Einführung in Sage

Einheit 3

Jochen Schulz

Georg-August Universität Göttingen

24. Januar 2010

Aufbau

1. Mengen
2. Zahlen

Aufbau

Mengen

Unter einer ‚Menge' verstehen wir jede Zusammenfassung M von bestimmten wohlunterschiedenen Objekten m unserer Anschauung oder unseres Denkens (welche die ‚Elemente' von M genannt werden) zu einem Ganzen.

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

- Die Objekte heißen Elemente der Menge.
- Ist x ein Element der Menge M, so schreibt man $x \in M$.
- Man sagt, eine Menge M ist in einer Menge N enthalten, wenn für alle $x \in M$ auch $x \in N$ gilt. Man schreibt $M \subset N$.
- Gilt $M \subset N$ und $N \subset M$, so sind die beiden Mengen gleich. Man schreibt $M = N$.

Mengen in Sage

- Mengen in Sage haben den Typ `sets.set`.
- Es ist eine ungeordnete Menge von beliebigen Objekten.
- Mengen werden als Listen in der Funktion `Set([])` angegeben.
- Leere Mengen werden durch `leere_menge = Set([])` definiert.
- Einträge werden durch Kommata getrennt.
- (Nummerierter) Zugriff auf Elemente der Menge M kann durch `M[n]` erfolgen ($n \geq 0$)
- Mit `M[i:j]` können auch Intervalle ausgegeben werden.

Beispiele für Mengen

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

```
  {x, 2, 3, pi, sqrt(2)}
```

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

```
  {y, 1, 2, {y, 1}, x}
```

Befehle für Mengen I

- Die Anzahl der Elemente in einer Menge:

```
>> M1.cardinality()
```

```
  5
```

- Zugriff auf Elemente:

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

```
1
[1, 2, {y, 1}]
```

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, c, 5, a, b, 4}, {1, 2, 3}, {b, a})
```

- Prüfen, ob ein Element enthalten ist:

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

```
(True, False)
```

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

```
True
```

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

```
>> Set(powerset([1,2,3]))
```

```
Set of elements of [[], [1], [2], [1, 2], [3],
   [1, 3], [2, 3], [1, 2, 3]]
```

filter()

Äußerst nützlich und häufig verwendet ist es Mengen durch bestimmte Vorschriften als Teilmengen einer größeren Menge zu erzeugen. Durch den Aufruf

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

wird aus der Menge `M` eine Menge `M1` extrahiert, die aus den Elementen $x \in M$ besteht, für die `f(x)` eine wahre Aussage ergibt.
`f` ist eine Abbildung auf die Boolschen Werte True/False.

Beispiel - filter()

```
>> 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)
```

```
 {90, 60, 30}
```

```
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

Wie kann ich prüfen, ob eine Menge `A1` eine Teilmenge einer Menge `A` ist?

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

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

```
  True
```

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

```
  False
```

Aufbau

Natürliche Zahlen (nach Peano)

Die Menge $\mathbb{N}$ ist definiert durch:

1. $0 \in \mathbb{N}$
2. Es gibt eine Nachfolgerabbildung $nf : \mathbb{N} \to \mathbb{N} \setminus \{0\}$
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

- Man stelle sich die Nachfolgefunktion *nf* als $nf(m) = m + 1$ vor.
- Es besteht ein enger Zusammenhang zwischen den natürlichen Zahlen und vollständiger Induktion.
- Man identifiziert die so erzeugte Folge von Zahlen als $0, 1, 2, 3, \ldots$.
- Sie haben keinen eigenen Datentyp in Sage, aber es gibt 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. Für $(x, y) \in R$ schreibt man auch $x \sim_R y$ oder einfach $x \sim y$.

1. **Reflexivität:** für alle $x \in M$ gilt $x \sim x$.
2. **Symmetrie:** für alle $x, y \in M$ folgt aus $x \sim y$ das $y \sim x$.
3. **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 $\overline{a}$ oder $a \bmod R$ für ein Äquivalenzklasse A.

