

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В ИЗУЧЕНИИ ФУНКЦИОНАЛЬНОГО И ЛОГИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Дацун Н.Н.
Кафедра ПМИ ДонГТУ
datsun@pmi.donetsk.ua

Abstract

Datsun N.N. Computer Technologies in Learning Functional and Logic Programming. This paper contains analyses of Lisp and Prolog realizations for teaching applications. Reasons of our own realization of the artificial intelligence languages are formulated. Characteristics of following systems are described, such as: SDMLisp for Windows, Fuzzy Prolog for DOS and Datalog for Windows. This software was developed and applied on "Applied Mathematics and Informatics" department.

Введение

Рассмотрены вопросы компьютерной поддержки обучения функциональному и логическому программированию студентов специальности 7.080403 "Программное обеспечение ВТ и автоматизированных систем" (ПО ВТиАС). Выполнен анализ реализаций языков Lisp и Prolog для учебных приложений. Обоснован выбор трансляторов, используемых в учебном процессе, и сформулированы причины выполнения собственных реализаций языков искусственного интеллекта. Приведены характеристики программных продуктов, разработанных и используемых на кафедре ПМИ: SDMLisp for Windows, Fuzzy Prolog for DOS и Datalog for Windows.

1. Структура дисциплины "Функциональное и логическое программирование"

Дисциплина "Функциональное и логическое программирование" (ФиЛП) для студентов специальности ПО ВТиАС в ДонГТУ преподается на 5-м семестре и состоит из четырех разделов: функциональное программирование и язык Lisp; логическое программирование и язык Prolog; логическое программирование и базы данных; дедуктивные базы данных (БД) и язык Datalog

2. Выбор программных средств для поддержки обучения

2.1. Анализ реализаций языка Lisp

Язык Lisp с момента своего создания имел относительно эффективные реализации на больших и малых ЭВМ. Идеология этого языка находится в глубоком противоречии с традиционной архитектурой фон Неймана. Семантический разрыв между архитектурой аппаратуры и архитектурой таких динамических по своей природе языков как Lisp и Prolog, может быть устранен принципиально только при аппаратной поддержке реализации языка. В случае же реализации языка Lisp на ЭВМ с традиционной архитектурой основная проблема заключается в организации памяти, ее распределении и утилизации.

Основной особенностью языка Lisp, влияющей на ограниченность его распространения, является его обособленность (с точки зрения сферы применимости) и относительная замкнутость по отношению к другим видам программного обеспечения. Это серьезный недостаток на фоне всеобщей тенденции в информатике к интегрированности и преобладанию графических пользовательских интерфейсов.

Первая Lisp-система для персонального компьютера - Apple Lisp - была разработана для Apple II в 1980 г. Актуальность применимости языка Lisp для создания коммерческого программного обеспечения, а также необходимость реализации Lisp'a для PS/2 и интеграции языка с MS Windows были осознаны в начале 1990-х гг. Промышленный стандарт языка Lisp - Common Lisp - получает все большее распространение. Назовем существующие реализации Лиспа для персональных компьютеров и рабочих станций [1]: muLISP фирмы IBM для IBM PC с MS DOS Golden Common Lisp 4.0 фирмы Gold Hill Inc. для Compaq 386 и PS/2 моделей 70 и 80; Allegro Common Lisp 4.0 фирмы Franz Inc. для рабочих станций с UNIX; Common Lisp 4.0 фирмы Lucid Inc. для рабочих станций Appolo, Sun, DEC, IBM, NCR, Prime с X-Windows.

Среди названных реализаций доступным для учебных приложений является система muLISP фирмы IBM для IBM PC с MS DOS. При выполнении лабораторных работ по дисциплине "ФиЛП" студентами используется транслятор muLISP-86.

