Division mit Rest

Die **Division mit Rest** oder der **Divisionsalgorithmus** ist ein mathematischer Satz aus der Algebra und der Zahlentheorie. Er besagt, dass es zu zwei Zahlen n und $m \neq 0$ eindeutig bestimmte Zahlen a und b gibt, für die

$$n = m \cdot a + b, \quad 0 \leq b < |m|$$

gilt. Die Zahlen a und b lassen sich durch die schriftliche Division ermitteln.

Die Division mit Rest ist auch für Polynome definiert. Die allgemeinste mathematische Struktur, in der es eine Division mit Rest gibt, ist der euklidische Ring.

Inhaltsverzeichnis

Natürliche Zahlen
- Beispiele
- Bestimmung des Restes für spezielle Teiler

Ganze Zahlen
- Beispiel
- Implementierung in Computersystemen

Modulo
- Beispiele

Grundrechenarten modulo einer natürlichen Zahl
- Beispiele

Verallgemeinerung: Reelle Zahlen

Polynome

Anwendung
- Programmierung
- Weitere Anwendungen

Siehe auch

Literatur

Weblinks

Natürliche Zahlen

Wenn zwei natürliche Zahlen, der Dividend a und der Divisor b (ungleich 0), mit Rest dividiert werden sollen, wenn also

$$a : b$$

berechnet werden soll, so wird gefragt, wie man die Zahl a als Vielfaches von b und einem „kleinen Rest“ darstellen kann:

$$a = b \cdot c + r$$

Hier ist c der so genannte *Ganzzahlquotient* und r der *Rest*. Entscheidende Nebenbedingung ist, dass r eine Zahl in $\{0, 1, \ldots, b-1\}$ ist. Hierdurch wird r eindeutig bestimmt.

Der **Rest** ist also die Differenz zwischen dem Dividenden und dem größten Vielfachen des Divisors, das höchstens so groß ist wie der Dividend. Ein Rest ungleich 0 ergibt sich folglich genau dann, wenn der Dividend kein Vielfaches des Divisors ist. Man sagt auch: Der Dividend ist nicht durch den Divisor teilbar, weshalb ein Rest übrigbleibt.

Liegt der Divisor fest, so spricht man beispielsweise auch vom *Neunerrest* einer Zahl, also dem Rest, der sich bei Division dieser Zahl durch neun ergibt.

Beispiele

- 7 : 3 = 2, Rest 1, da 7 = 3 × 2 + 1 („drei passt zweimal in sieben und es bleibt eins übrig“ – der Rest ist also eins)
- 2 : 3 = 0, Rest 2, da 2 = 3 × 0 + 2
- 3 : 3 = 1, Rest 0, da 3 = 3 × 1 + 0

Bestimmung des Restes für spezielle Teiler

Häufig kann man den Rest an der Dezimaldarstellung ablesen:

- Bei Division durch 2: Der Rest ist 1, wenn die letzte Ziffer ungerade ist, bzw 0, wenn die letzte Ziffer gerade ist.
- Bei Division durch 3: Der Rest ist gleich dem Rest, den die iterierte Quersumme bei Division durch 3 lässt.
- Bei Division durch 5: Der Rest ist gleich dem Rest, den die letzte Ziffer bei Division durch 5 lässt.
- Bei Division durch 9: Der Rest ist gleich dem Rest, den die iterierte Quersumme bei Division durch 9 lässt.
- Bei Division durch 10: Der Rest ist die letzte Ziffer.

Ähnliche, wenn auch etwas kompliziertere Regeln existieren für etliche weitere Teiler.

