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

# Informatik D: Einführung in die Theoretische Informatik

# Klausur — SoSe 2014 — 7. Okt. 2014

Nebentermin, Prüfungsnr. 1007049 — Gruppe: Zwickl/Groschn

## 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! Falsche Kreuzchen können zu Punkteabzug innerhalb der Teilaufgabe führen.
- 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 | Σ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Punkte (max) | 12 | 10 | 8 | 16 | 12 | 12 | 10 | 12 | 18 | 20 | **130** |
| Punkte (erreicht) | | | | | | | | | | | |

| **Punkte** | 0..64 | 65..72 | 73..79 | 80..84 | 85..89 | 90..95 | 96..100 | 101..105 | 106..110 | 111..117 | 118..130 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **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: Chomsky-Hierarchie **(12 Punkte)**

### (a) Sprachen und Automaten (6 Punkte)

Zu jeder Sprachklasse in der Chomsky-Hierarchie gibt es entsprechende Automaten. Vervollständigen Sie die folgende Tabelle! Geben Sie dabei alle Automaten an, die genau die Sprachklasse beschreiben, bzw. umgekehrt.

| Sprachklasse | Automaten |
|---|---|
| deterministisch kontextfrei | |
| regulär | |
| | Nicht-determ. linear-beschränkter Automat |

### (b) Definition (6 Punkte)

Definieren Sie rekursiv, was ein regulärer Ausdruck über einem Alphabet $\Sigma$ ist.

Basisfälle:

- Symbole aus $\Sigma$ sind reguläre Ausdrücke.
- ______ ist ein regulärer Ausdruck.

Rekursion:
Seien $E_1$ und $E_2$ reguläre Ausdrücke, dann sind auch die folgenden Ausdrücke regulär:

**Aufgabe 2: Sprachen klassifizieren** **(10 Punkte)**

Zu welcher Sprachklasse gehören die folgenden Sprachen? Kreuzen Sie dabei *alle* korrekten Antworten an.

| | regulär | determ. kontextfrei | kontextfrei | kontextsensitiv | rek. aufzählbar |
|---|---|---|---|---|---|
| $\left\{ w \;\middle|\; \begin{array}{l} w = \mathbb{W}(\mathcal{M}) \text{ ist die Kodierung einer Turingmaschine } \mathcal{M}, \\ \text{und } \mathcal{M} \text{ hält auf Eingabe } \varepsilon. \end{array} \right\}$ | ☐ | ☐ | ☐ | ☐ | ☐ |
| $\{w_1 w_2 \mid w_1, w_2 \in \{a, b, c\}^*, \lvert w_1 \rvert = \lvert w_2 \rvert\}$ | ☐ | ☐ | ☐ | ☐ | ☐ |
| $\{w_1\, c\, w_2 \mid w_1, w_2 \in \{a, b\}^*, \lvert w_1 \rvert = \lvert w_2 \rvert\}$ | ☐ | ☐ | ☐ | ☐ | ☐ |
| $\{w \mid w \text{ ist die Binärdarstellung einer geraden natürlichen Zahl}\}$ | ☐ | ☐ | ☐ | ☐ | ☐ |

**Aufgabe 3: Reguläre Grammatik und endlicher Automat** **(8 Punkte)**

Wandeln Sie – entsprechend dem Vorgehen aus der Vorlesung – die folgende reguläre Grammatik in einen endlichen Automaten um:

$$
\begin{aligned}
S &\to aS \mid aP \mid e \\
P &\to eQ \mid iR \mid i \\
Q &\to oP \mid oQ \mid iQ \\
R &\to uS
\end{aligned}
$$

## Aufgabe 4: Pumping Lemma **(16 Punkte)**

### (a) Definition (4 Punkte)

Vervollständigen Sie die Definition des Pumping Lemmas für kontextfreie Sprachen.

Sei $L$ eine kontextfreie Sprache. Es gibt eine Zahl $n := n(L)$, sodass sich jedes Wort

$z \in L$ mit $\left\{ \begin{array}{l} \square\ |z| \leq n \\ \square\ |z| = n \\ \square\ |z| \geq n \end{array} \right\}$ zerlegen lässt als $\left\{ \begin{array}{l} \square\ z = uvwxy \\ \square\ z = uvw \\ \square\ z = uv^*w \end{array} \right\}$

mit den Eigenschaften:

__________, __________, __________.

### (b) Anwendung (12 Punkte)

Zeigen Sie mittels des Pumping Lemmas, dass die Sprache $\{aa^{4i}bb^{2i}cc^{i} \mid i \geq 0\}$ nicht kontextfrei ist.

## Aufgabe 5: Heidewitzka! **(12 Punkte)**

Wir betrachten die folgende Sprache:

– Die Wörter benutzen keine Buchstaben außer '$a$', '$b$', '$c$' und '$d$'.
– Die Anzahl der '$c$'s in einem Wort entspricht der Anzahl der '$a$'s und '$b$'s zusammen.

### (a) Automatenklasse (2 Punkte)

Welches ist die eingeschränkteste nichtdeterministische Automatenklasse, die diese Sprache beschreiben kann?

### (b) Automatenkonstruktion (10 Punkte)

Geben Sie einen solchen Automaten an, der genau diese Sprache beschreibt.

**Aufgabe 6: Rechnende Turingmaschine** **(12 Punkte)**

Gegeben eine unär kodierte Zahl $\alpha \geq 1$. Geben Sie eine Turingmachine an, die die folgende Funktion berechnet:

$$f(\alpha) := \begin{cases} \alpha/2 & \alpha \in \mathbb{N}_g \\ \text{undef} & \text{sonst} \end{cases}$$

## Aufgabe 7: WHILE und GOTO **(10 Punkte)**

### (a) WHILE-Programm (6 Punkte)

Geben Sie ein WHILE-Programm an, das $x_2 := \lfloor x_1/2 \rfloor$ berechnet. Sie dürfen beliebige Vergleiche $(=, \neq, >, \geq, \leq, <)$ mit Konstanten verwenden.

### (b) GOTO-Programm (4 Punkte)

Gegeben sei folgendes GOTO-Programm mit Eingabe $x_1, x_2$ und Ausgabe $x_3$:

$$
\begin{aligned}
& x_3 := 0 \\
& \textbf{if } x_2 = 0 \textbf{ goto } L_2 \\
& x_4 := x_1 \\
L_0:\ & \textbf{if } x_4 = 0 \textbf{ goto } L_2 \\
& x_4 := x_4 - 1 \\
& x_5 = x_2 \\
L_1:\ & x_5 = x_5 - 1 \\
& x_3 = x_3 + 3 \\
& \textbf{if } x_5 = 0 \textbf{ goto } L_0 \\
& \textbf{goto } L_1 \\
L_2:\ & \textbf{halt}
\end{aligned}
$$

Was berechnet dieses GOTO-Programm?

## Aufgabe 8: Busy Beaver **(12 Punkte)**

**(a) Definition** (4 Punkte)

Was ist ein $n$-Busy-Beaver?

**(b) Funktion** (2 Punkte)

Was ist die Busy-Beaver-Funktion?

**(c) Berechenbarkeit** (6 Punkte)

Warum ist diese Funktion nicht berechenbar?

## Aufgabe 9: $\boldsymbol{P}$ und $\boldsymbol{NP}$ (18 Punkte)

Wir nehmen im Folgenden immer an, dass $P \neq NP$ ist.

### (a) Polynomialzeit (4 Punkte)

Sowohl $P$ als auch $NP$ beschreiben Probleme, die gemäß Definition in polynomieller Zeit gelöst werden können. Was ist der Unterschied und warum sind Probleme in $NP$ im Allgemeinen dennoch „schwerer"?

### (b) Zeuge (4 Punkte)

Betrachten Sie ein beliebiges $NP$-vollständiges Problem. Wir können für jede Ja- und für jede Nein-Instanz einen Zeugen (d.h. einen Beweis, dass die Instanz eine Ja- bzw. Nein-Instanz ist) angeben. Was ist dabei der wesentliche Unterschied?

### (c) Zusammenhänge (10 Punkte)

Welche Aussagen stimmen?
*(Achtung: Pro Frage gibt es $+2/0/-1$ Punkte bei einer richtigen/keinen/falschen Antwort! Sie erhalten jedoch natürlich mindestens 0 Punkte für die gesamte Aufgabe.)*

| korrekt | falsch | |
|---|---|---|
| ☐ | ☐ | Optimierungsprobleme können $NP$-schwer, aber nicht $NP$-vollständig sein. |
| ☐ | ☐ | Sei $\mathcal{X}$ ein Problem, das in $NP \cap Co\text{-}NP$ liegt, aber nicht in $P$. Dann gilt $\mathcal{X} \in NPI$. |
| ☐ | ☐ | Es gibt Probleme, die in $P$, aber nicht in $NP$ liegen. |
| ☐ | ☐ | Wenn ein schwach $NP$-vollständiges Problem in polynomieller Zeit gelöst werden kann, so sind die Werte der Eingabezahlen durch eine Konstante beschränkt. |
| ☐ | ☐ | Co-Primes liegt in $P$. |

## Aufgabe 10: ***NP*-vollständig** (20 Punkte)

Wir definieren das folgende Problem:

**Problem:** 3-PARTITION

**Gegeben:** Eine Menge $A = \{a_1, \ldots, a_n\}$ von natürlichen Zahlen.

**Gefragt:** Existiert eine Aufteilung von $A$ in paarweise disjunkte Teilmengen $X, Y, Z$ (d. h. $A = X \cup Y \cup Z$ und $X \cap Y = X \cap Z = Y \cap Z = \emptyset$), sodass $\sum_{x \in X} x = \sum_{y \in Y} y = \sum_{z \in Z} z$ ?

Wir wollen zeigen, dass 3-PARTITION ***NP***-vollständig ist.
Dafür müssen wir zeigen, dass es... (2 Punkte)

- ☐ ...(i) in ***P*** liegt und (ii) sich alle Probleme aus ***P*** darauf reduzieren lassen.
- ☐ ...(i) in ***P*** liegt und (ii) sich alle Probleme aus ***NP*** darauf reduzieren lassen.
- ☐ ...(i) in ***NP*** liegt und (ii) sich alle Probleme aus ***P*** darauf reduzieren lassen.
- ☐ ...(i) in ***NP*** liegt und (ii) sich alle Probleme aus ***NP*** darauf reduzieren lassen.

Zeigen Sie (i) für 3-PARTITION. (4 Punkte)

Um (ii) zu zeigen, reduzieren wir von...

☐ CLIQUE ☐ SUBSETSUM ☐ PARTITION ☐ PCP

Definieren Sie das von Ihnen angekreuzte Problem. (4 Punkte)

Warum reicht es, nur von *einem* Problem zu reduzieren? (2 Punkte)

Beweisen Sie nun (ii). (8 Punkte)



*Viel Erfolg!*