



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»*

Преподаватель: Строганов Ю.В.,
Студент: Мирзоян С.А.,
Группа: ИУ7-65Б

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)

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1...	-попытка унификации: T1=T2 ... -результат: Успех и подстановка, или Нет	Комментарий, вывод...

При желании, можно усложнить свою базу знаний, введя варианты: **строение**: (Дом, офис, торговый центр), **участок**: (садовый, территория под застройку, территория под агро-работы), **Водный_транспорт**: варианты названий.

Теоретические вопросы.

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

Знания сформулированы в CLAUSES в виде предложений. Эти предложения - знания о некоторой предметной области.

На формальном уровне – это отношения между различными объектами.

- 2. Что содержит тело правила?**

Содержит условие истинности заголовка правила.

- 3. 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?**

Использование переменных в формулировании знаний позволяют уточнять значения и переносить их. Формулировка знаний с использованием переменных носит более общий характер по отношению к знанию, состоящему только из констант. Использование знаний с одинаковой арностью при использовании одной переменной носит менее общий характер относительно отношению к знанию с использованием нескольких переменных.

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

С квантором \forall («для любого»).

Уникальность переменных:

Именованные переменные - в пределах одного предложения.

Анонимные - везде.

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

Предложения в разделе DOMAINS используются для объявления доменов, не являющимися стандартными 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?

В разделе PREDICATES описываются предикаты, их аргументность (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?

Первым шагом является унификация вопроса и первого предложения базы знаний. Алгоритм унификации попарно сопоставляет термы, пытаясь построить для них общий пример. Назначение - подбор нужного в данный момент правила. Унификация может завершаться успехом или неудачей (тупиковой ситуацией).

8. В каком случае запускается механизм отката?

Если достигнут конец БЗ (тупиковая ситуация) или резольвента пуста. После этого происходит откат к предыдущему состоянию резольвенты.

Текст программы

```
1. domains
2.
3.     property =
4.         auto own(symbol Brand, symbol Color, integer Cost);
5.         building own(symbol Type, integer Floors, integer Cost);
6.         land own(symbol Type, integer Plottage, integer Cost);
7.         water transport own(symbol Type, integer Cost).
8.
9.     address = address(symbol City, symbol Street, integer, integer).
10.
11.     predicates
12.         person(symbol Lastname, symbol Number, address)
13.         auto(symbol Lastname, symbol Brand, symbol Color, integer
            Cost)
14.         deposit(symbol Lastname, symbol Bank, integer client_check,
            integer Sum)
15.
16.         owner(symbol Lastname, property Property)
17.
```

```

18.         find_property(symbol Lastname, symbol Propername, integer
           Cost)

19.     clauses

20.         person(ivanov, "321-777", address(moscow, ilmenskiy, 14,
           1018)).

21.         person(petrov, "378-770", address(pyatigorsk, lenina, 73,
           527)).

22.         person(sidorov, "983-683", address(moscow, tverskaya, 103,
           17)).

23.         person(johnson, "596-055", address(san_andreas, groove, 10,
           7)).

24.

25.         auto(ivanov, bmw, white, 7800000).

26.         auto(petrov, volvo, red, 1300000).

27.         auto(sidorov, audi, green ,1600000).

28.         auto(johnson, comet, green, 50000).

29.

30.         deposit(ivanov, vtb, 00232973, 300000).

31.         deposit(petrov, sberbank, 00101023, 36000).

32.         deposit(sidorov, alfabank, 00014584, 200000).

33.         deposit(johnson, vivaldi, 98761234, 12356531).

34.

35.

36.         owner(ivanov, building_own(supermarket, 1, 20000000)).

37.         owner(petrov, building_own(townhouse, 3, 15000000)).

38.         owner(petrov, land_own(field, 40, 4000000)).

39.         owner(petrov, water_transport_own(hydrocycle, 200000)).

40.         owner(johnson, land_own(mamas_home, 54, 100000)).

41.         owner(johnson, water_transport_own(ship, 33400)).

42.         owner(johnson, building_own(casino_las_venturas, 4,
           1120000)).

43.

44.

45.         owner(Lastname, auto_own(Brand, Color, Cost)):-
           auto(Lastname, Brand, Color, Cost).

46.         find_property(Lastname, Brand, Cost):-owner(Lastname,
           auto_own(Brand, _, Cost)).

47.         find_property(Lastname, Type, Cost):-owner(Lastname,
           building_own(Type, _, Cost)).

