und wollen zeigen, dass $A \to B$ eine wahre Aussage ist. Wir zeigen die kontrapositive Aussage $\neg B \to \neg A$: Wenn a ungerade ist, dann ist auch a^2 ungerade. Ist a ungerade, so ist $a = 2k+1$ für eine ganze Zahl k. Durch das Quadrieren erhalten wir

$$a^2 = (2k+1)^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1\,.$$

Da k eine ganze Zahl ist, ist auch $(2k^2 + 2k)$ eine ganze Zahl. Also ist a^2 ungerade, wie behauptet. □

Die Widerspruchsregel ist die am häufigsten benutzte Beweisregel in der Mathematik überhaupt. In diesem Buch werden wir diese Regel sehr oft verwenden. An dieser Stelle demonstrieren wir die Regel mit einem wichtigen Satz.

Satz 2.4: Satz von Euklid
Es gibt unendlich viele Primzahlen.

Beweis:
Wir beweisen den Satz durch einen Widerspruch. Dazu nehmen wir das Gegenteil des Satzes an. Es sei $\{p_1, p_2, \ldots, p_k\}$ die endliche Menge aller Primzahlen. Wir betrachten dann die Zahl $n = p_1 p_2 \cdots p_k + 1$. Nun haben wir nur zwei Möglichkeiten: Entweder n ist prim oder nicht. Wir zeigen, dass unsere Annahme in beiden Fällen zu einem Widerspruch führt.

Fall 1: Angenommen n ist eine Primzahl. Da nach Annahme $\{p_1, p_2, \ldots, p_k\}$ *alle* Primzahlen enthält, muss n eine von diesen Zahlen sein. Das ist aber ein Widerspruch, da nach ihrer Definition die Zahl n echt größer als jede dieser Zahlen ist.

Fall 2: Angenommen n ist keine Primzahl. Dann muss n durch eine Primzahl p ohne Rest teilbar sein. Nach unserer Annahme muss p eine der Zahlen $p_1, p_2, \ldots, p_k$ sein und damit das Produkt $p_1 p_2 \cdots p_k$ ohne Rest teilen. Dann ergibt aber n geteilt durch p den Rest 1, ein Widerspruch.

Da wir in beiden möglichen Fällen einen Widerspruch zu unserer Annahme »es gibt nur endlich viele Primzahlen« erhalten haben, war diese Annahme falsch. □

2.4 Induktion: Beweis von $\forall x\ P(x)$

Als nächstes werden wir ein einfaches aber überraschend mächtiges Beweisprinzip für Aussagen der Form $\forall n \in \mathbb{N}:\ P(n)$ kennenlernen – die *Induktion.* [1]

2.4.1 Das Induktionsprinzip

Die Grundidee der Induktion[2] beruht auf dem axiomatischen Aufbau der natürlichen Zahlen nach Peano: Man kann jede natürliche Zahl dadurch erhalten, indem man, beginnend mit der 0, wiederholt 1 addiert. Entsprechend beweist man eine Eigenschaft $P(n)$

1 In der Literatur benutzt man oft den Namen »vollständige Induktion«, obwohl keine »nicht vollständige« Induktion bekannt ist!

2 Wer hat die Induktion erfunden? Das ist nicht ganz klar. Klar ist nur, dass Francesco Maurolico die Induktion in seinem Buch *Arithmeticorum Libri Due* (1575) benutzt hat, um zu zeigen, dass die Summe der ersten n ungeraden Zahlen gleich n^2 ist; siehe Aufgabe 2.7.

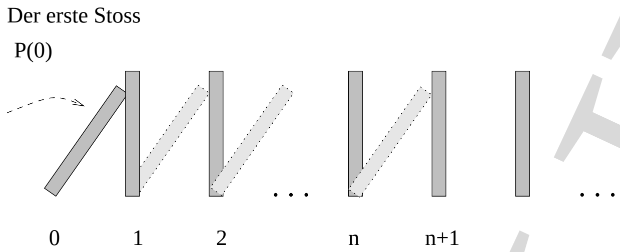


Bild 2.2: Induktionsbasis: Der erste Dominostein fällt, wenn er angestoßen wird. Induktionsschritt: Der $(n+1)$-erste Stein fällt, *falls* der n-te fällt.

für jede natürliche Zahl n, indem man zuerst die Eigenschaft $P(0)$ – die so genannte *Induktionsbasis* – beweist, und anschließend zeigt, dass für beliebige natürliche Zahlen n aus $P(n)$ auch $P(n+1)$ folgt – der so genannte *Induktionsschritt*:

(a) Induktionsbasis: Zeige, dass $P(0)$ gilt.

