**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 курсу «Функциональное и логическое программирование».

Студент: Власова Е. В.

Группа: ИУ7-64Б.

Преподаватели: Толпинская Н. Б.

Строганов Ю. В.

*2020 г.*

**Практическая часть.**

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.

**Листинг программы:**

|  |
| --- |
| domains  home = address(CITY, STREET, HOUSE, FLAT)  SURNAME = string  PHONE = string  CITY, STREET = string  HOUSE, FLAT = integer  MARKA, COLOR = string  PRICE = integer  BANK = string  NUMBER, MONEY = integer  predicates  info(SURNAME, PHONE, home)  cars(SURNAME, CITY, MARKA, COLOR, PRICE)  banks(SURNAME, CITY, BANK, NUMBER, MONEY)  find\_by\_car(SURNAME, CITY, PHONE, MARKA, COLOR, BANK)  clauses  info("Petrov", "88885353535", address("Moscow", "Saykin street", 10, 143)).  info("Petrov", "81111111111", address("Kursk", "Non street", 1, 10)).  info("Krot", "89995353566", address("Moscow", "Nwot street", 5, 60)).  %2 owners  cars("Petrov", "Moscow", "TOYOTA", "RED", 256000). %PERSON 1  cars("Krot", "Moscow", "TOYOTA", "RED", 256000). %PERSON 3  %1 owner  cars("Petrov", "Kursk", "LADA", "GREEN", 100000). %PERSON 2  %0 owners  cars("Krotova", "Moscow", "Merc", "BLUE", 100000).    banks("Petrov", "Moscow", "Sberbank", 356, 50000). %PERSON 1  banks("Petrov", "Kursk", "Tinkoff", 357, 9000). %PERSON 2  banks("Krot", "Moscow", "Rocket-bank", 20, 2000). %PERSON 3    find\_by\_car(SURNAME, CITY, PHONE, MARKA, COLOR, BANK):- cars(SURNAME, CITY, MARKA, COLOR, \_),  info(SURNAME, PHONE, address(CITY, \_, \_, \_)),  banks(SURNAME, CITY, BANK, \_, \_).  goal  find\_by\_car(SURNAME, CITY, PHONE, "TOYOTA", "RED", BANK). |

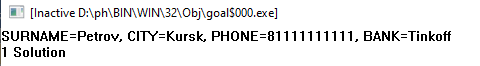
**Задание 1.**Используя **конъюнктивное правило и простой вопрос**, обеспечить возможность поиска:

По Марке и Цвету автомобиля найти Фамилию, Город, Телефон и Банки, в которых владелец автомобиля имеет вклады.

|  |
| --- |
| find\_by\_car(SURNAME, CITY, PHONE, MARKA, COLOR, BANK) :- cars(SURNAME, CITY, MARKA, COLOR, \_),  info(SURNAME, PHONE,  address(CITY, \_, \_, \_)), banks(SURNAME, CITY, BANK, \_, \_). |

1. У машины один владелец:

|  |
| --- |
| Goal  find\_by\_car(SURNAME, CITY, PHONE, "LADA", "GREEN", BANK). |

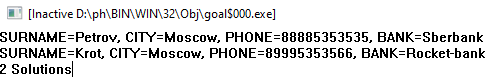


Порядок формирования ответа для примера 1:

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат |
| 0 | find\_by\_car(SURNAME, CITY, PHONE, MARKA, COLOR, BANK)  =  find\_by\_car(SURNAME, CITY, PHONE, «LADA», «GREEN», BANK) | Заносится в стек find\_by\_car(SURNAME, CITY, PHONE, «LADA», «GREEN», BANK) |
| 1 | Раскрывается правило, то есть надо доказать каждое утверждение в теле правила (последовательно)  cars(SURNAME, CITY, MARKA, COLOR, PRICE)  =  cars(SURNAME, CITY, «LADA», «GREEN», PRICE) | Прямой ходб  Заносится в стек  cars(SURNAME, CITY, «LADA», «GREEN», \_) |
| 2 | cars(SURNAME, CITY, «LADA», «GREEN», \_)  По функтору и списку аргументов ищется определение отношения  cars("Petrov", "Moscow", "TOYOTA", "RED", 256000) | Определение отношения найдено |
| 3 | Унификация cars(SURNAME, CITY, «LADA», «GREEN», \_) и cars("Petrov", "Moscow", "TOYOTA", "RED", 256000) | Результат сравнения термов false, прямой ход |
| 4 | Унификация cars(SURNAME, CITY, «LADA», «GREEN», \_) и  cars("Krot", "Moscow", "TOYOTA", "RED", 256000) | Результат сравнения термов false, прямой ход |
| 5 | Унификация cars(SURNAME, CITY, «LADA», «GREEN», \_)  и  cars("Petrov", "Kursk", "LADA", "GREEN", 100000) | Результат сравнения термов true  SURNAME = "Petrov"  CITY = "Kursk"  Переход к следующему утверждению в теле правила (прямой ход) |
| 6 | Следующее утверждение  info(SURNAME, PHONE, address(CITY, \_, \_, \_)) | Заносится в стек  info(SURNAME, PHONE, address(CITY, \_, \_, \_)) |
| 7 | Раскрывается правило  info(SURNAME, PHONE, address(CITY, \_, \_, \_))  ->  info("Petrov", PHONE, address("Kursk", \_, \_, \_))  По функтору и списку аргументов ищется определение отношения (с самого начала) | Определение отношения найдено |
| 8 | Унификация info("Petrov", PHONE, address("Kursk", \_, \_, \_)) и  info("Petrov", "88885353535", address("Moscow", "Saykin street", 10, 143)) | Результат сравнения термов false, прямой ход |
| 9 | Унификация info("Petrov", PHONE, address("Kursk", \_, \_, \_)) и  info("Petrov", "81111111111", address("Kursk", "Non street", 1, 10) | Результат сравнения термов true  PHONE = "81111111111"  Анонимные переменные не связываются со значением  Переход к следующему утверждению в теле правила (прямой ход) |
| 10 | Следующее утверждение  banks(SURNAME, CITY, BANK, \_, \_) | Заносится в стек  banks(SURNAME, CITY, BANK, \_, \_) |
| 11 | Раскрывается правило  banks(SURNAME, CITY, BANK, \_, \_)  ->  banks("Petrov", "Kursk", BANK, \_, \_)  По функтору и списку аргументов ищется определение отношения | Определение отношения найдено |
| 12 | Унификация banks("Petrov", "Kursk", BANK, \_, \_)  и  banks("Petrov", "Moscow", "Sberbank", 356, 50000) | Результат сравнения термов false, прямой ход |
| 13 | Унификация banks("Petrov", "Kursk", BANK, \_, \_)  и  banks("Petrov", "Kursk", "Tinkoff", 357, 9000) | Результат сравнения термов true  BANK = "Tinkoff"  Вывод результата, переход к следующей строке |
| 14 | Унификация banks("Petrov", "Kursk", BANK, \_, \_)  и  banks("Krot", "Moscow", "Rocket-bank", 20, 2000). | Результат сравнения термов false. В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся banks(SURNAME, CITY, BANK, \_, \_) |
| 15 | Унификация info("Petrov", PHONE, address("Kursk", \_, \_, \_)) и  info("Krot", "89995353566", address("Moscow", "Nwot street", 5, 60)) | Результат сравнения термов false. В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся info(SURNAME, PHONE, address(CITY, \_, \_, \_)) |
| 16 | Унификация cars(SURNAME, CITY, «LADA», «GREEN», \_) и  cars("Krotova", "Moscow", "Merc", "BLUE", 100000) | Результат сравнения термов false. В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся cars(SURNAME, CITY, «LADA», «GREEN», \_) |
| 17 | Стек пуст | Стек пуст, завершение программы |

1. У машины несколько владельцев:

|  |
| --- |
| goal  find\_by\_car(SURNAME, CITY, PHONE, "TOYOTA", "RED", BANK). |



Порядок формирования ответа для примера 2:

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат |
| 0 | find\_by\_car(SURNAME, CITY, PHONE, MARKA, COLOR, BANK)  =  find\_by\_car(SURNAME, CITY, PHONE, "TOYOTA", "RED", BANK) | Заносится в стек find\_by\_car(SURNAME, CITY, PHONE, "TOYOTA", "RED", BANK) |
| 1 | Раскрывается правило, то есть надо доказать каждое утверждение в теле правила (последовательно)  cars(SURNAME, CITY, MARKA, COLOR, PRICE)  =  cars(SURNAME, CITY, "TOYOTA", "RED", \_) | Прямой ход  Заносится в стек  cars(SURNAME, CITY, "TOYOTA", "RED", \_) |
| 2 | cars(SURNAME, CITY, "TOYOTA", "RED", \_)  По функтору и списку аргументов ищется определение отношения  cars("Petrov", "Moscow", "TOYOTA", "RED", 256000) | Определение отношения найдено |
| 3 | Унификация cars(SURNAME, CITY, "TOYOTA", "RED", \_) и cars("Petrov", "Moscow", "TOYOTA", "RED", 256000) | Результат сравнения термов true  SURNAME = "Petrov"  CITY = "Moscow"  Переход к следующему утверждению в теле правила (прямой ход) |
|  | Следующее утверждение  info(SURNAME, PHONE, address(CITY, \_, \_, \_)) |  |
| 4 | Раскрывается правило  info(SURNAME, PHONE, address(CITY, \_, \_, \_))  ->  info("Petrov", PHONE, address("Moscow", \_, \_, \_))  По функтору и списку аргументов ищется определение отношения (с самого начала) | Определение отношения найдено  Заносится в стек  info("Petrov", PHONE, address("Moscow", \_, \_, \_)) |
| 5 | Унификация info("Petrov", PHONE, address("Moscow", \_, \_, \_)) и  info("Petrov", "88885353535", address("Moscow", "Saykin street", 10, 143)) | Результат сравнения термов true  PHONE = "88885353535"  Анонимные переменные не связываются со значением  Переход к следующему утверждению в теле правила (прямой ход) |
|  | Следующее утверждение  banks(SURNAME, CITY, BANK, \_, \_) |  |
| 6 | Раскрывается правило  banks(SURNAME, CITY, BANK, \_, \_)  ->  banks("Petrov", "Moscow", BANK, \_, \_)  По функтору и списку аргументов ищется определение отношения | Определение отношения найдено  Заносится в стек  banks("Petrov", "Moscow", BANK, \_, \_) |
| 7 | Унификация banks("Petrov", "Moscow", BANK, \_, \_)  и  banks("Petrov", "Moscow", "Sberbank", 356, 50000) | Результат сравнения термов true  BANK = "Sberbank"  Вывод результата, переход к следующей строке |
| 8 | Унификация banks("Petrov", "Moscow", BANK, \_, \_)  и  banks("Krot", "Moscow", "Rocket-bank", 20, 2000). | Результат сравнения термов false. В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся banks("Petrov", "Moscow", BANK, \_, \_) |
|  | Следующее утверждение  info(SURNAME, PHONE, address(CITY, \_, \_, \_)) |  |
| 9 | Унификация info("Petrov", PHONE, address("Moscow", \_, \_, \_)) и  info("Petrov", "81111111111", address("Kursk", "Non street", 1, 10)) | Результат сравнения термов false. Прямой ход |
| 10 | Унификация info("Petrov", PHONE, address("Moscow", \_, \_, \_)) и  info("Krot", "89995353566", address("Moscow", "Nwot street", 5, 60)) | Результат сравнения термов false. В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся info("Petrov", PHONE, address("Moscow", \_, \_, \_)) |
|  | Следующее утверждение cars(SURNAME, CITY, "TOYOTA", "RED", PRICE) |  |
| 11 | Унификация cars(SURNAME, CITY, "TOYOTA", "RED", \_) и  cars("Krot", "Moscow", "TOYOTA", "RED", 256000) | Результат сравнения термов true  SURNAME = "Krot"  CITY = "Moscow"  Переход к следующему утверждению в теле правила (прямой ход) |
|  | Следующее утверждение  info(SURNAME, PHONE, address(CITY, \_, \_, \_)) |  |
| 12 | Раскрывается правило  info(SURNAME, PHONE, address(CITY, \_, \_, \_))  ->  info("Krot", PHONE, address("Moscow", \_, \_, \_))  По функтору и списку аргументов ищется определение отношения (с самого начала) | Определение отношения найдено  Заносится в стек  info("Krot", PHONE, address("Moscow", \_, \_, \_)) |
| 13 | Унификация info("Krot", PHONE, address("Moscow", \_, \_, \_))  И info("Petrov", "88885353535", address("Moscow", "Saykin street", 10, 143)) | Результат сравнения термов false, прямой ход |
| 14 | Унификация info("Krot", PHONE, address("Moscow", \_, \_, \_))  И info("Petrov", "81111111111", address("Kursk", "Non street", 1, 10)) | Результат сравнения термов false, прямой ход |
| 15 | Унификация info("Krot", PHONE, address("Moscow", \_, \_, \_))  И info("Krot", "89995353566", address("Moscow", "Nwot street", 5, 60)) | Результат сравнения термов true  PHONE = "89995353566"  Переход к следующему утверждению в теле правила (прямой ход) |
|  | Следующее утверждение  banks(SURNAME, CITY, BANK, \_, \_) |  |
| 16 | Раскрывается правило  banks(SURNAME, CITY, BANK, \_, \_)  ->  banks("Krot", "Moscow", BANK, \_, \_)  По функтору и списку аргументов ищется определение отношения | Определение отношения найдено  Заносится в стек  banks("Krot", "Moscow", BANK, \_, \_) |
| 17 | Унификация banks("Krot", "Moscow", BANK, \_, \_)  И banks("Petrov", "Moscow", "Sberbank", 356, 50000) | Результат сравнения термов false, прямой ход |
|  | Унификация banks("Krot", "Moscow", BANK, \_, \_)  И banks("Petrov", "Kursk", "Tinkoff", 357, 9000) | Результат сравнения термов false, прямой ход |
| 18 | Унификация banks("Krot", "Moscow", BANK, \_, \_)  И banks("Krot", "Moscow", "Rocket-bank", 20, 2000) | Результат сравнения термов true  BANK = "Rocket-Bank"  Вывод результата, переход к следующей строке.  В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся banks("Krot", "Moscow", BANK, \_, \_) |
| 19 |  | В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся info("Krot", PHONE, address("Moscow", \_, \_, \_)) |
|  | Следующее утверждение cars(SURNAME, CITY, "TOYOTA", "RED", \_) |  |
| 20 | Унификация cars(SURNAME, CITY, "TOYOTA", "RED", \_) и cars("Petrov", "Kursk", "LADA", "GREEN", 100000) | Результат сравнения термов false, прямой ход |
| 21 | Унификация cars(SURNAME, CITY, "TOYOTA", "RED", \_) и cars("Krotova", "Moscow", "Merc", "BLUE", 100000) | Результат сравнения термов false, прямой ход. В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся cars(SURNAME, CITY, "TOYOTA", "RED", \_) |
| 22 | Стек пуст | Стек пуст, завершение цикла. |

1. У машины нет владельца (он отсутствует в базе):

|  |
| --- |
| find\_by\_car(SURNAME, CITY, PHONE, "Merc", "BLUE", BANK). |



Порядок формирования ответа для примера 1:

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат |
| 0 | find\_by\_car(SURNAME, CITY, PHONE, MARKA, COLOR, BANK)  =  find\_by\_car(SURNAME, CITY, PHONE, "Merc", "BLUE", BANK) | Заносится в стек find\_by\_car(SURNAME, CITY, PHONE, "Merc", "BLUE", BANK) |
| 1 | Раскрывается правило, то есть надо доказать каждое утверждение в теле правила (последовательно)  cars(SURNAME, CITY, MARKA, COLOR, PRICE)  =  cars(SURNAME, CITY, "Merc", "BLUE", \_) | Прямой ходб  Заносится в стек  cars(SURNAME, CITY, "Merc", "BLUE", \_) |
| 2 | cars(SURNAME, CITY, "Merc", "BLUE", \_)  По функтору и списку аргументов ищется определение отношения  cars("Petrov", "Moscow", "TOYOTA", "RED", 256000) | Определение отношения найдено |
| 3 | Унификация cars(SURNAME, CITY, "Merc", "BLUE", \_) и cars("Petrov", "Moscow", "TOYOTA", "RED", 256000) | Результат сравнения термов false, прямой ход |
| 4 | Унификация cars(SURNAME, CITY, "Merc", "BLUE", \_) и  cars("Krot", "Moscow", "TOYOTA", "RED", 256000) | Результат сравнения термов false, прямой ход |
| 5 | Унификация cars(SURNAME, CITY, "Merc", "BLUE", \_)  и  cars("Petrov", "Kursk", "LADA", "GREEN", 100000) | Результат сравнения термов false, прямой ход. |
| 6 | Унификация cars(SURNAME, CITY, "Merc", "BLUE", \_)  и  cars("Krotova", "Moscow", "Merc", "BLUE", 100000) | Результат сравнения термов true  SURNAME = "Krotova"  CITY = "Moscow"  Переход к следующему утверждению в теле правила (прямой ход) |
|  | Следующее утверждение info(SURNAME, PHONE, address(CITY, \_, \_, \_)) |  |
| 7 | info(SURNAME, PHONE, address(CITY, \_, \_, \_))  =  info("Krotova", PHONE, address("Moscow",, \_, \_, \_)) | Заносится в стек info("Krotova", PHONE, address("Moscow",, \_, \_, \_)) |
| 8 | Унификация info("Krotova", PHONE, address("Moscow",, \_, \_, \_))  И info("Petrov", "88885353535", address("Moscow", "Saykin street", 10, 143)) | Результат сравнения термов false, прямой ход. |
| 9 | Унификация info("Krotova", PHONE, address("Moscow",, \_, \_, \_))  И info("Petrov", "81111111111", address("Kursk", "Non street", 1, 10)) | Результат сравнения термов false, прямой ход. |
| 10 | Унификация info("Krotova", PHONE, address("Moscow",, \_, \_, \_))  И info("Krot", "89995353566", address("Moscow", "Nwot street", 5, 60)) | Результат сравнения термов false, прямой ход. В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся info("Krotova", PHONE, address("Moscow",, \_, \_, \_)) |
|  | Следующее утверждение cars(SURNAME, CITY, "Merc", "BLUE", \_) | В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся cars(SURNAME, CITY, "Merc", "BLUE", \_) |
| 11 | Стек пуст | Стек пуст, завершение программы |

**Задание 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?

**АВТОМОБИЛИ – СПРАВОЧНИК - БАНКИ**

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат |
| 0 | find\_by\_car(SURNAME, CITY, PHONE, MARKA, COLOR, BANK)  =  find\_by\_car(SURNAME, CITY, PHONE, "TOYOTA", "RED", BANK) | Заносится в стек find\_by\_car(SURNAME, CITY, PHONE, "TOYOTA", "RED", BANK) |
| 1 | Раскрывается правило, то есть надо доказать каждое утверждение в теле правила (последовательно)  cars(SURNAME, CITY, MARKA, COLOR, \_)  =  cars(SURNAME, CITY, "TOYOTA", "RED", \_) | Прямой ход  Заносится в стек  cars(SURNAME, CITY, "TOYOTA", "RED", \_) |
| 2 | cars(SURNAME, CITY, "TOYOTA", "RED", \_)  По функтору и списку аргументов ищется определение отношения  cars("Petrov", "Moscow", "TOYOTA", "RED", 256000) | Определение отношения найдено |
| 3 | Унификация cars(SURNAME, CITY, "TOYOTA", "RED", \_) и cars("Petrov", "Moscow", "TOYOTA", "RED", 256000) | Результат сравнения термов true  SURNAME = "Petrov"  CITY = "Moscow"  Переход к следующему утверждению в теле правила (прямой ход) |
| 4 | Следующее утверждение  info(SURNAME, PHONE, address(CITY, \_, \_, \_)) |  |
| 5 | Раскрывается правило  info(SURNAME, PHONE, address(CITY, \_, \_, \_))  ->  info("Petrov", PHONE, address("Moscow", \_, \_, \_))  По функтору и списку аргументов ищется определение отношения (с самого начала) | Определение отношения найдено  Заносится в стек  info("Petrov", PHONE, address("Moscow", \_, \_, \_)) |
| 6 | Унификация info("Petrov", PHONE, address("Moscow", \_, \_, \_)) и  info("Petrov", "88885353535", address("Moscow", "Saykin street", 10, 143)) | Результат сравнения термов true  PHONE = "88885353535"  Анонимные переменные не связываются со значением  Переход к следующему утверждению в теле правила (прямой ход) |
|  | Следующее утверждение  banks(SURNAME, CITY, BANK, \_, \_) |  |
| 7 | Раскрывается правило  banks(SURNAME, CITY, BANK, \_, \_)  ->  banks("Petrov", "Moscow", BANK, \_, \_)  По функтору и списку аргументов ищется определение отношения | Определение отношения найдено  Заносится в стек  banks("Petrov", "Moscow", BANK, \_, \_) |
| 8 | Унификация banks("Petrov", "Moscow", BANK, \_, \_)  и  banks("Petrov", "Moscow", "Sberbank", 356, 50000) | Результат сравнения термов true  BANK = "Sberbank"  Вывод результата, переход к следующей строке |
| 9 | Унификация banks("Petrov", "Moscow", BANK, \_, \_)  и  banks("Petrov", "Kursk", "Tinkoff", 357, 9000) | Результат сравнения термов false. Прямой ход |
| 10 | Унификация banks("Petrov", "Moscow", BANK, \_, \_)  и  banks("Krot", "Moscow", "Rocket-bank", 20, 2000). | Результат сравнения термов false. В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся banks("Petrov", "Moscow", BANK, \_, \_) |
| 11 | Следующее утверждение  info(SURNAME, PHONE, address(CITY, \_, \_, \_)) |  |
| 12 | Унификация info("Petrov", PHONE, address("Moscow", \_, \_, \_)) и  info("Petrov", "81111111111", address("Kursk", "Non street", 1, 10)) | Результат сравнения термов false. Прямой ход |
| 13 | Унификация info("Petrov", PHONE, address("Moscow", \_, \_, \_)) и  info("Krot", "89995353566", address("Moscow", "Nwot street", 5, 60)) | Результат сравнения термов false. В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся info("Petrov", PHONE, address("Moscow", \_, \_, \_)) |
|  | Следующее утверждение cars(SURNAME, CITY, "TOYOTA", "RED", PRICE) |  |
| 14 | Унификация cars(SURNAME, CITY, "TOYOTA", "RED", \_) и  cars("Krot", "Moscow", "TOYOTA", "RED", 256000) | Результат сравнения термов true  SURNAME = "Krot"  CITY = "Moscow"  Переход к следующему утверждению в теле правила (прямой ход) |
|  | Следующее утверждение  info(SURNAME, PHONE, address(CITY, \_, \_, \_)) |  |
| 15 | Раскрывается правило  info(SURNAME, PHONE, address(CITY, \_, \_, \_))  ->  info("Krot", PHONE, address("Moscow", \_, \_, \_))  По функтору и списку аргументов ищется определение отношения (с самого начала) | Определение отношения найдено  Заносится в стек  info("Krot", PHONE, address("Moscow", \_, \_, \_)) |
| 16 | Унификация info("Krot", PHONE, address("Moscow", \_, \_, \_))  И info("Petrov", "88885353535", address("Moscow", "Saykin street", 10, 143)) | Результат сравнения термов false, прямой ход |
| 17 | Унификация info("Krot", PHONE, address("Moscow", \_, \_, \_))  И info("Petrov", "81111111111", address("Kursk", "Non street", 1, 10)) | Результат сравнения термов false, прямой ход |
| 18 | Унификация info("Krot", PHONE, address("Moscow", \_, \_, \_))  И info("Krot", "89995353566", address("Moscow", "Nwot street", 5, 60)) | Результат сравнения термов true  PHONE = "89995353566"  Переход к следующему утверждению в теле правила (прямой ход) |
|  | Следующее утверждение  banks(SURNAME, CITY, BANK, \_, \_) |  |
| 19 | Раскрывается правило  banks(SURNAME, CITY, BANK, \_, \_)  ->  banks("Krot", "Moscow", BANK, \_, \_)  По функтору и списку аргументов ищется определение отношения | Определение отношения найдено  Заносится в стек  banks("Krot", "Moscow", BANK, \_, \_) |
| 20 | Унификация banks("Krot", "Moscow", BANK, \_, \_)  И banks("Petrov", "Moscow", "Sberbank", 356, 50000) | Результат сравнения термов false, прямой ход |
| 21 | Унификация banks("Krot", "Moscow", BANK, \_, \_)  И banks("Petrov", "Kursk", "Tinkoff", 357, 9000) | Результат сравнения термов false, прямой ход |
| 22 | Унификация banks("Krot", "Moscow", BANK, \_, \_)  И banks("Krot", "Moscow", "Rocket-bank", 20, 2000) | Результат сравнения термов true  BANK = "Rocket-Bank"  Вывод результата, переход к следующей строке.  В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся banks("Krot", "Moscow", BANK, \_, \_) |
| 23 |  | В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся info("Krot", PHONE, address("Moscow", \_, \_, \_)) |
|  | Следующее утверждение cars(SURNAME, CITY, "TOYOTA", "RED", \_) |  |
| 24 | Унификация cars(SURNAME, CITY, "TOYOTA", "RED", \_) и cars("Petrov", "Kursk", "LADA", "GREEN", 100000) | Результат сравнения термов false, прямой ход |
| 25 | Унификация cars(SURNAME, CITY, "TOYOTA", "RED", \_) и cars("Krotova", "Moscow", "Merc", "BLUE", 100000) | Результат сравнения термов false, прямой ход. В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся cars(SURNAME, CITY, "TOYOTA", "RED", \_) |
| 26 | Стек пуст | Стек пуст, завершение цикла. |

**СПРАВОЧНИК – АВТОМОБИЛИ – БАНКИ**

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат |
| 0 | find\_by\_car(SURNAME, CITY, PHONE, MARKA, COLOR, BANK)  =  find\_by\_car(SURNAME, CITY, PHONE, "TOYOTA", "RED", BANK) | Заносится в стек find\_by\_car(SURNAME, CITY, PHONE, "TOYOTA", "RED", BANK) |
| 1 | Раскрывается правило, то есть надо доказать каждое утверждение в теле правила (последовательно)  info(SURNAME, PHONE, home) =  info(SURNAME, PHONE, address(CITY, \_, \_, \_)) | Прямой ход  Заносится в стек  info(SURNAME, PHONE, address(CITY, \_, \_, \_)) |
| 2 | Унификация info(SURNAME, PHONE, address(CITY, \_, \_, \_))  и info("Petrov", "88885353535", address("Moscow", "Saykin street", 10, 143)) | Результат сравнения термов true  SURNAME = "Petrov"  CITY = "Moscow"  PHONE = "88885353535"  Анонимные переменные не связываются со значением  Переход к следующему утверждению в теле правила (прямой ход) |
|  | Следующее утверждение  cars(SURNAME, CITY, MARKA, COLOR, \_) |  |
| 3 | Раскрывается правило  cars(SURNAME, CITY, MARKA, COLOR, \_)  =  cars(«Petrov», «Moscow», "TOYOTA", "RED", \_)  По функтору и списку аргументов ищется определение отношения | Определение отношения найдено  Заносится в стек  cars(«Petrov», «Moscow», "TOYOTA", "RED", \_) |
| 4 | Унификация cars(«Petrov», «Moscow», "TOYOTA", "RED", \_)  и  cars("Petrov", "Moscow", "TOYOTA", "RED", 256000) | Результат сравнения термов true  Анонимные переменные не связываются со значением  Переход к следующему утверждению в теле правила (прямой ход) |
|  | Следующее утверждение  banks(SURNAME, CITY, BANK, \_, \_) |  |
| 5 | Раскрывается правило  banks(SURNAME, CITY, BANK, \_, \_)  ->  banks("Petrov", "Moscow", BANK, \_, \_)  По функтору и списку аргументов ищется определение отношения | Определение отношения найдено  Заносится в стек  banks("Petrov", "Moscow", BANK, \_, \_) |
| 6 | Унификация banks("Petrov", "Moscow", BANK, \_, \_)  и  banks("Petrov", "Moscow", "Sberbank", 356, 50000) | Результат сравнения термов true  BANK = "Sberbank"  Вывод результата, переход к следующей строке |
| 7 | Унификация banks("Petrov", "Moscow", BANK, \_, \_)  и  banks("Krot", "Moscow", "Rocket-bank", 20, 2000). | Результат сравнения термов false. В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся banks("Petrov", "Moscow", BANK, \_, \_) |
|  | Следующее утверждение  cars(«Petrov», «Moscow», "TOYOTA", "RED", \_) |  |
| 8 | Унификация cars(«Petrov», «Moscow», "TOYOTA", "RED", \_)  и  cars("Krot", "Moscow", "TOYOTA", "RED", 256000) | Результат сравнения термов false. Прямой ход |
|  | Унификация cars(«Petrov», «Moscow», "TOYOTA", "RED", \_)  и  cars("Petrov", "Kursk", "LADA", "GREEN", 100000) |  |
| 9 | Унификация cars(«Petrov», «Moscow», "TOYOTA", "RED", \_)  и  cars("Krotova", "Moscow", "Merc", "BLUE", 100000) | Результат сравнения термов false. В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся cars(«Petrov», «Moscow», "TOYOTA", "RED", \_) |
|  | Следующее утверждение  info(SURNAME, PHONE, address(CITY, \_, \_, \_)) |  |
| 10 | Унификация info(SURNAME, PHONE, address(CITY, \_, \_, \_)) и  info("Petrov", "81111111111", address("Kursk", "Non street", 1, 10)) | Результат сравнения термов true  SURNAME = "Petrov"  CITY = "Kursk"  Переход к следующему утверждению в теле правила (прямой ход) |
|  | Следующее утверждение  cars(SURNAME, CITY, "TOYOTA", "RED", \_) |  |
| 11 | Раскрывается правило  cars(SURNAME, CITY, "TOYOTA", "RED", \_)  ->  cars(«Petrov», «Kursk», "TOYOTA", "RED", \_)  По функтору и списку аргументов ищется определение отношения | Определение отношения найдено  Заносится в стек  cars(«Petrov», «Kursk», "TOYOTA", "RED", \_) |
| 12 | Унификация cars(«Petrov», «Kursk», "TOYOTA", "RED", \_)  И cars("Petrov", "Moscow", "TOYOTA", "RED", 256000) | Результат сравнения термов false, прямой ход |
| 13 | Унификация cars(«Petrov», «Kursk», "TOYOTA", "RED", \_)  И cars("Krot", "Moscow", "TOYOTA", "RED", 256000) | Результат сравнения термов false, прямой ход |
| 14 | Унификация cars(«Petrov», «Kursk», "TOYOTA", "RED", \_)  И cars("Petrov", "Kursk", "LADA", "GREEN", 100000) | Результат сравнения термов false, прямой ход |
| 15 | Унификация cars(«Petrov», «Kursk», "TOYOTA", "RED", \_)  И cars("Krotova", "Moscow", "Merc", "BLUE", 100000) | Результат сравнения термов false, прямой ход  Вывод результата, переход к следующей строке.  В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся cars(«Petrov», «Kursk», "TOYOTA", "RED", \_) |
|  | Следующее утверждение  info(SURNAME, PHONE, address(CITY, \_, \_, \_)) |  |
| 16 | Унификация  info(SURNAME, PHONE, address(CITY, \_, \_, \_)) и info("Krot", "89995353566", address("Moscow", "Nwot street", 5, 60)) | Результат сравнения термов true  SURNAME = "Krot"  PHONE = "89995353566"  CITY = "Moscow"  Переход к следующему утверждению в теле правила (прямой ход) |
|  | Следующее утверждение  cars(SURNAME, CITY, "TOYOTA", "RED", \_) |  |
| 17 | Раскрывается правило  cars(SURNAME, CITY, "TOYOTA", "RED", \_)  ->  cars(«Krot», «Moscow», "TOYOTA", "RED", \_)  По функтору и списку аргументов ищется определение отношения | Определение отношения найдено  Заносится в стек  cars(«Krot», «Moscow», "TOYOTA", "RED", \_) |
| 18 | Унификация cars(«Krot», «Moscow», "TOYOTA", "RED", \_)  И cars("Petrov", "Moscow", "TOYOTA", "RED", 256000) | Результат сравнения термов false, прямой ход |
| 19 | Унификация cars(«Krot», «Moscow», "TOYOTA", "RED", \_)  И cars("Krot", "Moscow", "TOYOTA", "RED", 256000) | Результат сравнения термов true  Переход к следующему утверждению в теле правила (прямой ход) |
|  | Следующее утверждение banks(SURNAME, CITY, BANK, \_, \_) |  |
| 20 | Раскрывается правило  banks(SURNAME, CITY, BANK, \_, \_)  ->  banks("Krot", "Moscow", BANK, \_, \_)  По функтору и списку аргументов ищется определение отношения | Определение отношения найдено  Заносится в стек  banks("Krot", "Moscow", BANK, \_, \_) |
| 21 | Унификация banks("Krot", "Moscow", BANK, \_, \_)  И banks("Petrov", "Moscow", "Sberbank", 356, 50000) | Результат сравнения термов false, прямой ход |
| 22 | Унификация banks("Krot", "Moscow", BANK, \_, \_)  И banks("Petrov", "Kursk", "Tinkoff", 357, 9000) | Результат сравнения термов false, прямой ход |
| 23 | Унификация banks("Krot", "Moscow", BANK, \_, \_)  И banks("Krot", "Moscow", "Rocket-bank", 20, 2000) | Результат сравнения термов true  BANK = "Rocket-Bank"  Вывод результата, переход к следующей строке.  В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся banks("Krot", "Moscow", BANK, \_, \_) |
|  | Следующее утверждение cars(SURNAME, CITY, "TOYOTA", "RED", \_) |  |
| 24 | Унификация cars(«Krot», «Moscow», "TOYOTA", "RED", \_)  и cars("Petrov", "Kursk", "LADA", "GREEN", 100000) | Результат сравнения термов false, прямой ход |
| 25 | Унификация cars(SURNAME, CITY, "TOYOTA", "RED", \_) и cars("Krotova", "Moscow", "Merc", "BLUE", 100000) | Результат сравнения термов false, прямой ход. В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся cars(«Krot», «Moscow», "TOYOTA", "RED", \_)  В базе знаний больше нет утверждений с заданным именем  Из стека достаётся info(SURNAME, PHONE, address(CITY, \_, \_, \_)) |
| 26 | Стек пуст | Стек пуст, завершение цикла. |

**Вывод:** Prolog просматривает все предложения последовательно, сравнивая текущий терм со всеми в разделе clauses. Поэтому не зависимо от последовательности предикатов, объём всегда будет одинаков.

**Порядок работы алгоритма унификации** вопроса и подходящего заголовка правила

Унификация терма find\_by\_car(SURNAME, CITY, PHONE, MARKA, COLOR, BANK) и find\_by\_car(SURNAME, CITY, PHONE, "TOYOTA", "RED", BANK)

|  |  |  |  |
| --- | --- | --- | --- |
| Шаг унификации | Результирующая ячейка | Рабочее поле | Стек |
| 0 |  |  | find\_by\_car(SURNAME, CITY, PHONE, MARKA, COLOR, BANK)  =  find\_by\_car(SURNAME, CITY, PHONE, "TOYOTA", "RED", BANK) |
| 1 |  | find\_by\_car(SURNAME, CITY, PHONE, MARKA, COLOR, BANK)  = find\_by\_car(SURNAME, CITY, PHONE, "TOYOTA", "RED", BANK) | MARKA = "TOYOTA"  COLOR = "RED" |
| 2 | MARKA = "TOYOTA" | MARKA = "TOYOTA" | COLOR = "RED" |
| 3 | COLOR = "RED" | COLOR = "RED" | Стек пуст |
| 4 |  | Очистка поля |  |

**Порядок работы алгоритма унификации** вопроса и подходящего заголовка правила

Унификация терма cars(SURNAME, CITY, "TOYOTA", "RED", PRICE) и cars("Petrov", "Moscow", "TOYOTA", "RED", 256000)

|  |  |  |  |
| --- | --- | --- | --- |
| Шаг унификации | Результирующая ячейка | Рабочее поле | Стек |
| 0 |  |  | cars(SURNAME, CITY, "TOYOTA", "RED", PRICE)  =  cars("Petrov", "Moscow", "TOYOTA", "RED", 256000) |
| 1 |  | cars(SURNAME, CITY, "TOYOTA", "RED", \_)  =  cars("Petrov", "Moscow", "TOYOTA", "RED", 256000) | SURNAME = "Petrov"  CITY="Moscow"  "TOYOTA"="TOYOTA"  "RED"="RED" |
| 2 | SURNAME = "Petrov" | SURNAME = "Petrov" | CITY="Moscow"  "TOYOTA"="TOYOTA"  "RED"="RED" |
| 3 | SURNAME = "Petrov"  CITY="Moscow" | CITY="Moscow" | "TOYOTA"="TOYOTA"  "RED"="RED" |
| 4 | SURNAME = "Petrov"  CITY="Moscow" | "TOYOTA"="TOYOTA" | "RED"="RED" |
| 5 | SURNAME = "Petrov"  CITY="Moscow" | "RED"="RED" | Стек пуст |
| 6 |  | Очистка поля | Стек пуст |

**Теоретическая часть.**

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

Знания о предметной области выражаются на языке Пролог в виде предложений, называемых утверждениями (clauses).

1. **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

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

Три. «Телефонный справочник», «Автомобили», «Вкладчики банков»

1. **Что такое пример терма, это частный случай терма, пример? Как строитс япример?**

Пусть *θ* = {*X*1 = *t*1*,X*2 = *t*2*,...,Xn* = *tn*} – подстановка, тогда результат применения подстановки к терму обозначается: *Aθ*. Применение подстановки заключается в замене каждого вхождения переменной *Xi* на соответствующий терм. Терм *B* называется примером терма *A*, если существует такая подстановка *θ*, что *B* = *Aθ*.

В процессе выполнения программы – система, используя встроенный алгоритм унификации, пытается обосновать возможность истинности вопроса, строя подстановки и примеры термов (вопроса и формулировки знания), используя базу знаний.

1. **Что такое наиболее общий пример?**

Терм *C* называется общим примером термов *A* и *B*, если существуют такие подстановки *θ*1 и *θ*2, что *C* = *Aθ*1 и *C* = *Bθ*2

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

Унификация двух термов – это основной шаг доказательства. В процессе работы система выполняет большое число унификаций. Унификация – операция, которая позволяет формализовать процесс логического вывода.

С помощью алгоритма унификации происходит двунаправленная передача параметров процедурам. Например, из внешнего мира в программу для дальнейшего использования или из программы во внешний мир – значения интересующего нас параметра.

1. **В каком случае запускается механизм отката?**

Откат дает возможность получить много решений в одном вопросе к программе.

Во всех точках программы, где существуют альтернативы, в стек заносятся точки возврата.

Если впоследствии окажется, что выбранный вариант не приводит к успеху, то осуществляется откат к последней из имеющихся в стеке точек программы, где был выбран один из альтернативных вариантов.

Выбирается очередной вариант, программа продолжает свою работу. Если все варианты в точке уже были использованы, то регистрируется неудачное завершение и осуществляется переход на предыдущую точку возврата, если такая есть.

При откате все связанные переменные, которые были означены после этой точки, опять освобождаются.

**8. Виды и назначение переменных в Prolog. Примеры из задания. Почему использованы те или другие переменные (примеры из задания)?**   
При поступлении вопроса с переменной в Пролог-систему. Например:   
info ("Mark" , X).   
X – переменная, входящая в вопрос, изначально является неконкретизированной. Пролог просматривает базу данных в поисках факта, сопоставимого с вопросом. Если неконкретизированная переменная появляется в качестве одного из аргументов, то Пролог считает, что такой аргумент сопоставим с любым другим аргументом, находящимся в том же факте. При обнаружении такого факта переменная X становится конкретизированной, обозначая объект, являющийся вторым аргументом найденного факта.   
info(SURNAME, PHONE, address(CITY, \_, \_, \_))  
Это относится только к именованным переменным. Анонимные переменные не могут быть связаны со значением. Используются, если нас не интересует значение данного параметра.  
Если составные термы, факты, правила и вопросы не содержат переменных, то они называются основными. Составные термы, факты, правила и вопросы в момент фиксации в программе могут содержать переменные, тогда они называются неосновными.