|  |  |
| --- | --- |
|  | **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»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ДИСЦИПЛИНА «Функциональное и логическое программирование»

**Лабораторная работа № 15**

**“Структура программы на Prolog и ее реализация”**

|  |  |
| --- | --- |
| **Студент**  Чалый А. А.  **Группа** ИУ7 – 62 Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Толпинская Н.Б. |  |

Москва.

2021 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 программы 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,

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)

Листинг 1. owners.pro

domains

lname, phone, city, street = symbol.

building\_n, apartment, amount = integer.

model, color, bank, account = symbol.

address = address(city, street, building\_n, apartment).

cost, size = integer.

property =

building(cost, size);

sector(cost, size);

ship(cost, color);

car(model, color, cost).

predicates

phone\_book(lname, phone, address).

investors(lname, bank, account, amount).

own(lname, property).

own\_type(lname, symbol, cost).

own\_cost(lname, symbol, cost).

total\_cost(lname, cost).

clauses

phone\_book("Chalyy", "+123", address("Moscow", "Postal", 1, 2)).

phone\_book("Chalyy", "+000", address("SPB", "Neva", 2, 4)).

phone\_book("Gag", "+345", address("Sevastopol", "Anotherstrt", 4, 8)).

phone\_book("Dkul", "+333", address("Moscow", "Postal", 8, 16)).

investors("Chalyy", "RNCB", private, 999999999).

investors("Dkul", "Sberbank", public, 10).

investors("Gag", "Raiffeisen", public, 10000).

own("Chalyy", building(12000, 40)).

own("Chalyy", car("Audi", black, 150)).

own("Gag", ship(1000, grey)).

own\_type(Lname, building, Cost) :- own(Lname, building(Cost, \_)).

own\_type(Lname, sector, Cost) :- own(Lname, sector(Cost, \_)).

own\_type(Lname, ship, Cost) :- own(Lname, ship(Cost, \_)).

own\_type(Lname, car, Cost) :- own(Lname, car(\_, \_, Cost)).

Таблица 1. own\_type("Chalyy ", PropType, Cost).

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат |
| 1 | own\_type("Chalyy", PropType, Cost). = phone\_book("Chalyy", "+123", address("Moscow", "Postal", 1, 2)).  Унификация не успешна.  Разные функторы | Переход к следующему предложению |
| … | ... | … |
| 5 | own\_type("Chalyy", PropType, Cost). = investors("Chalyy", "RNCB", private, 999999999).  Унификация не успешна.  Разные функторы | Переход к следующему предложению |
| … | … | … |
| 9 | own\_type("Chalyy", PropType, Cost). = own("Chalyy", building(12000, 40)).  Унификация не успешна.  Разные функторы | Переход к следующему предложению |
| … | … | … |
| 12 | own\_type("Chalyy", PropType, Cost). = own\_type(Lname, building, Cost)  Унификация успешна.  Θ = { “Chalyy”=Lname,  PropType = building,  Cost=Cost.  } | Прямой ход.  1.Преобразование резольвенты с помощью редукции. Замена цели телом того правила заголовок которого унифицируется с целью.  2. Применяется подстановка.  Заменяется резольвента:  own(“Chalyy”, building(Cost, \_)). |
| 13 | own("Chalyy", building(Cost, \_)). = phone\_book("Chalyy", "+123", address("Moscow", "Postal", 1, 2)).  Унификация не успешна.  Разные функторы | Переход к следующему предложению |
| … | … | … |
| 17 | own("Chalyy", building(Cost, \_)). = investors("Chalyy", "RNCB", private, 999999999).  Унификация не успешна.  Разные функторы | Переход к следующему предложению |
| … | … | … |
| 20 | own("Chalyy", building(Cost, \_)). = own("Chalyy", building(12000, 40)).  Унификация успешна  Θ = { “Chalyy”=Lname,  PropType = building,  Cost=12000.  } | Решение найдено:  PropType = building  Cost = 12000  Откат. |
| 21 | own("Chalyy", building(Cost, \_)). = own("Winter", car("Audi", black, 150)).  Унификация не успешна.  building!=car | Переход к следующему предложению |
| … | … | … |
| 23 | own("Winter", building(Cost, \_)). = own\_type(Lname, building, Cost)  Унификация не успешна.  Разные функторы  … | …  Откат к предыдущему состоянию резольвенты  own\_type("Chalyy", PropType, Cost). |
| 26 | own\_type("Chalyy", PropType, Cost). = own\_type(Lname, sector, Cost)  Унификация успешна.  Θ = { “Chalyy”=Lname,  PropType = sector,  Cost=Cost.  } | Прямой ход.  1.Преобразование резольвенты с помощью редукции. Замена цели телом того правила заголовок которого унифицируется с целью.  2. Применяется подстановка.  Заменяется резольвента:  own(“Chalyy”, sector(Cost, \_)). |
| 26-40 | own("Chalyy", sector(Cost, \_)).  Сравнение с термами из базы знаний.  Со всеми унификация не успешна  -результат: Нет | Откат к предыдущему состоянию резольвенты  own\_type("Chalyy", PropType, Cost). |
| 41 | -попытка унификации: own\_type("Chalyy", PropType, Cost). = own\_type(Lname, ship, Cost)  Унификация успешна.  Θ = { “Chalyy”=Lname,  PropType = ship,  Cost=Cost.  } | Прямой ход.  1.Преобразование резольвенты с помощью редукции. Замена цели телом того правила заголовок которого унифицируется с целью.  2. Применяется подстановка.  Заменяется резольвента:  own(“Chalyy”, ship(Cost, \_)). |
| 41-56 | own("Chalyy", ship(Cost, \_)).  Сравнение с термами из базы знаний.  Со всеми унификация не успешна | Откат к предыдущему состоянию резольвенты  own\_type("Chalyy", PropType, Cost). |
| 57 | own\_type("Chalyy", PropType, Cost). = own\_type(Lname, car, Cost)  Унификация успешна.  Θ = { “Chalyy”=Lname,  PropType = car,  Cost=Cost.  } | Прямой ход.  1.Преобразование резольвенты с помощью редукции. Замена цели телом того правила заголовок которого унифицируется с целью.  2. Применяется подстановка.  Заменяется резольвента:  own(“Chalyy”, car(Cost, \_)). |
| 58 | own("Chalyy", car(Cost, \_)). = phone\_book("Chalyy", "+123", address("Moscow", "Postal", 1, 2)).  Унификация не успешна.  Разные функторы | Переход к следующему предложению |
| … | … | … |
| 62 | own("Chalyy", car(Cost, \_)). = investors("Chalyy", "RNCB", private, 999999999).  Унификация не успешна.  Разные функторы | Переход к следующему предложению |
| … | … | … |
| 65 | own("Chalyy", car(Cost, \_,\_)). = own("Chalyy", building(12000, 40)).  Унификация не успешна.  car!=bulding | Переход к следующему предложению |
| 66 | own("Chalyy", car(Cost, \_,\_)). = own("Chalyy", car("Audi", black, 150)).  Унификация успешна  Θ = { “Chalyy”=Lname,  PropType = car  Cost=150.  } | Решение найдено:  PropType = car  Cost = 150  Откат. |
| 67 | own("Chalyy", car(Cost, \_,\_)). = own("Gag", ship(1000, grey)).  Унификация не успешна.  “Chalyy”!=”Gag” | Переход к следующему предложению |
| … | … | … |
| 68 | own("Chalyy", car(Cost, \_)). = own\_type(Lname, building, Cost)  Унификация не успешна.  Разные функторы  … | …  Откат к предыдущему состоянию резольвенты  own\_type("Chalyy", PropType, Cost). |
| … | … | Резольвента пуста.  Система завершает работу. |

**Контрольные вопросы:**

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

Знания сформулированы в clauses (факты и правила). Это знания о предметной области.

**Что содержит тело правила?**

В заголовке правила находится знание о предметной области, а в теле содержится условия истинности этого знания.

**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?**

Связанная с каким-то значением переменная, в рамках одного предложения, может быть использована в других местах.

Чем больше переменных содержит формулировка правила, тем более общим будет являться терм.

**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 используется.

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

DOMAINS — раздел описания доменов. Этот раздел используется для описания используемых структур данных.

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

PREDICATES — раздел описания предикатов. Это описание используется для проверки корректности «типов» знаний.

**Унификация каких термов запускается на самом первом шаге работы системы?**

Вопроса и первого терма в clauses.

**Каковы назначение и результат использования алгоритма унификации?**

Унификация – попытка сопоставить два терма. Результат: успех/неудача.

**В каком случае запускается механизм отката?**

Механизм отката запустится в случае неудачи алгоритма унификации.