

Einführung in die theoretische Informatik
Sommersemester 2019 – Übungsblatt Lösungsskizze 13

Klausurvorbereitung

Dieses Übungsblatt besteht aus einer Auswahl von Klausuraufgaben aus vorherigen Jahren und soll der Klausurvorbereitung dienen. Da in der Übung vermutlich nicht alle Aufgaben besprochen werden können, empfehlen wir den Rest zu Hause nachzuarbeiten.

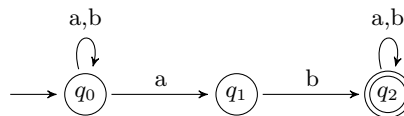
AUFGABE 13.1. (*Endliche Automaten (1)*)

Sei $r = (a|b)^*ab(a|b)^*$ ein regulärer Ausdruck über dem Alphabet $\Sigma = \{a, b\}$.

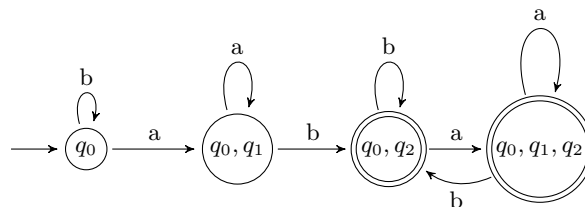
- (a) Konstruieren Sie einen NFA A mit $L(A) = L(r)$.
- (b) Konstruieren Sie einen DFA B mit $L(B) = L(A)$.
- (c) Konstruieren Sie den minimalen DFA C mit $L(C) = L(B)$.
- (d) Konstruieren Sie den minimalen DFA D mit $L(D) = \Sigma^* \setminus L(C)$.
- (e) Konstruieren Sie einen regulären Ausdruck r' mit $L(r') = L(D)$. Verwenden Sie hierzu Ardens Lemma.

Lösungsskizze

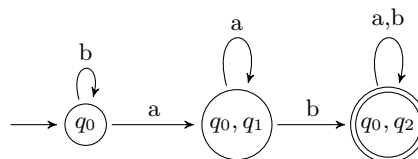
- (a) NFA A bzw. gleich B oder C , da DFA auch NFA:



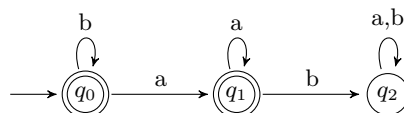
- (b) DFA B bzw. gleich DFA C aus nächstem Schritt (ist ja offensichtlich):



- (c) DFA C :



- (d) DFA D :



- (e) Aufstellen des LSGs und lösen:

$$\begin{aligned} r_{q_0} &\equiv ar_{q_1} \mid br_{q_0} \mid \varepsilon \\ r_{q_1} &\equiv ar_{q_1} \mid br_{q_2} \mid \varepsilon \\ r_{q_2} &\equiv (a|b)r_{q_2} \end{aligned}$$

Endergebnis: $r' = r_{q_0} = b^*a^*$

AUFGABE 13.2. (Kontextfreie Sprachen (1))

Sei $\Sigma = \{a, b, c\}$.

- (a) Geben Sie für die Sprache $L = \{(aba)^n b^m (ba)^n\}$ kontextfreie Grammatik G an, so dass $L(G) = L$ gilt.
- (b) Sei $G = (\{S\}, \Sigma, \{S \rightarrow \varepsilon \mid aSc \mid aSbSc\}, S)$ eine kontextfreie Grammatik. Zeigen Sie, dass für alle in G ableitbaren Wörter w die Eigenschaft $|w|_a = |w|_c$ gilt.

Lösungsskizze

- (a) $G = (\{S, X, B\}, \Sigma, P, S)$ mit $P: S \rightarrow Xb \quad X \rightarrow abaXba \mid B \quad B \rightarrow bB \mid \varepsilon$
- (b) Sei $u \in L(G)$. Wir zeigen $|u|_a = |u|_c$ mit Induktion über die Erzeugung von u .
 $\varepsilon: |\varepsilon|_a = 0 = |\varepsilon|_c$.
 aSc : Sei $|w|_a = |w|_c$. Dann gilt $|awc|_a = 1 + |w|_a = 1 + |w|_c = |awc|_c$.
 $aSbSc$: Sei $|w|_a = |w|_c$ und $|v|_a = |v|_c$. Dann gilt $|avbwc|_a = 1 + |v|_a + |w|_a = 1 + |v|_c + |w|_c = |avbwc|_c$.

AUFGABE 13.3. (Kontextfreie Sprachen (2))

- (a) Zeigen Sie, dass folgende Sprache $L = \{a^m b^n \mid m \geq n \geq 0\}$ **deterministisch** kontextfrei ist, indem Sie einen deterministischen Kellerautomaten (DPDA) angeben, der L auf Endzustand akzeptiert. Ihr DPDA darf maximal vier Zustände besitzen.
- (b) Gibt es einen DPDA, der L auf leeren Keller akzeptiert?

