TU Ilmenau, Fachgebiet Automaten und Logik
Prof. Dr. Dietrich Kuske,
M.Sc. Christopher Hugenroth, M.Sc. Sebastian Pfau
SS 2024

TECHNISCHE UNIVERSITÄT
ILMENAU

Berechenbarkeit und Komplexität – Übung 2

Abgabe: bis Freitag, der 03. Mai 2024, um 11:00 Uhr am Fachgebiet oder vor der Übung.

Geben Sie bitte Ihre Matrikelnummer an.
Heften Sie zudem alle Ihre Lösungsblätter geeignet zusammen.

Bonusaufgaben

Bonusaufgaben können schriftlich bearbeitet werden. Ihre Lösungen geben Sie bitte bis zum oben angegebenen Termin ab. Die Abgaben werden von uns korrigiert und die erreichten Punkte werden mittels eines Faktors in Bonuspunkte für die Klausur umgerechnet.

Aufgabe 1* — 3 Punkte

Sei $a : \mathbb{N}^2 \to \mathbb{N}$ mit $a(k,n) := \mathrm{ack}(k,n) \bmod 2$, d.h., $a(k,n)$ ist die Parität des Wertes $\mathrm{ack}(k,n)$.
Zeigen oder widerlegen Sie, dass die Funktion a Loop-berechenbar ist.

Aufgabe 2* — 2 Punkte

Bestimmen Sie den Wert $f_P(3)$ (vgl. Vorlesung) für das folgende Loop-Programm P.

```
x3 := x2;
x1 := x1 + 1;
loop x1 do
    x2 := x2 + x3;
    x1 := x1 + 1
end;
x3 := 0
```

Aufgabe 3* — 2+2+2 Punkte

Bearbeiten Sie die folgenden Teilaufgaben.

(a) Zeigen Sie, dass für jedes Loop-Programm P ohne Loop-Schleife ein $k \in \mathbb{N}$ existiert, sodass $f_P(n) \leq n + k$ für alle $n \in \mathbb{N}$.

(b) Zeigen Sie, dass für jedes Loop-Programm P mit höchstens einer Loop-Schleife ein $k \in \mathbb{N}$ existiert, sodass $f_P(n) \leq k \cdot n + k$ für alle $n \in \mathbb{N}$.

(c) Zeigen Sie, dass es eine Loop-berechenbare Funktion f gibt, sodass jedes Loop-Programm, das f berechnet mindestens 2 Loop-Schleifen braucht.

Hinweis: Orientieren Sie sich am Beweis vom Beschränkungslemma (Folie 3.8f).

Aufgabe 4* — 4 Punkte

Loop-Programme mit Stack verfügen über alle Befehle, die Loop-Programme besitzen. Zusätzlich besitzen sie einen Stack auf den mit dem Befehl $\mathrm{push}(x_i)$ der Wert von x_i gelegt werden kann und von dem mit dem Befehl $x_i = \mathrm{pop}$ der oberste Wert ausgelesen werden kann, welcher dann gleichzeitig vom Stack entfernt wird. Falls der Stack leer ist gibt pop den Wert 0 zurück.
Begründen Sie, dass für jede Funktion, die von einem Loop-Programm mit Stack berechnet wird, ein Loop-Programm (ohne Stack) existiert, welches die gleiche Funktion berechnet. Beschreiben Sie dazu, wie die Befehle $\mathrm{push}(x_i)$ und $x_i = \mathrm{pop}$ in einem Loop-Programm ohne Stack umgesetzt werden können.

Präsenzaufgaben

Aufgabe 5

Bearbeiten Sie die folgenden Teilaufgaben. Geben Sie jeweils ein kurze Begründung Ihrer Antwort mit an.

(a) Geben Sie eine möglichst einfache Beschreibung der Funktion $e\colon \mathbb{N} \to \mathbb{N}$ an, welche durch das folgende GOTO-Programm berechnet wird.

1 **if** $x_1 = 0$ **then** 6;
2 **if** $x_2 = 0$ **then** 9;
3 $x_1 := x_1 \dot{-} 1$;
4 $x_2 := x_2 \dot{-} 1$;
5 **goto** 1;
6 **if** $x_2 = 0$ **then** 12;
7 $x_1 := 0$;
8 **goto** 0;
9 **if** $x_1 = 0$ **then** 12;
10 $x_1 := 0$;
11 **goto** 0;
12 $x_1 := 1$;
13 **goto** 0;

(b) Geben Sie eine möglichst einfache Beschreibung der Funktion $d\colon \mathbb{N} \to \mathbb{N}$ an, welche durch das folgende WHILE-Programm berechnet wird.

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

(c) Wir betrachten das μrek-Programm

$$F = \mathrm{SEARCH}(\mathrm{SUBST}(\mathrm{SUB}; \mathrm{PROJ}_1^1, \mathrm{REC}(\mathrm{CONST}_1^0, \mathrm{SUBST}(\mathrm{MULT}; \mathrm{CONST}_2^2, \mathrm{PROJ}_1^2)))) \,.$$

Geben Sie eine möglichst einfache Beschreibung von $[\![F]\!]$ an.

Aufgabe 6

WHILE-Programme mit Stack sind definiert wie LOOP-Programme mit Stack nur, dass jede LOOP-Schleife durch eine WHILE-Schleife ersetzt wird. Bearbeiten Sie die folgenden Teilaufgaben.

(a) Gibt es für jede Funktion, die von einem WHILE- Programm mit Stack berechnet wird ein WHILE-Programm (ohne Stack), welches die gleiche Funktion berechnet? Begründen Sie ihre Antwort.

(b) Geben Sie ein WHILE-Programm mit Stack an, das die Ackermann-Funktion berechnet.