Die Menge $\mathbb{B}$

- Die Menge $\mathbb{B} = \{TRUE, FALSE\}$ der *boolschen* Werte wurde bereits erwähnt
- Wegen der häufigen Verwendung dieser Menge in der Informatik betrachten wir sie noch etwas genauer
- Die wichtigsten *Operationen* auf $\mathbb{B}$ sind sog. *innere* Operationen, d.h. der Definitions- und Bildbereich ist immer $\mathbb{B}^n$ bzw. $\mathbb{B}$
  (Man sagt auch, $\mathbb{B}$ ist *abgeschlossen* über diesen Operationen; die Abgeschlossenheit der Operatoren ist eine wesentliche Eigenschaft einer Algebra)
- Sie lassen sich wegen der Endlichkeit von Definitions- und Bildbereich mit Hilfe von *Wahrheitstafeln* explizit angeben.

Operationen auf $\mathbb{B}$

- *Negation*: $\neg : \mathbb{B} \to \mathbb{B}$

x	$\neg x$
TRUE	*FALSE*
FALSE	*TRUE*

- *Konjunktion*: $\wedge : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$

x	y	$x \wedge y$
TRUE	*TRUE*	*TRUE*
TRUE	*FALSE*	*FALSE*
FALSE	*TRUE*	*FALSE*
FALSE	*FALSE*	*FALSE*

- *Disjunktion*: $\vee : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$

x	y	$x \vee y$
TRUE	*TRUE*	*TRUE*
TRUE	*FALSE*	*TRUE*
FALSE	*TRUE*	*TRUE*
FALSE	*FALSE*	*FALSE*

- Aus diesen drei Operationen lassen sich z.B. alle möglichen zweistelligen Operationen ableiten, z.B.

  $x \Rightarrow y$ als „Abkürzung“ für $\neg x \vee y$.
- Einige Rechenregeln ($x, y, z \in \mathbb{B}$)
  - Kommutativ-Gesetze

    $$x \wedge y = y \wedge x$$
    $$x \vee y = y \vee x$$
  - Assoziativ-Gesetze

    $$x \wedge (y \wedge z) = (x \wedge y) \wedge z$$
    $$x \vee (y \vee z) = (x \vee y) \vee z$$
  - Distributiv-Gesetze:

    $$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$
    $$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$

- Diese Gesetze lassen sich am einfachsten mittels Wahrheitstafeln beweisen, z.B.

x	y	$x \wedge y$	$y \wedge x$
TRUE	*TRUE*	*TRUE*	*TRUE*
TRUE	*FALSE*	*FALSE*	*FALSE*
FALSE	*TRUE*	*FALSE*	*FALSE*
FALSE	*FALSE*	*FALSE*	*FALSE*

(Versuchen Sie's mal :-))

Warum ist die Boolsche Algebra so wichtig?

- Die Menge der Wahrheitswerte $\mathbb{B}$ dient häufig als Abstraktion der binären Steuerungsbefehle eines Computers.
- Operationen auf binär repräsentierten Daten kann man als logische Operationen auf $\mathbb{B}$ abbilden.
- Prozessoren etc. bestehen aus vielen kleinen "Transistoren", die letztlich logische Operationen ausführen können.
- 0 oder *FALSE* steht dabei z.B. für "Strom aus", 1 oder *TRUE* entspr. für "Strom an".
- Ein weiterer Aspekt ist, dass die Boolsche Algebra eine wichtige Grundlage von Logik-Kalkülen.

Beweisprinzip der „vollständigen Induktion“

- Ein fundamentales mathematisches Beweisprinzip ist die *vollständige Induktion*:
- Sei $p : \mathbb{N}_0 \to \mathbb{B}$ ein totales Prädikat, d.h. für jedes Urbild $x \in \mathbb{N}_0$ gibt es ein Bild $p(x) \in \mathbb{B}$.
- Falls gilt
  1. p gilt für $n = 0$, d.h. $p(0)$ (*Induktionsanfang*) und
  2. für beliebiges $n \in \mathbb{N}_0$ gilt der *Induktionsschluss*:
     „Falls $p(n)$ (*Induktionsvoraussetzung*), dann $p(n + 1)$.“

  dann: $p(n)$ für alle $n \in \mathbb{N}_0$.
- Die vollständige Induktion (nach n) ermöglicht es, eine Aussage (Prädikat) „p“ für alle $n \in \mathbb{N}_0$ zu beweisen.

Beispiel

- Sei $p : \mathbb{N}_0 \rightarrow \mathbb{B}$ mit
- $p : x \mapsto = (\sum_{i=0}^{x} i, \frac{x \cdot (x+1)}{2})$, oder in Infixschreibweise:
- $p : x \mapsto \sum_{i=0}^{x} i = \frac{x \cdot (x+1)}{2}$.
- Zu beweisen: Gültigkeit von $p(x)$ für alle $x \in \mathbb{N}_0$, d.h. $p(x) \mapsto \mathit{TRUE}$ für alle $x \in \mathbb{N}_0$.

  d.h. für alle $x \in \mathbb{N}_0$ gilt: $\sum_{i=0}^{x} i = \frac{x \cdot (x+1)}{2}$

- Für einen Induktionsbeweis müssen wir die beiden Beweisverpflichtungen der vorherigen Folie abarbeiten:
    1. Wir müssen zeigen, dass $p(0)$ gilt (Induktionsanfang)
    2. Wir müssen zeigen, dass für ein beliebiges $n \in \mathbb{N}_0$ aus $p(n)$ immer $p(n+1)$ folgt (Induktionsschluss).
- Zu 1. (Induktionsanfang):
    - Zu zeigen, dass $p(0)$, d.h. $\sum_{i=0}^{0} i = \frac{n \cdot (n+1)}{2}$.
    - $\sum_{i=0}^{0} i = 0$
    - $\frac{0 \cdot (0+1)}{2} = 0 \; \surd$

Beweisprinzip der „vollständigen Induktion“

- Zu 2. (Induktionsschluss): Wir müssen zeigen, dass wenn $p(n)$ für ein beliebiges $n \in \mathbb{N}_0$ gilt, dass daraus $p(n+1)$ folgt.
- Wir können also für $n \in \mathbb{N}_0$ $p(n)$ als Induktionsvoraussetzung (IV) annehmen, d.h. annehmen, dass $\sum_{i=0}^{n} i = \frac{n\cdot(n+1)}{2}$ gilt.
- Zu zeigen ist, dass daraus die Gültigkeit von $p(n+1)$, d.h. $\sum_{i=0}^{n+1} i = \frac{(n+1)\cdot(n+1+1)}{2}$ folgt:

LMU

- Dazu schauen wir uns zunächst die „linke Seite“ von $p(n+1)$ an. Kommen wir durch Umformen und unter Verwendung der IV zur „rechten Seite“?

$$\sum_{i=0}^{n+1} i = 0 + 1 + \ldots + n + (n+1) = \sum_{i=0}^{n} i + (n+1)$$

(Jetzt IV anwenden: $p(n)$, d.h. $\sum_{i=0}^{n} i = \frac{n\cdot(n+1)}{2}$ einsetzen)

$$= \frac{n\cdot(n+1)}{2} + n + 1$$
$$= \frac{n\cdot(n+1)+2\cdot(n+1)}{2} = \frac{(n+1)\cdot(n+2)}{2}$$
$$= \frac{(n+1)\cdot(n+1+1)}{2} \qquad \surd$$

- Ja, und damit ist $p(x)$ bewiesen für alle $x \in \mathbb{N}$.

Induktive Definition der Menge $\mathbb{N}_0$

- Warum ist dieses Beweisprinzip gültig?

  D.h., wie kann man sicher sein, dass p für alle Zahlen aus $\mathbb{N}_0$ gilt, wenn $p(0)$ gilt und man für ein beliebiges festes $n \in \mathbb{N}_0$ von $p(n)$ auf $p(n+1)$ schließen kann?

- Das folgt aus dem (*induktiven*) Aufbau der Menge $\mathbb{N}$:

  Die Menge $\mathbb{N}_0$ lässt sich durch folgende Regeln *induktiv definieren*:

  1. $0 \in \mathbb{N}_0$
  2. Ist $n \in \mathbb{N}_0$, dann ist auch $n+1 \in \mathbb{N}_0$.
  3. Außer den Elementen gemäß Regeln 1 und 2 enthält $\mathbb{N}_0$ keine weiteren Objekte.

