Prof. Javier Esparza
Philipp Czerner, Martin Helfrich

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2021 – Übungsblatt 13

- Dieses Übungsblatt besteht aus einer Auswahl von Klausuraufgaben aus vorherigen Jahren und beeinhaltet keinen neuen Stoff aus der Vorlesung. Verwenden Sie die letzte Übung für allgemeine Fragen zum Vorlesungsstoff in Hinblick auf die Klausurvorbereitung. (siehe Aufgabe Ü13.1)
- Das Übungsblatt ist in zwei Teile gegliedert: den Vorbereitungsteil, den Sie vor der Übung selbstständig bearbeiten sollen, und den Übungs-/Nachbereitungsteil, der Aufgaben enthält, die in der Übung besprochen werden und Aufgaben, die von Ihnen anschließend zur Nachbereitung verwendet werden können. Insbesondere werden nicht alle Aufgaben aus dem Übungs-/Nachbereitungsteil in der Übung besprochen.
- Das ist nicht das Hausaufgabenblatt! Die Hausaufgaben finden Sie auf einem separaten Blatt.

Vorbereitung (→ vor der Übung selbständig zu bearbeiten)

Individualaufgabe Ü13.1. (*letzte Übung ⇒ (letzte) Chance für Fragen*)

Diese Woche findet die letzte Übung statt. Bitte lassen Sie deshalb Ihrem Tutor rechtzeitig (also so früh wie möglich) Fragen auf Zulip zukommen, die sie noch haben. Z.B. könnten das allgemeine Fragen zu Konzepten wie *Reduktion* oder Techniken wie *Pumping Lemma* sein oder aber auch Fragen zu Übungsaufgaben oder Hausaufgaben, die Sie gerne besprechen würden. Bitte geben Sie in solchen Fällen immer das Kürzel der Aufgabe UND die konkrete Frage an.

Bitte geben Sie Ihrem Tutor genügen Zeit, sich auf die Fragen vorzubereiten. Falls Ihr Tutor es nicht mehr schaffen sollte, sich vorzubereiten, oder Sie nach dem letzten Tutorium noch Fragen haben, finden Sie natürlich auf Zulip noch Unterstützung.

Individualaufgabe Ü13.2. (*Kahoot*)

Falls Sie das Kahoot aus der Vorlesung verpasst haben: Spielen Sie es jetzt! Link[1] Es kann hilfreich sein, den passenden Stream (Nr. 24) nebenbei anzuschauen. Dann können Sie sich mit Ihren Mitstudenten/-innen messen und bekommen die Lösungen vom Professor erklärt!

Individualaufgabe Ü13.3. (*Automata Tutor*)

Wenn Sie Automata Tutor noch nicht verwendet haben, folgen Sie erst den Schritten in Ü1.2, um sich richtig zu registrieren.

Wir haben zusätzlich zu den Übungsaufgaben der letzten Wochen auch die Hausaufgaben, die auf AT bearbeitet wurden, frei zugänglich gemacht. Somit haben sie ab jetzt die Möglichkeit alle AT-Aufgaben zu üben, ohne das die Anzahl der Versuche eingeschränkt ist. Somit können Sie AT auch sehr gut zur Klausurvorbereitung verwenden.

[1] Falls der Link nicht mehr funktioniert, teilen Sie dies bitte der Übungsleitung mit. Die Teilnehmeranzahl ist leider auf 2000 begrenzt.

Außerdem wollen wir nochmal auf die Möglichkeit hinweisen, dass sie unter `Home > My Autogenerated Problems` für einige der Aufgabentypen selbständig neue Aufgaben generieren können.

Übung und Nachbereitung

Übungsaufgabe Ü13.4. (*Endliche Automaten*)

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 $L(B) = L(A)$.

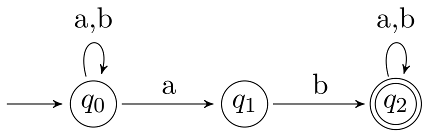
(c) Konstruieren Sie den minimalen DFA C mit $L(C) = L(B)$.

(d) Konstruieren Sie den minimalen DFA D mit $L(D) = \Sigma^* \setminus L(C)$.

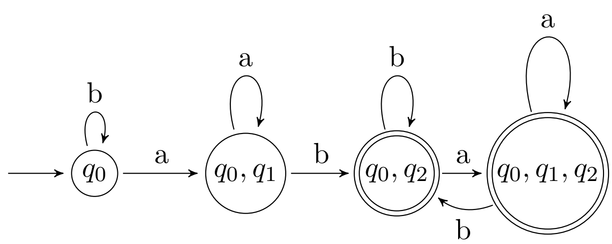
(e) Konstruieren Sie einen regulären Ausdruck r' mit $L(r') = L(D)$. Verwenden Sie hierzu Ardens Lemma.

Lösungsskizze.

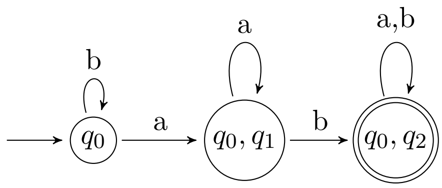
(a) NFA A bzw. gleich B oder C, da DFA auch NFA:



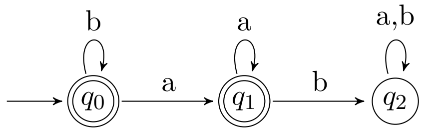
(b) DFA B bzw. gleich DFA C aus nächstem Schritt (ist ja offensichtlich):



(c) DFA C:



(d) DFA D:



(e) Aufstellen des LSGs und lösen:

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

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

Übungsaufgabe Ü13.5. (*Kontextfreie Sprachen (1)*)

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

(a) Geben Sie für die Sprache $L = \{(aba)^n b^m (ba)^n b\}$ eine kontextfreie Grammatik G an, sodass $L(G) = L$ gilt.

(b) Sei $G = (\{S\}, \Sigma, \{S \to \varepsilon \mid aSc \mid aSbSc\}, S)$ 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: $S \to Xb \qquad X \to abaXba \mid B \qquad B \to bB \mid \varepsilon$

(b) Sei $u \in L(G)$. Wir zeigen $|w|_a = |w|_c$ mit Induktion über die Erzeugung von u.

ε: $|\varepsilon|_a = 0 = |\varepsilon|_c$.

aSc: Sei $|w|_a = |w|_c$. Dann gilt $|awc|_a = 1 + |w|_a = 1 + |w|_c = |awc|_c$.

$aSbSv$: Sei $|w|_a = |w|_c$ und $|v|_a = |v|_c$. Dann gilt $|avbwc|_a = 1 + |v|_a + |w|_a = 1 + |v|_c + |w|_c = |avbwc|_c$.

Übungsaufgabe Ü13.6. (*Kontextfreie Sprachen (2)*)

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.

Lösungsskizze. DPDA mit Initialkonfiguration qX und $F = \{q, p\}$:

$$qX \xrightarrow{a} qAX \qquad qA \xrightarrow{a} qAA \qquad qA \xrightarrow{b} p\epsilon \qquad pA \xrightarrow{b} p\epsilon$$

X dient als Keller-Ende-Symbol.

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

Phase II (Zustand p): sobald das erste b gelesen wird, reagiert der DPDA nur noch bs, 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.

Übungsaufgabe Ü13.7. (*Pumping Lemma*)

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

Lösungsskizze.

- Angenommen, L wäre kontextfrei. Dann können wir das PL für kontextfreie Sprachen anwenden.
- Sei $n \in \mathbb{N}_+$ eine PL-Zahl für L.
- Wähle $z = a^n c^n a^n$. Es gilt $|z| = 3n \geq n$ und $z \in L$.
- Nach PL gibt es eine Zerlegung $z = uvwxy$ mit 1. $vx \neq \varepsilon$, 2. $|vwx| \leq n$ und 3. $\forall i \in \mathbb{N}_0 . uv^i wx^i y \in L$.
- 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 $|vx|_a > 0$ oder $|vx|_c > 0$.
- Also haben wir einen Widerspruch zur Annahme hergeleitet und somit ist L nicht kontextfrei.

Übungsaufgabe Ü13.8. (*Entscheidbarkeit*)

Betrachten Sie die Menge $A = \{w \in \{0,1\}^* \mid |L(M_w)| \leq 42\}$.

(a) Beweisen Sie, dass die Menge nicht semi-entscheidbar ist, indem Sie $\overline{H_0} \leq A$ zeigen.

