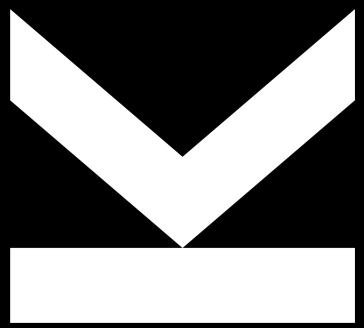
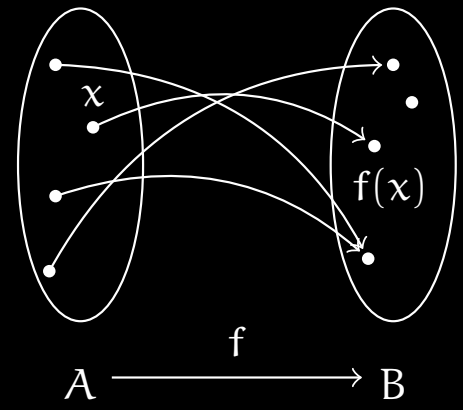
Diskrete Strukturen
Vorlesungszusammenfassung



Manuel Kauers · Institute for Algebra · JKU

Funktionen

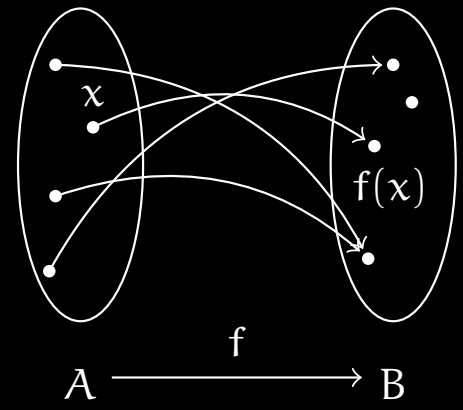
Funktionen



Was ist das?

- Informal: eine Zuordnung der Elemente einer Menge A auf Elemente einer Menge B
- Formal: $f \subseteq A \times B$ mit $\forall\, x \in A\ \exists_1\, y \in B : (x, y) \in f$

Funktionen



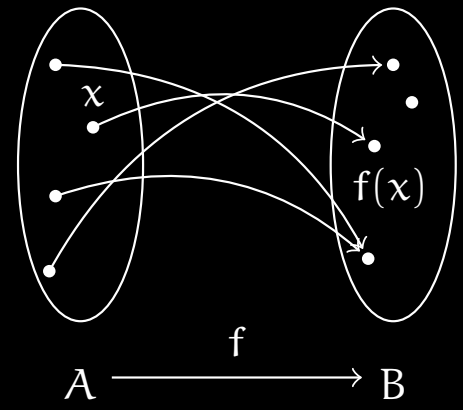
Was ist das?

- Informal: eine Zuordnung der Elemente einer Menge A auf Elemente einer Menge B
- Formal: $f \subseteq A \times B$ mit $\forall\, x \in A\ \exists_1\, y \in B : (x, y) \in f$

Wofür braucht man das?

- Funktionen drücken z.B. aus, wie man das, was man hat, überführt in das, was man will.
- Manche diskrete Strukturen lassen sich mit Hilfe von Funktionen beschreiben.
- Viele physikalische Zusammenhänge werden durch Funktionen ausgedrückt.

Funktionen



Was ist das?

