

Национальный исследовательский университет «Московский авиационный институт»

Факультет №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

РЕФЕРАТ

на тему

Логическое программирование

Студент:	Мохляков П. А
Группа:	108
Преподаватель:	Поповкин А. В.
Подпись:	
Оценка:	
Дата:	

Москва
2019

Содержание

Введение.....	2
Prolog.....	3
SAT. Задача вычислимости булевых функций.....	4
ASP. Программирование набора ответов.....	5
Заключение.....	6
Список использованных источников	7

Введение

Логическое программирование — парадигма программирования, основанная на автоматическом доказательстве теорем, а также раздел дискретной математики, изучающий принципы логического вывода информации на основе заданных фактов и правил вывода. Логическое программирование основано на теории и аппарате математической логики с использованием математических принципов резолюций.

Логическое программирование – это подход к программированию, при котором программа задается совокупностью правил без явного указания последовательности их применения.

Наиболее удивительным примером использования идей, методов и понятий теории логического вывода в программировании является логическое программирование.

Самым известным языком логического программирования является Prolog.

Prolog

Prolog (Programming in Logic, в оригинале: programmation en logique) был разработан в Марселе в начале 70-х Аленом Колмероз. В основу языка легла процедурная интерпретация логических выражений. Логическое выражение можно прочесть как: выражение «a» верно, если все выражения предпосылки к нему верны.

Таким образом «a» верно, если я могу доказать «b», доказать «c» и «d». Из всего этого следует, что каждая программа — это набор теорем для вывода утверждений, а каждое выражение доказывается. Отрицание в этом языке отличается от логического. В прологе отрицание можно высказать выражением как: «если я не смог доказать выражение «a», то оно не верно».

Пролог является языком полным по Тьюрингу, поэтому любой алгоритм можно реализовать на данном языке.

Задачи, которые можно решить с помощью Prolog

Prolog решает следующие задачи:

1. Разработка экспертных систем;
2. Решение задач и вывод решения, доказательство теорем;
3. Символьные и численные вычисления (например: решение уравнений, дифференцирование, интегрирование и т. д.);
4. Создание упрощенных версий систем Искусственного Интеллекта;
5. Снабжать существующие программы естественно-языковым интерфейсом;
6. Трансляция с одного языка на другой, также с одного языка программирования на другой.

Хороший пример задачи, которая хорошо формулируется и решается на прологе, — это набор правил, согласно которому выполняется или нет определенное условие. Однако вам самим придется задать алгоритм поиска решения — что является пространством допустимых значений, в каком порядке они обходятся и тд. По сути вы моделируете задачу в виде правил вывода и с помощью правил вывода задаете процедуру поиска решения и допустимое пространство решений.

Далее нам следует рассмотреть такие понятия как SAT и ASP

SAT. Задача вычислимости булевых функций.

Экземпляром задачи SAT является булева формула, состоящая только из имен переменных, скобок и операций И, ИЛИ и НЕ. Задача заключается в следующем: можно ли назначить всем переменным, встречающимся в формуле, значения ложь и истина так, чтобы формула стала истинной.

Согласно теореме Кука, доказанной Стивенем Куком в 1971 году, задача SAT для булевых формул, записанных в конъюнктивной нормальной форме, является NP-полной. Требование о записи в конъюнктивной форме существенно, так как, например, задача SAT для формул, представленных в дизъюнктивной нормальной форме, тривиально решается за линейное время в зависимости от размера записи формулы (для выполнимости формулы требуется только наличие хотя бы одной конъюнкции, не содержащей одновременно x и $\text{NOT } x$ для некоторой переменной x).

Данная форма программирования является императивной, то есть описывающий как решить задачу и как представить результат.

ASP. Программирование набора ответов.

Программирование наборов ответов — форма декларативного программирования, ориентированная на сложные (в основном NP-трудные) задачи поиска, основывающееся на свойствах стабильной семантики логического программирования. Задача поиска сводится к вычислению устойчивой модели и наборов решателей — программ для генерации устойчивых моделей, которые используются для поиска.

Что такое декларативный подход? Если вкратце, то мы описываем задачу и её свойства, а не как её решать. В этом случае задача чаще всего представляется в виде: Проблема = модель + Поиск

Где мы регулярно встречаемся с таким подходом? Например, в базах данных SQL — это декларативный язык запросов, а поиском ответа на этот запрос занимается СУБД.

Заключение

В настоящее время логическое программирование практически нигде не используется в реальных проектах, так как языки, которые используют другие парадигмы, такие как объектно ориентированное или функциональное программирование, более просты для понимания и создание сложных алгоритмов, легко расширяются и дают большую скорость разработки.

В связи со всеми этими обстоятельствами языки с парадигмой логического программирования практически не развиваются. Не смотря на это у языков с данной парадигмой есть еще один путь развития - это самообучающиеся системы. За счет развития методов перебора, позволяющих максималльно упростить описание моделируемых объектов.

Также логические языки программирования хорошо подходят для создания баз знаний и экспертных систем, при помощи такой формы декларативного программирования, как программирование набора ответов.

Список использованных источников

Что такое логическое программирование и зачем оно нам нужно // Habr URL:
<https://habr.com/ru/post/322900/>

В. Н. Нефедов, В. А. Осипова Курс дискретной математики. Москва:
Издательство МАИ, 1992.

С. С. Гайсарян В. Е. Зайцев Курс информатики. 12 изд. Москва: Издательство
Вузовская книга, 2012.

Логическое программирование // Wikipedia URL:
https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5

Плэнер // Wikipedia URL:
<https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D1%8D%D0%BD%D0%B5%D1%80>

Пролог // Wikipedia URL:
[https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BB%D0%BE%D0%B3_\(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F\)](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BB%D0%BE%D0%B3_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F))

Декларативное программирование // Wikipedia URL:
https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BA%D0%BB%D0%B0%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5

Задача выполнимости булевых формул // Wikipedia URL:
https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%B2%D1%8B%D0%BF%D0%BE%D0%BB%D0%BD%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D0%B8_%D0%B1%D1%83%D0%BB%D0%B5%D0%B2%D1%8B%D1%85_%D1%84%D0%BE%D1%80%D0%BC%D1%83%D0%BB

Вычислительная сложность // Wikipedia URL:
https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%BB%D0%BE%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D1%8C

Программирование наборов ответов // Wikipedia URL:
https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BD%D0%B0%D0%B1%D0%BE%D1%80%D0%BE%D0%B2_%D0%BE%D1%82%D0%B2%D0%B5%D1%82%D0%BE%D0%B2