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

Einführung in die Theoretische Informatik

Klausur — SoSe 2022 — 19. September 2022

Nebentermin, Prüfungsnr. 1007049 — Gruppe: Accelleratii incredibus/Velocitus incalcublii

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	9	Σ
Punkte (max)	8	8	12	10	16	14	12	6	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) Codeeigenschaften (2 Punkte)

Betrachten Sie die nebenstehende Tabelle für die Quelle (Σ, p). Ist der gegebene Code $\mathbb{C}$ ein binärer Blockcode minimaler erwarteter Codewortlänge? Begründen Sie.

σ	a	b	c
p_σ	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$
$\mathbb{C}(\sigma)$	1	01	00

(b) Entropie (6 Punkte)

Für $n \geq 1$ sei $\Sigma = \{a_0, a_1, ..., a_n\}$ ein Alphabet und $q(a_i)$ die Wahrscheinlichkeit von a_i. Sei $q(a_0) = 2^{-n}$ und $q(a_j) = 2^{-j}$ für alle $1 \leq j \leq n$. Zeigen Sie, dass die Entropie von (Σ, q) gegeben ist durch:

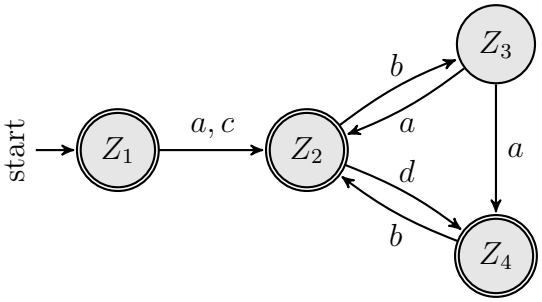
$$\mathrm{H}_{\Sigma,q} = 2 - 2^{1-n}$$

Hinweis: Sie dürfen die folgende Hilfsaussage ohne Beweis benutzen:

$$\sum_{i=1}^{n} i \cdot 2^{-i} = 2 - (n+2) \cdot 2^{-n}$$

Aufgabe 2: Umwandlung DEA → RegEx **(8 Punkte)**

Gegeben sei der rechts abgebildete DEA. Wandeln Sie ihn – gemäß dem Vorgehen aus der Vorlesung! – in einen regulären Ausdruck 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 $L := \{a^i d^j a^k \mid i, j, k \in \mathbb{N},\ j \leq 2,\ i \leq (2-j)k\}$ nicht regulär ist.

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

Betrachten Sie die folgende **reguläre** Sprache $L \coloneqq \{a^i b^j a^k \mid i, k \geq 0,\ j = (i + k) \bmod 2\}$.

Geben Sie eine TM M mit Bandalphabet $\Gamma \coloneqq \{\square, a, b\}$ an, die **niemals** ein $\square$ durch ein anderes Symbol überschreibt. Bei Eingabe $w \in \{a, b\}^+$ soll M die folgende Funktion f berechnen und das Ergebnis in Unärcodierung (mit Symbol a) ausgeben:

$$f(w) \coloneqq \begin{cases} |w|, & \text{falls } w \in L, \\ \texttt{undef}, & \text{sonst.} \end{cases}$$

Ihre TM darf **höchstens 8 Zustände** haben. Die Darstellung als Springende TM ist nicht erlaubt. Es existiert eine natürliche Lösung mit 6 Zuständen (und eine schlaue mit 5).

Aufgabe 5: Berechenbarkeit und Entscheidbarkeit (16 Punkte)

(a) Turing-Vollständigkeit (10 Punkte)

Betrachten Sie die folgende Programmiersprache WILE E. COYOTE:

Variablen, Konstanten, Zuweisung, Addition, Subtraktion und Verkettung sind identisch zur WHILE-Sprache. Für die folgenden Befehle seien A und B (möglicherweise leere) Anweisungsblöcke, $\mathtt{x}$ eine Variable und M eine Menge von Variablen.

$\mathtt{Wile}\{A\}$	führt A aus, solange die Summe aller in A auftretenden Variablen gerade ist (die Überprüfung erfolgt stets vor Schleifeneintritt);
$\mathtt{E.x}\{A\}\{B\}$	falls $\mathtt{x} \neq 0$ ist, führe A aus, sonst B;
$\mathtt{Coyote}(M)\{A\}\{B\}$	falls die Summe aller Variablen in M gerade ist, führe A aus, sonst B;
$\mathtt{BeepBeep}$	gibt die Imitation einer alten Autohupe aus.

Zeigen oder widerlegen Sie, dass WILE E. COYOTE Turing-vollständig ist.

Beispiel: Der folgende WILE E. COYOTE-Code berechnet $\mathtt{x}_1 \coloneqq 2 \cdot \mathtt{x}_1 + 1$ und hupt währenddessen $\lceil \mathtt{x}_1/2 \rceil$ mal:

$$\mathtt{x}_2 \coloneqq \mathtt{x}_1;\ \mathtt{Wile}\big\{\mathtt{x}_1 \coloneqq \mathtt{x}_1 + 1;\ \mathtt{E.x}_2\{\mathtt{x}_2 \coloneqq \mathtt{x}_2 - 1\}\{\};\ \mathtt{Coyote}(\{\mathtt{x}_1\})\{\mathtt{BeepBeep}\}\{\}\big\}$$

Fortsetzung von Aufgabe 5:

(b) Entscheidbarkeit (6 Punkte)

Eine deterministische akzeptierende TM M ist *voll*, wenn es eine Eingabe $w \in \Sigma^*$ gibt, sodass der Berechnungsweg von $M(w)$ jeden Zustand besucht.

