

Prof. Dr. Moritz Sinn  
Julian Winter M.Sc.

## Praktikum Zuverlässige Softwaresysteme

*Allgemeiner Hinweis:* In diesem Praktikum verwenden wir *Dafny*<sup>1</sup>. Um das Praktikum zu bestehen, müssen Sie mindestens 4 der 5 Aufgaben erfolgreich bearbeiten.

**Vor dem Praktikumstermin:** Gehen Sie die in der Vorlesung behandelten Beispiele  $F(x)$  bis  $L(x)$  sowie  $M(x, b)$  und  $N(x, y, b)$  durch (siehe Buch *Program Proofs*, Kapitel 3, die Beispiele finden sich als Code im Ilias) und stellen Sie sicher, dass Sie verstehen, warum diese Beispiele terminieren und wie man dies beweisen kann.

### Aufgabe 1    Rekursion und Terminierung

Gegeben ist folgende Funktion:

```
function G(n: nat): nat
{
  if n == 0 then 0 else n - G(G(n - 1))
}
```

- (a) Dafny findet automatisch das Terminierungsargument für  $G(n)$ . Welches? Warum bzw. wie beweist der von Dafny gefundene Ausdruck die Terminierung des Beispiels?

- (b) Wir definieren die Funktion  $G'(n)$  wie folgt:

```
function G'(n: nat): nat
{
  if n == 0 then 1 else n - G'(G'(n - 1))
}
```

terminiert  $G'(n)$ ? Warum, bzw. warum nicht?

### Aufgabe 2    Rekursion und Terminierung

- (a) Zeigen Sie, dass die folgende rekursive Funktion terminiert, in dem Sie einen entsprechenden *nicht*-lexikographischen *decreases*-Ausdruck ergänzen.

---

<sup>1</sup><https://dafny.org>

```

function K(x: nat, y: nat, z: nat): int
  // decreases ...
{
  if x < 10 ∨ y < 5 then
    x + y
  else if z = 0 then
    K(x - 1, y, 5)
  else
    K(x, y - 1, z - 1)
}

```

- (b) Zeigen Sie, dass die folgende rekursive Funktion terminiert, in dem Sie einen entsprechenden *decreases*-Ausdruck ergänzen.

```

function K'(x: nat, y: nat, z: nat): int
  // decreases ...
{
  if x < 10 ∨ y < 5 then
    x + y
  else if z = 0 then
    K'(x - 1, y-2, 5)
  else
    K'(x, y + 1, z - 1)
}

```

### Aufgabe 3    Wechselseitige Rekursion und Terminierung I

- (a) Für die folgenden wechselseitig rekursiven Funktionen findet Dafny automatisch das Terminierungsargument. Wie lautet es und warum bzw. wie beweist es die Terminierung?

```

method RequiredStudyTime(c: nat) returns (hours: nat)
  //keine Implementierung

method Outer(a: nat)
{
  if a ≠ 0 {
    var b := RequiredStudyTime(a - 1);
    Inner(a, b);
  }
}

method Inner(a: nat, b: nat)
  requires 1 ≤ a
{
  if b = 0 {
    Outer(a - 1);
  } else {
    Inner(a, b - 1);
  }
}

```

- (b) Zeigen Sie, dass die folgenden wechselseitig rekursiven Funktionen terminieren, in dem Sie für jede der beiden Methoden einen entsprechenden *decreases*-Ausdruck ergänzen.

```
method RequiredStudyTime(c: nat) returns (hours: nat)
    //keine Implementierung

method Outer(a: nat)
    //decreases ...
{
    if a ≠ 0 {
        var b := RequiredStudyTime(a - 1);
        Inner(a - 1, b);
    }
}

method Inner(a: nat, b: nat)
    //decreases ...
{
    if b = 0 {
        Outer(a);
    } else {
        Inner(a, b - 1);
    }
}
```

#### Aufgabe 4    Wechselseitige Rekursion und Terminierung II

Zeigen Sie, dass die folgenden wechselseitig rekursiven Funktionen terminieren, in dem Sie für jede der beiden Methoden eine entsprechenden *decreases*-Ausdruck ergänzen.

```
function F(n: nat): nat
    // decreases ...
{
    if n = 0 then 1 else n - M(F(n - 1))
}

function M(n: nat): nat
    // decreases ...
{
    if n = 0 then 0 else n - F(M(n - 1))
}
```

### Aufgabe 5    Rekursion und Terminierung

Zeigen Sie, dass die folgende rekursive Funktion terminiert, in dem Sie einen entsprechenden *decreases*-Ausdruck ergänzen.

```
function F(x: nat, y: nat): int
{
  if 1000 ≤ x then
    x + y
  else if x % 2 = 0 then
    F(x + 2, y + 1)
  else if x < 6 then
    F(2 * y, y)
  else
    F(x - 4, y + 3)
}
```