```

```

48.         find_property(Lastname, Type, Cost):-owner(Lastname,
land_own(Type, _, Cost)).
49.         find_property(Lastname, Type, Cost):-owner(Lastname,
water_transport_own(Type, Cost)).
50.     goal
51.     %find_property(johnson, Propername, _).
52.     %find_property(petrov, Propername, Cost).

```

Примеры работы программы

Задание 1. Поиск названий всех объектов собственности заданного субъекта.

find_property(johnson, Propername, _).

```

Propername=comet
Propername=casino_las_venturas
Propername=mamas_home
Propername=ship
4 Solutions

```

Задание 2. Поиск названий и стоимости всех объектов собственности заданного субъекта.

find_property(johnson, Propername, Cost).

```

Propername=comet, Cost=50000
Propername=casino_las_venturas, Cost=1120000
Propername=mamas_home, Cost=100000
Propername=ship, Cost=33400
4 Solutions

```

Описание порядка поиска объектов

Поиск названий и стоимости всех объектов собственности заданного субъекта.

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
	<p>Попытка унификации:</p> <pre>find_property(petrov, Own_name, Cost) = person(ivanov, "321-777", address(moscow, ilmenskiy, 14, 1018))</pre> <p>Результат: Неудача (разные функторы).</p>	Прямой ход, переход к следующему предложению
	<p>Попытка унификации:</p> <pre>find_property(petrov, Own_name, Cost) = person(petrov, "378-770", address(pyatigosk, lenina, 73, 527))</pre> <p>Результат: Неудача (разные функторы).</p>	Прямой ход, переход к следующему предложению

	<p>Попытка унификации: <code>find_property(petrov, Own_name, Cost)</code> <code>=</code> <code>person(sidorov, "983-683", address(moscow, tverskaya, 103, 17))</code></p> <p>Результат: Неудача (разные функторы).</p>	Прямой ход, переход к следующему предложению
	<p>Попытка унификации: <code>find_property(petrov, Own_name, Cost)</code> <code>=</code> <code>person(vasilev, "596-055", address(samara, lesnaya, 10, 7))</code></p> <p>Результат: Неудача (разные функторы).</p>	Прямой ход, переход к следующему предложению
	<p>Попытка унификации: <code>find_property(petrov, Own_name, Cost)</code> <code>auto(ivanov, bmw, white, 7800000)</code></p> <p>Результат: Неудача (разные функторы).</p>	Прямой ход, переход к следующему предложению
	<p>Попытка унификации: <code>find_property(petrov, Own_name, Cost)</code> <code>auto(petrov, volvo, red, 1300000)</code></p> <p>Результат: Неудача (разные функторы).</p>	Прямой ход, переход к следующему предложению
	<p>Попытка унификации: <code>find_property(petrov, Own_name, Cost)</code> <code>auto(sidorov, audi, green ,1600000)</code></p> <p>Результат: Неудача (разные функторы).</p>	Прямой ход, переход к следующему предложению
	<p>Попытка унификации: <code>find_property(petrov, Own_name, Cost)</code> <code>deposit(ivanov, vtb, 00232973, 300000)</code></p> <p>Результат: Неудача (разные функторы).</p>	Прямой ход, переход к следующему предложению
	<p>Попытка унификации: <code>find_property(petrov, Own_name, Cost)</code> <code>deposit(petrov, sberbank, 00101023, 36000)</code></p> <p>Результат: Неудача (разные функторы).</p>	Прямой ход, переход к следующему предложению
	<p>Попытка унификации: <code>find_property(petrov, Own_name, Cost)</code> <code>deposit(sidorov, alfabank, 00014584, 200000)</code></p> <p>Результат: Неудача (разные функторы).</p>	Прямой ход, переход к следующему предложению
	<p>Попытка унификации: <code>find_property(petrov, Own_name, Cost)</code> <code>=</code> <code>owner(Surname, auto_own(Brand, Color, Cost))</code></p> <p>Результат: Неудача (разные функторы).</p>	Прямой ход, переход к следующему предложению
...	<p>Производится попытка унификации со всеми предложениями <code>owner(...)</code></p> <p>Результат: Неудача (разные функторы).</p>	Прямой ход, переход к следующему предложению

12	<p>Попытка унификации: <code>find_property(petrov, Own_name, Cost)</code></p> <p><code>find_property(Surname, Brand, Cost)</code> Результат: Успех (выполняется подстановка) <code>{ Surname = petrov, Brand = Own_name, Cost = Cost}</code>).</p>	<p>Прямой ход. Сравнение начинается с начала с предложением <code>owner(petrov, auto_own(Own_name, _, Cost))</code>.</p>
3	<p>Попытка унификации: <code>owner(petrov, auto_own(Own_name, _, Cost))</code> <code>=</code> <code>person(ivanov, "321-777", address(moscow, ilmenskiy, 14, 1018))</code></p> <p>Результат: Неудача (разные функторы).</p>	<p>Прямой ход, переход к следующему предложению</p>
	<p>Производится попытка унификации со всеми предложениями <code>person(...)</code> Результат: Неудача (разные функторы).</p>	<p>Прямой ход, переход к следующему предложению</p>
14	<p>Попытка унификации: <code>owner(petrov, auto_own(Own_name, _, Cost))</code></p> <p><code>auto(ivanov, bmw, white, 7800000)</code> Результат: Неудача (разные функторы).</p>	<p>Прямой ход, переход к следующему предложению</p>
	<p>Производится попытка унификации со всеми предложениями <code>auto(...)</code> Результат: Неудача (разные функторы).</p>	<p>Прямой ход, переход к следующему предложению</p>
15	<p>Попытка унификации: <code>owner(petrov, auto_own(Own_name, _, Cost))</code></p> <p><code>deposit(ivanov, vtb, 00232973, 300000)</code> Результат: Неудача (разные функторы).</p>	<p>Прямой ход, переход к следующему предложению</p>
...	<p>Производится попытка унификации со всеми предложениями <code>deposit(...)</code> Результат: Неудача (разные функторы).</p>	<p>Прямой ход, переход к следующему предложению</p>
16	<p>Попытка унификации: <code>owner(petrov, auto_own(Own_name, _, Cost))</code> <code>=</code> <code>owner(Surname, auto_own(Brand, Color, Cost))</code> Результат: Успех (выполняется подстановка) <code>{ Surname = petrov, Brand = Own_name, Cost = Cost}</code>).</p>	<p>Прямой ход. Сравнение начинается с начала с предложением <code>auto(petrov, Own_name, _, Cost)</code>.</p>
17	<p>Попытка унификации: <code>auto(petrov, Own_name, _, Cost)</code> <code>=</code> <code>person(ivanov, "321-777", address(moscow, ilmenskiy, 14, 1018))</code></p> <p>Результат: Неудача (разные функторы).</p>	<p>Прямой ход, переход к следующему предложению</p>
	<p>Производится попытка унификации со всеми предложениями <code>person(...)</code> Результат: Неудача (разные функторы).</p>	<p>Прямой ход, переход к следующему предложению</p>

18	<p>Попытка унификации: <code>auto(petrov, Own_name, _, Cost) .</code> <code>=</code> <code>auto(ivanov, bmw, white, 7800000) .</code> Результат: Неудача (<code>petrov <> ivanov</code>).</p>	Прямой ход, переход к следующему предложению
19	<p>Попытка унификации: <code>auto(petrov, Own_name, _, Cost) .</code> <code>auto(petrov, volvo, red, 1300000) .</code> Результат: Успех (выполняется подстановка)</p>	Вывод: <code>Own_name = volvo,</code> <code>Cost = 1300000</code> Откат к <code>auto(petrov, Own_name, _, Cost) .</code>
20	<p>Попытка унификации: <code>auto(petrov, Own_name, _, Cost) .</code> <code>=</code> <code>auto(sidorov, audi, green, 1600000) .</code> Результат: Неудача (<code>petrov <> sidorov</code>).</p>	Прямой ход, переход к следующему предложению
...	<p>Производится попытка унификации со всеми предложениями <code>deposit(...)</code>, <code>owner(...)</code> и <code>find_property(...)</code> Результат: Неудача (разные функторы).</p>	Откат к шагу 16
21	<p>Попытка унификации: <code>owner(petrov, auto_own(Own_name, _, Cost))</code> <code>=</code> <code>owner(ivanov, building_own(supermarket, 1, 20000000))</code> Результат: Неудача (<code>petrov <> ivanov</code>).</p>	Прямой ход, переход к следующему предложению
...	<p>Производится попытка унификации со всеми предложениями <code>owner(...)</code> и <code>find_property(...)</code> Результат: Неудача</p>	Откат к шагу 12
22	<p>Попытка унификации: <code>find_property(petrov, Own_name, Cost)</code> <code>find_property(Surname, Type, Cost)</code> Результат: Успех (выполняется подстановка) <code>{ Surname = petrov, Type = Own_name, Cost = Cost }.</code></p>	Прямой ход. Сравнение начинается с начала с предложением <code>owner(petrov, building_own(Own_name, _, Cost)) .</code>
23	<p>Попытка унификации: <code>owner(petrov, building_own(Own_name, _, Cost))</code> <code>=</code> <code>person(ivanov, "321-777", address(moscow, ilmenskiy, 14, 1018))</code> Результат: Неудача (разные функторы).</p>	Прямой ход, переход к следующему предложению
...	<p>Производится попытка унификации со всеми предложениями <code>person(...)</code>, <code>auto(...)</code> и <code>deposit(...)</code> Результат: Неудача (разные функторы).</p>	Прямой ход, переход к следующему предложению
24	<p>Попытка унификации: <code>owner(petrov, building_own(Own_name, _, Cost))</code> <code>=</code> <code>owner(Surname, auto_own(Brand, Color, Cost))</code> Результат: Неудача (<code>building_own <> auto_own</code>).</p>	Прямой ход, переход к следующему предложению

25	<p>Попытка унификации: <code>owner(petrov, building_own(Own_name, _, Cost))</code> <code>=</code> <code>owner(ivanov, building_own(supermarket, 1, 20000000))</code> Результат: Неудача (<code>petrov <> ivanov</code>).</p>	Прямой ход, переход к следующему предложению
26	<p>Попытка унификации: <code>owner(petrov, building_own(Own_name, _, Cost))</code> <code>=</code> <code>owner(petrov, building_own(townhouse, 3, 15000000))</code> Результат: Успех (выполняется подстановка)</p>	Вывод: <code>Own_name = townhouse,</code> <code>Cost = 15000000</code> Откат к <code>owner(petrov, building_own(Own_name, _, Cost))</code>
...	<p>Попытка унификации: Результат: Неудача при сравнении со следующими термами до конца БЗ.</p>	Откат к 22
27	<p>Попытка унификации: <code>find_property(petrov, Own_name, Cost)</code> <code>find_property(Surname, Type, Cost)</code> Результат: Успех (выполняется подстановка) <code>{ Surname = petrov, Type = Own_name, Cost = Cost}</code>.</p>	Прямой ход. Сравнение начинается с начала с предложением <code>owner(petrov, land_own(Own_name, _, Cost))</code> .
28	<p>Попытка унификации: <code>owner(petrov, land_own(Own_name, _, Cost))</code> <code>=</code> <code>person(ivanov, "321-777", address(moscow, ilmenskiy, 14, 1018))</code>. Результат: Неудача (разные функторы).</p>	Прямой ход, переход к следующему предложению
...	<p>Производится попытка унификации со всеми предложениями <code>person(...)</code>, <code>auto(...)</code> и <code>deposit(...)</code> Результат: Неудача (разные функторы).</p>	Прямой ход, переход к следующему предложению
29	<p>Попытка унификации: <code>owner(petrov, land_own(Own_name, _, Cost))</code> <code>=</code> <code>owner(Surname, auto_own(Brand, Color, Cost))</code> Результат: Неудача (<code>land_own <> auto_own</code>).</p>	Прямой ход, переход к следующему предложению
30	<p>Попытка унификации: <code>owner(petrov, land_own(Own_name, _, Cost))</code> <code>=</code> <code>owner(ivanov, building_own(supermarket, 1, 20000000))</code> Результат: Неудача (<code>land_own <> building_own</code>).</p>	Прямой ход, переход к следующему предложению
31	<p>Попытка унификации: <code>owner(petrov, land_own(Own_name, _, Cost))</code> <code>=</code> <code>owner(petrov, land_own(field, 40, 4000000))</code> Результат: Успех (выполняется подстановка)</p>	Вывод: <code>Own_name = field,</code> <code>Cost = 4000000</code> Откат к <code>owner(petrov, land_own(Own_name, _, Cost))</code>
...	<p>Попытка унификации: Результат: Неудача при сравнении со следующими термами до конца БЗ.</p>	Откат к 27
32	<p>Попытка унификации: <code>find_property(petrov, Own_name, Cost)</code> <code>find_property(Surname, Type, Cost)</code></p>	Прямой ход. Сравнение начинается с начала с предложением <code>owner(petrov,</code>

	Результат: Успех (выполняется подстановка) { Surname = petrov, Type = Own_name, Cost = Cost}.	water_transport_own(Own_name, Cost)).
33	Попытка унификации: owner(petrov, water_transport_own(Own_name, Cost)) = person(ivanov, "321-777", address(moscow, ilmenskiy, 14, 1018)) Результат: Неудача (разные функторы).	Прямой ход, переход к следующему предложению
...	Производится попытка унификации со всеми предложениями person(...), auto(...) и deposit(...) Результат: Неудача (разные функторы).	Прямой ход, переход к следующему предложению
34	Попытка унификации: owner(petrov, water_transport_own(Own_name, Cost)) = owner(Surname, auto_own(Brand, Color, Cost)) Результат: Неудача (water_transport_own <> auto_own).	Прямой ход, переход к следующему предложению
35	Попытка унификации: owner(petrov, water_transport_own(Own_name, Cost)) = owner(ivanov, building_own(supermarket, 1, 20000000)) Результат: Неудача (petrov <> ivanov).	Прямой ход, переход к следующему предложению
36	Попытка унификации: owner(petrov, water_transport_own(Own_name, Cost)) = owner(petrov, building_own(townhouse, 3, 15000000)) Результат: Неудача (water_transport_own <> building_own).	Прямой ход, переход к следующему предложению
37	Попытка унификации: owner(petrov, water_transport_own(Own_name, Cost)) = owner(petrov, land_own(field, 40, 4000000)) Результат: Неудача (water_transport_own <> land_own).	Прямой ход, переход к следующему предложению
38	Попытка унификации: owner(petrov, water_transport_own(Own_name, Cost)) = owner(petrov, water_transport_own(hydrocycle, 200000)) Результат: Успех (выполняется подстановка)	Вывод: Own_name = hydrocycle, Cost = 4000000 Откат к owner(petrov, water_transport_own(Own_name, Cost))
...	Попытка унификации: owner(petrov, water_transport_own(Own_name, Cost)) и find_property(...) Результат: Неудача (разные функторы).	Прямой ход, переход к следующему предложению.
39	Попытка унификации: owner(petrov, water_transport_own(Own_name, Cost)) и find_property(Surname, Type, Cost) Результат: Неудача (разные функторы).	Откат к шагу 32, а затем достижение конца БЗ. Завершение программы.