Die Anzahl der Aufgaben, das Punkteschema, die Themen-
schwerpunkte, etc. können sich in der echten Klausur unterscheiden!

Universität Osnabrück / FB6 / Theoretische Informatik Prof. Chimani, Beyer

Informatik D: Einführung in die Theoretische Informatik

PROBEKLAUSUR — SoSe 2016

Gruppe: Paradox

Unbedingt ausfüllen

Matrikelnummer | Studiengang/Abschluss | Fachsemester

Nachname | Vorname

Unterschrift | Identifikator (Beliebiges Wort zur Identifikation im anonymen Notenaushang)

Grundregeln

- Die Bearbeitungszeit der Klausur beträgt **120 Minuten**.
- Sie schreiben diese Klausur **vorbehaltlich** der Erfüllung der **Zulassungsvoraussetzung**. Das heißt: Wir werden Ihre Zulassung vor Korrektur prüfen; die Tatsache, dass Sie die Klausur mitschreiben, bedeutet keine implizite Zulassung.
- Es sind **keine Unterlagen** und auch **keine** anderen **Hilfsmittel** erlaubt.
- Benutzen sie nur dokumentenechten (blauen oder zur Not schwarzen) **Kugelschreiber**! Bleistiftlösungen werden nicht gewertet!
- Es zählt die Antwort, die sich im dafür vorgesehenen Kästchen befindet! Soll eine andere Antwort gewertet werden, so ist diese **eindeutig** zu kennzeichnen! Falsche Kreuzchen können zu Punkteabzug innerhalb der Teilaufgabe führen.
- Jegliches Schummeln, und auch der Versuch desselben, führt zum Ausschluss von der Klausur und einer Bewertung mit **5,0**.

Wird vom Korrektor/Prüfer ausgefüllt

Aufgabe	1	2	3	4	5	6	7	8	9	Σ
Punkte (max)	24	8	8	18	10	12	18	12	22	**132**
Punkte (erreicht)										

Punkte	0..65	66..73	74..81	82..86	87..91	92..96	97..101	102..106	107..112	113..119	120..132
Note	**5,0**	**4,0**	**3,7**	**3,3**	**3,0**	**2,7**	**2,3**	**2,0**	**1,7**	**1,3**	**1,0**

Note:

Aufgabe 1: Sprachen **(24 Punkte)**

(a) Sprachfamilien und Automaten (8 Punkte)

Zu jeder Sprachfamilie gibt es entsprechende Automaten.
Vervollständigen Sie die folgende Tabelle:

Name der Sprachfamilie		regulär	
Automaten	DTM, NTM		DKA mit Akz. durch

(b) Normalformen (4 Punkte)

Definieren Sie Chomsky- und Greibach-Normalform.

Chomsky-Normalform	*Greibach-Normalform*

(c) Grammatikregeln (4 Punkte)

Zu welcher Sprachfamilie sind die Normalformen in Aufgabe (b) äquivalent?

Welche Form haben die Grammatiken dieser Sprachfamilie im Allgemeinen?

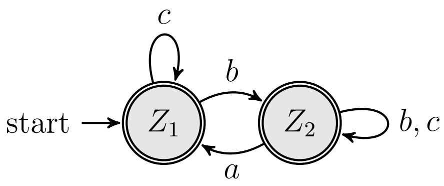
(d) Sprachen klassifizieren (8 Punkte)

Zu welcher Sprachklasse gehören die folgenden Sprachen? Kreuzen Sie dabei *alle* korrekten Antworten an. (Hinweis: Typ 3 ist regulär.)

Typ:	3	2	1	0
$L_1 \cup L_2$, wobei L_1 regulär und L_2 kontextsensitiv	☐	☐	☐	☐
$\{www \mid w \in \{a, c, e\}^*\}$	☐	☐	☐	☐
$\{\mathbb{W}(P) \mid P$ ist eine Ja-Instanz für das PCP$\}$, wobei $\mathbb{W}(X)$ eine beliebige PCP-Instanz X als Wort kodiert.	☐	☐	☐	☐
$\{w \in \Sigma^* \mid$ es gibt eine MPCP-Instanz mit w als Lösungswort$\}$	☐	☐	☐	☐

Aufgabe 2: Umwandeln **(8 Punkte)**

Wandeln Sie—gemäß dem Vorgehen aus der Vorlesung!—den folgenden deterministischen endlichen Automaten in einen regulären Ausdruck um:



Aufgabe 3: Deterministische Kellerautomaten **(8 Punkte)**

Geben Sie einen DKA für $\{w \mid w \in \{(,)\}^* \text{ ist ein korrekt geklammerter Ausdruck}\}$ an.

Aufgabe 4: Pumping Lemma **(18 Punkte)**

(a) Definition (4 Punkte)

Wie lautet das Pumping Lemma für reguläre Sprachen?

Sei L eine reguläre Sprache. Dann...

(b) Beweisidee (2 Punkte)

Wir betrachten einen DEA zu L. Um das Pumping Lemma zu beweisen, wollen wir, dass unsere betrachteten Wörter aus L im DEA über einen Kreis laufen müssen.
Wie erreichen wir das?

(c) Anwendung (12 Punkte)

Beweisen Sie, dass $L := L_1 \cup L_2$ mit $L_1 := \{a^i b^{i/2} \mid i \in \mathbb{N}_g\}$ und $L_2 := \{a^j b^{2j} \mid j \in \mathbb{N}_u\}$ keine reguläre Sprache ist.

Aufgabe 5: Zusammenhänge **(10 Punkte)**

Welche Aussagen stimmen?

Achtung: Pro Frage gibt es 2/0/−1 Punkte bei einer richtigen/keinen/falschen Antwort! Insgesamt gibt es jedoch keine negativen Punkte innerhalb Aufgabe 5.

korrekt	falsch	
☐	☐	Sei L eine kontextsensitive Sprache. Dann können die Eigenschaften des Pumping Lemmas für reguläre Sprachen auf keinen Fall erfüllt sein.
☐	☐	Man kann zeigen, dass reguläre Ausdrücke mindestens so mächtig sind wie reguläre Grammatiken, indem man zeigt, wie man jeden regulären Ausdruck in einen äquivalente reguläre Grammatik umwandeln kann.
☐	☐	Man kann zeigen, dass WHILE-Programme mindestens so mächtig sind wie Kellerautomaten, indem man zeigt, wie man jeden Kellerautomaten in ein äquivalentes WHILE-Programm umwandeln kann.
☐	☐	Es ist ein offenes Problem, ob NDTMen, die nur innerhalb der Grenzen des Eingabewortes arbeiten, die kontextsensitiven Sprachen beschreiben.
☐	☐	Falls $\mathsf{P} \neq \mathsf{NP}$, gibt es keinen Polynomialzeitalgorithmus, der entscheidet, ob eine gegebene Variablenbelegung eine gegebene logische Formel erfüllt.

Aufgabe 6: Rechnende Turingmaschine **(12 Punkte)**

Gegeben eine unär kodierte Zahl α. Geben Sie eine Turingmachine an, die die folgende Funktion berechnet:

$$f(\alpha) := \begin{cases} \alpha/2 & \alpha \in \mathbb{N}_g \cup \{0\} \\ \text{undef} & \text{sonst} \end{cases}$$

Aufgabe 7: Grundlagen **(18 Punkte)**

(a) Algorithmik (4 Punkte)

Beschreiben Sie die Bedeutung der Begriffe *Ja-Instanz* und *Nein-Instanz*.

Sei $G = (V, E)$ ein (nicht notwendig zusammenhängender) Graph. Welche Laufzeit (in $\mathcal{O}$-Notation) hat eine Tiefensuche auf G?

(b) Berechenbarkeit (8 Punkte)

Definieren Sie den Begriff *Turing-vollständig*.

Nennen Sie drei Turing-vollständige Programmiersprachen und begründen Sie, warum sie auf einem realen Computer *formal* nicht Turing-vollständig sind.

Warum ist die Funktion $S(n)$, die die Anzahl der Schritte eines n-Busy-Beavers angibt, nicht berechenbar?

(c) Komplexitätstheorie (6 Punkte)

Was besagt das Cook-Levin-Theorem?

Definieren Sie die Komplexitätsklasse ***P***.

Was ist ***NPI***?

Aufgabe 8: Unentscheidbarkeit (12 Punkte)

Zeigen Sie mithilfe des Wortproblems, dass es unentscheidbar ist, ob die Sprache $\mathcal{L}(M)$ für eine gegebene akzeptierende Turingmaschine M regulär ist.

Nutzen Sie die folgenden Beweisschnipsel. Genau ein Schnipsel pro Zeile ist richtig. Geben Sie eine Reihenfolge der richtigen Schnipsel an, indem Sie Zahlen 1 bis 8 in die Felder schreiben.

	$N(w)$ akzeptiert genau dann, wenn $w \in \{0^i1^i \mid i \in \mathbb{N}\}$ (nicht regulär).		Wenn $N(w)$ akzeptiert, dann ist $\mathcal{L}(N') = \Sigma^*$. Wenn $N(w)$ nicht akzeptiert, dann ist $\mathcal{L}(N') = \{0^i1^i \mid i \in \mathbb{N}\}$.		
	Dann existiert ein Algorithmus $\mathcal{A}$ mit $\mathcal{A}(w) := \begin{cases} 1 & \text{wenn } w \text{ regulär,} \\ 0 & \text{sonst.} \end{cases}$		Dann existiert ein Algorithmus $\mathcal{A}$ mit $\mathcal{A}(M) := \begin{cases} 1 & \text{wenn } \mathcal{L}(M) \text{ regulär,} \\ 0 & \text{sonst.} \end{cases}$		
	• Konstruiere akzeptierende TM $N'(x)$, die: — Überprüft, ob $x \in \{0^i1^i \mid i \in \mathbb{N}\}$. — Falls ja, akzeptiert. — Falls nein, $N(w)$ simuliert.		• Konstruiere akzeptierende TM $N'(x)$, die: — Überprüft, ob $x \in \{0^i1^i \mid i \in \mathbb{N}\}$. — Falls ja, akzeptiert. — Falls nein, nicht akzeptiert.		
	Widerspruch zur Semientscheidbarkeit des Wortproblems.		Widerspruch zur Unentscheidbarkeit des Wortproblems.		Widerspruch zu $1 - \mathcal{B}$.
	• `return` $\mathcal{A}(N')$;		• `return` $N'(w)$;		• `return` $1 - \mathcal{B}(N', w)$;
	Angenommen, $\mathcal{L}(M)$ wäre regulär.		Angenommen, das Wortproblem über $\mathcal{L}(M)$ wäre entscheidbar.		Angenommen, es wäre entscheidbar, ob $\mathcal{L}(M)$ regulär ist.
	Definiere $\mathcal{B}(N, \mathcal{B}(N)) :=$		Definiere $\mathcal{B}(N, w) :=$		Definiere Turingmaschine N mit Eingabe w:
	Es folgt: $\mathcal{B}(N, w) = 1$ $\iff \mathcal{A}(N') = 1$ $\iff \mathcal{L}(N')$ regulär $\iff N$ akzeptiert w		Es folgt: $\mathcal{B}(N, w) = 1$ $\iff \mathcal{A}(N') = 1$ $\iff w \in \mathcal{L}(N')$ $\iff N$ akzeptiert w		Es folgt: N akzeptiert w $\iff \mathcal{A}(N') = 1$ $\iff N'$ akzeptiert w $\iff \mathcal{L}(N)$ regulär

Raum für Notizen (unbewertet)

Aufgabe 9: *NP*-Vollständigkeit (22 Punkte)

Filmmaterial ist teuer. Als Produktionsfirma, die auf traditionellen Filmrollen dreht, haben wir von alten Filmen noch Reststücke von Filmrollen. Diese können wir zusammenkleben um eine neue Filmrolle zu erhalten. Uns ist vorher bekannt, wie lang die neue Filmrolle sein soll. Beim Zusammenkleben von Restmaterialien wird die neue Filmrolle gegebenenfalls zu lang, d. h. wir haben einen Überschuss. Wir definieren daher das *(Film-)Rollenklebeproblem* wie folgt:

Gegeben:
- n Rollenreststücke mit unterschiedlichen Längen $\ell_1, \ldots, \ell_n$,
- eine Gesamtlänge F für eine neue Filmrolle.

Minimiere den Überschuss, der entsteht, wenn wir aus einigen Reststücken die neue Filmrolle mit Länge $\geq F$ zusammenkleben. (Das heißt, es gibt eine Auswahl an Reststücken-Indizes I, sodass $\sum_{i \in I} \ell_i \geq F$. Der Überschuss $\sum_{i \in I} \ell_i - F$ soll minimiert werden.)

(a) Entscheidungsproblem (4 Punkte)

Formulieren Sie das zugehörige Entscheidungsproblem RKP zum Optimierungsproblem.

(b) Prinzipielles Vorgehen (8 Punkte)

Um zu zeigen, dass RKP *NP*-vollständig ist, zeigt man im Normalfall, dass es...

Schritt 1		Schritt 2
☐ ...in *P* liegt	und	☐ co-semi-entscheidbar ist.
☐ ...in *NP* liegt		☐ *NP*-schwer ist.
☐ ...in *Co-NP* liegt		☐ nicht *NP*-schwer ist.
☐ ...semi-entscheidbar ist		☐ nicht *P*-vollständig ist.

Für Schritt 1 ist zu zeigen, wie man einen Zeugen ...

Für Schritt 2 benötigen wir ☐ eine Reduktion ☐ einen Widerspruch ☐ ein Problem

☐ auf ☐ von ☐ zu einem Problem, von dem wir bereits wissen, dass es

☐ in *NP* liegt. ☐ *NP*-schwer ist. ☐ unentscheidbar ist.

(c) Problem für Schritt 2 definieren (4 Punkte)

Finden Sie ein solches in der Vorlesung behandeltes Problem, geben Sie den Namen an und definieren Sie es.

(d) Schritt 2 beweisen (6 Punkte)

Führen Sie Schritt 2 mit diesem Problem durch. Begründen/beweisen Sie alle dafür notwendigen Eigenschaften.

Viel Erfolg!