(b) Induktionsschritt $n \mapsto n+1$: Für beliebiges $n \in \mathbb{N}$ zeige, dass $P(n) \rightarrow P(n+1)$ gilt.

Man nennt n die *Induktionsvariable* oder den *Induktionsparameter*.

Auf den ersten Blick scheint ein solches Argument etwas verwirrend: Es scheint als ob wir in dem Induktionsschritt die zu beweisende Aussage »$\forall n \in \mathbb{N} : P(n)$« zugrunde legen. Dies ist aber nicht der Fall, denn man kann die Implikation $P(n) \rightarrow P(n+1)$ auch dann beweisen, wenn man nichts über den tatsächlichen Wahrheitswert von $P(n)$ weiß: Um $P(n) \rightarrow P(n+1)$ zu beweisen, können wir *annehmen*, dass $P(n)$ wahr ist!

Nach (a) wissen wir, dass $P(0)$ gilt, und nach (b) wissen wir, dass auch $P(0) \rightarrow P(1)$ gilt. Dann muss auch $P(1)$ gelten (modus ponens). Aus der Gültigkeit von $P(1)$ und $P(1) \rightarrow P(2)$ können wir wiederum die Gültigkeit von $P(2)$ schließen, usw. Da wir jede natürliche Zahl n durch die wiederholte Addition von 1 aus 0 erreichen können, folgt daraus die Gültigkeit von $P(n)$ für alle n.

Ein Standardbeispiel der Induktion ist eine unendliche Folge der Dominosteine mit der Aussage $P(n)$ interpretiert als »der n-te Dominostein fällt um« (Bild 2.2). Die Induktionsbasis ist »der Stein 0 fällt um« (da wir ihn anstoßen). Der Induktionsschritt $P(n) \rightarrow P(n+1)$ ist »der Stein $n+1$ fällt um, falls der Stein n umfällt«.

Beachte, dass man nicht unbedingt von Null starten muss. Will man eine Aussage $\forall n \geq m : P(n)$, also die Aussage »$P(n)$ gilt für alle $n \geq m$« für eine feste natürliche Zahl m beweisen, so ist $P(m)$ die Induktionsbasis.

Nicht immer ist es im Induktionsschritt einfach, alleine von $P(n)$ auf $P(n+1)$ zu schließen. Betrachtet man den Induktionsschritt genauer, so sieht man, dass man eigentlich sogar die Gültigkeit von $P(0) \wedge \cdots \wedge P(n)$ als Voraussetzung nutzen kann. D. h. es reicht, die Gültigkeit von

$$P(0) \wedge P(1) \wedge \cdots \wedge P(n) \rightarrow P(n+1)$$

für alle n zu zeigen. Diese Variante nennt man *verallgemeinerte Induktion*.

2.4.2 Das Prinzip des »kleinsten Verbrechers«

Es gibt auch eine andere Variante der Induktion, die oft leichter anzuwenden ist. Will man die Gültigkeit der Aussage $P(n)$ für alle n beweisen, so kann man einen Widerspruchsbeweis führen. Zeige zuerst, dass $P(0)$ gilt. Nimm dann an, dass $P(n)$ *nicht* für alle n gilt. Dann muss es die *kleinste* Zahl n geben, für die die Aussage $P(n)$ nicht gilt; diese Zahl n nennt man das »kleinste Gegenbeispiel« oder den »kleinsten Verbrecher«. Da n die kleinste solche Zahl ist, muss $P(m)$ für alle $0 \leq m \leq n-1$ gelten. Es reicht dann zu beweisen, dass es keinen solchen »kleinsten Verbrecher« geben kann.

Warum ist dieses Prinzip dasselbe wie die Induktion? Sei n ein »kleinster Verbrecher«. Zunächst müssen wir den Fall $n = 0$ ausschließen, und das ist genau die Induktionsbasis. Für $n > 0$ wissen wir, dass $P(n-1)$ gilt, nicht aber $P(n)$. Um einen Widerspruch zu erhalten, reicht es somit die Gültigkeit der Implikation $P(n-1) \to P(n)$, also den Induktionsschritt zu beweisen.

2.4.3 Falsche Anwendungen

Um die Gefahren bei Anwendung des Induktionsprinzips zu zeigen, beginnen wir mit einigen falschen Anwendungen.

Beispiel 2.5:

Unsere erste (nicht ernst gemeinte) Behauptung ist: *In einen Koffer passen unendlich viele Paare von Socken.*

»Beweis« durch Induktion: Induktionsbasis: $n = 1$. Ein paar Socken passt in einen leeren Koffer.

