1 Mathematische Konstruktionen

1.1 Zuweisung

Die Zuweisung ist die Standardform der Nominaldefinition in der Mathematik. Dabei wird die linke Seite durch die rechte Seite definiert, schematisch:

$$x =_{\text{def}} y$$

Die linke Seite x wird als Name oder Abkürzung für die üblicherweise komplizierte, rechte Seite y eingeführt. Beide Seiten x und y dürfen in Beweisen, Rechnungen oder Umformungen beliebig gegeneinander ausgetauscht werden.

Beispiele: Folgende Definitionen sind Beispiele für Zuweisungen:

- $x =_{\text{def}} 2$
- $x =_{\text{def}} 2n + 1$
- $f(x) =_{\text{def}} x^2$
- $p \mid q \iff_{\text{def}}$ es gibt eine ganze Zahl k mit $q = k \cdot p$

Im Gegensatz zur Definition „$x =_{\text{def}} y$“ behauptet der Ausdruck „$x = y$“ eine Gleichheit, wofür eine Begründung nötig ist.

1.2 Iteration

Die iterative Definitionsform dient zum Ausdrücken von Wiederholungen in variablen, aber bestimmten Grenzen. Typische Anwendungen finden sich in Summen- oder Produktdefinitionen:

$$\begin{aligned} \sum_{k=1}^{n} a_k \quad &=_{\text{def}} \quad a_1 + a_2 + \cdots + a_n \\ \prod_{k=1}^{n} a_k \quad &=_{\text{def}} \quad a_1 \cdot a_2 \cdot \ \cdots \ \cdot a_n \end{aligned}$$

Iterationen entsprechen `for`-Schleifen in Programmiersprachen.

Beispiel: Die Fakultätsfunktion ist für alle natürlichen Zahlen n definiert als

$$n! =_{\text{def}} \prod_{k=1}^{n} k \ \text{ für } n > 0, \quad 0! =_{\text{def}} 1.$$

Ein Code-Fragment in Java sieht wie folgt aus:

```
int h=1;
for (int k=1; k<=n; k++) h=h*k;
```

Der Index k in der Produktdefinition übernimmt die Rolle der Laufvariable.

Ein typisches Problem bei iterativen Definitionen ist das Finden wertgleicher Ausdrücke ohne Verwendung der Laufvariable k (explizite Darstellung).

Beispiel: $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$ für alle natürlichen Zahlen n.

1.3 Rekursion

Bei der rekursiven Definitionsform darf die definierte Seite (linke Seite) auf der definierenden Seite (rechte Seite) vorkommen (auch mehrmals):

$$x =_{\text{def}} \ldots x \ldots$$

Da x wieder auf der rechten Seite eingesetzt werden kann, ergeben sich Schachtelungen:

$$x, \quad \ldots x \ldots, \quad \ldots(\ldots x \ldots)\ldots, \quad \ldots(\ldots(\ldots x \ldots)\ldots)\ldots, \quad \text{usw. usf.}$$

Für den Auschluss unendlicher Schachtelungen müssen Abbruchbedingungen festgelegt werden.

Beispiele: Einige Beispiele für rekursive Definitionen sind folgende:

- Die Fakultätsfunktion kann ebenfalls rekursiv definiert werden:

  $$n! =_{\text{def}} n \cdot (n-1)!, \quad 0! =_{\text{def}} 1$$

  Die rekursive Form wird bestimmt durch die Verwendung des Symbols !, das auf beiden Seiten der Definition vorkommt. Man könnte abweichend von der üblichen mathematischen Notation auch $\mathsf{fak}(n) =_{\text{def}} n \cdot \mathsf{fak}(n-1)$ und $\mathsf{fak}(0) =_{\text{def}} 1$ definieren.
  Durch wiederholtes Einsetzen der Definition erhalten wir beispielsweise

  $$4! = 4 \cdot 3! = 12 \cdot 2! = 24 \cdot 1! = 24 \cdot 0! = 24.$$

- Die Folge der Fibonacci-Zahlen ist wie folgt rekursiv definiert:

  $$F_n =_{\text{def}} F_{n-1} + F_{n-2} \quad \text{fr } n \geq 2, \quad F_1 =_{\text{def}} 1, \quad F_0 =_{\text{def}} 0$$

  Beispielsweise ergibt sich:

  $$\begin{aligned} F_5 &= F_4 + F_3 \\ &= F_3 + F_2 + F_2 + F_1 \\ &= F_2 + F_1 + F_1 + F_0 + F_1 + F_0 + F_1 \\ &= F_1 + F_0 + F_1 + F_1 + F_0 + F_1 + F_0 + F_1 \\ &= 5 \cdot F_1 + 3 \cdot F_0 \\ &= 5 \end{aligned}$$

- Eine in der Berechenbarkeitstheorie prominente Funktion ist die Ackermann-Funktion $A(x, y)$, die für natürliche Zahlen x und y durch folgende kompliziertere rekursive Definition gegeben ist:

  $$\begin{aligned} A(0, y) &=_{\text{def}} y + 1 & & \\ A(x, 0) &=_{\text{def}} x & & \text{falls } x \geq 1 \\ A(x, y) &=_{\text{def}} A(x-1, A(x, y-1)) & & \text{falls } x \geq 1, y \geq 1 \end{aligned}$$

  Beispielsweise ergibt sich:

  $$A(1, y) = A(0, A(1, y-1)) = A(1, y-1) + 1 = \cdots = A(1, 0) + y = y + 1$$

Typische Probleme bei rekursiven Definitionen sind zum einen der Nachweis der Terminierung, die nicht immer offensichtlich sein muss, und zum anderen die Auflösung oder Abschätzung der rekursiven Definition, d.h. das Finden einer äquivalenten oder näherungsweise äquivalenten Definition, bei der die linke Seite nicht mehr auf der rechten Seite vorkommt.

Beispiel: Für die oben rekursiv definierten Funktionen ergeben sich beispielsweise folgende Ungleichungen und Gleichungen:

- $\left(\frac{n}{2}\right)^{\frac{n}{2}} \leq n! \leq n^n$ für alle natürlichen Zahlen n.
- $F_n = \frac{1}{\sqrt{5}} \cdot \left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right]$ für alle natürlichen Zahlen n.
- $A(5, y) \geq \left.2^{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}\right\} y\text{−mal}$ für alle natürlichen Zahlen y.

1.4 Strukturelle Induktion

Während es bei der rekursiven Definition um das Zerlegen einer Größe geht, steht bei der induktiven Definition das Zusammensetzen von Größen aus kleineren im Vordergrund. Typische Anwendungen sind Konstruktionen von Mengen und Begriffsinstanzen. Die allgemeine Form der induktiven Definition einer Menge A ist durch folgendes Schema beschrieben:

1. Induktionsanfang: Lege die Basiselemente der Menge fest.
2. Induktionsschritt: Lege Operationen zur Konstruktion neuer Elemente der Menge aus bereits bestehenden Elementen fest.
3. Nichts sonst ist ein Element dieser Menge.

Beispiele: Folgende Mengendefinition sollen das Schema der induktiven Definition verdeutlichen:

- Die Menge der natürlichen Zahlen ist wie folgt induktiv definiert:
    1. Induktionsanfang: 0 ist eine natürliche Zahl.
    2. Induktionsschritt: Ist n eine natürliche Zahl, so ist auch $n+1$ (Inkrementierung von n) eine natürliche Zahl.
    3. Nichts sonst ist eine natürliche Zahl.
- Die Menge der korrekten Klammerausdrücke, d.h. der endlichen Folgen von Symbolen (oder), ist wie folgt induktiv definiert:
    1. Induktionsanfang: () ist ein korrekter Klammerausdruck.
    2. Induktionsschritt: Sind H_1 und H_2 korrekte Klammerausdrücke, so sind auch (H_1) (Einklammerung von H_1) und H_1H_2 (Konkatenation von H_1 und H_2) korrekte Klammerausdrücke.
    3. Nichts sonst ist ein korrekter Klammerausdruck.

Suchbäume

Wir wollen an einem größeren Fallbeispiel das Zusammenwirken von induktivem Definieren und induktivem Beweisen, wie es bereits aus Beweisen mittels vollständiger Induktion von $n-1$ nach n bekannt ist, studieren. Eine für die Informatik sehr wichtige Datenstruktur sind Suchbäume. Suchbäume dienen der Suche nach Elementen (Schlüssel) in einer geordneten, variablen Menge (Wörterbuch) mittels binärer Suche.

Die zugrunde liegenden kombinatorischen Strukturen sind volle, gewurzelte Binarbäume, die eine Verallgemeinerung von Listen darstellen. In einer Liste hat jedes Element bis auf das letzte genau einen Nachfolger und jedes Element bis auf das erste genau einen Vorgänger. Verlangt man nur die Eigenschaft, dass jedes Element bis auf eines genau einen Vorgänger besitzt (und Kreise ausgeschlossen werden), gelangt man zu Bäumen. Eine Sonderklasse von Bäumen sind volle, gewurzelte Binärbäume. Ein voller, gewurzelter Binärbaum besteht aus Knoten und Kanten, die Knoten mittels eines Pfeils $\rightarrow$ geordnet verknüpfen, sowie einem ausgezeichneten Knoten r als der Wurzel des Baumes.

Formal ist ein voller, gewurzelter Binärbaum T zunächst einmal ein Tripel (V, E, r), wobei V die Menge der Knoten (die durch natürliche Zahlen beschrieben werden) und E die Menge der Kanten (d.h., Paare (u, v) von Knoten aus V mit $u \rightarrow v$) bezeichnen sowie $r \in V$ gilt. Die interne Struktur der Kantenmenge ist damit noch nicht festgelegt. Dies geschieht induktiv durch das Einhängen zweier Bäume unter eine gemeinsame neue Wurzel:

1. Induktionsanfang: Für jede natürliche Zahl r ist der Knoten r ein voller, gewurzelter Binärbaum.

   Formal: $(\{r\}, \emptyset, r)$ ist ein voller, gewurzelter Binärbaum.

2. Induktionsschritt: Sind T_1 und T_2 volle gewurzelte Binärbäume mit den Wurzeln r_1 und r_2 (alle Knoten seien paarweise verschieden), so ist die Kollektion der Knoten und Kanten von T_1 und T_2 sowie den neuen Kanten $r \rightarrow r_1$ und $r \rightarrow r_2$ mit der neuen Wurzel $r \neq r_1, r_2$ ein voller, gewurzelter Binärbaum.

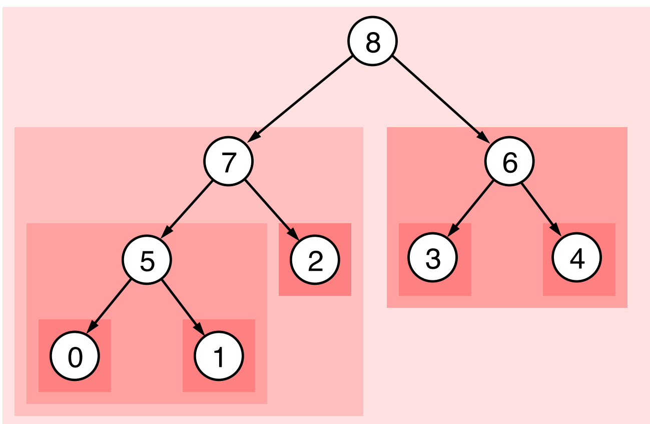
   Formal: Sind $T_1 = (V_1, E_1, r_1)$ und $T_2 = (V_2, E_2, r_2)$ volle, gewurzelte Binärbäume mit $V_1 \cap V_2 = \emptyset$ und ist $r \notin V_1 \cup V_2$, so ist

   $$f(T_1, T_2, r) =_{\mathrm{def}} (\{V_1 \cup V_2 \cup \{r\}, E_1 \cup E_2 \cup \{(r, r_1), (r, r_2)\}, r)$$

   ein voller, gewurzelter Binärbaum.

3. Nichts sonst ist ein voller, gewurzelter Binärbaum.

Beispielsweise lässt sich folgender Baum mit der angegebenen Operation (formal beschrieben durch die Funktion f) konstruieren:



Entlang der induktiven Definition können nun Eigenschaften, die für alle vollen, gewurzelten Binärbäum gelten, bewiesen werden. Für eine beispielhafte Eigenschaft führen wir noch zwei Begriffe ein. Es sei $T = (V, E, r)$ ein voller, gewurzelter Binärbaum. Ein Knoten $v \in V$ heißt Blatt (bzw. Blattknoten), falls es kein $u \in V$ mit $(v, u) \in E$ gibt; sonst heißt v innerer Knoten. Blätter sind also

Knoten ohne ausgehende Kanten; alle anderen Knoten sind innere Knoten.

Proposition 1.1

Für einen vollen, gewurzelten Binärbaum T seien n_T die Anzahl innerer Knoten und m_T die Anzahl der Blätter. Dann gilt stets $n_T = m_T - 1$.

Beweis: (Induktion über den Aufbau der Bäume) Es sei T ein beliebiger voller, gewurzelter Binärbaum.

- Induktionsanfang: Besteht T aus nur einem Knoten r, d.h. $T = (\{r\}, \emptyset, r)$, so gilt $n_T = 0$ und $m_T = 1$.
- Induktionsschritt: Besteht T aus mehr als einem Knoten, so ist T aus zwei geeigneten Bäumen T_1 und T_2 mit den Wurzeln r_1 und r_2 zusammengesetzt, d.h. $T = f(T_1, T_2, r)$ für geeignete Bäume $T_1 = (V_1, E_1, r_1)$ und $T_2 = (V_2, E_2, r_2)$ mit $V_1 \cap V_2 = \emptyset$ und $r \notin V_1 \cup V_2$. Insbesondere gilt, dass die Blätter bzw. inneren Knoten von T_1 und T_2 auch Blätter bzw. innere Knoten von T sind, da in T nur die Paare (r, r_1) und (r, r_2) hinzukommen. Mithin folgt:

$$\begin{aligned} n_T &= n_{T_1} + n_{T_2} + 1 && (r \text{ ist ein innerer Knoten von } T) \\ &= (m_{T_1} - 1) + (m_{T_2} - 1) + 1 && (\text{nach Induktionsvoraussetzung}) \\ &= (m_{T_1} + m_{T_2}) - 1 \\ &= m_T - 1 \end{aligned}$$

Damit ist die Proposition bewiesen. ■

Vollständige Induktion

Ein Spezialfall des induktiven Beweisens ist der Beweis entlang der Struktur der natürlichen Zahlen: der Beweis mittels vollständiger Induktion von $n-1$ nach n gemäß obiger induktiver Definition der natürlichen Zahlen. Da die Menge der natürlichen Zahlen auf unterschiedliche Art und Weise induktiv definiert werden kann, ergeben sich mit anderen induktiven Definitionen auch jeweils andere Formen der Induktion. Wir werden in einem späteren Abschnitt noch einmal darauf zurückkommen.

Wir wollen beispielhaft eine vollständige Induktion von $n-1$ nach n durchführen.

Tipp!

Gewöhnen Sie sich an, **Beweise mit vollständiger Induktion immer nach n** (nicht nach $n+1$) zu führen. Damit vermeiden Sie eine häufige Fehlerquelle bei komplexeren Induktionsbeweisen.

Proposition 1.2

Für alle natürlichen Zahlen n gilt

$$\sum_{k=0}^{n} (-1)^k \cdot k^2 = (-1)^n \cdot \frac{n(n+1)}{2}.$$

Beweis: (Induktion) Wir führen einen Beweis mittels vollständiger Induktion von $n-1$ nach n.

- Induktionsanfang $n = 0$: $\sum_{k=0}^{0}(-1)^k \cdot k^2 = (-1)^0 \cdot 0^2 = 0 = (-1)^0 \cdot \frac{0(0+1)}{2}$.
- Induktionsschritt $n > 0$: Es gilt also $n = (n-1)+1$. Wir nehmen an, die Aussage gilt bereits für $n-1$ (Induktionsvoraussetzung). Somit gilt

$$\begin{aligned}
\sum_{k=0}^{n}(-1)^k \cdot k^2 &= (-1)^n \cdot n^2 + \sum_{k=0}^{n-1}(-1)^k \cdot k^2 \\
&= (-1)^n \cdot n^2 + (-1)^{n-1} \cdot \frac{(n-1)n}{2} \qquad \text{(nach Induktionsvoraussetzung)} \\
&= (-1)^n \cdot \left(n^2 - \frac{(n-1)n}{2}\right) \\
&= (-1)^n \cdot \frac{n(n+1)}{2}
\end{aligned}$$

Damit ist die Proposition bewiesen. ■