



КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Тема Работа программы на Prolog

Студент Пересторонин П.Г.

Группа ИУ7-63Б

Оценка

Преподаватель Толпинская Н. Б.

Москва — 2021 г.

Оглавление

1	Лабораторная работа №11	2
2	Лабораторная работа №12	4
3	Лабораторная работа №13	6
3.1	Поиски ответов на вопрос	7
3.2	Теоретические вопросы	8
3.2.1	Что такое терм?	8
3.2.2	Что такое предикат в матлогике (математике)?	8
3.2.3	Что описывает предикат в Prolog?	9
3.2.4	Назовите виды предложений в программе и приведите примеры таких предложений из Вашей программы. Какие предложения являются основными, а какие – не основными? Каковы: синтаксис и семантика (формальный смысл) этих предложений (основных и неосновных)?	9
3.2.5	Каковы назначение, виды и особенности использования переменных в программе на Prolog? Какое предложение БЗ сформулировано в более общей – абстрактной форме: содержащее или не содержащее переменных?	10
3.2.6	Что такое подстановка?	10
3.2.7	Что такое пример терма? Как и когда строится? Как Вы думаете, система строит и хранит примеры? . . .	10

1 Лабораторная работа №11

Задание: запустить среду Visual Prolog 5.2. Настроить утилиту TestGoal. Запустить тестовую программу, проанализировать реакцию системы и множество ответов. Разработать свою программу – «Телефонный справочник». Протестировать работы программы.

```
1 domains
2   name = string
3   phone = integer
4   surname = string
5
6 predicates
7   entry(phone, name, surname)
8
9 clauses
10  entry(666, "Dima", "Yacubovich").
11  entry(777, "Lexa", "Romanovich").
12  entry(888, "Lexa", "Petrovich").
13  entry(1, "Mixail", "Nintendovich").
14  entry(800, "Sergey", "Kononenkovich").
15
16 goal
17  entry(Phone, "Lexa", Surname).
```

Программа на языке **Prolog** представляет собой базу знаний и вопрос. База знаний — набор фактов и правил, которые формируют базу знаний о предметной области. Факт — частный случай правила, состоит только из заголовка и с его помощью фиксируется *истинностное* отношение между объектами предметной области. С помощью правила также фиксируются знания, однако правила обладают телом, в котором фиксируется условие истинности правила. При поиске ответа на вопрос **Prolog** рассматривает альтернативные варианты и находит все возможные решения — множества значений переменных, при которых на поставленный вопрос можно ответить ’’да’’.

Программа состоит из разделов (структура программы), каждый имеет свой заголовок:

- **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 описаны все разделы.

2 Лабораторная работа №12

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).

```
1 domains
2   name, surname, university = string
3
4 predicates
5   student(name, surname, university)
6
7 clauses
8   student("Alexey", "Romanov", "BMSTU").
9   student("Alexey", "Romanov", "MSU").
10  student("Mikhail", "Nitenko", "BMSTU").
11  student("Sergey", "Kononenko", "BMSTU").
12  student("Dmitry", "Yacuba", "MSU").
13  student("Pavel", "Perestoronin", "BMSTU").
14  student("Anton", "Olenev", University) :- student("Alexey", "Romanov", University),
15                                         student("Mikhail", "Nitenko", University).
16
17 goal
18   student(Name, Surname, "MSU").
19   /*
20   Name=Alexey, Surname=Romanov
21   Name=Dmitry, Surname=Yacuba
22   */
23   %student("Alexey", "Romanov", University).
24   /*
25   University=BMSTU
26   University=MSU
27   */
28   %student(Name, Surname, "ITMO").
29   /*
30   No solution
31   */
```

Порядок формирования результатов:

1. `student(Name, Surname, "MSU")`. При сравнении вопроса с 2 предложениями базы знаний унификация вопроса и предложения базы знаний проходит успешно: совпадает функтор, арность, успешно унифицируются все аргументы (а переменные `Name` и `Surname` конкретизируются значениями аргументов функтора предложения БЗ, стоящими на тех же позициях соответственно, и возвращаются в качестве решений)
2. `student("Alexey", "Romanov", University)`. Аналогичная ситуация, только переменная `University` в этой ситуации конкретизируется значениями *BMSTU* и *MSU*
3. `student(Name, Surname, "ITMO")`. Унификация вопроса не проходит ни с одним предложением базы знаний (не проходит унификация третьего аргумента, потому что для 2 констант унификация успешно проходит только при их совпадении, а в базе знаний нет функтора `student` с 3 аргументом "ITMO")

3 Лабораторная работа №13

Задание: составить программу, то есть модель предметной области – базу знаний, объединив в ней информацию – знания:

- «Телефонный справочник»: Фамилия, №тел, Адрес – структура (Город, Улица, №дома, №кв)
- «Автомобили»: Фамилия_владельца, Марка, Цвет, Стоимость и др.
- «Вкладчики банков»: Фамилия, Банк, счет, сумма, др.

Владелец может иметь несколько телефонов, автомобилей, вкладов (Факты). Используя правила, обеспечить возможность поиска:

1. А. По № телефона найти: Фамилию, Марку автомобиля, Стоимость автомобиля (может быть несколько)

В. Используя сформированное в предыдущем пункте правило, по №телефона найти только Марку автомобиля (автомобилей может быть несколько)
2. Используя простой, не составной вопрос: по Фамилии (уникальна в городе, но в разных городах есть однофамильцы) и Городу проживания найти: Улицу, проживания, Банки, в которых есть вклады и №телефона.

Для задания 1 и 2 для одного из вариантов ответов, и для А. и для В., описать словесно порядок поиска ответа на вопрос, указав, как выбираются знания, и, при этом, для каждого этапа унификации, выписать подстановку – наибольший общий унификатор, и соответствующие примеры термов.

```
1 domains
2   surname = string
3   city, street = string
4   house, flat = integer
5   phone = string
6   address = addr(city, street, house, flat)
7   mark = string
8   color = string
9   price = integer
10  bank = string
```

```

11 id, amount = integer
12
13 predicates
14 phone(surname, phone, address)
15 car(surname, mark, color, price)
16 bank_depositor(surname, bank, id, amount)
17 car_by_phone(phone, surname, mark, price)
18 only_mark_by_phone(phone, mark)
19 data_by_surname_and_city(surname, city, street, bank, phone)
20
21 clauses
22 phone("Perestoronin", "+79999999999", addr("Moscow", "Lesnaya", 12, 2)).
23 phone("Romanov", "+71111111111", addr("Moscow", "Lesnaya", 13, 87)).
24 phone("Nitenko", "+73333333333", addr("Ekaterinburg", "Kamennaya", 13, 87)).
25 phone("Yacuba", "+66666666666", addr("Moscow", "Wall-street", 123, 87)).
26 car("Nitenko", "bmw", "green", 1000).
27 car("Yacuba", "volkswagen", "red", 10000).
28 car("Yacuba", "golfR", "black", 20000).
29 car("Romanov", "bike", "white", 10).
30 car("Perestoronin", "mercedes", "yellow", 30000).
31 bank_depositor("Nitenko", "Sber", 22, 1000).
32 bank_depositor("Yacuba", "Sber", 33, 10000).
33 bank_depositor("Yacuba", "Alfa", 44, 20000).
34 bank_depositor("Romanov", "Sper", 238, 10).
35 bank_depositor("Perestoronin", "Maze", 1, 10000).
36
37 car_by_phone(Phone, Surname, Mark, Price) :- phone(Surname, Phone, _), car(Surname,
    Mark, _, Price).
38 only_mark_by_phone(Phone, Mark) :- car_by_phone(Phone, _, Mark, _).
39 data_by_surname_and_city(Surname, City, Street, Bank, Phone) :- phone(Surname, Phone,
    addr(City, Street, _, _)), bank_depositor(Surname, Bank, _, _).
40
41 goal
42 %car_by_phone("+66666666666", Surname, Mark, Price).
43 %only_mark_by_phone("+66666666666", Mark).
44 data_by_surname_and_city("Yacuba", "Moscow", Street, Bank, Phone).

```

3.1 Поиски ответов на вопрос

Выполнены на отдельных листах и приложены к отчету.

3.2 Теоретические вопросы

3.2.1 Что такое терм?

Терм - основной элемент языка Prolog. Терм – это:

1. Константа:

- Число (целое, вещественное),
- Символьный атом (комбинация символов латинского алфавита, цифр и символа подчеркивания, начинающаяся со строчной буквы),
- Строка: последовательность символов, заключенных в кавычки.

2. Переменная:

- Именованная – обозначается комбинацией символов латинского алфавита, цифр и символа подчеркивания, начинающейся с прописной буквы или символа подчеркивания,
- Анонимная - обозначается символом подчеркивания

3. Составной терм: Это средство организации группы отдельных элементов знаний в единый объект, синтаксически представляется: $f(t_1, t_2, \dots, t_m)$, где f - функтор (отношение между объектами), t_1, t_2, \dots, t_m – термы, в том числе и составные.

3.2.2 Что такое предикат в матлогике (математике)?

Предикат (n -местный, или n -арный) — это функция с множеством значений $\{0, 1\}$ (или {ложь, истина}), определённая на множестве $M^n = (M_1, M_2, \dots, M_n)$. Таким образом, каждый набор элементов множества характеризуется либо как «истинный», либо как «ложный».

3.2.3 Что описывает предикат в Prolog?

Предикат в Prolog описывает отношение между аргументами процедуры. Процедурой в Prolog является совокупность всех правил, описывающих определенное отношение.

3.2.4 Назовите виды предложений в программе и приведите примеры таких предложений из Вашей программы. Какие предложения являются основными, а какие — не основными? Каковы: синтаксис и семантика (формальный смысл) этих предложений (основных и неосновных)?

В Prolog есть два типа предложений: правила и факты. Правило имеет вид: $A :- B_1, \dots, B_n$. A называется заголовком правила, а B_1, \dots, B_n — телом правила. Заголовок содержит некоторое знание, а тело — условие истинности этого знания. Факт является частным случаем правила — в нем отсутствует тело.

Пример факта из программы: `car('Yacuba', 'volkswagen', 'red', 10000)`.

Пример правила из программы: `car_by_phone(Phone, Surname, Mark, Price) :- phone(Surname, Phone, _), car(Surname, Mark, _, Price)`.

Основными называются предложения, не содержащие переменных. Предложения, содержащие переменные называются неосновными.

Синтаксис предложения: **заголовок (составной терм) :- тело (один или последовательность термов)**.

Предложения используются для формирования базы знаний о некоторой предметной области. Основное предложение описывает отношение конкретных объектов предметной области. Неосновное предложение описывает множество отношений, потому что переменная, входящая в предложение базы знаний, рассматривается только с квантором всеобщности.

3.2.5 Каковы назначение, виды и особенности использования переменных в программе на Prolog? Какое предложение БЗ сформулировано в более общей – абстрактной форме: содержащее или не содержащее переменных?

Переменные предназначены для обозначения некоторого неизвестного объекта предметной области. Переменные бывают именованными и анонимными. Именованные переменные уникальны в рамках предложения, а анонимная переменная – любая уникальна. В разных предложениях может использоваться одно имя переменной для обозначения разных объектов.

В ходе выполнения программы выполняется связывание переменных с различными объектами, этот процесс называется конкретизацией. Это относится только к именованным переменным. Анонимные переменные не могут быть связаны со значением.

В более общей форме сформулировано предложение, содержащее переменные, так как заранее неизвестно, каким объектом будет конкретизирована переменная.

3.2.6 Что такое подстановка?

Пусть дан терм: (X_1, X_2, \dots, X_n) . Подстановка — множество пар, вида: $\{X_i = t_i\}$, где X_i — переменная, а t_i — терм.

3.2.7 Что такое пример терма? Как и когда строится? Как Вы думаете, система строит и хранит примеры?

Пусть $\Theta = \{X_1 = t_1, X_2 = t_2, \dots, X_n = t_n\}$ — подстановка, A - терм. Результат применения подстановки к терму обозначается $A\Theta$.

Примером терма A называется терм B , если существует подстановка Θ такая, что $B = A\Theta$.

Примеры термов строятся в ходе логического вывода. Для построения примера терма его переменные конкретизируются.