

Einführung in die Theoretische Informatik
Klausur — SoSe 2024 — 8. Juli 2024

Haupttermin

Gruppe: Terbium / Thulium

Unbedingt ausfüllen

Matrikelnummer	Studiengang/Abschluss	Fachsemester
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
Nachname	Vorname	
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	
Unterschrift	Identifikator <small>(Beliebiges Wort zur Identifikation im anonymen Notenaushang)</small>	
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	

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)	8	8	12	8	18	10	10	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**(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 $H_{\Sigma, p}$ an, ohne $\mathcal{I}(p)$ zu verwenden.

(b) Erwarteter Informationsgewinn**(6 Punkte)**

Butler Hogarths Mops *Bello* ist international aufgewachsen und kläfft daher viersprachig, wobei die einzelnen Kläffgeräusche mit den folgenden Wahrscheinlichkeiten auftreten:

$$p(\text{wuff}) = 1/2$$

$$p(\text{bark}) = \begin{cases} 1/4 & \text{falls Bello bereits wuff gekläfft hat} \\ 1/8 & \text{sonst} \end{cases}$$

$$p(\text{guau}) = \begin{cases} 1/8 & \text{falls Bello bereits wuff gekläfft hat} \\ 1/4 & \text{sonst} \end{cases}$$

$$p(\text{ouaf}) = 1/8$$

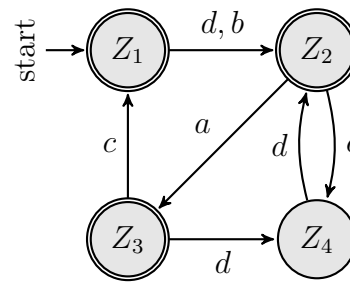
Bestimmen Sie den erwarteten Informationsgewinn, wenn Bello dreimal nacheinander kläfft. Vereinfachen Sie Ihr Ergebnis so weit wie möglich; insbesondere so, dass es keine log-Terme mehr enthält.

Hinweis: Bedenken Sie den erwarteten Informationsgewinn eines einzelnen Kläffens.

Aufgabe 2: Umwandlung DEA \rightarrow 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 $L := \{b^k a^j c^\ell \mid j \geq 1, k = (j \bmod 3) + 1, \ell = j - 1\}$ nicht regulär ist.

Aufgabe 4: Co-Semi-Entscheidbarkeit

(8 Punkte)

Sei $\mathbb{W}(\mathcal{M}) \in \{0, 1\}^*$ die Codierung einer rechnenden Turingmaschine \mathcal{M} und

$$L := \{\mathbb{W}(\mathcal{M}) \mid \mathcal{M} \text{ berechnet eine injektive Funktion } f: \mathbb{N} \rightarrow \mathbb{N}\}.$$

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

Hinweis: Eine Funktion $f: \mathbb{N} \rightarrow \mathbb{N}$ ist *injektiv*, wenn sie zwei unterschiedliche Eingabewerte niemals auf denselben Ausgabewert abbildet, d. h. $x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$ für alle $x_1, x_2 \in \mathbb{N}$.

Aufgabe 5: NP-Vollständigkeit

(18 Punkte)

(a) Definition

(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{P}_1 ist **NP**-schwer, wenn gilt: $\mathcal{P}_2 \in$ existiert eine Reduktion von auf .

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

(b) Reduktion

(12 Punkte)

AUSDAUERNDERAUTOMAT

Gegeben: Ein nicht-deterministischer endlicher Automat \mathcal{A} mit Eingabealphabet Σ , und ein Wort $w \in \Sigma^*$.

Gefragt: Existiert ein Weg in \mathcal{A} , den \mathcal{A} beim Lesen von w abgehen kann, sodass jeder Zustand in \mathcal{A} mindestens einmal besucht wird und \mathcal{A} am Ende w akzeptiert?

