

VIP - eine integrierte Programmierumgebung für Prolog

Andreas Krall, Eva Kühn
Institut für Praktische Informatik
Abteilung für Programmiersprachen und Übersetzerbau
Technische Universität Wien
Argentinierstraße 8
A-1040 Wien

Kurzfassung

Dieser Beitrag beschreibt den Entwurf und die Implementierung von VIP (Vienna Integrated Prolog). VIP besteht aus drei Komponenten: Der Kern umfaßt einen Zwischencodeinterpreter, einen inkrementellen Compiler, einen Memory-Manager, einen Window-Manager und eine Betriebssystemschnittstelle. Zur Programmierumgebung gehören ein syntaxgesteuerter Prologeditor, ein intelligentes Debugging-System sowie diverse Testhilfen. Auf der Applikationsebene werden das logikorientierte Datenbanksystem VIP-DBS, eine Expertsystemshell, ein Maskengenerator, ein Spreadsheetprogramm, der Parsergenerator VIP-PAGE und ein natürlichsprachiges Interface für Deutsch zur Verfügung gestellt. VIP-Prolog ist eine Übermenge von DEC-10 Prolog. Die Erweiterungen enthalten ein Modulkonzept, die Unterstützung globaler Variablen und eine große Auswahl an Systemprädikaten.

Schlüsselworte: Prolog, Programmierumgebung

1. Einleitung

Seitdem Prolog erfunden wurde, ist sehr viel Forschung auf dem Gebiet der logischen Programmierung betrieben worden. Die Implementierung von Warren [10] zeigte, daß es möglich ist, effiziente Prologsysteme zu entwickeln. Weiters hat sich gezeigt, daß das Konzept der logischen Programmierung ein sehr wesentliches Paradigma in der Welt der Programmiersprachen darstellt. Neben der Idee der funktionalen Programmierung stellt das Konzept der logischen Programmierung einen neuen Weg dar, die Welt 'assertional' (= d.h. durch Zusicherungen) abzubilden. Die meisten Forschungsaktivitäten auf dem Gebiet der logischen Programmierung beschäftigen sich heute mit Parallelismus, erweiterten Sprachkonzepten und mehrwertiger Logik.

Wir vertreten die Meinung, daß es wichtig ist, sich auf die Methoden des Software Engineering zu konzentrieren, die beim logischen Programmieren verwendet werden können. Diese Idee ist vorerst unabhängig davon, welches Modell der Logik man betrachtet. Weiters gilt sogar, daß die Betrachtung passender Programmiermethoden umso wichtiger ist, je komplizierter das Modell ist.

Konkret wollen wir zeigen, was 'strukturierte Programmierung' im Sinne prozeduraler Sprachen für Prolog bedeutet. Weiters soll ein gutes Prologsystem einen neuen, guten Programmierstil für Prolog unterstützen. Unterstützung beim Testen, Debuggingwerkzeuge, ein gut entworfenes Benutzerinterface, ein Modulkonzept und wiederverwendbarer Programmcode sind Hauptvoraussetzungen für diese Forderung. Die unterstützte Sprache darf nicht eingeschränkt sein, sondern soll sogar durch die Unterstützung sogenannter 'add-on' Modulen erweiterbar und veränderbar sein. Um ernsthaft Programme entwickeln zu können, ist es wichtig, daß das Prologsystem effizient implementiert ist. Alle diese Begriffe sind wichtige Voraussetzungen, um die Programmentwicklung 'im großen' zu ermöglichen.

Wir werden im folgenden das Projekt VIP (Vienna Integrated Prolog) vorstellen, das versucht, die oben genannten Ziele zu verwirklichen. VIP ist ein Forschungsprojekt am Institut für Praktische Informatik, Abteilung für Programmiersprachen und Übersetzerbau. VIP wurde bisher von der Nationalbank und vom Bundesministerium für Forschung und Wissenschaft gefördert. Im zweiten Kapitel wird ein Überblick über die Architektur des VIP Systems gegeben. Im dritten Kapitel beschreiben wir die implementierte Sprache und im vierten Kapitel die Implementierung der abstrakten VIP Maschine.

2. Die Architektur von VIP

Die Architektur von VIP besteht aus drei Ebenen:

- dem Kernel
- der Programmierungsumgebung
- der Applikationsschicht

2.1. Der Kernel

Der VIP Kernel besteht aus einem Zwischencodeinterpreter mit inkrementellem Compiler und einem Memory Management. Die implementierte Sprache ist eine Obermenge von DEC-10 Prolog. Die nachfolgenden Kapitel werden sich mit Sprache und Kernimplementierung beschäftigen. Die Prologsprache weist ein Modulkonzept auf und ist mittels Bibliotheken beliebig erweiterbar. Gleichzeitig wird die Forderung nach einem 'high-speed' Prologinterpreter, der auf herkömmlichen Computerarchitekturen läuft, erfüllt.

2.2. Die Programmierungsumgebung

Von dieser Schicht des VIP Systems werden die Forderungen nach Debuggingwerkzeugen und nach einer gut entworfenen Benutzerschnittstelle erfüllt. Wir präsentieren hier nur einige Entwurfsüberlegungen: