Федеральное государственное бюджетное образовательное учреждение высшего образования «Чувашский государственный университет им. И.Н. Ульянова»

Кафедра вычислительной техники

Лабораторная работа №1

по дисциплине функциональное и логическое программирование

Основы логического программирования

Вариант №16

Выполнил: Кузнецов Даниил Андреевич студент группы ИВТ-42-23

Проверил: доцент Обломов И.А.

Чебоксары, 2025

**Краткие теоретические сведения**

Существуют 3 вида предложений в языке Prolog: факты, правила, вопросы.

Формат задания факта: имяПредиката(арг1, арг2, ...).

Имя предиката обязательно должно начинаться со строчной буквы. Аргументы должны быть перечислены через запятую и тоже должны начинаться со строчной буквы. В качестве аргументов также могут выступать и числовые константы.

Факт всегда является истинным и не требует доказательства.

Формат задания правила: имяПравила :- функтор1(арг1, арг2, ...), функтор2(арг1, арг2, ...).

Имя правила также начинается со строчной буквы. Функторами могут являться как факты, так и другие правила. Функторы могут быть соединены с помощью запятых (логическое «И») или точек с запятыми (логическое «ИЛИ»).

Правило необходимо доказать, прежде чем принять за истину. Язык Prolog доказывает правила автоматически.

Формат задания вопроса: ?- функтор(арг1, арг2, Перем), write(Перем).

В качестве функтора можно использовать любое имя правила или факта, написанное со строчной буквы. В вопросе могут участвовать переменные, которые пишутся с прописной буквы. Значение переменных конкретизируется после выполнения вопроса на основе фактов и правил. После этого можно добавить вывод с помощью команды write(значение). Если значение будет найдено, то Prolog система выведет сообщение «Yes» и значение переменной, в противном случае – сообщение «No».

**Задание к лабораторной работе**

1. Используя предикаты parent(name,name), man(name), woman(name), married(name,name), записать факты, описывающие Вашу семью. Записать не менее 8 правил вывода для любых родственных отношений в Вашей семье, например, мать, отец, сестра, брат, племянник, племянница, тетя, дядя, внук, внучка, бабушка, дедушка, двоюродный брат, двоюродная сестра и т.д.

2. Написать факты и правила, моделирующие комбинационную схему НЕ-И-ИЛИ.

**Описание родственных связей в виде дерева**

german

anatoliy

zinaida

nikolay

natalia

andrey

daniil

**Описание комбинационной схемы**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

&

&

1

X1

X2

X3

X4

Y

**Тексты программ**

% lab1\_1.pro

parent(german, anatoliy).

parent(nikolay, zinaida).

parent(anatoliy, natalia).

parent(zinaida, natalia).

parent(natalia, daniil).

parent(andrey, daniil).

man(german).

man(nikolay).

man(anatoliy).

man(andrey).

man(daniil).

woman(zinaida).

woman(natalia).

maried(zinaida, anatoliy).

maried(natalia, andrey).

mother(X, Y) :- parent(X, Y), woman(X).

father(X, Y) :- parent(X, Y), man(X).

daughter(X, Y) :- parent(Y, X), woman(X).

son(X, Y) :- parent(Y, X), man(X).

grandmother(X, Y) :- parent(X, Z), parent(Z, Y), woman(X).

grandfather(X, Y) :- parent(X, Z), parent(Z, Y), man(X).

sister(X, Y) :- parent(Z, X), parent(Z, Y), woman(X).

brother(X, Y) :- parent(Z, X), parent(Z, Y), man(X).

uncle(X, Y) :- brother(X, Z), parent(Z, Y).

aunt(X, Y) :- sister(X, Z), parent(Z, Y).

nephew(X, Y) :- (brother(Z, Y); sister(Z, Y)), parent(Z, X), man(X).

niece(X, Y) :- (brother(Z, Y); sister(Z, Y)), parent(Z, X), woman(X).

% lab1\_2.pro

and\_(0, 0, 0).

and\_(0, 1, 0).

and\_(1, 0, 0).

and\_(1, 1, 1).

or\_(0, 0, 0).

or\_(0, 1, 1).

or\_(1, 0, 1).

or\_(1, 1, 1).

not\_(0, 1).

not\_(1, 0).

not\_and(X1, X2, Y) :- not\_(X1, Z1), not\_(X2, Z2), and\_(Z1, Z2, Y).

not\_and\_or(X1, X2, X3, X4, Y) :-

not\_and(X1, X2, Z1), not\_and(X3, X4, Z2),

or\_(Z1, Z2, Y).

?- not\_and\_or(X1, X2, X3, X4, Y),

write(X1), write(" "),

write(X2), write(" "),

write(X3), write(" "),

write(X4), write(" "),

write(Y), write(nl).

**Выводы**

Изучены предложения на языке Prolog: факты, правила вопросы.