Vorkurs
Formale Methoden der Informatik

Bettina Esser und Michael Kaibel

2. September bis 13. September 2024, Universität Bonn

Informatik V

Relationen

Relationen - Motivation

- Wir wollen modellieren, was es bedeutet, dass zwei (oder mehrere) Objekte miteinander in einer **Beziehung** zueinander stehen.

Relationen - Motivation

- Wir wollen modellieren, was es bedeutet, dass zwei (oder mehrere) Objekte miteinander in einer **Beziehung** zueinander stehen.
- Wir könnten weiterhin so eine Beziehung immer durch Prädikate ausdrücken. Das ist allerdings aufwendig und irgendwann unübersichtlich.

Relationen - Motivation

- Wir wollen modellieren, was es bedeutet, dass zwei (oder mehrere) Objekte miteinander in einer **Beziehung** zueinander stehen.
- Wir könnten weiterhin so eine Beziehung immer durch Prädikate ausdrücken. Das ist allerdings aufwendig und irgendwann unübersichtlich.
- **Idee: Modelliere eine Beziehung als die Menge von Tupeln mit Variablenwerten, für die die Beziehung wahr ist.**
- Bsp.: Auf $\mathbb{N}$ könnte man schreiben: „$\leq$“ $= \{(1,1),(1,2),\ldots,(2,2),(2,3),\ldots\}$.

Relationen - Definition

Definition

Eine (binäre) Relation R auf A und B ist eine allgemeine Menge von Tupeln mit Werten in A, B, d.h. $R \subseteq A \times B$.

Hierbei heißt $A = \mathrm{dom}(R)$ Definitionsbereich (Domain) von R und $B = \mathrm{codom}(R)$ Wertebereich oder Zielbereich (Codomain).

Gilt $A = B$, so nennt man R einfach nur eine binäre Relation „auf A".

Falls $(a, b) \in R$, nennt man a in Relation zu b stehend, und schreibt auch aRb.

Relationen - Definition

Definition

Eine (binäre) Relation R auf A und B ist eine allgemeine Menge von Tupeln mit Werten in A, B, d.h. $R \subseteq A \times B$.

Hierbei heißt $A = \text{dom}(R)$ Definitionsbereich (Domain) von R und $B = \text{codom}(R)$ Wertebereich oder Zielbereich (Codomain).

Gilt $A = B$, so nennt man R einfach nur eine binäre Relation „auf A".

Falls $(a, b) \in R$, nennt man a in Relation zu b stehend, und schreibt auch aRb.

Beispiel

- „ $\leq$ " $= \{(n, m) \in \mathbb{N}^2 | \exists k \in \mathbb{N}_0 : m = n + k\}$ ist eine sehr wichtige Relation.

Relationen - Definition

Definition

Eine (binäre) Relation R auf A und B ist eine allgemeine Menge von Tupeln mit Werten in A, B, d.h. $R \subseteq A \times B$.

Hierbei heißt $A = \mathrm{dom}(R)$ Definitionsbereich (Domain) von R und $B = \mathrm{codom}(R)$ Wertebereich oder Zielbereich (Codomain).

Gilt $A = B$, so nennt man R einfach nur eine binäre Relation „auf A".

Falls $(a, b) \in R$, nennt man a in Relation zu b stehend, und schreibt auch aRb.

Beispiel

- $„\leq" = \{(n, m) \in \mathbb{N}^2 | \exists k \in \mathbb{N}_0 : m = n + k\}$ ist eine sehr wichtige Relation.
- Die **Identitätsrelation** $„=" = \{(a, a) | a \in A\}$ spielt offensichtlich eine zentrale Rolle.

Relationen - Definition

Definition

Eine (binäre) Relation R auf A und B ist eine allgemeine Menge von Tupeln mit Werten in A, B, d.h. $R \subseteq A \times B$.

Hierbei heißt $A = \mathrm{dom}(R)$ Definitionsbereich (Domain) von R und $B = \mathrm{codom}(R)$ Wertebereich oder Zielbereich (Codomain).

Gilt $A = B$, so nennt man R einfach nur eine binäre Relation „auf A".

Falls $(a, b) \in R$, nennt man a in Relation zu b stehend, und schreibt auch aRb.

