**Лабораторная работа №1.**

**Цель работы: Изучить основные конструкции языка программирования Пролог для решения задач вычисления функций в экспертных системах.**

1. Теоретические сведения

Пролог – это один из языков программирования, который широко используется для создания баз знаний и экспертных систем. Фундаментальными свойствами языка Пролог являются:

1. механизм вывода с поиском и возвратом;
2. встроенный механизм сопоставления с образцом;
3. простая и легко изменяемая структура данных;
4. отсутствие указателей, операторов присваивания и перехода;
5. естественность рекурсии.

Пролог базируется на сочетании указанных свойств, а удобство его – на естественности его принципов для логических задач, сводящихся к рекурсии.

Программы на Прологе состоит из фактов и правил для получения других фактов и ответов на вопросы. Факты по определению считаются доказанными (истинными).

Конкретному применению логической программы соответствует понятие запроса (цели). Достичь цели – означает показать, что она логически следует из фактов и правил программы. Если вопрос содержит переменные, то система должна найти к тому же конкретные объекты, которые будучи подставленными вместо переменных, обеспечат достижение цели.

В алфавит языка входят следующие символы:

1. прописные и строчные буквы латинского алфавита (A-Z, a-z); регистр букв программы влияет на работу программы;
2. цифры (0-9);
3. спецсимволы: ! @ # $ ( ) & \* - + / < > ; , ? . \ \_ " '

1.2 Структура программы

Пролог – программа включает в себя следующие разделы:

constants – задаются константы;

domains - имена и структуры объектов, используемых в задаче;

predicates – названия отношений между объектами;

clauses – факты и правила, описывающие эти отношения;

goal – описание цели решения.

Для обозначения логических связок для конъюнкции – and, для дизъюнкции – or, следование обозначаются конструкцией «:-» (двоеточие и минус) или слово if. Комментарии обрамляются комбинациями /\*…\*/.

* 1. Домены

Типы данных в Прологе называют доменами. Домен характеризует множество значений, которые может принимать переменная предиката в ходе выполнения программы. Связывание домена с конкретным аргументом (местом) предиката осуществляется в секции predicates. Сами переменные в объявлении не нуждаются. Домены подразделяются на простые и структурированные, стандартные и нестандартные. К стандартным относятся :-symbol – символьная константа (имя), имеет две формы записи:

1. последовательность букв, цифр и знаков препинания, которая начинается со строчной буквы;
2. последовательность символов, заключенная в двойные кавычки (примеры: "apple", "person", "Студент Иванов С.В." ) ;

-string – строка символов (любая последовательность символов, заключенная в двойные кавычки).

-char – отдельный символ между апострофами.

-integer – целое число (в диапазоне –32768 до 32767).

-real – действительное число (пример: -34.567, 0.654, 9.76e+3).

Переменная с именем <<\_>> (анонимная) используется, если значение переменной несущественно. В примере book (Author, \_, \_, \_, \_) речь идет о всех книгах, написанных некоторым <<переменным>> автором, причем пропущенные названия, издательство, место и год рождения роли не играют.

Объявление новых доменов с исползованием стандартных имеет вид

<имя> = <имя\_стандартного\_домена>

Примеры объявлений:

domains

a=integer

fax=symbol

r, dup=real

Введение авторских наименований доменов позволяет внести в программу больше семантики и обеспечивает контроль типов значений переменных, поскольку смешивать в ходе выполнения программы переменные формально различных типов (доменов) нельзя. В Прологе допускается использование структуры, состоящей из нескольких простых или сложных объектов, например:

domains

articles=book(title,author)

author=author(init,fam)

title, author, fam=symbol

init=string

1.4 Предикаты

Раздел predicates должен содержать полный перечень предикатов пользователя, применяемых в программе. Описание предиката включает <имя предиката> и <список доменов его аргументов>:

<имя предиката> (<d1>,<d2>,…<dN>), где <d1>,<d2>,…<dN>- имена стандартных доменов, объявленных в разделе domains.

Предикаты используются для представления как данных, так и правил их обработки. Примеры:

Predicates

Add(integer,integer,real)

lk (fr)

* 1. Утверждения

В разделе clauses описываются утверждения относительно предикатов. Отрицание предиката pr задаётся в форме not(pr) и допустимо только для правой части правила. Существует два типа утверждений: факты и правила. Факт – это имя предиката с заключённым в скобки списком аргументов. Факты используются для констатации того, что выполнено некоторое отношение между объектами. Обычно они записываются относительно предметных констант Однако в фактах могут использоваться и переменные, что позволяет укрупнять факты, уменьшая их общее число и трудоёмкость их поиска в базе данных.

Например, факт mult(x,0,0) объединяет все факты относительно произведения произвольного числа на нуль.

Правило состоит из головной цели – предиката, за которым следует сперва двоеточие с дефисом (:-), а затем тело правила – предикаты (хвостовые цели) разделённые запятыми или точками с запятой. В конце утверждения ставится точка. Предполагается, что переменные в фактах и головных целях утверждения х связаны ??? всеобщности. Переменные в хвостовых целевых утверждениях связаны кванторами существования, а совокупность хвостовых целей рассматривается как логическое произведение.

Пример интерпретации: правило человек (х):- родитель (X,Y), человек (Y). Читается << Всякий Х сын при условии, что родители Х является Y и объект Y – человек >>. В ходе вычислений вместо переменной может быть подставлен другой объект. В этом случае говорят, что переменная конкретизирована. Область действия переменной ограничивается утверждением, и передавать информацию из одного утверждения в другое через свободные переменные нельзя. В Прологе отсутствуют локальные переменные для сохранения промежуточных результатов, поэтому предикаты Пролога дополняются << аргументами-накопителями>>.

* 1. Выражения и стандартные предикаты

В теле правила, помимо объявленных в программе предикатов, могут использоваться стандартные операции сравнения. Стандартные предикаты выполняют различные функции по вводу/выводу различных типов данных, работе с файлами и т.п. Для записи арифметических выражений используются знаки арифметических операций и стандартные функции.

# Таблица 1. Стандартные функции

|  |  |  |
| --- | --- | --- |
| № | Обозначение в математике | Обозначение в Прологе |
| 1 | |x| | abs(x) |
| 2 | ex | exp(x) |
| 3 | sin x | sin(x) |
| 4 | cos x | cos(x) |
| 5 | tg x | tan(x) |
| 6 | arctg x | artan(x) |
| 7 | ln x | ln(x) |
| 8 | log x | log(x) |
| 9 |  | sqrt(x) |
| 10 | xz | exp(z\*ln(x)) |

* 1. Предикаты ввода/вывода

Ввод осуществляется с помощью предикатов:

readchar(<имя переменной 1>, < имя переменной 2>, …) – ввод символьных переменных;

readint(<список переменных>) – ввод целочисленных переменных;

readreal(<список переменных>) – ввод вещественных переменных.

Вывод осуществляется предикатом write(<список вывода>)

Предикат nl обеспечивает переход на новую строку.

1.8 Цель

Раздел goal содержит внутренний запрос к программе. Для такого запроса Пролог осуществляет поиск только первого подходящего решения, при этом вывод организуется программистом. Целей может быть несколько: тогда они перечисляются через запятую.

Пример цели:

goal

synonym(brave, x),

write("A synonym for 'brave' is"),nl,

write(" ' ", x," ' "),nl

2. Задание к работе

Вычислить функцию, заданную согласно варианта (см. таблицу 1).

Написать и отладить программу.

:

# Таблица 1

|  |  |  |
| --- | --- | --- |
| Вариант | Вид выражения | Исходные данные |
| 1 | 2 | 3 |
| 1 |  | x=1,1 |
| 2 |  | x=0,5 |
| 3 |  | x=1,2 |
| 4 |  | x=1,05 |
| 5 |  | х=1,1 |
| 6 |  | х=1,33 |
| 7 |  | х=1,21 |
| 8 |  | х=1,08 |
| 9 |  | х=3,9 |
| 10 |  | х=1,37 |
| 11 |  | х=1,05 |
| 12 |  | х=1,022 |
| 13 |  | x=1,2 |
|  |  | x=1,8 |
|  |  | x=1,01 |
|  |  | x=1,77 |
|  |  | X=1 |
|  |  | x=1.08 |
|  |  | x=1.03 |
|  |  | x=1.19 |
|  |  | x=1.27 |
|  |  | x=1.21 |
|  |  | x=1.37 |
|  |  | x=1.04 |
|  |  | x=1.06 |

3. Контрольные вопросы

1. Основные секции Пролог-программы.
2. Какие стандартные домены имеются в Прологе.
3. Стандартные предикаты ввода/вывода. Описание и параметры.
4. Основные стандартные функции.
5. Алфавит языка Пролог.