Aufbau der Menge $\mathbb{N}_0$

- Die Elemente der Menge $\mathbb{N}_0$ werden gemäß dieser induktiven Definition der Reihe nach „konstruiert“:
  - Zunächst wird 0 gemäß Regel 1 als Element von $\mathbb{N}_0$ festgelegt.
  - Wegen Regel 2 ist dann $0 + 1 = 1$ Element von $\mathbb{N}_0$.
  - Erneute Anwendung von Regel 2 ergibt $1 + 1 = 2$ als Element von $\mathbb{N}_0$ usw.
- „+1“ können wir auch die Nachfolger-Funktion nennen:

$$\mathit{successor} : \mathbb{N}_0 \to \mathbb{N}_0 \text{ mit } n \mapsto n + 1$$

- Jede Zahl aus $\mathbb{N}_0$ wird erzeugt durch entsprechend häufige Anwendung (Komposition) von *successor* auf 0, z.B.:

$$3 = \mathit{successor}(\mathit{successor}(\mathit{successor}(0))), \text{ also gilt: } 3 \in \mathbb{N}_0$$

- Dieser „induktive Aufbau“ der Menge $\mathbb{N}_0$ ist der Grund für die Gültigkeit des Beweisprinzips der vollständigen Induktion.
- Das Prinzip der vollständigen Induktion vollzieht genau diesen Erzeugungsmechanismus der Menge $\mathbb{N}_0$ nach:
  - Der Induktionsanfang verifiziert $p(0)$.
  - Mit dem Induktionsschluss (aus $p(n)$ folgt immer $p(n+1)$), angewendet auf $n = 0$, erhält man $p(0+1)$, d.h. $p(1)$.
  - Mit einem weiteren Induktionsschluss, angewendet auf $n = 1$, erhält man $p(1+1)$, d.h. $p(2)$
  - usw.

- Da $\mathbb{N}_0$ nur Elemente enthält, die gemäß der induktiven Definition von $\mathbb{N}_0$ konstruiert sind, gilt dann also p tatsächlich für alle Zahlen aus $\mathbb{N}_0$.
- Tatsächlich gilt dies für alle Mengen, die induktiv definiert sind (bzw. definiert werden können).
- Z.B. ist auch die Menge $\mathbb{N}$ induktiv aufgebaut, d.h. die Menge $\mathbb{N}$ lässt sich analog durch folgende Regeln induktiv definieren:
  1. $1 \in \mathbb{N}$
  2. Ist $n \in \mathbb{N}$, dann ist auch $n + 1 \in \mathbb{N}$.
  3. Außer den Elementen gemäß Regeln 1 und 2 enthält $\mathbb{N}$ keine weiteren Objekte.

- Damit funktioniert das Beweisprinzip der vollständigen Induktion auch analog für Prädikate $p : \mathbb{N} \rightarrow \mathbb{B}$ über $\mathbb{N}$. Induktionsanfang ist dann $p(1)$.
- Es funktioniert sogar für partielle Prädikate $p : \mathbb{N} \rightarrow \mathbb{B}$, die nur für all $n \geq k$ gelten (wobei $k \in \mathbb{N}$). Induktionsanfang ist dann $p(k)$.

Weiteres Beispiel:

- Sei $p(x) : \mathbb{N} \to \mathbb{B}$ mit $x \mapsto 3|(x^3 - x)$, d.h. „$(x^3 - x)$ ist teilbar durch 3 für alle $x \in \mathbb{N}$“.
- Zeige: $p(n)$ für alle $n \in \mathbb{N}$ durch Induktion nach n:
- *Induktionsanfang*: zu zeigen, dass $p(1)$ d.h. $1^3 - 1$ ist durch 3 teilbar:

  $1^3 - 1 = 0$ und 0 ist (wie wir bereits gesehen haben) durch 3 teilbar, denn es gibt $k \in \mathbb{N}_0$ mit $k \cdot 3 = 0$, nämlich $k = 0$. $\surd$

- *Induktionsschluss*:
  Induktionsvoraussetzung (IV): $p(n)$, d.h. $n^3 - n$ ist durch 3 teilbar.
  Zeige: $p(n+1)$, d.h. $(n+1)^3 - (n+1)$ ist durch 3 teilbar.

$$
\begin{aligned}
(n+1)^3 - (n+1) &= n^3 + 3n^2 + 3n + 1 - n - 1 \\
&= (n^3 - n) + (3n^2 + 3n) = (n^3 - n) + 3n \cdot (n+1)
\end{aligned}
$$

Dieser Ausdruck ist durch 3 teilbar, wenn die beiden Summanden durch 3 teilbar sind [4]. Nach IV ist $p(n)$, d.h. $n^3 - n$ ist durch 3 teilbar.
$3n \cdot (n+1)$ ist offensichtlich ebenfalls durch 3 teilbar. $\surd$

[4] Wirklich???

Rekursive Definition von Abbildungen

- Eine weitere Konsequenz der induktiven Definition von $\mathbb{N}_0$ ist, dass wir nun Abbildungen auf $\mathbb{N}_0$ *rekursiv* definieren können.
- Die rekursive Definition einer Funktion f mit Definitionsbereich $\mathbb{N}_0$ bedeutet intuitiv:
  - $f(0)$ wird explizit festgelegt.
  - $f(n+1)$ für ein beliebiges $n \in \mathbb{N}_0$ wird auf $f(n)$ "zurückgeführt", d.h. in Abhängigkeit von $f(n)$ definiert.
  - Die Werte der Funktion $f(0)$, $f(1)$, $f(2)$ usw. sind dann wie oben erzeugbar, was $f(x)$ für alle $x \in \mathbb{N}_0$ festlegt.
- Sprechweise: Induktive und rekursive Definitionen sehen sich sehr ähnlich. Wir sagen im Folgenden:
  - Mengen werden induktiv definiert.
  - Abbildungen werden rekursiv definiert.

Beispiel: Fakultäts-Funktion

- Die Fakultäts-Funktion $! : \mathbb{N}_0 \to \mathbb{N}_0$ ist rekursiv definiert wie folgt (in Funktionsschreibweise):
  - $!(0) = 1$
  - $!(n + 1) = (n + 1) \cdot !(n)$
- bzw. in der allgemein üblichen Postfixschreibweise:
  - $0! = 1$
  - $(n + 1)! = (n + 1) \cdot (n!)$
- Der Wert von 3! ergibt sich entsprechend:
  $3! = 3 \cdot (2!) = 3 \cdot 2 \cdot (1!) = 3 \cdot 2 \cdot 1 \cdot (0!) = 3 \cdot 2 \cdot 1 \cdot 1 = 6$

Beispiel: Fakultäts-Funktion

- Oft wird äquivalent statt der Rückführung von $n+1$ auf n der Fall $n \neq 0$ auf $n-1$ zurückgeführt:

$$n! = \begin{cases} 1, & \text{falls } n = 0, \\ n \cdot (n-1)! & \text{sonst.} \end{cases}$$

- Die Berechnung von 3! (siehe letzte Folie) veranschaulicht gut, dass diese Definitionen äquivalent sind.

- Die Summenformel aus dem obigen Beispiel zum Beweis durch vollständige Induktion lässt sich ebenfalls rekursiv definieren:

$$\sum_{i=0}^{n} i = \begin{cases} 0, & \text{falls } n = 0, \\ \sum_{i=0}^{n-1} i + n & \text{sonst.} \end{cases}$$

- Das Beweisprinzip der vollständigen Induktion eignet sich besonders gut, wenn in der zu beweisenden Aussage rekursiv definierte (oder definierbare) Abbildungen auftreten.

Induktive Definition von Mengen

- Nicht nur $\mathbb{N}_0$ und $\mathbb{N}$ können induktiv definiert werden.
- Das Prinzip der induktiven Definition läßt sich leicht verallgemeinern.
- Eine Menge *M* läßt sich induktiv definieren durch:
  1. gewisse Elemente von *M* werden explizit angegeben;
  2. eine oder mehrere Regeln zur Erzeugung weiterer Elemente von *M* aus schon vorhandenen Elementen aus *M*.
- Für induktiv definierte Mengen gilt analog das Beweisprinzip der (strukturellen) Induktion und die Möglichkeit der rekursiven Funktionsdefinitionen.

Induktive Definition von Folgen

- Folgen haben wir oben als n-Tupel definiert, also als Elemente aus M^*.
- Hierbei gilt offensichtlich, dass eine Folge der Länge 1, $(a) \in M$, mit ihrem einzigen Element $a \in M$ identisch ist.
- Unter Ausnutzung dieser Eigenschaft können wir Folgen auch induktiv definieren (das hat den Vorteil, dass Funktionen rekursiv definiert werden können und Prädikate induktiv bewiesen werden können!).

Induktive Definition von Folgen

- Wir benötigen dazu noch Hilfsfunktionen, die das Anfügen eines Elementes $a \in M$ an eine Folge $x \in M^*$ ermöglichen:

$$\mathit{postfix} : M^* \times M \to M^*$$

$$\text{mit } \mathit{postfix}(x, a) = x \circ (a)$$

- Analog auch das Anfügen einer Folge $x \in M^*$ an ein Element $a \in M$:

$$\mathit{prefix} : M \times M^* \to M^*$$

$$\text{mit } \mathit{prefix}(a, x) = (a) \circ x$$

Induktive Definition von Folgen

- Damit kann eine Folge $x \in M^*, x = (x_1, \ldots, x_n)$ schrittweise aufgebaut werden, d.h. die beiden Hilfsfunktionen werden zur Erzeugung neuer Folgen verwendet.
- Ausgehend von der leeren Folge () werden die Elemente x_1, x_2, …, x_n angefügt:

$$
\begin{aligned}
x &= \mathit{postfix}(\ldots \mathit{postfix}(\mathit{postfix}((), x_1), x_2), \ldots, x_n) \\
&= () \circ (x_1) \circ (x_2) \circ \ldots \circ (x_n)
\end{aligned}
$$

Induktive Definition von Folgen

Induktive Definition von M^* unter Verwendung von *postfix*:

1. $() \in M^*$
2. Ist $x \in M^*$ und $a \in M$, dann ist $postfix(x, a) \in M^*$.

Analoge Definition unter Verwendung von *prefix*:

1. $() \in M^*$
2. Ist $a \in M$ und $x \in M^*$, dann ist $prefix(a, x) \in M^*$.

Beispiel

- Die Folge $(1,2,3) \in \mathbb{N}^*$ kann gemäß der induktiven Definition mit *postfix* wie folgt “erzeugt” werden:
  - $() \in \mathbb{N}^*$ gemäß Regel 1
  - *postfix*$((),1)$ = $(1) \in \mathbb{N}^*$ gemäß Regel 2
  - *postfix*$((1),2)$ = $(1,2) \in \mathbb{N}^*$ gemäß Regel 2
  - *postfix*$((1,2),3)$ = $(1,2,3) \in \mathbb{N}^*$ gemäß Regel 2
- Analog funktioniert es natürlich auch mit *postfix*.

Basierend auf der *prefix*-Definition lassen sich nützliche Funktionen definieren:

- Das *erste Element* einer nicht-leeren Folge $\mathit{first} : M^+ \to M$ ist definiert als

$$\mathit{first}(\mathit{prefix}(a, x)) = a$$

- Der *Rest* einer nicht-leeren Folge $\mathit{rest} : M^+ \to M^*$ mit

$$\mathit{rest}(\mathit{prefix}(a, x)) = x$$

- Die Bedeutung dieser Funktionen ist offensichtlich. Für eine nicht-leere Folge $(x_1, \ldots, x_n) \neq ()$ gilt

$$\mathit{first}(x_1, x_2, \ldots, x_n) = x_1$$

  und

$$\mathit{rest}(x_1, x_2, \ldots, x_n) = (x_2, \ldots, x_n).$$

- Analog kann man Funktionen auf Basis der *postfix*-Definition definieren, die den Zugriff auf das letzte Element („*last*") bzw. dem Kopf („*head*") einer nicht-leeren Folge erlauben.

