Lehrstuhl für Grundlagen der Software-Zuverlässigkeit und der theoretischen Informatik
School of Computation, Information and Technology
Technische Universität München

Esolution
Sticker mit SRID hier einkleben

Hinweise zur Personalisierung:

- Ihre Prüfung wird bei der Anwesenheitskontrolle durch Aufkleben eines Codes personalisiert.
- Dieser enthält lediglich eine fortlaufende Nummer, welche auch auf der Anwesenheitsliste neben dem Unterschriftenfeld vermerkt ist.
- Diese wird als Pseudonym verwendet, um eine eindeutige Zuordnung Ihrer Prüfung zu ermöglichen.

Einführung in die Theoretische Informatik

Klausur:	IN0011 / Endterm	**Datum:**	Samstag, 22. Juli 2023
Prüfer:	Prof. Javier Esparza	**Uhrzeit:**	09:00 – 12:00

	A 1	A 2	A 3	A 4	A 5	A 6	A 7	A 8
I								

Bearbeitungshinweise

- Diese Klausur umfasst **20 Seiten** mit insgesamt **8 Aufgaben**.
  Bitte kontrollieren Sie jetzt, dass Sie eine vollständige Angabe erhalten haben.
- Die Gesamtpunktzahl in dieser Klausur beträgt 100 Punkte.
- Zum Bestehen brauchen Sie 45 Punkte.
- Das Heraustrennen von Seiten aus der Prüfung ist untersagt.
- **Wenn Sie eine Aufgabe nicht im vorgesehenen Platz lösen können, verwenden Sie die zusätzlichen Blätter am Ende der Klausur und kennzeichnen Sie dies deutlich bei der entsprechenden Aufgabe.**
- Als Hilfsmittel sind zugelassen:
  - ein **doppelseitig beschriebenes DIN A4 Merkblatt**
  - ein **analoges Wörterbuch** Deutsch $\leftrightarrow$ Muttersprache **ohne Anmerkungen**
- Mit * gekennzeichnete Teilaufgaben sind ohne Kenntnis der Ergebnisse vorheriger Teilaufgaben lösbar.
- **Es werden nur solche Ergebnisse gewertet, bei denen der Lösungsweg erkennbar ist.** Auch Textaufgaben sind **grundsätzlich zu begründen**, sofern es in der jeweiligen Teilaufgabe nicht ausdrücklich anders vermerkt ist.
- Schreiben Sie weder mit roter / grüner Farbe noch mit Bleistift.
- Schalten Sie alle mitgeführten elektronischen Geräte vollständig aus, verstauen Sie diese in Ihrer Tasche und verschließen Sie diese.
- $\mathbb{N} := \{0, 1, 2, \dots\}$ und $\mathbb{N}_+ := \mathbb{N} \setminus \{0\}$

Hörsaal verlassen von _____ bis _____ / Vorzeitige Abgabe um _____

Aufgabe 1 Quiz Reguläre und Kontextfreie Sprachen (18 Punkte)

Bestimmen Sie für jede der folgenden Aussagen, ob sie wahr oder falsch ist. Falls die Aussage wahr ist, geben Sie eine kurze Begründung an. Andernfalls widerlegen Sie die Aussage, gegebenenfalls mit einem geeigneten Gegenbeispiel und Begründung, dass das Gegenbeispiel korrekt ist.
Wichtig: Antworten ohne Begründung werden nicht bewertet!

0 1 2 3

a)* **Aussage:** Für alle Sprachen A, B, C gilt $(A \setminus C)B = AB \setminus CB$.

Die Aussage ist falsch. Sei $A = \{aa\}$, $B = \{\epsilon, a\}$ und $C = \{aaa\}$. Dann gilt $(A \setminus C)B = \{aa, aaa\}$ und $AB \setminus CB = \{aa\}$.

0 1 2 3

b)* **Aussage:** Wenn $L \subseteq \Sigma^*$ regulär ist und $a \in \Sigma$, dann ist die Sprache aller Wörter aus L, die nicht mit a enden, regulär.

Die Aussage ist wahr. Die Sprache $\Sigma^*\{a\}$ ist regulär, da Σ^* und $\{a\}$ regulär sind und reguläre Sprachen unter Konkatenation abgeschlossen sind. Durch die Abschlusseigenschaften der regulären Sprachen ist damit auch $L \setminus \Sigma^*\{a\}$ regulär, was die gesuchte Sprache ist.

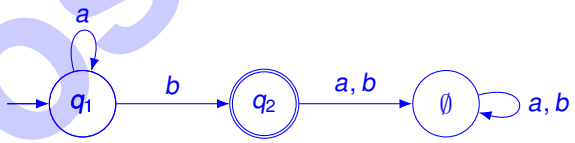
Alternative: Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA für L. Sei $q_f \notin F$. Der NFA $M' := (Q \cup \{q_f\}, \Sigma, \delta', q_0, \{q_f\})$ mit $\delta' := \delta \cup \{(q, x, q_f) \mid \delta(q, x) \in F \wedge x \in (\Sigma \setminus \{a\})\}$ akzeptiert die gewünschte Sprache.

0 1 2 3

c)* **Aussage:** Es gibt einen minimalen DFA M mit maximal 4 Zuständen, sodass die minimale Pumping-Lemma-Zahl für $L(M)$ kleiner als die Anzahl der Zustände von M ist.

