

## Kapitel 0

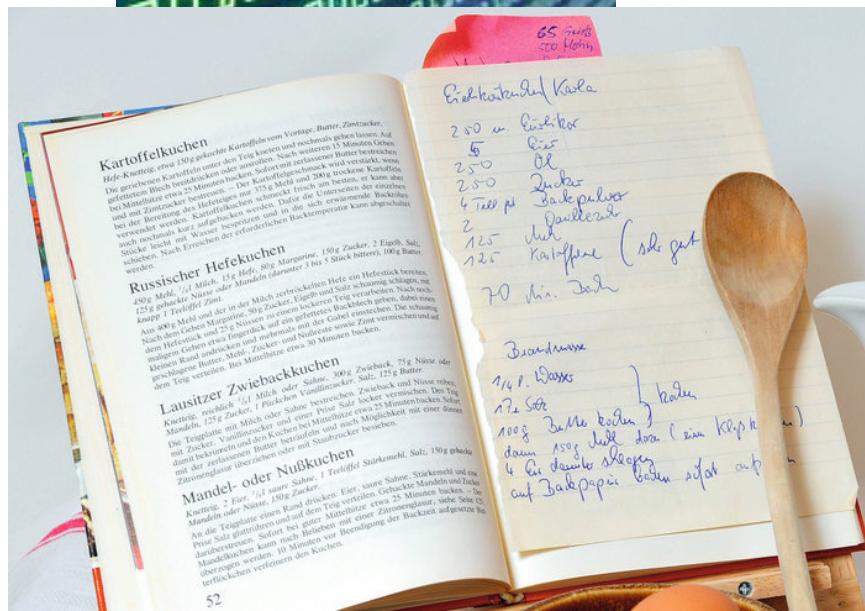
### *Einführung*

0.1

### *Motivation*

Prof. Dr. Robert Preis  
Fachbereich Informatik  
Fachhochschule Dortmund  
[Robert.Preis@fh-dortmund.de](mailto:Robert.Preis@fh-dortmund.de)

# Was ist Informatik ?



Ist das wirklich alles Informatik ?

# Informatik ist ...

In-for-ma-tion + Au-to-ma-tik

Aber was genau ist *Information* ?

Aber was genau ist *Automatik* (Automatisierung) ?

Bitte verwechseln Sie den Begriff „*Informatik*“ nicht mit z.B.

- *Datenverarbeitung*
- *Informationsverarbeitung*
- *IT (Informationstechnik)*

...das ist alles NICHT DASSELBE wie Informatik!

# Was ist Information ?

z.B. „*Der Kaffee kostet 50cent*“.

Eine Information ist eigentlich so etwas wie ein Wort oder ein Satz.

Eine Information kann auch aus mehreren Objekten bestehen:

(50c), (20c,20c,10c), (20c,10c,10c,10c), (10c,10c,10c,10c,10c)

...dies entspricht allen Zahlungsmöglichkeiten mit mind. 10c-Münzen!

Eigentlich ist eine Information nur eine Menge von Objekten, d.h. z.B.

- {„*Der Kaffee kostet 50cent*“} (1 Element)
- {(50c),(20c,20c,10c),(20c,10c,10c,10c),(10c,10c,10c,10c,10c)} (4 Elem.)
- {*Aal, Aas, ab, Abbildung, abblitzen, ABC, Abend, Abenteuer, ...*}  
(Alle Wörter im Duden, ziemlich viele Elemente)
- {2,4,6,8,10,12,14,...}  
(alle natürlichen geraden Zahlen, unendlich viele Elemente)

*Information* und *Menge* ist dasselbe. Wir nennen es *formale Sprache*.

# Ist ein Element (Wort) in einer Menge (formalen Sprache) ?

*Gehört die 17 zu der Menge  $M=\{3,18,3,9,17,5,4\}$ ?*

*d.h.  $17 \in \{3,18,3,9,17,5,4\}$ ?*

...einfach zu lösen, weil es nur eine Aufzählung ist...

*Ist  $18.374.019.283.947$  eine gerade Zahl?*

*d.h.  $18.374.019.283.947 \in \{2,4,6,8,10,\dots\}$ ?*

*Woher wissen Sie das? Haben Sie diese Zahl jemals in Ihrem Leben gesehen ?*

...einfach zu lösen, weil es eine einfache Struktur ist.