Ganze Zahlen

Ist b eine negative ganze Zahl, dann gibt es keine natürlichen Zahlen zwischen 0 und $b - 1$, die für den Rest r in Frage kämen. Es gibt drei Möglichkeiten:

1. Stattdessen kann man fordern, dass der Rest r zwischen 0 und $|b| - 1$ (dem Betrag von b minus 1) liegt.
2. Alternativ kann man aber auch verlangen, dass der Rest r zwischen $b + 1$ und 0 liegt, also dasselbe Vorzeichen hat wie b.
3. Eine dritte Möglichkeit ist, den betragskleinsten Rest r zu wählen. Diese Variante liefert für $a = b \cdot c + r$ die beste Näherung $b \cdot c$ für a.

Beispiel

Dividiert man negative Zahlen, ergibt sich folgendes Bild:

```
 7 :  3 =  2 Rest  1
−7 :  3 = −2 Rest −1
```

Übertragen auf negative Teiler, folgt:

```
 7 : −3 = −2 Rest  1
−7 : −3 =  2 Rest −1
```

(Hierbei wird für die Wahl von Quotient und Rest zunächst so getan, als gäbe es keine Vorzeichen, sie werden sozusagen nach der „eigentlichen Berechnung wieder hinzugefügt“). Als Ganzzahlquotient wird hier immer ein Wert bestimmt, dessen Betrag nicht größer als der Betrag des (rationalen) Quotienten ist. Der Rest und sein Vorzeichen folgen aus der Wahl des Quotienten.

Wie groß der Rest bei einer Division nun ausfällt, ist eigentlich Geschmackssache. Denn es steht jedem frei, nur einen Teil einer gegebenen Größe zu teilen, den verbleibenden *Rest* erklärt er einfach zum „Rest“. Lassen wir hierbei auch zu, dass „Schulden“ gemacht werden dürfen, sind beispielsweise alle folgenden Ergebnisse zulässig:

```
 7 :  3 =  1 Rest  4
 7 :  3 =  2 Rest  1
 7 :  3 =  3 Rest −2
```

oder

```
−7 :  3 = −1 Rest −4
−7 :  3 = −2 Rest −1
−7 :  3 = −3 Rest  2
```

Zur Normierung wird in der Mathematik die Konvention verwendet, die Vorzeichen der Reste aus denen der Teiler zu beziehen, wie in den folgenden Beispielen dargestellt:

```
 7 :  3 =  2 Rest  1
−7 :  3 = −3 Rest  2
 7 : −3 = −3 Rest −2
−7 : −3 =  2 Rest −1
```

Hierbei kann die Zugehörigkeit einer Zahl zu einer Restklasse direkt am Rest abgelesen werden.

Implementierung in Computersystemen

DIV- und MOD-Befehle bzw. Operatoren (für ganzzahlige Division und Restbildung) sind in den meisten Programmiersprachen (und sogar in CPUs) genau diesem Alltagsansatz entsprechend implementiert.

Einige Programmiersprachen und Computeralgebrasysteme tragen dieser Vielfalt von Konventionen Rechnung, indem sie zwei Modulo- oder Rest-Operatoren zur Verfügung stellen. Im Beispiel Ada hat:

- (A **rem** B) dasselbe Vorzeichen wie A und einen Absolutwert kleiner als der Absolutwert von B
- (A **mod** B) dasselbe Vorzeichen wie B und einen Absolutwert kleiner als der Absolutwert von B

Siehe auch: Liste von Operatoren für den Rest einer Division

Modulo

Modulo berechnet den Rest b der Division n geteilt durch m. Man kann eine Funktion definieren, die jedem Zahlenpaar (n, m) einen eindeutigen Teilerrest b zuordnet. Diese nennt man *Modulo* (von lat. modulus, Kasus Ablativ, also: ‚(gemessen) mit dem Maß (des …)‘; siehe auch wikt:modulo) und kürzt sie meistens mit *mod* ab.

In vielen Programmiersprachen wird die Funktion durch % (Prozentzeichen) dargestellt und als Operator behandelt. Frühe Programmiersprachen kannten den Operator *mod* noch nicht, nur den Datentyp des ganzzahligen Werts *integer* (abgekürzt INT); darum wurde der Divisionsrest nach der Formel (n/m-INT(n/m))*m errechnet und wegen der damaligen Rechenungenauigkeit beim Dividieren dann auf den ganzzahligen Wert gerundet.

Wir betrachten die Funktion

$$\mathrm{mod}\colon \mathbb{Z} \times (\mathbb{Z} \setminus \{0\}) \to \mathbb{Z}, \quad (a, m) \mapsto a \bmod m := a - \left\lfloor \frac{a}{m} \right\rfloor \cdot m.$$

Die Gaußklammer $\lfloor \cdot \rfloor$ bezeichnet die größte ganze Zahl, die nicht größer als die Zahl in der Gaußklammer ist, also ohne den Rest der Division a / m. Hier gilt stets

$$(a + km) \bmod m = a \bmod m \text{ für alle } k \in \mathbb{Z},$$

aber im Allgemeinen ist

$$(-a) \bmod m \neq -(a \bmod m), \text{ z. B. } (-2) \bmod 3 = 1 \neq -2 = -(2 \bmod 3).$$

Ist m positiv, so ist $a \bmod m \geq 0$ für alle a.

Neben dieser „mathematischen Variante“ wird oft auch eine ähnliche Funktion als Modulo bezeichnet, die für negative Argumente unterschiedliche Ergebnisse liefert und „symmetrische Variante“ genannt wird:

$$(a \bmod m) := a - m \cdot (a \operatorname{div} m)$$

Dabei bezeichnet $a \operatorname{div} m$ den zur Null hin gerundeten Quotienten a / m, gegeben durch $a \operatorname{div} m = \operatorname{sgn}(a)\ \operatorname{sgn}(m) \left\lfloor \frac{|a|}{|m|} \right\rfloor$, wobei $\operatorname{sgn}(x)$ die Vorzeichenfunktion bezeichnet. Für diese Variante gilt

$$(-a) \bmod m = -(a \bmod m),$$

aber im Allgemeinen

$$(a + km) \bmod m \neq a \bmod m, \text{ z. B. } (1 - 3) \bmod 3 = (-2) \bmod 3 = -2 \neq 1 = 1 \bmod 3.$$

$a \bmod m$ hat stets dasselbe Vorzeichen wie a, oder es gilt $a \bmod m = 0$.

Gilt $a \geq 0$ und $m > 0$, so ergeben beide Varianten dasselbe Ergebnis. In Programmiersprachen ist die implementierte Variante nicht einheitlich. So verwenden Ruby, Perl und Python die mathematische Variante, wohingegen C, Java, JavaScript und PHP die symmetrische einsetzen, was besonders wichtig bei Portierungen ist. Der in der Googlesuche enthaltene Rechner verwendet die mathematische Variante.

Steht in einer Sprache wie C(++) oder Java nur die symmetrische Variante zur Verfügung, kann man Ergebnisse nach der mathematischen Variante erhalten mit:

$a \bmod b$ = `((a % b) + b) % b`,

wobei % die symmetrische Modulooperation bezeichnet und $\bmod$ die mathematische.

Beispiele

- $17 \bmod 3 = 2$, da $17 = 5 \cdot 3 + 2$ („3 passt fünfmal in 17 und es bleiben 2 übrig“ – der Rest ist also 2)
- $2 \bmod 3 = 2$, da $2 = 0 \cdot 3 + 2$
- $3 \bmod 3 = 0$, da $3 = 1 \cdot 3 + 0$
- $-8 \bmod 6 = -8 - \left\lfloor \frac{-8}{6} \right\rfloor \cdot 6 = -8 - (-2) \cdot 6 = -8 + 12 = 4$

Aus $a \bmod m = b \bmod m$ folgt *nicht* $a = b$, sondern nur, dass sich a und b um ein ganzzahliges Vielfaches von m unterscheiden, also: $a = b + (k \cdot m)$ mit $k \in \mathbb{Z}$. Eine derartige Gleichung kann auch einfacher mit Hilfe der in der Zahlentheorie verbreiteten Kongruenzrelation geschrieben werden:

$$a \equiv b \pmod{m} \text{ oder auch } a \equiv_m b$$

Üblich ist auch die Schreibweise

$$a = b \pmod{m},$$

sowohl mit als auch ohne die Klammer, und zwar nicht nur, wo dies ohne die Klammer bei Betrachtung als Restoperator stimmen würde, etwa $1 = 11 \pmod{10}$, sondern auch sonst:

$$11 = 1 \pmod{10} \text{ oder gar}$$
$$11 = 21 \pmod{10}$$

Hintergrund ist hier, dass man dann die durch den Repräsentanten 1 eindeutig bestimmte Äquivalenzklasse der zu 1 modulo m kongruenten Zahlen als ein Element des Restklassenrings $\mathbb{Z}_m$ versteht; in diesem Sinne sind die beiden Ausdrücke als verschiedene Repräsentanten derselben Äquivalenzklasse tatsächlich *gleich*. In der Praxis ergibt sich kein Unterschied zur Verwendung des Kongruenzsymbols.

Grundrechenarten modulo einer natürlichen Zahl

Ist die Zahl m eine Primzahl, so kann man die aus den reellen Zahlen gewohnten Grundrechenarten mit anschließender Modulo-Berechnung anwenden und erhält einen sogenannten endlichen Körper.

Beispiele

- Modulo 3: $((2 \bmod 3) + (2 \bmod 3)) \bmod 3 = 4 \bmod 3 = 1 \bmod 3$
- Modulo 5: $((3 \bmod 5) \cdot (3 \bmod 5)) \bmod 5 = 9 \bmod 5 = 4 \bmod 5$

Verallgemeinerung: Reelle Zahlen

Sind a und b reelle Zahlen, a ungleich 0, dann kann man eine Division mit Rest folgendermaßen definieren: Der *ganzzahlige* Quotient c und Rest r im halboffenen Intervall $[0, |b|)$ sind diejenigen (eindeutig bestimmten) Zahlen, die die Gleichung $a = b \cdot c + r$ erfüllen.

Auch hier gibt es die Alternativen, dem Rest dasselbe Vorzeichen wie b zu geben oder den betragskleinsten Rest zu wählen. Letztere Alternative entspricht der Rundung: Die Division mit Rest von a durch 1 liefert eine ganze Zahl c und eine reelle Zahl r mit Betrag kleiner oder gleich 1/2, die die Gleichung $a = c + r$ erfüllen. Die Zahl c ist der auf ganze Zahlen gerundete Wert von a.

Es ist zu beachten, dass hierbei der Quotient nicht aus derselben Menge (der reellen Zahlen) genommen wird wie Divisor und Dividend.

Polynome

Bei der Division mit Rest für Polynome muss das als Divisor auftretende Polynom $f(X)$ aus dem Polynomring $R[X]$ (über R, einem kommutativen Ring mit 1) eine Voraussetzung erfüllen: Der Leitkoeffizient von $f(X)$ muss eine Einheit von R sein (insbes. ist $f(X)$ nicht das Nullpolynom). Unter dieser Bedingung gibt es zu jedem $g(X) \in R[X]$ eindeutig bestimmte Polynome $q(X), r(X) \in R[X]$ mit

$$g(X) = q(X) \cdot f(X) + r(X)$$
$$\deg(r) < \deg(f)$$

Dabei wird dem Nullpolynom ein kleinerer Grad als jedem anderen Polynom gegeben, beispielsweise $-\infty$.

Beispiel

$$2X^2 + 4X + 5 = (2X + 2)(X + 1) + 3 \in \mathbb{R}[X]$$

Die Polynome $q(X)$ und $r(X)$ lassen sich durch Polynomdivision bestimmen.

