



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»*

Студент ИУ7-64Б
Лозовский А.А.

Преподаватель
Толпинская Н.Б

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).

1. Используя правила, обеспечить возможность поиска:
 - а. По № телефона найти: Фамилию, Марку автомобиля, Стоимость автомобиля (может быть несколько),
 - б. 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, № телефона.

Для заданий 1 и 2: для одного из вариантов ответов, и для «а» и для «б», 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.

Ответы на вопросы:

1. Что такое терм?

Терм – основной элемент языка. Терм – это константа (число, символьный атом, строка), переменная (именованная, анонимная), составной терм.

2. Что такое предикат в матлогике (математике)?

Предикат – это логическая формула от одного или нескольких аргументов. Можно сказать, что предикат – это функция, отображающая множество произвольной природы в множество {ложно, истинно}.

3. Что описывает предикат в Prolog?

Предикат в Prolog – название отношений, существующих между объектами.

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)?

Предложения бывают двух видов – факты и правила.

1. Правило состоит **тела** и **головы (заголовка)**. Синтаксически правило оформляется следующим образом:

заголовок :- тело.

Причем, заголовок и тело – это термы; символ ":" – это специальный символ-разделитель.

2. Факт – частный случай правила. Факт – предложение, в котором отсутствует тело.

Пример факта из программы: **automobile**("Surname1", "Ford", "Black", 1600000).

Пример правила из программы: **search_by**(Surname, PhoneNum, CarBrand, CarPrice) :-

phonebook (Surname, PhoneNum, _),

automobile (Surname, CarBrand, _, CarPrice).

Предложения, не содержащие переменные, называются составными. Предложения в момент фиксации в программе, не содержащие переменные, называются неосновными

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?

Переменные предназначены для передачи значений «во времени и в пространстве». Они служат частью процесса сопоставления, и не являются «хранилищем» информации.

Переменные бывают:

- Именованные – обозначается комбинацией символов латинского алфавита, цифр и символа подчеркивания, начинающейся с прописной буквы или символа подчеркивания. Уникальны в рамках одного предложения
- Анонимные – обозначаются символом подчеркивания «_». Любая анонимная переменная уникальна.

Во время вычисления, именованные переменные могут конкретизироваться (связываться с различными объектами), причем, она может быть ре-конкретизирована, путем «отката» вычислительного процесса и отмены ранее проведенной конкретизации для нахождения новых решений.

Анонимные переменные не могут быть связаны со значениями.

Предложение БЗ, содержащее переменные сформулировано в более общей, абстрактной форме.

6. Что такое подстановка?

Подстановкой называется множество пар, вида: $\{ X_i = t_i \}$, где X_i – переменная, а t_i – терм. Т.е происходит конкретизация переменной термом. Применение подстановки заключается в замене каждого вхождения переменной x_i на соответствующий терм.

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

Примером терма В называется такой терм А, если существует такая подстановка в терм А, которая в результате будет эквивалентна терму В. Я думаю, что примеры термов строятся при доказательстве заданной пользователем цели или при вычислении внутренних целей, хранятся примеры до получения ответа, либо до момента «отката».

Текст программы

domains

adress = adress(symbol City, symbol Street, integer HouseNum, integer FlatNum)

predicates

phonebook (symbol Surname, symbol PhoneNum, adress Addr)

automobile(symbol Surname, symbol Brand, symbol Colour, integer Price)

investors (symbol Surname, symbol Bank, symbol Acc_number, integer Value)

search_by(symbol Surname, symbol PhoneNum, symbol CarBrand, integer CarPrice)

search_by(symbol PhoneNum, symbol CarBrand)

search_by(symbol Surname, symbol City, symbol PhoneNum, symbol Street, symbol Bank)

clauses

phonebook("Surname1", "0-000-111-222", adress("Moscow", "Unnatov", 14, 128)).

phonebook("Surname1", "0-000-111-222", adress("Moscow", "Mishina", 1, 10)).

phonebook("Surname1", "5-666-777-888", adress("St.Peterburg", "Unnatov", 14, 128)).

```

phonebook("Surname2", "1-222-333-444", adress("Moscow", "8th March", 12, 153)).
phonebook("Surname3", "3-444-555-666", adress("Moscow", "Mishina", 1, 10)).
phonebook("Surname3", "3-444-555-666", adress("Moscow", "Volgogradskaya", 1, 10)).
phonebook("Surname5", "9-999-000-111", adress("St.Peterburg", "Samarcand", 12, 153)).
phonebook("Surname6", "0-111-222-333", adress("St.Peterburg", "Usievicha", 1, 10)).

