Theorie der Informatik (10948-01)
Übungsklausur FS 2015

Prof. Dr. Malte Helmert
Gabriele Röger
Universität Basel
Departement Mathematik und Informatik

Name: ________________________________

Immatrikulationsnummer: ________________________________

- Die Klausur besteht aus einem **Multiple-Choice-Teil** und 6 **weiteren Aufgaben**.
- Bitte **schreiben Sie Ihren Namen und Ihre Immatrikulationsnummer auf dieses Titelblatt**.
- Als Hilfsmittel ist **ein von Ihnen vorbereitetes A4-Blatt** (beidseitig beschrieben) erlaubt. Weitere Hilfsmittel wie Vorlesungsfolien, Skripte, Bücher, weitere Notizen oder Taschenrechner sind nicht erlaubt. Des Weiteren sind alle elektronischen Geräte (wie z.B. Mobiltelefone) auszuschalten.
- Für die Bearbeitung der Aufgaben haben Sie **105 Minuten** Zeit.
- Benutzen Sie zur Bearbeitung der Aufgaben jeweils den Platz unterhalb der Aufgaben sowie falls nötig den Platz auf der Rückseite.
- Falls Sie mehrere Lösungsansätze einer Aufgabe erarbeiten, markieren Sie deutlich, welcher gewertet werden soll.

	Erreichbare Punkte	**Erzielte Punkte**
Multiple-Choice-Fragen	20	
Aufgabe 1	10	
Aufgabe 2	10	
Aufgabe 3	10	
Aufgabe 4	10	
Aufgabe 5	10	
Aufgabe 6	10	
Gesamt	80	
Note	(1,0–6,0)	

Multiple-Choice-Fragen (10× 2 Punkte)

(a) Welche der folgenden Behauptungen zu aussagenlogischen Formeln sind wahr?

☐ Ist φ erfüllbar, dann ist $\neg\varphi$ unerfüllbar.

☐ Ist φ unerfüllbar, gilt für jedes ψ, dass $\varphi \models \psi$.

☐ Zu jeder Formel gibt es eine gleich grosse logisch äquivalente Formel in KNF.

☐ Ist *WB* eine unerfüllbare Wissensbasis, kann man mit dem Resolutionskalkül die leere Klausel $\square$ aus *WB* ableiten.

(b) Welche der folgenden Behauptungen zur Prädikatenlogik sind wahr?

☐ $(\forall x(P(x) \wedge \exists y(Q(y,x) \vee (x = y))) \vee P(y))$ ist ein Satz.

☐ $(\forall x\varphi \wedge \forall x\psi) \equiv \forall x(\varphi \wedge \psi)$

☐ $(\forall x\varphi \vee \forall x\psi) \equiv \forall x(\varphi \vee \psi)$

☐ Sind die Formelmengen Φ und Ψ erfüllbar, so ist auch $\Phi \cup \Psi$ erfüllbar.

(c) Welche der folgenden Aussagen über Sprachen und Grammatiken sind wahr?

☐ Für jede Sprache gibt es eine Grammatik, die sie erzeugt.

☐ Jede kontextfreie Sprache kann von einer kontextsensitiven Grammatik erzeugt werden.

☐ Für $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ gilt $\varepsilon \in \Sigma^*$ und $\mathtt{aba} \in \Sigma^*$.

☐ Das leere Wort ε ist in jeder unendlichen Typ-0-Sprache enthalten.

(d) Welche der folgenden Aussagen zu regulären Sprachen sind wahr?

☐ Jede endliche Sprache ist regulär.

☐ Zu jedem NFA mit n Zuständen gibt es einen NFA mit $n+1$ Zuständen, aber nur einem Endzustand, der dieselbe Sprache akzeptiert.

☐ Akzeptiert ein NFA mit n Zuständen eine Sprache, so hat jeder Minimalautomat für diese Sprache höchstens n Zustände.

☐ Mit dem Pumping-Lemma kann man beweisen, dass eine Sprache regulär ist.

(e) Welche der folgenden Aussagen zu kontextfreien Sprachen und PDAs sind wahr?

☐ Jede Sprache, die durch eine kontextfreie Grammatik beschrieben werden kann, kann auch von einem PDA erkannt werden.

☐ Die Sprache $L = \{\mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}_0\}$ ist kontextfrei.

☐ Jede kontextfreie Sprache kann von einem PDA mit nur einem Zustand erkannt werden.

☐ Ist L_1 eine kontextfreie Sprache und L_2 eine reguläre Sprache, so ist $L_1 \cup L_2$ kontextfrei.

(f) Welche der folgenden Aussagen sind wahr? Betrachten Sie bei dieser Frage nur *numerische* Funktionen $f : \mathbb{N}_0^k \to \mathbb{N}_0$, keine Funktionen mit *Wörtern* als Eingabe.

- ☐ Turing-Maschinen sind weniger mächtig als WHILE-Programme.
- ☐ Zu jeder deterministischen Turingmaschine kann ein GOTO-Programm konstruiert werden, das dieselbe Funktion berechnet.
- ☐ Zu jedem GOTO-Programm kann eine deterministische Turingmaschine konstruiert werden, die dieselbe Funktion berechnet.
- ☐ Die Ackermann-Funktion ist WHILE-berechenbar.

(g) Welche der folgenden Probleme sind entscheidbar?

- ☐ „Hält ein gegebenes GOTO-Programm an, wenn alle Eingabevariablen den Wert 0 aufweisen?"
- ☐ die Sprache $L \cup \bar{L}$, wobei L semientscheidbar ist
- ☐ das Travelling Salesperson Problem (TSP)
- ☐ die Sprache $\{\varepsilon\}$

(h) Sei X ein unentscheidbares Problem. Welche der folgenden Aussagen folgen?

- ☐ Alle Probleme Y mit $X \leq Y$ sind unentscheidbar.
- ☐ Alle Probleme Y mit $Y \leq X$ sind unentscheidbar.
- ☐ X ist das spezielle Halteproblem.
- ☐ X und $\bar{X}$ sind semi-entscheidbar.

(i) Welche der folgenden Aussagen beschreiben die Beweisidee des Satzes von Cook und Levin?

- ☐ Übersetze die Funktionsweise einer nichtdeterministischen Turingmaschine mit polynomieller Laufzeit in eine Logikformel.
- ☐ Zeige, dass das Erfüllbarkeitsproblem der Aussagenlogik unentscheidbar ist.
- ☐ Reduziere jedes Problem in NP auf SAT.
- ☐ Konstruiere zu jeder polynomiell grossen Logikformel eine Turingmaschine, die diese erfüllt.

(j) Seien X und Y NP-vollständige Probleme. Was folgt?

- ☐ Wenn es für X effiziente Algorithmen gibt, dann auch für Y.
- ☐ Wenn es für X effiziente Algorithmen gibt, dann auch für BinPacking und TSP.
- ☐ $X \leq_{\mathrm{p}} \mathrm{SAT}$
- ☐ $\mathrm{SAT} \leq_{\mathrm{p}} X$

Aufgabe 1 (4+2+4 Punkte)

(a) Geben Sie für folgende aussagenlogische Formel ein Modell an und beweisen Sie mit der Semantik der Aussagenlogik, dass das Modell tatsächlich eine erfüllende Belegung für die Formel ist.

$$((A \land (B \lor C)) \land \neg C)$$

(b) Seien A, B und C atomare Aussagen. Geben Sie für jede der folgenden Eigenschaften eine aussagenlogische Formel über $\{A, B, C\}$ an.

erfüllbar:	
allgemeingültig:	
unerfüllbar:	
besitzt genau 3 Modelle:	

(c) Zeigen Sie durch Anwendung von Äquivalenzen, dass folgende aussagenlogische Formeln äquivalent sind.

$$((B \lor \neg C) \rightarrow (A \lor B)) \equiv ((A \lor B) \lor C)$$

Zusätzlicher Platz für Aufgabe 1:

Aufgabe 2 (4+6 Punkte)

Sei $\Sigma = \{\mathtt{a}, \mathtt{b}\}$.

(a) Geben Sie einen DFA an, der die Sprache akzeptiert, die durch den regulären Ausdruck $\mathtt{a}^*\mathtt{b}(\mathtt{ab})^*$ beschrieben wird. Es reicht aus, wenn Sie den DFA graphisch durch ein Diagramm angeben.

(b) Verwenden Sie das Pumping-Lemma, um zu zeigen, dass $L = \{\mathtt{a}^n\mathtt{b}^{2n} \mid n \geq 0\}$ nicht regulär ist.

Zusätzlicher Platz für Aufgabe 2:

Aufgabe 3 (4+6 Punkte)

Betrachen Sie die folgende Sprache

$$L = \{\mathtt{a}^n\mathtt{b}^m\mathtt{c}^{2n} \mid n, m \geq 0\}$$

über dem Alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$.

(a) Geben Sie eine kontextfreie Grammatik G an, die L erzeugt, also $\mathcal{L}(G) = L$. Geben Sie hierzu alle Komponenten von G an.

(b) Konstruieren Sie einen Kellerautomaten (PDA) M, der genau L akzeptiert. Es reicht aus, wenn Sie M graphisch durch ein Diagramm angeben.

Zur Erinnerung: Ein PDA wird mit dem Kellersymbol `#` auf dem Keller initialisiert und akzeptiert genau dann, wenn das Eingabewort abgearbeitet und der Keller leer ist (es gibt also keinen akzeptierenden Endzustand).

