

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 <u>«Программное обеспечение ЭВМ и информационные технологии»</u>	

Лабораторная работа № 14

Дисциплина Функциональное и логическое программирование

Тема <u>Работа программы на Prolog</u>
Студент Ильясов И. М.
Группа ИУ7-63Б
Оценка (баллы)
Преполаватель Толпинская Н. Б., Строганов Ю. В.

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 <u>структура</u> (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", "Krilatskaya", 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).



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).



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

Ответ на задание 1

 Car_Color = "White", Car_Mark = "Toyota", find_result(Car_Mark, Car_Color, Surname, City, Phone, Bank_Name) – ненужные промежуточные шаги опущены.

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

Результат – find result("Toyota", "White", Surname, City, Phone, Bank_Name). 3 Производится сравнение car(Surname, "Toyota", "White", City) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Результат – неудача. 4 Производится сравнение car(Surname, "Toyota", "White", City) и саr("Gorbunov", "Audi", "White", City) и саr("Gorbunov", "White", City), проверка следующей строки " (все остальные факты саг до следующего не улификцируются) 5 Производится сравнение car(Surname, "Toyota", "White", City) и саr("Gorbunov", "Toyota", "White", City рани a "Saint-Petersburg",		"White".	City). Car_Mark = "Toyota", Car_Color =
Производится сравнение саг(Surname, "Toyota", "White", _, City) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Результат — неудача. Производится сравнение саг(Surname, "Toyota", "White", _, City) и саг("Gorbunov", "Audi", "White", _, City) и саг("Gorbunov", "White", _, City), проверка следующей строки Производится сравнение саг(Surname, "Toyota", "White", _, City) и саг("Gorbunov", "White", _, City) и саг("Gorbunov", "White", _, City), проверка следующей строки Производится сравнение саг(Surname конкретизируется с "Gorbunov", City — с "Korolev". Результат — неудача. Производится сравнение саг(Surname, "Toyota", "White", _, City) и саг("Gorbunov", "тоyota", "White", _, City) и саг("Gorbunov", "тоyota", "White", _, City) и саг("Gorbunov", унификации аbonement(Surname, Phone, address(City, _, _, _)). При этом переменная Surname равна "Gorbunov", City — с "Saint-Petersburg". Саг Mark — "Toyota", "Car Mark — "Toyota", Car Color — "White". Производится сравнение аbonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)), и abonement("Gorbunov", "Gity равна "Saint-Petersburg", _, _, _)), проверка следующей строки Производится сравнение аbonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)), проверка следующей строки Производится сравнение аbonement("Gorbunov", "Glavnaya", 55, 122)). Ропое конкретизируется с "89251472838". Результат — неудача.		Результат – find_result("Toyota", "White",	"White"
"Toyota", "White", _, City) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Результат – неудача. 4 Производится сравнение саг(Surname, "Toyota", "White", _, City) и саг("Gorbunov", "Audi", "White", _, "Korolev"). Surname конкретизируется с "Gorbunov", City — с "Korolev". Результат — неудача. 5 Производится сравнение саг(Surname, "Toyota", "White", _, City) и саг("Gorbunov", "City _ с "Saint-Petersburg"). Випатав конкретизируется с "Gorbunov", "Toyota", "White", "Saint-Petersburg"). 1 Производится сравнение аномине премения Surname равна "Gorbunov", "City двна "Saint-Petersburg", Car_Mark - "Toyota", Car_Color - "White". 6 Производится сравнение аномине премения Surname равна "Gorbunov", "City двна "Saint-Petersburg", Car_Mark - "Toyota", Car_Color - "White". 6 Производится сравнение аномине премения Surname равнение аноменени ("Gorbunov", "Flone, address("Saint-Petersburg", _, _, _)), проверка следующей строки "White". 6 Производится сравнение аномине премения Surname равнение аноменение ("Gorbunov", "Glavnaya", 55, 122)). Рhone конкретизируется с "89251472838", дезультат — неудача.		Surname, City, Phone, Bank_Name).	
abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Результат — неудача. ——————————————————————————————————	3	Производится сравнение car(Surname,	Откат к car(Surname, "Toyota",
address("Korolev", "Glavnaya", 55, 122)). Результат – неудача. (остальные факты abonement также не унифицируются) 4 Производится сравнение car(Surname, "Toyota", "White", _, City) и саг("Gorbunov", "White", _, City) и саг("Gorbunov", "White", _, City), проверка следующей строки 5 Производится сравнение саг до следующего не унифицируются) 5 Производится сравнение саг (Surname, "Toyota", "White", _, City) и саг("Gorbunov", City — с "Saint-Petersburg"). Surname конкретизируется с "Gorbunov", City — с "Saint-Petersburg". Peзультат — саг("Gorbunov", "Toyota", "White", "Saint-Petersburg"). """ 6 Производится сравнение abonement("Gorbunov", "Toyota", "White", "Saint-Petersburg"). """ 6 Производится сравнение abonement("Gorbunov", "Hone, address("Saint-Petersburg", _, _, _,)). и abonement("Gorbunov", "Boyota", "White". """ 6 Производится сравнение abonement("Gorbunov", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Peзультат — неудача.		"Toyota", "White", _, City) и	"White", _, City), проверка следующей
Результат — неудача. ——————————————————————————————————		abonement("Gorbunov", "89251472838",	строки
(остальные факты abonement также не унифицируются) 4 Производится сравнение саг(Surname, "Toyota", "White", _, City) и саг("Gorbunov", "White", _, City) и саг("Gorbunov", "White", _, City), проверка следующей строки конкретизируется с "Gorbunov", City — с "Korolev". Результат — неудача. 5 Производится сравнение саг(Surname, "Toyota", "White", _, City) и саг("Gorbunov", "Toyota", "White", _, Saint-Petersburg"). Surname конкретизируется с "Gorbunov", City — с "Saint-Petersburg". Результат — саг("Gorbunov", "Toyota", "White", "Saint-Petersburg"). 6 Производится сравнение abonement("Gorbunov", "Toyota", "Saint-Petersburg"). 6 Производится сравнение abonement("Gorbunov", "Reg251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Результат — неудача.		address("Korolev", "Glavnaya", 55, 122)).	
4 Производится сравнение car(Surname, "Toyota", "White", _, City) и car("Gorbunov", "White", _, City), проверка следующей строки Откат к car(Surname, "Toyota", "White", _, City), проверка следующей строки "Аиді", "White", _, "Korolev"). Surname конкретизируется с "Gorbunov", City – с "Korolev". ————————————————————————————————————		Результат – неудача.	
4 Производится сравнение car(Surname, "Toyota", "White", _, City) и car("Gorbunov", "White", _, City), проверка следующей строки Откат к car(Surname, "Toyota", "White", _, City), проверка следующей строки "Аиді", "White", _, "Korolev"). Surname конкретизируется с "Gorbunov", City – с "Korolev". ————————————————————————————————————			
4 Производится сравнение car(Surname, "Toyota", "White", _, City) и car("Gorbunov", "White", _, City) и car("Gorbunov", "White", _, City), проверка следующей строки "Audi", "White", _, "Korolev"). Surname конкретизируется с "Gorbunov", City — с "Korolev". Результат — неудача. 5 Производится сравнение car(Surname, "Toyota", "White", _, City) и car("Gorbunov", "Toyota", "White", _, "Saint-Petersburg"). Прямой ход, производится переход к унификации abonement(Surname, Phone, address(City, _, _, _)). При этом переменная Surname равна "Gorbunov", City — с "Saint-Petersburg". Результат — car("Gorbunov", "Toyota", "White", "Saint-Petersburg"). 1 6 Производится сравнение аbonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)). и abonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)), проверка следующей строки 6 Производится сравнение аbonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)), проверка следующей строки 6 Производится сравнение аbonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)), проверка следующей строки 6 Производится сравнение аbonement ("Gorbunov", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Результат — неудача.		(остальные факты abonement также не	
"Toyota", "White", _, City) и саг("Gorbunov", "White", _, City), проверка следующей строки "Audi", "White", _, "Korolev"). Surname конкретизируется с "Gorbunov", City – с "Korolev". Результат – неудача. "Тоуота", "White", _, City) и саг("Gorbunov", "Toyota", "White", _, City) и саг("Gorbunov", City — с "Saint-Petersburg"). Surname конкретизируется с "Gorbunov", City — с "Saint-Petersburg". Результат – саг("Gorbunov", "Toyota", "White", "Saint-Petersburg"). ""White", "Saint-Petersburg". Производится сравнение аbonement("Gorbunov", "Toyota", "White", "Saint-Petersburg"). """ """ """ """ """ """ """		унифицируются)	
"Audi", "White", _, "Korolev"). Surname конкретизируется с "Gorbunov", City – с "Korolev". Результат – неудача. Производится сравнение саг(Surname, "Toyota", "White", _, City) и саг("Gorbunov", City – с "Saint-Petersburg"). Surname конкретизируется с "Gorbunov", City – с "Saint-Petersburg". Результат – саг("Gorbunov", "Toyota", "White", "Saint-Petersburg"). Производится сравнение аbonement("Gorbunov", "Toyota", "White", "Saint-Petersburg"). Производится сравнение аbonement("Gorbunov", "Toyota", "White". Производится сравнение abonement("Gorbunov", "Gity — c "Saint-Petersburg"). Производится сравнение abonement("Gorbunov", "Gity pasha "Saint-Petersburg", — address("Saint-Petersburg", — —)). и abonement("Gorbunov", "Inposepka следующей строки"), проверка следующей строки "89251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Результат – неудача.	4	Производится сравнение car(Surname,	Откат к car(Surname, "Toyota",
конкретизируется с "Gorbunov", City – с "Korolev". Результат – неудача. (все остальные факты саг до следующего не унифицируются) Лроизводится сравнение саг(Surname, "Toyota", "White", _, City) и саг("Gorbunov", City — с "Saint-Petersburg"). Surname конкретизируется с "Gorbunov", City — с "Saint-Petersburg". Результат — саг("Gorbunov", "Toyota", "White", "Saint-Petersburg"). Производится сравнение аbonement("Gorbunov", "Toyota", "White", "Saint-Petersburg"). Производится сравнение abonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)). и abonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)), и poверка следующей строки Производится сравнение abonement("Gorbunov", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Результат — неудача.		"Toyota", "White", _, City) и car("Gorbunov",	"White", _, City), проверка следующей
"Когоlev". Результат – неудача. (все остальные факты саг до следующего не унифицируются) Производится сравнение саг(Surname, "Toyota", "White", _, City) и саг("Gorbunov", "Toyota", "White", _, Saint-Petersburg"). Surname конкретизируется с "Gorbunov", City — с "Saint-Petersburg". — с "Saint-Petersburg". Phone, address(City, _, _, _)). При этом переменная Surname paвна "Gorbunov", City paвна "Saint-Petersburg", Car_Mark — "Toyota", Car_Color — "White". 6 Производится сравнение abonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)). и abonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)). и abonement("Gorbunov", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Откат к abonement("Gorbunov", проверка следующей строки — производится сравнение abonement("Gorbunov", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Откат к abonement("Gorbunov", проверка следующей строки		"Audi", "White", _, "Korolev"). Surname	строки
— Сезультат — неудача. — Откат к abonement("Gorbunov", "89251472838", даdress("Когоlev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Результат — неудача. — Откат к аbоnement("Gorbunov", проверка следующей строки переверка следующей строки перементая Surname равна "Gorbunov", Сіту равна "Saint-Petersburg", Саг_Mark — "Toyota", Саг_Mark — "		конкретизируется с "Gorbunov", City – с	
(все остальные факты саг до следующего не унифицируются) Лроизводится сравнение саг(Surname, "Toyota", "White", _, City) и саг("Gorbunov", "Toyota", "White", _, "Saint-Petersburg"). Surname конкретизируется с "Gorbunov", City _ c "Saint-Petersburg". Peзультат — саг("Gorbunov", "Toyota", "White", "Saint-Petersburg"). Производится сравнение abonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)). и abonement("Gorbunov", "R9251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Pезультат — неудача. Прямой ход, производится переход к унификации abonement(Surname, Phone, address(City, _, _, _)). При этом переменная Surname paвна "Gorbunov", City paвна "Saint-Petersburg", Car_Mark — "Toyota", Car_Color — "White". Откат к abonement("Gorbunov", "Phone, address("Saint-Petersburg", _, _, _)), проверка следующей строки		"Korolev".	
унифицируются)5Производится сравнение car(Surname, "Toyota", "White", _, City) и car("Gorbunov", "Toyota", "White", _, "Saint-Petersburg"). Surname конкретизируется с "Gorbunov", City — с "Saint-Petersburg".Phone, address(City, _, _, _)). При этом переменная Surname равна "Gorbunov", City равна "Saint-Petersburg", "Car_Mark — "Toyota", Car_Color — "White".6Производится сравнение abonement("Gorbunov", Phone, address("Saint- petersburg", _, _, _)). и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Pезультат — неудача.Откат к abonement("Gorbunov", phone, address("Saint-Petersburg", _, _, _,)), проверка следующей строки		Результат – неудача.	
унифицируются)5Производится сравнение car(Surname, "Toyota", "White", _, City) и car("Gorbunov", "Toyota", "White", _, "Saint-Petersburg"). Surname конкретизируется с "Gorbunov", City — с "Saint-Petersburg".Phone, address(City, _, _, _)). При этом переменная Surname равна "Gorbunov", City равна "Saint-Petersburg", "Car_Mark — "Toyota", Car_Color — "White".6Производится сравнение abonement("Gorbunov", Phone, address("Saint- petersburg", _, _, _)). и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Pезультат — неудача.Откат к abonement("Gorbunov", phone, address("Saint-Petersburg", _, _, _,)), проверка следующей строки			
Производится сравнение car(Surname, "Toyota", "White", _, City) и car("Gorbunov", унификации abonement(Surname, Phone, address(City, _, _, _)). При этом переменная Surname paвна "Gorbunov", City — с "Saint-Petersburg". City равна "Saint-Petersburg", Peзультат — car("Gorbunov", "Toyota", "White", "Saint-Petersburg"). Производится сравнение abonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)). и abonement("Gorbunov", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Pезультат — неудача.		(все остальные факты саг до следующего не	
"Toyota", "White", _, City) и car("Gorbunov", "Toyota", "White", _, "Saint-Petersburg"). Surname конкретизируется с "Gorbunov", City _ c "Saint-Petersburg". Peзультат — car("Gorbunov", "Toyota", "White", "Saint-Petersburg"). Производится сравнение abonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)). и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Pезультат — неудача. "Yнификации abonement(Surname, Phone, address(City, _, _, _)). При этом переменная Surname paвна "Gorbunov", City paвна "Saint-Petersburg",		унифицируются)	
"Toyota", "White", _, "Saint-Petersburg"). Surname конкретизируется с "Gorbunov", City — с "Saint-Petersburg". Peзультат — car("Gorbunov", "Toyota", "White", "Saint-Petersburg"). Производится сравнение abonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)). и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Результат — неудача. Phone, address(City, _, _, _)). При этом переменная Surname paвна "Gorbunov", City paвна "Saint-Petersburg", Car_Mark — "Toyota", Car_Color — "White". OTKAT к abonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)), проверка следующей строки	5	Производится сравнение car(Surname,	Прямой ход, производится переход к
Surname конкретизируется с "Gorbunov", City — с "Saint-Petersburg". Результат — car("Gorbunov", "Toyota", "White", "Saint-Petersburg"). Производится сравнение abonement("Gorbunov", Phone, address("Saint-Petersburg",,,)). и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Результат — неудача. переменная Surname paвна "Gorbunov", City равна "Saint-Petersburg", Car_Mark — "Toyota", Car_Color — "White". Откат к abonement("Gorbunov", Phone, address("Saint-Petersburg",,,)), проверка следующей строки		"Toyota", "White", _, City) и car("Gorbunov",	унификации abonement(Surname,
— с "Saint-Petersburg".City равна " Saint-Petersburg ",Результат — саг("Gorbunov", "Toyota",Car_Mark — "Toyota", Car_Color —"White", "Saint-Petersburg")—6Производится сравнение abonement("Gorbunov", Phone, address("Saint- Petersburg", _, _, _)). и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838".Откат к abonement("Gorbunov", address("Saint-Petersburg", _, _, _)), проверка следующей строки—проверка следующей строки		"Toyota", "White", _, "Saint-Petersburg").	Phone, address(City, _, _, _)). При этом
Результат — car("Gorbunov", "Toyota", "White", "Saint-Petersburg"). б Производится сравнение abonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)). и abonement("Gorbunov", "Saint-Petersburg", _, _, _)). и abonement("Gorbunov", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Результат — неудача. Сат_Mark — "Toyota", Car_Color — "White". Откат к abonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)), проверка следующей строки		Surname конкретизируется с "Gorbunov", City	переменная Surname равна "Gorbunov",
"White", "Saint-Petersburg"). Производится сравнение abonement("Gorbunov", Phone, address("Saint- Petersburg", _, _, _)). и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Результат — неудача. "White". Откат к abonement("Gorbunov", phone, address("Saint-Petersburg", _, _, _)), проверка следующей строки		- c "Saint-Petersburg".	City равна " Saint-Petersburg ",
6 Производится сравнение abonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)). и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". проверка следующей строки Результат – неудача.		Результат – car("Gorbunov", "Toyota",	Car_Mark – "Toyota", Car_Color –
abonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)), проверка следующей строки "89251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Результат — неудача.		"White", "Saint-Petersburg").	"White".
abonement("Gorbunov", Phone, address("Saint-Petersburg", _, _, _)), проверка следующей строки "89251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Результат — неудача.			
Реtersburg", _, _, _)). и abonement("Gorbunov", проверка следующей строки "89251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Результат – неудача.	6	Производится сравнение	Откат к abonement("Gorbunov", Phone,
"89251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с "89251472838". Результат — неудача		abonement("Gorbunov", Phone, address("Saint-	address("Saint-Petersburg", _, _, _)),
55, 122)). Phone конкретизируется с "89251472838". Результат — неудача		Petersburg", _, _, _)). и abonement("Gorbunov",	проверка следующей строки
"89251472838". Результат – неудача. 		"89251472838", address("Korolev", "Glavnaya",	
Результат – неудача		55, 122)). Phone конкретизируется c	
		"89251472838".	
(все остальные факты abonement до		Результат – неудача.	
(все остальные факты abonement до			
		(все остальные факты abonement до	
следующего не унифицируются)		следующего не унифицируются)	
7 Производится сравнение Прямой ход, производится переход к	7	Производится сравнение	Прямой ход, производится переход к
abonement("Gorbunov", Phone, address("Saint- унификации deposit(Surname,		abonement("Gorbunov", Phone, address("Saint-	унификации deposit(Surname,
Petersburg", _, _, _)). и abonement("Gorbunov", Bank_Name, _, _, City). При этом		Petersburg", _, _, _)). и abonement("Gorbunov",	Bank_Name, _, _, City). При этом

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

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

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

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

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

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.

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

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

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.

Oтвет на задание 3 find_result("Toyota", "White", Surname, City, Phone, Bank_Name)

Шаг	Результиру-ющая	Рабочее поле	Пункт	Стек
ун-	ячейка		ал-ма	
ции				
1			1	find_result("Toyota", "White", Surname, City, Phone, Bank_Name) = find_result(Car_Mark, Car_Color, Surname, City, Phone, Bank_Name)
2		find_result("Toyota", "White", Surname, City, Phone, Bank_Name) = 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).		Car_Mark = "Toyota" Car_Color = "White" car(Surname, Car_Mark, Car_Color, _, City) abonement(Surname, Phone, address(City, _, _, _)) deposit(Surname, Bank_Name, _, _, City)
3	Car_Mark = "Toyota"	← Car_Mark = "Toyota"	Γ	Car_Color = "White" car(Surname, Car_Mark, Car_Color, _, City) abonement(Surname, Phone, address(City, _, _, _)) deposit(Surname, Bank_Name, _, _, City)
4	Car_Mark = "Toyota" Car_Color = "White"	← Car_Color = "White"	Γ	car(Surname, Car_Mark, Car_Color, _, City) abonement(Surname, Phone, address(City, _, _, _)) deposit(Surname, Bank_Name, _, _, City)
		(перебор фактов abonement)		

5	Car_Mark =	car(Surname, "Toyota", "White", _,		Surname = "Gorbunov"
	"Toyota"	City)		"Toyota" = "Toyota"
	Car_Color = "White"	car("Gorbunov", "Audi", "White", 2500000, "Korolev").		"White" = "White"
		Неуспешная унификация!		abonement(Surname, Phone, address(City, _, _, _))
				deposit(Surname, Bank_Name, _, _, City)
		(перебор оставшихся фактов саг до нужного)		
6	Car_Mark = "Toyota"	car(Surname, "Toyota", "White", _, City)	e	Surname = "Gorbunov"
	Car_Color = "White"	car("Gorbunov", "Toyota", "White",		"Toyota" = "Toyota" "White" = "White"
		600000, "Saint-Petersburg"). Успешная унификация →		abonement(Surname, Phone, address(City, _ , _ , _))
				deposit(Surname, Bank_Name, _, _, City)
7	Car_Mark =	← Surname = "Gorbunov"	Γ	"Toyota" = "Toyota"
	"Toyota"			"White" = "White"
	Car_Color = "White" Surname =			abonement(Surname, Phone, address(City, _, _, _))
	"Gorbunov"			deposit(Surname, Bank_Name, _, _, City)
8	Car Mark =	"Toyota" = "Toyota"	б	"White = "White"
	"Toyota"			abonement(Surname, Phone,
	Car_Color = "Red"			address(City, _, _, _))
	Surname = "Gorbunov"			deposit(Surname, Bank_Name, _, _, City)
9	Car_Mark = "Toyota"	"White" = "White"	б	abonement(Surname, Phone, address(City, _, _, _))
	Car_Color = "White"			deposit(Surname,
	Surname = "Gorbunov"			Bank_Name, _, _, City)
10	Car_Mark = "Toyota"	abonement(Surname, Phone, address(City, _, _, _))		abonement(Surname, Phone, address(City, _, _, _))
	Car_Color = "White"	abonement("Gorbunov",		deposit(Surname,
	Surname = "Gorbunov"	"89251472838", address("Korolev", "Glavnaya", 55, 122)).		Bank_Name, _, _, City)
	Goroune v	Неуспешная унификация!		
		(перебор оставшихся фактов abonement до нужного)		
11	Car_Mark = "Toyota"	abonement(Surname, Phone, address(City, _, _, _))		"Gorbunov" = "Gorbunov" Phone = "165615253616"
	Car_Color = "White" Surname = "Containing or ""	abonement("Gorbunov", "165615253616", address("Saint-Petersburg", "Nevskaya", 51, 122)).		City = "Saint-Petersburg"
	"Gorbunov"			

		Успешная унификация 🔾		deposit(Surname, Bank_Name, _, _, City)
12	Car_Mark = "Toyota" Car_Color = "White" Surname = "Gorbunov"	"Gorbunov" = "Gorbunov"	6	Phone = "165615253616" City = "Saint-Petersburg" deposit(Surname, Bank_Name, _, _, City)
13	Car_Mark = "Toyota" Car_Color = "White" Surname = "Gorbunov" Phone = "165615253616"	← Phone = "165615253616"	г	City = "Saint-Petersburg" deposit(Surname, Bank_Name, _, _, City)
14	Car_Mark = "Toyota" Car_Color = "White" Surname = "Gorbunov" Phone = "165615253616" City = "Saint-Petersburg"	← City = "Saint-Petersburg"	Γ	deposit(Surname, Bank_Name, _, _, City)
		(перебор фактов abonement, car до deposit)		
15	Car_Mark = "Toyota" Car_Color = "White" Surname = "Gorbunov" Phone = "165615253616" City = "Saint-Petersburg"	deposit(Surname, Bank_Name, _, _, City) deposit("Gorbunov", "Sberbank", "123456789", 30000, "Korolev") Неуспешная унификация!		deposit(Surname, Bank_Name, _, _, City)
		(перебор оставшихся фактов deposit до нужного)		
16	Car_Mark = "Toyota" Car_Color = "White" Surname = "Gorbunov" Phone = "165615253616" City = "Saint-Petersburg"	deposit(Surname, Bank_Name, _, _, City) deposit("Gorbunov", "Sberbank", "563281726", 100000, "Saint-Petersburg") Успешная унификация →		"Gorbunov" = "Gorbunov" Bank_Name = "Sberbank"

17	Car_Mark = "Toyota"	"Gorbunov" = "Gorbunov"	б	Bank_Name = "Sberbank"
	Car_Color = "White"			
	Surname = "Gorbunov"			
	Phone = "165615253616"			
	City = "Saint- Petersburg"			
18	Car_Mark = "Toyota"	← Bank_Name = "Sberbank"	Γ	
	Car_Color = "White"			
	Surname = "Gorbunov"			
	Phone = "165615253616"			
	City = "Saint- Petersburg"			
	Bank_Name = "Sberbank"			
19-21		Откат к шагу 16, перебор фактов deposit с оставшимися фактами и правилами (find, find_result) Неуспешная унификация!		deposit(Surname, Bank_Name, _, _, City)
22-38		Откат к шагу 11, перебор фактов abonement с оставшимися фактами и правилами (car, deposit, find, find_result) Неуспешная унификация!		abonement(Surname, Phone, address(City, _, _, _)) deposit(Surname, Bank_Name, _, _, City)
39-46		Откат к шагу 6, перебор фактов саг с оставшимися фактами и правилами (deposit, find, find_result) Неуспешная унификация!		car(Surname, Car_Mark, Car_Color, _, City) abonement(Surname, Phone, address(City, _, _, _))
				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: -B1, ..., Bn, 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).

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.