Anwendung

Programmierung

Die Division mit Rest (Modulo) wird in der Programmierung relativ häufig verwendet. Der entsprechende Operator heißt in unterschiedlichen Programmiersprachen oft `mod` oder %. Man kann etwa prüfen, ob eine gegebene Zahl x gerade ist, indem man folgende Abfrage durchführt: `if ((x mod 2) == 0)`. Modulo kann man auch nutzen, wenn man in einer Schleife lediglich bei jedem x-ten Schleifendurchlauf einen speziellen Programmcode ausführen will. Auch bei vielen Berechnungen und Algorithmen ist der Operator sinnvoll einsetzbar. Allgemein kann man mit `mod` prüfen, ob eine Zahl durch eine andere genau teilbar ist: Nur dann liefert der Modulo-Operator den Wert 0. Des Weiteren muss man in der Programmierung oft auf ganze Vielfache einer Zahl ergänzen (z. B. 4 Bytes) und kann durch den Modulo errechnen, wie viele „Pad-Bytes“ noch fehlen. Durch die Funktion `divMod` werden Ganzzahlquotient und Rest zusammen berechnet.

Beispiel

> Man programmiert eine Uhr und hat die Zeit als Sekundenwert seit 0 Uhr gegeben. Dann kann man den Sekundenwert mod 3600 berechnen. Ist dieser gleich 0, so weiß man, dass eine volle Stunde angefangen hat. Diese Information kann man nutzen, um z. B. ein akustisches Signal (Gong zur vollen Stunde) auszulösen. Mit der Berechnung Sekundenwert mod 60 erhält man die Sekunden seit der letzten vollen Minute, die oftmals von Digitaluhren als letzte zwei Stellen angezeigt werden.

Weitere Anwendungen

- Berechnung der Prüfziffer der Internationalen Standardbuchnummer
- Prüfsummen-Formel Luhn-Algorithmus zur Bestätigung von Identifikationsnummern wie Kreditkartennummern und kanadische Sozialversicherungsnummern
- Kalenderberechnung (die relativ komplizierte Berechnung des Osterdatums)
- Bei der International Bank Account Number (IBAN)
- In der Kryptografie, beim Diffie-Hellman-Schlüsselaustausch oder beim RSA-Kryptosystem

Siehe auch

- Hashfunktion und die dort genannten Verfahren
- Kleiner fermatscher Satz
- Satz von Euler
- Liste von Operatoren für den Rest einer Division

Literatur

- Kristina Reiss, Gerald Schmieder: *Basiswissen Zahlentheorie. Eine Einführung in Zahlen und Zahlenbereiche.* Springer-Verlag, Berlin u. a. 2005, ISBN 3-540-21248-5
- Peter Hartmann: *Mathematik für Informatiker Ein praxisbezogenes Lehrbuch.* 4. überarbeitete Auflage. Vieweg, Wiesbaden 2006, ISBN 3-8348-0096-1, S. 62, (online).
- Albrecht Beutelspacher, Marc-Alexander Zschiegner: *Diskrete Mathematik für Einsteiger Mit Anwendungen in Technik und Informatik.* Vieweg, Wiesbaden 2007, ISBN 978-3-8348-0094-7, S. 65, (online).

Weblinks

- Modulo online berechnen
- Java ist auch eine Insel: Der Restwert-Operator %
- Ganzzahlige Division und der Rest
- Division mit Rest – der heimliche Hauptsatz der Algebra (PDF; 86 kB)

- Video: *Division mit Rest (Teil 1)*. Pädagogische Hochschule Heidelberg (PHHD) 2012, zur Verfügung gestellt von der Technischen Informationsbibliothek (TIB), doi:10.5446/19767.
- Video: *Division mit Rest (Teil 2)*. Pädagogische Hochschule Heidelberg (PHHD) 2012, zur Verfügung gestellt von der Technischen Informationsbibliothek (TIB), doi:10.5446/19768.
