Министерство науки и образования РФ

Федеральное государственное автономное образовательное

учреждение высшего профессионального образования

«Санкт-Петербургский государственный электротехнический

университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»

(СПбГЭТУ «ЛЭТИ»)

Кафедра вычислительной техники

Отчёт

по лабораторной работе № 1

по дисциплине «Интеллектуальные системы»

Тема: «Разработка и отладка простой программы на языке Visual Prolog»

Выполнили студенты гр. 8307 Николаев Д.Е.

Репин С.А.

Такшеев А.Д.

Преподаватель Родионов С.В.

1. Результаты выполнения программы DOG.PRO.

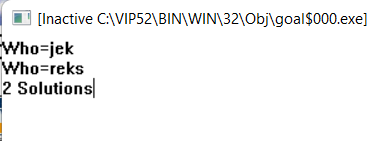


Рисунок 1.1 – результат выполнения программы DOG.PRO

Результатом программы стал вывод списка всех собак при выполнении запроса

*dog(Who).*

В секции предложений были записаны факты

*dog(“reks”).*

*parent(“jek”,”reks”).*

Первый факт говорит о том, что reks – собака, а второй описывает родителя reks. Если посмотреть на правило вывода, то получится что jek тоже собака.

1. Трасса выполнения программы DOG.PRO с пояснениями.

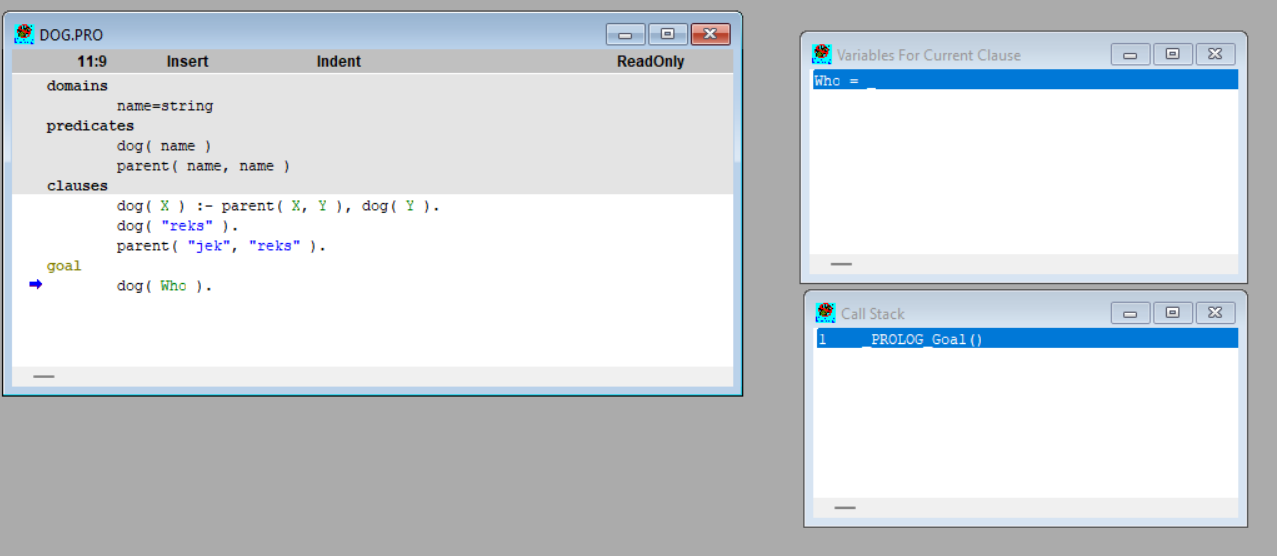


Рис. 2.1 – старт трассы

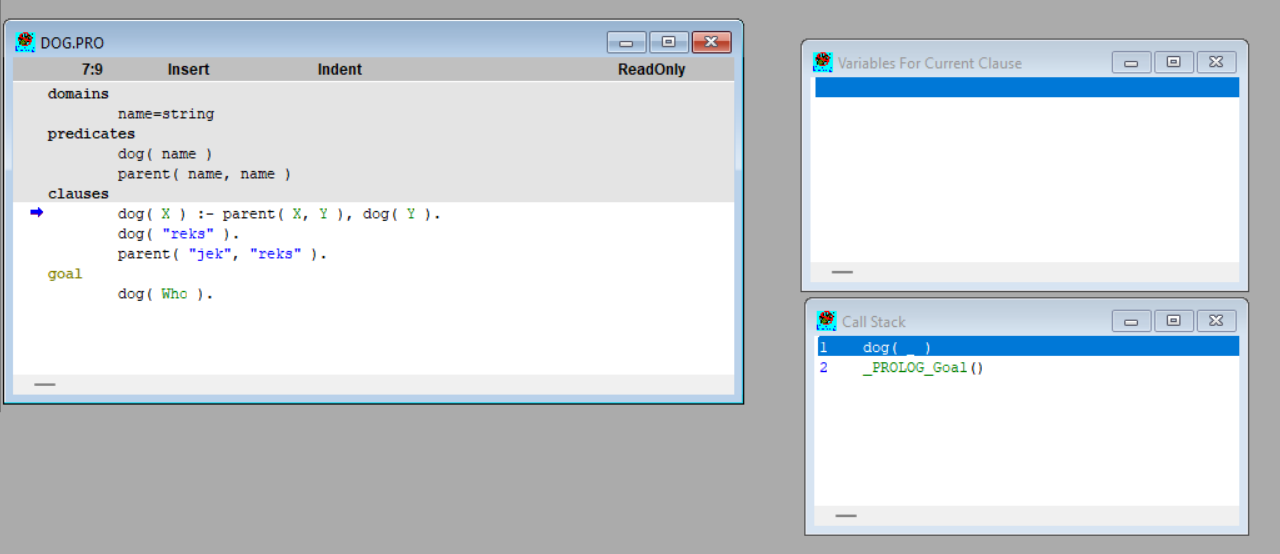


Рис.2.2 – вызываем любое правило/ факт с предикатом dog

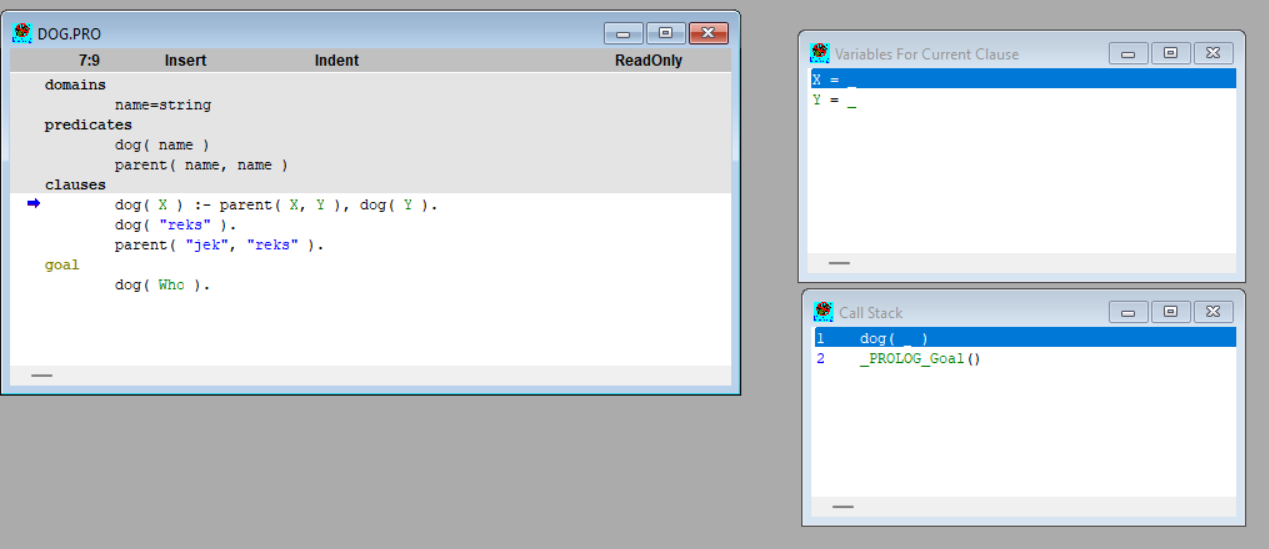


Рис. 2.3 – вызвали правило, чтобы проверить является ли X собакой, надо проверить является ли X родителем для собаки Y

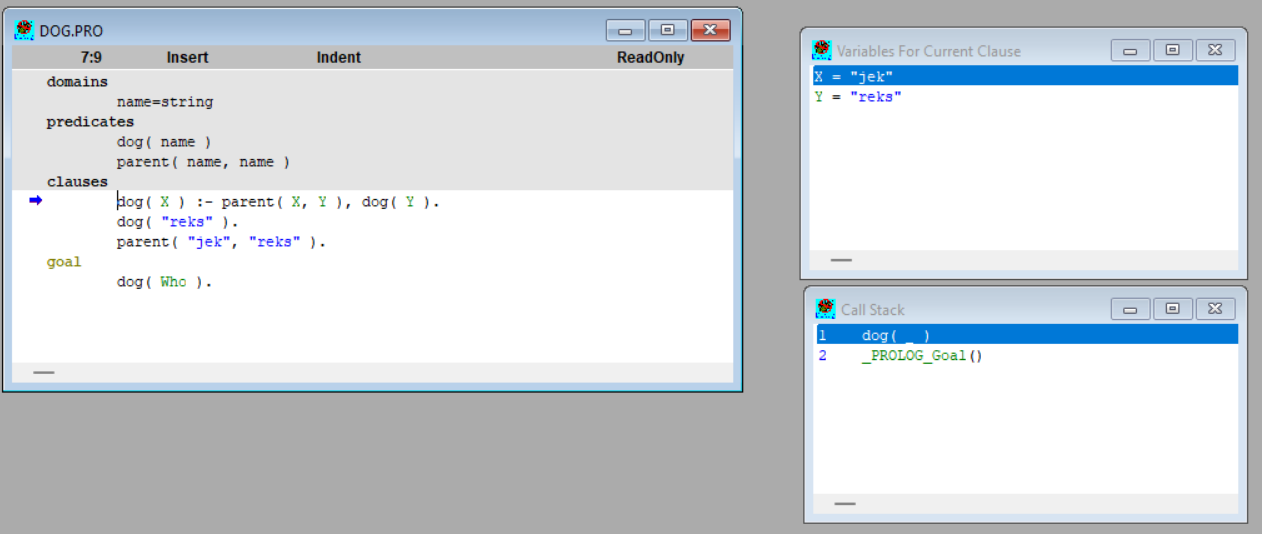


Рис. 2.4 – нашли факт о том что jek родитель reks

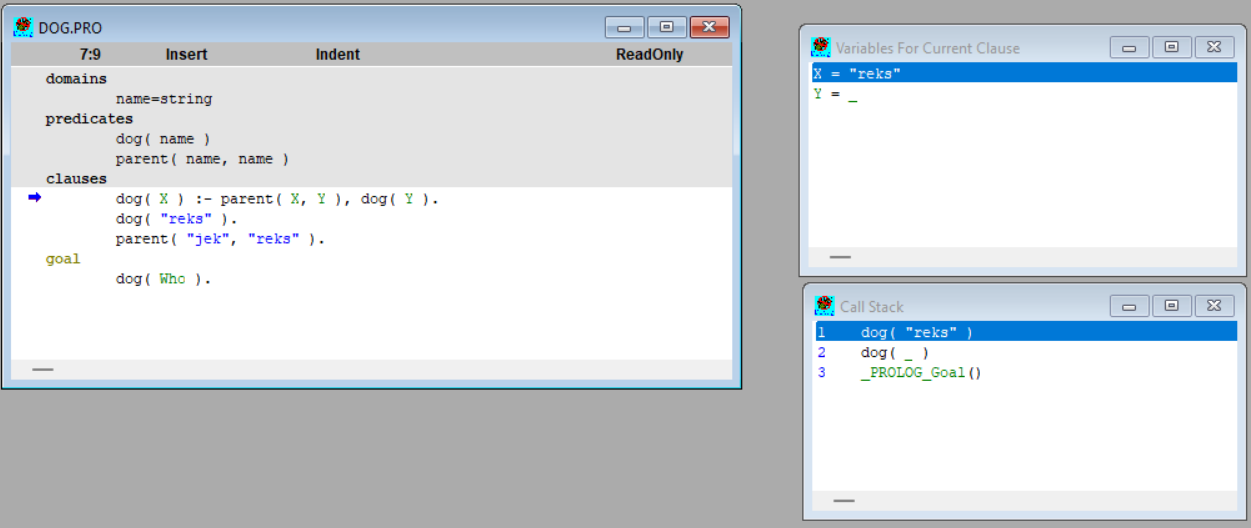


Рис 2.5 – теперь нужно проверить является ли reks родителем для какой-либо собаки



Рис 2.6 – reks родителем не является, но является собакой (есть факт об этом)

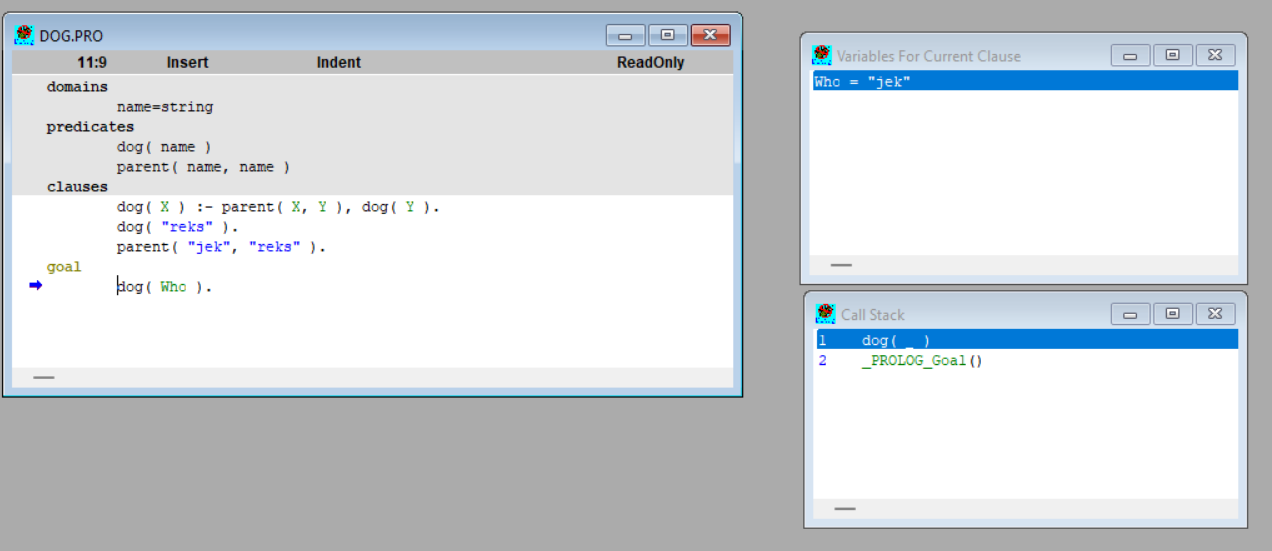
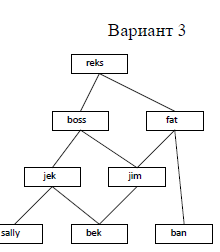


Рис. 2.7 – так ка reks собака, а jek её родитель, то jek тоже является собакой

Больше нет фактов, которые могут активировать правила, поэтому программа завершается.

1. Текст программы DOG1.PRO, трассы выполнения запросов и объяснение результатов их выполнения.

Вариант 3



Текст программы:

% Номер варианта - 3

domains

name=string

predicates

dog( name )

parent( name, name )

grandparent( name, name )

clauses

dog( X ) :-

parent( X, Y ),

dog( Y ).

dog( "sally" ).

dog( "bek" ).

dog( "ban" ).

dog( "fat" ).

parent( "jek", "sally" ).

parent( "jek", "bek" ).

parent( "jim", "bek" ).

parent( "boss", "jek" ).

parent( "boss", "jim" ).

parent( "fat", "ban" ).

parent( "fat", "jim" ).

parent( "reks", "boss" ).

parent( "reks", "fat" ).

grandparent( X, Z ) :-

parent( X, Y ),

parent( Y, Z ).

goal

% Кто является собакой ?

dog( Who ).

% Кто является родителем ?

% parent( Who, \_ ).

% Кто является внуком (внучкой) ?

% grandparent( \_, Who ).

% Bek - собака ?

% dog( "bek" ).

% Кто родитель собаки Bek ?

% parent( Who, "bek" ).

