# 1 Grundbegriffe und elementare Logik

**Junktoren**

|  |  |  |
| --- | --- | --- |
| Zeichen | Prädikat | Bezeichnung |
|  |  | **not** A |
|  |  | A **and** B |
|  |  | A **or** B |
|  |  | **if** A **then** B |
|  |  | A **==** B |

**Beweistechniken**

* **Direkter Beweis** A => B
* **Beweis durch Widerspruch:** Annahme A wäre Falsch => Widerspruch
* **Beweis durch Kontraposition**: A => B beweisen
* **Beweis durch Gegenbeispiel:** Beispiel wo nicht stimmt
* **Beweis durch Äquivalenz:** A ⬄ B beweisen A=> B und B=> A

Existiert genau ein:

(n Z): gerade Zahl: x=2n ungerade: x= 2n+1, d. 3 teilb. x= 3n, nicht d. 3 teilb. x=3n+1 und x=3n+2

**Junktor Regeln**

|  |  |  |
| --- | --- | --- |
| Doppelte Negation |  |  |
| Kommutativität |  |  |
| Assoziativität |  |  |
| Distributivität |  |  |
| De Morgan |  |  |
| Implikation | A => B | ⬄ |
| Äquivalenz | A ⬄ B | ⬄ *(B=>A)*  ⬄ |
| Tautologie / Widerspruch |  |  |

**Quantor Regeln**

|  |  |  |
| --- | --- | --- |
| Vertauschungsregel |  |  |
| Vertauschungsregel |  |  |
| Allquantor |  |  |
| Existenzquantor |  |  |

**Aussage**

Unter einer Aussage wollen wir ein «sprachliches Gebilde» verstehen, welchem zumindest im Prinzip ein Wahrheitswert «wahr» oder «falsch» zugeordnet werden kann.

**Prädikat**

Ein Prädikat ist im Wesentlichen eine Aussage, die freie Variablen enthält.   
Aussagen sind 0-stellige Prädikate. Aussage mit N freien Variablen sind N-stellige Prädikate.

**Quantoren**

|  |  |  |
| --- | --- | --- |
| Zeichen | Bezeichnung | Beispiel |
|  | Allquantor |  |
|  | Existenzquantor |  |

Distributiv gilt nicht! Aber gleiche Quantoren dürfen zsm gefasst werden. Reihenfolge ist wichtig!

**Reihenfolge der Bindung: (Syntaxbaum zeichnen!)**

**Junktoren Wahrheitstabellen**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | **B** |  |  | A => B | A ⬄ B |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 |

# 2 Syntax und Semantik

**Gleiche Regeln bei Aussagenlogischen Formeln wie bei Junktoren**

**Äquivalente Formeln** werden folgendermassen geschrieben:

1. Impliaktion eliminieren, 2. Negation in Klammern hinein, 3. Distributiv Gesetz anwendenf
2. Kann auch aus Wahrheitstabelle (0 => Knf (minterm), 1 => DNf)

**Negations Normalform (NNF)**

* Alle Negationen kommen in Literalen vor (direkt vor Buchstabe nicht vor ()
* Es gibt keine Implikationen

**Disjunktiver Normalform (DNF)**

* or p ∧ (q ∧ p) or p∨(q∨ (q ∧ p))

**Konjunktiver Normalform (KNF)**

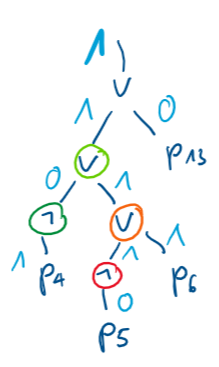
* or (p ∨ q) ∧ (p ∨ (p ∨ p)) or p ∧ (q ∧ p)

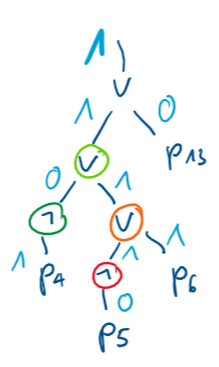
**Wahrheitstabelle / Baum / Belegung**

* Der letzte Eintrag der ersten Zeile ist die Formel *F*.
* Wenn *A* in einer Spalte vor *B* erscheint, dann ist *A* eine Teilformel von *B*.