Lösungsskizze

- (a) DPDA mit Initialkonfiguration qX und $F = \{q, p\}$:

$$qX \xrightarrow{a} qAX \quad qA \xrightarrow{a} qAA \quad qA \xrightarrow{b} p\varepsilon \quad pA \xrightarrow{b} p\varepsilon$$

X dient als Keller-Ende-Symbol.

Phase I (Zustand q): DPDA merkt sich, solange nur a s gelesen werden, die Anzahl der a s auf dem Stack.

Phase II (Zustand p): sobald das erste b gelesen wird, reagiert der DPDA nur noch b s, und das nur solange noch ein A auf dem Stack liegt.

Damit kann eine Eingabe nur vollständig gelesen werden, falls sie von der Form $a^m b^n$ mit $m \geq n \geq 0$ ist.

Da beide Zustände akzeptierend sind, wird somit genau die angegebene Sprache akzeptiert.

- (b) Nein, da L nicht präfixfrei ist (siehe Vorlesung).

AUFGABE 13.4. (Pumping Lemma)

Sei $\Sigma = \{a, b, c\}$. Zeigen Sie, dass die Sprache $L = \{wc^n w^R \mid w \in \{a, b\}^n \wedge n \in \mathbb{N}_0\}$ nicht kontextfrei ist. Führen Sie einen Widerspruchsbeweis unter Verwendung des Pumping Lemmas für kontextfreie Sprachen.

Lösungsskizze

S Angenommen, L wäre kontextfrei. Dann können wir das PL für kontextfreie Sprachen anwenden.

S Sei $n \in \mathbb{N}$ eine PL-Zahl (für L).

- Wähle $z = a^n c^n a^n$. Offensichtlich ist $|z| \geq n$ und $z \in L$.

S Nach PL gibt es eine Zerlegung $z = uvwxy$ mit 1. $vx \neq \varepsilon$, 2. $|vwx| \leq n$ und 3. $\forall i \in \mathbb{N}_0. uv^i w x^i y \in L$.

- Sei $z' = uv^2wx^2y$. Da wegen (2) nur maximal einer der beiden a -Blöcke gepumpt werden kann, gilt, dass $z' = a^n cz''$ oder $z' = z''ca^n$. Da mindestens einer der beiden a -Blöcke noch Länge n hat, muss $|z'| = 3n$ gelten. Es ist aber wegen (1) auch $|z'| > 3n$. Dies ist ein Widerspruch zu (3).

Alternativ: Fallunterscheidung über $|z|_a > 0$ oder $|z|_c > 0$.

S Also haben wir einen Widerspruch zur Annahme hergeleitet und somit ist L nicht kontextfrei.

AUFGABE 13.5. (Entscheidbarkeit)

Betrachten Sie die Menge $A = \{w \in \{0, 1\}^* \mid \exists M. enc(M) = w \wedge |L_H(M)| \leq 42\}$.

- (a) Beweisen Sie, dass die Menge nicht semi-entscheidbar ist, indem Sie $H_0 \leq A$ zeigen.
- (b) Geben Sie einen Semientscheidungsalgorithmus für \bar{A} an.

Hinweis: Verwenden Sie **nicht** den Satz von Rice. Verwenden Sie für diese Aufgabe die folgende Definition des Halteproblems auf dem leeren Band $H_0 := \{w \in \{0, 1\}^* \mid \exists M. w = enc(M) \wedge M[\varepsilon] \downarrow\}$.

Lösungsskizze

- (a) A ist die Menge der Wörter, die eine Turing Maschine encodieren, die auf maximal 42 Eingaben hält. Wir zeigen $\overline{H_0} \leq A$ mittels folgender Reduktion:
- Gegeben ein Wort w , das keine TM encodiert, gib die TM M_\perp zurück, die auf keinem Input hält.
 - Gegeben ein Wort w , das eine TM M encodiert, gib die TM M' zurück, die zuerst das Band löscht und dann M simuliert.

Berechenbarkeit: Diese Reduktion ist berechenbar, da nach VL Löschen des Bandes und Simulation einer TM berechenbar sind.

Korrektheit: Fallunterscheidung

- w keine TM: In diesem Fall wird die TM M_\perp zurück gegeben, die auf keinem Input hält. Es gilt $|L_H(M_\perp)| = 0 \leq 42$ und $M_\perp \in A$. Dies ist korrekt, da wenn w keine TM ist gilt, dass $w \in \overline{H_0}$.
 - $w \in H_0, w = enc(M)$: M hält nicht auf dem leeren Band, folglich hält M' auf keinem Input, da es immer das Band löscht und M auf dem leeren Band simuliert. Damit ist $|L_H(M')| = 0 \leq 42$ und $M' \in A$.
 - $w \notin \overline{H_0}, w = enc(M)$: $w \in H_0 \implies M[\varepsilon] \downarrow$. Da M auf dem leeren Band hält, hält M' für jeden Input, da es immer M auf dem leeren Band simuliert. Damit ist $|L_H(M')| = \infty > 42$ und $M' \notin A$.
- (b) \overline{A} ist die Menge der Wörter, die keine TM encodieren oder die eine TM encodieren die auf mindestens 43 Eingaben hält. Semientscheidungsalgorithmus: Wenn w keine TM encodiert, gib 1 aus. Wenn w eine TM M encodiert, simuliere M mittels Diagonalisierung auf allen Inputs. Sobald 43 der Simulationen halten, gib 1 aus.

AUFGABE 13.6. (Reduktionen)

Mit SUBSET bezeichnen wir das Problem:

Gegeben: Ein ε -NFA N mit n Zuständen über dem Alphabet $\Sigma = \{0, 1\}$ und eine Zahl $m \leq n$.

Frage: Gibt es ein Wort w der Länge m , so dass N dieses Wort nicht akzeptiert?

...und in Mengenschreibweise:

$$\text{SUBSET} := \{(N, m) \mid N = (Q, \Sigma, \delta, q_0, F) \text{ ist } \varepsilon\text{-NFA} \wedge m \leq |Q| \wedge L(N) \cap \Sigma^m \neq \Sigma^m\}$$

- (a) Zeigen Sie, dass SUBSET NP-vollständig ist. Verwenden Sie hierbei eine geeignete Reduktion von SAT auf SUBSET.
- (b) Wenden Sie Ihre Reduktion auf $(x_1 \vee \neg x_3 \vee x_4) \wedge (\neg x_1 \vee x_2 \vee x_3)$ an. Geben Sie den konstruierten ε -NFA graphisch an.

Hinweis: Betrachten Sie die Menge aller nichterfüllenden Belegungen für die Formel F , die genau m Variablen hat, als Wörter der Länge m .

Lösungsskizze

- (a) Da aus der Vorlesung $\text{SAT} \leq_p \text{3KNF-SAT}$ bekannt ist, zeigen wir $\text{3KNF-SAT} \leq_p \text{SUBSET}$.

- Idee:
Der ε -NFA N erkennt alle nicht erfüllenden Belegungen von F .
- Reduktion:
Sei $F = \bigwedge_{i=1}^k C_i$ eine Formel in 3-KNF mit m Variablen (x_1, x_2, \dots, x_m) und mit $C_i = \bigvee_{j=1}^3 L_{ij}$ für jedes i .
Der ε -NFA wird dann wie folgt definiert $N = (\{q_0\} \cup \{q_{ij} \mid i \in [1; k] \wedge j \in [0; m]\}, \{0, 1\}, \delta, q_0, \{q_{im} \mid i \in [1; k]\})$ mit δ :

$$\begin{aligned} \delta = & \{(q_0, \varepsilon, q_{i0}) \mid i \in [1; k]\} \\ & \cup \{(q_{i(k-1)}, 0, q_k) \mid k \in [1, m] \wedge \neg x_k \notin \{L_{i1}, L_{i2}, L_{i3}\}\} \\ & \cup \{(q_{i(k-1)}, 1, q_k) \mid k \in [1, m] \wedge x_k \notin \{L_{i1}, L_{i2}, L_{i3}\}\} \end{aligned}$$

Die Reduktion bildet dann F auf (N, m) ab.

- Korrektheit:
 F ist erfüllbar gdw. nicht alle Belegungen nicht erfüllend sind. Sei $\sigma(F) = 0$ eine nicht erfüllende Belegung. Dann existiert insbesondere eine Klausel C_i mit $\sigma(C_i) = \sigma(L_{i1}) + \sigma(L_{i2}) + \sigma(L_{i3}) = 0$. Das Wort $w = \sigma(x_1)\sigma(x_2)\dots\sigma(x_m)$ wird dann von q_{i0} aus akzeptiert und es gilt somit $w \in L(N)$. Der Beweis ist analog für $\sigma(F) = 1$
- Polynomielle Reduzierbarkeit:
Der konstruierte ε -NFA hat $\mathcal{O}(km)$ Zuständen und ein konstantes Alphabet und hat somit maximal Größe polynomiell in der Größe von F . Er kann auch in polynomieller Zeit berechnet werden, da jede Komponente während des Lesens einer Klausel C direkt konstruiert werden kann.

- **SUBSET \in NP:**

Rate $w \in \Sigma^m$ (dies ist möglich, da m kleiner als N ist). Übersetze den ϵ -NFA in einen NFA N' und prüfe $w \notin L(N')$ mithilfe des polynomiellen Algorithmus für das Wortproblem (auf NFAs) der Vorlesung.

(b) $(N, 4)$ mit N :

