



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 Куприй А. А.

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:**

Составить 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). В разных городах есть однофамильцы, в одном городе – фамилия уникальна. Владельцев может быть несколько (не более 3-х), один и ни одного.

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!!!

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 таблицы, демонстрирующие порядок работы алгоритма унификации вопроса и подходящего заголовка правила и укажите результаты его работы: ответ и побочный эффект.

## **Вопросы:**

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

Знание сформулировано в заголовке, как в составном терме  $f(t_1, \dots, t_n)$ . Оно содержит знание о том, что между аргументами  $t_1, \dots, t_n$  существует отношение. «f» - название этого отношения.

## **Что такое процедура?**

Процедурой называется совокупность правил, заголовки которых имеют одно и то же имя и одну и ту же аргументность (местность), т. е. это совокупность правил, описывающих одно определенное отношение.

## **Сколько в БЗ текущего задания процедур?**

В моей программе 4 процедуры – phonebook, depositor, automobile и search\_by.

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

Примером терма В называется такой терм А, если существует такая подстановка в терм А, которая в результате будет эквивалентна терму В. Например, есть терм  $\text{cook}(X, Y)$  и есть  $\text{cook}(\text{“Яблоки”}, \text{“Груши”})$ , тогда при подстановке  $\text{cook}(X, Y)$  и  $\text{cook}(\text{“Яблоки”}, \text{“Груши”})$  получится  $\text{cook}(\text{“Яблоки”}, \text{“Груши”})$ , являющийся примером терма  $\text{cook}(\text{“Яблоки”}, \text{“Груши”})$ . Пример терма строится при сопоставлении (конкретизации) более общего терма с возможными подстановками.

## **Что такое наиболее общий пример?**

Терм С называется общим примером термов А и В, если существуют такие подстановки  $\theta_1$  и  $\theta_2$ , что  $C = A\theta_1$  и  $C = B\theta_2$ , т. е. С является общим примером для В и А, причем, А и В являются примерами С, а С для них не является.

Назначение и результат работы алгоритма унификации. Что значит двунаправленная передача параметров при работе алгоритма унификации, поясните на примере одного из случаев пункта 3. Алгоритм унификации - основной вычислительный шаг работы программы. Предназначен для «сопоставления термов», формализует процесс логического вывода. Завершается либо «успехом», либо «неудачей». Результат работы программы – наиболее общий унификатор.

Двунаправленная передача параметров – после унификации, переменные могут быть конкретизированы, в этом случае, конкретизированные переменные переходят на следующий шаг доказательства (в подцели, например).

### **В каком случае запускается механизм отката?**

Механизм отката запускается в случае, если во алгоритм унификации завершился тупиковой ситуацией, неудачей.

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

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.

Пример: `search_by (name1, moscow, Number, Street, Bank).` - используются именованные свободные переменные, для получения набора значений при вычислении цели.

Пример: `search_by (Number, Brand) :- search_by(_, Number, Brand, _).` – анонимные переменные, используются для игнорирования некоторых значений программой.

## Текст программы

### domains

name = symbol.

number = integer.

city, street = symbol.

house, apartment = integer.

brand, color = symbol.

price = integer.

bank = symbol.

account, amount = integer.

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

### predicates

nondeterm phonebook(name, number, address).

nondeterm automobile(name, brand, color, price).

nondeterm sdepositor(name, bank, account, amount).

nondeterm search\_by(Brand, Color, Name, City, Number, Bank).

### clauses

phonebook(name1, 22211100, address(moscow, lenina, 14, 128)).

phonebook(name1, 88877766, address(sp, alexandrova, 14, 128)).

phonebook(name2, 44433322, address(moscow, ahmatovoy, 12, 153)).

phonebook(name3, 66655544, address(moscow, baumanskaya, 1, 10)).

phonebook(name3, 55544422, address(moscow, moscowskaya, 1, 10)).

phonebook(name4, 11111111, address(sp, alexandrova, 1, 10)).

automobile(name1, ford, black, 1600000).

automobile(name2, ford, black, 1600000).

automobile(name3, ford, black, 1600000).

automobile(name4, volvo, silver, 1300000).

automobile(name3, nissan, red, 1300000).

depositor(name1, sberbank, 444433332, 700000).

depositor(name2, sberbank, 222233332, 200000).

depositor(name4, tochka, 333333332, 300000).

depositor(name4, vtb, 444433332, 700000).

search\_by(Brand, Color, Name, City, Number, Bank) :-

automobile(Name, Brand, Color, \_),

phonebook(Name, Number, address(City, \_, \_, \_)),

depositor(Name, Bank, \_, \_).

goal

search\_by(ford, black, Name, City, Number, Bank).

% search\_by(volvo, silver, Name, City, Number, Bank).

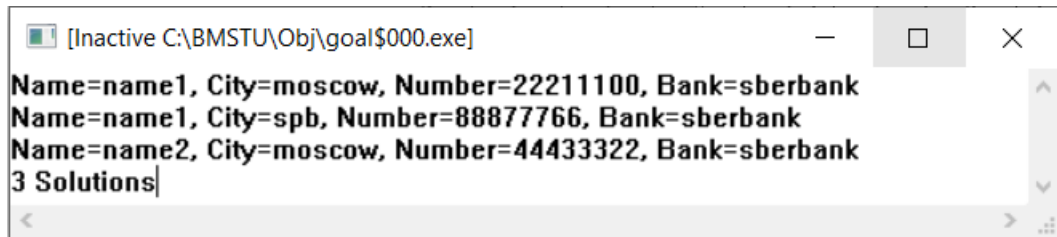
% search\_by(nissan, red, Name, City, Number, Bank).



## Результаты работы программы:

**goal**

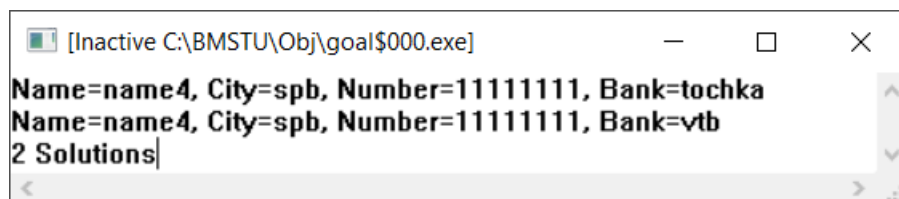
**search\_by(ford, black, Name, City, Number, Bank).**



```
[Inactive C:\BMSTU\Obj\goal$000.exe]  
Name=name1, City=moscow, Number=22211100, Bank=sberbank  
Name=name1, City=spb, Number=88877766, Bank=sberbank  
Name=name2, City=moscow, Number=44433322, Bank=sberbank  
3 Solutions|
```

**goal**

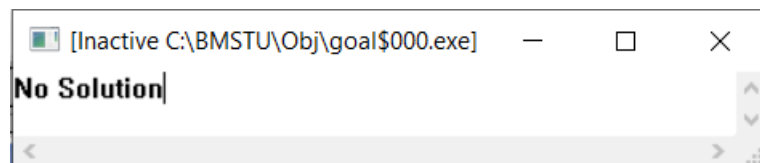
**search\_by(volvo, silver, Name, City, Number, Bank).**



```
[Inactive C:\BMSTU\Obj\goal$000.exe]  
Name=name4, City=spb, Number=11111111, Bank=tochka  
Name=name4, City=spb, Number=11111111, Bank=vtb  
2 Solutions|
```

**goal**

**search\_by(nissan, red, Name, City, Number, Bank).**



```
[Inactive C:\BMSTU\Obj\goal$000.exe]  
No Solution|
```

## Описание порядка поиска объектов

**Цель:** search\_by(ford, black, Name, City, Number, Bank).

Пример работы алгоритма унификации поэтапно, для нахождения первого набора значений, при условии, что необходимо условие для доказательства цели найдено.

Таблица 1

№ шага	Сравниваемые термы; подстановка, если есть	Дальнейшие действия: прямой ход или откат
1	<p>Сопоставляем: search_by(ford, black, Name, City, Number, Bank). search_by(Brand, Color, Name, City, Number, Bank).</p> <ol style="list-style-type: none"><li>1) Кладем в стек ford= Brand, black=Color, Name=Name, City= City, Number =Number, Bank= Bank.</li><li>2) Достаем из стека: ford= Brand – константа и именованная свободная переменная, (добавить в результирующую ячейку Brand=ford), переход к следующему шагу.</li><li>3) Достаем из стека: black=Color - константа и именованная свободная переменная, (добавить в результирующую ячейку Color =black) переход к следующему шагу.</li><li>4) Достаем из стека Name=Name – две свободные именованные переменные (связанные переменные).</li><li>5) Достаем из стека City= City – две свободные именованные переменные (связанные переменные).</li><li>6) Достаем из стека Number =Number – две свободные именованные переменные (связанные переменные).</li><li>7) Достаем из стека Bank= Bank – две свободные именованные переменные (связанные переменные).</li><li>8) Очистка рабочего поля. Неудача=0</li></ol> <p>Результат: Brand=ford, Color=black</p>	<p>Унификация выполнена успешно, прямой ход. Переходим к Automobile (Name, ford, black, _).</p>

