Lehrstuhl für Logik und Verifikation
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.

Theoretische Informatik

Klausur:	IN0011 / Wiederholung	**Datum:**	Dienstag, 1. Oktober 2019
Prüfer:	Prof. Dr. Tobias Nipkow	**Uhrzeit:**	08:00 – 11:00

	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 Prüfung beträgt 60 Punkte.
- Das Heraustrennen von Seiten aus der Prüfung ist untersagt.
- Als Hilfsmittel sind zugelassen:
  - ein **nicht-programmierbarer Taschenrechner**
  - ein **analoges Wörterbuch** Deutsch ↔ 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.

Hörsaal verlassen von ________ bis ________ / Vorzeitige Abgabe um ________

Aufgabe 1 **Quiz 1: Reguläre Sprachen, kontextfreie Sprachen, Sprachen allgemein (9 Punkte)**

0 1 2 3 4 5 6 7

a)* Geben Sie für jede der folgenden Aussagen an, ob diese korrekt ist oder nicht. Begründen Sie ihre Antwort jeweils *kurz*. Antworten ohne Begründung werden nicht gewertet. Falls eine Aussage nicht gilt, geben Sie (soweit angebracht) entsprechende Gegenbeispiele an. Antworten wie „Ja, siehe Vorlesung“ sind ohne weiteren Kontext *nicht* ausreichend.

1. Seien $A, B \subseteq \Sigma^*$ Sprachen. Dann ist $|AB| = |A| \cdot |B|$.
2. Zu jeder regulären Sprache L gibt es eine nicht mehrdeutige Grammatik, die L erzeugt.
3. Jede endliche Teilmenge einer kontextfreien Sprache ist kontextfrei.
4. Sei $A \subseteq \Sigma^*$ eine Sprache. Dann gibt es höchstens eine unendlich große Myhill–Nerode-Klasse bezüglich $\equiv_A$.
5. Sei M ein NFA über dem Alphabet Σ, in dem jeder Zustand ein Endzustand ist. Dann akzeptiert M alle Wörter in Σ^*.
6. Sei $A \subseteq \Sigma^*$ regulär. Dann ist $\{w \mid w \in A \land w^R \in A\}$ ebenfalls regulär ($(w_1 \dots w_n)^R = w_n \dots w_1$ bezeichnet die Spiegelung von w).
7. Seien $A, B \subseteq \Sigma^*$ regulär mit $\varepsilon \notin A$. Dann sind alle Lösungen $X \subseteq \Sigma^*$ der Gleichung $X = AX \cup B$ regulär.

b)* Nennen Sie jeweils ein Beispiel für die folgenden Objekte:

0
1
2

1. Eine mehrdeutige kontextfreie Grammatik.
2. Eine deterministisch kontextfreie Sprache (DCFL), die nicht von einem mit leerem Keller akzeptierenden DPDA akzeptiert werden kann.

Begründen Sie jeweils, warum das Objekt die jeweilige Eigenschaft hat.

Aufgabe 2 **CYK (6 Punkte)**

Gegeben ist die Grammatik $\mathsf{G} = (\{A, B, S, T, U, X, Y\}, \{a, b, c\}, P, S)$ mit den folgenden Produktionen P:

$$
\begin{array}{ll}
S \to TU \mid UT & U \to AY \\
T \to AX \mid BT \mid BA & Y \to BU \mid b \\
X \to AX \mid c & \\
A \to a & B \to b
\end{array}
$$

0 1 2 3 4

a)* Bestimmen Sie mithilfe des CYK-Algorithmus ob $w = baacab$ in $\mathsf{L}(\mathsf{G})$ liegt. Geben Sie die vollständige CYK-Tabelle an. Halten Sie sich dabei strikt an die Darstellung aus den Übungen und begründen Sie, warum aus Ihrer Tabelle hervorgeht, dass das Wort in der Sprache ist bzw. nicht.

0 1 2

b)* Zeichnen Sie für $w = baacab$ einen Ableitungsbaum bezüglich G.

