Logik in der Informatik

Nicole Schweikardt

WS 2009 / 10

Heute:

- Einführung ins Thema
- Organisatorisches
- Start mit Kapitel 1

0. Einführung ins Thema

Logik

- … und Mathematik
- … und Informatik

Logik als "Fundament der Mathematik"

Hilberts Programm (ca 1900-1928, initiiert von David Hilbert)

Ziel: formale Grundlegung der Mathematik

Mittel: mathematische Logik:

- mathematische Strukturen als logische Strukturen
- mathematische Aussagen als logische Formeln
- math. Beweise durch "syntaktisches Schließen" (Symbolmanipulation: Axiome & Schlussregeln)

Ansatz: Rückführung der Mathematik auf Arithmetik + Mengenlehre

Hilberts Programm:

2 Kernfragen:

1) Kann jede math. Aussage durch math. Schließen bewiesen oder widerlegt werden?

2) Gibt es ein Verfahren, das zu jeder math. Aussage entscheidet, ob sie wahr oder falsch ist?

Beachte: Es gilt 1) $\Rightarrow$ 2)

Eine äquivalente Formulierung des Entscheidungsproblems 2) ist das Allgemeingültigkeitsproblem der Logik erster Stufe:

Allgemeingültigkeitsproblem der Logik erster Stufe:

Eingabe: Eine Formel φ

Frage: Gilt für alle zu φ passenden Interpretationen $\mathcal{I}$: $\mathcal{I}$ erfüllt φ?

Beispiel: Sei φ die Formel

$$\forall x\ \exists y \exists z\ y > x \wedge z = y+2 \wedge \forall u \forall v \big((u \cdot v = y \rightarrow (u=1 \vee v=1)) \wedge (u \cdot v = z \rightarrow (u=1 \vee v=1)) \big)$$

Beachte: φ besagt: "es gibt unendlich viele Primzahlzwillinge"

... Der Nachweis, dass die Formel φ in der Arithmetik der nat. Zahlen erfüllt ist, würde also ein berühmtes offenes Problem aus der Zahlentheorie lösen (... nämlich die Frage, ob es unendlich viele Primzahlzwillinge gibt).

2 wichtige Aspekte der logischen Fundierung.

1) Präzisierung von Aussagen ("Logik für Penible")
2) Automatisierung des Beweisens ("Logik für Faule")

2 Spielverderber:

1) Kurt Gödel (1931)

+: jede gültige Aussage kann durch syntaktisches Schließen bewiesen werden
(Vollständigkeitssatz)

−: in der Arithmetik gibt es Aussagen, die weder beweisbar noch widerlegbar sind
(Unvollständigkeitssatz)

$\Rightarrow$ Hilberts (1) funktioniert nicht!

2 Spielverderber

2) Alan Turing (1936)

+ Der Begriff "automatisch entscheiden" lässt sich einfach und sauber definieren
  (⇝ Turingmaschine)

– Für die Arithmetik gibt es kein automatisches Verfahren — sie ist unentscheidbar

⇒ Hilberts (2) funktioniert nicht.

Logik & Mathematik: Geschichte

um 325 v. Chr.:

- Aristoteles: Syllogismen
- Euklid: Versuch einer Axiomatisierung der Geometrie

um 1700.

Leibniz formuliert das Ziel einer universellen Sprache zur Formulierung aller math. Aussagen und eines Kalküls zur Herleitung aller wahren Aussagen

um 1850:

Axiomatisierung der Analysis

1854: Boole: Formalisierung der Aussagenlogik

1879: Frege: Formalisierung der Logik erster Stufe

um 1880:

Cantorsche Mengenlehre,
Rückführung der Analysis + Arithmetik auf die Mengenlehre

um 1900:

Antinomien: Cantorsche Mengenlehre führt zu Widersprüchen (↗ "Menge aller Mengen, die sich nicht selbst als Element enthalten")

⟹ Notwendigkeit einer neuen Grundlegung der Mathematik / Mengenlehre ⟹

um 1900: Hilberts Programm

Ziel: • Formalisierung der Mathematik
• Nachweis der Widerspruchsfreiheit der Arithmetik

um 1910:
Russell, Whitehead: Mengenlehre mit Typen

um 1920:
Zermelo, Fraenkel: Axiomatische Mengenlehre

1930: Gödels Vollständigkeitssatz

1931: Gödels Unvollständigkeitssätze
1936: Church / Turing: Es gibt kein Programm, das für alle mathemat. Aussagen entscheidet, ob sie wahr oder falsch sind

Logik in der Informatik

Anwendungsbereiche der Logik in der Informatik:

- Logische Programmierung
- automatisches Beweisen
- Programm-Verifikation
- Model Checking (automatische Verifikation)
- Logik als Datenbank-Anfragesprache

Model Checking (automatische Verifikation):

2 Beispiele zur Motivation:

1) Der Pentium-Fehler:

Pentium-Prozessor (1993):

- zur Effizienz-Steigerung der Division wurden Wertetabellen verwendet
- ABER: 5 Einträge waren falsch!

$\rightsquigarrow$ ca 1 Fehler je 9 Milliarden Divisionen
(⇒ Fehler durch "Testen" nicht leicht zu finden)

Kosten: ca 475 Millionen Dollar

Intel hat danach viele Experten für automat. Verifikation gesucht!

2) Die Ariane 5-Rakete

Messwerte wurden von 64-Bit-Zahlen in 16-Bit-Zahlen umgewandelt.

- Das hatte bei Ariane 4 gut funktioniert
- ABER: aufgrund der technischen Änderungen waren die Werte bei Ariane 5 größer als erwartet

⇝ überlauf! System schaltete sich ab, Backup-System übernahm

Problem: ... dort lief aber das selbe Programm

Kosten: ca 600 Millionen Euro

Prinzip der automatischen Verifikation

1) Modelliere das zu testende System durch ein Transitionssystem $\mathcal{T}$ (= eine bestimmte logische Struktur; ein beschrifteter Graph)

2) Drücke die (erwünschte oder unerwünschte) Systemeigenschaft durch eine Formel φ einer geeigneten Logik aus

3) Teste, ob $\mathcal{T}$ die Formel φ erfüllt automatisch

Logik als Grundlage für Datenbank-Anfragesprachen

Grundprinzip:

- Datenbank $\hat{=}$ logische Struktur $\mathfrak{A}$
- Anfrage $\hat{=}$ Formel φ einer geeigneten Logik
- Auswerten der Anfrage auf der Datenbank $\hat{=}$ Testen, ob "$\mathfrak{A}$ erfüllt φ" gilt

Details: Vorlesung "Diskrete Modellierung"

Organisatorisches:

- Webseite der Vorlesung:
  www.informatik.uni-frankfurt.de/~tkshp/lehre/WS0910/LI

wichtig: regelmäßig draufschauen:

- "Aktuelles"
- "Logbuch" (Infos zu den einzelnen Vorlesungsstunden)

- Übungen: (André Hernich)
  - Di, nach der Vorlesung:
    Austeilen des aktuellen Übungsblatts
  - Di, in der Übungsstunde:
    Besprechung des letzten Übungsblatts
    Wichtig: SIE rechnen vor
  - Abgabe Ihrer Lösungen: gar nicht!
  - Nötig für Schein:

1) $\geq 40\%$ der erreichbaren Übungspunkte

2) Bestehen von 2 90-minütigen schriftlichen Tests

(letzte Üb.stunde vor Weihnachten +
letzte Üb.stunde des Semesters)

+ 3) regelmäßige aktive Teilnahme an den Übungsstunden.