Klausur im WS 22/23, 07.02.2023

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. **55 Punkte**, eine sehr gute Note gibt es **ab ca. 85 Punkten**.

Name: ____________________

Matrikelnummer: ____________

Note: ____________

Aufgabe	1	2	3	4	5	Σ
Punkte	20	13	29	20	28	110
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) Quantencomputer können alle NP-vollständige Probleme effizient lösen.	☐	☐	
(b) Bei allen Automatenmodellen (D/NEA, (D)PDA, (D)LBA,(N)TM) akzeptieren die nichtdeterministen und die deterministischen Varianten die gleiche Sprachklasse.	☐	☐	
(c) Alle formalen Sprachen können von einer Turing-Maschine akzeptiert werden.	☐	☐	
(d) NP-vollständige Probleme sind nicht lösbar.	☐	☐	
(e) Ein Problem K ist NP-vollständig, wenn es in NP liegt und alle anderen Probleme in NP sich in Polynomialzeit auf K reduzieren lassen.	☐	☐	
(f) Transduktoren können pro gelesenes Eingabezeichen nur ein Ausgabezeichen schreiben.	☐	☐	
(g) Endliche Automaten halten immer an. Kellerautomaten und Turing-Maschinen können auch unendlich lange laufen	☐	☐	
(h) Die Verneinung von „Theoretische Informatik ist wichtig für alle Teilbereiche der Informatik“ ist „Theoretische Informatik ist für keinen Teilbereich der Informatik wichtig.“	☐	☐	
(i) Alle Entscheidungsprobleme liegen in NP.	☐	☐	
(j) Jede reguläre Grammatik, die ε nicht erzeugt, lässt sich in Chomsky-Normalform überführen.	☐	☐	

AUFGABE 2, 13 PUNKTE
LOGIK UND KOMPLEXITÄTSTHEORIE

Folgendes sei gegeben:

- die Menge der Probleme $B = \{$ Halteproblem, Traveling Salesperson Problem, Rucksackproblem, Primzahlentest-Problem, Pfadexistenz-Problem, Fleißiger Biber Problem $\}$
- die Menge aller **polynomiellen** Komplexitätsklassen $K = \{$ P, NP, PSPACE, NSPACE $\}$
- Aussageform $E(p)$: „p ist entscheidbar" (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 $k_1, k_2 \in K$)

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

(a) (2 Punkte) $L(\text{Traveling Salesperson Problem}, \text{NP})$

(b) (3 Punkte) $\neg\,(E(\text{Halteproblem}) \wedge E(\text{Rucksackproblem}))$

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

(d) (2 Punkte) $\neg\,\exists_{p \in B}\; E(p)$

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

Aufgabe 3, 29 Punkte
Reguläre Sprachen

3.1 Reguläre Ausdrücke

Alle an der HTWG stattfindenden Prüfungen haben eine **Prüfungsnummer**. Alle Prüfungsnummern sind (vereinfacht) wie folgt aufgebaut:

- Jede Prüfungsnummer hat 5 Ziffern, welche jeweils die Werte 0-9 annehmen können.
- Die erste Ziffer entspricht der Versionsnummer der SPO der Prüfung.
- Die zweite und dritte Ziffer entsprechen der Modulnummer der Prüfung.
- Die vierte Ziffer entspricht dem Semester, welchem die Prüfung zugeordnet ist.
- Ist die letzte Ziffer eine gerade Zahl, so ist die Prüfung benotet.
- Ist die letzte Ziffer eine ungerade Zahl, so ist die Prüfung unbenotet.

Angenommen, Sie müssen aus einer großen Text-Datei (in welcher Prüfungsnummern, -namen und -ergebnisse stehen) Prüfungsnummern herausfiltern (die betreffenden Strings mit einem regulären Ausdruck matchen).

(a) (2 Punkte) Geben Sie einen regulären Ausdruck an, welcher alle Prüfungsnummern findet, die unbenoteten Prüfungen zugeordnet sind.

(b) (3 Punkte) Geben Sie einen regulären Ausdruck an, welcher alle Prüfungsnummern von benoteten Prüfungen nach AIN SPO 3, die dem 1. oder 2. Semester zugeordnet sind, findet.

(c) (4 Punkte) Welche Zeichenketten findet der folgende reguläre Ausdruck? Beschreiben Sie diese so exakt wie möglich Im Zusammenhang mit Prüfungen, was könnten diese darstellen?

```
r = \d{5}( [A-Za-z][A-Za-z0-9\w,\+\.\-\/]*)+( \d)?
```

3.2 Endliche Automaten und reguläre Grammatiken

Eine **BSYS-PNR** bezeichnet eine Prüfungsnummer (siehe Aufgabe 3.1), die nach allen AIN SPOs (Nummern 1-3) dem Modul 13 (Betriebssysteme, 3. Semester) zugeordnet ist und deren Endziffer 0 oder 1 ist.
Wir betrachteten das Alphabet $\Sigma_B = \{0, 1, 2, \ldots, 5\}$, sowie die Sprache

$$\begin{aligned} L_B &= \{\omega \in \Sigma_B^* \mid \omega \text{ enthält mindestens eine valide BSYS-PNR } \} \\ &= \{11330, 21330, \ldots, 0113300, 0113301, \ldots, 1133011330, \ldots, 4531331210, \ldots\}. \end{aligned}$$

L_B enthält also alle aus den Ziffern 1-5 bestehende Strings, in denen mindestens eine BSYS-PNR vorkommt.

Geben Sie für L_B an

(a) (2 Punkte) einen erzeugenden regulären Ausdruck.

(b) ($4\frac{1}{2}$ Punkte) einen akzeptierenden nichtdeterministischen endlichen Automaten (NEA). Keine Punkte für einen NEA ohne mindestens ein nichtdeterministisches Element.

(c) ($8\frac{1}{2}$ Punkte) einen akzeptierenden deterministischen endlichen Automaten (DEA).

(d) (5 Punkte) Eine **reguläre** Grammatik, welche L_B erzeugt.

Aufgabe 4, 20 Punkte
Kontextfreie Sprachen

4.1 Wir betrachten das Alphabet $\Sigma_A = \{a, b\}$, sowie die Sprache

$$L_A = \{\omega = v_1 v_2 \dots v_n \mid n \in \mathbb{N}, \forall_{1\leq i\leq n} \exists_{k\in\mathbb{N}}\ v_i = a^k b^k\}$$
$$= \{ab, aabb, \dots, abab, abaabb, \dots, aaabbbababab, aabbaaabbbabaabb, \dots\}.$$

L_A enthält also alle Strings, die aus der Hintereinanderreiung beliebig vieler Strings mit jeweils gleich vielen as wie bs bestehen.

Weiterhin betrachten wir die Grammatik $G_A = (N, \Sigma_A, P, S)$ mit $N = \{S, T\}$ und der Produktionsmenge

$$P: \begin{array}{ll} S & \to SS \\ S & \to aTb \\ T & \to aTb \\ T & \to \varepsilon \end{array}$$

(a) (2 Punkte) Begründen Sie, weshalb die Grammatik G_A die Sprache L_A erzeugt, dass also $L_A = \mathscr{L}(G_A)$ gilt.

(b) Nutzen Sie die Grammatik G_A, um für das Wort $\omega_a = abaabb$

α) (2 Punkte) eine Ableitung aus dem Startsymbol S

β) (2 Punkte) den entsprechenden Syntaxbaum

anzugeben.

(c) (5 Punkte) Überführen Sie die Grammatik G_A in die Chomsky-Normalform.
Es reicht, wenn Sie die **Regelmenge** P' dieser äquivalenten Grammatik $G'_A = (N, \Sigma_A, P', S)$ in CNF angeben.

(d) (3 Punkte) Zeigen Sie mit Hilfe des CYK-Algorithmus, dass das Wort $abab$ von G_A erzeugt werden kann.

(e) (6 Punkte) Konstruieren Sie den Kellerautomaten P_A (PDA oder DPDA), welcher L_A akzeptiert.

erreichte Punkte: ______ von 9 möglichen

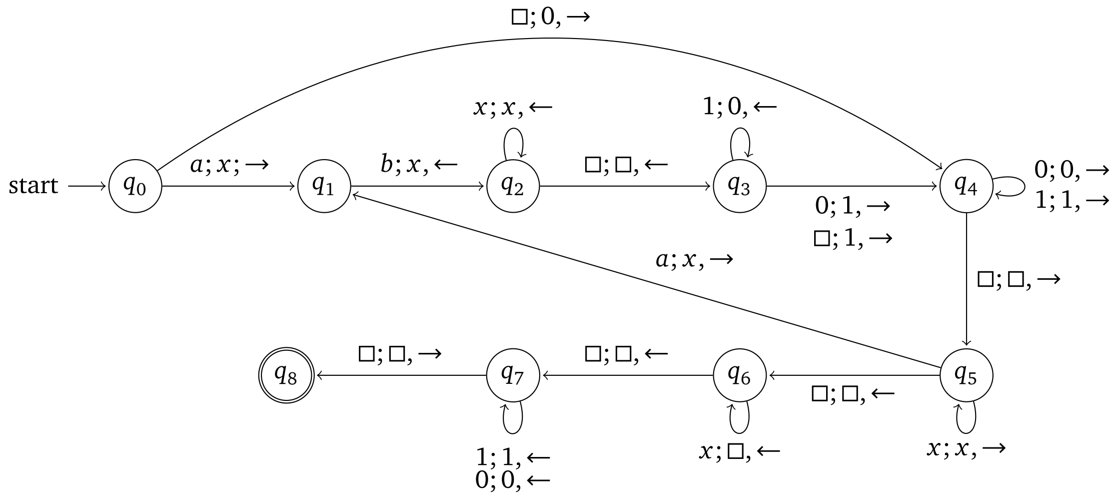

Abbildung 1: Erweitertes Zustandsübergangsdiagramm für T_X

Aufgabe 5, 28 Punkte
Berechenbarkeit, Entscheidbarkeit & Komplexität

5.1 Wir betrachten das Alphabet $\Sigma_X = \{a, b\}$ und die Funktion $f_X : \Sigma_X^* \to \{0,1\}^*$ welche von der Turing-Maschine $T_X = (Q, \Sigma_X, \Pi_X, \delta, q_0, F) = (\{q_0, q_1, \ldots, q_8\}, \{a, b\}, \{a, b, x, 0, 1, \Box\}, \{q_8\}, \delta)$ mit δ gegeben durch Abbildung 1 berechnet wird.

(a) (7 Punkte) Bestimmen Sie für die Worte $\omega_1 = \varepsilon$ (2 ½ Punkte) und $\omega_3 = \mathbf{ab}$ (4½ Punkte) alle Konfigurationen, welche T_X während der Verarbeitung der Worte durchläuft. Kürzen Sie sehr lange, uninteressante Berechnungsabschnitte durch „…“ bzw. „*“ ab!!

(b) (9 Punkte) Geben Sie für jedes der in der nebenstehenden Tabelle angegebenen Eingabewörter $\omega_i, i \in \{1, 2, \ldots, 9\}$ 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 „⊥“.

(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	$f_X(\omega_i)$
$\omega_1 = \varepsilon$	
$\omega_2 = a$	
$\omega_3 = ab$	
$\omega_4 = ba$	
$\omega_5 = aba$	
$\omega_6 = abab$	
$\omega_7 = ababb$	
$\omega_8 = ababab$	
$\omega_9 = abababab$	

5.2 Betrachten Sie T_X als akzeptierende Turing-Maschine.

(a) (2 Punkte) Geben Sie die Sprache $L_X = \mathscr{L}(T_X)$ an, die von T_X akzeptiert wird.

(b) (2 Punkte) Von welchem Chomsky-Typ ist L_X?
Begründen Sie Ihre Meinung!

(c) (2 Punkte) Ist L_X entscheidbar, semi-entscheidbar, oder unentscheidbar?
Begründen Sie Ihre Meinung!

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

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