Tutorium
Automaten und Formale Sprachen
Teil 0.

Christopher Blöcker, B. Sc.
inf9900@fh-wedel.de

SS 2012

Aufgabe

Es sei eine Aussage der Form

$$\bigwedge_{x \in \mathbb{N}} h(x)$$

zu beweisen, dabei sei h ein Prädikat.

Satz

$$\underbrace{h(0)}_{\text{Verankerung}} \wedge \underbrace{\bigwedge_{n \in \mathbb{N}} [h(n) \to h(\sigma(n))]}_{\text{Schritt}} \to \underbrace{\bigwedge_{n \in \mathbb{N}} h(n)}_{\textit{Schluss}}$$

Dabei bezeichne $\sigma(n)$ den Nachfolger von n.

Satz

$$\underbrace{h(0)}_{\text{Verankerung}} \quad \wedge \underbrace{\bigwedge_{n \in \mathbb{N}} [h(n) \rightarrow h(\sigma(n))]}_{\text{Schritt}} \rightarrow \underbrace{\bigwedge_{n \in \mathbb{N}} h(n)}_{\textit{Schluss}}$$

Dabei bezeichne $\sigma(n)$ den Nachfolger von n.

Satz (in Worten)

Sei $h(x)$ eine Aussageform über der Menge der natürlichen Zahlen.

Wenn $h(0)$ wahr ist — *Verankerung*

und

wenn für jedes beliebige $n \in \mathbb{N}$ aus der Wahrheit von $h(n)$ stets die Wahrheit von $h(\sigma(n))$ folgt, — *Schritt*

dann

gilt die Aussage für alle natürlichen Zahlen. — *Schluss*

Beispiel: Peano'sches Axiomensystem

Axiomatische Definition der Menge der Natürlichen Zahlen $\mathbb{N}$.

1. **Verankerung**: Es gelte $0 \in \mathbb{N}$
2. $\sigma : \mathbb{N} \to \mathbb{N}$ sei eine Funktion, die jedem $n \in \mathbb{N}$ einen (eindeutigen!) Nachfolger zuordnet
3. σ sei injektiv, also $\sigma(m) = \sigma(n) \to m = n$
4. 0 ist Nachfolger keiner natürlichen Zahl, d.h. $\bigwedge_{n \in \mathbb{N}} \sigma(n) \neq 0$
5. **Induktionsaxiom**: Jedes $n \in \mathbb{N}$ kann durch endlich häufige Anwendung von σ auf 0 erzeugt werden $n = \underbrace{(\sigma \circ \ldots \circ \sigma)}_{n \text{ mal}}(0)$

Mengendefinition nach Cantor

Eine Menge ist eine Zusammenfassung bestimmter wohlunterschiedener Objekte unserer Anschauung oder unseres Denkens zu einem Ganzen. Von jedem dieser Objekte muss eindeutig feststehen, ob es zur Menge gehört oder nicht. Die zur Menge gehörenden Objekte nennt man die Elemente der Menge.

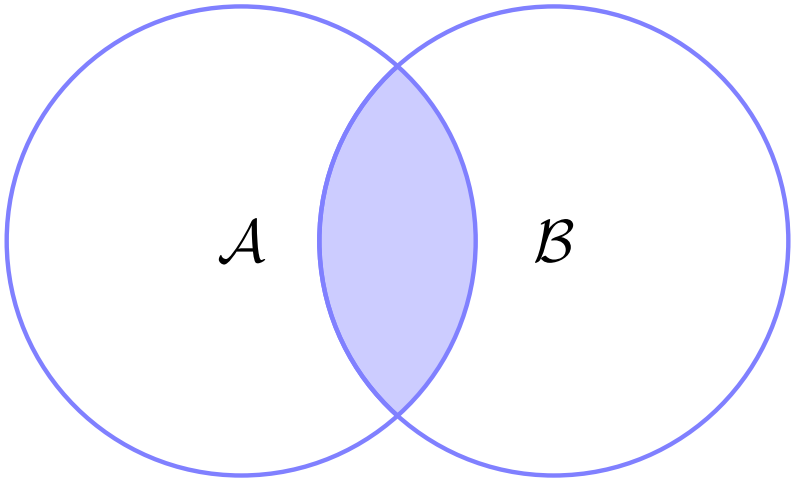
Notwendig

Wir fordern, dass die Elemente auf Gleichheit getestet werden können.

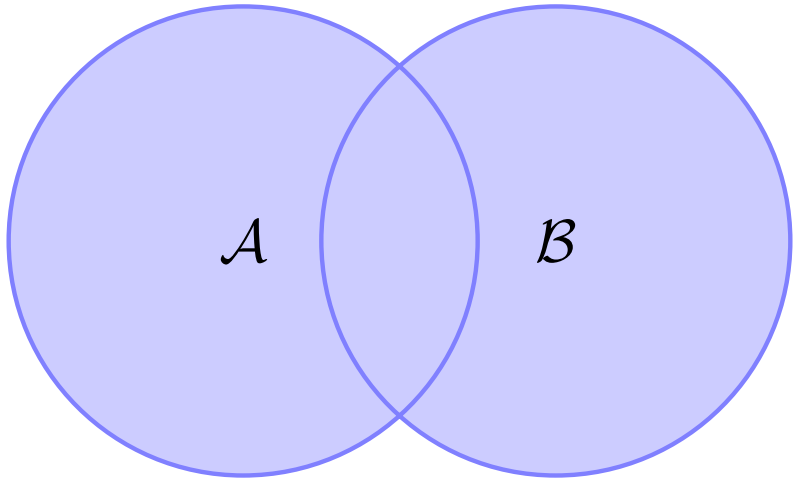
Mengenoperationen

$$
\begin{aligned}
\mathcal{A} \subseteq \mathcal{B} &\Leftrightarrow \bigwedge_{a \in \mathcal{A}} a \in \mathcal{B} \\
\mathcal{A} \cap \mathcal{B} &\Leftrightarrow \{ x \mid x \in \mathcal{A} \wedge x \in \mathcal{B} \} \\
\mathcal{A} \cup \mathcal{B} &\Leftrightarrow \{ x \mid x \in \mathcal{A} \vee x \in \mathcal{B} \} \\
\mathcal{A} \setminus \mathcal{B} &\Leftrightarrow \{ x \mid x \in \mathcal{A} \wedge x \notin \mathcal{B} \} \\
\mathcal{A} \triangle \mathcal{B} &\Leftrightarrow (\mathcal{A} \cup \mathcal{B}) \setminus (\mathcal{A} \cap \mathcal{B})
\end{aligned}
$$

