

11. Übungsblatt

Ausgabe: 26.06.22**Abgabe:** 03.07.22

11.1 Größer gemeinsamer Teiler(Finale)

Der größte gemeinsame Teiler kann auch rekursiv berechnet werden. In C sieht der rekursive Algorithmus so aus:

```
1 int ggt(int a, int b)
2 /** pre 0 ≤ a ∧ 0 ≤ b;
3     post \result == gcd(a,b);
4     */
5 {
6     int r;
7
8     if (a == b) return b;
9
10    if (a <= b) {
11        /** const ?? */ r= ggt(b- a, a);
12    } else {
13        /** const ?? */ r= ggt(b, a-b);
14    }
15    return r;
16 }
```

- (i) Annotieren Sie das Programm mit entsprechenden Spezifikationen beim Funktionsaufruf.
- (ii) Beweisen Sie die Korrektheit der rekursiven Berechnung des größten gemeinsamen Teilers mit dem in der Vorlesung vorgestellten erweiterten Floyd-Hoare-Kalkül.

Hier ist gcd die *mathematische* Funktion, und ggt die oben implementierte C-Funktion. Durch die Verifikation beweisen Sie, dass die beiden äquivalent sind; das ist genau die Spezifikation.

Zur Erinnerung noch einmal von Übungsblatt 10 die Gleichungen für die mathematische Funktion gcd:

$$\forall a. \text{gcd}(a, 0) = a \quad (1)$$

$$\forall a. \text{gcd}(a, a) = a \quad (2)$$

$$\forall a b. 0 < a < b \implies \text{gcd}(a, b) = \text{gcd}(b - a, a) \quad (3)$$

$$\forall a b. \text{gcd}(a, b) = \text{gcd}(b, a) \quad (4)$$

Hier geht es weiter →

11.2 *Binäre Produkte*

Das Binäre Produkt rekursiv:

```
1  int binprod(int m, int n)
2  /** pre ?;
3   post ?;
4   */
5  {
6   int r;
7
8   if (n == 0) return 0;
9   r = binprod(2 * m, n / 2);
10  return r + m * (n % 2);
11 }
```

- (i) Annotieren Sie das Programm mit Vor/Nachbedingungen
- (ii) Berechnen Sie die Verifikationsbedingungen
- (iii) Beweisen Sie die Verifikationsbedingungen