Erinnerung: Die Pumping-Lemma-Zahl n für eine Sprache L ist eine Zahl, sodass die Pumping-Lemma-Eigenschaft gilt. Genauer gesagt gibt es für alle $z \in L$ mit $|z| \geq n$ eine Zerlegung $z = uvw$ mit $v \neq \epsilon$, $|uv| \leq n$ und $\forall i \geq 0.\ uv^i w \in L$.

Die Aussage ist wahr. Betrachte den minimalen DFA für die Sprache $\{a^n b \mid n \in \mathbb{N}\}$:



Der Automat hat 3 Zustände, aber 2 ist eine gültige Pumping-Lemma-Zahl: Für jedes Wort $a^n b$ mit $n \geq 1$ definiere $u := \epsilon, v := a, w := a^{n-1}b$. Dann gilt (1) $a^n b = \epsilon a a^{n-1} b = uvw$, (2) $v \neq \epsilon$, (3) $|uv| = 1 \leq 2$ und (4) $uv^i w = a^i a^{n-1} b = a^{n-1+i} b \in L$ für alle $i \geq 0$.

d)* Definition: Analog zu rechtslinearen Grammatiken ist eine Grammatik G *linkslinear*, wenn jede Produktion von G die Gestalt $X \to Ya$ oder $X \to a$ hat.

Aussage: Die Sprache $L(G)$ einer linkslinearen Grammatik G ist regulär.

Hinweis: Sie dürfen verwenden, dass reguläre Sprachen unter Spiegelung abgeschlossen sind, d.h. für jede reguläre Sprache L ist auch L^R regulär.

0 1 2 3

Die Aussage ist wahr. Wir spiegeln zunächst die Produktionen, d.h. wir wandeln jede Produktion der Form $X \to Ya$ in $X \to aY$ um, erhalten wir eine rechtslineare Grammatik G'. Wir haben also $L(G')^R = L(G)$. Da rechtslineare Grammatiken reguläre Sprachen erzeugen und reguläre Sprachen unter Spiegelung abgeschlossen sind, muss auch $L(G)$ regulär sein.

e)* **Aussage:** Wenn die Sprache einer kontextfreien Grammatik regulär ist, dann ist die Grammatik nicht mehrdeutig.

0 1 2 3

Die Aussage ist falsch. Sei $G = (S, \{S \to SS \mid a\})$. Dann ist $L(G) = \{a^n \mid n \in \mathbb{N}_+\}$ regulär. Die Grammatik ist jedoch mehrdeutig, da man aaa sowohl mit $S \to SS \to aS \to aSS \to^2 aaa$ als auch mit $S \to SS \to Sa \to SSa \to^2 aaa$ herleiten kann.

0 1 2 3

f)* **Aussage:** Wenn $A, B \subseteq \Sigma^*$ kontextfrei sind und $\epsilon \notin A$, dann sind alle Lösungen der Gleichung $X = AX \cup B$ kontextfrei.

Sie dürfen das folgende Theorem für diese Aufgabe verwenden: Für alle Sprachen $X, A, B \subseteq \Sigma^*$ mit $\epsilon \notin A$ gilt: $X = AX \cup B \implies X = A^*B$.

Die Aussage ist wahr. Laut dem Theorem ist die einzige Lösung der Gleichung $X = A^*B$. Da A und B CFLs sind und CFLs unter Konkatenation und Stern abgeschlossen sind, muss auch X eine CFL sein.

Aufgabe 2 Vom Regulären Ausdruck zum ϵ-NFA und Zurück (15 Punkte)

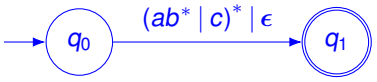
0 1 2 3 4 5

a)*
Gegeben sei der reguläre Ausdruck $r := (ab^* \mid c)^* \mid (b\emptyset)^*$ über dem Alphabet $\Sigma := \{a, b, c\}$. Berechnen Sie mit dem Algorithmus aus der Vorlesung einen ϵ-NFA N, sodass $L(N) = L(r)$ gilt. Geben Sie dabei nach **jeder** Anwendung einer Transformationsregel den Automaten graphisch an.

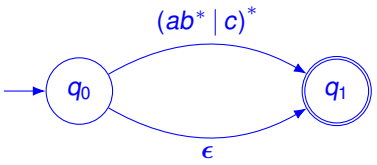
Wir konvertieren den Ausdruck zunächst, um Vorkommen von $\emptyset$ zu entfernen:

$$r = (ab^* \mid c)^* \mid (b\emptyset)^* \equiv (ab^* \mid c)^* \mid \emptyset^* \equiv (ab^* \mid c)^* \mid \epsilon.$$

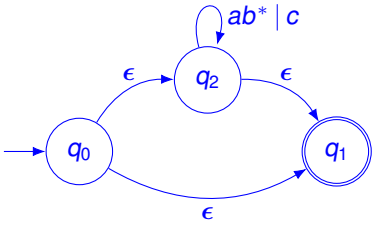
Schritt 1:



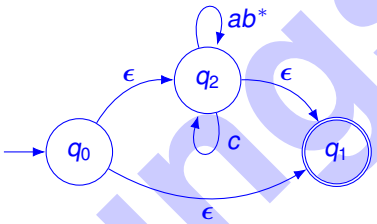
Schritt 2:



