

1 Mengen

2 Zahlen

1 Mengen

2 Zahlen

Unter einer Menge verstehen wir jede Zusammenfassung M von bestimmten wohlunterschiedenen Objekten m unserer Anschauung oder unseres Denkens zu einem Ganzen.

(G. Cantor; Beiträge zur Begründung der transfiniten Mengenlehre; Mathematische Annalen; Bd. 46; 1895; S. 481-512)

- **Element:** Ein Objekt x in der Menge M ($x \in M$).
- **Enthalten:** Es gilt für alle $x \in M$ auch $x \in N$ ($M \subset N$).
- **Gleichheit:** Es gilt $M \subset N$ und $N \subset M$ ($M = N$).


```
Set([<element1>,<element2>,...])
```

- Es ist eine **ungeordnete** Menge von beliebigen Objekten.
- Mengen in Sage haben den Typ **set**.
- Leere Mengen: `leere_menge = Set([])`.
- Zugriff: `M[n]` (Menge M , $n \geq 0$)
- Intervallzugriff: `M[i:j]`.

Beispiele für Mengen

```
M1 = Set([x, 2,3,pi,sqrt(2)]); M1
```

??

```
var('y');M2 = Set([y,1,Set([1,y]),2,x]); M2
```

??

Befehle für Mengen I

- Anzahl der Elemente in einer Menge:

```
M1.cardinality()
```

??

- Zugriff:

```
M2[1] ; M2[1:4]
```

?? [y, 2, x]

Befehle für Mengen II

- Vereinigung, Differenz, Schnitt:

```
L1 = Set([1,2,3,a,b]); L2 = Set([a,b,c,4,5])  
L1.union(L2), L1.difference(L2), L1.intersection(L2)
```

?? 1, 2, 3, 4, a, c, b, 5, 1, 2, 3, b, a)

- Prüfen, ob ein Element enthalten ist:

```
a in L1, c in L1
```

?? True, False)

```
Set([1,y]) in M2
```

?? rue

Befehle für Mengen III

- Auswählen von Elementen mit bestimmten Eigenschaften

```
M = Set(range(1,15))  
filter(is_prime,M)
```

```
?? 2, 3, 5, 7, 11, 13]
```

- Erzeugen der Potenzmenge

```
list(powerset([1,2,3]))  
[s for s in Set([1..3]).subsets()]
```

```
?? [], [1], [2], [1, 2], [3], [1, 3], [2, 3], [1, 2, 3]] [, 1, 2, 3, 1, 2, 1, 3,  
2, 3, 1, 2, 3]
```


Filter - filter()

Ein Filter erzeugt eine Teilmenge aus einer größeren Menge.

```
M1 = filter(<f>, <M>)
```

- $f(x)$ ist eine Abbildung auf die Booleschen Werte **True/False**.
- $M1$ ist die Teilmenge die aus den Elementen $x \in M$ besteht, für die $f(x)$ eine wahre Aussage ergibt.

Beispiel für Filter I

```
M = Set(range(1,101))
def f(x): return bool(mod(x,2)==0)
M2 = Set(filter(f,M))
def f(x): return bool(mod(x,15)==0)
M15 = Set(filter(f,M))
M2.intersection(M15)
```

?? Alternative:

```
M2 = Set([m for m in M if mod(m,2)==0])
M15 = Set([m for m in M if mod(m,15)==0])
```


Beispiel für Filter II

Ist Menge A1 eine Teilmenge von Menge A?

```
A = Set(range(1,11))  
A1 = Set(range(1,3))  
A2 = Set(range(9,12))
```

```
A.intersection(A1) == A1
```

?? rue

```
A.intersection(A2) == A2
```

?? else

1 Mengen

2 Zahlen

Natürliche Zahlen \mathbb{N} (nach Peano)

Natürliche Zahlen $\mathbb{N} := \{0, 1, 2, 3, \dots\}$:

- 1 $0 \in \mathbb{N}$
- 2 Es gibt eine Nachfolgerabbildung $nf: \mathbb{N} \rightarrow \mathbb{N} \setminus \{0\}$
- 3 nf ist injektiv.
- 4 Ist $M \subset \mathbb{N}$ mit $0 \in M$ und folgt für alle $m \in M$ das $nf(m) \in M$ gilt, so ist $M = \mathbb{N}$.

