

РЕФЕРАТ

По “Логическому программированию”

На тему

“Современные языки и системы логического программирования”

Работу выполнил студент группы

М8О-310Б-21

Барсуков Е.А.

Оглавление

Введение.....	3
Языки и системы логического программирования.....	4
Применение языков логического программирования.....	7
Вывод.....	8

Введение.

Логическое программирование - парадигма программирования, основанная на математической логике — программы задаются в форме логических утверждений и правил вывода. Программа формирует новые факты, основываясь на уже введенных.

Можно считать, что это упрощенная версия функционального программирования. Здесь используется некое подобие натурального для человека языка (символы и факты). Алгоритм решений не требуется, вместо него описываются мир задачи, объекты и их свойства, отношения между ними, факты. В ходе выполнения программы мы можем “задать ей вопрос”, на который будет дан ответ исходя из введенных правил.

Программа в ЛП работает как база данных: выдаёт по запросу пользователя логический факт из уже имеющихся в базе, зачастую используя метод полного перебора всех фактов и условий.

Самым первым языком логического программирования был Planner, созданный в 1969 году. Его использовали для снижения требований к вычислительным ресурсам и получения фактов без активного использования стека. Он стал основой для разработки более современного и самого популярного на сегодняшний день языка логического программирования — Prolog.

Prolog — уникальный язык программирования. С его помощью выполняются различные задачи в самых различных сферах, при этом уже много лет в языке не происходит существенных изменений.

На языках логического программирования разрабатываются быстрые прототипы прикладных программ, осуществляется управление производственными процессами, создаются динамические реляционные базы данных. Также их используют при написании языковых переводчиков или для создания языковых интерфейсов существующих программ.

Языки и системы логического программирования.

Существует множество различных языков логического программирования. Они заточены под определенные задачи, могут быть основаны друг на друге. Я расскажу о некоторых из них.

Начнем с самого популярного и известного языка – **Prolog**. Prolog - язык программирования, который основан не на алгоритме, как многие другие классические языки, а на логике предикатов и фактов. Если программа на процедурном языке является последовательностью инструкций, выполняющихся в заданном порядке, то Prolog содержит только описание задачи, а Prolog-машина выполняет поиск решения, руководствуясь только этим описанием и используя механизм поиска с возвратом и унификацию.

Разработка языка Prolog началась в 1970 г. Аланом Кулмером и Филиппом Русселом в университете города Марсель, Франция. Они хотели создать язык, который мог бы делать логические заключения на основе заданного текста. Их работа была мотивирована желанием примирить использование логики в качестве декларативного языка представления знаний с процессуальным подходом к представлению знаний, который был популярен в Северной Америке в конце 1960-х и начале 1970-х годов.

Название Prolog было выбрано Филиппом Русселом как аббревиатура от французского “PROgrammation en LOGique” (PROLOG) и первая реализация этого языка с использованием компилятора Николауса Вирта "AlgolW" была закончена в 1972 году. Основы же современной версии языка были заложены в 1973 году. Prolog постепенно распространялся среди тех, кто занимался логическим программированием, в основном благодаря личным контактам, а не через коммерциализацию продукта.

Prolog реализован практически для всех известных операционных систем и платформ, в том числе и для Java, .NET. В число операционных систем входят OS для мэйнфреймов, всё семейство Unix, Windows, OS для мобильных платформ.

Многие современные реализации языка имеют внутреннее расширение за счет ООП-архитектуры. Кроме проприетарных решений также существуют свободные реализации Prolog.

Базовым принципом языка является равнозначность представления программы и данных (декларативность), благодаря чему утверждения языка одновременно являются и записями, подобными записям в реляционной базе данных, и правилами, несущими в себе способы их обработки. Сочетание этих качеств приводит к тому, что по мере работы системы Prolog знания (и данные и правила) накапливаются. Поэтому Prolog-системы считают естественной

средой для накопления базы знаний и обучения студентов и школьников принципам логического программирования.

Стоит упомянуть о минусах языка. Prolog критикуется в первую очередь за свою недостаточную гибкость, отчего решения на обычных языках программирования в сочетании с базами данных оказываются более технологичными, чем аналогичные решения на Prolog. Негибкость заключается в трудности изучения языка, трудности отладки программы, неразвитости технологии программирования, плохой контролируемости промежуточных результатов. Некоторые из этих проблем нивелируются в современных реализациях языка, таких как, например, SWI-Prolog, на котором были выполнены лабораторные работы. Внутри него доступен полноценный дебаггер Prolog.

Datalog - декларативный язык для запросов к базам данных на основе исчисления доменов. Он разработан в 1978 году, а синтаксически является подмножеством или продолжением языка Prolog. Широкого применения в реальных базах данных не получил, но повлиял на формирование более поздних языков для запросов, таких как SQL.

В отличие от Prolog, операторы программы Datalog могут быть указаны в любом порядке. Более того, запросы Datalog к конечным множествам гарантированно завершатся, поэтому в Datalog нет оператора cut, как в Prolog. Это делает Datalog полностью декларативным языком.

Перечислим основные отличия Datalog от Prolog:

- запрещает сложные вложенные термы в качестве аргументов предикатов, например, $p(1, 2)$ допустимо, но не $p(f(1), 2)$,
- требует, чтобы каждая переменная, которая появляется в заголовке предложения, также появлялась в неарифметическом положительном (то есть не инвертированном) литерале в теле предложения,
- требует, чтобы каждая переменная, появляющаяся в отрицательном литерале в теле предложения, также появлялась в некотором положительном литерале в теле предложения.

Mercury – это функционально-логический язык программирования со строгой типизацией, системой режимов аргументов и детерминизмом предикатов. Синтаксис так же, как и в случае Datalog, частично унаследован от Prolog.

Разработчики Mercury сделали язык чисто декларативным, полностью убрав из него все императивные возможности. Добавили развитую систему типов, режимы аргументов, категории детерминизма предикатов и условные операторы. Оказалось, что такой подход имеет право на существование и, несмотря на некоторые трудности с организацией ввода-вывода, дает высокопроизводительный код. Причем оптимизация программы в этом случае производится автоматически самим компилятором, а не программистом, избавляя его от необходимости продумывания последовательности выполнения алгоритма.

Название Mercury, данное в честь бога скорости Меркурия, отражает направленность на получение быстро работающих программ. Фактически Mercury по многим тестам в разы опережает по производительности современные реализации Prolog.

Компилятор Mercury открыт и бесплатен, работает на всех современных операционных системах. Особенностью языка является то, что он способен транслировать код во многие популярные языки программирования: C, Java, Erlang, IL для платформы .NET.

Применение языков логического программирования

Языки логического программирования имеют множество применений. Среди них есть как стандартные, так и необычные. Например, с помощью Prolog можно легко и, что самое главное, математично описать вычисление производной:

$$d(X,X,1) :- !.$$
$$d(T,X,0) :- \text{atomic}(T).$$
$$d(U+V,X,DU+DV) :- d(U,X,DU), d(V,X,DV).$$
$$d(U-V,X,DU-DV) :- d(U,X,DU), d(V,X,DV).$$
$$d(-T,X,-R) :- d(T,X,R).$$
$$d(C*U,X,C*W) :- \text{atomic}(C), C \neq X, !, d(U,X,W).$$
$$d(U*V,X,Vd*U+Ud*V) :- d(U,X,Ud), d(V,X,Vd).$$
$$d(U/V,X,(Ud*V-Vd*U)/(V*V)) :- d(U,X,Ud), d(V,X,Vd).$$

Но, конечно, возможности языка этим не ограничиваются. Возможности Prolog и других языков ЛП не позволяют писать большие программы, но зато отлично подходят для прототипирования. Помимо чисто прототипирования, ЛП очень хорошо подходит для трансляции фрагмента логики в код. Таким образом, автоматические проверки и тому подобные вещи могут быть легко написаны на Prolog и других языках ЛП.

Первый интерпретатор Erlang был написан на Prolog, поскольку Prolog очень хорошо подходит для синтаксического анализа и кодирования логики. Более того, в Prolog есть встроенный синтаксический анализатор.

Языки логического программирования так же широко используются в NLP, особенно в синтаксисе и вычислительной семантике.

Вывод

Логическое программирование служит для работы с формальными и естественными языками, баз данных, запросных и экспертных систем и для других дискретных не вычислительных задач. Большинство языков логического программирования разработано в конце прошлого века. Но из-за этого факта они не теряют свою актуальность. Prolog, появившийся еще в 1972 году, используется до сих пор в своем кругу задач.

Можно сделать вывод, что сейчас нет необходимости разрабатывать новые языки логического программирования. Даже созданные более 40 лет назад языки не потерпели сильных изменений, их принцип работы сохранился. Они так же используются в своих определенных областях.