Einführung in die Theoretische Informatik

Klausur, Nebentermin SoSe 2020, 28. September 2020 — Phase A

Aufgabe A1: Informationstheorie (10 Punkte)

Dateinamen: `A1_nmuster-1`, `A1_nmuster-2`, ...

(a) Erwartete Codewortlänge (4 Punkte)

Betrachten Sie die nebenstehende Tabelle für die Quelle (Σ, p).

Ist der gegebene Code $\mathbb{C}$ ein Code minimaler erwarteter Codewortlänge? Begründen Sie.

σ	a	b	c	d	e
p_σ	$\frac{1}{12}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{4}$
$\mathbb{C}(\sigma)$	110	111	01	10	00

(b) Erwarteter Informationsgewinn (6 Punkte)

Sei m_k eine Nachricht der Länge $k \in \mathbb{N}$, wobei die einzelnen Zeichen jeweils unabhängig aus der oben angegebenen Quelle (Σ, p) stammen.

Bestimmen Sie den erwarteten Informationsgewinn von m_k.
Vereinfachen Sie Ihr Ergebnis, sodass maximal 3 log-Symbole benötigt werden.

Aufgabe A2: Rechnende Turingmaschine (10 Punkte)

Dateinamen: `A2_nmuster-1`, `A2_nmuster-2`, ...

Sei $\mathbb{U}(\gamma)$ die Unärkodierung einer natürlichen Zahl γ. Erstellen Sie eine deterministische TM, die bei Eingabe $\mathbb{U}(\alpha)\square\mathbb{U}(\beta)$ die folgende Funktion berechnet:

$$f(\alpha, \beta) \coloneqq |\beta - \alpha|$$

Die Ausgabe soll ebenfalls in Unärkodierung erfolgen. Ihre TM soll **nicht mehr als 16 Zustände** haben. Sie müssen die korrekte Syntax der Eingabe nicht prüfen.

Aufgabe A3: Randomisierte Algorithmen (10 Punkte)

Dateinamen: `A3_nmuster-1`, `A3_nmuster-2`, ...

Gegeben ein Array A, gefüllt mit einer geraden Anzahl $n \geq 4$ von natürlichen Zahlen. Eine dieser Zahlen taucht genau $n/2$ mal auf, alle anderen Zahlen jeweils genau einmal. Der folgende Las-Vegas-Algorithmus findet in erwartet konstanter Zeit das mehrmals auftretende Element:

while true do
 wähle i zufällig gleichverteilt aus $\{1, ..., n\}$;
 wähle j zufällig gleichverteilt aus $\{1, ..., n\}$;
 if $A[i] == A[j]$ **then return** $A[i]$;

(a) Monte-Carlo-Algorithmus (8 Punkte)

Modifizieren Sie den oben angegebenen Algorithmus so, dass Sie einen Monte-Carlo-Algorithmus mit Fehlerwahrscheinlichkeit echt kleiner als 1/2 erhalten. Beweisen Sie, dass die Fehlerschranke eingehalten wird.

(b) Klassifizierung (2 Punkte)

Warum liegt das behandelte Problem nicht in **ZPP**?

Einführung in die Theoretische Informatik

Klausur, Nebentermin SoSe 2020, 28. September 2020 — Phase B

Aufgabe B1: Reguläre Sprachen (6 Punkte)

Dateinamen: `B1_nmuster-1`, `B1_nmuster-2`, ...

Die Sprache $L := \{a^i b^{2j} c^k d^j \mid i, j, k \in \mathbb{N}\}$ ist nicht regulär. Gegeben sind außerdem die Sprachen

$$L_0 := \{ab^{2x} c^x d^y \mid x, y \in \mathbb{N}_g\}, L_1 := \{a^x b^{2x} c^y d \mid x, y \in \mathbb{N}_u\} \text{ und } L_2 := \{d^i \mid i \geq 1\}.$$

Genau zwei der drei Sprachen $L'_i := L \cap L_i$ mit $i \leq 2$ sind regulär. Geben Sie eine möglichst kurze formale Beschreibung dieser regulären Sprachen an. Sie müssen ihre Regularität nicht beweisen!

Aufgabe B2: Pumping Lemma (10 Punkte)

Dateinamen: `B2_nmuster-1`, `B2_nmuster-2`, ...

(a) Zerlegung (2 Punkte)

Betrachten Sie die reguläre Sprache $L := \{a^i b^j \mid i \bmod 3 = 1, j \bmod 2 = 0\}$. Geben Sie eine Zerlegung gemäß des Pumping Lemmas für das Wort $z = a^7 b^6$ an.

(b) Anwendung (8 Punkte)

Beweisen Sie, dass $L := \{a^i b^j c^k \mid i \bmod 2 = 1, k \leq j + (j \bmod 2)\}$ nicht regulär ist.

Aufgabe B3: *NP*-Schwere (16 Punkte)

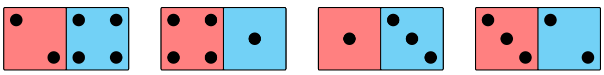
Dateinamen: `B3_nmuster-1`, `B3_nmuster-2`, ...

In *Dominu* ist es die Aufgabe der Spielenden, Spielsteine in einer Reihe aneinander zu legen. Ein Spielstein d besteht aus einem roten Feld $d[R]$ und einem blauen Feld $d[B]$, auf denen jeweils eine natürliche Zahl abgebildet ist (siehe Bild). Ziel ist es, eine Folge von mindestens k Spielsteinen auszuwählen, sodass (i) für zwei aufeinanderfolgende Steine das blaue Feld des vorderen mit dem roten Feld des hinteren übereinstimmen, (ii) das rote Feld des ersten Steins mit dem blauen Feld des letzten Steins übereinstimmen und (iii) keine zwei Steine dasselbe rote Feld haben.

DOMINU

Gegeben: Menge von Spielsteinen $D \subseteq \mathbb{N}^2$, $k \in \mathbb{N}$.

Gefragt: Gibt es ein $\ell \geq k$ und eine Folge $(d_1, ..., d_\ell) \in D^\ell$, sodass $d_i[B] = d_{i+1}[R]$ für alle $i < \ell$, $d_1[R] = d_\ell[B]$ und $d_i[R] \neq d_j[R]$ für alle $i \neq j$?

Beispiel: 

(a) Reduktion (12 Punkte)

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

Das bei Ihrer Reduktion verwendete Problem $\mathcal{X}$ muss aus der Vorlesung stammen. Geben Sie außerdem die Definition von $\mathcal{X}$ nach dem bekannten Schema (Gegeben/Gefragt) an.

(b) Ergänzung (4 Punkte)

Betrachten Sie die Problemvariante GERADESDOMINU, in welcher der Eingabewert k gerade sein muss. Skizzieren Sie, welche Änderungen Sie an Ihrer Konstruktion vornehmen müssen, um *NP*-Schwere von GERADESDOMINU zu zeigen.

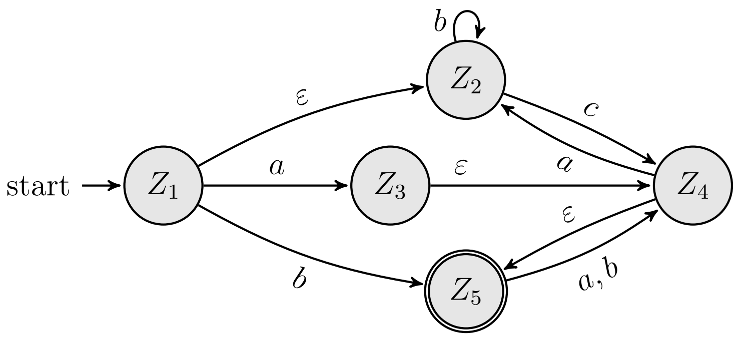
Einführung in die Theoretische Informatik

Klausur, Nebentermin SoSe 2020, 28. September 2020 — Phase C

Aufgabe C1: Umwandlung NDEA → DEA (8 Punkte)

Dateinamen: `C1_nmuster-1`, `C1_nmuster-2`, ...

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



Aufgabe C2: Berechnen und Entscheiden (14 Punkte)

Dateinamen: `C2_nmuster-1`, `C2_nmuster-2`, ...

(a) Berechenbarkeit (6 Punkte)

Begründen Sie für die folgende Funktion, ob sie berechenbar ist. Als Eingaben sind nur positive natürliche Zahlen erlaubt.

$$f(a,b) := \begin{cases} 3b^a \bmod 2b, & \text{wenn } \log_2 7 \leq a/b \\ \texttt{undef}, & \text{sonst.} \end{cases}$$

(b) Entscheidbarkeit (8 Punkte)

Ein gerichteter Graph $G = (V, E)$ heißt *schleifenlos*, wenn es keine Kante der Form $(v, v) \in E$ gibt. Eine Turingmaschine heißt *schleifenlos*, wenn der zugrundeliegende Graph schleifenlos ist. Beweisen oder widerlegen Sie:

Das Halteproblem für schleifenlose Turingmaschinen ist unentscheidbar.

Aufgabe C3: *FPT*-Algorithmus: Weder Suchbaum noch Kernel (6 Punkte)

Dateinamen: `C3_nmuster-1`, `C3_nmuster-2`, ...

r-CLIQUE$_{19}$	
Gegeben:	Graph $G = (V, E)$ mit Maximalgrad 19.
Parameter:	$r \in \mathbb{N}$.
Gefragt:	Gibt es eine Auswahl $S \subseteq V$ von genau r Knoten, sodass jedes Knotenpaar aus S benachbart ist?

Zeigen Sie durch Angabe eines deterministischen Algorithmus, dass r-CLIQUE$_{19}$ in ***FPT***-Zeit entscheidbar ist. Geben Sie eine Begründung für die ***FPT***-Laufzeit Ihres Algorithmus an.