# Diskrete Strukturen

## 1: Grundlagen

---

### 1.2 – Mengen

**Was sind Diskrete Strukturen?**

Diskrete Mathematik ≠ kontinuierliche Mathematik

z.B. Speicherplatz (Bits) z.B. Analysis, Lineare Algebra

**Was sind Mengen?**

- Zusammenfassung ‚M' von bestimmten unterschiedlichen Objekten ‚m' (~Georg Cantor)
- Mengen kennen keine Reihenfolge
- Auch Mengen können Objekte von Mengen sein
- Explizite Menge (extentionale Darstellung): Jedes Element wird einzeln aufgezählt (endlich)
- Implizite Menge (intensionale Darstellung): Eigentschaften von Elementen werden aufgezählt (endlich oder unendlich)

| | |
|---|---|
| $x \in M$ | – x ist Element von M (mindestens einmal enthalten) |
| $x \notin M$ | – x ist kein Element von M |
| $\emptyset$ oder {} | – leere Menge |
| $M_1 \subseteq M_2$ | – Teilmenge: Jedes Element von $M_1$ ist auch ein Element von $M_2$ |
| $M_1 \not\subseteq M_2$ | – keine Teilmenge: mindestens ein Element aus $M_1$ ist kein Element von $M_2$ |
| $M_2 \setminus M_1$ | – Differenzmenge: Elemente in $M_2$, die nicht in $M_1$ enthalten sind |
| $M_1 = M_2$ | – gleich/identisch, wenn $M_1 \subseteq M_2$ und $M_2 \subseteq M_1$ |
| $M_1 \Delta M_2$ | – symmetrische Differenz: Menge der Objekte, die Teil von nur einer der beiden Mengen ist |

- Mächtigkeit/Kardinalität von Mengen $|M|$ Anzahl der Elemente in der Menge – $n^k$

**Mengenterme**

- Assoziativgesetze

| | |
|---|---|
| $A = A \cap A$ | $A \cap B = B \cap A$ |
| $A = A \cup A$ | $A \cup B = B \cup A$ |
| $\emptyset = A \cap \emptyset$ | $A \cap (B \cap C) = (A \cap B) \cap C$ |
| $A = A \cup \emptyset$ | $A \cup (B \cup C) = (A \cup B) \cup C$ |

- Distributivgesetze

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$
$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$
$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$
$$A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C)$$
$$A \setminus (B \cap C) = (A \setminus B) \cup (A \setminus C)$$
$$A = A \cup (A \cap B)$$
$$A = A \cap (A \cup B)$$
$$A \Delta B = (A \setminus B) \cup (B \setminus A)$$

- Mit definiertem Universum $\Omega$

$$A \cap \overline{A} = \emptyset \qquad \overline{\overline{A}} = A$$
$$A \cup \overline{A} = \Omega \qquad \overline{A \cup B} = \overline{A} \cap \overline{B}$$
$$A \setminus B = A \cap \overline{B} \qquad \overline{A \cap B} = \overline{A} \cup \overline{B}$$

Was sind Tupel?

- Zusammenfassung einer festen endlichen Anzahl von Objekten unter der Beachtung der Reihenfolge und Vielfachheit. Syntax: $t = (a, b, \{a, b\})$
- Auch Mengen können Objekte von Tupeln sein
- Länge $|t| = \#t$ Anzahl der Objekte (einschließlich Vielfachheit)
- Zwei Tupel sind identisch, wenn sie diesselbe Länge haben und die Objekte an jeder Position identisch sind
- Kartesisches Produkt von Tupeln: $A \times B = \{(a, b) | a \in A, b \in B\}$
- Länge des Kartesischen Produktes: $|A \times B| = |A| * |B|$

## 1.3 – Relationen

### Was sind Relationen?

- Relationen sind Teilmengen des kartesischen Produkts einer/meherer Menge(n)
- Man findet Tabellen in Form von Relationen in Datenbanken

$\pi_{i_1, i_2, \ldots, i_k}(R) = \{(r_{i_1}, r_{i_2}, \ldots, {}_{i_j}) \mid (r_1, \ldots, r_k) \in R\}$ – Projektion: jedes Tupel in der Menge wird auf die Einträge an den Positionen $i_j$ reduziert

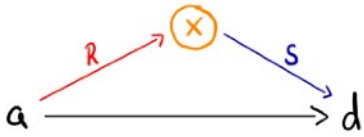
$R_1 \bowtie_P R_2 = \sigma_P(R_1 \times R_2) \subseteq R_1 \times R_2$ – Join: konkatenieren (verketten) von Tupeln under der Bedingung P

### Relationales Produkt / Verkettung

$R \subseteq A \times B$ und $S \subseteq C \times D$

$RS \subseteq A \times B$

$RS = \{(a, d) \mid$ es gibt $x \in B \cap C$ mit $(a, x) \in R$ und $(x, d) \in S\}$




### Binäre Relationen

- Binäre Relationen sind 2-stellige Relationen, die z.B. mittles relationalem Produkt mit sich selbst mehrfach verknüpft sind $R \subseteq A \times A$
- Veranschaulichung mittels Digraphen (directed graph)

$R^+ := \bigcup_{k \in \mathbb{N}} R^k$ – transitive Hülle

$R^* := \bigcup_{k \in \mathbb{N}_0} R^k$ – reflexiv-transitive Hülle

$R^{\leq k} := \bigcup_{i=0}^{k} R^i$ – „erreichbar in maximal k Schritten“

- Es gilt: $A$ endlich und $n = |A|$, dann $R^* = R^{\leq n-1}$

### Eigenschaften von binären Relationen

| | |
|---|---|
| $\forall a \in A : (a, a) \in R$ | reflexiv |
| $\forall a, b \in A : (a, b) \in R \Longrightarrow (b, a) \in R$ | symmetrisch |
| $\forall a, b \in A : (a, b) \in R \Longrightarrow (b, a) \notin R$ | asymmetrisch |
| $\forall a, b \in A : ((a, b) \in R$ und $(b, a) \in R) \Longrightarrow a = b$ | antisymmetrisch |
| $\forall a, b, c \in A : ((a, b) \in R$ und $(b, c) \in R) \Longrightarrow (a, c) \in R$ | transitiv |

### Klassifikation von Relationen auf einer Menge

- Partielle Ordnungen: reflexiv, antisymmetrisch, transitiv
- Totale Ordnungen: reflexiv, antisymmetrisch, transitiv & $\forall a, b \in R : aRb$ oder $bRa$
- Äquivalenzrelationen: relexiv, symmetrisch, transitiv

**Äquivalenzrelationen:**

- reflexiv, symmetrisch, transitiv
- Für $R \subseteq A \times B$:

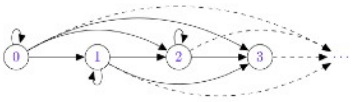
  $[a]_R = \{b \in A \mid aRb\}$ – Äquivalenzklasse eines Objekts a bzgl. R, also alles, was mit a in Relation steht

  $A/R = \{[a]_R \mid a \in A\}$ – Quotient von A bzgl. R

**Hasse-Diagramm:**

- Für $S^* = R : G_S$ ist Hasse Diagramm von $R$
- Hasse-Diagramm ist der normale Graph ohne reflexive und transitive Digraphen

Statt




nur


**Ordnungsrelationen:**

- $m \in A$ ist ein maximales Element bzgl. $R$, wenn: $mRA$ für $a \in A$, dann auch $aRm \Longrightarrow a = m$
  - Keine Kante zu anderem Element
- $m \in A$ ist das größte Element bzgl. $R$, wenn; $\forall a \in A : aRm$
  - Keine Kante zu anderem Element, aber von jedem anderen Element eine Kante
- Entsprechend definiert sind minimales und kleinstes Element
- Ist $R$ eine Ordnung, so ist auch $R^{-1}$ eine Ordnung (Pfeile umdrehen)

## 1.4 Funktionen

### Was ist eine Funktion?

$R \subseteq A \times A$ ist eine Funktion, wenn $\forall a \in A$ gibt es genau ein $b \in B : (a, b) \in R$

$f : A \to B = f \subseteq A \times B$ – Funktionen sind Teilmengen der binären Relation "alle Urbilder"$R$"alle Bilder"

$f : \mathbb{R} \to \mathbb{R} : x \mapsto x^2 + 1$ – Definition einer Funktion $f$ mittels $\mapsto$

$f : \times_{i=1}^{k} A_i \mapsto B$ – k-äre Funktion

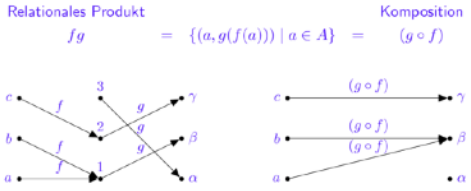
### Eigenschaften von Funktionen

$f(a) = f(a') \implies a = a'$ – injektiv, jedes $f(a)$ hat höchstens ein $a$

$\forall b \in B$ ein $a \in A \implies f(a) = b$ – surjektiv, jedes $f(a)$ hat mindestens ein $a$

$\forall b \in B : |f^{-1}(\{b\})| = 1$ – bijektiv (injektiv & surjejtiv), jedes $f(a)$ hat gen. ein $a$

### Kompositionen von Funktionen




- Kompositionen = Nacheinanderausführungen
  $f : A \to B$ und $g : B \to C$
  $(g \odot f)(a) := g(f(a))$
- Kompositionen sind assoziativ (Klammern können getauscht werden)
- Kompositionen sind *nicht* kommukativ (Reihenfolge nicht veränderbar)

## 1.5 Kardinalität: Vergleiche von Mengen

### Kardinalität von Mengen

$|A| \leq |B|$ („B mindestens so mächtig wie A") – es gibt injektive Funktion $f : A \to B$

$|A| = |B|$ („A und B sind gleichmächtig") – es gibt bijektive Funktion $f : A \to B$

$|A| < |B|$ („B ist echt mächtiger als A") – es gibt injektive Funktion $f : A \to B$ aber keine injekt. Funktion $g : B \to A$

### Satz von Cantor-Bernstein-Schröder

Sind $f : A \to B$ und $g : A \to B$ injektiv, dann gibt es $h : A \to B$ bijektiv

Abzählbarkeit von Mengen

- A ist abzählbar, wenn $|A| \leq |\mathbb{N}|$
- A ist abzählbar, wenn $\forall a \in A : (a, \in \mathbb{N})$ (jedes Element aus A einer eindeutigen natürlichen Zahl zuordnungbar ist
- Nicht abzählbar = überabzählbar

# 2: Graphen

## 2.1 Einführung Graphen

### Was ist ein Graph?

Graphen bestehen aus Knoten (V) und Kanten (E)

$G = (V, E) \quad E \subseteq V \times V$

## 2.2 Digraphen, ungerichtete und einfache Graphen

### Was sind Digraphen?

- Digraph > engl. directed graph > dt. gerichteter Graph
  $G = (V, E) \quad E \subseteq V \times V$
- Ein Diagraph ist endlich, falls $V$ endlich
- Ein Digrapgh ist bipartit, falls $V = A \cup B$ mit $A \cap B = \emptyset$ und $E \subseteq A \times B \cup B \times A$

### Pfade

- Folge von Knoten $v_0, v_1, v_2, \ldots, v_l$ von Knoten $v_i \in V$, wenn $i \in [l] : (v_{i-1}, v_i) \in E$
- Länge eines Pfades ist die Anzahl der Kanten von $v_0$ bis $v_l$ also $|v_0| = 0$
- Einfacher Pfad: kein Knoten wird mehrmals besucht
- In einem endlichen Digraphen hat ein einfacher Pfad maximale Länge $|V| - 1 = l_{max}$

### Zusammenhang

Sei $G = (V, E)$ ein Digraph.

| | |
|---|---|
| $uRv : R := (E \cup E^{-1})^* : u, v \in V$ | – G ist zusammenhängend (es gibt Graphen zwischen allen zusammenhängenden Knoten) |
| $uRv : R := E^* : u, v \in V$ | – G ist stark zusammenhängend (es gibt Digraphen zwischen allen stark zusammenhängenden Knoten) |
| $U \subseteq V$ | – U ist eine (starke) Zusammenhangskomponente, wenn G[U] (stark) zusammenhängend sind. |
| $U \not\subset U' \subset V$ und $G[U']$ | – Zusammenhangskomponente sind maximal, wenn sie nicht in einer anderen (starken) Zusammenhangskomponente sind. |

### Kreise

- Einen Pfad $v_0, v_1, \ldots, v_l \in V : v_0 = v_l$ nennt man Kreis/Zyklus
- Ein Kreis ist einfach, wenn $|\{v_0, \ldots, v_l\}| = l$, also kein Kreis teil des Kreises ist
- DAG (engl. directed acyclic graph, dt. gerichtet. antyzyklischer Graph) ist ein Graph ohne Kreis

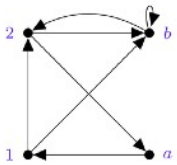
$uEv$ – u ist Vorgänger von v, v ist Nachfolger von u

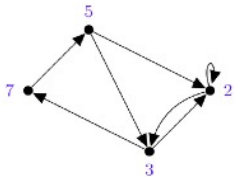
### Teilgraphen

| | |
|---|---|
| $G[U] = U \subseteq V$ für $(U, E \cap (U \times U))$ | – von U induzierter Teilgraph: alle Kanten zwischen den mitgenommenen Knoten müssen enthalten sein. |
| $G = (V_G, E_G) \;\; H = (V_H, E_H) : V_H \subseteq V_G \;\; E_H \subseteq E_G$ | – Teilgraph, bei dem nicht zwingend alle zusammengehörenden Knoten und Kanten enthalten sind. |

### Isomorphie

$G \cong H$ (isomorph) falls es eine Bijektion (Knotenumbenennung) $s : V_G \to V_H$ gibt, die die Kanten respektiert/erhält d.h. $u E_G v \iff s(u) E_H s(v)$




### Ungerichtete und einfach Graphen

- Ein Digraph $G = (V, E)$ ist ungerichtet, falls $E$ symmetrisch ist
- Ein Digraph ist einfach, wenn er ungerichtet (symmetrisch) und ohne Schleifen (irreflexiv) ist
- Statt Vorgänger und Nachfolger gibt es nur Nachbarschaften

### Kreise und Zyklen

- Def. des Kreises verändert: triviale Kreise (u, v, u) ausgeschlossen

  Pfad $u_0, u_1, \ldots, u_l$ mit $u_0 = u_l$ ist ein Kreis, falls $|\{u_i, u_{i+1 \pmod l}, u_{i+2 \pmod l}\}| = 3$ für $i \in [l]$

  > drei aufeinanderfolgende Knoten müssen unterschiedlich sein

### Weitere Graphklassen

- Multigraph: Kantenrelation ist Multimenge von Knotenpaaren – es könnte theoretisch 3 Kanten zwischen zwei Punkten geben
- Hypergraph: Kantenrelation beinhaltet Hyperkanten – eine Kante verbindet mehr als 2 Knoten gleichzeitig

### Wichtige Klassen von einfachen Graphen

- Vollständiger Graph: $K_n := ([n], \binom{[n]}{2})$, $|\text{Kante}_n| = \frac{n * (n-1)}{2}$
- Kreisgraph: $C_n := ([n], \{\{i, (i \bmod n) + 1\} \mid i \in [n]\})$, $|\text{Kante}_n| = n - 1$
- Pfadgraph: $P_n := ([n], \{\{i, i+1\} \mid i \in [n]\})$, $|\text{Kante}_n| = n - 2$
- Bipartitier Graph: $K_{m,n} := ([m+n], \{\{i, j\} \mid i \in [m], j \in [m+n] \backslash [m]\})$, $|\text{Kante}_n| = m * n$
  für $m \leq n$, m: Knoten links, n: Knoten rechts
- Gittergraph: $M_{m,n} := ([m] \times [n], \{\{(i,j),(k,l)\} \mid |i-k| + |j-l| = 1\})$
  für $m \leq n$, m: Zeilen, n: Spalten
- Hyperwürfel: $Q_n := (\{0,1\}^n, \{\{u, v\} \mid \sum_{i=1}^{n} |u_i - v_i| = 1\})$, mit $Q_0 := (\{\varepsilon\}, \varnothing)$, mit $n$ Dimensionen

### Perfekte Binärbäume

$B_h := ([2]^{\leq h}, \{\{u, ux\} \mid u \in [2]^{\leq h}, x \in [2]\})$, $h \in \mathbb{N}$

| | |
|---|---|
| $\lvert V \rvert \in B_h = 2^{h+1} - 1$ | Menge aller Knoten |
| $u \in V \mid \deg(u) \leq 1$ | Knoten $u$ werden Blätter genannt |
| $\lvert u \in V \mid \deg(u) \leq 1 \rvert = 2^h$ | Anzahl der Blätter |

### Gradfolge

- Jedem (einfachen) Graphen $G = (V, E)$ können wir eine Gradfolge zuordnen

  $(\deg(v_1), \deg(v_2), \ldots, \deg(v_n)$ für $V = \{v_1, v_2, \ldots, v_i\}$ Gradfolge

  $2|E| = \sum_{i \in [n]} deg(v_i)$ Hand-shaking Theorem für jeden einfachen Graphen

- Die Summe aller Grade eines realisierbaren Graphen ist eine gerade Zahl

### Algorithmus von Havel-Hakimi

- Eingabe: aufsteigend sortierte Gradfolge $(d_1, d_2, \ldots, d_n) : d_1 \geq 0$
- Abbruchbedingung (nicht erfüllbar) $d_1 < 0$
- Abbruchbedingung (erfüllbar) $d_n = 0$
- Sonst setze $(d'_1, \ldots, d'_{n-1}) := (d_1, \ldots, d_{n-d_n-1}, d_{n-d_n}, \ldots d_{n-1} - 1)$ und sortiere wieder aufsteigend

---

## 2.3 Bäume

### Was ist ein Baum?

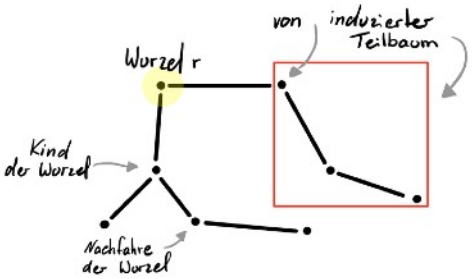
- Einfache Graphen, die zusammenhängend und kreisfrei sind, nennt man Bäume
- Blätter: $u \in V$ mit $deg(u) = 1$
- Graphen, deren maximale Zusammenhangskomponenten Bäume sind, nennt man Wald

### Eigenschaften von Bäumen

- jeder Baum mit $|V| \geq 2$ hat (min.) zwei Blätter
- $\forall$ Teilgraph $T = (V, E') : E' \subseteq E :$ T ist ein Baum nennt man Spannbaum von G
  $|E| = |V| - 1$

### Was ist ein Wurzelbaum?

- Ein Wurzelbaum $G = (V, E, r)$ ist ein Baum $G = (V, E)$ mit fest gewählter Wurzel $r \in V$
- Höhe $h_G(v)$ eines Knotens $v \in G$ ist die Länge des kürzesten Pfades zu r
  $h(G) = \min(h_g(v) \mid v \in V)$




---

## 2.4 Euler- und Hamiltonkreise

### Definition Kreis und Pfad

- Kreis: Anfang und Ende identisch
- Pfad: Anfang und Ende unterschiedlich

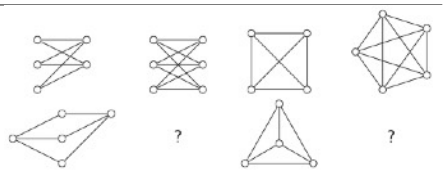
### Was ist ein Eulerkreis?

