**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(t1,t2,...,tn), ф-функтор, т-термы. Составной терм может фиксировать знание о том, что между объектами Т существует некоторое отношение(связь), природа которого может быть произвольной.

1. **Что такое предикат в матлогике (математике)?**

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

1. **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 арность (местность), т.е. это совокупность правил, описывающих одно определенное отношение. Отношение, определяемое процедурой, называется **предикатом**.

1. **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 :- B1,… , Bn. A называется заголовком правила, а B1,..., Bn – телом правила.   
Факт – это частный случай правила. Факт – это предложение, в котором отсутствует тело (т.е. тело пустое).   
Причем, A, B1,... , Bn – это термы; символ ":-" это специальный символ-разделитель.

Примеры из программы:

Факт:

|  |
| --- |
| cars("Petrov", "Moscow", "TOYOTA", "RED", 256000) |

Правило:

|  |
| --- |
| find\_by\_phone\_number\_car(SURNAME, PHONE, MARKA, PRICE):-  info(SURNAME, PHONE, address(CITY, \_, \_, \_)), cars(SURNAME, CITY, MARKA, \_, PRICE). |

1. **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.

1. **Что такое подстановка?**

Пусть дан терм: *A*(*X*1*,X*2*,...,Xn*)

Подстановкой называется множество пар, вида: {*Xi* = *ti*}, где *Xi* – переменная, а *ti* – терм.

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

Пусть *θ* = {*X*1 = *t*1*,X*2 = *t*2*,...,Xn* = *tn*} – подстановка, тогда результат применения подстановки к терму обозначается: *Aθ*. Применение подстановки заключается в замене каждого вхождения переменной *Xi* на соответствующий терм. Терм *B* называется примером терма *A*, если существует такая подстановка *θ*, что *B* = *Aθ*. 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),

1. 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:**

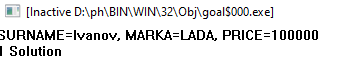
|  |
| --- |
| domains  home = address(CITY, STREET, HOUSE, FLAT)  SURNAME = string  PHONE = string  CITY, STREET = string  HOUSE, FLAT = integer  MARKA, COLOR = string  PRICE = integer  BANK = string  NUMBER, MONEY = integer  predicates  info(SURNAME, PHONE, home)  cars(SURNAME, CITY, MARKA, COLOR, 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). |

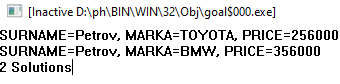
1. У человека одна машина:

|  |
| --- |
| Goal  find\_by\_phone\_number\_car(SURNAME, "81111111111", MARKA, PRICE). |



1. У человека несколько машин:

|  |
| --- |
| goal  find\_by\_phone\_number\_car(SURNAME, "88885353535", MARKA, PRICE). |



1. У человека нет машин:



Для примера 1:

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат |
| 0 | (Подстановка)  find\_by\_phone\_number\_car(SURNAME, PHONE, MARKA, PRICE)  =  find\_by\_phone\_number\_car(SURNAME, "81111111111", MARKA, PRICE). | Прямой ходб SURNAME=SURNAME,  PHONE = 81111111111  MARKA = MARKA, PRICE = PRICE |
| 1 | info(SURNAME, PHONE, address(CITY, \_, \_, \_)),  cars(SURNAME, CITY, MARKA, \_, PRICE)  (подстановка) ->  info(«Petrov», «81111111111», address(«Moscow», \_, \_, \_)) | Результат сравнения термов false, прямой ход |
| 2 | info(SURNAME, PHONE, address(CITY, \_, \_, \_)),  cars(SURNAME, CITY, MARKA, \_, PRICE)  (подстановка) ->  info(«Ivanov», «81111111111», address(«Kursk», \_, \_, \_)) | Результат сравнения термов true, откат |
| 3 | (Подстановка)  cars(SURNAME, CITY, MARKA, \_, PRICE)  =  cars(«Ivanov», «Kursk», MARKA, \_, PRICE) | Прямой ход, SURNAME = Ivanov, CITY = KURSK |
| 4 | cars(SURNAME, CITY, MARKA, \_, PRICE)  info("Krot", "89995353566", address("Moscow", "Nwot street", 5, 60))  (Разные функторы) | Результат сравнения термов false, прямой ход |
| 5 | cars(SURNAME, CITY, MARKA, \_, PRICE)  info("Petrov", "88005353535", address("Kursk", "Greds street", 6, 45))  (Разные функторы) | Результат сравнения термов false, прямой ход |
| 6 | cars(SURNAME, CITY, MARKA, \_, PRICE)  (Подстановка) ->  cars(Ivanov, Kursk, TOYOTA, \_, 256000) | Результат сравнения термов false, прямо ход |
| 7 | cars(SURNAME, CITY, MARKA, \_, PRICE)  (Подстановка) ->  cars(Ivanov, Kursk, BMW, \_, 356000) | Результат сравнения термов false, прямой ход |
| 8 | cars(SURNAME, CITY, MARKA, \_, PRICE)  (Подстановка) ->  cars("Ivanov", "Kursk", "LADA", \_, 100000) | Результат сравнения термов true, откат |
| 9 | cars(SURNAME, CITY, MARKA, \_, PRICE)  (Подстановка) ->  cars("Ivanov", "Kursk", "Merc", \_, 100000) | Результат сравнения термов false, прямqо 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, \_). |