Bemerkungen:

- Nachfolgefunktion: $nf(m) = m + 1$
- Es besteht ein enger Zusammenhang zwischen den natürlichen Zahlen und vollständiger Induktion.
- Sage: kein eigener Datentyp (aber: ganze Zahlen (Integer)).

Äquivalenzrelation

Sei M eine Menge. Eine **Äquivalenzrelation** R auf M ist eine Teilmenge

$$R \subseteq M \times M$$

mit den folgenden Eigenschaften (Schreibweise: $(x, y) \in R$, $x \sim_R y$, $x \sim y$):

- ❶ **Reflexivität:** für alle $x \in M$ gilt $x \sim x$.
- ❷ **Symmetrie:** für alle $x, y \in M$ folgt aus $x \sim y$ das $y \sim x$.
- ❸ **Transitivität:** für alle $x, y, z \in M$ und $x \sim y$, $y \sim z$ folgt $x \sim z$.

Äquivalenzklasse

- Sei \sim_R eine Äquivalenzrelation auf einer Menge M .
- Eine Teilmenge $A \subset M$ heißt **Äquivalenzklasse**, falls gilt:
 - (a) $A \neq \emptyset$.
 - (b) $x, y \in A \Rightarrow x \sim y$.
 - (c) $x \in A, y \in M, x \sim y \Rightarrow y \in A$.
- Eine Äquivalenzrelation zerlegt eine Menge in disjunkte Äquivalenzklassen.
- Andersrum definiert eine disjunkte Zerlegung einer Menge eine Äquivalenzrelation.
- Ein $a \in A$ ist ein **Repräsentant** der Äquivalenzklasse A . Man schreibt auch \bar{a} oder $a \bmod R$ für ein Äquivalenzklasse A .

Ganze Zahlen \mathbb{Z}

Ganze Zahlen: $\mathbb{Z} := \{0, 1, -1, 2, -2, \dots\}$

- Äquivalenzrelation auf $\mathbb{N} \times \mathbb{N}$:

$(m, n) \sim (p, q)$ genau dann, wenn $m + q = n + p$ gilt.

- Nichtnegative Zahlen: $(m, 0)$. Sie sind paarweise nicht äquivalent zueinander.
- Negative Zahlen: $(0, m)$.
- Die ganzen Zahlen \mathbb{Z} sind gegeben durch die Menge der Äquivalenzklassen.
- Addition:

$$\overline{(m, n)} + \overline{(u, v)} := \overline{(m + u, n + v)}$$

- Multiplikation:

$$\overline{(m, n)} \cdot \overline{(u, v)} := \overline{(mu + nv, mv + nu)}$$

Ganze Zahlen in Sage

Datentyp Integer.

Beispiele:

```
type(5), type(0), type(-5)
```

```
?? <type 'sage.rings.integer.Integer'>, <type  
'sage.rings.integer.Integer'>, <type 'sage.rings.integer.Integer'>) Division
```

```
type(5*4), type(5/4)
```

```
?? <type 'sage.rings.integer.Integer'>, <type  
'sage.rings.rational.Rational'>)
```


Division mit Rest

Seien $x \in \mathbb{Z}$, $a \in \mathbb{N}$. Dann gibt es eindeutig bestimmte Zahlen $n, r \in \mathbb{Z}$ mit $r \in \{0, 1, \dots, a-1\}$, so dass $x = na + r$ gilt.

Beispiele:

```
mod(45,7), floor(45/7)
```

?? 3, 6)

```
mod(-34,8), floor(-34/8)
```

?? 6, -5)

Äquivalenzrelation auf $\mathbb{Z} \times (\mathbb{Z} \setminus \{0\})$:

$(m, n) \sim (p, q)$ genau dann, wenn $mq = np$ gilt.

Statt (m, n) schreibt man $\frac{m}{n}$.

