



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

по Лабораторной работе №13

по курсу «Функциональное и логическое программирование»

на тему: «Структура программы на Prolog»

Студент ИУ7-63Б
(Группа)

(Подпись, дата)

Миронов Г. А.
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Толшинская Н. Б.
(И. О. Фамилия)

2022 г.

1 Практическая часть

Задание 13: составить программу, то есть модель предметной области – базу знаний, объединив в ней информацию – знания:

- «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). Используя правила, обеспечить возможность поиска:

1. А. По № телефона найти: Фамилию, Марку автомобиля, Стоимость автомобиля (может быть несколько)
В. Используя сформированное в предыдущем пункте правило, по №телефона найти только Марку автомобиля (автомобилей может быть несколько)
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.

Для задания 1 и 2 для одного из вариантов ответов, и для А. и для В., описать словесно порядок поиска ответа на вопрос, указав, как выбираются знания, и, при этом, для каждого этапа унификации, выписать подстановку – наибольший общий унификатор, и соответствующие примеры термов.

Листинг 1.1 – Реализация базы знаний, Часть 1

```
1 domains
2   surname, phone, city, street = symbol.
3   home, flat, id, amount = integer.
4   model, color, bank = symbol.
5   address = address(city, street, home, flat).
6
7   price, size = integer.
8   property =
9       building(price, size);
10      sector(price, size);
11      ship(price, color);
12      car(model, color, price).
13
14 predicates
15   person(surname, phone, address).
16   bank_depositor(surname, bank, id, amount).
17   own(surname, property).
18   own_type(surname, symbol, price).
19   own_price(surname, symbol, price).
20   total_price(surname, price).
21
22 clauses
23   person("Andreev", "+79999999999", address("Moscow", "Lesnaya",
24       12, 2)).
25   person("Borisov", "+71111111111", address("Moscow", "Lesnaya",
26       13, 87)).
27   person("Dmitriev", "+73333333333", address("Ekaterinburg", "
28       Kamennaya", 13, 87)).
29   person("Fedorov", "+66666666666", address("Moscow", "Hospital'
30       naya", 123, 87)).
31
32   own("Andreev", car("bmw", "green", 1000)).
33   own("Borisov", sector(300000, 3000)).
34   own("Dmitriev", car("lada", "black", 20000)).
35   own("Dmitriev", ship(20000, "black")).
36   own("Fedorov", car("lada", "black", 20000)).
37   own("Fedorov", building(200000, 120)).
38
39   bank_depositor("Andreev", "Sber", 22, 1000).
40   bank_depositor("Borisov", "Sber", 33, 10000).
41   bank_depositor("Dmitriev", "Alfa", 44, 20000).
42   bank_depositor("Fedorov", "Sber", 238, 10).
43   bank_depositor("Borisov", "Maze", 1, 10000).
44
45   own_type(L, building, Price) :- own(L, building(Price, _)).
46   own_type(L, sector, Price) :- own(L, sector(Price, _)).
47   own_type(L, ship, Price) :- own(L, ship(Price, _)).
48   own_type(L, car, Price) :- own(L, car(_, _, Price)).
```

Листинг 1.2 – Реализация базы знаний, Часть 1

```
47 own_price(L, sector, Price) :- own(L, sector(Price, _)), !.  
48 own_price(L, ship, Price) :- own(L, ship(Price, _)), !.  
49 own_price(L, car, Price) :- own(L, car(_, _, Price)), !.  
50 own_price(_, _, 0).  
51  
52 total_price(L, Total) :-  
53     own_price(L, building, Price1),  
54     own_price(L, sector, Price2),  
55     own_price(L, ship, Price3),  
56     own_price(L, car, Price4),  
57     Total = Price1 + Price2 + Price3 + Price4.  
58  
59 goal  
60     % 1. Type of property by lastname  
61     %own_type("Fedorov", PropType, _).  
62  
63     % 2. Type of property and price by lastname  
64     own_type("Fedorov", PropType, Price).  
65     %own_type("Dmitriev", PropType, Price).  
66  
67     %3. Total price of property  
68     %total_price("Fedorov", TotalPrice).
```

В Таблице 1.1 представлен порядок поиска ответа на вопрос 2.

Таблица 1.1 – Порядок формирования результата для 1-го вопроса

| Шаг | Сравниваемые термины; результаты | Дальнейшие действия | Резольвента | Подстановка |
|-----------------------------------|--|---|--|--|
| 1 | own_price("Fedorov", PropType, Price) и person("Andreev", "+79999999999", addr("Moscow", "Lesnaya", 12, 2)). Главные функторы не равны | Прямой ход Переход к след. предл. | own_price("Fedorov", PropType, Price) | |
| ... | ... | ... | ... | ... |
| 20 | own_price("Fedorov", PropType, Price) и own_price(L, building, PropType) | Прямой ход | own("Fedorov", building(Price, _)) | L = "Fedorov" PropType = building |
| ... | ... | ... | ... | ... |
| 36 | own("Fedorov", building(Price, _)) и own("Fedorov", building(200000, 120)) | Нашли ответ Откат | | L = "Fedorov" PropType = building Price=200000 |
| ... | ... | ... | ... | ... |
| 51 | own_price("Fedorov", PropType, Price) и own_price(L, sector, PropType) | Прямой ход | own("Fedorov", sector(Price, _)) | L = "Fedorov" PropType = sector |
| ... | ... | ... | ... | ... |
| Продолжение на следующей странице | | | | |

Таблица 1.1 – продолжение

| Шаг | Сравниваемые термины; результаты | Дальнейшие действия | Результата | Подстановка |
|---------------|---|--|--|--|
| 77 | own_price("Fedorov", PropType, Price) и own_price(L, ship, PropType) | Прямой ход | own("Fedorov", ship(Price, _)) | L = "Fedorov" PropType = ship |
| ... | ... | ... | ... | ... |
| 103 | own_price("Fedorov", PropType, Price) и own_price(L, car, PropType) | Прямой ход | own("Fedorov", car(_, _, Price)) | L = "Fedorov" PropType = car |
| ... | ... | ... | ... | ... |
| 112 | own("Fedorov", car(_, _, Price)) и own("Fedorov", car("lada", "black", 20000)) | Нашли ответ Откат | | L = "Fedorov" PropType = car Price = 20000 |
| ... | ... | ... | ... | ... |
| 130 | own_price("Fedorov", PropType, Price) и total_price(L, Total)) Главные функторы не равны | Завершение работы 2 подст. в рез-те | own_price("Fedorov", PropType, Price) | |
| Конец таблицы | | | | |

2 Контрольный вопросы

2.1 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.

2.2 Что содержит тело правила?

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

2.3 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?

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.

2.4 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.

2.5 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.

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

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

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

Вопроса и первого терма в `clauses`.

2.8 Каковы назначение и результат использования алгоритма унификации?

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

2.9 В каком случае запускается механизм отката?

Механизм отката запустится в случае неудачи алгоритма унификации.