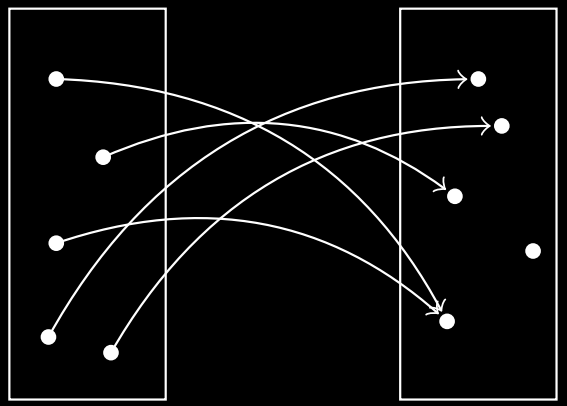
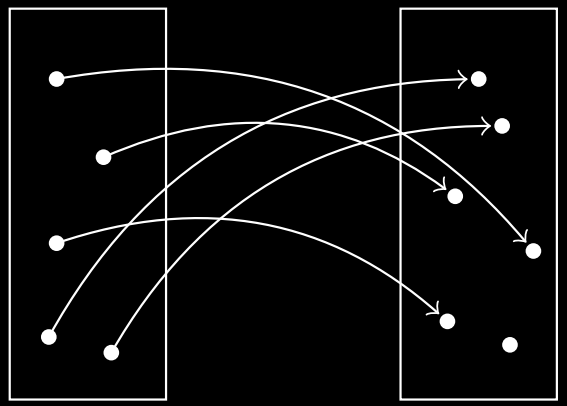
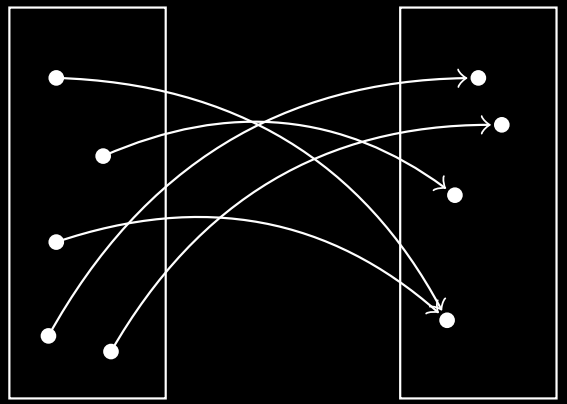
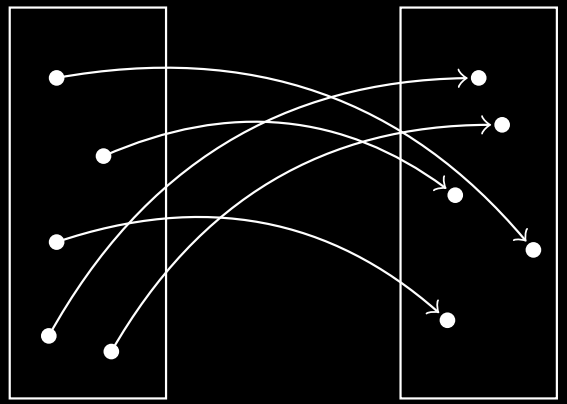
- Informal: eine Zuordnung der Elemente einer Menge A auf Elemente einer Menge B
- Formal: $f \subseteq A \times B$ mit $\forall\, x \in A\ \exists_1\, y \in B : (x, y) \in f$

Wofür braucht man das?

- Funktionen drücken z.B. aus, wie man das, was man hat, überführt in das, was man will.
- Manche diskrete Strukturen lassen sich mit Hilfe von Funktionen beschreiben.
- Viele physikalische Zusammenhänge werden durch Funktionen ausgedrückt.

Was muss man darüber wissen?

- Wie man Funktionen formal sauber definiert
- Wie man Funktionen auf Injektivität, Surjektivität, Bijektivität untersucht.

	nicht injektiv	injektiv
nicht surj.		
surjektiv		

Ordnungsrelationen

Ordnungsrelationen

Was ist das?

- Informal: eine Verallgemeinerung von $\leq$
- Formal: eine anti-symmetrische, reflexive, transitive Relation.

Ordnungsrelationen

Was ist das?

- Informal: eine Verallgemeinerung von $\leq$
- Formal: eine anti-symmetrische, reflexive, transitive Relation.

Wofür braucht man das?

- z.B. um Abhängigkeiten zwischen Softwarepaketen zu modellieren (wenn diese nicht zyklisch sein dürfen).

Ordnungsrelationen

Was ist das?

- Informal: eine Verallgemeinerung von $\leq$
- Formal: eine anti-symmetrische, reflexive, transitive Relation.

Wofür braucht man das?

- z.B. um Abhängigkeiten zwischen Softwarepaketen zu modellieren (wenn diese nicht zyklisch sein dürfen).

Was muss man darüber wissen?

- Unterschied zwischen partieller Ordnung und totaler Ordnung.
- Beispiele: $\leq$ für Zahlen, $\subseteq$ für Mengen, $\leq_{\text{lex}}$ für Wörter

Äquivalenzrelationen

Äquivalenzrelationen

Was ist das?

- Informal: eine Verallgemeinerung von $=$
- Formal: eine symmetrische, reflexive, transitive Relation.

Äquivalenzrelationen

Was ist das?

- Informal: eine Verallgemeinerung von $=$
- Formal: eine symmetrische, reflexive, transitive Relation.

Wofür braucht man das?

- Damit kann man von gegebenen mathematischen Objekten die uninteressanten Eigenschaften ausblenden.

Äquivalenzrelationen

Was ist das?

- Informal: eine Verallgemeinerung von $=$
- Formal: eine symmetrische, reflexive, transitive Relation.

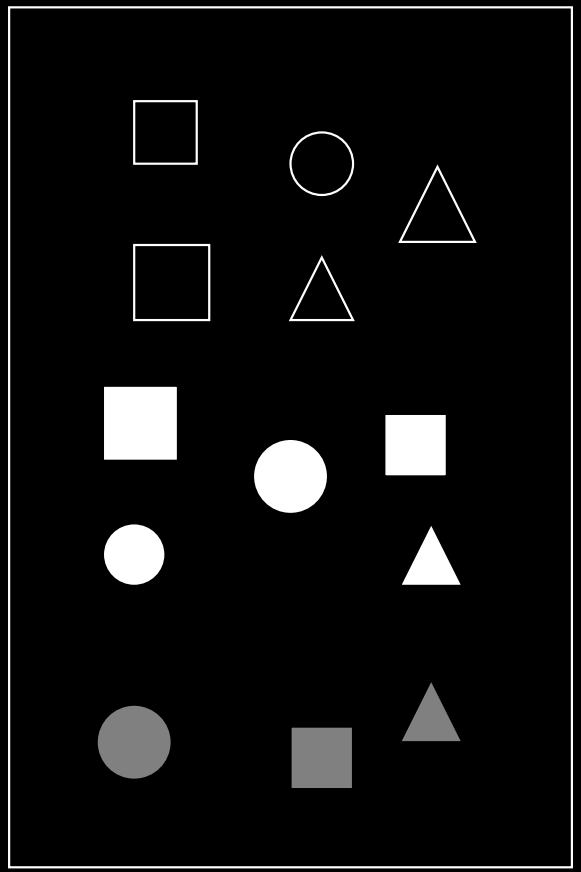
Wofür braucht man das?

- Damit kann man von gegebenen mathematischen Objekten die uninteressanten Eigenschaften ausblenden.

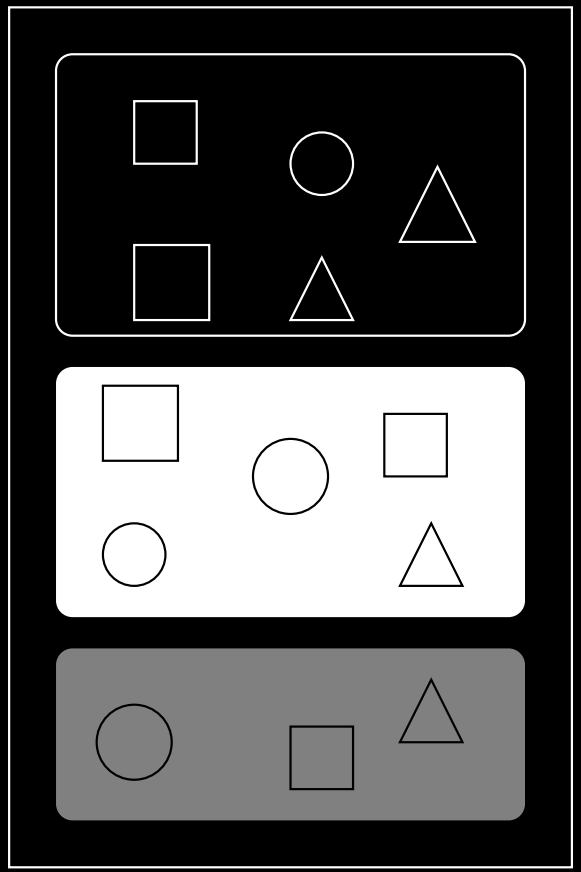
Was muss man darüber wissen?

- Wie man zeigt, dass eine gegebene Relation eine Äquivalenzrelation ist
- Wie man eine Funktionsdefinition $\mathrm{f}\colon \mathrm{A}/\!\sim \to \mathrm{B}$ auf Repräsentantenunabhängigkeit überprüft.

A

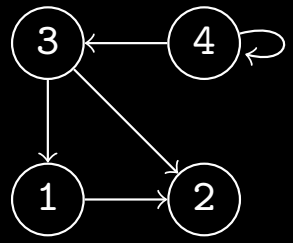


$A/\sim$



Graphen

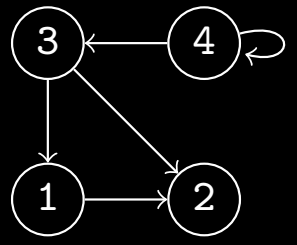
Graphen



Was ist das?

- Informal: ein Netzwerk von Knoten und Verbindungen.
- Formal: $G = (V, E)$ mit V endlich und $E \subseteq V \times V$.

Graphen



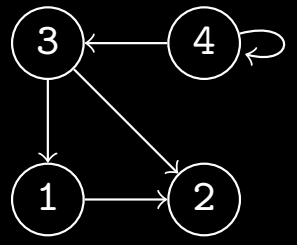
Was ist das?

- Informal: ein Netzwerk von Knoten und Verbindungen.
- Formal: $G = (V, E)$ mit V endlich und $E \subseteq V \times V$.

Wofür braucht man das?

- Graphen sind die wichtigste Datenstruktur in der Informatik. Damit kann man alle möglichen Dinge modellieren.

Graphen



Was ist das?

- Informal: ein Netzwerk von Knoten und Verbindungen.
- Formal: $G = (V, E)$ mit V endlich und $E \subseteq V \times V$.

Wofür braucht man das?

- Graphen sind die wichtigste Datenstruktur in der Informatik. Damit kann man alle möglichen Dinge modellieren.

Was muss man darüber wissen?

- Wie man Graphen im Computer codieren kann
- Wie man zwei gegebene Graphen auf Isomorphie überprüft
- Wie man in einem Graphen einen Pfad zwischen zwei gegebenen Knoten findet

Gruppen

Gruppen

Was ist das?

