| 7. Übungsblatt | Diskrete Strukturen | WS 2012/13 |
|---|---|---|

**Abgabe bis Freitag, 07.12.2012, 8 Uhr beim DS-Kasten im 4. OG vor Sekretariat Näher.**
**Die Aufgaben werden in derselben Woche in den Übungen besprochen.**

**1. Aufgabe:** (*3 Punkte*)
Beweisen Sie:

> Sind die Relationen $R \subseteq M \times N$ und $S \subseteq N \times O$ Funktionen, so ist auch $R \circ S \subseteq M \times O$ eine Funktion.
> Sind überdies $R$ und $S$ Surjektionen, so auch $R \circ S$.

**2. Aufgabe:** (*4 Punkte*)
Betrachten Sie die folgenden Relationen zwischen $A = \{0, 2, 4, 6\}$ und $B = \{a, b, c, d\}$:

1. $R_1 = \{(6, b), (2, a), (0, b), (4, c)\}$;
2. $R_2 = \{(2, b), (4, d), (0, a), (6, c)\}$;
3. $R_3 = \{(2, a), (4, c), (6, b)\}$;
4. $R_4 = \{(6, a), (0, b), (4, a), (0, d), (2, c)\}$.

Welche dieser Relationen sind partielle Funktionen oder auch sogar Funktionen (und welche nicht)?
Im “Funktionen-Fall”: Welche sind injektiv, surjektiv oder sogar bijektiv (und welche nicht)?

**3. Aufgabe:** (*3+1 Punkte*)
Die Funktionen $f : \mathbb{Q} \to \mathbb{Q}$ und $g : \mathbb{Q} \to \mathbb{Q}$ seien definiert durch:

$$f(x) = x^2 \quad \text{und} \quad g(x) = \begin{cases} 3x - 1 & \text{wenn } x \geq 0 \\ -x & \text{sonst} \end{cases}$$

Geben Sie Formeln an für $f \circ g$, $f \circ g \circ f$ und $g \circ g$.
Zeigen Sie: $f$ ist weder injektiv noch surjektiv.

Fortsetzung auf der nächsten Seite!

**4. Aufgabe:** *(1+3+2+3 Punkte)*

Angenommen, wir hätten eine Programmiersprache zur Verfügung, die als (einzigen) Datentyp die Menge $\mathbb{Z}$ der ganzen Zahlen mit den üblichen Operationen $+$, $-$ und $*$ anbietet, aber eben auch tatsächlich die Darstellung beliebig großer ganzer Zahlen gestattet. (Praktisch müsste das evtl. so aussehen, dass beim Lauf Ihres Programms möglicherweise eine Meldung angezeigt wird, dass das Programm nur dann weiterarbeiten kann, wenn Sie eine weitere Festplatte anschließen. In der Theoretischen Informatik müssen uns diese eher technischen Details nicht kümmern.) Sie dürfen z.B. mit $\mathbb{Z}$ `x,y;` die Variablen `x` und `y` deklarieren; mit `x` $\leftarrow$ **12;** `y` $\leftarrow$ `x` $*$ `x` $-$ **4;** bekommt `x` den Wert 12 zugewiesen, und `y` den Wert 140. Als weiteres syntaktisches Element sind Schleifen der Form `WHILE (z) DO { ... };` zulässig, wobei der Schleifenrumpf `...` solange ausgeführt wird, wie die Variable `z` einen Wert ungleich Null besitzt. Schließlich können Sie mit `GET x;` eine Zahleingabe vom Programmbenutzer einfordern und diese der Variablen `x` zuweisen sowie mit `PUT x;` den aktuellen Wert der Variablen `x` ausgeben.

Beispielsweise steht rechts ein Programm $\boldsymbol{P}$, das zu einer eingegebenen Zahl $\boldsymbol{n}$ die Zahl $\mathbf{2}^{\boldsymbol{n}}$ ausrechnet, sofern $\boldsymbol{n} \geq \mathbf{0}$:

```
ℤ x,y;
GET x;
y ← 1;
WHILE (x) DO {
x ← x − 1;
y ← 2 * y;
};
PUT y;
```

1. Was macht das soeben angegebene Programm $\boldsymbol{P}$, wenn der Benutzer eine negative Zahl eingibt?
2. Argumentieren Sie: Welche partielle Funktion entspricht also dem Programm $\boldsymbol{P}$ und warum?
3. Benutzen Sie die soeben eingeführten syntaktischen Elemente, um folgende Entwurfsentscheidung zu rechtfertigen: Die neue Version unserer einfachen Programmiersprache soll noch einfacher werden, und es wird auf die Multiplikation als vorgegebener Operation verzichtet. Wie kann man also "trotzdem" zwei Zahlen multiplizieren? Geben Sie dazu ensprechenden Programmcode an.
4. $\boldsymbol{f} : \mathbb{Z} \to \mathbb{Z}$ sei eine bijektive Funktion. Ihr Freund Fritz hat eine entsprechende Funktion programmiert, sodass Sie beispielsweise mit `x` $\leftarrow$ $\boldsymbol{f}$`(y);` der Variablen `x` den Wert zuweisen können, den die Funktion $\boldsymbol{f}$ liefert, wenn man sie auf den aktuellen Wert der Variablen `y`, z.B. 140, anwendet. Wie könnte man unter Verwendung der Hilfsfunktion $\boldsymbol{f}$ ein Programm schreiben, welches nach Einlesen einer Zahl $\boldsymbol{n}$ den Wert von $\boldsymbol{f}^{-1}(\boldsymbol{n})$ ausrechnet und ausgibt?