Induktionsschritt $n \mapsto n+1$: In einem Koffer sind n Paar Socken. Ein paar Socken passt immer noch rein, dies ist eine allgemeingültige Erfahrung. Also sind nun $n+1$ Paar Socken in dem Koffer. □

Wo ist der Fehler? Die Induktion ist ein *konstruktives* Beweisverfahren und solche Beweise erfordern auch konstruktive Argumente. Im Sockenbeispiel war das Argument »die Erfahrung sagt, dass immer noch ein paar Socken mehr in den Koffer passt« nicht konstruktiv. Ein konstruktives Argument sollte genau sagen *wo die Lücke für das weitere Paar Socken sein wird*!

Man vergisst oft, die Induktionsbasis zu verifizieren.

Beispiel 2.6:

Sei $P(n)$ die Aussage »$\forall n \in \mathbb{N} : n = n+1$«. Der Induktionsschritt $P(n) \to P(n+1)$ ist für alle n eine wahre Aussage, denn $n = n+1 \to (n+1) = (n+1)+1$ gilt. Aber $P(0)$ ist falsch, da $0 = 1$ nicht gilt.

Will man eine Aussage $\forall n \geq m : P(n)$ beweisen, so muss der Induktionsschritt $n \mapsto n+1$ für alle $n \geq m$, also auch für $n = m$ gelten.

Beispiel 2.7: Elefanten

Wir betrachten die Aussage $P(n) =$ »wenn sich unter n Tieren ein Elefant befindet, dann sind alle diese Tiere Elefanten«.

Induktionsbasis: $n = 1$: Wenn von einem Tier eines ein Elefant ist, dann sind alle diese Tiere Elefanten.

Induktionsschritt: $n \mapsto n+1$. Sei unter $n+1$ Tieren eines ein Elefant. Wir stellen die Tiere so in eine Reihe, dass sich dieser Elefant unter den ersten n Tieren befindet. Nach der Induktionsannahme sind dann alle diese ersten n Tiere Elefanten. Damit befindet sich aber auch unter den letzten n Tieren ein Elefant, womit diese auch alle Elefanten sein müssen. Also sind alle $n+1$ Tiere Elefanten. $\square$

Wo ist das Argument falsch? Im Fall $n + 1 = 2$ kann man den Elefanten zwar so stellen, dass er bei den ersten $n = 1$ Tieren steht. Folglich sind alle Tiere unter den ersten $n = 1$ Tieren Elefanten. Aber deshalb befinden sich unter den *letzten* $n = 2 - 1 = 1$ Tieren nicht notwendig Elefanten. Daher gilt $P(1) \to P(2)$ nicht.

Mit der Induktion kann man Aussagen nicht nur über natürliche Zahlen, sondern auch über die Elemente einer beliebigen Menge M beweisen. Will man eine Aussage der Form »für alle $x \in M$ gilt $Q(x)$« beweisen, so wählt man zunächst eine Abbildung $f : M \to \mathbb{N}$ aus, die jedem Objekt $x \in M$ seine »Länge« $f(x)$ zuweist, und probiert die Aussage $\forall n \in \mathbb{N} : P(n)$ mit

$$P(n) = \text{»für alle } x \in M \text{ mit } f(x) = n \text{ gilt } Q(x)\text{«} \tag{2.1}$$

mittels Induktion zu beweisen. Ist zum Beispiel M die Menge aller Graphen mit bestimmten Eigenschaften, so kann man als $f(x)$ die Anzahl der Knoten oder der Kanten in x nehmen. In dieser Situation ist n nur ein Parameter, der der ganzen Menge $f^{-1}(n)$ der Elemente aus M zugewiesen ist.

In solchen Fällen muss man aber sehr vorsichtig mit dem Induktionsschritt $n \mapsto n+1$ umgehen.

Ein häufiger Fehler ist der folgende. Man nimmt ein beliebiges Element $x \in M$ der Länge $f(x) = n + 1$ und »manipuliert« es, um ein Element x' der Länge $f(x') = n$ zu erhalten. Da x' eine kleinere Länge $f(x')$ als $f(x)$ hat, schließt man daraus, dass »nach Induktionsvoraussetzung« auch $Q(x')$ gelten muss. Dies muss aber nicht unbedingt der Fall sein: Das neue Element x' muss in der Menge M liegen, denn die Aussage $P(n)$ spricht nur über die Elemente in M! Man muss also noch $x' \in M$ nachweisen.

Beispiel 2.8: Jetzt knallt es aber richtig!

Wir wollen die folgende verblüffende Aussage »beweisen«:

Alle natürlichen Zahlen sind gleich.

