Gaußsche Summenformel

Die Gaußsche Summenformel lautet: Für alle natürlichen Zahlen $n \geq 1$ gilt

$$\mathrm{A}(n): \qquad 1+2+\cdots+n=\frac{n(n+1)}{2}$$

Sie kann durch vollständige Induktion bewiesen werden. Der Induktionsanfang ergibt sich unmittelbar:

$$\mathrm{A}(1): \qquad 1=\frac{1(1+1)}{2}$$

Im Induktionsschritt ist zu zeigen, dass aus der Induktionsvoraussetzung

$$\mathrm{A}(n): \qquad 1+2+\cdots+n=\frac{n(n+1)}{2}$$

die Induktionsbehauptung

$$\mathrm{A}(n+1): \qquad 1+2+\cdots+n+(n+1)=\frac{(n+1)\big((n+1)+1\big)}{2} \qquad \text{für } n \geq 1$$

folgt. Dies gelingt folgendermaßen (Die Induktionsvoraussetzung ist rot markiert.):

$$\begin{aligned}
1+2+\cdots+n+(n+1) &= \frac{n(n+1)}{2}+(n+1) \\
&= \frac{n(n+1)+2(n+1)}{2} && \text{(Hauptnenner 2)} \\
&= \frac{(n+2)(n+1)}{2} && \text{(Ausklammern von } (n+1)\text{)} \\
&= \frac{(n+1)(n+2)}{2} && \text{(Umdrehen nach Kommutativgesetz)} \\
&= \frac{(n+1)\big((n+1)+1\big)}{2} && \text{(mit } (n+1) \text{ an der Stelle von } n\text{)}
\end{aligned}$$

Schließlich der Induktionsschluss: Damit ist die Aussage $\mathrm{A}(n)$ für alle $n \geq 1$ bewiesen.

Summe ungerader Zahlen (Maurolicus 1575)

Die schrittweise Berechnung der Summe der ersten n ungeraden Zahlen legt die Vermutung nahe: Die Summe aller ungeraden Zahlen von 1 bis $2n-1$ ist gleich dem Quadrat von n:

$1 = 1$
$1 + 3 = 4$
$1 + 3 + 5 = 9$
$1 + 3 + 5 + 7 = 16$

Der zu beweisende allgemeine Satz lautet: $\sum_{i=1}^{n}(2i-1) = n^2$.

Induktionsanfang — Beweise, dass $A(1)$ eine wahre Aussage ist.

$$\mathrm{A}(1): \qquad \sum_{i=1}^{1}(2i-1) = 2\cdot 1 - 1 = 1 = 1^2$$

Induktionsvoraussetzung — Die Aussage $A(k)$ ist wahr für ein beliebiges $k \in \mathbb{N}$.

$$\mathrm{A}(n): \qquad \sum_{i=1}^{n}(2i-1) = 1 + 3 + \cdots + (2n-1) = n^2$$

Induktionsschritt — Beweise, dass wenn $A(n=k)$ wahr ist, auch $A(n=k+1)$ wahr sein muss.

$$\mathrm{A}(n+1): \qquad \sum_{i=1}^{n+1}(2i-1) = (n+1)^2$$

Beweis
Er ergibt sich über folgende Gleichungskette, bei der in der zweiten Umformung die Induktionsvoraussetzung angewandt wird:

$$\begin{aligned}
\sum_{i=1}^{n+1}(2i-1) &= {\color{red}1 + 3 + \cdots + (2n-1)} + (2(n+1)-1) && \text{Formel für die letzte Zahl ist: } 2n-1,\ n \text{ ist hier } n+1 \\
&= {\color{red}\sum_{i=1}^{n}(2i-1)} + (2(n+1)-1) && \text{andere Schreibweise mit dem Summenzeichen} \\
&= {\color{red}n^2} + 2(n+1) - 1 && \text{Ersetzen des Summenzeichens mit dem Ergebnis der Formel} \\
&= {\color{red}n^2} + 2n + 2 - 1 && \text{ausmultiplizieren} \\
&= {\color{red}n^2} + 2n + 1 && \text{mit erster Binomischer Formel: } (a+b)^2 = a^2 + 2ab + b^2 \\
&= (n+1)^2
\end{aligned}$$

(Die Induktionsvoraussetzung ist rot markiert.)

