“Theorie der Informatik” (CS206)
Organisation, Motivation
mathem. Grundlagen, Beweisformen

27. Februar 2013

Proff Malte Helmert und Christian Tschudin
Departement Mathematik und Informatik, Universität Basel

Willkommen bei CS206

Letztes Jahr (2012) war das Turing-Jahr !

Alan Turing, **23. 6. 1912** – 7. 6. 1954

`http://www.nzz.ch/nachrichten/hintergrund/wissenschaft/turingmaschine_im_rueckwaertsgang_1.14389143.html`

Uebersicht der Sitzung vom 27. Februar 2013

1. Organisatorisches
   - Zeiten
   - Personen
   - Leistungsüberprüfung
   - Textbücher
   - Vorlesungsinhalt
   - Uebungsorganisation
2. Motivation, Lernziele, Einführung in die Logik

Organisation – Zeiten

- Vorlesung:
  Montag 13–15 Uhr
  Mittwoch 15–17 Uhr
  jeweils Bernoullistrasse 16, Seminarraum 205
- Uebungen:
  - 2 Gruppen: Mo 15-17, Mi 17-19
  - Seminarraum 205 (zu überprüfen)
  - Start der Uebungssitzungen: tba
- Anmeldung: → **https://services.unibas.ch/**
  für die Uebungen: → **https://courses.cs.unibas.ch/**

Organisation – Personen

- Prof. Malte Helmert, <malte.helmert@unibas.ch>
  Bernoullistrasse 16, Büro 305
- Prof. Christian Tschudin, <christian.tschudin@unibas.ch>
  Bernoullistrasse 16, Büro 303
- Martin Wehrle, <martin.wehrle@unibas.ch>
- Tutoren:
  Florian Pommerening, Jendrik Seipp, Silvan Sievers

Organisation – Textbuch, Leistungsüberprüfung

- Zwei Bücher als Grundlage und als Ergänzung der Slides:
  – Uwe Schöning: *Logik für Informatiker*
  – Uwe Schöning: *Theoretische Informatik – kurzgefasst*
  beide in 5. Auflage, Spektrum Heidelberg, 2000 und 2008
  (ist bei Karger vorbestellt)
- Leistungsüberprüfung mit vier Elementen:
  – Aktive Teilnahme an Uebungen, inkl vertiefender Vortrag
  – Erreichen mind. 50% der Punkte aus Uebungsaufgaben
  – Klausur am 29. Mai 2013, bestimmt Schlussnote

Organisation – Vorlesungsinhalt FS2013

- Logik (4x2h)
- Automaten und Sprachen (8x2h)
- Berechenbarkeit (3x2h)
- Komplexitätstheorie (5x2h)
- 27. Mai: Fragestunde, Repetition
- 29. Mai: Klausur

Organisation – Inhalt bis nach Ostern 2013

27.2.	Einführung, Motivation, Beispiele von Beweisformen
4.3.	Syntax, Semantik, Inferenz / Aussagenlogik I
6.3.	Aussagenlogik II, Prädikatenlogik I
11.3.	Prädikatenlogik (II)
13.3.	Sprache und Grammatik, endl. Automat
18.3.	NFA, Pumping Lemma, Predictive Parsing
20.3.	Kontextfreie Sprachen, Kellerautomat, Turing-Maschine, Busy Beaver
25.3.	Turing-Maschine II, LOOP-Programme, prim.-rek. Funktionen
27.3.	Forts. LOOP-Programme und prim.-rek. Funktionen, GOTO-Programme
3.4.	μ-rek. Funktionen, Church-Turing-These, berechenbar/entscheidbar

Organisation – Uebungsbetrieb

- Woche N: Ausgabe des Uebungsblatts, jeweils am Montag (erstmals am 4. März 2013)
- Woche N+1: Abgabe (Courses) am Mittwoch (erstmals am 13. März)
- Woche N+2: Rückgabe Korrektur und Besprechung, Mo/Mi (erstmals am 18./20. März)

Der Uebungsbetrieb wird durch Assistent Martin Wehrle organisiert; er wird weitere Details bekanntgeben.
Präsenzübungen am 4. März!

Organisation – Fragen?

Motivation – Die Person Hilbert

- David Hilbert, 1862 –1943
- einflussreicher und universeller Mathematiker
- Beiträge zu Invariantentheorie, Axiomatisierung der Geometrie, Funktionalanalysis
- Mitbegründer einer Beweistheorie
- Unterscheidung zwischen Mathematik und Meta-Mathematik

Motivation – Hilberts 10tes Problem

- Mathematikerkongress 1900:
  David Hilbert stellt 23 wichtige Probleme der Mathematik vor.
- Problem # 10: **Entscheidbarkeit** für Diophantische Gleichungen
- Gegeben: Polynomgleichung $p(x_1, \ldots, x_n) = 0$
  mit ganzzahligen Koeffizienten.
- Frage: *Gibt es* ein Verfahren, das für beliebige solcher Gleichungen entscheiden kann, ob es eine ganzzahlige Lösung $(x_1, \ldots, x_n)$ gibt?
- Antwort (erst 1970 durch Matijasevic): Nein (!)

Motivation (Forts.)

Warum ist das (negativ beantwortete) 10te Hilbert-Problem wichtig?

- Beachte die Fragestellung:
  Es wird nicht nach einem (dem) Lösungsverfahren gefragt,
  nur **ob** es ein Verfahren gibt.
- Schlussfolgerung: Es gibt (in der Mathematik) Fragen, die nicht algorithmisch gelöst werden können.
- Dies schliesst nicht aus, dass eine gegebene Polynomgleichung gelöst werden kann. Aber es gibt kein *allgemeines* Verfahren.
- Später: Es kann wahre Sätze geben, die nicht beweisbar sind.

Motivation – Halteproblem

"Uebersetzung" der Unentscheidbarkeit des 10ten Hilbert-Problem für Computer:

- Programme können Eigenschaften haben, die
  nicht berechnet werden können.
- Berühmtestes Beispiel: das **Halteproblem**

  Gibt es ein Verfahren, das entscheidet, ob ein *beliebiges* Programm nach einer endlichen Anzahl von Berechnungsschritten abbricht?

  Antwort: Nein

Motivation – Halteproblem (Forts.)

Reisserische Variante:
Es kann nicht entscheiden werden, ob ein Programm ein Virus ist.

Virus: Ein Programm, dass den eigenen Code (`virusAction()`) in andere Programme (`ordinaryCode()`) einschleusen kann.

Betrachte:

```
BEGIN
   solve(Problem_X);    // virusAction, part A
   infect();            // virusAction, part B
   ordinaryCode();
END
```

Wenn `solve(Problem_X)` nie abbricht, dann ist es kein Virus.

Lernziele der Vorlesung CS206

Grundkenntnisse Theorie: “Das muss jede/r InformatikerIn wissen”

- Drei Aspekte im Vordergrund:
  – Konzepte der formalen Logik, Kalkül
  – Automatentheorie (endl. Automaten und Turingmaschine)
  – Komplexität
- Scheinbar komplexe Probleme können sich als einfach entpuppen, d.h. können reduziert werden.
- Scheinbar einfache Probleme können sich als unlösbar entpuppen. Oder “praktisch” unlösbar sein (ineffizient).

Formalisierung der Mathematik

Versuch (beginnend kurz vor Jahrhunderwende 1900) der Formalisierung

- Mechanisches Beweisen:
  – formales “Spiel mit Buchstaben”
  – losgelöst von einer Bedeutung
- Aufbau beginnt mit elementaren *Aussagen*,
  die wahr oder falsch sein können (Aussagenlogik)
- Später kommen Prädikate und Quantoren ($\exists$, $\forall$) dazu
- Axiomatik, insbesondere der natürlichen Zahlen (Peano)

Beispiel Peano-Axiome 1889 (Theorie der natürlichen Zahlen)

1. “1” ist eine natürliche Zahl
2. Jede natürliche Zahl ist zu sich selbst gleich (Reflektivität)
3. Für alle natZahlen a und b gilt: a=b dann und nur dann wenn b=a (Symmetrie)
4. Für alle natZahlen a, b und c gilt: wenn a=b und b=c dann a=c (Transitivität)
5. Falls a=b und b ist eine natZahl, dann ist a eine natZahl
6. Falls a eine natZahl ist, ist succ(a) auch eine natZahl
   (succ(x) ist die Nachfolgerfunktion (successor))
7. Falls a und b natZahlen, dann gilt: a=b dann und nur dann wenn succ(a) = succ(b)
8. Falls a eine natZahl, dann ist succ(a) nicht gleich 1
9. Für jede Menge K die “1” enthält, und für die gilt, dass für jede Zahl x aus K
   auch succ(x) in K ist, dann ist jede natZahl in K.

Die Elemente von N sind: 1, succ(1), succ(succ(1)), succ(succ(succ(1))), ...

Bewertung der Peano-Axiome

- Axiome “fassen” unser intuitives Verständnis der natürlichen Zahlen
- Könnten aus den Axiomen widersprüchliche Aussagen zu natürlichen Zahlen abgeleitet werden? (**Konsistenz**)
- Kann man beweisen, dass keine Inkonsistenzen entstehen können?

Gödel 1931: Konsistenzbeweis für Peano-Arithmetik kann nicht mit den Mitteln der Peano-Arithmetik selbst erbracht werden.

Mathematische Grundkonzepte (relevant für Informatik)

- Menge von Objekten (ungeordnet, keine Duplikate):
  - $\{4, 7, 12\}$
  - $\{x | x\ ist\ natuerliche\ Zahl, x\ ist\ gerade\}$
  - leere Menge: $\emptyset$, oder $\{\}$ geschrieben
- “ist-element-von” mit $\in$ oder $\notin$ angegeben:
  - 4 $\in$ $\{4, 7, 12\}$, 5 $\notin$ $\{4, 7, 12\}$
- Teilmenge $\subseteq$ und echte Teilmenge $\subset$:
  - $\{12, 4, 7\} \subseteq \{4, 7, 12\}$, und $\{4, 7\} \subset \{4, 7, 12\}$
- Schnittmenge $\cap$ und Vereinigung $\cup$:
  - $\{4, 7\} \cap \{4, 12\} = \{4\}$ und $\{7, 12\} \cup \{4, 7\} = \{4, 7, 12\}$

Mathematische Grundkonzepte (Forts I)

- Sequenz ist eine geordnete Liste von Objekten:
  - $< 4, 7, 12 >$ ist nicht das gleiche wie $< 12, 7, 4 >$
  - $< 4, 4 >$ ist nicht das gleiche wie $< 4 >$
  - Konvention: oft normale Klammern $(\ldots)$ statt $< \ldots >$ gebraucht
- Endliche vs unendliche Sequenz:
  - Unendliche Sequenz: die Ziffern von π, die Ziffern von $1/3$
  - eine endliche Sequenz wird *Tupel* genannt, auch k-Tupel (Sequenz mit k Elementen)
  - ein 2-Tupel ist ein *Paar*

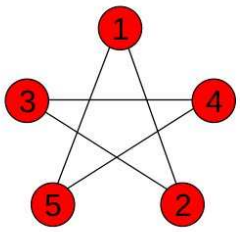
Mathematische Grundkonzepte (Forts II)

- Mächtigkeit, Länge:
  - $|X|$ für Mächtigkeit oder Länge von X
  - Sei $A = \{4, 7, 12\}$, dann ist $|A|$ die Anzahl Elemente von A (d.h. 3)
  - Sei $A =< 2, 3, 5 >$, dann ist $|A|$ die Länge der Sequenz A (d.h. 3)
- Potenzmenge (power set):
  - Potenzmenge $\mathcal{P}(A)$: Menge aller Teilmengen von A inkl A
  - Sei $A = \{0, 1\}$, dann ist $\mathcal{P}(A) = \{\{\}, \{0\}, \{1\}, \{0, 1\}\}$
  - Grund für Name: Sei $|A| = n$, dann ist $|\mathcal{P}(A)| = 2^n$
- Kartesisches Produkt, auch Kreuzprodukt genannt:
  - Sei $A = \{a, b\}$ und $B = \{1, 2, 3\}$, dann $A \times B = \{< a, 1 >, < a, 2 >, < a, 3 >, < b, 1 >, < b, 2 >, < b, 3 >\}$

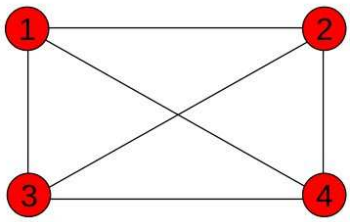
Mathematische Grundkonzepte (Forts III)

- Graph $G=(V,E)$ (vertices and edges)

G_1= ({1,2,3,4,5}, {{1,2}, {2,3), {3,4}, {4,5}, {5,1}})

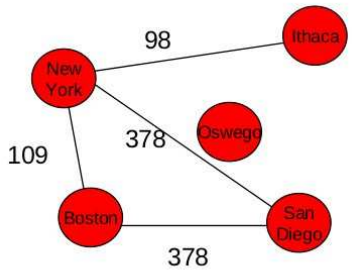


G_2= ({1,2,3,4}, {{1,2}, {1,3}, {1,4}, {2,3}, {2,4}, {3,4}})

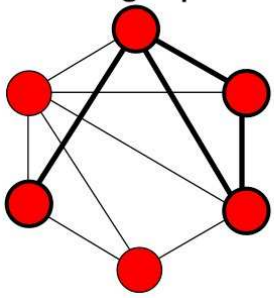


Mathematische Grundkonzepte (Forts IV)

- Labelled, weighted

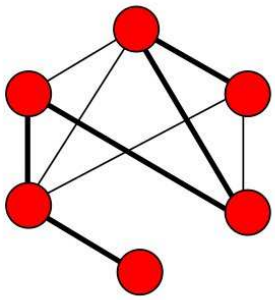


- Subgraph, induced subgraph

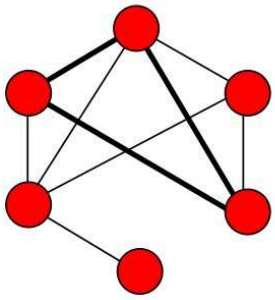


Mathematische Grundkonzepte (Forts V)

- (Simple) path

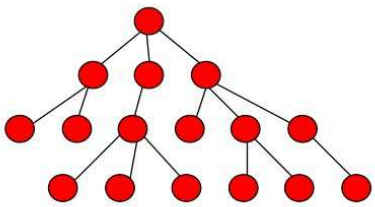


- (Simple) cycle

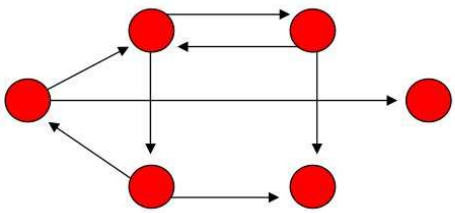


Mathematische Grundkonzepte (Forts VI)

- Tree



- Directed graph



Mathematische Grundkonzepte (Forts VII)

Zeichenketten (Wörter) und Sprachen

- Alphabet: Menge von Symbolen. Beispiel: $\Sigma = \{a, b, c\}$
- Wort (Zeichenkette, String) = endliche Sequenz von Symbolen über einem Alphabet. Beispiel: $w = aabbabcca$
- Länge $|w|$ eines Worts w = Länge der Sequenz = Anzahl Symbole in w
- Leeres Wort = ϵ
- $aabb$ ist ein Teilwort (subword) von $aaabbbbccc$
- Seien x, y Wörter, dann ist xy die Konkatenation
- Wiederholung $x^k = x \dots x$ (zum Beispiel $y^3 = yyy$)
- **"Sprache" = Menge von Wörtern (über einem Alphabet Σ)**

Meta-Mathematik

Meta-Mathematik bezeichnet die Untersuchung der Mathematik mit mathematischen Methoden.

- Formale Logik und Aussagelogik sind Teil der Meta-Mathematik
- Beispiel für meta-mathematische Aussage (siehe weiter oben): Gödel 1931 – Ein Konsistenzbeweis für die Peano-Arithmetik kann nicht mit den Mitteln der Peano-Arithmetik selbst erbracht werden.

Im Folgenden besuchen wir ein paar mathematische Beweisformen.

Mathematische Beweise

Liste von Möglichkeiten (Beispiele werden folgen):

- Direkter Beweis
- Beweis durch Konstruktion oder Gegenbeispiel
- Beweis durch Widerspruch
  (indirekter Beweis, reduction ad absurdum)
- Induktionsbeweis

Beweise sind für Menschen gedacht (es muss “überzeugend genug” sein); die Mechanisierung ist ein anderes Thema (automatische Theorem-Beweiser).

Beweise – direkt

Strategie: Ausgehend von Annahmen leite man Folgerungen ab bis man zum gewünschten Schluss kommt.

Beispiel: Seien a, b, c ganze Zahlen, $|$ “ist-Teiler-von”: Falls $a|b$ und $b|c$ dann $a|c$.

Beweis:

1. Aus $a|b$ erhalten wir (1): $\exists k_1 \; mit \; b = k_1 a$
2. Aus $b|c$ erhalten wir (2): $\exists k_2 \; mit \; c = k_2 b$
3. Aus (1) und (2) und Substitution erhalten wir (3): $\exists k_1, k_2 \; mit \; c = k_2 k_1 a$
4. Aus (3) erhalten wir (4): $\exists k \; mit \; c = ka$ (nämlich $k = k_1 k_2$)
5. Aus (4) erhalten wir die gesuchte Aussage $a|c$

Beweise – durch Konstruktion (I)

Ziel: Wir wollen beweisen, dass ein bestimmtes Objekt existiert.

Strategie: Zeige, wie dieses Objekt gefunden oder konstruiert wird.

Beispiel für Graphen:

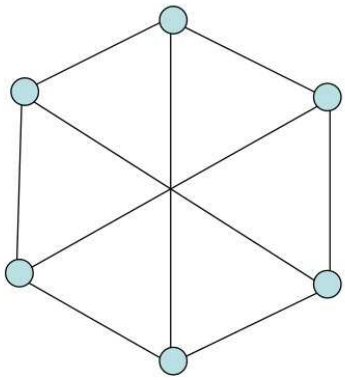
- Definition: Grad eines Knotens v = Anzahl Kanten, die von v ausgehen
- Definition: Ein Graph ist k-regulär wenn alle Knoten den Grad k haben.
- Theorem:
  Für alle geraden Zahlen $n > 2$ existiert ein 3-regulärer Graph mit n Knoten.

(Beweis siehe nächste Seite)

Beweise – durch Konstruktion (II)

G=(V,E) with

- V = {0,1,...,n-1} and
- E = {{i,i+1} | for $0 \leq i \leq n-2$ } $\cup$ {{n-1,0}} $\cup$ {{i, i+n/2} | $0 \leq i \leq n/2-1$ }}
- → every vertex has exactly three neighbours:
  - its predecessor in the cycle 0, 1, 2, …, n-1, 0
  - its successor in the cycle
  - its "mirror image" n/2 positions before/ahead in the cycle



Beweise – durch Widerspruch (I)

Beispiel eines Theorems, das mit Beweis durch Widerspruch bewiesen werden kann:

Theorem: $\sqrt{2}$ ist irrational

Strategie:

- Nehme an, dass das Theorem falsch sei
  d.h. $\sqrt{2}$ kann als Bruch (rationale Zahl) dargestellt werden
- Zeige, dass diese Annahme zu einem Widerspruch führt ...
- ... weshalb das Theorem wahr sein muss.

Beweise – durch Widerspruch (II)

Theorem: $\sqrt{2}$ ist irrational

Beweis: Nehme an, dass das Theorem falsch ist. Dann:

- $\sqrt{2} = b/a$, wobei a, b ganze Zahlen, Bruch gekürzt
- $2 = b^2/a^2$
- $2a^2 = b^2$, d.h. b^2 ist gerade, d.h. b ist gerade.
  Schreiben wir also $b = 2c$
- $2a^2 = 4c^2$, nun Division durch 2
- $a^2 = 2c^2$, d.h. a^2 ist gerade, d.h. a ist gerade.
- Widerspruch:
  die Annahme war, dass b und a keine gemeinsamen Teiler mehr haben.

Beweise – durch Induktion (I)

Ziel: Wir wollen eine Aussage $S(X)$ machen für eine Familie von Objekten (z.B. ganze Zahlen, Datenstrukturen)

Strategie:

- Basis (Verankerung): beweise direkt, dass S(X) für ein kleines X gilt
- Induktionsschritt: Nehme an, dass $S(Y)$ für $Y < X$ gilt, und beweise $S(X)$ unter dieser Annahme.

Anwendungsfelder:
– natürliche Zahlen
– induktiv definierte Objekte (Datenstrukturen, Mengen)

Beweise – durch Induktion (II)

Beispiel einer induktiv definierten Datenstruktur: binäre Bäume

Definition:

- Basis: ein einzelner Knoten v ist ein binärer Baum und ist die Wurzel dieses Baumes.
- Induktion: Seien T_1 und T_2 binäre Bäume mit den entsprechenden Wurzeln, dann ist der folgende Graph auch ein binärer Baum:
  – erzeuge einen neuen Knoten N als Wurzel
  – kopiere die Bäume T_1 und T_2
  – lege eine Kante von N nach T_1, und von N nach T_2

Beweise – durch Induktion (III)

Theorem: Ein binärer Baum mit n Blättern hat $2n - 1$ Knoten

Basis (Verankerung): Aussage stimmt für Baum mit 1 Knoten: $2 \times 1 - 1 = 1$

Induktion:

- Nehme an, die Aussage $S(T)$ gilt für Bäume kleiner als T, insbesondere seine beiden Teilbäume U und V.
- Dann besteht T aus der Wurzel und U und V
- Seien u und v die Anzahl Blätter von U und V: T hat t Blätter mit $t = u + v$
- Die Induktionsannahme war, dass U und V jeweils $2u - 1$ und $2v - 1$ Knoten haben.
- Somit hat T insgesamt $1 + (2u - 1) + (2v - 1) = 2(u + v) - 1 = 2t - 1$ Knoten.

Streit um Beweisformen

In der Mathematik waren nicht immer alle Beweisformen akzeptiert:

- Reine “Existenz-Beweise”, die das Vorhandensein eines Elements behaupten ohne dessen Konstruktion zeigen zu können, wurden angefochten.
- Ab 1908 kam es zum Streit zwischen Brouwer (“intuitive Logik”) und Hilbert (“Formalist”) um die Anwendung des Prinzips

  “Satz vom ausgeschlossenen Dritten”

  auf unendliche Mengen, die sog. Grundlagenkrise der Mathematik.
- Das “Hilbert-Programm” der Formalisierung der Mathematik wurde durch die Ergebnisse von Turing relativiert (Unentscheidbarkeit)

Heute “versöhnlich” gelöst, siehe Wikipedia zu “Grundlagenkrise der Mathematik”.

Aussagenlogik

Aussage = Satz, der entweder wahr oder falsch sein kann.

- Beispiele:
  “5 ist gerade”, “*Internationalization* hat 20 Buchstaben” (i18n)

Logische Konnektoren:

$A \wedge B$	“und”	wahr wenn A und B wahr sind
$A \vee B$	“oder”	wahr wenn mind einer der Terme A und B wahr ist
$\neg A$	Negation	wahr wenn A falsch ist
$A \rightarrow B$	“falls dann”	wenn A wahr ist, muss B auch wahr sein
$A \leftrightarrow$ B	“genau dann wenn”	A und B sind gleichzeitig wahr oder falsch