Lehrstuhl für Theoretische Informatik
Fakultät für Informatik
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 / Retake	**Datum:**	Mittwoch, 13. Oktober 2021
Prüfer:	Prof. Dr. Dr. h.c. Javier Esparza	**Uhrzeit:**	11:30 – 14:30

Bearbeitungshinweise

- Diese Klausur umfasst **16 Seiten** mit insgesamt **9 Aufgaben**.
  Bitte kontrollieren Sie jetzt, dass Sie eine vollständige Angabe erhalten haben.
- Die Gesamtpunktzahl in dieser Prüfung beträgt 100 Punkte.
- 45 Punkte sind *hinreichend* zum Bestehen.
- Sie müssen Ihre Klausur eingescannt bis 14:45 online auf TUMExam einreichen:
  - Sie müssen nur die von Ihnen bearbeiteten Seiten und das **unterschriebene** Deckblatt hochladen.
  - Achten Sie darauf, dass sowohl Ihre **Lösungen als auch die Barcodes klar lesbar** sind.
  - In begründeten Fällen (wie z.B. bei technischen Problemen) können Sie der Übungsleitung bis 14:45 per Email (theoleitung@in.tum.de) ihre Klausur zukommen lassen, als PDF oder SHA256-Prüfsumme.
- Sie müssen die Klausur alleine bearbeiten. Die Klausur ist open-book (Kofferklausur), allerdings dürfen Sie in keinster Weise Unterstützung von anderen Personen **erhalten** oder diesen **geben** (in Person, Chat, Foren, Diskussiongruppen, etc.). Eine solche Unterstützung wird als Unterschleif bewertet und mit den Konsequenzen, wie in der APSO beschrieben, geahndet.
- Sie dürfen jegliche Art von Literatur (auch im Internet) benutzen. Sollten Sie dabei auf Lösungsansätze stoßen, die Sie für die Klausur verwenden möchten, so müssen Sie diese Teile klar und deutlich zitieren (Literaturverweis bzw. Link). Die Lösung selber müssen Sie dennoch weiterhin selbstständig in die Klausur übertragen. Ihnen ensteht durch eine Zitation kein Nachteil.
- Sie können uns Fragen via Zulip oder E-Mail (theoleitung@in.tum.de) stellen, oder uns auf diesem Wege auf Probleme hinweisen. Inhaltliche Fragen werden wir nicht beantworten; falls Ihnen eine Aufgabenstellung mehrdeutig erscheint, notieren Sie bitte Ihre Interpretation der Aufgabe.
- Damit wir Sie während der Klausur erreichen können, behalten Sie bitte Ihr E-Mail Postfach im Blick.
- Mit * gekennzeichnete Teilaufgaben sind ohne Kenntnis der Lösung vorheriger Teilaufgaben lösbar.
- Sie dürfen Ergebnisse vorheriger Teilaufgaben auch dann verwenden, wenn Sie diese nicht lösen konnten.
- **Es werden nur solche Ergebnisse gewertet, bei denen der Lösungsweg erkennbar ist**. Alle Aufgaben sind **grundsätzlich zu begründen**, sofern es nicht ausdrücklich anders vermerkt ist.
- Schreiben Sie weder mit roter noch grüner Farbe.
- Ihre Lösungen müssen **handschriftlich** verfasst sein (digital oder auf Papier)!
- $0 \in \mathbb{N}$.

Aufgabe 1 Reguläre und kontextfreie Sprachen (13 Punkte)

Für die folgenden Fragen ist eine Begründung nur erforderlich, wenn explizit danach gefragt wird.
Die Länge eines regulären Ausdrucks über dem Alphabet Σ ist die Länge der üblichen Repräsentation über dem Alphabet $\Sigma \cup \{(,), *, |, \epsilon, \emptyset\}$.

0 1 2

a)* Geben Sie Sprachen $A, B \subseteq \{a, b\}^*$ mit $A \neq B$ und $|AB| < |A||B|$ an.

$A = \{ba, baa\}, B = \{ab, aab\}$

0 1 2

b)* Geben Sie einen regulären Ausdruck r für die Sprache $\{w \in \{0,1\}^* : |w|_1 \leq 3\}$ an, mit $|r| \leq 25$.

$0^*(1|\epsilon)\,0^*(1|\epsilon)\,0^*(1|\epsilon)0^*$

0 1 2 3

c)* Geben Sie eine kontextfreie Grammatik G für die Sprache $\{a^{12}\}$ an. G muss in Chomsky-Normalform sein und höchstens 5 Produktionen haben. Beachten Sie, dass z.B. $S \to SS \mid \varepsilon$ zwei Produktionen sind.