Beispiel

- „ $\leq$ " $= \{(n, m) \in \mathbb{N}^2 | \exists k \in \mathbb{N}_0 : m = n + k\}$ ist eine sehr wichtige Relation.
- Die **Identitätsrelation** „ $=$ " $= \{(a, a) | a \in A\}$ spielt offensichtlich eine zentrale Rolle.
- Die **leere Relation** $R = \emptyset$ und die **triviale Relation** $R = A \times B$ haben keinerlei Informationsgehalt, sollten aber nicht vergessen werden.

Eigenschaften von Relationen I

Viele Relationen, denen man begegnet, haben ein bestimmtes Muster oder verhalten sich ähnlich wie andere Relationen. Wir führen einige Begriffe ein, um viele interessante Relationen schon einmal zu charakterisieren.

Eigenschaften von Relationen I

Viele Relationen, denen man begegnet, haben ein bestimmtes Muster oder verhalten sich ähnlich wie andere Relationen. Wir führen einige Begriffe ein, um viele interessante Relationen schon einmal zu charakterisieren.

Definition

- Eine Relation R auf A heißt reflexiv, wenn für alle $a \in A$ gilt: aRa.

Eigenschaften von Relationen I

Viele Relationen, denen man begegnet, haben ein bestimmtes Muster oder verhalten sich ähnlich wie andere Relationen. Wir führen einige Begriffe ein, um viele interessante Relationen schon einmal zu charakterisieren.

Definition

- Eine Relation R auf A heißt reflexiv, wenn für alle $a \in A$ gilt: aRa.
- Eine Relation R auf A heißt symmetrisch, wenn für alle $x, y \in A$ gilt: $xRy \Rightarrow yRx$.

Eigenschaften von Relationen I

Viele Relationen, denen man begegnet, haben ein bestimmtes Muster oder verhalten sich ähnlich wie andere Relationen. Wir führen einige Begriffe ein, um viele interessante Relationen schon einmal zu charakterisieren.

Definition

- Eine Relation R auf A heißt reflexiv, wenn für alle $a \in A$ gilt: aRa.
- Eine Relation R auf A heißt symmetrisch, wenn für alle $x, y \in A$ gilt: $xRy \Rightarrow yRx$.
- Eine Relation R auf A heißt transitiv, wenn für alle $x, y, z \in A$ gilt: $(xRy \wedge yRz) \Rightarrow xRz$.

Eigenschaften von Relationen I

Viele Relationen, denen man begegnet, haben ein bestimmtes Muster oder verhalten sich ähnlich wie andere Relationen. Wir führen einige Begriffe ein, um viele interessante Relationen schon einmal zu charakterisieren.

Definition

- Eine Relation R auf A heißt reflexiv, wenn für alle $a \in A$ gilt: aRa.
- Eine Relation R auf A heißt symmetrisch, wenn für alle $x, y \in A$ gilt: $xRy \Rightarrow yRx$.
- Eine Relation R auf A heißt transitiv, wenn für alle $x, y, z \in A$ gilt: $(xRy \wedge yRz) \Rightarrow xRz$.
- Eine Relation R auf A heißt antisymmetrisch, wenn für alle $x, y \in A$ gilt: $(xRy \wedge yRx) \Rightarrow x = y$.

Eigenschaften von Relationen I

Viele Relationen, denen man begegnet, haben ein bestimmtes Muster oder verhalten sich ähnlich wie andere Relationen. Wir führen einige Begriffe ein, um viele interessante Relationen schon einmal zu charakterisieren.

Definition

- Eine Relation R auf A heißt reflexiv, wenn für alle $a \in A$ gilt: aRa.
- Eine Relation R auf A heißt symmetrisch, wenn für alle $x, y \in A$ gilt: $xRy \Rightarrow yRx$.
- Eine Relation R auf A heißt transitiv, wenn für alle $x, y, z \in A$ gilt: $(xRy \wedge yRz) \Rightarrow xRz$.
- Eine Relation R auf A heißt antisymmetrisch, wenn für alle $x, y \in A$ gilt: $(xRy \wedge yRx) \Rightarrow x = y$.
- Eine Relation R auf A heißt asymmetrisch, wenn für alle $x, y \in A$ gilt: $xRy \implies \neg(yRx)$.

Eigenschaften von Relationen II

Definition

- Eine Relation R auf A heißt total, wenn für alle $x, y \in A$ gilt: $(xRy \vee yRx)$.

