

## Übungen zur Semantik von Programmiersprachen

---

### Aufgabe 1 (H) (*Anwendung des Knaster-Tarski Fixpunkt-Theorems*)

**Abgabe: Montag, 21. 1. 2002**

Für eine Funktion  $h : M \rightarrow N$  ist die Erweiterung  $\tilde{h} : \wp(M) \rightarrow \wp(N)$  folgendermaßen definiert:

$$\tilde{h}(M_1) = \{n \mid \exists m \in M_1. h(m) = n\}$$

für  $M_1 \subseteq M$ .

- (a) Zeigen Sie die Gültigkeit des Banach'schen Dekompositionstheorems:

Gegeben seien zwei beliebige Mengen  $X$  und  $Y$  und zwei beliebige Funktionen  $f : X \rightarrow Y$  und  $g : Y \rightarrow X$ .

Dann gibt es  $X_1, X_2 \subseteq X$  mit  $X_1 \cup X_2 = X$  und  $X_1 \cap X_2 = \emptyset$  und  
 $Y_1, Y_2 \subseteq Y$  mit  $Y_1 \cup Y_2 = Y$  und  $Y_1 \cap Y_2 = \emptyset$ , sodaß

$$\tilde{f}(X_1) = Y_1 \tag{1}$$

$$\tilde{g}(Y_2) = X_2 \tag{2}$$

Gehen Sie dazu folgendermaßen vor:

- i) Stellen Sie den Inhalt des Theorems zeichnerisch dar.
  - ii) Geben Sie eine Funktion  $\Phi : \wp(X) \rightarrow \wp(X)$  unter Zuhilfenahme der Eigenschaften (1) und (2) an, so daß  $X_2$  ein Fixpunkt von  $\Phi$  ist.
  - iii) Weisen Sie die Existenz eines Fixpunkts von  $\Phi$  nach.
- (b) Zeigen Sie nun das Theorem von Schröder-Bernstein:

Gegeben seien zwei beliebige Mengen  $X$  und  $Y$ . Wenn es zwei injektive Funktionen  $f : X \rightarrow Y$  und  $g : Y \rightarrow X$  gibt, dann gibt es eine Bijektion  $b : X \rightarrow Y$ .

*Hinweis:* Konstruieren Sie die Funktion  $b$  unter Verwendung des Theorems aus (a).

### Aufgabe 2 (H) (*Beweis im Hoare-Kalkül*) **Abgabe: Montag, 21. 1. 2002**

Zeigen Sie, dass gilt

```
{a ≥ 0}
n = 0; m = 0; k = 0;
while k < a do
    (n = n + 1; k = k + m + 1; m = m + 2)
{n - 1 < √a ≤ n}
```

Hinweis: Die Invariante der Schleife ist etwas trickreich. Berechnen Sie ein paar einfache Beispiele und versuchen Sie herauszufinden, in welcher Beziehung  $m$ ,  $n$  und  $k$  zueinander stehen.

### Aufgabe 3 (Ü) (*Invarianten*)

Berechnen Sie die Werte von *pre* und *vc* für die Nachbedingung  $\{n - 1 < \sqrt{a} \leq n\}$  und das Programm aus Aufgabe 2.

Benutzen Sie dazu

- (a) die richtige Invariante
- (b) eine zu schwache Invariante
- (c) eine falsche Invariante

### Aufgabe 4 (Ü) (*Normalisierung von Hoare-Beweisen*)

Die Regeln des Hoare-Kalküls sind bis auf die Konsequenzregel syntaxgesteuert (es existiert jeweils genau eine Regel für jedes Statement). Obwohl die Konsequenzregel prinzipiell immer anwendbar ist, kann man sich in der Beweissuche jedoch auf bestimmte Situationen beschränken.

Wir betrachten dazu folgendes System *K* aus der Semikolon-Regel, der **if**-Regel und den abgeleiteten Regeln

$$\frac{P \implies Q}{\{P\} \text{ skip } \{Q\}} \quad \frac{P \implies Q[a/x]}{\{P\} x := a \{Q\}} \\ \frac{P \implies I \quad \{I \wedge b\} s \{I\} \quad I \wedge \neg b \implies Q}{\{P\} \text{ while } b \text{ do } s \{Q\}}$$

Zeigen Sie nun folgende Behauptung: Jeder Beweis im Hoare-Kalkül kann in einen Beweis im System *K* überführt werden. Formal:  $\vdash \{P\}s\{Q\} \implies \vdash_K \{P\}s\{Q\}$

### Aufgabe 5 (P) (*Nichtdeterminismus*)

**Abgabe: Montag, 21. 1. 2002 an kleing@in.tum.de**

Erweitern Sie die denotationelle Semantik von WHILE in Gofer aus Aufgabe 5, Blatt 9, um das nichtdeterministische Statement

$$s_1 \text{ or } s_2$$

Ändern Sie ihr Programm dazu so, dass zu einem Statement und einem Zustand nicht nur ein Folgezustand berechnet wird, sondern die Liste aller möglichen Folgezustände. Testen Sie ihr Programm mit den beiden Anweisungen

**while**  $\neg(x = 0)$  **do**  $(x := 0 \text{ or skip})$

und

**while**  $\neg(x = 0)$  **do**  $(\text{skip or } x := 0)$

### Prüfung:

Die Semestralprüfung wird vorraussichtlich am 6. 2. 2002 in Form einer mündlichen Prüfung stattfinden. Wenn Sie einen Schein erwerben möchten, melden Sie sich bitte spätestens bis zum 28. 1. 2002 per Email an kleing@in.tum.de an.