

Kapitel 1

Rekursion und Induktion

Die Syntax logischer Systeme ist üblicherweise rekursiv definiert, genauso formale Beweiskalküle. Dieses Kapitel ist eine kurze Einführung in solche rekursiven Definition sowie in Beweise per vollständiger Induktion über rekursiven Definitionen.

Studierenden der Informatik sollte der Inhalt des Kapitels aus der Vorlesung [Formale Systeme, Automaten, und Prozesse](#) bekannt sein.

1.1 Rekursive Definitionen

Rekursive Definition von Funktionen über den natürlichen Zahlen

Eine Funktion $f : \mathbb{N} \rightarrow Y$ (für eine beliebige Menge Y) lässt sich wie folgt definieren:

Rekursionsanfang. Definiere $f(0)$.

Rekursionsschritt. Definiere $f(n+1)$ aus $f(n)$.

Man spricht von einer **rekursiven** (oder auch **induktiven**) Definition.

Varianten

- ▶ Im Rekursionsanfang definiert man nicht nur $f(0)$, sondern auch andere Werte.
- ▶ Im Rekursionsschritt greift man zur Definition von $f(n+1)$ nicht nur auf $f(n)$, sondern auch auf andere Werte $f(i)$ für $i \leq n$ zurück, möglicherweise auf mehrere.

Beispiel 1.1 (Fakultät)

Die **Fakultätsfunktion** $fak : \mathbb{N} \rightarrow \mathbb{N}$ ist rekursiv wie folgt definiert:

$$\begin{aligned} fak(0) &:= 1 \\ fak(n+1) &:= fak(n) \cdot (n+1) \end{aligned} \quad \text{für } n \in \mathbb{N}.$$

Dann gilt $fak(n) = 1 \cdot 2 \cdot \dots \cdot n$ für alle $n \in \mathbb{N}_+$. Man schreibt $fak(n) =: n!$.

Beispiel 1.2 (Fibonacci Folge)

Wir definieren $fib : \mathbb{N} \rightarrow \mathbb{N}$ durch

$$\begin{aligned} fib(0) &:= 1 \\ fib(1) &:= 1 \\ fib(n+2) &:= fib(n) + fib(n+1) \end{aligned} \quad \text{für } n \in \mathbb{N}.$$

Man nennt die Folge $fib(0), fib(1), fib(2), \dots$ die **Fibonacci Folge**.

Rekursive Definition von Mengen

Definition 1.3

Eine **rekursive Definition** (oder auch **induktive Definition**) einer Menge X besteht aus:

Basisregeln der Form

$$„x \in X“$$

Rekursiven Regeln der Form

$$„\text{Wenn } x_1, \dots, x_k \in X, \text{ dann } x \in X“,$$

wobei x von $x_1, \dots, x_k$ abhängt.

Intuitiv ist die definierte Menge X dann die Menge aller Elemente, deren Zugehörigkeit zu X durch endlichmaliges Anwenden der Regeln gezeigt werden kann.

Beispiel: Zweierpotenzen

Beispiel 1.4

Die Menge $Z \subseteq \mathbb{N}$ sei rekursiv wie folgt definiert:

Basisregel.

$$1 \in Z.$$

Rekursive Regeln. Für alle $n \in \mathbb{N}$:

$$\text{Wenn } n \in Z, \text{ dann } 2n \in Z.$$

Dann gilt

$$Z = \{2^k \mid k \in \mathbb{N}\}.$$

Sehr häufig sind unsere rekursiv definierten Mengen **Sprachen** über einem endlichen oder unendlichen **Alphabet**.

Zur Erinnerung

- ▶ Ein **Alphabet** ist eine nichtleere Menge, deren Elemente wir als **Zeichen** (oder **Symbole** oder **Buchstaben**) bezeichnen.

Im Folgenden sei Σ ein Alphabet.

- ▶ **Wörter** über Σ sind einfach endliche Tupel von Elementen von Σ , die wir aber üblicherweise ohne Kommas und Klammern schreiben. Beispielsweise schreiben wir ***aab*** statt (a, a, b) .
- ▶ Das leere Wort bezeichnen wir mit ϵ .
- ▶ Die Menge aller Wörter über Σ bezeichnen wir mit Σ^* . Also ist $\Sigma^* := \bigcup_{i \in \mathbb{N}} \Sigma^i$
- ▶ Die **Länge** eines Wortes $w = a_1 \dots a_n \in \Sigma^n$ ist $|w| := n$.
- ▶ Eine **Sprache über Σ** ist eine Teilmenge von Σ^* .

