



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Дисциплина: «Функциональное и логическое
программирование»

Лабораторная работа №12

Студент: Левушкин И. К.

Группа: ИУ7-62Б

Преподаватели: Толпинская Н. Б.,

Строганов Ю. В.

Москва, 2020 г.

1 Задание

Составить программу – базу знаний, с помощью которой можно определить, например, множество студентов, обучающихся в одном ВУЗе. Студент может одновременно обучаться в нескольких ВУЗах. Привести примеры возможных вариантов вопросов и варианты ответов (не менее 3-х). Описать порядок формирования вариантов ответа. Исходную базу знаний сформировать с помощью только фактов. *Исходную базу знаний сформировать, используя правила. *Разработать свою базу знаний (содержание произвольно).

2 Реализация программы

Ниже приведен листинг кода программы - базы знаний, соответствующей поставленной задаче.

Листинг 1: Листинг кода программы

```
1 domains
2     student = person(fio , string Phone, birthday).
3     fio = name(string First , string Last).
4     birthday = b_date(string Month, integer Day, integer Year).
5     info = inf(string Faculty, integer Department, integer Group).
6 predicates
7     univ_list(string Unv, student, info).
8
9 clauses
10    univ_list("Bauman", person(name("Ilya", "Levushkin"), "8-985-977-14-92",
11    b_date("december", 11, 1999)), inf("IU", 7, 62)).
12
13    univ_list("Bauman", person(name("Pavel", "Pavlovich"), "8-985-977-14-93",
14    b_date("may", 12, 1999)), Info):-
15    univ_list("Bauman", person(name("Ilya", "Levushkin"), _, _), Info).
16
17    univ_list("Bauman", person(name("Pavel", "Pavlov"), "8-985-977-14-94",
18    b_date("may", 13, 2000)), Info):-
19    univ_list("Bauman", person(name("Ilya", "Levushkin"), _, _), Info).
20
21    univ_list("Bauman", person(name("Anton", "Antonich"), "8-985-977-14-95",
22    b_date("may", 12, 2000)), Info):-
23    univ_list("Bauman", person(name("Ilya", "Levushkin"), _, _), Info).
24
25    univ_list("Bauman", person(name("Anton", "Anton"), "8-985-977-14-96",
26    b_date("may", 13, 2000)), Info):-
27    univ_list("Bauman", person(name("Ilya", "Levushkin"), _, _), Info).
28
29    univ_list("Bauman", person(name("Michel", "Michelich"), "433-9906",
30    b_date("may", 12, 1982)), Info):-
31    univ_list("Bauman", person(name("Ilya", "Levushkin"), _, _), Info).
```

```

32  univ_list("Bauman", person(name("Michel", "Michel"), "433-9907",
    b_date("may", 13, 1982)), inf("IU", 6, 63)).
33
34  univ_list(Unv, person(name("Dima", "Dimaich"), "8-985-977-14-97",
    b_date("may", 12, 2001)), Info):-
35  univ_list(Unv, person(name("Michel", "Michel"), _, _), Info).
36
37  univ_list(Unv, person(name("Dima", "Dima"), "8-985-977-14-98", b_date(
    "may", 13, 2001)), Info):-
38  univ_list(Unv, person(name("Michel", "Michel"), _, _), Info).
39
40  univ_list(Unv, person(name("Vova", "Vovaich"), "8-985-977-14-99",
    b_date("may", 12, 2002)), Info):-
41  univ_list(Unv, person(name("Michel", "Michel"), _, _), Info).
42
43
44
45
46
47
48  univ_list("MSU", person(name("Ilya", "Levushkin"), Phone, Birthday),
    inf("VMK", 5, 99)):-
49  univ_list("Bauman", person(name("Ilya", "Levushkin"), Phone, Birthday)
    , _).
50
51  univ_list("MSU", person(name("Vova", "Vova"), "8-985-977-14-00",
    b_date("may", 13, 2002)), Info):-
52  univ_list("MSU", person(name("Ilya", "Levushkin"), _, _), Info).
53
54  univ_list("MSU", person(name("Roma", "Romaich"), "8-985-977-14-01",
    b_date("may", 12, 2003)), Info):-
55  univ_list("MSU", person(name("Ilya", "Levushkin"), _, _), Info).
56
57  univ_list("MSU", person(name("Roma", "Roma"), "8-985-977-14-02",
    b_date("may", 13, 2003)), Info):-
58  univ_list("MSU", person(name("Ilya", "Levushkin"), _, _), Info).
59
60  univ_list("MSU", person(name("Olya", "Olyaich"), "8-985-977-14-03",
    b_date("may", 12, 2004)), Info):-
61  univ_list("MSU", person(name("Ilya", "Levushkin"), _, _), Info).
62
63  univ_list("MSU", person(name("Olya", "Olya"), "8-985-977-14-04",
    b_date("may", 13, 2004)), Info):-
64  univ_list("MSU", person(name("Ilya", "Levushkin"), _, _), Info).
65
66  univ_list("MSU", person(name("Sveta", "Svetaich"), "8-985-977-14-05",
    b_date("may", 12, 2005)), Info):-
67  univ_list("MSU", person(name("Ilya", "Levushkin"), _, _), Info).
68
69  univ_list("MSU", person(name("Sveta", "Sveta"), "8-985-977-14-06",
    b_date("may", 13, 2005)), Info):-
70  univ_list("MSU", person(name("Ilya", "Levushkin"), _, _), Info).

