



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 №13

*По курсу: «Функциональное и логическое
программирование»*

Темы: «Структура программы на Prolog и ее реализация»

Студент: Зайцева А. А.
Группа: ИУ7-62Б
Преподаватели: Толпинская Н. Б.,
Строганов Ю. В.

Москва, 2022 г.

Практическая часть

Задание. Создать базу знаний «**Собственники**», дополнив базу знаний, хранящую знания (лаб. 12):

- «**Телефонный справочник**»: Фамили, №тел, Адрес – структура (Город, Улица, №дома, №кв),
- «**Автомобили**»: Фамилия_владельца, Марка, Цвет, Стоимость, и др.,
- «**Вкладчики банков**»: Фамилия, Банк, счёт, 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 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)

domains

```
city, street = symbol.  
house, flat = integer.  
address_struct = address(city, street, house, flat).  
surname = symbol.  
phone = integer.  
model, color = symbol.  
price, year = integer.  
bank = symbol.  
sum, account = integer.  
size = integer.  
  
ownership = building(price, address_struct);  
              area(price, size);  
              water_transport(price, color);  
              car(price, model, color).
```

predicates

```
phone_record(surname, phone, address_struct).  
depositor(surname, bank, account, sum).  
  
own(surname, ownership).  
  
ownershipsNamePrice(surname, symbol, price).  
sumCostInner(surname, symbol, price).  
sumCost(surname, price).
```

clauses

```
phone_record(rich, 7777772, address(london, green, 1, 10)).  
phone_record(rich, 7777771, address(london, green, 1, 10)).  
phone_record(rich, 1111111, address(moscow, zelenaya, 2, 20)).  
phone_record(middle, 9999999, address(moscow, ivanovskaya, 3, 2)).  
phone_record(poor, 3333331, address(karaganda, pit, 23, 5)).  
phone_record(poor, 3333332, address(perm, pit, 36, 7)).  
phone_record(poor, 3333333, address(kop, leet, 2, 53)).  
  
depositor(rich, gosbank, 10, 10000000).  
depositor(rich, mosbank, 15, 9000000).  
depositor(middle, mosbank, 17, 20000).  
depositor(middle, newbank, 345, 0).
```

```

own(rich, building(100, address(kop, leet, 2, 53))).
own(rich, area(10, 500)).
own(rich, water_transport(1, green)).
own(rich, car(5, model1, red)).

own(middle, car(3, model2, red)).
own(middle, building(90, address(moscow, leninskaya, 2, 53))).

```

```

% 1, 2
ownshipsNamePrice(Surname_, building, Price_) :- own(Surname_, building(Price_, _)).
ownshipsNamePrice(Surname_, area, Price_) :- own(Surname_, area(Price_, _)).
ownshipsNamePrice(Surname_, water_transport, Price_) :- own(Surname_, water_transport(Price_, _)).
ownshipsNamePrice(Surname_, car, Price_) :- own(Surname_, car(Price_, _, _)).

```

```

% 3
sumCostInner(Surname_, building, Price_) :- own(Surname_, building(Price_, _)), !.
sumCostInner(Surname_, area, Price_) :- own(Surname_, area(Price_, _)), !.
sumCostInner(Surname_, water_transport, Price_) :- own(Surname_, water_transport(Price_, _)), !.
sumCostInner(Surname_, car, Price_) :- own(Surname_, car(Price_, _, _)), !.
sumCostInner(_, _, 0). % if Surname_ has no some kind of ownership

```

```

sumCost(Surname_, Price_) :-
    sumCostInner(Surname_, building, Price1_),
    sumCostInner(Surname_, area, Price2_),
    sumCostInner(Surname_, water_transport, Price3_),
    sumCostInner(Surname_, car, Price4_),
    Price_ = Price1_ + Price2_ + Price3_ + Price4_.

```

goal

```

% 1
% ownshipsNamePrice(rich, Ownership, _).
% Ownership=building
% Ownership=area
% Ownership=water_transport
% Ownership=car
% 4 Solutions

```

```

% ownshipsNamePrice(Surname, car, _).
% Surname=rich
% Surname=middle
% 2 Solutions

```

```

% 2
ownshipsNamePrice(middle, Ownership, Price).
% Ownership=building, Price=90
% Ownership=car, Price=3
% 2 Solutions

```

```

% ownshipsNamePrice(Surname, building, Price).
% Surname=rich, Price=100
% Surname=middle, Price=90
% 2 Solutions

```

```

% 3
% sumCost(middle, Price). % =sumCost("middle", Price).
% Price=93
% 1 Solution

```

```

% sumCost(rich, Price).
% Price=116
% 1 Solution

```

```

% sumCost(poor, Price).
% Price=0
% 1 Solution

```

ownshipsNamePrice(middle, Ownnership, Price).

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
0		Состояние резольвенты: f("Model_2", "Red", Surname, City, Phone, Bank). Поиск способа доказательства подцели с начала БЗ
1	Сравниваемые термы: f("Model_2", "Green", Surname, City, Phone, Bank). car("Middle", "Model_2", "Green", 5000000, 1900). Результат: Унификация неуспешна (разные главные функторы)	Прямой ход, переход к следующему предложению
2-9
10	Сравниваемые термы: f("Model_1", "Red", Surname, City, Phone, Bank). f(Model_, Color_, Surname, City, Phone, Bank) Результат: Унификация успешна Подстановка: {Model_="Model_2", Color_="Green", Surname=Surname, City=City, Phone=Phone, Bank=Bank}	Образование новой резольвенты: 1. Редукция верхней подцели: замена f("Model_1", "Red", Surname, City, Phone, Bank) телом найденного правила. Полученная конъюнкция целей: car(Surname, Model_, Color_, _, _), phone_record(Surname, Phone, address(City, _, _)), depositor(Surname, Bank, _, _) 2. Применение подстановки к полученной конъюнкции целей. Новое состояние резольвенты: car(Surname, "Model_2", "Green", _, _), phone_record(Surname, Phone, address(City, _, _)), depositor(Surname, Bank, _, _) Поиск способа доказательства новой подцели (верхней) с начала БЗ
11	Сравниваемые термы: car(Surname, "Model_2", "Green", _, _), car("Middle", "Model_2", "Green", 5000000, 1900). Результат: Унификация успешна Подстановка: {Surname="Middle"}	Образование новой резольвенты: 1. Редукция верхней подцели: удаление car(Surname, "Model_2", "Green", _, _), так как найденное правило – факт (тело пустое). Полученная конъюнкция целей: phone_record(Surname, Phone, address(City, _, _)), depositor(Surname, Bank, _, _) 2. Применение подстановки к полученной конъюнкции целей. Новое состояние резольвенты: phone_record("Middle", Phone, address(City, _, _)), depositor("Middle", Bank, _, _) Поиск способа доказательства новой подцели (верхней) с начала БЗ
12	Сравниваемые термы: phone_record("Middle", Phone, address(City, _, _)), car("Middle", "Model_2", "Green", 5000000, 1900). Результат: Унификация неуспешна	Прямой ход, переход к следующему предложению
13-19
20	Сравниваемые термы: phone_record("Middle", Phone, address(City, _, _)), phone_record("Middle", 9999999, address("Moscow", "Ivanovskaya", 3, 2)). Результат: Унификация успешна Подстановка: {Phone=9999999, City="Moscow"}	Образование новой резольвенты: 1. Редукция верхней подцели: удаление phone_record("Middle", Phone, address(City, _, _)), depositor("Middle", Bank, _, _), так как найденное правило – факт (тело пустое). Полученная конъюнкция целей: depositor("Middle", Bank, _, _) 2. Применение подстановки к полученной конъюнкции целей. Новое состояние резольвенты: depositor("Middle", Bank, _, _) Поиск способа доказательства новой подцели (верхней) с начала БЗ
21	Сравниваемые термы: depositor("Middle", Bank, _, _)	Прямой ход, переход к следующему предложению

	<p>car("Middle", "Model_2", "Green", 5000000, 1900).</p> <p>Результат: Унификация неуспешна</p>	
22-25
26	<p>Сравниваемые термы: depositor("Middle", Bank, _, _) depositor("Middle", "Mosbank", 17, 20000).</p> <p>Результат: Унификация успешна</p> <p>Подстановка: {Bank="Mosbank"}</p>	<p>Образование новой резолюенты: Редукция верхней подцели: удаление depositor("Rich", Bank, _, _), так как найденное правило – факт (тело пустое).</p> <p>Новое состояние резолюенты: пуста</p> <p>Решение найдено: формирование подстановки {Surname= "Middle", City="Moscow", Phone=9999999, Bank="Mosbank"} в качестве побочного эффекта.</p> <p>Система должна получить все возможные ответы</p>
		<p>Бэктрэкнинг («обратная трассировка»):</p> <p>Отмена последней редукции (на шаге 26),</p> <p>восстановление предыдущего состояния резолюенты (с шага 20): depositor("Middle", Bank, _, _)</p> <p>реконкретизация переменных, которые были конкретизированы на предыдущем шаге (26) ({Bank="Mosbank"})</p> <p>переход к следующему предложению относительно шага 26</p>
27	<p>Сравниваемые термы: depositor("Middle", Bank, _, _)</p> <p>phone_record("Rich", 7777772, address("London", "Green", 1, 10)).</p> <p>Результат: Унификация неуспешна</p>	<p>Прямой ход, переход к следующему предложению</p>
28-30
		<p>Конец БЗ. Из данного состояния невозможен переход в новое состояние.</p> <p>Бэктрэкнинг («обратная трассировка»):</p> <p>Отмена последней редукции (на шаге 20),</p> <p>восстановление предыдущего состояния резолюенты (с шага 11): phone_record("Middle", Phone, address(City, _, _)), depositor("Middle", Bank, _, _)</p> <p>реконкретизация переменных, которые были конкретизированы на предыдущем шаге (20) ({Phone=9999999, City="Moscow"})</p> <p>переход к следующему предложению относительно шага 20</p>
31	<p>Сравниваемые термы: phone_record("Middle", Phone, address(City, _, _)), f(Model_, Color_, Surname, City, Phone, Bank)</p> <p>Результат: Унификация неуспешна (разные главные функторы)</p>	<p>Конец БЗ. Из данного состояния невозможен переход в новое состояние.</p> <p>Бэктрэкнинг («обратная трассировка»):</p> <p>Отмена последней редукции (на шаге 11),</p> <p>восстановление предыдущего состояния резолюенты (с шага 10): car(Surname, "Model_2", "Green", _, _), phone_record(Surname, Phone, address(City, _, _)), depositor(Surname, Bank, _, _)</p> <p>реконкретизация переменных, которые были конкретизированы на предыдущем шаге (11) ({Surname="Middle"})</p> <p>переход к следующему предложению относительно шага 11</p>
32	<p>Сравниваемые термы: car(Surname, "Model_2", "Green", _, _), car("Rich", "Model_2", "Green", 5000000, 1900).</p> <p>Результат: Унификация успешна</p> <p>Подстановка: {Surname="Rich"}</p>	<p>Образование новой резолюенты: 1. Редукция верхней подцели: удаление car(Surname, "Model_2", "Green", _, _), так как найденное правило – факт (тело пустое).</p> <p>Полученная конъюнкция целей: phone_record(Surname, Phone, address(City, _, _)), depositor(Surname, Bank, _, _)</p> <p>2. Применение подстановки к полученной конъюнкции целей.</p> <p>Новое состояние резолюенты:</p>

		<p>phone_record("Rich", Phone, address(City, _, _)), depositor("Middle", Bank, _, _)</p> <p>Поиск способа доказательства новой подцели (верхней) с начала БЗ</p>
33	<p>Сравниваемые термы: phone_record("Rich", Phone, address(City, _, _)),</p> <p>car("Middle", "Model_2", "Green", 5000000, 1900).</p> <p>Результат: Унификация неуспешна</p>	<p>Прямой ход, переход к следующему предложению</p>
34-38
39	<p>Сравниваемые термы: phone_record("Rich", Phone, address(City, _, _)),</p> <p>phone_record("Rich", 7777772, address("London", "Green", 1, 10)).</p> <p>Результат: Унификация успешна Подстановка: {Phone=7777772, City="London"}</p>	<p>Образование новой резолювенты: 1. Редукция верхней подцели: удаление phone_record("Rich", Phone, address(City, _, _)), так как найденное правило – факт (тело пустое).</p> <p>Полученная конъюнкция целей: depositor("Rich", Bank, _, _)</p> <p>2. Применение подстановки к полученной конъюнкции целей.</p> <p>Новое состояние резолювенты: depositor("Rich", Bank, _, _)</p> <p>Поиск способа доказательства новой подцели (верхней) с начала БЗ</p>
40	<p>Сравниваемые термы: depositor("Rich", Bank, _, _)</p> <p>car("Middle", "Model_2", "Green", 5000000, 1900).</p> <p>Результат: Унификация неуспешна</p>	<p>Прямой ход, переход к следующему предложению</p>
41-43
44	<p>Сравниваемые термы: depositor("Rich", Bank, _, _)</p> <p>depositor("Rich", "Gosbank", 10, 10000000).</p> <p>Результат: Унификация успешна Подстановка: {Bank="Gosbank"}</p>	<p>Образование новой резолювенты: Редукция верхней подцели: удаление depositor("Rich", Bank, _, _), так как найденное правило – факт (тело пустое).</p> <p>Новое состояние резолювенты: пуста</p> <p>Решение найдено: формирование подстановки {Surname= Rich", City="London", Phone=7777772, Bank="Gosbank"} в качестве побочного эффекта.</p> <p>Система должна получить все возможные ответы</p>
		<p>Бэктрэкнинг («обратная трассировка»):</p> <p>Отмена последней редукции (на шаге 44),</p> <p>восстановление предыдущего состояния резолювенты (с шага 39): depositor("Rich", Bank, _, _)</p> <p>реконкретизация переменных, которые были конкретизированы на предыдущем шаге (44) ({Bank="Gosbank"})</p> <p>переход к следующему предложению относительно шага 44</p>
45	<p>Сравниваемые термы: depositor("Rich", Bank, _, _)</p> <p>depositor("Middle", "Mosbank", 17, 20000).</p> <p>Результат: Унификация неуспешна</p>	<p>Прямой ход, переход к следующему предложению</p>
46-49
		<p>Конец БЗ. Из данного состояния невозможен переход в новое состояние.</p> <p>Бэктрэкнинг («обратная трассировка»):</p> <p>Отмена последней редукции (на шаге 39),</p> <p>восстановление предыдущего состояния резолювенты (с шага 32):</p>

		<p>phone_record("Rich", Phone, address(City, _, _)), depositor("Middle", Bank, _, _)</p> <p>реконкретизация переменных, которые были конкретизированы на предыдущем шаге (39) ({Phone=7777772, City="London"})</p> <p>переход к следующему предложению относительно шага 39</p>
50	<p>Сравниваемые термы: phone_record("Rich", Phone, address(City, _, _)),</p> <p>phone_record("Rich", 1111111, address("Moscow", "Zelenaya", 2, 20)).</p> <p>Результат: Унификация успешна Подстановка: {Phone=1111111, City="Moscow"}</p>	<p>Образование новой резольвенты: 1. Редукция верхней подцели: удаление phone_record("Rich", Phone, address(City, _, _)), так как найденное правило – факт (тело пустое).</p> <p>Полученная конъюнкция целей: depositor("Rich", Bank, _, _)</p> <p>2. Применение подстановки к полученной конъюнкции целей.</p> <p>Новое состояние резольвенты: depositor("Rich", Bank, _, _)</p> <p>Поиск способа доказательства новой подцели (верхней) с начала БЗ</p>
51	<p>Сравниваемые термы: depositor("Rich", Bank, _, _)</p> <p>car("Middle", "Model_2", "Green", 5000000, 1900).</p> <p>Результат: Унификация неуспешна</p>	<p>Прямой ход, переход к следующему предложению</p>
52-54
55	<p>Сравниваемые термы: depositor("Rich", Bank, _, _)</p> <p>depositor("Rich", "Gosbank", 10, 10000000).</p> <p>Результат: Унификация успешна Подстановка: {Bank="Gosbank"}</p>	<p>Образование новой резольвенты: Редукция верхней подцели: удаление depositor("Rich", Bank, _, _), так как найденное правило – факт (тело пустое).</p> <p>Новое состояние резольвенты: пуста</p> <p>Решение найдено: формирование подстановки {Surname= Rich", City="" Moscow", Phone=1111111, Bank="Gosbank"} в качестве побочного эффекта.</p> <p>Система должна получить все возможные ответы</p>
		<p>Бэктрэкнинг («обратная трассировка»):</p> <p>Отмена последней редукции (на шаге 55),</p> <p>Восстановление предыдущего состояния резольвенты (с шага 50): depositor("Rich", Bank, _, _)</p> <p>реконкретизация переменных, которые были конкретизированы на предыдущем шаге (55) ({Bank="Gosbank"})</p> <p>переход к следующему предложению относительно шага 55</p>
56	<p>Сравниваемые термы: depositor("Rich", Bank, _, _)</p> <p>depositor("Middle", "Mosbank", 17, 20000).</p> <p>Результат: Унификация неуспешна</p>	<p>Прямой ход, переход к следующему предложению</p>
57-60
		<p>Конец БЗ. Из данного состояния невозможен переход в новое состояние.</p> <p>Бэктрэкнинг («обратная трассировка»):</p> <p>Отмена последней редукции (на шаге 50),</p> <p>Восстановление предыдущего состояния резольвенты (с шага 49):</p> <p>phone_record("Rich", Phone, address(City, _, _)), depositor("Middle", Bank, _, _)</p>

		<p>реконкретизация переменных, которые были конкретизированы на предыдущем шаге (50) ({Phone=1111111, City="Moscow"})</p> <p>переход к следующему предложению относительно шага 50</p>
61	<p>Сравниваемые термы: phone_record("Rich", Phone, address(City, _, _)),</p> <p>phone_record("Middle", 9999999, address("Moscow", "Ivanovskaya", 3, 2)).</p> <p>Результат: Унификация неуспешна (пары компонент не унифицируются успешно)</p>	<p>Прямой ход, переход к следующему предложению</p>
62	<p>Сравниваемые термы: phone_record("Rich", Phone, address(City, _, _)),</p> <p>f(Model_, Color_, Surname, City, Phone, Bank)</p> <p>Результат: Унификация неуспешна (разные главные функторы)</p>	<p>Конец БЗ. Из данного состояния невозможен переход в новое состояние.</p> <p>Бэктрэкнинг («обратная трассировка»):</p> <p>Отмена последней редукции (на шаге 32),</p> <p>восстановление предыдущего состояния резольвенты (с шага 10):</p> <p>car(Surname, "Model_2", "Green", _, _), phone_record(Surname, Phone, address(City, _, _)), depositor(Surname, Bank, _, _)</p> <p>реконкретизация переменных, которые были конкретизированы на предыдущем шаге (32) ({Surname="Rich"})</p> <p>переход к следующему предложению относительно шага 32</p>
63	<p>Сравниваемые термы: car(Surname, "Model_2", "Green", _, _), car("Nobody", "Model_0", "Red", 1000000, 2022).</p> <p>Результат: Унификация неуспешна</p>	<p>Прямой ход, переход к следующему предложению</p>
64-69
		<p>Конец БЗ. Решение не найдено, и из данного состояния невозможен переход в новое состояние.</p> <p>Бэктрэкнинг («обратная трассировка»):</p> <p>Отмена последней редукции (на шаге 10),</p> <p>восстановление предыдущего состояния резольвенты (с шага 0):</p> <p>car(Surname, "Model_2", "Green", _, _), phone_record(Surname, Phone, address(City, _, _)), depositor(Surname, Bank, _, _)</p> <p>реконкретизация переменных, которые были конкретизированы на предыдущем шаге (10)({Model_="Model_2", Color_="Green"})</p> <p>переход к следующему предложению относительно шага 10</p>
		<p>Конец БЗ. Решение не найдено, и из данного состояния невозможен переход в новое состояние.</p> <p>Бэктрэкнинг («обратная трассировка»):</p> <p>Отмена последней редукции (на шаге 0),</p> <p>восстановление предыдущего состояния резольвенты: резольвента пуста</p> <p>Завершение работы. На вопрос удалось ответить утвердительно, 3 подстановки были возвращены в качестве побочного эффекта.</p>