- Ein Kreis bei dem jeder Knoten mindestens einmal und jede Kante genau einmal besucht wird
- Ein einfacher zusammenhängender Graph $G = (V, E)$ mit $v \in V : deg(v) = \frac{\mathbb{N}}{2}$
  $\{v_0, v_1, \ldots, v_l\} = V$ und $\{\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_l, v_0\}\}_M = E$ Eulerkreis

### Was ist ein Hamiltonkreis?

- Ein Pfad bei dem jeder Knoten genau einmal besucht wird
- Hinreichende Bedingung: $G = (V, E) : |V| \geq 3$ mit $v \in V : deg(v) = \frac{|V|}{2}$
- Kein Algorithmus: NP-Vollständiges Problem
  $\{v_0, v_1, \ldots, v_{l-1}\}_M = V$ Hamiltonkreis

## 2.5 Planare Graphen und Knotenfärbungen




### Was ist Planarität?

- Ein einfacher Graph $G = (V, E)$ ist planar, falls man ihn ohne Kantenüberschneidung zeichnen kann
- Graphen, die $K_5$ oder $K_{3,3}$ als Minoren haben, sind nicht planar

$f = |E| - |V| + 2$ Eulersche Polyederformel
Anzahl der Flächen, in die ein Graph bei überschneidungsfreier Darstellung zerschneidet (umschließende Fläche wird mitgezählt)

- Hinreichende Bedingungen: Für jeden Planaren Graphen $G = (V, E)$ gilt
  1. $f - |E| + |V| \geq 2$
  2. $|V| \geq 3 \Longrightarrow |E| \leq 3|V| - 6$
  3. $|V| \geq 3 \Longrightarrow u \in V : deg(u) \leq 5$

### Was ist ein Minor?

- $H = (V_H, E_H)$ ist ein Minor von $G = (V_G, E_G)$ falls man aus G durch
  1. Entfernen von Kanten
  2. Entfernen von Knoten : $deg(v) = 0$
  3. Kantenkontraktion

  einen zu H isomorphen Graphen erzeugen kann

### Was sind Knotenfärbungen?

- Eine Abbildung $c : V \to \mathbb{N}$ ist eine Knotenfärbung von einem einfachen Graphen $G = (V, E)$ falls $\{u, w\} \in E : c(u) \neq c(w)$
- Die chromatische Zahl $\chi(G)$ ist die minimale Anzahl von Farben für eine Knotenfärbung von G

$\chi(G) := \min\{\, |c(V)| \mid c : V \to \mathbb{N}\}$ Trivial $G = (V, E) : E \geq 1 \Longrightarrow 2 \leq \chi(G) \leq |V|$

$$\chi(G) \leq \frac{1}{2} + \sqrt{2|E| + \frac{1}{4}}$$

- Fünf-Farben-Satz: Jeder einfache zusammenhängende planare Graph kann mit 5 Farben gefärbt werden

## 2.6 Matchings

matchings…

# 3: Logik

## 3.1 Einführung

**Logik und Inferenzen**

- Logik ist die Wissenschaft des Schließens (Schlüsse ziehen)
- Logik untersucht Inferenzen auf Korrektheit

| | |
|---|---|
| $A \to B$ | Inferenz |
| $A$ | Annahme (Prämisse, Antezedens, Hypothese) |
| $B$ | Konklusion (Konsequenz) |

| (allgemein) Gültig, Tautologie | Erfüllbar | Widerspruch |
|---|---|---|
| In allen Welten erfüllbar | In mindestens einer Welt wahr | In keiner Welt erfüllbar |

## 3.2 Aussagenlogik

**Was sind Syntax und Semantik?**

- Syntax legt fest, welche Zeichenketten wohlgeformte Ausdrücke sind
- Formeln sind wohlgeformte Ausdrücke
- Semantik legt die Bedeutung von Formeln fest

**Syntax: Vokabular**

- Wahrheitskonstanten: true, fals
- Menge $V$ : $|V| = \infty$ von Aussagenvariablen: $p, q, r, s, t, \ldots$
- Logische Operatoren: $\wedge$ , $\vee$ , $\neg$, $\to$
- Hilfssymbole: $(,)$

**Syntax: Formationsregeln**

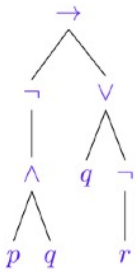
1. true und false sind Formeln.
2. Eine Aussagenvariable ist eine Formel.
3. Ist $F$ eine Formel, so ist auch $\neg F$ eine Formel.
4. Sind $F$ und $G$ Formeln, so sind $(F \wedge G), (F \vee G)$ und $(F \to G)$ ebenfalls Formeln.
5. Ein Ausdruck ist dann eine Formeln, wenn er durch Regeln 1 – 4 konstruiert werden kann.

**Syntax: Syntaxbaum**

- Die Konstruktion einer Formel stellt man als Syntaxbaum dar

  $(\neg(p \wedge q)) \to (q \vee \neg r)$ Bsp. Formel mit Syntaxbaum rechts

- Teilformeln sind aussagenlogische Formeln, die vollständig und zusammenhängend in einem Syntaxbaum vorkommen




**Syntax: Bindungsregeln**

- $\neg$ bindet stärker als $\wedge$
- $\wedge$ bindet stärker als $\vee$
- $\vee$ bindet stärker als $\to$

### Semantik

- Die Formel $F = (p \wedge q)$ kann in vier Welten ausgewertet werden
  1. p ist wahr, q ist war – F ist wahr
  2. p ist wahr, q ist falsch – F ist falsch
  3. p ist falsch, q ist wahr – F ist falsch
  4. p ist falsch, q ist falsch – F ist falsch
- Bedeutung einer Formel ist eine Funktion, die jede Welt dem Wahrheitswert der Formel in dieser Welt zuordnet

### Belegungen

- Eine Belegung ist eine Funktion, die jeder Aussagenvariable einen Wert zuordnet

  $\beta : V' \to \{0,1\}$ mit $V' \subseteq V$ Bsp. Belegung

  $\beta : V' \to \{0,1\} : V' = V_F$ minimal passende Belegung

### Weitere Operanden

- Ausschließendes-Oder: $\oplus$, XOR
- Bikonditional: $\leftrightarrow$
- If-Then-Else: ITE

### Logische Äquivalenz

- Zwei Formeln $F$ und $G$ sind logisch äquivalent ($F \equiv G$), wenn sich jede zu $F$ und $G$ passende Belegung gleich auswertet

  $\beta : V_{F,G} \to \{0,1\} : [F](\beta) = [G](\beta) \Longrightarrow F \equiv G$
- Äquivalenztest: $F \leftrightarrow G$ muss allgemein gültig sein (Tautologie)

