|  |  |
| --- | --- |
| 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 № 15**

**Структура программы на Prolog и ее реализация**

|  |  |
| --- | --- |
| **Студент: Луговой Д.М.**  **Группа: ИУ7-61Б**  **Преподаватель:** Толпинская Н. Б. |  |

Москва.

2020 г.

**Цель работы –** изучить структуру, особенности и принципы оформления программы, и способ выполнения программы на Prolog.

**Задачи работы**

Приобрести навыки декларативного описания предметной области с использованием фактов, правил и некоторых специальных разделов программы.

Изучить порядок использования фактов и правил в программе на Prolog, принципы и особенности сопоставления и отождествления термов, на основе механизма унификации.

**Задание**

Создать базу знаний **«Собственники»**, дополнив 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.,

знаниями о дополнительной **собственности** владельца. **Преобразовать** знания об автомобиле к форме знаний о собственности.

Вид собственности (кроме автомобиля):

* **Строение, стоимость** и другие его характеристики;
* **Участок, стоимость** и другие его характеристики;
* **Водный\_транспорт, стоимость** и другие его характеристики.

Описать и использовать вариантный домен: **Собственность**. Владелец может иметь, но **только один** объект **каждого вида собственности (**это касается и **автомобиля)**, или не иметь некоторых видов собственности.

Используя **конъюнктивное правило и разные формы** задания **одного вопроса (пояснять** для какого №задания – какой вопрос), обеспечить возможность поиска:

1. Названий всех объектов собственности заданного субъекта,
2. Названий и стоимости всех объектов собственности заданного субъекта,
3. \* Разработать правило, позволяющее найти суммарную стоимость всех объектов собственности заданного субъекта.

Для 2-го пункт и **одной** фамилии **составить таблицу**, отражающую конкретный порядок работы системы, с объяснениями порядка работы и особенностей использования доменов (указать конкретные Т1 и Т2 и полную подстановку на каждом шаге)

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат (к чему приводит?) |
| 1… | -попытка унификации: Т1=Т2 …  -результат: Успех и подстановка,  или Нет | Комментарий, вывод… |
| 2 | … | … |

При желании, можно усложнить свою базу знаний, введя варианты: **строение: (Дом, офис, торговый центр), участок: (садовый, территория под застройку, территория под агро-работы), Водный\_транспорт: варианты названий.**

**Ответы на вопросы**

* **В каком фрагменте программы сформулировано знание? Это знание о чем на формальном уровне?**

Знания сформулированы в заголовках правил, правила находятся в разделе описания предложений базы знаний – CLAUSES. Заголовок как составной терм 𝑓(𝑡1, 𝑡2, . . . ,𝑡𝑚) содержит знание об отношении между его аргументами 𝑡1, 𝑡2, . . . ,𝑡𝑚.

* **Что содержит тело правила?**

Тело правила содержит условие истинности знания, содержащегося в заголовке правила.

* **Что дает использование переменных при формулировании знаний? В чем отличие формулировки знания с помощью термов с одинаковой арностью при использовании одной переменной и при использовании нескольких переменных?**

Переменные предназначены для обозначения некоторого неизвестного объекта предметной области, знания сформулированных с их помощью получаются более общими, чем правила, сформулированные только с помощью констант, так как переменные могут быть конкретизированы разными объектами. Знание, сформулированное с использованием одной переменной, является менее общим, чем знания, сформулированное с использованием нескольких переменных.

* **С каким квантором переменные входят в правило, в каких пределах переменная уникальна?**

В правила переменные входят с квантором всеобщности (∀, «для любых»). Именованные переменные уникальны в пределах одного предложения, любая анонимная переменная уникальна.

* **Какова семантика (смысл) предложений раздела DOMAINS? Когда, где и с какой целью используется это описание?**

Предложения в разделе DOMAINS служат для определения доменов. Домены объявляются для объектов, которые из-за своей природы или структуры не могут быть описаны стандартными доменами. Это описание позволяет создавать домены сложной структуры и различать объекты, внешне одинаковые, но имеющие разные природу, несущие разную смысловую нагрузку.

* **Какова семантика (смысл) предложений раздела PREDICATES? Когда, и где используется это описание? С какой целью?**

Предложения в разделе PREDICATES служат для описания предикатов. Предикаты описывают некоторое отношение между аргументами процедуры. Это описание используется для определения количества аргументов отношения и их природы и структуры – доменов.

* **Унификация каких термов запускается на самом первом шаге работы системы? Каковы назначение и результат использования алгоритма унификации?**

На самом первом шаге работы системы запускается унификация вопроса и первого предложения из базы знаний. Алгоритм унификации предназначен для логического вывода, система использует его для того, чтобы "увидеть одинаковость" термов и дать ответ «Да» на поставленный вопрос. Унификация является основным вычислительным шагом работы программы. Результатом работы унификации является успех или неудача, побочным эффектом является построенная в ходе унификации подстановка.

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

Механизм отката запускается в случае попадания в тупиковую ситуацию, т.е. если из текущего состояния перейти в новое состояние невозможно, или в случае, если осуществляется поиск всех возможных ответов и очередной ответ был найден.

**Текст программы**

domains

name, phone, city, street, color, brand, money , bank, account, building\_type, area\_type, watercraft\_type, property\_name = symbol

house, apartment = integer

addr = address(city, street, house, apartment)

property = car(brand, color, money); building(building\_type, money); area(area\_type, money); watercraft(watercraft\_type, money)

predicates

phonebook(name, phone, addr).

owner(name, property).

depositor(name, bank, account, money).

property\_info(property, property\_name, money).

get\_property\_by\_name(name, property\_name, money).

clauses

property\_info(car(Brand, \_, CarMoney), Type, Money) :- Type=Brand, Money=CarMoney.

property\_info(building(Building, BuildingMoney), Type, Money) :- Type=Building, Money=BuildingMoney.

property\_info(area(Area, AreaMoney), Type, Money) :- Type=Area, Money=AreaMoney.

property\_info(watercraft(Watercraft, WatercraftMoney), Type, Money) :- Type=Watercraft, Money=WatercraftMoney.

get\_property\_by\_name(Name, Type, Money) :- owner(Name, Property), property\_info(Property, Type, Money).

owner("Ivanov", car("Audi", "Blue", "3000000")).

owner("Ivanov", building("Apartment", "10000000")).

owner("Ivanov", watercraft("Yacht", "25000000")).

owner("Petrov", car("BMW", "Black", "3500000")).

owner("Petrov", area("Field", "500000")).

owner("Sidorov", car("BMW", "Red", "2000000")).

phonebook("Ivanov", "79836457823", address("Moscow", "Tverskaya", 4, 112)).

phonebook("Sidorov", "79285920831", address("Tver", "Orlova", 17, 22)).

phonebook("Ivanov", "79260112361", address("Moscow", "Tverskaya", 4, 112)).

phonebook("Petrov", "79256239576", address("St-Petersburg", "Leninskaya", 19, 26)).

phonebook("Sidorov", "79278456344", address("Moscow", "Puskinskaya", 2, 34)).

depositor("Sidorov", "Sberbank", "1238123127", "5000000").

depositor("Ivanov", "Tinkoff", "5872874928", "300000").

depositor("Sidorov", "VTB", "123123213", "2000000").

depositor("Petrov", "Sberbank", "123213213", "3000000").

Предикат property\_info позволяет получить название и стоимость для любого объекта, принадлежащего вариативному домену property. Предикат get\_property\_by\_name позволяет получить все названия и стоимости объектов собственности по имени владельца.

**Примеры работы**

|  |  |
| --- | --- |
| Вопрос | Результат |
| Задание 1. Поиск названий всех объектов собственности заданного субъекта  **get\_property\_by\_name("Ivanov", Type, \_).** | Type=Audi  Type=Apartment  Type=Yacht  3 Solutions |
| Задание 2. Поиск названий и стоимости всех объектов собственности заданного субъекта  **get\_property\_by\_name("Ivanov", Type, Money).** | Type=Audi, Money=3000000  Type=Apartment, Money=10000000  Type=Yacht, Money=25000000  3 Solutions |

Порядок работы системы для 2-го примера:

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат (к чему приводит?) |
| 1 | - Попытка унификации get\_property\_by\_name("Ivanov", Type, Money) = property\_info(car(Brand, \_, CarMoney), Type, Money)  - Результат: не унифицируемы, так как имеют разные функторы | Прямой ход, переход к следующему знанию в БЗ |
| 2 | - Попытка унификации get\_property\_by\_name("Ivanov", Type, Money) = property\_info(building(Building, BuildingMoney), Type, Money)  - Результат: не унифицируемы, так как имеют разные функторы | Прямой ход, переход к следующему знанию в БЗ |
| 3 | - Попытка унификации get\_property\_by\_name("Ivanov", Type, Money) = property\_info(area(Area, AreaMoney), Type, Money)  - Результат: не унифицируемы, так как имеют разные функторы | Прямой ход, переход к следующему знанию в БЗ |
| 4 | - Попытка унификации get\_property\_by\_name("Ivanov", Type, Money) = property\_info(watercraft(Watercraft, WatercraftMoney), Type, Money)  - Результат: не унифицируемы, так как имеют разные функторы | Прямой ход, переход к следующему знанию в БЗ |
| 5 | - Попытка унификации get\_property\_by\_name("Ivanov", Type, Money) = get\_property\_by\_name(Name, Type, Money)  - Результат: Успех, подстановка {Name=”Ivanov”, Type=Type, Money=Money} | Прямой ход, переход к началу БЗ |
| 6 | - Попытка унификации owner(“Ivanov”, Property) = property\_info(car(Brand, \_, CarMoney), Type, Money)  - Результат: не унифицируемы, так как имеют разные функторы | Прямой ход, переход к следующему знанию в БЗ |
| 7 | - Попытка унификации owner(“Ivanov”, Property) = property\_info(building(Building, BuildingMoney), Type, Money)  - Результат: не унифицируемы, так как имеют разные функторы | Прямой ход, переход к следующему знанию в БЗ |
| 8 | - Попытка унификации owner(“Ivanov”, Property) = property\_info(area(Area, AreaMoney), Type, Money)  - Результат: не унифицируемы, так как имеют разные функторы | Прямой ход, переход к следующему знанию в БЗ |
| 9 | - Попытка унификации owner(“Ivanov”, Property) = property\_info(watercraft(Watercraft, WatercraftMoney), Type, Money)  - Результат: не унифицируемы, так как имеют разные функторы | Прямой ход, переход к следующему знанию в БЗ |
| 10 | - Попытка унификации owner(“Ivanov”, Property) = get\_property\_by\_name(Name, Type, Money)  - Результат: не унифицируемы, так как имеют разные функторы | Прямой ход, переход к следующему знанию в БЗ |
| 11 | - Попытка унификации owner(“Ivanov”, Property) = owner("Ivanov", car("Audi", "Blue", "3000000"))  - Результат: Успех, подстановка {Property= car("Audi", "Blue", "3000000")} | Прямой ход, переход к началу БЗ |
| 12 | - Попытка унификации property\_info(car("Audi", "Blue", "3000000"), Type, Money) = property\_info(car(Brand, \_, CarMoney), Type, Money)  - Результат: Успех, подстановка {Brand="Audi", \_="Blue", CarMoney="3000000", Type = Type, Money = Money} | Прямой ход, запуск унификации вручную с помощью = |
| 13 | - Попытка унификации Type="Audi"  - Результат: Успех, подстановка {Type="Audi"} | Прямой ход, запуск унификации вручную с помощью = |
| 14 | - Попытка унификации Money="3000000"  - Результат: Успех, подстановка {Money="3000000"} | Резольвента пуста, вывод подстановки {Type=”Audi”, Money="3000000”}, откат к предыдущему состоянию резольвенты |
| 15 | - Попытка унификации property\_info(car("Audi", "Blue", "3000000"), Type, Money) = property\_info(building(Building, BuildingMoney), Type, Money)  - Результат: не унифицируемы, так как car("Audi", "Blue", "3000000") и building(Building, BuildingMoney) не унифицируемы | Прямой ход, переход к следующему знанию в БЗ |
| 16 | - Попытка унификации property\_info(car("Audi", "Blue", "3000000"), Type, Money) = property\_info(area(Area, AreaMoney), Type, Money)  - Результат: не унифицируемы, так как car("Audi", "Blue", "3000000") и area(Area, AreaMoney) не унифицируемы | Прямой ход, переход к следующему знанию в БЗ |
| 17 | - Попытка унификации property\_info(car("Audi", "Blue", "3000000"), Type, Money) = property\_info(watercraft(Watercraft, WatercraftMoney), Type, Money)  - Результат: не унифицируемы, так как car("Audi", "Blue", "3000000") и watercraft(Watercraft, WatercraftMoney) не унифицируемы | Прямой ход, переход к следующему знанию в БЗ |
| 18 - 32 | - Попытка унификации property\_info(car("Audi", "Blue", "3000000"), Type, Money) = …  - Результат: не унифицируемы, так как имеют разные функторы | Прямой ход, переход к следующему знанию в БЗ |
| 33 | - Попытка унификации property\_info(car("Audi", "Blue", "3000000"), Type, Money) = depositor("Petrov", "Sberbank", "123213213", "3000000")  - Результат: не унифицируемы, так как имеют разные функторы | Откат к предыдущему состоянию резольвенты |
| 34 | - Попытка унификации owner(“Ivanov”, Property) = owner("Ivanov", building("Apartment", "10000000"))  - Результат: Успех, подстановка {Property=building("Apartment", "10000000")} | Прямой ход, переход к началу БЗ |
| 35 | - Попытка унификации property\_info(building("Apartment", "10000000"), Type, Money) = property\_info(car(Brand, \_, CarMoney), Type, Money)  - Результат: не унифицируемы, так как building("Apartment", "10000000") и car(Brand, \_, CarMoney) не унифицируемы | Прямой ход, переход к следующему знанию в БЗ |
| 36 | - Попытка унификации property\_info(building("Apartment", "10000000"), Type, Money) = property\_info(building(Building, BuildingMoney), Type, Money)  - Результат: Успех, подстановка { Building ="Apartment", BuildingMoney="10000000", Type = Type, Money = Money} | Прямой ход, запуск унификации вручную с помощью = |
| 37 | - Попытка унификации Type="Apartment"  - Результат: Успех, подстановка {Type="Apartment"} | Прямой ход, запуск унификации вручную с помощью = |
| 38 | - Попытка унификации Money="10000000"  - Результат: Успех, подстановка {Money="10000000"} | Резольвента пуста, вывод подстановки {Type=”Apartment ”, Money="10000000"}, откат к предыдущему состоянию резольвенты |
| 39 | - Попытка унификации property\_info(building("Apartment", "10000000"), Type, Money) = property\_info(area(Area, AreaMoney), Type, Money)  - Результат: не унифицируемы, так как building("Apartment", "10000000") и area(Area, AreaMoney) не унифицируемы | Прямой ход, переход к следующему знанию в БЗ |
| 40 | - Попытка унификации property\_info(building("Apartment", "10000000"), Type, Money) = property\_info(watercraft(Watercraft, WatercraftMoney), Type, Money)  - Результат: не унифицируемы, так как building("Apartment", "10000000") и watercraft(Watercraft, WatercraftMoney) не унифицируемы | Прямой ход, переход к следующему знанию в БЗ |
| 41 - 55 | - Попытка унификации property\_info(building("Apartment", "10000000"), Type, Money) = …  - Результат: не унифицируемы, так как имеют разные функторы | Прямой ход, переход к следующему знанию в БЗ |
| 56 | - Попытка унификации property\_info(building("Apartment", "10000000"), Type, Money) = depositor("Petrov", "Sberbank", "123213213", "3000000")  - Результат: не унифицируемы, так как имеют разные функторы | Откат к предыдущему состоянию резольвенты |
| 57 | - Попытка унификации owner(“Ivanov”, Property) = owner("Ivanov", watercraft("Yacht", "25000000"))  - Результат: Успех, подстановка {Property= watercraft("Yacht", "25000000")} | Прямой ход, переход к началу БЗ |
| 58 | - Попытка унификации property\_info(watercraft("Yacht", "25000000"), Type, Money) = property\_info(car(Brand, \_, CarMoney), Type, Money)  - Результат: не унифицируемы, так как watercraft("Yacht", "25000000") и car(Brand, \_, CarMoney) не унифицируемы | Прямой ход, переход к следующему знанию в БЗ |
| 59 | - Попытка унификации property\_info(watercraft("Yacht", "25000000"), Type, Money) = property\_info(building(Building, BuildingMoney), Type, Money)  - Результат: не унифицируемы, так как watercraft("Yacht", "25000000") и building(Building, BuildingMoney) не унифицируемы | Прямой ход, переход к следующему знанию в БЗ |
| 60 | - Попытка унификации property\_info(watercraft("Yacht", "25000000"), Type, Money) = property\_info(area(Area, AreaMoney), Type, Money)  - Результат: не унифицируемы, так как watercraft("Yacht", "25000000") и area(Area, AreaMoney) не унифицируемы | Прямой ход, переход к следующему знанию в БЗ |
| 61 | - Попытка унификации property\_info(watercraft("Yacht", "25000000"), Type, Money) = property\_info(watercraft(Watercraft, WatercraftMoney), Type, Money)  - Результат: Успех, подстановка {Watercraft ="Yacht", WatercraftMoney ="25000000", Type = Type, Money = Money} | Прямой ход, запуск унификации вручную с помощью = |
| 62 | - Попытка унификации Type="Yacht"  - Результат: Успех, подстановка {Type="Yacht"} | Прямой ход, запуск унификации вручную с помощью = |
| 63 | - Попытка унификации Money="25000000"  - Результат: Успех, подстановка {Money="25000000"} | Резольвента пуста, вывод подстановки {Type=”Yacht”, Money="25000000"}, откат к предыдущему состоянию резольвенты |
| 64 - 78 | - Попытка унификации property\_info(watercraft("Yacht", "25000000"), Type, Money) = …  - Результат: не унифицируемы, так как имеют разные функторы | Прямой ход, переход к следующему знанию в БЗ |
| 79 | - Попытка унификации property\_info(watercraft("Yacht", "25000000"), Type, Money) = depositor("Petrov", "Sberbank", "123213213", "3000000")  - Результат: не унифицируемы, так как имеют разные функторы | Откат к предыдущему состоянию резольвенты |
| 80 | - Попытка унификации owner(“Ivanov”, Property) = owner("Petrov", car("BMW", "Black", "3500000"))  - Результат: не унифицируемы, так как “Ivanov” и “Petrov” не унифицируемы | Прямой ход, переход к следующему знанию в БЗ |
| 81 | - Попытка унификации owner(“Ivanov”, Property) = owner("Petrov", area("Field", "500000"))  - Результат: не унифицируемы, так как “Ivanov” и “Petrov” не унифицируемы | Прямой ход, переход к следующему знанию в БЗ |
| 82 | - Попытка унификации owner(“Ivanov”, Property) = owner("Sidorov", car("BMW", "Red", "2000000"))  - Результат: не унифицируемы, так как “Ivanov” и “Sidorov” не унифицируемы | Прямой ход, переход к следующему знанию в БЗ |
| 83 - 90 | - Попытка унификации owner(“Ivanov”, Property) = …  - Результат: не унифицируемы, так как имеют разные функторы | Прямой ход, переход к следующему знанию в БЗ |
| 91 | - Попытка унификации owner(“Ivanov”, Property) = depositor("Petrov", "Sberbank", "123213213", "3000000")  - Результат: не унифицируемы, так как имеют разные функторы | Откат к предыдущему состоянию резольвенты |
| 92 - 105 | - Попытка унификации get\_property\_by\_name("Ivanov", Type, Money) = …  - Результат: не унифицируемы, так как имеют разные функторы | Прямой ход, переход к следующему знанию в БЗ |
| 106 | - Попытка унификации get\_property\_by\_name("Ivanov", Type, Money) = depositor("Petrov", "Sberbank", "123213213", "3000000")  - Результат: не унифицируемы, так как имеют разные функторы | Конец работы |

**Замечания к лабораторной работе 13**

1. Что такое подстановка?

Пусть дан терм: (𝑋1,𝑋2, . . . ,𝑋𝑛). Подстановка - множество пар, вида:

{𝑋𝑖 = 𝑡𝑖}, где 𝑋𝑖 – переменная, а 𝑡𝑖 – терм.

Что такое пример терма? Как и когда строится? Как Вы думаете,

система строит и хранит примеры?

Пусть Θ = {𝑋1 = 𝑡1,𝑋2 = 𝑡2, . . . ,𝑋𝑛 = 𝑡𝑛} – подстановка, 𝐴 - терм. Результат применения подстановки к терму обозначается 𝐴Θ.

Примером терма 𝐴 называется терм 𝐵, если существует подстановка Θ такая, что 𝐵 = 𝐴Θ.

Примеры термов строятся в ходе логического вывода. Для построения примера терма его переменные конкретизируются.

А в Вашем примере строится подстановка?

**Ответ:** Да, в примере разобран ход логического вывода, для каждой успешной унификации приведена подстановка.

Пример: Сравниваются find\_name\_brand\_money( "79260112361", Name, Brand, Money) и find\_name\_brand\_money(Phone, Name, Brand, Money), подстановка - {Phone="79260112361", Name=Name, Brand=Brand, Money=Money}.

1. Зачем в БЗ кавычки, без них нельзя?

**Ответ:** Кавычки в базе знаний используются для того, чтобы начинать символьные константы с большой буквы, так как symbol — это либо последовательность букв, цифр и подчерков, где первый символ - строчная буква, либо последовательность любых символов, заключенная в кавычки.

1. Приведите пример для вопроса по заданию 1.

**Ответ:** По № телефона найти: Фамилию, Марку автомобиля, Стоимость автомобиля (может быть несколько) find\_name\_brand\_money("79260112361", Name, Brand, Money).

1. Какой вопрос (цель) какую задачу решает? Если Вы знаете ФИО, а надо найти 3, то параметров 4. А можно вернуть не все? Как это сделать в задании b?

**Ответ:** Таблица с примерами вопросов на каждое задание

|  |  |
| --- | --- |
| Вопрос | Результат |
| Задание 1.а. По № телефона найти: Фамилию, Марку автомобиля, Стоимость автомобиля (может быть несколько)  **find\_name\_brand\_money("79260112361", Name, Brand, Money).** | Name=Ivanov, Brand=Mercedes, Money=5000000  1 Solution |
| Задание 1.b. Используя сформированное в пункте а) правило, по № телефона найти: только Марку автомобиля (автомобилей может быть несколько)  **find\_name\_brand\_money("79256239576", \_, Brand, \_).** | Brand=BMW  Brand=Audi  2 Solutions |
| Здания 2. Используя простой, не составной вопрос: по Фамилии (уникальна в городе, но в разных городах есть однофамильцы) и Городу проживания найти: Улицу проживания, Банки, в которых есть вклады и №телефона.  **find\_street\_bank\_phone("Sidorov", "Moscow", Street, Bank, Phone).** | Street=Puskinskaya, Bank=Sberbank, Phone=79278456344 Street=Puskinskaya, Bank=VTB, Phone=79278456344  2 Solutions |

Для того, чтобы вернуть не все результаты, вместо именованной переменной нужно использовать анонимную переменную, которая не может быть конкретизирована. В задании b вместо переменных Name и Money передаются анонимные переменные, таким образом возвращается только модель автомобиля.

1. Какой вопрос (цель) какую задачу из задания решает? Если Вы знаете ФИО, а надо найти 3, то параметров 4. А можно вернуть не все? Как это сделать в задании b? Было Задание: b. Используя сформированное в пункте «а» правило,!!!!

**Ответ** приведен в предыдущем пункте.

**Замечания к лабораторной работе 14**

1. Задание 1

Порядок формирования результата для 1-го вопроса, шаг 3: Сравниваются find\_name\_city\_phone\_bank("Audi","Blue", Name, City, Phone, Bank) и find\_name\_city\_phone\_bank(Brand,Color, Name, City, Phone, Bank), подстановка - {Brand="Audi",Color="Blue"} НЕ полная!!, выпишите полную, А Name и Name из разных термов – это одна и та же переменная? А когда будут установлены значения других переменных Name, City, Phone, Bank?

**Ответ: Подстановка:** {Brand="Audi",Color="Blue", Name=Name, City=City, Phone=Phone, Bank=Bank}. Name и Name из разных термов – две разные переменные, так как переменная уникальна в пределах предложения. При унификации они становятся сцепленными, т.е. именами одного объекта, и при конкретизации получают одинаковое значение. Переменные Name, City, Phone и Bank будут конкретизированы на следующих шагах доказательства, т.е. при унификации тела правила find\_name\_city\_phone\_bank.

1. Работа алгоритма унификации:

А что находится **в результирующей ячейке –** При **успешной** унификации 2-термов?

**Ответ:** При успешной унификации термов в результирующей ячейке находится подстановка.

1. Задание 3

Порядок работы алгоритма унификации для 1-го случая из задания 2: А алгоритм **унификации** запускается **только для 2-**термов, и **заново** для 2- других**,** или иначе? Я просила таблицу работы алгоритма только для 2-х термов.

**Ответ:** Алгоритм унификации запускается для 2-х термов, она может завершиться успехом или неудачей, при успешной унификации унифицируемая подцель резольвенты заменяется телом унифицированного правила из БЗ и к резольвенте применяется найденная подстановка, затем для самой первой подцели из резольвенты и первого правила из базы знаний алгоритм унификации запускается заново. При неудачной унификации осуществляется попытка унифицировать первую подцель резольвенты со следующим правилом из БЗ, если правил нет, то выполняется откат. Т.е. для очередной пары термов процесс унификации запускается заново.

Таблица работы алгоритма унификации для вопроса и соответствующего правила:

|  |  |  |  |
| --- | --- | --- | --- |
| шаг унификации | результирующая ячейка | рабочее поле | стек |
| 0 |  |  | find\_name\_city\_phone\_bank( "Audi", "Blue", Name, City, Phone, Bank) = find\_name\_city\_phone\_bank( Brand, Color, Name, City, Phone, Bank) |
| 1 |  | find\_name\_city\_phone\_bank( "Audi", "Blue", Name, City, Phone, Bank) = find\_name\_city\_phone\_bank( Brand, Color, Name, City, Phone, Bank) ---> | Brand = "Audi"  Color = "Blue"  Name = Name  City = City Phone = Phone  Bank = Bank |
| 2 | Brand = "Audi" | <--- Brand = "Audi" | Color = "Blue"  Name = Name  City = City Phone = Phone  Bank = Bank |
| 3 | Brand = "Audi"  Color = "Blue" | <--- Color = "Blue" | Name = Name  City = City Phone = Phone  Bank = Bank |
| 4 | Brand = "Audi"  Color = "Blue"  Name = Name | <--- Name = Name | City = City Phone = Phone  Bank = Bank |
| 5 | Brand = "Audi"  Color = "Blue"  Name = Name  City = City | <--- City = City | Phone = Phone  Bank = Bank |
| 6 | Brand = "Audi"  Color = "Blue"  Name = Name  City = City Phone = Phone | <--- Phone = Phone | Bank = Bank |
| 7 | Brand = "Audi"  Color = "Blue"  Name = Name  City = City Phone = Phone  Bank = Bank | <--- Bank = Bank |  |
| Вывод: | **подстановка** | Т.к. стек пуст – **успех** и  в результирующей ячейке подстановка | |

1. А что являетсяпризнаком необходимости **завершить** работу алгоритма?

Почему на 12-м шаге в стек попали новые термы? Ощущение, что по Вашей логике, алгоритм унификации работает бесконечно…, иначе, а как система наконец поняла, что надо остановиться?

**А как вообще система «понимает», что надо остановиться?**

**Ответ:** Алгоритм может быть завершен в двух случаях: либо при неудачной унификации флаг неудача будет установлен в 1 и будет осуществлен выход из алгоритма унификации, либо при успешной унификации признаком завершения будет пустой стек. Корректная таблица для алгоритма унификации приведена выше. Система «понимает», что нужно остановиться, когда все правила в БЗ будут отмечены, как просмотренные, и невозможно будет сделать откат, т.е. резольвента будет содержать исходный вопрос.