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

Informatik D: Einführung in die Theoretische Informatik

Klausur — SoSe 2015 — 27. Juli 2014

Haupttermin, Prüfungsnr. 1007049 — Gruppe: Rebellen (Luke, Han)

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	8	16	12	16	12	12	14	10	20	**132**
Punkte (erreicht)											

Punkte	0..65	66..73	74..81	82..86	87..91	92..96	97..101	102..106	107..112	113..119	120..132
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: Sprache vs. Grammatik **(12 Punkte)**

(a) Hierarchie und Automaten (8 Punkte)

Zu jeder Sprache gibt es entsprechende Automaten. Vervollständigen Sie die folgende Tabelle:

Automaten	Name der Sprachfamilie
	monoton
DTM,	
	kontextfrei

(b) Grammatikdefinition (4 Punkte)

Welche Sprachenklasse wird durch Grammatiken in Greibach-Normalform genau beschrieben?

Sei V die Menge der Variablen, und Σ das betrachetet Alphabet. Definieren Sie oben genannte Normalform.

Alle Regeln haben die Form...

Aufgabe 2: Sprachen klassifizieren **(8 Punkte)**

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

	regulär	determ. kontextfrei	kontextfrei	kontextsensitiv	rek. aufzählbar
$\{w_1w_2 \mid w_1, w_2 \in \{a,b,c\}^*, 3\lvert w_1\rvert \bmod 5 = 2\lvert w_2\rvert \bmod 5\}$	☐	☐	☐	☐	☐
$\{ww^R \mid w$ ist Wort einer DKF Sprache über $\Sigma = \{$♪, ♩, 𝅗𝅥$\}\}$	☐	☐	☐	☐	☐
$\{a^ib^jc^{i+j} \mid i \geq j \geq 2\}$	☐	☐	☐	☐	☐

Aufgabe 3: Pumping Lemma **(16 Punkte)**

(a) Definition (4 Punkte)

Vervollständigen Sie die Definition des Pumping Lemmas für kontextfreie Sprachen.

- ☐ Sei L eine kontextfreie Sprache. Für jede Zahl n gibt es ein Wort $z \in L$, sodass $|z| \geq n$ und eine Zerlegung $z = uvwxy$ existiert, mit...
- ☐ Sei L eine kontextfreie Sprache. Es gibt eine Zahl n, sodass für jedes Wort $z \in L$ mit $|z| \geq n$ eine Zerlegung $z = uvwxy$ existiert, mit...
- ☐ Sei L eine nicht-kontextfreie Sprache L. Es gibt es eine Zahl n, sodass ein Wort $z \in L$ mit $|z| \geq n$ existiert, für das eine Zerlegung $z = uvwxy$ existiert, mit...

...den folgenden drei Eigenschaften:

(1) , (2) , (3) .

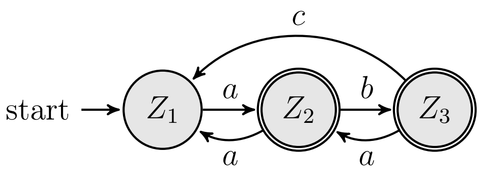
(b) Anwendung (12 Punkte)

Zeigen Sie mittels Pumping Lemma, dass die Sprache $L := \{a^{\lfloor i/2 \rfloor} b^{\lfloor i/3 \rfloor} (cc)^{\lceil i/5 \rceil} \mid i \geq 3\}$ nicht kontextfrei ist.

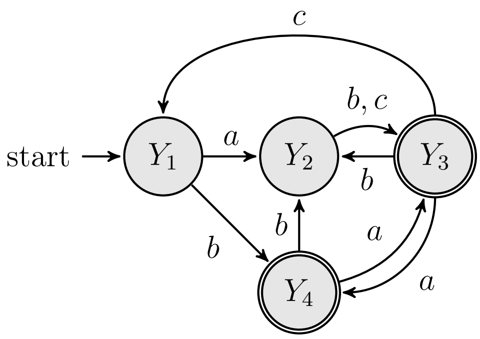
Aufgabe 4: Abgeschlossenheit regulärer Sprachen (12 Punkte)

Erstellen Sie – nach dem Vorgehen aus der Vorlesung! – aus den beiden gegebenen DEAs $\mathcal{A}_1$ und $\mathcal{A}_2$ einen DEA $\mathcal{B}$ mit $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$.

$\mathcal{A}_1$:



$\mathcal{A}_2$:



Aufgabe 5: Algorithmische Fragestellungen (16 Punkte)

(a) Graphen (8 Punkte)

Ein *einfacher* ungerichteter Graph besteht aus einer Menge von Knoten und einer Menge von Kanten. Eine Kante ist eine zwei-elementige Teilmenge der Knoten. Wieviele Kanten kann ein solcher Graph mit n Knoten maximal haben?

Gegeben ein endlicher Automat $\mathcal{A}$, der die Sprache L beschreibt. Sie möchten das Leerheitsproblem für L lösen. Dazu interpretieren Sie $\mathcal{A}$ als einen gerichteten Graphen G mit n Knoten und m Kanten.

Welchen Algorithmus benutzen Sie, um das Problem auf G zu lösen?

Was ist die benötigte Laufzeit in $\mathcal{O}$-Notation?

Welche Datenstruktur benutzt man typischerweise, um in diesem Algorithmus die Knoten zu betrachten?

☐ Stack ☐ Queue ☐ PriorityQueue (Heap) ☐ Binärer Suchbaum

(b) Dynamische Programmierung (8 Punkte)

Beschreiben Sie das generelle Konzept der *dynamischen Programmierung*:

Nennen Sie zwei Beispiele (z.B. aus Informatik D), in denen dynamische Programmierung klassischerweise zum Einsatz kommt:

-
-

Aufgabe 6: Kellerautomat **(12 Punkte)**

Gegeben die folgende kontextfreie Sprache L (als Grammatik in EBNF):

$$
\begin{aligned}
S &\to aA\{B\} \\
A &\to aAa \mid a \\
B &\to b[B]c
\end{aligned}
$$

Beschreiben Sie die Sprache L in Mengenschreibweise (also nach dem Muster $\{ab^i \mid i \geq 1\}$). Vereinfachen Sie soweit wie möglich!

Erstellen Sie einen Kellerautomaten, der genau L akzeptiert.

Aufgabe 7: Rechnende Turingmaschine (12 Punkte)

Gegeben eine unär kodierte Zahl $\alpha > 0$. Geben Sie eine Turingmachine an, die die folgende Funktion berechnet:

$$f(\alpha) := \begin{cases} \alpha/3 & \alpha \bmod 3 = 0 \\ \text{undef} & \text{sonst} \end{cases}$$

Aufgabe 8: Turing-Vollständigkeit **(14 Punkte)**

Betrachten Sie die folgende Programmiersprache MultFun:

Variablen in MultFun werden mit $\$i$, $i \geq 0$, bezeichnet. In jeder Variable $\$i \in \mathbb{Z}$ wird eine ganze Zahl gespeichert.

Die einzelnen Anweisungen in MultFun werden nacheinander ausgeführt und jeweils durch ein Komma voneinander getrennt. Wir haben die folgenden Anweisungen zur Verfügung:

Anweisung	Beschreibung
$!\,c \mapsto \$j$	kopiert den Wert der Konstante $c \in \mathbb{Z}$ nach $\$j$.
$!\,\$i \mapsto \$j$	kopiert den Wert von $\$i$ nach $\$j$
$+\,\$i \mapsto \$j$	erhöht (Addition) $\$j$ um $\$i$
$-\,\$i \mapsto \$j$	verkleinert (Subtraktion) $\$j$ um $\$i$
$*\,\$i \mapsto \$j$	speichert den Wert des Produkts $\$i * \$j$ in $\$j$
$?\,\$i\,\{\mathcal{A}\}$	führt die Anweisungen $\mathcal{A}$ aus, falls $\$i > 0$
$\{\mathcal{A}\}\,\$i$	führt die Anweisungen $\mathcal{A}$ aus (mind. 1-mal), solange $\$i > 0$

Beispiel: Das folgende MultFun-Programm berechnet $(12 - \$1 + 5) \cdot \$2$ und speichert den Wert in $3:

$$!12 \mapsto \$3, \quad -\$1 \mapsto \$3, \quad !5 \mapsto \$4, \quad +\$4 \mapsto \$3, \quad *\$2 \mapsto \$3$$

(a) Definition (4 Punkte)

Definieren Sie Turing-Vollständigkeit:

(b) Beweis der Turing-Vollständigkeit (8 Punkte)

Zeigen Sie, dass die Programmiersprache MULTFUN Turing-vollständig ist. (Beschreiben Sie vor dem eigentlichen Beweis auch kurz die allgemeine Idee des Vorgehens.)

(c) Turing-Äquivalenz (2 Punkte)

Was müssten Sie darüberhinaus zeigen, um zu beweisen, dass MULTFUN sogar *Turing-äquivalent* ist?

Aufgabe 9: Berechenbarkeit **(10 Punkte)**

Ein *Cosmonaut* ist eine Turingmaschine mit trinärem Alphabet $\Gamma = \{\square, \star, \bullet\}$ (Leere, Stern, Schwarzes Loch), bei der die folgenden drei Eigenschaften erfüllt sind:

(i) Das Speicherband ist initial nur mit $\square$-Symbolen gefüllt.

(ii) Der SL-Kopf ändert die Richtung nur, falls gerade das Symbol $\bullet$ gelesen wird.
Falls aktuell also kein $\bullet$ gelesen wird, darf der SL-Kopf nur stehenbleiben oder einen Schritt in die Richtung machen, in die er als letztes (ggf. vor dem Stehenbleiben) einen gemacht hat.

(iii) Sie hält an.

(a) Semi-Entscheidbarkeit (6 Punkte)

Zeigen Sie, dass es semi-entscheidbar ist, ob eine gegebene TM $\mathcal{T}$ ein Cosmonaut ist:

(b) Unentscheidbarkeit (4 Punkte)

Ein *Crazy Cosmonaut* ist ein Cosmonaut, der die größtmögliche Schrittanzahl erzielt.

Zeigen Sie, dass es unentscheidbar ist, ob eine gegebene TM $\mathcal{T}$ ein Crazy Cosmonaut ist:

Aufgabe 10: NP-Vollständigkeit (20 Punkte)

Ein Problem $\mathcal{X}$ ist NP-schwer, genau dann wenn...

Ein Problem $\mathcal{X}$ ist NP-vollständig, genau dann wenn...

Betrachten Sie das folgende Problem

Problem: PinPacking

Gegeben: Mehrere Nadeln (Pins) und Nadelkissen. Die Nadeln sind gegeben als Menge $\mathcal{P}$ von n 2-Tupeln, d.h. $\{(s_i, c_i)\}_{1 \leq i \leq n}$. Dabei gibt s_i die Größe einer Nadel (=wie viel Platz sie benötigt) und c_i die Anzahl der Nadeln dieser Größe an. Es gilt, dass $s_i \neq s_{i'}$ für $i \neq i'$. Die Nadelkissen sind gegeben als ℓ-elementige Liste $\mathcal{K} = \langle p_j \rangle_{1 \leq j \leq \ell}$; das j-te Nadelkissen bietet dabei p_j viel Platz.

Frage: Kann man mindestens eine Nadel jeder vorhandenen Größe auswählen, sodass all diese gewählten Nadeln auf die gegebenen Kissen gesteckt werden können?

Wir wollen im Folgenden zeigen, dass PinPacking NP-vollständig ist.

Definieren Sie, wie ein (im Weiteren nutzbarer) Zeuge für dieses Problem aufgebaut ist:

Wir benötigen im Beweis ein weiteres NP-vollständiges Problem. Welches? Definieren Sie es:

Beweisen Sie nun, dass PinPacking ***NP***-vollständig ist:

Beweisteil 1.

Beweisteil 2.

Viel Erfolg!