|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № 14**

|  |  |
| --- | --- |
| **Дисциплина** Функциональное и логическое программирование  **Тема** \_Работа программы на Prolog\_  **Студент** \_Ильясов И. М.\_  **Группа** \_ИУ7-63Б\_  **Оценка (баллы)** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Преподаватель** \_Толпинская Н. Б., Строганов Ю. В.\_ |  |

Москва, 2020 г.

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

**Задачи работы**:

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

**Задание лабораторной работы**

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

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

Владелец может иметь несколько телефонов, автомобилей, вкладов (Факты). В разных городах есть однофамильцы, в одном городе – фамилия уникальна.

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

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

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

1. Для каждого из трех вариантов **словесно подробно** описать порядок формирования ответа (в виде таблицы). При этом, указать – отметить моменты очередного запуска алгоритма унификации и полный результат его работы. Обосновать следующий шаг работы системы. Выписать унификаторы – подстановки. Указать моменты, причины и результат отката, если он есть.
2. Для случая нескольких владельцев (2-х):

приведите примеры (таблицы) работы системы **при разных порядках** следования в БЗ процедур, и знаний в них: (**«Телефонный справочник», «Автомобили», «Вкладчики банков»,** или: **«Автомобили», «Вкладчики банков», «Телефонный справочник»**). Сделайте **вывод:** Одинаковы ли: множество работ и объем работ в разных случаях?

1. 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).



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**

1. Car\_Color = "White",

Car\_Mark = "Toyota",

find\_result(Car\_Mark, Car\_Color, Surname, City, Phone, Bank\_Name) – ненужные промежуточные шаги опущены.

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат (к чему приводит?) |
| 1 | Производится сравнение find\_result(“Toyota”, “White”, Surname, City, Phone, Bank\_Name) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).  Результат – неудача. | Откат к find\_result(“Toyota”, “White”, Surname, City, Phone, Bank\_Name) |
|  | … | … |
| 2 | Производится сравнение find\_result(“Toyota”, “White”, Surname, City, Phone, Bank\_Name) и find\_result(Car\_Mark, Car\_Color, Surname, City, Phone, Bank\_Name). Car\_Mark конкретизируется с “Toyota”, Car\_Color – с “White”.  Результат – find\_result(“Toyota”, “White”, Surname, City, Phone, Bank\_Name). | Прямой ход, производится переход к унификации car(Surname, Car\_Mark, Car\_Color, \_, City), abonement(Surname, Phone, address(City, \_, \_, \_)), deposit(Surname, Bank\_Name, \_, \_, City). Car\_Mark = “Toyota”, Car\_Color = “White” |
| 3 | Производится сравнение car(Surname, “Toyota”, “White”, \_, City) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).  Результат – неудача. | Откат к car(Surname, “Toyota”, “White”, \_, City), проверка следующей строки |
|  | …  (остальные факты abonement также не унифицируются) | … |
| 4 | Производится сравнение car(Surname, “Toyota”, “White”, \_, City) и car("Gorbunov", "Audi", “White”, \_, “Korolev”). Surname конкретизируется с “Gorbunov”, City – с “Korolev”.  Результат – неудача. | Откат к car(Surname, “Toyota”, “White”, \_, City), проверка следующей строки |
|  | …  (все остальные факты car до следующего не унифицируются) | … |
| 5 | Производится сравнение car(Surname, “Toyota”, “White”, \_, City) и car("Gorbunov", "Toyota", “White”, \_, “Saint-Petersburg”). Surname конкретизируется с “Gorbunov”, City – с “Saint-Petersburg”.  Результат – car(“Gorbunov”, “Toyota”, “White”, “Saint-Petersburg”). | Прямой ход, производится переход к унификации abonement(Surname, Phone, address(City, \_, \_, \_)). При этом переменная Surname равна “Gorbunov”, City равна “ Saint-Petersburg ”, Car\_Mark – “Toyota”, Car\_Color – “White”. |
|  | … | … |
| 6 | Производится сравнение abonement(“Gorbunov”, Phone, address(“Saint-Petersburg”, \_, \_, \_)). и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Phone конкретизируется с “89251472838”.  Результат – неудача. | Откат к abonement(“Gorbunov”, Phone, address(“Saint-Petersburg”, \_, \_, \_)), проверка следующей строки |
|  | …  (все остальные факты abonement до следующего не унифицируются) | … |
| 7 | Производится сравнение abonement(“Gorbunov”, Phone, address(“Saint-Petersburg”, \_, \_, \_)). и abonement("Gorbunov", "165615253616", address("Saint-Petersburg", "Glavnaya", 55, 122)). Phone конкретизируется с “165615253616”.  Результат – abonement(“Gorbunov”, “165615253616”, address(“Saint-Petersburg”, \_, \_, \_)) | Прямой ход, производится переход к унификации deposit(Surname, Bank\_Name, \_, \_, City). При этом переменная Surname равна “Gorbunov”, City равна “Saint-Petersburg”, Phone – “165615253616”, Car\_Mark – “Toyota”, Car\_Color – “White”. |
|  | … | … |
| 8 | Производится сравнение deposit(“Gorbunov”, Bank\_Name, \_, \_, “Saint-Petersburg”). и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).  Результат – неудача. | Откат к сравнение deposit(“Gorbunov”, Bank\_Name, \_, \_, “Saint-Petersburg”), проверка следующей строки |
|  | …  (все остальные факты abonement не унифицируются) | … |
| 9 | Производится сравнение deposit(“Gorbunov”, Bank\_Name, \_, \_, “Saint-Petersburg”). и deposit("Gorbunov", "Sberbank", "123456789", 30000, "Korolev"). Bank\_Name конкретизируется с “Sberbank”.  Результат – неудача. | Откат к сравнение deposit(“Gorbunov”, Bank\_Name, \_, \_, “Saint-Petersburg”), проверка следующей строки |
|  | …  (все остальные факты deposit до следующего не унифицируются) | … |
| 10 | Производится сравнение deposit(“Gorbunov”, Bank\_Name, \_, \_, “Saint-Petersburg”). и deposit("Gorbunov", "Sberbank", "563281726", 100000, "Saint-Petersburg"). Bank\_Name конкретизируется с “Sberbank”.  Результат – deposit(“Gorbunov”, “Sberbank”, \_, \_, “Saint-Petersburg”) | Прямой ход, производится подстановка значений, которые были найдены в исходный вопрос. При этом Surname = “Gorbunov”, Bank\_Name = “Sberbank”, Phone = “165615253616”, City = “Saint-Petersburg”, Car\_Mark = “Toyota”, Car\_Color = “White”. |
|  | … | … |
| 11 | car(“Gorbunov”, “Toyota”, “White”, \_, “Saint-Petersburg”), abonement(“Gorbunov”, “165615253616”, address(“Saint-Petersburg”, \_, \_, \_)), deposit(“Gorbunov”, “Sberbank”, \_, \_, “Saint-Petersburg”) |  |
|  | … | … |
|  | (Результат – 1 решение) | |

1. Car\_Color = "Black",

Car\_Mark = "Tesla",

car(Surname, Car\_Mark, Car\_Color, \_, City),

abonement(Surname, Phone, address(City, \_, \_, \_)),

deposit(Surname, Bank\_Name, \_, \_, City) – ненужные промежуточные шаги опущены.

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат (к чему приводит?) |
| 1 | Производится сравнение find\_result(“Tesla”, “Black”, Surname, City, Phone, Bank\_Name) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).  Результат – неудача. | Откат к find\_result(“Tesla”, “Black”, Surname, City, Phone, Bank\_Name) |
|  | … | … |
| 2 | Производится сравнение find\_result(“Tesla”, “Black”, Surname, City, Phone, Bank\_Name) и find\_result(Car\_Mark, Car\_Color, Surname, City, Phone, Bank\_Name). Car\_Mark конкретизируется с “Tesla”, Car\_Color – с “Black”.  Результат – find\_result(“Tesla”, “Black”, Surname, City, Phone, Bank\_Name). | Прямой ход, производится переход к унификации car(Surname, Car\_Mark, Car\_Color, \_, City), abonement(Surname, Phone, address(City, \_, \_, \_)), deposit(Surname, Bank\_Name, \_, \_, City). Car\_Mark = “Tesla”, Car\_Color = “Black” |
| 3 | Производится сравнение car(Surname, “Tesla”, “Black”, \_, City) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Surname конкретизируется с “Gorbunov”, City – с “Korolev”.  Результат – неудача. | Откат к car(Surname, “Tesla”, “Black”, \_, City), проверка следующей строки |
|  | …  (все остальные факты abonement не унифицируются) | … |
| 4 | Производится сравнение car(Surname, “Tesla”, “Black”, \_, City) и car("Gorbunov", "Audi", “White”, \_, “Korolev”). Surname конкретизируется с “Gorbunov”, City – с “Korolev”.  Результат – неудача. | Откат к car(Surname, “Tesla”, “Black”, \_, City), проверка следующей строки |
|  | …  (все остальные факты car до следующего не унифицируются) | … |
| 5 | Производится сравнение car(Surname, “Tesla”, “Black”, \_, City) и car("Ilyasov", "Tesla", “Black”, \_, “Moscow”). Surname конкретизируется с “Ilyasov”, City – с “Moscow”.  Результат – car(“Ilyasov”, “Tesla”, “Black”, “Moscow”). | Прямой ход, производится переход к унификации abonement(Surname, Phone, address(City, \_, \_, \_)). При этом переменная Surname равна “Ilyasov”, City равна “Moscow”, Car\_Mark – “Tesla”, Car\_Color – “Black”. |
|  | …  (все остальные факты deposit не унифицируются) | … |
| 6 | Производится сравнение abonement(“Ilyasov”, Phone, address(“Moscow”, \_, \_, \_)). и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).  Результат – неудача. | Откат к abonement(“Ilyasov”, Phone, address(“Moscow”, \_, \_, \_)), проверка следующей строки |
| 7 | Производится сравнение abonement(“Ilyasov”, Phone, address(“Moscow”, \_, \_, \_)). и abonement("Ilyasov", "89969503880", address("Moscow", "Severnaya", 12, 75)). Phone конкретизируется с “89969503880”.  Результат – abonement(“Ilyasov”, 89969503880, address(“Moscow”, \_, \_, \_)). | Прямой ход, производится переход к унификации deposit(Surname, Bank\_Name, \_, \_, City). При этом переменная Surname равна “Ilyasov”, City равна “Moscow”, Phone – “89969503880”, Car\_Mark – “Tesla”, Car\_Color – “Black”. |
|  | …  (все остальные факты abonement не унифицируются) | … |
| 8 | Производится сравнение deposit(“Ilyasov”, Bank\_Name, \_, \_, “Moscow”). и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Bank\_Name конкретизируется с “Sberbank”.  Результат – неудача. | Откат к deposit(“Ilyasov”, Bank\_Name, \_, \_, “Moscow”), проверка следующей строки |
|  | …  (все остальные факты abonement не унифицируются) | … |
| 9 | Производится сравнение deposit(“Ilyasov”, Bank\_Name, \_, \_, “Moscow”). и deposit("Ilyasov", "Sberbank", "987654321", 20000, "Moscow"). Bank\_Name конкретизируется с “Sberbank”.  Результат – deposit(“Ilyasov”, “Sberbank”, \_, \_, “Moscow”). | Откат, производится подстановка значений, которые были найдены в исходный вопрос. При этом Surname = “Ilyasov”, Bank\_Name = “Sberbank”, Phone = “89969503880”, City = “Moscow”, Car\_Mark = “Tesla”, Car\_Color = “Black”. |
| 10 | car(“Ilyasov”, “Tesla”, “Black”, \_, “Moscow”), abonement(“Ilyasov”, “987654321”, address(“Moscow”, \_, \_, \_)), deposit(“Ilyasov”, “Sberbank”, \_, \_, “Moscow”) |  |
| 11 | Производится сравнение car(Surname, “Tesla”, “Black”, \_, City) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Surname конкретизируется с “Gorbunov”, City – с “Korolev”.  Результат – неудача. | Откат к car(Surname, “Tesla”, “Black”, \_, City), проверка следующей строки |
|  | …  (все остальные факты abonement не унифицируются) | … |
| 12 | Производится сравнение car(Surname, “Tesla”, “Black”, \_, City) и car("Gorbunov", "Audi", “White”, \_, “Korolev”). Surname конкретизируется с “Gorbunov”, City – с “Korolev”.  Результат – неудача. | Откат к car(Surname, “Tesla”, “Black”, \_, City), проверка следующей строки |
|  | …  (все остальные факты car до следующего не унифицируются) | … |
| 13 | Производится сравнение car(Surname, “Tesla”, “Black”, \_, City) и car("Sidenko", "Tesla", “Black”, \_, “Moscow”). Surname конкретизируется с “Sidenko”, City – с “Moscow”.  Результат – car(“Sidenko”, “Tesla”, “Black”, “Moscow”). | Прямой ход, производится переход к унификации abonement(Surname, Phone, address(City, \_, \_, \_)). При этом переменная Surname равна “Sidenko”, City равна “Moscow”, Car\_Mark – “Tesla”, Car\_Color – “Black”. |
|  | …  (все остальные факты car не унифицируются) | … |
| 14 | Производится сравнение abonement(“Sidenko”, Phone, address(“Moscow”, \_, \_, \_)). и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).  Результат – неудача. | Откат к abonement(“Sidenko”, Phone, address(“Moscow”, \_, \_, \_)), проверка следующей строки |
|  | …  (все остальные факты abonement до следующего не унифицируются) | … |
| 15 | Производится сравнение abonement(“Sidenko”, Phone, address(“Moscow”, \_, \_, \_)). и abonement("Sidenko", "89691929395", address("Moscow", "Semenovskaya", 25, 53)). Phone конкретизируется с “89691929395”.  Результат – abonement(“Sidenko”, “89691929395”, address(“Moscow”, \_, \_, \_)). | Прямой ход, производится переход к унификации deposit(Surname, Bank\_Name, \_, \_, City). При этом переменная Surname равна “Sidenko”, City равна “Moscow”, Phone – “89691929395”, Car\_Mark – “Tesla”, Car\_Color – “Black”. |
|  | …  (все остальные факты abonement не унифицируются) | … |
| 16 | Производится сравнение deposit(“Sidenko”, Bank\_Name, \_, \_, “Moscow”). и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Bank\_Name конкретизируется с “Sberbank”.  Результат – неудача. | Откат к deposit(“Sidenko”, Bank\_Name, \_, \_, “Moscow”), проверка следующей строки |
|  | …  (все остальные факты deposit до следующего не унифицируются) | … |
| 17 | Производится сравнение deposit(“Sidenko”, Bank\_Name, \_, \_, “Moscow”). и deposit("Sidenko", "Tinkoff", "135798642", 60000, "Moscow"). Bank\_Name конкретизируется с “Tinkoff”.  Результат – deposit(“Sidenko”, “Tinkoff”, \_, \_, “Moscow”). | Откат, производится подстановка значений, которые были найдены в исходный вопрос. При этом Surname = “Sidenko”, Bank\_Name = “Tinkoff”, Phone = “89251472838”, City = “Moscow”, Car\_Mark = “Tesla”, Car\_Color = “Black”. |
| 18 | car(“Sidenko”, “Tesla”, “Black”, \_, “Moscow”), abonement(“Ilyasov”, “135798642”, address(“Moscow”, \_, \_, \_)), deposit(“Sidenko”, “Tinkoff”, \_, \_, “Moscow”) |  |
|  | … | … |
|  | (Результат – 2 решения) | |

1. Car\_Color = "White",

Car\_Mark = "Tesla",

car(Surname, Car\_Mark, Car\_Color, \_, City),

abonement(Surname, Phone, address(City, \_, \_, \_)),

deposit(Surname, Bank\_Name, \_, \_, City) – ненужные промежуточные шаги опущены.

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат (к чему приводит?) |
| 1 | Производится сравнение find\_result(“Tesla”, “White”, Surname, City, Phone, Bank\_Name) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)).  Результат – неудача. | Откат к find\_result(“Tesla”, “White”, Surname, City, Phone, Bank\_Name) |
|  | … | … |
| 2 | Производится сравнение find\_result(“Tesla”, “White”, Surname, City, Phone, Bank\_Name) и find\_result(Car\_Mark, Car\_Color, Surname, City, Phone, Bank\_Name). Car\_Mark конкретизируется с “Tesla”, Car\_Color – с “White”.  Результат – find\_result(“Tesla”, “White”, Surname, City, Phone, Bank\_Name). | Прямой ход, производится переход к унификации car(Surname, Car\_Mark, Car\_Color, \_, City), abonement(Surname, Phone, address(City, \_, \_, \_)), deposit(Surname, Bank\_Name, \_, \_, City). Car\_Mark = “Tesla”, Car\_Color = “White” |
| 3 | Производится сравнение car(Surname, “Tesla”, “White”, \_, City) и abonement("Gorbunov", "89251472838", address("Korolev", "Glavnaya", 55, 122)). Surname конкретизируется с “Gorbunov”, City – с “Korolev”.  Результат – неудача. | Откат к car(Surname, “Tesla”, “White”, \_, City), проверка следующей строки |
|  | …  (все остальные факты abonement не унифицируются) | … |
| 4 | Производится сравнение car(Surname, “Tesla”, “White”, \_, City) и car("Gorbunov", "Audi", “White”, \_, “Korolev”). Surname конкретизируется с “Gorbunov”, City – с “Korolev”.  Результат – неудача. | Откат к car(Surname, “Tesla”, “White”, \_, City), проверка следующей строки |
|  | …  (все остальные факты car не унифицируются) | … |
| 5 | Производится сравнение car(Surname, “Tesla”, “Black”, \_, City) и deposit("Gorbunov", "Sberbank", "123456789", 30000, "Korolev"). Surname конкретизируется с “Ilyasov”, City – с “Moscow”.  Результат – неудача | Откат к car(Surname, “Tesla”, “White”, \_, City), проверка следующей строки |
|  | … | … |
|  | (Результат – 0 решений) | |

**Ответ на задание 2**

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

**Ответ на задание 3**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Шаг  ун-ции | Результиру-ющая ячейка | Рабочее поле | Пункт  ал-ма | Стек |
| 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” | г | 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” | г | car(Surname, Car\_Mark, Car\_Color, \_, City)  abonement(Surname, Phone, address(City, \_, \_, \_))  deposit(Surname, Bank\_Name, \_, \_, City) |
|  | … | …  (перебор фактов abonement) | … | … |
| 5 | Car\_Mark = “Toyota”  Car\_Color = “White” | car(Surname, “Toyota”, “White”, \_, City)  car("Gorbunov", "Audi", "White", 2500000, "Korolev").  Неуспешная унификация! |  | Surname = “Gorbunov”  “Toyota” = “Toyota”  “White” = “White”  abonement(Surname, Phone, address(City, \_, \_, \_))  deposit(Surname, Bank\_Name, \_, \_, City) |
|  | … | …  (перебор оставшихся фактов car до нужного) | … | … |
| 6 | Car\_Mark = “Toyota”  Car\_Color = “White” | car(Surname, “Toyota”, “White”, \_, City)  car(“Gorbunov”, “Toyota”, “White”, 600000, “Saint-Petersburg”).  Успешная унификация 🡪 | е | Surname = “Gorbunov”  “Toyota” = “Toyota”  “White” = “White”  abonement(Surname, Phone, address(City, \_, \_, \_))  deposit(Surname, Bank\_Name, \_, \_, City) |
| 7 | Car\_Mark = “Toyota”  Car\_Color = “White”  Surname = “Gorbunov” | 🡨 Surname = “Gorbunov” | г | “Toyota” = “Toyota”  “White” = “White”  abonement(Surname, Phone, address(City, \_, \_, \_))  deposit(Surname, Bank\_Name, \_, \_, City) |
| 8 | Car\_Mark = “Toyota”  Car\_Color = “Red”  Surname = “Gorbunov” | “Toyota” = “Toyota” | б | “White = “White”  abonement(Surname, Phone, address(City, \_, \_, \_))  deposit(Surname, Bank\_Name, \_, \_, City) |
| 9 | Car\_Mark = “Toyota”  Car\_Color = “White”  Surname = “Gorbunov” | “White” = “White” | б | abonement(Surname, Phone, address(City, \_, \_, \_))  deposit(Surname, Bank\_Name, \_, \_, City) |
| 10 | Car\_Mark = “Toyota”  Car\_Color = “White”  Surname = “Gorbunov” | abonement(Surname, Phone, address(City, \_, \_, \_))  abonement(“Gorbunov”, “89251472838”, address(“Korolev”, “Glavnaya”, 55, 122)).  Неуспешная унификация! |  | abonement(Surname, Phone, address(City, \_, \_, \_))  deposit(Surname, Bank\_Name, \_, \_, City) |
|  | … | …  (перебор оставшихся фактов abonement до нужного) | … | … |
| 11 | Car\_Mark = “Toyota”  Car\_Color = “White”  Surname = “Gorbunov” | abonement(Surname, Phone, address(City, \_, \_, \_))  abonement(“Gorbunov”, “165615253616”, address(“Saint-Petersburg”, “Nevskaya”, 51, 122)).  Успешная унификация 🡪 |  | “Gorbunov” = “Gorbunov”  Phone = “165615253616”  City = “Saint-Petersburg”  deposit(Surname, Bank\_Name, \_, \_, City) |
| 12 | Car\_Mark = “Toyota”  Car\_Color = “White”  Surname = “Gorbunov” | “Gorbunov” = “Gorbunov” | б | 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” | г | 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”  Car\_Color = “White”  Surname = “Gorbunov”  Phone = “165615253616”  City = “Saint-Petersburg” | “Gorbunov” = “Gorbunov” | б | Bank\_Name = “Sberbank” |
| 18 | Car\_Mark = “Toyota”  Car\_Color = “White”  Surname = “Gorbunov”  Phone = “165615253616”  City = “Saint-Petersburg”  Bank\_Name = “Sberbank” | 🡨 Bank\_Name = “Sberbank” | г |  |
| **Вывод**  **подце-ли** | Подстановка | Так как стек пуст – успех  🡨 Подстановка |  |  |
| 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 car 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) |

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

**Ответы на вопросы**

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

Правила – предложение вида , 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.

1. **Что такое процедура?**

Процедура – совокупность правил, заголовки которых одинаковы.

1. **Сколько в БЗ текущего задания процедур?**

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

1. **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.

1. **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.

1. **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).

1. **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.

1. **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.