Prof. Dr. Jasmin Blanchette
Jannis Limperg
Luca Maio

Ludwig-Maximilians-Universität München
Institut für Informatik
Abgabe bis 1. Mai 2024, 10:00 Uhr

Lösungsvorschlag zur Übung 2 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.

FSK2-1 *Grammatiken und Elimination von ε-Produktionen* (2 Punkte)

a) Geben Sie kontextfreie Grammatiken (ggf. mit ε-Produktionen) an, die folgende Sprachen über dem Alphabet $\Sigma = \{a, b\}$ erkennen:

i) $L_1 = \{caw \mid c \in \Sigma,\ w \in \Sigma^*\}$

LÖSUNGSVORSCHLAG:
$G_1 = (\{S, C, W\}, \Sigma, P_1, S)$ mit

$$P_1 = \{S \to CaW,\ C \to a,\ C \to b,\ W \to \varepsilon,\ W \to aW,\ W \to bW\}$$

ii) $L_2 = \{aawbb \mid w \in \Sigma^*\}$

LÖSUNGSVORSCHLAG:
$G_2 = (\{S, W\}, \Sigma, \{S \to aaWbb,\ W \to \varepsilon,\ W \to aW,\ W \to bW\}, S)$

b) Betrachten Sie die Grammatik $G = (\{A, B, C\}, \{a, b\}, P, A)$ mit Produktionen

$$P = \{A \to aBbC,\ A \to AaA,\ B \to \varepsilon,\ B \to CC,\ C \to BB\}$$

Geben Sie eine zu G äquivalente Grammatik G' ohne ε-Produktionen an. Verwenden Sie den Algorithmus zur Elimination von ε-Produktionen aus der Vorlesung und geben Sie die Zwischenschritte Ihrer Berechnung an. (Das ermöglicht uns, Ihnen bei kleinen Fehlern noch Teilpunkte zu geben.)

LÖSUNGSVORSCHLAG:

- Finde die Menge W der Variablen D mit $D \Rightarrow^* \varepsilon$, wobei D nicht das Startsymbol ist.
  - Initial $W := \{B\}$, da $B \to \varepsilon$ und $B \neq A$.

- Wegen $C \to BB$ und $B \in W$ muss $C \in W$ sein, also $W := \{B, C\}$.
  - Wegen $B \to CC$ und $C \in W$ muss $B \in W$ sein, aber das war es auch vorher schon.
  - Es gibt keine weiteren Produktionen der Form $D \to E_1 \dots E_n$ mit $E_1, \dots, E_n \in W$, also ist $W = \{B, C\}$.

- Entferne Produktionen der Form $D \to \varepsilon$, wobei D nicht das Startsymbol ist:

$$P' := \{A \to aBbC,\ A \to AaA,\ B \to CC,\ C \to BB\}$$

- Für Produktionen der Form $D \to uEv \in P'$ mit $|uv| > 0$ und $E \in W$, füge $D \to uv$ zu P' hinzu.
  - Wegen $B \to CC \in P'$: $P' := P' \cup \{B \to C\}$.
  - Wegen $C \to BB \in P'$: $P' := P' \cup \{C \to B\}$.
  - Wegen $A \to aBbC \in P'$: $P' := P' \cup \{A \to abC\}$.
  - Wegen $A \to aBbC \in P'$: $P' := P' \cup \{A \to aBb\}$.
  - Wegen $A \to aBb \in P'$: $P' := P' \cup \{A \to ab\}$.
  - Wegen $A \to abC \in P'$: $P' := P' \cup \{A \to ab\}$, aber diese Produktion war bereits enthalten.
  - Es gibt keine weiteren Produktionen der gewünschten Form in P', also ist

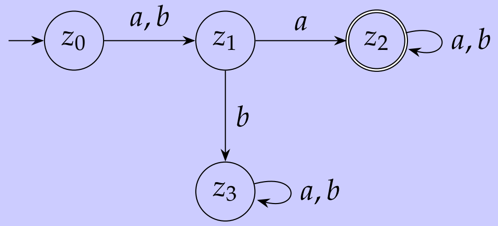
$$P' = \{A \to aBbC,\ A \to AaA,\ B \to CC,\ C \to BB,\ B \to C,\ C \to B,$$
$$A \to abC,\ A \to aBb,\ A \to ab\}$$