Beispiel:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C |  |  |  |
| 0 | 0 | 0 | False | True | True |
| 0 | 0 | 1 | False | True | True |
| 0 | 1 | 0 | False | True | True |
| 0 | 1 | 1 | True | False | False |
| 1 | 0 | 0 | False | True | True |
| 1 | 0 | 1 | False | True | True |
| 1 | 1 | 0 | False | True | True |
| 1 | 1 | 1 | True | False | True |





**Belegung (völlig willkürlich, dass (wahr/falsch) eigentlich)**

* kann Ausgewertet werden mit: Wahrheitstabelle, Baum, Belegung

Die Funktion ordnet jeder aussagenlogischen Formel ihren Wahrheitswert bezüglich dessen Belegung zu.



Beispiel: Bestimmen Sie der Formel **(als Gleichungslösung)**

Eine aussagenlogische Formel *A* heisst

Allgemeingültig (Taut.) Alle Belegungen

Unerfüllbar (Wid.) Alle Belegungen

Erfüllbar Min. Eine Belegung

Widerlegbar Min. Eine Belegung

Gültig/Wahr(unter 1 Belegung)

# 3 Mengen, Relationen und Graphen

**Rechenregeln**

Mächtigkeit/Kardinalität: |X| = Anz. Elemente (|∅| = |{}| = 0 und |{∅}|=1 (+ {} )

Zwei Mengen die keine gemeinsamen Objekte haben heissen disjunkt.

Kommutativität der Vereinigung und des Schnittes:

Assoziativgestze von Schnitt und Vereinigung:

Distributivgesetze von :

Idempotenzgesetz:

Regeln von De Morgan:

Notationen

* Explizite Schreibweise: {1, 2, 3, 4, …}
* =
* =
* Bsp. Prädikat: {F(x |xN} ={2x|x

Zahlenmengen

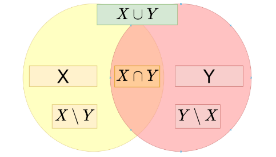
* = Natürliche Zahlen = {0, 1, 2, 3, …}
* ∅ (leere Menge) = {} (nicht reflexif!)

Teilmengen

* X ist eine *Teilmenge* von Y:
* X ist eine echte *Teilmenge* () von Y:
* A\Y [mit A = Grundmenge] = Komplement/ Komplementärmenge von X[anderer Teilmenge]

Operationen

* Vereinigung
* Schnittmenge}
* Differenz



**Potenzmenge**

mit Mächtigkeit

* Jede Potenzmenge enthält die leere Menge

Beispiel

* }

**Kartesisches Produkt** = Kreuzprodukt (Mächtigkeit = |AxB| = |A|\*|B|, keine Kommutativität) ( ein Element davon ist ein Tupel)

Das *kartesische Produkt* von , ist die Menge aller -Tupel mit Einträgen aus den Mengen das alle möglichen Kombinationen enthält

Beispiel =

∅ × {∅} = ∅

**Partitionen und Blöcke**

Eine Partition Zerlegung in Teilmengen (Beliebige disjunkte.)

* Die Elemente von *P* sind nicht leer und paarweise disjunkt
* Partition = Menge aller Äquivalenzklassen

Die Elemente einer Partition werden Blöcke genannt.



**Surjektiv, Injektiv und Bijektiv**

Surjektiv Elemente der Zielmenge werden *mindestens* einmal getroffen (rechtstotal).   
Injektiv Elemente der Zielmenge werden *maximal* einmal getroffen (linkseindeutig).

Funktion: linkstotal (mind 1 Pfeil nach B für jedes Element von A), rechtseindeutig (max. 1 Pfeil nach recht für jedes Element von A)

Bijektive Relation: rechtstotal, linkseindeutig (Surjektiv + Injektiv = Bijektiv)

**Rechenregeln:**

Falls X=> Y Injektiv und Y=>Z Injektiv dann ist auch X=>Z Injektiv

Falls X=> Y Surjektiv und Y=>Z Surjektiv dann ist auch X=>Z Surjektiv

**Bsp:** abababab auf endliche Wörter mit a,b,c =>

ab =>b, aa => a, bb => c, ba => c zbsp.

Binär, bei 0 trennen

**Ordnungstypen**

Es sei *R* eine binäre Relation auf der Menge *M*.