Eigenschaften von Relationen II

Definition

- Eine Relation R auf A heißt total, wenn für alle $x, y \in A$ gilt: $(xRy \vee yRx)$.
- Eine reflexive, symmetrische, transitive Relation auf R heißt Äquivalenzrelation.

Eigenschaften von Relationen II

Definition

- Eine Relation R auf A heißt total, wenn für alle $x, y \in A$ gilt: $(xRy \vee yRx)$.
- Eine reflexive, symmetrische, transitive Relation auf R heißt Äquivalenzrelation.
- Eine reflexive, antisymmetrische, transitive Relation auf R heißt (partielle) Ordnung (poset).

Eigenschaften von Relationen II

Definition

- Eine Relation R auf A heißt total, wenn für alle $x, y \in A$ gilt: $(xRy \vee yRx)$.
- Eine reflexive, symmetrische, transitive Relation auf R heißt Äquivalenzrelation.
- Eine reflexive, antisymmetrische, transitive Relation auf R heißt (partielle) Ordnung (poset).
- Eine Ordnung, die außerdem noch total ist, heißt Totalordnung.

Eigenschaften von Relationen: Beispiele

Beispiel

- Auf jeder Menge ist „ $=$ “ eine Äquivalenzrelation.

Eigenschaften von Relationen: Beispiele

Beispiel

- Auf jeder Menge ist „ $=$ “ eine Äquivalenzrelation.
- Auf der leeren Menge ist die leere Relation eine Totalordnung.

Beispiel

- Auf jeder Menge ist „ $=$ “ eine Äquivalenzrelation.
- Auf der leeren Menge ist die leere Relation eine Totalordnung.
- Auf jeder Menge ist „ $=$ “ sowohl symmetrisch als auch antisymmetrisch!

Eigenschaften von Relationen: Beispiele

Beispiel

- Auf jeder Menge ist „ $=$ “ eine Äquivalenzrelation.
- Auf der leeren Menge ist die leere Relation eine Totalordnung.
- Auf jeder Menge ist „ $=$ “ sowohl symmetrisch als auch antisymmetrisch!
- Auf $P(X)$ ist $\subseteq$ eine partielle Ordnung.

Eigenschaften von Relationen: Beispiele

Beispiel

- Auf jeder Menge ist „ $=$ “ eine Äquivalenzrelation.
- Auf der leeren Menge ist die leere Relation eine Totalordnung.
- Auf jeder Menge ist „ $=$ “ sowohl symmetrisch als auch antisymmetrisch!
- Auf $P(X)$ ist $\subseteq$ eine partielle Ordnung.
- Auf der „Menge aller aussagenlogischen Ausdrücke“ ist „ $\equiv$ “ eine Äquivalenzrelation.

Eigenschaften von Relationen: Beispiele

Beispiel

- Auf jeder Menge ist „ $=$ “ eine Äquivalenzrelation.
- Auf der leeren Menge ist die leere Relation eine Totalordnung.
- Auf jeder Menge ist „ $=$ “ sowohl symmetrisch als auch antisymmetrisch!
- Auf $P(X)$ ist $\subseteq$ eine partielle Ordnung.
- Auf der „Menge aller aussagenlogischen Ausdrücke“ ist „ $\equiv$ “ eine Äquivalenzrelation.
- Die **Ungleichrelation** „ $\neq$ “ auf $\mathbb{Z}$ ist symmetrisch und erfüllt ansonsten keine der anderen Eigenschaften.
- Auf R ist „ $<$ “ **keine** Totalordnung, da sie nicht reflexiv ist. „ $\leq$ “ jedoch schon. Man nennt $<$ auch eine **strenge Totalordnung**.

Fazit zu Relationen

- xRy schreiben ist auch umständlich und kann unübersichtlich werden. Verwende anstatt R ein **allgemeines Relationszeichen** wie $\sim$, also $x \sim y$. Wenn der Kontext nicht klar ist, schreib etwas wie: $x \sim_R y$.

Fazit zu Relationen

- xRy schreiben ist auch umständlich und kann unübersichtlich werden. Verwende anstatt R ein **allgemeines Relationszeichen** wie $\sim$, also $x \sim y$. Wenn der Kontext nicht klar ist, schreib etwas wie: $x \sim_R y$.
- Partielle Ordnungen und Totalordnungen sollen im Kopf auch wirklich wie Ordnungen funktionieren - niedrigwertige Objekte liegen unter höherwertigen Objekten.

Fazit zu Relationen

- xRy schreiben ist auch umständlich und kann unübersichtlich werden. Verwende anstatt R ein **allgemeines Relationszeichen** wie $\sim$, also $x \sim y$. Wenn der Kontext nicht klar ist, schreib etwas wie: $x \sim_R y$.
- Partielle Ordnungen und Totalordnungen sollen im Kopf auch wirklich wie Ordnungen funktionieren - niedrigwertige Objekte liegen unter höherwertigen Objekten.
- Äquivalenzrelationen tauchen häufiger auf, als man denkt. Wie in der Aussagenlogik bereits geraten: Es macht Sinn, zwei unter einer Äquivalenzrelation in Beziehung zueinander stehende Objekte im richtigen Kontext als ein und das selbe Objekt zu sehen. Der Begriff der **Äquivalenzklasse** modelliert diese Idee (LudS).

Funktionen

Was für Funktionen kennt ihr so?

Funktionen in der Schule

- Eingabe und Ausgabe üblicherweise reelle Zahlen

Funktionen in der Schule

- Eingabe und Ausgabe üblicherweise reelle Zahlen
- Stetig (salopp gesagt haben ”keine Sprünge”)

Funktionen in der Schule

- Eingabe und Ausgabe üblicherweise reelle Zahlen
- Stetig (salopp gesagt haben "keine Sprünge")
- Meistens durch elementare arithmetische Operationen darstellbar
  - $+, -, \cdot, /, e^x \ldots$

Funktionen in der Schule

- Eingabe und Ausgabe üblicherweise reelle Zahlen
- Stetig (salopp gesagt haben "keine Sprünge")
- Meistens durch elementare arithmetische Operationen darstellbar
  - $+, -, \cdot, /, e^x \dots$
- **Allgemeine Funktionen in Mathe müssen keine dieser Eigenschaften haben**

Funktionen im Programmieren

- Können 0 oder mehr Eingabeparameter verschiedener Datentypen haben

Funktionen im Programmieren

- Können 0 oder mehr Eingabeparameter verschiedener Datentypen haben
- Dürfen in Verhalten von externen Zuständen (globale Variablen, Attribute eines Objekt o.Ä. abhängen)

Funktionen im Programmieren

- Können 0 oder mehr Eingabeparameter verschiedener Datentypen haben
- Dürfen in Verhalten von externen Zuständen (globale Variablen, Attribute eines Objekt o.Ä. abhängen)
- Können einen Wert zurückgeben, müssen das aber nicht

Funktionen im Programmieren

- Können 0 oder mehr Eingabeparameter verschiedener Datentypen haben
- Dürfen in Verhalten von externen Zuständen (globale Variablen, Attribute eines Objekt o.Ä. abhängen)
- Können einen Wert zurückgeben, müssen das aber nicht
- Können externe Werte verändern (globale Variablen, Objektattribute u.Ä.)

Funktionen im Programmieren

- Können 0 oder mehr Eingabeparameter verschiedener Datentypen haben
- Dürfen in Verhalten von externen Zuständen (globale Variablen, Attribute eines Objekt o.Ä. abhängen)
- Können einen Wert zurückgeben, müssen das aber nicht
- Können externe Werte verändern (globale Variablen, Objektattribute u.Ä.)
- Nur "pure Funktionen" sind Funktionen im mathematsichen Sinn

Funktionen im Programmieren

- Können 0 oder mehr Eingabeparameter verschiedener Datentypen haben
- Dürfen in Verhalten von externen Zuständen (globale Variablen, Attribute eines Objekt o.Ä. abhängen)
- Können einen Wert zurückgeben, müssen das aber nicht
- Können externe Werte verändern (globale Variablen, Objektattribute u.Ä.)
- Nur "pure Funktionen" sind Funktionen im mathematsichen Sinn
    - Geben bei gleichen Eingabeparametern immer die gleiche Ausgabe
    - Können keine externen Zustände verändern
    - (müssen min. 1 Parameter erhalten und eine Rückgabe geben)

- Mathematische Funktion hat Eingabemenge A und Ausgabemenge B

Allgemeine Mathematische Funktionen - Intuition

- Mathematische Funktion hat Eingabemenge A und Ausgabemenge B
- Alles, was jedem Element $a \in A$ der Eingabemenge eine eindeutige Ausgabe $b \in B$ zuordnet, ist eine Funktion

Allgemeine Mathematische Funktionen - Intuition

- Mathematische Funktion hat Eingabemenge A und Ausgabemenge B
- Alles, was jedem Element $a \in A$ der Eingabemenge eine eindeutige Ausgabe $b \in B$ zuordnet, ist eine Funktion
  - Es muss keinen logischen Zusammenhang zwischen Eingabe und Ausgabe geben

Allgemeine Mathematische Funktionen - Intuition

- Mathematische Funktion hat Eingabemenge A und Ausgabemenge B
- Alles, was jedem Element $a \in A$ der Eingabemenge eine eindeutige Ausgabe $b \in B$ zuordnet, ist eine Funktion
  - Es muss keinen logischen Zusammenhang zwischen Eingabe und Ausgabe geben
  - Die Ausgabe muss nicht für Menschen berechenbar sein, solange sie nachweislich eindeutig ist

Allgemeine Mathematische Funktionen - Intuition

- Mathematische Funktion hat Eingabemenge A und Ausgabemenge B
- Alles, was jedem Element $a \in A$ der Eingabemenge eine eindeutige Ausgabe $b \in B$ zuordnet, ist eine Funktion
  - Es muss keinen logischen Zusammenhang zwischen Eingabe und Ausgabe geben
  - Die Ausgabe muss nicht für Menschen berechenbar sein, solange sie nachweislich eindeutig ist
    - Z.B. ist die Halteproblem-Funktion nachweislich für beinahe alle Eingaben nicht berechenbar

$$H(P, w) = 1 \text{ falls das programm } P \text{ auf Eingabe } w \text{ terminiert, 0 sonst}$$

Funktionen Formal: Spezielle Relationen

Funktionen werden formalisiert als spezielle Relationen modelliert. Diese Modellierung ist wichtig um Mathe ordentlich aufzubauen, man nutzt sie später aber nicht sehr oft.

Funktionen Formal: Spezielle Relationen

Funktionen werden formalisiert als spezielle Relationen modelliert. Diese Modellierung ist wichtig um Mathe ordentlich aufzubauen, man nutzt sie später aber nicht sehr oft.

Definition

Eine **Funktion** $f: A \to B$ ist eine binäre Relation $f \subseteq A \times B$ mit der folgenden Eigenschaft:

- $\forall x \in A \exists! y \in B : x \sim_f y$

Funktionen Formal: Spezielle Relationen

Funktionen werden formalisiert als spezielle Relationen modelliert. Diese Modellierung ist wichtig um Mathe ordentlich aufzubauen, man nutzt sie später aber nicht sehr oft.

Definition

Eine **Funktion** $f: A \to B$ ist eine binäre Relation $f \subseteq A \times B$ mit der folgenden Eigenschaft:

- $\forall x \in A \exists! y \in B : x \sim_f y$
- Wörtlich: Jedes Objekt in A steht mit genau einem Objekt aus B in Beziehung.

Funktionen Formal: Spezielle Relationen

Funktionen werden formalisiert als spezielle Relationen modelliert. Diese Modellierung ist wichtig um Mathe ordentlich aufzubauen, man nutzt sie später aber nicht sehr oft.

Definition

Eine **Funktion** $f\colon A \to B$ ist eine binäre Relation $f \subseteq A \times B$ mit der folgenden Eigenschaft:

- $\forall x \in A \exists! y \in B : x \sim_f y$
- Wörtlich: Jedes Objekt in A steht mit genau einem Objekt aus B in Beziehung.
- Das eindeutige $y \in B$ mit $x \sim_f y$ schreibt man als $y = f(x)$ oder $x \mapsto y$.

Funktionen Formal: Spezielle Relationen

Funktionen werden formalisiert als spezielle Relationen modelliert. Diese Modellierung ist wichtig um Mathe ordentlich aufzubauen, man nutzt sie später aber nicht sehr oft.

Definition

Eine **Funktion** $f : A \to B$ ist eine binäre Relation $f \subseteq A \times B$ mit der folgenden Eigenschaft:

- $\forall x \in A \exists! y \in B : x \sim_f y$
- Wörtlich: Jedes Objekt in A steht mit genau einem Objekt aus B in Beziehung.
- Das eindeutige $y \in B$ mit $x \sim_f y$ schreibt man als $y = f(x)$ oder $x \mapsto y$.

Ein Wort der Vorsicht: Funktionen sind zwar per Modellierung Relationen, aber sie sind eigentlich etwas anderes. Relationen im allgemeinen modellieren Beziehungen, Funktionen modellieren Zuweisungen!

Funktionen: Beispiele

- $f : \mathbb{R} \to \mathbb{R}$ mit der Vorschrift $f(x) = x^2 + 5x + 8$ ist eine sogenannte Polynomfunktion.

Funktionen: Beispiele

- $f: \mathbb{R} \to \mathbb{R}$ mit der Vorschrift $f(x) = x^2 + 5x + 8$ ist eine sogenannte Polynomfunktion.
- $f: \mathbb{R} \to \mathbb{R}$ mit der Vorschrift $f(x) = \frac{1}{x}$ ist **keine** wohldefinierte Funktion, da $f(0)$ undefiniert ist. Die Vorschrift definiert allerdings eine Funktion $f: \mathbb{R} \setminus \{0\} \to \mathbb{R}$.

Funktionen: Beispiele

- $f : \mathbb{R} \to \mathbb{R}$ mit der Vorschrift $f(x) = x^2 + 5x + 8$ ist eine sogenannte Polynomfunktion.
- $f : \mathbb{R} \to \mathbb{R}$ mit der Vorschrift $f(x) = \frac{1}{x}$ ist **keine** wohldefinierte Funktion, da $f(0)$ undefiniert ist. Die Vorschrift definiert allerdings eine Funktion $f : \mathbb{R} \setminus \{0\} \to \mathbb{R}$.
- Die Vorschrift $F(f)(x) = \int\limits_0^x f(t)\mathrm{d}t$ definiert eine Funktion von der Menge aller auf $[0, 1]$ stetigen Funktionen in die Menge aller auf $[0, 1]$ stetig differenzierbaren Funktionen (Aus der Schule bekannt als: Fundamentalsatz der Integralrechnung).

Funktionen: Beispiele

- $f : \mathbb{R} \to \mathbb{R}$ mit der Vorschrift $f(x) = x^2 + 5x + 8$ ist eine sogenannte Polynomfunktion.
- $f : \mathbb{R} \to \mathbb{R}$ mit der Vorschrift $f(x) = \frac{1}{x}$ ist **keine** wohldefinierte Funktion, da $f(0)$ undefiniert ist. Die Vorschrift definiert allerdings eine Funktion $f : \mathbb{R} \setminus \{0\} \to \mathbb{R}$.
- Die Vorschrift $F(f)(x) = \int\limits_0^x f(t)\mathrm{d}t$ definiert eine Funktion von der Menge aller auf $[0, 1]$ stetigen Funktionen in die Menge aller auf $[0, 1]$ stetig differenzierbaren Funktionen (Aus der Schule bekannt als: Fundamentalsatz der Integralrechnung).
- Die Funktion $Id : A \to A$ mit $Id(x) = x$ nennen wir die Identitätsfunktion. A kann dabei eine beliebige Menge sein
- Eine Funktion darf formell nur einen Input und einen Output haben. Mehrere In- und Outputs lassen sich mittels karthesischem Produkt modellieren: $f : A_1 \times A_2 \to B_1 \times B_2$ nimmt quasi zwei Inputs unterschiedlichen Typs und gibt zwei Outputs unterschiedlichen Typs.

Etwas mehr zu mehreren Eingaben bzw. Ausgaben

- Schreibweise für mehrere Eingaben bzw. Ausgaben oft erstmal etwas unintuitiv

Etwas mehr zu mehreren Eingaben bzw. Ausgaben

- Schreibweise für mehrere Eingaben bzw. Ausgaben oft erstmal etwas unintuitiv
- $f : A \times B \times C \to D \times E$ hat formal als Eingabe und Ausgabe Tupel
  - Wir schreiben dann $f(a, b, c) = (d, e)$ bzw. $f(a, b, c) = d, e$

Etwas mehr zu mehreren Eingaben bzw. Ausgaben

- Schreibweise für mehrere Eingaben bzw. Ausgaben oft erstmal etwas unintuitiv
- $f : A \times B \times C \to D \times E$ hat formal als Eingabe und Ausgabe Tupel
  - Wir schreiben dann $f(a, b, c) = (d, e)$ bzw. $f(a, b, c) = d, e$
- $f : \mathbb{N} \times \mathbb{Q} \to \{0, 1\}$ entspricht in Java

```
bool f(int n, double q) {...}
```

Verknüpfung von Funktionen

- Funktionen können hintereinandergehängt/ineinander eingesetzt werden

Verknüpfung von Funktionen

- Funktionen können hintereinandergehängt/ineinander eingesetzt werden
- Seien $f : B \to C$ und $g : A \to B$ dann definieren wir die Funktion $h = f \circ g$ mit $h : A \to C$ und $h(x) = f(g(x))$

Verknüpfung von Funktionen

- Funktionen können hintereinandergehängt/ineinander eingesetzt werden
- Seien $f : B \to C$ und $g : A \to B$ dann definieren wir die Funktion $h = f \circ g$ mit $h : A \to C$ und $h(x) = f(g(x))$
- Setzt voraus, dass Zielbereich von g der Definitionsbereich von f ist

Definition

Sei eine Funktion $f : A \to B$ gegeben.

- f heißt surjektiv, wenn für alle $y \in B$ gilt: $\exists x \in A : f(x) = y$.

Funktionen: Besondere Eigenschaften

Definition

Sei eine Funktion $f: A \to B$ gegeben.

- f heißt surjektiv, wenn für alle $y \in B$ gilt: $\exists x \in A : f(x) = y$.
- Wörtlich: Jeder Wert im Zielbereich wird tatsächlich getroffen.

Funktionen: Besondere Eigenschaften

Definition

Sei eine Funktion $f : A \to B$ gegeben.

- f heißt surjektiv, wenn für alle $y \in B$ gilt: $\exists x \in A : f(x) = y$.
- Wörtlich: Jeder Wert im Zielbereich wird tatsächlich getroffen.
- f heißt injektiv, wenn für alle $x, x' \in A$ gilt: $f(x) = f(x') \Rightarrow x = x'$.

Funktionen: Besondere Eigenschaften

Definition

Sei eine Funktion $f : A \to B$ gegeben.

- f heißt surjektiv, wenn für alle $y \in B$ gilt: $\exists x \in A : f(x) = y$.
- Wörtlich: Jeder Wert im Zielbereich wird tatsächlich getroffen.
- f heißt injektiv, wenn für alle $x, x' \in A$ gilt: $f(x) = f(x') \Rightarrow x = x'$.
- Wörtlich: f trifft keinen Zielwert doppelt.

Funktionen: Besondere Eigenschaften

Definition

Sei eine Funktion $f: A \to B$ gegeben.

- f heißt surjektiv, wenn für alle $y \in B$ gilt: $\exists x \in A : f(x) = y$.
- Wörtlich: Jeder Wert im Zielbereich wird tatsächlich getroffen.
- f heißt injektiv, wenn für alle $x, x' \in A$ gilt: $f(x) = f(x') \Rightarrow x = x'$.
- Wörtlich: f trifft keinen Zielwert doppelt.
- f heißt bijektiv, wenn f sowohl surjektiv als auch injektiv ist.

Funktionen: Besondere Eigenschaften

Definition

Sei eine Funktion $f : A \to B$ gegeben.

- f heißt surjektiv, wenn für alle $y \in B$ gilt: $\exists x \in A : f(x) = y$.
- Wörtlich: Jeder Wert im Zielbereich wird tatsächlich getroffen.
- f heißt injektiv, wenn für alle $x, x' \in A$ gilt: $f(x) = f(x') \Rightarrow x = x'$.
- Wörtlich: f trifft keinen Zielwert doppelt.
- f heißt bijektiv, wenn f sowohl surjektiv als auch injektiv ist.
- Wörtlich: f trifft jeden Zielwert genau ein mal. (eng: one-to-one correspondence).

Funktionen: Besondere Eigenschaften

Definition

Sei eine Funktion $f : A \to B$ gegeben.

