



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 по лабораторной работе №
по дисциплине «Функциональное и логическое
программирование»**

Тема

Студент Зайцева А.А.

Группа ИУ7-62Б

Оценка (баллы) _____

Преподаватель Толпинская Н.Б., Строганов Ю. В.

Лабораторная работа №11

Задание

Разработать свою программу — «Телефонный справочник». Протестировать работы программы.

Решение

```
1 domains
2   surname = string.
3   name = string.
4   address = string.
5   phone = integer.
6
7 predicates
8   nondeterm record(name, surname, address, phone).
9
10 clauses
11   record("Ivanov", "Ivan", "Moscow, Ivanovskay 5", 2294055).
12   record("Petrov", "Petr", "Volgograd, Leninskaya 1", 8456372).
13   record("Sidorov", "Ivan", "Ekaterinburg, Lesnaya 2/4", 8994527).
14   record("Svidorov", "Pavel", "Moscow, Lesnaya 8", 8994558).
15   record("Ivanov", "Ivan", "Ekaterinburg, Glavnaya 4", 6994566).
16
17 goal
18   % record(Surname, _, _, Phone). =>
19   % Surname=Ivanov, Phone=2294055
20   % Surname=Petrov, Phone=8456372
21   % Surname=Sidorov, Phone=8994527
22   % Surname=Svidorov, Phone=8994558
23   % Surname=Ivanov, Phone=6994566
24   % 5 Solutions
25
26   % record("Ivanov", "Ivan", "Moscow, Ivanovskay 5", Phone). =>
27   % Phone=2294055
28   % 1 Solution
29
30   % record("Petrov", "Petr", "Volgograd, Leninskaya 2", Phone). =>
31   % No Solution
32
```

```
33 record(Surname, "Ivan", _, _).  
34 % Surname=Ivanov  
35 % Surname=Sidorov  
36 % Surname=Ivanov  
37 % 3 Solutions
```

Лабораторная работа №12

Постановка задачи

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.

Решение

```
1 domains
2   name, surname, university, group = string
3   course = integer
4
5 predicates
6   student(name, surname, university, group, course)
7
8 clauses
9   student("Alexey", "Romanov", "BMSTU", "IU7-63B", 3).
10  student("Pavel", "Perestoronin", "BMSTU", "IU7-63B", 5).
11  student("Ivan", "Cvetkov", "BMSTU", "IU7-43B", 2).
12  student("Alexey", "Voyakin", "BMSTU", "IU7-64B", 3).
13  student("Alexey", "Romanov", "MSU", "VMK-1", 1).
14  student("Pavel", "Perestoronin", "MAI", "EM-8", 2).
15  student("Ilya", "Bryukhov", "MAI", "ACS12", 2).
16  student("Vladimir", "Nesterov", "MIPT", Group, Course)
17    :- student("Alexey", "Romanov", "BMSTU", Group, _), student("Pavel", "
      Perestoronin", "BMSTU", Group, Course).
18
19 goal
20  student(Name, Surname, "MIPT", _, _).
21  student(Name, Surname, "BMSTU", _, 3).
22  student("Alexey", \_, \_, "IU7-63B", \_)
```

Данная база знаний содержит информацию о студентах (имя, фамилия, вуз, группа, курс).

С помощью первого вопроса получают все студенты, которые учатся в МФТИ. Происходит проход сверху вниз по всем фактам предиката *student(name, surname, university, group, course)* и осуществляется унификация с *study(Name, Surname, "BMSTU", _, _)*. Унифика-

цию успешно проходит один факт: *study("Vladimir", "Nesterov", "BMSTU", "IU7-63B", 5)*.

С помощью второго вопроса получают все студенты МГТУ, которые обучаются на 3 курсе. Происходит проход по всем фактам предиката *student(name, surname, university, group, course)* и осуществляется унификация с *study(Name, Surname, "BMSTU", _, 3)*. Успешно унификацию проходят факты *study("Alexey", "Romanov", "BMSTU", _, 3)* и *study("Alexey", "Voyakin", "BMSTU", _, 3)*.

С помощью второго вопроса получают все Алексеи, которые обучаются в группе ИУ7-63Б. Происходит проход по всем фактам предиката *student(name, surname, university, group, course)* и осуществляется унификация с *study(Name, _, _, "IU7-63B", _)*. Успешно унификацию проходит *study("Alexey", "Romanov", "BMSTU", "IU7-63B", 3)*.

Лабораторная работа №13

Постановка задачи

Составить программу, то есть модель предметной области — базу знаний, объединив в ней информацию — знания:

- «Телефонный справочник»: фамилия, № телефона, адрес - структура (город, улица, № дома, № квартиры);
- «Автомобили»: фамилия владельца, марка, цвет, стоимость и др.;
- «Вкладчики банков»: фамилия, банк, счет, сумма и др.