Beispiel 1.5

Sei Σ ein beliebiges Alphabet. Die Menge $PAL \subseteq \Sigma^*$ ist wie folgt rekursiv definiert:

Basisregeln.

- ▶ $\varepsilon \in PAL$;
- ▶ $x \in PAL$ für alle $x \in \Sigma$.

Rekursive Regeln.

Für alle $w \in \Sigma^*$ und alle $x \in \Sigma$:

Wenn $w \in PAL$, dann $xwx \in PAL$. (★)

PAL ist die Menge aller **Palindrome** über dem Alphabet Σ , also aller Wörter, die von vorne und hinten gleich gelesen sind. Wenn wir für ein Wort $w = a_1 \dots a_n \in \Sigma^*$ mit $w^{-1} := a_n \dots a_1$ das umgekehrte Wort bezeichnen, gilt also

$$PAL = \{w \in \Sigma^* \mid w = w^{-1}\}.$$

Wir verzichten auf einen Beweis der Aussage

$$PAL = \{w \in \Sigma^* \mid w = w^{-1}\}.$$

Um diese Aussage zu beweisen, würde man $PAL \subseteq \{w \in \Sigma^* \mid w = w^{-1}\}$, also *für alle $w \in PAL$ gilt $w = w^{-1}$* per vollständiger Induktion über den Aufbau von PAL beweisen (vgl. Abschnitt 1.2) und $PAL \supseteq \{w \in \Sigma^* \mid w = w^{-1}\}$, also *für alle $w \in \Sigma^*$ mit $w = w^{-1}$ gilt $w \in PAL$* per vollständiger Induktion über die Länge $n := |w|$.

Regeln und Regelschemata

- ▶ Selbst wenn das Alphabet Σ endlich ist, enthält die rekursive Definition der Menge PAL unendlich viele Regeln, nämlich eine rekursive Regel

$$\text{Wenn } w \in PAL, \text{ dann } xwx \in PAL. \quad (\star)$$

für jedes $w \in \Sigma^*$ und jedes $x \in \Sigma$.

- ▶ Allerdings haben all diese Regeln die gleiche Gestalt $(\star)$. Wir bezeichnen $(\star)$ auch als ein **Regelschema**.
- ▶ Jede einzelne Regel für ein $w \in \Sigma^*$ und $x \in \Sigma$ ist eine **Instanziierung** des Regelschemas $(\star)$.
- ▶ Rekursive Definitionen bestehen typischerweise aus den Instanziierungen endlich vieler Regelschemata.

Beispiel 1.6

Die Menge $BIN \subseteq \{0, 1\}^*$ sei wie folgt rekursiv definiert:

Basisregeln.

$$0, 1 \in BIN.$$

Rekursive Regeln.

Für alle $w \in \{0, 1\}^* \setminus \{0\}$:

Wenn $w \in BIN$, dann $w0, w1 \in BIN$.

BIN ist die Menge aller **Binärzahlen**.

Beispiel: Arithmetische Terme

Beispiel 1.7

Die Menge $ART \subseteq (\mathbb{N} \cup \{+, \cdot, -, (,)\})^*$ sei wie folgt rekursiv definiert:

Basisregeln.

- Für alle $n \in \mathbb{N}$:

$$n \in ART.$$

Rekursive Regeln.

- Für alle $t \in (\mathbb{N} \cup \{+, \cdot, -, (,)\})^*$:

(i) Wenn $t \in ART$, dann $-t \in ART$.

- Für alle $t, u \in (\mathbb{N} \cup \{+, \cdot, -, (,)\})^*$:

(ii) Wenn $t, u \in ART$, dann $(t + u) \in ART$.

(iii) Wenn $t, u \in ART$, dann $(t \cdot u) \in ART$.

(iv) Wenn $t, u \in ART$, dann $(t - u) \in ART$.

Die Semantik rekursiver Definitionen

Im Folgenden sei $\mathcal{D}$ eine rekursive Definition einer Menge X , also eine Menge von Basisregeln der Gestalt " $x \in X$ " und rekursiven Regeln der Gestalt "Wenn $x_1, \dots, x_k \in X$ dann y in X ".

Frage

Welche Elemente genau sind in X ?

