

Logikprogrammierung

Prof. Dr. Michael Hanus

mh@informatik.uni-kiel.de, Tel. 880-7271, R. 706 / CAP-4

Sebastian Fischer

sebf@informatik.uni-kiel.de, Tel. 880-7267, R. 705 / CAP-4

Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität zu Kiel

SS 2006

Einführung

Termine

Übungen

Prüfungsmodalitäten

Inhalt

Literatur

Prolog

Termine

Vorlesung

Montag, 10:15 – 12:00 Uhr, Sem-Ü1

Mittwoch, 10:15 – 12:00 Uhr, Sem-Ü1

Übungen

Donnerstag, 14:15 - 16:00 Uhr, Sem-Ü2

Übungsbetrieb (↔ Sebastian Fischer)

- ▶ Anmeldung zu den Übungen:
in der Studierendendatenbank
(www.informatik.uni-kiel.de/ifi/studium-lehre/studidb/)
- ▶ Bearbeitung und Abgabe in **Zweiergruppen**
- ▶ Ausgabe der Übungsblätter: mittwochs (Vorlesung)
- ▶ Abgabe der Übungsblätter: (eine Woche später)
- ▶ Beginn der Übungen: Donnerstag, 13.4.2006

Bachelorprüfung:

- ▶ Modulprüfung am Vorlesungsende (mündlich oder schriftlich)

Leistungsnachweis (Diplominformatik, Nebenfächler):

- ▶ mindestens 50% der Übungspunkte
- ▶ Prüfung am Vorlesungsende

Inhalt

Kurzübersicht:

1. Einführung in Prolog, elementare Programmiertechniken
2. Grundlagen:
 - ▶ Deklarative Semantik
 - ▶ Unifikation, SLD-Resolution
3. Nichtdeklarative Bestandteile
4. Fortgeschrittene Programmiertechniken
5. Erweiterungen: Constraints, Nebenläufigkeit
6. Grundlagen der Implementierung

Literatur

- ▶ M. Hanus: Problemlösen mit Prolog, 2. Auflage, Teubner 1987
- ▶ L. Sterling, E. Shapiro: The Art of Prolog, 2nd Ed., MIT Press, 1994
- ▶ U. Schöning: Logik für Informatiker, 2. Auflage, B.I., 1989
- ▶ J.W. Lloyd: Foundations of Logic Programming, 2nd Ed., Springer 1987
- ▶ H. Aït-Kaci: Warren's Abstract Machine, MIT Press, 1991
- ▶ P. Deransart, A. Ed-Dbali, L. Cervoni: Prolog: The Standard Springer, 1996
- ▶ K. Marriott, P.J. Stuckey: Programming with Constraints: An Introduction, MIT Press, 1998

Prolog

Verwendete Programmiersprache in der Vorlesung: **Prolog**

Gründe:

- ▶ meistverbreitetste logische Sprache
- ▶ de facto - Standard
- ▶ viele Implementierungen verfügbar

Nachteil: wenig Konzepte für den Entwurf großer Systeme
Verbesserung: Modulkonzepte, Typen \rightsquigarrow **Gödel, Mercury**

Entstehung von Prolog

- 1965: Resolutionsprinzip von Robinson [Robinson 1965]
Verallgemeinerung des modus ponens
(wenn A und $A \Rightarrow B$, dann B)
mit Variablensubstitution, genannt **Unifikation**
- 1972-75: Colmerauer/Roussel in Marseille:
Effiziente Implementierung der Resolution → **Prolog**
- 1974: Kowalski: Predicate Logic as Programming Language
Neue Forschungsdisziplin mit dem Prinzip:
Rechnen (auf Computern) = Beweisen

Entstehung von Prolog

- 1977: Pereira/Warren: **DEC-10-Prolog**
Anwendungen von Prolog für
- Expertensysteme
 - Verarbeitung natürlicher Sprache
 - Datenbanken
 - Übersetzerbau
 - Symbolische Mathematik
 - ...
- 1981-1995 : Prolog als Implementierungssprache von Japans
5. Computergeneration
- 1987 - ... **Constraint Logic Programming:**
Verallgemeinerung der Logikprogrammierung
mit beliebigen „Einschränkungen“

Eigenschaften von Prolog

Vorteile:

- ▶ Einfache Syntax
- ▶ Klares Grundkonzept
- ▶ Einfache Semantik (mathematische Strukturen)
- ▶ schnelle Implementierung (Prototyping)
- ▶ Symbolverarbeitung einfach
- ▶ Automatische Speicherverwaltung

Nachteile:

- ▶ Programmausführung manchmal nicht so effizient
- ▶ Speicherintensiv (Rekursion)
- ▶ Noch kein Standard für Module, Typen, ...