```

```

automobile("Surname1", "Ford", "Black", 1600000).
automobile("Surname1", "Volvo", "Silver", 1300000).
automobile("Surname2", "Nissan", "Green", 970000).
automobile("Surname4", "Nissan", "Yellow", 900000).
automobile("Surname5", "Volkswagen", "Pink", 900000).
automobile("Surname6", "Volvo", "Red", 900000).

```

```

investors("Surname1", "Sberbank", "0000 4444 3333 2222", 700000).
investors("Surname1", "VTB", "0000 4444 3333 2222", 700000).
investors("Surname2", "Sberbank", "0000 2222 3333 2222", 200000).
investors("Surname3", "VTB", "0000 3333 3333 2222", 300000).
investors("Surname4", "Sberbank", "0000 5555 3333 2222", 400000).

```

```

search_by(Surname, PhoneNum, CarBrand, CarPrice) :-
    phonebook (Surname, PhoneNum, _),
    automobile (Surname, CarBrand, _, CarPrice).

```

```

search_by(PhoneNum, CarBrand) :-
    search_by(_, PhoneNum, CarBrand, _).

```

```

search_by(Surname, City, PhoneNum, Street, Bank) :-
    phonebook (Surname, PhoneNum, adress(City, Street, _, _)),
    investors (Surname, Bank, _, _).

```

Примеры целей и результатов работы программы

1. **Goal** search_by(Surname, "0-000-111-222", CarBrand, CarPrice).

Result Surname=Surname1, CarBrand=Ford, CarPrice=1600000
Surname=Surname1, CarBrand= Volvo, CarPrice=1300000
Surname=Surname1, CarBrand=Ford, CarPrice=1600000
Surname=Surname1, CarBrand= Volvo, CarPrice=1300000

2. **Goal** search_by("0-000-111-222", CarBrand).

Result CarBrand=Ford
CarBrand=Volvo
CarBrand=Ford
CarBrand=Volvo

3. **Goal** search_by("Surname1", "Moscow", PhoneNum, Street, Bank).

Result PhoneNum=0-000-111-222, Street=Unnatov, Bank=Sberbank
PhoneNum=0-000-111-222, Street=Unnatov, Bank=VTB
PhoneNum=0-000-111-222, Street=Mishina, Bank=Sberbank
PhoneNum=0-000-111-222, Street=Mishina, Bank=VTB

Описание порядка поиска объектов

Цель: search_by(Surname, "0-000-111-222", CarBrand, CarPrice).

1. Производится поиск процедуры search_by с арностью, соответствующей арности цели.

Таблица 1.

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	<p>search_by(Surname, "0-000-111-222", CarBrand, CarPrice) и search_by(Surname, PhoneNum, CarBrand, CarPrice) → производим унификацию.</p> <ol style="list-style-type: none"> 1. Surname унифицируется с Surname – не конкретизированные переменные. 2. PhoneNum унифицируется с "0-000-111-222" – не конкретизированная переменная и константа, PhoneNum конкретизируется значением "0-000-111-222" 3. CarBrand унифицируется с CarBrand - не конкретизированные переменные. 4. CarPrice унифицируется с CarPrice – не конкретизированные переменные. <p>Результат: search_by(Surname, "0-000-111-222", CarBrand, CarPrice)</p>	<p>Унификация выполнена, прямой ход. Выполнение первой подцели в теле search_by → phonebook (Surname, PhoneNum, _)</p>
2	<p>Унификация phonebook (Surname, "0-000-111-222", _) и phonebook("Surname1", "0-000-111-222", address(...)). Произведем унификацию:</p> <ol style="list-style-type: none"> 1. Surname унифицируется с "Surname1" – Surname конкретизируется константой "Surname1". 2. "0-000-111-222" унифицируется с "0-000-111-222" – две одинаковые константы. 3. «_» - анонимная переменная, значение игнорируется. <p>Результат: phonebook("Surname1", "0-000-111-222", _)</p>	<p>Унификация выполнена, прямой ход. Выполнение следующей подцели в теле search_by → automobile (Surname, CarBrand, _, CarPrice).</p>
3	<p>Унификация automobile("Surname1", CarBrand, _, CarPrice) и automobile("Surname1", "Ford", "Black", 1600000). Производим унификацию:</p> <ol style="list-style-type: none"> 1. "Surname1" унифицируется с "Surname1" две одинаковые константы. 2. CarBrand унифицируется с "Ford" – свободная переменная конкретизируется константой "Ford". 3. «_» - значение игнорируется. 4. CarPrice унифицируется с 1600000-свободная переменная конкретизируется константой 1600000. <p>Результат: automobile("Surname1", "Ford", _, 1600000)</p>	<p>Унификация выполнена, Surname="Surname1", CarBrand="Ford", CarPrice=1600000. Откат к automobile ("Surname1", CarBrand,_, CarPrice)</p>

4	<p>Унификация automobile("Surname1", CarBrand, _, CarPrice) и automobile("Surname1", "Volvo", "Silver", 1300000). Производим унификацию:</p> <ol style="list-style-type: none"> 1. "Surname1" унифицируется с "Surname1" две одинаковые константы. 2. CarBrand унифицируется с "Volvo" – свободная переменная конкретизируется константой "Volvo". 3. «_» - значение игнорируется. 4. CarPrice унифицируется с 1300000-свободная переменная конкретизируется константой 1300000. <p>Результат: automobile("Surname1", "Volvo", _, 1300000)</p>	<p>Унификация выполнена, Surname="Surname1", CarBrand="Volvo", CarPrice=1300000. Откат к automobile ("Surname1", CarBrand,_, CarPrice)</p>
5	<p>Соответствий не найдено. Унификация не выполнена.</p>	<p>Откат к phonebook (Surname, "0-000-111-222", _)</p>
6	<p>Унификация phonebook (Surname, "0-000-111-222", _) и phonebook("Surname1", "0-000-111-222", address(...)). Произведем унификацию:</p> <ol style="list-style-type: none"> 4. Surname унифицируется с "Surname1" – Surname конкретизируется константой "Surname1". 5. "0-000-111-222" унифицируется с "0-000-111-222" – две одинаковые константы. 6. «_» - анонимная переменная, значение игнорируется. <p>Результат: phonebook("Surname1", "0-000-111-222", _)</p>	<p>Унификация выполнена, прямой ход. Выполнение следующей подцели в теле search_by -> automobile (Surname, CarBrand, _, CarPrice).</p>
7	<p>Унификация automobile("Surname1", CarBrand, _, CarPrice) и automobile("Surname1", "Ford", "Black", 1600000). Производим унификацию:</p> <ol style="list-style-type: none"> 1. "Surname1" унифицируется с "Surname1" две одинаковые константы. 2. CarBrand унифицируется с "Ford" – свободная переменная конкретизируется константой "Ford". 3. «_» - значение игнорируется. 4. CarPrice унифицируется с 1600000-свободная переменная конкретизируется константой 1600000. <p>Результат: automobile("Surname1", "Ford", _, 1600000)</p>	<p>Унификация выполнена, Surname="Surname1", CarBrand="Ford", CarPrice=1600000. Откат к automobile ("Surname1", CarBrand,_, CarPrice)</p>
8	<p>Унификация automobile("Surname1", CarBrand, _, CarPrice) и automobile("Surname1", "Volvo", "Silver", 1300000). Производим унификацию:</p> <ol style="list-style-type: none"> 5. "Surname1" унифицируется с "Surname1" две одинаковые константы. 6. CarBrand унифицируется с "Volvo" – свободная переменная конкретизируется константой "Volvo". 	<p>Унификация выполнена, Surname="Surname1", CarBrand="Volvo", CarPrice=1300000. Откат к automobile ("Surname1", CarBrand,_, CarPrice)</p>