Antwort

X ist die Menge aller Elemente, die man durch endlichmales Anwenden von Regeln in $\mathcal{D}$ erhält.

Aber was genau bedeutet das?

Definition 1.8

Sei $\mathcal{D}$ eine rekursive Definition einer Menge X .

Eine **Ableitung** eines Elementes z in $\mathcal{D}$ ist eine Folge $(y_1, \dots, y_n)$, so dass

- ▶ $y_n = z$;
- ▶ für alle $i \leq n$ gibt es
 - ▶ eine Basisregel „ $x \in X$ “ in $\mathcal{D}$, so dass $y_i = x$,
 - ▶ oder eine rekursive Regel „Wenn $x_1, \dots, x_k \in X$, dann $x \in X$ “ in $\mathcal{D}$ und $j_1, j_2, \dots, j_k < i$, so dass $y_{j_1} = x_1, y_{j_2} = x_2, \dots, y_{j_k} = x_k$ und $y_i = x$.

Formal ist jetzt X die Menge aller ableitbaren Elemente, also

$$X = \{z \mid z \text{ hat eine Ableitung in } \mathcal{D}\}.$$

Beispiel: Arithmetische Terme

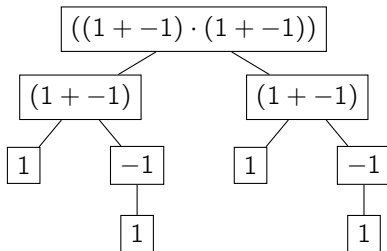
Eine Ableitung von $((1 + -1) \cdot (1 + -1))$ in der rekursiven Definition von ART (vgl. Beispiel 1.7):

$$(1, \quad -1, \quad (1 + -1), \quad ((1 + -1) \cdot (1 + -1))).$$

Eine verständlichere Darstellung dieser Ableitung ist:

- | | | |
|----|------------------------------|----------------------------|
| 1. | 1 | Basisregel |
| 2. | -1 | Rek. Regel (i) auf 1 |
| 3. | $(1 + -1)$ | Rek. Regel (ii) auf 1, 2 |
| 4. | $((1 + -1) \cdot (1 + -1)),$ | Rek. Regel (iii) auf 3, 3. |

Alternativ können wir die Ableitung von $((1 + -1) \cdot (1 + -1))$ in der rekursiven Definition von *ART* auch graphisch mit einem **Ableitungsbaum** darstellen:



Bemerkung 1.9

Generell lassen sich Ableitungen in rekursiven Definitionen durch **Ableitungsbäume** darstellen. Wir verzichten auf eine formale Definition.

Man beachte, dass lineare Ableitungen deutlich kleiner sein können als Ableitungsbäume, weil man in Ableitungsbäumen unter Umständen Teilableitungen wiederholen muss.

Beispielsweise hat die Ableitung von $((1 + -1) \cdot (1 + -1))$ in der rekursiven Definition von ART nur 4 Schritte, der Ableitungsbaum hat aber 9 Knoten.

Definition 1.10

Sei $\mathcal{D}$ eine rekursive Definition einer Menge X .

Eine Menge Y ist **abgeschlossen unter $\mathcal{D}$** , wenn sie folgende Bedingungen erfüllt:

- (i) Für alle Basisregeln „ $x \in X$ “ in $\mathcal{D}$ gilt $x \in Y$.
- (ii) Für alle rekursiven Regeln „Wenn $x_1, \dots, x_k \in X$, dann $x \in X$ “ in $\mathcal{D}$ gilt:

Wenn $x_1, \dots, x_k \in Y$, dann $x \in Y$.

Beispiel 1.11

Sei $\mathcal{D}$ die rekursive Definition der Menge $X := PAL$ der Palindrome über dem Alphabet $\Sigma := \{a, b\}$ (vgl. Beispiel 1.5).

- Die Menge PAL ist abgeschlossen unter $\mathcal{D}$.
- Die Menge

$$Y := PAL \cup \{wabw^{-1} \mid w \in \Sigma^*\},$$

wobei w^{-1} die Umkehrung des Wortes w ist, ist ebenfalls abgeschlossen unter $\mathcal{D}$.

Die kleinste abgeschlossene Menge

Satz 1.12

Sei $\mathcal{D}$ eine rekursive Definition einer Menge X .

(1) X ist abgeschlossen unter $\mathcal{D}$.

(2) Für jede unter $\mathcal{D}$ abgeschlossene Menge Y gilt $X \subseteq Y$.

