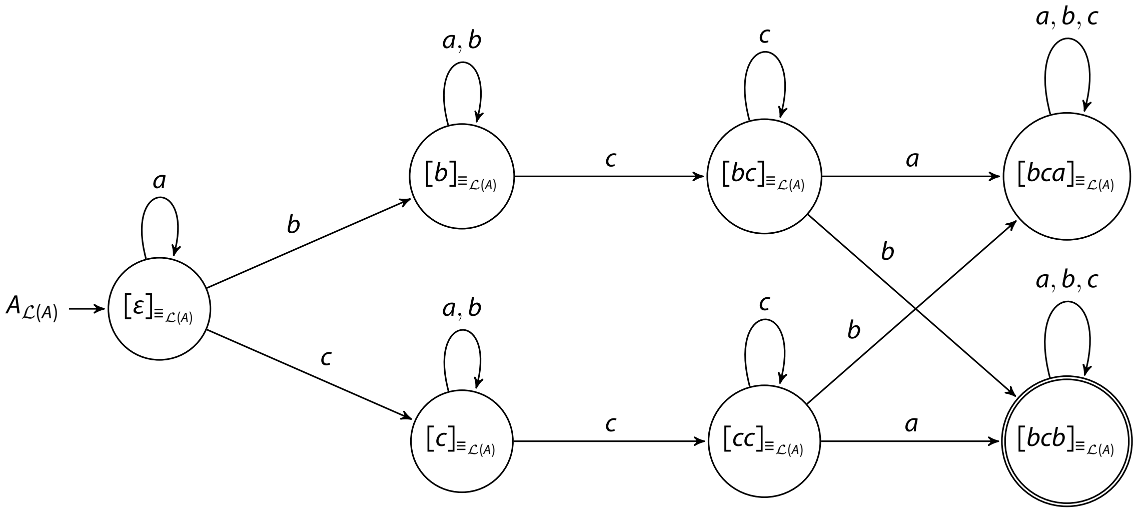
3: $s \xrightarrow{b} t, w \xrightarrow{b} t, s \xrightarrow{c} x, w \xrightarrow{c} x$ (⇒ 2 neu)

4: $w \xrightarrow{a} s, s \xrightarrow{a} w$

Bei der vierten Iteration konnten keine neuen Paare unterschieden werden. Die leeren Zellen verraten nun, welche Zustände verschmolzen werden müssen, um einen minimalen DFA für $\mathcal{L}(A)$ zu erzeugen: (Es reicht, einen der beiden unteren Automaten zu zeichnen.)



Es sollen alle Äquivalenzklassen der Nerode-Rechtskongruenz von $\mathcal{L}(A)$ aufgelistet werden. Jede dieser Klassen ist eine Sprache über dem Alphabet $\{a, b, c\}$ und ist einem der Zustände des minimalen DFA zugewiesen, nämlich den einzigartigen Zustand q mit $q_0 \xrightarrow{w} q$ für jedes Wort w aus der Äquivalenzklasse. Die Klassen indentifizieren sich am Besten mit einem Repräsentanten, z.B. einem kürzesten Wort, das den Zustand ansteuert.



Ein minimaler Automat ist nützlich, um alle Nerode-Rechtskongruenzen zu berechnen. Betrachte dazu jeweils einen Zustand als einzigen Final-Zustand und erzeuge einen regulären Ausdruck mit Arden's Lemma. Diese Sprachen sollten paarweise disjunkt sein, und die Vereinigung sollte Σ^* sein.

$$
\begin{aligned}
[\varepsilon]_{\equiv_{\mathcal{L}(A)}} &= a^* \\
[b]_{\equiv_{\mathcal{L}(A)}} &= a^* b(a \cup b)^* \\
[c]_{\equiv_{\mathcal{L}(A)}} &= a^* c(a \cup b)^* \\
[bc]_{\equiv_{\mathcal{L}(A)}} &= a^* b(a \cup b)^* c^+ \\
[cc]_{\equiv_{\mathcal{L}(A)}} &= a^* c(a \cup b)^* c^+ \\
[bcb]_{\equiv_{\mathcal{L}(A)}} &= a^* \big(b(a \cup b)^* c^+ b \cup c(a \cup b)^* c^+ a\big)(a \cup b \cup c)^* \\
[bca]_{\equiv_{\mathcal{L}(A)}} &= a^* \big(b(a \cup b)^* c^+ a \cup c(a \cup b)^* c^+ b\big)(a \cup b \cup c)^*
\end{aligned}
$$

Übungsaufgabe 7.8: Pumping-Lemma Das Pumping-Lemma stellt ein vergleichsweise einfaches notwendiges Kriterium an reguläre Sprachen. Jede reguläre Sprache L besitzt eine Pumping-Konstante $p_L \in \mathbb{N}$, sodass für jedes längere Wort $w \in L$ mit $|w| \geq p_L$ zerlegbar ist in $w = xyz$ mit $|xy| \leq p_L$ und $y \neq \varepsilon$, sodass für alle $i \in \mathbb{N}$ das gepumpte Wort $xy^iz \in L$ auch in der Sprache liegt.

