

Einführung in die theoretische Informatik
Sommersemester 2017 – Abgabebblatt Lösungsskizze 1

Abgabebblatt

Wir unterscheiden zwischen Übungs- und Abgabebblättern. Mit diesem *Abgabebblatt* fragen wir eine *Auswahl* an Kernaspekten ab, die auf den Trainingsblättern 1 bis 4 zu finden sind. Auch die Kernaspekte, die sich nicht auf Abgabebblättern finden, sind für die Klausur relevant. Wenn Sie die Abgabebblätter regelmäßig abgeben, können Sie zusätzlich einen *Bonus* von 0,3 (nach Bestehen) für beide Klausuren erreichen. Die genauen Regularien, wann Sie einen Notenbonus erhalten und wann nicht, finden Sie auf der Vorlesungshomepage.

AUFGABE 1.1. (*Übersetzungen*)

8 Punkte

Sei $\Sigma = \{a, b\}$ und $L := \{w \mid (w_1 = a \wedge \forall 3 \leq i \leq |w|. w_i = a) \vee (w_1 = w_2 = a \wedge \forall 3 \leq i \leq |w|. w_i = b)\}$.

- Geben Sie einen regulären Ausdruck r an, so dass $L(r) = L$.
- Geben Sie einen ε -NFA N gemäß Vorlesung an, so dass $L(r) = L(N)$.
- Geben Sie einen DFA D mithilfe der Potenzmengenkonstruktion an, so dass $L(D) = L(N)$.
- Geben Sie eine rechtlineare Grammatik G gemäß Vorlesung an, so dass $L(G) = L(D)$.

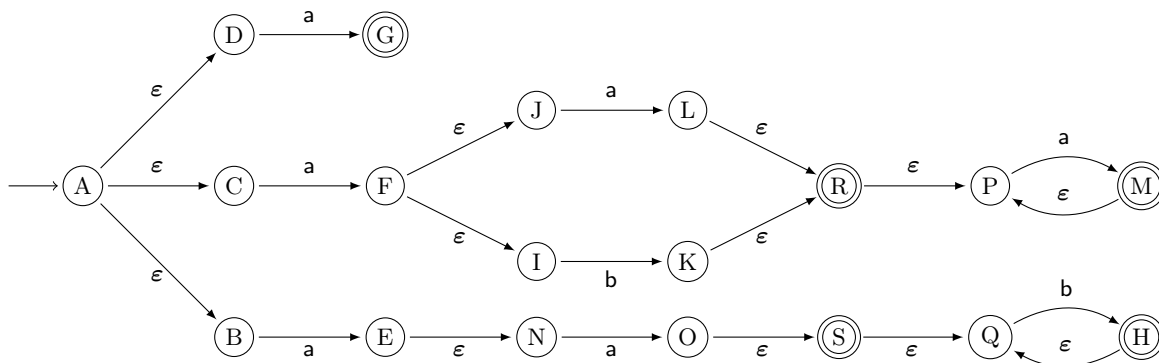
Hinweise:

- Sie bekommen für eine Teilaufgabe nur dann Punkte, wenn Ihr Automat, Ihr Ausdruck bzw. Ihre Grammatik die folgenden Wörter korrekt erkennt:
 - *nicht* enthalten: ε , $abab$, $aaaab$
 - enthalten: a , aa , $aaaa$
- Sie dürfen von den in der Vorlesung beschriebenen Konstruktionsverfahren abweichen, *sofern* Sie diese Änderungen genau dokumentieren und deren Korrektheit begründen.

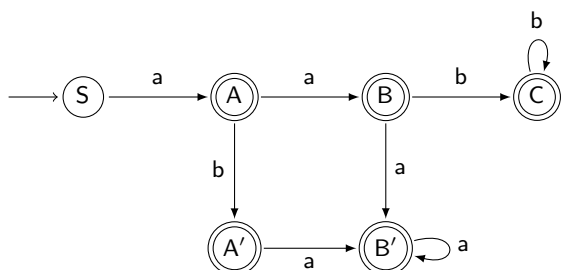
Lösungsskizze

Punkteverteilung: 2P je Teilaufgabe.

- $a \mid a(a \mid b)a^* \mid aab^*$. Alternativ: $a(\varepsilon \mid a \mid b)a^* \mid aab^*$
- Der ε -Automat zum regulären Ausdruck gemäß Vorlesungsschema:



- Mithilfe der Potenzmengenkonstruktion für ε -Automaten von Übungsblatt 3 erhält man:



Alle nicht eingezeichneten Kanten, die aber um eine totale Transitionsrelation zu erhalten, notwendig sind, führen zu einem Fehlerzustand. Um die Darstellung der rechtslinearen Grammatik übersichtlicher zu gestalten, benennen wir die Zustände der oberen Reihe mit S, A, B, C und die der unteren mit A' und B' . Weiterhin haben wir von A drei anstatt zwei ε -Kanten, da wir die Konstruktion von binärem $|$ auf ternäres $|$ erweitert haben.

- Die Grammatik $G = (\{S, A, B, C, A', B'\}, \Sigma, P, S)$ hat die folgenden Produktionen:

- $S \rightarrow a \mid aA$
- $A \rightarrow a \mid b \mid aB \mid bA'$
- $A' \rightarrow a \mid aB'$
- $B \rightarrow a \mid b \mid aB' \mid bC$
- $B' \rightarrow a \mid aB'$
- $C \rightarrow b \mid bC$

AUFGABE 1.2. (Abschlusseigenschaften)

5 Punkte

Sei Σ ein Alphabet und $L_1, L_2 \subseteq \Sigma^*$. Wir definieren:

$$\text{REPEAT}(L_1, L_2) := \{w^{(1)} \dots w^{(n)} \mid n \in \mathbb{N} \wedge (\forall 0 < i < n. (w^{(i)} \in L_1 \wedge w^{(i)} \notin L_2)) \wedge w^{(n)} \in L_2\}$$

Die Sprache $\text{REPEAT}(L_1, L_2)$ enthält alle Wörter, die *solange* Wörter aus der Sprache L_1 konkateniert bis ein Wort aus der Sprache L_2 auftritt. Beweisen Sie die folgende Aussage

Wenn L_1 und L_2 regulär sind, dann ist auch $\text{REPEAT}(L_1, L_2)$ regulär.

Lösungsskizze

Punkteverteilung: 2.5P für Umformung, 2.5P für die Begründung, dass reguläre Sprache unter den verwendeten Operation abgeschlossen sind.

Wir beweisen zunächst, dass $\text{REPEAT}(L_1, L_2) = (L_1 \cap \overline{L_2})^* L_2$. Dazu

$$\begin{aligned} w &\in \text{REPEAT}(L_1, L_2) \\ \Leftrightarrow w &= w^{(1)} \dots w^{(n)} \wedge (\forall 0 < i < n. (w^{(i)} \in L_1 \wedge w^{(i)} \notin L_2)) \wedge w^{(n)} \in L_2 \\ \Leftrightarrow w &= w^{(1)} \dots w^{(n)} \wedge (\forall 0 < i < n. (w^{(i)} \in L_1 \cap \overline{L_2})) \wedge w^{(n)} \in L_2 \\ \Leftrightarrow w &= w^{(1)} \dots w^{(n)} \wedge w^{(1)} \dots w^{(n-1)} \in (L_1 \cap \overline{L_2})^* \wedge w^{(n)} \in L_2 \\ \Leftrightarrow w &\in (L_1 \cap \overline{L_2})^* L_2 \end{aligned}$$

Da reguläre Sprachen unter Schnitt, Komplement, Stern und Konkatenation abgeschlossen sind, muss $\text{REPEAT}(L_1, L_2)$ regulär sein, wenn L_1 und L_2 regulär sind.

Definition

Sei Σ ein endliches Alphabet und $X \subseteq 2^\Sigma$.

- Sei $w \in \Sigma^*$. Wir definieren $\ell(w) := \{x \mid |w|_x > 0\}$. $\ell(w)$ ist die Menge aller im Wort w auftretenden Buchstaben.
- Wir definieren die Sprache $L_X := \{w \in \Sigma^* \mid \ell(w) \in X\}$.

Beispiele:

- $\Sigma^* = L_{2^\Sigma}$
- Die Sprache aus Aufgabe 2.4 von Übungsblatt 2 über dem Alphabet $\Sigma = \{a, b, c\}$ lässt sich aufschreiben als L_X mit $X = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}\}$.

AUFGABE 1.3. (Regularität)

5 Punkte

Sei Σ ein endliches Alphabet. Zeigen Sie, dass L_X für alle $X \subseteq 2^\Sigma$ regulär ist, indem Sie eine passende Konstruktion für einen DFA D in Abhängigkeit von X angeben und beweisen, dass $L(D) = L_X$

Lösungsskizze

Punkteverteilung: 2P für Konstruktion, 2P für die Induktion, 1P für Umformung.

Wir konstruieren einen DFA $D = (Q, \Sigma, \delta, q_0, F)$ in Abhängigkeit von $X \subseteq 2^\Sigma$ wie folgt:

- $Q := \{q_P \mid P \in 2^\Sigma\}$
- $\delta(q_P, x) := q_{P \cup \{x\}}$ für alle $x \in \Sigma$
- $q_0 := q_\emptyset$
- $F := \{q_P \mid P \in X\}$

Mit Induktion über die Länge des Wortes $w \in \Sigma^*$ können wir zeigen, dass $\hat{\delta}(q_0, w) = q_{\ell(w)}$. Dann gilt

$$w \in L_D \Leftrightarrow \hat{\delta}(q_0, w) \in F \Leftrightarrow q_{\ell(w)} \in F \Leftrightarrow \ell(w) \in X,$$

was zu zeigen war.

AUFGABE 1.4. (Pumping Lemma)

3 Punkte

Sei $\Sigma = \{a, b\}$. Zeigen Sie mithilfe des Pumping Lemmas, dass $L := \{w \in \Sigma^* \mid |w|_a - 2 \cdot |w|_b = 0\}$ nicht regulär ist.

Lösungsskizze

Punkteverteilung: 1P für passendes Wort. 1P für passende Zerlegung. 1P für korrekter Beweis.

Angenommen, L wäre regulär. Dann können wir das Pumping Lemma anwenden.

Sei $n \in \mathbb{N}$ eine Pumping-Lemma-Zahl. Dann gilt $a^{6n}b^{3n} \in L$. Es gibt also für $z = a^{6n}b^{3n}$ eine Zerlegung $z = uvw$ mit $v \neq \varepsilon$ und $|uv| \leq n$, so dass $(*) uv^i w \in L$ für alle $i \in \mathbb{N}_0$. Deshalb $uv = a^k$ für $0 < k \leq n$. Also $u = a^i$ und $v = a^j$ mit $i + j = k$ und $j > 0$.

Dann ist aber $uv^2w = a^i a^{2j} a^{6n-k} b^{3n} \notin L$, denn $i + 2j + 6n - k = 6n + j$ und $6n + j - 2 \cdot 3n \neq 0$ für $j > 0$. Dies steht im Widerspruch zu $(*)$, d.h. unsere ursprüngliche Annahme, dass L regulär ist, ist falsch.

AUFGABE 1.5. (Entscheidungsverfahren)

4 Punkte

Zeigen Sie mithilfe des passenden Entscheidungsalgorithmus aus der Vorlesung, dass die beiden regulären Ausdrücke $(1|10)^*$ und $1^*(101^*)^*$ die gleiche Sprache akzeptieren.

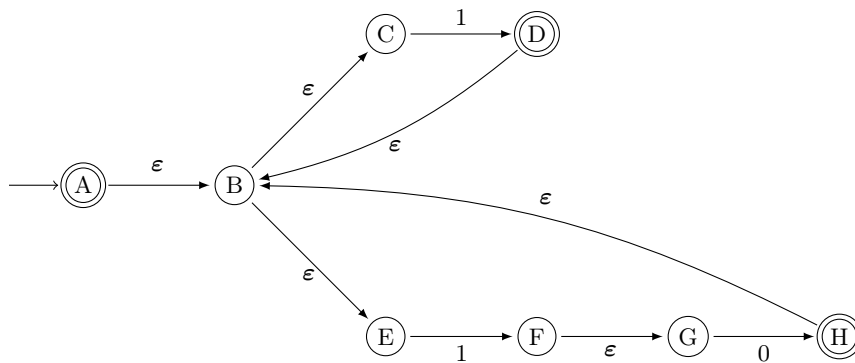
Hinweis: Sie dürfen von den in der Vorlesung beschriebenen Konstruktionsverfahren abweichen, *sofern* Sie diese Änderungen genau dokumentieren und deren Korrektheit begründen.

Lösungsskizze

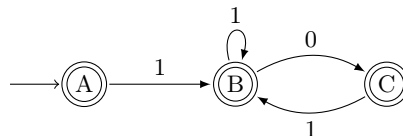
Punkteverteilung: jeweils 1P für korrekte Automaten für die RE. 1P für korrekten Produktautomaten. 1P für korrekte Analyse des Automaten.

Um das Verfahren aus der Vorlesung anzuwenden, müssen wir die regulären Ausdrücke zunächst in DFAs übersetzen. Dazu übersetzen wir die regulären Ausdrücke zunächst gemäß Vorlesung in ε -NFAs, dann mithilfe des Algorithmus von Übungsblatt 3 determinisiert und schließlich beide Schnitte bildet.

Für den regulären Ausdruck $(1|10)^*$ ergibt sich streng nach Vorlesungsalgorithmus:

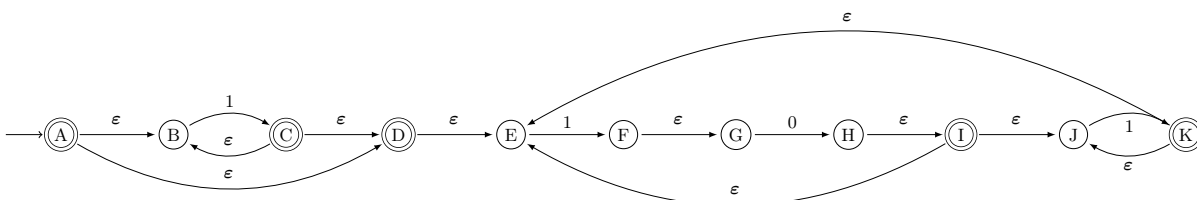


Determinisieren liefert:

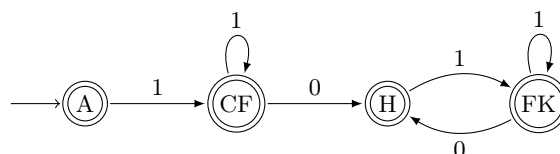


Alle nicht eingezeichneten Kanten, die aber um eine totale Transitionsrelation zu erhalten, notwendig sind, führen zu einem Fehlerzustand.

Für den regulären Ausdruck $1^*(101^*)^*$ ergibt sich streng nach Vorlesungsalgorithmus:

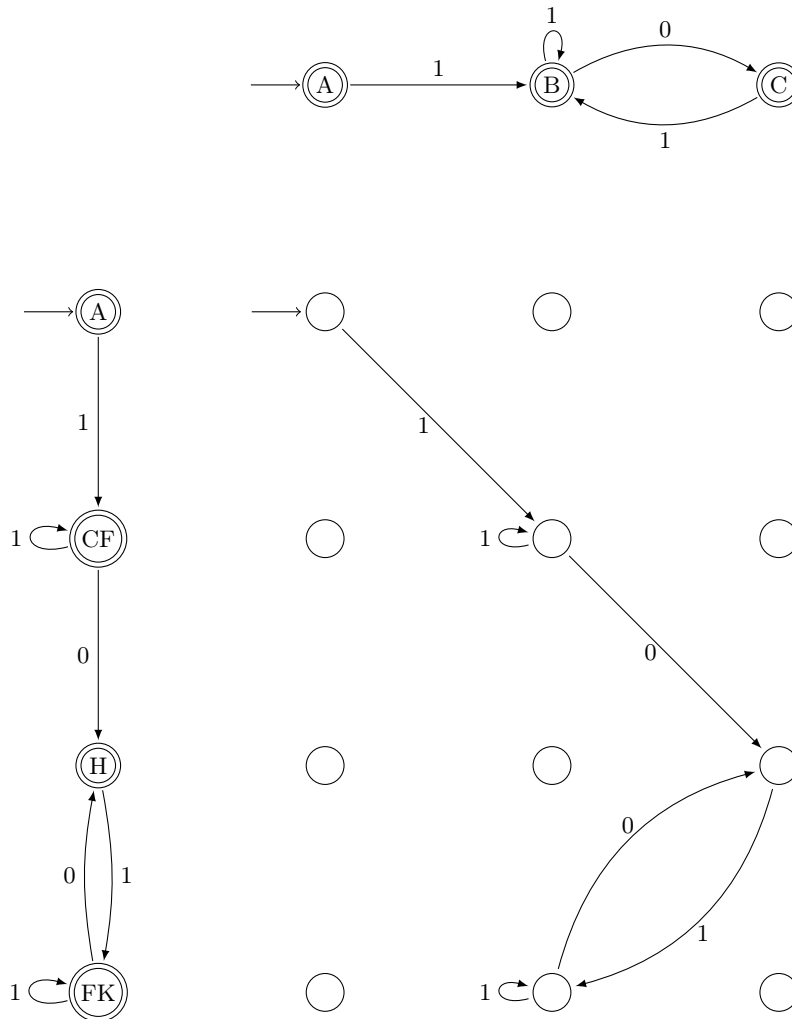


Determinisieren liefert:



Alle nicht eingezeichneten Kanten, die aber um eine totale Transitionsrelation zu erhalten, notwendig sind, führen zu einem Fehlerzustand.

Der Produktautomat der beiden DFAs ergibt sich wie folgt. Alle nicht eingezeichneten Kanten, die aber um eine totale Transitionsrelation zu erhalten, notwendig sind, führen zu einem Fehlerzustand.



Die Akzeptanzbedingung ist $w \in L((1 \mid 10)^*) \Delta L(1^*(101^*)^*)$ (symmetrische Differenz, siehe Übungsblatt 1). Das heißt genau die Zustände im Produktautomat sind akzeptierend, die aus einem nicht-akzeptierenden und einem akzeptierenden Zustand der beiden ursprünglichen Automaten bestehen. Da alle gezeichneten Zustände Paare von akzeptierenden Zuständen der beiden Automaten sind und der Fehlerzustand von beiden synchron erreicht wird, ist die Menge der akzeptierenden Zustände für die Produktkonstruktion leer. Damit ist die Sprache leer und die beiden regulären Ausdrücke beschreiben die gleiche Sprache.