- Informal: eine Abstraktion des Zahlenraums $(\mathbb{Z}, +)$
- Formal: eine Menge G mit einer Verküpfung $\circ$: $\mathrm{G} \times \mathrm{G} \to \mathrm{G}$, die die Gruppenaxiome erfüllt.

Gruppen

Was ist das?

- Informal: eine Abstraktion des Zahlenraums $(\mathbb{Z}, +)$
- Formal: eine Menge G mit einer Verküpfung $\circ$: $\mathrm{G} \times \mathrm{G} \to \mathrm{G}$, die die Gruppenaxiome erfüllt.

Wofür braucht man das?

- Vor allem zur Beschreibung von Symmetrien (s.u.)
- Für modulares Rechnen (s.u.)

Gruppen

Was ist das?

- Informal: eine Abstraktion des Zahlenraums $(\mathbb{Z}, +)$
- Formal: eine Menge G mit einer Verküpfung $\circ\colon G \times G \to G$, die die Gruppenaxiome erfüllt.

Wofür braucht man das?

- Vor allem zur Beschreibung von Symmetrien (s.u.)
- Für modulares Rechnen (s.u.)

Was muss man darüber wissen?

- Die Gruppenaxiome und wann zwei Gruppen isomorph sind
- Einige Beispiele von Gruppen
- Was Untergruppen sind und wie man sie erzeugt

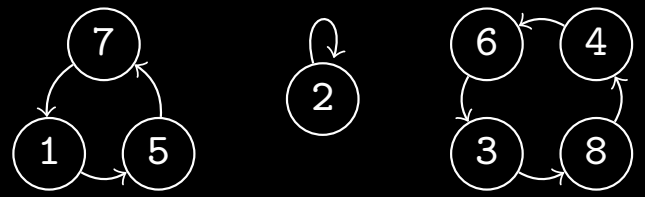
Beispiel: Die Symmetrische Gruppe

Beispiel: Die Symmetrische Gruppe

- Die Menge S_n aller bijektiven Funktionen $\pi\colon \{1,\ldots,n\} \to \{1,\ldots,n\}$ bildet mit der Verkettung $\circ$ als Verknüpfung eine (im Fall $n \geq 3$ nicht-kommutative) Gruppe.

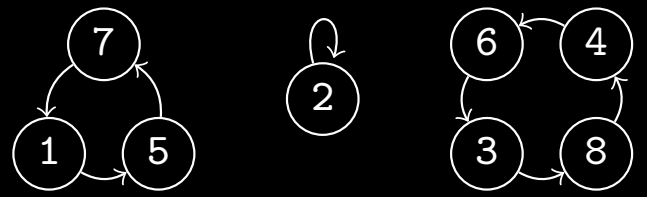
Beispiel: Die Symmetrische Gruppe

- Die Menge S_n aller bijektiven Funktionen $\pi\colon \{1,\dots,n\} \to \{1,\dots,n\}$ bildet mit der Verkettung $\circ$ als Verknüpfung eine (im Fall $n \geq 3$ nicht-kommutative) Gruppe.
- Jedes $\pi \in S_n$ lässt sich in eindeutiger Weise in paarweise disjunkte Zyklen zerlegen.



Beispiel: Die Symmetrische Gruppe

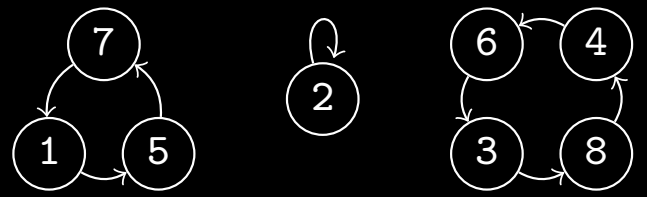
- Die Menge S_n aller bijektiven Funktionen $\pi\colon \{1,\dots,n\} \to \{1,\dots,n\}$ bildet mit der Verkettung $\circ$ als Verknüpfung eine (im Fall $n \geq 3$ nicht-kommutative) Gruppe.
- Jedes $\pi \in S_n$ lässt sich in eindeutiger Weise in paarweise disjunkte Zyklen zerlegen.



- Beispiel: $S_3 = \{\mathrm{id},\ (1\,2),\ (1\,3),\ (2\,3),\ (1\,2\,3),\ (1\,3\,2)\}$.

Beispiel: Die Symmetrische Gruppe

- Die Menge S_n aller bijektiven Funktionen $\pi\colon \{1,\dots,n\} \to \{1,\dots,n\}$ bildet mit der Verkettung $\circ$ als Verknüpfung eine (im Fall $n \geq 3$ nicht-kommutative) Gruppe.
- Jedes $\pi \in S_n$ lässt sich in eindeutiger Weise in paarweise disjunkte Zyklen zerlegen.

