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

Einführung in die Theoretische Informatik

Klausur — SoSe 2022 — 11. Juli 2022

Haupttermin, Prüfungsnr. 1007049 — Gruppe: Charles Leclique / Jacky Clickx

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)	8	8	12	10	10	20	18	12	**98**
Punkte (erreicht)									

Punkte	0.. 48	49..50	51..53	54..56	57..60	61..64	65..68	69..73	74..77	78..82	83..98
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 **(8 Punkte)**

(a) Definition (2 Punkte)

Sei (Σ, p) eine Informationsquelle; für $\sigma \in \Sigma$ sei p_σ die zugehörige Wahrscheinlichkeit.

Geben Sie die Definition von $\mathrm{H}_{\Sigma,p}$ an, ohne $\mathcal{I}(p)$ zu verwenden.

(b) Quellencodierungstheorem (6 Punkte)

Sei $n \geq 2$ konstant, $V \coloneqq \{v_1, ..., v_n\}$ eine Knotenmenge und

$$\mathcal{G} \coloneqq \{G = (V, E) \mid G \text{ ist ein gerichteter Graph ohne Schleifen}\}.$$

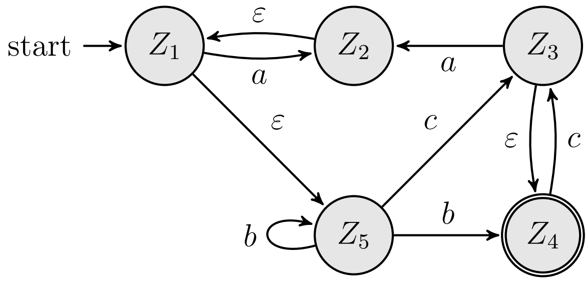
Wir betrachten eine Quelle, bei der jedes Element aus $\mathcal{G}$ gleich wahrscheinlich ist.

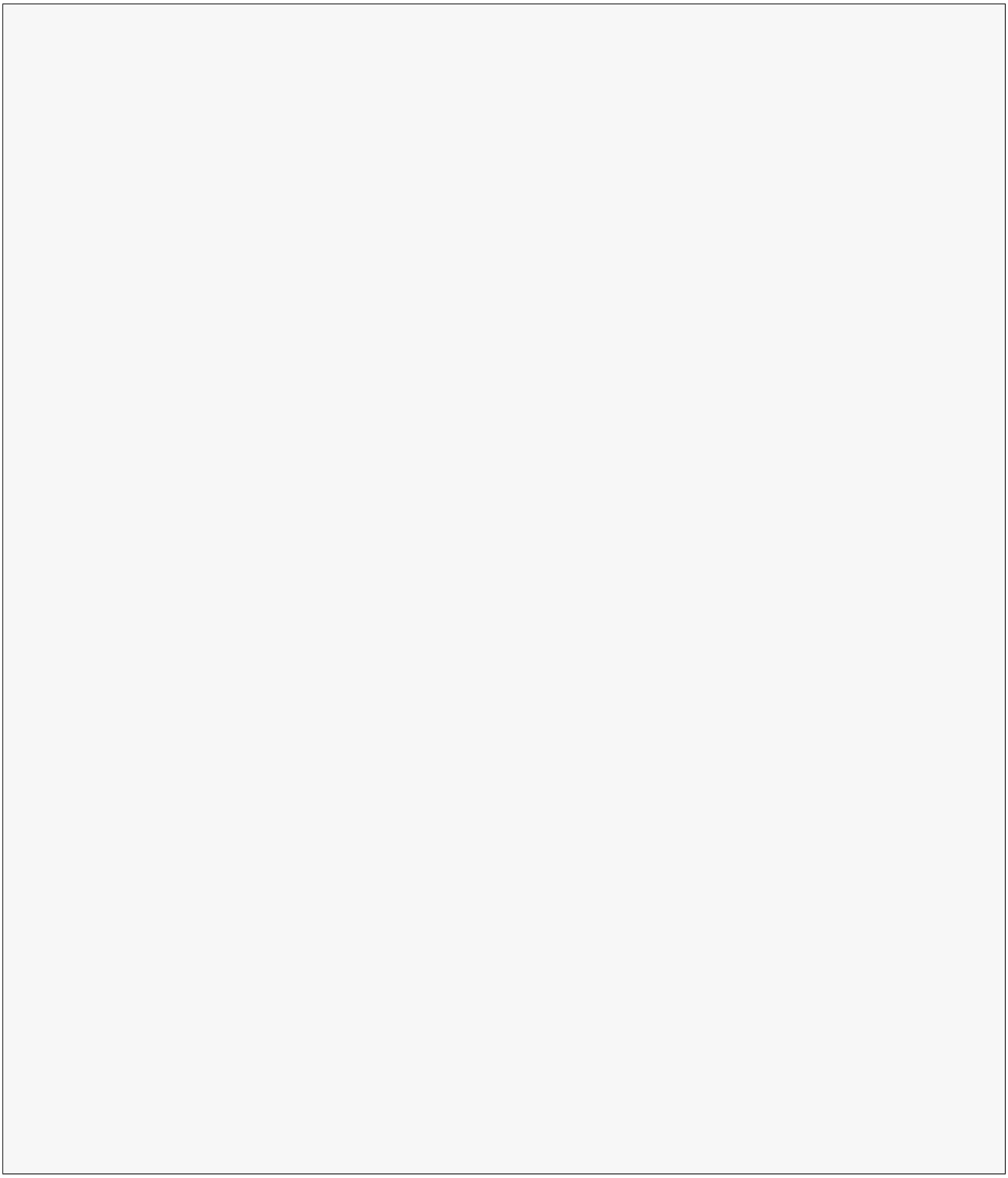
Geben Sie (mit Begründung) eine möglichst gute untere Schranke (in Abhängigkeit von n) für die minimale erwartete Codewort-Länge dieser Quelle an. Kürzen Sie den Ausdruck so weit wie möglich.