*Ist  $18.374.019.283.947$  eine Primzahl?*

*d.h.  $18.374.019.283.947 \in \{\text{alle Primzahlen}\}$ ?*

*Sie sehen die Zahl heute zum ersten Mal !*

...schwieriger zu lösen, weil man das umständlicher prüfen muss.

*Aber was machen wir, wenn es schwieriger wird?*

# Was ist Automatik (Automatisierung)?

Immer wenn es mit Informationen schwierig wird machen wir:

1. **Wir schreiben ein Programm** (Code, Verfahren, Ablaufplan, Rezept,...)  
*...hier sind wir aktiv und versuchen alles so zu machen, dass es später läuft...*
2. **Wir lassen das Programm auf einem Computer laufen** (durch einen Studenten ausführen, durch einen Arbeiter ausführen, durch eine Hausfrau kochen, ...)  
*...hier schauen wir nur zu und hoffen, dass alles gut geht...*

**Die Kunst der Automatisierung ist es,  
etwas so zu schaffen,  
dass es nachher auch ohne uns  
fehlerfrei funktioniert !**

*...dafür müssen wir sehr penibel und sehr genau arbeiten !*

# Was ist dann genau Informatik?

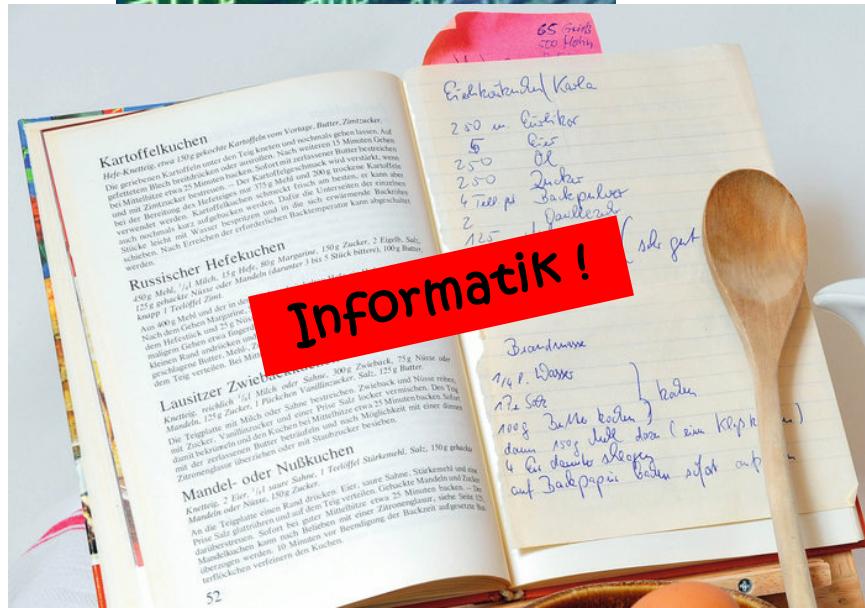
Wikipedia sagt:

**Informatik** (englisch: *computer science* oder *computing science*) ist die

- |                                     |                           |
|-------------------------------------|---------------------------|
| „Wissenschaft                       | (Nicht tun, sondern wie?) |
| von der systematischen Verarbeitung | (Automatik)               |
| von Informationen,                  | (Information)             |
| besonders der                       |                           |
| automatischen Verarbeitung          |                           |
| mit Hilfe von Digitalrechnern“.     | (Push durch Computer)     |

Historisch hat sich die Informatik einerseits als Formalwissenschaft aus der Mathematik entwickelt, andererseits als Ingenieursdisziplin aus dem praktischen Bedarf nach einer schnellen und insbesondere automatischen Ausführung von Berechnungen.

# Was ist nun wirklich Informatik ?



# Was ist Informatik ?

Grundlegende Konzepte und Methoden zur Lösung konkreter **informatischer Probleme** (z.B. Entwicklung von Datenstrukturen oder von Programmiersprachen, Softwaretechnik als Teilgebiet)

Anwendung informatischer Methoden in **informatikfremden Gebieten**, wie Biologie, Betriebswirtschaft oder Medizin

Angewandte Informatik

Praktische Informatik

Technische Informatik

Theoretische Informatik

Abstraktion, Modellbildung und grundlegende Fragestellungen, die mit der Struktur, Verarbeitung, Übertragung und Wiedergabe von Informationen in Zusammenhang stehen

Architektur, Entwurf, Realisierung, Bewertung, Betrieb von Rechner-, Kommunikations- und eingebetteten Systemen auf der **Ebene der Hardware als auch der systemnahen Software**

Nach Wikipedia

# Und was ist Theoretische Informatik ?

## Wikipedia:

Die **theoretische Informatik** beschäftigt sich mit der Abstraktion, Modellbildung und grundlegenden Fragestellungen, die mit der Struktur, Verarbeitung, Übertragung und Wiedergabe von Informationen in Zusammenhang stehen.

Ihre Inhalte sind

- **Automatentheorie,**
- **Theorie der formalen Sprachen,**
- Berechenbarkeits- und Komplexitätstheorie,
- aber auch Logik und formale Semantik
- sowie die Informations-, Algorithmen- und Datenbanktheorie.



# Inhalt „Theoretische Informatik“

## 0 Einführung (ca. 1 Vorlesung)

Motivation, Organisation

## 1 Formale Sprachen und Grammatiken (ca. 2 Vorlesungen)

Sprachen, Grammatiken, Chomsky Hierarchie

## 2 Reguläre Sprachen (ca. 5-6 Vorlesungen)

Deterministischer Endlicher Automat (DEA),  
Nichtdeterministischer Endlicher Automat (NEA),  $\epsilon$ -NEA,  
Minimierung, Äquivalenz,  
Reguläre Ausdrücke,  
Abschlusseigenschaften, Grenzen

## 3 Kontextfreie Sprachen (ca. 3 Vorlesungen)

Push-Down-Automat (PDA),  
Chomsky Normalform, Wortproblem (CYK),  
Abschlusseigenschaften, Grenzen

## 4 Turingmaschinen und Berechenbarkeit (ca. 1-2 Vorlesungen)

# Kapitel 1: Formale Sprachen und Grammatiken

1. Ist **175.830.894.078.798.408** eine gerade Zahl?
2. Ist **7.748.109.384** eine natürliche Zahl ?
  - Anders: *Ist 7.748.109.384 in der Menge der natürlichen Zahlen ?*
3. Ist „**1749373**“ in der Menge {alle Primzahlen} ?
4. Ist „**gehen**“ in der Menge {alle deutsche Wörter} ?
5. Ist „**Ich heiße Paul.**“ in der Menge {alle deutsche Sätze} ?
6. Ist 

```
public class HelloWorld {  
    public static void main(String[ ] args) {  
        System.out.println("HelloWorld");  
    }  
}
```

ein Java-Programm, d.h. in der Menge {alle gültigen Java-Programme}?
7. Ist **(17-7)\*(27-35+((4+3\*(2+5)))** ein gültiger arithmetischer Ausdruck?
  - Auch diesen Ausdruck haben Sie noch nie vorher gesehen !

# Grammatik

- Ist

(())(((())((())(0))))((())((())(0))((0))))

ein gültiger Klammerausdruck?

- Regeln:
  - gleich viele Klammer-AUF und Klammer-ZU !
  - immer erst Klammer-AUF und danach Klammer-ZU !

Aber wie überprüfe ich das ?

Das ist die Syntax !

Aber wie sieht eine Syntax aus ?

Grammatik für Klammerausdrücke:

- Startsymbol S
- Regeln:
  1.  $S \rightarrow SS$
  2.  $S \rightarrow (S)$
  3.  $S \rightarrow ()$

1. *Alles, was durch die Regeln erzeugt wird ist ein korrekter Klammerausdruck.*
2. *Jeden korrekten Klammerausdruck kann man durch diese Regeln erzeugen.*

*D.h.: Alles, was man durch diese Grammatik erzeugen kann sind*

*EXAKT*

*alle korrekten Klammerausdrücke.*

# Eine Grammatik bei der Arbeit

- Startsymbol S
- Regeln:  $S \rightarrow SS, S \rightarrow (S), S \rightarrow ()$

Eine Ableitung:  $S \rightarrow SS \rightarrow (S)S \rightarrow (SS)S \rightarrow ((S)S) \rightarrow (((S))) \rightarrow (((())())()$

Sind alle Grammatiken so einfach?

Nein, denn z.B. recht kompliziert ist:

- Startsymbol S
- Regeln:  $S \rightarrow ACaB, Ca \rightarrow aaC, CB \rightarrow DB, aD \rightarrow Da, AD \rightarrow AC, CB \rightarrow E, aE \rightarrow Ea, AE \rightarrow \varepsilon$

## Einfache Grammatik:

Links: nur ein Zeichen (S)

Rechts: wenige Zeichen (SS)

## Komplizierte Grammatik:

mehrere Zeichen (Ca)

viele Zeichen (ACaB)

*Je nachdem, was man bei Grammatiken zulässt  
kann man mehr oder weniger komplizierte Mengen beschreiben.*

# Einfache und komplexe Grammatiken

Die Information/Sprache/Grammatik ist... Ich brauche...

- |                        |                                  |
|------------------------|----------------------------------|
| 1. <i>sehr einfach</i> | <i>nur ein Blatt Papier!</i>     |
| 2. <i>einfach</i>      | <i>nur einen Taschenrechner!</i> |
| 3. <i>komplex</i>      | <i>nur einen PC!</i>             |
| 4. <i>sehr komplex</i> | <i>nur einen Superrechner!</i>   |

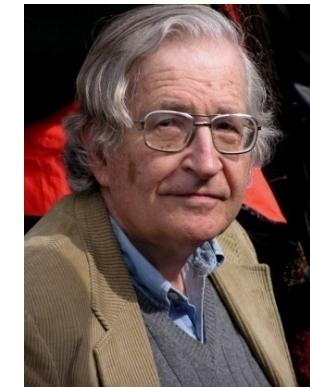
D.h.:

*Einfache Informationen benötigen auch nur einfache Automaten!*

*Komplizierte Informationen benötigen komplexe Automaten!*

***Es ist die Aufgabe eines Informatikers, zwischen einfachen und komplizierten Informationen (Problemen) zu unterscheiden, um die notwendigen Ressourcen für die Problemlösung zu ermitteln!***

# Die Chomsky-Hierarchie zur Klassifizierung von Grammatiken und Maschinen



Avram Noam Chomsky (geb. 7.12.1928 in Philadelphia, USA)

Professor für Linguistik am

Massachusetts Institute of Technology (MIT)

→ [http://de.wikipedia.org/wiki/Noam\\_Chomsky](http://de.wikipedia.org/wiki/Noam_Chomsky)

Typ	Typ-3 ⊂	Typ-2 ⊂	Typ-1 ⊂	Typ-0
Grammatik	$X \rightarrow aY$	$X \rightarrow aYXbY$	$aXb \rightarrow aYb$	$a^x b^y c^z$
Sprache	dieses Semester, Kapitel 2	dieses Semester, Kapitel 3	kontextsensitiv	Einführung dieses Semester, Kapitel 4
Beispiel	Endlicher Automat	Nichtdet. Kellerautomat	$a^n b^n c^n$	{
Maschine			Linear beschränkter Automat	Turing-maschine
Typisches Problem	Gerade Zahlen	Gültiger Klammerausdruck		Java-Programm

# Palindrom

**Studie: Neun von zehn Schülern schreiben "Rentner" verkehrt herum**



Erlangen (dpo) - Eine Umfrage des Goethe-Instituts Erlangen brachte Erschreckendes zu Tage: Nahezu 90 Prozent der Schüler in Deutschland schreiben das Wort "Rentner" aus bisher ungeklärten Gründen rückwärts. Auch Begriffe wie "Neffen", "Radar" oder "Reliefpfeiler" bereiten den meisten große Schwierigkeiten. Nur bei etwa einem Zehntel der Schüler konnte eine korrekte Schreibweise festgestellt werden. Betroffen seien alle Schulformen. mehr...