Rekursive Funktionen über Folgen

- Da Folgen induktiv definiert sind, können wir rekursive Abbildungen über Folgen sehr einfach definieren.
- Die Projektion auf Folgen

$$\pi : M^n \times I_n \to M \quad \text{mit } n \in \mathbb{N}.$$

mit $\pi(x, i) = x_i$ können wir nun rekursiv definieren, z.B.:

$$\pi(x, i) = \begin{cases} \mathit{first}(x), & \text{falls } i = 1, \\ \pi(\mathit{rest}(x), i - 1) & \text{sonst.} \end{cases}$$

Verallgemeinerung

- Die induktive Struktur von Folgen lässt sich leicht verallgemeinern.
- Jede nicht-leere Folge ist zusammengesetzt aus einem Element $a \in M$ und einer anderen Folge x:

$$y = prefix(a, x)$$

- Abstrahiert von der konkreten Funktion *prefix* ("der Erzeugungsmechanismus") ist y durch das Paar (a, x) bestimmt, wobei x selbst wieder auf analoge Weise (wie y) bestimmt ist.

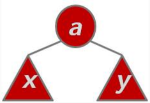
Verallgemeinerung

- Eine Verallgemeinerung kann darin bestehen, dass wir Objekte einführen, die aus einem Element a und mehreren "Resten" bestehen, z.B. zwei Resten x_1 und x_2 statt aus nur einem, d.h. "der Erzeugungsmechanismus hängt zwei Reste anünd y ist somit durch das Tripel (a, x_1, x_2) gegeben.
- Hierbei gilt induktiv, dass die "Reste" x_1 und x_2 selbst von der gleichen Art (wie y) sind.
- Solche Objekte "mit zwei Resten" heißen *Binärbäume* (über M).

Beispiel: Binärbäume

- Die Definition von Binärbäumen ist denkbar einfach, analog zu den leeren Folgen benötigen wir den *leeren Baum*, den wir mit ε bezeichnen.
- Induktive Definition der Menge $binarytree_M$ der Binärbäume über M:
  1. $\varepsilon \in binarytree_M$
  2. Wenn $a \in M$ und $x, y \in binarytree_M$, dann ist $(a, x, y) \in binarytree_M$.
- Hierbei heißt a *Wurzel*, x *linker Teilbaum*, y *rechter Teilbaum* eines Binärbaumes (a, x, y).
- Die Wurzel des Baumes und die Wurzeln von Teilbäumen (linker bzw. rechter Unterbaum) heißen *Knoten*.

- Ein Binärbaum der Gestalt $(a, \varepsilon, \varepsilon)$ heißt *Blatt* (des Baumes).
- Ein von ε verschiedener Binärbaum heißt *nicht-leer*.
- Es ist auch üblich, Bäume graphisch darzustellen. Die kanonische Darstellung für (a, x, y) ist:



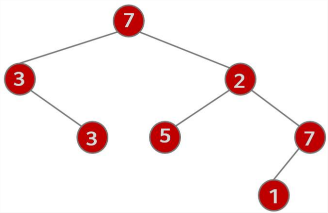
- Nochmal: wir haben eigentlich nix wesentlich anderes wie bei Folgen gemacht, nur dass wir jetzt an jedes Element aus M zwei Folgen (die jetzt Bäume heißen) anhängen.

- Beispiel: Das Objekt

$$(7,(3,\varepsilon,(3,\varepsilon,\varepsilon)),(2,(5,\varepsilon,\varepsilon),(7,(1,\varepsilon,\varepsilon),\varepsilon)))$$

  ist ein Binärbaum über $\mathbb{N}_0$.

- Graphisch:



- Entsprechend der induktiven Definition lassen sich auch leicht wieder rekursive Funktionen über Binärbäumen definieren, die auf die einzelnen Elemente zugreifen.
- Folgende Funktionen sind zunächst wiederum "Zugriffs"-Funktionen:
  - $\mathit{root} : \mathit{binarytree}_M \setminus \{\varepsilon\} \to M$
    mit $\mathit{root}(a, x, y) = a$
  - $\mathit{left} : \mathit{binarytree}_M \setminus \{\varepsilon\} \to \mathit{binarytree}_M$
    mit $\mathit{left}(a, x, y) = x$
  - $\mathit{right} : \mathit{binarytree}_M \setminus \{\varepsilon\} \to \mathit{binarytree}_M$
    mit $\mathit{right}(a, x, y) = y$

Damit können wir z.B. die Anzahl der Knoten rekursiv bestimmen:

$$nodes : binarytree_M \to \mathbb{N}_0$$

$$nodes(z) = \begin{cases} 0, & \text{falls } z = \varepsilon, \\ 1 + nodes(left(z)) + nodes(right(z)) & \text{sonst.} \end{cases}$$

Beispiel: Sei $z = (7, \underbrace{(3, \varepsilon, (3, \varepsilon, \varepsilon))}_{z_l := left(z)}, \underbrace{(2, (5, \varepsilon, \varepsilon), (7, (1, \varepsilon, \varepsilon), \varepsilon))}_{z_r := right(z)})$

$$nodes(z) = 1 + nodes(z_l) + nodes(z_r), \text{ da } z \neq \varepsilon$$

mit

$$z_l = (3, \underbrace{\varepsilon}_{left(z_l)}, \underbrace{(3, \varepsilon, \varepsilon)}_{right(z_l)}) \quad \text{und} \quad z_r = (2, \underbrace{(5, \varepsilon, \varepsilon)}_{left(z_r)}, \underbrace{(7, (1, \varepsilon, \varepsilon), \varepsilon)}_{right(z_r)})$$

Rekursive Funktionen über Binärbäume

$$nodes(z_l) = nodes((3, \underbrace{\varepsilon}_{left(z_l)}, \underbrace{(3, \varepsilon, \varepsilon)}_{right(z_l)})) = 1 + \underbrace{nodes(left(z_l))}_{=0\ (da\ \ left(z_l)=\varepsilon)} + nodes(right(z_r))$$

$$= 1 + 0 + nodes((3, \varepsilon, \varepsilon)) = 1 + 0 + 1 + nodes(\varepsilon) + nodes(\varepsilon) = 1 + 0 + 1 + 0 + 0 = 2$$

$$nodes(z_r) =$$

$$= nodes((2, (5, \varepsilon, \varepsilon), (7, (1, \varepsilon, \varepsilon), \varepsilon))) = 1 + nodes((5, \varepsilon, \varepsilon)) + nodes((7, (1, \varepsilon, \varepsilon), \varepsilon))$$

$$= 1 + 1 + nodes(\varepsilon) + nodes(\varepsilon) + 1 + nodes((1, \varepsilon, \varepsilon)) + nodes(\varepsilon)$$

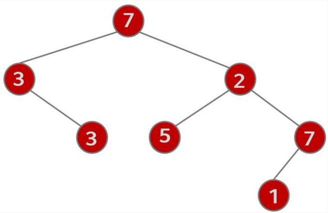
$$= 1 + 1 + 0 + 0 + 1 + 1 + 0 + 0 + 0 = 4$$

Insgesamt:

$$nodes(z_l) = 2 \quad \text{und} \quad nodes(z_r) = 4$$

d.h.

$$nodes(z) = 1 + nodes(z_l) + nodes(z_r) = 1 + 2 + 4 = 7$$



Schlussbemerkung

- Folgen bestehen aus der Multimenge ihrer Element (z.B. $(3, 2, 2, 3, 2)$).
- Ein Binärbaum besteht ebenfalls aus der Multimenge seiner Knoten.
- In beiden Fällen sind die Multimengen in bestimmter Weise angeordnet.
- Dadurch enthalten sowohl Folgen als auch Bäume mehr Information als die Multimengen ihrer Elemente bzw. Knoten.
  - Bei Folgen sind die Elemente *linear* angeordnet.
  - Bäume beschreiben eine *verzweigte* Struktur der Elemente.

Schlussbemerkung

- Wie vorher angedeutet können wir natürlich nicht nur zwei Reste anhängen, sondern generell $n \in \mathbb{N}$:

$$(a, x_1, \ldots, x_n)$$

- Diese Objekte heißen *n-äre Bäume*, jeder Knoten hat n Teilbäume (n heißt in diesem Kontext auch (Verzweigungs-)Grad, engl. fanout).
- Bäume eignen sich sehr gut, hierarchische Strukturen zu repräsentieren.