

Informatik D: Einführung in die Theoretische Informatik

Klausur — SoSe 2013 — 11. Juli 2013

(Prüfungsnr. 1007049)

Gruppe: Joker, Pinguin

Unbedingt ausfüllen

Matrikelnummer	Studiengang/Abschluss	Fachsemester
<input type="text"/>	<input type="text"/>	<input type="text"/>
Nachname	Vorname	
<input type="text"/>	<input type="text"/>	
Unterschrift	Identifikator <small>(Beliebiges Wort zur Identifikation im anonymen Notenaushang)</small>	
<input type="text"/>	<input 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	8	9	10	11	12	Σ
Punkte (max)	8	6	10	6	8	12	12	16	8	14	12	12	<b>124</b>
Punkte (erreicht)													

<b>Punkte</b>	0..61	62..67	68..74	75..80	81..86	87..93	94..99	100..105	106..111	112..118	119..124
<b>Note</b>	<b>5,0</b>	<b>4,0</b>	<b>3,7</b>	<b>3,3</b>	<b>3,0</b>	<b>2,7</b>	<b>2,3</b>	<b>2,0</b>	<b>1,7</b>	<b>1,3</b>	<b>1,0</b>

Note:

### Aufgabe 1: Sprachen vs. Automaten

(8 Punkte)

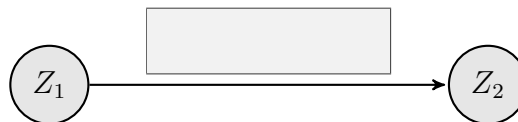
Geben Sie zu den Sprachklassen jeweils alle Automatenklassen an, die zum Entscheiden des Wortproblems genau dieser Sprachklasse (aber keine größeren) genutzt werden können. Vergessen Sie nicht Determinismus/Nicht-Determinismus, etc. zu spezifizieren!

Sprachklasse	Automaten
Deterministisch kontextfreie Sprachen	
Kontextsensitive Sprachen	
Rekursiv aufzählbare Sprachen	
Reguläre Sprachen	

### Aufgabe 2: Übergänge

(6 Punkte)

Welche Daten stehen an einem Zustandsübergang eines Kellerautomaten? Beschreiben Sie diese kurz aber präzise. Was bedeuten Sie?



*Erklärung:*

**Aufgabe 3: Sprachen klassifizieren****(10 Punkte)**

Zu welcher Sprachklasse gehören die folgenden Sprachen? Kreuzen Sie dabei **alle** korrekten Antworten an. (*Falsche Kreuzchen führen zu Punkteabzug innerhalb der Teilaufgabe.*)

	regulär	determ. kontextfrei	kontextfrei	kontextsensitiv	rek. aufzählbar
$\{w^Racwa \mid w \in \{b, c\}^*\}$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$\{ab^{i-2}c \mid i \geq 2\}$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$\{w \mid w \text{ kodiert eine rationale Zahl im Dezimalsystem}\}$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$\left\{ w \mid \begin{array}{l} w \text{ kodiert einen Algorithmus, der} \\ \text{auf das Leerwort angewendet anhält} \end{array} \right\}$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Aufgabe 4: Regulärer Ausdruck****(6 Punkte)**

Erstellen Sie einen regulären Ausdruck, der zur folgenden Grammatik äquivalent ist.

$$S \rightarrow bAbB$$

$$A \rightarrow aA \mid a$$

$$B \rightarrow cc \mid dd$$

**Aufgabe 5: Kontextfreie Grammatik****(8 Punkte)**

Erstellen Sie eine kontextfreie Grammatik, die genau alle Palindrome gerader Länge über dem Alphabet  $\Sigma = \{a, b, c\}$  umfasst, die mit „ab“ beginnen. Achten Sie darauf, dass die Grammatik den strikten Regeln der kontextfreien Grammatiken gehorcht!

### Aufgabe 6: Turingmaschine

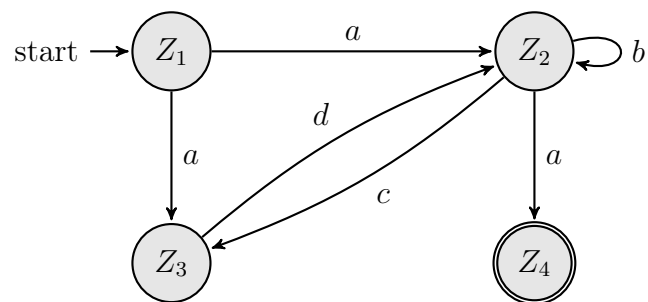
(12 Punkte)

Erstellen Sie eine Turingmaschine, die genau die Sprache  $\{ab^{2i} \mid i \geq 0\}$  akzeptiert.

### Aufgabe 7: Reguläre Grammatik

(12 Punkte)

Wandeln Sie—gemäß dem Vorgehen aus der Vorlesung!— den folgenden nicht-deterministischen endlichen Automaten in eine reguläre Grammatik (strikte Regeln!) um.



## Aufgabe 8: Pumping Lemma

(16 Punkte)

### (a) Definition

(4 Punkte)

Vervollständigen Sie die Definition des Pumping Lemmas für reguläre Sprachen.  
(Falsche Kreuzchen führen zu Punkteabzug innerhalb der Teilaufgabe.)

