Klausur im WS 20/21, 01.02.2021

Theoretische Informatik
Angewandte Informatik

Prof. Dr. Barbara Staehle, HTWG Konstanz

Bearbeitungszeit: 60 Minuten

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 die Aufgaben auf einem Extra-Blatt Papier und beschriften Sie **nicht** die Angabe!
- Sie müssen weder zum Bestehen noch für eine sehr gute Note alle Aufgaben korrekt bearbeiten. Zum Bestehen reichen ca. 55 Punkte, eine sehr gute Note gibt es ab ca. 90 Punkten.

Name: ______________________

Matrikelnummer: ____________

Note: ____________

Aufgabe	1	2	3	4	Σ
erreichbare Punkte	20	19	53	33	125
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) Alle Probleme, die in der Klasse NP enthalten sind, sind nicht lösbar.	☐	☐	
(b) Eine Typ 1 Sprache wird von beschränkten nichtdeterministischen Turing-Maschinen und von beschränkten deterministischen Turing-Maschinen akzeptiert.	☐	☐	
(c) $\emptyset$ ist eine formale Sprache über dem Alphabet $\Sigma = \{0, 1\}$	☐	☐	
(d) Wenn man von einer beliebigen Menge das Komplement bildet und dann den Durchschnitt von diesem mit der Ausgangsmenge bildet, ergibt sich die leere Menge.	☐	☐	
(e) Alle formalen Sprachen sind semi-entscheidbar.	☐	☐	
(f) Für ein Problem der Größe n benötigt Algorithmus 1 die Zeit $t_1(n) \in O(n^2)$, Algorithmus 2 benötigt die Zeit $t_2(n) \in O(\log n)$. Für große n ist Algorithmus 2 daher immer schneller.	☐	☐	
(g) Die logische Aussage $x \Leftrightarrow y$ ist nur wahr, wenn x und y wahr sind.	☐	☐	
(i) Alle Sprachen die von einem regulären Ausdruck erzeugt werden können, werden auch auch von einer Turing-Maschine akzeptiert.	☐	☐	
(j) Eine clevere Informatikerin oder ein cleverer Informatiker wird in naher Zukunft einen Algorithmus für das Traveling Salesperson Problem finden, der dieses exakt in Zeit $O(n \log n)$ löst.	☐	☐	
(k) Alle Turing-Maschinen beenden jede Berechnung immer nach endlich vielen Schritten.	☐	☐	

AUFGABE 2 LOGIK UND KOMPLEXITÄTSTHEORIE

Folgendes sei gegeben:

- die Menge der Probleme $B = \{$ Halteproblem, Traveling Salesperson Problem, Rucksackproblem, Primzahlen-Problem, Pfadexistenz-Problem, Fleißiger Biber Problem $\}$
- die Menge aller **polynomiellen** Komplexitätsklassen $K = \{$ P, NP, PSPACE, NSPACE $\}$
- Aussageform $U(p)$: „p ist unentscheidbar" (für $p \in B$)
- Aussageform $L(p, k)$: „p liegt in Komplexitätsklasse k" (für $p \in B, k \in K$)
- Aussageform $T(k_1, k_2)$: „k_1 ist eine echte Teilmenge von k_2: $k_1 \subset k_2$" (für $k_1, k_2 \in K$)

TEILAUFGABE 2.1 12 PUNKTE

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

a) (3 Punkte) $\neg(U(\text{Halteproblem} \wedge U(\text{Rucksackproblem}))$

b) (3 Punkte) $\neg L(\text{Traveling Salesperson Problem}, \text{NP})$

c) (3 Punkte) $T(\text{PSPACE}, \text{NPSPACE}) \Leftrightarrow T(\text{NP}, \text{P})$

d) (3 Punkte) $\forall_{p \in B}\ \exists_{k \in K}\ L(p, k)$

TEILAUFGABE 2.2 9 PUNKTE

Erstellen Sie ein Venn-Diagramm, das

- die Mengen aller Entscheidungsprobleme A
- die Menge aller entscheidbaren Probleme E
- die Menge aller semi-entscheibaren Problem S
- sowie alle Komplexitätsklassen aus $K = \{$ P, NP, PSPACE, NSPACE $\}$

enthält. Achten Sie darauf, dass die Inklusionsbeziehungen korrekt dargestellt sind, die Größenverhältnisse sind irrelevant!
Zeichnen Sie in Ihr Diagramm weiterhin **zwei beliebige Probleme aus B** ein.

Aufgabe 3 Formale Sprachen, Grammatiken und akzeptierende Automaten

Wir betrachten das Alphabet $\Sigma_1 = \{0,1\}$, sowie folgende formale Sprachen über Σ_1^*:

- $L_1 = \{0^{2^n} \mid n \in \mathbb{N}_0\} = \{0, 00, 0000, 00000000, \ldots\}$
- $L_2 = \{0^n 1^{2n} \mid n \in \mathbb{N}\} = \{011, 001111, 000111111, \ldots\}$
- $L_3 = \{(0^n 1^m)^k \mid n, m, k \in \mathbb{N}\} = \{01, 0001, 0111101010111, \ldots\}$
- $L_4 = \{0\omega\omega^R 0 \mid \omega \in \Sigma_1^* \wedge \omega^R \text{ ist } \omega \text{ rückwärts gelesen }\} = \{00, 0110, 010110011010, \ldots\}$

Teilaufgabe 3.1 12 Punkte

Lösen Sie die folgenden Aufgaben gerne mit Hilfe einer Tabelle oder einer Stichpunktliste, in der Sie für eine Sprache, alles geforderte in einer Zeile notieren!
Geben Sie für jede der Sprachen L_1, L_2, L_3, L_4 an:

a) (4 Punkte) ihren (numerisch maximalen) Chomsky-Typ.

b) (4 Punkte) den schwächsten Automatentyp, der die Sprache akzeptiert.
Verdeutlichung: z.B. sind endliche Automaten schwächer als Kellerautomaten.

c) (4 Punkte): die kleinste Zeit-Komplexitätsklasse in der die Sprache enthalten ist.
Wählen Sie hierfür unter TIME(O(1)), TIME(O(n)), P, NP

Teilaufgabe 3.2 6 Punkte

Geben Sie für die Sprachen L_2, L_3 und L_4, jeweils eine **auf jeden Fall kontextfreie, wenn möglich sogar reguläre Grammatik** an, welche die Sprache erzeugt.

Teilaufgabe 3.3 2 Punkte

Geben Sie für jede der Sprachen L_1, L_2, L_3 und L_4, für welche es möglich ist, einen **regulären Ausdruck** an, der die Sprache erzeugt.

Teilaufgabe 3.4 3 Punkte

Geben Sie für die Sprache $L_3 = \{(0^n 1^m)^k \mid n, m, k \in \mathbb{N}\} = \{01, 0001, 0111101010111, \ldots\}$ einen **endlichen Automaten (DEA oder NEA)** an, der die Sprache akzeptiert.

Teilaufgabe 3.5 6 Punkte

Geben Sie einen Kellerautomaten **Kellerautomaten (PDA oder DPDA)** an, welcher die Sprache $L_4 = \{0\omega\omega^R 0 \mid \omega \in \Sigma_1^* \wedge \omega^R \text{ ist } \omega \text{ rückwärts gelesen }\} = \{00, 0110, 010110011010, \ldots\}$ akzeptiert.

Teilaufgabe 3.6 7 Punkte

Geben Sie eine **Turing-Maschine (TM oder NTM)** an, welche die Sprache $L_2 = \{0^n 1^{2n} \mid n \in \mathbb{N}\} = \{011, 001111, 000111111, \ldots\}$ akzeptiert.

TEILAUFGABE 3.7 10 PUNKTE

Betrachtet wird immer noch $\Sigma_1 = \{0,1\}$. Zusätzlich sei nun die Grammatik $G_5 = (N, \Sigma_1, P, S)$ mit $N = \{S\}$ gegeben mit den Regeln

$$P: \begin{array}{lll} S & \to & 0S1 \\ S & \to & 1S0 \end{array} \qquad \begin{array}{lll} S & \to & SS \\ S & \to & \varepsilon \end{array}$$

a) Nutzen Sie die Grammatik G_5, um für das Wort $\omega_{50} = 0111010010$

   1) (2 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) (3 Punkte) Überführen Sie die Grammatik G_5 in die Chomsky-Normalform.
   Es reicht, wenn Sie die **Regelmenge** P' dieser äquivalenten Grammatik in CNF angeben.

d) (1 Punkt) Geben Sie an wie sich der Ableitungsbaum des Wortes ω_{50} nach P' (in CNF) vom Ableitungsbaum nach den Produktionsregeln aus P unterscheidet.

TEILAUFGABE 3.8 7 PUNKTE

Betrachtet wird immer noch $\Sigma_1 = \{0,1\}$. Zusätzlich sei nun gegeben die Sprache
$L_6 = \{\omega \in \Sigma_1 \mid \omega \text{ enthält den String 010 mindestens einmal}\} = \{010, 010010, 1101001, 101000100, \ldots\}$.
Geben Sie für L_6 an

a) (1 Punkt) einen erzeugenden regulären Ausdruck.

b) (3 Punkte) einen akzeptierenden nichtdeterministischen endlichen Automaten (NEA).
   Keine Punkte für einen NEA, der nicht mindestens ein nichtdeterministisches Element enthält.

c) (3 Punkte) einen akzeptierenden deterministischen endlichen Automaten (DEA).

Aufgabe 4 Berechenbarkeit, Entscheidbarkeit & Komplexität

Teilaufgabe 4.1 11 Punkte

Vervollständigen Sie den folgenden Lückentext, in dem Sie auf Ihrer Lösung für jeden Buchstaben angeben, durch welche (kein, ein oder mehrere passende Worte) Sie ihn ersetzen möchten.

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 „–“. **Für nicht genannte Buchstaben gibt es keine Punkte.**

___**A**___ alle formalen Sprachen werden von einer Turing-Maschine akzeptiert. Formale Sprachen vom Chomsky-Typ 2 werden zusätzlich von ___**B**___ akzeptiert. Turing-Maschinen akzeptieren nicht nur Sprachen, sie berechnen auch Funktionen . Allerdings gilt, das alle Funktionen, die ein ___**C**___ nicht berechnen kann, kann eine Turing-Maschine nicht berechnen. Dies hat sich ___**D**___ geändert, seit die Überlegenheit der Quantencomputer (Quantum Supremacy) bewiesen wurde: Quantencomputer können ___**E**___ als herkömmliche Computer und damit Turing-Maschinen.

Eine formale Sprache L heißt entscheidbar, falls eine Turing-Maschine T existiert, die ___**F**___.
L heißt semi-entscheidbar, falls eine Turing-Maschine T existiert, die ___**G**___. Falls L nicht entscheidbar ist, heißt L ___**H**___.

Für alle ___**I**___ Probleme oder formalen Sprachen, kann die Zeit- und Raumkomplexität bestimmt werden. Vor allem die Zeitkomplexität ist wichtig, da nur Probleme aus z.B. der Klasse ___**J**___ für große Instanzen effizient lösbar sind. Dies wird sich mit zunehmendem technologischem Fortschritt ___**K**___ ändern.

Teilaufgabe 4.2 8 Punkte

Gesucht ist die Turing-Maschine T_4, welche die totale Funktion f_4 berechnet. T_4 erhält als Input eine natürliche Zahl n in Binärdarstellung gibt das Ergebnis n mod 4 (den Rest von n bei Division durch 4) in Binärdarstellung zurück.

Beispiele (in Dezimaldarstellung, Ihre Turing-Maschine soll aber Binär-Zahlen verarbeiten!):
$f_4(8) = 0, f_4(9) = 1, f_4(15) = 3, \ldots$

a) (6 Punkte) Geben Sie T_4 an! Verwenden Sie hierzu das Eingabealphabet $\Sigma_4 = \{0, 1\}$.

b) (2 Punkte) Wenn T_4 Zahlen in **Dezimaldarstellung** verarbeiten müsste, würde Sie dann im Vergleich zur Variante mit der Binärdarstellung

   1) mehr, weniger oder gleich viele Zustände brauchen?

   2) mehr, weniger oder gleich viele Arbeitsschritte für die gleiche Rechenaufgabe benötigen?

   Geben Sie auf Ihrer Lösung die jeweils richtigen Wörter an!

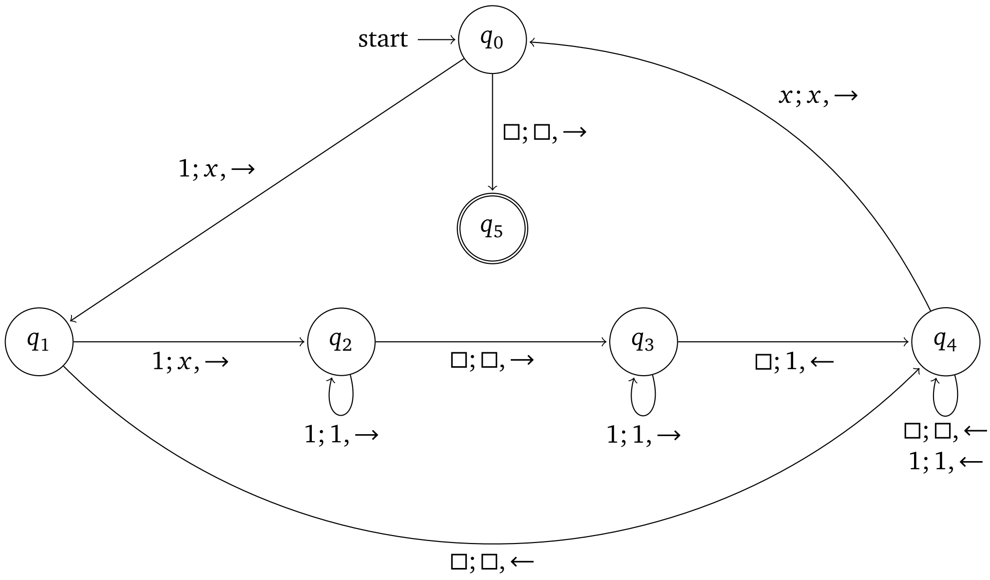

Abbildung 1: Erweitertes Zustandsübergangsdiagramm für T_x

Teilaufgabe 4.3 14 Punkte

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

a) Bestimmen Sie für die Worte $\omega_1 = 1$ (2 Punkte) und $\omega_2 = 11$ (3 Punkte) jeweils alle Konfigurationen welche die TM T_x während der Verarbeitung der Worte durchläuft.

b) (7 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	$f_0(\omega_i)$
$\omega_1 = \varepsilon$	
$\omega_2 = 1$	
$\omega_3 = 11$	
$\omega_4 = 111$	
$\omega_5 = 1111$	
$\omega_6 = 11111$	
$\omega_7 = 111111$	

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?