

Maschinelles Beweisen

Vorlesung mit Übung

WS 2006/2007

F. von Henke, H. Pfeifer

Organisatorisches

Vorlesung: Do 14 – 16

Übungen: Do 16 – 18

Sprechstunde: nach der Vorlesung, oder nach Vereinbarung

friedrich.von-henke@uni-ulm.de

Vorlesungsunterlagen: Kopien der Vorlesungsfolien (PDF) werden über die Webseite zur Vorlesung verfügbar gemacht (Zugang über die Webseite des “Instituts für KI”).

Voraussetzungen: Logik (wie für Vordiplom Informatik)

Die Lehrveranstaltung findet parallel zur *Einführung in die KI* statt und setzt diese nicht voraus.

Es wird im laufenden Semester auch ein Praktikum “Maschinelles Beweisen mit PVS” angeboten; die Vorlesung ist in gewisser Weise Voraussetzung für die Teilnahme daran.

Einordnung der Lehrveranstaltung

Grundlegende Vorlesung für das Gebiet der maschinell unterstützten Deduktion
Grundlage für *formale Methoden* in der Informatik

Folgeveranstaltungen:

- weiterführende Vorlesungen, wie
 - *Computergestützte Modellierung und Verifikation*
- Seminare, Praktika, Bachelor-, Master-, Diplomarbeiten
- Mitarbeit in Forschungsprojekten der Abteilung
- Tätigkeit als studentische Hilfskraft

Maschinelles Beweisen und dessen Anwendungen sind seit vielen Jahren ein Forschungsschwerpunkt des Instituts.

Maschinelles Beweisen

Maschinelles Beweisen generell:

Logische Ableitung oder logisches Schließen mit Unterstützung der Maschine, d.h. des Rechners

- nach den Regeln einer formalen Logik (z.B. Prädikatenlogik) – die Logik definiert, was ein 'Beweis' ist.
- Maschinelle Behandlung ("Mechanisierung") des Beweises soll dazu führen, dass der Vorgang des Beweises und das Resultat (der Beweis) nachvollziehbar, überprüfbar und wiederholbar werden.

Historisches

- Idee des Theorembeweisens mit maschineller Unterstützung existiert seit Beginn der Computer-Entwicklung in den 40er Jahren : A. Turing
- Wichtiges Gebiet in den Anfängen des Gebiets der Künstlichen Intelligenz (KI):
J. McCarthy (1962) - *proof checking*
A. Robinson (ca 1969) - Resolutionskalkül
- intensive Entwicklung von Theorembeweisern in den 70er und 80er Jahren
Fokus des Interesses hauptsächlich auf “automatischem Beweisen”: der Rechner soll einen Beweis (zu einem mathematischem Theorem, . . .) selbständig finden
- zunächst hauptsächlich: Beweisen von mathematischen Theoremen
- Querverbindung zur (Programm-) *Verifikation*: z.B. Beweis von Verifikationsbedingungen, die bei Anwendung des Hoare-Kalküls erzeugt werden.

Historisches (2)

- parallele Entwicklungen:
 - Beweiser für Prädikatenlogik erster Stufe, überwiegend mit Resolution
 - Entwicklung von *Entscheidungsprozeduren* für spezielle Theorien:
 - Aussagenlogik
 - lineare Arithmetik
 - einfache Datenstrukturen (Arrays, Listen, . . .)
 - Termersetzungssysteme: Beweisen in Gleichungslogik
- heute: Mathematiker sind i.a. nicht besonders interessiert an Theorembeweisern als Werkzeug
aber: Kombinationen von Theorembeweisern und Systemen für *symbolisches Rechnen* (z.B. Mathematika, Maple) werden entwickelt
auch: Bestrebungen, mathematische Inhalte (einschließlich zugehöriger Ableitungen) durchgängig maschinell zu formalisieren und für Präsentation aufzubereiten (etwa online im Internet).

Historisches (3)

Hauptinteresse heute bei

- Modellierung und formaler Analyse *kritischer* Systeme:
Sicherheit, Zuverlässigkeit, Vertrauenswürdigkeit, Fehlertoleranz . . .
– Eigenschaften, die oft summarisch als “nicht-funktional” bezeichnet werden –
von Systemen und System-Komponenten
Beweiser als notwendige Inferenz-Komponente beim praktischen Einsatz formaler Methoden
- Beweiser-Module als Hilfskomponenten zur Unterstützung von Inferenzen
(“eingebettete Intelligenz”)
- “proof-carrying Code”: (mobile) Software wird begleitet von einem leicht überprüfbar Nachweis,
dass sie gewisse Eigenschaften besitzt.
- Viele Anwendungen benutzen eine *Logik höherer Ordnung* (im Gegensatz zu PL1)
- Typischerweise Einsatz von Kombinationen von Methoden, weniger Ableitung in einem ‘reinen’
Kalkül

Grade der Automatisierung

- **Beweisprüfung** (*proof checking*): Maschine überprüft einen vorgelegten Beweis daraufhin, ob es sich tatsächlich um einen Beweis handelt.
- **Automatisches Beweisen**: Maschine versucht, einen Beweis “automatisch”, d.h. ohne Mitwirkung eines menschlichen Benutzers, zu konstruieren.
- **Interaktives Beweisen**: Maschine agiert als “Beweisassistent” eines Benutzers
 - Routine-Aktivitäten werden so weit wie möglich von der Maschine übernommen
 - Benutzer hat (manuelle) Kontrolle auf der obersten Ebene der Beweis-Konstruktion
- **Taktisches Beweisen**: Verwendung von “Taktiken” oder “Strategien” – programmierte Beweiskonstruktion bzw. Beweissuche

Grade der Automatisierung (2)

Voll-automatisches Beweisen ist inhärent schwierig bzw. prinzipiell unmöglich, wegen

- Unentscheidbarkeit der Prädikatenlogik erster Stufe, bzw.
- Komplexität der Entscheidungsprobleme für einfachere Logiken (Aussagenlogik u.a.)

Je einfacher die Logik ist, desto größer ist die Chance des vollautomatischen Beweisens

Beispiel: gewisse Klassen von aussagenlogischen Problemen lassen sich mit neueren Ansätzen wie *binary decision diagrams* oder SAT-Solvern effizient behandeln.

Themengebiete der Vorlesung

Logiken und zugehörige Beweisverfahren

- Gleichungslogik und Termersetzung
- Aussagenlogik
- Prädikatenlogik 1. Stufe
- Rekursive Definitionen und Beweis durch Induktion
- Prädikatenlogik höherer Stufe (*Higher-order Logic*)
(getypter) Lambda-Kalkül und Sequenzenkalkül
- Entscheidungsprozeduren
- Weitere Themen je nachdem, wieviel Zeit noch bleibt