



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 №13*

*По курсу: «Функциональное и логическое  
программирование»*

Студентка ИУ7-65Б  
Оберган Т.М

Преподаватели  
Толпинская Н.Б  
Строганов Ю.В

*Москва, 2020 г.*

### **Задание:**

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.

## **Вопросы:**

### **Что такое терм?**

Термы – слова, которые описывают сущности изучаемого мира.

### **Что такое предикат в матлогике (математике)?**

Предикат – функция с множеством значений  $\{0, 1\}$ , определенная на множестве  $M = M_1 * M_2 * \dots * M_n$ .

### **Что описывает предикат в Prolog?**

Предикаты – слова, которые описывают атрибуты/свойства сущностей, их поведение и отношение.

**Виды предложений в программе, примеры таких предложений.**

- Факты – утверждения, которые всегда истинны. Предложения с пустым телом.  
`investors("Autumn", "Sberbank", public, 10).`
- Правила- утверждения, истинность которых зависит от некоторых условий. Имеют голову и непустое тело.  
`car_by_lname(Phone, Lname, Model, Price) :-  
 phone_book(Lname, Phone, _),  
 car(Lname, Model, _, Price).`
- Вопросы – с их помощью пользователь спрашивает систему о том, какие утверждения являются истинными. Предложения, состоящие только из тела.  
`car_by_lname("+123", Last_name, Model, Price).`

### **Какие предложения являются основными, а какие – не основными?**

Если составные термы, факты, правила и вопросы не содержат переменных, то они называются основными.

Составные термы, факты, правила и вопросы в момент фиксации в программе могут содержать переменные, тогда они называются неосновными.

### **Каков синтаксис этих предложений**

Переменные обозначаются идентификаторами, начинающимися с заглавной буквы.

### **Каково назначение переменных**

Переменные обозначают некоторый неизвестный объект из некоторого множества объектов.

### **Особенности использования переменных в программе на Prolog?**

В момент фиксации утверждений в программе не имеют значения. Значения для переменных могут быть установлены системой только в процессе поиска ответа на вопрос, т.е. реализации программы.

### **Виды переменных**

- Именованные – есть имя – комбинация символов.
- Анонимные – нет имени – символ подчеркивания.

или

- Связанная (конкретизирована) – имеется объект, который в данный момент обозначает данная переменная.
- Свободная (не конкретизирована)

### **Какое предложение БЗ сформулировано в более общей – абстрактной форме: содержащее или не содержащее переменных?**

Не содержащее переменных.

### **Что такое подстановка?**

Подстановка - множество пар вида  $\{X_i = t_i\}$ , где  $X_i$  – переменная, а  $t_i$  – терм.

### **Что такое пример терма?**

Терм В называется примером терма А, если существует такая подстановка  $\Theta$ , что  $B = A\Theta$ , где  $A\Theta$  – результат применения подстановки к терму.

### **Как и когда строится?**

Примеры строятся во время алгоритма унификации.

## Текст программы:

domains

lname, phone, city, street = symbol.

building, apartment = integer.

model, color, bank, account = symbol.

price, amount = integer.

address = address(city, street, building, apartment).

predicates

phone\_book(lname, phone, address).

car(lname, model, color, price).

investors(lname, bank, account, amount).

car\_by\_lname(phone, lname, model, price).

address\_and\_bank(lname, city, street, bank, phone).

clauses

phone\_book("Winter", "+123", address("Moscow", "Postal", 1, 2)).

phone\_book("Spring", "+111", address("Spb", "Somestr", 2, 4)).

phone\_book("Summer", "+345", address("Sevastopol", "Anotherstr", 4, 8)).

phone\_book("Autumn", "+333", address("Moscow", "Postal", 8, 16)).

car("Winter", "IDK", red, 200000).

car("Winter", "SmallOne", black, 150).

car("Summer", "Apple", green, 1000000).

investors("Winter", "RNCB", private, 999999999).

investors("Autumn", "Sberbank", public, 10).

investors("Autumn", "Raiffeisen", public, 10000).

car\_by\_lname(Phone, Lname, Model, Price) :-

phone\_book(Lname, Phone, \_),

car(Lname, Model, \_, Price).

address\_and\_bank(Lname, City, Street, Bank, Phone) :-

phone\_book(Lname, Phone, address(City, Street, \_, \_)),

investors(Lname, Bank, \_, \_).

goal

% 1.a find by phone number

%car\_by\_lname("+123", Last\_name, Model, Price). %(1)

%car\_by\_lname("+333", Last\_name, Model, Price).

%car\_by\_lname("+000", Last\_name, Model, Price).

% 1.b find car model using 1.a

%car\_by\_lname("+123", \_, Model, \_).

%car\_by\_lname("+333", \_, Model, \_). %(2)

%car\_by\_lname("+000", \_, Model, \_).

% 2

%address\_and\_bank("Winter", "Moscow", Street, Bank, Phone).

%address\_and\_bank("Autumn", "Moscow", Street, Bank, Phone).


%address\_and\_bank("Spring", "SPB", Street, Bank, Phone).

Не совсем понятно как во втором задании можно обойтись без составного правила/вопроса. А если ограничение только на вопрос, то не понимаю в чем смысл ограничения.


## Результаты работы программы:

1.a


---

 [Inactive D:\VIP52\BIN\WIN\32\TestGoal\Obj\goal\$000.exe]  
**Last\_name=Winter, Model=IDK, Price=200000**  
**Last\_name=Winter, Model=SmallOne, Price=150**  
**2 Solutions**

---

 [Inactive D:\VIP]  
**No Solution**


---

 [Inactive D:\VIP]  
**No Solution**


(1)

1.b


---

 [Inactive D:\VIP52\BIN\  
**Model=IDK**  
**Model=SmallOne**  
**2 Solutions**

---


 [Inactive D:\VIP]  
**No Solution**

---


 [Inactive D:\VIP]  
**No Solution**

2.


---

 [Inactive D:\VIP52\BIN\WIN\32\TestGoal\Obj\g  
**Street=Postal, Bank=RNCB, Phone=+123**  
**1 Solution**

---

 [Inactive D:\VIP52\BIN\WIN\32\TestGoal\Obj\goal\$000.exe]  
**Street=Postal, Bank=Sberbank, Phone=+333**  
**Street=Postal, Bank=Raiffeisen, Phone=+333**  
**2 Solutions**

---

 [Inactive D:\VIP]  
**No Solution**

## Порядок поиска ответа на вопрос:

Для одного из вариантов ответов 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. Описание рекомендуется оформить в виде таблицы.

**Унификация** – процесс сравнения и связывания.

1.a

car\_by\_lname("+123", Last\_name, Model, Price).

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	car_by_lname(Phone, Lname, Model, Price) и car_by_lname("+123", Last_name, Model, Price) сравнимы. Связываются: Phone = "+123"	Car_by_lname убирается из стека. Помещаем в стек phone_book(Lname, Phone, _) и car(Lname, Model, _, Price).
2	phone_book(Lname, "+123", _) и phone_book("Winter", "+123", address("Moscow", "Postal", 1, 2)) сравнимы. Связываются: Lname = "Winter"	phone_book убирается из стека
3	car("Winter", Model, _, Price) и car("Winter", "IDK", red, 200000) сравнимы. Связываются: Model = "IDK" Price = 200000	car убирается из стека стек вопросов пуст, <b>ВЫВОДЯТСЯ</b> Lname, Model, Price: "Winter", "IDK", 200000 в стек заносится последний убранный вопрос: car(Lname, Model, _, Price), а Model и Price развязываются



4	<p>car("Winter", Model, _, Price) и car("Winter", "SmallOne", black, 150) сравнимы.</p> <p>Связываются: Model = "SmallOne" Price = 150</p>	<p>car убирается из стека стек вопросов пуст, <b>ВЫВОДЯТСЯ</b> Lname, Model, Price: "Winter", "SmallOne", 150</p> <p>в стек заносится последний убранный вопрос: car(Lname, Model, _, Price), а Model и Price развязываются</p>
5	<p>car("Winter", Model, _, Price) и car("Summer", "Apple", green, 1000000) не унифицируемы.</p>	<p>Откат.</p> <p>Т.к. все факты car пройденны, в стек заносится последний вопрос (который перед car, т.к. car уже в стеке): phone_book(Lname, Phone, _), а Lname развязывается</p>
6	<p>phone_book(Lname, "+123", _) и phone_book("Spring", "+111", address("Spb", "Somestrt", 2, 4)) не унифицируемы.</p>	
7	<p>phone_book(Lname, "+123", _) и phone_book("Summer", "+345", address("Sevastopol", "Anotherstr", 4, 8)) не унифицируемы.</p>	
8	<p>phone_book(Lname, "+123", _) и phone_book("Autumn", "+333", address("Moscow", "Postal", 8, 16)) не унифицируемы.</p>	<p>Откат.</p> <p>Т.к. все факты phone_book пройденны, в стек заносится последний вопрос (который перед phone_book): car_by_lname(Phone,</p>

		Lname, Model, Price) и развязывается Phone.
8		Т.к. дошли до последнего car_by_lname, а откатываться не к чему, система завершает свою работу.

1.b

car\_by\_lname("+333", \_, Model, \_).

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	car_by_lname(Phone, Lname, Model, Price) и car_by_lname("+333", Last_name, Model, Price) сравнимы.  Связываются: Phone = "+333"	Car_by_lname убирается из стека.  Помещаем в стек phone_book(Lname, Phone, _) и car(Lname, Model, _, Price).
2	phone_book(Lname, "+333", _) и phone_book("Winter", "+123", address("Moscow", "Postal", 1, 2)) не унифицируемы.	
3	phone_book(Lname, "+333", _) и phone_book("Summer", "+345", address("Sevastopol", "Anotherstrt", 4, 8)) не унифицируемы.	
4	phone_book(Lname, "+333", _) и phone_book("Autumn", "+333", address("Moscow", "Postal", 8, 16)) унифицируемы.  Lname = "Autumn"	phone_book убирается из стека.

5	car("Autumn", Model, _, Price) и car("Winter", "IDK", red, 200000) не унифицируемы.	
6	car("Autumn", Model, _, Price) и car("Winter", "SmallOne", black, 150) не унифицируемы.	
7	car("Autumn", Model, _, Price) и car("Summer", "Apple", green, 1000000) не унифицируемы.	Откат. Т.к. все факты car пройденны, в стек возвращается: phone_book(Lname, Phone, _), а Lname развязывается
8		Откат. Т.к. все факты phone_book пройденны, в стек возвращается: car_by_lname(Phone, Lname, Model, Price) и развязывается Phone.
9		Т.к. дошли до последнего car_by_lname, а откатываться не к чему, система выводит <b>NoSolution</b> (т.к. не было найдено решения) и завершает свою работу.

для вопроса `address_and_bank("Winter", "Moscow", Street, Bank, Phone)`.

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	<code>address_and_bank("Winter", "Moscow", Street, Bank, Phone)</code> и <code>address_and_bank(Lname, City, Street, Bank, Phone)</code> унифицируемы. Lname = "Winter" City = "Moscow"	<code>address_and_bank</code> убирается из стека. в стек помещаются <code>phone_book(Lname, Phone, address(City, Street, _, _))</code> и <code>investors(Lname, Bank, _, _)</code> .
2	<code>phone_book("Winter", Phone, address("Moscow", Street, _, _))</code> и <code>phone_book("Winter", "+123", address("Moscow", "Postal", 1, 2))</code> унифицируются. Phone = "+123" Street = "Postal"	<code>phone_book</code> убирается из стека.
3	<code>investors("Winter", Bank, _, _)</code> и <code>investors("Winter", "RNCB", private, 999999999)</code> унифицируются. Bank = "RNCB"	<code>investors</code> убирается из стека. стек пуст, <b>выводим решение:</b> Phone = "+123" Street = "Postal" Bank = "RNCB" Откат.
4	<code>investors("Winter", Bank, _, _)</code> и <code>investors("Autumn", "Sberbank", public, 10)</code> не унифицируются	
5	<code>investors("Winter", Bank, _, _)</code> и <code>investors("Autumn", "Raiffeisen", public, 10000)</code> не унифицируются	Дошли до конца фактов <code>investors</code> . Откат. В стек возвращается <code>phone_book</code> .

6	phone_book("Winter", Phone, address("Moscow", Street, _, _)) и не унифицируются.	
7	phone_book("Winter", Phone, address("Moscow", Street, _, _)) и не унифицируются.	
8	phone_book("Winter", Phone, address("Moscow", Street, _, _)) и не унифицируются.	Дошли до конца phone_book.  Откат.  В стек возвращается address_and_bank.
9		Дошли до конца address_and_bank, откатываться не к чему, система завершает работу.