Aufgabe 3 Rekursion auf regulären Ausdrücken (9 Punkte)

Sei $a \in \Sigma$ ein beliebiges, aber festes Zeichen. Für eine Sprache $A \subseteq \Sigma^*$ definieren wir das Prädikat P, das angibt, ob es ein Wort in A gibt, das das Zeichen a enthält. Formal:

$$P(A) \longleftrightarrow \exists w{\in}A.\ |w|_a \geq 1$$

a)* Finden Sie für $A, B \subseteq \Sigma^*$ einen einfachen Ausdruck für $P(AB)$ in Abhängigkeit von $P(A)$ und $P(B)$. Beweisen Sie Ihre Behauptung. 0 1 2

Vorsicht: Vergessen Sie nicht den Fall, dass eine der Sprachen leer ist.

0
1
2

b)* Geben Sie eine rekursive Definition eines Prädikats $Q(r)$ auf regulären Ausdrücken an, sodass $Q(r) \longleftrightarrow P(\mathsf{L}(r))$.
Sie dürfen hierbei auf das aus der Übung bekannte Prädikat is_empty(r) zurückgreifen, das bestimmt, ob die zu einem regulären Ausdruck r gehörende Sprache leer ist; d.h. is_empty(r) $\longleftrightarrow \mathsf{L}(r) = \emptyset$.

c) Beweisen Sie die Korrektheit Ihrer Konstruktion durch strukturelle Induktion über r. Sie müssen dabei *nur* die Fälle $r_1 \mid r_2$ (Vereinigung) und $r = r_1 r_2$ (Konkatenation) behandeln und dürfen die anderen 4 Fälle weglassen.

Geben Sie jeweils klar an, welche Aussage zu zeigen ist und ggf. was die Induktionshypothesen sind und wo Sie sie verwenden.

Sie dürfen Ihr Ergebnis aus Aufgabe a) verwenden, selbst wenn Sie es nicht beweisen konnten. Auch hier müssen Sie aber angeben, wo Sie es genau verwenden.

0 1 2 3 4

Aufgabe 4 **NFA Konstruktion (5 Punkte)**

Wir betrachten die Sprache der Wörter die durch das Löschen genau eines Zeichens aus einem fixen Wort w hervorgehen:

$$\mathsf{L}_w = \{uv \mid u, v \in \Sigma^* \wedge \exists c \in \Sigma.\ ucv = w\}$$

Wobei Σ ein gegebenes Alphabet ist.

0 1 2

a)* Sei $\Sigma = \{$t,h,e,o$\}$. Zeichnen Sie einen ε-NFA, der die Sprache L_{theo} akzeptiert. Der Automat darf **maximal 10 Zustände** umfassen.

b)* Seien nun Σ und $w = w_1 \dots w_n \in \Sigma^*$ beliebig. Konstruieren Sie einen ε-NFA $\mathsf{M_w}$, der die Sprache L_w akzeptiert, indem Sie explizit das Tupel $\mathsf{M_w} = (Q, \Sigma, \delta, q_0, F)$ angeben. Informelle Beschreibungen des Automaten werden nicht bewertet. Der Automat darf **maximal $\mathbf{2(n+1)}$ Zustände** umfassen.

0
1
2
3

Aufgabe 5 Pumping Lemma (8 Punkte)

Sei $\Sigma = \{a, b\}$. Wir betrachten die folgende Sprache über dem Alphabet $\Sigma' = \{a, b, \#\}$:

$$\mathsf{L} = \{p\#w \mid p, w \in \Sigma^* \wedge p \text{ ist ein Präfix von } w\}$$

Dabei ist p genau dann ein Präfix von w, wenn es ein $v \in \Sigma^*$ gibt mit $pv = w$.

0 1 2 3 4 5 6

a)* Zeigen Sie, dass L nicht kontextfrei ist, indem Sie einen Widerspruchsbeweis unter Verwendung des Pumping Lemmas für kontextfreie Sprachen führen.

b)* Geben Sie eine unendliche Menge von Myhill–Nerode Äquivalenzklassen für die Sprache L an. Es reicht hierbei aus, für jede Klasse jeweils einen Repräsentanten anzugeben. Zeigen Sie außerdem, dass die Klassen paarweise verschieden sind.

0
1
2

Aufgabe 6 **Kontextfreie Sprachen (2) (6 Punkte)**

Wir betrachten die Sprache $\mathsf{L} = \{\mathsf{a}^k\mathsf{b}^k \mid k \geq 0\}$ und die Grammatik $G = (\{S\}, \{\mathsf{a}, \mathsf{b}\}, P, S)$ mit den folgenden Produktionen P:

$$S \to \varepsilon \mid \mathsf{a}S\mathsf{b}$$

Zeigen Sie, dass G genau die Sprache L produziert, also $\mathsf{L}(G) = \mathsf{L}$.

Aufgabe 7 Unabhängige Knotenmenge (8 Punkte)

In einem beliebigen ungerichteten Graph $G = (V, E)$ nennen wir eine Knotenmenge $S \subseteq V$ unabhängig, falls keine zwei Knoten aus S eine Kante aus E bilden. Wir betrachten das Entscheidungsproblem UK:
Gegeben: Ein ungerichteter Graph $G = (V, E)$ und eine natürliche Zahl k.
Zu entscheiden: Gibt es eine unabhängige Teilmenge von Knoten $S \subseteq V$ mit $|S| \geq k$?

a)* Zeigen Sie, dass UK $\in$ NP.

0
1

b)* Zeigen Sie, dass UK NP-schwer ist, indem Sie eine polynomielle Reduktion von 3KNF-SAT auf UK angeben und deren Korrektheit beweisen. Führen Sie Ihre Reduktion auch explizit für die Formel $(x_1 \vee \neg x_3 \vee x_4) \wedge (\neg x_1 \vee x_2 \vee x_3)$ durch.
Hilfestellung: Verwenden Sie genau einen Knoten für jedes Vorkommen eines Literals in jeder Klausel (also höchstens $3m$ Knoten bei m Klauseln).

0
1
2
3
4
5
6
7

Aufgabe 8 Quiz 2: Berechenbarkeit und Komplexität (9 Punkte)

a)* Geben Sie für jede der folgenden Aussagen an, ob diese korrekt ist oder nicht. Begründen Sie jeweils *kurz*. Antworten ohne Begründung werden nicht gewertet. Falls eine Aussage nicht gilt, geben Sie (soweit angebracht) entsprechende Gegenbeispiele an. Antworten wie „Ja, siehe Vorlesung" sind ohne weiteren Kontext *nicht* ausreichend.

0 1 2 3 4 5

1. Die Sprache $A := \{w \mid \varphi_w = \chi_{H_0}\}$ ist entscheidbar. Wie aus der Vorlesung bekannt bezeichnet dabei φ_w die von der durch w kodierten Turingmaschine berechnete Funktion, und χ_{H_0} die charakteristische Funktion von H_0.
2. Der Wertebereich $\{y \in \Sigma^* \mid \exists x \in \Sigma^*.\ y = f(x)\}$ einer berechenbaren Funktion f ist semi-entscheidbar.
3. Für jedes Problem A in P gilt $A \leq_{\mathsf{p}} \mathsf{SAT}$.
4. Jede unentscheidbare Sprache hat eine unentscheidbare echte Teilmenge.
5. Sei $A \subseteq \Sigma^*$ eine reguläre Sprache. Dann ist A in deterministisch-linearer Zeit entscheidbar, d.h. es gibt eine DTM M mit $L(M) = A$ und Konstanten $a, b > 0$ mit $\forall w \in \Sigma^*.\ \mathrm{time}_M(w) \leq a\,|w| + b$.

0
1
2
3
4

b)* Nennen Sie jeweils ein Beispiel für die folgenden Objekte:

1. Eine Sprache, die nicht in NP ist.
2. Eine totale Funktion, die nicht berechenbar ist.
3. Zwei unentscheidbare Sprachen, deren Schnitt entscheidbar ist.
4. Eine Sprache, die nicht Typ 0 in der Chomsky-Hierarchie hat.

Begründen Sie jeweils *kurz*, warum ihr Beispiel diese Eigenschaften hat (nur „laut Vorlesung“ ist *nicht* ausreichend). Die Unentscheidbarkeit von aus der Vorlesung als unentscheidbar bekannten Sprachen müssen Sie nicht begründen.

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