Федеральное государственное бюджетное образовательное учреждение высшего

образования

«Чувашский государственный университет им. И.Н. Ульянова»

Кафедра вычислительной техники

Лабораторная работа № 1

Основы логического программирования

Вариант 9

Выполнил: Иванов В.С.

студент группы ИВТ-41-22

Проверил: кандидат технических наук

Обломов Игорь Александрович

Чебоксары, 2024

Цель работы: изучение основ логического программирования.

Пролог – это язык программирования, предназначенный для обработки символьной информации. Он очень хорошо приспособлен для решения задач, в которых фигурируют некоторые объекты и отношения между ними.

Предложения в Прологе подразделяются на следующие виды:

* факт;
* правило;
* вопрос.

Факт представляет собой некое утверждение, которое всегда истинно.

Правило – более сложная конструкция, которую необходимо доказать, вывести из программы или согласовать с программой.

Вопрос является основным средством общения пользователя с Пролог-программой, т.е. своеобразным интерфейсом с программой.

Отметим несколько важных особенностей Пролог-программы. Во-первых, любое предложение заканчивается символом точки. Во-вторых, имена отношений и имена их аргументов необходимо писать латинскими символами.

Имена людей в отношении начинаются со строчной буквы, поскольку они представляют собой *атомы*.

Отличительной особенностью переменных в Прологе является то, что они всегда начинаются с прописного символа или с символа подчеркивания. Другая особенность переменных – их лексический диапазон ограничен рамками того предложения, в котором они присутствуют. Другими словами, если в одной и той же программе есть переменные с одним и тем же именем, то это суть разные объекты.

Вопрос является основным средством общения пользователя с Пролог-программой, т.е. своеобразным интерфейсом с программой. В общем виде вопрос записывается в следующей форме:

?-вопрос.

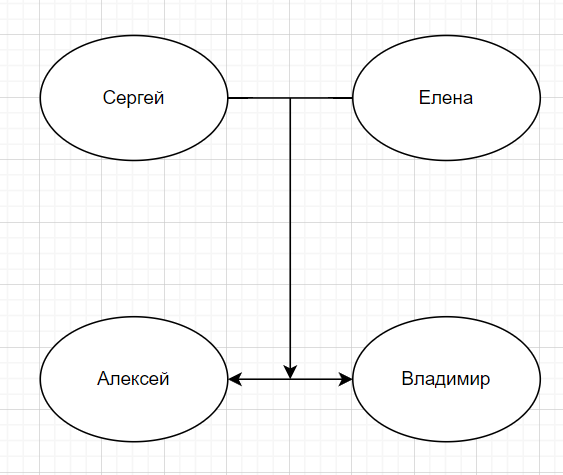
Здесь «вопрос» представляет собой имя отношения между объектами, например, применительно программы о родственных отношениях, Пролог-системе можно задать следующий вопрос:

?-родитель(мария, петр).

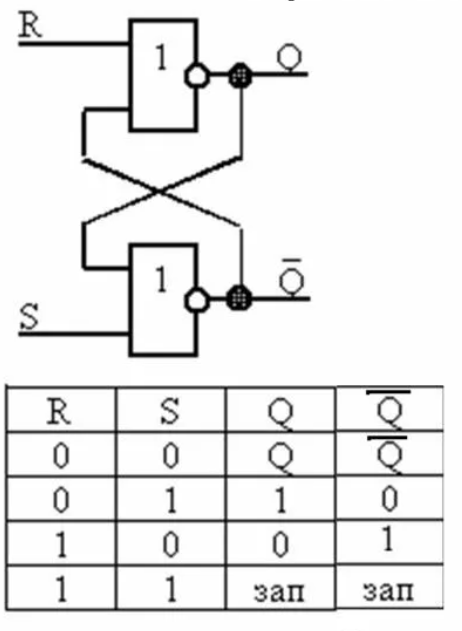
**Задание:**

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 логический элемент или комбинационную схему согласно Вашему варианту, представленному в табл. 3.( Асинхронный RS-триггер).

****

Дерево родственных отношений.



Модель асинхронного RS-триггера

Текст программы:

%1 ex

parent("Sergey", "Vladimir").

parent("Sergey","Alex").

parent("Helen","Vladimir").

parent("Helen","Alex").

man("Sergey").

man("Alex").

man("Vladimir").

woman("Helen").

married("Sergey", "Helen").

father(X,Y):-

parent(X,Y),man(X), write(X).

mother(X,Y):-

parent(X,Y),woman(X),write(X).

brother(X,Y):-

parent(A,X), parent(A,Y), X\=Y,man(X),write(X).

son(X,Y):-

parent(X,Y), man(Y),write(Y).

?-father(X, "Vladimir"), nl.

?-mother(X, "Alex"), nl.

?-brother("Alex", "Vladimir"), nl.

?-son("Helen", X), nl.

?-parent(X, "Vladimir"), write(X), nl.

?-parent(X, "Alex"), write(X), nl.

?-married(X, "Helen"), write(X), nl.

?-married("Sergey", X), write(X), nl.

%2 ex

or\_(0,0,0).

or\_(0,1,1).

or\_(1,0,1).

or\_(1,1,1).

no(1,0).

no(0,1).

or\_no(X,Y,Z):-or\_(X,Y,W), no(W,Z).

rs\_async(S, R, Q, Not\_Q):- or\_no(S,Q,Not\_Q), or\_no(R, Not\_Q, Q).

%?-read(S), read(R), rs\_async(S,R,Q,Not\_Q), write(Q), write(Not\_Q).

?- write("S R Q Not\_Q"), nl.

?- rs\_async(0, 0, Q, Not\_Q), write("0 0 "), write(Q), write(" "), write(Not\_Q), nl.

?- rs\_async(0, 1, Q, Not\_Q), write("0 1 "), write(Q), write(" "), write(Not\_Q), nl.

?- rs\_async(1, 0, Q, Not\_Q), write("1 0 "), write(Q), write(" "), write(Not\_Q), nl.

?- rs\_async(1, 1, Q, Not\_Q), write("1 1 "), write(Q), write(" "), write(Not\_Q), nl.

Вывод: изучил основы логического программирования.