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

Einführung in die Theoretische Informatik

Klausur — SoSe 2023 — 25. September 2023

Nebentermin, Prüfungsnr. 1007049 — Gruppe: dQw4w9WgXcQ

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	Σ
Punkte (max)	10	8	10	8	18	8	12	**74**
Punkte (erreicht)								

Punkte	0.. 36	37..38	39..40	41..42	43..45	46..48	49..51	52..55	56..58	59..62	63..74
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: Information **(10 Punkte)**

(a) Quellencodierungstheorem (4 Punkte)

Wie lautet das Quellencodierungstheorem?

Sei (Σ, p) *eine Quelle. Dann gilt:*

(i)

(ii)

(b) Erwartete Codewortlänge (6 Punkte)

Die Quelle (Σ, p), die durch die nebenstehende Tabelle gegeben ist, hat Entropie $H_{\Sigma,p} \approx 2{,}22$.

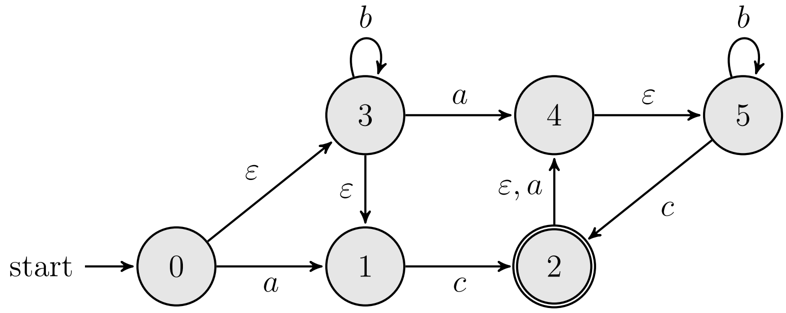
σ	a	b	c	d	e
p_σ	$\frac{1}{12}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{4}$

Geben Sie einen binären Präfixcode für (Σ, p) mit minimaler erwarteter Codewortlänge an.

Bestimmen Sie außerdem die erwartete Codewortlänge Ihrer Lösung als Dezimalzahl.

Aufgabe 2: Umwandlung NDEA → DEA (8 Punkte)

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



Aufgabe 3: Rechnende Turingmaschine (10 Punkte)

Sei $e(n)$ die Anzahl der Einsen in der Binärcodierung $\mathbb{B}(n)$ einer Zahl $n \in \mathbb{N}$. Erstellen Sie eine rechnende Turingmaschine mit Bandalphabet $\Gamma := \{\square, 0, 1, \$\}$ und **höchstens 10 Zuständen**, die bei Eingabe $\mathbb{B}(n)$ die folgende Funktion berechnet und das Ergebnis binär codiert ausgibt:

$$f(n) := \begin{cases} n/2^{e(n)}, & \text{falls } n \bmod 2^{e(n)} = 0, \\ \texttt{undef}, & \text{sonst.} \end{cases}$$

Hinweis: Durch $2^{e(n)}$ zu dividieren ist dasselbe wie $e(n)$-mal durch 2 zu dividieren. Denken Sie darüber nach, was bei einer Division durch 2 passieren soll.

Aufgabe 4: Co-Semi-Entscheidbarkeit **(8 Punkte)**

Sei $\mathbb{W}(M) \in \{0,1\}^*$ eine eindeutige Codierung einer rechnenden TM M. Sei

$$L \coloneqq \{\mathbb{W}(M) \mid M \text{ berechnet bei Eingabe einer Binärzahl } x \text{ \underline{nie} } x^2\}.$$

Zeigen Sie, dass L co-semi-entscheidbar ist.

Aufgabe 5: *NP*-Vollständigkeit **(18 Punkte)**

(a) Definitionen und Eigenschaften (6 Punkte)

Aufgabe: Vervollständigen Sie.

Eine Reduktion von $\mathcal{X}$ auf $\mathcal{Y}$ ist eine ______ Funktion f, die eine beliebige ____-Instanz I in eine ____-Instanz $f(I)$ verwandelt, sodass gilt: I ist eine ____-Instanz ____ $f(I)$ ist eine ____-Instanz.

Ein Problem $\mathcal{A}$ ist *NP*-schwer, wenn gilt: für ______ $\mathcal{B} \in$ ____ existiert eine ______ Reduktion von ____ auf ____.

Ein Problem $\mathcal{Q}$ ist *NP*-vollständig, wenn gilt: ______ und ______.

(b) Reduktion (12 Punkte)

Luca möchte in allen Museen ihrer Stadt Protestaktionen durchführen. Um von Museum zu Museum zu gelangen, nutzt sie ihren E-Roller. Da dieser nur einen sehr kleinen Akku hat, muss Luca nach jedem Museumsbesuch eine Ladestation anfahren. Am Ende des Tages möchte sie wieder an den Startpunkt zurückkehren.

Der Stadtplan ist gegeben als ungerichteter Graph $G = (V, E)$. Dabei besteht $V = \mathcal{Z} \cup \mathcal{L}$ aus zwei Typen von Knoten: *Zielknoten* $\mathcal{Z}$ (Museen und Startpunkt) und *Ladeknoten* $\mathcal{L}$.

MUSEUMSTOUR

Gegeben: Ein ungerichteter Graph $G = (\mathcal{Z} \cup \mathcal{L}, E)$ mit $\mathcal{Z} \cap \mathcal{L} = \varnothing$.

Gefragt: Gibt es eine Rundtour in G, sodass *(i)* jeder Knoten aus $\mathcal{Z}$ genau einmal besucht wird und *(ii)* keine zwei aufeinanderfolgenden Knoten denselben Typ haben?

Zeigen Sie, dass MUSEUMSTOUR *NP*-schwer ist. Das in Ihrer Reduktion verwendete Ausgangsproblem $\mathcal{X}$ muss aus der Vorlesung stammen.

Definieren Sie das bei Ihrer Reduktion verwendete Problem $\mathcal{X}$:

Name:

Gegeben:

Gefragt:

Reduktionsbeweis:

Aufgabe 6: Dynamische Programmierung (8 Punkte)

Arnold S. hat sein Fitnessstudio gewechselt. Um sich direkt Respekt zu verschaffen, möchte er alle verfügbaren Gewichtsscheiben auf einer Hantelstange stemmen. Dazu muss er die Gewichte auf die rechte und linke Seite aufteilen, sodass diese genau gleich schwer sind. Jede Scheibe i ist mit ihrem Gewicht $w_i > 0$ beschriftet.

HANTELSTANGE

Gegeben: Menge $S = \{1, ..., n\}$ von beschrifteten Gewichten.

Gefragt: Gibt es eine Aufteilung von S in zwei disjunkte Mengen $L, R \subseteq S$, sodass $L \cup R = S$ und $\sum_{i \in L} w_i = \sum_{i \in R} w_i$?

Sei $W \coloneqq \sum_{i \in S} w_i$ das Gesamtgewicht. Geben Sie einen deterministischen Algorithmus an, der HANTELSTANGE mittels dynamischer Programmierung in Zeit $\mathcal{O}(n \cdot W)$ exakt löst.

Berechnen Sie dafür eine Matrix M mit Einträgen $M[x, y] \in \{\texttt{true}, \texttt{false}\}$ für $0 \leq x \leq n$ und $0 \leq y \leq W$. Dabei ist $M[x, y]$ genau dann $\texttt{true}$, wenn eine Menge $X \subseteq \{i \in S \mid i \leq x\}$ mit $\sum_{i \in X} w_i = y$ existiert.

// Initialisierung:

// Dynamische Programmierungs-Schleife:

// Ergebnis:

Aufgabe 7: Randomisierte Algorithmen **(12 Punkte)**

(a) Definitionen (6 Punkte)

Vervollständigen Sie die folgenden Definitionen.

Co-RP enthält alle Entscheidungsprobleme, für die ein Monte-Carlo-Algorithmus mit { ☐ im Worst Case / ☐ im Erwartungswert } polynomieller Laufzeit existiert, der

Ja-Instanzen mit Wahrscheinlichkeit { ☐ $= 1$ ☐ $= 0$ / ☐ $\geq \frac{1}{2}$ ☐ $\leq \frac{1}{2}$ } und

Nein-Instanzen mit Wahrscheinlichkeit { ☐ $= 1$ ☐ $= 0$ / ☐ $\geq \frac{1}{2}$ ☐ $\leq \frac{1}{2}$ } korrekt beantwortet.

Falls er nicht korrekt antwortet, antwortet er { ☐ mit dem Gegenteil / ☐ „keine Ahnung" / ☐ gar nicht }.

ZPP enthält alle Entscheidungsprobleme, für die ein { ☐ Monte-Carlo / ☐ Las-Vegas }-Algorithmus mit im Worst Case polynomieller Laufzeit existiert, der

Ja-Instanzen mit Wahrscheinlichkeit { ☐ $= 1$ ☐ $= 0$ / ☐ $\geq \frac{1}{2}$ ☐ $\leq \frac{1}{2}$ } und

Nein-Instanzen mit Wahrscheinlichkeit { ☐ $= 1$ ☐ $= 0$ / ☐ $\geq \frac{1}{2}$ ☐ $\leq \frac{1}{2}$ } korrekt beantwortet.

Falls er nicht korrekt antwortet, antwortet er { ☐ mit dem Gegenteil / ☐ „keine Ahnung" / ☐ gar nicht }.

(b) *Co-RP* vs. *ZPP* (6 Punkte)

Sei $\mathcal{X} \in$ *ZPP*. Zeigen Sie, dass $\mathcal{X} \in$ *Co-RP*.

Notizen:

Viel Erfolg!