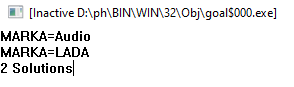
1. У человека одна машина:

|  |
| --- |
| find\_by\_phone\_number\_mark("81111111111", MARKA). |



1. У человека несколько машин:

|  |
| --- |
| find\_by\_phone\_number\_mark("88885353535", MARKA). |



1. У человека нет машин:

|  |
| --- |
| find\_by\_phone\_number\_mark("88885353000", MARKA). |



Для примера 2:

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат |
| 0 | (Подстановка)  find\_by\_phone\_number\_mark(88885353535, MARKA)  =  find\_by\_phone\_number\_car(88885353535, MARKA) | Прямой ход  PHONE = 88885353535  MARKA = MARKA |
| 1 | find\_by\_phone\_number\_car(SURNAME, PHONE, MARKA, PRICE)  =  find\_by\_phone\_number\_car(\_,88885353535, MARKA, \_) | Прямой ход,  PHONE = 88885353535  MARKA = MARKA |
| 2 | info(SURNAME, PHONE, address(CITY, \_, \_, \_)),  cars(SURNAME, CITY, MARKA, \_, PRICE)  (подстановка) ->  info(SURNAME, PHONE, address(CITY, \_, \_, \_))  =  info(«Petrov», «88885353535», address(«Moscow», \_, \_, \_)) | Результат сравнения термов true, откат |
| 3 | (Подстановка)  cars(SURNAME, CITY, MARKA, \_, PRICE)  =  cars(«Petrov», «Moscow», MARKA, \_, PRICE) | Прямой ход  SURNAME = PETROV  CITY = MOSCOW |
| 4 | cars(SURNAME, CITY, MARKA, \_, PRICE)  info("Ivanov", "81111111111", address("Kursk", "Non street", 1, 10))  Разные функторы | Результат сравнения термов false, прямой ход |
| 5-7 | Аналогично с 4 | Результат сравнения термов false, прямой ход |
| 8 | cars(SURNAME, CITY, MARKA, \_, PRICE)  (подстановка)->  cars(«Petrov», «Moscow», "TOYOTA", \_, 256000) | Результат сравнения термов true, откат |
| 9 | cars(SURNAME, CITY, MARKA, \_, PRICE)  (подстановка)->  cars(«Petrov», «Moscow», "BMW", \_, 356000) | Результат сравнения термов true, откат |
| 10 | cars(SURNAME, CITY, MARKA, \_, PRICE)  (подстановка)->  cars(«Petrov», «Moscow», " LADA", \_, 100000) | Результат сравнения термов false, прямой ход |
| 11 | cars(SURNAME, CITY, MARKA, \_, PRICE)  (подстановка)->  cars(«Petrov», «Moscow», " Merc", \_, 100000) | Результат сравнения термов false, прямой ход |
| … | Аналогично далее | … |

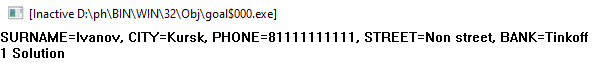
**В)** Используя простой, не составной вопрос: по Фамилии (уникальна в городе, но в разных городах есть однофамильцы) и Городу проживания найти: Улицу проживания, Банки, в которых есть вклады и №телефона.

|  |
| --- |
| goal  info("Ivanov", PHONE, address("Kursk", STREET, \_, \_)), banks("Ivanov", BANK, \_, \_). |

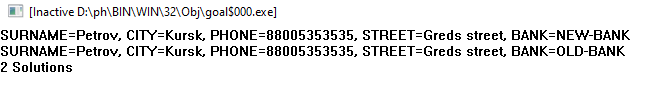
Примеры работы:

1. Когда у человека один счёт в банке:

Система вывод всевозможные ответы «Да» на заданный вопрос.



1. У человека несколько несколько счетов:



1. У человека нет счетов:

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