2	<p>Сопоставляем:  automobile (Name, ford, black, _)  automobile(name1, ford, black, 1600000).</p> <ol style="list-style-type: none"> <li>1) Кладем в стек Name= name1, ford=ford, black=black.</li> <li>2) Достаем из стека Name= name1 - константа и именованная свободная переменная, (добавить в результирующую ячейку Name= name1), переход к следующему шагу.</li> <li>3) Достаем из стека ford=ford – равные константы, следующий шаг</li> <li>4) Достаем из стека black=black – равные константы, следующий шаг</li> <li>5) Стек пуст, очистка рабочего поля. Неудача=0</li> </ol> <p>Результат: Name= name1</p>	<p>Унификация выполнена, прямой ход. Выполнение следующей подцели в теле search_by → automobile(Name, Brand, _, Price).</p>
3	<p>Сопоставляем:  phonebook(name1, Number, address(City, _, _))  phonebook(name1, 22211100, address(moscow, lenina, 14, 128)).</p> <ol style="list-style-type: none"> <li>1) Кладем в стек name1=name1, Number=22211100, address(City, _, _)= address(moscow, lenina, 14, 128)</li> <li>2) Достаем из стека name1=name1 - равные константы, следующий шаг</li> <li>3) Достаем из стека Number=22211100 - константа и именованная свободная переменная, (добавить в результирующую ячейку Number=22211100), переход к следующему шагу.</li> <li>4) Достаем из стека address(City, _, _)= address(moscow, lenina, 14, 128). – составные термы – кладем в стек moscow=City.</li> <li>5) Стек пуст, очистить поле. Неудача=0.</li> </ol> <p>Результат: Number=22211100, City=moscow</p>	<p>Унификация выполнена успешно, прямой ход.  Переходим к depositor(name1, Bank, _, _).</p>
4	<p>Сопоставляем:  depositor(name1, Bank, _, _).  depositor(name1, sberbank, 444433332, 700000).</p> <ol style="list-style-type: none"> <li>1) Кладем в стек name1=name1, Bank=sberbank.</li> <li>2) Достаем из стека name1=name1 - равные константы, следующий шаг</li> </ol>	<p>Унификация выполнена успешно,  Name= name1  Number=22211100, City=moscow,  Brand=ford,  Color =black,  Bank=sberbank.  Обратный ход</p>

	<p>3) Достаем из стека Bank=sberbank - константа и именованная свободная переменная, (добавить в результирующую ячейку Bank=sberbank), переход к следующему шагу</p> <p>4) Стек пустой. Очистка поля. Неудача=0.</p> <p>Результат: Bank=sberbank</p>	
--	--	--

Составные термы унифицируемы, если:

- 1) имеют одинаковые главные функторы,
- 2) имеют одинаковую арность.

**Цель:** search\_by (22211100, Brand).

Выполнение поиска унифицируемых термов.

Таблица 2

№ шага	Сравниваемые термы; подстановка, если есть	Дальнейшие действия: прямой ход или откат
1	<p>Сопоставляем: search_by(ford, black, Name, City, Number, Bank). phonebook(name1, 22211100, address(moscow, lenina, 14, 128)).</p> <p>Кладем в стек search_by(...) = phonebook(...).</p> <p>Вытаскиваем из стека search_by(...) = phonebook(...).</p> <p>Термы составные: главные функторы и арность не совпадают – унификация невозможна.</p>	<p>Прямой ход, переход к следующему предложению. (по аналогии, до конца процедуры phonebook)</p>
2	<p>Сопоставляем: search_by(ford, black, Name, City, Number, Bank). automobile(name1, ford, black, 1600000).</p> <p>Кладем в стек search_by(...) = automobile(...).</p> <p>Вытаскиваем из стека search_by(...) = automobile(...).</p> <p>Термы составные: главные функторы и арность не совпадают – унификация невозможна.</p>	<p>Прямой ход, переход к следующему предложению. (по аналогии, до конца процедуры automobile)</p>

3	<p>Сопоставляем:  search_by(ford, black, Name, City, Number, Bank). depositor(name1, sberbank, 444433332, 700000).</p> <p>Кладем в стек search_by(...) = depositor (...).</p> <p>Вытаскиваем из стека search_by(...) = depositor (...).</p> <p>Термы составные: главные функторы и арность не совпадают – унификация невозможна.</p>	<p>Прямой ход, переход к следующему предложению.  (по аналогии, до конца процедуры depositor)</p>
4	<p>Сопоставляем:  search_by(ford, black, Name, City, Number, Bank). search_by(Brand, Color, Name, City, Number, Bank) кладем в стек: search_by(...) = search_by(...).</p> <p>Вытаскиваем из стека: search_by(...) = search_by(...).</p> <p>Составные термы: имена главных функторов совпали, арность совпала – термы унифицируемы. В стек заносятся: Brand=ford, Color=black, Name= Name, City=City, Number=Number, Bank= Bank.</p> <p>Результат унификации: успех</p> <p>Наиболее общий унификатор: Brand=ford, Color=black.</p>	<p>Прямой ход. Переход в тело правила.</p>

### Задание 1.

3 владельца одного автомобиля.

