



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 № 14**

Дисциплина Функциональное и логическое программирование

Тема Работа программы на Prolog

Студент Ильясов И. М.

Группа ИУ7-63Б

Оценка (баллы) \_\_\_\_\_

Преподаватель Толпинская Н. Б., Строганов Ю. В.

Москва, 2020 г.

**Цель работы** – получить навыки построения модели предметной области, разработки и оформления программы на Prolog, изучить принципы, логику формирования программы и отдельные шаги выполнения программы на Prolog.

**Задачи работы:**

- приобрести навыки декларативного описания предметной области с использованием фактов и правил;
- изучить способы использования термов, переменных, фактов и правил в программе на Prolog, принципы и правила сопоставления и отождествления, порядок унификации.

**Задание лабораторной работы**

Используя базу знаний, хранящую знания (лаб. 13):

- **«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). В разных городах есть однофамильцы, в одном городе – фамилия уникальна.

Используя **конъюнктивное правило и простой вопрос**, обеспечить возможность поиска:

По Марке и Цвету автомобиля найти Фамилию, Город, Телефон и Банки, в которых владелец автомобиля имеет вклады. Лишней информации не находить и не передавать!!!

Владельцев может быть **несколько** (не более 3-х), **один** и **ни одного**.

1. Для каждого из трех вариантов **словесно подробно** описать порядок формирования ответа (в виде таблицы). При этом, указать – отметить моменты очередного запуска алгоритма унификации и полный результат его работы. Обосновать следующий шаг работы системы. Выписать унификаторы – подстановки. Указать моменты, причины и результат отката, если он есть.
2. Для случая нескольких владельцев (2-х):  
приведите примеры (таблицы) работы системы **при разных порядках** следования в БЗ процедур, и знаний в них: (**«Телефонный справочник»**, **«Автомобили»**, **«Вкладчики банков»**, или: **«Автомобили»**, **«Вкладчики банков»**, **«Телефонный справочник»**).  
Сделайте **вывод**: Одинаковы ли: множество работ и объем работ в разных случаях?
3. Оформите 2 таблицы, демонстрирующие **порядок работы алгоритма унификации** вопроса и подходящего заголовка правила (для двух случаев из пункта 2) и укажите результаты его работы: ответ и побочный эффект.

## Текст программы

### domains

```
surname, phone, city, street = string.  
house, flat = integer.  
address = address(city, street, house, flat).
```

```
car_mark, car_color = string.  
car_cost = integer.
```

```
bank_name, bank_cardnumber = string.  
bank_sum = integer.
```

### predicates

```
abonement(surname, phone, address).  
car(surname, car_mark, car_color, car_cost, city).  
deposit(surname, bank_name, bank_cardnumber, bank_sum, city).
```

```
find(phone, car_mark, car_cost).  
find(phone, car_mark).
```

### clauses

```
abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).  
abonement("Ilyasov", "89969503880", address("Moscow", "Severnaya", 12, 75)).  
abonement("Sidenko", "89691929395", address("Moscow", "Semenovskaya", 25, 53)).  
abonement("Stepanov", "185818582839", address("Korolev", "Krilianskaya", 12, 155)).  
abonement("Gorbunov", "165615253616", address("Saint-Petersburg", "Nevskaya", 51, 122)).
```

```
car("Gorbunov", "Audi", "White", 2500000, "Korolev").  
car("Ilyasov", "Mitsubishi", "Red", 800000, "Moscow").  
car("Ilyasov", "Tesla", "Black", 6000000, "Moscow").  
car("Sidenko", "Tesla", "Black", 6000000, "Moscow").  
car("Sidenko", "Mercedes", "White", 3000000, "Moscow").  
car("Stepanov", "Mercedes", "Black", 5000000, "Korolev").  
car("Stepanov", "Audi", "White", 2500000, "Korolev").  
car("Stepanov", "Mercedes", "White", 3000000, "Korolev").  
car("Gorbunov", "Toyota", "White", 600000, "Saint-Petersburg").
```

```
deposit("Gorbunov", "Sberbank", "123456789", 30000, "Korolev").  
deposit("Ilyasov", "Sberbank", "987654321", 20000, "Moscow").  
deposit("Sidenko", "Tinkoff", "135798642", 60000, "Moscow").  
deposit("Stepanov", "Alfa", "156273727", 20000, "Korolev").  
deposit("Gorbunov", "Sberbank", "563281726", 100000, "Saint-Petersburg").
```

```
find(Phone, Car_Mark, Car_Cost) :- abonement(Surname, Phone, address(City, _, _)),  
                                car(Surname, Car_Mark, _, Car_Cost, City).
```

```
find(Phone, Car_Mark) :- find(Phone, Car_Mark, _).
```

```
find_result(Car_Mark, Car_Color, Surname, City, Phone, Bank_Name) :- car(Surname, Car_Mark,  
Car_Color, _, City), abonement(Surname, Phone, address(City, _, _)), deposit(Surname, Bank_Name, _, _, City).
```

### goal

```
Car_Color = "White",  
Car_Mark = "Toyota",  
find_result(Car_Mark, Car_Color, Surname, City, Phone, Bank_Name).
```

## Примеры работы программы

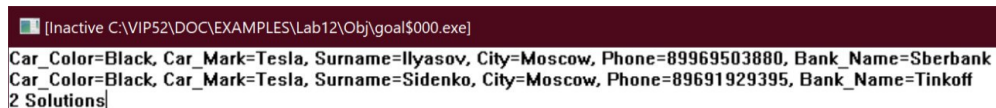
На рисунке 1 приведен ответ на вопрос об информации о владельцах белой Тойоты (1 владелец).



```
[Inactive C:\VIP52\DOC\EXAMPLES\Lab12\Obj\goal$000.exe]
Car_Color=White, Car_Mark=Toyota, Surname=Gorbunov, City=Saint-Petersburg, Phone=165615253616, Bank_Name=Sberbank
1 Solution
```

Рисунок 1. Ответ на вопрос о владельцах белой Тойоты (1 владелец).

На рисунке 2 приведен ответ на вопрос об информации о владельцах черной Теслы (2 владельца).



```
[Inactive C:\VIP52\DOC\EXAMPLES\Lab12\Obj\goal$000.exe]
Car_Color=Black, Car_Mark=Tesla, Surname=Ilyasov, City=Moscow, Phone=89969503880, Bank_Name=Sberbank
Car_Color=Black, Car_Mark=Tesla, Surname=Sidenko, City=Moscow, Phone=89691929395, Bank_Name=Tinkoff
2 Solutions
```

Рисунок 2. Ответ на вопрос о владельцах черной Теслы (2 владельца).

На рисунке 3 приведен ответ на простой вопрос о владельцах белой Теслы (ни одного владельца).



```
[Inactive C:\VIP52\DOC\EXAMPLES\Lab12\Obj\goal$000.exe]
No Solution
```

Рисунок 3. Ответ на вопрос о владельцах белой Теслы (ни одного владельца).

## Ответ на задание 1

- 1) Car\_Color = "White",  
Car\_Mark = "Toyota",  
find\_result(Car\_Mark, Car\_Color, Surname, City, Phone, Bank\_Name) – ненужные промежуточные шаги опущены.

| № шага | Сравниваемые термины; результат; подстановка, если есть   | Дальнейшие действия: прямой ход или откат (к чему приводит?)   |
|--------|---|--|
| 1      | Производится сравнение find_result("Toyota", "White", Surname, City, Phone, Bank_Name) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).<br>Результат – неудача.                 | Откат к find_result("Toyota", "White", Surname, City, Phone, Bank_Name)  |
|        | ...   | ...  |
| 2      | Производится сравнение find_result("Toyota", "White", Surname, City, Phone, Bank_Name) и find_result(Car_Mark, Car_Color, Surname, City, Phone, Bank_Name). Car_Mark конкретизируется с "Toyota", Car_Color – с | Прямой ход, производится переход к унификации car(Surname, Car_Mark, Car_Color, _, City), abonement(Surname, Phone, address(City, _, _)), deposit(Surname, Bank_Name, _, _). |

|   |  |  |
|---|--|--|
|   | <p>“White”.</p> <p>Результат – find_result(“Toyota”, “White”, Surname, City, Phone, Bank_Name).</p>  | <p>City). Car_Mark = “Toyota”, Car_Color = “White”</p>   |
| 3 | <p>Производится сравнение car(Surname, “Toyota”, “White”, _, City) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).</p> <p>Результат – неудача.</p>  | <p>Откат к car(Surname, “Toyota”, “White”, _, City), проверка следующей строки</p>   |
|   | <p>...</p> <p>(остальные факты abonement также не унифицируются)</p>   | <p>...</p>   |
| 4 | <p>Производится сравнение car(Surname, “Toyota”, “White”, _, City) и car("Gorbunov", "Audi", “White”, _, “Korolev”). Surname конкретизируется с “Gorbunov”, City – с “Korolev”.</p> <p>Результат – неудача.</p>  | <p>Откат к car(Surname, “Toyota”, “White”, _, City), проверка следующей строки</p>   |
|   | <p>...</p> <p>(все остальные факты car до следующего не унифицируются)</p>   | <p>...</p>   |
| 5 | <p>Производится сравнение car(Surname, “Toyota”, “White”, _, City) и car("Gorbunov", "Toyota", “White”, _, “Saint-Petersburg”). Surname конкретизируется с “Gorbunov”, City – с “Saint-Petersburg”.</p> <p>Результат – car(“Gorbunov”, “Toyota”, “White”, “Saint-Petersburg”).</p> | <p>Прямой ход, производится переход к унификации abonement(Surname, Phone, address(City, _, _, _)). При этом переменная Surname равна “Gorbunov”, City равна “Saint-Petersburg”, Car_Mark – “Toyota”, Car_Color – “White”.</p> |
|   | <p>...</p>   | <p>...</p>   |
| 6 | <p>Производится сравнение abonement(“Gorbunov”, Phone, address(“Saint-Petersburg”, _, _, _)). и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с “89251472838”.</p> <p>Результат – неудача.</p>                             | <p>Откат к abonement(“Gorbunov”, Phone, address(“Saint-Petersburg”, _, _, _)), проверка следующей строки</p>   |
|   | <p>...</p> <p>(все остальные факты abonement до следующего не унифицируются)</p>   | <p>...</p>   |
| 7 | <p>Производится сравнение abonement(“Gorbunov”, Phone, address(“Saint-Petersburg”, _, _, _)). и abonement("Gorbunov",</p>  | <p>Прямой ход, производится переход к унификации deposit(Surname, Bank_Name, _, _, City). При этом</p>   |

|    |  |  |
|----|--|--|
|    | "165615253616", address("Saint-Petersburg", "Glavnaya", 55, 122)). Phone конкретизируется с "165615253616".<br>Результат – abonent("Gorbunov", "165615253616", address("Saint-Petersburg", _, _))  | переменная Surname равна "Gorbunov", City равна "Saint-Petersburg", Phone – "165615253616", Car_Mark – "Toyota", Car_Color – "White".  |
|    | ...  | ...  |
| 8  | Производится сравнение deposit("Gorbunov", Bank_Name, _, _, "Saint-Petersburg"). и abonent("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).<br>Результат – неудача.  | Откат к сравнение deposit("Gorbunov", Bank_Name, _, _, "Saint-Petersburg"), проверка следующей строки  |
|    | ...<br>(все остальные факты abonent не унифицируются)  | ...  |
| 9  | Производится сравнение deposit("Gorbunov", Bank_Name, _, _, "Saint-Petersburg"). и deposit("Gorbunov", "Sberbank", "123456789", 30000, "Korolev"). Bank_Name конкретизируется с "Sberbank".<br>Результат – неудача.  | Откат к сравнение deposit("Gorbunov", Bank_Name, _, _, "Saint-Petersburg"), проверка следующей строки  |
|    | ...<br>(все остальные факты deposit до следующего не унифицируются)  | ...  |
| 10 | Производится сравнение deposit("Gorbunov", Bank_Name, _, _, "Saint-Petersburg"). и deposit("Gorbunov", "Sberbank", "563281726", 100000, "Saint-Petersburg"). Bank_Name конкретизируется с "Sberbank".<br>Результат – deposit("Gorbunov", "Sberbank", _, _, "Saint-Petersburg") | Прямой ход, производится подстановка значений, которые были найдены в исходный вопрос. При этом Surname = "Gorbunov", Bank_Name = "Sberbank", Phone = "165615253616", City = "Saint-Petersburg", Car_Mark = "Toyota", Car_Color = "White". |
|    | ...  | ...  |
| 11 | car("Gorbunov", "Toyota", "White", _, "Saint-Petersburg"), abonent("Gorbunov", "165615253616", address("Saint-Petersburg", _, _)), deposit("Gorbunov", "Sberbank", _, _, "Saint-Petersburg")   |  |
|    | ...  | ...  |
|    | (Результат – 1 решение)  |  |

- 2) Car\_Color = "Black",  
 Car\_Mark = "Tesla",  
 car(Surname, Car\_Mark, Car\_Color, \_, City),  
 abonement(Surname, Phone, address(City, \_, \_)),  
 deposit(Surname, Bank\_Name, \_, \_, City) – ненужные промежуточные шаги опущены.

| № шага | Сравниваемые термы; результат; подстановка, если есть   | Дальнейшие действия: прямой ход или откат (к чему приводит?)   |
|--------|---|--|
| 1      | Производится сравнение find_result("Tesla", "Black", Surname, City, Phone, Bank_Name) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).<br>Результат – неудача.  | Откат к find_result("Tesla", "Black", Surname, City, Phone, Bank_Name)   |
|        | ...   | ...  |
| 2      | Производится сравнение find_result("Tesla", "Black", Surname, City, Phone, Bank_Name) и find_result(Car_Mark, Car_Color, Surname, City, Phone, Bank_Name). Car_Mark конкретизируется с "Tesla", Car_Color – с "Black".<br>Результат – find_result("Tesla", "Black", Surname, City, Phone, Bank_Name). | Прямой ход, производится переход к унификации car(Surname, Car_Mark, Car_Color, _, City), abonement(Surname, Phone, address(City, _, _)), deposit(Surname, Bank_Name, _, _, City). Car_Mark = "Tesla", Car_Color = "Black" |
| 3      | Производится сравнение car(Surname, "Tesla", "Black", _, City) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Surname конкретизируется с "Gorbunov", City – с "Korolev".<br>Результат – неудача.  | Откат к car(Surname, "Tesla", "Black", _, City), проверка следующей строки   |
|        | ...<br>(все остальные факты abonement не унифицируются)   | ...  |
| 4      | Производится сравнение car(Surname, "Tesla", "Black", _, City) и car("Gorbunov", "Audi", "White", _, "Korolev"). Surname конкретизируется с "Gorbunov", City – с "Korolev".<br>Результат – неудача.   | Откат к car(Surname, "Tesla", "Black", _, City), проверка следующей строки   |
|        | ...<br>(все остальные факты car до следующего не унифицируются)   | ...  |
| 5      | Производится сравнение car(Surname, "Tesla", "Black", _, City) и car("Ilyasov",   | Прямой ход, производится переход к унификации abonement(Surname,   |

|    |  |  |
|----|--|--|
|    | "Tesla", "Black", _, "Moscow"). Surname конкретизируется с "Ilyasov", City – с "Moscow".<br>Результат – car("Ilyasov", "Tesla", "Black", "Moscow").  | Phone, address(City, _, _). При этом переменная Surname равна "Ilyasov", City равна "Moscow", Car_Mark – "Tesla", Car_Color – "Black".   |
|    | ...<br>(все остальные факты deposit не унифицируются)  | ...  |
| 6  | Производится сравнение abonent("Ilyasov", Phone, address("Moscow", _, _)) и abonent("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).<br>Результат – неудача.   | Откат к abonent("Ilyasov", Phone, address("Moscow", _, _)), проверка следующей строки  |
| 7  | Производится сравнение abonent("Ilyasov", Phone, address("Moscow", _, _)) и abonent("Ilyasov", "89969503880", address("Moscow", "Severnaya", 12, 75)).<br>Phone конкретизируется с "89969503880".<br>Результат – abonent("Ilyasov", 89969503880, address("Moscow", _, _)). | Прямой ход, производится переход к унификации deposit(Surname, Bank_Name, _, City). При этом переменная Surname равна "Ilyasov", City равна "Moscow", Phone – "89969503880", Car_Mark – "Tesla", Car_Color – "Black".    |
|    | ...<br>(все остальные факты abonent не унифицируются)  | ...  |
| 8  | Производится сравнение deposit("Ilyasov", Bank_Name, _, "Moscow"). и abonent("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).<br>Bank_Name конкретизируется с "Sberbank".<br>Результат – неудача.  | Откат к deposit("Ilyasov", Bank_Name, _, "Moscow"), проверка следующей строки  |
|    | ...<br>(все остальные факты abonent не унифицируются)  | ...  |
| 9  | Производится сравнение deposit("Ilyasov", Bank_Name, _, "Moscow"). и deposit("Ilyasov", "Sberbank", "987654321", 20000, "Moscow"). Bank_Name конкретизируется с "Sberbank".<br>Результат – deposit("Ilyasov", "Sberbank", _, "Moscow").                                    | Откат, производится подстановка значений, которые были найдены в исходный вопрос. При этом Surname = "Ilyasov", Bank_Name = "Sberbank", Phone = "89969503880", City = "Moscow", Car_Mark = "Tesla", Car_Color = "Black". |
| 10 | car("Ilyasov", "Tesla", "Black", _, "Moscow"), abonent("Ilyasov", "987654321",   |  |



|    |   |   |
|----|---|---|
|    | address("Moscow", _, _, _), deposit("Ilyasov", "Sberbank", _, _, "Moscow")  |   |
| 11 | Производится сравнение car(Surname, "Tesla", "Black", _, City) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Surname конкретизируется с "Gorbunov", City – с "Korolev".<br>Результат – неудача.  | Откат к car(Surname, "Tesla", "Black", _, City), проверка следующей строки  |
|    | ...<br>(все остальные факты abonement не унифицируются)   | ...   |
| 12 | Производится сравнение car(Surname, "Tesla", "Black", _, City) и car("Gorbunov", "Audi", "White", _, "Korolev"). Surname конкретизируется с "Gorbunov", City – с "Korolev".<br>Результат – неудача.                                 | Откат к car(Surname, "Tesla", "Black", _, City), проверка следующей строки  |
|    | ...<br>(все остальные факты car до следующего не унифицируются)   | ...   |
| 13 | Производится сравнение car(Surname, "Tesla", "Black", _, City) и car("Sidenko", "Tesla", "Black", _, "Moscow"). Surname конкретизируется с "Sidenko", City – с "Moscow".<br>Результат – car("Sidenko", "Tesla", "Black", "Moscow"). | Прямой ход, производится переход к унификации abonement(Surname, Phone, address(City, _, _, _)). При этом переменная Surname равна "Sidenko", City равна "Moscow", Car_Mark – "Tesla", Car_Color – "Black". |
|    | ...<br>(все остальные факты car не унифицируются)   | ...   |
| 14 | Производится сравнение abonement("Sidenko", Phone, address("Moscow", _, _, _)). и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).<br>Результат – неудача.  | Откат к abonement("Sidenko", Phone, address("Moscow", _, _, _)), проверка следующей строки  |
|    | ...<br>(все остальные факты abonement до следующего не унифицируются)   | ...   |
| 15 | Производится сравнение abonement("Sidenko", Phone,  | Прямой ход, производится переход к унификации deposit(Surname,  |

|    |   |   |
|----|---|---|
|    | address("Moscow", _, _, _). и<br>abonement("Sidenko", "89691929395",<br>address("Moscow", "Semenovskaya", 25, 53)).<br>Phone конкретизируется с "89691929395".<br>Результат – abonement("Sidenko",<br>"89691929395", address("Moscow", _, _, _)).         | Bank_Name, _, _, City). При этом<br>переменная Surname равна "Sidenko",<br>City равна "Moscow", Phone –<br>"89691929395", Car_Mark – "Tesla",<br>Car_Color – "Black".   |
|    | ...<br>(все остальные факты abonement не<br>унифицируются)  | ...   |
| 16 | Производится сравнение deposit("Sidenko",<br>Bank_Name, _, _, "Moscow"). и<br>abonement("Gorbunov", "89251472838",<br>address("Korolev", "Glavnaya", 55, 122)).<br>Bank_Name конкретизируется с "Sberbank".<br>Результат – неудача.                       | Откат к deposit("Sidenko", Bank_Name,<br>_, _, "Moscow"), проверка следующей<br>строки  |
|    | ...<br>(все остальные факты deposit до следующего<br>не унифицируются)  | ...   |
| 17 | Производится сравнение deposit("Sidenko",<br>Bank_Name, _, _, "Moscow"). и<br>deposit("Sidenko", "Tinkoff", "135798642",<br>60000, "Moscow"). Bank_Name<br>конкретизируется с "Tinkoff".<br>Результат – deposit("Sidenko", "Tinkoff", _, _,<br>"Moscow"). | Откат, производится подстановка<br>значений, которые были найдены в<br>исходный вопрос. При этом Surname =<br>"Sidenko", Bank_Name = "Tinkoff",<br>Phone = "89251472838", City =<br>"Moscow", Car_Mark = "Tesla",<br>Car_Color = "Black". |
| 18 | car("Sidenko", "Tesla", "Black", _, "Moscow"),<br>abonement("Ilyasov", "135798642",<br>address("Moscow", _, _, _)), deposit("Sidenko",<br>"Tinkoff", _, _, "Moscow")  |   |
|    | ...   | ...   |
|    | (Результат – 2 решения)   |   |

- 3) Car\_Color = "White",  
Car\_Mark = "Tesla",  
car(Surname, Car\_Mark, Car\_Color, \_, City),  
abonement(Surname, Phone, address(City, \_, \_, \_)),  
deposit(Surname, Bank\_Name, \_, \_, City) – ненужные промежуточные шаги опущены.