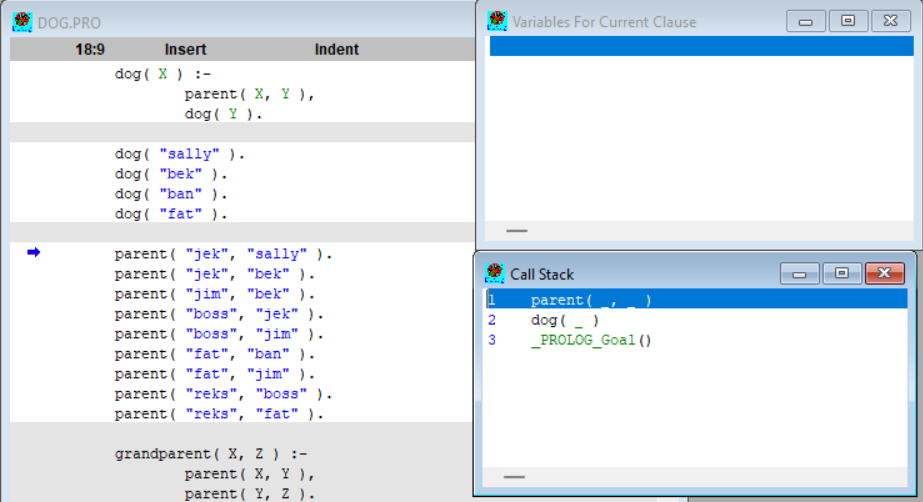
% Кому Bek является родителем ?

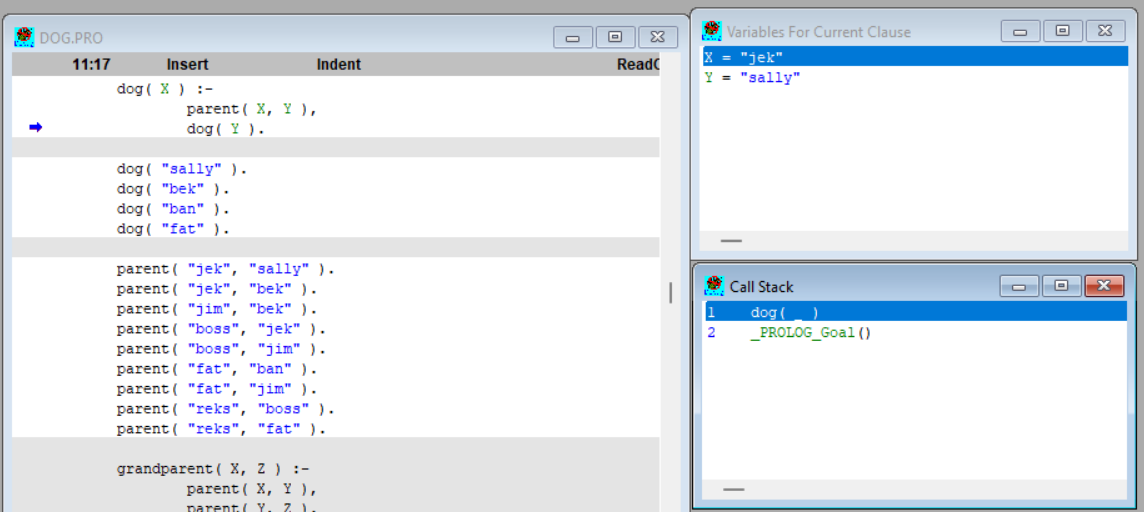
% parent( "bek", Who ).

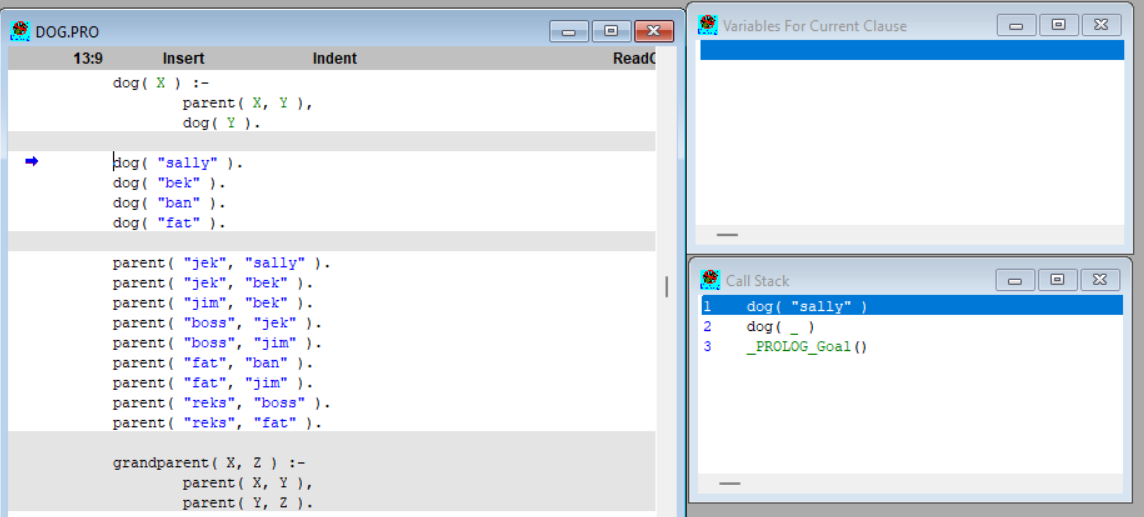
Запрос Кто является собакой?

Трасса:









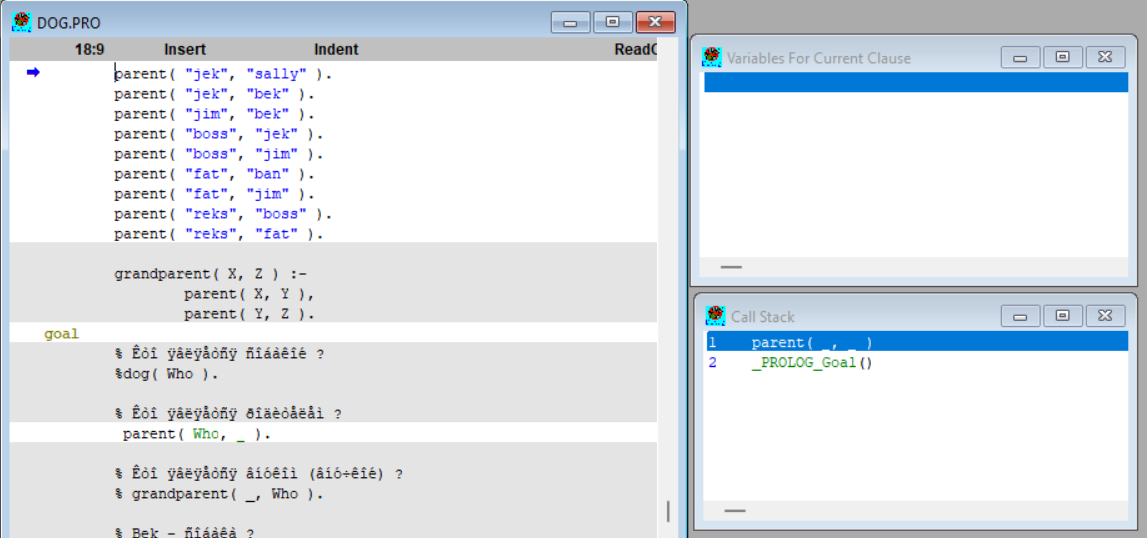
Результат:

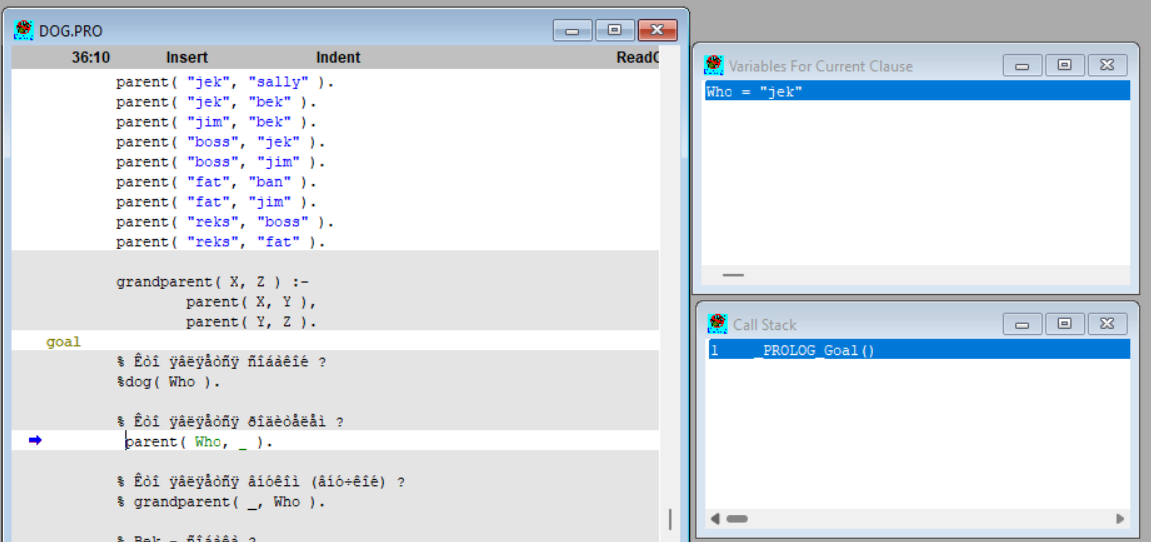


Трасса аналогична выполнению программы DOG.PRO. Результатом стали 18 решений о том кто является собакой. Собаки повторяются так как в программе при нахождении собаки факт об этом не записывается, поэтому каждый раз нам необходимо узнавать кто является собакой родителем (самой старшей собакой).

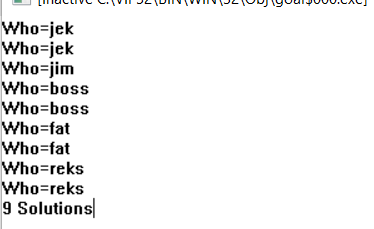
Запрос Кто является родителем?

Трасса:





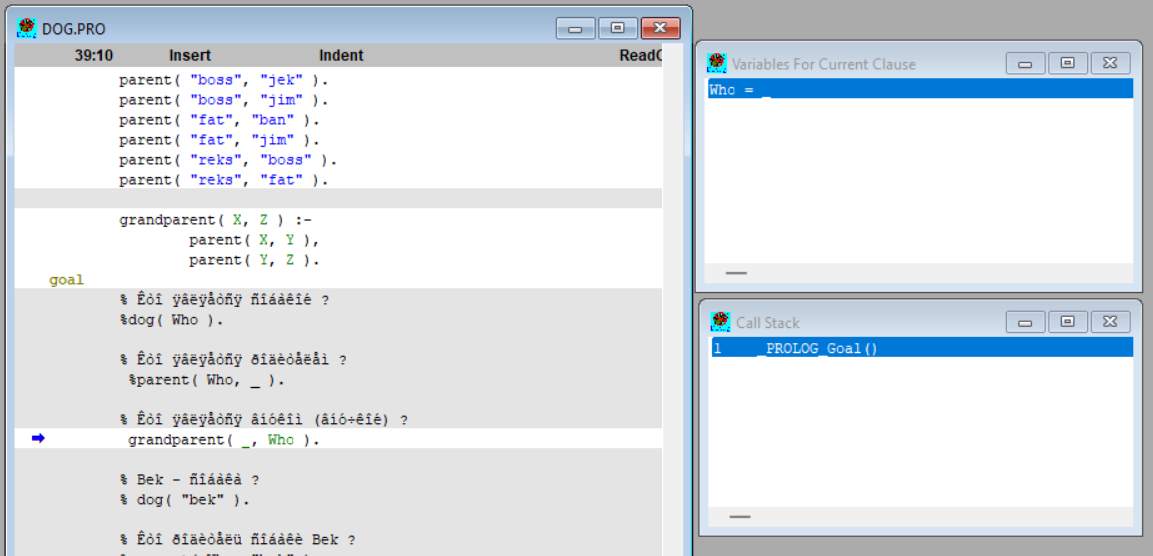
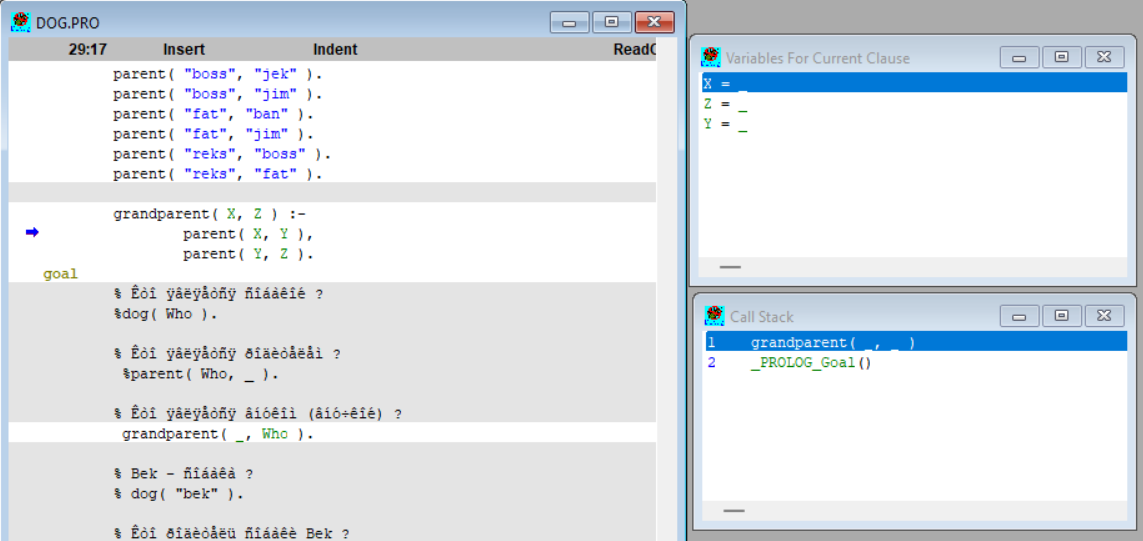
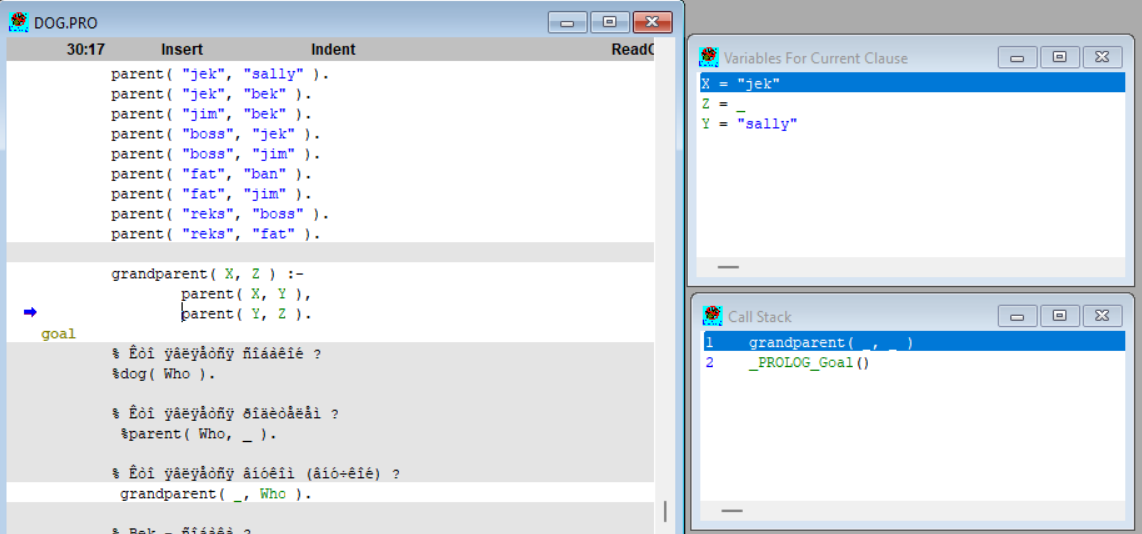
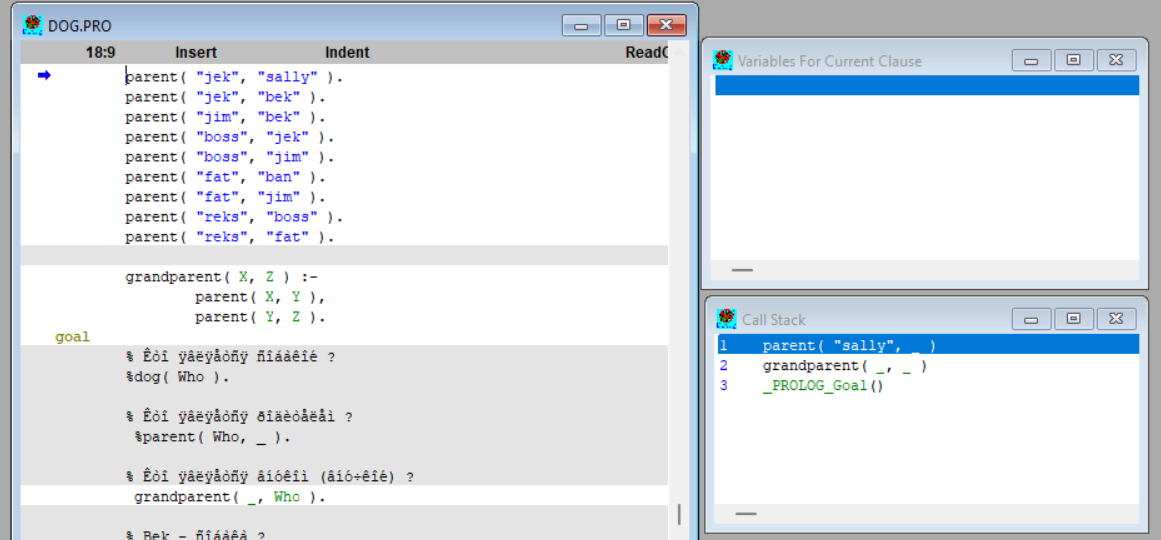
Результат:



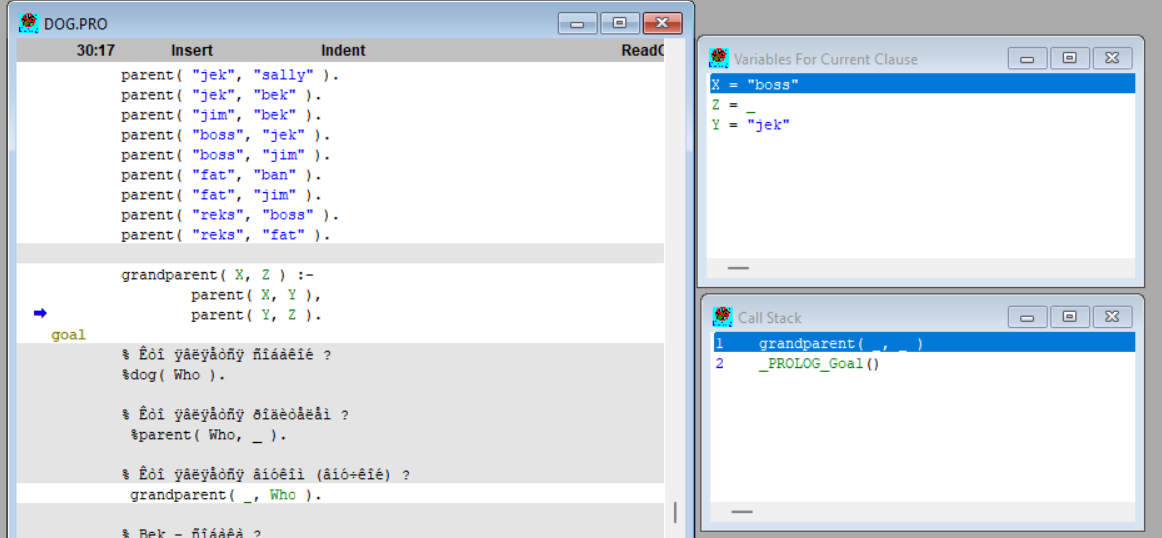
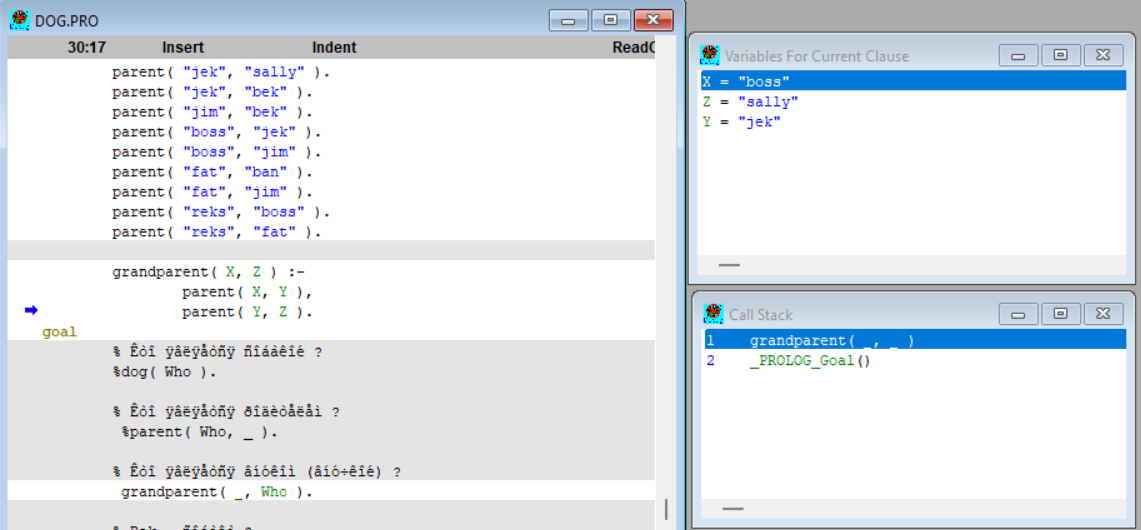
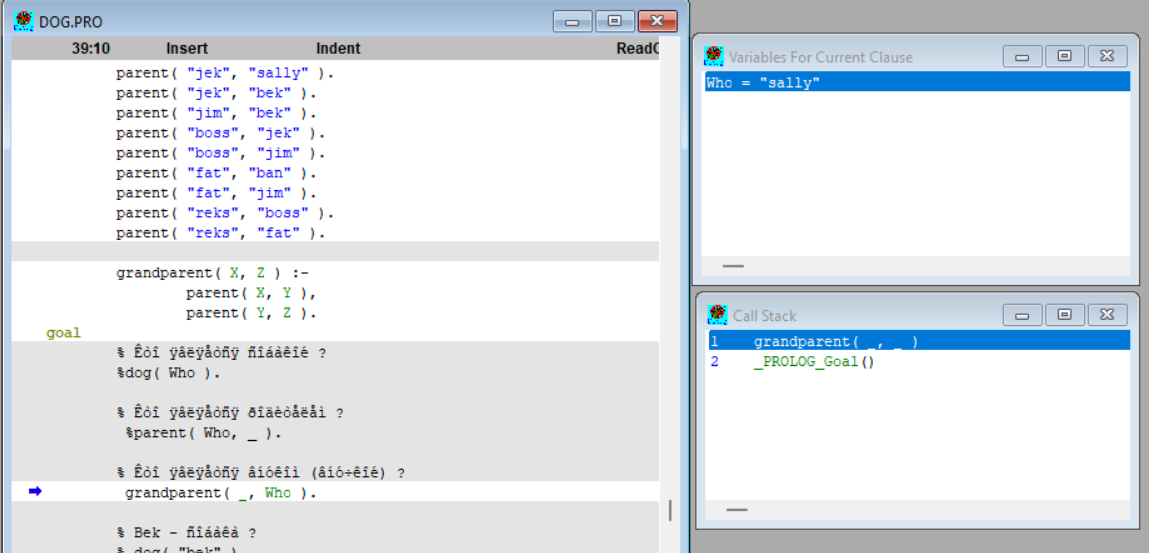
Результатом стал список всех родителей. Результатов 9, так как существует 9 фактов, удовлетворяющих запросу.

Запрос Кто является внуком (внучкой)?

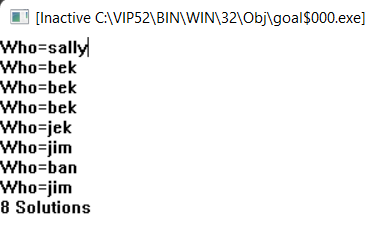
Трасса:

…

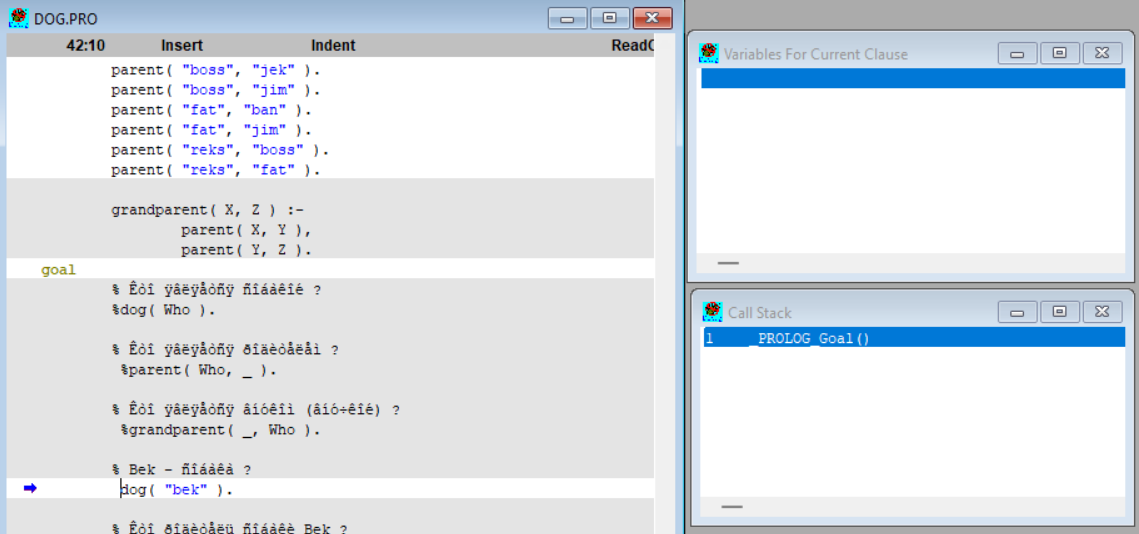
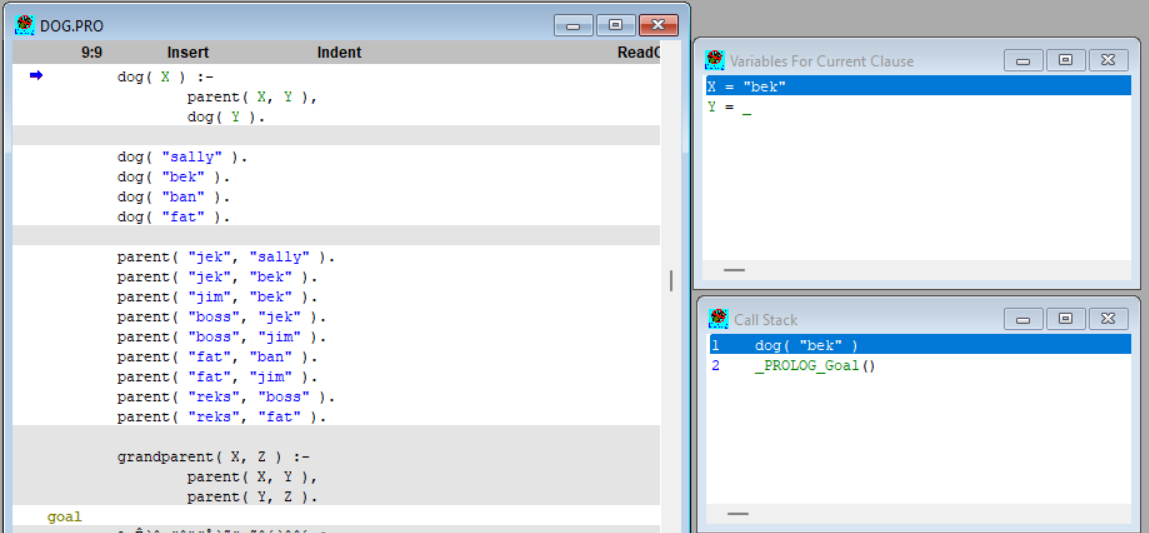
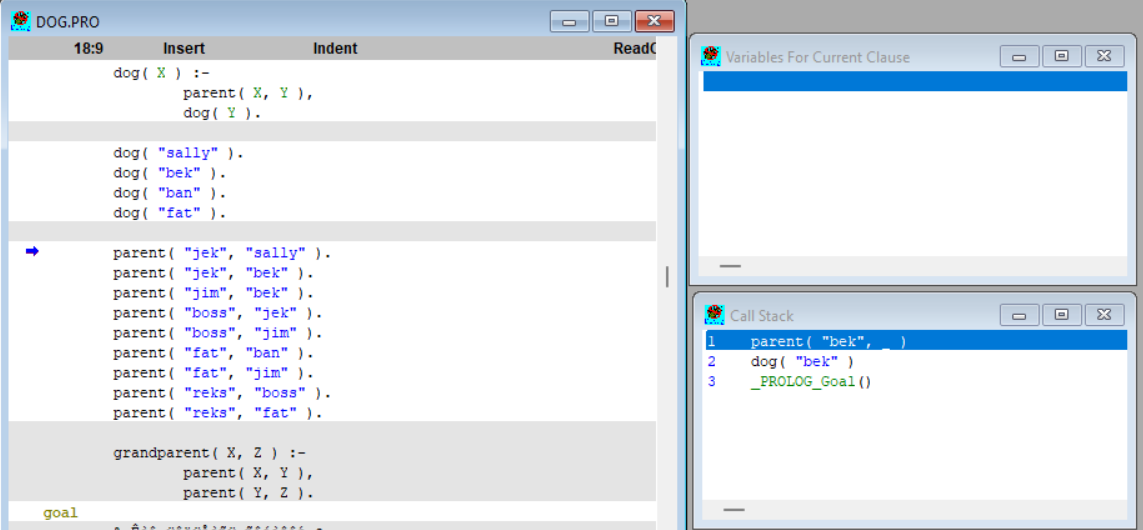
Результат:



Результатом стало 8 решений. Такое количество решений связано с количеством дедушек (бабушек), а точнее количеством связей между внуком (внучкой) и дедушкой (бабушкой). Так например bek записан 3 раза так как у него есть 3 связи с дедушкой (бабушкой): к boss через jek и jim, к fat через jim.

Запрос Bek – собака?

Трасса:

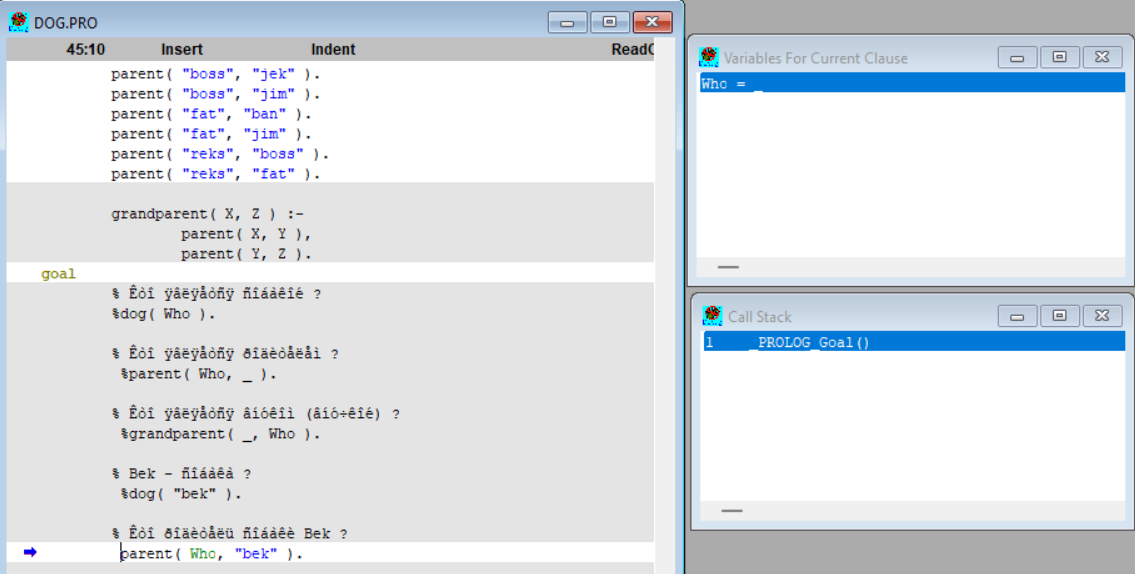
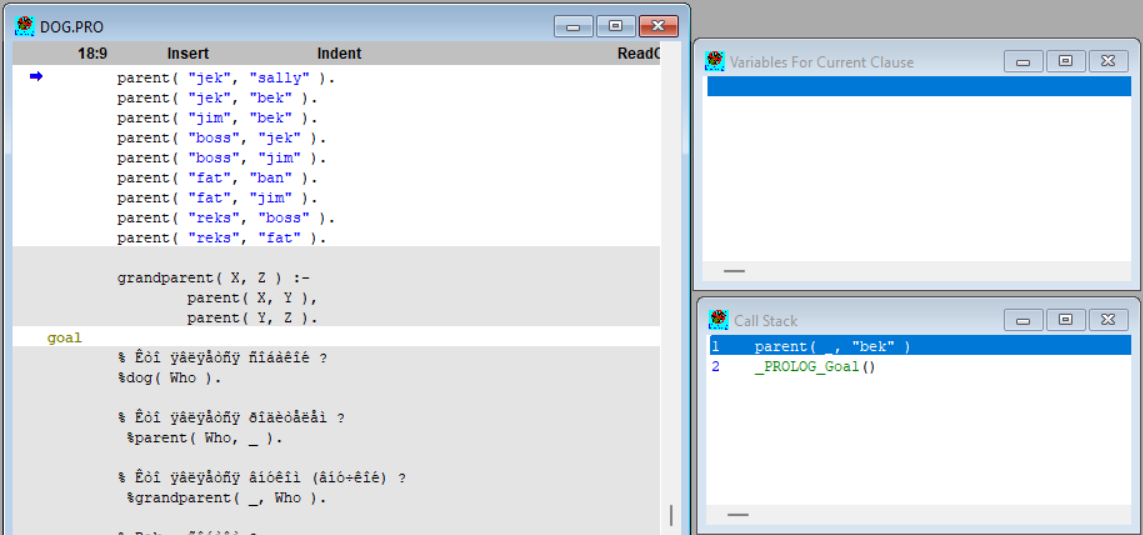
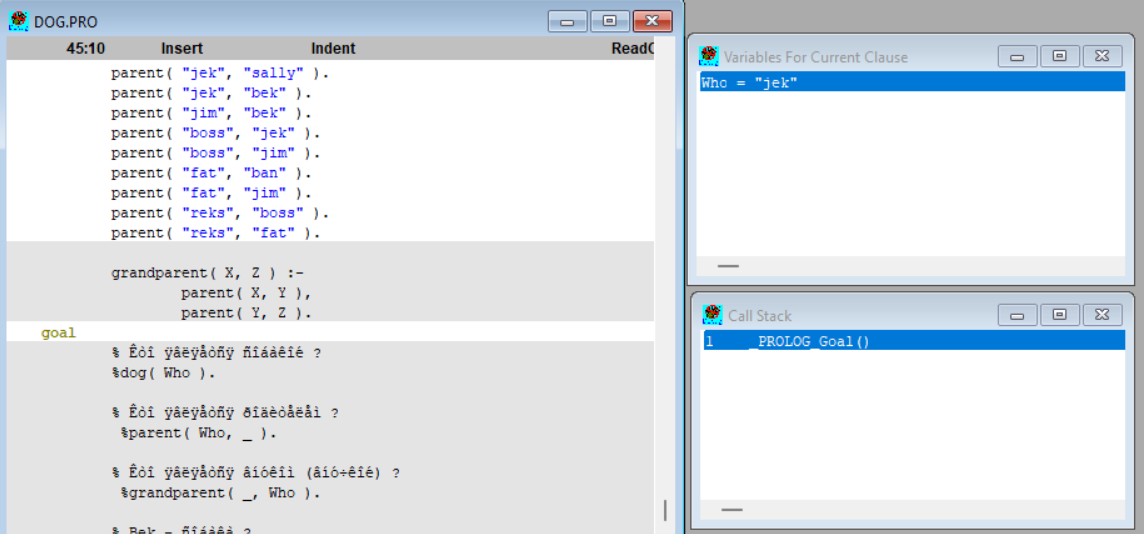
Результат:



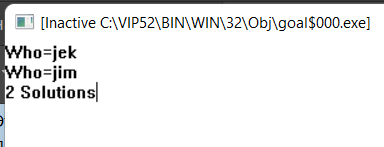
Результатом стал ответ yes – истина, то есть существует такой факт как «bek является собакой».

Запрос Кто родитель собаки bek?

Трасса:

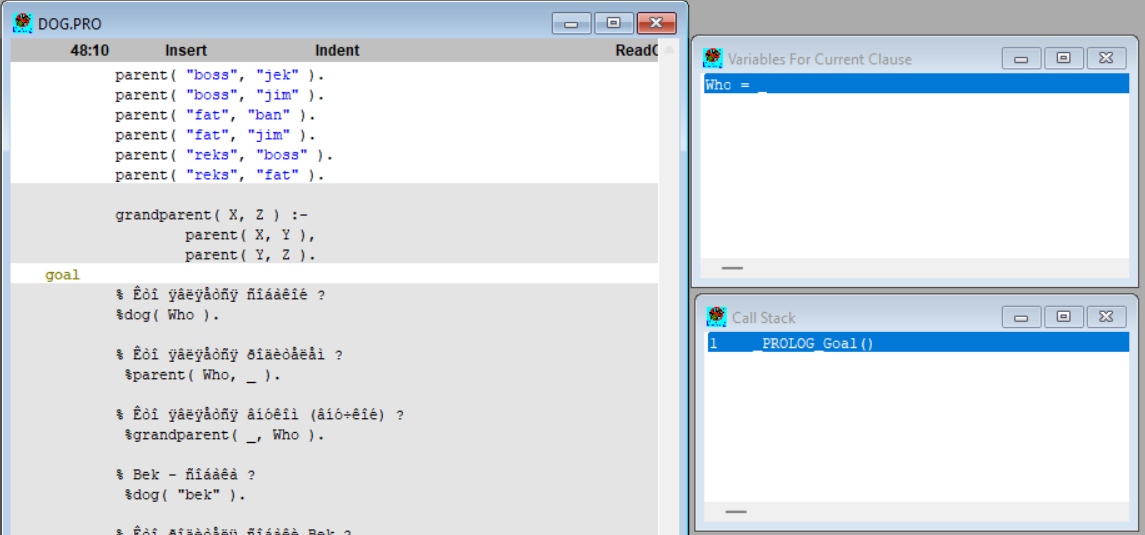
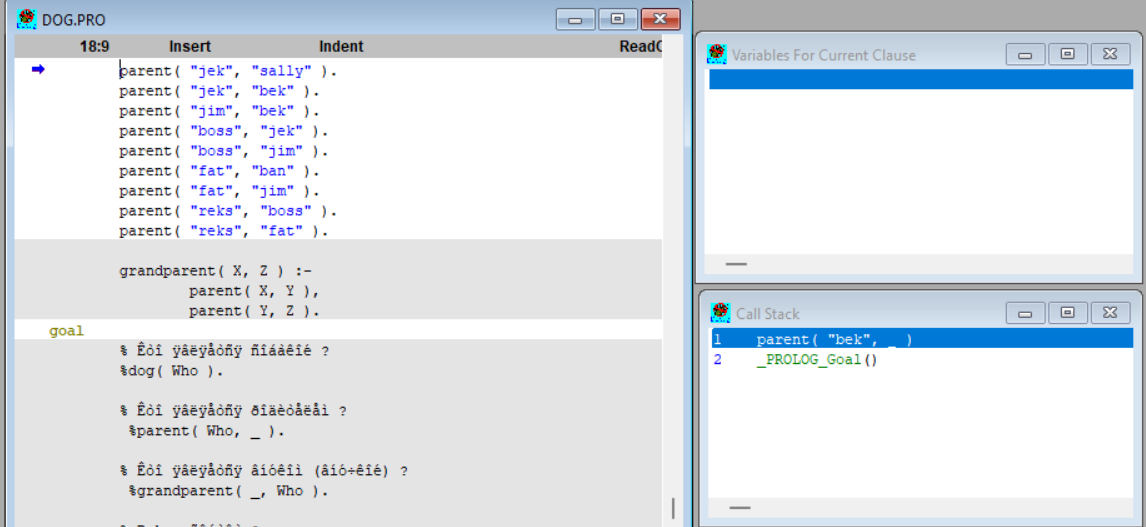
Результат:



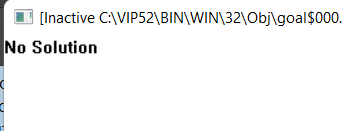
Результатом стало два решения – родители bek

Запрос Кому Bek является родителем?

Трасса:

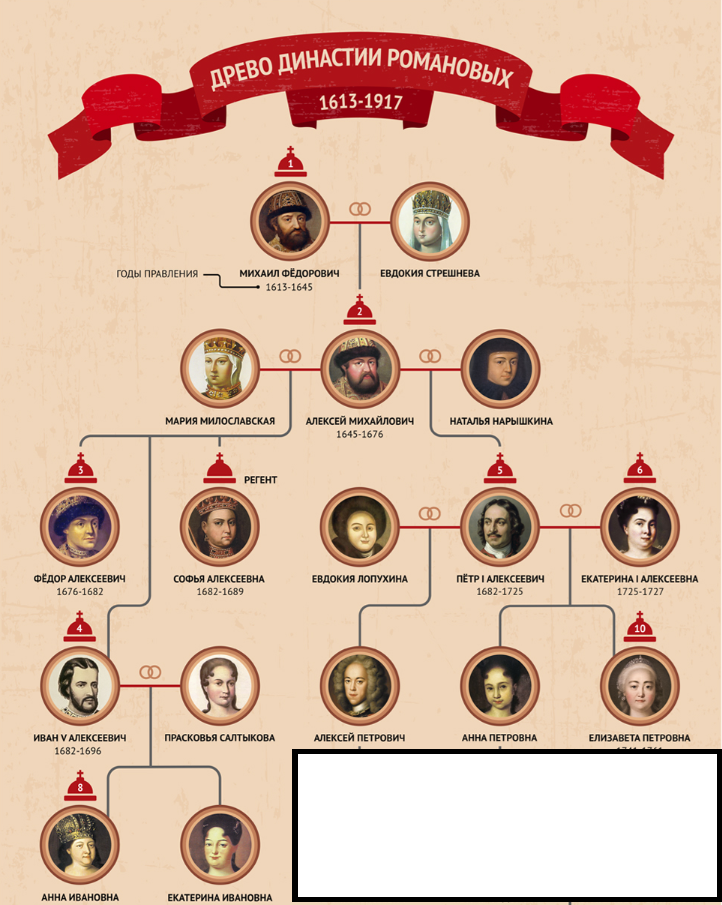
 

Результат:



Результатом стал ответ «Нет решения» так как у bek нет детей

1. Описание родственных связей в семье (в виде дерева).



1. Текст программы и результаты её работы для 5-6 разных запросов.

Текст программы:

domains

name=string

predicates

man( name )

woman( name )

parent( name, name )

married( name, name )

мать( name, name )

отец( name, name )

брат( name, name )

сестра( name, name )

дядя( name, name )

тетя( name, name )

племянник( name, name )

племянница( name, name )

дедушка( name, name )

бабушка( name, name )

внук( name, name )

внучка( name, name )

двоюродный\_брат( name, name )

двоюродная\_сестра( name, name )

clauses

man( "Михаил Федорович" ).

man( "Алексей Михайлович" ).

man( "Федор Алексеевич" ).

man( "Петр Алексеевич" ).

man( "Иван Алексеевич" ).

man( "Алексей Петрович" ).

woman( "Евдокия Стрешнева" ).

woman( "Мария Милославская" ).

woman( "Наталья Нарышкина" ).

woman( "Софья Алексеевна" ).

woman( "Евдокия Лопухина" ).

woman( "Екатерина Алексеевна" ).

woman( "Прасковья Салтыкова" ).

woman( "Анна Петровна" ).

woman( "Елизавета Петровна" ).

woman( "Анна Ивановна" ).

woman( "Екатерина Ивановна" ).

parent( "Михаил Федорович", "Алексей Михайлович" ).

parent( "Евдокия Стрешнева", "Алексей Михайлович" ).

parent( "Мария Милославская", "Федор Алексеевич" ).

parent( "Алексей Михайлович", "Федор Алексеевич" ).

parent( "Мария Милославская", "Софья Алексеевна" ).

parent( "Алексей Михайлович", "Софья Алексеевна" ).

parent( "Мария Милославская", "Иван Алексеевич" ).

parent( "Алексей Михайлович", "Иван Алексеевич" ).

parent( "Алексей Михайлович", "Петр Алексеевич" ).

parent( "Наталья Нарышкина", "Петр Алексеевич" ).

parent( "Петр Алексеевич", "Алексей Петрович" ).

parent( "Евдокия Лопухина", "Алексей Петрович" ).

parent( "Петр Алексеевич", "Анна Петровна" ).

parent( "Екатерина Алексеевна", "Анна Петровна" ).

parent( "Петр Алексеевич", "Елизавета Петровна" ).

parent( "Екатерина Алексеевна", "Елизавета Петровна" ).

parent( "Иван Алексеевич", "Анна Ивановна" ).

parent( "Прасковья Салтыкова", "Анна Ивановна" ).

parent( "Иван Алексеевич", "Екатерина Ивановна" ).

parent( "Прасковья Салтыкова", "Екатерина Ивановна" ).

married( "Михаил Федорович", "Евдокия Стрешнева" ).

married( "Алексей Михайлович", "Мария Милославская" ).

married( "Алексей Михайлович", "Наталья Нарышкина" ).

married( "Иван Алексеевич", "Прасковья Салтыкова" ).

married( "Петр Алексеевич", "Евдокия Лопухина" ).

married( "Петр Алексеевич", "Екатерина Алексеевна" ).

отец( X, Y ) :-

parent( X, Y ),

man( X ).

мать( X, Y ) :-

parent( X, Y ),

woman( X ).

брат( X, Y ) :-

parent( Z, X ),

parent( Z, Y ),

man( X ),

X <> Y.

сестра( X, Y ) :-

parent( Z, X ),

parent( Z, Y ),

woman( X ),

X <> Y.

дядя( X, Y ) :-

брат( X, Z ),

parent( Z, Y ).

дядя( X, Y ) :-

married( X, Z ),

тетя( Z, Y ).

тетя( X, Y ) :-

сестра( X, Z ),

parent( Z, Y ).

тетя( X, Y ) :-

married( X, Z ),

дядя( Z, Y ).

племянник( X, Y ) :- man( X ), дядя( Y, X ).

племянник( X, Y ) :- man( X ), тетя( Y, X ).

племянница( X, Y ) :- woman( X ), дядя( Y, X ).

племянница( X, Y ) :- woman( X ), тетя( Y, X ).

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

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

внук( X, Y ) :- man( X ), дедушка( Y, X ).

внук( X, Y ) :- man( X ), бабушка( Y, X ).

внучка( X, Y ) :- woman( X ), дедушка( Y, X ).

внучка( X, Y ) :- woman( X ), бабушка( Y, X ).

двоюродный\_брат( X, Y ) :- man( X ), дядя( Z, Y ), parent( Z, X ).

двоюродный\_брат( X, Y ) :- man( X ), тетя( Z, Y ), parent( Z, X ).

двоюродная\_сестра( X, Y ) :- woman( X ), дядя( Z, Y ), parent( Z, X ).

двоюродная\_сестра( X, Y ) :- woman( X ), тетя( Z, Y ), parent( Z, X ).

goal

%Кто является двоюродной сестрой и для кого?

двоюродная\_сестра( Who, Whose ).

%Мужчины в семье

%man( Who ).

% Кто является сестрой и для кого?

%сестра( Who, Whose ).

%кто является племянником и для кого?

%племянник( Who, Whose ).

%Является ли Пётр Алексеевич внуком?

%внук( "Петр Алексеевич", \_ ).

%Женат ли Алексей Михайлович?

%married( "Алексей Михайлович", \_ ).

Результаты работы:

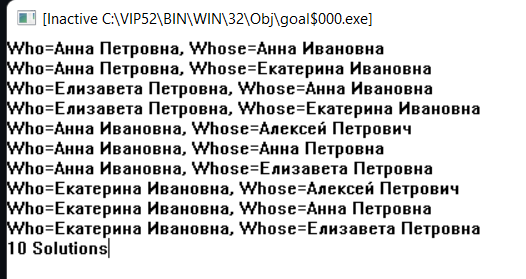


Рис 5.1 – Результат выполнения запроса «Кто является двоюродной сестрой и для кого?»

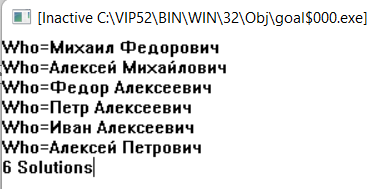


Рис 5.2 – Результат выполнения запроса «Мужчины в семье»

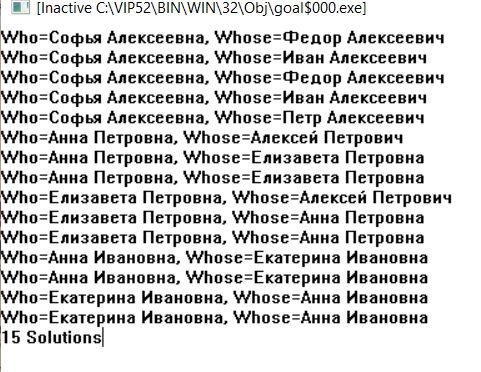


Рис 5.3 – Результат выполнения запроса «Кто является сестрой и для кого?»

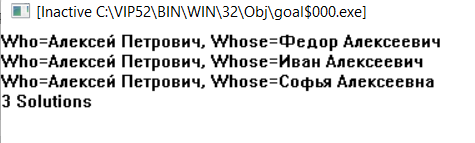


Рис 5.4 – Результат выполнения запроса «Кто является племянником и для кого?»

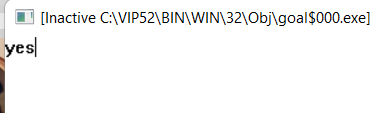


Рис. 5.5 – Результат выполнения запроса «Является ли Петр Алексеевич внуком?»

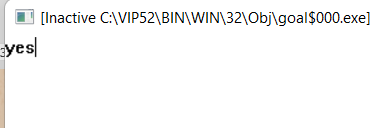


Рис.5.6 – Результат выполнения запроса «Женат ли Алексей Михайлович?»