Таблица 3

№ шага	Сравниваемые термы; подстановка, если есть	Дальнейшие действия: прямой ход или откат
1	<p>Сопоставляем:  automobile(Name, ford, black, _), с термами аналогично терму search_by (см. Таблица 2), пока не будет найден терм удовлетворяющий условиям унификации.</p>	<p>Прямой ход, переход к следующему предложению. (по аналогии, до конца процедуры phonebook)</p>
2	<p>Сопоставляем:  automobile(Name, ford, black, _),  automobile(name1</p>	<p>Прямой ход, переход к следующей подцели правила.</p>

	<p>, ford, black, 1600000).</p> <p>Составные термы, унифицируемые.</p> <p>Результат унификации: успех</p> <p>Наиболее общий унификатор: Name=name1.</p>	
3	<p>Сопоставляем: phonebook(name1, Number, address(City, _, _, _)) phonebook(name1, 22211100, address(moscow, lenina, 14, 128)).</p> <p>Составные термы, унифицируемые.</p> <p>Результат унификации: успех</p> <p>Наиболее общий унификатор: Number=22211100, City=moscow.</p>	<p>Прямой ход, переход к следующей подцели правила.</p>
4	<p>Сопоставляем: depositor(name1, Bank, _, _). с термами аналогично терму search_by (см. Таблица 2), пока не будет найден терм удовлетворяющий условиям унификации.</p>	<p>Прямой ход, переход к следующему предложению. (по аналогии, до конца процедур phonebook, automobile)</p>
5	<p>Сопоставляем: depositor(name1, Bank, _, _). depositor(name1, sberbank, 444433332, 700000).</p> <p>Составные термы, унифицируемые. Результат унификации: успех</p> <p>Наиболее общий унификатор: Bank=sberbank.</p>	<p>Истинное тело найдено.</p> <p>Name=name1, Number=22211100, City=moscow, Bank=sberbank.</p> <p>Откат, переход к следующему предложению.</p>
6	<p>Сопоставляем: depositor(name1, Bank, _, _). depositor(name2, sberbank, 222233332, 200000).</p> <p>Составные термы, унифицируемые.</p> <p>Результат унификации: неудача, константы name1 и name2 не совпадают.</p>	<p>Унификация завершилась неудачей, прямой ход, переход к следующему предложению.</p>
7	<p>Сопоставляем: depositor(name1, Bank, _, _). с термами в процедуре depositor. Все термы</p>	<p>Унификация завершилась неудачей. Откат</p>

	унифицируемы, но унификация завершается неудачей.	
8	Сопоставляем: phonebook(name1, Number, address(City, _, _)) phonebook(name1, 88877766, address(spб, alexandrova, 14, 128)) Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: Number=88877766, City=spб.	Прямой ход, переход к следующей подцели правила
9	Сопоставляем: depositor(name1, Bank, _, _). depositor(name1, sberbank, 444433332, 700000). Составные термы, унифицируемые. Результат унификации: успех. Наиболее общий унификатор: Bank=sberbank.	Истинное тело найдено. Name=name1, Number=88877766, City=spб, Bank=sberbank. Откат, переход к следующему предложению.
10	Сопоставляем: depositor(name1, Bank, _, _). с термами в процедуре depositor. Все термы унифицируемы, но унификация завершается неудачей.	Унификация завершилась неудачей. Откат
11	Сопоставляем: phonebook(name1, Number, address(City, _, _)) с термами в процедуре phonebook. Все термы унифицируемы, но унификация завершается неудачей.	Унификация завершилась неудачей. Откат
12	Сопоставляем: automobile(Name,ford, black, _), automobile(name2, ford, black, 1600000). Составные термы, унифицируемые. Результат унификации: успех. Наиболее общий унификатор: Name=name2.	Прямой ход, переход к следующей подцели правила.
13	Сопоставляем: phonebook(name2, Number, address(City, _, _)) с термами в процедуре phonebook, пока не найдем успешно унифицируемый терм, либо пока не закончится БЗ.	Прямой ход, переход к следующему предложению.
14	Сопоставляем: phonebook(name2, Number, address(City, _, _)) phonebook(name2, 44433322, address(moscow, ahmatovoy, 12, 153)) Составные термы, унифицируемые. Результат унификации: успех	Прямой ход, переход к следующей подцели правила.

	Наиболее общий унификатор: Number=44433322, City=moscow.	
15	Сопоставляем: depositor(name2, Bank, _, _) с термами в процедуре depositor, пока не найдем успешно унифицируемый терм, либо пока не закончится БЗ.	Прямой ход, переход к следующему предложению.
16	Сопоставляем: depositor(name2, Bank, _, _). depositor(name2, sberbank, 222233332, 200000). Составные термы, унифицируемые. Результат унификации: успех. Наиболее общий унификатор: Bank=sberbank.	Истинное тело найдено. Name=name2, Number=44433322, City=moscow, Bank=sberbank. Откат, переход к следующему предложению
17	Сопоставляем: depositor(name1, Bank, _, _). с термами в процедуре depositor. Все термы унифицируемы, но унификация завершается неудачей.	Унификация завершилась неудачей. Откат
18	Сопоставляем: phonebook(name1, Number, address(City, _, _)) с термами в процедуре phonebook. Все термы процедуры унифицируемы, но унификация завершается неудачей.	Унификация завершилась неудачей. Откат
19	Сопоставляем: automobile(Name, ford, black, _), с термами в процедуре phonebook. Все термы процедуры унифицируемы, но унификация завершается неудачей. Далее до конца БЗ термы не унифицируемы.	Унификация завершилась неудачей. Откат
20	Сопоставляем: search_by(ford, black, Name, City, Number, Bank), с термами до конца БЗ, унифицируемых термов не найдено.	Конец

Пример работы программы на цели для 1 владельца автомобиля.

Таблица 4

№ шага	Сравниваемые термы; подстановка, если есть	Дальнейшие действия: прямой ход или откат



1	<p>Сопоставляем:  search_by(volvo, silver,  Name, City, Number, Bank).  search_by(Brand, Color,  Name, City, Number, Bank)  кладем в стек: search_by(...)  = search_by(...).  Вытаскиваем из стека: search_by(...) =  search_by(...).  Составные термы: имена главных  функторов совпали, аргументы совпали –  термы унифицируемы. В стек заносятся:  Brand=volvo, Color=silver, Name= Name,  City=City, Number=Number, Bank= Bank.  Результат унификации: успех  Наиболее общий унификатор:  Brand=volvo, Color=silver.</p>	Прямой ход. Переход в тело правила.
2	<p>Сопоставляем:  automobile(Name, volvo, silver,, _), с  термами аналогично терму search_by  (см. Таблица 2), пока не будет найден  терм удовлетворяющий условиям  унификации.</p>	Прямой ход, переход к следующему предложению. (по аналогии, до конца процедуры phonebook)
3	<p>Сопоставляем:  automobile(Name,  volvo, silver,, _),  automobile(name4,  volvo, silver,  1300000)  Составные термы,  унифицируемые.  Результат  унификации: успех  Наиболее общий унификатор:  Name=name4.</p>	Прямой ход, переход к следующей подцели правила.
4	<p>Сопоставляем:  phonebook(name4, Number, address(City,  _, _, _)) phonebook(name4, 11111111,  address(spbl, alexandrova, 1, 10)).  Составные термы, унифицируемые.  Результат унификации: успех  Наиболее общий унификатор:  Number=11111111, City=spbl.</p>	Прямой ход, переход к следующей подцели правила.
5	<p>Сопоставляем:  depositor(name4, Bank, _, _). с термами  аналогично терму search_by (см.  Таблица 2), пока не будет найден терм  удовлетворяющий условиям унификации.</p>	Прямой ход, переход к следующему предложению. (по аналогии, до конца процедур phonebook, automobile)

6	Сопоставляем: depositor(name4, Bank, _, _). depositor(name4, tochka, 33333332, 300000). Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: Bank=tochka.	Истинное тело найдено. Name=name4, Number=1111111, City=spb, Bank=tochka. Откат, переход к следующему предложению.
7	Сопоставляем: depositor(name4, Bank, _, _). с термами в процедуре depositor. Все термы унифицируемы, но унификация завершается неудачей.	Унификация завершилась неудачей. Откат
8	Сопоставляем: phonebook(name4, Number, address(City, _, _, _)) с термами в процедуре phonebook. Все термы процедуры унифицируемы, но унификация завершается неудачей.	Унификация завершилась неудачей. Откат
9	Сопоставляем: automobile(Name, volvo, silver,, _), с термами в процедуре phonebook. Все термы процедуры унифицируемы, но унификация завершается неудачей. Далее до конца БЗ термы не унифицируемы.	Унификация завершилась неудачей. Откат
10	Сопоставляем: search_by(volvo, silver, Name, City, Number, Bank), с термами до конца БЗ, унифицируемых термов не найдено.	Конец

Пример работы программы на цели для машины с 1 владельцем, без вклада.

Таблица 5

№ шага	Сравниваемые термы; подстановка, если есть	Дальнейшие действия: прямой ход или откат
1	Сопоставляем: search_by(nissan, red, Name, City, Number, Bank) search_by(Brand, Color, Name, City, Number, Bank) кладем в	Прямой ход. Переход в тело правила.

	<p>стек: search_by(...) = search_by(...).</p> <p>Вытаскиваем из стека: search_by(...) = search_by(...).</p> <p>Составные термы: имена главных функторов совпали, арность совпала – термы унифицируемы. В стек заносятся: Brand=nissan, Color=red, Name= Name, City=City, Number=Number, Bank= Bank.</p> <p>Результат унификации: успех</p> <p>Наиболее общий унификатор: Brand=nissan, Color=red.</p>	
2	<p>Сопоставляем: automobile(Name, nissan, red, _), с термами аналогично терму search_by (см. Таблица 2), пока не будет найден терм удовлетворяющий условиям унификации.</p>	<p>Прямой ход, переход к следующему предложению. (по аналогии, до конца процедуры phonebook)</p>
3	<p>Сопоставляем: automobile(Name, nissan, red, _), automobile(name3, nissan, red, 1300000). Составные термы, унифицируемые. Результат унификации: успех</p> <p>Наиболее общий унификатор: Name=name3.</p>	<p>Прямой ход, переход к следующей подцели правила.</p>
4	<p>Сопоставляем: phonebook(name3, Number, address(City, _, _, _)) phonebook(name3, 66655544, address(moscow, baumanskaya, 1, 10)). Составные термы, унифицируемые. Результат унификации: успех</p> <p>Наиболее общий унификатор: Number= 66655544, City=moscow.</p>	<p>Прямой ход, переход к следующей подцели правила.</p>
5	<p>Сопоставляем: depositor(name3, Bank, _, _). с термами пока не будет найден подходящий терм, либо пока не закончится БЗ. БЗ закончилась.</p> <p>Унифицируемых термов найдено не было.</p>	<p>Откат</p>

6	Сопоставляем: phonebook(name3, Number, address(City, _, _, _)) с термами в процедуре phonebook. Все термы процедуры унифицируемы, но унификация завершается неудачей.	Унификация завершилась неудачей. Откат
7	Сопоставляем: automobile(Name, nissan, red, _), с термами в процедуре phonebook. Все термы процедуры унифицируемы, но унификация завершается неудачей. Далее до конца БЗ термы не унифицируемы.	Унификация завершилась неудачей. Откат
8	Сопоставляем: search_by(nissan, red, Name, City, Number, Bank), с термами до конца БЗ, унифицируемых термов не найдено.	0 Solutions

Задание 2.

«Телефоны» «Автомобили» «Инвесторы»

Таблица 6

№ шага	Сравниваемые термы; подстановка, если есть	Дальнейшие действия: прямой ход или откат
1	Сопоставляем: search_by(volvo, silver, Name, City, Number, Bank). search_by(Brand, Color, Name, City, Number, Bank) кладем в стек: search_by(...) = search_by(...). Вытаскиваем из стека: search_by(...) = search_by(...). Составные термы: имена главных функторов совпали, арность совпала – термы унифицируемы. В стек	Прямой ход. Переход в тело правила.

	<p>заносятся: Brand=volvo, Color=silver, Name= Name, City=City, Number=Number, Bank= Bank.</p> <p>Результат унификации: успех</p> <p>Наиболее общий унификатор: Brand=volvo, Color=silver.</p> <p>Для поиска было проведено 16 сопоставлений.</p>	
2	<p>Сопоставляем: automobile(Name, volvo, silver,, _), с термами аналогично терму search_by (см. Таблица 2), пока не будет найден терм удовлетворяющий условиям унификации.</p>	<p>Прямой ход, переход к следующему предложению. (по аналогии, до конца процедуры phonebook)</p>
3	<p>Сопоставляем: automobile(Name, volvo, silver,, _), automobile(name4, volvo, silver, 1300000) Составные термы, унифицируемые.</p> <p>Результат унификации: успех</p> <p>Наиболее общий унификатор: Name=name4.</p> <p>Для поиска было проведено 10 сопоставлений</p>	<p>Прямой ход, переход к следующей подцели правила.</p>
4	<p>Сопоставляем: phonebook(name4, Number, address(City, _, _, _)) phonebook(name4, 11111111, address(spb, alexandrova, 1, 10)).</p> <p>Составные термы, унифицируемые. Результат унификации: успех</p> <p>Наиболее общий унификатор: Number=11111111, City=spb.</p> <p>Для поиска было проведено 6 сопоставлений.</p>	<p>Прямой ход, переход к следующей подцели правила.</p>
5	<p>Сопоставляем: depositor(name4, Bank, _, _). с термами аналогично терму search_by (см. Таблица 2), пока не будет найден терм удовлетворяющий условиям унификации.</p>	<p>Прямой ход, переход к следующему предложению. (по аналогии, до конца процедур phonebook, automobile)</p>

6	Сопоставляем: depositor(name4, Bank, _, _). depositor(name4, tochka, 33333332, 300000). Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: Bank=tochka. Для поиска было проведено 14 сопоставлений	Истинное тело найдено. Name=name4, Number=1111111, City=spb, Bank=tochka. Откат, переход к следующему предложению.
7	Сопоставляем: depositor(name4, Bank, _, _). с термами в процедуре depositor. Все термы унифицируемы, но унификация завершается неудачей. Было проведено 2 сопоставления	Унификация завершилась неудачей. Откат
8	Сопоставляем: phonebook(name4, Number, address(City, _, _, _)) с термами в процедуре phonebook. Все термы процедуры унифицируемы, но унификация завершается неудачей. Было проведено 10 сопоставлений	Унификация завершилась неудачей. Откат
9	Сопоставляем: automobile(Name, volvo, silver,, _), с термами в процедуре phonebook. Все термы процедуры унифицируемы, но унификация завершается неудачей. Далее до конца БЗ термы не унифицируемы. Было проведено 6 сопоставлений	Унификация завершилась неудачей. Откат
10	Сопоставляем: search_by(volvo, silver, Name, City, Number, Bank), с термами до конца БЗ, унифицируемых термов не найдено.	Конец

Итого было проведено 48 сопоставлений.

«Автомобили» «Телефоны» «Инвесторы»

Таблица 7

№ шага	Сравниваемые термы; подстановка, если есть	Дальнейшие действия: прямой ход или откат
1	<p>Сопоставляем:  search_by(volvo, silver, Name, City, Number, Bank).  search_by(Brand, Color, Name, City, Number, Bank)  кладем в стек: search_by(...) = search_by(...).  Вытаскиваем из стека: search_by(...) = search_by(...).  Составные термы: имена главных функторов совпали, аргументность совпала – термы унифицируемы. В стек заносятся: Brand=volvo, Color=silver, Name= Name, City=City, Number=Number, Bank= Bank.  Результат унификации: успех  Наиболее общий унификатор: Brand=volvo, Color=silver.  Для поиска было проведено 16 сопоставлений.</p>	
2	<p>Сопоставляем:  automobile(Name, volvo, silver, _), с термами аналогично терму search_by (см. Таблица 2), пока не будет найден терм удовлетворяющий условиям унификации.</p>	<p>Прямой ход, переход к следующему предложению. (по аналогии, до конца процедуры phonebook)</p>
3	<p>Сопоставляем:  automobile(Name, volvo, silver, _),  automobile(name4, volvo, silver, 1300000)  Составные термы, унифицируемые.  Результат унификации: успех  Наиболее общий унификатор: Name=name4. Для поиска было проведено 4 сопоставлений</p>	<p>Прямой ход, переход к следующей подцели правила.</p>
4	<p>Сопоставляем:  phonebook(name4, Number, address(City, _, _, _)) phonebook(name4, 1111111, address(spb, alexandrova, 1, 10)).</p>	<p>Прямой ход, переход к следующей подцели правила.</p>

	<p>Составные термы, унифицируемые.  Результат унификации: успех  Наиболее общий унификатор:  Number=11111111, City=spb.  Для поиска было проведено 11 сопоставлений.</p>	
5	<p>Сопоставляем:  depositor(name4, Bank, _, _). с термами аналогично терму search_by (см. Таблица 2), пока не будет найден терм удовлетворяющий условиям унификации.</p>	<p>Прямой ход, переход к следующему предложению. (по аналогии, до конца процедур phonebook, automobile)</p>
6	<p>Сопоставляем:  depositor(name4, Bank, _, _).  depositor(name4, tochka, 333333332, 300000). Составные термы, унифицируемые.  Результат унификации: успех  Наиболее общий унификатор:  Bank=tochka.  Для поиска было проведено 14 сопоставлений</p>	<p>Истинное тело найдено.  Name=name4,  Number=11111111,  City=spb,  Bank=tochka.  Откат, переход к следующему предложению.</p>
7	<p>Сопоставляем:  depositor(name4, Bank, _, _). с термами в процедуре depositor.  Все термы унифицируемы, но унификация завершается неудачей. Было проведено 2 сопоставления</p>	<p>Унификация завершилась неудачей. Откат</p>
8	<p>Сопоставляем:  phonebook(name4, Number, address(City, _, _, _)) с термами в процедуре phonebook. Все термы процедуры унифицируемы, но унификация завершается неудачей.  Было проведено 5 сопоставлений</p>	<p>Унификация завершилась неудачей. Откат</p>
9	<p>Сопоставляем:  automobile(Name, volvo, silver,, _), с термами в процедуре phonebook. Все термы процедуры унифицируемы, но унификация завершается неудачей. Далее до конца БЗ термы не унифицируемы.</p>	<p>Унификация завершилась неудачей. Откат</p>



	Было проведено 11 сопоставлений	
10	Сопоставляем: search_by(volvo, silver, Name, City, Number, Bank), с термами до конца БЗ, унифицируемых термов не найдено.	Конец

Итого было проведено 48 сопоставлений.