Zeigen Sie, dass das folgende Problem semi-entscheidbar ist:

VolleTM	
Gegeben:	Deterministische akzeptierende TM M mit Eingabealphabet Σ.
Gefragt:	Ist M voll?

Aufgabe 6: Komplexitätstheorie **(14 Punkte)**

(a) Definitionen und Eigenschaften (10 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.

Es { ☐ gilt ☐ gilt nicht }: Für alle Instanzpaare $I \neq I'$ im Definitionsbereich von f ist $f(I) \neq f(I')$.

Es { ☐ gilt ☐ gilt nicht }: Im Wertebereich von f kann es Instanzen geben, die nicht erzeugt werden.

Ein Problem $\mathcal{A}$ ist **NP**-[ ], wenn gilt:

für [ ] $\mathcal{B} \in$ [ ] existiert eine [ ]

Reduktion von [ ] auf [ ].

Es { ☐ gilt ☐ gilt nicht }: Jedes Problem in **NP** kann auf SAT reduziert werden.

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

[ ] und [ ].

Es { ☐ gilt ☐ gilt nicht }: Falls $\mathbf{P} \neq \mathbf{NP}$, dann ist $\mathbf{NP} \setminus \mathbf{P}$ eine Teilmenge von **NPC**.

(b) Schwache NP-Vollständigkeit (4 Punkte)

Ist CLIQUE unter der Annahme $\mathbf{P} \neq \mathbf{NP}$ schwach **NP**-vollständig? Begründen Sie.

Aufgabe 7: *NP*-Reduktion **(12 Punkte)**

TeureEinsen

Gegeben: Aussagenlogische Formel F in konjunktiver Normalform, wobei jede Klausel genau zwei Literale enthält; $k \in \mathbb{N}$.

Gefragt: Gibt es eine erfüllende Belegung für F, sodass höchstens k Variablen auf `true` gesetzt sind?

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

Hinweis: Denken Sie *graph*isch.

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

Name:

Gegeben:

Gefragt:

Fortsetzung von Aufgabe 7:

Aufgabe 8: Fixed Parameter Tractability: Kernelization **(6 Punkte)**

ParaMaxSat
Parameter: $k \in \mathbb{N}$.
Gegeben: Menge C von aussagenlogischen Klauseln mit je höchstens 3 Variablen.
Gefragt: Gibt es eine Variablenbelegung, die mindestens k Klauseln aus C erfüllt?

Zeigen Sie, dass das Problem ParaMaxSat einen Kernel der Größe $\leq 2k$ besitzt.

Geben Sie dazu einen polynomiellen Algorithmus an, der eine gegebene ParaMaxSat-Instanz entweder entscheidet oder eine äquivalente ParaMaxSat-Instanz zurückgibt, die aus maximal $2k$ Klauseln besteht. Begründen Sie.

Hinweis: Betrachten Sie für „zu große“ Instanzen die beiden Variablenbelegungen, bei denen alle Variablen auf denselben Wert gesetzt werden.

Aufgabe 9: Randomisierte Algorithmen (12 Punkte)

(a) Algorithmus analysieren (6 Punkte)

Betrachten Sie den randomisierten Algorithmus `BogoHamilton`, der das Problem GerichteterHamiltonKreis entscheiden soll. Sei A ein Array, $G \coloneqq (V, E)$ der gerichtete Eingabegraph und $n \coloneqq |V|$.

Geben Sie an, welche Erfolgswahrscheinlichkeit der Algorithmus für JA- und welche er für NEIN-Instanzen garantiert.

`BogoHamilton`:

$X \leftarrow V$
for $i \coloneqq 1, ..., n$**:**
  wähle v zufällig gleichverteilt aus X
  $A[i] \leftarrow v$
  $X \leftarrow X \setminus \{v\}$
if $A[1], ..., A[n]$ ist ein Hamiltonkreis**:**
  return JA
return NEIN

(b) Wahrscheinlichkeit pumpen (6 Punkte)

Sei $\mathcal{A}$ ein randomisierter Algorithmus, der JA- und NEIN-Instanzen jeweils mit Wahrscheinlichkeit $p \coloneqq 2/3$ korrekt erkennt. Wir betrachten den Algorithmus $\mathcal{A}_k$, bei dem $\mathcal{A}$ genau $k \in \mathbb{N}_u$ mal ausgeführt wird; der Rückgabewert entspricht der Mehrheit der Ergebnisse der einzelnen Durchläufe.

Geben Sie an, welche Erfolgswahrscheinlichkeit $\mathcal{A}_3$ für JA-Instanzen garantiert. Kürzen Sie soweit wie möglich.

Notizen:

Viel Erfolg!