Klausur im WS 21/22, 10.02.2022

Theoretische Informatik für Angewandte Informatik

Prof. Dr. Barbara Staehle, HTWG Konstanz

Bearbeitungszeit: 60 Minuten

Hinweise:

- Falls Sie für die Aufgaben alle Punkte haben wollen, begründen Sie Ihre Antworten, bzw. stellen Sie den Lösungs- / Rechenweg nachvollziehbar dar.
- Lösen Sie sofern möglich, die Aufgaben auf dem Angabenblatt. Falls nicht genügend Platz vorhanden ist, nutzen Sie zusätzliches Papier.
- **Die Klausur enthält mehr Aufgaben, als Sie in der Bearbeitungszeit lösen können. Wählen Sie klug aus welche Aufgaben Sie lösen!** Sie müssen weder zum Bestehen noch für eine sehr gute Note alle Aufgaben korrekt bearbeiten. Zum Bestehen reichen ca. **50 Punkte**, eine sehr gute Note gibt es **ab ca. 70** Punkten.

Name: _____

Matrikelnummer: _____

Note: _____

Aufgabe	1	2	3	4	5	Σ
Punkte	20	29	29	17	25	120
Erreicht						

AUFGABE 1, 20 PUNKTE
WAHR ODER FALSCH?

Sind folgenden Aussagen wahr oder falsch? Begründen Sie Ihre Entscheidung (kurz).

Punktvergabe: w/f richtig: 1 Punkt; w/f richtig und Begründung sinnvoll: 2 Punkte

Aussage	wahr	falsch	kurze Begründung
(a) 'Transduktor' ist kein äquivalenter Name für „Turing-Maschine".	☐	☐	
(b) Bei allen Automatenmodellen (D/NEA, (D)PDA, (D)LBA,(N)TM) akzeptieren die nichtdeterministen und die deterministischen Varianten die gleiche Sprachklasse.	☐	☐	
(c) Alle Probleme, die in der Klasse NP enthalten sind, sind nicht lösbar.	☐	☐	
(d) Der Begriff „unentscheidbar" ist nur für Turing-Maschinen wichtig. Für state-of-the-art Hardware gibt es keine unentscheidbaren Probleme - man kann für jedes Problem einen Algorithmus finden.	☐	☐	
(e) Alle formalen Sprachen sind entscheidbar.	☐	☐	
(f) Mit dem Pumping-Lemma für reguläre Sprachen kann man nicht beweisen, dass eine Sprache regulär ist. Dies macht man z.B., indem man einen regulären Ausdruck angibt, der die Sprache erzeugt.	☐	☐	
(g) Für jede Grammatik G ist der Syntaxbaum der Ableitung für jedes aus G ableitbare Wort immer eindeutig.	☐	☐	
(h) Es gilt $P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXP \subseteq NEXP$	☐	☐	
(i) Kommen in einer logischen Aussage mehrere Existenz- und Allquantoren vor, kann man deren Reihenfolge beliebig vertauschen.	☐	☐	
(j) Alle Turing-Maschinen beenden jede Berechnung immer nach endlich vielen Schritten.	☐	☐	

AUFGABE 2, 29 PUNKTE
LOGIK, MENGEN UND FORMALE SPRACHEN

2.1 LOGIK

Folgendes sei gegeben:

- die Menge aller formalen Sprachen F
- die Menge aller (für die theoretische Informatik definierten) Grammatiken T
- die formale Sprache $L_3 \in F$ mit $L_3 = \{(ab)^n \mid n \in \mathbb{N}\}$ (Chomsky-Typ 3)
- die formale Sprache $L_1 \in F$ mit $L_1 = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ (Chomsky-Typ 1)
- Aussageformen $C(X, i)$: „X ist vom Chomsky-Typ i“ (für $X \in T \cup F, i \in \{0, 1, 2, 3\}$)
- Aussageform $E(L, G)$: „L wird von G erzeugt: $\mathscr{L}(G) = L$“ (für $G \in T, L \in F$)

