Automatisierte Logik und Programmierung II

Sommersemester 2014

Christoph Kreitz

Theoretische Informatik, Raum 1.18, Telephon 3060

kreitz@cs.uni-potsdam.de

http://cs.uni-potsdam.de//alup2-ss14

1. Lehrziele
2. Rückblick & Ausblick
3. Organisatorisches

Ziel der Veranstaltung

Computergestütztes logisches Schließen

- **Mathematische Beweisführung**
  - Aufdeckung und Korrektur von Fehlern (Beweisprüfung)
  - Automatische Suche nach neuen Beweisen (Theorembeweisen)
- **Unterstützung für Entwurf zuverlässiger Software**
  - Fehlersuche und Korrektheitsbeweise (Verifikation)
  - Verbesserung der Performanz (Optimierung)
  - Erzeugung aus Spezifikationen (Synthese)
- **Inferenzmaschine für KI-Systeme**
  - Problemlöser und Planer für Roboter, ...

Themen des ersten Semesters

Inferenzkalküle für Mathematik & Programmierung

- **Beweisen $\hat{=}$ Anwendung formaler Regeln**
  - Umgeht Mehrdeutigkeiten der natürlichen Sprache
  - Erlaubt schematische Lösung mathematischer Probleme
- **Kernbestandteile:**
  - Formale Sprache (Syntax + Semantik)
  - Ableitungssystem (Axiome + Inferenzregeln)
  - Notwendige Eigenschaften: korrekt, vollständig, automatisierbar
  - Nützliche Eigenschaften: konstruktiv, ausdrucksstark, lesbar
- **Vorgestellte Kalküle**
  - Prädikatenlogik (Logisches Schließen)
  - λ-Kalkül (Programmierung)
  - Einfache Typentheorie (Programmeigenschaften)
  - **Konstruktive Typentheorie (CTT)** (Uniformer Kalkül)

Eigenschaften der konstruktiven Typentheorie

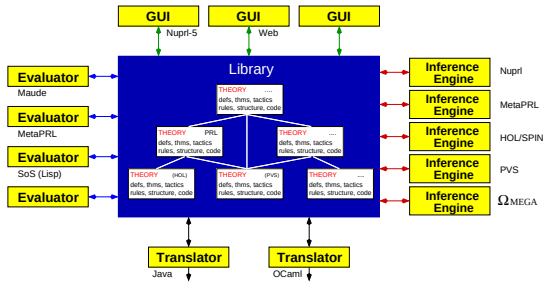
- **Extrem ausdrucksstarkes Inferenzsystem**
  - Vereinheitlicht und erweitert Logik, λ-Kalkül & einfache Typentheorie
  - Direkte Darstellung der zentralen Konzepte (keine Simulation)
  - Formalisierung "natürlicher" Gesetze als Regeln
- **Umfangreicher Formalismus**
  - Viele vordefinierte Basiskonstrukte, mehr als 150 Inferenzregeln
    Prädikatenlogik höherer Stufe, mathematische Grundkonzepte, Programmierkonstrukte
  - Programmkonstruktion durch konstruktive Beweisführung möglich
  - Abhängige Datentypen machen Wohlgeformtheit unentscheidbar
  - Gestützt auf eigenständige konstruktive semantische Theorie
- **Probleme bei der praktischen Anwendung**
  - Beweise erfordern viel Schreibarbeit → *interaktive Beweissysteme*
  - Beweise sind unübersichtlich (komplexer Beweisbaum)
  - Beweise oft schwer zu finden (viele Regeln und Parameter)

    → *Automatisierung der Beweisführung*

Themen des zweiten Teils

Konstruktion und Einsatz von Beweissystemen

- **Aufbau von Beweisassystemen**
  - Implementierung interaktiver Beweisassistenten
  - Das NuPRL Logical Programming Environment
  - Das Isabelle Beweissystem unter ProofGeneral



- **Automatisierung des formalen Schließens**
  - Taktiken und Beweisplanung: benutzerdefinierte Beweisstrategien
  - Entscheidungsprozeduren: Lösung entscheidbarer Teilprobleme
  - Integration externer Beweisprozeduren als Black-Box Systeme
- **Mathematische Assistenzsysteme**
  - Wissensverwaltung und Benutzerinteraktion
- **Anwendungen formaler Beweissysteme**
  - Formalisierung mathematischen Wissens
  - Synthese effizienter Algorithmen aus formalen Spezifikationen
  - Korrektheitserhaltende Optimierung von Algorithmen

⋮

Organisatorisches

- **Zuordnung: theoretische/angewandte Informatik**
- **Veranstaltungen**
  - Vorlesung (Mi 10:15–11:45 (1.02), Do 14:15–15:45 (0.02))
    · Präsentation der zentralen Konzepte / Ideen
    Keine Veranstaltung am 30.4.2014
  - Sprechstunde (Fr 10:30–11:30 . . . und immer wenn die Türe offen ist)
    · Fachberatung / Klärung von Schwierigkeiten mit der Thematik
  - Übungsaufgaben (gelegentlich)
    · Anregung und Herausforderungen zum Selbsttraining
  - Optional: selbstgewähltes praktisches Beweiserprojekt
- **Lehrmaterialien**
  - Vorlesungsfolien, Skript (1995), Fachartikel, Manuals,...
  - Bei Interesse: Nuprl und Isabelle Systeme für eigene Experimente
- **Erfolgskriterium: Abschlußklausur am 17. Juli 2014**
  - Alternativ mündliche Prüfung nach Vereinbarung