$S \to A_8A_4 \qquad A_8 \to A_4A_4 \qquad A_4 \to A_2A_2 \qquad A_2 \to A_1A_1 \qquad A_1 \to a$

0 1 2 3

d)* Gibt es eine kontextfreie Grammatik G, sodass jeder PDA für $L(G)$, der über Finalzustände akzeptiert, mindestens 2 Zustände hat? Falls ja, geben Sie ein solches G an und begründen kurz, dass es die geforderte Eigeschaft erfüllt. Falls nein, begründen Sie kurz, wieso dies unmöglich ist.

Ja, gibt es: $S \to a$. Da $\varepsilon \notin L(G) = \{a\}$, darf der Startzustand nicht akzeptierend sein, aber da $L(G) \neq \emptyset$ muss es einen akzeptierenden Zustand geben.

0 1 2 3

e)* Sei $S \subset \mathbb{N}$ endlich mit $S \neq \emptyset$. Geben Sie den minimalen DFA für $L := \{a^n : n \in S\}$ über dem Alphabet $\Sigma := \{a\}$ formal an.

Mit $m := \max S$ haben wir Zustände $\{0, ..., m+1\}$, $\delta(q, a) := q + 1$ für $q \leq m$ und $\delta(m+1, a) := m + 1$. Die Finalzustände sind genau S, Startzustand ist 0.

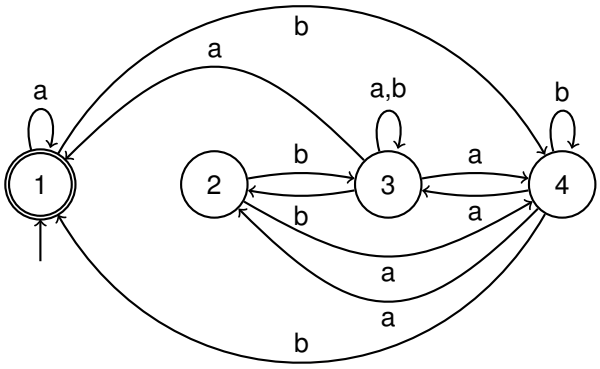
Aufgabe 2 **Potenzkonstruktion (10 Punkte)**

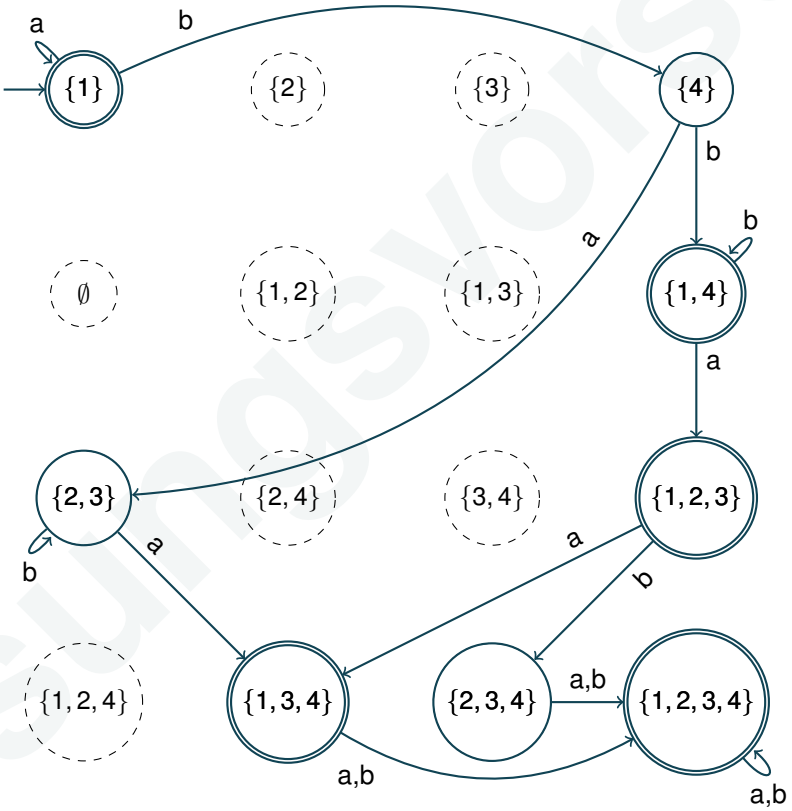
Berechnen Sie mit Hilfe der Potenzmengenkonstruktion einen DFA für den folgenden NFA.

