FernUniversität in Hagen

Bitte hier unbedingt Matrikelnummer und Adresse eintragen, sonst keine Bearbeitung möglich.

Postanschrift: FernUniversität 58084 Hagen

(Name, Vorname)

(Straße, Nr.)

(PLZ, Wohnort)

(Land)

FERNUNIVERSITÄT
EINGANG

MI

Bitte zurück an:
FERNUNIVERSITÄT
58084 Hagen

Fakultät für Mathematik und Informatik

Kurs 01657 – Grundlagen der Theoretischen Informatik A

Klausur am 19. Februar 2011

- ☐ Berlin
- ☐ Bern
- ☐ Bochum
- ☐ Bregenz
- ☐ Frankfurt
- ☐ Hamburg
- ☐ Karlsruhe
- ☐ Köln
- ☐ München
- ☐ Wien
- ☐ sonstige

Aufgabe	1	2	3	4	5	6	7	8	9	Summe
bearbeitet										
maximale Punktzahl	12	9	16	7	10	10	14	10	12	100
erreichte Punktzahl										

Datum: ____________________

Korrektor: ____________________

Vorbemerkungen zur Klausur zum Kurs 01657 „Grundlagen der Theoretischen Informatik A “

Es müssen Ihnen 8 Blätter Klausurtext vorliegen:

- das Deckblatt für Ihre Lösungen,
- eine Teilnahmebescheinigung für das Finanzamt,
- diese Vorbemerkungen,
- 3 Blätter mit 9 Aufgaben,
- 2 Blätter Anhang

Füllen Sie bitte vor der Bearbeitung das Deckblatt aus (Matrikel-Nr., Name, Adresse) und legen Sie es mit den Ausweisen zur Kontrolle bereit.

Für die Bearbeitung haben Sie 3 Stunden Zeit. Außer Schreibzeug sind keinerlei Hilfsmittel erlaubt.

Schreiben Sie bitte leserlich und lassen Sie einen breiten Korrekturrand. Beginnen Sie möglichst für jede Aufgabe eine neue Seite.

Folgende Unterlagen sind zusammengeheftet abzugeben:

- das ausgefüllte Deckblatt (bitte bearbeitete Aufgaben ankreuzen),
- die ausgefüllte Finanzamt-Bescheinigung (sofern gewünscht),
- Ihre Lösungsblätter, möglichst in der Reihenfolge der Aufgaben. Schreiben Sie zur Sicherheit auf jedes Blatt nochmals Name und Matrikel-Nr. Vergessen Sie bitte nicht, den ausgefüllten Fragebogen von Aufgabe 1 abzugeben!

Die Klausur ist bestanden, wenn Sie 40 von 100 Punkten erreichen. Wir bemühen uns, die Klausur zügig zu korrigieren und zurückzusenden. Sollten Sie nach zwei Wochen noch keine Nachricht haben, können Sie Ihr Ergebnis im Lehrgebiet erfragen.

Viel Erfolg!

Mit freundlichen Grüßen

IHRE KURSBETREUER

Aufgabe 1 *(12 Punkte)*

Es wird jede **richtig beantwortete Frage mit 1 Punkt**, jede **falsch beantwortete Frage mit -1 Punkt** und jede nicht beantwortete Frage mit 0 Punkten bewertet. Insgesamt können für jede Teilaufgabe nur nicht negative Punktzahlen erzielt werden.

Um eine Frage zu beantworten, kreuzen Sie das entsprechenden Kästchen in der Zeile an. Falls Sie eine Frage nicht beantworten wollen, dann machen Sie bitte kein Kreuz in der Zeile.

Vergessen Sie bitte nicht, Ihre Lösung für diese Aufgabe abzugeben!

(i) Welche der folgenden Aussagen ist/sind korrekt?

korrekt	falsch	
[ ]	[ ]	Eine Funktion $f :\subseteq \mathbb{N} \to \mathbb{N}$ ist berechenbar genau dann, wenn es eine Turing-Maschine M mit $f_M = f$ gibt.
[ ]	[ ]	Eine Funktion $f :\subseteq \mathbb{N} \to \mathbb{N}$ ist berechenbar genau dann, wenn es eine verallgemeinerte Registermaschine M mit $f_M = f$ gibt.
[ ]	[ ]	Eine Menge $M \subseteq \mathbb{N}$ ist rekursiv genau dann, wenn die Funktion $\mathrm{cf}_M :\subseteq \mathbb{N} \to \mathbb{N},\quad \mathrm{cf}_M(n) = \begin{cases} 1 & \text{falls } n \in M \\ 0 & \text{sonst} \end{cases}$ existiert.
[ ]	[ ]	Es gibt berechenbare Zahlenfunktionen, die weder primitiv-rekursiv noch LOOP-berechenbar sind.
[ ]	[ ]	Es gibt totale, berechenbare Zahlenfunktionen, die weder primitiv-rekursiv noch LOOP-berechenbar sind.
[ ]	[ ]	Eine Menge $M \subseteq \mathbb{N}$ ist rekursiv-aufzählbar genau dann, wenn die Funktion $d_M :\subseteq \mathbb{N} \to \mathbb{N},\quad d_M(n) = \begin{cases} 2^n & \text{falls } n \in M \\ \mathrm{div} & \text{sonst} \end{cases}$ für alle $n \in \mathbb{N}$, berechenbar ist.

(ii) Welche der folgenden Mengen ist rekursiv, rekursiv aufzählbar aber nicht rekursiv, nicht rekursiv aufzählbar?

rekursiv	r.a. und nicht rekursiv	nicht r.a.	
[ ]	[ ]	[ ]	$\{(i,x) \mid x \in \mathrm{Def}(\varphi_i) \wedge \Phi_i(x) < i^2 \cdot x\}$
[ ]	[ ]	[ ]	$\{(i,x) \mid x \in \mathrm{Def}(\varphi_i) \wedge \Phi_i(x) < \varphi_i(x) \cdot x\}$
[ ]	[ ]	[ ]	$\{(i,x) \mid x \in \mathrm{Def}(\varphi_i) \wedge \Phi_i(x) > \varphi_i(x) \cdot x\}$
[ ]	[ ]	[ ]	$\{(i,x) \mid \varphi_i(x) < i^2 \cdot x\}$
[ ]	[ ]	[ ]	$\{(i,j) \mid \varphi_i = \varphi_{i^2+j}\}$
[ ]	[ ]	[ ]	$\{(i,x) \mid \Phi_i(x) \in \mathbb{N} \wedge \Phi_i(x)$ ist keine Primzahl$\}$

Aufgabe 2 *(9 Punkte, je 3 pro Teilaufgabe)*

(i) Erläutern Sie den Zusammenhang zwischen Wort- und Zahlenberechenbarkeit.

(ii) Erläutern Sie den Zusammenhang zwischen verallgemeinerten Registermaschinen und berechenbaren Funktionen.

(iii) Erläutern Sie den Zusammenhang zwischen rekursiv-aufzählbaren und rekursiven Teilmengen von $\mathbb{N}$.

Aufgabe 3 *(16 Punkte)*

Gegeben sei das folgende While-Programm P:

$$((\text{WHILE } R_1 \neq 0 \text{ DO } ((R_1-; R_2+); R_0+); (\text{WHILE } R_2 \neq 0 \text{ DO } (R_2-; (R_0+; R_0+))))$$

(i) Welche einstellige Funktion $AC \circ \tau(P) \circ EC$ wird von P berechnet. (3 Punkte)

(ii) Zeigen Sie Ihre Aussage aus Teil (i). (13 Punkte)

(Hinweis: Die Syntax und Semantik der While-Programme finden Sie im Anhang.)

Aufgabe 4 *(7 Punkte)*

In dieser Aufgabe soll eine Bandmaschine konstruiert werden, die das (arithmetische) Dekrementieren von Dualzahlen berechnet, d.h. geben Sie eine ausführlich kommentierte Bandmaschine an, die die Funktion $f : \{0,1\}^* \to \{0,1\}^*$, definiert durch

$$(\forall n \in \mathbb{N}) \quad (\forall a_1, ..., a_n \in \{0,1\})(\exists b_1, ..., b_n \in \{0,1\}) \\ (f(a_1...a_n) = b_1...b_n \wedge \textstyle\sum_{i=1}^{n} b_i 2^{n-i} = (\sum_{i=1}^{n} a_i 2^{n-i}) \dot{-} 1))),$$

berechnet.
Korrektheitsbeweise müssen nicht geführt werden.

Aufgabe 5 *(10 Punkte)*

Zeigen Sie mittels Diagonalisierung, dass es Teilmengen M von $\mathbb{N}$ gibt, so dass weder M noch $\mathbb{N} \setminus M$ rekursiv-aufzählbar ist.

(Hinweis: Betrachten Sie φ_i auf den Eingaben $2 \cdot i$ und $2 \cdot i + 1$.)

Aufgabe 6 *(10 Punkte)*

Sei $\Sigma = \{0,1\}$ und bezeichne $W^{(1)}$ die Menge der berechenbaren Funktionen $f :\subseteq \Sigma^* \to \Sigma^*$. Definieren Sie eine surjektive Abbildung $\gamma : \Sigma^* \to W^{(1)}$, so dass die Funktion $t_\gamma :\subseteq \Sigma^* \times \Sigma^* \to \Sigma^*$, definiert durch $(\forall u, v \in \Sigma^*)(t_\gamma(u,v) = \gamma(u)(v))$, berechenbar ist. (Beweis)

(Hinweis: Benutzen Sie Standardnummerierungen.)

Aufgabe 7 *(14 Punkte, je 7 pro Teilaufgabe)*

(i) Zeigen Sie, dass zu jeder berechenbaren Funktion $g :\subseteq \mathbb{N} \to \mathbb{N}$ eine totale, berechenbare Funktion $f : \mathbb{N} \to \mathbb{N}$ existiert mit

$$(\forall i, n \in \mathbb{N})(\varphi_{f(i)}(n) = g(i))$$

(ii) Zeigen Sie mit Hilfe von Teil (i), dass sich jede rekursiv-aufzählbare Menge auf K_φ reduzieren lässt.

Aufgabe 8 *(10 Punkte)*

Zeigen Sie:
Eine Menge $A \subseteq \mathbb{N}$ ist genau dann rekursiv, wenn sie endlich ist oder eine totale, berechenbare Funktion $f : \mathbb{N} \to \mathbb{N}$ existiert mit

a) $(\forall i, j \in \mathbb{N})(i^2 < j \Rightarrow f(i) < f(j))$ und

b) $f(\mathbb{N}) = A$.

Aufgabe 9 *(12 Punkte)*

Geben Sie nicht-äquivalente Nummerierungen von $\mathbb{Z}$ an, d.h. definieren Sie Nummerierungen ν_1 und ν_2 von $\mathbb{Z}$, so dass <u>nicht</u> $\nu_1 \equiv \nu_2$ gilt. Beweisen Sie, dass tatsächlich $\nu_1 \not\equiv \nu_2$.

Anhang

Satz 3.2.5 (berechenbare Zahlenfunktionen)
Die wie folgt definierten Funktionen f und Tests t auf den natürlichen Zahlen sind berechenbar.

a) $\tilde{0} : \mathbb{N}^0 \to \mathbb{N},\ \tilde{0}() := 0$ (nullstellige Nullfunktion).
b) $Z : \mathbb{N} \to \mathbb{N},\ Z(x) := 0$ (einstellige Nullfunktion).
c) $S : \mathbb{N} \to \mathbb{N},\ S(x) := x + 1$ (Nachfolgerfunktion).
d) $V : \mathbb{N} \to \mathbb{N},\ V(x) := x \dot{-} 1$ (Vorgängerfunktion).
e) $\mathrm{pr}_i^{(k)} : \mathbb{N}^k \to \mathbb{N},\ \mathrm{pr}_i^{(k)}(x_1, \ldots, x_k) := x_i$ für $1 \leq i \leq k$ (Projektion).
f) $f(x_1, \ldots, x_n) := k$ für $n, k \in \mathbb{N}$ (konstante Funktionen).
g) $f(x, y) := x + y$ (Summe).
h) $f(x, y) := x \dot{-} y$ (arithmetische Differenz).
i) $f(x, y) := x \cdot y$ (Produkt).
j) $q, r :\subseteq \mathbb{N}^2 \to \mathbb{N}$ mit $x = q(x, y) \cdot y + r(x, y)$ und $r(x, y) < y$ für $y \neq 0$ und $q(x, y) = r(x, y) = \mathrm{div}$ für $y = 0$ (Quotient und Rest).
k) $f(x, y) := x^y$ mit $0^0 := 1$ (Exponentiation).
l) $f(x) := (\sqrt{x}$ falls x Quadratzahl, div sonst).
m) $\max(x, y)$.
n) $\min(x, y)$.
o) $f(x) := (1$ falls x Primzahl, 0 sonst).
p) $f(x) :=$ die x–te Primzahl.
q) $f(x, y) := (1$ falls $x < y$, 0 sonst). (Entsprechend für $\leq, =, \geq, >, \neq$ anstelle von $<$.)
r) Es sei $A \subseteq \mathbb{N}$ endlich. $f(x) := (0$ falls $x \in A$, 1 sonst).
s) Es sei $g : \mathbb{N} \to \mathbb{N}$ berechenbar. $f(x) :=$ (die kleinste Zahl y mit $g(y) = x$ falls $x \in \mathrm{Bild}(g)$, div sonst).
t) Es sei $g : \mathbb{N} \to \mathbb{N}$ berechenbar, und es gelte $(\forall y)\, g(y) < g(y + 1)$. Dann sei $f(x) := \max\{y \mid g(y) \leq x\}$.
u) $f(a, b) := \mathrm{ggT}(a, b)$ falls $a, b > 0$, $f(a, b) := 0$ sonst.

Definition 4.4.1 (WHILE–Programme)
Für jedes $i \in \mathbb{N}$ sei R_i eine Kurzschreibweise für die Zeichenreihe $R0\ldots0$ (R gefolgt von i Nullen).

a) Für jedes $i \in \mathbb{N}$ sind R_i+ und R_i- WHILE–Programme.

b) Wenn die Zeichenreihen P und Q WHILE–Programme sind und $i \in \mathbb{N}$ ist, dann sind auch

$$\begin{array}{l} (P;Q) \\ (\text{ IF } R_i = 0 \text{ THEN } P \text{ ELSE } Q) \\ (\text{ WHILE } R_i \neq 0 \text{ DO } P) \end{array}$$

WHILE–Programme.

c) Keine anderen Zeichenreihen sind WHILE–Programme.

Definition 4.4.3 (Semantik der WHILE–Programme)
Für jedes WHILE–Programm P sei eine Funktion $\tau(P) :\subseteq D \longrightarrow D$, wobei $D = \mathbb{N}^{\mathbb{N}}$, definiert durch:

a) $\tau(R_i+)(a_0, a_1, \ldots) := (a_0, a_1, \ldots, a_{i-1}, a_i + 1, a_{i+1}, \ldots)$
$\tau(R_i-)(a_0, a_1, \ldots) := (a_0, a_1, \ldots, a_{i-1}, a_i \dot{-} 1, a_{i+1}, \ldots)$

b) $\tau(P;Q) := \tau(Q) \circ \tau(P)$

c) $\tau(\text{ IF } R_i = 0 \text{ THEN } P \text{ ELSE } Q)(d) := \begin{cases} \tau(P)(d) & \text{falls } d(i) = 0 \\ \tau(Q)(d) & \text{sonst} \end{cases}$

d) $\tau(\text{ WHILE } R_i \neq 0 \text{ DO } P)(d)$
$$:= \begin{cases} \tau(P)^t(d) & \text{falls } t := \min\{s \mid \tau(P)^s(d)(i) = 0\} \text{ existiert} \\ \text{div} & \text{sonst.} \end{cases}$$

für alle $i \in \mathbb{N}$ und alle WHILE–Programme P und Q;

e) $f :\subseteq \mathbb{N}^k \longrightarrow \mathbb{N}$ heißt WHILE–berechenbar, gdw. $f = \text{AC} \circ \tau(P) \circ \text{EC}$ für ein WHILE–Programm P gilt (wobei EC und AC die Eingabe– und Ausgabecodierung der k–stelligen Registermaschinen ist).