

Aufgabe 1)

Betrachten Sie folgendes Programm:

```
{ n, f, k ∈ N }      φ

f := 1;
k := n;

while(k > 0) do
  begin
    k := k - 1;
    f := k • f;
  end

{ Nachbedingung }    ψ
```

- Formulieren Sie die Invariantenbedingungen, die für jeden Schleifendurchlauf gültig sind. Beweisen Sie das durch vollständige Induktion.
- Zeigen Sie, dass dann die Schleife irgendwann terminiert (wann genau?). Formulieren und beweisen Sie direkt unter Verwendung von a), was diese dann berechnet hat.
- Verändern Sie die Abbruchbedingung der Schleife so, dass das Programm eine Fakultät berechnet. Welche Fakultät berechnet das Programm genau? Beweisen Sie auch das unter Verwendung von a)

Aufgabe 2)

Betrachten Sie folgendes Programm:

Gegeben seien n Zahlen $a[1] \dots a[n] \in \mathbb{Q}$.

```

m := 0;
k := 0;
while (k < n) do
  begin
    m := k * m;
    k := k + 1;
    m := (m + a[k])/k ;
  end

```

- Geben Sie eine Nachbedingung für m an! Brauchen Sie Vorbedingungen dafür?
- Geben Sie die Invariantenbedingungen m_i und k_i an, die nach jedem Schleifendurchlauf erfüllt sind!
- Geben Sie an, nach wie vielen Durchläufen die Schleife abbricht.
- Beweisen Sie die Gültigkeit der Invariantenbedingung und des in c) festgestellten Werts für die Durchläufe mit vollständiger Induktion und folgern Sie daraus die Nachbedingung.

Aufgabe 3)

$\{ b \in \mathbb{Z}, e \in \mathbb{N} \}$ φ

```

result := 1;
base := b;
exp := e;

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}

```

$\{ result = b^e \}$ ψ

- a) Beweisen Sie mit vollständiger Induktion über k : $b^e = base_k^{exp_k} \cdot result_k$
(Hierbei sind $base_k$, exp_k und $result_k$ die Werte der Variablen nach dem k -ten Schleifendurchlauf)
Tipp: Unterscheiden Sie in Ihrem Induktionsbeweis die beiden Fälle, ob exp_k gerade ist oder nicht!
- b) Folgern Sie daraus, dass das Programm mit der gewünschten Nachbedingung stoppt.

Aufgabe 4)

```
procedure f(n): integer
begin
  if (n = 0)
    return n
  else
    return n * f(n-1);
end;
```

- a) Geben Sie eine Vorbedingung an, bei der dieses Programm¹ terminiert! Was berechnet das Programm unter dieser Vorbedingung?
- b) Beweisen Sie Ihre Antwort auf die Frage in a) mit vollständiger Induktion nach n .

Aufgabe 5) * (für Einserkandidaten)**

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.

¹ Die Syntax und Funktionsweise entspricht der aus dem Foliensatz GTI2Prozeduren (Übersicht zu Kap. 2.4), welche nicht mit Pascal identisch ist.