

Abitur 2018 III Aufgabe 4

(Abitur 2018)

Stichwörter: Ein-Adress-Befehl-Assembler

(a) Gegeben ist folgendes Programm:

```
load 100
cmp 101
jmpnn 12
load 101
store 102
jmp 14
store 102
hold
```

Der Zustand der Registermaschine wird im Folgenden durch die Inhalte des Akkumulators A, des Befehlszählers BZ, des Statusregisters SR sowie der Speicherzellen 100 bis 102 beschrieben.

Veranschaulichen Sie die durchlaufenen Zustände bei der Ausführung des Programms anhand einer geeigneten Tabelle. Gehen Sie von folgendem Anfangszustand aus: Der Befehlszähler BZ enthält den Wert 0, die Speicherzelle 100 den Wert 4 und die Speicherzelle 101 den Wert 5. Geben Sie an, was das Programm in Abhängigkeit von den Startwerten in den Speicherzellen 100 und 101 leistet.

Gegeben ist folgendes Struktogramm für die Methode $c(n)$ für natürliche Zahlen $n \geq 3$.

(b) Übertragen Sie diesen Algorithmus der Methode c in ein Programm für die gegebene Registermaschine. Machen Sie auch die Speicherzelle deutlich, in der der Wert der Variablen n zu Beginn und am Ende des Programms steht.

Assembler

```
# n := n / 2 + 1;
                                LOAD 100
                                DIVI 2
                                ADDI 1
                                STORE 100
                                JMP solange
vorbereitung:  LOADI 15
                                STORE 100

# WHILE n >= 3 DO
solange:      LOAD 100
                                CMPI 3
                                JMPN ende
                                # n := (n + 1) / 2;
                                falsch_durch_2: LOAD 100
                                ADDI 1
                                DIVI 2
                                STORE 100
                                JMP solange
# IF (n % 4) = 0 THEN
                                MODI 4
                                JMPP falsch_durch_4
                                # n := n / 4;
                                LOAD 100
                                DIVI 4
                                STORE 100
                                JMP solange
                                ende:      HOLD
                                n:          WORD 15

# IF (n % 2) = 0 THEN
falsch_durch_4: LOAD 100
                                MODI 2
                                JMPP falsch_durch_2
```

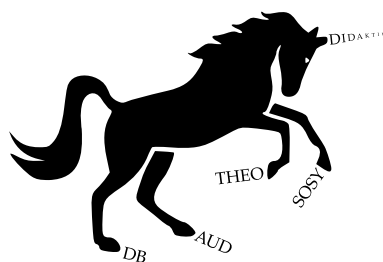
Minisprache

```
PROGRAM abi;
VAR n;

BEGIN
  n := 15;
  WHILE n >= 3 DO
    IF (n % 4) = 0 THEN
      n := n / 4;
```

```
ELSE
  IF (n % 2) = 0 THEN
    n := n / 2 + 1;
  ELSE
    n := (n + 1) / 2;
  END;
END;
END
END abi.
```

- (c) Versehentlich wurde die Bedingung $n \geq 3$ bei der Implementierung des Algorithmus durch $n \geq 2$ ersetzt. Erläutern Sie kurz, welches Problem bei der Ausführung des Programms auftreten kann.



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net. Der TeX-Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Module/50_TECH/10_Ein-Adress/Aufgabe_Abitur-2018-III.tex