Quelle:  
[www.der-postillion.com](http://www.der-postillion.com)

# Kapitel 2: Reguläre Sprachen

## Automaten für einfache Grammatiken

Für jede Grammatik können wir einen Automaten bauen, der entscheidet, ob ein Wort durch eine Grammatik erzeugt werden kann oder nicht.

**Menge der Zahlen, die durch 3 teilbar sind:**

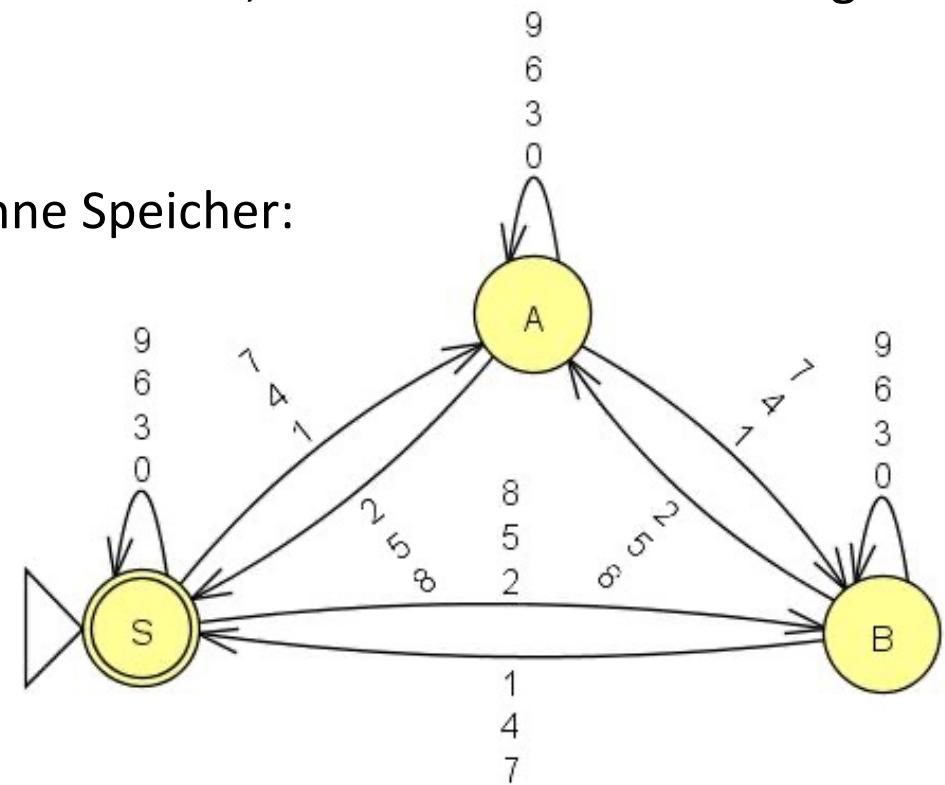
Idee: Wir gehen von links nach rechts und merken uns, welchen Rest die bisherige Summe (geteilt durch 3) hat!

Grammatik:

$$\begin{aligned} S &\rightarrow 0S \mid 1A \mid 2B \mid 3S \mid 4A \mid \\ &5B \mid 6S \mid 7A \mid 8B \mid 9S \mid \epsilon \\ A &\rightarrow 0A \mid 1B \mid 2S \mid 3A \mid 4B \mid \\ &5S \mid 6A \mid 7B \mid 8S \mid 9A \\ B &\rightarrow 0B \mid 1S \mid 2A \mid 3B \mid 4S \mid \\ &5A \mid 6B \mid 7S \mid 8A \mid 9B \end{aligned}$$

*Wir müssen uns 1 von 3 Dingen merken!  
D.h. wir haben 3 Zustände!*

Automat ohne Speicher:



# Kapitel 3: Kontextfreie Sprachen

## Automaten für kompliziertere Grammatiken

Menge aller korrekten Klammerausdrücke:

((())(((())((())( ))))((())((())(( ))))

