Universität Osnabrück / FB6 / Theoretische Informatik Prof. Dr. M. Chimani

Informatik D: Einführung in die Theoretische Informatik

Klausur — SoSe 2014 — 30. Juli 2014

Haupttermin, Prüfungsnr. 1007049 Gruppe: Reis (Risotto, Nasi-Goreng)

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	10	Σ
Punkte (max)	12	12	16	12	10	12	12	12	20	12	**130**
Punkte (erreicht)											

Punkte	0..64	65..72	73..79	80..84	85..89	90..95	96..100	101..105	106..110	111..117	118..130
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: Sprache vs. Grammatik **(12 Punkte)**

(a) Hierarchie und Automaten (10 Punkte)

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

Chomsky-Typ	Name der Sprachfamilie	Automaten
0		
		NLBA
	regulär	

(b) Grammatikdefinition (2 Punkte)

Definieren Sie kontextfreie Grammatiken (für Sprachen L mit $\varepsilon \notin L$).

Alle Regeln haben die Form...

V → (V ∪ ∑)+

Aufgabe 2: Sprachen **(12 Punkte)**

Welche Aussagen stimmen?
(Achtung: Pro Frage gibt es +2/0/−2 Punkte bei einer richtigen/keinen/falschen Antwort! Sie erhalten jedoch natürlich mindestens 0 Punkte für die gesamte Aufgabe.)

korrekt	falsch	
■	☐	Die Sprache $\{0^k 1^\ell 2^k 22^\ell \mid k, \ell \geq 0\}$ ist kontextfrei.
■	☐	Nicht-deterministische Kellerautomaten, die maximal ein Symbol im Keller speichern können, sind nur so mächtig wie deterministische endliche Automaten.
☐	■	Man kann jeden deterministischen Kellerautomat in einen nicht-deterministischen Kellerautomaten umformen, dessen Keller immer maximal 2 Elemente enthält.
☐	■	Es gibt endliche Mengen die nicht das Alphabet einer Sprache sein können.
■	☐	Das Alphabet einer Sprache kann unendlich groß sein.
☐	■	Σ^* ist die Potenzmenge von Σ.

Aufgabe 3: Pumping Lemma **(16 Punkte)**

(a) Definition (4 Punkte)

Wie lautet das Pumping Lemma für reguläre Sprachen?

(b) Anwendung (12 Punkte)

Beweisen Sie, dass $\{c^j a^i c a^{3i} c \mid i, j \geq 0\}$ keine reguläre Sprache ist.

Aufgabe 4: RegEx vs. DEA **(12 Punkte)**

Geben Sie einen deterministischen endlichen Automaten an, der dem folgenden regulären Ausdruck entspricht:

$$(\varnothing^*|23^*2|2^*)1$$

(Es gibt genug Platz, damit Sie Zwischenschritte aufschreiben können. Markieren Sie Ihr Endergebnis bitte entsprechend.)

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

Geben Sie für die Sprache $\{a^j c b^{j-i} c d^i \mid j \geq i \geq 0\}$ einen Kellerautomaten an, der durch leeren Keller akzeptiert.

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

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

$$f(\alpha) := \begin{cases} \text{undef} & \alpha \in \mathbb{N}_g \cup \{0\} \\ \alpha - 1 & \text{sonst} \end{cases}$$

Aufgabe 7: LOOP-Berechenbarkeit **(12 Punkte)**

(a) LOOP-Programm (8 Punkte)

Geben Sie ein LOOP-Programm an (eingeschränkte Definition, d.h. keine Addition von Variablen oder höhere Rechenoperationen), dass der folgenden Codezeile entspricht:

$$x_4 := 3 \cdot x_1 \cdot x_2$$

(b) Turing-vollständig (4 Punkte)

Begründen Sie, warum LOOP-Programme nicht so mächtig wie Turingmaschinen sind.

Aufgabe 8: Entscheidbarkeit **(12 Punkte)**

Wir definieren das Ergebnis des *Klebeoperators* $\circ$ als die Zahl, die durch das Hintereinanderschreiben der Dezimaldarstellungen ihrer einzelnen Argumente repräsentiert wird. Wir können mehrere Klebeoperationen gesammelt schreiben, z.B.

$$\bigcirc_{i=1}^{4} i^2 \;=\; 1 \circ 4 \circ 9 \circ 16 \;=\; 14916.$$

Betrachten Sie das folgende Problem:

Gegeben: Drei jeweils m-elementige Mengen $\mathcal{A} := \{a_i\}_{1 \leq i \leq k}$, $\mathcal{B} := \{b_i\}_{1 \leq i \leq k}$, $\mathcal{C} := \{c_i\}_{1 \leq i \leq k}$ mit Elementen aus $\mathbb{N}$.
Frage: Gibt es eine Sequenz $s(1), s(2), \ldots, s(n)$ mit $n \geq 1$ und $s(i) \in \{1, \ldots, k\}$ für alle $1 \leq i \leq n$, so dass

$$\bigcirc_{i=1}^{n} a_{s(i)} - \bigcirc_{i=1}^{n} b_{s(i)} = \bigcirc_{i=1}^{n} c_{s(i)}. \qquad \text{(„Gleichung“)}$$

(a) Beispiele (4 Punkte)

Geben Sie jeweils ein Beispiel einer Ja- und einer Nein-Instanz für dieses Problem an:

Ja-Instanz

Nein-Instanz

(b) Semi-Entscheidbarkeit (4 Punkte)

Zeigen Sie, dass das Problem semi-entscheidbar ist:

(c) Unentscheidbarkeit (4 Punkte)

Beschreiben Sie kurz die notwendige Reduktion (von? nach? wie?) um zu begründen, warum das Problem nicht entscheidbar ist:

Aufgabe 9: *P* vs. *NP* (20 Punkte)

(a) Definition (4 Punkte)

Definieren Sie die Komplexitätsklasse *P*.

(b) Basiszusammenhänge (10 Punkte)

Welche Aussagen stimmen?

(Achtung: Pro Frage gibt es +2/0/−2 Punkte bei einer richtigen/keinen/falschen Antwort! Sie erhalten jedoch natürlich mindestens 0 Punkte für die gesamte Aufgabe.)

korrekt	falsch	
☐	■	Das Problem „Finde das kleinste Element aus n gegebenen Zahlen“ liegt in NP.
■	☐	Das Problem „Gegeben ein gewichteter Graph und eine Zahl W. Kann man einen aufspannenden Baum finden, dessen Gesamtgewicht maximal W ist?“ liegt in P.
☐	■	Das Problem „Gegeben ein Graph auf $n \geq 8$ Knoten. Kann man einen Hamiltonkreis finden, der maximal $n-3$ Kanten enthält?“ ist NP-vollständig.
■	☐	Wenn ein stark NP-vollständiges Problem einen pseudopolynomiellen Algorithmus erlaubt, gilt $\mathsf{P} = \mathsf{NP}$.
■	☐	Sei $\mathcal{A}$ das zu einem Optimierungsproblem $\mathcal{B}$ zugehörige Entscheidungsproblem. Wenn $\mathcal{A} \in \mathsf{P}$, gibt es einen deterministischen polynomiellen Algorithmus für $\mathcal{B}$.

(c) Zeuge (6 Punkte)

Was versteht man, wenn man über P und NP spricht, unter einem *Zeugen*?

(Vollständige!) Definition:

Warum kann ein Zeuge nur polynomiell groß sein?

Eine det. TM kann in polynomieller Zeit nur polynomiell viele "Elemente" eines Zeugen betrachten. Ist dieser superpolynomiell groß, kann er nicht in polynomieller Zeit komplett betrachtet und daher auch nicht verifiziert werden.

Aufgabe 10: ***NP*-vollständig** (12 Punkte)

Sie kennen das Problem SAT, in dem eine Formel in konjunktiver Normalform gegeben ist, und jede Klausel *mindestens* ein Literal enthält. Sie kennen auch den Spezialfall des 3-SAT, in dem jede Klausel *maximal* drei Literale enthält. Wir definieren nun das folgende Problem:

> **Problem:** EXAKT-5-SAT
>
> **Gegeben:** Eine aussagenlogische Formel F in konjunktiver Normalform mit *genau* fünf Literalen pro Klausel.
>
> **Frage:** Ist F erfüllbar?

Um zu zeigen, dass EXAKT-5-SAT ***NP***-vollständig ist, zeigt man im Normalfall, dass...

☐ ...es vollständig in ***P*** liegt	und	☐ in ***Co-NP*** liegt.
☐ ...es in ***P*** liegt		☐ nicht ***P***-vollständig ist.
☒ ...es in ***NP*** liegt		☒ ***NP***-schwer ist.
☐ ...es in ***NPI*** liegt		☐ nicht ***NP***-schwer ist.
Punkt 1		*Punkt 2*

Punkt 1 ist trivial, daher beschränken wir uns auf Punkt 2. Dazu benötigen wir eine Reduktion ☐ über das ☒ von dem ☐ zu dem Problem ☒ 3-SAT ☐ SAT ☐ PARTITION .

Geben Sie die notwendige Reduktion an, begründen Sie ihre notwendigen Eigenschaften und beweisen Sie, dass der korrekte *Punkt 2* gilt.

Viel Erfolg!