

Theoretische Informatik I

Prof. Dr. Carsten Lutz
AG Theorie der künstlichen Intelligenz
MZH Raum 3090

Homepage der Vorlesung:

<http://www.informatik.uni-bremen.de/tdki/lehre/ws11/theoinf>



Organisatorisches

Vorlesung: Mo 14:00 – 16:00 NW1 H1

Hauptsächlich Folien,
ausgesuchte Beispiele + Beweise an der Tafel

Skript:

- Verfügbar auf Webseite
- Teile I + II: VL Theoretische Informatik I
- Teile III + IV: VL Theoretische Informatik II
- Zusätzliches Material in der VL wird angekündigt

Mitschreiben!



Literatur

- Skript zur Vorlesung (Webseite)
- Dexter Kozen, *Automata and Computability*, Springer Verlag 2007
- John Hopcroft, Rajeev Motwani, Jeff Ullmann, *Introduction to Automata Theory, Languages, and Computation* (3rd edition), Addison Wesley, 2006
- Uwe Schöning, *Theoretische Informatik-kurzgefasst*, Spektrum Akademischer Verlag, 2001
- Ingo Wegener, *Theoretische Informatik-Eine algorithmenorientierte Einführung*, Teubner, 1999.



Übungsgruppen

- 7 Gruppen zu unterschiedlichen Terminen, Zuordnung beim ESO-Frühstück
- Beginn kommende Woche
- Jede Woche ein Aufgabenblatt auf VL-Homepage, das in der Übungsgruppe **gemeinsam gelöst** wird
- Jede zweite Woche werden die Aufgaben **abgegeben und korrigiert**
- Die Bearbeitung der Aufgaben erfolgt in **Gruppen von 2-3 Personen**
- In der kommenden Woche muss nichts abgegeben werden.



Scheine / Prüfungen

Prüfungsmodalitäten

- Es gibt 6 gewertete Übungsblätter, **insgesamt müssen 50% der Punkte** erreicht werden
- Note wird über alle Blätter **gemittelt**, geht in Bachelor-Note ein!
- Zusätzlich **Fachgespräch** am Ende des Semesters (Prüfungsleistung, Änderung der Note möglich)
- Übungen gehören zum Vorlesungsstoff!



Theoretische Informatik – Eine kurze Einführung



Theoretische Informatik

Schafft formale und kulturelle Grundlage für die Informatik

Kultur:

- Gemeinsames Grundwissen: welche allgemeinen **Konzepte und Methoden** sind zentral für die Disziplin und “common knowledge”?
- Gemeinsame Sprache: welche **zentralen Begriffe** werden von allen verstanden?

Konkrete Anwendungen und Realisierungen

Praktische Informatik

Technische Informatik

Theoretische Informatik



Theoretische Informatik

Dijkstra: In der Informatik geht es genauso wenig um Computer, wie in der Astronomie um Teleskope

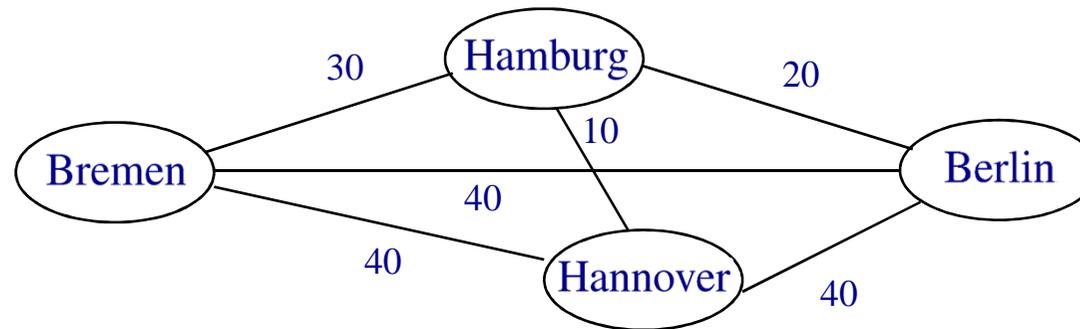
Schwerpunkte:

- **Schaffen von mathematischen Modellen und Abstraktionen**
Was ist die Essenz einer Programmiersprache / eines Computers / einer Anwendung, was Beiwerk?
- **Bereitstellen von Berechnungsmodellen und algorithmischen Techniken**
Was macht einen Computer aus? Wie unterscheidet sich ein PC von einem DNA Computer und einem Quantencomputer?
- **Verständnis der Grenzen der (effizienten) Berechenbarkeit**
Kann ich alles berechnen, was ich beschreiben kann (bei vollständiger Information)? Wie effizient kann ich Dinge berechnen?



Theoretische Informatik – Beispiel 1

Aufgabe: Finde den günstigsten Weg für eine Rundreise



Das Programm ist nicht sehr effizient?

Das liegt nicht an Dir!

Denn:

- unbekannt, ob dieses Problem (**Travelling Salesman**) effizient lösbar
- Frage äquivalent zum **wichtigsten offenen Problem** in der Informatik/Mathematik



Theoretische Informatik – Beispiel 2

Aufgabe: Entwerfe eine Raketensteuerung



Theoretische Informatik – Beispiel 2

Aufgabe: Entwerfe eine Raketensteuerung



Problem: fehlerhafte Konversion einer Fließkommazahl in ganze Zahl

Theoretische Informatik – Beispiel 2

Klassische Methode zum Finden von Bugs:

Testen! (nach Möglichkeit systematisch)

Problem: i.d.R. zu viele mögliche Eingaben, um **alle** zu testen

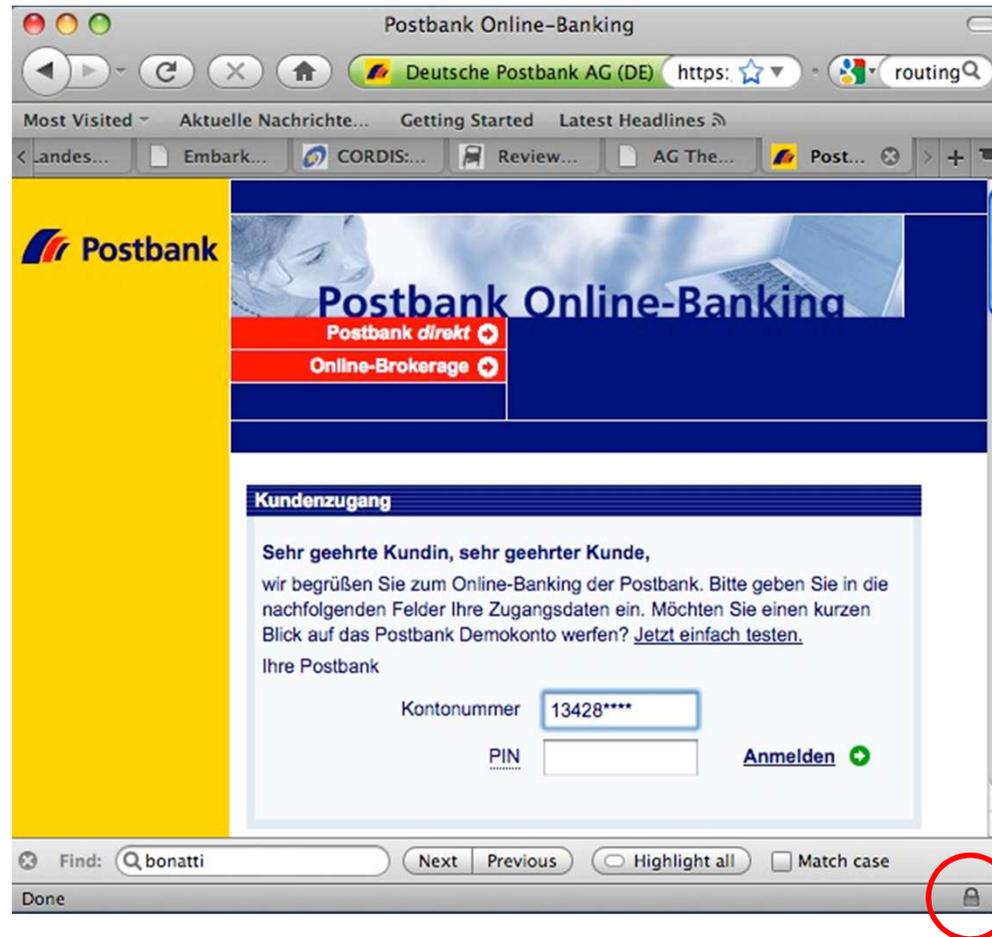
In kritischen Anwendungen viel besser: **Verifikation**

- erlaubt einen **formalen Beweis** der Korrektheit
(automatische Analyse des Programmes, kein Testen)
- basiert auf mathematischen Methoden, insb. Logik
- Teilgebiet der theoretischen Informatik



Theoretische Informatik – Beispiel 3

Aufgabe: Verwende Online-Banking, ohne bestohlen zu werden



Was genau bedeutet das?



Theoretische Informatik – Beispiel 3

Das Schloss bedeutet natürlich **verschlüsselte Übertragung**

Aber es bleiben berechnete Fragen:

- Kann jemand den Schlüssel abfangen?
Sehr leicht sogar, aber das macht nichts
- Kann man ganz sicher sein, dass niemand einen Trick gefunden hat, die Verschlüsselung ohne Schlüssel zu brechen?
Nein, kann man nicht!
- Kann ich der Verschlüsselung vertrauen?
Ja, durchaus!

Diese Fragen werden in der **Kryptographie** studiert.



Theoretische Informatik

Besteht aus **vielen Teilgebieten**:

- Automaten und formale Sprachen - **TheoInf I**
- Komplexitätstheorie und Theorie der Berechenbarkeit - **TheoInf II**
- Verifikation und mathematische Logik
- Kryptographie
- Algorithmentheorie
- Datenbanktheorie
- etc.



Zur Rolle der Mathematik



Mathematik

Ab jetzt keine explodierenden Raketen und Screenshots, sondern: **Mathematik**

Exakte + formale Beschreibungen essentiell in der Theoretischen Informatik

Lasst Euch nicht abschrecken:

- Mit etwas gutem Willen erkennt man leicht, warum die mathematischen Definitionen und Resultate **wichtig für die Informatik** sind
- Mathe ist nicht gleich Mathe:
 - Wir benötigen oft **andere mathematische Methoden**, als Ihr aus der Schule kennt.
 - Wir wollen nicht **rechnen** (=langweilig), sondern Dinge **beweisen** (=verstehen)



Mathematik

Die notwendige Mathematik lernen Sie:

- hier!
- in Mathe I
- durch viel Übung, z.B. in den Übungsgruppen

