# Theorie der Informatik (10948)
# Probeklausur

FS 2019
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 **120 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 allgemeinen Aussagen über aussagenlogische Formeln $\varphi$ und $\psi$ sind wahr?

- ☐ Wenn $\varphi$ nicht allgemeingültig ist, dann ist $\varphi$ erfüllbar.
- ☐ Wenn $\varphi \models \psi$ und $\psi \models \varphi$ gelten, dann gilt $\varphi \equiv \psi$.
- ☐ Ist $(\varphi \leftrightarrow \psi)$ erfüllbar, sind $\varphi$ und $\psi$ logisch äquivalent.
- ☐ Wenn $\varphi \equiv \psi$ gilt und $\varphi$ allgemeingültig ist, dann ist $\psi$ allgemeingültig.

(b) Welche der folgenden Aussagen zur Aussagenlogik sind wahr?

- ☐ Ist mindestens eine der Formelmengen $\Phi$ oder $\Psi$ unerfüllbar, so ist auch $\Phi \cup \Psi$ unerfüllbar.
- ☐ Zu jeder DNF-Formel gibt es eine *gleich grosse* logisch äquivalente KNF-Formel.
- ☐ Ist *WB* eine unerfüllbare Wissensbasis, kann man mit dem Resolutionskalkül die leere Klausel $\square$ aus *WB* ableiten.
- ☐ Ist Formel $\varphi$ unerfüllbar, gilt für jede Formel $\psi$, dass $\varphi \models \psi$.

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

- ☐ $(\forall x(P(x) \land \exists y(R(y,x) \lor \neg Q(y,x))) \lor P(y))$ ist ein Satz (= eine geschlossene Formel).
- ☐ Bei einer geschlossenen Formel ist die Variablenbelegung irrelevant, wenn man bestimmen möchte, ob die Formel unter einer Interpretation wahr ist.
- ☐ $\forall x \forall y \mathrm{P}(x,y) \models \forall x \exists y \mathrm{P}(y,x)$
- ☐ $(\forall x \varphi \land \forall x \psi) \equiv \forall x(\varphi \land \psi)$

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

- ☐ Für jede durch eine reguläre Grammatik erzeugbare Sprache gibt es einen endlichen Automaten (DFA, NFA), der sie akzeptiert.
- ☐ Jede endliche Sprache ist regulär.
- ☐ Der reguläre Ausdruck $a^*b^*|ab^*aa^*$ beschreibt eine Sprache, die die Wörter $\varepsilon$ und *abbbaa* enthält, aber nicht das Wort *abbaba*.
- ☐ Das Äquivalenzproblem für reguläre Sprachen ist entscheidbar.

(e) Welche der folgenden Aussagen über Sprachen und Automaten sind wahr?

- ☐ Reguläre Sprachen können nicht das leere Wort $\varepsilon$ enthalten.
- ☐ Es gibt Sprachen, die von keinem Kellerautomaten (PDA) akzeptiert werden können.
- ☐ Für jede kontext-freie Sprache gibt es eine deterministische Turingmaschine, die sie akzeptiert.
- ☐ Jede von einer deterministischen Turingmaschine akzeptierte Sprache ist entscheidbar.

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

- ☐ WHILE-Programme sind mächtiger als GOTO-Programme.
- ☐ Zu jedem WHILE-Programm kann ein LOOP-Programm konstruiert werden, das dieselbe Funktion berechnet.
- ☐ Jede totale Funktion ist LOOP-berechenbar.
- ☐ Jede primitiv rekursive Funktion kann von einem LOOP-Programm berechnet werden.

(g) Sei $X$ ein unentscheidbares Problem. Welche der folgenden Aussagen folgen daraus?

- ☐ $X$ ist semi-entscheidbar.
- ☐ $X$ ist nicht semi-entscheidbar.
- ☐ Alle Probleme $Y$ mit $X \leq Y$ sind unentscheidbar.
- ☐ Alle Probleme $Y$ mit $Y \leq X$ sind unentscheidbar.

(h) Welche der folgenden Probleme/Sprachen sind entscheidbar?

- ☐ Terminiert ein gegebenes GOTO-Programm, wenn alle Eingabeparameter 0 sind?
- ☐ Die Sprache $\bar{L}$, wobei $L$ entscheidbar ist.
- ☐ Gibt es in einem gegebenen Graphen eine Knotenüberdeckung (vertex cover) der Grösse höchstens $K$?
- ☐ Berechnet ein gegebenes WHILE-Programm eine gegebene $\mu$-rekursive Funktion?

(i) Seien $A$, $B$ und $C$ Probleme in NP. Sei ferner $B$ NP-hart und gelte $A \leq_{\mathrm{p}} B$ und $B \leq_{\mathrm{p}} C$. Welche der folgenden Aussagen können Sie daraus ableiten?

- ☐ $A$ ist NP-vollständig.
- ☐ $A$ ist nicht NP-vollständig.
- ☐ $B$ ist NP-vollständig.
- ☐ $C$ ist NP-vollständig.

(j) Sei $X$ ein Problem in P und $Y$ ein NP-vollständiges Problem. Was folgt daraus?

- ☐ Es gibt einen deterministischen, polynomiellen Algorithmus für $X$.
- ☐ Falls gilt, dass $X \leq_{\mathrm{p}} Y$, dann ist P = NP.
- ☐ Wenn es einen deterministischen polynomiellen Algorithmus für $Y$ gibt, dann gibt es einen deterministischen polynomiellen Algorithmus für SAT
- ☐ $Y$ ist entscheidbar.

**Aufgabe 1** (4 + 4 + 2 Punkte)

(a) Verwenden Sie logische Äquivalenzen um die folgende Formel in DNF zu bringen. Wenden Sie in jedem Schritt nur eine Äquivalenz an.

$$\psi = (\neg(A \rightarrow \neg(B \vee (D \wedge E))) \rightarrow C)$$

(b) Beweisen Sie ohne Wahrheitstabelle oder Äquivalenzumformungen, dass

$$\varphi = (((A \vee B) \rightarrow (B \wedge C)) \rightarrow (\neg A \vee B))$$

eine Tautologie ist. Wenn Sie möchten, können Sie die Implikationen durch die entsprechenden Formeln ohne $\rightarrow$ ersetzen. Alternativ können Sie auch direkt mit der entsprechenden Semantik der Implikationen argumentieren.

(c) Geben Sie ein Modell mit Universum $U = \{a, b, c, d\}$ für folgende prädikatenlogische Formel für Signatur $\langle \{x, y, z\}, \{\mathrm{k}\}, \{\}, \{P\} \rangle$ mit 3-stelligem $P$ an.

$$\chi = \forall x \exists y P(x, y, \mathrm{k})$$

*Zusätzlicher Platz für Aufgabe 1:*

**Aufgabe 2** (7 + 3 Punkte)

(a) Beweisen Sie mit Hilfe des Pumping-Lemmas: die Sprache

$$L = \{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k \mid i, j, k \geq 0, i = j + k\}$$

ist nicht regulär.

(b) Geben Sie einen DFA an, der die Sprache akzeptiert, die durch den regulären Ausdruck $\gamma = \mathtt{0}^*\mathtt{1}(\mathtt{01})^*$ beschrieben wird.

*Zusätzlicher Platz für Aufgabe 2:*

**Aufgabe 3** (5 + 5 Punkte)

Betrachte Sprache $L = \{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k \mid i, j, k \geq 0, j = i + k\}$.

(a) Geben Sie eine kontext-freie Grammatik an, die $L$ erzeugt.

(b) Geben Sie einen Kellerautomaten (Push-Down Automaton, PDA) an, der $L$ akzeptiert.

Geben Sie jeweils eine vollständige Beschreibung mit allen Komponenten an.

*Zusätzlicher Platz für Aufgabe 3:*

**Aufgabe 4** (4 + 1 + 5 Punkte)

(a) Welche Funktion $f$ entsteht durch das primitive Rekursionsschema aus den folgenden beiden Funktionen? Beschreiben Sie $f$ so einfach wie möglich.

$$g(a) = 0$$
$$h(a, b, c) = 2a + c$$

(b) Ist $f$ WHILE-berechenbar? Ist $f$ $\mu$-rekursiv? Ist $f$ LOOP-berechenbar? Ist $f$ Turing-berechenbar?

(c) Simulieren Sie das syntaktische Konstrukt

$$\textbf{IF } x_i < c \textbf{ THEN } P \textbf{ END}$$

(mit $i, c \in \mathbb{N}_0$, $P$ einem LOOP-Programm) durch ein LOOP-Programm, das nur die Konstrukte aus der Basisdefinition verwendet.

*Zur Erinnerung:* LOOP-Programme sind induktiv wie folgt definiert:

- $x_i := x_j + c$ ist ein LOOP-Programm für alle $i, j, c \in \mathbb{N}_0$
- $x_i := x_j - c$ ist ein LOOP-Programm für alle $i, j, c \in \mathbb{N}_0$
- Wenn $P_1$ und $P_2$ LOOP-Programme sind, dann auch $P_1; P_2$
- Ist $P$ ein LOOP-Programm, dann ist es auch LOOP $x_i$ DO $P$ END für alle $i \in \mathbb{N}_0$

Beachten Sie, dass *kein* Ausdruck $x_i := c - x_j$ mit $i, j, c \in \mathbb{N}_0$ enthalten ist.

*Zusätzlicher Platz für Aufgabe 4:*

**Aufgabe 5** (4 + 6 Punkte)

(a) Geben Sie in jedem Aufgabenteil ein Beispiel für eine Sprache $L_i$ mit den genannten Eigenschaften (ohne Begründung), oder erklären Sie warum keine solche Sprache existiert (mit kurzer Begründung).

1. $L_1$ ist unentscheidbar und $L_1$ und $\overline{L_1}$ sind semi-entscheidbar.
2. $L_2$ ist eine Typ-0-Sprache und entscheidbar.
3. $L_3$ ist eine Typ-0-Sprache und unentscheidbar.
4. $L_4$ ist in NP und unentscheidbar.

(b) Welche der folgenden informell beschriebenen algorithmischen Probleme sind berechenbar? Geben Sie jeweils eine kurze Begründung (jeweils 1 Satz).

1. Gegeben zwei NFAs $M$ und $M'$, gibt es eine Eingabe $w$, die sowohl von $M$ als auch von $M'$ akzeptiert wird?
2. Gegeben eine deterministische Turing-Maschine $M$, berechne ein WHILE-Programm, welches dieselbe Funktion berechnet wie $M$.
3. Gegeben ein WHILE-Programm $P$ und ein GOTO-Programm $P'$, berechnen $P$ und $P'$ dieselbe Funktion?

*Zusätzlicher Platz für Aufgabe 5:*

**Aufgabe 6** (4 + 6 Punkte)

Betrachten Sie die folgenden Entscheidungsprobleme:

HittingSet:

- *Gegeben:* endliche Menge $T$, Menge von Mengen $S = \{S_1, \dots, S_n\}$ mit $S_i \subseteq T$ für alle $i \in \{1, \dots, n\}$, eine natürliche Zahl $K \in \mathbb{N}_0$
- *Gefragt:* Gibt es eine Menge $H$ mit höchstens $K$ Elementen, die aus jeder Menge aus $S$ mindestens ein Element enthält?

  Formal: Gibt es eine Menge $H$ mit $|H| \leq K$ und $H \cap S_i \neq \emptyset$ für alle $i \in \{1, \dots, n\}$?

VertexCover:

- *Gegeben:* ungerichteter Graph $G = \langle V, E \rangle$, natürliche Zahl $K \in \mathbb{N}_0$
- *Gefragt:* Enthält $G$ eine Knotenüberdeckung der Grösse $K$ oder weniger, d. h., eine Knotenmenge $C \subseteq V$ mit $|C| \leq K$ und $\{u, v\} \cap C \neq \emptyset$ für alle $\{u, v\} \in E$?

(a) Zeigen Sie HittingSet $\in$ NP, indem Sie einen nichtdeterministischen, polynomiellen Algorithmus angeben.

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

*Zusätzlicher Platz für Aufgabe 6:*