1. Zahlensysteme

Bei der heute üblichen Schreibweise für Zahlen handelt es sich um ein *Stellenwertsystem* zur Basis 10, d. h. jede der auftretenden Ziffern wird noch mit einer Wertigkeit versehen, die sich aus der relativen Position ergibt:

$$302,15 = 3 \cdot 10^2 + 0 \cdot 10^1 + 2 \cdot 10^0 + 1 \cdot 10^{-1} + 5 \cdot 10^{-2}$$
$$= 3 \cdot 100 + 2 + \tfrac{1}{10} + \tfrac{5}{100}$$

Auf diese Weise kann man mit 10 Ziffern beliebig große (und beliebig kleine) Zahlen auf einheitliche Weise darstellen. Die Vorteile gegenüber anderen Verfahren (z. B. römische Zahlzeichen oder einfache "Strichlisten") sind offensichtlich - nicht nur für das Hinschreiben, sondern vor allem für das schriftliche Rechnen!

(Beachte: Voraussetzung für die Entwicklung von Stellenwertsystemen war die Erfindung der Null!)

Das **Aussprechen** von Zahlwörtern ist ein ganz anderes, vorwiegend sprachliches Problem. Dabei gibt es in vielen Sprachen eigenartige Besonderheiten, z. B. die Vertauschung von Einer- und Zehnerstelle im Deutschen oder die Umschreibungen für 70, 80 und 90 im Französischen. Hier soll nur auf einen Punkt hingewiesen werden, der häufig zu Missverständnissen bei englischen Texten führt. Dazu ist in der folgenden Tabelle die Benennung großer Zahlen auf deutsch und englisch gegenübergestellt:

10^6	Million	million
10^9	Milliarde	billion
10^{12}	Billion	trillion
10^{15}	Billiarde	quadrillion

(Im Französischen werden etwa dieselben Begriffe wie im Deutschen verwendet.) Früher galt obige Tabelle nur für das *amerikanische* Englisch, während es in Großbritannien die "milliard" gab und auch die "billion" gleichbedeutend mit der deutschen Billion war (noch bis in die 80er Jahre). Seitdem hat sich der Sprachgebrauch aber nahezu komplett an das amerikanische Englisch angeglichen. Bei älteren Texten muss man mit der "billion" also besonders vorsichtig sein.

Stellenwertsysteme kann man statt zur Basis 10 auch zu der jeder beliebigen anderen Basis (natürlich ganzzahlig und ≥ 2) bilden. Ein Beispiel zur Basis 7:

$$452,3_7 = 4 \cdot 7^2 + 5 \cdot 7^1 + 2 \cdot 7^0 + 3 \cdot 7^{-1}$$

Dezimal entspricht das also der Zahl

$$4 \cdot 49 + 35 + 2 + \tfrac{3}{7} = 233\tfrac{3}{7}.$$

Man beachte, dass bei einer Zahl in der 7-er-Darstellung nur die Ziffern 0 bis 6 auftreten dürfen! Bei Benutzung einer Basis > 10 muss man dagegen zunächst entsprechend viele neue Ziffern erfinden.

Es ist bekannt, dass die Dezimalbruchentwicklung **rationaler** Zahlen (das sind solche, die als Quotient zweier ganzer Zahlen entstehen) entweder nach endlich vielen Stellen abbricht oder periodisch wird, z. B.

$$\frac{1}{7} = 0,142857142857142857... = 0,\overline{142857}$$

$$\frac{1}{22} = 0,04545454545... = 0,0\overline{15}$$

Dagegen besitzen **irrationale** Zahlen (wie z. B. $\sqrt{2}$) stets unendliche, nicht-periodische Darstellungen. Da die Unterscheidung zwischen rationalen und irrationalen Zahlen grundsätzlicher Natur ist, gilt dies für alle Basen. (Genaueres zu rationalen und irrationalen Zahlen folgt in einer späteren Vorlesung.)

Ob die Darstellung einer rationalen Zahl endlich oder periodisch ist, hängt dagegen wesentlich von der Basis ab. Eine rationale Zahl

$$r = \frac{p}{q}, \quad p \text{ und } q \text{ ganzzahlig}$$

hat z. B. im 10er-System genau dann eine endliche Entwicklung, wenn der Nenner q keine anderen Primfaktoren als 2 und 5 besitzt.

Gewissensfrage: Wer erinnert sich noch daran, wie man einen periodischen Dezimalbruch wieder in einen "gemeinen Bruch" umrechnet?

2. Das Binärsystem

Neben dem Dezimalsystem sind für Informatiker vor allem noch die Systeme zur Basis 2 ("Binärsystem") und zur Basis 16 ("Hexadezimalsystem", siehe nächster Abschnitt) wichtig. Das Binärsystem kommt mit nur zwei Ziffern 0 und 1 aus. Die Binärdarstellung ganzer Zahlen ist wahrscheinlich vielen geläufig. Wie mit jedem Stellenwertsystem kann man aber natürlich auch gebrochene Zahlen darstellen:

$$11010,1011_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4}$$

Dezimal ergibt das

$$16 + 8 + 2 + \tfrac{1}{2} + \tfrac{1}{8} + \tfrac{1}{16} = 26,6875$$

Eine rationale Zahl $r = p/q$ hat im Binärsystem nur genau dann eine endliche Darstellung, wenn der Nenner keine Primfaktoren außer der 2 besitzt (also eine Zweierpotenz ist). So wird etwa der Wert $\frac{1}{10}$, der eine endliche Dezimalentwicklung hat, im Binärsystem schon periodisch:

$$\frac{1}{10} = 0,1_{10} = 0,000110011001100... = 0,0\overline{0011}_2$$

Das hat ganz praktische Konsequenzen: Wenn in einem Programm ein Statement

```
x=0.1
```

steht, so ergibt sich bereits bei der Umwandlung in die interne Binärdarstellung der erste Rundungsfehler!

<u>Hinweis:</u> Das bekannte Phänomen, dass Dezimalbruch-Darstellungen nicht immer eindeutig sind, da man z. B. die Zahl 1 auch als $0,\overline{9}$ schreiben kann, tritt bei allen Basen auf. In der Binärdarstellung gilt z. B. $1 = 0,\overline{1}$.

<u>Empfehlung:</u>
Man übe die schriftliche Ausführung der vier Grundrechenarten im Binärsystem, z. B.:

1011,101 + 101,1 = ...

10001,001 - 101,1 = ...

1,01 · 10,11 = ...

11,0111 : 10,11 = ...

3. Das Hexadezimalsystem

Intern arbeiten Computer (bzw. deren Prozessoren) praktisch ausschließlich mit der Binärdarstellung. Für die Arbeit mit Bleistift und Papier oder die Benutzung im Programmcode ist diese jedoch sehr unpraktisch, da selbst relativ kleine Zahlen schon sehr viele Stellen benötigen. Stattdessen benutzt man daher lieber das Hexadezimalsystem (Basis 16), das gelegentlich auch noch als "Sedezimalsystem" bezeichnet wird. (Das ist zwar sprachlich eigentlich korrekter, hat sich aber trotzdem nicht durchgesetzt.) Kurz spricht man üblicherweise von Hex-Zahlen.

Als Ziffern für die Werte 10 bis 15 nimmt man die ersten sechs Buchstaben des Alphabets, also $A = 10$, $B = 11$, $C = 12$, $D = 13$, $E = 14$, $F = 15$. (Groß- oder Kleinschreibung sollte dabei eigentlich egal sein, ältere Software-Produkte akzeptieren aber manchmal nur Großbuchstaben als Hex-Ziffern.) Das liefert eine wesentlich kompaktere Darstellung, die sich andererseits aber sehr einfach in die Binärdarstellung umrechnen lässt. (Je vier Binärziffern lassen sich zu einer Hex-Ziffer zusammenfassen und umgekehrt.)

Beispiel: $46476_{10} = B58C_{16} = 1011\ 0101\ 1000\ 1100_2$

Ein typischer Anwendungsfall für die Benutzung von Hex-Zahlen ist die differenzierte Vorgabe von 24-Bit-Farbwerten in Programmen oder HTML-Dokumenten. Für die Standardfarben gibt es i. a. einprägsame Kurznamen, für feinere Nuancen kann man die Rot-, Grün- und Blauwerte aber exakt vorgeben. Typischer Eintrag im BODY-Tag eines HTML-Dokuments:

```
BGCOLOR="#C0F0F0"
```

Das Lattenkreuz kennzeichnet den Wert als Hex-Wert. Die ersten beiden Stellen liefern den Rotanteil (von 0-255), die nächsten beiden den Grünanteil und die letzten beiden den Blauanteil. `#FFFFFF` bedeutet weiß, `#000000` schwarz. Der Wert `#F0F0F0` würde also ein sehr helles, fast schon weißes Grau liefern. Durch die Reduktion des Rot-Anteils entsteht ein helles, "pastelliges" Cyan. In Dezimaldarstellung lautet derselbe Wert 12644592, dem man die resultierende Farbe deutlich weniger einfach ansehen kann.

Für Puristen: Das BGCOLOR-Schlüsselwort gilt nach der aktuellen HTML-Spezifikation als veraltet, daher sicherheitshalber noch die Stylesheet-Syntax:

```
BODY {background-color: #C0F0F0}
```

<u>Beachte:</u> Auch fast alle Programmiersprachen erlauben die Angabe von Farbwerten in dieser Form, wobei aber üblicherweise die umgekehrte Byte-Reihenfolge zu beachten ist (erst Blau, dann Grün, dann Rot). Außerdem ist die Symbolik zur Kennzeichnung von Hex-Werten natürlich sprachspezifisch unterschiedlich.

<u>Hinweis:</u> Die meisten Taschenrechner, die einen Hex-Modus haben (z. B. der zu den Bordmitteln von Windows gehörige), arbeiten in diesem Modus nur mit ganzen Zahlen. Das ist auch fast immer ausreichend, obwohl im Hexadezimalsystem gebrochene Zahlen natürlich genauso gut dargestellt werden können wie in jedem anderen.

4. Operatoren, Präzedenz, Gleichheitszeichen

Jeder kennt die Regel "Punktrechnung geht vor Strichrechnung", die besagt, dass in einer Rechenaufgabe wie

$$13 \cdot 7 + 9 : 5$$

zuerst die Multiplikation ($13 \cdot 7$) und die Division ($9 : 5$) jeweils für sich auszuführen und anschließend die Ergebnisse zu addieren sind. Andere Ausführungs-Reihenfolgen müssen explizit durch Klammersetzung erzwungen werden. In jeder Programmiersprache gibt es weitere solche "Präzedenzregeln", die die Auswertung ungeklammerter Ausdrücke in allen Fällen regeln, insbesondere wenn außer den vier Grundrechenarten noch weitere Operatoren enthalten sind (Vergleichsoperatoren, logische Operatoren etc. ð siehe Programmiervorlesung).

Im folgenden soll nur kurz auf die unterschiedliche Verwendung des Gleichheitszeichens in der Mathematik und beim Programmieren eingegangen werden.

In fast allen Programmiersprachen tritt das Gleichheitszeichen in zwei Hauptfunktionen auf:

als "Zuweisungsoperator" (in Java und C das einfache Gleichheitszeichen)

```
x = a + b
```

als "Vergleichsoperator" (in Java und C das doppelte Gleichheitszeichen)

```
if (a==b) ...
```

Beim Zuweisungsoperator wird der rechtsstehende Ausdruck berechnet und der linksstehenden Variablen zugewiesen. Rechte und linke Seite sind in diesem Falle also nicht gleichberechtigt (wie sonst bei Gleichungen). Daher ist z. B. auch

```
x = x + 7
```

eine korrekte und sinnvolle Zuweisung, die man aber nicht als Gleichung im üblichen Sinn lesen sollte.

In mathematischen Texten hat das Gleichheitszeichen normalerweise nicht den Charakter eines Operators, sondern stellt nur die Gleichheit zweier Ausdrücke fest. Aber auch dabei gibt es einige Nuancen, z. B. kann es sich um eine behauptete (und bewiesene) Gleichheit handeln, oder um eine geforderte Gleichheit oder auch um eine Gleichheit "per definitionem".

Typisches Beispiel für eine behauptete und bewiesene Gleichheit ist der Satz des Pythagoras:

Wenn a und b die Katheten eines rechtwinkligen Dreiecks sind und c die Hypotenuse ist, **dann** gilt $a^2 + b^2 = c^2$.

Als Beispiel für eine geforderte Gleichheit kann man etwa die Nullstellensuche bei Polynomen betrachten. Gegeben sei das Polynom $f(x) = 2x^5 - x^4 + 13x^2 + 3x - 7$, gesucht sind alle x-Werte, für die $f(x) = 0$ gilt. Hier wird nicht etwa behauptet, dass $f(x)$ immer den Wert Null liefert, sondern man möchte wissen, für welche x der "Wunsch" $f(x) = 0$ in Erfüllung geht. Solch gezieltes "Gleichsetzen" zweier Ausdrücke wird manchmal durch ein Ausrufezeichen über dem Gleichheitszeichen verdeutlicht:

$$f(x) = 2x^5 - x^4 + 13x^2 + 3x - 7 \overset{!}{=} 0.$$

Ebenso benutzt man manchmal einen Doppelpunkt vor dem Gleichheitszeichen, um Definitionsgleichungen zu kennzeichnen, etwa:

$$\cosh(x) := \tfrac{1}{2}(e^x + e^{-x})$$

$$n! := n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 3 \cdot 2 \cdot 1$$

In beiden Fällen geht es darum, dass auf der linken Seite eine neue Funktion, Kurzschreibweise oder sonstige Symbolik steht, deren Bedeutung durch die rechte Seite definiert wird.

<u>Beachte:</u> Diese typographischen Zusätze sollen nur verdeutlichen, welche Rolle die Gleichung im jeweiligen Kontext spielt und dienen damit dem besseren Verständnis. Sie ändern aber nicht die Bedeutung des Gleichheitszeichens an sich, können also immer auch wegbleiben - insbesondere, wenn aus dem Zusammenhang heraus keine Missverständnisse zu befürchten sind.