* Totale Ordnung Kein unvergleichbares Element (+ Halbordnung)
* Wohlordnung mind. 1 min. Element pro Teilmenge (+ Totale Ordnung)
* Wohldefinierte Funktion Jedes Element aus Äquivalenzklassse hat gleichen Fuktionswert

**Relationen (Teilmenge eines Kartesischen Produkts)**

2 Stellige (binäre) Relation mit , dann schreiben wir auch .

Verbindung von Relationen mit Komposition

**Darstellung**

Aufzählend: R= {(1,2), (2,1)}, Beschreibend: aRb «b ist … a», Tabelle/Matrix, Graph, Bipartiter Graph (siehe unten rechts), gerichteter Graph ( nur Homogene Relationen (unten links)

**Unvergleichbar**

Zwei Elemente heissen R-unvergleichbar, falls weder noch gilt.

**Minimal / Maximal**

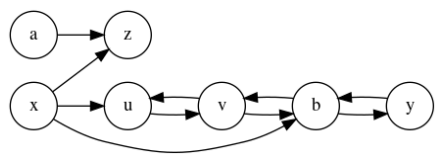
Ein Element einer Teilmenge von *M* heisst

* *R-minimal* in X, falls es kein anderes Element mit *yRx* gibt. (obe)
* *R-maximal* in X, falls es kein anderes Element mit *xRy* gibt. (unne)

**Graphen**

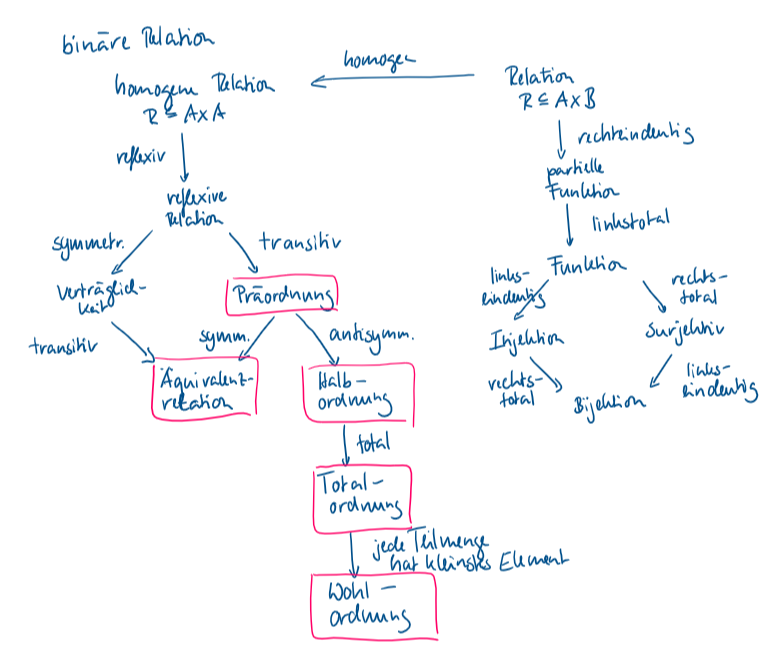
Die Relation auf sei durch den Graph gegeben.

Minimal (Nur ausgehende Pfeile) =   
Maximal (Nur eingehende Pfeile) =



Eine binäre Relation auf eine Menge heisst:

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung 

**Abzählbar**

Eine Menge heisst *abzählbar*, wenn … (oder) … .

* Gleichmächtig wie N
* können durchnummeriert werden (Wiederholungen erlaubt)
* Schubfachprinzip: Wenn n Objekte auf m Behälter verteilt werden und n>m gilt, dann gibt es keine Injektive Funktion und gäbe es eine Injektive Funktion wäre die Menge m unendlich
  + Die Menge A ist Abzählbar
  + Es gibt eine surjektive Funktion
  + Es gibt eine injektive Funktion
  + Es gibt eine bijektive Funktion
  + Es gibt eine bijektive Funktion

**Überabzählbar**

Eine Menge heisst *überabzählbare* wenn sie…

* Nicht abzählbar ist

**Hasse-Diagramm**

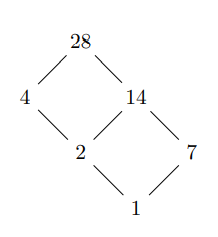
Das Hasse-Diagramm bietet eine Vereinfachte Darstellung einer Halbordnungs-Relation.

