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

Informatik D: Einführung in die Theoretische Informatik

Klausur — SoSe 2019 — 10. Juli 2019

Haupttermin, Prüfungsnr. 1007049 — Gruppe: Menorca / Formentera

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!
- 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	Σ
Punkte (max)	12	8	12	10	12	20	14	16	**104**
Punkte (erreicht)									

Punkte	0..51	52..56	57..60	61..65	66..69	70..74	75..78	79..83	84..87	88..92	93..104
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: Sprachgrundlagen **(12 Punkte)**

(a) Faktenwissen (4 Punkte)

Von welchem Maschinenmodell werden genau die deterministisch kontextfreien Sprachen beschrieben?

Welche Einschränkungen haben die Regeln einer Grammatik, die eine reguläre Sprache beschreibt?

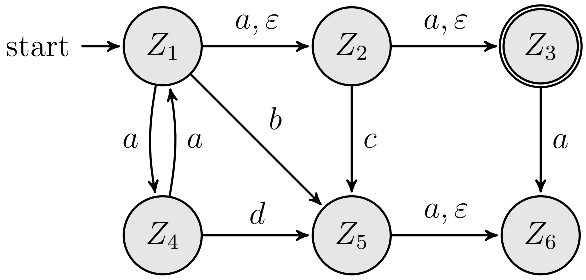
(b) Zuordnung von Sprachen (8 Punkte)

Zu welcher der *fünf* in der Vorlesung besprochenen Sprachfamilien innerhalb der Chomsky-Hierarchie gehören die folgenden Sprachen? Geben Sie dabei die *kleinste* Sprachfamilie an, die gerade mächtig genug ist, die entsprechende Sprache zu erkennen.

$\mathcal{L}(M)$, wobei M eine TM mit nur 4 Zuständen ist	
$\{\alpha^n\beta^{n+1} \mid n \geq 2\} \cup \{\alpha^{n+1}\beta^n \mid n \geq 2\}$	
$\left\{a0a^R0a \mid a \in \{1\}^*\right\} \cap \left\{1b1b1 \mid b \in \{0,1\}^*\right\}$	
$\left\{z \;\middle|\; \begin{array}{r} z \text{ ist Dezimaldarstellung einer} \\ \text{durch 3 teilbaren Zahl} \end{array}\right\} \varnothing^* \{nlba\}$	

Aufgabe 2: Umwandlung NDEA → DEA (8 Punkte)

Wandeln Sie den folgenden endlichen Automaten – gemäß dem Vorgehen aus der Vorlesung – in einen deterministischen endlichen Automaten um.



Aufgabe 3: Pumping Lemma **(12 Punkte)**

(a) Definition (4 Punkte)

Wie lautet das Pumping Lemma für kontextfreie Sprachen?

Sei L eine kontextfreie Sprache. Dann...

... $z = uvwxy$ mit den Eigenschaften

(1) ______, (2) ______ und (3) ______.

(b) Anwendung (8 Punkte)

Beweisen Sie, dass $L := \{\alpha^k \beta^\ell \gamma^k \delta^m \mid k, m \geq 0, \ell = k \cdot m\}$ nicht kontextfrei ist.

Aufgabe 4: Rechnende Turingmaschine **(10 Punkte)**

Für eine Zahl $n \in \mathbb{N}$ sei $\mathbb{U}_a(n)$ ihre Unärkodierung mittels Symbol a und $\mathbb{B}(n)$ ihre Binärkodierung. Seien $n, k \in \mathbb{N}_+$ gegeben durch den Eingabestring $\mathbb{U}_a(k)\square\mathbb{B}(n)$. Konstruieren Sie eine Turingmaschine, die $\mathbb{B}(f(n,k))$ für die folgende Funktion berechnet:

$$f(n,k) := \begin{cases} n \cdot (k \bmod 3) & \text{wenn } k \text{ gerade ist,} \\ \mathsf{undef} & \text{sonst.} \end{cases}$$

Aufgabe 5: Berechenbarkeit **(12 Punkte)**

(a) Definition (4 Punkte)

Vervollständigen Sie die folgende Definition ohne den Begriff *berechnen* zu verwenden.

Eine Funktion f mit Definitionsbereich $\mathsf{Def} \subseteq \mathbb{N}^k$ *ist berechenbar, wenn …*

(b) Beweis (8 Punkte)

Gegeben eine Funktion g_c, die abhängig ist von einer Konstanten $c \geq 11$. Bei Eingabe einer Turingmaschine T mit Eingabealphabet Σ liefert $g_c(T)$ die Anzahl der TM-Eingaben $x \in \Sigma^c$, auf denen T hält. — Ist g_c berechenbar? Begründen Sie.

Aufgabe 6: ***NP*-Vollständigkeit** **(20 Punkte)**

(a) Definition (8 Punkte)

Definieren Sie ***NP***-Vollständigkeit, ohne den Begriff „***NP***-schwer“ zu benutzen. Definieren Sie auch den Begriff der Reduktion. Sie können davon ausgehen, dass ***NP*** definiert ist.

