



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»

ОТЧЕТ

к лабораторной работе №12

*По курсу: «Функциональное и логическое
программирование»*

Студент ИУ7-64Б
Лозовский А.А.

Преподаватель
Толпинская Н.Б

Москва, 2020 г.

Задание: Составить программу – базу знаний, с помощью которой можно определить, например, множество студентов, обучающихся в одном ВУЗе. Студент может одновременно обучаться в нескольких ВУЗах.

База знаний

1) Были выделены следующие **домены**:

- name – название учебного заведения.
- human – составной домен, включающий в себя следующие компоненты:
 - string FirstName – Имя человека;
 - string LastName – Фамилия;
 - integer Age – Возраст;
 - string City – Город, где родился.
- info – составной домен, включающий в себя следующие компоненты:
 - name University – Название учебного заведения;
 - string Group – Группа, в которой обучается человек;
 - string Direction – Направление обучения.

2) **Предикаты** (Все предикаты недетерминированы (т.е могут вернуть множество значений))

- university(name) – проверяет, является ли заведение университетом.
- school(name) – проверяет, является ли заведение школой.
- college(name) – проверяет, является ли заведение колледжем.
- study(human, info) – информация о студенте.

Наложим на задачу ограничение, для корректности работы программы, пусть наименования учебных заведений уникальны.

В программе используются «_» или анонимные переменные для игнорирования неиспользуемых значений.

3) Условия представлены в виде фактов.

domains

name = string

human = human(string FirstName, string LastName, integer Age, string City)

info = info(name University, string Group, string Direction)

predicates

nondeterm university(name)

nondeterm school(name)

nondeterm college(name)

nondeterm study(human, info)

clauses

university("BMSTU").

university("HSE").

university("MSU").

```
school("School 1289").
college("Technical College 4").
```

```
study( human("Tolya", "Terentiev", 21, "Moscow"), info("HSE", "171", "CSF") ).
study( human("Kolya", "Polyakov", 18, "Moscow"), info("HSE", "171", "CSF") ).
study( human("Artyom", "Cobanov", 21, "Moscow"), info("HSE", "172", "CSF") ).
study( human("Artyom", "Cobanov", 21, "Moscow"), info("BMSTU", "172", "CSF") ).
study( human("Sergey", "Samoylov", 25, "Moscow"), info("HSE", "171", "CSF") ).
```

```
study( human("Tolya", "Terentiev", 21, "Moscow"), info("BMSTU", "171", "CSS") ).
study( human("Kolya", "Polyakov", 18, "Moscow"), info("MSU", "171", "CSC") ).
study( human("Artyom", "Cobanov", 21, "Moscow"), info("Technical College 4", "171", "CC") ).
study( human("Sergey", "Samoylov", 25, "Moscow"), info("Technical College 4", "171", "CC") ).
```

Цели:

1) Получим множество студентов, обучающихся в HSE.

```
goal
    study(Human, info("HSE",_, _)).
```

Prolog «просмотрит» все факты, касающиеся отношения study и подставит вместо переменной Human все имеющиеся в этих фактах константы. Таким образом, системой будет дан ответ, содержащий 4 решения для поставленной задачи:

```
Human=human("Tolya","Terentiev", 21, "Moscow")
Human=human("Kolya","Polyakov", 18, "Moscow")
Human=human("Artyom","Cobanov", 21, "Moscow")
Human=human("Sergey","Samoylov", 25, "Moscow")
4 Solutions
```

2) Получим, является ли «Technical College 4» университетом.

```
goal
    university("Technical College 4").
```

Prolog с помощью механизма вывода, который основан на сопоставлении образцов, будет пытаться доказать поставленную гипотезу. Prolog извлекает хранящуюся (известную) информацию и пытается проверить истинность гипотезы (другими словами — ответить на вопрос), запрашивая для этого информацию, о которой уже известно, что она истинна. «Просмотрев» все имеющиеся в разделе clauses предложения, Prolog «найдет» соответствующий факт и ответит:

```
no
```

3) Получим множество студентов, обучающихся по направлению “CSF”.

```
goal
    study(Human, info(_, _, "CSF")).
```

Результат:

```
Human=human("Tolya","Terentiev",21,"Moscow")
Human=human("Kolya","Polyakov",18,"Moscow")
Human=human("Artyom","Cobanov",21,"Moscow")
Human=human("Artyom","Cobanov",21,"Moscow")
Human=human("Sergey","Samoylov",25,"Moscow")
5 Solutions
```

4) Получим Имена, фамилии, возраст студентов, старше 21.

```
goal
    study(human(Name, LastName, Age, _), _, Age > 21.
```

Результат:

```
Name=Sergey, LastName=Samoylov, Age=25
Name=Sergey, LastName=Samoylof, Age=25
2 Solutions
```

Дополним текущую базу знаний следующими правилами (недетерминированными):

- student(human) – проверяет, является ли человек студентом
- universities – выводит список всех университетов
- hse_students – выводит список студентов ВШЭ.

predicates

```
student(human(Name, LastName, _, _)):- study(human(Name, LastName, _, _), info(University, _, _)),
                                         university(University),
                                         write(Name, " ", LastName, " is student\n").

universities :- university(Name),
                write(Name, " is University.\n"),
                fail.

hse_students :- study(human(Name, LastName, _, _), info("HSE", "171", _)),
                write(Name, LastName, " studies at hse),
                fail.
```

Цели:

1) Получить список всех университетов

```
goal
    universities.
```

Результат:

```
BMSTU is University.
HSE is University.
MSU is University.
no
```

В конце программа выдала результат no, поскольку используется предикат fail, потому что Visual Prolog найдет только первое подходящее решение для и распечатает его. Если же программа будет содержать предикат fail в теле правила, Visual Prolog отработает по-другому. Будет найдено первое решение, отмечена точка возврата, поскольку обращение к **study(human(Name, LastName, _, _), info(University, _, _))** дает не единственное решение, выведено на экран сообщение: **BMSTU is University**, после чего будет вызван предикат fail, который всегда возвращает неуспех. Таким образом, будет инициализирован откат, и Visual Prolog станет находить следующее решение для предиката **study**.

2) Проверить, является ли человек студентом.

```
goal
    student(human("Sergey", "Samoylof", 25, "Moscow")).
```

Сергей учится в колледже, колледж не является университетом -> Сергей не является студентом.

Результат:

```
no
```

3) Получить всех студентов ВШЭ.

```
goal
    hse_students.
```

Результат:

TolyaTerentiev studies at hse
KolyaPolyakov studies at hse
SergeySamoylov studies at hse
no

Собственная база знаний:

База знаний:

В качестве темы для базы знаний была выбрана тематика: «Фильмография».

Были выделены следующие **домены**:

- name = string – название сериалов и фильмов
- info – составной домен со следующими параметрами:
 - string Leader – Режиссер
 - string Genre – Жанр
 - integer Data – Дата выхода
 - integer Rate – рейтинг

Были выделены следующие предикаты (недетерминированные):

- tv_show(name, info) – Информация о сериале
- movie(name, info) – Информация о фильме
- oscar(name) – Есть ли у фильма оскар.
- show_movies_shows(string Leader, name Names) – вывод всех фильмов и сериалов, которые выпустил один режиссер.
- latest(string Leader) – выводит последние фильмы (последние – те, которое вышли позднее 2012г.).

domains

name = string

info = info(string Leader, string Genre, integer Data, integer Rate)

predicates

nondeterm tv_show(name, info)

nondeterm movie(name, info)

oscar(name)

good_movie(name)

nondeterm show_movies_shows(string Leader, name Names)

nondeterm latest(string Leader)

clauses

tv_show("show1", info("leader_1", "drama", 1990, 3)).

tv_show("show2", info("leader_2", "comedy", 2019, 2)).

tv_show("name3", info("leader_3", "drama", 2015, 5)).

movie("movie1", info("leader_1", "comedy", 2019, 9)).

movie("movie2", info("leader_3", "comedy", 2012, 6)).

```
movie("movie3", info("leader_2", "comedy", 2014, 7)).
```

```
oscar("movie1").
```

```
oscar("movie3").
```

```
good_movie(Name):-movie(Name, info(_, _, _, Rate)),  
                    Rate > 7,  
                    oscar(Name).
```

```
latest(Name):-movie(Name, info(_, _, Data, _)),  
                Data > 2012.
```

```
show_movies_shows(Leader, Names):-tv_show(Names, info(Leader,_,_,_));  
                                    movie(Names, info(Leader,_,_,_)).
```

Цели:

goal

```
tv_show("show1", info("leader_1", "drama", 1990, 3)).
```

Результат: Yes.

goal

```
tv_show("movie1", info("leader_1", "comedy", 2019, 9)).
```

Результат: No

goal

```
good_movie("movie3").
```

Результат: No

goal

```
good_movie("movie1").
```

Результат: Yes.

goal

latest("movie1").

Результат: Yes.

goal

latest(Name).

Результат:

Name=movie1

Name=movie3

2 Solutions

goal

latest("show2").

Результат: No.

goal

show_movies_shows("leader_1", Names).

Результат:

Name=show1

Name=movie1

2 Solutions

Вопросы к лабораторной работе:

1) Что собой представляет программа на Prolog?

Ответ: Программа на prolog представляет собой набор фактов и правил, обеспечивающих получение заключений на основе этих утверждений.

2) Какова ее структура?

Ответ: Prolog использует упрощенную версию синтаксиса логики предикатов — предикаты первого порядка. Он включает в себя механизм вывода, который основан на сопоставлении образцов термов, например, терма вопроса и терма факта. Программа на Prolog представляет собой: базу знаний и вопрос

Структура:

- 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).

3) Как она реализуется?

(Prolog, программа) извлекает хранящуюся (известную в программе) информацию с помощью подбора ответов на запросы. База знаний содержит истинностные знания, используя которые программа выдает ответ на запрос.