Gegeben sei folgende Methode:[1]

```java
public class GeoSum {
  // Math.pow(q, n) == q^n
  double geoSum(int n, double q) {
    if (n == 0) {
      return 1 - q;
    } else {
      return (1 - q) * Math.pow(q, n) + geoSum(n - 1, q);
    }
  }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/aufgaben/sosy/totale_korrektheit/GeoSum.java`

Weisen Sie mittels vollständiger Induktion nach, dass

$$\text{geoSum}(n, q) = 1 - q^{n+1}$$

Dabei können Sie davon ausgehen, dass $q > 0$, $n \in \mathbb{N}_0$

Induktionsanfang — Beweise, dass $A(1)$ eine wahre Aussage ist.

$$f(0) : \text{geoSum}(0, q) = 1 - q^{0+1} = 1 - q^1 = 1 - q$$

Induktionsvoraussetzung — Die Aussage $A(k)$ ist wahr für ein beliebiges $k \in \mathbb{N}$.

$$f(n) : \text{geoSum}(n, q) = 1 - q^{n+1}$$

Induktionsschritt — Beweise, dass wenn $A(n = k)$ wahr ist, auch $A(n = k + 1)$ wahr sein muss.

$$\begin{aligned}
f(n+1) : \text{geoSum}(n+1, q) &= (1-q)^{(n+1)+1} + \text{geoSum}(n, q) \\
&= (1-q)^{n+1+1} + (1-q)^{n+1} \\
&= 1 - q^{n+1} + q^{n+1} \cdot (1-q) \\
&= 1 - q^{n+1} + q^{n+1} - q^{n+2} \\
&= 1 - q^{(n+1)+1}
\end{aligned}$$

Aufgabe 1: „Formale Verifikation“[2]

Gegeben sei folgende Methode zur Berechnung der Anzahl der notwendigen Züge beim Spiel *„Die Türme von Hanoi“*:[3]

[1] **sosy:e-klausur**.
[2] **examen:46116:2014:03**.
[3] **sosy:pu:5:1**.

```java
int hanoi(int nr, char from, char to) {
  char free = (char) ('A' + 'B' + 'C' - from - to);
  if (nr > 0) {
    int moves = 1;
    moves += hanoi(nr - 1, from, free);
    System.out.println("Move piece nr. " + nr + " from " + from + " to " + to);
    moves += hanoi(nr - 1, free, to);
    return moves;
  } else {
    return 0;
  }
}
```

Code-Beispiel auf Github ansehen:

(a) Beweisen Sie formal mittels vollständiger Induktion, dass zum Umlegen von k Scheiben (z. B. vom Turm A zum Turm C) insgesamt $2^k - 1$ Schritte notwendig sind, also dass für $k \geq 0$ folgender Zusammenhang gilt:

$$\text{hanoi}(k, \text{'A'}, \text{'C'}) = 2^k - 1$$

Zu zeigen:

$$\text{hanoi}(k, \text{'A'}, \text{'C'}) = 2^k - 1$$

Induktionsanfang — Beweise, dass $A(1)$ eine wahre Aussage ist. —

$k = 0$

$$\text{hanoi}(0, \text{'A'}, \text{'C'}) = 0$$

$$2^0 - 1 = 1 - 1 = 0$$

Induktionsvoraussetzung — Die Aussage $A(k)$ ist wahr für ein beliebiges $k \in \mathbb{N}$.

$$\text{hanoi}(k, \text{'A'}, \text{'C'}) = 2^k - 1$$

Induktionsschritt — Beweise, dass wenn $A(n = k)$ wahr ist, auch $A(n = k + 1)$ wahr sein muss.

$k \to k + 1$

$$
\begin{aligned}
\text{hanoi}(k+1, 'A', 'C') &= 1 + \text{hanoi}(k, 'A', 'B') + \text{hanoi}(k, 'B', 'C') \\
&= 1 + 2^k - 1 + 2^k - 1 \\
&= 2 \cdot 2^k - 1 \\
&= 2^{k+1} - 1
\end{aligned}
$$

(b) Geben Sie eine geeignete Terminierungsfunktion an und begründen Sie kurz Ihre Wahl!

Betrachte die Argumentenfolge $k, k-1, k-2, \ldots, 0$.

$\Rightarrow$ Terminierungsfunktion: $T(k) = k$

Nachweis für ganzzahlige $k \geq 0$:

- $T(k)$ ist auf der Folge der Argumente streng monoton fallend bei jedem Rekursionsschritt.
- Bei der impliziten Annahme k ist ganzzahlig und $k \geq 0$ ist $T(k)$ nach unten durch 0 beschränkt.

Aufgabe 5[4]

5. a) Zeigen Sie mit Hilfe vollständiger Induktion, dass das folgende Programm bzgl. der Vorbedingung $x > 0$ und der Nachbedingung drei_hoch $x = 3^x$ partiell korrekt ist!

