



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»

Отчёт

по лабораторной работе № 15

Название: Структура программы на Prolog и ее реализация

Дисциплина: Функциональное и логическое программирование

Студент

ИУ7-65Б

(Группа)

Д.В. Сусликов

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

Н.Б. Толпинская

(Подпись, дата)

(И.О. Фамилия)

Москва, 2021

Цель работы - изучить структуру, особенности и принципы оформления программы, и способ выполнения программы на Prolog

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, 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 T1 и T2 и полную подстановку на каждом шаге)

Листинг:

domains

```
number, surname, city, street, brand, color, bank = symbol.  
building_num, room_num, price, bill, summa, square, cost = in  
address = address(city, street, building_num, room_num).
```

```
property =      car(brand, color, cost);  
                building(cost, street, building_num);  
                ground(cost, square);  
                water_transport(cost, color).
```

predicates

```
phonebook(surname, number, address).  
depositor(surname, bank, bill, summa).  
  
own(surname, property).  
own_by_name(surname, symbol).  
own_price_by_name(surname, symbol, cost).
```

clauses

```
phonebook("Surname1", "11111", address("City1", "St.1", 1, 17))  
phonebook("Surname2", "22222", address("City2", "St.2", 2, 18))  
phonebook("Surname3", "33333", address("City1", "St.1", 4, 28))  
phonebook("Surname4", "11111", address("City3", "St.4", 4, 24))  
  
depositor("Surname1", "Bank1", 3221, 1200).  
depositor("Surname1", "Bank2", 1233, 4000).  
depositor("Surname2", "Bank2", 4356, 2000).  
  
own("Surname1", car("Brand1", "Red", 12345)).  
own("Surname1", ground(4321, 1000)).  
own("Surname1", water_transport(22233, "Black")).  
own("Surname2", car("Brand2", "Blue", 30000)).
```

```

own("Surname2", building(49000, "St.1", 13)).
own("Surname3", car("Brand3", "Grey", 32213)).
own("Surname3", ground(4567, 1100)).

```

```

own_by_name(Name, Property):- own(Name, car(_, _, _)), Proper
own_by_name(Name, Property):- own(Name, building(_, _, _)), P
own_by_name(Name, Property):- own(Name, ground(_, _)), Proper
own_by_name(Name, Property):- own(Name, water_transport(_, _))

```

```

own_price_by_name(Name, Property, Price):- own(Name, car(_, _
own_price_by_name(Name, Property, Price):- own(Name, building
own_price_by_name(Name, Property, Price):- own(Name, ground(P
own_price_by_name(Name, Property, Price):- own(Name, water_tr

```

goal

```

% Task1.
%own_by_name("Surname1", Property).
%own_by_name("Surname3", Property).
%own_by_name("Surname4", Property).

% Task2.
%own_price_by_name("Surname1", Property, Price).
%own_price_by_name("Surname3", Property, Price).
%own_price_by_name("Surname4", Property, Price).

```

1) Названия всех объектов собственности заданного субъекта.

**Property=Car
Property=Ground
Property=Water Transport
3 Solutions|**

Пример 1.1

**Property=Car
Property=Ground
2 Solutions|**

Пример 1.2

No Solution|

Пример 1.3

2) Названия и стоимости всех объектов собственности заданного субъекта.

**Property=Car, Price=12345
Property=Ground, Price=4321
Property=Water Transport, Price=22233
3 Solutions|**

Пример 2.1

**Property=Car, Price=32213
Property=Ground, Price=4567
2 Solutions|**

Пример 2.2

No Solution|

Пример 2.3

Таблица

own_price_by_name("Surname1 Property, Price).

Шаг	Сравнимые термы; результаты; подстановка, если есть	Дальнейшие действия
1	own_price_by_name("Surname1 Property, Price) phonebook(...) Неудача. Не унифицируемы.	Переход к следующему заголовку БЗ
	...	
2	own_price_by_name("Surname1 Property, Price) own_price_by_name(Name, Property, Price) Успех. Унифицируемы. Подстановка: {Name = "Surname1 Property = Property, Price = Price}	Смена состояния резольвенты. Новое состояние резольвенты: own("Surname1 car(_, _, Price)), Property = "Car".
3	own("Surname1 car(_, _, Price)) phonebook(...) Неудача. Не унифицируемы.	Переход к следующему заголовку БЗ
	...	
4	own("Surname1 car(_, _, Price)) own("Surname1 car("Brand1 "Red 12345)) Успех. Унифицируемы. Подстановка: {"Surname1- "Surname1 Price = 12345}	Смена состояния резольвенты. Новое состояние резольвенты: Property = "Car"
5		Смена состояния резольвенты. Новое состояние резольвенты: Пустая Вывод: Property=Car, Price=12345 Новое состояние резольвенты: own("Surname1 car(_, _, Price)), Property = "Car".

6	<p>own("Surname1 car(_, _, Price))</p> <p>own("Surname1 ground(4321, 1000)).</p> <p>Неудача. Не унифицируемы.</p>	<p>Переход к следующему заголовку БЗ</p>
	...	
7	<p>own("Surname1 car(_, _, Price))</p> <p>own_price_by_name(Name, Property, Price)</p> <p>Неудача. Не унифицируемы.</p>	<p>Произведено сравнение со всеми записями БЗ.</p> <p>Смена состояния резольвенты.</p> <p>Новое состояние резольвенты: own_price_by_name("Surname1 Property, Price).</p>
8	<p>own_price_by_name("Surname1 Property, Price)</p> <p>own_price_by_name(Name, Property, Price)</p> <p>Успех. Унифицируемы.</p> <p>Подстановка: {"Surname1- Name, Property = Property, Price = Price }</p>	<p>Смена состояния резольвенты.</p> <p>Новое состояние резольвенты: own("Surname1 building(Price, _, _)), Property = "Building".</p>
9	<p>own("Surname1 building(Price, _, _))</p> <p>phonebook(...)</p> <p>Неудача. Не унифицируемы.</p>	<p>Переход к следующему заголовку БЗ</p>
	...	
10	<p>own("Surname1 building(Price, _, _))</p> <p>own_price_by_name(Name, Property, Price)</p> <p>Неудача. Не унифицируемы.</p>	<p>Произведено сравнение со всеми записями БЗ.</p> <p>Смена состояния резольвенты.</p> <p>Новое состояние резольвенты: own_price_by_name("Surname1", Property, Price).</p>

11	<p>own_price_by_name("Surname1 Property, Price). own_price_by_name(Name, Property, Price)</p> <p>Успех. Унифицируемы.</p> <p>Подстановка: {"Surname1- Name, Property = Property, Price = Price }</p>	<p>Смена состояния резольвенты. Новое состояние резольвенты: own("Surname1 ground(Price, _)), Property = "Ground".</p>
12	<p>own("Surname1 ground(Price, _)) phonebook(...)</p> <p>Неудача. Не унифицируемы.</p>	<p>Переход к следующему заголовку БЗ</p>
	...	
13	<p>own("Surname1 ground(Price, _)) own("Surname1 ground(4321, 1000)).</p> <p>Успех. Унифицируемы.</p> <p>Подстановка: {"Surname1- "Surname1 Price = 4321 }</p>	<p>Смена состояния резольвенты. Новое состояние резольвенты: Property = "Ground"</p>
14		<p>Смена состояния резольвенты. Новое состояние резольвенты: Пустая</p> <p>Вывод: Property=Ground, Price=4321</p> <p>Новое состояние резольвенты: own("Surname1 ground(Price, _)), Property = "Ground".</p>

15	<p>own("Surname1 ground(Price, _))</p> <p>own("Surname1 water_transport(22233, "Black")).</p> <p>Неудача. Не унифицируется.</p>	<p>Переход к следующему заголовку БЗ</p>
	...	
16	<p>own("Surname1 ground(Price, _))</p> <p>own_price_by_name(Name, Property, Price)</p> <p>Неудача. Не унифицируется.</p>	<p>Произведено сравнение со всеми записями БЗ.</p> <p>Смена состояния резольвенты.</p> <p>Новое состояние резольвенты: own_price_by_name("Surname1 Property, Price).</p>
17	<p>own_price_by_name("Surname1 Property, Price)</p> <p>own_price_by_name(Name, Property, Price)</p> <p>Успех. Унифицируемы.</p> <p>Подстановка: {"Surname1- Name, Property = Property, Price = Price}</p>	<p>Смена состояния резольвенты.</p> <p>Новое состояние резольвенты: own("Surname1 water_transport(Price, _)), Property = "Water Transport".</p>
18	<p>own("Surname1 water_transport(Price, _))</p> <p>phonebook(...)</p> <p>Неудача. Не унифицируемы.</p>	<p>Переход к следующему заголовку БЗ</p>
	...	
19	<p>own("Surname1 water_transport(Price, _))</p> <p>own("Surname1 water_transport(22233, "Black")).</p> <p>Успех. Унифицируемы.</p> <p>Подстановка: {"Surname1- "Surname1 Price = 22233}</p>	<p>Смена состояния резольвенты.</p> <p>Новое состояние резольвенты: Property = "Water Transport"</p>
20		<p>Смена состояния резольвенты.</p> <p>Новое состояние резольвенты: Пустая</p> <p>Вывод: Property=Water Transport, Price=22233</p> <p>Новое состояние резольвенты: own("Surname1 water_transport(Price, _)), Property = "Water Transport".</p>

21	<p>own("Surname1 water_transport(Price, _))</p> <p>own("Surname2 car("Brand2 "Blue 30000)).</p> <p>Неудача. Не унифицируется.</p>	<p>Переход к следующему заголовку БЗ</p>
	...	
22	<p>own("Surname1 water_transport(Price, _))</p> <p>own_price_by_name(Name, Property, Price)</p> <p>Неудача. Не унифицируется.</p>	<p>Произведено сравнение со всеми записями БЗ.</p> <p>Смена состояния резольвенты.</p> <p>Новое состояние резольвенты: own_price_by_name("Surname1 Property, Price).</p>
23	<p>own_price_by_name("Surname1 Property, Price)</p>	<p>Произведено сравнение со всеми записями БЗ.</p> <p>Смена состояния резольвенты.</p> <p>Новое состояние резольвенты:Пустая</p>

Ответы на вопросы

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

Знание содержится в заголовке предложений базы знаний. А предложения - в разделе CLAUSES. Знание о том, что между аргументами термина (тела правила) существует отношение.

2. Что содержит тело правила?

Тело содержит условие истинности знания.

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

Использование переменных в формулировании знаний позволяют уточнять значения и переносить их в пространстве и времени. Формулировка знаний с использованием переменных носит более общий характер по отношению к знанию, состоящему только лишь из констант.

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

Переменные в факты и правила входят с квантором всеобщности («любой» элемент из множества). Именованные переменные уникальны в рамках предложения, а анонимная – любая уникальная.

5. Какова семантика (смысл) предложений раздела DOMAINS? Когда, и где используется это описание?

DOMAINS – раздел описания доменов. В разделе объявляются любые нестандартные домены в формате: <имя домена> = <определение домена>. Используется с целью описания имен и семантики доменов, когда природа или структура объектов не может быть определена с помощью стандартных доменов.

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

В разделе PREDICATES описываются предикаты, их арифность (местность) и домены (типы и природа аргументов). С помощью описанных предикатов, можно создавать предложения в базе знаний. Предикаты используются для представления, как фактов,

так и правил.

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

На первом шаге работы происходит унификация вопроса и первого предложения базы знаний.

Алгоритм унификации необходим для того, чтобы подобрать знание, чтобы ответить на поставленный вопрос. Результатом работы алгоритма является значение переменной «неудача».

Если неудача = 1, то унификация невозможна;

если неудача = 0, то побочным действием работы алгоритма является содержимое результирующей ячейки –результирующая подстановка.

8. В каком случае запускается механизм отката?

В случае, когда унификация на текущем шаге завершается тупиковой ситуацией, или был получен ответ «да».