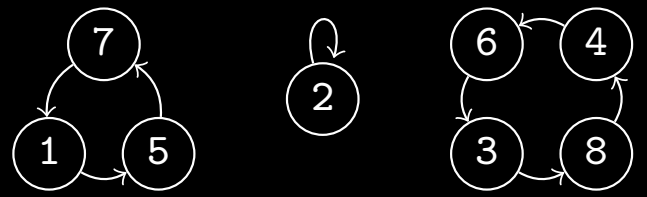


- Beispiel: $S_3 = \{\mathrm{id},\ (1\,2),\ (1\,3),\ (2\,3),\ (1\,2\,3),\ (1\,3\,2)\}$.
- Beispiel: Die von $\pi = (1\,2\,3)$ und $\sigma = (1\,2\,4)$ erzeugte Untergruppe von S_4 ist

$$\langle \pi, \sigma \rangle = \{\mathrm{id},\ \pi,\ \sigma,\ \pi^2,\ \pi\sigma,\ \sigma\pi,\ \sigma^2,\ \pi\sigma^2,\ \sigma\pi^2,\ \pi^2\sigma,\ \sigma^2\pi,\ \sigma\pi^2\sigma\}.$$

Beispiel: Die Symmetrische Gruppe

- Die Menge S_n aller bijektiven Funktionen $\pi\colon \{1,\dots,n\} \to \{1,\dots,n\}$ bildet mit der Verkettung $\circ$ als Verknüpfung eine (im Fall $n \geq 3$ nicht-kommutative) Gruppe.
- Jedes $\pi \in S_n$ lässt sich in eindeutiger Weise in paarweise disjunkte Zyklen zerlegen.

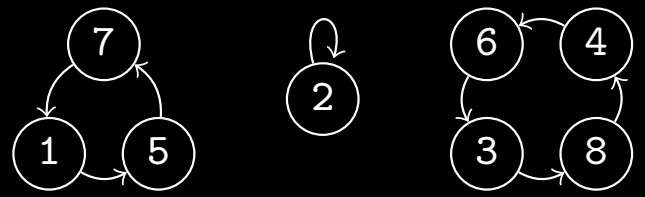


- Beispiel: $S_3 = \{\mathrm{id},\ (1\,2),\ (1\,3),\ (2\,3),\ (1\,2\,3),\ (1\,3\,2)\}$.
- Beispiel: Die von $\pi = (1\,2\,3)$ und $\sigma = (1\,2\,4)$ erzeugte Untergruppe von S_4 ist

$$\langle \pi, \sigma \rangle = \big\{\mathrm{id},\ (1\,2\,3),\ (1\,2\,4),\ (1\,3\,2),\ (1\,3)(2\,4),\ (1\,4)(2\,3),\ (1\,4\,2),\ (1\,4),\ (1\,3),\ (2\,4),\ (2\,3),\ (1\,2)\big\}.$$

Beispiel: Die Symmetrische Gruppe

- Die Menge S_n aller bijektiven Funktionen $\pi\colon \{1,\dots,n\} \to \{1,\dots,n\}$ bildet mit der Verkettung $\circ$ als Verknüpfung eine (im Fall $n \geq 3$ nicht-kommutative) Gruppe.
- Jedes $\pi \in S_n$ lässt sich in eindeutiger Weise in paarweise disjunkte Zyklen zerlegen.



- Beispiel: $S_3 = \{\mathrm{id},\ (1\,2),\ (1\,3),\ (2\,3),\ (1\,2\,3),\ (1\,3\,2)\}$.
- Beispiel: Die von $\pi = (1\,2\,3)$ und $\sigma = (1\,2\,4)$ erzeugte Untergruppe von S_4 ist

$$\langle \pi, \sigma \rangle = \big\{\mathrm{id},\ (1\,2\,3),\ (1\,2\,4),\ (1\,3\,2),\ (1\,3)(2\,4),\ (1\,4)(2\,3),\ (1\,4\,2),\ (1\,4),\ (1\,3),\ (2\,4),\ (2\,3),\ (1\,2)\big\}.$$

- Für alle $n \in \mathbb{N}$ gilt $|S_n| = n!$.

Gruppenoperationen

Gruppenoperationen

Was ist das?

- Informal: Die Elemente einer Gruppe G werden als Funktionen $X \to X$ einer Menge in sich selbst interpretiert.
- Formal: $*\colon G \times X \to X$ mit $e{*}x = x$ und $(g{\circ}h){*}x = g{*}(h{*}x)$ für alle $x \in X$, $g, h \in G$ und das Neutralelement e von G.

Gruppenoperationen

Was ist das?

- Informal: Die Elemente einer Gruppe G werden als Funktionen $X \to X$ einer Menge in sich selbst interpretiert.
- Formal: $*\colon G \times X \to X$ mit $e{*}x = x$ und $(g{\circ}h){*}x = g{*}(h{*}x)$ für alle $x \in X$, $g, h \in G$ und das Neutralelement e von G.

Wofür braucht man das?

- Mit Gruppenoperationen beschreibt man Symmetrien in der Menge X.

Gruppenoperationen

Was ist das?

