Klausur im SS 21, 06.07.2021

Theoretische Informatik
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 (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.
- Sie müssen weder zum Bestehen noch für eine sehr gute Note alle Aufgaben korrekt bearbeiten. Zum Bestehen reichen ca. 45 Punkte, eine sehr gute Note gibt es ab ca. 80 Punkten.

Name: ______________________________

Matrikelnummer: _________________

Note: _________________

Aufgabe	1	2	3	4	Σ
erreichbare Punkte	20	24	36	30	110
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: w/f richtig: 1 Punkt; w/f richtig und Begründung sinnvoll: 2 Punkte

Aussage	wahr	falsch	kurze Begründung
(a) $\mathbb{N} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$	☐	☐	
(b) Es gibt keine formale Sprache, die nicht vom Chomsky-Typ 0, 1, 2 oder 3 ist.	☐	☐	
(c) Ein DEA kann nicht erkennen, ob ein String zu jeder öffnenden Klammer auch eine schließende enthält.	☐	☐	
(d) Weil ein Quantencomputer in jedem Schritt unendlich viele Möglichkeiten parallel betrachten kann, löst er jedes Problem schneller als ein herkömmlicher Computer.	☐	☐	
(e) $\{a, b\} \times \{1, 2\} \neq \{1, 2\} \times \{a, b\}$	☐	☐	
(f) Ein Optimierungsproblem, wie z.B. die optimale Beladung eines LKWs mit Gütern kann rein theoretisch zwar exakt lösbar sein, aber rein praktisch so viel Zeit in Anspruch nehmen, dass es nur approximativ gelöst wird.	☐	☐	
(g) Wird aus einer Grammatik mit X verschiedenen Nonterminalsymbolen ein Wort der Länge $l > X$ abgeleitet, muss mindestens ein Nonterminal in der Ableitung mehrfach vorkommen.	☐	☐	
(i) Zukünftige Compiler werden zuverlässig für jedes Programm erkennen können, ob dieses in eine Endlosschleife läuft.	☐	☐	
(j) $L_a = \{ain, win\}$ ist eine formale Sprache vom Chomsky-Typ 3 über dem Alphabet $\Sigma_a = \{a, b, \ldots, z\}$.	☐	☐	
(k) Die Lösung eines Problems aus der Komplexitätsklasse NP kann man in Polynomialzeit verfizieren. Allerdings benötigt das Finden dieser Lösung meistens exponentielle Zeit.	☐	☐	

Aufgabe 2 Logik und reguläre Ausdrücke

Teilaufgabe 2.1 9 Punkte

Folgendes sei gegeben:

- m: beliebiger Mensch aus der Menge aller AIN-Student:innen M
- p: beliebige Programmiersprache aus der Menge aller Programmiersprachen P
- Aussageform $K(m, p)$: „m kann in p programmieren“
- Aussageform $L(m, p)$: „m liebt p“ (kurz für „m liebt es in p zu programmieren“)
- Alice, Bob und Charlie: AIN-Student:innen, also Menschen aus M.

Formulieren Sie die folgenden logischen Aussagen in Ihren eigenen Worten als deutsche Sätze:

a) (1.5 Punkte) $K(\text{Alice, Java}) \Rightarrow L(\text{Alice, Java})$

b) (1.5 Punkte) $\left(\forall_{m\in\{\text{Alice,Bob}\}} \neg L(m, \text{C++})\right) \wedge L(\text{Charlie}, \text{C++})$

c) (2 Punkte) $\forall_{m\in M} \exists_{p\in P} K(m, p)$

Formulieren Sie folgende Sätze als zusammengesetzte logische Aussagen (mit Quantoren bei Bedarf).

d) (2 Punkte) Alle AIN-Student:innen können entweder in Java oder in C++ programmieren, oder in beiden Sprachen.

e) (2 Punkte) Für alle AIN-Student:innen existiert mindestens eine Programmiersprache, die sie weder können noch lieben.

TEILAUFGABE 2.2 15 PUNKTE

Wir betrachten das Alphabet der 95 druckbaren ASCII-Zeichen Σ_A (Klein-, Großbuchstaben (ohne Umlaute), Ziffern, Sonderzeichen, Leerzeichen).

Geben Sie für die folgenden regulären Ausdrücke über Σ_A jeweils die formale Sprache an, welche diese erzeugen. Geben Sie die Sprache mathematisch genau an, oder beschreiben Sie diese mit Ihren eigenen Worten.

a) (2.5 Punkte) $r_a = 0|[-+]?[1-9][0-9]*$

b) (2.5 Punkte) $r_b =$ ([01][0–9]|2[0–3]):[0–5][0–9]:[0–5][0–9]

c) (3 Punkte) $r_c =$
(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])

Geben Sie den regulären Ausdruck an, welcher die folgenden formalen Sprachen erzeugt. Falls es nicht möglich ist, einen erzeugenden regulären Ausdruck anzugeben, kennzeichnen Sie dies bitte entsprechend.

d) (2 Punkte) $L_d = \{x^n y^m z^q \mid n, m, q \in \mathbb{N}_0\}$

e) (2 Punkte) $L_e = \{x^n y^n z^n \mid n \in \mathbb{N}\}$

f) (3 Punkte) $L_f =$ { Menge aller Passwörter, die ein bis beliebig viele Zeichen enthalten. Die Zeichen dürfen Groß- oder Kleinbuchstaben oder Ziffern sein und es muss mindestens eine Zahl enthalten sein, aber keine Sonderzeichen } = { 1, pass4word5, 01234, q5Txc67, 0815LanGesHochGeheimesPasswort999, ...}

AUFGABE 3 FORMALE SPRACHEN, GRAMMATIKEN UND AKZEPTIERENDE AUTOMATEN

Wir betrachten das Alphabet $\Sigma_1 = \{a, b, x\}$, sowie folgende formale Sprachen über Σ_1^*:

- $L_1 = \{\omega\omega^R \mid \omega \in \Sigma_1^*\} =$ { die Menge aller Palindrome gerader Länge über Σ_1^* }
  $= \{\varepsilon, aa, bb, xx, abba, bbabbabb, xbaabx, \ldots\}$
- $L_2 = \{\omega xx\omega^R \mid \omega \in \{a, b\}^*\} =$ { die Menge aller Palindrome gerader Länge über $\{a, b\}^*$ getrennt vom String xx
  $= \{xx, axxa, bxxb, abxxba, bbabxxbabb, \ldots\}$
- $L_3 = \{\omega\omega \mid \omega \in \Sigma_1^*\} =$ { die Menge aller doppelten Worte über Σ_1^*}
  $= \{\varepsilon, aa, bb, xx, abxabx, bbabbbab, bxxbxx, \ldots\}$
- $L_4 = \{ab\omega ab \mid \omega \in \Sigma_1^*\} =$ { die Menge aller Wörter über Σ_1^* die mit ab beginnen und mit ab enden }
  $= \{abab, abaab, abbab, abxab, abbbbab, abxaxbab, \ldots\}$

TEILAUFGABE 3.1 8 PUNKTE

Markieren Sie in der folgenden Tabelle die Kästchen, wodurch die Sprachen L_1, L_2, L_3 und L_4 jeweils erzeugt bzw. akzeptiert werden können mit einem „x". Markieren Sie die Kästchen mit einem „–", welche eine nicht mögliche Kombination zeigen. **Leere Kästchen geben keine Punkte!**
Abkürzungen:

- Typ$y$G: Grammatik vom Chomsky-Typ y, $y \in \{0, 1, 2, 3\}$
- restliche Abkürzungen: siehe Skript

Sprache	Typ0G	Typ1G	Typ2G	Typ3G	DEA	NEA	DPDA	PDA	LBA	(N)TM
L_1										
L_2										
L_3										
L_4										

TEILAUFGABE 3.2 5 PUNKTE

Geben Sie für die folgenden Sprachen eine erzeugende Grammatik an:

$$L_1 = \{\omega\omega^R \mid \omega \in \Sigma_1^*\} = \{ \text{ die Menge aller Palindrome gerader Länge über } \Sigma_1^* \}$$

$$L_4 = \{ab\omega ab \mid \omega \in \Sigma_1^*\} = \{ \text{ die Menge aller Wörter über } \Sigma_1^* \text{ die mit } ab \text{ beginnen und mit } ab \text{ enden } \}$$

Die erzeugende Grammatik soll **auf jeden Fall kontextfrei, wenn möglich sogar regulär** sein.

a) (2 Punkte) Grammatik G_1, welche L_1 erzeugt:

b) (3 Punkte) Grammatik G_4, welche L_4 erzeugt:

Teilaufgabe 3.3 5 Punkte

Geben Sie einen Kellerautomaten **Kellerautomaten (PDA oder DPDA)** an, welcher die Sprache

$L_2 = \{\omega x x \omega^R \mid \omega \in \{a, b\}^*\} =$ { die Menge aller Palindrome gerader Länge über $\{a, b\}^*$ getrennt vom String xx }

akzeptiert.

Teilaufgabe 3.4 7 Punkte

Geben Sie für die Sprache

$L_4 = \{ab\omega ab \mid \omega \in \Sigma_1^*\} =$ { die Menge aller Wörter über Σ_1^* die mit ab beginnen und mit ab enden }

an

a) (3 Punkte) einen akzeptierenden nichtdeterministischen endlichen Automaten (NEA).
Achten Sie darauf, dass Ihr NEA mindestens ein nichtdeterministisches Element enthält.

b) (4 Punkte) einen akzeptierenden deterministischen endlichen Automaten (DEA).

TEILAUFGABE 3.5   11 PUNKTE

Betrachtet wird immer noch $\Sigma_1 = \{a, b\}$. Zusätzlich sei nun die Grammatik $G_5 = (N, \Sigma_1, P, S)$ mit $N = \{S\}$ gegeben mit den Regeln

$$P: \begin{array}{lllllll} S & \rightarrow & aSb & \quad & S & \rightarrow & SS \\ S & \rightarrow & bSa & \quad & S & \rightarrow & \varepsilon \end{array}$$

a) Nutzen Sie die Grammatik G_5, um für das Wort $\omega_{50} = abbbaa$

   1) (3 Punkte) eine Ableitung aus dem Startsymbol S
   2) (2 Punkte) den entsprechenden Syntaxbaum

   anzugeben.

b) (2 Punkte) Welche Sprache L_5 wird von der Grammatik G_5 erzeugt?

c) (4 Punkte) Überführen Sie die Grammatik G_5 in die Chomsky-Normalform.
   Es reicht, wenn Sie die **Regelmenge** P' dieser äquivalenten Grammatik in CNF angeben.

AUFGABE 4 BERECHENBARKEIT, ENTSCHEIDBARKEIT & KOMPLEXITÄT

TEILAUFGABE 4.1 9 PUNKTE

Gegeben seien das Alphabet $\Sigma_2 = \{0, 1\}$ und die Sprache $L_6 \subseteq \Sigma_2^*$ mit

$$\begin{aligned} L_6 &= \{\omega \in \Sigma_2 \mid |\omega|_0 = 2|\omega|_1\} = \{\omega \in \Sigma_2^* \mid \omega \text{ enthält genau doppelt so viele Nullen wie Einsen }\} \\ &= \{\varepsilon, 001, 100, 010, 000110, 010010, 010010, 110000, \ldots, 100101001000, \ldots\}. \end{aligned}$$

a) (2 Punkte) Kreisen Sie die/das für L_6 zutreffenden Adjektiv(e) ein. Begründen Sie Ihre Meinung!

entscheidbar   semi-entscheidbar   unentscheidbar

b) (3 Punkte) Beschreiben Sie **knapp!** die Arbeitsweise einer Turing-Maschine, welche L_6 akzeptiert.

c) (4 Punkte) Kreisen Sie jeweils **die kleinste** (Raum- bzw. Zeit-)Komplexitätsklasse ein, der L_6 angehört und begründen Sie Ihre Meinung (kurz)!

- P
- TIME($O(n^2)$)
- TIME($O(1)$)
- NP
- SPACE($O(1)$)
- PSPACE
- NPSPACE
- SPACE($O(n^2)$)

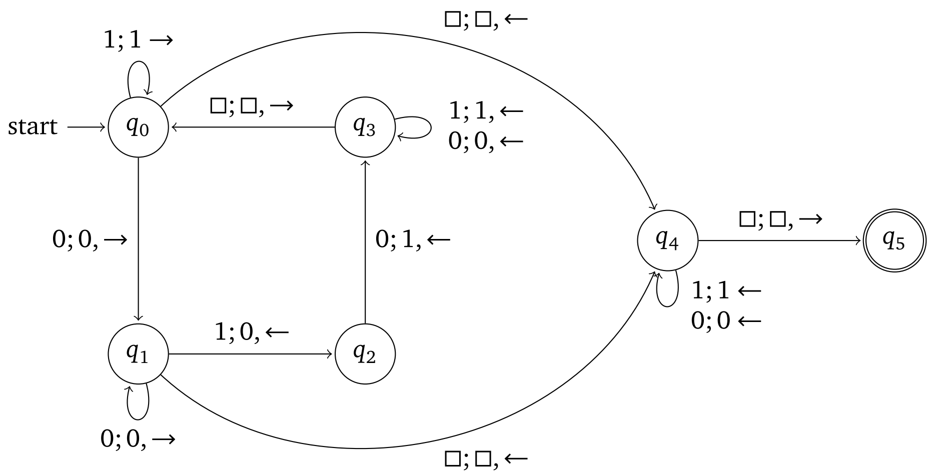

Abbildung 1: Erweitertes Zustandsübergangsdiagramm für T_x

TEILAUFGABE 4.2 21 PUNKTE

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

a) Bestimmen Sie für die Worte $\omega_1 = 0$ (2 Punkte) und $\omega_2 = 01$ (3 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) (10 Punkte) Geben Sie für jedes der in der nebenstehenden Tabelle angegebenen Eingabewörter $\omega_i, i \in \{0, 1, \ldots, 9\}$ 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 „$\perp$“.

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

ω_i	$f_x(\omega_i)$
$\omega_0 = \varepsilon$	
$\omega_1 = 0$	
$\omega_2 = 01$	
$\omega_3 = 11$	
$\omega_4 = 10$	
$\omega_5 = 00$	
$\omega_6 = 010$	
$\omega_7 = 011$	
$\omega_8 = 001$	
$\omega_9 = 101$	

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?

Hinweis: Aufgabe d) siehe nächste Seite!

d) (4 Punkte) Wenn T_x die gleiche Funktion f_x für Eingaben über dem Eingabealphabet $\{0, 1, \ldots, 9\}$ ausführen müsste, wie würden sich im Vergleich zur Variante mit dem Eingabealphabet $\{0, 1\}$ die Anzahl der benötigten Zustände und der benötigten Arbeitsschritte (für eine Eingabe der gleichen Länge) verändern? Kreuzen Sie in der unten stehenden Tabelle die zutreffenden Adjektive ein und begründen Sie Ihre Meinung!

Zustände	mehr	weniger	gleich viele
Arbeitsschritte	mehr	weniger	gleich viele