



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*

*По предмету: «Функциональное и логическое  
программирование»*

Преподаватель: Строганов Ю.В.

Студент: Мирзоян С.А.,

Группа: ИУ7-65Б

Москва, 2020 г.

## 1. Текст задания

Составить программу, т.е. модель предметной области – базу знаний, объединив в ней информацию – знания:

- **«Телефонный справочник»:** Фамилия, №тел, Адрес – структура (Город, Улица, №дома, №кв),
- **«Автомобили»:** Фамилия\_владельца, Марка, Цвет, Стоимость, и др.,
- **«Вкладчики банков»:** Фамилия, Банк, счет, сумма, др.

Владелец может иметь несколько телефонов, автомобилей, вкладов (Факты). Используя **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.

## 2. Ответы на вопросы

### 1. В какой части правила сформулировано знание? Это знание о чём, с формальной точки зрения?

Сформулировано в заголовке правила, это знание о том, подходит ли данное правило заданному вопросу или нет.

### 2. Что такое процедура?

Процедура - это совокупность правил, описывающих одно определенное отношение. Заголовки правил имеют одно и то же имя и одну и ту же аргументность.

### 3. Сколько в БЗ текущего задания процедур?

4 процедуры: person, car, deposit и person\_by\_car.

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

Пусть  $\Theta = \{X_1 = t_1, X_2 = t_2, \dots, X_n = t_n\}$  – подстановка, а  $A(X_1, \dots, X_n)$  – терм. Тогда результат применения подстановки к терму обозначается:  $A\Theta$ . Терм В называется примером терма А, если существует такая подстановка  $\Theta$ , что  $B = A\Theta$ . Соответственно, чтобы построить пример, необходимо заменить в терма А каждый  $X_i$  на соответствующий  $t_i$ .

### 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.

Терм С называется общим примером термов А и В, если существуют такие подстановки  $\Theta_1$  и  $\Theta_2$ , что  $C = A\Theta_1$  и  $C = B\Theta_2$

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

Для двух термов находится наибольший общий унификатор. Во время работы алгоритма переменные могут быть заменены значениями, при

которых тело данного правила истинно.

Двунаправленная передача данных – из внешнего мира в программу, и из программы во внешний мир.

Пример:

правило: `person_by_car(Brand, Color, Surname, Number, Bank) :-  
car(Surname, Brand, _, Color, _), person(Surname, Number, _),  
deposit(Surname, Bank, _, _).`

вызов: `person_by_car("Bugatti", "Black", Surname, Number, Bank).`

Передача в одном направлении: `Brand = "Bugatti", Color="Black"`

Передача в обратном направлении: `Surname, Number, Bank` заменяются на найденные для них значения.

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

При возникновении тупиковой ситуации при унификации.

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

Переменные бывают именованные и анонимные (обозначаются символом “\_”).

Пример:

`person_by_car(Brand, Color, Surname, Number, Bank) :- car(Surname,  
Brand, _, Color, _), person(Surname, Number, _), deposit(Surname, Bank,  
_, _).`

`Brand, Color` – переменные, по которым идет поиск,  
`Surname, Number, Bank` – переменные, значения которых являются результатом.

Остальные являются анонимными, так как их значения не представляют интереса и не влияют на ход работы.

### 3. Листинг кода

```
1.
    domains
2.    lastname, number, city, street, brand, color, bank, client_check =
    symbol.
3.    price, money = integer. % thousand
4.    address_t = address(city, street, integer, integer).
5.
6. predicates
7.    person(lastname, number, address_t).
8.    vehicle(lastname, brand, color, price).
9.    deposit(lastname, address_t, bank, client_check, money).
10.
11.        by_phn_num(number, lastname, brand, price).
12.        vehicle_phn(number, brand).
13.        lastname_city(lastname, city, street, bank, number).
14.
15.        vehicle_holder(Brand, Color, Lastname, Number, Bank).
16.
17.    clauses
18.        person("Ivanov", "000-000", address("Example", "street", 0,
0)).
19.        person("Ivanov", "111-111", address("City-17", "Gordon
street", 0, 0)).
20.        person("Petrov", "001-917", address("St. Petersburg",
"Lenina", 24, 42)).
21.        person("Sidorov", "555-555", address("Los Angeles", "Apple
street", 0, 1)).
22.        person("Johnson", "123-321", address("San Andreas", "Groove
street", 1, 2)).
23.
24.        vehicle("Ivanov", "Bugatti", "Black", 1178000).
25.        vehicle("Ivanov", "Aston Martin", "Grey", 230000).
26.        vehicle("Petrov", "Lada", "Red", 200).
27.        vehicle("Sidorov", "Ford", "Black", 400).
28.        vehicle("Petrov", "GAZ", "White", 200).
29.        vehicle("Sidorov", "Ford", "Red", 400).
30.        vehicle("Johnson", "Comet", "Green", 50000).
31.        vehicle("Tenpeny", "Rocket", "Black", 20000).
32.
```

```

33.         deposit("Ivanov", address("Example", "street", 0, 0),
        "Sberbank", "0-0-0-0", 999999999).

34.         deposit("Ivanov", address("Example", "street", 0, 0), "VTB",
        "0-0-0-1", 1).

35.         deposit("Ivanov", address("City-17", "Gordon street", 0, 0),
        "Tinkoff", "0-1-0-1", 987654).

36.         deposit("Petrov", address("St. Petersburg", "Lenina", 24, 42),
        "Alfa", "1-2-3-4", 999999999).

37.         deposit("Sidorov", address("Los Angeles", "Apple street", 0,
        1), "Mavrodi", "6-9-6-9", 1).

38.         deposit("Johnson", address("San Andreas", "Groove street", 1,
        2), "Vivaldi", "11-21-31-21", 987654).

39.

40.

41.         by_phn_num(Number, Lastname, Brand, Price) :- person(Lastname,
        Number, _), vehicle(Lastname, Brand, _, Price).

42.         vehicle_phn(Number, Brand) :- by_phn_num(Number, _, Brand, _).

43.         lastname_city(Lastname, City, Street, Bank, Number) :-
        person(Lastname, Number, address(City, Street, _, _)),
        deposit(Lastname, address(City, Street, _, _), Bank, _, _).

44.         vehicle_holder(Brand, Color, Lastname, Number, Bank) :-
        vehicle(Lastname, Brand, Color, _), person(Lastname, Number, _),
        deposit(Lastname, _, Bank, _, _).

45.

46.         goal

47.         %by_phn_num("001-917", Lastname, Brand, Price).

48.         %vehicle_phn("111-111", Brand).

49.         %lastname_city("Ivanov", "Example", Street, Bank, Number).

50.         vehicle_holder("Bugatti", "Black", Lastname, Number, Bank).

```

#### 4. Описание порядка поиска ответа

`vehicle_holder("Bugatti", "Black", Lastname, Number, Bank).`

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	<code>vehicle_holder("Bugatti", "Black", Lastname, Number, Bank).</code> и <code>vehicle_holder(Brand, Color, Lastname, Number, Bank) :- vehicle(Lastname, Brand, Color, _), person(Lastname, Number, _), deposit(Lastname, _, Bank, _, _).</code>  Запуск алгоритма унификации. Результат:  Brand = "Bugatti",  Color = "Black"	прямой ход. Нужно вычислить <code>vehicle</code> , <code>person</code> и <code>deposit</code>
2	<code>vehicle_holder("Lada", "Red", Lastname, Number, Bank).</code> и <code>vehicle("Ivanov", "Bugatti", , "Black", 1178000).</code>  Запуск алгоритма унификации. Результат:  Lastname = "Ivanov"	<code>vehicle</code> вычислен, теперь вычисляем <code>person</code> .
3	<code>person("Ivanov", Number, _)</code> и <code>person("Ivanov", "000-000", address("Example", "street", 0, 0)).</code>  Запуск алгоритма унификации. Результат:  Number = "000-000"	<code>person</code> вычислен, вычисляем <code>deposit</code> .
4	<code>deposit("Ivanov", Bank, _, _).</code> и <code>deposit("Ivanov", "Sberbank", "0-0-0-0", 999999999).</code>	<code>deposit</code> вычислен, получен результат 1, ищем остальные результаты

	Запуск алгоритма унификации. Результат: Bank = “Sberbank”	
5	deposit("Ivanov", Bank, _, _). и deposit("Ivanov", "VTB", "0-0-0-1", 1). Запуск алгоритма унификации. Результат: Bank = “VTB”	deposit вычислен, получен результат 2, ищем остальные результаты
6	deposit("Ivanov", Bank, _, _). и deposit("Petrov", "Alfa", "1-2-3-4", 999999999). Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “Petrov” не совпадают)	откат
7	deposit("Ivanov", Bank, _, _). и deposit("Sidorov", "Mavrodi", "6-9-6-9", 1). Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “Sidorov” не совпадают)	откат
8	deposit("Ivanov", Bank, _, _). и deposit("Johnson", address("San Andreas", "Groove street", 1, 2), "Vivaldi", "11-21-31-21", 987654). Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “Johnson” не совпадают)	откат. Все deposit при Lastname=“Ivanov” вычислены. Вычисляем оставшиеся person
9	person("Ivanov", Number, _) и person("Petrov", "001-917", address("St. Petersburg", "Lenina", 24, 42)).	откат



	<p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “Petrov” не совпадают)</p>	
10	<p>person("Ivanov", Number, _) и person("Sidorov", "555-555", address("Los Angeles", "Apple street", 0, 1)).</p> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “Sidorov” не совпадают)</p>	откат
11	<p>person("Ivanov", Number, _) и person("Krueger", "013-666", address("Springwood", "Elm street", 13, 13)).</p> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “Krueger” не совпадают)</p>	откат
12	<p>person("Ivanov", Number, _) и person("A", "123-456", address("B", "C avnenue", 13, 14)).</p> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “A” не совпадают)</p>	откат
13	<p>person("Ivanov", Number, _) и person("Johnson", "123-321", address("San Andreas", "Groove street", 1, 2)).</p> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “Johnson” не совпадают)</p>	Откат. Все person при Surname=“Ivanov” вычислены. Вычисляем новый vehicle
14	<p>vehicle(Lastname, "Bugatti", "Black", _) и</p>	откат

	<pre>vehicle("Petrov", "Lada", "White", 200).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Bugatti” и “Lada” не совпадают)</p>	
15	<pre>vehicle(Lastname, “Bugatti”, “Black”, _) и vehicle("Sidorov", "Lada", "White", 200).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Bugatti” и “Lada” не совпадают)</p>	откат
16	<pre>vehicle(Lastname, “Bugatti”, “Black”, _) и vehicle("Johnson", "Comet", "Green", 50000).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Bugatti” и “Comet” не совпадают)</p>	откат. vehicle вычислен, vehicle_holder вычислен, получено 2 результата.

vehicle\_holder("Comet", "Green", Lastname, Number, Bank).

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	<pre>vehicle_holder("Comet", "Green", Lastname, Number, Bank). и vehicle_holder (Brand, Color, Surname, Number, Bank) :- car(Surname, Brand, _, Color, _), person(Surname, Number, _), deposit(Surname, Bank, _, _).</pre>	прямой ход. Нужно вычислить vehicle, person и deposit

	<p>Запуск алгоритма унификации. Результат:</p> <p>Brand = “Comet”, Color = “Green”</p>	
2	<p>vehicle(Surname, “Comet”, “Green”, _) и vehicle(“Ivanov”, “Bugatti”, “Black”, 1178000).</p> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Comet” и “Bugatti” не совпадают)</p>	откат
...	...	...
3	<p>vehicle(Surname, “Comet”, “Green”, _) и vehicle(“Johnson”, “Comet”, “Green”, 50000).</p> <p>Запуск алгоритма унификации. Результат: Surname = “Petrov”</p>	Vehicle вычислен, вычисляем person
4	<p>person(“Johnson”, Number, _) и person(“Ivanov”, “000-000”, address(“Example”, “street”, 0, 0)).</p> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Johnson” и “Ivanov” не совпадают)</p>	откат
...	...	...
5	<p>person(“Johnson”, Number, _) и person(“Johnson”, “123-321”, address(“San Andreas”, “Groove street”, 1, 2)).</p> <p>Запуск алгоритма унификации. Результат: Number = “123-321”</p>	person вычислен, вычисляем deposit
6	<p>deposit(“Johnson”, Bank, , _).</p>	откат

	и deposit("Ivanov", "Sberbank", "0-0-0-0", 999999999) . Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “Johnson” не совпадают)	
...	...	...
8	deposit("Johnson", Bank, _, _). и deposit("Johnson", address("San Andreas", "Groove street", 1, 2), "Vivaldi", "11-21-31-21", 987654) . Запуск алгоритма унификации. Результат: Bank = “Vivaldi”	deposit вычислен, получен результат 1. Все deposit при Lastname=“Vivaldi” вычислены.  Vehical_holder вычислен, найден 1 результат

vehical\_holder ("Rocket", "Black", Lastname, Number, Bank) .

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	vehical_holder ("Ford", "Green", Lastname, Number, Bank) . и vehicle_holder(Brand, Color, Lastname, Number, Bank) :- vehicle(Lastname, Brand, Color, _), person(Lastname, Number, _), deposit(Lastname, _, Bank, _, _).  Запуск алгоритма унификации. Результат:  Brand = “Rocket”,  Color = “Black”	прямой ход. Нужно вычислить vehical, person и deposit
2	vehical_holder ("Ford", "Green", Lastname, Number, Bank) . и vehicle("Ivanov", "Bugatti", "Black", 1178000) .	откат