Achtung: Falls Sie die Klausur ausgedruckt haben oder digital bearbeiten, schreiben Sie bitte direkt in die folgende Vorlage. Falls Sie auf dem Template arbeiten, das Sie vor Klausurbeginn ausgedruckt haben, ordnen Sie die Zustände bitte trotzdem so wie in der Vorlage an.

Hinweis: Sie müssen nur den erreichbaren Teil des Automaten berechnen!

0 1 2 3 4 5 6 7 8 9 10





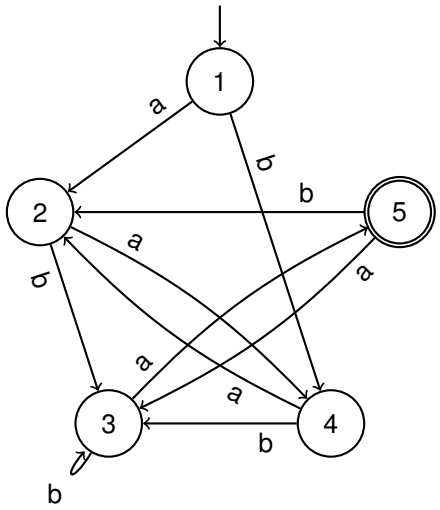
Aufgabe 3 **Minimierung (12 Punkte)**

0 1 2 3 4 5 6

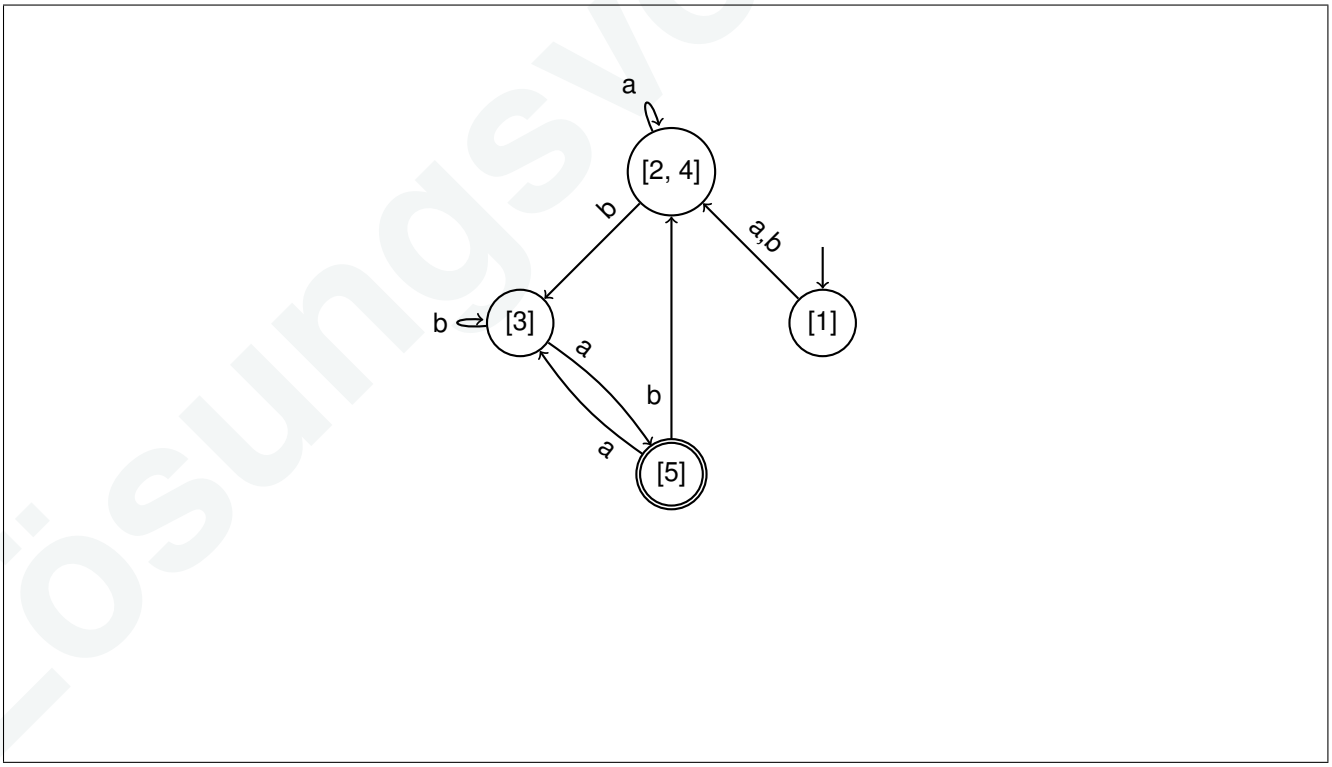
a)* Minimieren Sie den folgenden DFA mit dem erweiterten Minimierungsalgorithmus aus der Übungsaufgabe Ü4.4(b) (siehe https://teaching.model.in.tum.de/2021ss/theo/ex/ue04-solution.pdf?key=o25mcGBZ) und zeichnen Sie den minimierten DFA.

Füllen Sie dazu die Tabelle aus und geben Sie für jedes Paar an Zuständen (q, r), das sie „markieren“, ein **kürzestes** Wort w an, das sie unterscheidet (d.h. von $\delta(q, w), \delta(r, w)$ ist genau einer akzeptierend).

Hinweis: Falls Sie nicht den erweiterten Minimierungsalgorithmus sondern den Minimierungsalgorithmus aus der Vorlesung verwenden, um dem DFA zu minimieren, können Sie trotzdem noch bis zu 4 Punkte bekommen.

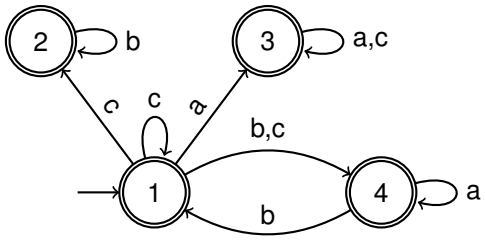


1				
ba	2			
a	*a*	3		
ba	=	*a*	4	
ε	ε	ε	ε	5

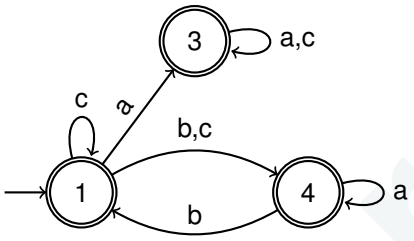


b)* Geben Sie einen NFA an, der die gleiche Sprache akzeptiert wie der folgende NFA, aber weniger Zustände hat. Erklären Sie **kurz**, wie Sie zu Ihrer Lösung gekommen sind.

0 1 2 3 4 5 6



Mit Zustand 2 akzeptiert der NFA genau die Wörter aus $L(c^*cb^*)$. Diese können aber auch ohne Zustand 2 erkannt werden, indem man die c's in Zustand 1 liest und die b's durch Wechseln zwischen Zuständen 1 und 4. Somit kann man Zustand 2 entfernen:



Wörter die **nicht** akzeptiert werden dürfen:
bc, *bbbc*, *cabbc*
ab, *bbab*, *cabab*
cacb

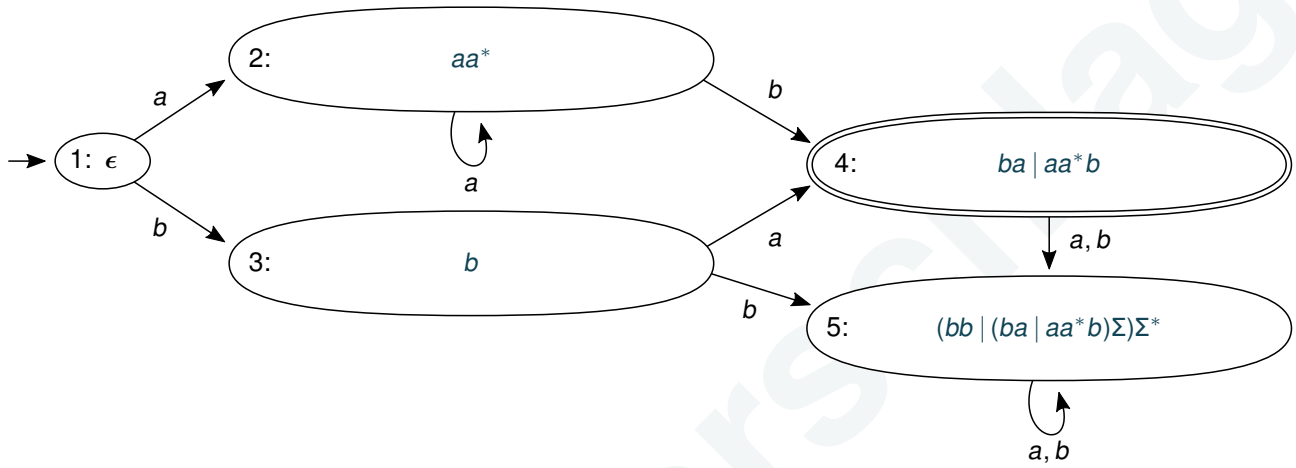
Aufgabe 4 Äquivalenzklassen (11 Punkte)

0 1 2 3 4 5 6

a)* Der unten gezeichnete DFA M ist der kanonische Minimalautomat für den regulären Ausdruck $r := ba \mid aa^*b$ über dem Alphabet $\Sigma := \{a, b\}$. Beschriften Sie jeden Zustand mit der entsprechenden Äquivalenzklasse von $L(r)$, indem Sie einen regulären Ausdruck für die Äquivalenzklasse angeben.
Beachten Sie, dass im minimalen DFA die Äquivalenzklasse eines Zustandes genau die Wörter enthält, die diesen Zustand vom Startzustand aus erreichen. Zum Beispiel hat Zustand 1 die Äquivalenzklasse $\{\varepsilon\}$, und wurde dementsprechend mit ϵ beschriftet.

Achtung: Es genügt nicht, einen Repräsentanten der Äquivalenzklasse zu notieren!

Hinweis: Sie können direkt in die Aufgabenstellung schreiben.



0 1 2 3 4 5

b)* Sei $\Sigma := \{a, b, c\}$ und $L := \{a^n c b^m : n > 7m \wedge n, m \in \mathbb{N}\}$. Beweisen Sie, dass $\equiv_L$ unendliche viele Äquivalenzklassen hat, indem Sie eine unendliche Folge an Wörten $w_0, w_1, w_2, ... \in \Sigma^*$ identifizieren, für die $w_i \not\equiv_L w_j$ für alle $i \neq j$ gilt.
Anmerkung: Aus dem Satz von Myhill-Nerode folgt dann, dass L nicht regulär ist.

Sei $w_n := a^{7n}c$ für $n \in \mathbb{N}$. Wir zeigen nun $w_n \not\equiv_L w_m$ für alle $n \neq m$, seien also $n, m \in \mathbb{N}$ mit $n \neq m$ beliebig. Wir können $n < m$ ohne Beschränkung der Allgemeingültigkeit annehmen. Für $v := b^n$ gilt dann $w_n v = a^{7n}cb^n \notin L$, aber $w_m v = a^{7m}cb^n \in L$, da $7m > 7n$ aus $m > n$ folgt. Dies zeigt $w_n \not\equiv_L w_m$ und schließt den Beweis ab.

Alternative Lösung mit Residualsprachen: Wir wählen wieder $w_n := a^{7n}c$ für $n \in \mathbb{N}$. Nun gilt $L^{w_n} = \{b^m : 7n > 7m\}$, also $|L^{w_n}| = n$. Daraus folgt dann $L^{w_n} \neq L^{w_m}$ für alle $n \neq m$, und nach Ü5.6(a) somit $w_n \not\equiv_L w_m$.

Aufgabe 5 CFG Konstruktion (10 Punkte)

a)* Sei $\Sigma := \{a, b, c\}$. Wir betrachten folgende Grammatik G:

$$S \to XX$$
$$X \to YS \mid a \mid b \mid c$$
$$Y \to a$$

0 1 2 3 4 5 6

Konstruieren Sie eine kontextfreie Grammatik H mit $L(H) = L(G) \cap \Sigma^4$. Ihre Grammatik H darf höchstens 15 Produktionen haben.
Hinweis: Die Variablen der Musterlösung kommen aus der Menge $\{S_1, ..., S_4, X_1, ..., X_4, Y_1, ..., Y_4, \}$.

Das Startsymbol ist S_4.

$$S_4 \to X_1X_3 \mid X_2X_2 \mid X_3X_1$$
$$S_2 \to X_1X_1$$
$$X_3 \to Y_1S_2 \mid Y_2S_1$$
$$X_2 \to Y_1S_1$$
$$X_1 \to a \mid b \mid c$$
$$Y_1 \to a$$

b)* Sei $G = (V, \Sigma, P, S)$ eine beliebige kontextfreien Grammatik in Chomsky-Normalform (CNF) und $n \in \mathbb{N}$. Geben Sie eine kontextfreie Grammatik $H = (V', \Sigma, P', S')$ an, sodass $L(H) = L(G) \cap \Sigma^n$ und $|P'| \leq n^2|P|$. Zeigen Sie außerdem, dass $|P'| \leq n^2|P|$ gilt.

0 1 2 3 4

Für jede Variable der ursprünglichen Grammatik führen wir n Variablen ein, die sich zusätzlich merken, wie lang das abgeleitete Wort ist. Wir wählen also $V' := \{X_i : X \in V, i \in \{1, ..., n\}\}$, und in unserer Konstruktion soll $L_H(X_i) = L_G(X) \cap \Sigma^i$ erfüllt sein. (Da G in CNF ist, kann ε nicht abgeleitet werden.)
Für jede Produktion der Form $(X \to YZ) \in P$ (wobei $X, Y, Z \in V$), führen wir die Produktionen

$$X_i \to Y_jZ_k \qquad \text{für alle } i, j, k \in \{1, ..., n\} \text{ mit } i = j + k$$

ein. Es gibt hier höchstens n^2 Möglichkeiten i, j, k zu wählen, da $j, k \in \{1, ..., n\}$ und sich i aus der Wahl von j, k ergibt. Für die Produktionen der Form $(X \to c) \in P$, mit $X \in V, c \in \Sigma$, erstellen wir die folgende Produktion:

$$X_1 \to a$$

Da G in CNF ist, sind alle Produktionen in einer dieser Formen. Unser Startsymbol ist $S' := S_n$.
Wir haben für jede Produktion höchstens n^2 neue Produktionen erstellt, also gilt $|P'| \leq n^2|P|$.

Aufgabe 6 Entscheidbarkeit und Komplexität (10 Punkte)

Entscheiden Sie, ob die folgenden Aussagen jeweils wahr sind, **unter der Annahme P $\neq$ NP**. Falls ja, geben Sie eine *kurze* Begründung an, falls nein, ein Gegenbeispiel.
Achtung: Wenn die Aussage falsch ist, müssen Sie ein konkretes Gegenbeispiel angeben, eine Begründung genügt nicht.

0 1 2

a)* Sei $L \subseteq \{0,1\}^*$ entscheidbar. Dann ist $L' := \{w \in \{0,1\}^* : L(M_w) = L\}$ unendlich.

Wahr. Man kann einer beliebigen TM M nutzlose Zustände hinzufügen. (Alternativ: $L' \neq \{0,1\}^*$ ist klar, also ist L' eine nichttriviale Eigenschaft von $L(M_w)$, nach Satz von Rice also unentscheidbar, und somit unendlich.)

0 1 2

b)* Seien $L_1, L_2, ...$ entscheidbar. Dann ist $L' := \bigcup_{n=1}^{\infty} L_i$ entscheidbar.

Falsch, sei $w_1, w_2, ... \in \Sigma^*$ eine Aufzählung der Wörter in $\mathcal{H}$, also $\mathcal{H} = \{w_1, w_2, ...\}$. Mit $L_i := \{w_i\}$ ist jedes L_i endlich und somit entscheidbar, aber $L' = \mathcal{H}$ ist unentscheidbar.

0 1 2

c)* Für jedes unendliche $L \subseteq \{0,1\}^*$ ist die Menge $A := \{L(M) \subseteq L : M \text{ ist TM}\}$ abzählbar.

Wahr. Jede TM kann durch ein Wort über $\{0,1\}^*$ repräsentiert werden, es gibt also nur abzählbar viele TMs und somit ist A auch abzählbar.

0 1 2

d)* Jede Sprache in NP ist unendlich.

Falsch, z.B. $\emptyset \in$ NP.

0 1 2

e)* Für jede Sprache $L \in$ NP gibt es eine Sprache $L' \in$ P und ein Polynom p, sodass $w \in L$ gilt, gdw. es ein u mit $w\#u \in L'$ und $|u| \leq p(|w|)$ gibt.

Wahr; da $L \in$ NP gibt es einen polynomiell beschränkten Verifikator M für L und wir können $L' := L(M)$ wählen, und p als das Polynom, das die Laufzeit von M beschränkt.

Aufgabe 7 **Reduktion (12 Punkte)**

In dieser Aufgabe betrachten wir reguläre Ausdrücke über einem Alphabet $\Sigma = \{a_1, ..., a_n\}$, für $n \in \mathbb{N}$. Die Länge $|r|$ eines regulären Ausdrucks r ist die Länge der üblichen Repräsentation über dem Alphabet $\Sigma \cup \{(,), *, |, \epsilon, \emptyset\}$. Z.B. hat der Ausdruck $(a_1|a_2)^*$ Länge 6. Wir verwenden r^k als Makro für die k-fache Konkatenation von r. Beachten Sie, dass $|r^k| = k\,|r|$, so hat etwa der Ausdruck $(a_1|a_2)^{10}$ Länge 50, **nicht** Länge 7.

a)* Ein Wort $w \in \Sigma^*$ ist eine *Permutation* von Σ genau dann, wenn es eine Bijektion $f : \{1, ..., n\} \to \{1, ..., n\}$ gibt, mit $w = a_{f(1)}a_{f(2)}...a_{f(n)}$. Für $n = 3$ wären z.B. $a_3a_1a_2$ und $a_1a_2a_3$ Permutationen, aber $a_2a_3a_3$, $a_3a_1a_2a_1$, und a_2a_1 nicht.