Sei $L$ eine reguläre Sprache. Es gibt eine Zahl $n := n(L)$ , so dass sich
$\left\{ \begin{array}{l} \input{checkbox} \text{ jedes} \\ \input{checkbox} \text{ mindestens ein} \\ \input{checkbox} \text{ kein} \end{array} \right\}$ Wort $z \in L$ mit $ z  \geq n$ so in drei Teile $u, v, w$
zerlegen lässt (d.h. $z = uvw$ ), dass die folgenden Eigenschaften gelten:
<input type="checkbox"/> $ u  \geq 1$ <input type="checkbox"/> $ vw  \geq n$ <input type="checkbox"/> $u^+vw \in L$
<input type="checkbox"/> $ v  \geq 1$ <input type="checkbox"/> $ uw  \leq n$ <input type="checkbox"/> $uv^iw \in L$ für alle $i \geq 0$

### (b) Anwendung

(12 Punkte)

Zeigen Sie, dass die Sprache  $\{b^i a^j b^{i+j} \mid i, j \in \mathbb{N}\}$  nicht regulär ist. Vergessen Sie nicht, Begründungen für Ihre Beweisschritte anzugeben.

### Aufgabe 9: Post'sches Korrespondenzproblem

(8 Punkte)

Sei  $\mathcal{T} = \{(x_i, y_i)\}_{1 \leq i \leq k}$  die Instanz eines Post'schen Korrespondenzproblems (PCP). Betrachten Sie den folgenden Algorithmus zur Lösung des PCPs:

```
forall  $\mathcal{S} \subseteq \mathcal{T}$ :  
  forall Permutationen  $\pi(\mathcal{S})$  von  $\mathcal{S}$ :  
    if  $\pi(\mathcal{S})$  ist eine zulässige Lösung:  
      return true  
return false
```

Der Algorithmus prüft jede mögliche Auswahl und Reihenfolge. Davon gibt es jedoch nur endlich (exponentiell) viele. Daher hält der Algorithmus nach endlicher Zeit und entscheidet das PCP. In der Vorlesung haben wir jedoch besprochen, dass das PCP unentscheidbar ist. Wo liegt der Fehler in obiger Argumentation?

### Aufgabe 10: Probleme in **P**

(14 Punkte)

Ordnen Sie die folgenden Probleme allen Komplexitätsklassen zu, die sie enthalten (unter der Annahme, dass  $P \neq NP$ ). (Achtung: Falsche Antworten zählen innerhalb der Aufgabe negativ!)

<b>P</b>	<b>NP</b>	<b>NP-vollständig</b>	<b>NP-schwer</b>	keines der genannten	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SubsetSum
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Wang Kachelung
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Entscheiden, ob ein gerichteter Graph Kreise enthält
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sortieren
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Wortproblem über deterministisch kontextfreie Sprachen
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Testen, ob ein Spannbaum mit Gewicht $\leq k$ existiert
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Suchen der kürzesten Rundtour, die alle Knoten eines Graphen genau einmal besucht

**Aufgabe 11: NP-Vollständigkeit**

(12 Punkte)

**(a) Optimierungs- vs. Entscheidungsproblem**

(6 Punkte)

Was ist das zu folgendem Optimierungsproblem zugehörige Entscheidungsproblem?

**Gegeben:** Eine Menge  $M = \{0, 1, 2, 3, \dots, m\}$  von Gegenständen. Jeder Gegenstand  $i$  ( $1 \leq i \leq m$ ) hat ein Gewicht  $g_i$  und einen Wert  $w_i$ . Zusätzlich haben wir eine Gewichtsschranke  $G$ .

**Gesucht:** Eine Teilmenge  $M' \subseteq M$  mit maximalem Gesamtwert, deren Gesamtgewicht maximal  $G$  ist.

**(b) Reduktionen**

(6 Punkte)

Welche zwei Bedingungen muss eine Reduktion von einem Problem  $\mathcal{X}$  auf ein Problem  $\mathcal{Y}$  erfüllen, um sie in einem **NP**-Vollständigkeitsbeweis zu benutzen?

Von welchem der beiden Probleme wird in einem solchen Beweis die **NP**-Härte gezeigt?

**Aufgabe 12: Komplexitätsfragen**

(12 Punkte)

Stimmen die folgenden Aussagen aus der Komplexitätstheorie?

(Achtung: Pro Frage gibt es +2/0/-2 Punkte bei einer richtigen/keinen/falschen Antwort!)

ja    nein

- |                          |                          |                                                                                                                                                               |
|--------------------------|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <input type="checkbox"/> | <input type="checkbox"/> | Falls $P = NP$ , dann gibt es keine <b>NP</b> -vollständigen Probleme.                                                                                        |
| <input type="checkbox"/> | <input type="checkbox"/> | Ein Problem, das in <b>Co-NP</b> liegt, kann nicht in <b>NP</b> liegen.                                                                                       |
| <input type="checkbox"/> | <input type="checkbox"/> | Jedes unentscheidbare Problem ist <b>NP</b> -schwer.                                                                                                          |
| <input type="checkbox"/> | <input type="checkbox"/> | Schwach <b>NP</b> -vollständige Probleme erlauben polynomielle Laufzeiten, wenn die Eingabegröße polynomiell in den vorkommenden Zahlenwerten beschränkt ist. |
| <input type="checkbox"/> | <input type="checkbox"/> | Wenn man für ein Entscheidungsproblem einen Exponentialzeit-Algorithmus kennt, dann liegt das Problem in <b>NP</b> , aber sicher nicht in <b>P</b> .          |
| <input type="checkbox"/> | <input type="checkbox"/> | Wenn man für ein Entscheidungsproblem einen Exponentialzeit-Algorithmus kennt, dann liegt das Problem in <b>NP</b> , aber vielleicht nicht in <b>P</b> .      |