```

В данной базе знаний использовались составные домены для структуризации информации. Структура данных приведена на рисунке (1):

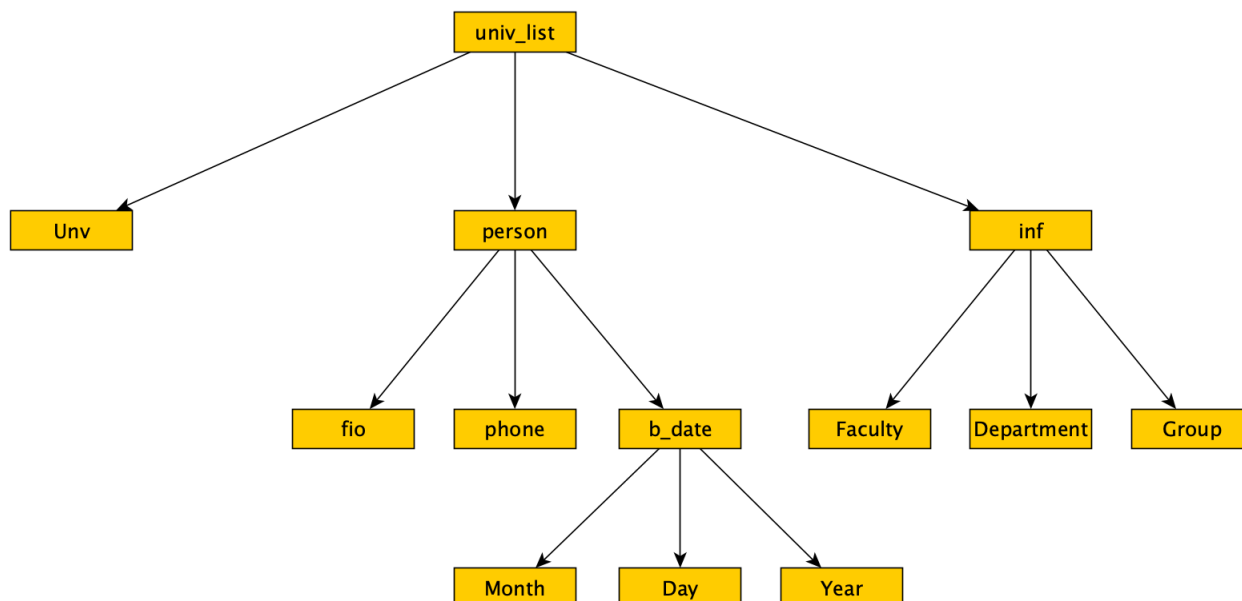


Рис. 1: Древоподобная структура данных программы.

Данная база знаний формировалась с помощью фактов и правил. Правила использовались по аналогии с примером *ch02e01.pro* из 11 лабораторной работы, чтобы присвоить одному терму те же параметры, что и другому (в данном случае присваивалась информация о факультете, кафедре и номере группы - *inf(string Faculty, integer Department, integer Group)*, а также название университета - *string Unv*).

3 Варианты вопросов и ответы на них в программе

Ниже приведены вопросы и ответы программы на них:

Существуют ли люди, обучающиеся в МГУ (MSU)?

Листинг 2: Пример 1

```

1 goal
2   univ_list("MSU", Student, Info).
3 Output:
4   Student=person(name("Ilya", "Levushkin"), "8-985-977-14-92", b_date("
5     december", 11, 1999)), Info=inf("VMK", 5, 99)
6   Student=person(name("Vova", "Vova"), "8-985-977-14-00", b_date("may"
7     , 13, 2002)), Info=inf("VMK", 5, 99)
8   Student=person(name("Roma", "Romaich"), "8-985-977-14-01", b_date("may"
9     , 12, 2003)), Info=inf("VMK", 5, 99)
10  Student=person(name("Roma", "Roma"), "8-985-977-14-02", b_date("may"
11    , 13, 2003)), Info=inf("VMK", 5, 99)
  
```

```

8      Student=person(name("Olya","Olyaich"),"8-985-977-14-03",b_date("may"
      ,12,2004)), Info=inf("VMK",5,99)
9      Student=person(name("Olya","Olya"),"8-985-977-14-04",b_date("may"
      ,13,2004)), Info=inf("VMK",5,99)
10     Student=person(name("Sveta","Svetaich"),"8-985-977-14-05",b_date("may"
      ,12,2005)), Info=inf("VMK",5,99)
11     Student=person(name("Sveta","Sveta"),"8-985-977-14-06",b_date("may"
      ,13,2005)), Info=inf("VMK",5,99)
12     8 Solutions

```

Существуют ли люди с именем Илья Левушкин (Ilya Levushkin)? День рождения не интересуется.

Листинг 3: Пример 2

```

1      goal
2      univ_list(Unv, person(name("Ilya","Levushkin"), Phone, _), Info).
3      Output:
4      Unv=Bauman, Phone=8-985-977-14-92, Info=inf("IU",7,62)
5      Unv=MSU, Phone=8-985-977-14-92, Info=inf("VMK",5,99)
6      2 Solutions

```

Поскольку Илья Левушкин обучается сразу в 2 ВУЗах найдено 2 решения.

Есть ли в базе знаний студенты, учащиеся в Бауманском университете, на факультете ИУ, кафедры 6 и из группы 63? Вывести всех таких студентов.

Листинг 4: Пример 3

```

1      goal
2      univ_list("Bauman", Student, inf("IU", 6, 63)).
3      Output:
4      Student=person(name("Michel","Michel"),"433-9907",b_date("may",13,1982))
5      Student=person(name("Dima","Dimaich"),"8-985-977-14-97",b_date("may"
      ,12,2001))
6      Student=person(name("Dima","Dima"),"8-985-977-14-98",b_date("may"
      ,13,2001))
7      Student=person(name("Vova","Vovaich"),"8-985-977-14-99",b_date("may"
      ,12,2002))
8      4 Solutions

```

Есть ли в базе знаний студенты, учащиеся в МГУ 2003-ого года рождения? Вывести имена и номера телефонов этих студентов.

Листинг 5: Пример 4

```

1      goal
2      univ_list("MSU", person(Name, Phone, b_date(_,_,2003)), _).
3      Output:
4      Name=name("Roma","Romaich"), Phone=8-985-977-14-01
5      Name=name("Roma","Roma"), Phone=8-985-977-14-02
6      2 Solutions

```

4 Назначение и использование переменных

Ниже приведен список переменных с их описанием:

- *Faculty* (Номер факультета), *Department* (Номер кафедры), *Group* (Номер группы), *Month* (Месяц), *Day* (День), *Year* (год), *First* (Имя), *Last* (Фамилия), *Phone* (номер телефона) - именованные переменные, связанные со стандартными доменами (*string*, *integer*), не требующие описания;
- *info* (учебная информация о студенте) - именованная переменная, связанная с составным доменом, образованным функтором *inf* с параметрами *string Faculty*, *integer Department*, *integer Group*;
- *birthday* (день рождения) - именованная переменная, связанная с составным доменом, образованным функтором *b_date* с параметрами *string Month*, *integer Day*, *integer Year*;
- *fio* (полное имя) - именованная переменная, связанная с составным доменом, образованным функтором *name* с параметрами *string First*, *string Last*;
- *student* (студент) - именованная переменная, связанная с составным доменом, образованным функтором *person* с параметрами *fio*, *string Phone*, *birthday*.

5 Порядок формирования результата работы программы

Формирование результата происходит с помощью механизма унификации, встроеного в систему и не доступного программисту.

Опишем процесс унификации для вопроса из примера 1 к нашей программе:

Для Пролога вопрос есть цель, которую необходимо достичь. Пролог берет вопрос `univ_list(«MSU», Student, Info)` и начинает последовательно сверху-вниз сравнивать его с фактами и правилами базы знаний. Там где обнаруживается предикат с таким же идентификатором как и у вопроса и с таким же количеством аргументов происходит сопоставление.

Так как в вопросе первый аргумент имеет конкретное значение, то Пролог проводит сопоставление сравнивая значение из вопроса и из фактов. Как только они совпадут, процесс сравнения переходит на вторые аргументы, третьи. В вопросе второй и третий аргументы ничем не обозначены, поэтому ему передаются значения из факта. О чем, собственно, Пролог и сообщает.

6 Ответы на вопросы

6.1 Что собой представляет программа на Prolog?

Программа на Prolog представляет собой базу знаний и вопрос.

- База знаний состоит из предложений - CLAUSES (отдельных знаний или утверждений): фактов и правил.
- Вопрос состоит только из тела – составного терма (или нескольких составных термов). Вопросы используются для выяснения выполнимости некоторого отношения между описанными в программе объектами. Система рассматривает вопрос как цель, к которой (к истинности которой) надо стремиться.

6.2 Какова структура программы на Prolog?

Программа на Prolog состоит из разделов. Каждый раздел начинается со своего заголовка. Структура программы:

- директивы компилятора — зарезервированные символьные константы
- CONSTANTS — раздел описания констант
- DOMAINS — раздел описания доменов
- DATABASE — раздел описания предикатов внутренней базы данных
- PREDICATES — раздел описания предикатов
- CLAUSES — раздел описания предложений базы знаний
- GOAL — раздел описания внутренней цели (вопроса).

В программе не обязательно должны быть все разделы.

6.3 Как программа реализуется на Prolog?

- Предложения бывают двух видов: факты и правила;
- Каждое предложение заканчивается точкой;
- Предложение более общего вида – правило имеет вид: $A : -B_1, \dots, B_n$
A называется заголовком правила, а B_1, \dots, B_n – телом правила;
- Факт – это предложение, в котором отсутствует тело (частный случай правила);

6.4 Как формируются результаты работы программы?

Ответ на поставленный вопрос система дает в логической форме – «Да» или «нет». Цель системы состоит в том, чтобы на поставленный вопрос найти возможность, исходя из базы знаний, ответить «Да». Вариантов ответить «Да» на поставленный вопрос может быть несколько. Система может (в нашем случае обучения - должна) быть настроена в режим получения всех возможных вариантов ответа «Да» на поставленный вопрос.

Поиск содержательного ответа на поставленный вопрос, с помощью имеющейся базы знаний, фактически заключается в поиске нужного знания, но какое знание понадобится – заранее неизвестно. Этот поиск осуществляется формально с помощью механизма *унификации*, встроенного в систему и не доступного программисту.