Министерство образования Российской Федерации

Московский Государственный Технический Университет им. Н.Э. Баумана

Отчет по лабораторной работе №16

По курсу «Функциональное и логическое

программирование»

Студент Медведев А.В.

Группа ИУ7-62 Преподаватель Толпинская Н.Б.

# Теоретическая часть

1. Основные принципы логического программирования

Пролог является языком нечистых предикатов. Предикат – логическая формула от одного или нескольких аргументов. Программа на алгоритмическом языке является последовательностью инструкций. Prolog же содержит только описание задачи, для решения которой происходит поиск по описанию с использованием механизма поиска с возвратом и унификация.

Основной структурной единицей языка является терм

Задача языка – доказать, является ли заданное целевое утверждение следствием из имеющихся фактов и правил. Решаемая задача представляется в виде совокупности утверждений. Все истинные утверждения хранятся в базе знаний, на основе которой решается задача (т.е для решения любой задачи используются имеющиеся знания).

Используются базовые домены (описание имени структуры объектов). Пролог имеет 6 базовых типов данных: симолы, целые числа, действительные числа, стороки, символьные имена, факты.

Еще оним принципом является

1. Основны структуры языка

Основной Элемент языка-терм

Терм:

* Простой
  + Константа (с маленькой буквы)
    - Символ
    - число
  + переменная (с большой буквы)
    - Именованная
    - Анонимная
* Составной (пример f(t1,t2,…,tn) Где F-функтор (имя отношения между объектами))

1. Особенность использования переменных

Именованные переменные уникальны в предикатах одного предложения, анонимные уникальны везде. Анонимные переменные не возвращают значение. Переменной можно обозначить любой объект. При описании переменная может потерять свое значение, но потом его можно вернуть.

1. Структура программы

Программа состоит из разделов

* 1. Domains – описание имен и структур объектов (не обязателен)
  2. Predicates – описание предикатов (названий отношений между объектами)
  3. Clauses – база знаний
  4. Goal – Раздел целевых утверждений

Программа состоит из предложений

* Факт (безусловная истина, формулируется составным термом)- частный случай правил.
* Правила (условная истина, способ порождения новых фактов на основе имеющихся)
  + Пример:

A:- B1, B2,B3. (правило)

A – заголовок , B1, B2,B3.- тело

Вопрос:

* Конъюктивидный (B1, B2, B3)
* Дизъюктивный (B1; B2; B3).

1. Понятие процедуры

Процедура - совокупность правил, которые имеют одни и те же функторы и одну и ту же арность, а заголовки одинаковые. Процедуры определяют отношения, называемые предикатами. Процедуры описывают одно отношение, которое невозможно сформулировать в виде одного предложения.  
Пример процедур:  
f(t1, t2, ..., tn)  
f(t1, t2, ..., tn)

1. Понятие домена

Домен - описание множестве значений, способ отличить природу объекта. Домены подразделяются на стандартные/базовые (например, целые числа, строки, символы) и на составные (создаются в программе на основе базовых). Домены составляют необязательный раздел программы на Прологе.ы

domains

NAME=symbol

NUM=string

CITY=string

STREET=string

HOUSE=integer

address=address(CITY,STREET,HOUSE)

person = person(NAME, address,NUM)

articles =

book(string Title, string Author) ;

horse(string Name) ;

boat ;

bankbook(real Balance).

# Практическая часть

Исходный код программы

DOMAINS

int\_list = integer\*

PREDICATES

lengthtr(int\_list, integer, integer)

length(int\_list, integer)

sumtr(int\_list, integer, integer)

removeone(int\_list,integer,int\_list)

removebypos(int\_list,integer,int\_list)

removeMember(int\_list,integer,int\_list)

removeMoreThen(int\_list,integer,int\_list)

removeEvenPos(int\_list,integer,int\_list)

mergelist(int\_list,int\_list,int\_list ).

member1(integer,int\_list)

listtoset(int\_list,int\_list)

CLAUSES

lengthtr([],M,M).

lengthtr([\_|T],M, R):- M1=M+1, lengthtr(T,M1,R).

length([],0).

length([\_|T],L):-length(T,L1), L=L1+1.

sumtr([],M,M).

sumtr([H|T],M,R):- M1=M+H, sumtr(T,M1,R).

removeone([X|Tail],X,Tail).

removeone([Y|Tail],X,[Y|Tail1]):- removeone(Tail,X,Tail1).

removebypos([\_|T],0, T).

removebypos([H|T], POS, [H|Z]):- POS1=POS-1, removebypos(T,POS1,Z).

removeMember([],\_,[]) :- !.

removeMember([X|Xs], X, Y) :- !, removeMember(Xs, X, Y).

removeMember([T|Xs], X, [T|Y]) :- removeMember(Xs, X, Y).

removeMoreThen([],\_,[]) :- !.

removeMoreThen([H|T],X, PTail):- H > X, !, removeMoreThen(T,X, PTail).

removeMoreThen([H|T],X, [H|PTail]):- removeMoreThen(T,X, PTail).

removeEvenPos([],\_,[]) :- !.

removeEvenPos([\_|T], POS, PTail):- POS1=POS+1, POS mod 2 = 1,!,removeEvenPos(T,POS1, PTail).

removeEvenPos([H|T], POS, [H|L]):- POS1=POS+1, removeEvenPos(T ,POS1, L).

mergelist([],L,L ).

mergelist([H|T],L,[H|M]):-mergelist(T,L,M).

member1(X,[H|\_]) :- X = H,!.

member1(X,[\_|T]) :- member1(X,T).

listtoset([],[]).

listtoset([H|T],C) :- member1(H,T),!, listtoset(T,C).

listtoset([H|T],[H|C]) :- listtoset(T,C).

GOAL

%lengthtr([1,2,3,4,5],0,R).

%length([1,2,3,4,5],Z).

%sumtr([1,2,3,4,5],0,R).

%removeone([1,2,3,4,5],3,R).

%removeone([1,3,2,3,5],3,R).

%removeone([1,3,2,3,5],1,R).

%removeone([1,3,2,3,5],5,R).

%removebypos([0,1,2,3,4,5], 2,X).

%removebypos([0,1,2,3,4,5], 0,X).

%removebypos([0,1,2,3,4,5], 5,X).

%removeMember([0,1,2,3,4,5],2, X).

%removeMember([1,3,2,3,5],3, X).

%removeMember([0,1,2,3,4,5],6,X).

%removeMoreThen([1,3,2,3,5],2, X).

%removeMoreThen([1,3,2,3,5],3, X).

%removeEvenPos([1,3,2,3,5],0, X).

%mergelist([1,2],[3,4,5],M).

listtoset([1,2,3,2,4,2,5],M).