Formulieren Sie die folgenden logischen Aussagen in Ihren eigenen Worten als deutsche Sätze **und geben Sie den Wahrheitswert der Aussage an**:

(a) (3 Punkte) $\neg(C(L_3, 3) \wedge \neg C(L_1, 1))$

(b) (3 Punkte) $\forall_{G \in T}\ \exists_{L \in F}\ E(L, G)$

(c) (3 Punkte) $\forall_{L \in F}\ \exists_{G \in T}\ E(L, G)$

Formulieren Sie folgende Sätze als zusammengesetzte logische Aussagen (bei Bedarf mit Quantoren) **und geben Sie den Wahrheitswert der Aussage an**.

(d) (3 Punkte) Es gibt keine formale Sprache, die nicht vom Typ 0 ist.

(e) (3 Punkte) Alle Sprachen vom Chomsky-Typ 1 werden von allen Grammatiken vom Chomsky-Typ 1 erzeugt.

2.2 CHOMSKY-HIERARCHIE

Es seinen die Grammatiken $G_i = (N, \Sigma_A, P_i, S) = (\{S, A\}, \{a, b\}, P_i, S)$ mit $i \in \{1, 2, 3, 4\}$ gegeben. Die Grammatiken sind also identisch, bis auf die Produktionsmengen, welche in der untenstehenden Tabelle angegeben sind.

(a) (10 Punkte) Kreuzen Sie für jede der Regelmenge an, von welchem Chomsky-Typ die dazugehörige Grammatik (maximal) ist. Wenn also eine Produktionsmenge die Grammatik zum Typ 0, 1 und 2 macht, dann wäre die Lösung „Typ 2“.
Geben Sie weiterhin (wenn möglich) **zwei verschiedene** Automatentypen an, welche die von der Grammatik erzeugte Sprache erkennen.

Produktionsmenge	Typ 0	Typ 1	Typ 2	Typ 3	ungültig	2 erkennende Automaten
$P_1 = \{S \to aA, S \to bS, S \to b, A \to \varepsilon\}$						
$P_2 = \{S \to Aa, S \to Sb, S \to b, A \to \varepsilon\}$						
$P_3 = \{S \to Ab, b \to \varepsilon, A \to Ab, A \to \varepsilon\}$						
$P_4 = \{S \to aA, aA \to SbS, bS \to aba, S \to \varepsilon\}$						

(b) (3 Punkte) Geben Sie für die Grammatik G_1 die Ableitung des Wortes bbb aus dem Startsymbol, sowie den dazugehörigen Syntaxbaum an.

(c) (1 Punkt) Geben Sie die von G_1 erzeugte Sprache, $\mathscr{L}(G_1)$ an.

AUFGABE 3, 29 PUNKTE
REGULÄRE SPRACHEN

3.1 Gegeben sei das Alphabet aller Kleinbuchstaben $\Sigma_I = \{a, b, \ldots, z\}$, sowie die Abkürzungen einiger fiktiver Bachelor-Studiengänge, welche **ain, aib, gin, gib, win, wib** lauten.

Die formale Sprache $L_I \subseteq \Sigma_I^*$ ist definiert als Menge aller möglichen Wörter, die als Teilwort mindestens einen der genannten Abkürzungen (ain, aib, gin, gib, win, wib) enthalten.

Beispiele: ain, zuvwinbcdaibjk, loggibbu, gehören zu L_I; aiai, wien, xyzgi aber nicht.

(a) (2 Punkte) Geben Sie den regulären Ausdruck r_I an, der L_I erzeugt.

(b) ($3\frac{1}{2}$ Punkte) Konstruieren Sie den NEA (nichtdeterministischen endlichen Automaten) N_I, der L_I akzeptiert. Achten Sie darauf, dass Ihr NEA **mindestens ein** nichtdeterministische Element enthält.

(c) ($5\frac{1}{2}$ Punkte) Konstruieren Sie den DEA (deterministischen endlichen Automaten) A_I, der L_I akzeptiert.

(d) (6 Punkte) Konstruieren Sie den DET (deterministischen endlichen Transduktor) T_I (egal ob Mealy- oder Moore-Automat) der eine beliebig lange Zeichenkette als Eingabe annimmt und als Ausgabe für jedes gelesene Zeichen eine 0 oder eine 1 ausgibt. T_I soll eine 1 schreiben, sobald als Teilwort einer der Strings **ain, aib, gin, gib, win, wib** gelesen wurde, ansonsten eine 0.
Beispiele: ain → 001, abginn → 000010, winginaib → 001001001, aixn → 0000.

(e) (4 Punkte) Geben Sie die **reguläre** Grammatik G_I ab, welche L_I erzeugt, für welche also $\mathscr{L}(G_I) = L_I$ gilt.

3.2 Wir betrachten das Alphabet $\Sigma_Z = \{3, 4, 5\}$. Geben Sie für die folgenden regulären Ausdrücke jeweils die formale Sprache an, welche diese erzeugen. Geben Sie für für die formalen Sprachen an, welcher reguläre Ausdruck diese erzeugt.

(a) (2 Punkte) $r_1 = 5[34][34]5$

(b) (2 Punkte) $r_2 = (34^*)^*$

(c) (2 Punkte) $L_4 = \{53^n4^m5 \mid n, m \in \mathbb{N}_0\}$

(d) (2 Punkte) $L_5 = \{345^n(34)^m \mid n, m \in \mathbb{N}\}$

AUFGABE 4, 17 PUNKTE
KONTEXTFREIE SPRACHEN

4.1 Wir betrachten das Alphabet $\Sigma_K = \{a, b, c\}$, sowie die Grammatik $G_K = (N, \Sigma_K, P, S)$ mit $N = \{S, T\}$ und der Produktionsmenge

$$P: \begin{array}{lcllcl} S & \rightarrow & TaSb & S & \rightarrow & ab \\ T & \rightarrow & cT & T & \rightarrow & \varepsilon \end{array}$$

(a) (2 Punkte) Welche Sprache $L_K = \mathscr{L}(G_K)$ wird von der Grammatik G_K erzeugt?

(b) (5 Punkte) Überführen Sie die Grammatik G_K in die Chomsky-Normalform.
Es reicht, wenn Sie die **Regelmenge** P' dieser äquivalenten Grammatik in CNF angeben.

(c) (4 Punkte) Konstruieren Sie den Kellerautomaten P_K (PDA oder DPDA), welcher L_K akzeptiert.

(d) (6 Punkte) Konstruieren Sie die Turing-Maschine T_K (TM oder NTM), welche L_K akzeptiert.

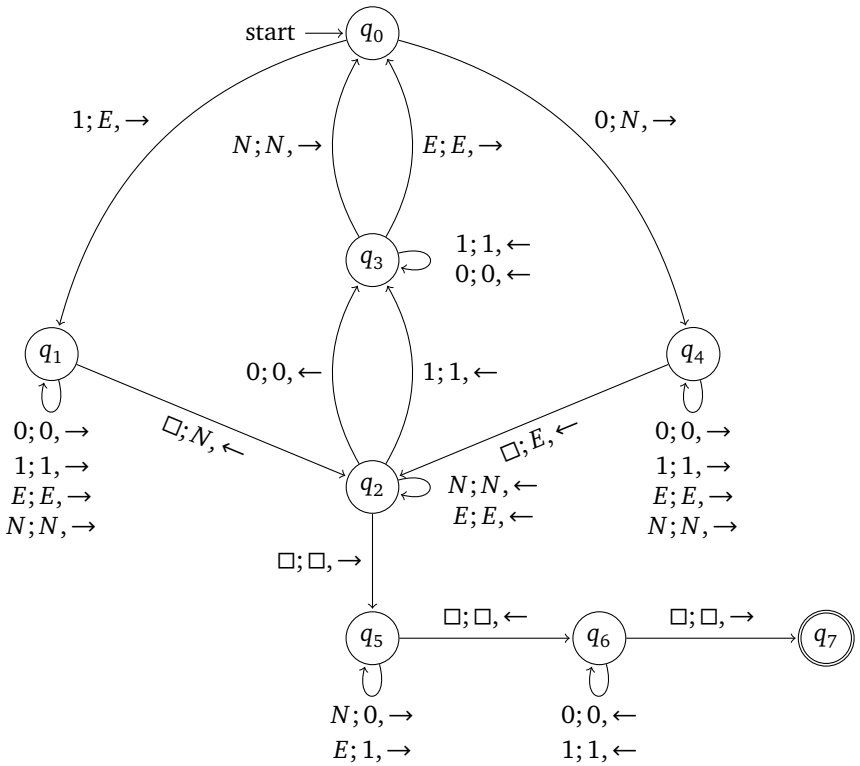

Abbildung 1: Erweitertes Zustandsübergangsdiagramm für T_x

AUFGABE 5, 25 PUNKTE
BERECHENBARKEIT, ENTSCHEIDBARKEIT & KOMPLEXITÄT

5.1 Wir betrachten das Alphabet $\Sigma_x = \{0, 1\}$ und die Funktion $f_x : \Sigma_x^* \rightarrow \Sigma_x^*$ welche von der Turing-Maschine $T_x = (Q, \Sigma_x, \Pi, \delta, q_0, F) = (\{q_0, q_1, \ldots, q_7\}, \{0, 1\}, \{0, 1, N, E, \square\}, \{q_5\}, \delta)$ mit δ gegeben durch Abbildung 1 berechnet wird.

(a) (7 Punkte) Bestimmen Sie für die Worte $\omega_1 = \mathbf{0}$ (3 Punkte) und $\omega_2 = \mathbf{11}$ (4 Punkte) jeweils alle Konfigurationen welche die TM T_x während der Verarbeitung der Worte durchläuft. Kürzen Sie sehr lange, uninteressante Berechnungsabschnitte durch „…“ bzw. „*“ ab!!

(b) (8 Punkte) Geben Sie für jedes der in der nebenstehenden Tabelle angegebenen Eingabewörter $\omega_i, i \in \{0, 1, \ldots, 7\}$ das von T_x berechnete Ergebnis $f_x(\omega_i)$ an.
Falls Sie der Meinung sind, dass T_x für ein Eingabewort ein undefiniertes Ergebnis liefert, verwenden Sie für das entsprechende Ergebnis das Symbol „$\perp$“.

(c) (2 Punkte) Beschreiben Sie die Funktion f_x, welche von der TM T_x berechnet wird. Konkret: was ist der Output von T_x für einen zulässigen Input?

ω_i	$f_x(\omega_i)$
$\omega_0 = \varepsilon$	
$\omega_1 = 0$	
$\omega_2 = 11$	
$\omega_3 = 00$	
$\omega_4 = 10$	
$\omega_5 = 01$	
$\omega_6 = 010$	
$\omega_7 = 011$	

5.2 (8 Punkte) Vervollständigen Sie den folgenden Lückentext. Die Länge des Feldes sagt wenig über die Länge des einzusetzenden Textes aus. Falls Sie eine Lücke leer lassen möchten, kennzeichnen Sie dies z.B. durch „–“. **Leere Lücken geben keine Punkte.**

Turing-Maschinen akzeptieren nicht nur Sprachen, sie berechnen auch Funktionen . Allerdings kann eine Turings-Maschine nur Funktionen berechnen, _____.
Dies hat sich _____ geändert, seit die Überlegenheit der Quantencomputer (Quantum Supremacy) bewiesen wurde: Quantencomputer können
_____ als herkömmliche Computer.

Für alle _____ Probleme oder formalen Sprachen kann die Zeit- und Raumkomplexität bestimmt werden. Vor allem die Zeitkomplexität ist wichtig, da nur Probleme aus der Klasse P _____. Probleme aus der Klasse NP, und vor allem die NP-vollständigen Probleme sind _____. Wenn Sie jedoch trotzdem eine effiziente Polyonomialzeitlösung für ein NP-vollstängiges Problem finden,
_____. Ein beispielhaftes NP-vollständiges Problem ist _____.