Klausur im WS 20/21, Januar 2020

Testklausur Theoretische Informatik Angewandte Informatik

Prof. Dr. Barbara Staehle, HTWG Konstanz

Bearbeitungszeit: maximal 30 Minuten, nur Techniktest!

Hinweise:

- $\mathbb{N} = \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \ldots\}$
- 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 (Vorder- und Rückseite). Falls nicht genügend Platz vorhanden ist, nutzen Sie zusätzliches Papier.
- Beschriften Sie das Kopfblatt der Angabe, sowie jedes zusätzlich genutzte Blatt mit Ihrem Namen und Ihrer Matrikelnummer.
- 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. 85 Punkten.

Name: ____________________________

Matrikelnummer: ________________

Note: ________________

Aufgabe	1	2	3	4	Σ
erreichbare Punkte	20	13	41	26	100
erreichte Punkte					

Aufgabe 1 Wahr oder Falsch?, 20 Punkte

Entscheiden Sie, ob die folgenden Aussagen wahr oder falsch sind. Begründen Sie Ihre Entscheidung (kurz).
Punktvergabe:

- richtiges Kreuz & falsche bzw. fehlende Begründung: +1 Punkt
- richtiges Kreuz & korrekte Begründung: +2 Punkte

Aussage	wahr	falsch	kurze Begründung
(1) $L_a = \{ain\}$ ist eine formale Sprache vom Chomsky-Typ 3 über dem Alphabet $\Sigma_a = \{a, b, \ldots, z\}$.	☐	☐	
(2) Ein endlicher deterministischer Automat kann nicht erkennen, ob ein String zu jeder öffnenden Klammer auch eine schließende enthält.	☐	☐	
(3) Jede Turing-berechenbare Funktion ist total.	☐	☐	
(4) Alle Chomsky Typ-2 Sprachen sind entscheidbar	☐	☐	
(5) 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.	☐	☐	
(6) Es gilt: $P \subset NP$.	☐	☐	
(7) Bei der Ableitung eines Wortes aus einer Grammatik wird in jedem Schritt mindestens ein Nonterminalsymbol durch kein oder beliebig viele andere Symbole ersetzt.	☐	☐	
(8) Das Halteproblem ist NP-vollständig.	☐	☐	
(9) „Transduktor“ ist kein äquivalenter Name für „Turing-Maschine“.	☐	☐	
(10) Für jede Grammatik G ist der Syntaxbaum der Ableitung für jedes aus G ableitbare Wort immer eindeutig.	☐	☐	

Aufgabe 2 Logik und reguläre Ausdrücke

Teilaufgabe 2.1 7 Punkte

Sei $N(x, y)$ die Aussageform „x ist effizienter als y" für x und y jeweils Programmiersprachen aus der Menge P aller existierenden Programmiersprachen. Übersetzen Sie die folgenden logischen Aussagen in deutsche Sätze:

a) (1.5 Punkte) $\forall_{x \in P \setminus \{\text{Java}\}} \; N(\text{Java}, x)$

b) (1.5 Punkte) $\exists_{x \in P \setminus \{\text{Scala}\}} \; N(x, \text{Scala})$

c) (2 Punkte) $\forall_{x \in P} \; \exists_{y \in P \setminus \{x\}} \; N(x, y)$

d) (2 Punkte) $\exists_{y \in P} \; \forall_{x \in P \setminus \{y\}} \; N(x, y)$

Teilaufgabe 2.2 6 Punkte

$\Sigma_A = \{x, y, z\}$. Geben Sie für die folgenden formalen Sprachen über Σ_A^* jeweils die regulären Ausdrücke an, welche diese Sprachen erzeugen. Falls dies nicht möglich ist, kennzeichnen Sie dies bitte entsprechend.

a) (1.5 Punkte) $L_1 = \{x^n y^m z^q \mid n, m, q \in \mathbb{N}_0\}$

b) (1.5 Punkte) $L_2 = \{(xyz)^n \mid n \in \mathbb{N}\}$

c) (1.5 Punkte) $L_3 = \{x^n y^n z^n \mid n \in \mathbb{N}\}$

d) (1.5 Punkte) $L_4 = \{x^n \mid n \in \mathbb{N}\} \cup \{y^n \mid n \in \mathbb{N}\} \cup \{z^n \mid n \in \mathbb{N}\}$

Aufgabe 3 Formale Sprachen und alles was dazu gehört

Teilaufgabe 3.1 21 Punkte

Gegeben sei das Alphabet $\Sigma_C = \{a, b\}$, sowie die Sprache $L_C \subseteq \Sigma_C^*$, mit $L_C = \mathscr{L}(G_C)$. Die L_C erzeugende Grammatik G_C ist gegeben durch $G_C = (N_C, \Sigma_C, P_C, S) = (\{S\}, \{a, b\}, P_C, S)$ mit

$$P_C = \{S \rightarrow aSb, S \rightarrow bb\}.$$

a) (3 Punkte) Geben Sie für G_C eine Grammatik G'_C in Chomsky-Normalform mit $\mathscr{L}(G_C) = \mathscr{L}(G'_C)$ an.

b) (5 Punkte) **Nicht klausurrelevant im WS20/21!!!:** Bestimmen Sie mit Hilfe des CYK-Algorithmus, ob die Wörter $abbb$ und $aabbb$ zur Sprache $L_C = \mathscr{L}(G_C)$ gehören.

c) (1 Punkt) Geben Sie $L_C = \mathscr{L}(G_C)$ an.

d) (5 Punkte) Geben Sie einen Kellerautomaten (PDA oder DPDA) an, welcher L_C akzeptiert.

e) (7 Punkte) Geben Sie eine Turing-Maschine (TM oder NTM) an, welche L_C akzeptiert.

Teilaufgabe 3.2 11 Punkte

Wir betrachten das Alphabet $\Sigma_A = \{x, y, z\}$ und die Sprache $L_A \subseteq \Sigma_A^*$, die durch den regulären Ausdruck

$$r_A = [xyz]^* xyz$$

erzeugt wird:

$$L_A = \mathscr{L}(r_A) = \{xyz, xxyz, yxyz, zxyz, xxzyxyz, xyzxyzzzzzxyz, \dots\}.$$

a) (3 Punkte) Geben Sie einen nichtdeterministischen endlichen Automaten (NEA) an, welcher L_A akzeptiert. Achten Sie darauf, dass Ihr Automat mindestens ein nichtdeterministisches Element enthält.

b) (5 Punkte) Geben Sie einen deterministischen endlichen Automaten (DEA) an, welcher L_A akzeptiert.

c) (3 Punkte) Geben Sie eine **reguläre** Grammatik an, welche L_A erzeugt.

Teilaufgabe 3.3 9 Punkte

Fertigen Sie ein Mengendiagramm an, das sieben ineinander geschachtelte Mengen darstellt. Beschriften Sie es dann so, dass dieses die Ihnen bekannten Inklusionsbeziehungen zwischen den unten stehenden Sprachklassen und Sprachen darstellt. Verwenden Sie für Ihre Beschriftung die in den Klammern angegebenen Abkürzungen.

a) (7 Punkte) Die Klassen der

- Chomsky Typ0-, Typ1-, Typ2-, Typ3-Sprachen, $\mathscr{L}_0, \mathscr{L}_1, \mathscr{L}_2, \mathscr{L}_3$
- von den jeweiligen Automatentypen erkannten Sprachen
  $\mathscr{L}(DEA), \mathscr{L}(NEA), \mathscr{L}(DPDA), \mathscr{L}(PDA), \mathscr{L}(LBA), \mathscr{L}(TM), \mathscr{L}(NTM)$
- entscheidbaren Sprachen (E)
- semi-entscheidbaren Sprachen (SE)
- formalen Sprachen (F)

b) (2 Punkte) Platzieren Sie die Sprachen L_C und L_A (siehe Aufgaben 3.1 und 3.2) in einer korrekten, **möglichst weit innen liegenden** Ellipse.

Hinweis: Manche Ellipsen müssen mehrfach beschriftet werden.

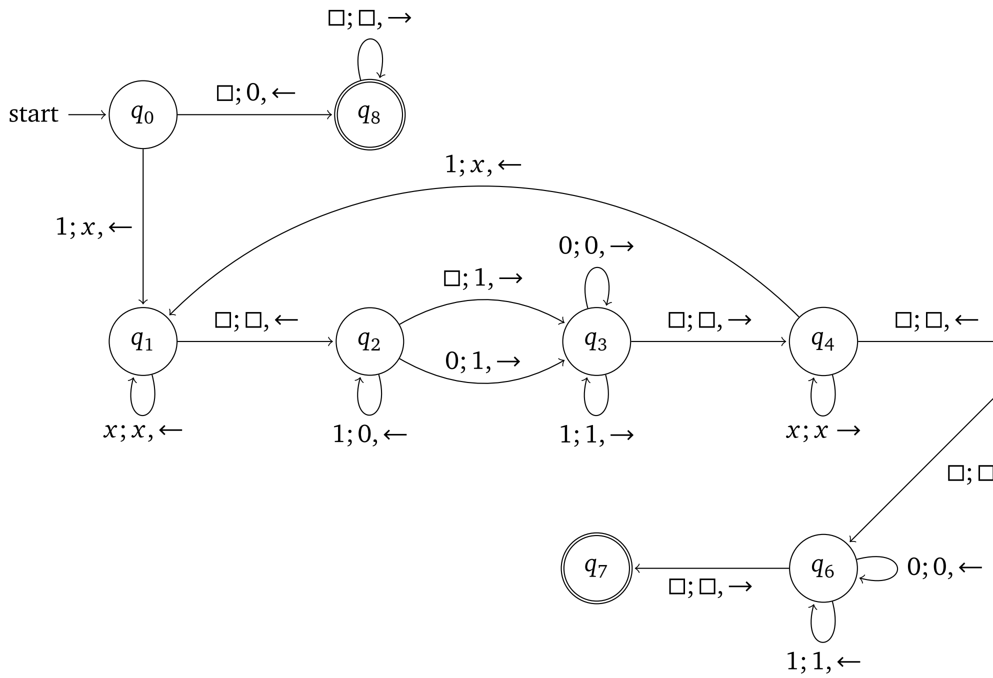

Abbildung 1: Erweitertes Zustandsübergangsdiagramm für T_x

AUFGABE 4 TURING-MASCHINEN UND ALLES WAS DAZU GEHÖRT

TEILAUFGABE 4.1 13 PUNKTE

Wir betrachten das Alphabet $\Sigma_x = \{0, 1\}$ und die Funktion $f_x : \Sigma_x^* \rightarrow \Sigma_x^*$ welche von der Turing-Maschine $T_x = (Q, \Sigma_x, \Pi, \delta, q_0, F) = (\{q_0, q_1, \ldots, q_8\}, \{0, 1\}, \{0, 1, x, \Box\}, \{q_7, q_8\}, \delta)$ mit δ gegeben durch Abbildung 4.1 berechnet wird.

a) Bestimmen Sie für die Worte $\omega_1 = 1$ (3 Punkte) und $\omega_2 = 11$ (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.
- das Ergebnis der Berechnung von f_x für diese Worte, $f_x(\omega_1)$ und $f_x(\omega_2)$.

b) (8 Punkte) Geben Sie für jedes der in der nebenstehenden Tabelle angegebenen Eingabewörter ω das von T_x berechnete Ergebnis $f_x(\omega)$ an.

Falls Sie der Meinung sind, dass T_x für ein Eingabewort ein undefiniertes Ergebnis liefert, verwenden Sie für das entsprechende Ergebnis das Symbol „$\perp$".

Hinweis: Sie müssen **keine** durchlaufenen Konfigurationen oder sonstige Begründungen angeben!

ω_i	$f_x(\omega_i)$
$\omega_3 = \varepsilon$	
$\omega_4 = 0$	
$\omega_5 = 110$	
$\omega_6 = 111$	
$\omega_7 = 1111$	
$\omega_8 = 1101$	
$\omega_9 = 11111$	
$\omega_{10} = 111111$	

c) (3 Punkte) Beschreiben Sie die Funktion f_x, welche von der TM T_x berechnet wird. Konkret:

- welche Eingabwörter werden akzeptiert?
- was ist der Output von T_x für ein korrektes Eingabewort?

d) (2 Punkte) Handelt es sich bei f_x um eine partielle oder eine totale Funktion? Begründen Sie Ihre Meinung.

TEILAUFGABE 4.2 13 PUNKTE

Vervollständigen Sie den folgenden Lückentext, in dem Sie **für jede Lücke kein, ein oder mehrere passende Worte finde**. 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 „A: –“. **Wenn Sie auf Ihrer Lösung nichts für einen Buchstaben angeben, bekommen Sie für diesen auch keine Punkte.**

Turing-Maschinen sind nur ein theoretisches Konstrukt, können aber alle Funktionen berechnen, die auch ein ____**A**____ berechnen kann. Dies hat sich ____**B**____ geändert, seit die Überlegenheit der Quantencomputer (Quantum Supremacy) bewiesen wurde: Quantencomputer können ____**C**____ als herkömmliche Computer und damit Turing-Maschinen.

Die Sprache L_x (aus Aufgabe 4.1) ist ein einfaches Beispiel einer ____**D**____ entscheidbaren Sprache. Weiterhin kann man auch schnell die jeweils Zeit- und Raumkomplexitätsklassen angeben, in denen L_x liegt: ____**E**____ und ____**F**____. L_x ist weiterhin auch ____**G**____ rekursiv und ____**H**____ rekursiv aufzählbar.

Für alle ____**I**____ Probleme oder formalen Sprachen, kann die Zeit- und Raumkomplexität bestimmt werden. Vor allem die Zeitkomplexität ist wichtig, die da nur Probleme aus der Klasse ____**J**____ für große Instanzen effizient lösbar sind. Probleme aus der Klasse NP und vor allem die NP-vollständigen Probleme sind ____**K**____. Wenn Sie jedoch trotzdem eine Polyonomialzeitlösung für ein NP-vollstängiges Problem finden, ____**L**____. Ein beispielhaftes NP-vollständiges Problem ist ____**M**____.