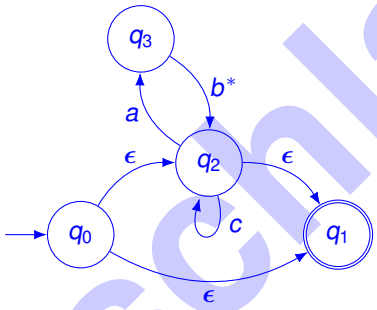
Schritt 3:



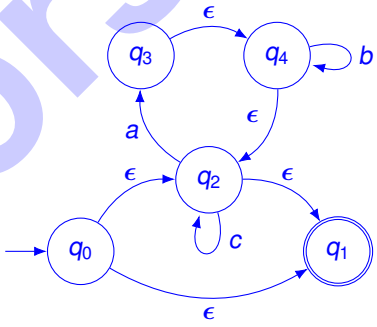
Schritt 4:



Schritt 5:



Schritt 6:



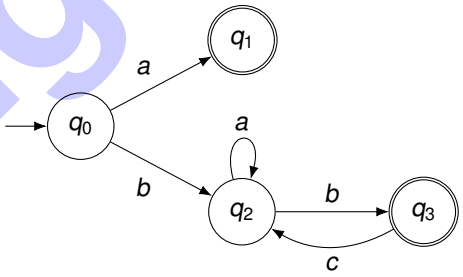
0
1
2
3

b)* Betrachten Sie nun den regulären Ausdruck $s := (0 \mid 1(10)^*(0 \mid 11)(01^*01 \mid 01^*00)^*1)^*$ über dem Alphabet $\Sigma := \{0, 1\}$. Wie viele Zustände hat der resultierende Automat, wenn Sie den Algorithmus ausführen würden? Begründen Sie Ihre Antwort!
Hinweis: Sie müssen den Algorithmus nicht ausführen.

Der Algorithmus startet mit zwei Zuständen und fügt für jede Konkatenation und jeden Stern einen neuen Zustand hinzu. Der Ausdruck enthält 12 Konkatenationen und 5 Sterne. Damit hat der resultierende Automat 2 + 12 + 5 = 19 Zustände.

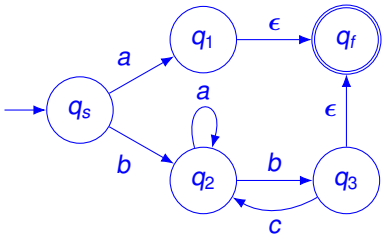
0
1
2
3
4
5
6
7

c)* Wir betrachten den NFA $N := \big(\{q_0, q_1, q_2, q_3\}, \{a, b, c\}, \delta, q_0, \{q_1, q_3\}\big)$, dessen Übergangsrelation δ durch folgende Skizze gegeben ist.

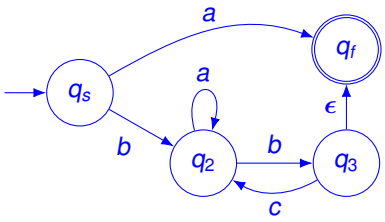


Berechnen Sie mit dem Algorithmus aus der Vorlesung einen regulären Ausdruck r, sodass $L(r) = L(N)$. Geben Sie dabei nach **jeder** Anwendung einer Transformationsregel den Automaten graphisch an und vereinfachen Sie die Ausdrücke in den Zwischenschritten nicht (Außnahme: Sie dürfen die Vereinfachung $\epsilon\alpha \equiv \alpha\epsilon \equiv \alpha$ anwenden).

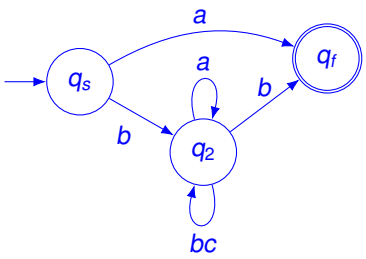
Schritt 1:



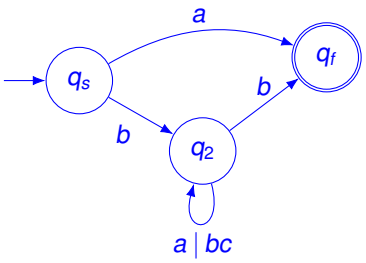
Schritt 2:



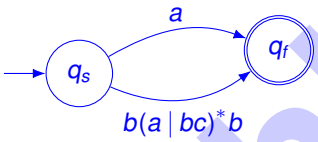
Schritt 3:



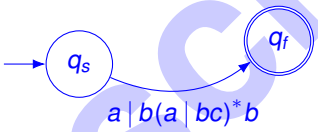
Schritt 4:



Schritt 5:



Schritt 6:



Aufgabe 3 Residualsprachen (12 Punkte)

Sei $\Sigma := \{a, b\}$ und $L \subseteq \Sigma^*$ eine reguläre Sprache mit

$$\epsilon, b \notin L, \qquad L^a = L^{bb} = L(a^*) \qquad \text{und} \qquad L^{ba} = \emptyset.$$

0 1 2 3

a)* Geben Sie einen regulären Ausdruck für L^b an und begründen Sie die Korrektheit des regulären Ausdrucks. Ihr regulärer Ausdruck darf maximal 5 Zeichen lang sein.
Beispielsweise enthält der Ausdruck abb^* 4 Zeichen und der Ausdruck $abb^* \mid bb$ 7 Zeichen.