| № шага | Сравниваемые термы; результат; подстановка, если есть                                      | Дальнейшие действия: прямой ход или откат (к чему приводит?)              |
|--------|--|---|
| 1      | Производится сравнение find_result("Tesla",<br>"White", Surname, City, Phone, Bank_Name) и | Откат к find_result("Tesla", "White",<br>Surname, City, Phone, Bank_Name) |

|   |   |   |
|---|---|---|
|   | abonement("Gorbunov", "89251472838",<br>address("Korolev", "Glavnaya", 55, 122)).<br>Результат – неудача.   |   |
|   | ...   | ...   |
| 2 | Производится сравнение find_result(“Tesla”,<br>“White”, Surname, City, Phone, Bank_Name) и<br>find_result(Car_Mark, Car_Color, Surname,<br>City, Phone, Bank_Name). Car_Mark<br>конкретизируется с “Tesla”, Car_Color – с<br>“White”.<br>Результат – find_result(“Tesla”, “White”,<br>Surname, City, Phone, Bank_Name). | Прямой ход, производится переход к<br>унификации car(Surname, Car_Mark,<br>Car_Color, _, City), abonement(Surname,<br>Phone, address(City, _, _, _)),<br>deposit(Surname, Bank_Name, _, _,<br>City). Car_Mark = “Tesla”, Car_Color =<br>“White” |
| 3 | Производится сравнение car(Surname,<br>“Tesla”, “White”, _, City) и<br>abonement("Gorbunov", "89251472838",<br>address("Korolev", "Glavnaya", 55, 122)).<br>Surname конкретизируется с “Gorbunov”, City<br>– с “Korolev”.<br>Результат – неудача.   | Откат к car(Surname, “Tesla”, “White”,<br>_, City), проверка следующей строки   |
|   | ...<br>(все остальные факты abonement не<br>унифицируются)  | ...   |
| 4 | Производится сравнение car(Surname,<br>“Tesla”, “White”, _, City) и car("Gorbunov",<br>"Audi", “White”, _, “Korolev”). Surname<br>конкретизируется с “Gorbunov”, City – с<br>“Korolev”.<br>Результат – неудача.   | Откат к car(Surname, “Tesla”, “White”,<br>_, City), проверка следующей строки   |
|   | ...<br>(все остальные факты car не унифицируются)   | ...   |
| 5 | Производится сравнение car(Surname,<br>“Tesla”, “Black”, _, City) и deposit("Gorbunov",<br>"Sberbank", "123456789", 30000, "Korolev").<br>Surname конкретизируется с “Ilyasov”, City –<br>с “Moscow”.<br>Результат – неудача  | Откат к car(Surname, “Tesla”, “White”,<br>_, City), проверка следующей строки   |
|   | ...   | ...   |
|   | (Результат – 0 решений)   |   |

## Ответ на задание 2

Так как у нас система будет проходить базу знаний до конца (начиная с первого факта и заканчивая последним), то количество и объем работы не изменится при разных следованиях фактов о телефонном справочнике, о машинах и о вкладах, потому что количество сравнений будет тем же и для каждого вопроса будет производиться попытка унификации элементов из БЗ. Как мне кажется, если система будет знать, где у нас конец процедуры, то можно будет уменьшить количество сравнений, подняв процедуру повыше.

## Ответ на задание 3

find\_result("Toyota", "White", Surname, City, Phone, Bank\_Name)

| Шаг<br>ун-<br>ции | Результирующая<br>ячейка                          | Рабочее поле  | Пункт<br>ал-ма | Стек  |
|-------------------|---|---|----------------|---|
| 1                 |   |   | 1              | find_result("Toyota",<br>"White", Surname, City,<br>Phone, Bank_Name)<br>=<br>find_result(Car_Mark,<br>Car_Color, Surname, City,<br>Phone, Bank_Name)   |
| 2                 |   | find_result("Toyota", "White",<br>Surname, City, Phone, Bank_Name)<br>=<br>find_result(Car_Mark, Car_Color,<br>Surname, City, Phone, Bank_Name)<br>:- car(Surname, Car_Mark,<br>Car_Color, _, City),<br>abonement(Surname, Phone,<br>address(City, _, _)),<br>deposit(Surname, Bank_Name, _, _,<br>City). |                | Car_Mark = "Toyota"<br>Car_Color = "White"<br>car(Surname, Car_Mark,<br>Car_Color, _, City)<br>abonement(Surname,<br>Phone, address(City, _, _<br>_))<br>deposit(Surname,<br>Bank_Name, _, _, City) |
| 3                 | Car_Mark =<br>"Toyota"                            | ← Car_Mark = "Toyota"   | r              | Car_Color = "White"<br>car(Surname, Car_Mark,<br>Car_Color, _, City)<br>abonement(Surname,<br>Phone, address(City, _, _<br>_))<br>deposit(Surname,<br>Bank_Name, _, _, City)                        |
| 4                 | Car_Mark =<br>"Toyota"<br><br>Car_Color = "White" | ← Car_Color = "White"   | r              | car(Surname, Car_Mark,<br>Car_Color, _, City)<br>abonement(Surname,<br>Phone, address(City, _, _<br>_))<br>deposit(Surname,<br>Bank_Name, _, _, City)   |
|                   | ...   | ...   | ...            | ...   |

|    |  |  |     |   |
|----|--|--|-----|---|
|    |  | (перебор фактов abonement)   |     |   |
| 5  | Car_Mark = "Toyota"<br>Car_Color = "White"                         | car(Surname, "Toyota", "White", _, City)<br><br>car("Gorbunov", "Audi", "White", 2500000, "Korolev").<br><br>Неуспешная унификация!                                |     | Surname = "Gorbunov"<br>"Toyota" = "Toyota"<br>"White" = "White"<br>abonement(Surname, Phone, address(City, _, _))<br>deposit(Surname, Bank_Name, _, _, City) |
|    | ...  | ...  | ... | ...   |
|    |  | (перебор оставшихся фактов car до нужного)   |     |   |
| 6  | Car_Mark = "Toyota"<br>Car_Color = "White"                         | car(Surname, "Toyota", "White", _, City)<br><br>car("Gorbunov", "Toyota", "White", 600000, "Saint-Petersburg").<br><br>Успешная унификация →                       | е   | Surname = "Gorbunov"<br>"Toyota" = "Toyota"<br>"White" = "White"<br>abonement(Surname, Phone, address(City, _, _))<br>deposit(Surname, Bank_Name, _, _, City) |
| 7  | Car_Mark = "Toyota"<br>Car_Color = "White"<br>Surname = "Gorbunov" | ← Surname = "Gorbunov"   | г   | "Toyota" = "Toyota"<br>"White" = "White"<br>abonement(Surname, Phone, address(City, _, _))<br>deposit(Surname, Bank_Name, _, _, City)                         |
| 8  | Car_Mark = "Toyota"<br>Car_Color = "Red"<br>Surname = "Gorbunov"   | "Toyota" = "Toyota"  | б   | "White" = "White"<br>abonement(Surname, Phone, address(City, _, _))<br>deposit(Surname, Bank_Name, _, _, City)  |
| 9  | Car_Mark = "Toyota"<br>Car_Color = "White"<br>Surname = "Gorbunov" | "White" = "White"  | б   | abonement(Surname, Phone, address(City, _, _))<br>deposit(Surname, Bank_Name, _, _, City)   |
| 10 | Car_Mark = "Toyota"<br>Car_Color = "White"<br>Surname = "Gorbunov" | abonement(Surname, Phone, address(City, _, _))<br><br>abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).<br><br>Неуспешная унификация! |     | abonement(Surname, Phone, address(City, _, _))<br>deposit(Surname, Bank_Name, _, _, City)   |
|    | ...  | ...  | ... | ...   |
|    |  | (перебор оставшихся фактов abonement до нужного)   |     |   |

|    |  |   |     |  |
|----|--|---|-----|--|
| 11 | Car_Mark =<br>"Toyota"<br><br>Car_Color = "White"<br><br>Surname =<br>"Gorbunov"   | abonement(Surname, Phone,<br>address(City, _, _))<br><br>abonement("Gorbunov",<br>"165615253616", address("Saint-<br>Petersburg", "Nevskaya", 51, 122)).<br><br>Успешная унификация → |     | "Gorbunov" = "Gorbunov"<br><br>Phone = "165615253616"<br><br>City = "Saint-Petersburg"<br><br>deposit(Surname,<br>Bank_Name, _, _, City) |
| 12 | Car_Mark =<br>"Toyota"<br><br>Car_Color = "White"<br><br>Surname =<br>"Gorbunov"   | "Gorbunov" = "Gorbunov"   | б   | Phone = "165615253616"<br><br>City = "Saint-Petersburg"<br><br>deposit(Surname,<br>Bank_Name, _, _, City)                                |
| 13 | Car_Mark =<br>"Toyota"<br><br>Car_Color = "White"<br><br>Surname =<br>"Gorbunov"<br><br>Phone =<br>"165615253616"                                      | ← Phone = "165615253616"  | г   | City = "Saint-Petersburg"<br><br>deposit(Surname,<br>Bank_Name, _, _, City)  |
| 14 | Car_Mark =<br>"Toyota"<br><br>Car_Color = "White"<br><br>Surname =<br>"Gorbunov"<br><br>Phone =<br>"165615253616"<br><br>City = "Saint-<br>Petersburg" | ← City = "Saint-Petersburg"   | г   | deposit(Surname,<br>Bank_Name, _, _, City)   |
|    | ...  | ...<br><br>(перебор фактов abonement, car до<br>deposit)  | ... | ...  |
| 15 | Car_Mark =<br>"Toyota"<br><br>Car_Color = "White"<br><br>Surname =<br>"Gorbunov"<br><br>Phone =<br>"165615253616"<br><br>City = "Saint-<br>Petersburg" | deposit(Surname, Bank_Name, _, _,<br>City)<br><br>deposit("Gorbunov", "Sberbank",<br>"123456789", 30000, "Korolev")<br><br>Неуспешная унификация!                                     |     | deposit(Surname,<br>Bank_Name, _, _, City)   |
|    | ...  | ...<br><br>(перебор оставшихся фактов<br>deposit до нужного)  | ... | ...  |
| 16 | Car_Mark =<br>"Toyota"<br><br>Car_Color = "White"<br><br>Surname =<br>"Gorbunov"   | deposit(Surname, Bank_Name, _, _,<br>City)<br><br>deposit("Gorbunov", "Sberbank",<br>"563281726", 100000, "Saint-<br>Petersburg")<br><br>Успешная унификация →                        |     | "Gorbunov" = "Gorbunov"<br><br>Bank_Name = "Sberbank"  |

|                      |   |   |   |  |
|----------------------|---|---|---|--|
|                      | Phone =<br>"165615253616"<br><br>City = "Saint-Petersburg"  |   |   |  |
| 17                   | Car_Mark =<br>"Toyota"<br><br>Car_Color = "White"<br><br>Surname =<br>"Gorbunov"<br><br>Phone =<br>"165615253616"<br><br>City = "Saint-Petersburg"                                  | "Gorbunov" = "Gorbunov"   | б | Bank_Name = "Sberbank"   |
| 18                   | Car_Mark =<br>"Toyota"<br><br>Car_Color = "White"<br><br>Surname =<br>"Gorbunov"<br><br>Phone =<br>"165615253616"<br><br>City = "Saint-Petersburg"<br><br>Bank_Name =<br>"Sberbank" | ← Bank_Name = "Sberbank"  | г |  |
| <b>Вывод подцели</b> | Подстановка   | Так как стек пуст – успех<br><br>← Подстановка  |   |  |
|                      |   | Откат к шагу 16, перебор фактов deposit с оставшимися фактами и правилами (find, find_result)<br><br>Неуспешная унификация!                 |   | deposit(Surname, Bank_Name, _, _, City)  |
|                      |   | Откат к шагу 11, перебор фактов abonement с оставшимися фактами и правилами (car, deposit, find, find_result)<br><br>Неуспешная унификация! |   | abonement(Surname, Phone, address(City, _, _, _))<br>deposit(Surname, Bank_Name, _, _, City)   |
|                      |   | Откат к шагу 6, перебор фактов car с оставшимися фактами и правилами (deposit, find, find_result)<br><br>Неуспешная унификация!             |   | car(Surname, Car_Mark, Car_Color, _, City)<br>abonement(Surname, Phone, address(City, _, _, _))<br>deposit(Surname, Bank_Name, _, _, City) |

При изменении порядка следования знаний в БЗ, как и было сказано, объем работы и количество сравнений останется таким же (будут проверяться все факты и правила из БЗ), просто прямой ход в случае успеха будет производиться раньше или позже в зависимости от порядка следования знаний. Допустим, теперь факты о car расположены над abonement.

| Шаг<br>уни-<br>ции | Результирующая<br>ячейка                      | Рабочее поле  | Пункт<br>ал-ма | Стек  |
|--------------------|---|---|----------------|---|
| 1                  |   |   | 1              | find_result("Toyota",<br>"White", Surname, City,<br>Phone, Bank_Name)<br>=<br>find_result(Car_Mark,<br>Car_Color, Surname, City,<br>Phone, Bank_Name)   |
| 2                  |   | find_result("Toyota", "White",<br>Surname, City, Phone, Bank_Name)<br>=<br>find_result(Car_Mark, Car_Color,<br>Surname, City, Phone, Bank_Name)<br>:- car(Surname, Car_Mark,<br>Car_Color, _, City),<br>abonement(Surname, Phone,<br>address(City, _, _)),<br>deposit(Surname, Bank_Name, _, _,<br>City). |                | Car_Mark = "Toyota"<br>Car_Color = "White"<br>car(Surname, Car_Mark,<br>Car_Color, _, City)<br>abonement(Surname,<br>Phone, address(City, _, _<br>_))<br>deposit(Surname,<br>Bank_Name, _, _, City) |
| 3                  | Car_Mark =<br>"Toyota"                        | ← Car_Mark = "Toyota"   | r              | Car_Color = "White"<br>car(Surname, Car_Mark,<br>Car_Color, _, City)<br>abonement(Surname,<br>Phone, address(City, _, _<br>_))<br>deposit(Surname,<br>Bank_Name, _, _, City)                        |
| 4                  | Car_Mark =<br>"Toyota"<br>Car_Color = "White" | ← Car_Color = "White"   | r              | car(Surname, Car_Mark,<br>Car_Color, _, City)<br>abonement(Surname,<br>Phone, address(City, _, _<br>_))<br>deposit(Surname,<br>Bank_Name, _, _, City)   |
| 5                  | Car_Mark =<br>"Toyota"<br>Car_Color = "White" | car(Surname, "Toyota", "White", _,<br>City)<br>car("Gorbunov", "Audi", "White",<br>2500000, "Korolev").<br>Неуспешная унификация!   |                | Surname = "Gorbunov"<br>"Toyota" = "Toyota"<br>"White" = "White"<br>abonement(Surname,<br>Phone, address(City, _, _<br>_))<br>deposit(Surname,<br>Bank_Name, _, _, City)                            |
|                    | ...   | ...<br>(перебор оставшихся фактов car<br>до нужного)  | ...            | ...   |
| 6                  | Car_Mark =<br>"Toyota"<br>Car_Color = "White" | car(Surname, "Toyota", "White", _,<br>City)<br>car("Gorbunov", "Toyota", "White",<br>600000, "Saint-Petersburg").   | e              | Surname = "Gorbunov"<br>"Toyota" = "Toyota"<br>"White" = "White"  |



|    |  |   |     |   |
|----|--|---|-----|---|
|    |  | Успешная унификация →   |     | abonement(Surname, Phone, address(City, _, _))<br>deposit(Surname, Bank_Name, _, _, City)   |
| 7  | Car_Mark = "Toyota"<br>Car_Color = "White"<br>Surname = "Gorbunov"                           | ← Surname = "Gorbunov"  | г   | "Toyota" = "Toyota"<br>"White" = "White"<br>abonement(Surname, Phone, address(City, _, _))<br>deposit(Surname, Bank_Name, _, _, City) |
| 8  | Car_Mark = "Toyota"<br>Car_Color = "Red"<br>Surname = "Gorbunov"                             | "Toyota" = "Toyota"   | б   | "White" = "White"<br>abonement(Surname, Phone, address(City, _, _))<br>deposit(Surname, Bank_Name, _, _, City)                        |
| 9  | Car_Mark = "Toyota"<br>Car_Color = "White"<br>Surname = "Gorbunov"                           | "White" = "White"   | б   | abonement(Surname, Phone, address(City, _, _))<br>deposit(Surname, Bank_Name, _, _, City)   |
|    | ...  | ...<br>(перебор фактов car)   | ... | ...   |
|    | ...  | ...<br>(перебор оставшихся фактов abonement до нужного)   | ... | ...   |
| 11 | Car_Mark = "Toyota"<br>Car_Color = "White"<br>Surname = "Gorbunov"                           | abonement(Surname, Phone, address(City, _, _))<br>abonement("Gorbunov", "165615253616", address("Saint-Petersburg", "Nevskaya", 51, 122)).<br>Успешная унификация → |     | "Gorbunov" = "Gorbunov"<br>Phone = "165615253616"<br>City = "Saint-Petersburg"<br>deposit(Surname, Bank_Name, _, _, City)             |
| 12 | Car_Mark = "Toyota"<br>Car_Color = "White"<br>Surname = "Gorbunov"                           | "Gorbunov" = "Gorbunov"   | б   | Phone = "165615253616"<br>City = "Saint-Petersburg"<br>deposit(Surname, Bank_Name, _, _, City)  |
| 13 | Car_Mark = "Toyota"<br>Car_Color = "White"<br>Surname = "Gorbunov"<br>Phone = "165615253616" | ← Phone = "165615253616"  | г   | City = "Saint-Petersburg"<br>deposit(Surname, Bank_Name, _, _, City)  |
| 14 | Car_Mark = "Toyota"  | ← City = "Saint-Petersburg"   | г   | deposit(Surname, Bank_Name, _, _, City)   |

|    |  |  |     |   |
|----|--|--|-----|---|
|    | Car_Color = "White"<br>Surname =<br>"Gorbunov"<br>Phone =<br>"165615253616"<br>City = "Saint-Petersburg"                           |  |     |   |
|    | ...  | ...<br>(перебор фактов car, abonent до deposit)  | ... | ...   |
| 15 | Car_Mark =<br>"Toyota"<br>Car_Color = "White"<br>Surname =<br>"Gorbunov"<br>Phone =<br>"165615253616"<br>City = "Saint-Petersburg" | deposit(Surname, Bank_Name, _, _, City)<br>deposit("Gorbunov", "Sberbank", "123456789", 30000, "Korolev")<br>Неуспешная унификация!          |     | deposit(Surname, Bank_Name, _, _, City)           |
|    | ...  | ...<br>(перебор оставшихся фактов deposit до нужного)  | ... | ...   |
| 16 | Car_Mark =<br>"Toyota"<br>Car_Color = "White"<br>Surname =<br>"Gorbunov"<br>Phone =<br>"165615253616"<br>City = "Saint-Petersburg" | deposit(Surname, Bank_Name, _, _, City)<br>deposit("Gorbunov", "Sberbank", "563281726", 100000, "Saint-Petersburg")<br>Успешная унификация → |     | "Gorbunov" = "Gorbunov"<br>Bank_Name = "Sberbank" |
| 17 | Car_Mark =<br>"Toyota"<br>Car_Color = "White"<br>Surname =<br>"Gorbunov"<br>Phone =<br>"165615253616"<br>City = "Saint-Petersburg" | "Gorbunov" = "Gorbunov"  | б   | Bank_Name = "Sberbank"                            |
| 18 | Car_Mark =<br>"Toyota"<br>Car_Color = "White"<br>Surname =<br>"Gorbunov"<br>Phone =<br>"165615253616"                              | ← Bank_Name = "Sberbank"   | г   |   |

|                      |   |  |  |  |
|----------------------|---|--|--|--|
|                      | City = "Saint-Petersburg"<br>Bank_Name = "Sberbank" |  |  |  |
| <b>Вывод подцели</b> | Подстановка   | Так как стек пуст – успех<br>← Подстановка   |  |  |
|                      |   | Откат к шагу 16, перебор фактов deposit с оставшимися фактами и правилами (find, find_result)<br>Неуспешная унификация!                |  | deposit(Surname, Bank_Name, _, _, City)  |
|                      |   | Откат к шагу 11, перебор фактов abonement с оставшимися фактами и правилами (deposit, find, find_result)<br>Неуспешная унификация!     |  | abonement(Surname, Phone, address(City, _, _, _))<br>deposit(Surname, Bank_Name, _, _, City)   |
|                      |   | Откат к шагу 6, перебор фактов car с оставшимися фактами и правилами (abonement, deposit, find, find_result)<br>Неуспешная унификация! |  | car(Surname, Car_Mark, Car_Color, _, City)<br>abonement(Surname, Phone, address(City, _, _, _))<br>deposit(Surname, Bank_Name, _, _, City) |

## Ответы на вопросы

- 1. В какой части правила сформулировано знание? Это знание о чем, с формальной точки зрения?**

Правила – предложение вида  $A : - B_1, \dots, B_n$ , то, что находится слева от знака  $:$  – является заголовком правила, то, что справа – телом правила. Факт (знание) – частный случай правила, у него нет тела. То есть знание сформулировано в заголовке правила.

- 2. Что такое процедура?**

Процедура – совокупность правил, заголовки которых одинаковы.

- 3. Сколько в БЗ текущего задания процедур?**

В БЗ текущего задания 4 процедуры – телефонная книга, машины, банк и правило.

- 4. Что такое пример терма, это частный случай терма, пример? Как строится пример?**

Пример терма – это результат подстановки некоторых конкретных значений в предикат, частный случай предиката. Строится после того, как задан вопрос. Хранится до окончания работы программы.

- 5. Что такое наиболее общий пример?**

S называется наиболее общим примером T1 и T2, если S такой их общий пример, который является более общим по отношению к любому другому их примеру.

**6. Назначение и результат работы алгоритма унификации. Что значит двунаправленная передача параметров при работе алгоритма унификации, поясните на примере одного из случаев пункта 3.**

Работа алгоритма унификации заключается в попарном сопоставлении термов и попытке построить для них общий пример. Алгоритм унификации производит двунаправленную передачу параметров процедурам. Двунаправленная передача параметров при работе алгоритма унификации – передача этих самых параметров извне в программу для дальнейшего использования или из программы во внешний мир (например, значение параметра, который нас интересует). Пример: пункты 5 и 7 в задании 3 – мы передаем неконкретизированную переменную Surname в процедуру, потом эта переменная конкретизируется и мы получаем интересующую нас информацию – фамилию владельца.

**7. В каком случае запускается механизм отката?**

Механизм отката запускается в случае, когда унификация (попытка «увидеть одинаковость») завершается тупиковой ситуацией (или неудачей). При этом происходит откат к предыдущему шагу.

**8. Виды и назначение переменных в Prolog. Примеры из задания. Почему использованы те или другие переменные (примеры из задания)?**

Переменные – один из видов термов в языке Prolog. Переменные бывают:

- именованные – обозначаются комбинацией символов латинского алфавита, цифр и символа подчеркивания, начинающейся с прописной буквы или символа подчеркивания;
- анонимные – символ подчеркивания.

**Примеры переменных в нашем задании:**

- Surname, City, Phone, Bank\_Name и др. – именованные. Нам эти данные интересны, поэтому мы их делаем именованными
- Символы подчеркивания – анонимные. Эти данные нам не нужны, мы их не выводим, поэтому их можно сделать анонимными.