|  |  |
| --- | --- |
| 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»

**ОТЧЕТ**

*к лабораторной работе № 15*

*По курсу: «Функциональное и логическое программирование»*

Студент ИУ7-65Б

Юмаев А.Р.

Преподаватели

Строганов Ю.В.

Толпинская Н.Б.

*Москва, 2020 г.*

**Задание**

**Создать базу знаний «Собственники»**, дополнив базу знаний, хранящую знания (лаб. 13):

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

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

**Вид собственности:**

1. Строение, стоимость и другие характеристики;
2. Участок, стоимость и другие его характеристики;
3. Водный транспорт, стоимость и другие его характеристики;
4. Автомобиль, марка, цвет, стоимость и др.

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 пункта 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).

**Ответы на вопросы**

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

Формально - знание о предметной области. Знание формируется в заголовке правила. Правила описаны в разделе clauses.

1. **Что содержит тело правила?**

Тело правила содержит условия истинности знания: термы-подцели, чью истинность необходимо доказать, чтобы знание было истинным.

1. **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 используются для передачи знаний (значений, с которыми сопоставлены в данный момент переменные) «во времени и пространстве». Кроме того, использование переменных позволяет сделать правило более общим. Например, cook(fried\_potato, potato, oil) – факт, cook(fried\_potato, Product1, Product2) - более общий вид, если задать такой вопрос, программа выдаст все наборы продуктов, с помощью которых можно получить fried\_potato.

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

Переменные входят в правило с квантором всеобщности. Именованная переменная уникальна в рамках предложения, в котором она используются. Любая анонимная переменная является уникальной.

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

В случае, если структура объектов, используемых для обозначения аргументов, между которыми устанавливается отношение в заголовке правил процедуры не может быть описана с помощью стандартных доменов, она записывается в раздел domains.

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

В случае если природа или структура объектов, обозначенных в качестве аргументов, между которыми устанавливается отношение значима во время работы системы, то оно описывается в разделе predicates.

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

На первом шаге работы программы унифицируется текущая цель(вопрос, заданный пользователем) и первое знание в БЗ (предложения в БЗ сопоставляются сверху вниз от первого (самого верхнего) до нижнего (последнего))

Для поиска ответа на вопрос системе необходимо найти подходящее знание в БЗ, для поиска такого знания используется алгоритм унификации. Формально, он помогает системе понять, что заголовок подошел: алгоритм попарно пытается сопоставить термы (текущую цель и термы из БЗ) и построить для них общий пример (для этого используется подстановка).

Алгоритм унификации может завершиться успехом и неудачей, если в результат – успех – в результирующей ячейке сформируется ниаболее общий унификатор (подстановка), в качестве побочного эффекта.

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

Механизм отката запускается в случае возникновения тупиковой ситуации.

**Текст программы**

|  |
| --- |
| domains   price, value, house, appartment = integer  surName, phone, colour, bank, account, city, street, propertyName = symbol  adress = adress(city, street, house, appartment)   % домен собственность  property = car(propertyName, price, colour);  ship(propertyName, price);  building(propertyName, price);  sector(propertyName, price)   predicates  owns(surName, property)  owns(surName, name, price)  count\_sum(surName, price)   phonebook (surName, phone, adress)  invests(surName, bank, symbol account, value)   clauses  owns(«Person1», car("Volvo", 123123123, "Black")).  owns(«Person1», ship("Ship", 45645645)).  owns(«Person1», building("House", 98765432)).  owns(«Person2», sector("Area", 876345765)).  owns(«Person3», car("BMW", 321321321, "Pink")).    phonebook(«Person1», «0-000-000-00-00»,  adress(«London», "Baker", 14, 128)).  phonebook(«Person1», «0-000-000-00-00»,  adress(«London», "Bakery", 1, 10)).  phonebook(«Person1», «1-111-111-11-11»,  adress("Samara", "superskaya", 14, 128)).  phonebook(«Person2», «2-222-222-22-22»,  adress("Moscow", "woy-street", 12, 153)).  phonebook(«Person3», «3-333-333-33-33»,  adress("Moscow", "superskaya", 1, 10)).  phonebook(«Person3», «4-444-444-44-44»,  adress("Samara", "Bakery", 1, 10)).  invests(«Person1», "Sberbank", "1111 1111 1111 1111 1111", 1500000).  invests(«Person1», "MoscowBank", "2222 2222 2222 2222 2222", 30000).  invests(«Person3», "Tinkoff", "3333 3333 3333 3333 3333", 4500000).  invests(«Person3», "VTB","4444 4444 4444 4444 4444", 10000).  invests(«Person4», "Sberbank", "5555 5555 5555 5555 5555", 400000).    owns(SurName, Name, Price):- owns(SurName, car(Name, Price, \_)).  owns(SurName, Name, Price):- owns(SurName, ship(Name, Price)).  owns(SurName, Name, Price):- owns(SurName, building(Name, Price)).  owns(SurName, Name, Price):- owns(SurName, sector(Name, Price)).      count\_sum(SurName, Count\_sum):- owns(SurName, Name, Price),   owns(SurName, Name1, Price1),   Name<>Name1,  owns(SurName, Name2, Price2),  Name<>Name2,   Name1<>Name2,  owns(SurName, Name3, Price3),  Name<>Name3, Name1<>Name3, Name2<>Name3,  Count\_sum=Price+Price1+Price2+Price3,!.    count\_sum(SurName, Count\_sum):- owns(SurName, Name, Price),   owns(SurName, Name1, Price1),   Name<>Name1,  owns(SurName, Name2, Price2),  Name<>Name2, Name1<>Name2,  Count\_sum=Price+Price1+Price2,!.    count\_sum(SurName, Count\_sum):- owns(SurName, Name, Price),   owns(SurName, Name1, Price1),   Name<>Name1,  Count\_sum=Price+Price1, !.    count\_sum(SurName, Count\_sum):- owns(SurName, \_, Price), Count\_sum = Price, !. |

**Примеры целей и результатов работы программы**

Поиск названий и стоимости всех объектов собственности заданного субъекта.

Цель:

owns(person("Name1", "SurName1"), Name, Price).

Результат:

Name=Volvo, Price=123123123

Name=Ship, Price=45645645

Name=House, Price=98765432

Поиск названий всех объектов собственности заданного субъекта.

Цель:

owns(«Person1», Name, \_)

Результат:

Name=Volvo

Name=Ship

Name=House

**Пояснения решения доп задачи**

Поскольку предикат owns может работать с любым видом собственности, проблема заключалась в том, чтобы как-то найти все виды и сохранить стоимость каждого, так как во время отката, значение терялось бы, поскольку у каждого собственника может быть всего лишь один объект каждой собственности, у которого есть имя – имя становится уникальным в контексте одного человека. На этом и построена работа данной процедуры. Были реализованы правила для любого возможного количества объектов собственности владельца, а поскольку домен собственности вариантный – нам нет нужны заботиться о порядке встречающихся в БЗ правил.

**Описание порядка поиска объектов**

**Цель:**  owns(«Person2», Name, Price).

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка,  если есть | Дальнейшие действия: прямой ход или откат (к чему приводит?) |
| 1 | Попытка унификации:  owns(«Person2», Name, Price),  owns(«Person1», car("Volvo", 123123123, "Black"))  Результат: неудача, разная арность термов. | Прямой ход, переход к следующему предложению. |
| ... | Попытка унификации:  owns(«Person2», Name, Price),  и термов пока не конец БЗ или пока не будет найден подходящий (имена главных функторов совпадают, одинаковая арность) терм. | Прямой ход, переход к следующему предложению. |
| 12 | Попытка унификации:  owns(«Person2», Name, Price),  owns(SurName, Name, Price).  Результат: успех  Подстановка: {SurName=«Person2», Name=Name, Price=Price}. | Прямой ход,  переход в тело правила |
| 13 | Попытка унификации:  owns(«Person2», car(Name, Price, \_)).  owns(«Person1», car("Volvo", 123123123, "Black"))..  Результат: неудача, разные константы. | Прямой ход, переход к следующему предложению. |
| ... | Попытка унификации:  owns(«Person2», car(Name, Price, \_))  и термов пока не конец БЗ или пока не будет найден подходящий (имена главных функторов совпадают, одинаковая арность) терм. | Конец БЗ, откат |
| 36 | Попытка унификации:  owns(«Person2», car(Name, Price, \_)),  count\_sum(SurName, Count\_sum).  Результат: неудача, разные имена главных функторов, термы не унифицируемы. | Прямой ход, переход к следующему предложению. |
| 37 | Попытка унификации:  owns(«Person2», Name, Price),  owns(SurName, Name, Price).  Результат: успех  Подстановка: {SurName=«Person2», Name=Name, Price=Price}. | Прямой ход,  переход в тело правила |
| 38 | Попытка унификации:  owns(«Person2», ship(Name, Price))  owns(«Person1», car("Volvo", 123123123, "Black"))  Результат: неудача, несовпадающие константы. | Прямой ход, переход к следующему предложению |
| 39 | Попытка унификации:  owns(«Person2», ship(Name, Price)) и термов пока не конец БЗ или пока не будет найден подходящий (имена главных функторов совпадают, одинаковая арность) терм. | Прямой ход, переход к следующему предложению |
| 40 | Попытка унификации:  owns(«Person2», ship(Name, Price))  count\_sum(SurName, Count\_sum).  Результат: неудача, разные имена главных функторов, термы не унифицируемы. | Конец БЗ,  откат |
| 41 | Попытка унификации:  owns(«Person2», Name, Price),  owns(SurName, Name, Price).  Результат: успех  Подстановка: {SurName=«Person2», Name=Name, Price=Price}. | Прямой ход,  переход в тело правила |
| ... | Попытка унификации:  owns(SurName, building(Name, Price))  и термов (начиная с верхнего), пока не конец БЗ или пока не будет найден подходящий (имена главных функторов совпадают, одинаковая арность) терм. | Прямой ход, переход к следующему предложению |
| 65 | Попытка унификации:  owns(SurName, building(Name, Price)) count\_sum(SurName, Count\_sum).  Результат: неудача, разные имена главных функторов, термы не унифицируемы. | Конец БЗ,  откат |
| 66 | Попытка унификации:  owns(«Person2», Name, Price),  owns(SurName, Name, Price).  Результат: успех  Подстановка: {SurName=«Person2», Name=Name, Price=Price}. | Прямой ход,  переход в тело правила |
| 67 | Попытка унификации:  owns(«Person2», sector(Name, Price)).  owns(«Person1», car("Volvo", 123123123, "Black"))..  Результат: неудача, разные константы. | Прямой ход, переход к следующему предложению |
| 68 | Попытка унификации:  owns(«Person2», sector(Name, Price)).  owns(«Person1», ship("Ship", 45645645)).  Результат: неудача, разные константы. | Прямой ход, переход к следующему предложению |
| 69 | Попытка унификации:  owns(«Person2», sector(Name, Price)).  owns(«Person1», building("House", 98765432)).  Результат: неудача, разные константы. | Прямой ход, переход к следующему предложению |
| 70 | Попытка унификации:  owns(«Person2», sector(Name, Price)).  owns(«Person2», sector("Area", 876345765)).  Результат: успех  Подстановка: {Name= "Area", Price=876345765} | Вывод  Name= "Area",  Price=876345765  Поиск следующих решений. |
| 71 | Попытка унификации:  owns(«Person2», sector(Name, Price)).  и термов пока не конец БЗ или пока не будет найден подходящий (имена главных функторов совпадают, одинаковая арность) терм. | Прямой ход, переход к следующему предложению |
| 72 | Попытка унификации:  owns(«Person2», sector(Name, Price)).  count\_sum(SurName, Count\_sum).  Результат: неудача, разные имена главных функторов, термы не унифицируемы. | Конец БЗ,  откат |
| ... | Попытка унификации:  owns(«Person2», Name, Price),  и термов пока не конец БЗ или пока не будет найден подходящий (имена главных функторов совпадают, одинаковая арность) терм. | Прямой ход, переход к следующему предложению |
| 76 | Попытка унификации:  owns(«Person2», Name, Price),  count\_sum(SurName, Count\_sum).  Результат: неудача, разные имена главных функторов, термы не унифицируемы. | Конец БЗ,  Конец работы. |

Таблица 1: Порядок поиска объектов