

Logische Programmierung

Stefan Brass

Institut für Informatik

Fachgebiet Datenbanken und Informationssysteme

Prof. Dr. Udo W. Lipeck

Organisatorisches

Sprechstunde:

Montags, 14⁰⁰–15⁰⁰ (und nach Vereinbarung)

Lange Laube 22 (Eingang Hausmannstr.), 2. OG

Telefon: 762-4953, Sekretariat: 762-4950

Praktische Übung:

Termin: _____ (nach Vereinbarung)

Raum B417, Beginn nächste Woche

Antrag auf Rechnernutzung im Sekretariat F456

Übungs-Schein:

Vier Hausaufgaben bis Semesterende vorführen.

Skript:

Kopien der Folien, Kosten: 5,-DM (eventuell +2,-DM)

Public Domain Prolog-Interpreter:

Für MS-DOS/ATARI-ST im Institut erhältlich.

Literaturhinweise (1)

Deutschsprachige Lehrbücher:

- Cordes/Kruse/Langendörfer/Rust:
Prolog — Eine methodische Einführung
Vieweg, 1990, 2. Auflage, 245 Seiten, DM 49.80
- Hanus: Problemlösen mit Prolog
Teubner, 1987, 2. Auflage, 224 Seiten, DM 27.80
- Kleine Büning/Schmitgen: Prolog
Teubner, 1988, 2. Auflage, 311 Seiten, DM 36,-

Englischsprachige Lehrbücher:

- Clocksin/Mellish: Programming in Prolog
Springer, 1987, 3. Auflage, 300 Seiten, DM 52,-
(das klassische Prolog-Lehrbuch)
- Dodd: Prolog — A Logical Approach
Oxford University Press, 1990, 556 Seiten, £ 17.50 \cong DM 57.16
- Hogger: Introduction to Logic Programming
Academic Press, 1984, 278 Seiten, \$ 39.95 \cong DM 77.94
- Kluźniak/Szpakowicz: Prolog for Programmers
Academic Press, 1985, 306 Seiten, DM 117.40
(von den Autoren von TOY-Prolog)

Literaturhinweise (2)

Englischsprachige Lehrbücher (Forts.):

- Maier/Warren: Computing with Logic
Benjamin/Cummings, 1988, 535 Seiten, \$ 36.30 \cong DM 70.65
(Aussagenlogik/Datalog/Prolog, Implementierung von Prolog)
- Merritt: Adventure in Prolog
Springer, 1990, 186 Seiten, DM 64,-
(ein einfaches Adventurespiel als durchgehendes Beispiel)
- Nilson/Małoszyński:
Logic, Programming and Prolog
Wiley, 1990, 289 Seiten, \$ 36.95 \cong DM 72.35
(betont logische Grundlagen, behandelt auch Alternativen zu Prolog)
- O'Keefe: The Craft of Prolog
MIT Press, 1990, 387 Seiten, \$ 37.50 \cong DM 73.38
(fortgeschritten, Programmiermethodik, Programmierstil)
- Sterling/Shapiro: Art of Prolog
MIT Press, 1986, 437 Seiten, \$ 39.95 \cong DM 77.95
(etwas fortgeschritten, Programmiertechniken, Anwendungsbeispiele)

Implementierung von Prolog:

- Ait-Kaci: Warren's Abstract Machine
MIT Press, 1991, 114 Seiten, \$30 \cong DM 59.77
(Grundlagen der Prolog-Compilierung)

Literaturhinweise (3)

Theoretische Grundlagen von Prolog:

- Lloyd: Foundations of Logic Programming
Springer, 1987, 2. Auflage, 212 Seiten
(Theoretische Grundlagen von Prolog: Logik, SLD-Resolution)
- Schmitt: Theorie der logischen Programmierung
Springer-Lehrbuch, 1992, 246 Seiten
(Logik und automatisches Beweisen für Prolog und Erweiterungen)

Bücher über automatisches Beweisen:

- Chang/Lee: Mechanical Theorem Proving
Academic Press, 1973, 331 Seiten (das klassische Lehrbuch)
- Duffy: Automated Theorem Proving
Wiley, 1991, 243 Seiten, \$ 41.50 \cong DM 74.31
- Hofbauer/Kutsche:
Grundlagen des maschinellen Beweisens
Vieweg, 1989, 172 Seiten, DM 39,80 (behandelt auch Termersetzung)
- Robinson: Logic and Logic Programming
In: Communications of the ACM 35(3), März 1992, Seite 40–65
(Geschichtlicher Überblick über automatisches Beweisen, vom Erfinder der Resolution)

Warum man Prolog lernen sollte

Horizontalerweiterung:

- eine ganz andere Programmiersprache:
„Programming in Logic“
anders als imperative, funktionale, objekt-orientierte Sprachen
- Deklarative Programmierung:
Nicht wie, sondern was
- Beziehung von Logik und Programmierung

Arbeitsersparnis:

- Patternmatching, Backtracking sind eingebaut
- Listen- und Baumstrukturen einfachst benutzbar
- Automatische Speicherverwaltung
- Eingebaute (Hauptspeicher-) Datenbank
- Grammatiken direkt ausführbar

Metaprogrammierung:

- Verarbeitung anderer Prologprogramme
In Prolog besonders einfach zu realisieren

Historisches

Logik:

- Frege: Mathematische Beweisschrift (1879)
- Whitehead/Russel: Principia Mathematica (1910)
- Gödel/Herbrand: Vollständigkeit (1930)
- Church/Turing: Unentscheidbarkeit (1936)

Automatisches Beweisen:

- Gilmore/Davis/Putnam: Erster Beweiser (1960)
- Robinson: Resolventenverfahren (1965)
- Loveland/Luckham: Lineare Resolution (1970)

Logisches Programmieren:

- Colmerauer: Frage-Antwort-System (1971)
- Colmerauer/Roussel: Prolog (1972/73)
Zum Vergleich: Fortran (1954), Lisp (1962), Pascal (1970), Ada (1979)
- Kowalski: Algorithm = Logic + Control (1973?)
- Warren: Compiler für Prolog (1977)

Inhalt (geplant)

Prolog Programmierkurs:

- Syntax, Unifikation, Vierport-Modell
- Eingebaute Prädikate
- Prolog vs. Pascal

Logische Grundlagen:

- Deklarative Semantik von Pure-Prolog
- Grundlagen des automatischen Beweisens

Programmiertechniken:

- Standard-Algorithmen
- Grammatiken in Prolog
- Programmierstil und Effizienz

Implementierung von Prolog:

- Interpreter, Compiler

Ausblick:

- Alternative Logische Programmiersprachen