(b) Geben Sie einen Semientscheidungsalgorithmus für $\overline{A}$ an.

Lösungsskizze.

(a) A ist die Menge der Wörter, deren korrespondierenden Turingmaschine auf maximal 42 Eingaben hält. Wir zeigen $\overline{H_0} \leq A$ mittels folgender Reduktion:

Reduktionsfunktion: Gegeben ein Wort w, gib die Kodierung w' einer TM zurück, die zuerst das Band löscht und dann M_w simuliert. Danach geht $M_{w'}$ in einen Endzustand und hält.

Totalität: Die Reduktion gibt für jede Eingabe w ein entsprechendes w' zurück.

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

Korrektheit:

- $w \in \overline{H_0}$: M_w hält nicht auf dem leeren Band, folglich hält $M_{w'}$ auf keinem Input, da es immer das Band löscht und M_w auf dem leeren Band simuliert. Damit ist $|L(M_{w'})| = 0 \leq 42$ und somit $w' \in A$.
- $w \notin \overline{H_0}$: Da M_w auf dem leeren Band hält, akzeptiert $M_{w'}$ jeden Input, da es immer M_w auf dem leeren Band simuliert und dann in einen Endzustand übergeht. Damit ist $|L(M_{w'})| = \infty > 42$ und $w' \notin A$.

(b) $\overline{A}$ ist die Menge der Wörter, die eine TM kodieren, die auf mindestens 43 Eingaben hält. Semientscheidungsalgorithmus: Simuliere M_w mit Dove-Tailing auf allen Inputs. (Also: Simuliere für alle Eingaben der Länge $\leq$1 für 1 Schritt, dann für alle Eingaben der Länge $\leq$2 für 2 Schritte, usw.) Sobald 43 der Simulationen halten, gib 1 aus.

Übungsaufgabe Ü13.9. (*polynomielle Reduktion*)

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, sodass N dieses Wort nicht akzeptiert?

In Mengenschreibweise:

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

(a) Zeigen Sie, dass SUBSET NP-vollständig ist. Verwenden Sie hierbei eine geeignete Reduktion von 3-KNF-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)
- Idee: Der ϵ-NFA N erkennt alle nicht erfüllenden Belegungen von F. Hierfür rät er zunächst die Klausel, welche nicht erfüllt ist und dann die unerfüllende Belegung.
- Reduktion: Sei $F = \bigwedge_{i=1}^{k} C_i$ eine Formel in 3-KNF mit m Variablen $(x_1, x_2, ...x_m)$ und mit $C_i = \bigvee_{j=1}^{l_i} L_{ij}$ für jedes i. Der ϵ-NFA wird dann wie folgt definiert

  $$N = \Big(\{q_0\} \cup \{q_{ij} \mid i \in [1, k] \wedge j \in [0, m]\}, \{0, 1\}, \delta, q_0, \{q_{im} \mid i \in [1, k]\}\Big),$$

  wobei

  $$\begin{aligned}\delta = \{&(q_0, \epsilon, q_{i0}) \mid i \in [1, k]\} \\ \cup \{&(q_{i(k-1)}, 0, q_{ik}) \mid k \in [1, m] \wedge \neg x_k \notin \{L_{i1}, \ldots, L_{il_i}\}\} \\ \cup \{&(q_{i(k-1)}, 1, q_{ik}) \mid k \in [1, m] \wedge x_k \notin \{L_{i1}, \ldots, L_{il_i}\}\}\end{aligned}$$

  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 $w = \sigma(x_1)\sigma(x_2)...\sigma(x_m)$ wird dann von q_{i0} aus akzeptiert und es gilt somit $w \in L(N)$.

  Besteht andererseits ein Lauf in N über q_{i_0}, so erfüllt die Belegung, beschrieben durch den Lauf, kein Literal der Klausel C_i. Damit ist die Belegung unerfüllend für F.
- 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.

- $\mathsf{SUBSET} \in NP$:
  Ein Zertifikat ist ein Wort $w \in \Sigma^m \setminus L(N)$. Da $m \leq |Q|$, ist $|w| = m$ polynomiell bezüglich $|N|$. Das Wortproblem (und damit $w \notin L(N)$) kann nun in polynomieller Zeit gelöst werden (vgl. Vorlesung).

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