



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 № 5

По дисциплине «Функциональное и логическое программирование»

Студент: Тимонин А. С.

Группа ИУ7-626

Преподаватель Толпинская Н. Б.

Москва.
2020 г.

Практическая часть

Создать базу знаний «Собственники», дополнив базу знаний, хранящую знания (лаб. 13):

- «Телефонный справочник»: Фамилия, Но телефона, Адрес – структура (Город, Улица, Но дома, Но 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 Но 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)

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.**

Листинг 1. Реализация программы

```
domains
    lastname, phone, city, street = symbol.
    house, flat, area, acres, price = integer.
    typeOfOwn, type, brand, color = symbol.
    bank = symbol.
    account, amount, sumPrice = integer.
    address = address(city, street, house, flat).
    ownership = car(brand, color, price);
                building(area, type, price);
                plot(acres, type, price);
                boat(brand, type, color, price).

predicates
    nondeterm phone_notice(lastname, phone, address).
    nondeterm bank_investors(lastname, city, bank, account, amount).
    nondeterm owner(lastname, city, ownership).

    nondeterm allOwners(lastname, city, typeOfOwn, price).
    nondeterm allOwners(lastname, city, typeOfOwn).
    nondeterm priceCar(lastname, city, price).
    nondeterm priceBuilding(lastname, city, price).
    nondeterm pricePlot(lastname, city, price).
    nondeterm priceBoat(lastname, city, price).
    nondeterm summaryPrice(lastname, city, sumPrice).

clauses
    phone_notice("Timonin", "7090898286725", address("Moscow", "Sovetskaya", 2,
37)).
    phone_notice("Ibragimov", "7090898286725", address("Vologda", "Vostochnaya", 5,
11)).
    phone_notice("Timonin", "434451374707", address("neMoscow", "Central", 4, 37)).
    phone_notice("Sardyn", "88418172007714", address("Orsk", "Zapandanya", 4, 33)).
    phone_notice("Savelko", "0180596669", address("Archangelsk", "Central", 7, 64)).
    phone_notice("Foreman", "510435028126", address("Orenburg", "Krygovaya", 4,
37)).
    phone_notice("Checkiki", "4104222849519", address("Archangelsk", "Vostochnaya",
4, 23)).
    phone_notice("Lapriyal", "66013578651255", address("Vologda", "Svetskaya", 2,
90)).
    phone_notice("Koverap", "5782027078", address("Dnepr", "Krygovaya", 4, 73)).
    phone_notice("Kolopyn", "5562794360972", address("Pripyat", "Zone", 1, 61)).
    phone_notice("Nebrogov", "79875467322", address("Rytia", "Mainernce", 1, 22)).

    bank_investors("Timonin", "Moscow", "VTB", 900, 1250).
    bank_investors("Timonin", "neMoscow", "SBERBANK", 2000, 13500).
    bank_investors("Koverap", "Dnepr", "TINKOFF", 2000, 3700).
```

```

bank_investors("Koverap", "neDnepr", "SBERBANK", 2100, 3600).
bank_investors("Foreman", "Orenburg", "SBERBANK", 1000, 1300).
bank_investors("Sardyn", "Orksk", "TINKOFF", 71000, 101).
bank_investors("Kolopyn", "Pripyat", "VTB", 1000, 12).
bank_investors("Lapriyal", "Vologda", "SBERBANK", 11000, 111).
bank_investors("Joker", "Acrham", "TINKOFF", 100, 130).
bank_investors("Lapriyal", "Vologda", "TINKOFF", 134000, 6000000).
bank_investors("Ibragimov", "Vologda", "VTB", 10, 250).

%owner(lastname, city, ownership)
owner("Timonin", "Moscow", car("BMW", "red", 50000)).
owner("Timonin", "Vologda", building(175, "Real", 125000)).
owner("Timonin", "Moscow", building(175, "Real", 125000)).
owner("Timonin", "Moscow", boat("BMX", "123", "Red", 1)).
owner("Ibragimov", "Vologda", plot(200, "Real", 12342)).
owner("Sardyn", "Orksk", building(213, "Unreal", 11111)).
owner("Koverap", "Dnepr", boat("BMX", "123", "Red", 1290)).
owner("Nebrogov", "Rytia", boat("KOL", "Real", "Green", 90000)).

allOwners(Lastname, City, Ownership, Price) :-
    owner(Lastname, City, car(_, _, Price)), Ownership = "Car".
allOwners(Lastname, City, Ownership, Price) :-
    owner(Lastname, City, building(_, _, Price)), Ownership =
"Building".
allOwners(Lastname, City, Ownership, Price) :-
    owner(Lastname, City, plot(_, _, Price)), Ownership = "Plot".
allOwners(Lastname, City, Ownership, Price) :-
    owner(Lastname, City, boat(_, _, Price)), Ownership = "Boat".

allOwners(Lastname, City, Ownership) :-
    allOwners(Lastname, City, Ownership, _).

priceCar(Lastname, City, Price) :- owner(Lastname, City, car(_, _, Price)),!.
priceCar(_, _, 0).

priceBuilding(Lastname, City, Price) :- owner(Lastname, City, building(_, _,
Price)),!.
priceBuilding(_, _, 0).

pricePlot(Lastname, City, Price) :- owner(Lastname, City, plot(_, _, Price)),!.
pricePlot(_, _, 0).

priceBoat(Lastname, City, Price) :- owner(Lastname, City, boat(_, _, _,
Price)),!.
priceBoat(_, _, 0).

summaryPrice(Lastname, City, SumPrice) :-
    priceCar(Lastname, City, CarPrice),
    priceBuilding(Lastname, City, BuildingPrice),
    pricePlot(Lastname, City, PlotPrice),
    priceBoat(Lastname, City, BoatPrice),
    SumPrice = CarPrice + BuildingPrice + PlotPrice + BoatPrice.


goal
allOwners("Timonin", "Moscow", Ownership).

%summaryPrice("Timonin", City, SumPrice).

```


```
Ownership=Car  
Ownership=Building  
Ownership=Boat  
3 Solutions|
```

Рисунок 1. allOwners("Timonin", "Moscow", Ownership).

 [Inactive C:\Users\ANTONT~1\AppData\Local\Temp\goal\$000.exe]

```
Ownership=Car, Cost=50000  
Ownership=Building, Cost=125000  
Ownership=Boat, Cost=1  
3 Solutions|
```

Рисунок 2. allOwners("Timonin", "Moscow", Ownership, Cost).

 [Inactive C:\Users\ANTONT~1\AppData\

```
City=Moscow, SumPrice=175001  
1 Solution|
```

Рисунок 3. summaryPrice("Timonin", City, SumPrice).

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	Подстановка Lastname = «Timonin», City = «Moscow» allOwners("Timonin", "Moscow", Ownership, Cost) allOwners(Lastname, City, Ownership, Cost)	Прямой ход
2	Сравнение «Timonin» = «Timonin», «Moscow» = «Moscow» owner("Timonin", "Moscow", car("BMW", "red", 50000)) owner(Lastname, City, car(_, _, Price)) Успех	Прямой ход
3	Подстановка: Cost = 50000, Ownership = «Car»	Прямой ход
4	Результат: Ownership = Car, Cost = 50000	Обратный ход
5	Подстановка Lastname = «Timonin», City = «Moscow» allOwners("Timonin", "Moscow", Ownership, Cost) allOwners(Lastname, City, Ownership, Cost)	Прямой ход
6	Сравнение «Timonin» = «Timonin», «Moscow» = «Vologda» owner("Timonin", "Vologda", building(175, "Real", 125000)) owner(Lastname, City, building(_, _, Price)) Нет	Прямой ход
7	Сравнение «Timonin» = «Timonin», «Moscow» = «Moscow» owner("Timonin", "Moscow", building(175, "Real", 125000)) owner(Lastname, City, building(_, _, Price)) Успех	Прямой ход
8	Подстановка: Cost = 125000, Ownership = Building»	Прямой ход
9	Результат: Ownership = Building, Cost = 125000	Обратный ход
10	Сравнение «Timonin» = «Sardyn», «Moscow» = «Orsk» owner("Sardyn", "Orsk", building(213, "Unreal", 1111)) owner(Lastname, City, building(_, _, Price)) Нет	Обратный ход
11	Подстановка Lastname = «Timonin», City = «Moscow» allOwners("Timonin", "Moscow", Ownership, Cost) allOwners(Lastname, City, Ownership, Cost)	Прямой ход
12	Сравнение «Timonin» = «Ibragimov», «Moscow» = «Vologda» owner("Ibragimov", "Vologda", plot(200, "Real", 12342)) owner(Lastname, City, plot(_, _, Price)) Нет	Обратный ход
13	Подстановка Lastname = «Timonin», City = «Moscow» allOwners("Timonin", "Moscow", Ownership, Cost) allOwners(Lastname, City, Ownership, Cost)	Прямой ход
14	Сравнение «Timonin» = «Timonin», «Moscow» = «Moscow» owner("Timonin", "Moscow", boat("BMX", "123", "Red", 1)) owner(Lastname, City, boat(_, _, _, Price)) Успех	Прямой ход

15	Подстановка: Cost = 1, Ownership = Boat	Прямой ход
16	Результат: Ownership = Boat, Cost = 1	Обратный ход
17	Сравнение «Timonin» = «Koverap», «Moscow» = «Dnepr» owner("Koverap", "Dnepr", boat("BMX", "123", "Red", 1290)) owner(Lastname, City, boat(____, Price)) Нет	Прямой ход
18	Сравнение «Timonin» = «Nebrogov», «Moscow» = «Rytia» owner("Nebrogov", "Rytia", boat("KOL", "Real", "Green", 90000)) owner(Lastname, City, boat(____, Price)) Нет	Обратный ход

Теоретическая часть

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

Знания о предметной области выражаются на языке Пролог в виде предложений, называемых утверждениями (clauses). Каждое утверждение заканчивается точкой и описывает какое-либо отношение, свойство, объект или закономерность. Структура утверждения проста и имеет одну из форм:

- <заголовок>. — факт
- <заголовок>:- <тело>. — правило, где заголовок является предикатом и полностью характеризует описываемое отношение.

Что содержит тело правила?

Правило используется Прологом для проверки условия вывода нового факта. Правила в Прологе записываются в разделе утверждений (clauses).

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 Прологе аналогично использованию их в логике (но не идентично). Вопросы, включающие переменные носят перечислительный характер: ответы в этом случае представляют собой списки термов. Например, вопрос «Кто в подчинении у Анри?» является вопросом «перечислительного» типа. Ответ БД библио мог бы быть [эмиль, жозеф]

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

На все переменные имени предиката имеют квантор всеобщности — \forall . На переменные в теле предиката, отсутствующих в имени, имеют квантор существования — \exists .

В Прологе область действия переменной - одно предложение. В различных предложениях может использоваться одно имя переменной для обозначения разных объектов. (Исключение — анонимная переменная)

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

Предложения раздела DOMAINS описывают новые типы данных, использующиеся в разделах PREDICATES, для описания отношений, в которых можно будет использовать более частных случаев типов данных.

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

Предикаты используются для описания фактов в разделе CLAUSES. Предикаты описывают какие домены будут принимать правило. Такое описание используется для создания договоренности, в какой последовательности или порядке будут следовать термы в отношениях, в которых они определены.

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

Для выполнения логического вывода используется механизм (алгоритм) унификации, встроенный в систему. **Унификация** — операция, которая позволяет формализовать процесс логического вывода (наряду с правилом резолюции). С практической точки зрения - это основной вычислительный шаг, с помощью которого происходит:

- Двухнаправленная передача параметров процедурам,
- Неразрушающее присваивание,
- Проверка условий (доказательство).

В каком случае запускается механизм отката?

Когда в программе возможен выбор нескольких вариантов, Пролог заносит в стек точку возврата, для последующего отката по этой точке возврата.

Пролог унифицирует выбранный вариант, если унификация прошла успешно, тогда пролог подготавливает ответ, и далее по точке возврата происходит унификация с другими вариантами. Если пролог не видит дальнейшие варианты, которые он мог бы проунифицировать, тогда по точке возврата программа возвращается на еще более раннюю стадию.

Исправления из прошлых лабораторных работ

ЧТО сравнивалось? Подстановка: $\text{Lastname} = \text{«Timonin»}$, $\text{Brand} = \text{Brand}$, $\text{Price} = \text{Price forphone(«Timonin», Brand, Price)}$

Сравнивается $\text{forphone}(\text{Lastname}, \text{Brand}, \text{Price})$ и $\text{forphone}(\text{«Timonin»}, \text{Brand}, \text{Price})$, где все аргументы равны: $\text{Lastname} = \text{«Timonin»}$, $\text{Brand} = \text{Brand}$, $\text{Price} = \text{Price}$. Так как сравнение произошло успешно, то выполняется подстановка «Timonin» на Lastname, Brand на Brand, Price на Price.

Назначение и результат работы алгоритма унификации. ...Слова правильные, но ответа нет!!! Система знает заранее для каких конкретно термов запускать алгоритм унификации????

Унификация позволяет отождествлять формулы логики первого порядка путем замены свободных переменных на термы.

В конце концов, алгоритм заканчивает свою работу, выдавая в качестве ответа множество подстановок, позволяющих унифицировать, и затем применить к ним правило резолюции для исчисления высказываний. Это Множество подстановок называется общим унификатором (ОУ) дизъюнктов.