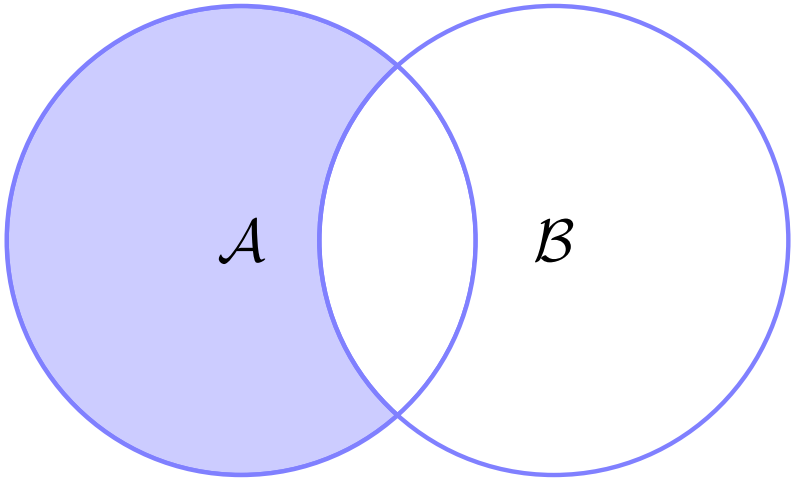
$\mathcal{A} \cap \mathcal{B}$



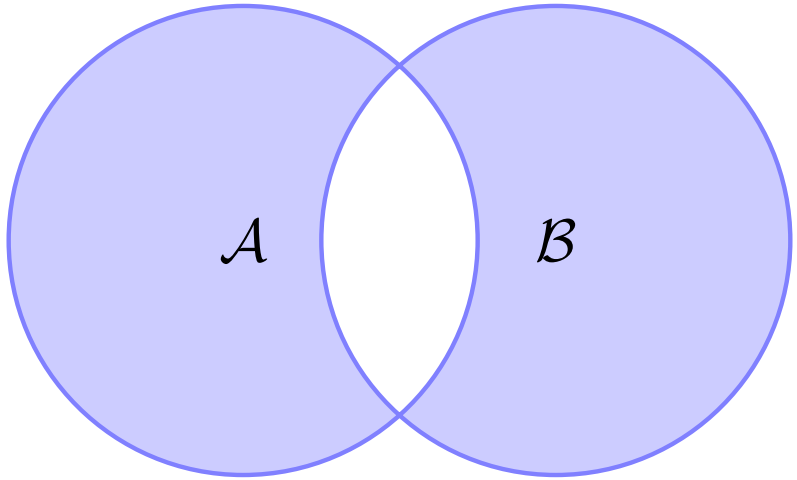
$\mathcal{A} \cup \mathcal{B}$



$\mathcal{A} \setminus \mathcal{B}$



$\mathcal{A} \Delta \mathcal{B}$



Frage

Es existiere ein Dorf, in dem es verboten sei, einen Bart zu tragen und

1. der Dorfbarbier denjenigen Männern den Bart schneide, die sich **nicht** selbst den Bart schneiden und
2. der Dorfbarbier der einzige Mann sei.

$\mathcal{M}$ sei die Menge der Männer, denen der Dorfbarbier den Bart schneidet, x sei der Dorfbarbier.

$$x \overset{?}{\in} \mathcal{M} \text{ bzw } \mathcal{M} \overset{?}{=} \emptyset$$

Antwort

Die Frage ist unentscheidbar, denn

$$x \in \mathcal{M} \Leftrightarrow x \notin \mathcal{M}.$$

Definition

Eine (2-stellige) Relation zwischen den Mengen $\mathcal{M}$ und $\mathcal{N}$ ist eine Teilmenge $\mathcal{R}$ der Produktmenge $\mathcal{M} \times \mathcal{N}$.

$$\mathcal{R} \subseteq \mathcal{M} \times \mathcal{N}.$$

Dabei steht $m \in \mathcal{M}$ mit $n \in \mathcal{N}$ in Relation, wenn

$$(m, n) \in \mathcal{R}.$$

Man schreibt dafür auch

$$m \overset{\mathcal{R}}{\sim} n \text{ oder } m\mathcal{R}n.$$

Äquivalenzrelation

Sei $\mathcal{R} \subseteq \mathcal{M}^2$. $\mathcal{R}$ definiert eine Äquivalenzrelation, wenn gilt

1. Reflexivität : $\bigwedge_{x \in \mathcal{M}} x \overset{\mathcal{R}}{\sim} x$
2. Symmetrie : $\bigwedge_{x,y \in \mathcal{M}} (x \overset{\mathcal{R}}{\sim} y) \rightarrow (y \overset{\mathcal{R}}{\sim} x)$
3. Transitivität : $\bigwedge_{x,y,z \in \mathcal{M}} [(x \overset{\mathcal{R}}{\sim} y) \wedge (y \overset{\mathcal{R}}{\sim} z)] \rightarrow (x \overset{\mathcal{R}}{\sim} z)$

Ordnungsrelation

Sei $\mathcal{R} \subseteq \mathcal{M}^2$. $\mathcal{R}$ definiert eine Halbordnung, wenn gilt

1. Reflexivität : $\bigwedge_{x \in \mathcal{M}} x \overset{\mathcal{R}}{\sim} x$

oder Irreflexivität : $\bigwedge_{x \in \mathcal{M}} x \overset{\mathcal{R}}{\nsim} x$

2. Antisymmetrie : $\bigwedge_{x,y \in \mathcal{M}} [(x \overset{\mathcal{R}}{\sim} y) \wedge (y \overset{\mathcal{R}}{\sim} x)] \rightarrow (x = y)$

oder Asymmetrie : $\bigwedge_{x,y \in \mathcal{M}} (x \overset{\mathcal{R}}{\sim} y) \rightarrow (x \neq y)$

3. Transitivität : $\bigwedge_{x,y,z \in \mathcal{M}} [(x \overset{\mathcal{R}}{\sim} y) \wedge (y \overset{\mathcal{R}}{\sim} z)] \rightarrow (x \overset{\mathcal{R}}{\sim} z)$

Definition

Ist zusätzlich die Eigenschaft der Linearität erfüllt, so definiert $\mathcal{R}$ eine Totalordnung.

1. Linearität : $\bigwedge_{x,y \in \mathcal{M}} (x \overset{\mathcal{R}}{\sim} y) \vee (y \overset{\mathcal{R}}{\sim} x)$

Algebraische Struktur

Eine algebraische Struktur ist ein Paar, bestehend aus einer Menge $\mathcal{M}$ und einer Familie von n_i-stelligen Verknüpfungen f_i auf den Elementen der Menge mit $f_i \colon \mathcal{M}^{n_i} \to \mathcal{M}$.

$$(\mathcal{M}, (f_i))$$

Gruppoid

Ein Gruppoid besteht aus einer Menge $\mathcal{M}$ und einer inneren, binären Verknüpfung $* \colon \mathcal{M} \times \mathcal{M} \to \mathcal{M}$.

1. $(\mathbb{Z}, -)$
2. $\left(\mathbb{N}, *(a, b) = a^b\right)$

Halbgruppe

Eine Halbgruppe besteht aus einer Menge $\mathcal{M}$ und einer inneren, binären, **assoziativen** Verknüpfung $* \colon \mathcal{M} \times \mathcal{M} \to \mathcal{M}$.
Es muss gelten

$$\bigwedge_{a,b,c \in \mathcal{M}} a * (b * c) = (a * b) * c$$

Monoid

Ein Monoid ist eine Halbgruppe mit ausgezeichnetem neutralen Element e: $(\mathcal{M}, *, e)$ mit

$$\bigwedge_{a \in \mathcal{M}} e * a = a = a * e$$

Gruppe

Eine Gruppe ist ein Monoid mit einer zusätzlichen unären Operation $^{-1}\colon \mathcal{M} \to \mathcal{M}$, die zu jedem Element das inverse bezüglich $*$ bestimmt: $\left(\mathcal{M}, *, e, ^{-1}\right)$.

$$\bigwedge_{a \in \mathcal{M}} a * a^{-1} = e = a^{-1} * a$$

ABEL'sche Gruppe

Eine ABEL'sche Gruppe ist eine Gruppe mit einer **kommutativen** Operation $*\colon \mathcal{M} \times \mathcal{M} \to \mathcal{M}$ mit

$$\bigwedge_{a,b \in \mathcal{M}} a * b = b * a$$

Definition

Seien m, n zwei beliebige Zahlen in $\mathbb{N}$.

Die Zahl $d \in \mathbb{N}$ heisst **größter gemeinsamer Teiler** der Zahlen m und n (kurz: **ggT**), wenn gilt:

$$(d \mid m) \wedge (d \mid n) \wedge \bigwedge_{t \in \mathbb{N}} [(t \mid m) \wedge (t \mid n) \rightarrow (t \mid d)]$$

Definition

Seien m, n zwei beliebige Zahlen in $\mathbb{N}$.

Die Zahl $k \in \mathbb{N}$ heisst **kleinstes gemeinsames Vielfaches** der Zahlen m und n (kurz: **kgV**), wenn gilt:

$$(m \mid k) \wedge (n \mid k) \wedge \bigwedge_{s \in \mathbb{N}} [(m \mid s) \wedge (n \mid s) \rightarrow (k \mid s)]$$

Zusammenhang von kgV und ggT

Zwischen $\mathrm{ggT}(m,n)$ und $\mathrm{kgV}(m,n)$ zweier Zahlen $m,n \in \mathbb{N}$ besteht der folgende Zusammenhang:

$$\mathrm{ggT}(m,n) \cdot \mathrm{kgV}(m,n) = m \cdot n$$

Berechnung

Der **ggT** zweier Zahlen lässt sich mit Hilfe des Euklidischen Algorithmus berechnen.

Berechnung von $\mathrm{ggT}(m,n)$ mit $m,n \in \mathbb{N}, m \geq n$

$$\begin{array}{rcl} m & = & q_0 \cdot n + r_0 \\ n & = & q_1 \cdot r_0 + r_1 \\ \vdots & & \vdots \\ r_{n-1} & = & q_n \cdot r_n + 0 \end{array}$$

Es gilt: $\mathrm{ggT}(m,n) = r_{n-1}$.

Euklidischer Algorithmus (C - Rekursiv)

```c
unsigned ggt(unsigned m, unsigned n) {
    if (m < n)  return ggt(n, m);
    if (n == 1) return 1;
    if (n == 0) return m;
    return ggt(n, m % n);
}
```

Euklidischer Algorithmus (Haskell)

```haskell
module GGT where

ggt :: Integral a => a -> a -> a
ggt m n
    | m < n      = ggt n m
    | n == 1     = 1
    | n == 0     = m
    | otherwise  = ggt n (m `mod` n)
```