	<p>7. «_» - значение игнорируется.</p> <p>8. CarPrice унифицируется с 1300000-свободная переменная конкретизируется константой 1300000.</p> <p>Результат: automobile("Surname1", "Volvo", _, 1300000)</p>	
9	Соответствий не найдено. Унификация не выполнена.	<p>Откат к phonebook (Surname, "0-000-111-222", _)</p>
10	Соответствий не найдено. Унификация не выполнена.	<p>4 Solutions.</p>

Цель: search_by ("0-000-111-222", CarBrand).

1. Производится поиск процедуры search_by с арностью, соответствующей арности цели.

Таблица 2.

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	<p>search_by ("0-000-111-222", CarBrand) и search_by(PhoneNum, CarBrand) → производим унификацию.</p> <p>1. PhoneNum унифицируется с "0-000-111-222" – не конкретизированная переменная и константа, PhoneNum конкретизируется значением "0-000-111-222"</p> <p>2. CarBrand унифицируется с CarBrand - не конкретизированные переменные.</p> <p>Результат: search_by("0-000-111-222", CarBrand)</p>	<p>Унификация выполнена, прямой ход. Выполнение первой подцели в теле search_by → search_by (Surname, "0-000-111-222", CarBrand, CarPrice)</p>
2	Описание дальнейшей работы описано в Таблице 1.	<p>Откат к search_by (Surname, "0-000-111-222", CarBrand, CarPrice)</p>
3	Соответствий не найдено. Унификация не выполнена.	<p>4 Solutions.</p>

Цель: search_by("Surname1", "Moscow", PhoneNum, Street, Bank).

1. Производится поиск процедуры search_by с целью, соответствующей цели.

Таблица 3

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	<p>search_by ("Surname1", "Moscow", PhoneNum, Street, Bank). и search_by (Surname, City, PhoneNum, Street, Bank).→ производим унификацию.</p> <ol style="list-style-type: none"> 1. "Surname1" унифицируется с Surname – не конкретизированные переменная конкретизируется константой. 2. City унифицируется с "Moscow" – не конкретизированная переменная и константа, 3. PhoneNum конкретизируется с PhoneNum - не конкретизированные переменные. 4. Street унифицируется с Street - не конкретизированные переменные. 5. Bank унифицируется с Bank – не конкретизированные переменные. <p>Результат: search_by ("Surname1", "Moscow", PhoneNum, Street, Bank).</p>	<p>Унификация выполнена, прямой ход. Выполнение первой подцели в теле search_by → phonebook ("Surname1", PhoneNum, adress("Moscow", Street, _, _)).</p>
2	<p>Унификация phonebook ("Surname1", PhoneNum, adress("Moscow", Street, _, _)) и phonebook("Surname1", "0-000-111-222", adress("Moscow", "Unnatov", 14, 128)). Производим унификацию:</p> <ol style="list-style-type: none"> 1. "Surname1" унифицируется с "Surname1" – две константы. 2. PhoneNum унифицируется с "0-000-111-222" - не конкретизированная переменная и константа, PhoneNum конкретизируется значением "0-000-111-222". 3. "Moscow" унифицируется с "Moscow" – две константы. 4. Street унифицируется с "Unnatov" - не конкретизированная переменная и константа, Street конкретизируется константой "Unnatov". 5. «_» - анонимная переменная, значение игнорируется. 6. «_» - анонимная переменная, значения игнорируется. <p>Результат: phonebook ("Surname1", "0-000-111-222", adress("Moscow", "Unnatov", _, _))</p>	<p>Унификация успешно выполнена, прямой ход к подцели investors ("Surname1", Bank, _, _).</p>
3	<p>Унификация investors ("Surname1", Bank, _, _) и investors ("Surname1", "Sberbank", "0000 4444 3333 2222", 700000). Произведем унификацию:</p>	<p>Унификация выполнена, Surname="Surname1", PhoneNum = "0-000-111-222" Bank="Sberbank". Откат к investors ("Surname1", Bank, _, _)</p>

	<p>1. "Surname1" унифицируется с "Surname1" – две константы.</p> <p>2. Bank унифицируется с "Sberbank" – свободная переменная и константа, Bank конкретизируется константой "Sberbank".</p> <p>3. «_» - анонимная переменная, значение игнорируется.</p> <p>4. «_» - анонимная переменная, значения игнорируется.</p> <p>Результат: investors ("Surname1", "Sberbank", _, _).</p>	
4	<p>Унификация investors ("Surname1", Bank, _, _) и investors ("Surname1", "VTB", "0000 4444 3333 2222", 700000). Произведем унификацию:</p> <p>1. "Surname1" унифицируется с "Surname1" – две константы.</p> <p>2. Bank унифицируется с "VTB" – свободная переменная и константа, Bank конкретизируется константой "VTB".</p> <p>3. «_» - анонимная переменная, значение игнорируется.</p> <p>4. «_» - анонимная переменная, значения игнорируется.</p> <p>Результат: investors ("Surname1", "VTB", _, _).</p>	<p>Унификация выполнена, Surname="Surname1", PhoneNum = "0-000-111-222" Bank="VTB". Откат к investors ("Surname1", Bank, _, _)</p>
5	Соответствий не найдено. Унификация не выполнена.	<p>Откат к phonebook ("Surname1", PhoneNum, adress("Moscow", Street, _, _)).</p>
6	<p>Унификация phonebook ("Surname1", PhoneNum, adress("Moscow", Street, _, _)) и phonebook("Surname1", "0-000-111-222", adress("Moscow", "Mishina", 1, 10)). Производим унификацию:</p> <p>1. "Surname1" унифицируется с "Surname1" – две константы.</p> <p>2. PhoneNum унифицируется с "0-000-111-222" - не конкретизированная переменная и константа, PhoneNum конкретизируется значением "0-000-111-222".</p> <p>3. "Moscow" унифицируется с "Moscow" – две константы.</p> <p>4. Street унифицируется с "Mishina" - не конкретизированная переменная и константа, Street конкретизируется константой "Mishina".</p> <p>5. «_» - анонимная переменная, значение игнорируется.</p> <p>6. «_» - анонимная переменная, значения игнорируется.</p> <p>Результат: phonebook ("Surname1", "0-000-111-222", adress("Moscow", "Mishina", _, _))</p>	<p>Унификация успешно выполнена, прямой ход к подцели investors ("Surname1", Bank, _, _).</p>
7	<p>Унификация investors ("Surname1", Bank, _, _) и investors ("Surname1", "Sberbank", "0000 4444 3333 2222", 700000). Произведем унификацию:</p>	<p>Унификация выполнена, Surname="Surname1", PhoneNum = "0-000-111-222" Bank="Sberbank". Откат к investors ("Surname1", Bank, _, _)</p>

	1. "Surname1" унифицируется с "Surname1" – две константы. 2. Bank унифицируется с "Sberbank" – свободная переменная и константа, Bank конкретизируется константой "Sberbank". 3. «_» - анонимная переменная, значение игнорируется. 4. «_» - анонимная переменная, значения игнорируется. Результат: investors ("Surname1", "Sberbank", _, _).	
8	Унификация investors ("Surname1", Bank, _, _) и investors ("Surname1", "VTB", "0000 4444 3333 2222", 700000). Произведем унификацию: 1. "Surname1" унифицируется с "Surname1" – две константы. 2. Bank унифицируется с "VTB" – свободная переменная и константа, Bank конкретизируется константой "VTB". 3. «_» - анонимная переменная, значение игнорируется. 4. «_» - анонимная переменная, значения игнорируется. Результат: investors ("Surname1", "VTB", _, _).	Унификация выполнена, Surname="Surname1", PhoneNum = "0-000-111-222" Bank="VTB". Откат к investors ("Surname1", Bank, _, _)
9	Соответствий не найдено. Унификация не выполнена.	Откат к phonebook ("Surname1", PhoneNum, adress("Moscow", Street, _, _)).
10	Соответствий не найдено. Унификация не выполнена.	4 Solutions.