- Informal: Die Elemente einer Gruppe G werden als Funktionen $X \to X$ einer Menge in sich selbst interpretiert.
- Formal: $*\colon G \times X \to X$ mit $e*x = x$ und $(g \circ h)*x = g*(h*x)$ für alle $x \in X$, $g, h \in G$ und das Neutralelement e von G.

Wofür braucht man das?

- Mit Gruppenoperationen beschreibt man Symmetrien in der Menge X.

Was muss man darüber wissen?

- Was die Bahn und der Stabilisator eines Elements $x \in X$ sind
- Dass eine Gruppenoperation eine Äquivalenzrelation auf X erklärt, deren Äquivalenzklassen die Bahnen sind.

Beispiel 1: Tupel modulo Reihenfolge

Beispiel 1: Tupel modulo Reihenfolge

- Ist M irgendeine Menge, so operiert S_4 auf $X = M^4$ via

$$\pi * (x_1, x_2, x_3, x_4) := (x_{\pi(1)}, x_{\pi(2)}, x_{\pi(3)}, x_{\pi(4)}).$$

Beispiel 1: Tupel modulo Reihenfolge

- Ist M irgendeine Menge, so operiert S_4 auf $X = M^4$ via

$$\pi * (x_1, x_2, x_3, x_4) := (x_{\pi(1)}, x_{\pi(2)}, x_{\pi(3)}, x_{\pi(4)}).$$

- Beispiel: $(1\,3\,2) * (\blacksquare, \bullet, \blacktriangle, \blacktriangledown) = (\blacktriangle, \blacksquare, \bullet, \blacktriangledown)$

Beispiel 1: Tupel modulo Reihenfolge

- Ist M irgendeine Menge, so operiert S_4 auf $X = M^4$ via

$$\pi * (x_1, x_2, x_3, x_4) := (x_{\pi(1)}, x_{\pi(2)}, x_{\pi(3)}, x_{\pi(4)}).$$

- Beispiel: $(1\,3\,2) * (\blacksquare, \bullet, \blacktriangle, \blacktriangledown) = (\blacktriangle, \blacksquare, \bullet, \blacktriangledown)$
- Der Stabilisator eines Tupels ist die Untergruppe aller Permutationen, die es auf sich selbst abbilden. Beispiel:

$$\mathrm{Stab}((\blacksquare, \blacksquare, \bullet, \bullet)) = \{\mathrm{id},\ (1\,2),\ (3\,4),\ (1\,2)(3\,4)\}$$

Beispiel 1: Tupel modulo Reihenfolge

- Ist M irgendeine Menge, so operiert S_4 auf $X = M^4$ via

$$\pi * (x_1, x_2, x_3, x_4) := (x_{\pi(1)}, x_{\pi(2)}, x_{\pi(3)}, x_{\pi(4)}).$$

- Beispiel: $(1\,3\,2) * (■, ●, ▲, ▼) = (▲, ■, ●, ▼)$
- Der Stabilisator eines Tupels ist die Untergruppe aller Permutationen, die es auf sich selbst abbilden. Beispiel:

$$\mathrm{Stab}((■, ■, ●, ●)) = \{\mathrm{id},\ (1\,2),\ (3\,4),\ (1\,2)(3\,4)\}$$

- Die Bahn eines Tupels ist die Menge aller Tupel, die sich durch Umordnen der Komponenten erzeugen lassen. Beispiel:

$$G * (■, ■, ●, ●) = \{(■, ■, ●, ●),\ (■, ●, ■, ●),\ (●, ■, ■, ●),\ (■, ●, ●, ■),\ (●, ■, ●, ■),\ (●, ●, ■, ■)\}$$

Beispiel 1: Tupel modulo Reihenfolge

- Ist M irgendeine Menge, so operiert S_4 auf $X = M^4$ via

$$\pi * (x_1, x_2, x_3, x_4) := (x_{\pi(1)}, x_{\pi(2)}, x_{\pi(3)}, x_{\pi(4)}).$$

- Beispiel: $(1\,3\,2) * (■, ●, ▲, ▼) = (▲, ■, ●, ▼)$
- Der Stabilisator eines Tupels ist die Untergruppe aller Permutationen, die es auf sich selbst abbilden. Beispiel:

$$\mathrm{Stab}((■, ■, ●, ●)) = \{\mathrm{id},\ (1\,2),\ (3\,4),\ (1\,2)(3\,4)\}$$

- Die Bahn eines Tupels ist die Menge aller Tupel, die sich durch Umordnen der Komponenten erzeugen lassen. Beispiel:

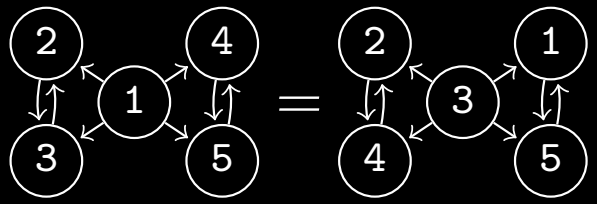
$$G * (■, ■, ●, ●) = \{(■, ■, ●, ●),\ (■, ●, ■, ●),\ (●, ■, ■, ●),\ (■, ●, ●, ■),\ (●, ■, ●, ■),\ (●, ●, ■, ■)\}$$

- Jedes Tupel gehört zu genau einer Bahn.

Beispiel 2: Die Automorphiegruppe eines Graphen

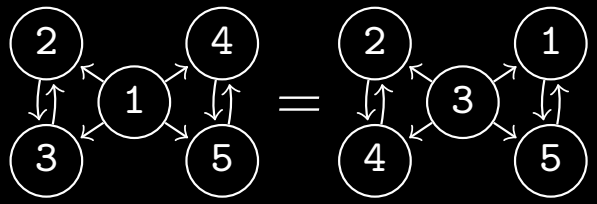
Beispiel 2: Die Automorphiegruppe eines Graphen

- S_n operiert auch auf der Menge aller Graphen $G = (V, E)$ mit $V = \{1, \ldots, n\}$. Beispiel:

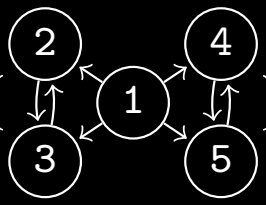
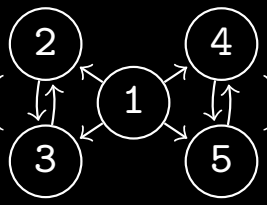
$(1\,3\,4) *$ 

Beispiel 2: Die Automorphiegruppe eines Graphen

- S_n operiert auch auf der Menge aller Graphen $G = (V, E)$ mit $V = \{1, \ldots, n\}$. Beispiel:

$(1\,3\,4) *$  $=$

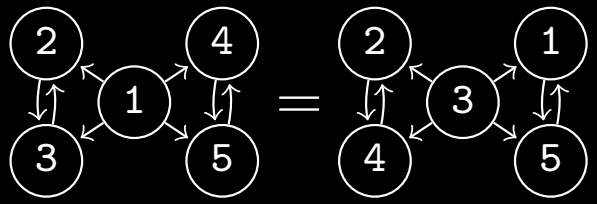
- Der Stabilisator eines Graphen ist seine Automorphiegruppe:

$\mathrm{Aut}($  $) = \mathrm{Stab}($  $) = \langle (2\,3), (4\,5), (2\,4)(3\,5) \rangle.$

Beispiel 2: Die Automorphiegruppe eines Graphen

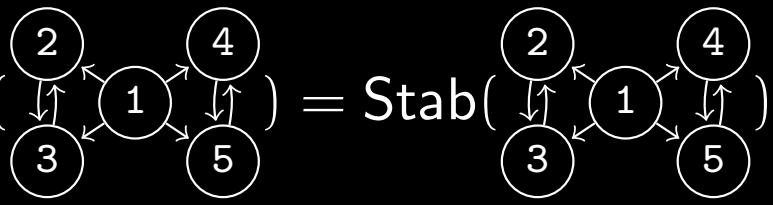
- S_n operiert auch auf der Menge aller Graphen $G = (V, E)$ mit $V = \{1, \ldots, n\}$. Beispiel:

$$(1\,3\,4) * \quad = \quad$$



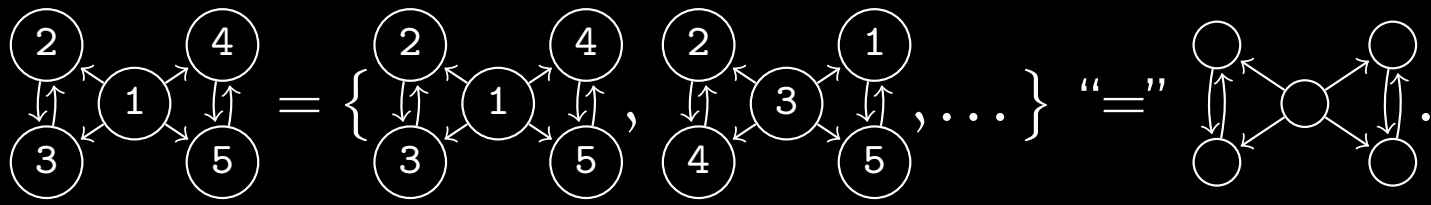
- Der Stabilisator eines Graphen ist seine Automorphiegruppe:

$$\mathrm{Aut}(\quad) = \mathrm{Stab}(\quad) = \langle (2\,3), (4\,5), (2\,4)(3\,5) \rangle.$$



- Die Bahn eines Graphen entspricht einem Graphen, bei dem man die Knotenbezeichnungen ignoriert:

$$S_n * \quad = \{\quad, \quad, \ldots\} \text{ “=” } .$$



Modulares Rechnen

Modulares Rechnen

Was ist das?

- Informal: Man rechnet in $\mathbb{Z}$ und nimmt immer, wenn die Zahlen zu lang werden, modulo m.
- Formal: $\mathbb{Z}_m = \mathbb{Z}/\equiv_m$, wobei $x \equiv_m y \iff m \mid x - y$.