- Die Äquivalenzklasse $\overline{(0, n)}$, $n \in \mathbb{Z}$ ist die 0 in \mathbb{Q} .
- Mit (n, m) gehören auch alle Erweiterungen (kn, km) zu einer Ä.-klasse.
- Addition:

$$\overline{\left(\frac{m}{n}\right)} + \overline{\left(\frac{p}{q}\right)} = \overline{\left(\frac{mq + pn}{nq}\right)},$$

Multiplikation:

$$\overline{\left(\frac{m}{n}\right)} \cdot \overline{\left(\frac{p}{q}\right)} = \overline{\left(\frac{mp}{nq}\right)}.$$

Rationale Zahlen in Sage

Datentyp rational.

```
var('a,b,c,d'); (a/b+c/d).simplify_rational()
```

```
?? a*d + b*c)/(b*d)
```

```
bool(a/b+c/d == (a/b+c/d).simplify_rational())
```

```
?? rue
```


Eine **Gruppe** ist ein Paar (G, \cdot) bestehend aus einer Menge G und einer Verknüpfung \cdot auf G , d.h. einer Abbildung

$$\cdot : G \times G \rightarrow G, \quad (a, b) \mapsto a \cdot b$$

mit folgenden Eigenschaften

(G1) $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ für alle $a, b, c \in G$.

(G2) Es existiert ein $e \in G$ (*neutrales Element*) mit $e \cdot a = a$ für alle $a \in G$ und zu jedem $a \in G$ existiert ein $a' \in G$ (*inverses Element*) mit $a' \cdot a = e$.

abelsche Gruppe: $a \cdot b = b \cdot a$ für alle $a, b \in G$.

Eigenschaften einer Gruppe

- Für ein neutrales Element gilt auch $a \cdot e = a$ für alle $a \in G$.
- Es gibt genau ein neutrales Element $e \in G$.
- Zu jedem $a \in G$ ist das inverse Element $a' \in G$ eindeutig und wird durch a^{-1} bezeichnet.
- Es gilt auch $a \cdot a' = e$.
- Für abelsche Gruppen schreibt man oft $+$ statt \cdot . Das Inverse zu a wird dann mit $-a$, das Neutrale mit 0 bezeichnet.

Ein **Körper** ist ein Tripel $(K, +, \cdot)$ bestehend aus einer Menge K und zwei Verknüpfungen $+$ und \cdot mit folgenden Eigenschaften:

- (K1) $(K, +)$ ist eine abelsche Gruppe. (Das neutrale Element heie 0. Das inverse Element zu $a \in K$ sei $-a$.)
- (K2) $(K \setminus \{0\}, \cdot)$ sei eine abelsche Gruppe. (Das neutrale Element dazu sei 1.)
- (K3) Distributivgesetze

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(a + b) \cdot c = (a \cdot c) + (b \cdot c) \text{ fr alle } a, b, c \in K.$$

(Ein Krper ist ein kommutativer unitrer Ring)

Gruppen:

- $(\mathbb{Z}, +)$, die ganzen Zahlen mit Addition.
- $(\mathbb{Z}/n\mathbb{Z}, +)$, die Restklassen modulo n mit Addition.
- $(\mathbb{Q}, +)$, $(\mathbb{Q} \setminus \{0\}, \cdot)$
- $(\text{Add}(M, \mathbb{R}), +)$, die reellwertigen Funktionen auf einer Menge M mit punktweiser Addition.

Körper:

- Die rationalen Zahlen \mathbb{Q} mit den Verknüpfungen $+$ und \cdot .
- Die reellen Zahlen \mathbb{R} mit den Verknüpfungen $+$ und \cdot .
- Die komplexen Zahlen \mathbb{C} mit den Verknüpfungen $+$ und \cdot .
- Für p Primzahl $\mathbb{Z}/p\mathbb{Z}$, die Restklassen modulo p mit $+$ und \cdot .

Körper und Gruppen in Sage

- Die ganzen Zahlen \mathbb{Z} : ZZ
- Die rationalen Zahlen \mathbb{Q} : QQ
- Die reellen Zahlen \mathbb{R} : RR
- Die komplexen Zahlen \mathbb{C} : CC

```
QQ(5.01), RR(5/3)
```

```
?? 501/100, 1.666666666666667)
```

```
RR.is_field(), ZZ.is_field()
```

```
?? True, False)
```


Sei K ein Körper. Er heißt **angeordnet**, wenn es einen **Positivbereich** $P \subset K$ gibt mit

- Die Mengen P , $\{0\}$, und $-P := \{-x \mid x \in P\}$ sind disjunkt.
- $K = P \cup \{0\} \cup -P$.
- Aus $x, y \in P$ folgt $x + y \in P$ und $x \cdot y \in P$.

Man definiert:

$x > y$ genau dann, wenn $x - y \in P$,

$x \geq y$ genau dann, wenn $x - y \in P \cup \{0\}$.

Analog definiert man $<$ und \leq .

Sei K ein angeordneter Körper.

- **obere Schranke** $y \in K$: Für $M \subset K$, wenn für alle $x \in M$ die Relation $x \leq y$ gilt.
- nach oben **beschränkt**: Wenn eine Teilmenge M von K eine obere Schranke besitzt (analog **untere Schranke**).
- **Maximum** von M : Eine obere Schranke y einer Teilmenge $M \subset K$, wenn $y \in M$ (analog **Minimum**).
- **Supremum**: Die kleinstmögliche obere Schranke y einer Teilmenge $M \subset K$ (analog **Infimum**) (Nicht notwendigerweise in M oder K).

- Sei M die Menge aller Teilmengen von \mathbb{Q} mit oberer Schranke.
- Äquivalenzrelation: Zwei Elemente aus M seien äquivalent, wenn sie dieselben Mengen von oberen Schranken haben.
- Die entstehenden Äquivalenzklassen nennt man **reelle Zahlen**.

Bemerkungen

- Es lassen sich die üblichen Verknüpfungen auf \mathbb{R} definieren.
- Die reellen Zahlen können auch als Vervollständigung von \mathbb{Q} definiert werden oder durch den Dedekindschen Schnitt.
- Die rationalen Zahlen sind als Äquivalenzklassen der einelementigen Mengen $\{x\}$, $x \in \mathbb{Q}$ enthalten.

Reelle Zahlen in Sage - Gleitkommazahlen

Datentyp `RealNumber`

Problem: keine exakte Darstellung möglich => Approximation: die Gleitkommazahlen

- Gleitkommazahlen haben in Sage den Datentyp `float`.
- Die Anzahl der signifikanten Stellen kann durch die Objekt-Methode `n(digits=<digits>)` gesteuert werden

Beispiel: $\sqrt{2}$

```
(sqrt(2)).n(digits=200)
```

?? .4142135623730950488016887 242096980785696718753769480731767

- **Relativer Fehler:** Sei $rd(x)$ die 'gerundete' Gleitkomma-Zahl zu $x \in \mathbb{R}$.
Dann gilt

$$\frac{|x - rd(x)|}{|x|} \leq \varepsilon$$

mit $\varepsilon = b^{1-t}$ (b =Basis, t =Anzahl signifikante Stellen).

- Rundungsfehler können sich innerhalb eines Verfahrens verstärken.
(*Fehlerfortpflanzung*).
- Katastrophale Auswirkungen möglich! Z.B. Absturz der Arianne-Rakete 1996.

Warnung! Die Subtraktion zweier fast gleichgroßer Gleitkommazahlen ist zu vermeiden.

Rechnen mit Zahlen I

- Berechnen einer numerischen Näherung

```
(pi).n(digits=22), (exp(1)).n(digits=22)
```

?? 3.141592653589793238463, 2.718281828459045235360)

- Sage rechnet mit floats, sobald mindestens eine Zahl in Gleitkommadarstellung gegeben ist

```
(1.0+(5/2*3))/(1/7+7/9)^2
```

?? 0.0286860879905

```
(1+(5/2*3))/(1/7+7/9)^2
```

?? 7473/6728

Rechnen mit Zahlen II

- Viele Sage Funktionen liefern numerische Werte beim Einsetzen von Gleitkommazahlen.

```
sqrt(64.0), sin(3.14), sin(7/5)
```

```
?? 8.000000000000000, 0.00159265291648683, sin(7/5))
```

- Ausdrücke werden nicht automatisch umgewandelt

```
2/3*sin(2), 0.6666666666666666*sin(2)
```

```
?? 2/3*sin(2), 0.6666666666666666*sin(2))
```

```
float(2/3*sin(2))
```

```
?? .60619828455045444
```



```
x = 10^(-2); ((1.0+x)-1.0)/x
```

```
?? .0000000000000000
```

```
x = 10^(-4); ((1.0+x)-1.0)/x
```

```
?? .99999999999999890
```

```
x = 10^(-16); ((1.0+x)-1.0)/x
```

```
?? .0000000000000000
```


Wichtige Funktionen für Zahlen

<code>abs</code>	Absolutbetrag
<code>ceil</code>	Aufrunden
<code>floor</code>	Abrunden
<code>round</code>	Runden
<code>sqrt</code>	Wurzel
<code>digits</code>	Anzahl Stellen

Darstellung von Gleitkommazahlen I

$$x = (-1)^s \cdot (0.a_1 a_2 \dots a_t) \cdot b^e, \quad a_1 \neq 0$$

- $b \in \mathbb{N} \setminus \{0, 1\}$ ist die Basis
- $a_1 \neq 0$ erzwingt die Eindeutigkeit der Darstellung.
- $s \in \{0, 1\}$ das Vorzeichen.
- Es sei $a_i \in \{0, 1, \dots, b-1\}$.
- t ist die Anzahl der *signifikanten Stellen*.
- x hat den Wert $(-1)^s b^e \sum_{k=1}^t a_k b^{-k}$.
- Man spricht von einer **b -adischen Darstellung** oder einer Darstellung zur Basis b .

Darstellung von Gleitkommazahlen II

Beispiele:

$$73 = 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

⇒ Binärdarstellung $1001001 = 0.1001001 \cdot 2^7$.

$$73 = 1 \cdot 8^2 + 1 \cdot 8^1 + 1 \cdot 8^0$$

⇒ Oktaldarstellung $111 = 0.111 \cdot 8^3$.

```
73.str(2)
```

```
?? 001001
```

```
73.str(8)
```

```
?? 11
```


Darstellung von Gleitkommazahlen III

$$2.45 = 1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} + 0 \cdot 2^{-5} + \dots$$

⇒ Binärdarstellung 10.01110...

$$2.45 = 2 \cdot 8^0 + 3 \cdot 8^{-1} + 4 \cdot 8^{-2} + 6 \cdot 8^{-3} + 3 \cdot 8^{-4} + 1 \cdot 8^{-5} + \dots$$

⇒ Oktaldarstellung 2.34631...

```
(2.45).str(2)
```

?? 0.011100110011001100110011001100110011001100110011010

```
(2.45).str(8)
```

?? .346314631463146320

Der Körper \mathbb{C} der **komplexen Zahlen**: Die Menge $\mathbb{R}^2 = \mathbb{R} \times \mathbb{R}$ mit

- Addition: $(k, l) + (n, m) = (k + n, l + m)$
- Multiplikation: $(k, l) \cdot (n, m) = (kn - lm, km + ln)$
- **$i := (0, 1)$** mit
 - $i^2 = (0, 1) \cdot (0, 1) = (-1, 0)$
 - $\forall (x, y) \in \mathbb{C} : (x, y) = x \cdot (1, 0) + y \cdot (0, 1) = x + iy$
- **Betrag**: $|z| = |(x, y)| := \sqrt{x^2 + y^2}$ (Sage: `abs`)

- **Fundamentalsatz der Algebra:** Jedes nicht konstante Polynom (mit komplexen Koeffizienten) hat mindestens eine Nullstelle in \mathbb{C} .
- Polarkoordinaten (r, φ) zu $(x, y) \in \mathbb{C}$

$$r := \sqrt{x^2 + y^2}, \quad \tan(\varphi) = \frac{y}{x}$$

- Es gilt: $z = (x, y)_{\text{Rechtwinklig}} = (r, \varphi)_{\text{Polar}} = re^{i\varphi}$

- Datentyp in Sage: `complex`
- Die imaginäre Einheit $i = (0, 1)$ ist in Sage `I`.

```
sqrt(-1), I^2
```

```
?? I, -1)
```

- Rechnen mit komplexen Zahlen

```
(1+2*I)*(4+I), (1/2+I)*(0.1+I/2)
```

```
?? 9*I + 2, -0.4500000000 + 0.3500000000*I)
```


- Keine automatische Trennung bzgl. Realteil und Imaginärteil
- `real()`: Realteil
- `imag()`: Imaginärteil

```
1/(sqrt(2)+I), real(1/(sqrt(2)+I)) + imag(1/(sqrt(2)+I))*I
```

```
?? /(sqrt(2) + I) 1/3*sqrt(2) - 1/3*I
```

```
real(1/(sqrt(2)+I)), imag(1/(sqrt(2)+I))
```

```
?? 1/3*sqrt(2), -1/3)
```