МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №1
по дисциплине «Логическое программирование»
Тема: Использование основных элементов языка
Вариант 3.

Студент гр. 0303 _____ Бодунов П.А.
Студент гр. 0303 _____ Болкунов В.О.
Студент гр. 0303 _____ Калмак Д.А.
Преподаватель _____ Родионов С.В.

Санкт-Петербург
2024

Цель работы.

Целью работы является изучение основ языка Пролог, освоение принципов работы правил, фактов и вопросов.

Задачи.

Для достижения поставленной цели требуется решить следующие задачи:

1) Изучить теоретический материал.

2) Выполнить задание с номером варианта, равным номеру бригады:

- создать набор фактов о родственных связях.
- создать правила в соответствии с заданием.
- проверить выполнение программы.

3) Составить отчет о выполнении работы.

4) Представить на проверку файл отчета и файл текста программы на языке GNU Prolog, решающей поставленную задачу.

Номер варианта и текст варианта задания должны быть представлены в форме комментариев в тексте программы. Номер группы и номер варианта должны присутствовать в имени файла с текстом программы.

Основные теоретические положения.

Программа на Прологе есть совокупность утверждений. Утверждения состоят из целей и хранятся в базе данных Пролога. Таким образом, база данных Пролога может рассматриваться как программа на Прологе. В конце утверждения ставится точка ".". Иногда утверждение называется предложением.

Основная операция Пролога – доказательство целей, входящих в утверждение.

Существуют два типа утверждений:

– факт – это одиночная цель, которая, безусловно, истинна;

– правило – состоит из одной головной цели и одной или более хвостовых целей, которые истинны при некоторых условиях.

Правило обычно имеет несколько хвостовых целей в форме конъюнкции целей.

Конъюнкцию можно рассматривать как логическую функцию И. Таким образом, правило согласовано, если согласованы все его хвостовые цели.

Примеры фактов:

собака(рекс).

родитель(голди, рекс).

Примеры правил:

собака (X) :- родитель (X,Y),собака (Y).

человек(X) :- мужчина(X).

Разница между правилами и фактами чисто семантическая. Хотя для правил мы используем синтаксис операторов (более подробное рассмотрение операторного и процедурного синтаксисов выходит за рамки нашего курса), нет никакого синтаксического различия между правилом и фактом.

Так, правило

собака (X) :- родитель(X,У),собака(У). может быть задано как

:-собака (X) ',' родитель(X,У),собака (Y).

Запись верна, поскольку :- является оператором "при условии, что", а ',' – это оператор конъюнкции. Однако удобнее записывать это как

собака (X) :- родитель (X,Y),собака (Y).

и читать следующим образом: " X – собака при условии, что родителем X является Y и Y – собака".

Задание.

В любом текстовом редакторе создайте файл parents.pl, в нем заданы следующие факты принадлежности лиц к определенному полу и отношения родства:

parent(tom, bob).

parent(ann, bob).

parent(tom, liza).

parent(bob, mary).

parent(bob, luk).

parent(luk, kate).

male(tom).

male(bob).

male(luk).

female(kate).

female(liza).

female(mary).

Самостоятельно создайте правила для поиска родственных связей: вариант 3 - племянник, кузина (двоюродная сестра).

В оболочке Пролога откройте этот файл parents.pl (меню File/Consult).

Приведите примеры вызова соответствующих правил (вопрос и полученные результаты).

Также создайте правило, возвращающее название типа родства для двух заданных лиц.

Выполнение работы.

1. Порядок выполнения

Созданы факты о принадлежности к полу: мужчина и женщина.

Факт для определения мужчин: male(X), где X - имя мужчины.

Факт для определения женщин: female(X), где X - имя женщины.

Задан набор фактов о родственных связях: родитель.

Факт о родителях: parent(X, Y), где Y является родителем X.

Созданы правила в соответствии с заданием: племянник и кузина.

Вспомогательным правилом для племянника и кузины является родственная связь: брат или сестра.

Правило для брата или сестра: sibling(A, B), где B является братом или сестрой A при условии, что у них общий родитель X и A и B не являются одним человеком.

Правило для племянника: nephew(A, B), где B является племянником A при условии, что B - мужчина, а родитель B является братом или сестрой A.

Правило для кузины, или двоюродной сестры: cousina(A, B), где B является кузиной A при условии, что B - женщина, а родитель A - X1, и родитель B - X2, являются друг другу братом или сестрой.

Создано правило, возвращающее название типа родства для двух заданных лиц. Для связей родитель, племянник и кузина созданы по две проверки, поскольку связь односторонняя, а для брата или сестры создана одна проверка, поскольку связь является взаимной.

Тестирование программы представлено в разделе 3 Примеры вызова соответствующих правил и результаты выполнения.

