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

Informatik D: Einführung in die Theoretische Informatik

Klausur — SoSe 2013 — 14. Okt. 2013

Nebentermin, Prüfungsnr. 1007049 — Gruppe: Fisole

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	11	Σ
Punkte (max)	6	10	8	12	16	12	12	12	12	8	20	**128**
Punkte (erreicht)												

Punkte	0..63	64..71	72..79	80..85	86..90	91..96	97..101	102..106	107..112	113..120	121..128
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: Sprachen vs. Automaten **(6 Punkte)**

Geben Sie die jeweils größte Sprachklasse an, die mit dem gegebenen Automaten beschrieben werden kann.

Automaten	**Sprachklasse**
Deterministischer Endlicher Automat	
Nicht-Deterministischer Kellerautomat, Akzeptanz durch leeren Keller	
Deterministische Turingmaschine	

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
$\{das, ist, eine, sprache\}$	☐	☐	☐	☐	☐
$\{w \mid w$ beschreibt eine immer-terminierende Turingmaschine$\}$	☐	☐	☐	☐	☐
$\{ww^R \mid w \in \{a,b\}\} \cap \{a^i bb^i a \mid i \geq 0\}$	☐	☐	☐	☐	☐
$\{w \mid w$ ist ein korrekt geklammerter algebraischer Ausdruck$\}$	☐	☐	☐	☐	☐

Aufgabe 3: Regulärer Ausdruck vs. Reguläre Grammatik **(8 Punkte)**

Erstellen Sie (nicht unbedingt nach dem Vorgehen der Vorlesung) eine reguläre Grammatik (gemäß den strikten Regeln), die zu folgendem regulären Ausdruck äquivalent ist:

$$(ba)^*(c^+|b)$$

Aufgabe 4: Kellerautomat **(12 Punkte)**

Geben Sie einen Kellerautomaten an, der genau die Sprache $\{a^i(cd)^ja^i \mid i, j \geq 0\}$ akzeptiert.

Aufgabe 5: Normalformen **(16 Punkte)**

(a) Definitionen (6 Punkte)

Geben Sie die größte Sprachfamilie an, deren Grammatiken immer in Chomsky- *und* Greibach-Normalform geschrieben werden können?

Seien V die Variablen und Σ die Symbole einer solchen Sprache. Welche Form müssen die Regeln in diesen Normalformen immer haben (formal, kein Beschreibungstext)?

Chomsky-Normalform (CNF):

Greibach-Normalform (GNF):

(b) Transformation (10 Punkte)

Gegeben eine Grammatik G (nicht in CNF), die in CNF umgeformt werden kann. Was sind, gemäß dem Vorgehen aus der Vorlesung, die notwendigen Schritte um G in CNF zu verwandeln? Beschreiben Sie das Ziel jedes Schritts in einem kurzen Satz.

Schritt 1:

Schritt 2:

Teilschritt 2.a:

Teilschritt 2.b:

Schritt 3:

Aufgabe 6: Pumping Lemma **(12 Punkte)**

Mittels des Pumping Lemmas kann man zeigen, dass die Sprache $L = \{a^j f b^j g c^j \mid j \geq 0\}$ nicht kontextfrei ist. Der folgende „Beweis" enthält jedoch **drei Fehler**.

(Fehlerhafter) Beweis:

Angenommen die Sprache L wäre kontextfrei, dann würde das Pumping Lemma für irgendeine Wortmindestlänge $n := n(L)$ gelten. Betrachten wir das Wort $z = a^n f b^n g c^n \in L$.

Das Pumping Lemma garantiert, dass eine Zerlegung $z = uvwxy$ in fünf Teile mit den Eigenschaften $|v| \geq 1, |x| \geq 1$ und $|vwx| \leq n$ existiert, so dass $uv^i wx^i y \in L$ für alle $i \geq 0$ (insbesondere also auch $uwy \in L$).

Wir wissen, dass vx weder das Symbol „f" noch „g" enthält, da diese Symbole sonst in $uwy \in L$ fehlen würden.

Es bleibt also die folgende Fallunterscheidung:

1. Angenommen „f" liegt in u. Dann liegen auch alle „a"s in u, und v muss „b"s oder „c"s enthalten. Im Wort uwy gibt es also noch n viele „a"s, die Anzahl der „b"s und/oder „c"s hat aber gegenüber z abgenommen. Widerspruch.

2. Angenommen „g" liegt in y. Der Beweis ist analog zu Fall 1, wobei nun die Anzahl der „c"s n bleibt, und die der „a"s und/oder „b"s abnimmt.

3. Falls weder Fall 1 noch 2 vorliegt, dann sind „f" und „g" in w. Also liegen auch alle „b"s in w; v enthält „a"s und/oder x enthält „c"s. Im Wort uwy gibt es also noch n viele „b"s, die Anzahl der „a"s und „c"s hat aber gegenüber z abgenommen. Widerspruch.

Markieren Sie die Fehler direkt im Text oberhalb, und geben Sie die notwendigen Korrekturen unterhalb an. (In Ihren Korrekturen, sollten sie möglichst wenig neu schreiben müssen!)
Hinweis: Der zweite Fehler befindet sich in Fall 1 der Fallunterscheidung.

Korrekturen:

Fehler 1:

Fehler 2:

Fehler 3:

Aufgabe 7: Rechnende Turingmaschine (12 Punkte)

Gegeben eine binär kodierte natürliche Zahl $\alpha > 1$. Geben Sie eine deterministische Turingmaschine an, die $\alpha + 2$ berechnet.

Aufgabe 8: WHILE (12 Punkte)

Geben Sie ein WHILE-Programm an, das $x_3 := 3x_1 - x_2$ berechnet (es gilt $x_1 > x_2$). Benutzen Sie dabei (zusätzlich zur Verkettung „;“) nur folgende Sprachkonstrukte:

- $x_i := c$ und $x_i := x_j \pm c$, für Konstanten $c \in \mathbb{N}$
- $\mathsf{while}(x_i \neq 0)\ \{\ldots\}$

Aufgabe 9: Kolmogorov-Komplexität **(12 Punkte)**

Zeigen Sie, dass das Wort $w = abab \ldots ab$ mit 1200 Zeichen eine Kolmogorov-Komplexität $K_{\mathsf{GOTO}^+}(w)$ unter 600 hat. Dabei ist GOTO^+ eine Erweiterung der GOTO-Programmiersprache mit folgenden Anweisungen:

- $x_i := c$ und $x_i := x_j \pm c$, für Konstanten $c \in \mathbb{N}$
- goto L_i
- if$(x_i \neq 0)$ goto L_j
- halt

(normale GOTO-Programmiersprache)

- print(S), wobei der Parameter S eine Symbolkette ist, die ausgegeben werden soll

Geben Sie auch eine Begründung an, warum das, was Sie da tun, die gesuchte Aussage zeigt!

GOTO^+ :

Begründung:

Aufgabe 10: Komplexitäten (8 Punkte)

Nehmen Sie innerhalb dieser Aufgabe an, dass $\mathsf{P} \neq \mathsf{NP}$. Geben Sie jeweils ein Problem (außer dem TSP, siehe nächste Aufgabe) an, dass in der gesuchten Komplexitätsklasse liegt.

P:

$\mathsf{NP} \setminus \mathsf{P}$:

stark NP-vollständig:

schwach NP-vollständig:

Aufgabe 11: NP-Vollständigkeit: TSP (20 Punkte)

Das *Traveling Salesman Problem (TSP)* ist wie folgt definiert:

> Gegeben ein ungerichteter vollständiger Graph $G = (V, E)$ mit Kantenkosten $c : E \to \mathbb{N}^+$, und eine natürliche Zahl K. Gibt es eine Rundtour mit Kosten $\leq K$, die jeden Knoten aus V genau einmal besucht?

Es ist trivial, dass TSP in NP liegt. Zeigen Sie, dass TSP auch NP-schwer, und damit insgesamt NP-vollständig ist.

Der Beweis erfolgt durch Reduktion

☐ von ☐ auf

☐ 3-SAT ☐ HAMILTONKREIS ☐ EULERTOUR ☐ SUBSETSUM ☐ HALTEPROBLEM

Definieren Sie das angekreuzte Problem:

Mittels der Reduktion müssen wir zeigen, dass

- ☐ das angekreuzte Problem ein Spezialfall des TSP ist.
- ☐ TSP ein Spezialfall des angekreuzten Problems ist.

Beschreiben Sie die notwendige (deterministisch in Polynomialzeit durchführbare) Reduktion:

Durch welche zwei Beweisschritte zeigt man, dass die (deterministisch in Polynomialzeit durchführbare) Reduktion korrekt ist?

Zeigen Sie, dass Ihre (deterministisch in Polynomialzeit durchführbare) Reduktion korrekt ist:

Viel Erfolg!