0 1 2 3 4 5 6

Geben Sie (für beliebiges n) einen regulären Ausdruck r_{kP} über Σ an, sodass $L(r)$ genau die Wörter enthält, die **keine** Permutationen sind, und r_{kP} Länge polynomiell in n hat. Sie können Σ als Makro für $(a_1 \mid a_2 \mid ... \mid a_n)$ verwenden (das Makro hat Länge $2n+1$) und $\bigcup_{i\in M} r_i$ anstelle von $r_{a_1} \mid ... \mid r_{a_k}$ schreiben, für eine beliebige Menge $M = \{a_1, ..., a_k\}$.

Es gibt drei Fälle, die wir abdecken müssen: Das Wort hat Länge echt kleiner als n, echt größer als n, oder ein Zeichen kommt zweifach vor.

$r_{kP} := (\Sigma \mid \epsilon)^{n-1} \mid \Sigma^{n+1}\Sigma^* \mid \bigcup_{i\in\{1,...,n\}} \Sigma^* a_i \Sigma^* a_i \Sigma^*$

b)* Wir betrachten das Nicht-Universalitätsproblem für reguläre Ausdrücke, was wir als RE-NONUNI bezeichnen. Es ist folgendermaßen definiert:

0 1 2 3 4 5 6

Eingabe: Ein regulärer Ausdruck r über $\Sigma = \{a_1, ..., a_n\}$.
Ausgabe: Ist $L(r) \neq \Sigma^*$?

Außerdem betrachten wir HAMILTON, also die Frage, ob ein Graph einen Hamilton-Kreis besitzt (siehe Folie 353).

Zeigen Sie HAMILTON $\leq_p$ RE-NONUNI, indem Sie eine entsprechende Reduktionsfunktion beschreiben und argumentieren, dass diese die notwendigen Bedingungen erfüllt.
Hinweis: Sie können Teilaufgabe a) verwenden und dürfen ohne Beweis annehmen, dass r_{kP} in polynomieller Zeit konstruiert werden kann.

Sei $G = (V, E)$ ein ungerichteter Graph mit $V = \{1, ..., n\}$. Ein Hamilton-Kreis ist eine Permutation der Knoten, bei denen sowohl aufeinanderfolgende Knoten adjazent sind, als auch der erste und der letzte Knoten. Wir erzeugen einen regulären Ausdruck r_G, der genau die Wörter erzeugt, die keinen Hamilton-Kreis repräsentieren.
Wir reduzieren also auf folgenden regulären Ausdruck r_G, wobei wir r_{kP} aus a) verwenden, um die Wörter zu erzeugen, die keine Permutationen sind.

$$r_G := r_{kP} \mid \bigcup_{(u,v)\notin E} \Sigma^* a_u a_v \Sigma^* \mid \bigcup_{(u,v)\notin E} a_u \Sigma^* a_v$$

Hierbei verwenden wir $\bigcup_{(u,v)\notin E} r_{uv}$ als Makro für $r_{u_1v_1} \mid ... \mid r_{u_mv_m}$ wobei $u_1v_1, ..., u_mv_m$ eine Auflistung aller Knotenpaare ist, die nicht adjazent sind.
Nun argumentieren wir, dass unsere Reduktion polynomiell berechenbar ist. Aus a) wissen wir, dass $|r_{kP}|$ polynomiell berechenbar ist. Da es höchstens n^2 viele Knotenpaare gibt, die nicht adjazent sind, ist r_G auch polynomiell berechenbar.
Es verbleibt zu zeigen, dass unsere Reduktion korrekt ist. Wenn G einen Hamilton-Kreis besitzt, entspricht dieser einer Permutation der Knoten, sodass sowohl aufeinanderfolgende, als auch der erste und letzte Knoten adjazent sind. Wir können diese Permutation als Wort w repräsentieren. Da w eine Permutation ist, gilt $w \notin L(r_{kP})$, und weiterhin folgt $w \notin L(r_G)$ daraus, dass w einem geschlossenen Pfad entspricht. Also ist $L(r_G) \neq \Sigma^*$.
Umgekehrt, wenn G keinen Hamilton-Kreis besitzt, kann kein Wort $w \in \Sigma^*$ einer Permutation entsprechen, die ein Hamilton-Kreis ist (da es keine solche Permutation gibt). Falls w keine Permutation ist, muss $w \in L(r_{kP})$ gelten, ansonsten gibt es zwei nichtadjazente Knoten in w, die aufeinander folgen, oder jeweils am Anfang und Ende sind, es folgt also $w \in L(r_G)$. Damit gilt $L(r_G) = \Sigma^*$.

Lösungbox Aufgabe 7 (fortlaufend)

Aufgabe 8 Die Sandwich-Sprache (12 Punkte)

a)* Sei $B := \{w \in \{a,b\}^* : |w|_a = |w|_b\}$. Geben Sie eine kontextfreie, nicht reguläre Sprache $A \subseteq \{a,b\}^*$ an und eine reguläre Sprache $L \subseteq \{a,b\}^*$, sodass $A \subseteq L \subseteq B$.

0 1 2 3 4 5 6

$A := \{(ab)^n(ba)^n : n \in \mathbb{N}\}$, $L := \{ab, ba\}^*$

b)* Sei $A := \{a^n b^n : n \in \mathbb{N}\}$, $B := \{w \in \{a,b\}^* : |w|_a = |w|_b\}$. Beweisen Sie, dass es **keine** reguläre Sprache $L \subseteq \{a,b\}^*$ mit $A \subseteq L \subseteq B$ gibt.

0 1 2 3 4 5 6

Sei L eine beliebige Sprache mit $A \subseteq L \subseteq B$. Wir zeigen nun, dass L nicht regulär ist, indem wir das Pumping-Lemma verwenden.
Sei also $n \in \mathbb{N}$ beliebig. Wir wählen $z := a^n b^n$. Da $z \in A \subseteq L$, gilt $z \in L$, und $|z| \geq n$ ist offensichtlich. Sei nun $z = uvw$ eine Zerlegung, die die Eigenschaften des PL erfüllt. Da $|uv| \leq n$ und $v \neq \varepsilon$, gilt $v = a^m$ für ein $m > 0$.
Nach PL muss $uv^2w \in L$ gelten, es gilt aber $uv^2w = a^{n+m}b^n \notin B$ und somit $uv^2w \notin L$.
Die Eigenschaft des PL kann also nicht erfüllt sein, und L ist somit nicht regulär.

Aufgabe 9 ResidualspRAchen (10 Punkte)

Sei $\Sigma := \{0, 1, \$\}$. Für eine beliebige Sprache $L \subseteq \Sigma^*$ und ein Wort $w \in \Sigma^*$ bezeichnen wir $L^w := \{u \in \Sigma^* : wu \in L\}$ als die *Residualsprache bezüglich* w von L. Anschaulich gesprochen enthält L^w also alle Wörter in L, die mit w beginnen, aber jeweils ohne das führende w.
Wir schreiben $\mathrm{RA} := \{L(M_w) \subseteq \Sigma^* : w \in \{0, 1\}^*\}$ für die Menge der rekursiv aufzählbaren / semi-entscheidbaren Sprachen über dem Alphabet Σ.

0 1 2 3 4

a)* Zeigen Sie: Für $L \in \mathrm{RA}$ und $w \in \Sigma^*$ ist $L^w \in \mathrm{RA}$.

Da $L \in \mathrm{RA}$ gibt es eine TM M mit $L(M) = L$. Sei M' die TM, die w vor die Eingabe hängt, und dann M ausführt. Da $x \in L^w$ äquivalent zu $wx \in L$ ist, gilt $L(M') = L^w$.

0 1 2 3 4 5 6

b)* Sei $\mathcal{H} := \{w\$x : w \in \{0, 1\}^*, x \in L(M_w)\}$ das (allgemeine) Halteproblem. Zeigen Sie: $\{\mathcal{H}^w : w \in \Sigma^*\} = \mathrm{RA}$.

Es gilt $\mathcal{H} \in \mathrm{RA}$, da wir die TM M_w auf Eingabe x simulieren können (mithilfe einer universellen Turingmaschine, wie in Satz 5.46). Nach a) gilt also $\{\mathcal{H}^w : w \in \Sigma^*\} \subseteq \mathrm{RA}$.

Für eine beliebige Sprache $L \in \mathrm{RA}$ gibt es eine TM M mit $L = L(M)$. Außerdem gibt es ein $w \in \Sigma^*$ mit $M_w = M$. Nun gilt

$$\mathcal{H}^{w\$} = \{x \in \Sigma^* : x \in L(M_w)\} = L$$

Also ist $\{L^w : w \in \Sigma^*\} \supseteq \mathrm{RA}$ und die Aussage gezeigt.

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