2.2. Анализ реализаций языка Prolog

Стандартом de facto языка Prolog является DEC-10 Prolog (Эдинбургская реализация Д.Уоррена), синтаксиса которого придерживаются практически все реализации языка. Большинство современных диалектов Prolog'a выполнены для операционной системы UNIX[2]: CProlog Эдинбургского Университета, Quintos Prolog фирмы Quintos Computer Systems Inc., SKW фирмы Silogic Inc., UNSW Prolog Университета Нового Южного Уэльса Среди реализаций Пролога для персональных компьютеров и рабочих станций отметим следующие [1, 2]; Turbo-Prolog фирмы Borland Int. для IBM PC в MS DOS; Prolog-2 фирмы Expert Systems Ltd. для IBM PC XT/AT с MS-DOS или Windows 3.0; Prolog III фирмы Prologia и Марсельского университета; Quintus Prolog 3.0 фирмы Quintus Computer Systems Inc. для рабочих станций Sun 3 и 4 с X-Windows; МПролог (МБТУ) для IBM PC под MS DOS, PC DOS; Arity Prolog фирмы Arity Corp.; Prolog-86 фирмы MICRO-AI (версия UNSW для MS-DOS); Chalcedony Prolog фирмы Chalcedony (версия UNSW для MS-DOS и Apple Macintosh).

Для учебных приложений основными критериями транслятора Prolog'a являются интегрированность среды, наличие компилятора, возможности интерфейса с другими языками и расширения языка в сторону пограничных областей информатики. По этим параметрам и проведем анализ реализаций языка Prolog [1,2].

Среда разработки программ. Текстовые редактор. Специфика UNIX отражается на среде разработки программ: в CProlog и UNSW отсутствуют встроенные редакторы (пользователю следует пользоваться редакторами UNIX), а Quintus Prolog и SKW можно запускать либо как самостоятельный процесс, либо как подпроцесс под управлением редактора EMACS. Реализации в MS-DOS имеют встроенные полноэкранные (Prolog-2) или полиэкранные (Turbo-Prolog) редакторы. Отладчики. Все рассматриваемые диалекты Prolog'a имеют отладчики, работающие на дереве доказательства целей. Однако отладчики систем Arity Prolog, SWK и Prolog-2 не могут работать с откомпилированным кодом.

Компиляторы. Компиляторы есть в Quintus-Prolog (пошаговый), SKW и Arity Prolog (внешние). Транслятор Turbo-Prolog является компилятором в отличие от всех прочих реализаций.

Интерфейс с другими языками. Во всех реализациях, основанных на CProlog'e (CProlog, UNSW), допустим интерфейс с языком C путем модификации исходного текста соответствующего интерпретатора. SKW допускает подключение к Prolog-программе функций языка C. В Quintus-Prolog и Turbo-Prolog есть возможность сопряжения с модулями на языках C, Pascal, Assembler, Fortran (в Quintus-Prolog также допустим интерфейс с Cobol- и Lisp-программами), а также механизм отображения структур данных названных языков на структуры данных Prolog'a для передачи их аргументами. Prolog-2 допускает подключение объектных файлов, оттранслированных с других языков программирования, но ответственность за отображение структур данных в Prolog лежит на программисте.

Расширения языка. Расширения языка Prolog представлены следующими направлениями: DCG-расширение (CProlog, SKW, Quintus-Prolog, Prolog-2, Arity Prolog) для написания процедур нисходящего грамматического разбора; система управления реляционными базами данных (SKW); язык SQL (Arity Prolog), пакет прикладных программ для разработки экспертных систем (Arity Prolog); библиотеки для обеспечения совместимости с DEC-10 Prolog (Prolog-86); средства графики, управление окнами, цветом, звуком (Turbo-Prolog).

Среди названных реализаций наиболее подходящим для учебных приложений является система Turbo-Prolog с эффективным компилятором, с развитой интегрированной средой и многооконным интерфейсом, со средствами разработки больших программ (project). При выполнении лабораторных работ по дисциплине "ФиЛПе" студентами используется транслятор Turbo-Prolog 2.0 для MS-DOS.

При выполнении лабораторной работы по теме "Системы связывания Пролога с реляционными базами данных" студентами используется транслятор PROLOG-PLUS alpha version (КТУ) для MS-DOS. Эта CPR-система поддерживает интерфейс языка Prolog с .dbf-файлами и модульный принцип программирования на Prolog'e.

2.3. Анализ реализаций языка Datalog

Анализ реализаций языка DataLog [3] показывает, что все реализации находятся на этапе "Исследовательский проект" жизненного цикла программного продукта.

2.4. Причины разработки собственного программного обеспечения

Следующие причины побудили нас к созданию своих собственных систем программирования для поддержки процесса обучения по дисциплине "Функциональное и логическое программирование":

- экспансия Windows-технологии в программном обеспечении;
- недоступность для отечественного университетского образования фирменных продуктов для языков Lisp и Prolog;
- "закрытость" компилятора Turbo-Prolog, не допускающая расширения транслятора на случай логического вывода в условиях неопределенности, а позволяющая только его непосредственное программирование;
- наличие только исследовательских проектов по дедуктивным БД и Datalog'у.

В настоящее время в учебном процессе при изучении "ФиЛП" используются собственные разработки кафедры "Прикладной математики и информатики". Это интегрированные среды SDMLisp for Windows, Fuzzy Prolog for DOS и Datalog for Windows. Охарактеризуем названные программные продукты для учебных приложений.

2. Характеристики программных продуктов

2.1. SDMLisp for Windows

SDMLisp for Windows (реализация Д.М.Салашника - дипломный проект студента ФВТИ) - интерпретатор языка Common Lisp с интегрированной средой типа Delphi и GUI-интерфейсом:

- допустима работа в режиме интерпретации и псевдокомпиляции, имеется встроенный отладчик,
- распределение Lisp-памяти выполняется по собственному алгоритму (нет необходимости в "сборщике мусора" в чистом виде);
- не реализованы разрушающие функции, которые в учебных приложениях не используются.

2.2. Fuzzy Prolog for DOS

Fuzzy Prolog for DOS (реализация А.В.Бондаренко - научно-исследовательская работа студента ФВТИ) - интерпретатор языка Prolog близкий по синтаксису и возможностям к эдинбургской версии, допускающий логический вывод как традиционный, так и в условиях неопределенности:

- при логическом выводе в условиях неопределенности с каждым утверждением связывается коэффициент уверенности CF (в стиле MYCIN). Он задается после утверждения в виде десятичного числа в диапазоне 0..100;

$$P(X,2):-q(X),r(a,X,b):70.$$

- если коэффициент уверенности не задан, то он принимается равным 100;
- вывод считается в условиях неопределенности, если при доказательстве цели хотя бы одна из подцелей сопоставляется с утверждением, имеющим CF, отличный от 0.

2.3. Datalog for Windows

Datalog for Windows (реализация Д.Г.Жирного [4]) - транслятор языка Datalog, допускающий интерфейс с несколькими коммерческими реляционными СУБД;

- особенностью Datalog'a является наличие множества его вычислительных моделей. В реализации выбрана инфляционная модель как наиболее естественная для программ с отрицанием. Опытная эксплуатация транслятора в учебном процессе показала необходимость наличия минимум двух вычислительных моделей (для линейных и взаимнорекурсивных программ), а также механизма выбора модели;
- экстенциональная БД может быть как во внешней памяти (Datalog связывается с файлами в форматах многих коммерческих СУБД - dBase, FoxPro, Paradox и др.через IDAPI), так и в теле Datalog-программы в виде множества фактов;
- при вычислении проводится оптимизация порядка предикатов в запросе для повышения скорости получения результатов по нему;
- степень прозрачности интерфейса логического вывода - полная;
- связывание логического вывода с БД динамическое.

Все разработанные интегрированные среды имеют встроенные текстовые редакторы, а Datalog for Windows - собственный редактор БД и другие сервисные средства.

Литература

1. Пантелеймонов А. Основные языки программирования искусственного интеллекта. / КомпьютерПресс, 1991, N9, с. 33-38.
2. Малпас Дж. Реляционный язык Пролог и его применение. - М: Наука, 1990, 464 с.
3. Ceri S., Gottlob G., Tanca L. Datalog: a self-contained tutorial. / Программирование, 1991, N 5, с. 9-31.
4. Жирный Д.Г., Дацун Н.М. Реалізація інтегрованого середовища Дейталогу для навчання логічному програмуванню та його оцінка / В кн. "ІТОН'97". Збірник тез Всеукраїнської конференції молодих науковців "Інформаційні технології у науці та освіті", Черкаси, 1997, с.27.