



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

*По предмету: «Функциональное и логическое
программирование»*

Студент: Лаврова А. А.,
Группа: ИУ7-65Б
Преподаватель: Толпинская Н. Б.
Строганов Ю. В.

Москва, 2020 г

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 T1 и T2 и полную подстановку на каждом шаге)

Листинг программы:

domains

```
surname = symbol.  
telephone = symbol.  
city = symbol.  
street = symbol.  
house = integer.  
address = address(city, street, house).
```

```

model = symbol.
color = symbol.
price = integer.

area = integer.

name_of_bank = symbol.
account = integer.
amount = integer.

property = automobile(model, color, price);
          building(area, price, city);
          territory(area, price, city);
          boat(model, color, price).

```

predicates

```

phonebook(surname, telephone, address).
bank(surname, name_of_bank, account, amount).
own(surname, property, city).

all_own(surname, string, price, city).
all_own(surname, string, city).

find_car(surname, price, city).
find_building(surname, price, city).
find_territory(surname, price, city).
find_boat(surname, price, city).

```

```

find_sum(surname, integer, city).

```

clauses

```

phonebook("Ivanov", "8111111111", address("Moscow", "Street1", 1)).
phonebook("Petrov", "8222222222", address("Moscow", "Street2", 2)).
phonebook("Sidorov", "8333333333", address("Kazan", "Street3", 3)).
phonebook("Vasechkin", "8444444444", address("Chelyabinsk",
"Street4", 4)).
phonebook("Lolaev", "8555555555", address("Zlatoust", "Street5",
5)).

bank("Lolaev", "HomeCreditBank", 100, 100).
bank("Ivanov", "Sberbank", 200, 200).
bank("Vasechkin", "VTB", 300, 300).
bank("Lolaev", "Pochta Bank", 400, 400).

own("Ivanov", automobile("Kia", "red", 1550000), "Moscow").
own("Ivanov", automobile("Lada", "grey", 1000000), "Moscow").
own("Vasechkin", automobile("Land Rover", "black", 5000000),
"Chelyabinsk").
own("Sidorov", automobile("Honda", "white", 2000000), "Kazan").

own("Ivanov", building(100, 9000000, "Moscow"), "Moscow").
own("Petrov", building(35, 3000000, "Moscow"), "Moscow").
own("Sidorov", building(50, 2500000, "Kazan"), "Kazan").
own("Vasechkin", building(70, 2000000, "Chelyabinsk"),
"Chelyabinsk").
own("Lolaev", building(46, 500000, "Zlatoust"), "Zlatoust").

```

```

    own("Ivanov", territory(500, 3000000, "Moscow"), "Moscow").
    own("Sidorov", territory(150, 1000000, "Kazan"), "Kazan").
    own("Vasechkin", territory(100, 700000, "Chelyabinsk"),
"Chelyabinsk").
    own("Lolaev", territory(100, 100000, "Zlatoust"), "Zlatoust").

    own("Petrov", boat("Model1", "green", 100000), "Moscow").
    own("Sidorov", boat("Model2", "white", 70000), "Kazan").
    own("Lolaev", boat("Model3", "black", 50000), "Zlatoust").

    all_own(Surname, Property, Price, City) :- own(Surname, automobile(_,
_, Price), City), Property = "Automobile".
    all_own(Surname, Property, Price, City) :- own(Surname, building(_,
Price, _), City), Property = "Building".
    all_own(Surname, Property, Price, City) :- own(Surname, territory(_,
Price, _), City), Property = "Territory".
    all_own(Surname, Property, Price, City) :- own(Surname, boat(_, _,
Price), City), Property = "Boat".
    all_own(Surname, Property, City) :- all_own(Surname, Property, _,
City).

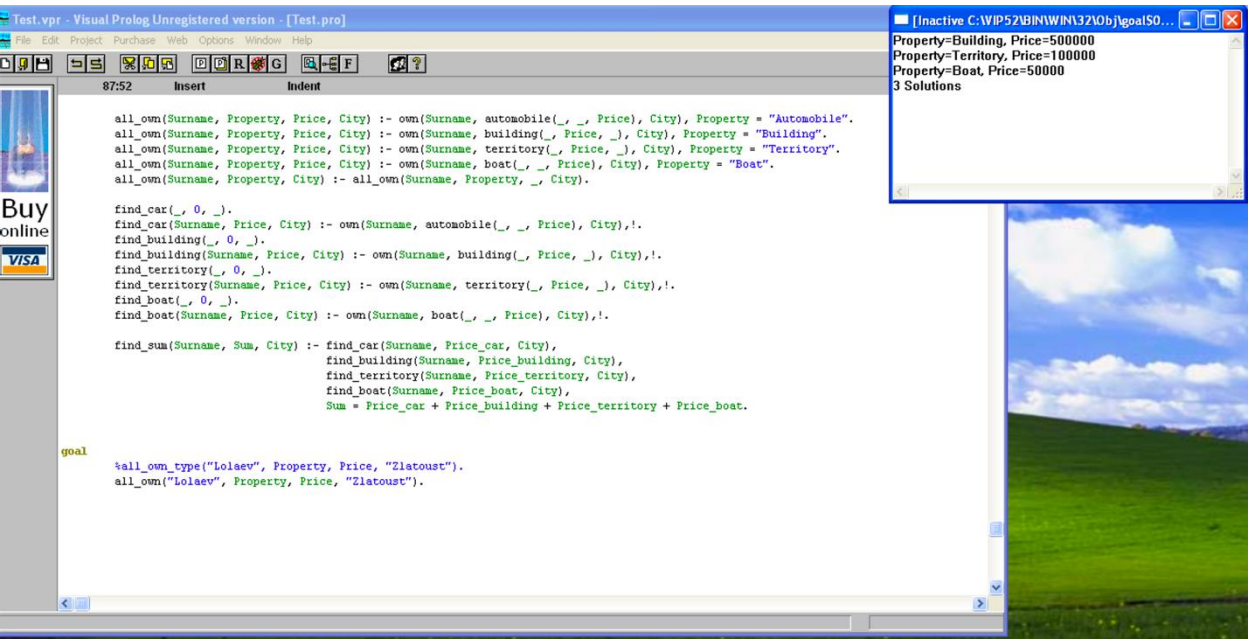
    find_car(_, 0, _).
    find_car(Surname, Price, City) :- own(Surname, automobile(_, _,
Price), City),!.
    find_building(_, 0, _).
    find_building(Surname, Price, City) :- own(Surname, building(_,
Price, _), City),!.
    find_territory(_, 0, _).
    find_territory(Surname, Price, City) :- own(Surname, territory(_,
Price, _), City),!.
    find_boat(_, 0, _).
    find_boat(Surname, Price, City) :- own(Surname, boat(_, _, Price),
City),!.

    find_sum(Surname, Sum, City) :- find_car(Surname, Price_car, City),
                                find_building(Surname, Price_building, City),
                                find_territory(Surname, Price_territory, City),
                                find_boat(Surname, Price_boat, City),
                                Sum = Price_car + Price_building +
Price_territory + Price_boat.

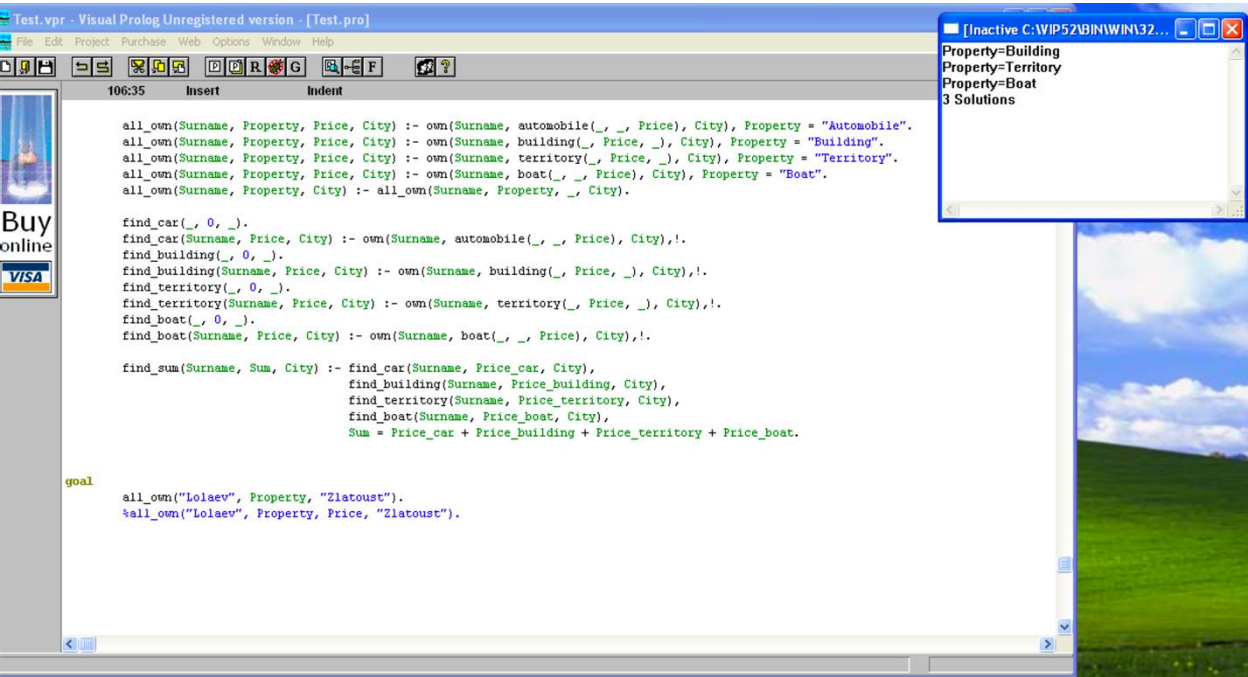
goal
    all_own("Lolaev", Property, "Zlatoust").
    %all_own("Lolaev", Property, Price, "Zlatoust").

```

Примеры работы программы:



Вопрос: Какой собственностью и по какой стоимости владеет Лолаев из Златоуста?



Вопрос: какой собственностью владеет Лолаев из Златоуста?

Порядок поиска ответа на вопрос all_own("Lolaev", Property, Price, "Zlatoust").

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
--------	---	--

1	<p>Попытка унификации all_own("Lolaev", Property, Price, "Zlatoust").</p> <p>и</p> <p>phonebook("Ivanov", "8111111111", address("Moscow", "Street1", 1)).</p> <p>Неудача</p>	Прямой ход
	... (ряд неудачных попыток)	...
2	<p>Попытка унификации all_own("Lolaev", Property, Price, "Zlatoust").</p> <p>и</p> <p>all_own(Surname, Property, Price, City)</p> <p>Удача Результат: all_own("Lolaev", Property, Price, "Zlatoust").</p>	Прямой ход, Унификация Surname = "Lolaev", City = "Zlatoust", Property = "Automobile"
3	<p>Попытка унификации own("Lolaev", automobile(_, _, Price), "Zlatoust)</p> <p>и</p> <p>phonebook("Ivanov", "8111111111", address("Moscow", "Street1", 1)).</p> <p>Неудача</p>	Прямой ход
	... (ряд неудачных попыток)	...
4	<p>Попытка унификации own("Lolaev", automobile(_, _, Price), "Zlatoust)</p> <p>и</p> <p>find_sum(Surname, Sum, City)</p> <p>Неудача</p>	Откат до all_own("Lolaev", Property, Price, "Zlatoust").
5	<p>Попытка унификации own("Lolaev", automobile(_, _, Price), "Zlatoust)</p> <p>и</p>	Прямой ход, Унификация Surname = "Lolaev", City = "Zlatoust", Property = "Building"

	<p>all_own(Surname, Property, Price, City)</p> <p>Удача Результат: all_own("Lolaev", Property, Price, "Zlatoust").</p>	
6	<p>Попытка унификации own("Lolaev", building(_, _, Price), ""Zlatoust) и phonebook("Ivanov", "8111111111", address("Moscow", "Street1", 1)).</p> <p>Неудача</p>	Прямой ход
	... (ряд неудачных попыток)	...
7	<p>Попытка унификации own("Lolaev", building(_, _, Price), ""Zlatoust) и own("Lolaev", building(46, 500000, "Zlatoust"), "Zlatoust").</p> <p>Удача Результат: own("Lolaev", building(_, 500000, _), ""Zlatoust") Выводим Building, Price = 500000</p>	<p>Прямой ход Унификация Price = 500000</p>
	... (ряд неудачных попыток)
8	<p>Попытка унификации own("Lolaev", building(_, _, Price), ""Zlatoust) и find_sum(Surname, Sum, City)</p> <p>Неудача</p>	Откат до all_own("Lolaev", Property, Price, "Zlatoust").
9	<p>Попытка унификации own("Lolaev", automobile(_, _, Price), ""Zlatoust) и</p>	<p>Прямой ход, Унификация Surname = "Lolaev", City = "Zlatoust", Property = "Territory"</p>

	all_own(Surname, Property, Price, City) Удача Результат: all_own("Lolaev", Property, Price, "Zlatoust").	
	... (ряд неудачных попыток)	...
10	Попытка унификации own("Lolaev", building(_, _, Price), ""Zlatoust) и own("Lolaev", territory(100, 100000, "Zlatoust"), "Zlatoust"). Удача Результат: own("Lolaev", building(_, 100000, _), ""Zlatoust") Выводим Territory, Price = 100000	Прямой ход Унификация Price = 100000
	... (ряд неудачных попыток)	...
11	Попытка унификации own("Lolaev", building(_, _, Price), ""Zlatoust) и find_sum(Surname, Sum, City) Неудача	Откат до all_own("Lolaev", Property, Price, "Zlatoust").
12	Попытка унификации own("Lolaev", automobile(_, _, Price), ""Zlatoust) и all_own(Surname, Property, Price, City) Удача Результат: all_own("Lolaev", Property, Price, "Zlatoust").	Прямой ход, Унификация Surname = "Lolaev", City = "Zlatoust", Property = "Boat"
	... (ряд неудачных попыток)	...
13	Попытка унификации	Прямой ход

	own("Lolaev", building(_, _, Price), ""Zlatoust) и own("Lolaev", boat("Model3", "black", 50000), "Zlatoust"). Удача Результат: own("Lolaev", boat(_, _, 50000), "Zlatoust") Выводим Territory, Boat = 50000	Унификация Price = 50000
	... (ряд неудачных попыток)	...
14	Попытка унификации own("Lolaev", building(_, _, Price), ""Zlatoust) и find_sum(Surname, Sum, City) Неудача	Откат до all_own("Lolaev", Property, Price, "Zlatoust").
	... (ряд неудачных попыток)	...
		Дошли до конца БЗ. Завершение программы.

Теоретическая часть

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

Знание сформулировано в заголовке правила. Содержит знание о том, что между аргументами существует отношение.

2) Что содержит тело правила?

Тело содержит условия истинности знания

3) Что дает использование переменных при формулировании знаний?

В чем отличие формулировки знания с помощью термов с одинаковой арифметичностью при использовании одной переменной и при использовании нескольких переменных?

Переменные предназначены для передачи значений во времени и в пространстве. Формулировка знаний с использованием переменных носит более общий характер по отношению к знанию, состоящему только лишь из констант. Терм при использовании нескольких переменных более общий, чем при использовании одной переменной.

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

Переменные входят в правило с квантором всеобщности. Именованные переменные уникальны в рамках предложения, а любая анонимная переменная уникальна.

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

Предложения в разделе DOMAINS используются для объявления используемых доменов, не являющимися стандартными доменами в Prolog. Раздел доменов используется для описания структур (вариантных доменов).

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

Если природа или структура объектов, обозначенных аргументами, между которыми устанавливается отношение в заголовке правил процедуры, важна во время работы системы, то она должна быть указана в разделе PREDICATES с помощью соответствующих доменов.

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

На первом шаге запускается унификация вопроса и первого предложения. Алгоритм унификации необходим для формализации процесса логического вывода и подбора подходящего знания. В результате алгоритма унификации мы выясняем, является ли вопрос логическим следствием программы или нет, и в случае, если является, получаем подстановку. Результат унификации – ответ «да» или «нет».

8) В каком случае запускается механизм отката?

В случае попадания в тупиковую ситуацию или в случае, если Prolog ответил на вопрос и пытается найти другие ответы.