



**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»

Лабораторная работа № 13.

По курсу «Функциональное и логическое программирование».

Студент: Власова Е. В.

Группа: ИУ7-64Б.

Преподаватели: Толпинская Н. Б.

Строганов Ю. В.

2020 г.

Теоретическая часть.

1. Что такое терм?

Терм – это

- Константа. Константы бывают численные и символьные.
- Переменная
- Составной терм. Составные термы - $f(t_1, t_2, \dots, t_n)$, f – функтор, t – термы. Составной терм может фиксировать знание о том, что между объектами T существует некоторое отношение (связь), природа которого может быть произвольной.

2. Что такое предикат в матлогике (математике)?

Предикат – утверждение, которое содержит переменные, принимающие значение 1 или 0 (истинно или ложно) в зависимости от значений переменных.

3. Что описывает предикат в Prolog?

В Prolog существует понятие процедуры. **Процедурой** называется совокупность правил, заголовки которых имеют одно и то же имя и одну и ту же аргументность (местность), т.е. это совокупность правил, описывающих одно определенное отношение. Отношение, определяемое процедурой, называется **предикатом**.

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)?

Существуют два вида предложений: факты и правила. 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 частный случай правила. Факт – это предложение, в котором отсутствует тело (т.е. тело пустое). Причем, A, B_1, \dots, B_n – это термы; символ $:-$ это специальный символ-разделитель.

Примеры из программы:

Факт:

<code>cars("Petrov", "Moscow", "TOYOTA", "RED", 256000)</code>
--

Правило:

```
find_by_phone_number_car(SURNAME, PHONE, MARKA,  
PRICE):-  
info(SURNAME, PHONE, address(CITY, _, _, _)),  
cars(SURNAME, CITY, MARKA, _, PRICE).
```

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?

На момент фиксации в программе, переменные обозначают некоторый (любой), неизвестный объект предметной области. Значения переменных подбираются системой в процессе унификации (сопоставления) двух составных термов из программы на Prolog (для этого алгоритм унификации автоматически и многократно запускается системой).

В более общей форме сформулировано предложение, содержащее переменные.

6. Что такое подстановка?

Пусть дан терм: $A(X_1, X_2, \dots, X_n)$

Подстановкой называется множество пар, вида: $\{X_i = t_i\}$, где X_i – переменная, а t_i – терм.

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, тогда результат применения подстановки к терму обозначается: $A\theta$. Применение подстановки заключается в замене каждого вхождения переменной X_i на соответствующий терм. Терм B называется примером терма A , если существует такая подстановка θ , что $B = A\theta$. 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.

Практическая часть.

Составить программу, т.е. 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 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 задания1 и задания2:

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.

Листинг программы:

```
domains
home = address(CITY, STREET, HOUSE, FLAT)
SURNAME = string
```

```

PHONE = string
CITY, STREET = string
HOUSE, FLAT = integer
MARKA, COLOR = string
PRICE = integer
BANK = string
NUMBER, MONEY = integer

predicates
    info(SURNAME, PHONE, home)
    cars(SURNAME, CITY, MARKA, COLOR, PRICE)
    banks(SURNAME, CITY, BANK, NUMBER, MONEY)
    find_by_phone_number_car(SURNAME, PHONE, MARKA, PRICE)
    find_by_phone_number_mark(PHONE, MARKA)

clauses
    info("Petrov", "88885353535", address("Moscow", "Saykin street", 10, 143)).
%PERSON 1
    info("Ivanov", "81111111111", address("Kursk", "Non street", 1, 10)).
%PERSON 2
    info("Krot", "89995353566", address("Moscow", "Nwot street", 5, 60)).
%PERSON 3
    info("Petrov", "88005353535", address("Kursk", "Greds street", 6, 45)).
%PERSON 4
    cars("Petrov", "Moscow", "TOYOTA", "RED", 256000). %PERSON 1
    cars("Petrov", "Moscow", "BMW", "BLACK", 356000). %PERSON 1

    cars("Ivanov", "Kursk", "LADA", "GREEN", 100000). %PERSON 2

    cars("Krot", "Moscow", "Merc", "BLUE", 100000). %PERSON 3

    cars("Petrov", "Kursk", "Audio", "WHITE", 456000). %PERSON 4
    cars("Petrov", "Kursk", "LADA", "RED", 156000). %PERSON 4

    banks("Petrov", "Moscow", "Sberbank", 356, 50000). %PERSON 1
    banks("Ivanov", "Kursk", "Tinkoff", 357, 9000). %PERSON 2
    banks("Krot", "Moscow", "Rocket-bank", 20, 2000). %PERSON 3
    banks("Petrov", "Kursk", "NEW-BANK", 333, 50000). %PERSON 4
    banks("Petrov", "Kursk", "OLD-BANK", 555, 30000). %PERSON 4

    find_by_phone_number_car(SURNAME, PHONE, MARKA, PRICE):-
info(SURNAME, PHONE, address(CITY, _, _, _)),
cars(SURNAME, CITY, MARKA, _, PRICE).

```

```

find_by_phone_number_mark(PHONE, MARKA):-
find_by_phone_number_car(_, PHONE, MARKA, _).
goal
find_by_phone_number_mark("88005353535", MARKA).

```

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

```

find_by_phone_number_car(SURNAME, PHONE, MARKA, PRICE):-
info(SURNAME, PHONE, address(CITY, _, _, _)),
cars(SURNAME, CITY, MARKA, _, PRICE).


```

1. У человека одна машина:

```

Goal
find_by_phone_number_car(SURNAME, "8111111111", MARKA,
PRICE).

```

 [Inactive D:\ph\BIN\WIN\32\Obj\goal\$000.exe]

SURNAME=Ivanov, MARKA=LADA, PRICE=100000


1 Solution

2. У человека несколько машин:

```

goal
find_by_phone_number_car(SURNAME, "88885353535", MARKA,
PRICE).

```


 [Inactive D:\ph\BIN\WIN\32\Obj\goal\$000.exe]

SURNAME=Petrov, MARKA=TOYOTA, PRICE=256000

SURNAME=Petrov, MARKA=BMW, PRICE=356000

2 Solutions

3. У человека нет машин:

 [Inactive D:\ph\BIN\WIN\32\Obj\goal\$000.exe]

No Solution

Для примера 1:

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат
	(Подстановка) find_by_phone_number_car(SURNAME, PHONE, MARKA, PRICE) = find_by_phone_number_car(SURNAME, "8111111111", MARKA, PRICE).	Прямой ход SURNAME=SURNAME, PHONE = 8111111111 MARKA = MARKA, PRICE = PRICE
	info(SURNAME, PHONE, address(CITY, _, _, _)),	Результат сравнения термов false, прямой ход


	<code>cars(SURNAME, CITY, MARKA, _, PRICE)</code> (подстановка) -> <code>i</code>	
	<code>info(SURNAME, PHONE, address(CITY, _, _)),</code> <code>cars(SURNAME, CITY, MARKA, _, PRICE)</code> (подстановка) -> <code>i</code>	Результат сравнения термов true, откат
	(Подстановка) <code>cars(SURNAME, CITY, MARKA, _, PRICE)</code> <code>=</code> <code>cars(«Ivanov», «Kursk», MARKA, _, PRICE)</code>	Прямой ход, SURNAME = Ivanov, CITY = KURSK
	<code>cars(SURNAME, CITY, MARKA, _, PRICE)</code> <code>info("Krot", "89995353566",</code> <code>address("Moscow", "Nwot street", 5, 60))</code> Разные функторы)	Результат сравнения термов false, прямой ход
	<code>cars(SURNAME, CITY, MARKA, _, PRICE)</code> <code>info("Petrov", "88005353535",</code> <code>address("Kursk", "Greds street", 6, 45))</code> Разные функторы)	Результат сравнения термов false, прямой ход
	<code>cars(SURNAME, CITY, MARKA, _, PRICE)</code> (Подстановка) -> <code>cars(Ivanov, Kursk, TOYOTA, _, 256000)</code>	Результат сравнения термов false, прямо ход
	<code>cars(SURNAME, CITY, MARKA, _, PRICE)</code> (Подстановка) -> <code>c</code>	Результат сравнения термов false, прямой ход
	<code>cars(SURNAME, CITY, MARKA, _, PRICE)</code> (Подстановка) -> <code>c</code>	Результат сравнения термов true, откат
	<code>cars(SURNAME, CITY, MARKA, _, PRICE)</code> (Подстановка) -> <code>cars("Ivanov", "Kursk", "Merc", _, 100000)</code>	Результат сравнения термов false, прямго ход
...	Аналогично далее	...

Б) Используя сформированное в пункте **а)** правило, по № телефона найти: только Марку автомобиля (автомобилей может быть несколько).

`find_by_phone_number_mark(PHONE, MARKA):-`
`find_by_phone_number_car(_, PHONE, MARKA, _).`

1. У человека одна машина:

`find_by_phone_number_mark("8111111111", MARKA).`


 [Inactive D:\ph\BIN\WIN\32\Obj\goal\$000.exe]

MARKA=LADA

1 Solution

2. У человека несколько машин:

`find_by_phone_number_mark("8888535353", MARKA).`

 [Inactive D:\ph\BIN\WIN\32\Obj\goal\$000.exe]


MARKA=Audio

MARKA=LADA

2 Solutions

3. У человека нет машин:

`find_by_phone_number_mark("88885353000", MARKA).`

 [Inactive D:\ph\BIN\WIN\32\Obj\goal\$000.exe]

No Solution

Для примера 2:

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат
	(Подстановка) <code>find_by_phone_number_mark(88885353535, MARKA)</code> <code>find_by_phone_number_car(88885353535, MARKA)</code>	Прямой ход PHONE = 88885353535 MARKA = MARKA
	<code>find_by_phone_number_car(SURNAME, PHONE, MARKA, PRICE)</code> = <code>find_by_phone_number_car(_,88885353535, MARKA, _)</code>	Прямой ход, PHONE = 88885353535 MARKA = MARKA
	<code>info(SURNAME, PHONE, address(CITY, _, _)),</code> <code>cars(SURNAME, CITY, MARKA, _, PRICE)</code> (подстановка) -> <code>info(SURNAME, PHONE, address(CITY, _, _))</code> = <code>i</code> <code>n</code>	Результат сравнения термов true, откат
	(Подстановка) <code>cars(SURNAME, CITY, MARKA, _, PRICE)</code> = <code>cars(«Petrov», «Moscow», MARKA, _, PRICE)</code>	Прямой ход SURNAME = PETROV CITY = MOSCOW
	<code>cars(SURNAME, CITY, MARKA, _, PRICE)</code> <code>info("Ivanov", "8111111111",</code>	Результат сравнения термов false, прямой ход

	address("Kursk", "Non street", 1, 10)) Разные функторы	
	Аналогично с 4	Результат сравнения термов false, прямой ход
	cars(SURNAME, CITY, MARKA, _, PRICE) (подстановка)-> cars(«Petrov», «Moscow», "TOYOTA", _, 256000)	Результат сравнения термов true, откат
	cars(SURNAME, CITY, MARKA, _, PRICE) (подстановка)-> cars(«Petrov», «Moscow», "BMW", _, 356000)	Результат сравнения термов true, откат
	cars(SURNAME, CITY, MARKA, _, PRICE) (подстановка)-> cars(«Petrov», «Moscow», " LADA", _, 100000)	Результат сравнения термов false, прямой ход
	cars(SURNAME, CITY, MARKA, _, PRICE) (подстановка)-> cars(«Petrov», «Moscow», " Merc", _, 100000)	Результат сравнения термов false, прямой ход
...	Аналогично далее	...


В) 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.

```
goal
info("Ivanov", PHONE, address("Kursk", STREET, _, _)), banks("Ivanov",
BANK, _, _).
```

Примеры работы:


1. Когда у человека один счёт в банке:

Система вывод всевозможные ответы «Да» на заданный вопрос.

 [Inactive D:\ph\BIN\WIN\32\Obj\goal\$000.exe]

SURNAME=Ivanov, CITY=Kursk, PHONE=8111111111, STREET=Non street, BANK=Tinkoff
1 Solution


2. У человека несколько несколько счетов:

 [Inactive D:\ph\BIN\WIN\32\Obj\goal\$000.exe]

SURNAME=Petrov, CITY=Kursk, PHONE=88005353535, STREET=Greds street, BANK=NEW-BANK
SURNAME=Petrov, CITY=Kursk, PHONE=88005353535, STREET=Greds street, BANK=OLD-BANK
2 Solutions

3. У человека нет счетов:

В данном случае система не найдёт вариант ответить «Да».

 [Inactive D:\ph\BIN\WIN\32\Obj\goal\$000.exe]

No Solution