2. Текст программы с комментариями

```
% Задача о родственных связях. Вариант 3: Племянник, кузина (двоюродная сестра).
% Создание фактов: мужчины. male(X), где X - имя мужчины.
male(tom).
male(bob).
male(luk).
male(vlad).
male(claus).
male(alex).
```

```
% Создание фактов: женщины. female(X), где X - имя женщины.
female(kate).
female(liza).
female(mary).
female(lena).
female(alice).
female(ksenia).

% Создание фактов: родитель. parent(X, Y), где Y является родителем X.
parent(tom, bob).
parent(ann, bob).
parent(tom, liza).
parent(bob, mary).
parent(bob, luk).
parent(luk, kate).
parent(vlad, tom).
parent(vlad, lena).
parent(ksenia, tom).
parent(ksenia, lena).
parent(alice, ann).
parent(alice, claus).
parent(alex, ann).
parent(alex, claus).

% Создание правила: брат или сестра. sibling(A, B), где B является братом или сестрой A при условии, что у них общий родитель X и A и B не являются одним человеком.
sibling(A, B) :- parent(A, X), parent(B, X), A \= B.

% Создание правила: племянник. nephew(A, B), где B является племянником A при условии, что B - мужчина, а родитель B является братом или сестрой A.
nephew(A, B) :- male(B), parent(B, X), sibling(A, X).

% Создание правила: кузина. cousina(A, B), где B является кузиной A при условии, что B - женщина, а родитель A - X1, и родитель B - X2, являются друг другу братом или сестрой.
cousina(A, B) :-
   female(B), parent(B, X1),
   parent(A, X2), sibling(X1, X2).

% Правило определения родственных связей для двух людей. Для связей родитель, племянник и кузина созданы по две проверки, поскольку связь односторонняя, а для брата или сестры создана одна проверка, поскольку связь является взаимной.
who(A, B) :-
    (parent(A, B), write(B), write(' is parent of '), write(A);
    (parent(B, A), write(A), write(' is parent of '), write(B);
    (sibling(A, B), write(A), write(', '), write(B), write(' are siblings'));
    (nephew(A, B), write(B), write(' is nephew of '), write(A);
```

```
          (nephew(B,  A),  write(A),  write('  is  nephew  of  ')),
write(B);
          (cousina(A,  B),  write(B),  write('  is  cousina  of  ')),
write(A);
          (cousina(B,  A),  write(A),  write('  is  cousina  of  ')),
write(B).
```

3. Примеры вызова соответствующих правил и результаты выполнения

Вызов факта *parent* для проверки связи родитель представлен на рис. 1:

```
| ?- parent(ksenia, tom).
```

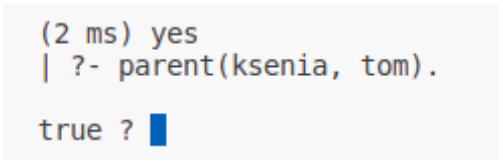

Рисунок 1 - Вызов факта parent

Вызов правила *sibling* для проверки связи брат-сестра представлен на рис. 2:

```
| ?- sibling(tom, ann).
```

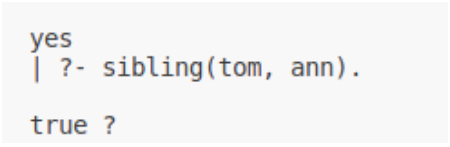

Рисунок 2 - Вызов правила sibling

Вызов правила *nephew* для поиска племянников представлен на рис. 3:

```
| ?- nephew(A, B).
```

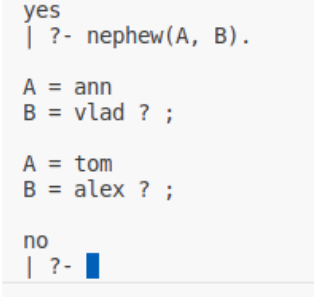

Рисунок 3 - Вызов правила nephew

Вызов правила cousina для поиска двоюродных сестёр (кузин) представлен на рис. 4:

```
| ?- cousina(A, B).
```

```
| ?- cousina(A, B).

A = vlad
B = alice ? ;

A = ksenia
B = alice ? ;

A = alice
B = ksenia ? ;

A = alex
B = ksenia ? ;

no
| ?-
```

Рисунок 4 - Вызов правила cousina

Вызов правил who для установления родственной связи представлены на рис. 5-8:

```
| ?- who(alice, vlad).
```

```
| ?- who(alice, vlad).
alice is cousina of vlad

true ?

yes
```

Рисунок 5 - Вызов правила who для идентификаторов alice и vlad

```
| ?- who(vlad, alice).
```

```
| ?- who(vlad, alice).
alice is cousina of vlad

true ?

yes
```

Рисунок 6 - Вызов правила who для идентификаторов vlad и alice

```
| ?- who(tom, alex).
```

```
| ?- who(tom, alex).
alex is nephew of tom

true ?

yes
```

Рисунок 7 - Вызов правила who для идентификаторов tom и alex

```
| ?- who(alex, tom).
```

```
| ?- who(alex, tom).
alex is nephew of tom

true ?

yes
```

Рисунок 8 - Вызов правила who для идентификаторов alex и tom

Выводы

В результате выполнения лабораторной работы были описаны правила на языке GNU Prolog, позволяющие решать задачи родственных связей: определение родителя, брата/сестры, племянника и двоюродной сестры (кузины). Были приведены примеры вызова правил для поиска людей связанных конкретными связями и для определения связи между двумя персонами.

Роли членов бригады:

Бодунов Пётр 0303 написание кода, написание комментариев в код, оформление отчета.

Болкунов Владислав 0303 написание кода, написание комментариев в код, оформление отчета.

Калмак Даниил 0303 написание кода, написание комментариев в код, оформление отчета.

Трудности:

1. В правиле sibling была предпринята попытка оптимизации, заключающаяся в переносе условия неравенства в начало. Однако так как переменные A, B до вызова правил parent ещё не связанные - данное выражение будет ложным (так как для несвязанных переменных верно утверждение A = B). В итоге данное правило работает только при передаче конкретных значений и не подходит для поиска вариантов решения. В правиле с условием неравенства в конце таких проблем не возникает, что использовано в конечном варианте программы.

```
sibling(A, B) :- A \= B, parent(A, X), parent(B, X).

| ?- sibling(tom, ann).

true ?

| ?- sibling(A, B).

no
```