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

1.
 - по № телефона найти: фамилию, марку автомобиля, стоимость автомобиля (может быть несколько);
 - используя сформированное в пункте А 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 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 унификатор, и соответствующие примеры термов.

Решение

```

1 domains
2   surname, phone, city, street, house, apartment = string
3   address = address(city, street, house, apartment)
4   model, color, cost = string
5   bank, sum = string
6
7 predicates
8   man(surname, phone, address)
9   car(surname, model, color, cost)
10  deposit(surname, bank, sum)
11
12  car_by_phone(phone, surname, model, cost)
13  car_model_by_phone(phone, model)
14  bank_and_street_by_surname_and_city(surname, city, bank, street)
15
16 clauses
17   man("Alexey", "89096412389", address("Krasnogorsk", "Lesnaya", "17", "5"))
18   .
19   man("Vladimir", "890955550987", address("Moscow", "Sovetskaya", "134", "15
20   ")).
21   man("Mikhail", "8100500321", address("Khimki", "Lesnaya", "27", "501")).
22   man("Pavel", "87654329867", address("Moscow", "Tikhaya", "105", "52")).
23   man("Alexey", "89096421389", address("Moscow", "Baumanskaya", "170", "1"))
24   .
25
26   car("Alexey", "AMG", "Black", "100000").
27   car("Vladimir", "Volvo", "Red", "50000").
28   car("Pavel", "Nissan", "White", "709000").
29   car("Mikhail", "Cadillac", "Black", "1000000").
30
31   deposit("Alexey", "Tinkoff", "10000000000").
32   deposit("Alexey", "Sber", "1000").
33   deposit("Pavel", "Sber", "0").
34   deposit("Mikhail", "Alpha", "10").
35
36   car_by_phone(Phone, Surname, Model, Cost)
37   :- man(Surname, Phone, _), car(Surname, Model, _, Cost).
38
39   car_model_by_phone(Phone, Model) :- car_by_phone(Phone, _, Model, _).
40
41   bank_and_street_by_surname_and_city(Surname, City, Bank, Street)
42   :- man(Surname, _, address(City, Street, _, _)), deposit(Surname, Bank,
43   _).
44
45 goal
46   %car_by_phone("89096412389", Surname, Model, Cost).
47   %car_model_by_phone("89096412389", Model).
48   bank_and_street_by_surname_and_city("Alexey", "Moscow", Bank, Street).

```


Теоретическая часть

1. Что собой представляет программа на языке пролог?

Программа на **Prolog** представляет собой базу знаний и вопрос.

- **База знаний** состоит из предложений – фактов и правил, – используя которые программа выдает ответ на вопрос. Каждое предложение должно заканчиваться точкой.
 - **Правило** имеет вид: $A :- B_1, \dots, B_n$, где A – заголовок правила (составной терм, который содержит знание); B_1, \dots, B_n – тело правила (составные термы, которые содержат условия истинности этого знания), символ $:-$ – это специальный символ-разделитель.
 - **Факт** – это частный случай правила – предложение, в котором отсутствует тело (то есть тело пустое).
- **Вопрос** – это частный случай правила – предложение, которое состоит только из тела. Используется, чтобы определить, выполняется ли некоторое отношение между описанными в программе объектами. Система рассматривает вопрос как цель, к которой (к истинности которой) надо стремиться. Ответ на вопрос может оказаться логически положительным или отрицательным, в зависимости от того, может ли быть достигнута соответствующая цель.

Базис пролога – матлогика. Используется символьная обработка, декларативная методология.

2. Какова структура программы на Prolog?

Программа на Prolog состоит из следующих разделов, каждый из которых начинается со своего заголовка.

- 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.

3. Как реализуется программа на Prolog? Как формируются результаты работы программы?

Ответ: Программа на Prolog представляет собой базу знаний и вопрос. База знаний состоит из предложений – фактов и правил, которые задают истинные знания. Ответ на вопрос может оказаться логически положительным или отрицательным, в зависимости от того, может ли быть достигнута соответствующая цель.

Вопрос рассматривается системой как цель: найти возможность, исходя из базы знаний, ответить «Да» на поставленный вопрос. Вариантов ответить «Да» на может быть несколько. При поиске ответа рассматриваются альтернативные варианты и находятся все возможные решения (методом проб и ошибок) - множества значений переменных, при которых на поставленный вопрос можно ответить - «Да».

Для выполнения логического вывода используется механизм унификации, встроенный в систему. Унификация – операция, которая позволяет формализовать процесс логического вывода. С практической точки зрения - это основной вычислительный шаг, с помощью которого происходят:

- двунаправленная передача параметров процедурам,
- неразрушающее присваивание,
- проверка условий (доказательство).

В процессе работы система выполняет большое число унификаций. Попытка "увидеть одинаковость" – сопоставимость двух термов, может завершаться успехом или тупиковой ситуацией (неудачей). В последнем случае включается механизм отката к предыдущему шагу.

4. Что такое терм?

Терм - основной элемент языка Prolog. Терм – это:

1. константа (используется для обозначения объекта предметной области):

- число (целое, вещественное),
- символьный атом – комбинация символов латинского алфавита, цифр и ' _ ' (символа подчеркивания), начинающаяся со строчной буквы), используется для обозначения конкретного объекта предметной области или для обозначения конкретного отношения,
- строка – последовательность символов, заключенных в кавычки;

2. переменная:

- именованная – комбинация символов латинского алфавита, цифр и ' _ ', начинающаяся с прописной буквы или символа подчеркивания, может связываться с различными объектами (конкретизироваться),
 - анонимная - обозначается символом ' _ ', не может быть связана со значением;
3. составной терм – средство фиксации информации о том, что между объектами существует определенная связь, синтаксически представляется так: $f(t_1, t_2, \dots, t_m)$, где f - функтор (символьная константа, обозначающая имя отношения между объектами), t_1, t_2, \dots, t_m – термы (в том числе и составные), являющиеся аргументами (арность – число аргументов).

5. Что такое предикат в матлогике (математике)?

Предикат в математической логике – это (логическая) функция со множеством значений 0, 1 (истина/ложь), определенная на некотором множестве параметров. Предикат называю n -арным, если он определен на n -ой декартовой степени множества M . Таким образом, каждый набор параметров характеризуется либо как «истинный», либо как «ложный».

6. Что описывает предикат в Prolog?

Процедура – совокупность правил, описывающих определенное отношение (заголовки имеют одно и то же имя и одинаковую арность). Предикат – отношение, определяемое процедурой. Таким образом, предикат в Prolog описывает отношение между аргументами процедуры.

7. 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:

- **Правило** имеет вид: $A : -B_1, \dots, B_n$, где A – заголовок правила (составной терм, который содержит знание); B_1, \dots, B_n - тело правила (составные термы, которые содержат условия истинности этого знания), символ $:-$ – это специальный символ-разделитель;
- **Факт** – это частный случай правила – в нем отсутствует тело (то есть тело пустое).

TODO Примеры из программы:

- факт: `car("Mikhail", "Cadillac", "Black", "5000000").`

- правило: $car_by_phone(Phone, Surname, Model, Cost) :- man(Surname, Phone, _), car(Name, Model, _, Cost).$

Основными называются предложения, не содержащие переменных. Они предназначены для описания отношений, формирования базы знаний.

Предложения, содержащие переменные называются неосновными. Они предназначены для поиска ответа в базе знаний.

8. 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 передачи информации «во времени (за конечное число шагов получить результат) и пространстве (передача значений через параметры)», для повышения уровня абстракции. Они бывают именованными и анонимными.

Именованная переменная представляет собой комбинацию символов латинского алфавита, цифр и `'_'`, начинающуюся с прописной буквы или символа подчеркивания. В процессе выполнения программы именованные переменные могут связываться с различными объектами – конкретизироваться. Именованная переменная является уникальной в рамках предложения. В разных предложениях может использоваться одно имя переменной для обозначения разных объектов.

Анонимная переменная обозначается символом `'_'`. Она не может быть связана со значением. Любая анонимная переменная уникальна.

В момент фиксации система не знает, какой объект представляет переменная. Во многих языках последовательность действий при работе с переменными такая: задать значение переменной, а затем работать с самой переменной. В прологе же особый способ работы с переменными – значение переменной не задается, система сама подбирает такое(-ие) значение(-я), чтобы условие было истинным.

Предложение БЗ, содержащее переменные, сформулировано в более общей форме, так как переменные не имеют значения и могут конкретизироваться различными объектами в ходе работы системы.

TODO??

9. Что такое подстановка?

Подстановка - это функция, действующая из множества X переменных в множество T термов программы, (при этом каждой переменной $X_i \in X$ ставится в соответствие терм $t_i \in T$).

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.

10. Что такое пример терма? Как и когда строится? Как Вы думаете система строит и хранит термы?

Пусть $\omega : X_1 = t_1, X_2 = t_2, \dots, X_n = t_n$ – подстановка. Тогда результат применения подстановки к терму обозначается $A\omega$. Применение подстановки заключается в замене каждого вхождения переменной x_i на соответствующий терм. Терм B называется примером терма A , если существует такая подстановка ω , что $B = A\omega$.

В процессе выполнения программы система, используя встроенный алгоритм унификации, пытается обосновать возможность истинности вопроса, строя подстановки и примеры термов (вопроса и формулировки знания), используя базу знаний. Построение и подстановки производится путём конкретизации переменных. Сами термы хранятся в стеке.