Abbildungen, Funktionen

Jörn Loviscach

Versionsstand: 9. November 2009, 19:27

1 Begriffe Abbildung, Funktion

Eine Abbildung [mapping, map] namens f ordnet jedem Element x eines Definitionsbereichs [domain] D genau ein Element y eines Wertevorrats / einer Zielmenge [codomain, target set, *ungenau:* range] W zu. Diese Beziehung zwischen f und den beiden Mengen schreibt man [1] . Diese Formel sagt aber noch nichts darüber, wie die Abbildung im Detail funktioniert. Das schreibt man als Zuordnungsvorschrift, schulmäßig als Gleichung [2] oder professionell mit einem Abbildungspfeil [3] . Statt $f(x)$ wird dann typischerweise eine Formel angegeben wie bei [4] . Achtung: Streng mathematisch ist der Name der Funktion f, nicht $f(x)$. Letzteres ist ein ausgerechneter Wert $\in W$.

Wenn Definitionsbereich und Zielmenge Mengen von Zahlen statt von Punkten in der Ebene oder von geometrischen Objekten oder von Funktionen (!) oder ... sind, nennt man eine Abbildung meist eine Funktion [function]. Mathematisch sind Abbildungen und Funktionen aber dasselbe.

Definitionsbereich und Wertevorrat dürfen dieselbe Menge sein:

[5]

$f : D \to \mathbb{R}$ habe die Abbildungsvorschrift $x \mapsto \frac{x^2+7}{x-3}$. Was ist der maximal mögliche Definitionsbereich $D \subseteq \mathbb{R}$? [6]

Nicht nur sin und ln, sondern die meisten üblichen Operationen mit Zahlen sind Funktionen – allerdings oft nicht als $f(x)$ geschrieben:

- Addition reeller Zahlen:

  [7]

- Division reeller Zahlen:

  [8]

- Fakultät:

  9

- Binomialkoeffizient:

  10

2 Vorstellungen

Es gibt mehre übliche Vorstellungen von Abbildungen/Funktionen. Hier sind vier davon.

Vorstellung 1: Maschine. Eine Abbildung f ist eine Maschine, in die man ein Teil aus D als Rohmaterial hineinwerfen kann und aus der man dann ein Teil aus W als Produkt zurückerhält:

11

Alle Teile aus D müssen akzeptiert werden; nur Teile aus W dürfen produziert werden. Es wird immer genau ein Teil produziert (nicht null, nicht sieben). Wenn man dasselbe Teil aus D nochmal hinein wirft, muss wieder dasselbe Teil aus W produziert werden wie beim ersten Mal.

Vorstellung 2: Tabelle. Eine Abbildung f ist eine Tabelle (gegebenenfalls unendlich lang), aus der man man für jedes $x \in D$ das zugehörige $f(x) \in W$ ablesen kann:

12

In der x-Spalte der Tabelle muss jedes Element aus D genau einmal vorkommen (nicht nullmal, nicht siebenmal). In der $f(x)$-Spalte der Tabelle dürfen nur Elemente aus W stehen. Die Reihenfolge der Zeilen in der Tabelle ist egal.

Vorstellung 3: Pfeildiagramm:

13

Von jedem Element in D muss genau ein Pfeil starten und zu einem Element in W führen. Es ist erlaubt, dass mehrere Pfeile auf einem Element von W enden. Es ist auch erlaubt, dass Elemente von W von keinem Pfeil erreicht werden.

Vorstellung 4: Funktionskurven (offizieller Name: Funktionsgraphen):

14

Hier ist von Funktionen die Rede; D und W sind also Mengen von Zahlen. D ist ein Bereich auf der x-Achse, W ein Bereich auf der y-Achse. D und W können sich natürlich auch jeweils über die gesamten Achsen erstrecken. Zu jedem $x \in D$ (und nur für solche x) ist genau ein $y \in W$ markiert. Anmerkung: Fiese Funktionen bilden keine anschaulichen „Kurven“, sondern können zum Beispiel zu Staub zerfallen:

15

3 Was geht und was nicht

Eine Abbildung/Funktion braucht nicht unbedingt eine ausdrückliche Rechenvorschrift wie $x \mapsto \sin(x^{13})$, mit der man sie ausrechnen kann. Vielmehr kann eine Abbildung/Funktion in Worten beschrieben sein:

16

Oder sie ist nur indirekt gegeben:

17

Beim Programmieren in den üblichen Sprachen wie C kommen ebenfalls „Funktionen“ vor. Diese sind aber allgemeiner als die mathematischen Funktionen:

- Funktionen in C dürfen auch Zeichenketten, Pixel oder Fenster verarbeiten und heißen weiterhin Funktionen, nicht Abbildungen.
- Sie dürfen ohne Eingabe bleiben.
  [18]
- Sie dürfen ohne Ausgabe bleiben.
  [19]
- Sie müssen nicht bei gleicher Eingabe immer die gleiche Ausgabe liefern.
  [20]
- Das Wichtigste: Sie dürfen Nebeneffekte haben.
  [21]

In „funktionalen Programmiersprachen“ halten sich die Funktionen dagegen viel mehr an das Vorbild der Mathematik. Dies wird aktuell für Multi-Core-Rechner sehr interessant. Beispiele (Links): Erlang, F#.

4 Bildmenge

Die Menge der tatsächlich vorkommenden Elemente der Zielmenge W heißt Bild [image] oder Bildmenge $f(D)$:
[22]

Vorsicht, Namensverwirrung: Das Wort „Wertebereich“ [range] ist unklar. Es kann je nach Autor die Zielmenge = den Wertevorrat W bezeichnen oder aber die Bildmenge $f(D)$.[c1]

[c1] jl: Einige Autoren bezeichne Zielmenge = Wertevorrat W als menge. Das passiert ebenso im F schen mit dem unklaren Begriff gel].

Beispiele:

- Die Funktion $f_1 : [3,4] \to \mathbb{R}$ sei definiert durch $x \mapsto x^2$. Was ist die Bildmenge?
  [23]

  In Wolfram Alpha: `plot x^2 x=3 to 4`

- Die Funktion $f_2 : \mathbb{Z} \to R$ sei definiert durch $x \mapsto$ Anzahl der Nullen in der Dezimaldarstellung von x. Was ist die Bildmenge?

24

- Die Menge aller endlichen Zeichenketten heiße S. Die Abbildung $f_3 : S \to S$ habe die Abbildungsvorschrift $x \mapsto$ ersetze in x[c1] jeden Buchstaben `a` [c2] durch den Buchstaben `b`. Was ist die Bildmenge?

[c1] *text added by jl*

[c2] *removed text by jl:* in x

25

- Die Abbildung $f_4 : \mathbb{R} \to \mathbb{R}^2$ habe die Abbildungsvorschrift $t \mapsto (\cos(2\pi t)|\sin(2\pi t))$. Was ist die Bildmenge?

26

In Wolfram Alpha: `plot(cos(2 pi t),sin(2 pi t))`

- Die Abbildung $f_5 : \mathbb{R} \to \mathbb{R}^2$ habe die Abbildungsvorschrift $t \mapsto (e^t\cos(2\pi t)|e^t\sin(2\pi t))$. Was ist die Bildmenge?

27