### Äquivalenzumformungen

$(F \wedge F) \equiv F$ $(F \vee F) \equiv F$

$(F \wedge G) \equiv (G \wedge F)$ $(F \vee G) \equiv (G \vee F)$

$((F \wedge G) \wedge H) \equiv (G \wedge (F \wedge H))$ $((F \vee G) \vee H) \equiv (G \vee (F \vee H))$

$(F \wedge (F \vee G)) \equiv F$ $(F \vee (F \wedge G)) \equiv F$

$(F \wedge (G \vee H)) \equiv ((F \vee G) \wedge (F \vee H))$ $(F \vee (G \wedge H)) \equiv ((F \wedge G) \vee (F \wedge H))$

$\neg\neg F \equiv F$

$\neg(F \wedge G) \equiv (\neg F \vee \neg G)$ $\neg(F \vee G) \equiv (\neg F \wedge \neg G)$

$(F \wedge \neg F) \equiv \text{false}$ $(F \vee \neg F) \equiv \text{true}$

$(F \wedge \text{true}) \equiv F$ $(F \vee \text{false}) \equiv F$

$(F \to G) \equiv (\neg F \vee G)$

$(F \leftrightarrow G) \equiv ((F \to G) \wedge (G \to F)) \equiv \neg(F \oplus G)$

$(F \oplus G) \equiv ((F \vee G) \wedge (\neg F \vee \neg G)) \equiv ((F \wedge \neg G) \vee (\neg F \wedge G))$

### Normalformen

$\bigvee_{i=1}^{m} (\bigwedge_{j=1}^{m_i} L_{i,j})$ wobei $L_{i,j} \in \{p, \neg p \mid p \in V_F\}$ disjunktive Normalform (DNF)

$\bigwedge_{i=1}^{m} (\bigvee_{j=1}^{m_i} L_{i,j})$ wobei $L_{i,j} \in \{p, \neg p \mid p \in V_F\}$ konjunktive Normalform (KNF)

- Aufstellen mit Wahrheitstabelle, KV-Diagramm, Äquivalenzumformung

### Basen

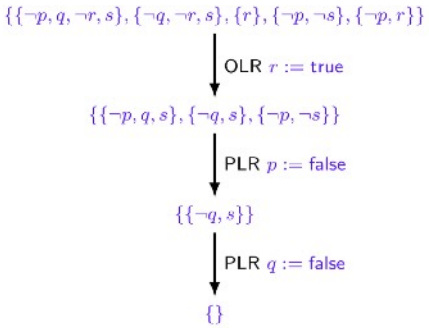
- Operatoren, mit denen man alle anderen Operatoren beschreiben kann, nennt man vollständig/vollständige Operatorbasis

### DPLL

- Algorithmus zum Nachweis der Erfüllbarkeit einer KNF
- Davis-Putnam-Logemann-Loveland
- Klauseln sind Disjunktionen von Literalen (oder, ∨)

  $\{p, \neg q, r\} \rightarrow (p \vee \neg q \vee r)$ $\{\}$ = false leere Klausel = false

  $\{\{\neg p\}, \{p, \neg q, r\}\} \rightarrow \neg p \wedge (p \vee \neg q \vee r)$ leere Klauselmenge = true
- OLR (one-literal rule): Klausel mit nur einem Literal vorhanden > mit der Variable weitermachen und sie auf den Wahrheitswert der Klausel setzen
- PLR (pure-liter rule): Ein Literal kommt in allen Klauseln nur normal oder nur als Komplement vor > mit der Variable weitermachen und sie auf den Wert des Literals setzen
- Fallunterscheidung: lexikographische Ordnung > jede Variable auf true und false setzen >
  - Alle Klauseln, die das Literal in der gesetzten Form enthalten entfernen
  - In allen Klausel, die das Komplement der Gesetzen Form enthalten, das Literal streichen
- Terminieren, wenn erfüllbare Belegung gefunden ist
- Zertifikat: erfüllbare Belegung




### Resolution

- Algorithmus zum Nachweis der Unerfüllbarkeit einer KNF
- „joinen" von zwei Klauseln auf Basis von einem Literal, das in der einen Klausel normal und in der anderen als Komplement vorkommt > Resolvent
- Terminiert bei leerer Klausel (nicht erfüllbar) □
- Lineare Resolution: der Erzeugte Resolvent muss im nächsten Schritt verwendet werden
- Positive Resolution: jeder neu erzeugte Resolvent muss positiv sein
- Positive Klausel: Klausel enthält ein Literal, welches nicht komplementiert ist

---

## 3.3 Prädikatenlogik

### Was ist Prädikatenlogik?

Die Prädikatenlogik ist eine Erweiterung der Aussagenlogik. Sie führt zusätzlich zu den Begriffen „und", „oder", „nicht" und „wenn … dann" noch Begriffe wie „für alle", „es gibt ein" und „ist" ein.

| | |
|---|---|
| $P(\bullet)$ | Prädikat ist eine Abbildung von z.B. $P(x)$ = „x ist sterblich" |
| $\bullet$ | Individuum, als Variable instantiert |
| $P(\bullet, \bullet)$ | Prädikat, dass Paare v. Individuen auf Aussagen abbildet |
| | z.B. $P(x, y)$ = „x mag y" – es sind auch Prädikate mit höhere Arität möglich |
| $m(\bullet)$ | Funktion, die ein Individuum auf ein andere Individium abbildet |
| | z.B. $m(x)$ = „Mutter von x" |

Syntax: Vokabular

| | |
|---|---|
| $x, y, z, \ldots$ | (Individuen)variablen |
| $a, b, c, \ldots$ | Konstanten |
| $P, Q, R, \ldots$ | Prädikatsymbole |
| $f, g, h, \ldots$ | Funktionssymbole |
| $=$ | Gleichheitssymbol |
| $\text{true}, \text{false}$ | logische Konstanten |
| $\wedge, \vee, \neg, \rightarrow$ | logische Operatoren |
| $\forall, \exists$ | Quantoren |
| $(,)$ | Hilfssymbole |

- Vrs, PrS, Fns bezeichnen (disjunkte) Mengen der Variablen, Prädikate und Funktionssymbole
- ar: PrS $\uplus$ FnS $\rightarrow \mathbb{N}$ ordnet jedem Prädikaten- und Funktionssymbol eine Stelligkeit/Arität zu

Syntax: Formationsregeln

1. Jede Variable und jede Konstante ist ein Term
2. Sind $t_1, \ldots, t_n$ Terme und $ar(f) = n$, dann ist $f(t_1, \ldots, t_n)$ ebenfalls ein Term
3. Sind $t_1, \ldots, t_n$ Terme und $ar(P) = n$, dann ist $P(t_1, \ldots, t_n)$ eine Formel
4. Sind $t$ und $u$ Terme, dann ist $t = u$ eine Formel
5. $\text{true}, \text{false}$ sind Formeln
6. Ist $F$ eine Formel, dann ist auch $\neg F$ eine Formel
7. Sind $F$ und $G$ eine Formel, dann sind $(F \wedge G), (F \vee G)$ und $(F \rightarrow G)$ ebenfalls Formeln
8. Ist $x$ eine Variable und $F$ eine Formel, dann sind $\forall x F$ und $\exists x F$ ebenfalls Formeln

$t := x \mid a \mid f(t_1, \ldots, t_n)$

$F := \text{true} \mid \text{false} \mid P(t_1, \ldots, t_n) \mid t_1 = t_2 \mid \neg F \mid (F \wedge G) \mid (F \vee G) \mid (F \rightarrow G) \mid \forall x F \mid \exists x F$

Syntaxbaum/Teilformel analog zur Aussagenlogik.

Gültigkeitsbereich (scope)

- Gültigkeitsbereich (scope) eines Vorkommens einer Variable $x$ in einer Formel $F$ ist die kleinste Unterformel von $F$ der Gestalt $\forall x G$ oder $\exists x G$, die das Vorkommen enthält *(x ist gebunden)*
- Wenn es diese Unterformel nicht gibt, ist die Formel $F$ selbst der scope *(x ist frei)*
- Formeln ohne freie Vorkommnise von Variablen sind geschlossen

| | |
|---|---|
| $\exists x P(x) \wedge Q(x)$ | Formel mit gebundenem und freien Vorkommen v. $x$ |
| $\forall x (P(x) \wedge \exists y (P(y) \vee Q(x, y)))$ | geschlossene Formel |

Semantik

- Semantik einer Formel ist die Funktion, die jeder möglichen „Welt“, die zu Formel „passt“, den Wahrheitswert der Formel (0 oder 1) dieser „Welt“ zuordnet.

Struktur

- Struktur ist der Fachbegriff für Welt
- Definition: Eine Struktur $S = (U_S, I_S)$ besteht aus einer nicht leeren Mege $U_S$ (Universum/ Grundmenge) und einer partiellen Abbildung $I_S$ (Interpretation) mit:
  - Falls $I_S(x)$ für $x \in$ VrS definiert ist, dann $x^S := I_S(x) \in U_S$
  - Falls $I_S(f)$ für $f \in$ FnS definiert ist, dann $f^S := I_S(f) : U_S^{ar(f)} \rightarrow U_S$
  - Falls $I_S(P)$ für $P \in$ PrS definiert ist, dann $f^S := I_S(f) \subseteq U_S^{ar(P)}$

### Semantik: Passende Struktur

- Definition: Eine Struktur $S = (U_S, I_S)$ passt zum Term $t$, falls $I_S$ für jede in $t$ vorkommende Variable und für jedes in $t$ vorkommende Funktionssymbol definiert ist.
- Definition: Eine Struktur $S = (U_S, I_S)$ passt zu einer Funktion $F$, falls $I_S$ für jede in $F$ frei vorkommende Variable und für jedes in $F$ vorkommende Funktions- und Prädikatensymbol definiert ist.

### Semantik: Formel

- Semantik einer Formel $F$ wird mit $[F](\bullet)$ bezeichnet
- Sie bildet jede zu $F$ passende Struktur $S$ auf einen Wahrheitswert $[F](S) \in \{0,1\}$ ab
- Definiton folgt Syntaxbaum
- Eine zu $F$ passende Struktur passt nicht automatisch zu allen Teilformeln von $F$
  Bsp: zu $F = \forall x P(x)$ passende Struktur $S$ muss $x^S$ nicht definieren und würde damit nicht zur Teilformel $P(x)$ passen

### Semantische Äquivalenzen

- Standardäquivalenzen für die Junktoren $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$ gelten auch in der Prädikatenlogik

| | |
|---|---|
| $\neg \forall F \equiv \exists x \neg F$ | DeMorgan für Quantoren |
| $\neg \exists F \equiv \forall x \neg F$ | DeMorgan für Quantoren |
| $(\forall x F \vee \forall x G) \models \forall x (F \vee G)$ | Distributivität für Quantoren |
| $\exists x (F \wedge G) \models (\exists x F \wedge \exists x G)$ | Distributivität für Quantoren |

Fehlender Teil Prädikatenlogik…

# 4: Kombinatorik

## 4.1 Einführung

**Was ist Kombinatorik?**
Kombinatorik beschäftig sich mit dem effektiven Abzählen von komplizierten Mengen.
Jedes Abzählproblem lässt sich auf Teilmengen von $[n]^k$ oder $\mathbb{N}_0^k$ zurückführen:

| | |
|---|---|
| $\{(s_1, \ldots, s_k) \in [n]^k : \lvert \{s_1, \ldots, s_k\} \rvert = k\}$ | $k$ verschiedene aus $n$, geordnet |
| $\{(s_1, \ldots, s_k) \in [n]^k : s_1 < s_2 < \ldots < s_k\}$ | $k$ verschiedene aus $n$, ungeordnet |
| $\{(s_1, \ldots, s_k) \in [n]^k : s_1 \leq s_2 \leq \ldots \leq s_k\}$ | $k$ aus $n$, ungeordnet |
| $\{(s_1, \ldots, s_k) \in \mathbb{N}_0^k : s_1 + s_2 + \ldots + s_k = n\}$ | <missing text> |
| $\{(s_1, \ldots, s_k) \in \mathbb{N}_0^k : s_1 + s_2 + \ldots + s_k \leq n\}$ | <missing text> |

## 4.2 Einfache Zählregeln

**Diskunkte Mengen**
- Sind $A, B$ beliebige Mengen für die gilt: $A \cap B = \varnothing$, dann auch $\lvert A \cup B \rvert = \lvert A \rvert + \lvert B \rvert$
- Falls $A \cap B \neq \varnothing$, dann $\lvert A \cup B \rvert = \lvert A \rvert < \lvert B \rvert$ und $\lvert A \cup B \rvert = \lvert A \rvert + \lvert B \rvert - \lvert A \cup B \rvert$

**Kartesisches Produkt von Mengen**
- Sind $A, B$ beliebige Mengen, gilt: $\lvert AB \rvert = \lvert A \times B \rvert = \lvert A \rvert \bullet \lvert B \rvert$

**Abbildungen von Mengen auf andere**
- Sind $A, B$ beliebige Mengen für die gilt: $f : A \to B$ mit $\lvert f^{-1}(b) \rvert = m$ konstant für alle $b \in B$, d.h. jeden Element auf $B$ hat genau $m$ Urbilder bzgl. $f$, dann auch $\lvert A \rvert = m \lvert B \rvert$ ($f$ ist bijektiv)

## 4.3 Urnenmodell

**Ziehen ohne Zurücklegen, mit Beachtung der Reihenfolge**

$$A_{n,k} := \{(s_1, \ldots, s_k) \in [n]^k : \lvert \{s_1, \ldots, s_k\} \rvert = k\}$$
$$\lvert A_{n,k} \rvert = n \bullet (n-1) \bullet \ldots \bullet (n-k+1) = \frac{n!}{(n-k)!}$$

**Ziehen ohne Zurücklegen, ohne Beachtung der Reihenfolge**

$$B_{n,k} = \{(s_1, \ldots, s_k) \in [n]^k : s_1 < s_2 < \ldots < s_k\}$$
$$\lvert B_{n,k} \rvert = \frac{n!}{(n-k)! \bullet k!} = \binom{n}{k}$$

**Ziehen mit Zurücklegen, ohne Beachtung der Reihenfolge**

$$C_{n,k} = \{(s_1, \ldots, s_k) \in [n]^k : s_1 \leq s_2 \leq \ldots \leq s_k\}$$

| (1,1,2,4,5,5,) | (2,1,0,1,2) | \|\|,\|,,\|,\|\| | (1,2,4,7,9,10) | $\lvert D_{n,k} \rvert = \binom{n+k-1}{n}$ |
|---|---|---|---|---|
| Münze > Person | Person > \|Münzen\| | | Pos. der Striche | |

## 4.4 Weitere Verteilungsprobleme

### Stirling Zahlen 2. Art

Beispiel 1: Verteilen von $k$ Geschenken auf $n$ Kinder, so dass jedes Kind mindestens ein Geschenk erhält. (Kinder und Geschenke sind unterscheidbar)

$F_{n,k} := \{(s_1, \ldots, s_k) \in [n]^k : |\{s_1, \ldots, s_k\}| = n\}$

$s_i$: ist das Kind, dass das $i$-te Geschenk erhält

- z.B. für $k = 6, n = 4$ das Tupel $(3,1,3,2,2,4)$
- Analog dazu: für jedes Kind die Untermenge der Geschenke $\rightarrow (\{2\}, \{4,5\}, \{1,3\}, \{6\})$
- Mathematisch: Ordne dem $k$-Tupel $(s_1, \ldots, s_k) \in F_{n,k}$ die geordnete Partition von $[k]$ in genau $n$ Klassen zu, die gegeben wird durch: $i, j \in [k]$ liegen in derselben Klasse, falls $s_i = s_j$

Beispiel 2: Verteilen von $k$ Geschenken auf $n$ Päckchen, so dass jedes Päckchen mindestens ein Geschenk enthält. (Päckchen sind nicht unterscheidbar)

- Entspricht einer ungeordneten Partition von $[k]$ in $[n]$ Äquivalenzklassen

  $\{\{2\}, \{4,5\}, \{1,3\}, \{6\}\} = \{\{4,5\}, \{2\}, \{1,3\}, \{6\}\}$ vgl. Bsp. 1 mit Tupel (geordnet)
- Für jede Partition von $n$ Elementen in $k$ (nicht leeren) Klassen gibt es $k!$ geordnete Partitionen

$S_{n,k}$ Stirling Zahlen zweiter Art (Anzahl Partitionen von $n$ Elementen in $k$ Klassen)

### Siebformel

Zwei Techniken:

- Abzählen des Komplements: statt $|X| : X \subseteq U$ direkt zu berechnen, kann es einfacher sein $X^C = U \setminus X$ zu bestimmen und $|X| = |U| - |X^C|$ zu berechnen
- Inklusion und Exklusion: Formel für Bestimmung von $|X|$, wenn $X = \bigcup_{i=1}^{n} Y_i$, wobei die $Y_i$ nicht zwingend disjunkt sind.
- Es gilt: $n^k = |F_{n,k}^C| + |F_{n,k}|$ - es reicht also auch $|F_{n,k}^C|$ zu bestimmen
- $|F_{n,k}^C|$ enthält alle $k$-Tupeln, in denen mindestens ein Element aus $[n]$ nicht vorkommt

  $|F_{n,k}^C| = \bigcup_{x \in [n]} ([n] \setminus \{x\})^k$ Problem: $|F_{n,k}^C| < \sum_{x \in [n]} |[n] \setminus \{x\}|$
- Diese Summation kann mit der Siebformel korrigiert werden!

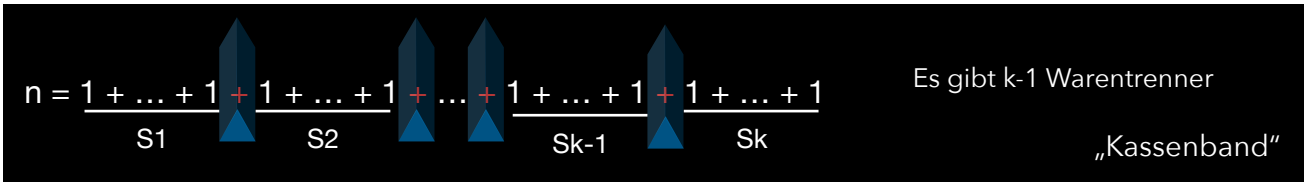
  $$|X| = \sum (-1)^{r-1} \sum_{I \subseteq [m] : |I| = r} |\bigcap_{i \in I} Y_i|$$

### Geordnete Zahlenpartitionen

Wieviele Möglichkeiten gibt es, $n$ Euro unter $k$ Kindern zu verteilen, so dass jedes Kind mindestens 1 Euro erhält? (Euros sind nicht unterscheidbar, Kinder schon)

$G_{n,k} := \{(s_1, \ldots, s_k) \in \mathbb{N}^k : s_1 + \ldots + s_k = n\}$ „sortierter Zählvektor"

- Jedes Tupel $(s_1, \ldots, s_k)$ kann geschrieben werden als:



- Anzahl der geordneten Partitionen ist gleich der Anzahl der Möglichkeiten, $k+1$ Pluszeichen aus insgesamt $n-1$ Pluszeichen auszuwählen

$$|G_{n,k}| = \binom{n-1}{k-1}$$

### (Ungeordnete) Zahlenpartitionen

Wieviele Möglichkeiten gibt es, $n$ Euro in $k$ Päckchen aufzuteilen, so dass jedes Päckchen mindestens 1 Euro enthält? (Euros und Päckchen sind nicht unterscheidbar)

$$P_{n,k} = \{(s_1, \ldots, s_k) \in \mathbb{N}^k : s_1 + \ldots + s_k = n, s_1 \leq \ldots \leq s_k\}$$

### Kombinatorik Zusammenfassung

$|\{s_1, \ldots, s_k) \in [n]^k\}| = n^k$

Ziehen mit Zurücklegen, unter Beachtung der Reihenfolge.

$|\{(s_1, \ldots, s_k) \in [n]^k | \{s_1, \ldots, s_k\}| = k\}| = \frac{n!}{(n-k)!}$

Ziehen ohne Zurücklegen, unter Beachtung der Reihenfolge.

$|\{(s_1, \ldots, s_k) \in [n]^k : s_1 < \ldots < s_k\}| = \frac{n!}{k!(n-k)!} = \binom{n}{k} = \binom{n}{n-k}$

Ziehen ohne Zurücklegen, ohne Beachtung der Reihenfolge.

$|\{(s_1, \ldots, s_k) \in [n]^k : s_1 \leq \ldots \leq s_k\}| = \frac{n!}{k!(n-k)!} = \binom{k+n+1}{k}$

Ziehen mit Zurücklegen, ohne Beachtung der Reihenfolge.

$|\{(s_1, \ldots, s_k) \in \mathbb{N}_0^k : s_1 + \ldots + s_k = n\}| = \binom{n+k-1}{n}$

$|\{(s_1, \ldots, s_k) \in \mathbb{N}_0^k : s_1 + \ldots + s_k \leq n\}| = \binom{n+k}{n}$

$|\{(s_1, \ldots, s_k) \in [n]^k : |\{s_1, \ldots, s_k\}| = n\}| = n! S_{k,n}$

mit $S_{n+1,k} = S_{n,k-1} + kS_{n,k}, S_{0,0} = 1, S_{n+1,0} = 0, S_{n,n} = 1$

„Stirling Zahlen 2. Art: Anzahl Partitionen von $[n]$ in $k$ Klassen."

$|\{(s_1, \ldots, s_k) \in \mathbb{N}_0^k : s_1 + \ldots + s_k = n, s_1 \leq \ldots \leq s_k\}| = |P_{n+k,k}|$

mit $|P_{n,k}| = |P_{n-1,k-1}| + |P_{n-k,k-1}|, |P_{0,0}| = 1, |P_{n+1,0}| = 0, |P_{n,n}| = 1$

„Anzahl Zahlpartition von $n$ in $k$ positive Summanden"

$|A_1 \times A_2 \times \ldots \times A_n| = \prod_{i \in [n]} |A_i|$ Produktregel

$|\bigcup_{i \in \mathbb{N}_0} A_i| = \sum_{i \in \mathbb{N}_0} |A_i|$ Summenregel

$|\bigcup_{i \in \mathbb{N}_0} A_i| \leq \sum_{i \in \mathbb{N}} |A_i|$ union bound

$|\bigcup_{i \in \mathbb{N}_0} A_i| = \sum_{I \subseteq [n] : I \neq \emptyset} (-1)^{|I|+1} |\bigcap_{i \in I} A_i|$ Siebformel

$|A| = m|B|$, falls es ein $f : A \to B$ mit $|f^{-1}(b)| = m$ konstant gibt.

# 5: Algebra

## 5.1 Gruppen

### Was sind Gruppen?

Gruppen sind algebraische Strukturen, die folgende Eigenschaften erfüllen

$\mathbb{G} := <$ 'menge', 'op', 'neutrl.' $>$

| | |
|---|---|
| 'op' : $\mathbb{G} \times \mathbb{G} \to \mathbb{G}$ | abgeschlossen |
| $\forall a, b, c \in \mathbb{G} : (a$ 'op' $b)$ 'op' $c = a$ 'op' $(b$ 'op' $c)$ | assoziativ |
| $\forall a \in \mathbb{G} : a$ 'op' 'neutr.' $=$ 'neutr.' 'op' $a = a$ | Neutrales |
| $\forall a \in \mathbb{G}\ \exists b \in \mathbb{G} : a$ 'op' $b = b$ 'op' $a =$ 'neutr.' | Inverses |
| $\forall a, b \in \mathbb{G} : a$ 'op' $b = b$ 'op' $a$ | kommutativ |

### Erweiterter Euklidischer Algorithmus (EEA)

$\alpha_{\text{neu}} = \beta_{\text{alt}} - k_{\text{neu}} * \alpha_{\text{alt}} \qquad \beta_{\text{neu}} = \alpha_{\text{alt}}$

### Inverses Element in multiplikativer Gruppe

Falls die Grundmenge $\mathbb{Z}_n^*$: EEA mit $a = a$ und $b = n \to \alpha = a^{-1}$

Weiterhin gilt: vorletztes Element von $< a > = a^{-1}$

### Kardinalität von $\mathbb{Z}_n^*$

Kardinalität von einer Teilergruppe kann mit der Eulerschen-Phi-Funktion berechnet werden,

$\phi(p^r \in \mathbb{P}) = (p-1) * p^{r-1}$

$\phi(m * n) = \phi(m) * \phi(n)$

### Ordnung eines Elements

Die Ordnung eines Elements entspricht der Kardinalität der induzierten Bahn.

$\text{ord}(a) = \text{ord}(a^{-1}) = | < a > |$

$< a > = \{a$ 'op' $\times\, k : k \in \mathbb{N}\}$

$\text{ord}(\pi) = \text{kgV}($ Zyklenlängen $)$

Ordnungen einer Teilergruppe können nur Teiler der Kardinalität der Gruppe sein.

### Gruppenexponent

Der Gruppenexponent von endlichen Gruppen ist das kleinste gemeinsame Vielfache aller Gruppenordnungen

$\text{kgV}(\text{ord}(a) : a \in \mathbb{G})$

### Zyklische Gruppen & Erzeuger

Zyklische Gruppen beinhalten Elemente (Erzeuger), deren Bahn gleich der Gruppenmenge ist

$\exists\ g \in \mathbb{G} : < g > = \mathbb{G}$

- Jede zyklische Gruppe ist auch kommutativ
- Falls , ist $\mathbb{G}$ zyklisch
- $\mathbb{Z}_n^*$ hat $\phi(\phi(n))$ Erzeuger
- $S_n$ (Permutationen) für $n \geq 3$ sind nie zyklisch
- Zyklisch, falls gilt $\mathbb{Z}_n^* : n \in \{2,4,p^r,2p^r\} : p \in \mathbb{P}$

### Untergruppen

Nicht leere Teilmenge $H \subseteq \mathbb{G}$ ist Untergruppe von $\mathbb{G}$, falls 'op' auf $H$ eine Gruppe mit all ihren Eigenschaften (s.o.) definiert.

### Modulo & Restklassen

$$a^k \mod n = (a \mod n)^k \mod n$$
$$a^k \mod n = (a^{(k \mod |\mathbb{G}|)}) \mod n$$
$$a^k = a^{(k \mod \mathrm{ord}(a))}$$