FSK2-2 ***DFAs und Minimierung*** (2 Punkte)

a) Geben Sie DFAs an, die folgende Sprachen über dem Alphabet $\Sigma = \{a, b\}$ erkennen:

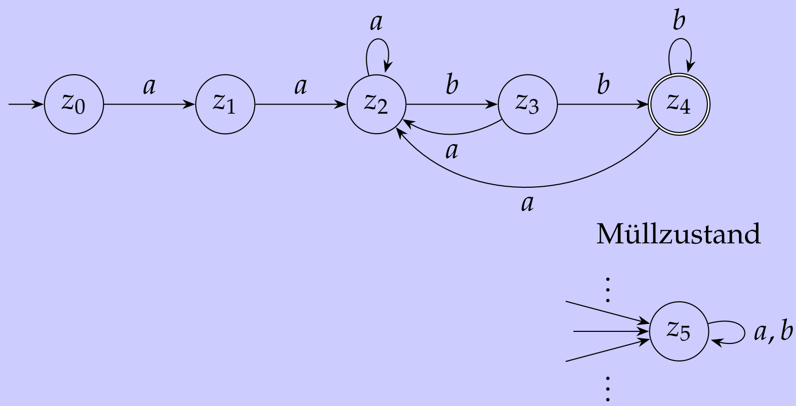
   i) $L_1 = \{caw \mid c \in \Sigma,\ w \in \Sigma^*\}$

   LÖSUNGSVORSCHLAG:



ii) $L_2 = \{aawbb \mid w \in \Sigma^*\}$

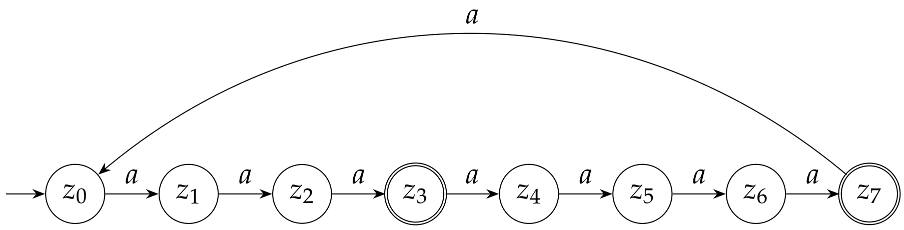
LÖSUNGSVORSCHLAG:



(Das sind die gleichen Sprachen wie in Aufgabe FSK2-1.)

b) Minimieren Sie die folgenden DFAs. Verwenden Sie die tabellarische Variante des Algorithmus zur Minimierung von DFAs aus der Vorlesung (nicht die graphische Variante und nicht den Algorithmus von letztem Jahr!). Geben Sie die Partitionstabelle und den minimalen DFA an.

i) DFA A_1 über dem Alphabet $\Sigma = \{a\}$:



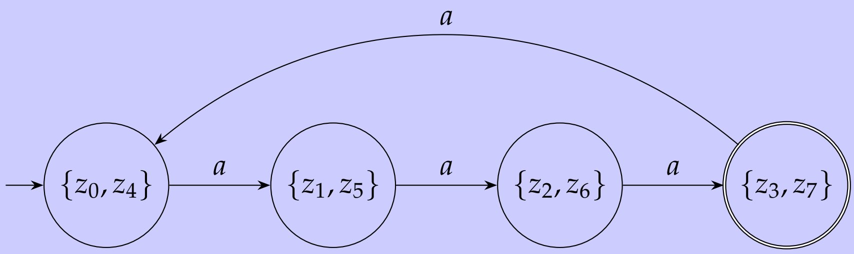
LÖSUNGSVORSCHLAG:
Partitionstabelle:

$z_0\quad z_1\quad z_2\quad z_4\quad z_5\quad z_6 \mid z_3\quad z_7$

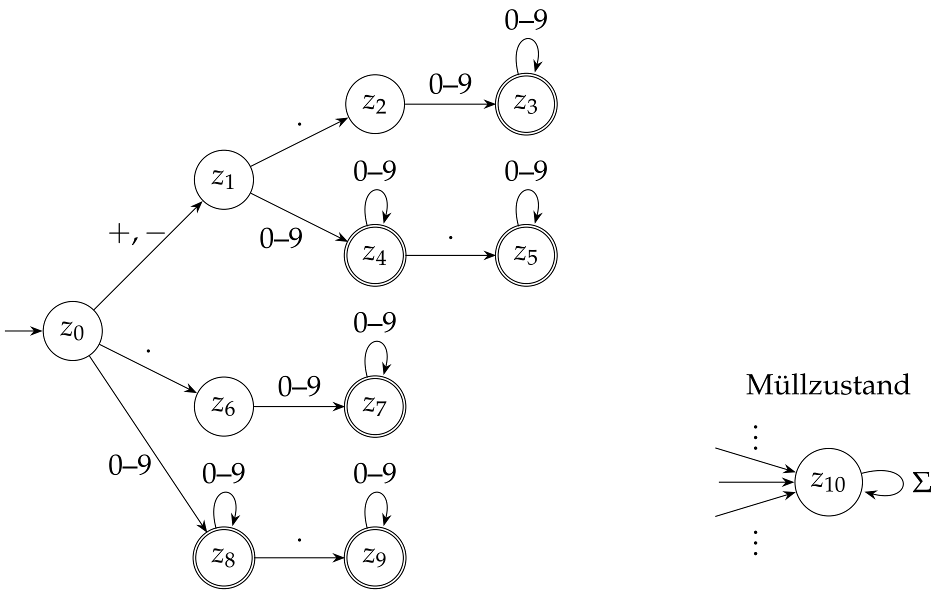
$z_0\quad z_1\quad z_4\quad z_5 \mid z_2\quad z_6 \mid z_3\quad z_7$ mit a

$z_0\quad z_4 \mid z_1\quad z_5 \mid z_2\quad z_6 \mid z_3\quad z_7$ mit a

Aus der Partitionierung ergibt sich der Minimalautomat:



ii) DFA A_2 über dem Alphabet $\Sigma = \{+, -, ., 0, \dots, 9\}$ (bekannt aus der Vorlesung):



LÖSUNGSVORSCHLAG:
Partitionstabelle:

$z_0\quad z_1\quad z_2\quad z_6\quad z_{10} \mid z_3\quad z_4\quad z_5\quad z_7\quad z_8\quad z_9$

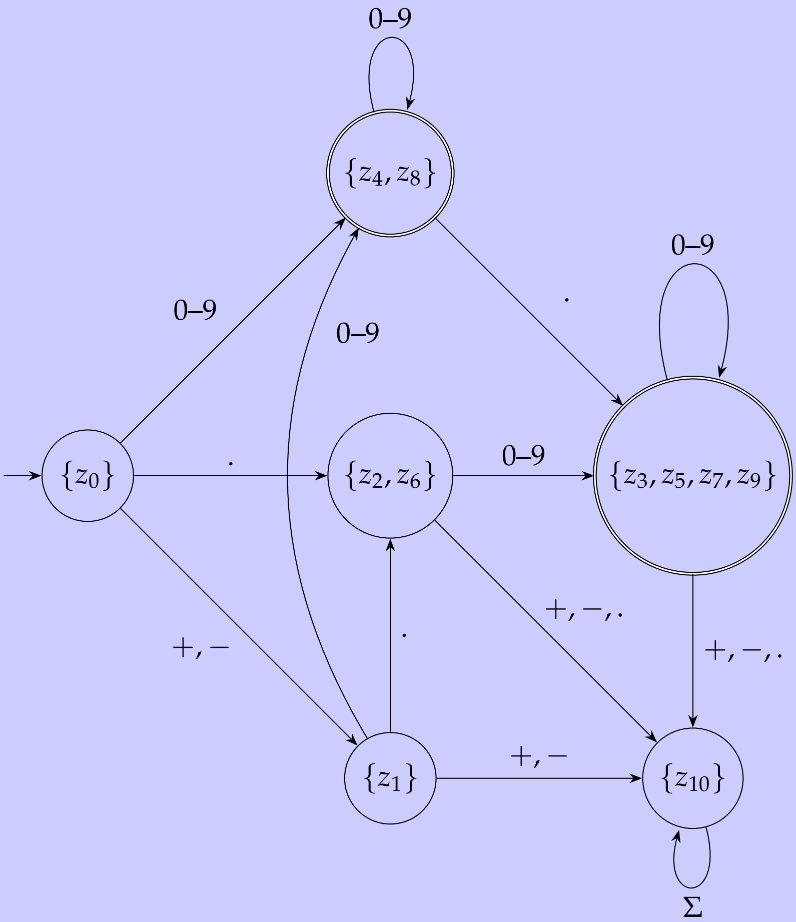
$z_0\quad z_1\quad z_2\quad z_6 \mid z_{10} \mid z_3\quad z_4\quad z_5\quad z_7\quad z_8\quad z_9$ mit 0

$z_0 \mid z_1\quad z_2\quad z_6 \mid z_{10} \mid z_3\quad z_4\quad z_5\quad z_7\quad z_8\quad z_9$ mit +

$z_0 \mid z_1 \mid z_2\quad z_6 \mid z_{10} \mid z_3\quad z_5\quad z_7\quad z_9 \mid z_4\quad z_8$ mit .

Sie dürfen (und sollten vernünftigerweise) Mengen von Symbolen, die immer zusammen vorkommen, für diesen Algorithmus als ein Symbol betrachten. Beispielsweise müssen Sie bei dem gegebenen Automaten nicht separat prüfen, ob 0, 1, 2, …, 9 zu einer Partitionierung führen; es genügt, ein Symbol aus dieser Menge zu prüfen.

Aus der Partitionierung ergibt sich der Minimalautomat:



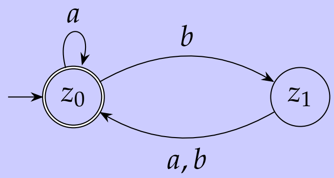
FSK2-3 ***Kleine Automaten*** (0 Punkte)

a) Sei A_1 ein DFA mit Alphabet Σ und genau einem Zustand. Zeigen oder widerlegen Sie: Es ist entweder $L(A_1) = \Sigma^*$ oder $L(A_1) = \emptyset$.

LÖSUNGSVORSCHLAG:
Stimmt. Der Automat hat genau einen Zustand z_0 (der also auch Startzustand ist) und muss Übergänge $z_0 \xrightarrow{a} z_0$ für jedes $a \in \Sigma$ haben. Wenn z_0 ein Endzustand ist, akzeptiert A_1 somit jedes Wort in Σ^*. Wenn z_0 kein Endzustand ist, akzeptiert A_1 kein Wort.

b) Sei A_2 ein DFA mit Alphabet $\Sigma = \{a, b\}$ und genau zwei Zuständen. Angenommen es gibt ein Wort $z \notin L(A_2)$ und für alle $i \in \mathbb{N}$ ist $a^i \in L(A_2)$. Zeigen oder widerlegen Sie: Für jeden solchen Automaten A_2 ist $bb \notin L(A_2)$.

LÖSUNGSVORSCHLAG:
Falsch. Beispielsweise gilt es in folgendem Automaten nicht:



c) Zeigen Sie: Für jeden DFA A mit Alphabet $\Sigma = \{a, b\}$ und genau vier Zuständen gilt: Wenn für jede natürliche Zahl $n \geq 1$ das Wort $a^{n^2} \in L(A)$ ist, dann ist auch $a^{12} \in L(A)$.

LÖSUNGSVORSCHLAG:
In A muss es mindestens eine Schleife aus a-Übergängen geben, da A beliebig lange a-Sequenzen erkennt, aber nur 4 Zustände hat.
Sei z_0 der Startzustand von A. Sei $z_w = \widetilde{\delta}(z_0, a^{12})$ der Zustand, in dem wir uns nach Lesen von a^{12} befinden. Dieser Zustand z_w muss Teil einer Schleife sein, da A nur 4 Zustände hat. Sei $s \in \{1, 2, 3, 4\}$ die Länge dieser Schleife.
Es gilt somit für jedes $k \in \mathbb{N}$: $\widetilde{\delta}(z_w, a^{ks}) = z_w$.
Wir zeigen, dass z_w ein Endzustand ist. Fallunterscheidung über s:

- Für $s = 1$ gilt $\widetilde{\delta}(z_0, a^{4^2}) = \widetilde{\delta}(z_0, a^{12+4}) = \widetilde{\delta}(z_w, a^{4 \cdot 1}) = z_w$
- Für $s = 2$ gilt $\widetilde{\delta}(z_0, a^{4^2}) = \widetilde{\delta}(z_0, a^{12+4}) = \widetilde{\delta}(z_w, a^{2 \cdot 2}) = z_w$
- Für $s = 3$ gilt $\widetilde{\delta}(z_0, a^{6^2}) = \widetilde{\delta}(z_0, a^{12+24}) = \widetilde{\delta}(z_w, a^{8 \cdot 3}) = z_w$
- Für $s = 4$ gilt $\widetilde{\delta}(z_0, a^{4^2}) = \widetilde{\delta}(z_0, a^{12+4}) = \widetilde{\delta}(z_w, a^{1 \cdot 4}) = z_w$

In allen Fällen gibt es also ein $n \in \mathbb{N}_{>0}$ mit $\widetilde{\delta}(z_0, a^{n^2}) = z_w$, sodass z_w ein Endzustand sein muss. Damit ist $a^{12} \in L(A)$. (Übrigens ist 12 die kleinste Nicht-Quadratzahl mit dieser Eigenschaft.)

FSK2-4 ***Grammatik-Konkatenation*** (0 Punkte)

Seien G und G' Typ i-Grammatiken (für $i \in \{0, \ldots, 3\}$), sodass $\varepsilon \notin L(G)$ und $\varepsilon \notin L(G')$.

Zeigen oder widerlegen Sie für alle i: Es gibt eine Grammatik G'' vom Typ i, sodass $L(G'') = L(G)L(G')$.

LÖSUNGSVORSCHLAG:
Ja, dies gilt für alle i.
Vorbereitungen: Sei $G = (V, \Sigma, P, S)$ und $G' = (V', \Sigma', P', S')$.
Um zu erreichen, dass V disjunkt zu V' und Σ' ist, führe für jedes $A \in V \cap (V' \cup \Sigma')$ eine neue Variable A' ein und ersetze A in G durch A' (sowohl in V als auch in P und eventuell in S). Führe analog für jedes $A \in V' \cap \Sigma$ eine neue Variable A' ein und ersetze A in G' durch A'.
Falls Σ und Σ' nicht disjunkt sind, führe für jedes $a \in \Sigma \cap \Sigma'$, welches auf der linken Seite einer Produktionsregel von P vorkommt, eine Variable V_a ein, ersetze jedes a in P durch V_a und füge die Produktionsregel $V_a \rightarrow a$ zu P hinzu. Führe analog für jedes $a \in \Sigma \cap \Sigma'$, welches auf der linken Seite einer Produktionsregel von P' vorkommt, eine Variable V'_a ein. Dies ändert Typ 2- und Typ 3-Grammatiken nicht, da dort keine Terminale auf linken Seiten vorkommen.
Beide Vorbereitungen ändern die erzeugten Sprachen $L(G)$ und $L(G')$ nicht.
Konstruktion von $G'' = (V'', \Sigma'', P'', S'')$:

- Für Typ 0, 1, 2:
  - Sei S'' eine neue Variable (d.h. $S'' \notin V \cup V'$)
  - $V'' = V \cup V' \cup \{S''\}$
  - $\Sigma'' = \Sigma \cup \Sigma'$
  - $P'' = P \cup P' \cup \{S'' \rightarrow SS'\}$
- Für Typ 3 verwende entweder die Abschlusseigenschaften regulärer Sprachen oder:
  - $S'' = S$
  - $V'' = V \cup V'$
  - $\Sigma'' = \Sigma \cup \Sigma'$
  - $P'' = P_n \cup P_c \cup P'$
    wobei $P_n = \{A \rightarrow aB \mid (A \rightarrow aB) \in P\}$
    und $P_c = \{A \rightarrow aS' \mid (A \rightarrow a) \in P\}$.