

## 1. Hausaufgabenblatt

### 1. Eigenschaften von Funktionen

30 Punkte

Geben Sie jeweils eine Funktion mit den geforderten Eigenschaften an, oder begründen Sie, dass eine solche Funktion nicht existiert. Geben Sie zu jeder Funktion auch ihren Graph als Menge an.

- (a) surjektives  $f_1 : \{1\} \rightarrow \{0, 1\}$ , das nicht injektiv ist
- (b) totales  $f_2 : \{0, 1, 2\} \rightarrow \{1, 3\}$ , das nicht surjektiv ist
- (c)  $f_3 : \{5\} \rightarrow \{0\}$ , das nicht total ist
- (d) bijektives  $f_4 : \{0, 1\} \rightarrow \{0, 1, 2\}$
- (e) totales  $f_5 : \emptyset \rightarrow \{0, 1\}$
- (f) surjektives  $f_6 : \{0, 1, 2\} \rightarrow \{0, 1, 2\}$ , das nicht injektiv ist

### 2. Syntax von While-Programmen

30 Punkte

Der unten stehende *fehlerhafte* Quelltext ist ein Versuch, ein While-Programm zu schreiben, das folgende Funktion  $f : \mathbb{Z} \rightarrow \mathbb{Z}$  berechnet:

$$f(x) = \begin{cases} 2^x & \text{falls } x \geq 0 \\ 0 & \text{falls } x < 0 \end{cases}$$

Nennen und begründen Sie alle Fehler des Quelltexts. Berichtigen Sie die Fehler, sodass ein gültiges While-Programm entsteht, das die Funktion  $f$  berechnet.

```
def g(f):  
    return f*2  
  
def f(x):  
    if x<0  
        return 0  
    else if x=0  
        return 1  
    else {  
        d = g(f(x-1));  
        return d  
    }
```

### 3. While-Programm für Binärdarstellung

40 Punkte

while

Gesucht ist ein kommentiertes While-Programm (mit Syntaxprüfer testen!), das die Binärdarstellung einer natürlichen Zahl  $n$  auf dem Bildschirm ausgibt. Gehen Sie dazu wie folgt vor.

- (a) Schreiben Sie ein While-Programm für die Funktion  $\text{divtwo}(x) = \begin{cases} \lfloor \frac{x}{2} \rfloor & \text{falls } x \geq 0 \\ 0 & \text{sonst} \end{cases}$ .

Verwenden Sie dabei nicht die Funktionen `prodZ` und `divZ`.

- (b) Verwenden Sie die Funktion `divtwo` und schreiben Sie ein While-Programm, das für Eingaben  $n \geq 0$  die Ziffern der Binärdarstellung von  $n$  nacheinander auf dem Bildschirm ausgibt. Ist  $n < 0$ , so soll 0 ausgegeben werden.

Abgabetermin: **Montag, 27.04.2026, 12:00 Uhr**

Lösungen müssen von einer Person je Abgabegruppe im WueCampus hochgeladen werden.

Jede Aufgabe in eine eigene Datei, die die Namen aller an der Lösung beteiligten Personen enthält.

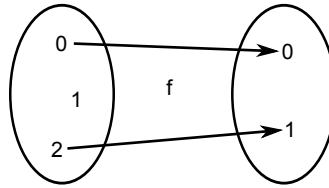
Begründen Sie Ihre Behauptungen. Verwenden Sie nur Definitionen, Notationen, Resultate und Verfahren aus der Vorlesung. Verwenden Sie keine externen Quellen (z.B. Wikipedia, ChatGPT usw.).

Aufgaben-Rating: ohne Chili = Wissen, 🌶️ = Anwenden, 🌶️🌶️ = Verallgemeinern, 🌶️🌶️🌶️ = schwierig

# Lösungshinweise

## Aufgabe 1:

Sie können die gesuchte Funktion auch durch eine Zeichnung angeben. Beispielsweise zeigt folgende Zeichnung eine surjektive Funktion  $f : \{0, 1, 2\} \rightarrow \{0, 1\}$ , die nicht total ist. Der Graph der Funktion ist  $G_f = \{(0, 0), (2, 1)\}$ .



## Aufgabe 2:

Im WueCampus finden Sie einen *Syntaxprüfer für While- und Loop-Programme*. Installieren Sie diesen auf ihrem Computer (die Dokumentation enthält eine Installationsanleitung). Mit Hilfe des Syntaxprüfers lässt sich die Aufgabe schnell lösen.

## Aufgabe 3:

(a) Testen Sie die Syntax Ihres Programms mit dem Syntaxprüfer! Insbesondere darf das Programm keine Multiplikation und keine Division verwenden (auch nicht die im Skript behandelten Funktionen `prodZ` und `divZ`). Für natürliche  $x$  ist `divtwo(x)` die größte Zahl  $y$  mit der Eigenschaft  $2y \leq x$ . Versuchen Sie, diese Beschreibung in ein Programm umzuwandeln. Ihr Programm soll eher leicht verständlich (kurz) als effizient (schnell) sein.

(b) Testen Sie die Syntax Ihres Programms mit dem Syntaxprüfer! Beachten Sie Eigenschaft 1.5 und Beispiel 1.6. Zur Ausgabe einer natürlichen Zahl (Binärziffer) dürfen Sie den Befehl `print(x)` verwenden. Sie können die Ziffern der Binärdarstellung in der Reihenfolge von rechts nach links, oder von links nach rechts ausgeben (die Ziffern können auf dem Bildschirm auch untereinander stehen). Der Rückgabewert Ihres Programms kann eine beliebige Zahl sein (z.B.  $-1$ ). Ihr Programm könnte folgende Form haben.

```
def divtwo(x):  
    ...  
    return ...  
  
def bin(n):  
    if (n <= 0):  
        print(0)  
    else:  
        ...  
        # hier könnte divtwo(...) aufgerufen werden  
        ...  
        # hier könnte print(...) aufgerufen werden  
        ...  
    return -1
```