* Pfeile werden weggelassen. Der Graph geht nach «oben».
* Verbindungen zwischen zwei Punkten werden weggelassen, wenn es bereits eine «indirekte Verbindung» gibt.
* Verbindungen von einem Punkt zu sich selbst werden weggelassen.

Beispiel

Teilbarkeitsrelation auf der Menge

* Minimale Elemente
* Maximale Elemente



**Äquivalenzklasse**

Gleiche Eigenschaften wie Partition

Notation : xRy : und dann gilt: |x|~ =|y|~ (Äquivalente Elemente repräsentieren stets dieselbe Äquivalenzklasse und jedes Element der Äquivalenzklasse ist auch Repräsentant derselben)

**Faktormenge = Menge aller Äquivalenzklassen**

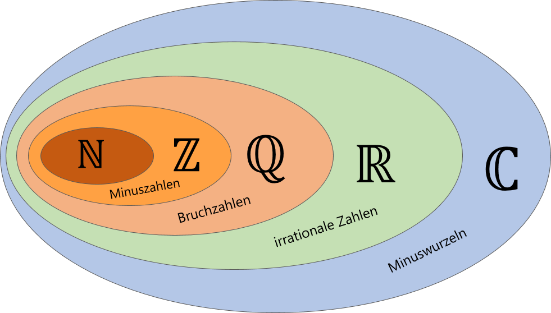
**DAG** (**D**irected **A**cyclic **G**raph) Gegenteil: Zyklus

Jeder endliche DAG besitzt (mindestens) eine topologische Sortierung.

**Topologische Sortierung = Reihenfolge eindeutig**

Es sei *M* eine endliche Menge und ein DAG. Eine lineare Ordnung ist eine *topologische Sortierung* von G, wenn für alle gilt:

|  |  |
| --- | --- |
| Abzählbar | Überabzählbar |
|  |  |
|  |  |
| (0,0) (0,1) Diagonalarg. |  |
|  |  |
|  | (unendlich grosse Teilmengen) |
| Menge aller Programme | Menge aller Funktionen |
| Jede endliche Menge / Teilmenge einer abzählbaren Menge | Beliebige Teilmenge einer abzählbaren Funktion (hat kein max /minima kann Menge sich selbst sein) |
| Vereinigung von abzählbar vielen abzählbaren Mengen |  |
| (und alles was abzählbar ist auch gekreuzt) |  |



# 4 Rekursive Strukturen und die natürlichen Zahlen

**Fakulatät: F!** f(n) = n\*(n-1)\*(n-2)\*…\*3\*2\*1

**Darstellungen:**

* **Aufzählend:** f(1), f(2), f(3), …
* **Explizit:** f(n) = …
* **Rekursiv:** f(1) =… , f(n) = f(n-1)… / f(n+1) = f(n) \*…

**Peano Axiome**

* 0 ist eine natürliche Zahl.
* Jede natürliche Zahl {\displaystyle n}n hat eine natürliche Zahl {\displaystyle n'}n' als Nachfolger.
* 0 ist kein Nachfolger einer natürlichen Zahl.
* Natürliche Zahlen mit gleichem Nachfolger sind gleich.
* Enthält die MengeX die 0 und mit jeder natürlichen Zahln auch deren Nachfolger n, so bilden die natürlichen Zahlen eine Teilmenge von X.

{\displaystyle 0\in \mathbb {N} }{\displaystyle \forall n(n\in \mathbb {N} \Rightarrow n'\in \mathbb {N} )}

**Induktion**

**IV: Induktionsverankerung:** (Überprüfung für kleinstes Element)

**IA: Induktionsannahme:** (Annahme für n ist die Aussage wahr)

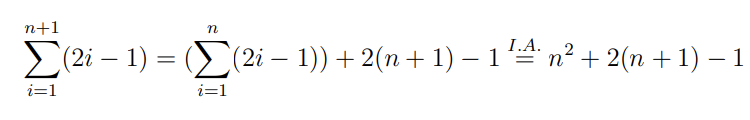
**IB: Induktionsbehauptung:** (Behauptung für n+1 ist die Aussage wahr)

**IS: Induktionsschritt:** Mit Hilfe der Annahme und dem Schritt n+ 1 muss sich die Behauptung für n+ 1 als wahr ergeben.

