

Hausarbeit im Modul „Berechenbarkeit und Komplexität“

Aufgabe 3

Jannik Novak, Giulia Benta, Tobias Phillip Rimkus

13. Januar 2022

Turing-Maechtigkeit von MinWHILE

Beweis durch strukturelle Induktion.

1. Induktionsanfang

Fuer $i, j, k \in \{0, 1, \dots, N_Q\}$ $i \neq j$, $i \neq k$, $k \neq j$
wobei $N_Q + 1$ die Anzahl der benutzten Variablen in einem WHILE-
beziehungsweise MinWHILE-Programm Q bezeichnet.

Jedes WHILE-Programm der Form:

$$(1) \quad x_i = x_j \pm c$$

kann durch das folgende MinWHILE-Programm simuliert werden:

```

1 while  $x_j \neq 0$  do
2    $x_i = x_i + 1$ ;
3    $x_k = x_k + 1$ ;
4    $x_j = x_j - 1$ 
5 end
6  $x_i = x_i \pm c$ ;
7 while  $x_k \neq 0$  do
8    $x_j = x_j + 1$ ;
9    $x_k = x_k - 1$ 
10 end

```

x_k muss sonst ungenutzte Variable sein -0.5p
 hier gibt $x_i = x_i + x_j$, (x_i muss auf null gesetzt werden) -1p

+1.5p

Zu zeigen ist nun, dass das Programm (2) das Programm (1) simuliert. Programm (1) weist der Variable x_i den Wert der Variable x_j addiert/subtrahiert mit der Konstante c zu. Programm (2) durchläuft eine WHILE-Schleife mit Abbruchbedingung $x_j = 0$ und inkrementiert die Variablen x_i und x_k mit jedem Schleifendurchlauf um 1/ dekrementiert die Variable x_j mit jedem Schleifendurchlauf um 1. Somit uebertraegt sich der Wert aus x_j in x_i und x_k . Anschliessend wird x_i mit der Konstante c addiert/subtrahiert und x_i zugewiesen. Somit ergibt sich: $x_i = x_j \pm c$ (FF)
 Zulezt wird der Variable x_j wieder ihr urspruenglicher Wert zugewiesen indem in einer weiteren WHILE-Schleife der Wert der Variablen x_k auf die Variable x_j uebertragen wird. +1p

2. Induktionsvoraussetzung

Sei P ein WHILE-Programm, so existiert ein MinWHILE-Programm P' welches P simuliert.

3. Induktionsschritt

(a) Seien P_1 und P_2 zwei WHILE-Programme. Dann ist auch $P_1;P_2$ ein WHILE-Programm. Wenn P_1' und P_2' MinWHILE-Programme sind die P_1 und P_2 simulieren, so simuliert das MinWHILE-Programm $P_1';P_2'$ das WHILE-Programm $P_1;P_2$. ✓ +1p

(b) Wenn P ein MinWHILE-Programm ist, dann ist auch:

```

1 while  $x_i \neq 0$  do
2   P
3 end

```

typo? In jedem Fall muss man aber ein Min-Programm P' benutzen welches P simuliert.

ein MinWHILE-Programm, welches P ausfuehrt und nach Ausfuehrung von P die Schleifenbedingung $x_i \neq 0$ prueft. ok..?

Simulation einer while schleife geht

-1p

(c) Wenn P ein WHILE-Programm ist, dann ist auch:

LOOP x_i DO P END

ein WHILE-Programm, welches P x_i mal ausfuehrt.

(d) Sei P ein WHILE-Programm und P' ein MinWHILE-Programm welches P simuliert. Betrachten wir nun das WHILE-Programm aus (c):

(1)

LOOP x_i DO P END

Dann koennen wir dieses WHILE-Programm mithilfe des MinWHILE-Konstrukts aus (b) wie folgt simulieren:

(2)

1	while $x_i \neq 0$ do	
2	P';	
3	$x_i = x_i - 1$	
4	end	

Zu zeigen ist nun, dass das Programm (2) das Programm (1) simuliert. Programm (1) fuehrt das Programm P x_i mal aus. Programm (2) prueft vor jeder Ausfuehrung ob $x_i \neq 0$ und bricht ab sobald $x_i = 0$. In jedem Schleifendurchlauf wird die Variable x_i um 1 dekrementiert, solange bis die Abbruchbedingung $x_i = 0$ erreicht ist. Somit wird das Programm P' x_i mal ausgefuehrt. Da P' das Programm P simuliert, gilt die obige Aussage.