(b) Reduktion (12 Punkte)

Sei $\mathbb{P} := \{p_i \mid i \in \mathbb{N}\}$ die Menge aller Primzahlen. Eine Zahl $\kappa \in \mathbb{N}$ heißt *quadratfrei*, wenn in ihrer Primfaktorzerlegung keine Primzahl mehrmals auftritt, z.B. ist $14 = 2 \cdot 7$ quadratfrei, aber $12 = 2 \cdot 2 \cdot 3$ nicht.

Auf einem Mathematikkongress soll die beste quadratfreie Zahl $\kappa \in \mathbb{N}$ so gewählt werden, dass sie allen Teilnehmenden gefällt. Vorher darf jede*r Teilnehmende beliebig viele Forderungen äußern: Eine Forderung ist dabei ein Tripel, bestehend aus einem Teilbarkeitszeichen aus $\{\oslash, \otimes\}$, einer Primzahl $p \in \mathbb{P}$ und einem*r Teilnehmenden t. Eine Forderung $(\oslash, p, t)$ ist genau dann erfüllt, wenn p ein Primfaktor von κ ist; eine Forderung $(\otimes, p, t)$ ist genau dann erfüllt, wenn p kein Primfaktor von κ ist.

> KONGRESSZAHL
>
> **Gegeben:** Eine Menge von Teilnehmenden T, eine Menge von Forderungen F
>
> **Gefragt:** Gibt es eine quadratfreie Zahl $\kappa \in \mathbb{N}$, sodass für jede*n Teilnehmende*n mindestens eine Forderung erfüllt wird?

Zeigen Sie, dass KONGRESSZAHL ***NP***-schwer ist.

Definieren Sie das bei Ihrer Reduktion verwendete Problem:

> ***Name:***
>
> ***Gegeben:***
>
> ***Gefragt:***

Aufgabe 7: Fixed Parameter Tractability **(14 Punkte)**

(a) Definition (4 Punkte)

Wie lautet die Definition von ***FPT***?

Ein Entscheidungsproblem liegt in ***FPT*** in Bezug auf Parameter ℓ, wenn es sich in Zeit

$\Big\{$ ☐ $\mathcal{O}\big(g(n) \cdot 2^c\big)$ ☐ $\mathcal{O}\big(g(n) \cdot 2^\ell\big)$ ☐ $\mathcal{O}\big(g(\ell) \cdot n^c\big)$ ☐ $\mathcal{O}\big(g(c) \cdot n^\ell\big)$ $\Big\}$ lösen lässt.

Dabei ist g $\Big\{$ ☐ beliebig, ☐ berechenbar, ☐ exponentiell, ☐ polynomiell $\Big\}$ und $c \in \mathbb{N}$.

(b) Kernelization (10 Punkte)

Im Teutoburger Wald brennen n Bäume, deren genaue Position bekannt ist. Ein Löschflugzeug kann beliebig viele brennende Bäume löschen, sofern sie exakt entlang einer Geraden stehen.

BÄUMELÖSCHEN

Gegeben: $n \in \mathbb{N}$ Bäume mit Koordinaten in $\mathbb{N}^2$, $\ell \in \mathbb{N}$.

Gefragt: Gibt es eine Menge von $\leq \ell$ Geraden, sodass jeder der n Bäume auf mindestens einer der Geraden liegt?

Geben Sie einen Algorithmus an, der in Polynomialzeit entweder das Problem entscheidet oder eine äquivalente Instanz erzeugt, deren Größe ausschließlich von ℓ abhängt.

Hinweis: Stellen Sie sich eine Gerade g durch ℓ Punkte vor. Was muss für eine Lösung gelten, falls g nicht ausgewählt wird?

Fortsetzung Aufgabe 7:

Aufgabe 8: Randomisierte Algorithmen (16 Punkte)

(a) Definition (4 Punkte)

Wie lautet die Definition von ***BPP*** ?

(b) Algorithmus (12 Punkte)

Gegeben eine Zahl $n \in \mathbb{N}_+$ und eine Blackbox B, die bei Eingabe eines Bitstrings aus $\{0,1\}^n$ in konstanter Zeit ein einzelnes Bit ausgibt. Es gibt zwei Arten von Blackboxen: Eine *konstante Box* gibt immer dasselbe Bit aus, wohingegen eine *balancierte Box* für genau die Hälfte der Eingaben 0 und für die andere Hälfte 1 ausgibt.

Geben Sie einen randomisierten Algorithmus an, der bei Eingabe n mit konstant vielen Blackboxanfragen entscheidet, ob B eine balancierte Box ist. Auf *NEIN*-Instanzen soll Ihr Algorithmus immer korrekt antworten und auf *JA*-Instanzen eine Fehlerwahrscheinlichkeit $< 1/6$ aufweisen. Beweisen Sie dies!

Geben Sie zuerst die kleinstmögliche randomisierte Komplexitätsklasse an, der dieser Algorithmus zugeordnet werden kann:

Algorithmus ***und*** *Analyse:*

Fortsetzung Aufgabe 8:

Notizen: