



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: Гасанзаде М.А.,

Группа: ИУ7-66Б

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 чем, с формальной точки зрения?

Знание зафиксировано в заголовке правила. Это знание о том. подходит ли данное правило заданному вопросу

2. Что такое процедура?

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

3. Сколько в БЗ текущего задания процедур?

Поскольку факт – частный случай правила, имеем 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 Θ , 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 Θ_1 и Θ_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.

Алгоритм унификации находит наибольший общий унификатор двух термов. Во время работы алгоритма переменные могут быть заменены значениями, при которых тело данного правила истинно. Эти значения

передадутся обратно во внешний мир. Это и есть 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 (например, при унификации `f(a,b)` и `f(c, d)`, если `a,b,c,d` – разные константы)

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

Переменные могут быть именованные или анонимные (у них имя – символ “_”). Пример:

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

Здесь `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. Например, нас не интересует, по какому адресу проживает найденный `person` или сколько денег у него на счету.

3. Листинг кода

```
domains
    surname, number, city, street, brand, model, color, bank, account =
symbol.
    price, money = integer. % thousand
    address_t = address(city, street, integer, integer).

predicates
    person(surname, number, address_t).
    car(surname, brand, model, color, price).
    deposit(surname, address_t, bank, account, money).
    all_by_phone(number, surname, brand, price).
    brand_by_phone(number, brand).
    by_surname_city(surname, city, street, bank, number).

clauses
    person("Ivanov", "000-000", address("Example", "street", 0, 0)).
    person("Petrov", "001-917", address("St. Petersburg", "Lenina", 24,
42)).
    person("Sidorov", "555-555", address("Los Angeles", "Apple street", 0,
1)).
    person("Krueger ", "013-666", address("Springwood", "Elm street", 13,
13)).
    person("A", "123-456", address("B", "C avnue", 13, 14)).
    person("Another", "123-321", address("One", "Good street", 3, 12)).
    person("One", "999-666", address("More", "Pioneer street", 3, 4)).
    person("Not", "987-654", address("Enough", "Bad Fantasy avenue", 9,
9)).
    car("Ivanov", "Bugatti", "La Voiture Noire", "Black", 1178000).
    car("Petrov", "Lada", "Kalina", "White", 200).
    car("Sidorov", "Lada", "Kalina", "White", 200).
    car("Noname", "Ford", "Focus", "Red", 400).
    deposit("Ivanov", "Sberbank", "0-0-0-0", 999999999).
    deposit("Ivanov", "VTB", "0-0-0-1", 1).
    deposit("Petrov", "Alfa", "1-2-3-4", 999999999).
    deposit("Sidorov", "Mavrodi", "6-9-6-9", 1).
    deposit("Another", "Bankname", "10-20-30-40", 40302010).
    person_by_car(Brand, Color, Surname, Number, Bank) :- car(Surname,
Brand, _, Color, _), person(Surname, Number, _), deposit(Surname, Bank, _,
_).

goal
    %person_by_car("Bugatti", "Black", Surname, Number, Bank).
    %person_by_car("Lada", "White", Surname, Number, Bank).
    %person_by_car("Ford", "Red", Surname, Number, Bank).
```

4. Описание порядка поиска ответа

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

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	person_by_car("Bugatti", "Black", Surname, Number, Bank). и person_by_car(Brand, Color, Surname, Number, Bank) :-	прямой ход. Нужно вычислить car, person и deposit

	<pre>car(Surname, Brand, _, Color, _), person(Surname, Number, _), deposit(Surname, Bank, _, _)).</pre> <p>Запуск алгоритма унификации. Результат:</p> <p>Brand = “Bugatti”, Color = “Black”</p>	
2	<pre>car(Surname, "Bugatti", _, "Black", _) и car("Ivanov", "Bugatti", "La Voiture Noire", "Black", 1178000).</pre> <p>Запуск алгоритма унификации. Результат:</p> <p>Surname = “Ivanov”</p>	car вычислен, вычисляем person
3	<pre>person("Ivanov", Number, _) и person("Ivanov", "000- 000", address("Example", "street", 0, 0)).</pre> <p>Запуск алгоритма унификации. Результат:</p> <p>Number = “000-000”</p>	person вычислен, вычисляем deposit.
4	<pre>deposit("Ivanov", Bank, _, _). и deposit("Ivanov", "Sberbank", "0-0-0-0", 999999999).</pre> <p>Запуск алгоритма унификации. Результат:</p> <p>Bank = “Sberbank”</p>	deposit вычислен, получен результат 1, ищем остальные результаты
5	<pre>deposit("Ivanov", Bank, _, _). и deposit("Ivanov", "VTB", "0-0-0-1", 1).</pre> <p>Запуск алгоритма унификации. Результат:</p> <p>Bank = “VTB”</p>	deposit вычислен, получен результат 2, ищем остальные результаты
6	<pre>deposit("Ivanov", Bank, _, _). и deposit("Petrov", "Alfa", "1-2-3-4", 999999999).</pre> <p>Запуск алгоритма унификации. Результат:</p> <p>термы не унифицируемы</p>	откат

	(константы “Ivanov” и “Petrov” не совпадают)	
7	<pre>deposit("Ivanov", Bank, _, _).</pre> и <pre>deposit("Sidorov", "Mavrodi", "6-9-6-9", 1).</pre> Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “Sidorov” не совпадают)	откат
8	<pre>deposit("Ivanov", Bank, _, _).</pre> и <pre>deposit("Another", "Bankname", "10-20-30-40", 40302010).</pre> Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “Another” не совпадают)	откат. Все deposit при Surname=“Ivanov” вычислены. Вычисляем оставшиеся person
9	<pre>person("Ivanov", Number, _)</pre> и <pre>person("Petrov", "001-917", address("St. Petersburg", "Lenina", 24, 42)).</pre> Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “Petrov” не совпадают)	откат
10	<pre>person("Ivanov", Number, _)</pre> и <pre>person("Sidorov", "555-555", address("Los Angeles", "Apple street", 0, 1)).</pre> Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “Sidorov” не совпадают)	откат
11	<pre>person("Ivanov", Number, _)</pre> и <pre>person("Krueger", "013-666", address("Springwood", "Elm street", 13, 13)).</pre> Запуск алгоритма унификации. Результат:	откат

	термы не унифицируемы (константы “Ivanov” и “Kueger” не совпадают)	
12	<pre>person("Ivanov", Number, _) и person("A", "123-456", address("B", "C avvenue", 13, 14)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “A” не совпадают)</p>	откат
13	<pre>person("Ivanov", Number, _) и person("Another", "123- 321", address("One", "Good street", 3, 12)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “Another” не совпадают)</p>	откат
14	<pre>person("Ivanov", Number, _) и person("One", "999-666", address("More", "Pioneer street", 3, 4)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “One” не совпадают)</p>	откат
15	<pre>person("Ivanov", Number, _) и person("Not", "987-654", address("Enough", "Bad Fantasy avenue", 9, 9)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “Not” не совпадают)</p>	откат. Все person при Surname=“Ivanov” вычислены. Вычисляем новый car
16	<pre>car(Surname, "Bugatti", _, "Black", _) и car("Petrov", "Lada", "Kalina", "White", 200).</pre>	откат

	Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Bugatti” и “Lada” не совпадают)	
17	<p>car(Surname, “Bugatti”, _, “Black”, _) и car("Sidorov", "Lada", "Kalina", "White", 200).</p> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Bugatti” и “Lada” не совпадают)</p>	откат
18	<p>car(Surname, “Bugatti”, _, “Black”, _) и car("Noname", "Ford", "Focus", "Red", 400).</p> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Bugatti” и “Ford” не совпадают)</p>	откат. car вычислен, person_by_car вычислен, получено 2 результата.

person_by_car("Lada", "White", Surname, Number, Bank).

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	<p>person_by_car("Lada", "White", Surname, Number, Bank). и person_by_car(Brand, Color, Surname, Number, Bank) :- car(Surname, Brand, _, Color, _), person(Surname, Number, _), deposit(Surname, Bank, _, _).</p> <p>Запуск алгоритма унификации. Результат:</p>	прямой ход. Нужно вычислить car, person и deposit

	Brand = “Lada”, Color = “White”	
2	<pre>car(Surname, "Lada", _, "White", _) и car("Ivanov", "Bugatti", "La Voiture Noire", "Black", 1178000).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Lada” и “Bugatti” не совпадают)</p>	откат
3	<pre>car(Surname, "Lada", _, "White", _) и car("Petrov", "Lada", "Kalina", "White", 200).</pre> <p>Запуск алгоритма унификации. Результат: Surname = “Petrov”</p>	car вычислен, вычисляем person
4	<pre>person("Petrov", Number, _) и person("Ivanov", "000-000", address("Example", "street", 0, 0)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Petrov” и “Ivanov” не совпадают)</p>	откат
5	<pre>person("Petrov", Number, _) и person("Petrov", "001-917", address("St. Petersburg", "Lenina", 24, 42)).</pre> <p>Запуск алгоритма унификации. Результат: Number = "001-917"</p>	person вычислен, вычисляем deposit
6	<pre>deposit("Petrov", Bank, _, _). и deposit("Ivanov", "Sberbank", "0-0-0-0", 999999999).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы</p>	откат

	(константы “Ivanov” и “Petrov” не совпадают)	
7	<pre>deposit("Petrov", Bank, _, _).</pre> и <pre>deposit("Ivanov", "VTB", "0-0-0-0", 1).</pre> Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ivanov” и “Petrov” не совпадают)	откат
8	<pre>deposit("Petrov", Bank, _, _).</pre> и <pre>deposit("Petrov", "Alfa", "1-2-3-4", 999999999).</pre> Запуск алгоритма унификации. Результат: Bank = “Alfa”	deposit вычислен, получен результат 1. Ищем остальные
9	<pre>deposit("Petrov", Bank, _, _).</pre> и <pre>deposit("Sidorov", "Mavrodi", "6-9-6-9", 1).</pre> Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Petrov” и “Sidorov” не совпадают)	откат
10	<pre>deposit("Petrov", Bank, _, _).</pre> и <pre>deposit("Another", "Bankname", "10-20-30-40", 40302010).</pre> Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Petrov” и “Another” не совпадают)	откат. Все deposit при Surname=“Petrov” вычислены. Вычисляем оставшиеся person
11	<pre>person("Petrov", Number, _)</pre> и <pre>person("Sidorov", "555-555", address("Los Angeles", "Apple street", 0, 1)).</pre> Запуск алгоритма унификации. Результат: термы не унифицируемы	откат

	(константы “Petrov” и “Sidorov” не совпадают)	
12	<pre>person("Petrov", Number, _) и person("Krueger", "013-666", address("Springwood", "Elm street", 13, 13)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Petrov” и “Krueger” не совпадают)</p>	откат
13	<pre>person("Petrov", Number, _) и person("A", "123-456", address("B", "C avenue", 13, 14)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Petrov” и “A” не совпадают)</p>	откат
14	<pre>person("Petrov", Number, _) и person("Another", "123-321", address("One", "Good street", 3, 12)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Petrov” и “Another” не совпадают)</p>	откат
15	<pre>person("Petrov", Number, _) и person("One", "999-666", address("More", "Pioneer street", 3, 4)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Petrov” и “One” не совпадают)</p>	откат

16	<pre>person("Petrov", Number, _) и person("Not", "987-654", address("Enough", "Bad Fantasy avenue", 9, 9)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы "Petrov" и "Not" не совпадают)</p>	откат. Все person при Surname="Petrov" вычислены. Вычисляем новый car
17	<pre>car(Surname, "Lada", _, "White", _) и car("Sidorov", "Lada", "Kalina", "White", 200).</pre> <p>Запуск алгоритма унификации. Результат: Surname = "Sidorov"</p>	car вычислен, вычисляем person
18	<pre>person("Sidorov", Number, _) и person("Ivanov", "000-000", address("Example", "street", 0, 0)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы "Sidorov" и "Ivanov" не совпадают)</p>	откат
19	<pre>person("Sidorov", Number, _) и person("Petrov", "001-917", address("St. Petersburg", "Lenina", 24, 42)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы "Sidorov" и "Petrov" не совпадают)</p>	откат
20	<pre>person("Sidorov", Number, _) и person("Sidorov", "555-555", address("Los Angeles", "Apple street", 0, 1)).</pre> <p>Запуск алгоритма унификации. Результат: Number="555-555"</p>	person вычислен, вычисляем deposit

21	<pre>deposit("Sidorov", Bank, _, _). и deposit("Ivanov", "Sberbank", "0-0-0-0", 999999999).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы "Sidorov" и "Ivanov" не совпадают)</p>	откат
22	<pre>deposit("Sidorov", Bank, _, _). и deposit("Ivanov", "VTB", "0-0-0-0", 1).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы "Sidorov" и "Ivanov" не совпадают)</p>	откат
23	<pre>deposit("Sidorov", Bank, _, _). и deposit("Petrov", "Alfa", "1-2-3-4", 999999999).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы "Sidorov" и "Petrov" не совпадают)</p>	откат
24	<pre>deposit("Sidorov", Bank, _, _).и deposit("Sidorov", "Mavrodi", "6-9-6-9", 1).</pre> <p>Запуск алгоритма унификации. Результат: Bank = "Mavrodi"</p>	deposit вычислен, получен результат 2. Ищем остальные
25	<pre>deposit("Sidorov", Bank, _, _). и deposit("Another", "Bankname", "10-20-30-40", 40302010).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы "Sidorov" и "Another" не совпадают)</p>	откат. Все deposit при Surname="Sidorov" вычислены. Вычисляем оставшиеся person

26	<pre>person("Sidorov", Number, _) и person("Krueger", "013-666", address("Springwood", "Elm street", 13, 13)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Sidorov” и “Krueger” не совпадают)</p>	откат
27	<pre>person("Sidorov", Number, _) и person("A", "123-456", address("B", "C avnenue", 13, 14)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Sidorov” и “A” не совпадают)</p>	откат
28	<pre>person("Sidorov", Number, _) и person("Another", "123-321", address("One", "Good street", 3, 12)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Sidorov” и “Another” не совпадают)</p>	откат
29	<pre>person("Sidorov", Number, _) и person("One", "999-666", address("More", "Pioneer street", 3, 4)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Sidorov” и “One” не совпадают)</p>	откат
30	<pre>person("Sidorov", Number, _) и person("Not", "987-654", address("Enough", "Bad Fantasy avenue", 9, 9)).</pre>	откат. Все person при Surname=”Sidorov”

	Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Sidorov” и “Not” не совпадают)	вычислены. Вычисляем новый car
31	<pre>car(Surname, "Lada", _, "White", _) и car("Noname", "Ford", "Focus", "Red", 400).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Lada” и “Ford” не совпадают)</p>	car вычислен. person_by_car вычислен, получено 2 результата

person_by_car("Ford", "Red", Surname, Number, Bank).

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	<pre>person_by_car("Ford", "Red", Surname, Number, Bank). и person_by_car(Brand, Color, Surname, Number, Bank) :- car(Surname, Brand, _, Color, _), person(Surname, Number, _), deposit(Surname, Bank, _, _).</pre> <p>Запуск алгоритма унификации. Результат: Brand = “Ford”, Color = “Red”</p>	прямой ход. Нужно вычислить car, person и deposit
2	<pre>car(Surname, "Ford", _, "Red", _) и car("Ivanov", "Bugatti", "La Voiture Noire", "Black", 1178000).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы</p>	откат

	(константы “Ford” и “Bugatti” не совпадают)	
3	<pre>car(Surname, "Ford", _, "Black", _) и car("Petrov", "Lada", "Kalina", "White", 200).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ford” и “Lada” не совпадают)</p>	откат
4	<pre>car(Surname, "Ford", _, "Black", _) и car("Sidorov", "Lada", "Kalina", "White", 200).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Ford” и “Lada” не совпадают)</p>	откат
5	<pre>car(Surname, "Ford", _, "Black", _) и car("Noname", "Ford", "Focus", "Red", 400).</pre> <p>Запуск алгоритма унификации. Результат: Surname=”Noname”</p>	прямой ход, вычисляем person
6	<pre>person("Noname", Number, _) и person("Ivanov", "000-000", address("Example", "street", 0, 0)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Noname” и “Ivanov” не совпадают)</p>	откат
7	<pre>person("Noname", Number, _) и person("Petrov", "001-917", address("St. Petersburg", "Lenina", 24, 42)).</pre>	откат

	<p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Noname” и “Petrov” не совпадают)</p>	
8	<pre>person("Noname", Number, _) и person("Sidorov", "555-555", address("Los Angeles", "Apple street", 0, 1)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Noname” и “Sidorov” не совпадают)</p>	откат
9	<pre>person("Noname", Number, _) и person("Krueger", "013-666", address("Springwood", "Elm street", 13, 13)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Noname” и “Krueger” не совпадают)</p>	откат
10	<pre>person("Noname", Number, _) и person("Another", "123-321", address("One", "Good street", 3, 12)).</pre> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы “Noname” и “Another” не совпадают)</p>	откат
11	<pre>person("Noname", Number, _) и person("One", "999-666", address("More", "Pioneer street", 3, 4)).</pre> <p>Запуск алгоритма унификации. Результат:</p>	откат

	термы не унифицируемы (константы "Noname" и "One" не совпадают)	
1ц	<p>person("Noname", Number, _) и person("Not", "987-654", address("Enough", "Bad Fantasy avenue", 9, 9)).</p> <p>Запуск алгоритма унификации. Результат: термы не унифицируемы (константы "Noname" и "Not" не совпадают)</p>	<p>откат. Все person при Surname="Noname" вычислены. car вычислен. person_by_car вычислен, результатов не найдено.</p>

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

```

person_by_car("Lada", "White", Surname, Number, Bank). = person_by_car(Brand,
Color, Surname, Number, Bank) :- car(Surname, Brand, _, Color, _),
person(Surname, Number, _), deposit(Surname, Bank, _, _).

```

шаг унификации	результатирующая ячейка	рабочее поле	пункт алгоритма	стек
0			1.	<p>person_by_car("Bugatti", "Black", Surname, Number, Bank). =</p> <p>person_by_car(Brand, Color, Surname, Number, Bank) :-</p> <p>car(Surname, Brand, _, Color, _),</p> <p>person(Surname,</p>

				Number, _), deposit(Surname, Bank, _, _).
1		person_by_car("Lada ", " White ", Surname, Number, Bank). = person_by_car(Brand, Color, Surname, Number, Bank) :- car(Surname, Brand, _, Color, _), person(Surname, Number, _), deposit(Surname, Bank, _, _).---→	e)	car(Surname, “Lada”, _, “White”, _) = car("Petrov", "Lada", "Kalina", "White", 200). person(Surname, Number, _) deposit(Surname, Bank, _, _).
2	Surname = "Petrov"	car(Surname, “Lada”, _, “White”, _) = car("Petrov", "Lada", "Kalina", "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", 999999999).
3	Surname = "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", 999999999).

4	Surname = "Petrov", Number = "001-917", Bank = "Alfa"	deposit("Petrov",,, Bank, _, _). = deposit("Petrov", "Alfa", "1-2-3-4", 999999999). ←-----	e)	
Вы вод:	подстанов ка	Т.к. стек пуст – успех и в рез. ячейке подстановка		

Так как Prolog обрабатывает правило в порядке следования предикатов в его теле, а не в базе знаний, изменение порядка следования предикатов в БЗ не изменит таблицу.