Ganze Zahlen $\mathbb{Z}$

Einführung der ganzen 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.
- Die Tupel der Form $(m, 0)$ sind paarweise nicht äquivalent zueinander. Dies sind die nichtnegativen Zahlen. Die negativen Zahlen werden durch $(0, m)$ identifiziert.
- Die ganzen Zahlen $\mathbb{Z}$ sind gegeben durch die Menge der Äquivalenzklassen.

Verknüpfungen

- 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

Ganze Zahlen in Sage haben den Datentyp `Integer`. Man kann sie addieren, subtrahieren und multiplizieren. Das Ergebnis ist wieder vom Typ `Integer`.

```
>> 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 $r \in \{0, 1, \ldots, a-1\}$, so dass $x = na + r$ gilt. (Beispiele: $x = 45$, $a = 7$ und $x = -34$, $a = 8$)

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

```
  6, 3
```

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

```
(6, -5)
```

rationale Zahlen $\mathbb{Q}$

Die rationalen Zahlen $\mathbb{Q}$ sind gegeben durch die folgende Äquivalenzrelation auf $\mathbb{Z} \times (\mathbb{Z} \setminus \{0\})$:

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

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

Rationale Zahlen in Sage haben den Datentyp `rational`. Man kann sie in Sage beliebig addieren, subtrahieren, multiplizieren und dividieren. Das Ergebnis ist wieder eine rationale Zahl. Die rationalen Zahlen bilden einen Körper.

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

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

```
>> a/b*c/d == (a/b*c/d).simplify_rational()
```

```
a*c/(b*d) == a*c/(b*d)
```

Gruppe

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

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

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 $a' \cdot a = e$.

Gilt zusätzlich $a \cdot b = b \cdot a$ für alle $a, b \in G$ so heißt die Gruppe *abelsch*.

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.

Körper

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 heiße 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

$$\begin{aligned} a \cdot (b + c) &= (a \cdot b) + (a \cdot c) \\ (a + b) \cdot c &= (a \cdot c) + (b \cdot c) \text{ für alle } a, b, c \in K. \end{aligned}$$

Beispiele

Gruppen:

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

Anordnung

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:

$$\begin{aligned} rclx > y \quad &\text{genau dann, wenn} \quad x - y \in P, \\ x \geq y \quad &\text{genau dann, wenn} \quad x - y \in P \cup \{0\}. \end{aligned}$$

Analog definiert man $<$ und $\leq$.

Schranken

Sei K ein angeordneter Körper.

- $y \in K$ heißt obere Schranke von $M \subset K$, wenn für alle $x \in M$ die Relation $x \leq y$ gilt.
- Hat eine Teilmenge M von K eine obere Schranke, so heißt M nach oben beschränkt (analog untere Schranke).
- Eine obere Schranke y einer Teilmenge M von K heißt Maximum von M, wenn $y \in M$ (analog Minimum).
- Die kleinstmögliche obere Schranke y einer Teilmenge M von K heißt Supremum (analog Infimum). Insbesondere müssen sie nicht Element der Menge M sein oder gar als Element von K existieren.

Reelle Zahlen

- Sei M die Menge aller Teilmengen von $\mathbb{Q}$ mit oberer Schranke.
- Zwei Elemente aus M seien äquivalent, wenn sie dieselben Mengen von oberen Schranken haben. Auf diese Weise kann eine Äquivalenzrelation definiert werden.
- Die entstehenden Äquivalenzklassen nennt man reelle Zahlen und die Menge dieser Zahlen bezeichnet man mit $\mathbb{R}$.

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.
- Der Datentyp in Sage ist `RealNumber`

Gleitkommazahlen I

Problem:
Die reellen Zahlen werden nicht exakt im Computer abgebildet. Es wird nur eine bestimmte Anzahl von Nachkommastellen betrachtet und die letzten Stellen gerundet.
Computer arbeiten also in der Regel nur mit Approximationen an die gesuchte reelle Zahl.
Beispiel: $\sqrt{2}$

```
>> (sqrt(2)).n(200)
```

```
1.4142135623730950488016887
2420969807856967187537694807317 67
```

Gleitkommazahlen II

Darstellung von Gleitkommazahlen:

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

- $b \in \mathbb{N} \setminus \{0, 1\}$ 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.

Gleitkommazahlen

Beispiele:

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

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

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

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

Rundungsfehler

- Sei $rd(x)$ die 'gerundete' Gleitkomm-Zahl zu $x \in \mathbb{R}$. Es gilt für den *relativen Fehler*

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

  mit $\varepsilon = b^{1-t}$.
- 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.

Beispiele

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

$\Rightarrow$ Binärdarstellung $10.01110\ldots$

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

$\Rightarrow$ Oktaldarstellung $2.34631\ldots$

Sage

- Gleitkommazahlen haben in Sage den Datentyp `float`.
- Gleitkommazahlen werden zur Basis 10 ausgegeben.
- Die Anzahl der signifikanten Stellen kann man durch die globale Variable `DIGITS` steuern (Default: 10). Möglich sind Werte zwischen 1 und $2^{31} - 1$. ?? .n ()
- Intern werden zusätzliche Schutzstellen verwendet. Z.B. wird bei DIGITS=10 intern mit ca. 19 Stellen gerechnet. ?? nachlesen

Rechnen mit Zahlen I

- Approximation durch `float`. Berechnen einer numerischen Näherung zu einem Ausdruck.

```
>> (pi).n(22), (exp(1)).n(22)
```

```
(3.14159, 2.71828)
```

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

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

```
10.0286860879905
```

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

```
67473/6728
```

Rechnen mit Zahlen II

- Ausdrücke werden nicht automatisch umgewandelt

```
>> 2/3*sin(2), 0.666666666666666*sin(2)
```

```
(2/3*sin(2), 0.666666666666666*sin(2))
```

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

```
0.606198284550454 44
```

- 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))
```

Auslöschung

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

```
  1.000000000000000
```

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

```
 0.999999999999890
```

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

```
  0.000000000000000
```

Wichtige Funktionen für Zahlen

`abs`	Absolutbetrag
`ceil`	Aufrunden
`floor`	Abrunden
`round`	Runden
mehr! `sqrt`	Wurzel

Komplexe Zahlen

Die Menge $\mathbb{R}^2 = \mathbb{R} \times \mathbb{R}$ versehen mit der Addition

$$(k, l) + (n, m) = (k + n, l + m)$$

und der Multiplikation

$$(k, l) \cdot (n, m) = (kn - lm, km + ln)$$

ist der Körper $\mathbb{C}$ der komplexen Zahlen.
Das Element $i := (0, 1)$ hat die Eigenschaft

$$i^2 = (0, 1) \cdot (0, 1) = (-1, 0).$$

Jedes $(x, y) \in \mathbb{C}$ hat die Eigenschaft

$$(x, y) = x \cdot (1, 0) + y \cdot (0, 1) = x + iy.$$

Eigenschaften von $\mathbb{C}$

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

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

- Betrag $|z| = |(x, y)| := \sqrt{x^2 + y^2}$ (Sage: `abs`)
- Es gilt: $z = (x, y)_{\text{Rechtwinklig}} = (r, \varphi)_{\text{Polar}} = re^{i\varphi}$
- $\mathbb{C}$ ist kein angeordneter Körper!

$\mathbb{C}$ in Sage I

- 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)
```

$\mathbb{C}$ in Sage II

- Ergebnisse werden nicht automatisch bzgl. Realteil und Imaginärteil getrennt; Dies kann erzwungen werden durch `rectform`:

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

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

- Mittels `real()` und `imag()` erhält man Real- und Imaginärteil.

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

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