Also ist X die *kleinste unter $\mathcal{D}$ abgeschlossene Menge*.

Korollar 1.13

Sei $\mathcal{D}$ eine rekursive Definition einer Menge X . Dann ist

$$X = \bigcap_{\substack{Y \text{ abgeschlossen} \\ \text{unter } \mathcal{D}}} Y.$$

Beweis von Satz 1.12.

(1) Für alle Basisregeln „ $x \in X$ “ in $\mathcal{D}$ ist

(x)

eine Ableitung von x in $\mathcal{D}$, also gilt $x \in X$.

Betrachten wir eine rekursive Regel „Wenn $x_1, \dots, x_k \in X$, dann $x \in X$ “ in $\mathcal{D}$, und nehmen wir an, dass $x_1, \dots, x_k \in X$. Für $1 \leq i \leq k$ sei

$(y_{i1}, \dots, y_{in_i})$

eine Ableitung von x_i in $\mathcal{D}$.

Dann ist

$$(y_{11}, \dots, \underbrace{y_{1n_1}}_{=x_1}, y_{21}, \dots, \underbrace{y_{2n_2}}_{=x_2}, y_{31}, \dots, \underbrace{y_{(k-1)n_{k-1}}}_{=x_{k-1}}, y_{k1}, \dots, \underbrace{y_{kn_k}}_{=x_k}, x)$$

eine Ableitung von x in $\mathcal{D}$. Also gilt $x \in X$.

- (2) Sei Y abgeschlossen unter $\mathcal{D}$. Um zu zeigen, dass $X \subseteq Y$ zeigen wir, dass jedes beliebige $z \in X$ auch in Y liegt.

Sei also $z \in X$, und sei

$$y_1, \dots, y_n$$

eine Ableitung von z in $\mathcal{D}$.

Wir zeigen per Induktion über i , $1 \leq i \leq n$, dass $y_i \in Y$. Insbesondere ist dann $z = y_n \in Y$.

Induktionsanfang: $i = 1$.

Dann enthält $\mathcal{D}$ eine Basisregel „ $x \in X$ “ mit $x = y_1$.

Weil Y abgeschlossen ist gilt $y_1 = x \in Y$.

Induktionsschritt: $i \rightarrow i + 1$, wobei $i + 1 \leq n$.

Wenn $\mathcal{D}$ eine Basisregel „ $x \in X$ “ mit $x = y_{i+1}$ enthält, so gilt wie im Induktionsanfang $y_{i+1} = x \in Y$.

Sonst enthält $\mathcal{D}$ eine rekursive Regel „Wenn $x_1, \dots, x_k \in X$, dann $x \in X$ “, und es gibt $j_1, j_2, \dots, j_k \leq i$, so dass $y_{j_1} = x_1, y_{j_2} = x_2, \dots, y_{j_k} = x_k$ und $y_{i+1} = x$.

Nach Induktionsannahme sind $y_{j_1}, y_{j_2}, \dots, y_{j_k} \in Y$.

Weil Y abgeschlossen ist, gilt dann auch $y_{i+1} = x \in Y$.



Rekursive Definition von Funktionen über rekursiv definierten Mengen

Sei $\mathcal{D}$ eine rekursive Definition einer Menge X , und sei Y eine beliebige Menge.

Eine Funktion $f : X \rightarrow Y$ lässt sich wie folgt definieren.

Rekursionsanfang.

Für alle Basisregeln „ $x \in X$ “ in $\mathcal{D}$, definiere $f(x)$.

Rekursionsschritt.

Für alle rekursiven Regeln „Wenn $x_1, \dots, x_k \in X$, dann $x \in X$ “, definiere $f(x)$ aus $f(x_1), \dots, f(x_k)$.

Wir sagen: f ist **rekursiv** (oder **induktiv**) **über** $\mathcal{D}$ definiert.

Oft geben wir die rekursive Definition $\mathcal{D}$ von X nicht explizit an, sondern gehen davon aus, dass sie aus dem Kontext bekannt ist. Wir sagen dann auch: f ist **rekursiv** (oder **induktiv**) **über den Aufbau von X** definiert.

Achtung

Rekursiv definierte Funktionen sind möglicherweise nicht “wohldefiniert”, weil verschiedene Ableitungen eines Elements x zu verschiedenen Funktionswerten führen können.

Wollen wir also eine Funktion wie auf der vorherigen Seite beschrieben rekursiv definieren, müssen wir noch beweisen, dass sie tatsächlich wohldefiniert ist.

Definition 1.14

Eine rekursive Definition $\mathcal{D}$ einer Menge X ist **eindeutig**, wenn es für jedes $x \in X$ genau eine Regel gibt, die x erzeugt, also entweder eine Basisregel „ $x \in X$ “ oder eine rekursive Regel “Wenn $x_1, \dots, x_k \in X$ dann $x \in X$ “, so dass $x_1, \dots, x_k \in X$.

Beobachtung 1.15

Sei $\mathcal{D}$ eine eindeutige rekursive Definition einer Menge X . Dann ist jede über $\mathcal{D}$ rekursiv definierte Funktion wohldefiniert.

Bemerkungen

- ▶ *Wir werden in dieser Vorlesung Funktionen rekursiv nur über eindeutigen rekursiven Definitionen definieren.*
- ▶ *Es ist leicht zu sehen, dass eine rekursive Definition $\mathcal{D}$ einer Menge X genau dann eindeutig ist, wenn jedes $x \in X$ einen eindeutigen Ableitungsbaum hat.*

Beispiel 1.16

Die rekursiven Definitionen der Mengen *PAL*, *BIN*, *ART* sind eindeutig.

Beispiel 1.17

Folgende rekursive Definition einer Menge $L \subseteq \{a, b\}^*$ ist nicht eindeutig:

Basisregel:

$$ab \in L.$$

Rekursive Regeln: Für alle $v, w \in \{a, b\}^*$:

$$\text{Wenn } v, w \in L \text{ dann } vw \in L.$$

Für dieselbe Menge lässt sich aber auch eine eindeutige rekursive Definition angeben.

Zu Beispiel 1.16:

Die Eindeutigkeit muss man in jedem Fall einzeln beweisen. Wir geben hier einen Beweis für die Eindeutigkeit der rekursiven Definition von ART .

Zur Erinnerung

Seien $v = a_1 \dots a_m, w = b_1 \dots b_n \in \Sigma^*$. Dann ist v ein **Präfix** von w (wir schreiben $v \sqsubseteq w$), wenn $m \leq n$ und $a_i = b_i$ für alle $i \in [m]$. Gilt außerdem $m < n$, so ist v ein **echtes Präfix** von w (wir schreiben $v \sqsubset w$).

Behauptung 1

Seien $t, u \in ART$. Dann ist t kein echtes Präfix von u .

Beweis.

Seien $t = a_1 \dots a_m, u = a_1 \dots a_n \in ART$.

Per Induktion über die Länge n von u zeigen wir, dass $t \not\sqsubset u$.

Induktionsanfang: $n = 1$

Weil alle Wörter in ART mindestens die Länge 1 haben, gilt $|t| \geq 1$. Also ist $t \not\sqsubseteq u$.

Induktionsschritt: $n > 1$.

Dann wird u mittels einer der rekursiven Regeln der Schemata (ii)–(iv) gebildet.

Fall 1: u wird mittels einer Regel vom Schema (i) gebildet.

Dann gilt $b_1 = -$ und $u' := b_2 \dots b_n \in ART$. Wenn $a_1 \neq -$, dann gilt $t \not\sqsubseteq u$. Nehmen wir an, $a_1 = -$. Dann wird auch t mittels einer Regel vom Schema (i) gebildet. Also $t' := a_2 \dots a_m \in ART$. Nach Induktionsannahme gilt $t' \not\sqsubseteq u'$. Also $t = -t' \not\sqsubseteq -u' = u$.

Fall 2: u wird mittels einer Regel von einem der Schemata (ii)–(iv) gebildet.

Dann ist $u = (u' * u'')$, wobei $u', u'' \in ART$ und $* \in \{+, \cdot, -\}$.

Angenommen, $t \sqsubset u$. Wir führen diese Annahme zum Widerspruch.

Weil $t \sqsubset u$ ist $a_1 = ($, also wird auch t mittels einer Regel von einem der Schemata (ii)–(iv) gebildet.

Also $t = (t' \star t'')$, wobei $t', t'' \in ART$ und $\star \in \{+, \cdot, -\}$.

Fall 2a: $|t'| < |u'|$.

Dann ist $t' \sqsubset u'$, was der Induktionsannahme *widerspricht*.

Fall 2b: $|t'| = |u'|$.