Anders ausgedrückt, kann eine Sprache nicht regulär sein, wenn es für jede potenzielle Zustandszahl eines NFAs ein akzeptiertes Wort mit mindestens einem Schleifen-Durchlauf gibt, sodass jede potenzielle erste Schleife, die für dieses Wort durchlaufen werden muss, mindestens ein ungewolltes Wort zuviel akzeptieren lassen würde.

a) Sei $L_0 := \{ w \in \{a, b\}^* \mid |w|_a \geq |w|_b \}$. Es ist zu zeigen, dass L_0 nicht regulär ist.

Beweis
Sei $0 < p_L \in \mathbb{N}$.
Wähle das Wort $x = a^{p_L} b^{p_L} \in L_0$.
Sei $x = u.v.w$ eine Zerlegung mit $|uv| \leq p_L$ und $v \neq \varepsilon$.
So wie das Wort gewählt wurde, gilt immer $v \in a^+$.
Betrachte $u.v^0.w = a^{p_L - |v|} b^{p_L} \notin L_0$, weil $|uw|_a = p_L + 1 - |v| < p_L + 1 = |uw|_b$.
Da dies für alle p_L gilt, ist L_0 nach dem Pumping-Lemma nicht regulär. □

b) Sei $L_1 := \{ w \in \{a, b\}^* \mid |w|_a \leq |w|_b \text{ oder } 2|w|_b \leq |w|_a \}$ Es ist zu zeigen, dass L_1 nicht regulär ist.

Beweis
Sei $p_L \in \mathbb{N}$.
Wähle das Wort $x = a^{2(p_L+1)} b^{p_L+1} \in L_1$.
Sei $x = u.v.w$ eine Zerlegung mit $|uv| \leq p_L$ und $v \neq \varepsilon$.
So wie das Wort gewählt wurde, gilt immer $v \in a^+$ und $|v| \leq p$.
Betrachte $u.v^0.w = a^{2p_L+2-|v|} b^{p_L+1} \notin L_1$, weil $|uw|_a = 2p_L + 2 - |v| > p_L + 1 = |uw|_b$ und $2|uw|_b = 2p_L + 2 > 2p_L + 2 - |v| = |uw|_a$.
Da dies für alle p_L gilt, ist L_0 nach dem Pumping-Lemma nicht regulär. □

c) Sei $L_2 := \{ a^n b^m \mid n, m \in \mathbb{N} \text{ und } (n \neq 1 \text{ oder } \exists \ell \in \mathbb{N}\colon m = \ell^2) \}$.

Es kann mit dem Pumping-Lemma nicht (sofort) gezeigt werden, dass L_2 nicht regulär ist:

Gegenbeweis:
Wähle $p_L = 3$. Sei $w = a^n b^m$ mit $n + m \geq p_L$ und $n = 1$ oder $m = \ell^2$.
Falls $n = 0$ oder $n = 2$, wähle eine Zerlegung mit $y = b$. Anderenfalls ist $n = 1$ oder $n \geq 3$.
Wähle eine Zerlegung mit $y = a$. Weder $i = 0$ noch $i \geq 2$ können $xy^iz \in L_2$ verhindern.
Nach dem Pumping-Lemma kann so keine Aussage über L_2 getroffen werden.

Um trotzdem Regularität widerlegen zu können, kann man eine Abschluss-Eigenschaften nutzen: Falls L_2 regulär ist, dann sind es auch e.g. $\overline{L_2}$, L_2^{reverse} oder $L_2 \cap ab^*$.

Beweis
Sei $p_L \in \mathbb{N}$.
Betrachte $ab^{p_L^2} \in L_2 \cap ab^*$.
Bei den Zerlegungen unterscheiden wir zwei Fälle:
Falls $y \in ab^*$, wähle $i \neq 1$ beliebig. E.g. folgt einfach $xy^0z \notin L_2 \cap ab^*$.
Sonst ist $y \in b^+$. Falls $p_L^2 - |y|$ nicht quadratisch ist, wähle $i = 0$.
Anderenfalls betrachte $q = p_L - \sqrt{p_L^2 - |y|}$. Es gilt mit der zweiten binomischen Formel $p_L^2 - |y| = (p_L - q)^2 = p_L^2 - 2p_L q + q^2$. Wähle $i \geq 2$ beliebig, denn schon in den reellen Zahlen hat das Polynom $q \mapsto q^2 - 2p_L q + |y|$ keine dritte Nullstelle.
Nach dem Pumping-Lemma ist $L_2 \cap ab^*$ nicht regulär. Damit ist auch L_2 nicht regulär. □