|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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),

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.

**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 = M1 \* M2 \* … \* Mn.

**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 {Xi = ti}, где Xi – переменная, а ti – терм.

**Что такое пример терма?**

Терм В называется примером терма А, если существует такая подстановка Θ, что В=АΘ, где АΘ – 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", "Somestrt", 2, 4)).

phone\_book("Summer", "+345", address("Sevastopol", "Anotherstrt", 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).

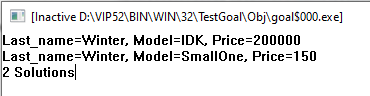
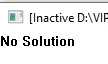
%address\_and\_bank("Autumn", "Moscow", Street, Bank, Phone).

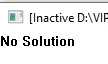
%address\_and\_bank("Spring", "SPB", Street, Bank, Phone).

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.

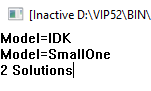
**Результаты работы программы:**

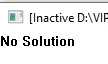
1.a

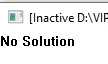
 (1)  




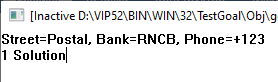
1.b

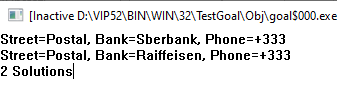


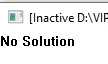




2.







**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.а

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,  **выводятся**  Lname, Model, Price: “Winter”, “IDK”, 200000  в стек заносится последний убранный вопрос:  car(Lname, Model, \_, Price), а Model и Price развязываются |
| 4 | car(“Winter”, Model, \_, Price) и car("Winter", "SmallOne", black, 150) сравнимы.  Связываются: Model = “SmallOne” Price = 150 | car убирается из стека  стек вопросов пуст,  **выводятся**  Lname, Model, Price: “Winter”, “SmallOne”, 150  в стек заносится последний убранный вопрос:  car(Lname, Model, \_, Price), а Model и Price развязываются |
| 5 | car(“Winter”, Model, \_, Price) и car("Summer", "Apple", green, 1000000) не унифицируемы. | Откат.  Т.к. все факты car пройдены, в стек заносится последний вопрос (который перед car, т.к. car уже в стеке): phone\_book(Lname, Phone, \_), а Lname развязывается |
| 6 | phone\_book(Lname, “+123”, \_) и phone\_book("Spring", "+111", address("Spb", "Somestrt", 2, 4)) не унифицируемы. |  |
| 7 | phone\_book(Lname, “+123”, \_) и phone\_book("Summer", "+345", address("Sevastopol", "Anotherstrt", 4, 8)) не унифицируемы. |  |
| 8 | phone\_book(Lname, “+123”, \_) и phone\_book("Autumn", "+333", address("Moscow", "Postal", 8, 16)) не унифицируемы. | Откат.  Т.к. все факты phone\_book пройдены, в стек заносится последний вопрос (который перед phone\_book): car\_by\_lname(Phone, 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 пройдены, в стек возвращается: phone\_book(Lname, Phone, \_), а  Lname развязывается |
| 8 |  | Откат.  Т.к. все факты phone\_book пройдены, 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 **NoSolution** (т.к. не было найдено решения) и завершает свою работу. |

2

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 | address\_and\_bank("Winter", "Moscow", Street, Bank, Phone) и address\_and\_bank(Lname, City, Street, Bank, Phone) унифицируемы.  Lname = “Winter” City = “Moscow” | address\_and\_bank убирается из стека.  в стек помещаются  phone\_book(Lname, Phone, address(City, Street, \_, \_)) и investors(Lname, Bank, \_, \_). |
| 2 | phone\_book(“Winter”, Phone, address(“Moscow”, Street, \_, \_)) и phone\_book("Winter", "+123", address("Moscow", "Postal", 1, 2)) унифицируются.  Phone = “+123” Street = “Postal” | phone\_book убирается из стека. |
| 3 | investors("Winter", Bank, \_, \_) и investors("Winter", "RNCB", private, 999999999) унифицируются.  Bank = “RNCB” | investors убирается из стека.  стек пуст, **выводим** **решение**: Phone = “+123” Street = “Postal” Bank = “RNCB”  Откат. |
| 4 | investors("Winter", Bank, \_, \_) и investors("Autumn", "Sberbank", public, 10) не унифицируются |  |
| 5 | investors("Winter",, Bank, \_, \_) и investors("Autumn", "Raiffeisen", public, 10000) не унифицируются | Дошли до конца фактов investors. Откат.  В стек возвращается phone\_book. |
| 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. |