```
(define (drei_hoch x)
  (cond ((= x 0) 1)
    (else (* 3 (drei_hoch (- x 1))))
  )
)
```

Induktionsanfang — Beweise, dass $A(1)$ eine wahre Aussage ist.

drei_hoch $1 = 3 \cdot (\text{drei\_hoch}\, 0) = 3 \cdot 1 = 3$

Induktionsvoraussetzung — Die Aussage $A(k)$ ist wahr für ein beliebiges $k \in \mathbb{N}$.

für alle $x < x0$ gilt drei_hoch $x = 3x$

Induktionsschritt — Beweise, dass wenn $A(n = k)$ wahr ist, auch $A(n = k + 1)$ wahr sein muss.

x->x+1

[4] examen:66112:2003:09.

$$\begin{aligned}\text{drei\_hoch}\,(x+1) &= 3 \cdot \text{drei\_hoch}\,(-(x+1)1))\\ &= 3 \cdot (\text{drei\_hoch}\,x)\\ &= 3 \cdot 3^x\\ &= 3^{x+1}\end{aligned}$$

Aufgabe 1: „Rekursion und Induktion“[5]

(a) Gegeben sei die Methode `BigInteger lfBig(int n)` zur Berechnung der eingeschränkten Linksfakultät:[6]

```
import java.math.Biginteger;
import static java.math.BigInteger.*;

public class LeftFactorial {
  // returns the left factorial !n
  BigInteger lfBig(int n) {
    if (n <= 0 || n >= Short.MAX_VALUE) {
      return ZERO;
    } else if (n == 1) {
    return ONE;
    } else {
      return sub(mul(n, lfBig(n - 1)), mul(n - 1, lfBig(n - 2)));
    }
  }
}
```

Implementieren Sie unter Verwendung des Konzeptes der *dynamischen Programmierung* die Methode `BigInteger dpBig(int n)`, die jede $!n$ auch bei mehrfachem Aufrufen mit dem gleichen Parameter höchstens einmal rekursiv berechnet. Sie dürfen der Klasse `LeftFactorial` genau ein Attribut beliebigen Datentyps hinzufügen und die in `lfBig(int)` verwendeten Methoden und Konstanten ebenfalls nutzen.

(b) Betrachten Sie nun die Methode `lfLong(int)` zur Berechnung der vorangehend definierten Linksfakultät ohne obere Schranke. Nehmen Sie im Folgenden an, dass der Datentyp `long` unbeschränkt ist und daher kein Überlauf auftritt.

```
long lfLong(int n) {
  if (n <= 0) {
    return 0;
  } else if (n == 1) {
    return 1;
  } else {
    return n * lfLong(n - 1) - (n - 1) * lfLong(n - 2);
  }
}
```

Beweisen Sie *formal* mittels *vollständiger Induktion*:

[5] **examen:66115:2014:03**.
[6] **aud:fs:1**.

$$\forall n \geq 0 : \text{lfLong}(n) \equiv \sum_{k=0}^{n-1} k!$$

Induktionsanfang — Beweise, dass $A(1)$ eine wahre Aussage ist. —

$$n = 1 \Rightarrow \text{lfLong}(1) = 1 = \sum_{k=0}^{n-1} k! = 0! = 1$$

$$n = 2 \Rightarrow \text{lfLong}(2) = 2 \cdot \text{lfLong}(1) - 1 \cdot \text{lfLong}(0) = 2 = \sum_{k=0}^{1} k! = 1! + 0! = 1 + 1 = 2$$

Induktionsvoraussetzung — Die Aussage $A(k)$ ist wahr für ein beliebiges $k \in \mathbb{N}$.

$$\text{lfLong}(n) = \sum_{k=0}^{n-1} k!$$

gilt:

Induktionsschritt — Beweise, dass wenn $A(n = k)$ wahr ist, auch $A(n = k + 1)$ wahr sein muss.

$$
\begin{aligned}
\text{lfLong}(n+1) &= (n+1)\cdot \text{lfLong}(n) - n\cdot \text{lfLong}(n-1) \\
&= (n+1)\cdot \sum_{k=0}^{n-1} k! - n\cdot \sum_{k=0}^{n-2} k! \\
&= (n+1)\cdot \left((n-1)! + \sum_{k=0}^{n-2} k!\right) - n\cdot \sum_{k=0}^{n-2} k! \\
&= (n+1)(n-1)! + (n+1)\cdot \sum_{k=0}^{n-2} k! - n\cdot \sum_{k=0}^{n-2} k! \\
&= (n+1)(n-1)! \cdot \sum_{k=0}^{n-2} k! + n\cdot \sum_{k=0}^{n-2} k! - n\cdot \sum_{k=0}^{n-2} k! \\
&= (n+1)(n-1)! + \sum_{k=0}^{n-2} k! \\
&= n\cdot(n-1)! + (n-1)! + \sum_{k=0}^{n-2} k! \\
&= n\cdot(n-1)! + \sum_{k=0}^{n-1} k! \\
&= n! + \sum_{k=0}^{n-1} k! \\
&= \sum_{k=0}^{n} k! \\
&= \sum_{k=0}^{(n+1)-1} k!
\end{aligned} \tag{1}
$$

Aufgabe 4: Vollständige Induktion[7]

Sie dürfen im Folgenden davon ausgehen, dass keinerlei Under- oder Overflows auftreten.[8]
Gegeben sei folgende rekursive Methode für $n \geq 0$:

```java
long sumOfSquares (long n) {
  if (n == 0)
    return 0;
  else
    return n * n + sumOfSquares(n - 1);
}
```

(a) Beweisen Sie formal mittels vollständiger Induktion:

$$\forall n \in \mathbb{N} : \texttt{sumOfSquares(n)} = \frac{n(n+1)(2n+1)}{6}$$

[7] **sosy:ab:8**.
[8] **examen:66115:2017:03**.

Sei $f(n) : \frac{n(n+1)(2n+1)}{6}$

Induktionsanfang — Beweise, dass $A(1)$ eine wahre Aussage ist. —

Für $n = 0$ gilt:

`sumOfSquares(0)` $\stackrel{\texttt{if}}{=} 0 = f(0)$

Induktionsvoraussetzung — Die Aussage $A(k)$ ist wahr für ein beliebiges $k \in \mathbb{N}$.

Für ein festes $n \in \mathbb{N}$ gelte:

`sumOfSquares(0)` $= f(n)$

Induktionsschritt — Beweise, dass wenn $A(n = k)$ wahr ist, auch $A(n = k + 1)$ wahr sein muss.

$n \to n + 1$

`sumOfSquares(n+1)` $\stackrel{\texttt{else}}{=}$

`(n+1)*(n+1)*sumOfSquares(n)` $\stackrel{\texttt{I.H.}}{=}$

$(n + 1) \cdot (n + 1) + f(n)$

$(n + 1)^2 + \frac{n(n+1)(2n+1)}{6}$

$\frac{6(n+1)^2}{6} + \frac{n(n+1)(2n+1)}{6}$

$\frac{6(n+1)^2+n(n+1)(2n+1)}{6}$

$\frac{(n+1)\cdot(6(n+1)+n(2n+1))}{6}$

$\frac{(n+1)\cdot(6n+6+2n^2+n))}{6}$

$\frac{(n+1)\cdot(2n^2+7n+6))}{6}$ $\frac{(n+1)\cdot(n+2)(2n+3))}{6}$

Neben $2n^2 + 7n + 6 = (n + 2)(2n + 3) = n \cdot 2n + 2 \cdot 2n + 3 \cdot n + 2 \cdot 6$

(b) Beweisen Sie die Terminierung von `sumOfSquares(n)` für alle $n \geq 0$.

Sei $T(n) = n$. Die Funktion $T(n)$ ist offenbar ganzzahlig. In jedem Rekursionsschritt wird n um eins verringert, somit ist $T(n)$ streng monoton fallend. Durch die Abbruchbedingung `n==0` ist $T(n)$ insbesondere nach unten beschränkt. Somit ist T eine gültige Terminierungsfunktion.