In diesem Fall ist $M = \mathbb{N} \times \mathbb{N}$ die Menge aller Paare $x = (a, b)$ der natürlichen Zahlen und wir wollen zeigen, dass $a = b$ für alle solche Paare gilt. Dazu nehmen wir die Längenfunktion $f(a, b) = \max\{a, b\}$ und betrachten die Aussagen

$$P(n) = \text{»für alle } (a, b) \in \mathbb{N} \times \mathbb{N} \text{ mit } \max\{a, b\} = n \text{ gilt } a = b\text{«}.$$

Induktionsbasis $n = 0$ ist richtig, denn aus $a, b \in \mathbb{N}$ und $\max\{a, b\} = 0$ folgt $a = 0$ und $b = 0$, also $a = b$.

Induktionsschritt: $n \mapsto n+1$. Nehmen wir an, dass $P(n)$ gilt, und betrachten ein beliebiges Paar von Zahlen $a, b \in \mathbb{N}$ mit $\max\{a, b\} = n+1$. Dann ist $\max\{a-1, b-1\} = n$ und nach Induktionsannahme gilt $a-1 = b-1$ und damit auch $a = b$. $\square$

Wo ist der Fehler? Aus $\max\{a, b\} = n+1$ folgt zwar $\max\{a-1, b-1\} = n$, aber die Induktionsvoraussetzung wird damit nicht unbedingt erfüllt: Wenn wir zum Beispiel $a = 0$ und $b = 1$ betrachten, dann ist zwar $\max\{a-1, b-1\} = 0$ immer noch eine natürliche Zahl, die Zahl $a-1 = -1$ aber nicht!

Um eine Aussage $P(n)$ von der Form (2.1) zu beweisen, wird auch gerne der folgende Fehler gemacht. Man nimmt ein beliebiges Element x' der Länge $f(x') = n$, manipuliert es, um ein ebenfalls »beliebiges« Element x der Länge $f(x) = n+1$ zu erhalten, und weist anschließend die Implikation $Q(x') \to Q(x)$ nach. Ist das gelungen, so »folgert« man daraus, dass die Aussage $Q(x)$ auch für *alle* Elemente x der Länge $n+1$ gelten muss. (Dieses Vorwärts-Verfahren ist man ja schließlich von den Zahlen her gewohnt.) Die Gültigkeit von $Q(x)$ für die konstruierten Elemente x garantiert aber alleine noch nicht, dass die Aussage $Q(x)$ auch für *alle* Elemente x der Länge $n+1$ gelten muss. Es kann Elemente der Länge $n+1$ in M geben, die nicht durch diese Konstruktion erreicht werden.

Beispiel 2.9:

Wenn ein ungerichteter Graph $G = (V, E)$ zusammenhängend ist, also wenn für zwei beliebige Knoten $u, v \in V$ mindestens ein Weg von u nach v existiert, dann muss jeder Knoten mit mindestens einem anderen Knoten benachbart (d. h. adjazent) sein. Ist die Umkehrung auch richtig? Natürlich nicht: Als Gegenbeispiel kann man einen Graphen nehmen, der nur aus zwei knotendisjunkten Kanten besteht. Trotzdem wollen wir die folgende Aussage »beweisen«.

Behauptung 2.10: Eine falsche Behauptung!

Wenn jeder Knoten in einem ungerichteten Graphen mit mindestens einem anderen Knoten benachbart ist, dann ist der Graph zusammenhängend.

»Beweis«: Wir gehen induktiv vor und wählen die Zahl n der Knoten als Induktionsvariable. Die Länge eines Graphen $G = (V, E)$ ist also $f(G) = |V|$. Sei $P(n)$ die folgende Aussage: Wenn in einem Graphen G mit n Knoten jeder Knoten mit mindestens einem anderen Knoten benachbart ist, dann ist G zusammenhängend. Wir wollen »zeigen«, dass $P(n)$ für alle n gilt.

Für die Induktionsbasis $n = 1$ ist $P(n)$ offensichtlich richtig, da es keinen anderen Knoten gibt. Der Fall $n = 2$ ist auch offensichtlich, da in diesem Fall der Graph nur aus einer Kante besteht.

Induktionsschritt $n \mapsto n+1$. Wir nehmen einen beliebigen Graphen $G_n = (V, E)$ mit $|V| = n$ Knoten, in dem jeder Knoten mit mindestens einem anderen Knoten benachbart ist. Dann fügen wir einen neuen Knoten $x \notin V$ hinzu. Da x mit mindestens einem anderen Knoten benachbart sein muss und als Nachbarn nur Knoten aus V in Frage kommen, verbinden wir x mit mindestens einem Knoten aus V. Sei G_{n+1} der dabei enstehende Graph mit $n+1$ Knoten. Wir wollen »zeigen«, dass auch G_{n+1} zusammenhängend sein muss. Dafür betrachten wir zwei beliebige Knoten u und v aus $V \cup \{x\}$. Gehören diese beiden Knoten zu V, so muss es nach Induktionsvoraussetzung einen Weg von u nach v (im Graphen G_n und somit auch in G_{n+1}) geben.

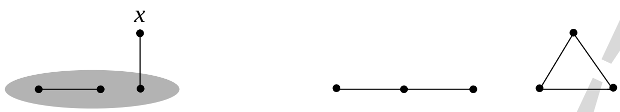


Bild 2.3: Der Graph links liegt in M_4, da jeder sein Knoten einen Nachbarn hat. Die Menge M_3 besteht aus zwei Graphen rechts.

Sei nun $v \notin V$, d. h. $v = x$. Nach Definition von G_{n+1} muss der Knoten x mindestens einen Nachbarn $y \in V$ haben. Nach Induktionsvoraussetzung muss es daher einen Weg von u nach y in G_n geben, der sich durch die Kante $\{x, y\}$ zu einem Weg von u nach x (über y) erweitern lässt. Also ist der Graph G_{n+1} zusammenhängend. □

Wo ist der Fehler? Sei M_n die Menge aller Graphen mit n Knoten, in denen jeder Knoten mit mindestens einem anderen Knoten benachbart ist. In dem obigen »Beweis« geht man davon aus, dass man *jeden* Graphen in M_{n+1} aus mindestens einem Graphen in M_n auf die beschriebene Weise konstruieren kann. Dies ist aber falsch! So gehört zwar der in Bild 2.3 links gezeichnete Graph zu M_4, er kann aber aus keinem der Graphen in M_3 konstruiert werden, da M_3 nur aus den in Bild 2.3 rechts gezeichneten zwei Graphen besteht.

2.4.4 Richtige Anwendungen

Nun (endlich) folgen einige Beispiele für die Sätze, die man leicht (und richtig!) mittels Induktion beweisen kann.

Satz 2.11: Bernoulli-Ungleichung

Für jede reelle Zahl $x \geq -1$ und für jede natürliche Zahl $n \geq 1$ gilt

$$(1+x)^n \geq 1 + nx\,.$$

Beweis:

Wir führen den Beweis mittels Induktion über n.

Induktionsbasis: Für $n = 1$ gilt $1 + x \geq 1 + x$.

Induktionsschritt $n \mapsto n+1$: Gilt $(1+x)^n \geq 1 + nx$, so gilt auch

$$\begin{aligned}(1+x)^{n+1} &= (1+x)^n \cdot (1+x) \\ &\geq (1+nx) \cdot (1+x) && 1+x \geq 0 \\ &= 1 + nx + x + nx^2 \\ &= 1 + (n+1)x + nx^2 && nx^2 \geq 0 \\ &\geq 1 + (n+1)x\,.\end{aligned}$$

□

Wir zeigen nun mittels verallgemeinerter Induktion, dass jede natürliche Zahl, die größer oder gleich 2 ist, sich als Produkt von Primzahlen darstellen lässt.[3] Es gilt zum Beispiel $1815 = 3 \cdot 5 \cdot 11 \cdot 11$.

Satz 2.12: Primzahldarstellung
Sei n eine natürliche Zahl, und $n \geq 2$. Dann ist n Produkt von Primzahlen.

Beweis:
Wir führen den Beweis mittels verallgemeinerter Induktion.

Induktionsbasis $n = 2$: Da 2 eine Primzahl ist, ist 2 triviales Produkt von sich selbst, also Produkt einer Primzahl.

Induktionsschritt $n \to n + 1$: Sei n beliebig und nehmen wir an, dass *alle* Zahlen von 2 bis n sich als Produkte von Primzahlen schreiben lassen. Wir zeigen, dass dann $n + 1$ ebenfalls ein Produkt von Primzahlen ist. Dazu machen wir eine Fallunterscheidung.

Fall 1: Angenommen $n+1$ ist eine Primzahl. Dann ist $n+1$ die einzige Primzahl, aus der das Produkt $n + 1$ besteht.

Fall 2: Angenommen $n+1$ ist keine Primzahl. Dann gibt es zwei echte Teiler von $n+1$, also natürliche Zahlen a und b mit $2 \leq a, b < n+1$, so dass $n+1 = ab$ gilt. Da a und b beide kleiner als $n+1$ sind, können wir die Induktionsvoraussetzung nutzen und die Zahlen a und b als Produkte von Primzahlen schreiben. Dann ist $n+1 = ab$ auch ein Produkt von Primzahlen. □

Die Folge der sogenannten *harmonischen Zahlen* $H_1, H_2, H_3, \ldots$ ist definiert durch

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}\,.$$

Satz 2.13: Harmonische Zahlen
Für alle $n = 0{,}1, \ldots$ gilt:

$$1 + \frac{n}{2} \leq H_{2^n} \leq n + 1\,.$$

Beweis:
Induktionsbasis: Für $n = 0$ gilt $1 + \frac{0}{2} \leq H_{2^0} = 1 \leq 0 + 1$.
Induktionsvoraussetzung: Es gilt $1 + \frac{n}{2} \leq H_{2^n} \leq n + 1$ für ein festgelegtes n. Unter Anwendung der Induktionsvoraussetzung folgt

$$H_{2^{n+1}} = \overbrace{1 + \frac{1}{2} + \ldots + \frac{1}{2^n}}^{\geq 1 + \frac{n}{2}} + \overbrace{\frac{1}{2^n + 1} + \ldots + \frac{1}{2^{n+1}}}^{2^n \text{ Zahlen}}\,.$$

3 Zur Erinnerung: $p \in \mathbb{N}$ ist eine *Primzahl* genau dann, wenn $p \geq 2$ gilt und p *nur* durch 1 und p teilbar ist. Achtung: 1 ist also *keine* Primzahl!

Die letzte Summe besteht aus 2^n Zahlen und die kleinste davon ist $\frac{1}{2^{n+1}}$. Somit trägt diese Summe mindestens $2^n \cdot \frac{1}{2^{n+1}} = \frac{1}{2}$ bei, was die erwünschte Ungleichung

$$H_{2^{n+1}} \geq 1 + \frac{n}{2} + \frac{1}{2} = 1 + \frac{n+1}{2}$$

ergibt. Da $\frac{1}{2^n+1}$ die größte der letzten 2^n Zahlen ist, gilt auch

$$H_{2^{n+1}} \leq n + 1 + \frac{2^n}{2^n+1} \leq n + 2 = (n+1) + 1\,. \qquad \square$$

Nun zeigen wir, wie man einige wichtige Eigenschaften von Bäumen mittels Induktion relativ leicht beweisen kann.

Satz 2.14:

Jeder Baum mit n Knoten hat $n-1$ Kanten.

Beweis:

Wir beweisen den Satz mittels Induktion über n. Induktionsbasis $n = 1$ ist offenbar richtig. Wir nehmen nun an, dass die Aussage des Satzes wahr ist für alle Bäume mit höchstens n Knoten. Sei B ein beliebiger Baum mit $n+1$ Knoten. Da $n+1 \geq 2$, besitzt B mindestens ein Blatt (siehe Aufgabe 1.5), also einen Knoten v vom Grad $d(v) = 1$. Sei e die zu v adjazente Kante mit dem zweiten Endpunkt u, also $e = \{v, u\}$, und sei B' der Baum mit n Knoten, der durch Entnahme des Knoten v und der Kante e entsteht. Dann hat B' nach Induktionsannahme hat $n-1$ Kanten. Für B ergibt sich damit eine Kantenzahl von $(n-1) + 1 = n$. $\square$

In vielen Anwendungen ist ein Knoten des Baumes B als Startknoten ausgezeichnet; dann spricht man über einen *Wurzelbaum*. In einem solchen Baum kann das Verhältnis der einzelnen Knoten des Baumes zueinander begrifflich gut beschrieben werden. Dazu benutzt man den Begriff der *Tiefe* eines Knotens.

Als *Tiefe* eines Knotens v von B wird der Abstand von v zur Wurzel (d. h. die Anzahl der Kanten in dem eindeutigen Weg von v zur Wurzel) bezeichnet. Die *Tiefe* $t(B)$ von B ist die maximale Tiefe eines Knotens von T. Alle Knoten gleicher Tiefe bilden ein *Knotenniveau*. Als *Kinder* eines Knotens v von B werden sämtliche Knoten bezeichnet, die zu v benachbart sind und deren Tiefe die von v um eins übersteigt; v heißt *Vater* seiner Kinder. Knoten ohne Kinder heißen *Blätter* (Bild 2.4).

In einem *binären Baum* hat jeder innere Knoten genau zwei Kinder. Was kann man dann über die Tiefe $t(B)$ eines binären Baums B sagen, wenn man die Anzahl der Blätter $|B|$ kennt? Man kann leicht zeigen, dass es sowohl Bäume mit $t(B) = |B| - 1$ wie auch Bäume mit $t(B) = \log_2 |B|$ geben kann (siehe Bild 2.5). Wir werden nun beweisen, dass $\log_2 |B|$ bereits die untere Schranke für die Tiefe ist.

Satz 2.15:

Für jeden binären Wurzelbaum B gilt: $t(B) \geq \log_2 |B|$, d. h.

$$\text{Tiefe}(B) \geq \log_2(\text{Anzahl der Blätter}).$$

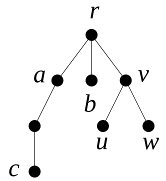


Bild 2.4: Dieser Wurzelbaum mit der Wurzel r hat die Tiefe 3; u und w sind Kinder des Knotens v, und v besitzt die Tiefe 1; a, b und v bilden ein Knotenniveau und c, b, u, w sind die Blätter.

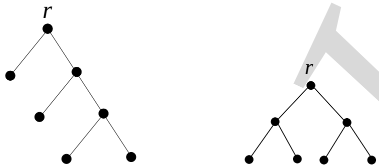


Bild 2.5: Diese beiden binären Bäume haben $|B| = 4$ Blätter; der erste Baum hat die Tiefe $3 = |B| - 1$, während der zweite die Tiefe $2 = \log_2 |B|$ hat.

Beweis:

Wir führen den Beweis mittels Induktion über die Tiefe $t = t(B)$.

Basis $t = 0$: In diesem Fall besteht B nur aus einem Knoten. Dieser Knoten ist ein Blatt, es gilt also $|B| = 1$. Da $1 \leq 2^0$ bzw. $\log_2 1 \leq 0$ gilt, ist die Behauptung für $t = 0$ wahr.

Induktionsschritt $t - 1 \mapsto t$: Sei die Behauptung bereits für alle binären Bäume der Tiefe $\leq t-1$ bewiesen und sei B ein beliebiger binärer Baum der Tiefe $t = t(B)$. Wir wollen zeigen, dass $|B| \leq 2^{t(B)}$ gilt, was äquivalent zu $t(B) \geq \log_2 |B|$ ist.

Sei r die Wurzel von B und seien $\{r, u\}$ und $\{r, v\}$ die beiden mit r inzidenten Kanten. Wir betrachten die beiden in u und v wurzelnden Teilbäume B_u und B_v von B. Beide diese Teilbäume haben Tiefe höchstens $t - 1$ und für die Anzahl der Blätter gilt: $|B_u| + |B_v| = |B|$. Nach Induktionsannahme gilt also

$$|B_u| \leq 2^{t(B_u)} \quad \text{und} \quad |B_v| \leq 2^{t(B_v)}.$$

Wir erhalten damit

$$|B| = |B_u| + |B_v| \leq 2^{t(B_u)} + 2^{t(B_v)} \leq 2 \cdot 2^{t-1} \leq 2^t. \qquad \square$$

2.5 Induktion und Entwurf von Algorithmen

Anwendungen der Induktion findet man in allen mathematischen Gebieten, von Mengenlehre bis Geometrie, von Differenzialrechnung bis Zahlentheorie. Sogar der Beweis des großen Fermat'schen Satzes (Andrew Wiles, 1993) verwendet die Induktion (neben vielen anderen Argumenten).

Die Induktion ist auch in der Informatik wichtig, denn rekursive Algorithmen sind

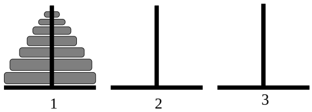


Bild 2.6: Das Spiel »Türme von Hanoi«.

induktive Beschreibungen von Objekten. Deshalb ist Induktion nicht nur für den *Beweis*, dass ein gegebener Algorithmus korrekt ist, geeignet – man kann sie auch für den *Entwurf* der Algorithmen benutzen! Das allgemeine Schema – als *dynamisches Programmieren* bekannt – ist das folgende:

Dynamisches Programmieren Um ein Problem zu lösen, löse zuerst *Teilprobleme* und kombiniere die Lösungen der Teilprobleme zu einer Lösung des ursprünglichen Problems.

2.5.1 Türme von Hanoi

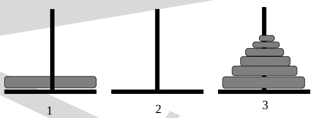
Das folgende Spiel »Türme von Hanoi« wurde 1883 von Eduard Lucas erfunden. Wir haben drei Stäbe 1, 2 und 3. Ursprünglich besitzt der 1. Stab n Ringe, wobei die Ringe in absteigender Größe auf dem Stab aufgereiht sind (mit dem größten Ring als unterstem Ring). Die Stäbe 2 und 3 sind leer (siehe Bild 2.6). Ein Zug besteht darin, einen zuoberst liegenden Ring von einem Stab zu einem anderen zu bewegen. Der Zug ist aber nur dann erlaubt, wenn der Ring auf einen größeren Ring gelegt wird oder wenn der Stab leer ist. Das Spiel ist erfolgreich beendet, wenn alle Ringe von Stab 1 nach Stab 2 bewegt wurden.

Man kann dieses Problem mittels Induktion lösen. Sei $P(n)$ die Aussage: »Wir haben einen Algorithmus, der das Problem für n Scheiben löst.«

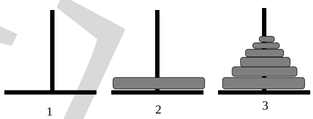
Induktionsbasis: $P(0)$ ist wahr, da wir dann überhaupt keine Scheiben haben.

Induktionsschritt: Angenommen, wir wissen bereits, wie man das Problem für $n-1$ Scheiben lösen kann und wollen einen Algorithmus für n Scheiben entwerfen. Dazu beobachten wir, dass das Problem, n Scheiben von Stab 1 zu Stab 2 zu befördern, sich wie folgt lösen lässt:

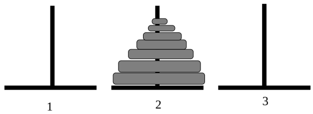
1. Zunächst verlege die obersten $n-1$ Scheiben von Stab 1 zum als Hilfsstab genutzten Stab 3 (nach Induktionsannahme wissen wir, wie dies zu tun ist):



2. Dann verlege die jetzt zuoberst liegende Scheibe von Stab 1 nach Stab 2:



3. Anschließend verlege wieder $n-1$ Scheiben vom Hilfsstab 3 nach Stab 2 (nach Induktionsannahme wissen wir, wie dies zu tun ist):



Damit können wir ein rekursives Programm Hanoi$(n; a, b, c)$, das die gegebenen n Scheiben von Stab a auf Stab b befördert, wie folgt beschreiben.

Algorithmus 2.16: Hanoi$(n; a, b, c)$

1. Rufe Hanoi$(n-1; a, c, b)$ auf (befördere die obersten $n-1$ Scheiben von Stab a zu Stab c).
2. Verlege die zuoberst liegende Scheibe von Stab a nach Stab b.
3. Rufe Hanoi$(n-1; c, b, a)$ auf (verlege $n-1$ Scheiben vom Hilfsstab c nach Stab b).

Zu beachten ist dabei, dass sich die Rollen der drei Stäbe (Ausgangs-, Ziel- und Hilfsstab) ständig wechseln.

2.5.2 Das Rucksackproblem

Für diejenigen Leser, die das vorherige Beispiel als zu »spielerisch« empfunden haben, geben wir eine ernsthaftere algorithmische Anwendung der Induktion an.

Eine mathematische Modellierung einer alltäglichen Situation: Geht ein Bergsteiger auf eine mehrtägige Wanderung, hat er sich reiflich zu überlegen, welche Dinge er unbedingt mitführen sollte. Dabei muss er die begrenzte Tragfähigkeit seines Körpers beachten, sein Rucksack sollte also nicht zu schwer sein.

Kann unser Bergsteiger unter n Gegenständen auswählen, hat jeder Gegenstand j eine bestimmte Masse (oder Gewicht) g_j und einen für die Wanderung wichtigen Wert w_j, es ergibt sich somit ein Optimierungsproblem. Zusätzlich wollen wir die Tragfähigkeit des Bergsteigers nicht außer Acht lassen und nehmen an, dass der Bergsteiger höchstens eine Masse von G kg mit sich herumschleppen kann.

In der mathematischen Sprache klingt das Problem so: Es sind m Objekte $1, 2, \ldots, m$ mit Gewichten $g_1, g_2, \ldots, g_m$ und Werten $w_1, w_2, \ldots, w_m$ vorgegeben, ebenso wie eine Gewichtsschranke G (die Tragfähigkeit des Bergsteigers), alle sind positive natürliche Zahlen. D.h. die Eingabe besteht aus $2m+1$ natürlichen Zahlen (Gewichte, Werte und die Gewichtsschranke). Der Rucksack ist mit einer Auswahl von Objekten so zu bepacken, dass einerseits die Gewichtsschranke G nicht überschritten wird und dass andererseits der Gesamtwert maximal ist. Wir suchen also unter Beachtung der Gewichtsschranke G eine Auswahl der Objekte mit maximalem Wert, d. h. eine Teilmenge $I \subseteq \{1, \ldots, m\}$ der Objekte mit

$$\sum_{i \in I} g_i \leq G, \quad \text{so dass der Gesamtwert} \quad \sum_{i \in I} w_i \quad \text{größtmöglich ist}\,.$$

Einfachheitshalber betrachten wir ein etwas leichteres Problem: Wir wollen nur den Gesamtwert

$$P(m, G) := \max \Big\{ \sum_{i \in I} w_i \;:\; I \subseteq \{1, \ldots, m\} \text{ und } \sum_{i \in I} g_i \leq G \Big\}$$