Modulares Rechnen

Was ist das?

- Informal: Man rechnet in $\mathbb{Z}$ und nimmt immer, wenn die Zahlen zu lang werden, modulo m.
- Formal: $\mathbb{Z}_m = \mathbb{Z}/\equiv_m$, wobei $x \equiv_m y \iff m \mid x - y$.

Wofür braucht man das?

- Man kann damit Pseudozufallszahlengeneratoren konstruieren.
- Viele Krypto-Verfahren basieren auf Rechnungen in $\mathbb{Z}_m$.
- Statt in $\mathbb{Z}$ ist es manchmal effizienter in $\mathbb{Z}_m$ zu rechnen.

Modulares Rechnen

Was ist das?

- Informal: Man rechnet in $\mathbb{Z}$ und nimmt immer, wenn die Zahlen zu lang werden, modulo m.
- Formal: $\mathbb{Z}_m = \mathbb{Z}/\equiv_m$, wobei $x \equiv_m y \iff m \mid x - y$.

Wofür braucht man das?

- Man kann damit Pseudozufallszahlengeneratoren konstruieren.
- Viele Krypto-Verfahren basieren auf Rechnungen in $\mathbb{Z}_m$.
- Statt in $\mathbb{Z}$ ist es manchmal effizienter in $\mathbb{Z}_m$ zu rechnen.

Was muss man darüber wissen?

- Den (erweiterten) euklidischen Algorithmus und den chinesischen Restsatz.
- Dass $\mathbb{Z}_p \setminus \{0\}$ mit $\cdot$ genau dann eine Gruppe ist, wenn p eine Primzahl ist, und wie man modulare Inverse ausrechnet.

Zählfunktionen

Zählfunktionen

Was ist das?

- Den Elementen x einer Menge X wird eine Größe $w(x) \in \mathbb{N}$ zugeordnet, und zwar so, dass es für jedes $n \in \mathbb{N}$ nur endlich viele $x \in X$ mit $w(x) = n$ gibt. Dann ist $a\colon \mathbb{N} \to \mathbb{N}$, $a(n) := |\{x \in X : w(x) = n\}|$ die Zählfunktion für X.

Zählfunktionen

Was ist das?

- Den Elementen x einer Menge X wird eine Größe $w(x) \in \mathbb{N}$ zugeordnet, und zwar so, dass es für jedes $n \in \mathbb{N}$ nur endlich viele $x \in X$ mit $w(x) = n$ gibt. Dann ist $a\colon \mathbb{N} \to \mathbb{N}$, $a(n) := |\{x \in X : w(x) = n\}|$ die Zählfunktion für X.

Wofür braucht man das?

- Um abzuschätzen, wie lange eine Rechnung dauert, muss man oft verstehen, wie viele Objekte eines bestimmten Typ es gibt.

Zählfunktionen

Was ist das?

- Den Elementen x einer Menge X wird eine Größe $w(x) \in \mathbb{N}$ zugeordnet, und zwar so, dass es für jedes $n \in \mathbb{N}$ nur endlich viele $x \in X$ mit $w(x) = n$ gibt. Dann ist $a\colon \mathbb{N} \to \mathbb{N}$, $a(n) := |\{x \in X : w(x) = n\}|$ die Zählfunktion für X.

Wofür braucht man das?

- Um abzuschätzen, wie lange eine Rechnung dauert, muss man oft verstehen, wie viele Objekte eines bestimmten Typ es gibt.

Was muss man darüber wissen?

- Was Partitionen von Mengen bzw. von Zahlen sind
- Das Prinzip des kombinatorischen Beweises
- Die kombinatorische Bedeutung von $\binom{n}{k}$ und der Catalanzahlen C_n

Rekurrenzen

Rekurrenzen

Was ist das?

- Gleichungen, durch die man Probleme auf ein oder mehrere kleinere Probleme gleichen Typs zurückführen kann.

Rekurrenzen

Was ist das?

- Gleichungen, durch die man Probleme auf ein oder mehrere kleinere Probleme gleichen Typs zurückführen kann.

Wofür braucht man das?

- Um Zählfunktionen effizient berechnen zu können.
- Zur Komplexitätsanalyse von Algorithmen.

Rekurrenzen

Was ist das?

- Gleichungen, durch die man Probleme auf ein oder mehrere kleinere Probleme gleichen Typs zurückführen kann.

Wofür braucht man das?

- Um Zählfunktionen effizient berechnen zu können.
- Zur Komplexitätsanalyse von Algorithmen.

Was muss man darüber wissen?

- Welche verschiedenen Typen von Rekurrenzen es gibt
- Das Prinzip des Induktionsbeweises
- Die Pascal-Rekurrenz für $\binom{n}{k}$ und ihre wichtigsten Folgerungen
- Die O-Notation und das Master-Theorem zur Lösung von Divide-and-Conquer-Rekurrenzen