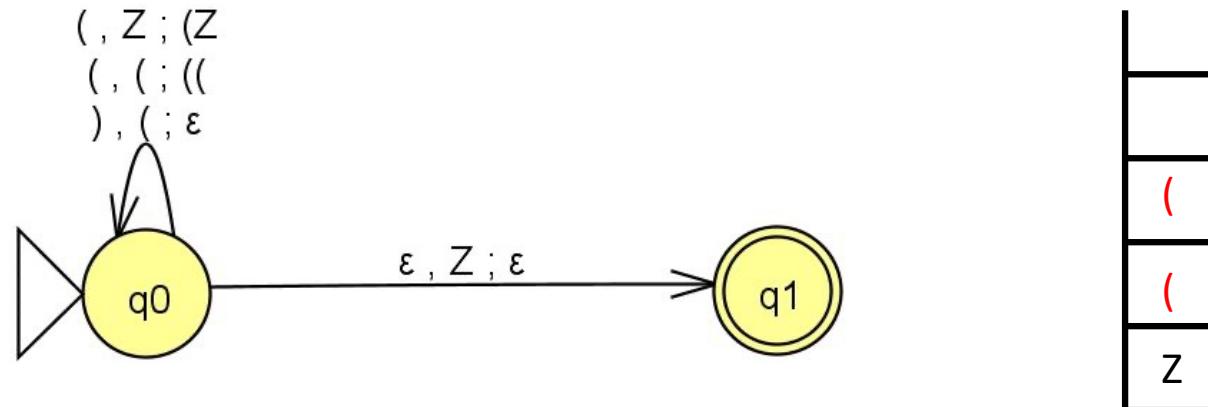
Grammatik:  $S \rightarrow SS, S \rightarrow (S), S \rightarrow ()$

Was müssen wir uns merken, wenn wir von links nach rechts durchgehen?

Die bisherige Anzahl der noch nicht geschlossenen „(-Zeichen!

*D.h. die Zahl, die wir uns merken müssen, kann beliebig groß werden.  
Wir brauchen einen unendlich großen Speicher, in dem wir sie uns merken!*

Automat mit einfachem Keller-Speicher (LIFO – Last In, First Out):



# Kapitel 4: Turingmaschinen

## Automaten für komplexe Grammatiken

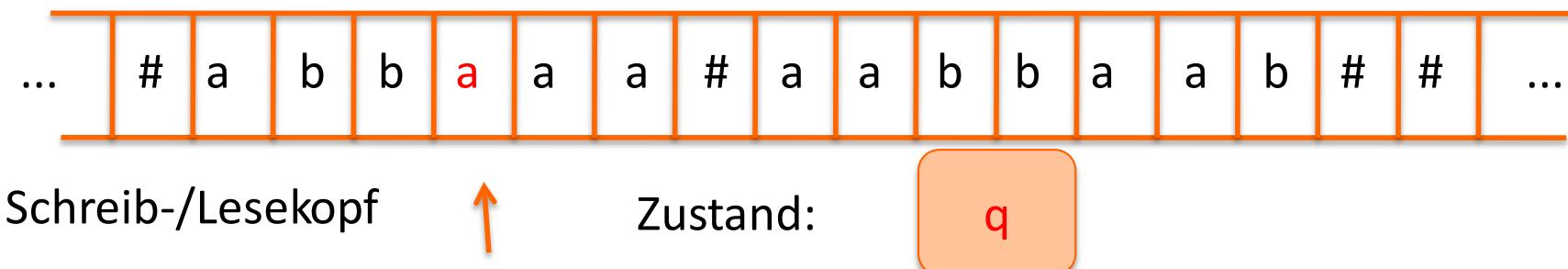
Einige Grammatiken sind so kompliziert, dass ein einfacher Speicher nicht ausreicht. Wir brauchen einen flexiblen Speicher.

**Menge aller Wörter aus a's, wobei die Anzahl eine Zweierpotenz ist.**

Grammatik:

$S \rightarrow ACaB, Ca \rightarrow aaC, CB \rightarrow DB, aD \rightarrow Da, AD \rightarrow AC, CB \rightarrow E, aE \rightarrow Ea, AE \rightarrow \epsilon$

Eine Turingmaschine hat ein unendliches Speicherband und einen flexiblen Zugriff:



*...eine Turingmaschine kann alles berechnen,  
was ein Computer (elektronischer Rechner) kann...*

# Fundamentale Fragen der Theoretischen Informatik

*Was ist ein Problem und wie beschreibe ich es formal korrekt?*

- Wie beschreibe ich die Eingabe und Ausgabe eines Problems?

*Welche Maschine/Computer brauche ich, um ein Problem zu lösen?*

- Waschmaschine, Kaffeemaschine, Cola-Automat, Elektronischer Rechner

*Wie schwer ist es, das Problem zu lösen?*

- Laufzeit, Speicherbedarf?

*Damit ist doch alles gesagt...oder ?*

Eine wichtige Fragestellung fehlt aber noch (kommt erst im Master...):

*Ist das Problem überhaupt lösbar?*

# Unlösbare Probleme ?

Sage ich bei dem Satz

*„Ich sage jetzt nicht die Wahrheit.“*

die Wahrheit oder nicht ?

**Behauptung: Ich sage die Wahrheit!**

Dann würde der Satz ja stimmen.

D.h. ich würde nicht die Wahrheit sagen...das ist ein Widerspruch!

**Behauptung: Ich sage nicht die Wahrheit!**

Dann würde ich ja lügen und der Satz ja nicht stimmen.

Dann würde ja das Gegenteil „Ich sage jetzt die Wahrheit“ stimmen.

Das ist aber ein Widerspruch zur Behauptung!

**Keine der beiden Behauptungen ist wahr !**

# Gibt es unlösbare Probleme?

*In Dortmund wohnt ein Barbier der genau diejenigen männlichen Einwohner von Dortmund rasiert, die sich nicht selbst rasieren.*

## **PROBLEM: Rasiert sich der Barbier selbst oder nicht ?**

**Behauptung:** Der Barbier rasiert sich selbst!

Geht nicht, weil er nur die Gruppe von Männern rasiert, die sich nicht selbst rasieren, und zu dieser Gruppe gehört er nicht!

**Behauptung:** Der Barbier rasiert sich nicht selbst!

Geht nicht, denn dann gehört er zu der Gruppe der Menschen, die sich nicht selbst rasieren...und genau diese Gruppe wird vom ihm rasiert und somit müsste er sich selbst rasieren!

**Keine der beiden Behauptungen ist wahr !**

**Deshalb kann es keinen solchen Barbier geben !**

# Problem der Endlosschleife bei P(x)

**Frage:** Hält  $P(x)$  (Programm  $P$  bei der Eingabe  $x$ ) oder gibt es eine Endlosschleife?

**Annahme:** Es gibt ein Programm „**Halten( $P,x$ )**“, das entscheidet ob  $P(x)$  hält.

Dann gibt es auch folgendes Programm **Widerspruch( $P$ )**:

Falls  $\text{Halten}(P,P)$  „ja“ liefert, gehe in Endlosschleife, sonst halte an.

**Problem: Hält Widerspruch(Widerspruch) ?**

**Behauptung:** **Widerspruch(Widerspruch)** hält!

Dann würde laut Annahme  $\text{Halten}(W.,W.)$ , „ja“ liefern, d.h.  $W.(W.)$  würde in eine Endlosschleife gehen ... Widerspruch zur Behauptung !

**Behauptung:** **Widerspruch(Widerspruch)** hält nicht!

Dann würde laut Annahme  $\text{Halten}(W.,W.)$  „nein“ liefern, d.h.  $W.(W.)$  würde anhalten ... Widerspruch zur Behauptung !

**Keine der beiden Behauptungen ist wahr !**

**Also kann es kein Programm **Widerspruch( $P$ )** geben !**

**Also kann es kein Programm **Halten( $P,x$ )** geben !**

# Problemklassen (im Master)

1. Unlösbarer Probleme
  - Z.B. Endlosschleife
2. Lösbarer Probleme, aber nicht effizient
3. Effizient lösbarer Probleme

Fundamentale Fragen in der  
Theoretischen Informatik:

**Was können Computer *überhaupt* lösen?**

**Berechenbarkeitstheorie**

**Was können Computer *effizient* lösen?**

**Komplexitätstheorie**

Wie bereits erwähnt:

***Es ist die Aufgabe eines Informatikers, herauszufinden, ob ein Problem lösbar ist und falls ja, wie aufwendig die Lösung ist.***