Dann ist $t' = u'$ und damit auch $\star = \ast$. Also $t'' \sqsubset u''$, was der Induktionsannahme *widerspricht*.

Fall 2c: $|t'| > |u'|$.

Dann ist $u' \sqsubset t'$, was wiederum der Induktionsannahme *widerspricht*. ┘

Wir zeigen jetzt, dass die rekursive Definition von ART eindeutig ist. Sei $t = a_1 \dots a_n \in ART$. Seien R, S Regeln der rekursiven Definition, die t erzeugen. Weil $t \in ART$, gibt es mindestens eine solche Regel. Wir zeigen, dass $R = S$.

Fall 1: R ist eine Basisregel.

Sagen wir, R ist die Regel „ $n \in ART$ “ für ein $n \in \mathbb{N}$. Dann ist $t = n$. Dann muss auch S eine Basisregel sein, denn nur Basisregeln erzeugen Wörter der Länge 1. Sei etwa S die Regel „ $m \in ART$ “ für ein $m \in \mathbb{N}$. Dann gilt $t = m$ und damit $m = n$, also $R = S$.

Fall 2: R ist eine Regel des Schemas (i).

Sagen wir, R ist die Regel „Wenn $t' \in ART$ dann $-t' \in ART$ “. Dann ist $t = -t'$. Weil nur Regeln vom Schema (i) Wörter erzeugen, die mit „-“ beginnen, muss auch S vom Schema (i) sein, etwa „Wenn $u' \in ART$ dann $-u' \in ART$ “. dann gilt $t = -u'$. Es folgt $u' = t'$ und damit $R = S$.

Fall 3: R ist eine Regel eines der Schemata (ii)–(iv).

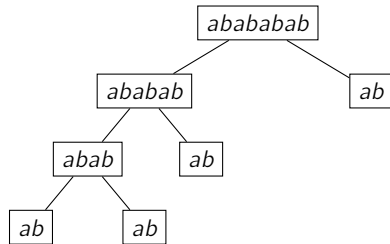
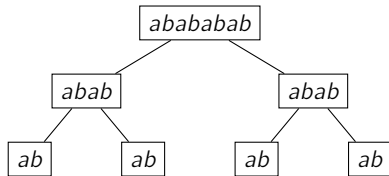
Sagen wir, R ist die Regel „Wenn $t', t'' \in ART$ dann $(t' * t'') \in ART$ “ für geeignete t', t'' und ein $* \in \{+, \cdot, -\}$. Dann ist $t = (t' * t'')$.

Weil nur Regeln der Schemata (ii)–(iv) Wörter erzeugen, die mit „(“ beginnen, muss auch S von einem dieser Schemata sein. Sei etwa S die Regel „Wenn $u', u'' \in ART$ dann $(u' \star u'') \in ART$ “ für geeignete u', u'' und ein $\star \in \{+, \cdot, -\}$. Dann ist $t = (u' \star u'')$.

Aus $t = (t' \star t'') = (u' \star u'')$ folgt, dass $t' \sqsubseteq u'$ oder $u' \sqsubseteq t'$. Nach Behauptung 1 ist dann $t' = u'$, und daraus folgen $\star = \cdot$ und $t'' = u''$. Also $R = S$. $\lrcorner$

Zu Beispiel 1.17:

Das Wort $abababab \in L$ hat folgende Ableitungsbäume:



Eine eindeutige rekursive Definition von L ist:

Basisregel:

$$ab \in L.$$

Rekursive Regeln: Für alle $w \in \{a, b\}^*$:

$$\text{Wenn } w \in L \text{ dann } wab \in L.$$

Beispiel 1.18

Wir definieren die Funktion $\text{höhe} : ART \rightarrow \mathbb{N}$ rekursiv über den Aufbau von ART (vgl. Beispiel 1.7):

Rekursionsanfang. Für alle $n \in \mathbb{N}$:

$$\text{höhe}(n) := 0.$$

Rekursionsschritt. Für alle $t, u \in ART$ und $*$ $\in \{+, \cdot, -\}$:

$$\text{höhe}(-t) := \text{höhe}(t) + 1$$

$$\text{höhe}(t * u) := \max \{ \text{höhe}(t), \text{höhe}(u) \} + 1.$$

Dann ist $\text{höhe}(t)$ die Höhe des Ableitungsbaums von t .

