

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. Задача о
родственных связях
Вариант 1

Студентка гр. 1304

Чернякова В.А.

Студентка гр. 1304

Ярусова Т.В.

Студент гр. 1304

Байков Е.С.

Студент гр. 1304

Мамин Р.А.

Преподаватель

Родионов С.В.

Санкт-Петербург

2025

Цель работы.

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.

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, указанных в задании. Вариант 1 – дядя, невестка (жена сына или жена брата).

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.

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,Y), собака (Y). может быть задано как
:-собака (X) ', ' родитель (X,Y), 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".

Порядок выполнения работы.

В файле *parents.pl* были прописаны факты, указанные в задании к лабораторной работе №1.

Факт для определения пола:

- Мужчина – `male(name);`
- Женщина – `female(name);`

Факт для определения наличия родственных связей:

- `parent(X,Y)` – X является родителем Y.

Для решения задания из номера варианта 1 исходные данные были дополнены следующим образом:

- `female(elis).`
- `female(lilo).`
- `female(taty).`
- `parent(liza, elis).`
- `parent(lilo, kate).`
- `parent(taty, luk).`

Для наглядности полученные отношения можно отобразить в виде схемы, представленной на рисунке 1.

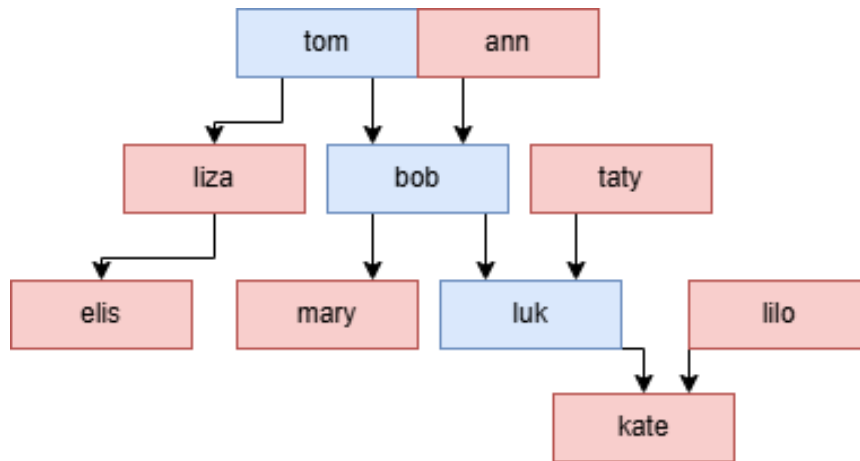


Рисунок 1. Родословная, заданная в программе

Правило дядя.

В родственных связях дядя – брат родителя.

Правило родитель написано ранее. Определим новое правило брат.

`brother(X, Y) :- male(X), parent(F, X), parent(F, Y), X \= Y.`

X является братом Y при условии, что X – мужчина, F – родитель и X и Y и при этом X и Y не являются одним человеком. На рисунке 2 представлена схема правила brother.

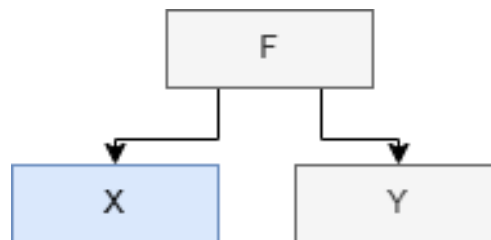


Рисунок 2. Правило brother

Данное правило дополнит правило для определения дяди.

`uncle(X, Y) :- brother(X, F), parent(F, Y).`

X является дядей Y при условии, что X является братом F и при этом F родитель Y. На рисунке 3 представлена схема правила uncle.

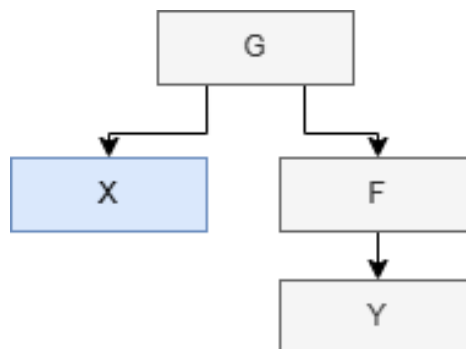


Рисунок 3. Правило uncle

Правило невестка.

В родственных связях невестка – жена брата или сына.

Правила для определения сына (тоже самое, что и определить родителя) и брата написано ранее. Определим новое правило брак.

```
marriage(X,Y):-parent(X,F),parent(Y,F),X\=Y.
```

X и Y состоят в браке при условии, что X и Y родители F и при этом X и Y не являются одним человеком. На рисунке 4 представлена схема правила marriage.

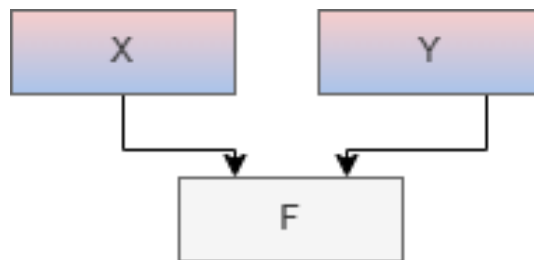


Рисунок 4. Правило marriage

Данное правило дополнит правило для определения невестки.

```
daughterinlaw(X,Y):-female(X),marriage(X,F),parent(Y,F);brother(F,Y).
```

X является невесткой Y при условии, что X – женщина, X состоит в браке с F и Y является родителем F ИЛИ F является братом Y. На рисунке 5 представлена схема правила daughterinlaw.

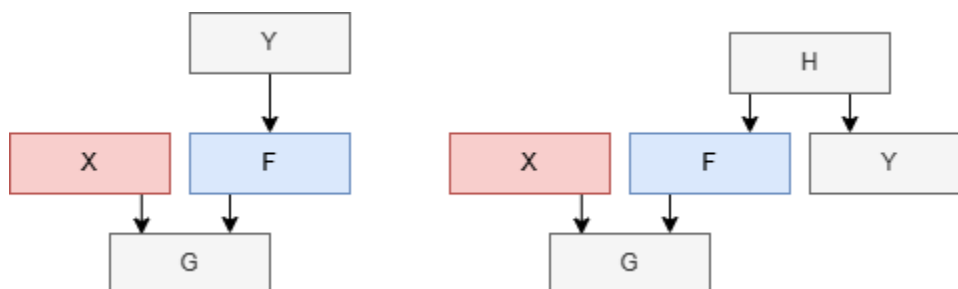


Рисунок 5. Правило daughterinlaw

Правило определения типа родства.

Тип родства определялся на основе написанных ранее правил.

Написано новое правило who, которое проверяет наличие одной из определенных родственных связей между парой людей и в случае, если эта связь между людьми есть, выводит ее название:

```
who(X,Y):-  
(parent(X,Y),write(X),write(' parent of '),write(Y));  
(brother(X,Y),write(X),write(' brother of '),write(Y));
```

```
(uncle(X,Y),write(X),write(' ucnle of '),write(Y));
(marriage(X,Y),male(X),write(X),write(' married to '),write(Y));
(daughterinlaw(X,Y),write(X),write(' daughterinlaw of '),write(Y));
```

Полный текст программы с комментариями смотри в приложении А.

Примеры вызова правил

Были выбраны следующие примеры:

- Является ли mary дядей kate. Смотри рисунок 6.

```
| ?- uncle(mary,kate) .
no
```

Рисунок 6. Является ли mary дядей kate

- Кто дядя elis. Смотри рисунок 7.

```
| ?- uncle(X,elis) .
X = bob ?
(125 ms) yes
```

Рисунок 7. Кто дядя elis

- Чей дядя tom. Смотри рисунок 8.

```
| ?- uncle^tom,X) .
no
```

Рисунок 8. Определение чей дядя tom.

- Является ли elis невесткой ann. Смотри рисунок 9.

```
| ?- daughterinlaw(elis,ann) .
no
```

Рисунок 9. Является ли elis невесткой ann

- Кто невестка для mary. Смотри рисунок 10.

```
| ?- daughterinlaw(X,mary) .
X = lilo ?
yes
```

Рисунок 10. Кто невестка для mary

- Чья невестка taty. Смотри рисунок 11.

```
| ?- daughterinlaw(taty,X) .
X = tom ?
yes
```

Рисунок 11. Чья невестка taty

- На рисунке 12 представлены примеры определения родства.

```

| ?- who(taty,liza).
taty daughterinlaw of liza

true ?

yes
| ?- who(luk,mary).
luk brother of mary

true ?

yes
| ?- who(luk,kate).
luk parent of kate

true ?

yes
| ?- who(bob,elis).
bob ucnle of elis

true ?

yes

```

Рисунок 12. Определение родственных связей

Выводы.

В ходе выполнения лабораторной работы были описаны правила на языке GNU Prolog. Данные правила позволяют решать задачи родственных связей, а именно определение брата, дяди, брака и невестки. Были приведены примеры вызова правил для поиска людей связанных конкретными родственными связями и для определения вида связи между двумя людьми.

Зоны ответственности членов бригады:

- Чернякова В.А. – написание программы;
- Ярусова Т.В. – написание программы;
- Байков Е.С. – тестирование программы;
- Мамин Р.А. – составление отчета.

Каждый участник бригады проконтролировал действия других участников и разобрался в проделанной ими работе.

В ходе выполнения лабораторной работы возникли следующие трудности:

- При первом написании программы не была учтена необходимость добавления в ряд правил условия неравенства. Так, например, в правиле brother

использовались факты $\text{parent}(F,X), \text{parent}(F,Y)$. Однако, если не добавить условие $X \neq Y$, то X для любого значения Y будет являться братом. Это происходит потому, что факт parent перебирает все возможные варианты, в результате чего правдивый факт с X может быть учтен дважды, что даст истинность правила в общем.

- Задание не предусматривает возможность определить брак между людьми как факт, а не правило. Из-за этого в программе брак определен как наличие общего ребенка. Такое определение не отражает реальную жизнь, но позволяет выполнить задание.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: parent.pl

% Задание: создать правила для поиска родственных связей указанный в задании.

% Создать правило возвращающее название типа родства для двух людей.

% Вариант 1 - дядя, невестка (жена сына или жена брата).

% Бригада 1 группы 1304 - Чернякова, Ярусова, Байков, Мамин.

% Создание фактов родитель. parent(X,Y). X родитель Y.

parent(tom, bob).

parent(ann, bob).

parent(tom, liza).

parent(bob, mary).

parent(bob, luk).

parent(luk, kate).

parent(liza, elis).

parent(lilo, kate).

parent(taty, luk).

% Создание фактов мужчина.

male(tom).

male(bob).

male(luk).

% Создание фактов женщина.

female(ann).

female(kate).

female(liza).

female(mary).

female(elis).

female(lilo).

female(taty).

% Правило брат. X является братом Y при условии, что X - мужчина, F - родитель и X и Y и при этом X и Y не являются одним человеком.

```
brother(X,Y):-male(X),parent(F,X),parent(F,Y),X\=Y.
```

% Правило дядя. X является дядей Y при условии, что X является братом F и при этом F родитель Y.

```
uncle(X,Y):-brother(X,F),parent(F,Y).
```

% Правило брак. X и Y состоят в браке при условии, что X и Y родители F и при этом X и Y не являются одним человеком.

```
marriage(X,Y):-parent(X,F),parent(Y,F),X\=Y.
```

% Правило невестка. X является невесткой Y при условии, что X - женщина, X состоит в браке с F и Y является родителем F ИЛИ F является братом Y.

```
daughterinlaw(X,Y):-  
female(X),marriage(X,F),(parent(Y,F);brother(F,Y)).
```

% Правило для вывода родственной связи между двумя людьми.

```
who(X,Y):-
```

```
(parent(X,Y),write(X),write(' parent of '),write(Y));
```

```
(brother(X,Y),write(X),write(' brother of '),write(Y));
```

```
(uncle(X,Y),write(X),write(' uncle of '),write(Y));
```

```
(marriage(X,Y),male(X),write(X),write(' married to '),write(Y));
```

```
(daughterinlaw(X,Y),write(X),write(' daughterinlaw of '),write(Y)).
```