Oder: Bei IB: Methode des Kleinsten Verbrechers: Kleinste natürliche Zahl mit nicht der Eigenschaft E(x). Jede Teilmenge (nicht leer der natürlichen Zahlen hat ein kleinstes Element. => Widerspruch.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung



Ein Bild, das Text enthält.

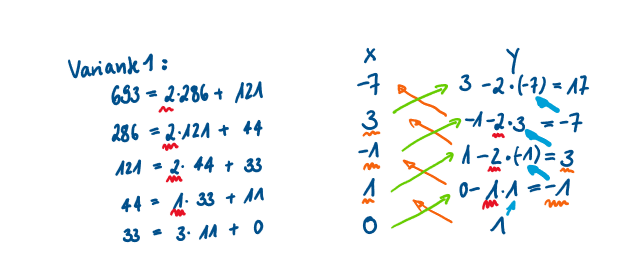
Automatisch generierte Beschreibung

Stetig absteigend in N-> N (Halbordnung) f(x) =< g(x): gn(x) = max(0, x − n)

# 5 Elementare Zahlentheorie

**Lemma von Bézout (= Lineare diophantische Gleichung)**

(es gibt x,y ≠0 sodass a, b existiert, kann mit erw. Euklid. Algorithmus ein Zahlenpaar bestimmt werden)



Variante 2

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Vor die grössere Zahl x/y kommt die kleinere von a/b.

**Teilbarkeit**

Sind , falls es kein gibt mit .

**Euklidischer Algorithmus**

Für mit 0 < n < m gilt

**Definition kgV und ggT**

Seien . *Kleinstes gemeinsames Vielfaches* von und :

Ist oder . *Grösster gemeinsamen Teiler* von und :

(Berechnung mit S. v. Euklid / Primfaktorzerlegung)

* n\*m = kgV(n,m)\*ggT(n,m)
* ggT(n,1) = 1 / ggT(n, 0) = n / ggT(n, n) = n /
* falls n, m teilerfremd: ggT(n,m) = 1
* ggT(n, m)= ggT(m, n)
* ggT(n, m) = ggT(n, m-n)
* ggT(n, m) = ggT(n, m-k\*n)

**Verknüpfungstabelle**

Die Verknüpfungstabelle der Multiplikation in

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Primzahlen**

Eine natürliche Zahl ist eine *Primzahl* , wenn gilt.

1. Jede Primzahl hat genau zwei unterschiedliche Teiler
2. Jede ganze Zahl besitzt mind. einen Primfaktor
3. Es gibt unendlich viele Primzahlen (bw. Durch Widerspruch Produkt + 1)

**Multiplikatives Inverse**

Sind mit , so sagen wir sei invers zu und schreiben auch für . Bei Restklassen von Primzahlen ist jedes invers zu .

* hat EIN Inverses in
* We look for x such that:
  + a \* x = 1 mod m => Erweiterter Euklidischer Alg.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**Restklassen (= Äquivalenzklassen)**

Es sei beliebig. Wir definieren eine Relation auf wie folgt

Es sei beliebig. Für jede ganze Zahl bezeichnen wir mit

= Restklasse

Restklassenmenge:

Die Äquivalenzklasse von der Relation heisst *Restklassenmenge* von . (Es wird standardmässig immer der kleinste Vertreter der Restklassenmenge angegeben)

**Rechenregeln**

Wir definieren die Verknüpfungen und



Primes Restklassensystem:

nur Restklassen die zu n ein Multiplikativ Inverses besitzen bzw. teilerfremd sind.

**Primfaktorzerlegung**

Es sei jeweils die *i-te* Primzahl. Für gibt es eine eindeutig bestimmte, endliche Folge mit , so dass

Beispiel

**Multiplikatives Inverses bestimmen**

1. (Pröbeln)
2. Erweiterter Euklidischer Algorithmus

**Satz von Euler, Kl. Satz von Fermat**

ggT(a,m) = 1 => aphi(m)

Euler’sche Phi-Funktion

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Kl. Satz von Fermat

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**Chinesischer Restsatz**

Simultane Kongruenz

Hier muss jeweils mit Alg. Inverses bestimmt werden

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung