fhwedel
UNIVERSITY OF APPLIED SCIENCES

Aufgabe 1)

Betrachten Sie folgendes Programm:

```
{ n,s,k ∈ N }          φ

s := 1;
k := n;

while(k > 0) do
  begin
     k := k - 1;
     s := k + s;
  end

{ Nachbedingung }      ψ
```

a) Formulieren Sie die Belegungswerte, die für jeden Schleifendurchlauf gültig sind. Beweisen Sie das durch vollständige Induktion.

b) Zeigen Sie, dass dann die Schleife irgendwann terminiert (wann genau?). Formulieren und beweisen Sie direkt unter Verwendung von a), was diese dann berechnet hat.

c) Verändern Sie das Programm so, dass es die Summe von 1 bis n berechnet berechnet. Sie dürfen dafür die Initialisierungswerte oder die Abbruchbedingung ändern, nicht aber die Reihenfolge der Anweisungen.

Aufgabe 2)

Betrachten Sie folgendes Programm:

```
Gegeben seien n Zahlen a[1] ... a[n] ∈ ℚ.

k := 1;
while (k < n) do
  begin
      k := k + 1;
      d := a[k] – a[k-1];
      if m > d
        then
            m := d;
  end
```

a) Geben Sie eine Nachbedingung für m an! Brauchen Sie Vorbedingungen dafür?

b) Bestimmen Sie die Invariantenbedingungen m_i, k_i und d_i an, die nach jedem Schleifendurchlauf erfüllt sind und beweisen Sie das mit vollständiger Induktion!

c) Beweisen Sie, dass die Schleife terminiert und folgern Sie dann die in a) angegebene Nachbedingung!

d) Ändern Sie das Programm so ab, dass es den Betrag des kleinsten Abstands ausgibt, der zwischen zwei hintereinander folgenden Zahlen auftreten kann!

Aufgabe 3)

Gegeben sei folgendes Programm:

```
{n: integer}

k := 0; s := 0;

while k ≤ n do
begin
  k := k + 1;
  s := k - s;
end {while}
```

a) Was berechnet dieses Programm? Geben Sie die genaue Abhängigkeit von n an!
b) Beweisen Sie a)!

Tipp: Unterscheiden Sie für den i-ten Schleifendurchlauf, ob i eine gerade oder eine ungerade Zahl ist.

Aufgabe 4)

```
{ m ∈ Z, n ∈ N }                                          φ

   result := 1;
   base := m;
   exp := n;

   while exp > 0
     begin
       if (exp MOD 2 = 1) then
         begin
           result := result * base;
           exp := exp - 1;
         end
       else
         begin
           base := base * base;
           exp := exp DIV 2;
         end;
     end; {while}

{ result = m^n }                                          ψ
```

a) Beweisen Sie mit vollständiger Induktion über k: $m^n = base_k^{\exp_k} \cdot result_k$ (Hierbei sind $base_k$, exp_k und $result_k$ die Werte der Variablen nach dem k-ten Schleifendurchlauf)

b) Folgern Sie daraus, dass das Programm mit der gewünschten Nachbedingung stoppt.