



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

к лабораторной работе № 15

По курсу: «Функциональное и логическое программирование»

Студент ИУ7-64Б
Лозовский А.А.

Преподаватель
Толпинская Н.Б.

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 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 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).

Ответы на вопросы

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

Знание формируется в заголовке правила (правила описываются в разделе CLAUSES), также знание может быть описано набором фактов и правил, формирующих процедуру. Знание о предметной области.

2. Что содержит тело правила?

Тело правила содержит условия истинности знания, содержащегося в заголовке этого правила. Тело состоит из термов.

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 дает возможность передачи знаний «во времени и пространстве». Например, связанная с каким-то значением переменная, в рамках одного предложения, может быть использована в последующих подцелях тела правила.

Чем больше переменных используется в формулировке правила, с помощью терма, тем более общим будет являться терм. Например, parents(«mother», Father, «child») примером терма parents(Mother, Father, «child»), т.е. parents(Mother, Father, «child») является более общим.

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. Любая анонимная переменная является уникальной.

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

Предложения в разделе DOMAINS описывают структуру объектов, используемых для обозначения аргументов, между которыми устанавливается отношение в заголовке правил процедуры, в случае, если она не может быть описана с помощью стандартных доменов.

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

Предложения в разделе PREDICATES описывают структуру предиката. Предложения указываются в данном разделе, в случае если природа или структура объектов, обозначенных в качестве аргументов между которыми устанавливается отношение значима во время работы системы.

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

На первом шаге работы системы унифицируется вопрос с первым термом (расположенном в самом верху раздела) в разделе CLAUSES. Унификация – основной вычислительный шаг работы программы, предназначенный для формализации процесса логического вывода. Унификация – попытка сопоставить два терма. Результатом алгоритма унификации является «успех» или «неудача». В случае удачи в результате также является наиболее общий унификатор двух унифицируемых термов.

8. В каком случае запускается механизм отката?

Механизм отката запускается в случае завершения алгоритма унификации неудачей.

Текст программы

domains

adress = **adress**(symbol City, symbol Street, integer HouseNum, integer FlatNum)

*/*Вариантный домен собственность, включает в себя автомобиль, участок, водный транспорт и строение. */*

ownership = **automobile** (symbol Name, integer Price, symbol Colour);
 arrea (symbol Name, integer Price, symbol Size);
 water_transport (symbol Name, integer Price);
 building (symbol Name, integer Price)

person = **person**(symbol FirstName, symbol LastName)

predicates

/ Вкладчики банков*/*

investors (person, symbol Bank, symbol Acc_number, integer Value)

*/*Телефонный справочник*/*

phonebook (person, symbol PhoneNum, adress Addr)

*/*Позволяет получить название собственности*/*

type(symbol, ownership)

*/*Позволяет получить стоимость собственности*/*

price(integer, ownership)

*/*Описывает отношения между человеком и собственностью*/*

owns(person, ownership)

*/*Правило для выполнения задания ЛР*/*

owns(person, symbol, integer)

clauses

phonebook(person("Name1", "Surname1"), "0-000-111-222", adress("Moscow", "Unnatov", 14, 128)).
phonebook(person("Name1", "Surname1"), "0-000-111-222", adress("Moscow", "Mishina", 1, 10)).
phonebook(person("Name1", "Surname1"), "5-666-777-888", adress("St.Peterburg", "Unnatov", 14, 128)).
phonebook(person("Name2", "Surname2"), "1-222-333-444", adress("Moscow", "8th March", 12, 153)).
phonebook(person("Name3", "Surname3"), "3-444-555-666", adress("Moscow", "Mishina", 1, 10)).
phonebook(person("Name3", "Surname3"), "3-444-555-666", adress("Moscow", "Volgogradskaya", 1, 10)).

investors(person("Name1", "Surname1"), "Sberbank", "0000 4444 3333 2222", 700000).
investors(person("Name1", "Surname1"), "VTB", "0000 4444 3333 2222", 700000).
investors(person("Name2", "Surname2"), "Sberbank", "0000 2222 3333 2222", 200000).
investors(person("Name3", "Surname3"), "VTB", "0000 3333 3333 2222", 300000).

type(Name, automobile(CarName, _, _)) :-Name=CarName, !.
type(Name, arrea(ArreaName, _, _)) :-Name=ArreaName, !.
type(Name, water_transport(TrName, _)) :-Name=TrName, !.
type(Name, building(BName, _)) :-Name=BName, !.

price(Price, automobile(_, CarPrice, _)) :-Price=CarPrice, !.
price(Price, arrea(_, ArreaPrice, _)) :-Price=ArreaPrice, !.
price(Price, water_transport(_, TransportPrice)) :-Price=TransportPrice, !.
price(Price, building(_, BPrice)) :-Price=BPrice, !.

owns(person("Name1", "Surname1"), automobile("Car", 2000000, "Black")).
owns(person("Name1", "Surname1"), arrea("Arrea", 1200000, "Black")).
owns(person("Name1", "Surname1"), water_transport("Ship", 23)).
owns(person("Name2", "Surname2"), automobile("Car", 2300000, "Red")).
owns(person("Name2", "Surname2"), building("House", 1203212)).

owns(Person, Name, Price) :-owns(Person, Ownership),
type(Name, Ownership),
price(Price, Ownership).

Примеры целей и результатов работы программы

Задание: поиск названий всех объектов собственности заданного субъекта.

1. **Goal** owns(person("Name2", "Surname2"), Name, _).

Result Name=Car

Name=House

Задание: поиск названий и стоимости всех объектов собственности заданного субъекта.

2. **Goal** owns(person("Name1", "Surname1"), Name, Price).

Result Name=Car, Price=2000000

Name=Arrea, Price=1200000

Name=Ship, Price=23

Описание порядка поиска объектов

Цель: owns(person("Name2", "Surname2"), Name, Price).

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
0	-попытка унификации: T1=T2 ... -результат: Успех и подстановка, или Нет	Комментарий, вывод
1	<ul style="list-style-type: none"> Попытка унификации: owns(person("Name2", "Surname2"), Name, Price), phonebook(person("Name1", "Surname1"),...). Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	Прямой ход, переход к следующему предложению.
2	<ul style="list-style-type: none"> Попытка унификации: owns(person("Name2", "Surname2"), Name, Price), phonebook(person("Name1", "Surname1"),...). Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	Прямой ход, переход к следующему предложению.
3	<ul style="list-style-type: none"> Попытка унификации: owns(person("Name2", "Surname2"), Name, Price), phonebook(person("Name1", "Surname1"),...). Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	Прямой ход, переход к следующему предложению.
4	<ul style="list-style-type: none"> Попытка унификации: owns(person("Name2", "Surname2"), Name, Price), phonebook(person("Name2", "Surname2"),...). Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	Прямой ход, переход к следующему предложению.
5	<ul style="list-style-type: none"> Попытка унификации: owns(person("Name2", "Surname2"), Name, Price), phonebook(person("Name3", "Surname3"),...). Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	Прямой ход, переход к следующему предложению.
6	<ul style="list-style-type: none"> Попытка унификации: owns(person("Name2", "Surname2"), Name, Price), phonebook(person("Name3", "Surname3"),...). Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	Прямой ход, переход к следующему предложению.
7	<ul style="list-style-type: none"> Попытка унификации: owns(person("Name2", "Surname2"), Name, Price), investors(person("Name1", "Surname1"),...). Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	Прямой ход, переход к следующему предложению.
8	<ul style="list-style-type: none"> Попытка унификации: owns(person("Name2", "Surname2"), Name, Price), investors(person("Name1", "Surname1"),...). 	Прямой ход, переход к следующему предложению.

	<ul style="list-style-type: none"> Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	
9	<ul style="list-style-type: none"> Попытка унификации: owns(person("Name2", "Surname2"), Name, Price), investors(person("Name2", "Surname2"),...). Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	Прямой ход, переход к следующему предложению.
10	<ul style="list-style-type: none"> Попытка унификации: owns(person("Name2", "Surname2"), Name, Price), investors(person("Name3", "Surname3"),...). Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	Прямой ход, переход к следующему предложению.
11	<ul style="list-style-type: none"> Попытка унификации: owns(person("Name2", "Surname2"), Name, Price), type(Name, automobile(CarName, _, _)). Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	Прямой ход, переход к следующему предложению
...	Аналогично выполняется программа со всеми предположениями type(...).	Прямой ход, переход к следующему предложению
18	<ul style="list-style-type: none"> Попытка унификации: owns(person("Name2", "Surname2"), Name, Price), owns(person("Name1", "Surname1"), automobile ("Car", 2000000, "Black")). Результат: неудача, несовпадающая аргументность. 	Прямой ход, переход к следующему предложению
...	Аналогично выполняется программа со всеми термами owns(person(...), ...) с отличной от вопроса аргументностью.	Прямой ход, переход к следующему предложению
23	<ul style="list-style-type: none"> Попытка унификации: owns(person("Name2", "Surname2"), Name, Price), owns(Person, Name, Price). Результат: успех Подстановка: {Person= person("Name2", "Surname2"), Name=Name, Price=Price}. 	Прямой ход, переход в тело правила.
...	<ul style="list-style-type: none"> Попытка унификации: owns(person("Name2", "Surname2"), Ownership), и первых 21 терма Результат: неудача 	Прямой ход, переход к следующему предложению
45	<ul style="list-style-type: none"> Попытка унификации: owns(person("Name2", "Surname2"), Ownership), owns(person("Name2", "Surname2"), automobile ("Car", 2300000, "Red")). Результат: успех Подстановка { person("Name2", "Surname2")= person("Name2", "Surname2"), Ownership=automobile ("Car", 2300000, "Red") } 	Прямой ход, переход в к следующей цели правила.
...	<ul style="list-style-type: none"> Попытка унификации: type(Name, automobile ("Car", 2300000, "Red")), и первых 10 термов Результат: неудача 	Прямой ход, переход к следующему предложению

56	<ul style="list-style-type: none"> Попытка унификации: type(Name, automobile ("Car", 2300000, "Red")), type(Name, automobile (CarName, _ , _)), Результат: успех Подстановка: { Name=Name, automobile ("Car", 2300000, "Red")= automobile (CarName, _ , _) } 	Прямой ход, переход в тело правила
57	<ul style="list-style-type: none"> Попытка унификации: Name="Car" Результат: успех Подстановка: {CarName="Car" } 	Прямой ход, выполнение предиката отсечения. Выход из правила, переход к следующей подцели.
58	<ul style="list-style-type: none"> Попытка унификации: price(Price, automobile ("Car", 2300000, "Red")), и первых 14 термов Результат: неудача 	Прямой ход, переход к следующему предложению
73	<ul style="list-style-type: none"> Попытка унификации: price(Price, automobile ("Car", 2300000, "Red")), price(Price, automobile (_ , CarPrice , _)), Результат: успех Подстановка: { Price=Price, automobile ("Car", 2300000, "Red")= automobile (_ , CarPrice , _) } 	Прямой ход, переход в тело правила
74	<ul style="list-style-type: none"> Попытка унификации: CarPrice=2300000 Результат: успех Подстановка: {CarPrice=2300000 } 	Прямой ход, выполнение предиката отсечения. Выход из правила. Name="Car", Price=2300000. Откат.
...	<ul style="list-style-type: none"> Попытка унификации: price(Price, automobile ("Car", 2300000, "Red")), и price(Price, arrea(_ , ArreaPrice, _)) price(Price, water_transport(_ , TransportPrice)) price(Price, building(_ , BPrice)) Результат: неудача 	Прямой ход переход к следующим предложениям.
...	<ul style="list-style-type: none"> Попытка унификации: price(Price, automobile ("Car", 2300000, "Red")), термов owns(...). Результат: неудача, главные функторы не совпадают, термы не унифицируемы. 	Конец БЗ. Откат
...	<ul style="list-style-type: none"> Попытка унификации: type(Name, automobile ("Car", 2300000, "Red")) и type(Name, arrea(ArreaName, _ , _)) type(Name, water_transport(TrName, _)) type(Name, building(BName, _)) Результат: неудача 	Прямой ход переход к следующим предложениям.

...	<ul style="list-style-type: none"> Попытка унификации: type(Name, automobile ("Car", 2300000, "Red")) со всеми терминами до конца БЗ. Результат: неудача 	Конец БЗ. Откат
-----	--	--------------------

Таблица 1: Порядок поиска объектов

Исправления ЛР №14

ИУ7-64Б Лозовский А.А. а где л.р.№13?

Л.Р. №13 я отправлял 18 апреля, отправил вам на почту письмо с отчетом еще раз.

3. Сколько в БЗ текущего задания процедур?

Конкретно в моей программе **6 процедур НЕТ!!** – phonebook, investors, automobile и 3 правила search_by. **А сколько описано предикатов?**

Ответ: В программе описано 4 предиката – phonebook, investors, automobile и search_by, поэтому в программе 4 процедуры.

4. Что такое пример терма, это частный случай терма, пример? Как строится пример?

... Пример терма строится при **сопоставлении**(конкретизации) более общего терма с **возможными подстановками**. **Сопоставить терм можно с термом, и то не со всяким, но не с подстановкой!!**

Ответ: Пример терма строится путем применения подстановки к терму, т.е конкретизацией одной или нескольких переменных терма.

6. Назначение и результат работы алгоритма унификации ... - Предназначен для «сопоставления термов», формализует процесс логического вывода **Какая цель сопоставлять?**

Ответ: Алгоритм унификации предназначен для формализации процесса логического вывода. При сопоставлении двух термов пытается построить для них общий пример, это нужно для того, чтобы формально определить подходящее, при поиске ответа на вопрос, знание. При помощи алгоритма унификации происходит двунаправленная передача параметров процедурам.

Задание 3 Для одного вопроса и одного подходящего терма заполните таблицу аккуратно!!!

Текст программы ЛР №14

domains

adress = adress(symbol City, symbol Street, integer HouseNum, integer FlatNum)

predicates

phonebook (symbol Surname, symbol PhoneNum, adress Adrr)

automobile(symbol Surname, symbol Brand, symbol Colour, integer Price)

investors (symbol Surname, symbol Bank, symbol Acc_number, integer Value)

search_by(symbol CarBrand, symbol CarColour, symbol Surname, symbol City, symbol PhoneNum, symbol Bank)

clauses

phonebook("Surname1", "0-000-111-222", adress("Moscow", "Unnatov", 14, 128)).

phonebook("Surname1", "5-666-777-888", adress("St.Peterburg", "Nevskiy", 14, 128)).

phonebook("Surname2", "1-222-333-444", adress("Moscow", "8th March", 12, 153)).

phonebook("Surname3", "3-444-555-666", adress("Moscow", "Mishina", 1, 10)).

phonebook("Surname4", "9-000-111-111", adress("St.Peterburg", "Nevskiy", 1, 10)).


```

/*3 drivers, 1 doesn't have investments*/
automobile("Surname1", "Ford", "Black", 1600000).
automobile("Surname2", "Ford", "Black", 1600000).
automobile("Surname3", "Ford", "Black", 1600000).

/*1 driver*/
automobile("Surname4", "Volvo", "Silver", 1300000).

/*no drivers, doesn't have investments*/
automobile("Surname3", "Nissan", "Red", 1300000).

investors("Surname1", "Sberbank", "0000 4444 3333 2222", 700000).
investors("Surname2", "Sberbank", "0000 2222 3333 2222", 200000).
investors("Surname4", "MoscowBank", "0000 3333 3333 2222", 300000).
investors("Surname4", "VTB", "0000 4444 3333 2222", 700000).

/*searches for surname, city, phone, bank*/
search_by(CarBrand, CarColour, Surname, City, PhoneNum, Bank) :-
    automobile(Surname, CarBrand, CarColour, _),
    phonebook(Surname, PhoneNum, adress(City, _, _)),
    investors(Surname, Bank, _, _).

```

Вопрос

Goal search_by("Volvo", "Silver", Surname, City, PhoneNum, Bank)

шаг	результатирующая ячейка	рабочее поле	пункт Алг.	стек
0			1.	search_by("Volvo", "Silver", Surname, City, PhoneNum, Bank) = search_by(CarBrand, CarColour, Surname, City, PhoneNum, Bank)
1		search_by("Volvo", "Silver", Surname, City, PhoneNum, Bank) = search_by(CarBrand, CarColour, Surname, City, PhoneNum, Bank) →	Е.	CarBrand="Volvo", CarColour="Silver", Surname= Surname, City=City, PhoneNum=PhoneNum, Bank=Bank
2	CarBrand="Volvo"	← CarBrand="Volvo"	Г.	CarColour="Silver", Surname= Surname, City=City, PhoneNum=PhoneNum, Bank=Bank
3	CarBrand="Volvo", CarColour="Silver"	← CarColour="Silver"	Г.	Surname= Surname, City=City, PhoneNum=PhoneNum, Bank=Bank
4	CarBrand="Volvo", CarColour="Silver", Surname= Surname	← Surname= Surname	-	City=City, PhoneNum=PhoneNum, Bank=Bank
5	CarBrand="Volvo", CarColour="Silver", Surname= Surname, City=City	← City=City	-	PhoneNum=PhoneNum, Bank=Bank

6	CarBrand="Volvo", CarColour="Silver", Surname= Surname, City=City, PhoneNum=PhoneNum	PhoneNum=PhoneNum ←	-	Bank=Bank
7	CarBrand="Volvo", CarColour="Silver", Surname= Surname, City=City, PhoneNum=PhoneNum, Bank=Bank	Bank=Bank	-	Переход в тело правила Surname= Surname, City=City, PhoneNum=PhoneNum, Bank=Bank – связанные переменные.
8	Новая цель: automobile(Surname, "Volvo", "Silver", _) далее будет рассматриваться сразу подходящий терм			
9		automobile(Surname, "Volvo", "Silver", _) = automobile("Surname4", "Volvo", "Silver", 1300000) →	Е.	Surname="Surname4", "Volvo"="Volvo", "Silver"="Silver", Анонимные переменные не могут быть связаны.
10	Surname= "Surname4",	Surname="Surname4", ←	Г.	"Volvo"="Volvo", "Silver"="Silver", Анонимные переменные не могут быть связаны.
11	Surname= "Surname4"	"Volvo"="Volvo"	Б.	"Silver"="Silver"
12	Surname= "Surname4"	"Silver"="Silver"	Б.	Стек пуст, очистка рабочего пространства
	Подстановка	Т.к. стек пуст – успех , Переход к следующей подцели тела.		
13	Новая цель: phonebook("Surname4", PhoneNum, adress(City, _, _, _)) далее будет рассматриваться сразу подходящий терм			
14		phonebook("Surname4", PhoneNum, adress(City, _, _, _)) и phonebook("Surname4", "9-000- 111-111", adress("St.Peterburg", "Nevskiy", 1, 10)). →	Е.	"Surname4"="Surname4", PhoneNum= "9-000-111-111", City="St.Peterburg".
15	Surname= "Surname4",	"Surname4"="Surname4"	Б.	PhoneNum= "9-000-111-111", City="St.Peterburg".
16	Surname= "Surname4", PhoneNum= "9-000-111- 111" ,	PhoneNum= "9-000-111-111" ←	Г.	City="St.Peterburg".
17	Surname= "Surname4", PhoneNum= "9-000-111- 111", City="St.Peterburg"	City="St.Peterburg" ←	Г.	Стек пуст, очистка рабочего пространства
	Подстановка	Т.к. стек пуст – успех , Переход к следующей подцели тела.		
18	Новая цель: investors("Surname4", Bank, _, _). далее будет рассматриваться сразу подходящий терм			

19		<code>investors("Surname4", Bank, _, _)</code> <code>=</code> <code>investors("Surname4",</code> <code>"MoscowBank", "0000 3333 3333</code> <code>2222", 300000)</code> \longrightarrow	Е.	<code>"Surname4"="Surname4",</code> <code>Bank="MoscowBank".</code>
20		<code>"Surname4"="Surname4"</code>	Б.	<code>Bank="MoscowBank".</code>
21	<code>Bank="MoscowBank".</code>	<code>Bank="MoscowBank".</code> \longleftarrow	Г.	Стек пуст, очистка рабочего пространства
	Подстановка	Т.к. стек пуст – успех , Переход к следующей подцели тела.		

Получена подстановка: {Surname= "Surname4", City="St.Peterburg", PhoneNum="9-000-111-111", Bank="MoscowBank"}, такая подстановка получилась поскольку переменные Surname, City, PhoneNum, Bank – связанные именованные переменные. (двунаправленная передача параметров)