Beispiel: Wert einer Binärzahl

Beispiel 1.19

Wir definieren die Funktion $wert : BIN \rightarrow \mathbb{N}$ rekursiv über den Aufbau von BIN (vgl. Beispiel 1.6):

Rekursionsanfang. $wert(0) := 0$, $wert(1) := 1$.

Rekursionsschritt. Für alle $v \in BIN \setminus \{0\}$:

$$wert(v0) := 2wert(v),$$

$$wert(v1) := 2wert(v) + 1.$$

Behauptung

Diese Definition ist "korrekt", das heißt, für $b = a_1 \dots a_n \in BIN$ gilt

$$wert(b) = \sum_{i=1}^n a_i 2^{n-i}.$$

Wie kann man so eine Behauptung beweisen? — Per **vollständiger Induktion** über die rekursive Definition von *BIN*.

1.2 Induktive Beweise

Induktionsprinzip für rekursiv definierte Mengen

Sei $\mathcal{D}$ eine rekursive Definition einer Menge X .

Wir wollen für alle $x \in X$ eine Aussage $A(x)$ beweisen.

Ein **Beweis per (vollständiger) Induktion über $\mathcal{D}$** funktioniert wie folgt:

Induktionsanfang.

Für alle Basisregeln „ $x \in X$ “ in $\mathcal{D}$, beweise $A(x)$.

Induktionsschritt.

Für alle rekursiven Regeln „Wenn $x_1, \dots, x_k \in X$, dann $x \in X$ “, beweise $A(x)$ unter Verwendung der **Induktionsannahmen** $A(x_1), \dots, A(x_k)$.

Oft geben wir die rekursive Definition $\mathcal{D}$ von X nicht explizit an, sondern nehmen an, dass sie aus dem Kontext bekannt ist. Wir sprechen dann auch von einem Beweis per **(vollständiger) Induktion über den Aufbau von X** .

Man spricht auch von Beweisen durch **strukturelle Induktion**.

Das Induktionsprinzip für rekursiv definierte Mengen lässt sich auf das Induktionsprinzip für die natürlichen Zahlen zurückführen, indem man induktive Beweise über den Aufbau einer rekursiven Definition $\mathcal{D}$ zurückführt auf induktive Beweise über die Länge einer Ableitung in $\mathcal{D}$.

Beispiel: Nachfolgerfunktion

Beispiel 1.20

Wir definieren rekursiv eine **Nachfolgerfunktion** $nach : BIN \rightarrow \{0, 1\}^*$:

Rekursionsanfang. $nach(0) := 1$, $nach(1) := 10$.

Rekursionsschritt. Für alle $v \in BIN \setminus \{0\}$:

$$nach(v0) := v1,$$

$$nach(v1) := nach(v)0.$$

Behauptung

Für alle $v \in BIN$ gilt $nach(v) \in BIN \setminus \{0\}$ und $wert(nach(v)) = wert(v) + 1$.

Beweis der Behauptung.

Induktion über den Aufbau von BIN .

Induktionsanfang.

$v = 0$: $nach(0) = 1 \in BIN \setminus \{0\}$ und $wert(1) = 1 = wert(0) + 1$.

$v = 1$: $nach(1) = 10 \in BIN \setminus \{0\}$ und $wert(10) = 2 = wert(1) + 1$.

Induktionsschritt.

$v = v'0$ für ein $v' \in BIN \setminus \{0\}$: Dann ist $nach(v) = v'1 \in BIN \setminus \{0\}$, und es gilt

$$\begin{aligned} wert(nach(v)) &= wert(v'1) \\ &= 2wert(v') + 1 \\ &= wert(v'0) + 1 \\ &= wert(v) + 1. \end{aligned}$$

$v = v'1$ für ein $v' \in \text{BIN} \setminus \{0\}$: Dann ist $\text{nach}(v) = \text{nach}(v')0$.

Nach Induktionsannahme ist $\text{nach}(v') \in \text{BIN} \setminus \{0\}$, also ist $\text{nach}(v) \in \text{BIN} \setminus \{0\}$.

Außerdem gilt

$$\begin{aligned}\text{wert}(\text{nach}(v)) &= \text{wert}(\text{nach}(v')0) \\ &= 2\text{wert}(\text{nach}(v')) \\ &= 2(\text{wert}(v') + 1) \\ &= (2\text{wert}(v') + 1) + 1 \\ &= \text{wert}(v) + 1.\end{aligned}$$