$L^b := L(ba^*)$
Begründung: Da $b \notin L$, gilt $L^b = \{a\}L^{ba} \cup \{b\}L^{bb} = \emptyset \cup \{b\}L(a^*)$

0 1 2 3

b) Geben Sie einen regulären Ausdruck für L an und begründen Sie die Korrektheit des regulären Ausdrucks. Ihr regulärer Ausdruck darf maximal 10 Zeichen lang sein.

$L := L(aa^* \mid bba^*)$
Begründung: Da $\epsilon \notin L$, gilt $L = \{a\}L^a \cup \{b\}L^b = \{a\}L(a^*) \cup \{b\}L(ba^*)$

0 1 2 3 4

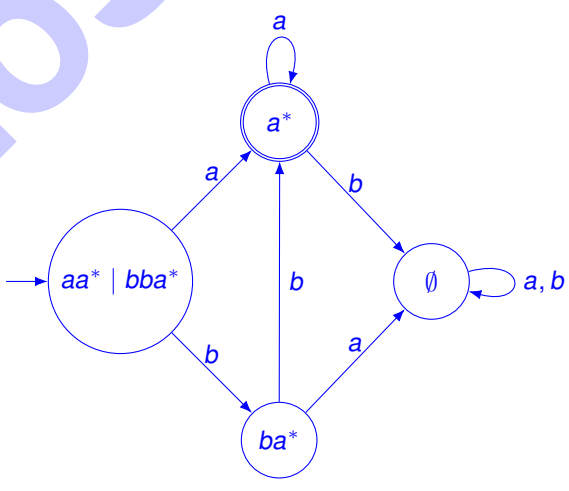
c)* Fakt: Es gibt keine weiteren Residualsprachen neben L, L^a, L^b, L^{ba}.

Entscheiden Sie für die folgenden Sprachen, zu welchen Residualsprachen L, L^a, L^b, L^{ba} sie gleich sind, indem Sie die rechten Seiten der Gleichungen mit L, L^a, L^b oder L^{ba} vervollständigen.

- $L^{aa} = L(a^*)^a = L^a$
- $L^{ab} = L(a^*)^b = \emptyset = L^{ba}$
- $L^{baa} = \emptyset^a = \emptyset = L^{ba}$
- $L^{bab} = \emptyset^b = \emptyset = L^{ba}$

0 1 2

d) Zeichnen Sie den kanonischen Minimalautomaten für L. Beschriften Sie die Zustände des Automaten jeweils mit einem regulären Ausdruck für die Residualsprache des jeweiligen Zustands. Je regulären Ausdruck dürfen Sie dabei maximal 10 Zeichen verwenden.



Aufgabe 4 Do You Even Recurse? (10 Punkte)

a)* Sei $P(r)$ das Prädikat, das für einen regulären Ausdruck r angibt, ob alle durch r erzeugten Wörter gerader Länge sind. Formal definieren wir

$$P(r) := \forall w \in L(r).\ \exists n \in \mathbb{N}.\ |w| = 2n.$$

Geben Sie, analog zu den rekursiven Funktionen aus den Übungs- und Hausaufgaben, eine rekursive, berechenbare Funktion ev an, sodass $\text{ev}(r) \iff P(r)$ für jeden regulären Ausdruck r gilt.
Für den Fall der Konkatenation $r_1 r_2$ dürfen Sie zusätzlich die Hilfsfunktionen empty(r) und odd(r) verwenden, wobei

$$\text{empty}(r) \iff L(r) = \emptyset \qquad \text{und} \qquad \text{odd}(r) \iff \forall w \in L(r).\ \exists n \in \mathbb{N}.\ |w| = 2n + 1.$$

0 1 2 3 4

- $\text{ev}(\emptyset) :=$ true
- $\text{ev}(\epsilon) :=$ true
- $\text{ev}(a) :=$ false
- $\text{ev}(r_1 \mid r_2) := \text{ev}(r_1) \wedge \text{ev}(r_2)$
- $\text{ev}(r^*) := \text{ev}(r)$
- $\text{ev}(r_1 r_2) := (\text{ev}(r_1) \wedge \text{ev}(r_2)) \vee (\text{odd}(r_1) \wedge \text{odd}(r_2)) \vee \text{empty}(r_1 r_2)$

b) Beweisen Sie die Korrektheit Ihrer Funktion mit struktureller Induktion. Genauer: Beweisen Sie per struktureller Induktion über r, dass $\text{ev}(r) \iff P(r)$.
Sie müssen nur die Fälle $\emptyset, \epsilon, a$ und r^* beweisen. Schreiben Sie für jeden Fall explizit die Induktionshypothese(n) auf und kennzeichnen Sie die Anwendung der Induktionshypothesen im Beweis deutlich. Zeigen Sie insbesondere auch, ob $P(\emptyset)$, $P(\epsilon)$, $P(a)$ gilt, indem Sie die Definition von P verwenden.

0 1 2 3 4 5 6

- Fall $r = \emptyset$: $\text{ev}(\emptyset) \iff \text{true} \iff \forall w \in \emptyset.\ \exists n \in \mathbb{N}.\ |w| = 2n \iff P(\emptyset)$
  Alternativ: $P(\emptyset) = \text{true} = \text{ev}(\emptyset)$ da die leere Menge keine Wörter enthält.
- Fall $r = \epsilon$: $\text{ev}(\epsilon) \iff \text{true} \iff |\epsilon| = 2 * 0 \iff \forall w \in \{\epsilon\}.\ \exists n \in \mathbb{N}.\ |w| = 2n \iff P(\epsilon)$
  Alternativ: Das leere Wort hat Länge 0, was gerade ist. Somit ist $P(\epsilon) = \text{true} = \text{ev}(\epsilon)$.
- Fall $r = a$:

  $\text{ev}(a) \iff \text{false} \iff \exists n \in \mathbb{N}.\ |a| = 1 = 2 * n \iff \forall w \in \{a\}.\ \exists n \in \mathbb{N}.\ |w| = 2n \iff P(a)$.

  Alternativ: Ein Buchstabe a hat Länge 1, was ungerade ist. Somit ist $P(a) = \text{false} = \text{ev}(a)$.

- Fall $r = s^*$: Induktionshypothese: $\text{ev}(s) \iff P(s)$.

$$\begin{aligned}\text{ev}(s^*) &\iff \text{ev}(s) \overset{IH}{\iff} P(s) \iff \forall w \in L(s).\ \exists n \in \mathbb{N}.\ |w| = 2n \\ &\overset{(*)}{\iff} \forall w \in L(s^*).\ \exists n \in \mathbb{N}.\ |w| = 2n \\ &\iff P(s^*)\end{aligned}$$

Beweis von $(*)$:

$\Longrightarrow$: Sei $w \in L(s^*)$. Dann gibt es $w_1, \ldots, w_k \in L(s)$ mit $w = w_1 \cdots w_k$. Nach Annahme gibt es $n_1, \ldots, n_k$ mit $|w_i| = 2n_i$. Dann gilt $|w| = |w_1 \cdots w_k| = 2n_1 + \cdots + 2n_k = 2(n_1 + \cdots + n_k)$.

$\Longleftarrow$: Wenn $w \in L(s)$, dann auch $w \in L(s^*)$. Somit gibt es nach Annahme n mit $|w| = 2n$.

Aufgabe 5 Kontextfreie Sprachen und Pumping Lemma (11 Punkte)

a)* Zeigen Sie unter Verwendung des Pumpinglemmas, dass folgende Sprache über dem Alphabet $\Sigma := \{a, b\}$ nicht kontextfrei ist: $L := \{a^i b^j a^k \mid i, j, k \in \mathbb{N} \wedge j = \min(i, k)\}$.

0 1 2 3 4 5 6 7

- Wir nehmen an, dass L kontextfrei ist und leiten einen Widerspruch her.
- Sei $n \in \mathbb{N}_+$ eine Pumping-Lemma-Zahl für L.
- Wähle $z := a^n b^n a^n \in L$, es gilt $|z| = 3n \geq n$.
- Es gibt also für z eine Zerlegung $z = uvwxy$ mit den Eigenschaften

  (1) $vx \neq \varepsilon$ (2) $|vwx| \leq n$ (3) $\forall i \in \mathbb{N}.\ uv^i wx^i y \in L$.

- Wir unterscheiden die folgenden Fälle:
  - $|vx|_a = 0$: Aus (1) und der Wahl von z folgt, dass $|vx|_b = k > 0$. Folglich ist $uv^0wx^0y = a^n b^{n-k} a^n \notin L$, da $n - k \neq \min(n, n)$. Dies ist ein Widerspruch zu (3)
  - $|vx|_a = k > 0$: Wir machen eine weitere Fallunterscheidung:
    - $|vx|_b = 0$: Dann ist $uv^0wx^0y = a^{n-k}b^n a^n$ oder $uv^0wx^0y = a^n b^n a^{n-k}$. Da $min(n-k, n) \neq n$ beziehungweise $min(n, n-k) \neq n$ und daher $uv^0wx^0y \notin L$ folgt wieder ein Widerspruch zu (3).
    - $|vx|_b = l > 0$. Wegen (2) berührt vwx nur einen der a Blöcke. uv^2wx^2y hat daher die Form $a^{n+k}b^{n+l}a^n$ oder $a^n b^{n+l}a^{n+k}$. Da $k, l > 0$ sind auch $n + k > n$ und $n + l > n$. Folglich ist die Anzahl der bs $(n + l)$ größer als $min(n + k, n) = min(n, n + k) = n$. Daher ist $uv^2wx^2y \notin L$, ein Widerspruch zu (3).
- Jeder Fall führt zu einem Widerspruch. Somit ist L nicht kontextfrei.

0 1 2 3 4

b)* Geben Sie eine kontextfreie Grammatik für die Sprache $L := \{a^i b^j a^k \mid i, j, k \in \mathbb{N} \wedge j = i + k\}$ über dem Alphabet $\Sigma := \{a, b\}$ an. Ihre Grammatik darf maximal 7 Produktion enthalten.
Beachten Sie: $S \to a \mid b$ ist eine Abkürzung für $S \to a$ und $S \to b$ und enthält somit 2 Produktionen.

$$S \to XY$$
$$X \to aXb \mid \varepsilon$$
$$Y \to bYa \mid \varepsilon$$

Aufgabe 6 Typ-0 Grammatiken (12 Punkte)

Erinnerung: Eine Typ-0 Grammatik kann sowohl Terminale als auch Nichtterminale auf beiden Seiten Ihrer Produktionen besitzen, d.h. $P \subseteq (V \cup \Sigma)^* \times (V \cup \Sigma)^*$.

a)* Gegeben eine Typ-0 Grammatik $G = (V, \Sigma, P, S)$, sei $G' := \big(V \cup \{S'\}, \Sigma, P', S'\big)$, wobei $S' \notin V$ und $P' := P \cup \{S' \to SS\}$.
Geben Sie eine Grammatik G mit $L(G') \neq L(G)L(G)$ und maximal 3 Produktionen an. Begründen Sie die Korrektheit Ihres Beispiels.

0 1 2 3

Sei G die Grammatik $SS \to a$ mit Startsymbol S. Es gilt $L(G) = \emptyset$ und $L(G') = \{a\}$.

b)* Geben Sie ein Verfahren an, das als Input eine Typ-0 Grammatik $G = (V, \Sigma, P, S)$ erhält, und eine Grammatik H mit $L(H) = L(G)L(G)$ und höchstens $3(|\Sigma| + |P|)$ Produktionen konstruiert. Erläutern Sie die Idee Ihres Verfahrens auch informell!

0 1 2 3 4 5 6 7

Idee: wir möchten die Idee aus Aufgabe a) weiterverfolgen, müssen dabei aber verhindern, dass die beiden Ableitungen vom ursprünglichen Startsymbol durch Konkatenation die linke Seite einer Produktion bilden können. Hierfür stellen wir sicher, dass (1) Terminale nicht mehr auf der linken Seite der Produktion vorkommen dürfen und (2) wir disjunkte Nichtterminale für die beide Ableitungen (beginnend vom neuen Startsymbol) verwenden.

Zwei Schritte:

1. Wir ersetzen jedes Vorkommen eines Terminals a in G durch ein neues Nichtterminal X_a und fügen die Produktion $X_a \to a$ hinzu. Damit erhalten wir eine Grammatik $G_1 = (V_1, \Sigma, P_1, S)$ mit Produktionen der Gestalt $X_1 \cdots X_n \to Y_1 \cdots Y_m$ und $X_a \to a$.
2. Sei $V_1' := \{X' \mid X \in V_1\}$ eine Kopie von V_1, disjunkt mit V_1, und sei P_1' die Menge von Produktionen, die man erhält, indem man jedes Vorkommen eines Nichtterminals X durch X' ersetzt. Sei $S_0 \notin V_1 \cup V_1'$. Sei H die Grammatik

$$H := \big(V_1 \cup V_1' \cup \{S_0\}, \Sigma, P_1 \cup P_1' \cup \{S_0 \to SS'\}, S_0\big).$$

Es gilt $L(H) = L(G)L(G)$: Jedes Wort von Nichterminalzeichen, das von S_0 abgeleitet werden kann, ist von der Gestalt $\alpha\beta$ mit $\alpha \in V_1^*, \beta \in (V_1')^*$. Jede Produktion in P kann nur auf α und jede Produktion in P' kann nur auf β angewendet werden. Da P bzw. P' von S bzw. S' aus $L(G)$ erzeugen, folgt somit $L(H) = L(G)L(G)$.

Alternative: siehe nächste Seite

Idee: wir möchten die Idee aus Aufgabe a) weiterverfolgen, müssen dabei aber verhindern, dass die beiden Ableitungen vom ursprünglichen Startsymbol durch Konkatenation die linke Seite einer Produktion bilden können. Hierfür stellen wir sicher, dass (1) Terminale nicht mehr auf der linken Seite der ursprünglichen Produktionen vorkommen, (2) die erzeugten Symbolketten der beiden Ableitungen durch ein neues Symbol getrennt werden und (3) das Trennsymbol erst entfernt werden kann, sobald beide Ableitungen für S beendet sind (d.h. alle Nichtterminale entfern sind).

Zwei Schritte:

1. Wir ersetzen jedes Vorkommen eines Terminals a in G durch ein neues Nichtterminal X_a und fügen die Produktion $X_a \to a$ hinzu. Damit erhalten wir eine Grammatik $G_1 = (V_1, \Sigma, P_1, S)$ mit Produktionen der Gestalt $X_1 \cdots X_n \to Y_1 \cdots Y_m$ und $X_a \to a$.
2. Sei $\# \notin V_1$. Für jedes Terminalzeichen x führen wir die Produktionen $\#x \to x\#$ und $x\# \to \#x$ ein, formal $P' := \{\#x \to x\# \mid x \in \Sigma\} \cup \{x\# \to \#x \mid x \in \Sigma\}$. Sei $S_0 \notin (V_1 \cup \{\#\})$. Sei H die Grammatik

$$H := \big(V_1 \cup \{\#, S_0\}, \Sigma, P_1 \cup P' \cup \{S_0 \to \#S\#S\#, \#\#\# \to \epsilon\}, S_0\big).$$

0 1 2

c) Wenden Sie Ihr Verfahren auf die folgende Grammatik mit Startsymbol S an:

$$S \to XaYa \quad aY \to bc \quad Xb \to cc$$

1. Schritt: $S \to XAYA, \quad AY \to BC, \quad XB \to CC, \quad A \to a, \quad B \to b, \quad C \to c$

2. Schritt: H enthält die Produktionen

$$S_0 \to SS',$$
$$S \to XAYA, \quad AY \to BC, \quad XB \to CC, \quad A \to a, \quad B \to b, \quad C \to c$$
$$S' \to X'A'Y'A', \quad A'Y' \to B'C', \quad X'B' \to C'C', \quad A' \to a, \quad B' \to b, \quad C' \to c$$

Alternative:

$$S_0 \to \#S\#S\#, \quad \#x \to x\#, \quad x\# \to \#x, \quad \#\#\# \to \epsilon, \quad (\forall x \in \{a, b, c\})$$
$$S \to XAYA, \quad AY \to BC, \quad XB \to CC, \quad A \to a, \quad B \to b, \quad C \to c$$

Aufgabe 7 Ask Me Anything (10 Punkte)

Kreuzen Sie richtige Antworten an ☒
Kreuze können durch vollständiges Ausfüllen gestrichen werden ■
Gestrichene Antworten können durch nebenstehende Markierung erneut angekreuzt werden ×■

a)* Sei $f : \Sigma^* \to \Sigma^*$ eine berechenbare, partielle Funktion. Der Graph von f ist die Menge

$$\mathit{Graph}(f) := \{(w, f(w)) \mid w \in \Sigma^*, f(w) \neq \bot\}.$$

Wählen Sie alle wahren Aussagen, die unabhängig von f gelten, aus. Beachten Sie insbesondere, dass beispielsweise jede entscheidbare Menge auch semi-entscheidbar ist.

- ☐ Das Komplement von $\mathit{Graph}(f)$ ist semi-entscheidbar.
- ☐ $\mathit{Graph}(f)$ ist entscheidbar.
- ☒ $\mathit{Graph}(f)$ ist semi-entscheidbar.

b)* Sei $\Sigma := \{a, b\}$ und sei $L \subseteq \Sigma^*$. Welche der folgenden Aussagen sind wahr?

- ☐ Wenn $L^a \in \text{NP}$, dann ist $L \in \text{NP}$.
- ☒ Wenn $L^a \in \text{NP}$ und $L^b \in \text{NP}$, dann ist $L \in \text{NP}$.
- ☒ Wenn $L \in \text{NP}$, dann ist $L^a \in \text{NP}$.

c)* Sei Σ ein Alphabet. Sei $\mathcal{K}$ die Menge aller kontextfreien Grammatiken mit Terminalen aus Σ. Gegeben eine Sprache $L \subseteq \Sigma^*$, sei $\min(L)$ die Menge der kürzesten Wörter in L. Welche der folgenden Mengen sind entscheidbar?

- ☒ $\{(G_1, G_2) \in \mathcal{K} \times \mathcal{K} \mid \min(L(G_1)) \subseteq L(G_2)\}$.
- ☐ $\{(G_1, G_2) \in \mathcal{K} \times \mathcal{K} \mid L(G_1) \setminus \min(L(G_1)) = L(G_2)\}$.
- ☐ $\{(G_1, G_2) \in \mathcal{K} \times \mathcal{K} \mid L(G_1) \setminus L(G_2) = \emptyset\}$.

d)* Seien A, B, C Sprachen über demselben Alphabet Σ. Welche der folgenden Aussagen sind wahr?

- ☒ $A \leq_p B$ genau dann wenn $\overline{A} \leq_p \overline{B}$.
- ☒ Sei H das (allgemeine) Halteproblem. Wenn $A \in \text{NP}$, dann $A \leq_p H$.
- ☐ A und B sind entscheidbar genau dann wenn $A \leq B$ und $B \leq A$.

e)* Erinnerung: Eine Familie von Sprachen ist eine Menge von Mengen.
Wählen Sie alle Aussagen, die zutreffen. Beachten Sie insbesondere, dass jede entscheidbare Menge auch semi-entscheidbar ist.

- ☒ Es gibt eine abzählbar unendliche Familie von entscheidbaren Sprachen $\{L_i\}_{i=1}^{\infty}$ über demselben Alphabet, sodass $\bigcup_{i=1}^{\infty} L_i$ nicht semi-entscheidbar ist.
- ☒ Es gibt eine abzählbar unendliche Familie von entscheidbaren Sprachen $\{L_i\}_{i=1}^{\infty}$ über demselben Alphabet, sodass $\bigcup_{i=1}^{\infty} L_i$ unentscheidbar ist.
- ☒ Es gibt eine abzählbar unendliche Familie von entscheidbaren Sprachen $\{L_i\}_{i=1}^{\infty}$ über demselben Alphabet, sodass $\bigcup_{i=1}^{\infty} L_i$ entscheidbar ist.

Aufgabe 8 Spiel, SAT und Sieg (12 Punkte)

Eine Klausel (d.h. eine Disjunktion von Literalen) ist

1. *positiv*, falls alle Literale positiv sind,
2. *negativ*, falls alle Literale negativ sind und
3. *gemischt*, sonst.

Beispielsweise ist die Klausel $(x \vee y \vee z)$ positiv, die Klausel $(\neg x \vee \neg z)$ negativ, und die Klausel $(x \vee \neg y)$ gemischt.
Eine Formel in konjunktiver Normalform (KNF) ist *unvermischt* wenn sie keine gemischte Klauseln enthält. Beispielsweise ist die Formel $(x \vee y) \wedge (\neg x \vee \neg z)$ unvermischt, aber die Formel $(x \vee y) \wedge (x \vee \neg y)$ nicht.
Sei UKNF-SAT die Menge aller Formeln in KNF, die unvermischt und erfüllbar sind.

0 1 2 3 4 5 6 7 8 9 10

a) Lösen Sie eine der folgenden beiden Aufgaben:

(i) Geben Sie einen polynomiellen Algorithmus an, der UKNF-SAT entscheidet. Begründen Sie die Korrektheit Ihres Algorithmus.

(ii) Reduzieren Sie KNF-SAT auf UKNF-SAT.

Korrigierte Angabe (am Anfang der Klausur verkündet): Geben Sie eine polynomielle Reduktion von KNF-SAT auf UKNF-SAT an. Beweisen Sie die Korrektheit der Reduktion.

Wir reduzieren KNF-SAT auf UKNF-SAT. Somit ist UKNF-SAT NP-hart und, unter der Ananhme $P \neq NP$, nicht in P, womit Aufgabe (i) unlösbar ist.

Sei F eine KNF Formel mit m Variablen. Für jede Variable x, die negiert in F vorkommt, führe eine neue Variable nx ein. Wir ersetzen jedes Vorkommen von $\neg x$ duch nx und fügen die Klauseln $(x \vee nx) \wedge (\neg x \vee \neg nx)$ hinzu. Sei G die resultierende Formel.

Die Reduktion ist polynomiell: Wir durchlaufen lediglich die Formel und führen syntaktische Ersetzungen und Erweiterungen durch. Zusätzlich zu der ursprünglichen Formel erstellen wir $\mathcal{O}(m)$ Klauseln in $\mathcal{O}(m)$ Schritten.

Korrektheit:

$\Longrightarrow$: Sei σ eine erfüllende Belegung von F. Wir erweiteren σ zu einer Belegung σ' für G indem wir $\sigma'(nx) := \sigma(\neg x)$ setzen. Die Klauseln aus F in G sind somit weiterhin erfüllt. Da $\sigma'(nx) = \sigma(\neg x) = 1 - \sigma(x)$ gilt, sind außerdem die Klauseln $(x \vee nx) \wedge (\neg x \vee \neg nx)$ erfüllt. Somit ist σ' eine erfüllende Belegung für G.

$\Longleftarrow$: Sei σ eine erfüllende Belegung von G. Da $(x \vee nx) \wedge (\neg x \vee \neg nx)$ unter σ erfüllt sind, gilt $\sigma(nx) = \sigma(\neg x)$. Somit erfüllt σ auch die Klauseln, die aus G hervorgehen, indem man alle Vorkomnisse nx durch $\neg x$ ersetzt. Nun sind die Klauseln aus F eine Teilmenge dieser Klauseln und somit σ eine erfüllende Belegung für F.

Alternative: Jede Klausel K ersetzen wir durch zwei Klauseln $K_1 := \{L \in K \mid L \text{ ist positiv}\} \cup \{x_K\}$ und $K_2 := \{L \in K \mid L \text{ ist negativ}\} \cup \{\neg x_K\}$. Sei G die resultierende Formel.

Die Reduktion ist polynomiell: Wir durchlaufen lediglich die Formel, teilen alle Klauseln anhand eines syntaktischen Tests in zwei neue Klauseln auf und führen dabei ein weiteres Literal in jede Klausel ein. Somit ist die Reduktion sogar linear.

Korrektheit:

$\Longrightarrow$: Sei σ eine erfüllende Belegung von F. Wir erweiteren σ zu einer Belegung σ' für G wie folgt: Für jede Klausel $K \in F$, wähle ein beliebiges $L \in K$ mit $\sigma(L) = 1$. Falls L positiv ist, setze $\sigma'(x_K) := 0$, ansonsten $\sigma'(x_K) := 1$.

Es gilt $\sigma'(K_1) = 1$: Falls $L \in K_1$, gilt $\sigma'(L) = \sigma(L) = 1$. Ansonsten gilt $\sigma'(x_K) = 1$ und $x_K \in K_1$.

Analog zeigt sich $\sigma'(K_2) = 1$. Somit ist σ' eine erfüllende Belegung für G.

$\Longleftarrow$: Sei σ eine erfüllende Belegung von G. Dann gibt es $L_1 \in K_1, L_2 \in K_2$ mit $\sigma(L_1) = \sigma(L_2) = 1$. Da weiters L_1 positiv und L_2 negativ ist, verwenden L_1, L_2 verschiedene Variablen. Somit verwendet höchstens eins der Literale die Variable x_K. Somit $\{L_1, L_2\} \cap K \neq \emptyset$. Somit gilt $\sigma(K) = 1$ und σ ist eine erfüllende Belegung für F.

b) Falls Sie einen polynomiellen Algorithmus für UKNF-SAT angegeben haben (Aufgabe (i)), wenden Sie Ihren Algorithmus auf die Formel $F := (x \vee y) \wedge (\neg x \vee \neg y \vee \neg z) \wedge \neg v$ Schritt für Schritt an.

Falls Sie hingegen KNF-SAT auf UKNF-SAT reduziert haben (Aufgabe (ii)), wenden Sie Ihre Reduktionsfunktion auf die Formel $F := (x \vee \neg y) \wedge (x \vee y \vee \neg z) \wedge \neg v$ an.

0
1
2

$(x \vee ny) \wedge (x \vee y \vee nz) \wedge nv \wedge (y \vee ny) \wedge (\neg y \vee \neg ny) \wedge (z \vee nz) \wedge (\neg z \vee \neg nz) \wedge (v \vee nv) \wedge (\neg v \vee \neg nv)$

Alternative: $(x \vee x_1) \wedge (\neg y \vee \neg x_1) \wedge (x \vee y \vee x_2) \wedge (\neg z \vee \neg x_2) \wedge x_3 \wedge (\neg v \vee \neg x_3)$

Zusätzlicher Platz für Lösungen. Markieren Sie deutlich die Zuordnung zur jeweiligen Teilaufgabe. Vergessen Sie nicht, ungültige Lösungen zu streichen.