	<p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ford” и “Bugatti” не совпадают)</p>	
3	<pre>vehical_holder ("Ford", "Red", Lastname, Number, Bank).</pre> <p>и</p> <pre>vehicle("Petrov", "Lada", "Red", 200).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ford” и “Lada” не совпадают)</p>	откат
4	<pre>vehical_holder ("Ford", "Red", Lastname, Number, Bank).</pre> <p>и</p> <pre>vehicle("Ivanov", "Aston Martin", "Grey", 230000).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ford” и “Lada” не совпадают)</p>	откат
...	...	...
5	<pre>vehical_holder ("Rocket", "black", Lastname, Number, Bank).</pre> <p>и</p> <pre>vehicle("Tenpeny", "Rocket", "black", 20000).</pre> <p>Запуск алгоритма унификации. Результат: Lastname=”Tenpeny”</p>	прямой ход, вычисляем person
6	<pre>person("Tenpeny", Number, _)</pre> <p>и</p> <pre>person("Ivanov", "000-000", address("Example", "street", 0, 0)).</pre> <p>Запуск алгоритма унификации. Результат:</p>	откат

	термы не унифицируемы (константы “Тенпену” и “Ivanov” не совпадают)	
7	<pre>person("Tenpeny", Number, _) и person("Petrov", "001- 917", address("St. Petersburg", "Lenina", 24, 42)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “ Tenpeny ” и “Petrov” не совпадают)</p>	откат
8	<pre>person("Tenpeny", Number, _) и person("Sidorov", "555- 555", address("Los Angeles", "Apple street", 0, 1)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “ Tenpeny ” и “Sidorov” не совпадают)</p>	откат
9	<pre>person("Tenpeny", Number, _) и person("Johnson", "123- 321", address("San Andreas", "Groove street", 1, 2)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “ Tenpeny ” и “Johnson” не совпадают)</p>	откат. Все person при Lastname=” Tenpeny ” вычислены. vehicle вычислен. Vehicle_holder вычислен, результатов не найдено.

**5. Сравнение объема работ при разном порядке следования правил в БЗ**  
Prolog обрабатывает правило в порядке следования предикатов в его теле, а не в базе знаний, следовательно, их порядок в БЗ не влияет ни на ход работы, ни, тем более, на результат. Таблицы, соответственно, полностью совпадут.

## 6. Порядок работы алгоритма унификации вопроса и подходящего заголовка правила

```
vehicle_holder(Lada", "White", Lastname, Number, Bank). =
vehicle_holder(Brand, Color, Surname, Number, Bank) :- vehicle(Lastname,
Brand, Color, _), person(Lastname, Number, _), deposit(Lastname, Bank, _, _).
```

Prolog обрабатывает правило в порядке следования предикатов в его теле.

Изменение порядка следования предикатов в БЗ не изменит таблицу.

шаг унификации	результатирующая ячейка	рабочее поле	пункт алгоритма	стек
0			1.	vehicle_holder ("Bugatti", "Black", Lastname, Number, Bank). = vehicle_holder (Brand, Color, Lastname, Number, Bank) :- vehicle(Lastname, Brand, _, Color, _), person(Lastname, Number, _), deposit(Lastname, Bank, _, _).
1		vehicle_holder ("Lada ", " White ", Lastname, Number, Bank). = vehicle_holder (Brand, Color, Lastname, Number, Bank) :- vehicle(Lastname, Brand, Color, _), person(Lastname, Number, _), deposit(Lastname, Bank, _, _).--- →	e)	vehicle(lastname, "Lada", _, "White", _) = vehicle("Petrov", "Lada", "White", 200).  person(Lastname, Number, _)  deposit(Lastname, Bank, _, _).

2	Lastname = "Petrov"	vehixle(Lastname, "Lada", "White", _) = vehicle("Petrov", "Lada", "White", 200). ←-----→	e)	person("Petrov", Number, _) = person("Petrov", "001-917", address("St. Petersburg", "Lenina", 24, 42)).  deposit("Petrov", Bank, _, _). = deposit("Petrov", "Alfa", "1-2-3-4", 9999999999).
3	Lastname = "Petrov", Number = "001-917"	person("Petrov", Number, _) = person("Petrov", "001-917", address("St. Petersburg", "Lenina", 24, 42)). ←-----→	e)	deposit("Petrov", Bank, _, _). = deposit("Petrov", "Alfa", "1-2-3-4", 9999999999).
4	Lastname = "Petrov", Number = "001-917", Bank = "Alfa"	deposit("Petrov", Bank, _, _). = deposit("Petrov", "Alfa", "1-2-3-4", 9999999999). ←-----	e)	
Вывод:	<b>подстановка</b>	Т.к. стек пуст – <b>успех</b> и в рез. ячейке подстановка		