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

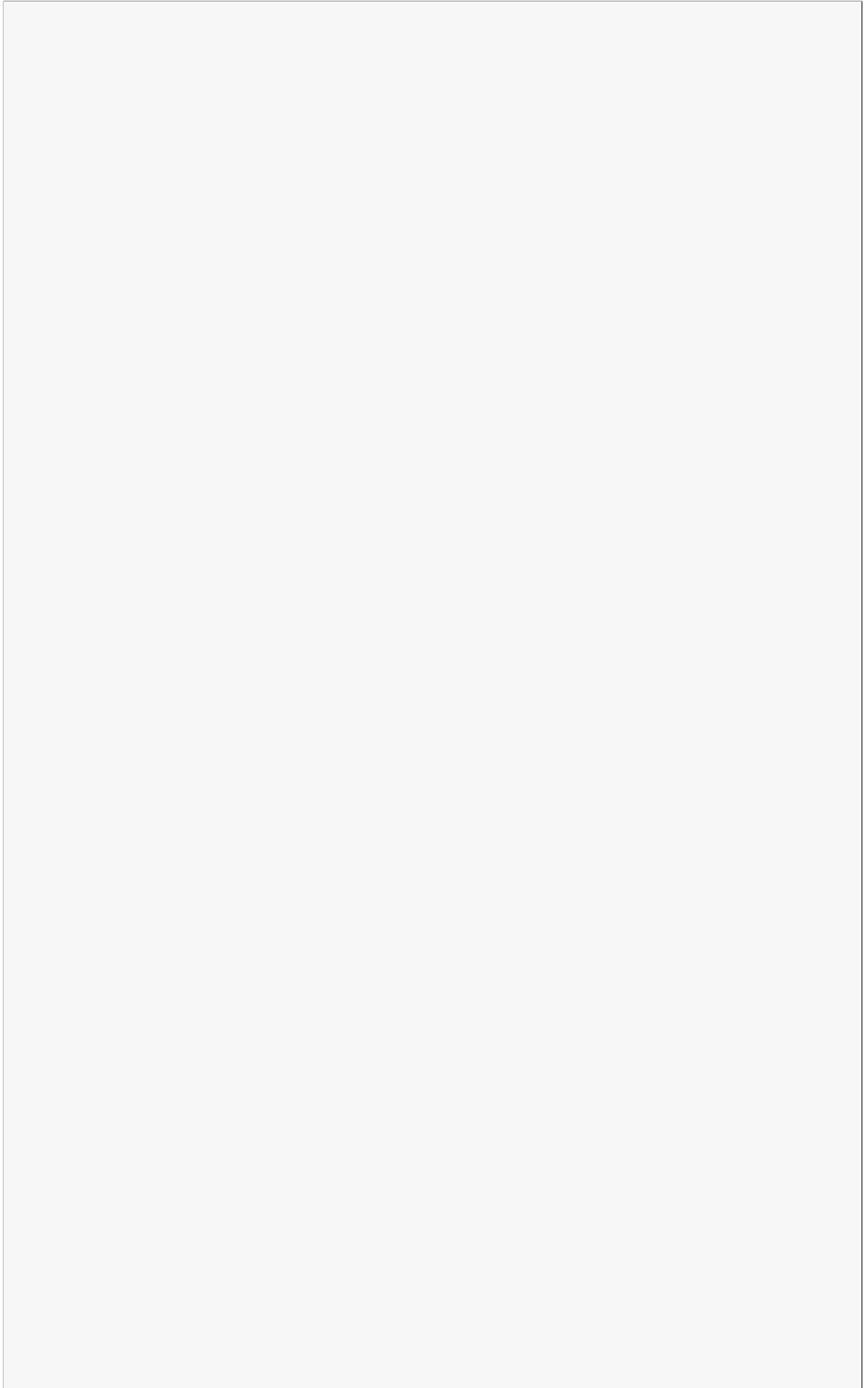
Definieren Sie das bei Ihrer Reduktion verwendete Problem \mathcal{X} :

Name:

Gegeben:

Gefragt:

Reduktionsbeweis:



Aufgabe 6: Fixed Parameter Tractability

(10 Punkte)

(a) Definition

(4 Punkte)

Wie lautet die Definition von *FPT*?

Ein Entscheidungsproblem \mathcal{X} mit Eingabelänge ℓ liegt in *FPT* in Bezug auf Parameter α , wenn...

(b) Tiefenbeschränkter Suchbaum

(6 Punkte)

5-CYCLE-COVERING

Parameter: $k \in \mathbb{N}$.

Gegeben: Ein ungerichteter Graph $G = (V, E)$.

Gefragt: Existiert eine Knotenteilmenge $S \subseteq V$ der Größe $|S| = k$, sodass alle Kreise der Länge maximal 5 in G einen Knoten aus S enthalten?

In einem Graph mit m Kanten kann man einen Kreis der Länge ≤ 5 in Zeit $\mathcal{O}(m)$ finden (bzw. `false` zurückgeben, falls kein solcher Kreis existiert).

Zeigen Sie: 5-CYCLE-COVERING liegt in *FPT* in Bezug auf k . Falls Sie einen Algorithmus nutzen, geben Sie eine (sinnvolle) obere Laufzeitschranke an.

Aufgabe 7: Randomisierte Algorithmen

(10 Punkte)

(a) **Definition**

(4 Punkte)

Definieren Sie *BPP*.

(b) **Algorithmus**

(6 Punkte)

Butler Hogarth möchte herausfinden, was das Lieblingsfutter seines Mopses *Moppel* ist. Für eine Mahlzeit füllt Hogarth Moppels Futternapf immer mit genau einer Portion eines bestimmten Futters. Er weiß: Handelt es sich um Moppels Lieblingsfutter, wird Moppel es immer fressen. Ansonsten frisst Moppel es nur mit einer Wahrscheinlichkeit von 75%.

Geben Sie einen *Co-RP*-Algorithmus an, um innerhalb möglichst weniger Mahlzeiten zu entscheiden, ob es sich bei den „Frischkäse-Flips“ um Moppels Lieblingsfutter handelt. Die entsprechende Fehlerwahrscheinlichkeit soll bei weniger als 50% liegen. Wieviele Mahlzeiten benötigt Ihr Algorithmus?

Hinweis: Sie dürfen annehmen, dass sowohl die Dauer einer einzelnen Mahlzeit als auch der zeitliche Abstand zur jeweils nächsten Mahlzeit konstant sind.

Algorithmus **und** Analyse:

Fortsetzung Aufgabe 7:

Notizen: