UNIVERSITÄT KOBLENZ · LANDAU

Theoretische Informatik II

WS 2007/2008

Jun.-Prof. Dr. Bernhard Beckert
Ulrich Koch

1. Teilklausur

11. 12. 2007

Persönliche Daten — bitte gut leserlich ausfüllen!

Vorname: ....

Nachname: ....

Matrikelnummer: ....

Klausurergebnis — bitte nicht ausfüllen!

Aufgabe 1: .... (18 Punkte)

Aufgabe 2: .... (8 Punkte)

Aufgabe 3: .... (6 Punkte)

Aufgabe 4: .... (6 Punkte)

Aufgabe 5: .... (7 Punkte)

Gesamtergebnis: ....

Note: ....

1 Multiple Choice (7+6+5 = 18 Punkte)

Hinweis:
Bei den folgenden Ankreuzaufgaben führen falsche Kreuze zu Punktabzug!
Dabei werden insgesamt jedoch keinesfalls weniger als 0 Punkte für die jeweilige Teilaufgabe vergeben.

(a) Registermaschinen (7 Punkte)

Entscheiden Sie durch Ankreuzen, ob die folgenden Aussagen richtig oder falsch sind.

	richtig	falsch
Im LOOP-Programm `loop` x_1 `do` S `end` kann die Anzahl der Ausführungen von `S` dadurch beeinflusst werden, dass `x1` in `S` geändert wird.	☐	☒
Die `if-then-else`-Anweisung lässt sich mit der `while`-Anweisung simulieren.	☒	☐
Jedes WHILE-Programm terminiert.	☐	☒
Es gibt zu jedem GOTO-Programm ein äquivalentes GOTO-Programm, das nur *einen* Rücksprung enthält. (Dabei ist „`goto` j“ ein Rücksprung, wenn das Sprungziel „$j:$ ...“ syntaktisch *vor* „`goto` j“ im Programm steht.)	☒	☐
Es gibt zu jedem WHILE-Programm ein äquivalentes LOOP-Programm.	☐	☒
Es gibt zu jedem GOTO-Programm ein äquivalentes WHILE-Programm.	☒	☐
Es gibt zu jedem WHILE-Programm eine äquivalente Turing-Maschine.	☒	☐

(b) Primitiv rekursive Funktionen (6 Punkte)

Entscheiden Sie durch Ankreuzen, ob die folgenden Aussagen richtig oder falsch sind.

	richtig	falsch
Für jede natürliche Zahl k ist die Funktion f mit $$f(n_1, \ldots, n_k) = \prod_{i=1}^{k} n_i$$ primitiv rekursiv.	☒	☐
Jede primitiv-rekursive Funktion ist auch total.	☒	☐
Jede totale Funktion ist primitiv-rekursiv.	☐	☒
Die Ackermann-Funktion ist primitiv rekursiv.	☐	☒
Es gibt zu jeder primitiv rekursiven Funktion ein äquivalentes LOOP-Programm.	☒	☐
Es gibt zu jeder primitiv rekursiven Funktion eine äquivalente Turing-Maschine.	☒	☐

(c) μ-rekursive Funktionen (5 Punkte)

Entscheiden Sie durch Ankreuzen, ob die folgenden Aussagen richtig oder falsch sind.

	richtig	falsch
Alle μ-rekursiven Funktionen lassen sich mit nur *einer* Anwendungen des μ-Operators definieren.	☒	☐
Jede primitiv rekursive Funktion ist μ-rekursiv.	☒	☐
Jede totale μ-rekursive Funktion ist primitiv-rekursiv.	☐	☒
Jede μ-rekursive Funktion ist total.	☐	☒
Es gibt zu jedem WHILE-Programm eine äquivalente μ-rekursive Funktion.	☒	☐

2 Registermaschinen 1 (6+2 = 8 Punkte)

(a) Ein WHILE-Programm (im Sinne dieser Aufgabe) hat eine der folgenden Formen:

- x_i `:=` c
- x_i `:=` x_j
- x_i `:=` c *op* x_j
- x_i `:=` x_j *op* c
- x_i `:=` x_j *op* x_k
- P_1 `;` P_2
- `if` x_i `= 0 then` P_1 `end`
- `while` $x_i \neq$ `0 do` P_1 `end`

Dabei ist $op \in \{+, -, *\}$, x_i, x_j, x_k sind Register, c ist eine Konstante aus $\mathbb{N}_0$, und P_1, P_2 sind WHILE-Programme.

Die Funktion $\log : \mathbb{N}_0^2 \to \mathbb{N}_0$ sei wie folgt definiert:

$$\log(x, y) = \begin{cases} \lfloor \log_x y \rfloor & \text{falls } x \geq 2 \text{ und } y \geq 1 \\ \textit{undefiniert} & \text{sonst} \end{cases}$$

i. Geben Sie ein Pseudocode-Programm für log an.

Dieses darf nur die arithmetischen Operatoren $+, -, *$ verwenden. Ansonsten muss es *nicht* den oben genannten Formvorgaben für WHILE-Programme genügen.

Lösung:

`if` $x \leq 1$ `then` *undef*;
`if` $y = 0$ `then` *undef*;
$z := 1$;
`while` $z \leq y$ `do`
  $z := z * x$;
  $r := r + 1$
`end`;
`output` $r \dot{-} 1$

ii. Geben Sie nun ein WHILE-Programm für log an.

Beachten Sie dabei auch, dass alle Register außer dem Ergebnis-Register am Ende der Programmausführung dieselben Werte wie am Anfang enthalten müssen.

Lösung:

$x_4 := x_1 \dot{-} 1$;
`if` $x_4 = 0$ `then` $x_1 := 1$; `while` $x_1 \neq 0$ `do` $x_1 := 1$ `end end`;
`if` $x_2 = 0$ `then while` $x_1 \neq 0$ `do` $x_1 := 1$ `end end`;
$x_4 := 1$;
$x_5 := x_4 \dot{-} x_2$;
$x_5 := 1 \dot{-} x_5$;
`while` $x_5 \neq 0$ `do`
  $x_4 := x_4 * x_1$;
  $x_3 := x_3 + 1$;
  $x_5 := x_4 \dot{-} x_2$;
  $x_5 := 1 \dot{-} x_5$
`end`;
$x_3 := x_3 \dot{-} 1$;
$x_4 := 0$;
$x_5 := 0$

(b) Ein GOTO-Befehl (im Sinne dieser Aufgabe) hat eine der folgenden Formen:

- x_i `:=` c
- x_i `:=` x_j
- x_i `:=` $x_i \pm$ `1`
- `if` x_i `= 0 goto` l

Dabei sind x_i, x_j Register, c ist eine Konstante aus $\mathbb{N}_0$ und l ist ein Label.

Ein GOTO-Programm (im Sinne dieser Aufgabe) hat die Form

$$l_1 : B_1; \ldots ; l_k : B_k \qquad (k \geq 1),$$

dabei sind $B_1, \ldots, B_k$ GOTO-Befehle und $l_1, \ldots, l_k$ paarweise verschiedene Labels.

Geben Sie ein GOTO-Programm an, das für keine Eingabe terminiert und aus höchstens drei Befehlen besteht.

Lösung:

$1: \; x_1 := 0;$
$2: \; \texttt{if } x_1 = 0 \texttt{ goto } 1$

3 Registermaschinen 2 (3+1+2 = 6 Punkte)

Gegeben sei das folgende LOOP-Programm P:

loop x_1 do
  $x_3 := x_3 + 1$
end;
// (1)
loop x_3 do
  $x_2 := x_2 + x_3$
end;
// (2)
$x_3 := 0$

(a) Füllen Sie die folgende Tabelle mit den Werten der Register x_1, x_2, x_3 an den beiden Programmstellen (1) und (2), und zwar jeweils

i. für die Eingabe 3 und

ii. für die Eingabe 5.

Eingabe 3	x_1	x_2	x_3
(1)	3	0	3
(2)	3	9	3

Eingabe 5	x_1	x_2	x_3
(1)	5	0	5
(2)	5	25	5

(b) Welche Werte berechnet P für folgende Eingaben?

i. Eingabe: 3 – Ausgabe: 9........

ii. Eingabe: 5 – Ausgabe: 25........

(c) Welche Funktion $f : \mathbb{N}_0 \to \mathbb{N}_0$ berechnet P?

$$f(n) = n^2$$

4 μ-rekursive Funktionen (6 Punkte)

Zeigen Sie, dass die Funktion

$$\begin{aligned} kgv &: \mathbb{N}_0^2 \to \mathbb{N}_0 \\ kgv(x,y) &= \text{das kleinste gemeinsame Vielfache von } x \text{ und } y \end{aligned}$$

μ-rekursiv ist.

Dabei dürfen Sie alle Funktionen als μ-rekursiv voraussetzen, für die dies in der Vorlesung oder auf einem Aufgabenblatt schon bewiesen wurde, und zudem die folgende Funktion:

$$teilt : \mathbb{N}_0^2 \to \mathbb{N}_0$$

$$teilt(n,m) = \begin{cases} 1 & \text{falls } n \neq 0 \text{ und } n|m \text{ („}n\text{ teilt }m\text{“)} \\ 0 & \text{sonst} \end{cases}$$

Lösung:

$$kgv(x,y) = \mu i((1 \dot{-} (teilt(x,i) * teilt(y,i))) = 0)$$

5 Primitiv rekursive Funktionen (1+3+3 = 7 Punkte)

Gegeben seien die folgenden primitiv rekursiven Funktionen:

- $f_1 = + \circ (\dot{-} \circ (\pi_1^2, c_5^2), * \circ (\pi_2^2, \pi_2^2))$
- $f_2 = \mathcal{PR}[(+1) \circ 0, * \circ ((+1) \circ \pi_1^2, \pi_2^2)]$

Dabei sei für alle $s, k \in \mathbb{N}_0$ und $n_1, \ldots, n_k \in \mathbb{N}_0^k$ definiert: $c_s^k(n_1, \ldots, n_k) = s$

(a) Welche Stelligkeiten haben f_1 bzw. f_2?

Stelligkeit von f_1: 2

Stelligkeit von f_2: 1

(b) Welche Werte berechnen f_1 bzw. f_2, falls alle Argumente...

i. gleich 2 sind?

$f_1(\ldots) = 4$

$f_2(\ldots) = 2$

ii. gleich 3 sind?

$f_1(\ldots) = 9$

$f_2(\ldots) = 6$

(c) Was berechnen f_1 und f_2 im allgemeinen?

$f_1(n, m) = (n \dot{-} 5) + m^2$

$f_2(n) = n!$