

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Бобрешов А. М., Коровченко И. С., Степкин В. А., Телков А. Ю.

Экспертные системы: лабораторные работы

(Учебное пособие)

Воронеж 2014

Содержание

1. Программирование на языке Visual Prolog.....	4
1.1. Теоретические основы логического программирования	4
1.1.1. Область применения языка Prolog	4
1.1.2. Общие сведения о языке Prolog.....	6
1.1.3. Программирование на языке Visual Prolog	16
2. Программирование на языке CLIPS.....	29
2.1. Общие сведения о языке CLIPS.....	29
2.2. Задача "Правдолюбцы и лжецы"	39
3. Список использованных источников	47

ней. При этом их поведение рационально в том смысле, что они способны к пониманию, а их действия направлены на достижение какой-либо цели. Такой агент может быть как роботом, так и встроенной программой. Об интеллектуальности агента можно говорить, если он взаимодействует с окружающей средой примерно так же, как действовал бы человек.

1.1.2. Общие сведения о языке Prolog

Теоретической основой Prolog является раздел символьной логики, называемый исчислением предикатов. Предикат — это логическая функция, которая выражает некоторое отношение между своими аргументами и принимает значение "истина", если это отношение имеется, или "ложь", если оно отсутствует.

В отличие от процедурных языков, где программист должен описать процедуру решения шаг за шагом, при использовании Prolog достаточно описать задачу и основные правила ее решения. Он имеет ряд свойств, которыми не обладают процедурные языки, что делает его мощным средством логического программирования. К числу таких свойств относятся:

- механизм вывода с поиском и возвратом,
- встроенный механизм сопоставления с образцом,
- простая, но выразительная структура данных с возможностью ее изменения.

Язык Prolog отличается единообразием программ и данных. Данные и программы лишь две различные точки зрения на объекты Prolog. В единой базе данных можно свободно создавать и уничтожать отдельные элементы. Поскольку не существует различия между программой и данными, то можно менять программу во время ее работы. Естественным методом программирования является рекурсия. Prolog является декларативным языком. Это означает, что имея необходимые факты и правила, он может использовать дедуктивные выводы для решения задач программирования.

Prolog программа содержит описание проблемы, которое представляет собой набор логических утверждений в форме фактов, таких как, например

Иван любит Марью.

Петр любит футбол.

или в форме правил, например,

Иван любит X, если Петр любит X

(Иван любит тоже самое, что любит и Петр).

На основе имеющихся правил и фактов Prolog делает выводы. Имея два вышеприведенных факта и одно правило, он придет к выводу, что «Иван любит футбол». А если задать запрос: «Кто любит футбол?», то он найдет все решения этой задачи. При этом он автоматически управляет решением задачи, стараясь во время выполнения программы найти все возможные наборы значений, удовлетворяющие запросу. Поиск производится по всей базе данных, ранее введенной в систему.

Термин база данных используется при объединении набора фактов для совместного их использования при решении некоторой конкретной задачи. Если кроме фактов в описании предметной области содержатся еще и правила, то часто используют термин база знаний.

В языке Prolog используется метод поиска с возвратом, который позволяет ему в случае нахождения одного решения пересмотреть все сделанные предположения еще раз, чтобы определить, не приводит ли новое значение переменной к еще одному решению. Синтаксис Пролога очень короток и прост. Поэтому его проще освоить, чем запутанный синтаксис традиционных языков программирования.

Программа на языке Prolog состоит из множества предложений (фраз). Каждое предложение может быть одного из трех типов: это либо факт, либо правило, либо запрос.

Факт — это фраза без условий, утверждение о том, что соблюдается некоторое отношение. Он записывается как имя предиката, за которым

следует заключенный в скобки список аргументов. Каждый факт должен заканчиваться точкой. Например:

`likes('Иван', 'Марья').`

`likes('Петр', 'футбол').`

Правило — это факт, истинность которого зависит от истинности других фактов. Состоит из головы и тела, разделенных знаком `:`, который читается как «если» и соответствует импликации. Голова правила — это предикат, а тело правила — последовательность предикатов, разделенных запятыми. Правило должно заканчиваться точкой, а запятая в теле правила означает конъюнкцию (логическое И). Например:

`likes('Иван', X) : likes('Петр', X).`

Интуитивный смысл правила состоит в том, что цель, являющаяся головой, будет истинной, если Пролог сможет показать, что все выражения (подцели) в теле правила являются истинными.

Голова : Подцель1, Подцель2, ... , ПодцельN.

Переменные, входящие в предикат головы правила, квантифицируются универсально. Это означает, что правило будет истинным для любых термов, которые удовлетворяют подцелям правила. С другой стороны, переменные, которые входят только в тело правила, квантифицируются экзистенциально. Учитывая изложенное, приведенное ниже правило

`good_student(Name) : student(Name , B) , B > 4.`

можно прочесть так:

Для любого Name, Name является хорошим студентом

ЕСЛИ существует средний балл B, такой что

Name является студентом со средним баллом B

И средний балл B больше 4.

Запрос — это последовательность предикатов, разделенных запятыми или точкой с запятой и завершающаяся точкой. На естественном языке запятая соответствует союзу «и», а на языке математической логики обозначает конъюнкцию. Точка с запятой соответствует союзу «или» и