Hinweis: Denken Sie über die Kardinalität von $|\mathcal{G}|$ nach.

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 reguläre Sprachen?

Sei L eine reguläre Sprache. Dann...

... $z = uvw$ mit den Eigenschaften

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

(b) Anwendung (8 Punkte)

Beweisen Sie, dass $L := \{c^j d^\ell \mid j, \ell \geq 2,\ j \bmod \ell = 0\}$ nicht regulär ist.

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

Sei $n \geq 1$ und $x, y \in \{0,1\}^n$ zwei Binärstrings. Die Funktion $\texttt{OR}(x,y)$ berechnet bitweise einen Binärstring z, dessen i-te Stelle $z_i = x_i \vee y_i$ ist.

Geben Sie eine TM mit Bandalphabet $\Gamma = \{\square, 0, 1, \$, \#\}$ an, die bei Eingabe $x\$y\#$ den Binärstring $\texttt{OR}(x,y)$ ausgibt. Dabei darf die TM **niemals** ein $\square$ durch ein anderes Symbol überschreiben.

Anmerkung: Sie dürfen die Darstellung als *Springende TM* aus der Übung benutzen. Es existiert eine Springende TM mit 7 Zuständen und eine klassische TM mit 9 Zuständen.

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

(a) Definition (4 Punkte)

Vervollständigen Sie die folgende Definition ohne die Begriffe „berechnen“, „berechenbar“, „Hälfte“ oder „charakteristische Funktion“ zu verwenden.

Eine Sprache $L \subseteq \Sigma^*$ ist semi-entscheidbar, wenn …

(b) Beweis (6 Punkte)

NICHT-LEERE INDEXSCHLACHT

Gegeben: Wort $w \in \Sigma^*$; berechenbare Funktion ϕ mit $\phi(i) \coloneqq M_i$, wobei M_i für alle $i \in \mathbb{N}$ jeweils eine akzeptierende TM ist.

Gefragt: Ist $\{i \in \mathbb{N} \mid M_i \text{ akzeptiert } w\} \neq \varnothing$?

Beweisen Sie, dass NICHT-LEERE INDEXSCHLACHT semi-entscheidbar ist.

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

(a) Definition (8 Punkte)

Definieren Sie ***NP***-Vollständigkeit, ohne den Begriff „***NP***-schwer“ zu benutzen. Definieren Sie dabei auch den Begriff der Reduktion. Sie müssen die Klasse ***NP*** nicht definieren.

(b) Reduktion (12 Punkte)

Sei $G = (V, E)$ ein ungerichteter Graph mit Kantenkosten $c\colon E \to \mathbb{N}$. Eine Kantenmenge $A \subseteq E$ heißt *ausgewogen*, wenn $\sum_{e \in A} c(e) = \sum_{e \in E \setminus A} c(e)$.

