Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное

образовательное учреждение высшего образования

‹‹Уфимский университет науки и технологий››

Кафедра ВМиК

**Отчёт по лабораторной работе №5**

по дисциплине «Логическое программирование»

Выполнил:студент группы ПРО-331Б

Семенов Л.А.

Проверил:

старший преподаватель

Шакирзянов А.А.

Уфа – 2024

**Цель работы**:

Изучение основ программирования на языке Prolog.

**Ход работы**

**1 задача**:

Условие: Три друга заняли первое, второе и третье места в соревнованиях универсиады. Друзья – разной национальности, зовут их по-разному и любят они разные виды спорта.

Иван предпочитает баскетбол и играет лучше, чем американец. Аргентинец Месси играет лучше теннисиста. Игрок в футбол занял первое место.

Кто является русским? Каким видом спорта занимается Ричард?

**Код программы:**

nation(russian).

nation(american).

nation(argentinian).

nation(french).

nation(chinese).

sport(basketball).

sport(tennis).

sport(football).

sport(swimming).

sport(athletics).

name(ivan).

name(messi).

name(richard).

name(pierre).

name(chen).

prize(1).

prize(2).

prize(3).

prize(4).

prize(5).

unique([]).

unique([X|Xs]) :- \+ member(X, Xs), unique(Xs).

solve(Mans):-

Mans = [man(X, XNat, XSp, XP), man(Y, YNat, YSp, YP), man(Z, ZNat, ZSp, ZP), man(W, WNat, WSp, WP), man(K, KNat, KSp, KP)],

name(X), name(Y), name(Z), name(W), name(K), unique([X, Y, Z, W, K]),

nation(XNat), nation(YNat), nation(ZNat), nation(WNat), nation(KNat), unique([XNat, YNat, ZNat, WNat, KNat]),

sport(XSp), sport(YSp), sport(ZSp), sport(WSp), sport(KSp), unique([XSp, YSp, ZSp, WSp, KSp]),

prize(XP), prize(YP), prize(ZP), prize(WP), prize(KP), unique([XP, YP, ZP, WP, KP]),

% Иван играет в баскетбол

member(man(ivan, \_, basketball, IvanPos), Mans),

% Иван играет лучше чем американец

member(man(\_, american, \_, AmerPos), Mans), IvanPos < AmerPos,

% Месси - аргентинец

member(man(messi, argentinian, \_, MessiPos), Mans),

% Месси играет лучше теннисиста

member(man(\_, \_, tennis, TennPos), Mans), MessiPos < TennPos,

% игрок в футбол занял первое место

member(man(\_, \_, football, 1), Mans).

start(Solve) :-

solve(Men),

member(man(richard, russian, SportRichard, \_), Men),

member(man(\_, american, \_, 2), Men),

member(man(\_, \_, tennis, \_), Men),

member(man(\_, \_, football, \_), Men),

writeln('1. Richard is Russian.'),

writeln('2. Richard plays the following sport:'),

writeln(SportRichard),

writeln('3. The man who is American is in 2nd place.'),

writeln('4. The man who plays tennis and football are:'),

member(man(TennisPlayer, \_, tennis, \_), Men),

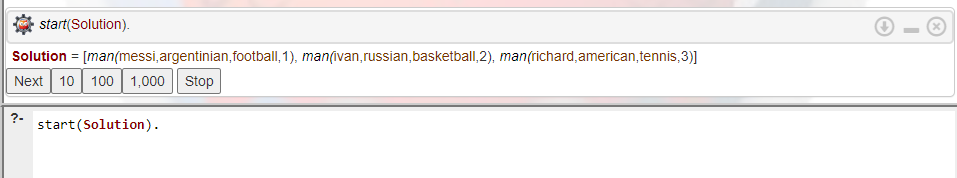
member(man(FootballPlayer, \_, football, \_), Men),

writeln(TennisPlayer),

writeln(FootballPlayer),

fail.

**Результат работы:**



**2 задача**:

Условие:

1. Если А виновен и В невиновен, то С виновен;  
2. С никогда не действует в одиночку;  
3. А никогда «не ходит на дело» вместе с С;  
4. Никто, кроме А, В и С, в преступлении не замешан, и по крайней мере один из этой тройки виновен.

Сержант поскреб в затылке и сказал: «Боюсь, что я смогу извлечь из этих фактов не слишком много, сэр. А вы можете, опираясь на них, доказать, кто из трех подозреваемых виновен и кто не виновен?»

«Не могу, – признался Крэг, – но, чтобы выдвинуть неопровержимое обвинение против одного из них, материала вполне достаточно». Чья виновность не вызывает сомнений?

**Код программы:**

возможное\_имя(а).

возможное\_имя(в).

возможное\_имя(с).

возможная\_виновность(виновен).

возможная\_виновность(невиновен).

генерация\_гипотезы(Гипотеза):-

возможная\_виновность(ВиновностьА),

возможная\_виновность(ВиновностьВ),

возможная\_виновность(ВиновностьС),

Гипотеза = [

подозреваемый(а, ВиновностьА),

подозреваемый(в, ВиновностьВ),

подозреваемый(с, ВиновностьС)

].

%1. Если А виновен и В невиновен, то С виновен

проверка\_условия(1, Гипотеза):-

входит\_в(подозреваемый(а, виновен), Гипотеза),

входит\_в(подозреваемый(в, невиновен), Гипотеза),

входит\_в(подозреваемый(с, виновен), Гипотеза).

проверка\_условия(1, Гипотеза):-

not(входит\_в(подозреваемый(а, виновен), Гипотеза)).

проверка\_условия(1, Гипотеза):-

not(входит\_в(подозреваемый(в, невиновен), Гипотеза)).

%2. С никогда не действует в одиночку;

проверка\_условия(2, Гипотеза):-

%%% значит есть есть еще один виновный, но это не С

входит\_в(подозреваемый(с, виновен), Гипотеза),

входит\_в(подозреваемый(Другой, виновен), Гипотеза),

not(Другой = с).

% 3. А никогда «не ходит на дело» вместе с С;

проверка\_условия(3, Гипотеза):-

%%% если С - виновен - то проверим что А невиновен

входит\_в(подозреваемый(с, виновен), Гипотеза),

входит\_в(подозреваемый(а, невиновен), Гипотеза).

проверка\_условия(3, Гипотеза):-

%%% но если С невиновен - то на А не накладываются ограничения

not(входит\_в(подозреваемый(с, виновен), Гипотеза)).

% 4. Никто, кроме А, В и С, в преступлении не замешан, и по крайней мере

% один из этой тройки виновен.

проверка\_условия(4, Гипотеза):-

входит\_в(подозреваемый(\_Виновный, виновен), Гипотеза).

входит\_в(X, Y):- member(X, Y).

проверка\_гипотезы(Гипотеза):-

проверка\_условия(1, Гипотеза),

проверка\_условия(2, Гипотеза),

проверка\_условия(3, Гипотеза),

проверка\_условия(4, Гипотеза).

%Проверка на то, существует ли хоть одна подходящая под условия задачи гипотеза,

%такая что в ней X виновен

проверка\_возможной\_виновности(Имя, Виновность):-

генерация\_гипотезы(Гипотеза),

проверка\_гипотезы(Гипотеза),

входит\_в(подозреваемый(Имя, Виновность), Гипотеза).

%Проверка на то, что существует хоть один вариант генерации при котором X виновен

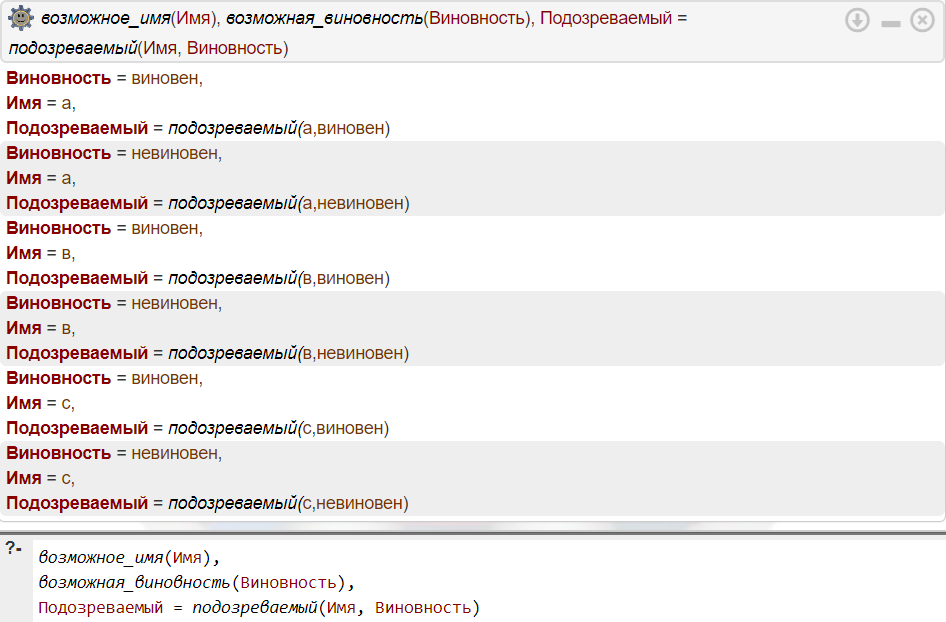
%и что не существует ни одного варианта когда он не виновен.

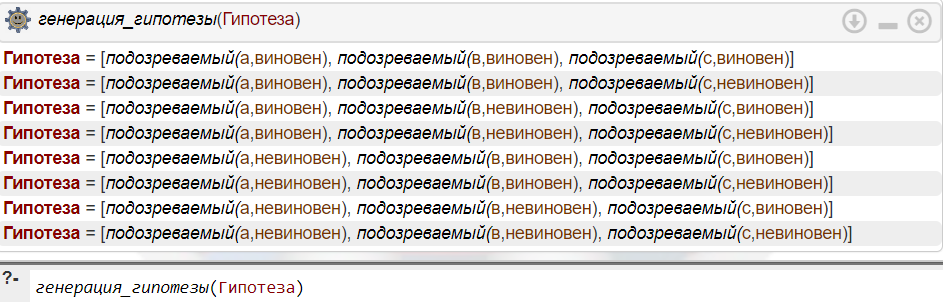
поиск\_виновного(Имя):-

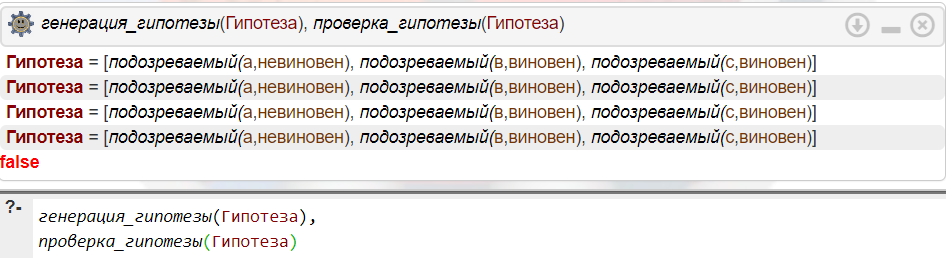
проверка\_возможной\_виновности(Имя, виновен),

not(проверка\_возможной\_виновности(Имя, невиновен)).

**Результат работы:**







**3 задача**:

Условие: Лабиринт представляет собой систему комнат, соединенных между собой переходами. В лабиринте имеется вход и выход, а также комната с золотым кладом. Кроме того, имеются комнаты, запрещенные для посещений: комната монстров и комната разбойников.

1. Найти путь в лабиринте от входа до выхода, не посещая дважды одной и той же комнаты;
2. Найти путь с посещением золотой комнаты;
3. Найти путь, избегающий запрещенных к посещению комнат.

**Код программы**

:- use\_rendering(graphviz).

%Определение графа

graph(Graph) :-

findall(edge(From, To), edge(From, To), Edges),

gvgraph(Edges, Graph).

%Преобразование графа в формат Graphviz

gvgraph(Edges, digraph([rankdir='LR', size=5|Statements])) :-

phrase(edges(Edges, \_{id:1}, \_), Statements).

%Обработка ребер графа

edges([], Options, Options) --> [].

edges([edge(From, To)|T], Options0, Options) -->

[ From -> To ],

edges(T, Options0, Options).

%Определение рёбер на графе

edge(shochi, out).

edge(shochi, perm).

edge(shochi, bulgakovo).

edge(ufa, out).

edge(ufa, perm).

edge(ufa, sterlitamak).

edge(bulgakovo, out).

edge(bulgakovo, moscow).

edge(bulgakovo, shochi).

edge(sterlitamak, perm).

edge(sterlitamak, moscow).

edge(sterlitamak, out).

%Определение предиката поиска в глубину

dfs(Node, Node, \_, [Node]). %Базовый вариант: когда достигнут конечный узел

dfs(Start, End, Visited, [Start|Path]) :-

edge(Start, Next),

\+ member(Next, Visited), % Убеждение в том, что мы раньше не посещали этот узел

dfs(Next, End, [Next|Visited], Path). %Рекурсивный поиск со следующего узла

%graph(Graph)

% Найди путь в лабиринте от входа до входа,не посещая дважды один и тот же город;

% 1- dfs(in, out, [], Path)

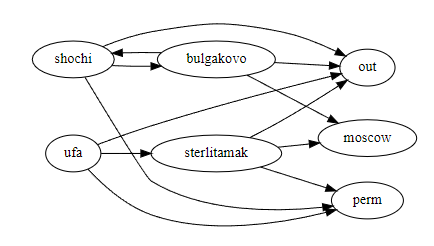
% Найти путь с посещением конкретного города;

% 2- dfs(in, out, [], Path), member(shakshi, Path)

% Найти путь,избегающий запрещенных к посещению комнат.

% 3- dfs(in, out, [], Path), \+ member(sterlitamak, Path), \+ member(magnitogorsk, Path)

**Результат работы:**

****

**Вывод.**

В данной лабораторной работы были освоены навыки на языке программирования Prolog и применения их на практике. Были рассмотрены задачи на установление соответствия, метод генерации гипотезы, задача поиска в пространстве состояний и визуализации. Предметная область – карта городов.