**Вывод:** Пролог проходит по всем предложениям последовательно сравнивая текущий терм со всеми в разделе clauses. Поэтому в какой бы последовательности не шли предикаты, число проходов – объем работы останется тем же.

### Задание 3.

Таблица работы алгоритма унификации.

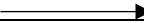

#### Goal

search\_by(volvo, silver, Name, City, Number, Bank)

Таблица 7

шаг	результатирующая ячейка	рабочее поле	пункт Алг.	стек
0			1.	search_by(volvo, silver, Name, City, Number, Bank) = search_by(Brand, Color, Name, City, Number, Bank)
1		search_by(volvo, silver, Name, City, Number, Bank) = search_by(Brand, Color, Name, City, Number, Bank) →	Е.	Brand=volvo, Color=silver, Name= Name, City=City, Number=Number, Bank=Bank
2	Brand=volvo	Brand=volvo ←	Г.	Color=silver, Name= Name, City=City, Number=Number, Bank=Bank
3	Brand=volvo, Color=silver	Color=silver ←	Г.	Name= Name, City=City, Number=Number, Bank=Bank
4	Brand=volvo, Color=silver, Name= Name	Name= Name ←	-	City=City, Number=Number, Bank=Bank
5	Brand=volvo, Color=silver, Name= Name, City=City	City=City ←	-	Number=Number, Bank=Bank
6	Brand=volvo, Color=silver, Name= Name, City=City, Number=Number	Number=Number ←	-	Bank=Bank
7	Brand=volvo, Color=silver, Name= Name, City=City, Number=Number, Bank=Bank	Bank=Bank	-	Переход в тело правила Name= Name, City=City, Number=Number, Bank=Bank – связанные переменные.

8	Новая цель: automobile(Name, volvo, silver, _) далее будет рассматриваться сразу подходящий терм.			
9		<pre> automobile(Name, volvo,             silver, _)             = automobile(name4, volvo,             silver, 1300000)           </pre>	Е.	Name=name4, volvo=volvo, silver=silver, Анонимные переменные не могут быть связаны.
10	<b>Name= name4,</b>	<pre> Name=name4,           </pre>	Г.	volvo=volvo, silver=silver, Анонимные переменные не могут быть связаны.
11	Name= name4	volvo=volvo	Б.	silver=silver
12	Name= name4	silver=silver	Б.	Стек пуст, очистка рабочего пространства
	Подстановка	Т.к. стек пуст – <b>успех</b> , Переход к следующей подцели тела.		
13	Новая цель: phonebook(name4, Number, adress(City, _, _)) далее будет рассматриваться сразу подходящий терм.			
14		<pre> phonebook(name4, Number, adress(City, _, _,                 _)) и phonebook(name4, 11111111, adress(spb, alexandrova, 1, 10)).           </pre>	Е.	name4=name4, Number= 11111111, City=spb.
15	Name= name4,	name4=name4	Б.	Number= 11111111, City=spb.
16	Name= name4, <b>Number= 11111111</b>	<pre> Number= 11111111           </pre>	Г.	City=spb.
17	Name= <b>name4</b> , Number= <b>11111111</b> ,  <b>City=spb</b>	<pre> City=spb           </pre>	Г.	Стек пуст, очистка рабочего пространства
	Подстановка	Т.к. стек пуст – <b>успех</b> , Переход к следующей подцели тела.		

18	Новая цель: depositor(name4, Bank, _, _). далее будет рассматриваться сразу подходящий терм.			
19		depositor(name4, Bank, _, _) = depositor(name4, tochka, 333333332, 300000)	Е.	name4=name4, Bank=tochka.
20		<div style="text-align: center;">  </div> name4=name4	Б.	Bank=tochka.
21	<b>Bank=tochka.</b>	Bank=tochka. <div style="text-align: center;">  </div>	Г.	Стек пуст, очистка рабочего пространства

Получена подстановка: {Name= name4, City=spb, Number=11111111, Bank=tochka},  
такая подстановка получилась поскольку переменные Name, City, Number, Bank –  
связанные именованные переменные. (двунаправленная передача параметров)