- f heißt surjektiv, wenn für alle $y \in B$ gilt: $\exists x \in A : f(x) = y$.
- Wörtlich: Jeder Wert im Zielbereich wird tatsächlich getroffen.
- f heißt injektiv, wenn für alle $x, x' \in A$ gilt: $f(x) = f(x') \Rightarrow x = x'$.
- Wörtlich: f trifft keinen Zielwert doppelt.
- f heißt bijektiv, wenn f sowohl surjektiv als auch injektiv ist.
- Wörtlich: f trifft jeden Zielwert genau ein mal. (eng: one-to-one correspondence).
- $g : B \to A$ heißt Umkehrfunktion zu f, wenn $f(g(y)) = y$ für alle $y \in B$ und $g(f(x)) = x$ für alle $x \in A$ gilt.

Funktionen: Besondere Eigenschaften

Definition

Sei eine Funktion $f : A \to B$ gegeben.

- f heißt surjektiv, wenn für alle $y \in B$ gilt: $\exists x \in A : f(x) = y$.
- Wörtlich: Jeder Wert im Zielbereich wird tatsächlich getroffen.
- f heißt injektiv, wenn für alle $x, x' \in A$ gilt: $f(x) = f(x') \Rightarrow x = x'$.
- Wörtlich: f trifft keinen Zielwert doppelt.
- f heißt bijektiv, wenn f sowohl surjektiv als auch injektiv ist.
- Wörtlich: f trifft jeden Zielwert genau ein mal. (eng: one-to-one correspondence).
- $g : B \to A$ heißt Umkehrfunktion zu f, wenn $f(g(y)) = y$ für alle $y \in B$ und $g(f(x)) = x$ für alle $x \in A$ gilt.

Funktionen: Beispiele für besondere Eigenschaften

Beispiel

- $f : \mathbb{R} \to \mathbb{R}^+$, $f(x) = e^x$ ist bijektiv, ihre Umkehrfunktion ist $g : \mathbb{R}^+ \to R$, $g(x) = \ln x$.

Funktionen: Beispiele für besondere Eigenschaften

Beispiel

- $f : \mathbb{R} \to \mathbb{R}^+$, $f(x) = e^x$ ist bijektiv, ihre Umkehrfunktion ist $g : \mathbb{R}^+ \to R$, $g(x) = \ln x$.
- Fakt: f ist genau dann bijektiv, wenn f eine Umkehrfunktion besitzt. (Beweis: LudS)

Beispiel

- $f : \mathbb{R} \to \mathbb{R}^+$, $f(x) = e^x$ ist bijektiv, ihre Umkehrfunktion ist $g : \mathbb{R}^+ \to R$, $g(x) = \ln x$.
- Fakt: f ist genau dann bijektiv, wenn f eine Umkehrfunktion besitzt. (Beweis: LudS)
- Schubfachprinzip: Wenn A, B endliche Mengen sind, und $|A| > |B|$, dann kann $f : A \to B$ nicht injektiv sein.

Beispiel

- $f : \mathbb{R} \to \mathbb{R}^+$, $f(x) = e^x$ ist bijektiv, ihre Umkehrfunktion ist $g : \mathbb{R}^+ \to R$, $g(x) = \ln x$.
- Fakt: f ist genau dann bijektiv, wenn f eine Umkehrfunktion besitzt. (Beweis: LudS)
- Schubfachprinzip: Wenn A, B endliche Mengen sind, und $|A| > |B|$, dann kann $f : A \to B$ nicht injektiv sein.
- Die Funktion $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, $f(x, y) = xy$ ist surjektiv, aber nicht injektiv.

Funktionen: Beispiele für besondere Eigenschaften

Beispiel

- $f: \mathbb{R} \to \mathbb{R}^+$, $f(x) = e^x$ ist bijektiv, ihre Umkehrfunktion ist $g: \mathbb{R}^+ \to R$, $g(x) = \ln x$.
- Fakt: f ist genau dann bijektiv, wenn f eine Umkehrfunktion besitzt. (Beweis: LudS)
- Schubfachprinzip: Wenn A, B endliche Mengen sind, und $|A| > |B|$, dann kann $f: A \to B$ nicht injektiv sein.
- Die Funktion $f: \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, $f(x, y) = xy$ ist surjektiv, aber nicht injektiv.
- Rätselstunde zum Ende: Versucht, eine Bijektion $f: \mathbb{N} \to \mathbb{Z}$ zu finden (ja, das geht!).