Zusätzlicher Platz für Aufgabe 3:

Aufgabe 4 (4+4+2 Punkte)

(a) Schreiben Sie ein LOOP-Programm, das die Summe der ersten n natürlichen Zahlen zurückliefert, also folgende Funktion berechnet:

$$f(n) = \sum_{i=1}^{n} i$$

Sie dürfen alle Syntaxkonstrukte aus Vorlesung und Übungen verwenden.

(b) Geben Sie an, wie sich das folgende syntaktische Konstrukt durch bekannte Konstrukte für LOOP-Programme simulieren lässt. Die Semantik sei wie folgt: x_i wird von 1 bis x_j hochgezählt und für jeden dieser Werte wird P einmal ausgeführt. Sie dürfen alle Syntaxkonstrukte aus Vorlesung und Übungen verwenden. Weiterhin dürfen Sie annehmen, dass x_i und x_j in P nicht verändert werden.

$$\begin{array}{l} \textbf{FOR } x_i = 1 \textbf{ TO } x_j \textbf{ DO} \\ \quad P \\ \textbf{END} \end{array}$$

(c) Welche einstellige Funktion berechnet das folgende WHILE-Programm? Ist diese Funktion LOOP-berechenbar? Begründen Sie Ihre Antwort zur zweiten Frage.

$$\begin{array}{l} x_2 := 1; \\ x_3 := 0; \\ \textbf{WHILE } x_2 \neq 0 \textbf{ DO} \\ \quad \textbf{IF } x_1 = x_3 \textbf{ THEN} \\ \quad\quad x_2 := 0 \\ \quad \textbf{END}; \\ \quad x_3 := x_3 + 2 \\ \textbf{END}; \\ x_0 := 1 \end{array}$$

Zusätzlicher Platz für Aufgabe 4:

Aufgabe 5 (3+1+6 Punkte)

(a) Beschreiben Sie informell (jeweils 1 Satz), was es bedeutet, dass eine gegebene Sprache $L \subseteq \Sigma^*$

- *entscheidbar*,
- *semi-entscheidbar*,
- *unentscheidbar* ist.

(b) Beschreiben Sie (ohne Beweis) die Beziehungen zwischen den Eigenschaften *entscheidbar*, *semi-entscheidbar*, *unentscheidbar*: welche Eigenschaften implizieren welche anderen? Welche Eigenschaften schliessen sich gegenseitig aus?

(c) Welche der folgenden Sprachen sind entscheidbar? Geben Sie jeweils eine kurze Begründung (1 Satz).

- $L_1 = \{w \in \{\mathtt{0},\mathtt{1}\}^* \mid M_w$ berechnet eine Funktion mit endlichem Definitionsbereich$\}$
- $L_2 = \{w \in \{\mathtt{0},\mathtt{1}\}^* \mid M_w$ berechnet eine LOOP-berechenbare Funktion $\}$
- $L_3 = \{w \in \{\mathtt{0},\mathtt{1}\}^* \mid M_w$ berechnet eine Turing-berechenbare Funktion $\}$

Hinweis: Verwenden Sie (wo möglich) den Satz von Rice, um Unentscheidbarkeit zu zeigen.

Zusätzlicher Platz für Aufgabe 5:

Aufgabe 6 (3+7 Punkte)

Betrachten Sie die folgenden Entscheidungsprobleme:

DirHamiltonPath:

- *Gegeben:* gerichteter Graph $G = \langle V, E \rangle$
- *Gefragt:* Enthält G einen Hamiltonpfad?

DirHamiltonPathWithEndPoints:

- *Gegeben:* gerichteter Graph $G = \langle V, E \rangle$, Startknoten $v_\mathrm{s} \in V$, Endknoten $v_\mathrm{e} \in V$
- *Gefragt:* Enthält G einen Hamiltonpfad von v_s nach v_e, also einen Hamiltonpfad $\pi = \langle v_1, \dots, v_n \rangle$ mit $v_1 = v_\mathrm{s}$ und $v_n = v_\mathrm{e}$?

(a) Zeigen Sie, dass DirHamiltonPathWithEndPoints in NP liegt, indem Sie einen nichtdeterministischen, polynomiellen Algorithmus angeben.

(b) Beweisen Sie, dass DirHamiltonPathWithEndPoints NP-hart ist. Sie dürfen dabei verwenden, dass das Problem DirHamiltonPath NP-vollständig ist.

Zur Erinnerung: Ein *Hamiltonpfad* in einem gerichteten Graphen $\langle V, E \rangle$ ist eine Knotenfolge $\pi = \langle v_1, \dots, v_n \rangle$, die einen Pfad definiert ($\langle v_i, v_{i+1} \rangle \in E$ für alle $1 \leq i < n$) und jeden Knoten des Graphen genau einmal enthält.

Zusätzlicher Platz für Aufgabe 6: