Lehrstuhl für Theoretische Informatik
Fakultät für Informatik
Technische Universität München

Eexam
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:**	Freitag, 6. August 2021
Prüfer:	Prof. Dr. Dr. h.c. Javier Esparza	**Uhrzeit:**	17:00 – 20:00

Bearbeitungshinweise

- Diese Klausur umfasst **18 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 20:15:00 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 20:15:00 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 zu technischen Problemen via Zulip oder E-Mail (theoleitung@in.tum.de) stellen. Inhaltliche Fragen werden wir nicht beantworten; falls Ihnen eine Aufgabenstellung mehrdeutig erscheint, notieren Sie bitte Ihre Interpretation der Aufgabe.
- 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 (12 Punkte)**

Für die folgenden Fragen ist eine Begründung nur erforderlich, wenn explizit danach gefragt wird.

0 1 2

a)* Geben Sie Sprachen $A, B, C \subseteq \{a, b, c\}^*$ mit $B \neq C$ und $AB = CA$ an.

0 1 2

b)* Geben Sie einen NFA M mit höchstens drei Zuständen an, der eine Sprache L mit $L = ab\,L \cup \{c\}$ akzeptiert.

0 1 2

c)* Geben Sie eine kontextfreie Grammatik für eine Sprache L mit $L = \{a\}L^*\{b\} \cup \{bb\}$ an, mit höchstens 4 Produktionen. Beachten Sie, dass z.B. $S \to SS \mid \varepsilon$ zwei Produktionen sind.

0 1 2 3

d)* Welche Sprachen können von kontextfreie Grammatiken über einem Alphabet Σ erzeugt werden, bei denen die rechte Seite jeder Produktion Länge 1 hat? Beachten Sie $|\varepsilon| = 0$.

0 1 2 3

e)* Seien $M_1 = (Q, \Sigma, \delta, q_0, F_1)$, $M_2 = (Q, \Sigma, \delta, q_0, F_2)$ zwei beliebige DFAs, die sich nur in ihren Finalzuständen unterscheiden, mit $F_1 \neq F_2$ und $L(M_1) = L(M_2)$. Gibt es solche M_1, M_2, in denen alle Zustände erreichbar sind? Wenn ja, geben Sie ein Beispiel an, wenn nein, begründen Sie dies kurz.

Aufgabe 2 CYK-Algorithmus (10 Punkte)

Die Grammatik G ist eine kontextfreie Grammatik in CNF mit 2 Terminalen (f, b) und 2 Nichtterminalen (F, B) **wobei F das Startsymbol ist**.

In der folgenden CYK-Tabelle für das Wort *fbbff* und die Grammatik G sind die unteren beiden Zeilen bereits ausgefüllt.

1,5				
1,4	2,5			
1,3	2,4	3,5		
1,2 F	2,3 —	3,4 F	4,5 B	
1,1 F	2,2 B	3,3 B	4,4 F	5,5 F
f	b	b	f	f

a)* Geben Sie die Grammatik G an.

b) Vervollständigen Sie die CYK-Tabelle am Anfang der Aufgabe. Sie können entweder die gesamte Tabelle in dem nachfolgenden Antwortbereich angeben oder direkt in die Tabelle oben schreiben.

c)* Gilt $fbbff \in L(G)$? Begründen Sie Ihre Antwort *kurz* mit Hilfe der ausgefüllten Tabelle aus Teilaufgabe b).
Falls Sie Teilaufgabe b) nicht bearbeitet haben, dürfen Sie annehmen, dass in jeder Zelle in den oberen 3 Zeilen der Tabelle exakt das Nichtterminal F steht.

0
1

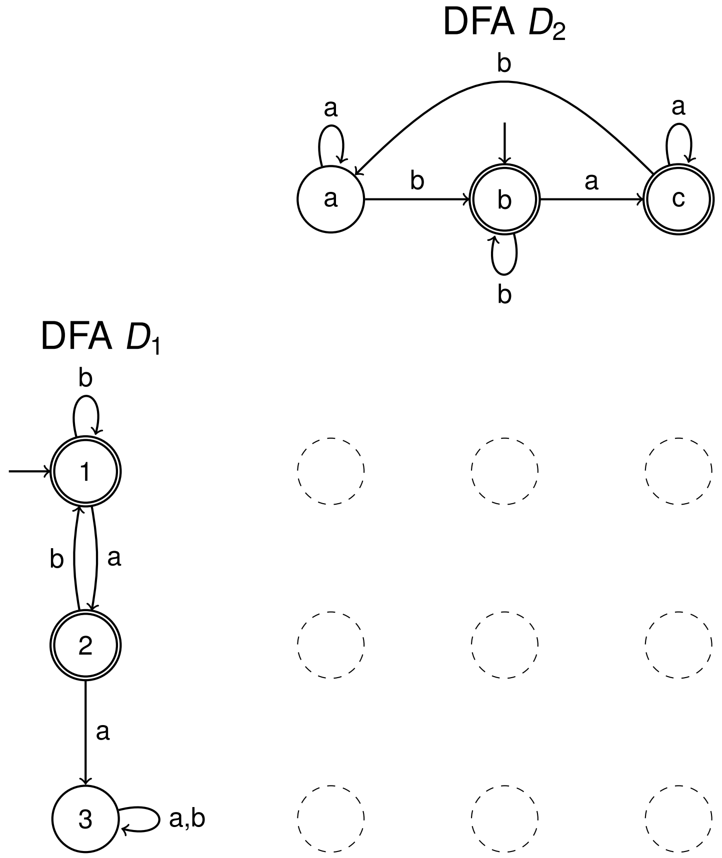
d)* Gilt $bbf \in L(G)$? Begründen Sie Ihre Antwort *kurz* mit Hilfe der ausgefüllten Tabelle aus Teilaufgabe b). Falls Sie Teilaufgabe b) nicht bearbeitet haben, dürfen Sie annehmen, dass in jeder Zelle in den oberen 3 Zeilen der Tabelle exakt das Nichtterminal F steht.

0
1
2

e)* Aus einer ausgefüllten CYK-Tabelle kann man den Syntaxbaum für ein Wort gewinnen. Wie würden Sie den CYK-Algorithmus abändern, um den Syntaxbaum für ein Wort effizienter ablesen zu können?

Aufgabe 3 **Produktkonstruktion (8 Punkte)**

a)* Berechnen Sie mit Hilfe der Produktkonstruktion einen DFA für $\{w \in \{a,b\}^* \mid w \in L(D_1) \wedge w \in L(D_2)\}$. Zeichnen Sie direkt in die Aufgabenstellung. Wenn Sie zusätzlichen Platz benötigen, finden Sie leere Seiten am Ende der Klausur.



b)* Beschreiben Sie, wie man die Produktkonstruktion abändern kann, um aus 2 DFAs $D_1 = (Q_1, \Sigma, \delta_1, q_0, F_1)$ und $D_2 = (Q_2, \Sigma, \delta_2, q_0', F_2)$ einen DFA zu gewinnen, der die Sprache $\{w \in \Sigma^* \mid w \notin L(D_1) \vee w \in L(D_2)\}$ akzeptiert.

Aufgabe 4 **Kontextfreie Sprachen (10 Punkte)**

0
1
2
3
4
5
6
7
8
9
10

Ist L kontextfrei? Wenn ja, geben Sie eine kontextfreie Grammatik an, wenn nein, widerlegen Sie es.

a)* $L := M \cap M^R$, wobei $M := \{b^i a^j b^k : 0 \leq i < j\}$.

b)* $L := \{b^i a^j b^k : 0 \leq i < j \vee 0 \leq k < j\}$

Aufgabe 5 PDA Konstruktion (14 Punkte)

Sei G die Grammatik mit Produktionen

$$S \to bS \mid aSca \mid aabS \mid \varepsilon$$

a)* Konstruieren Sie einen zu G äquivalenten Kellerautomaten mit einem Zustand. Verwenden Sie dazu eine aus der Vorlesung oder den Übungen bekannte Notation. Ihr PDA soll über leeren Keller akzeptieren.

0 1 2 3 4 5 6

b)* Finden Sie das lexikographisch erste Wort $w \in L(G)$ mit Länge 58. Begründen Sie Ihre Antwort kurz. Sie können w^k als Abkürzung für die k-fache Konkatenation verwenden, für ein Wort w und $k \in \mathbb{N}$.
Zur Erinnerung: $a <_{\text{lex}} b <_{\text{lex}} c$, und für zwei unterschiedliche Wörter u, v mit gleicher Länge gilt $u <_{\text{lex}} v$ genau dann, wenn $u_i <_{\text{lex}} v_i$ an der ersten Stelle i, an der sie sich unterscheiden. Z.B. gilt also $abac <_{\text{lex}} abba$.

0 1 2

c)* Konstruieren Sie einen **deterministischen** Kellerautomaten M, mit $L(M) = L(G')$, wobei G' die Grammatik mit den Produktionen $S \to aSba \mid aS \mid c$ ist. Ihr Automat soll über Finalzustände akzeptieren und höchstens 5 Zustände und 4 Kellersymbole haben.

0 1 2 3 4 5 6

Aufgabe 6 Entscheidbarkeit und Komplexität (12 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)* Wenn $L \subseteq \{a, b\}^*$ entscheidbar ist, dann ist L^a entscheidbar. Hier bezeichnet $L^a := \{w : aw \in L\}$ die Residualsprache bezüglich a von L.

0 1 2

b)* Seien M_1, M_2 beliebige Turingmaschinen, und L eine Sprache mit $L(M_1) \subseteq L \subseteq L(M_2)$. Dann ist L semi-entscheidbar.

0 1 2

c)* Es gibt keine reguläre Sprache $L \in$ NP.

0 1 2 3

d)* Sei L entscheidbar und M eine NTM. Dann ist $L(M) \setminus L$ rekursiv aufzählbar.

0 1 2 3

e)* Für NP-vollständige Sprachen $L_1, L_2 \subseteq \{0, 1\}^*$ ist $\{w_1\$w_2 : w_1 \in L_1, w_2 \in L_2\}$ stets NP-hart/NP-schwierig.

Aufgabe 7 **Reduktion (12 Punkte)**

In dieser Aufgabe betrachten wir reguläre Ausdrücke über einem Alphabet $\Sigma = \{0, 1\}$. 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 $(0|1)^*$ 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 $(0|1)^{10}$ Länge 50, **nicht** Länge 7.

a)* Sei $V = \{x_1, x_2, \dots, x_9\}$ eine Menge von booleschen Variablen. Wir identifizieren eine Belegung von V mit einem Wort $w \in \{0, 1\}^9$ der Länge 9. Hierfür legen wir also die Reihenfolge $x_1, \dots, x_9$ fest, sodass w_i mit der Belegung von x_i korrespondiert. Sei F die aussagenlogische Formel

0 1 2

$$F = (x_2 \vee x_5 \vee \neg x_7) \wedge (x_3 \vee \neg x_4 \vee \neg x_7)$$

Geben Sie einen regulären Ausdruck r_F mit $|r_F| \leq 150$ an, sodass $L(r_F)$ genau die Belegungen enthält, die F **nicht** erfüllen. Sie können Σ als Makro für $(0\,|\,1)$ verwenden (das Makro hat Länge 5).

b)* Geben Sie einen regulären Ausdruck r_V mit $|r_V| \leq 150$ für die Sprache $\{w \in \{0, 1\}^* : |w| \neq 9\}$ an.

0 1 2

c)* 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 7 8

Eingabe: Ein regulärer Ausdruck r über $\Sigma = \{0, 1\}$.
Ausgabe: Ist $L(r) \neq \Sigma^*$?

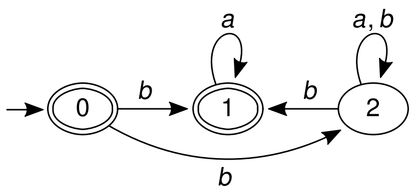
Zeigen Sie 3KNF-SAT $\leq_p$ RE-NONUNI, indem Sie eine entsprechende Reduktionsfunktion beschreiben und argumentieren, dass diese die notwendigen Bedingungen erfüllt.

Lösungbox Aufgabe 7 (fortlaufend)

Aufgabe 8 **Fast Azyklisch (10 Punkte)**

Ein NFA ist *fast azyklisch*, wenn alle Zyklen Schlingen sind. Formal: Ein NFA $A = (Q, \Sigma, \delta, q_0, F)$ ist fast azyklisch wenn es keine Zustände q, q' mit $q \neq q'$ gibt, sodass man von q aus q' erreichen kann, und von q' aus q erreichen kann.
Ein regulärer Ausdruck r ist *fast azyklisch*, wenn er in der Form $r = t_1 \mid t_2 \mid ... \mid t_k$ ist, wobei $t_1, ..., t_k$ reguläre Ausdrücke sind, die das Symbol $\mid$ nicht beinhalten. Beispielsweise ist $a^*b(ab^*a)^* \mid ba^*b \mid \epsilon$ fast azyklisch, und $a^*b(a \mid b)^*$ nicht.

a)* Geben Sie einen einen fast azyklischen regulären Ausdruck r an, sodass $L(r)$ die von diesem NFA akzeptierten Sprache ist:



b)* Sei M ein beliebiger, fast azyklischer NFA. Konstruieren Sie einen fast azyklischen regulären Ausdruck r mit $L(r) = L(M)$ und begründen Sie, wieso Ihre Konstruktion korrekt ist.

Aufgabe 9 **Alle kommen vor (12 Punkte)**

Sei $\Sigma_n = \{1, 2, \dots, n\}$. Sei $L_n \subseteq \Sigma_n^*$ die Menge der Wörter, in denen alle Buchstaben von Σ_n mindestens einmal vorkommen. Z.B. gilt $123, 2213, 333312 \in L_3$ und $11122, \varepsilon, 23, 33232323 \notin L_3$.

0 1 2

a)* Konstruieren Sie einen ϵ-NFA für $\overline{L_3}$ mit höchstens 4 Zuständen.

0 1 2 3 4 5

b)* Zeigen Sie: Jeder DFA für L_n hat mindestens 2^n Zustände.
Hinweis: Wenn Sie Teilaufgabe c) lösen, erhalten Sie die Punkte für b) automatisch.

c)* Zeigen Sie: Jeder NFA für L_n hat mindestens 2^n Zustände.
Hinweis: Sie dürfen folgende Aussage ohne Beweis verwenden. Für zwei beliebige Wörter $u, v \in \Sigma_n^*$, in denen jedes Zeichen genau einmal vorkommt, und beliebige Zerlegungen $u = u_1 u_2$, $v = v_1 v_2$ mit $u_1 \neq v_1$, gilt $u_1 v_2 \notin L_n$ oder $v_1 u_2 \notin L_n$.

0
1
2
3
4
5

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