Sei $W \subseteq V$ eine Knotenteilmenge. Eine Kante $\{v, w\} \in E$ bezeichnen wir als *Schnittkante*, wenn genau einer ihrer Endknoten in W liegt. Die Menge aller Schnittkanten ist also $\delta(W) \coloneqq \big\{\{v, w\} \in E \mid v \notin W, w \in W\big\}$.

AUSGEWOGENER SCHNITT

Gegeben: Zusammenhängender ungerichteter Graph $G = (V, E)$,
Kostenfunktion $c\colon E \to \mathbb{N}$.

Gefragt: Gibt es eine Knotenteilmenge $W \subseteq V$, sodass $\delta(W)$ ausgewogen ist?

Zeigen Sie, dass AUSGEWOGENER SCHNITT ***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:

Fortsetzung von Aufgabe 6:

Aufgabe 7: Algorithmische Methoden **(18 Punkte)**

(a) Fixed Parameter Tractability: Definition (4 Punkte)

Vervollständigen Sie die folgende Definition.

Ein Entscheidungsproblem $\mathcal{X}$ liegt in ***FPT*** in Bezug auf Parameter β, wenn ...

(b) Fixed Parameter Tractability: SubsetSum (6 Punkte)

Betrachten Sie das SubsetSum-Problem. Sei (A, b) eine Instanz und α die Anzahl an Bits der größten Zahl in A; der Wert von b darf beliebig groß sein.

Ist SubsetSum in FPT in Bezug auf Parameter α? Begründen Sie.

(c) Dynamische Programmierung (8 Punkte)

Sie möchten Himbeersträucher entlang einer Seite Ihres Gartenwegs pflanzen. Dazu haben Sie den zur Verfügung stehenden Platz in eine Reihe von Abschnitten $A_1, ..., A_N$ unterteilt. In jedem Abschnitt kann maximal ein Himbeerstrauch gepflanzt werden, jedoch dürfen keine zwei benachbarten Abschnitte gleichzeitig benutzt werden. Für jeden Abschnitt A_i haben Sie eine Schätzung h_i für den Ertrag an dort zu erntenden Himbeeren, falls in A_i ein Strauch gepflanzt wird. Sie möchten wissen, in welchen Abschnitten Sie Sträucher pflanzen sollten, um einen möglichst großen Gesamtertrag zu erreichen.

Fortsetzung von Aufgabe 7 (c):

HIMBEERSTRÄUCHER
Gegeben: Folge $(h_1, ..., h_N)$ positiver natürlicher Zahlen.
Gesucht: $\max_{X \in \mathcal{X}} \sum_{i \in X} h_i$, wobei $\mathcal{X} \coloneqq \big\{ Y \subseteq \{1, ..., N\} \mid \nexists\, i \in Y \text{ mit } (i+1) \in Y \big\}$.

Sei $h \coloneqq \sum_{i=1}^{N} h_i$. Geben Sie einen deterministischen Algorithmus an, der mittels dynamischer Programmierung das Problem HIMBEERSTRÄUCHER in Zeit $\mathcal{O}(N \cdot h)$ exakt löst. ~~Benutzen~~Berechnen Sie dafür eine Matrix M mit Einträgen $M[a, b] \in \{\texttt{true}, \texttt{false}\}$ für $0 \leq a \leq h$ und $0 \leq b \leq N$. Dabei ist $M[a, b]$ genau dann `true`, wenn es eine gültige Lösungsmenge $X \subseteq \{1, ..., b\}$ mit $\sum_{i \in X} h_i = a$ gibt.

// Initialisierung:

// Dynamische Programmierungs-Schleife:

// Ergebnis:

Aufgabe 8: Randomisierte Algorithmen **(12 Punkte)**

(a) Definition (4 Punkte)

Definieren Sie ***Co-RP***.

(b) Algorithmus (8 Punkte)

Bei einem *fairen* Würfel ist jede Seite gleich wahrscheinlich; bei einem *gezinkten* Würfel ergibt jeder Wurf eine 1. Wir betrachten einen Würfel W mit 5 Seiten, die mit $1, 2, ..., 5$ gelabelt sind. Wir wissen, dass W entweder fair oder gezinkt ist.

Geben Sie einen ***Co-RP***-Algorithmus an, der in möglichst kurzer Zeit mit $99,9\%$-iger Sicherheit entscheidet, ob W fair ist oder nicht. Zeigen Sie, dass die nötigen Eigenschaften erfüllt werden.

Hinweis: Sie dürfen annehmen, dass ein einzelner Würfelwurf konstante Zeit benötigt.

Algorithmus **und** Analyse:

Fortsetzung Aufgabe 8:

Notizen: