



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»

Дисциплина: «Функциональное и логическое  
программирование»

Исправление ошибок 14, 15, 16 лабораторных  
работ

Студент: Левушкин И. К.

Группа: ИУ7-62Б

Преподаватели: Толпинская Н. Б.,

Строганов Ю. В.

Москва, 2020 г.

# Исправление ошибок 15-ой лабораторной работы

## Замечание

*В таблице странная нотация: != в Prolog символ ! – значимый*

*Подстановка Building = moscow\_city, Building\_Cost = 987654321. Это множество – нотация: {...}*

**Исправленная таблица для поиска «Названия и стоимости всех объектов собственности по заданному субъекту»**

*Порядок работы системы на 3 примере:*

```
goal
owner(ryazanova, area_owner(Area, Area_Cost));
owner(ryazanova, building_owner(Building, Building_Cost));
owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));
owner(ryazanova, car_owner(Car, _, Car_Cost, _)).
%Вывод:
Building=moscow_city, Building_Cost=987654321
Car=bugatti, Car_Cost=999999999
2 Solutions
```

№ ша-га	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	T1 = owner(ryazanova, area_owner(Area, Area_Cost)); T2 = get_own_cost(Sername, building, Cost). Неудача (функторы owner и get_own_cost не равны).	Прямой ход к следующему предложению. Аналогичная ситуация в следующих 5 предложениях. Прямой ход к следующему предложению
7	T1 = owner(ryazanova, area_owner(Area, Area_Cost)); T2 = owner(Sername, car_owner(Model, Color, Cost, Probeg)). Неудача (функторы area_owner и car_owner не равны).	Прямой ход к следующему предложению.

8	<p>T1 = owner(ryazanova, area_owner(Area, Area_Cost));</p> <p>T2 = owner(levushkin, building_owner(dacha, 90000)).</p> <p>Неудача (ryazanova не равно levushkin).</p>	Прямой ход к следующему предложению.
9	<p>T1 = owner(ryazanova, area_owner(Area, Area_Cost));</p> <p>T2 = owner(levushkin, area_owner(dacha_area, 9000)).</p> <p>Неудача (ryazanova не равно levushkin)</p>	Прямой ход к следующему предложению.
10	<p>T1 = owner(ryazanova, area_owner(Area, Area_Cost));</p> <p>T2 = owner(levushkin, warercraft_owner(boat, 90)).</p> <p>Неудача (ryazanova не равно levushkin)</p>	Прямой ход к следующему предложению.
11	<p>T1 = owner(ryazanova, area_owner(Area, Area_Cost));</p> <p>T2 = owner(samkov, building_owner(fitness_club, 888888)).</p> <p>Неудача (ryazanova не равно samkov).</p>	Прямой ход к следующему предложению.
12	<p>T1 = owner(ryazanova, area_owner(Area, Area_Cost));</p> <p>T2 = owner(samkov, area_owner(park, 89898989)).</p> <p>Неудача (ryazanova не равно samkov).</p>	Прямой ход к следующему предложению.
13	<p>T1 = owner(ryazanova, area_owner(Area, Area_Cost));</p> <p>T2 = owner(ryazanova, building_owner(moscow_city, 987654321)).</p> <p>Неудача (функторы area_owner и building_owner не равны).</p>	Прямой ход к следующему предложению.
14	<p>T1 = owner(ryazanova, area_owner(Area, Area_Cost));</p> <p>T2 = phone_list(levushkin, «89859771492», adr(moscow, kantemirovskaya, 5, 1)).</p> <p>Неудача (функторы owner и phone_list не равны).</p>	Прямой ход к следующему предложению. Аналогичная ситуация в следующих 9 предложениях (phone_list, auto, bank_list). Откат, переход к предыдущему состоянию резольвенты. Поиск с начала предложений.

24	<p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = get_own_cost(Sername, building, Cost).</p> <p>Неудача (функторы owner и get_own_cost не равны).</p>	<p>Прямой ход к следующему предложению. Аналогичная ситуация в следующих 5 предложениях. Прямой ход к следующему предложению</p>
30	<p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = owner(Sername, car_owner(Model, Color, Cost, Probeg)).</p> <p>Неудача (функторы building_owner и car_owner не равны).</p>	<p>Прямой ход к следующему предложению.</p>
31	<p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = owner(levushkin, building_owner(dacha, 90000)).</p> <p>Неудача (ryazanova не равно levushkin).</p>	<p>Прямой ход к следующему предложению.</p>
32	<p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = owner(levushkin, area_owner(dacha_area, 9000)).</p> <p>Неудача (ryazanova не равно levushkin)</p>	<p>Прямой ход к следующему предложению.</p>
33	<p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = owner(levushkin, warercraft_owner(boat, 90)).</p> <p>Неудача (ryazanova не равно levushkin)</p>	<p>Прямой ход к следующему предложению.</p>
34	<p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = owner(samkov, building_owner(fitness_club, 888888)).</p> <p>Неудача (ryazanova не равно samkov).</p>	<p>Прямой ход к следующему предложению.</p>

35	<p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = owner(samkov, area_owner(park, 89898989)).</p> <p>Неудача (ryazanova не равно samkov).</p>	Прямой ход к следующему предложению.
36	<p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = owner(ryazanova, building_owner(moscow_city, 987654321)).</p> <p>Успех. Подсановка {Building = moscow_city, Building_Cost = 987654321}.</p>	<p>Вывод:</p> <p>Building = moscow_city,</p> <p>Building_Cost = 987654321,</p> <p>Прямой ход к следующему предложению. Реконкретизация Building, Building_Cost.</p>
37	<p>T1 = owner(ryazanova, building_owner(Building, Building_Cost));</p> <p>T2 = phone_list(levushkin, «89859771492», adr(moscow, kantemirovskaya, 5, 1)).</p> <p>Неудача (функторы owner и phone_list не равны).</p>	Прямой ход к следующему предложению. Аналогичная ситуация в следующих 9 предложениях (phone_list, auto, bank_list). Откат, переход к предыдущему состоянию резольвенты. Поиск с начала предложений.
47	<p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = get_own_cost(Sername, building, Cost).</p> <p>Неудача (функторы owner и get_own_cost не равны).</p>	Прямой ход к следующему предложению. Аналогичная ситуация в следующих 5 предложениях. Прямой ход к следующему предложению
53	<p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = owner(Sername, car_owner(Model, Color, Cost, Probeg)).</p> <p>Неудача (функторы watercraft_owner и car_owner не равны).</p>	Прямой ход к следующему предложению.

54	<p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = owner(levushkin, building_owner(dacha, 90000)).</p> <p>Неудача (ryazanova не равно levushkin).</p>	Прямой ход к следующему предложению.
55	<p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = owner(levushkin, area_owner(dacha_area, 9000)).</p> <p>Неудача (ryazanova не равно levushkin)</p>	Прямой ход к следующему предложению.
56	<p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = owner(levushkin, warercraft_owner(boat, 90)).</p> <p>Неудача (ryazanova не равно levushkin)</p>	Прямой ход к следующему предложению.
57	<p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = owner(samkov, building_owner(fitness_club, 888888)).</p> <p>Неудача (ryazanova не равно samkov).</p>	Прямой ход к следующему предложению.
58	<p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = owner(samkov, area_owner(park, 89898989)).</p> <p>Неудача (ryazanova не равно samkov).</p>	Прямой ход к следующему предложению.
59	<p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = owner(ryazanova, building_owner(moscow_city, 987654321)).</p> <p>Неудача (функторы watercraft_owner и building_owner не равны).</p>	Прямой ход к следующему предложению.

60	<p>T1 = owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));</p> <p>T2 = phone_list(levushkin, «89859771492», adr(moscow, kantemirovskaya, 5, 1)).</p> <p>Неудача (функторы owner и phone_list не равны).</p>	<p>Прямой ход к следующему предложению. Аналогичная ситуация в следующих 9 предложениях (phone_list, auto, bank_list). Откат, переход к предыдущему состоянию резольвенты. Поиск с начала предложений.</p>
70	<p>T1 = owner(ryazanova, car_owner(Car, _, Car_Cost, _));</p> <p>T2 = get_own_cost(Sername, building, Cost).</p> <p>Неудача (функторы owner и get_own_cost не равны).</p>	<p>Прямой ход к следующему предложению. Аналогичная ситуация в следующих 5 предложениях. Прямой ход к следующему предложению</p>
76	<p>T1 = owner(ryazanova, car_owner(Car, _, Car_Cost, _));</p> <p>T2 = owner(Sername, car_owner(Model, Color, Cost, Probeg)).</p> <p>Успех. Подстановка {ryazanova = Sername, Car = Model, Car_Cost = Cost}.</p>	<p>Прямой ход к auto(ryazanova, Car, _, Car_Cost, _). Поиск с начала предложений.</p>
77	<p>T1 = auto(ryazanova, Car, _, Car_Cost, _);</p> <p>T2 = get_own_cost(Sername, building, Cost).</p> <p>Неудача (функторы auto и get_own_cost не равны).</p>	<p>Прямой ход к следующему предложению. Аналогичная ситуация в следующих 15 предложениях (get_own_cost, get_sum_cost_owner, owner, phone_list). Прямой ход к следующему предложению</p>
93	<p>T1 = auto(ryazanova, Car, _, Car_Cost, _);</p> <p>T2 = auto(levushkin, volvo, grey, 3000000, 1000).</p> <p>Неудача (ryazanova не равно levushkin).</p>	<p>Прямой ход к следующему предложению.</p>
94	<p>T1 = auto(ryazanova, Car, _, Car_Cost, _);</p> <p>T2 = auto(samkov, volkswagen, pink, 1000000, 99999).</p> <p>Неудача (ryazanova не равно samkov).</p>	<p>Прямой ход к следующему предложению.</p>

95	<p>T1 = auto(ryazanova, Car, _, Car_Cost, _);</p> <p>T2 = auto(ryazanova, bugatti, gold, 999999999, 1).</p> <p>Успех. Подстановка {Car = bugatti, Car_Cost = 999999999}.</p>	<p>Вывод: Car = bugatti, Car_Cost = 999999999. Прямой ход к следующему предложению, реконкретизация Car, Car_Cost.</p>
96	<p>T1 = auto(ryazanova, Car, _, Car_Cost, _);</p> <p>T2 = bank_list(levushkin, sberbank, 1111, 900000).</p> <p>Неудача (функторы auto и bank_list не равны).</p>	<p>Прямой ход к следующему предложению. Аналогичная ситуация в следующих 3 предложениях. Откат, переход к предыдущему состоянию резольвенты - owner(ryazanova, car_owner(Car, _, Car_Cost, _)), реконкретизация Car, Car_Cost.</p>
100	<p>T1 = owner(ryazanova, car_owner(Car, _, Car_Cost, _));</p> <p>T2 = owner(levushkin, building_owner(dacha, 90000).</p> <p>Неудача (ryazanova не равно levushkin).)</p>	<p>Прямой ход к следующему предложению. Аналогичная ситуация в следующих 2 предложениях. Прямой ход к следующему предложению.</p>
103	<p>T1 = owner(ryazanova, car_owner(Car, _, Car_Cost, _));</p> <p>T2 = owner(samkov, building_owner(fitness_club, 888888)).</p> <p>Неудача (ryazanova не равно samkov).</p>	<p>Прямой ход к следующему предложению. Аналогичная ситуация в следующем предложении. Прямой ход к следующему предложению.</p>
105	<p>T1 = owner(ryazanova, car_owner(Car, _, Car_Cost, _));</p> <p>T2 = owner(ryazanova, building_owner(moscow_city, 987654321)).</p> <p>Неудача (функторы car_owner и building_owner не равны).</p>	<p>Прямой ход к следующему предложению.</p>



106	T1 = owner(ryazanova, car_owner(Car, _, Car_Cost, _)); T2 = phone_list(levushkin, «89859771492», adr(moscow, kantemirovskaya, 5, 1)). Неудача (функторы owner и phone_list не равны).	Прямой ход к следующему предложению. Аналогичная ситуация в следующих 9 предложениях (phone_list, auto, bank_list). Откат, переход к предыдущему состоянию резольвенты. Конец clauses. Опустошение резольвенты. Завершение работы (115 шагов).

## Замечание

*Правило:*

*get\_sum\_cost\_owner(Sername, Cost) :-*

*get\_own\_cost(Sername, building, Building\_Cost),*

*get\_own\_cost(Sername, area, Area\_Cost),*

*get\_own\_cost(Sername, watercraft, Watercraft\_Cost),*

*get\_own\_cost(Sername, car, Car\_Cost),*

*Cost = Building\_Cost + Area\_Cost + Watercraft\_Cost + Car\_Cost.*

*ИТЕРАЦИОННЫЙ способ мышления! Эффективнее было бы использовать один, универсальный предикат – «собственность»*

## Исправленный текст программы

### domains

```

town, street, sername, model,
watercraft, color, bank, area, building = symbol
phone = string
house, case, cost, probeg, account, money = integer
address = adr(town, street, house, case)
ownership = building_owner(building, cost) ;area_owner(area, cost)
;watercraft_owner(watercraft, cost)
;car_owner(model, color, cost, probeg)

```

### predicates

```

phone_list(sername, phone, address).
auto(sername, model, color, cost, probeg).

```

```

bank_list(sername, bank, account, money).

owner(sername, ownership).

get_own_cost(sername, ownership, cost, cost).

get_sum_cost_owner(sername, cost).

```

clauses

```

get_own_cost(Sername, building_owner(_, _), Cost, Result) :-
owner(Sername, building_owner(_, Building_Cost)),
New_cost = Cost + Building_Cost,
get_own_cost(Sername, area_owner(_, _), New_cost, Result),
!.

get_own_cost(Sername, building_owner(_, _), Cost, Result):-
get_own_cost(Sername, area_owner(_, _), Cost, Result),
!.

get_own_cost(Sername, area_owner(_, _), Cost, Result) :-
owner(Sername, area_owner(_, Area_Cost)),
New_cost = Cost + Area_Cost,
get_own_cost(Sername, watercraft_owner(_, _), New_cost, Result),
!.

get_own_cost(Sername, area_owner(_, _), Cost, Result):-
get_own_cost(Sername, watercraft_owner(_, _), Cost, Result),
!.

get_own_cost(Sername, watercraft_owner(_, _), Cost, Result) :-
owner(Sername, watercraft_owner(_, Watercraft_Cost)),
New_cost = Cost + Watercraft_Cost,
get_own_cost(Sername, car_owner(_, _, _, _), New_cost, Result),
!.

get_own_cost(Sername, watercraft_owner(_, _), Cost, Result):-
get_own_cost(Sername, car_owner(_, _, _, _), Cost, Result),
!.

get_own_cost(Sername, car_owner(_, _, _, _), Cost, Result) :-
owner(Sername, car_owner(_, _, Car_Cost, _)),

```

```

Result = Cost + Car_Cost,
!.

get_own_cost(_, car_owner(_, _, _, _), Cost, Cost):-
!.

get_sum_cost_owner(Sername, Cost) :-
get_own_cost(Sername, building_owner(_, _), 0, Cost).

owner(Sername, car_owner(Model, Color, Cost, Probeg)) :-
auto(Sername, Model, Color, Cost, Probeg).

owner(levushkin, building_owner(dacha, 90000)).
owner(levushkin, area_owner(dacha_area, 9000)).
owner(levushkin, watercraft_owner(boat, 90)).

owner(samkov, building_owner(fitness_club, 888888)).
owner(samkov, area_owner(park, 89898989)).

owner(ryazanova, building_owner(moscow_city, 987654321)).

phone_list(levushkin, "89859771492",
adr(moscow, kantemirovskaya, 5, 1)).
phone_list(samkov, "898999999",
adr(chelyabinsk, pushkinskaya, 4, 2)).
phone_list(ryazanova, "8911911911",
adr(moscow, baumanskaya, 9, 9)).

auto(levushkin, volvo, grey, 3000000, 1000).
auto(samkov, volkswagen, pink, 1000000, 99999).
auto(ryazanova, bugatti, gold, 999999999, 1).

bank_list(levushkin, sberbank, 1111, 900000).
bank_list(samkov, sberbank, 2222, 100).
bank_list(ryazanova, tinkoff, 3333, 99999999).

```

```

goal
bank_list(ryazanova, raiffeisen, 4444, 888888888).

%Названия всех объектов собственности по заданному субъекту
%Пример 1
owner(levushkin, area_owner(Area, _));
owner(levushkin, building_owner(Building, _));
owner(levushkin, watercraft_owner(Watercraft, _));
owner(levushkin, car_owner(Car, _, _, _)).

%Пример 2
owner(samkov, area_owner(Area, _));
owner(samkov, building_owner(Building, _));
owner(samkov, watercraft_owner(Watercraft, _));
owner(samkov, car_owner(Car, _, _, _)).

%Пример 3
owner(ryazanova, area_owner(Area, _));
owner(ryazanova, building_owner(Building, _));
owner(ryazanova, watercraft_owner(Watercraft, _));
owner(ryazanova, car_owner(Car, _, _, _)).

%Названия и стоимости всех объектов собственности
%по заданному субъекту
%Пример 1
owner(levushkin, area_owner(Area, Area_Cost));
owner(levushkin, building_owner(Building, Building_Cost));
owner(levushkin, watercraft_owner(Watercraft, Watercraft_Cost));
owner(levushkin, car_owner(Car, _, Car_Cost, _)).

%Пример 2
owner(samkov, area_owner(Area, Area_Cost));
owner(samkov, building_owner(Building, Building_Cost));
owner(samkov, watercraft_owner(Watercraft, Watercraft_Cost));
owner(samkov, car_owner(Car, _, Car_Cost, _)).

%Пример 3
owner(ryazanova, area_owner(Area, Area_Cost));
owner(ryazanova, building_owner(Building, Building_Cost));
owner(ryazanova, watercraft_owner(Watercraft, Watercraft_Cost));
owner(ryazanova, car_owner(Car, _, Car_Cost, _)).

```

## Замечание

Знания размещены в *CLAUSES* в виде предложений. Они представляют собой знания о некоторой предметной области, формально – отношения между различными объектами. Не четко!

## Исправление

*Вопрос:*

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

*Ответ:*

Знания размещены в CLAUSES в виде предложений.

Они представляют собой знания о некоторой предметной области, формально – утверждения : факты и правила.

## Замечание

*С помощью описанных предикатов, можно создавать предложения в базе знаний.*

*Предикаты используются для представления, как фактов, так и правил. Звучит неверно*

## Исправление

*Вопрос:*

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

*Ответ:*

В разделе PREDICATES описываются предикаты, их аридность (местность) и домены (типы и природа аргументов).

С помощью описанных предикатов, описываются процедуры в разделе CLAUSES.

Процедурой называется совокупность правил, заголовки которых имеют одно и то же имя и одну и ту же аридность (местность), т.е. это совокупность правил, описывающих одно определенное отношение.

## Исправление ошибок 16-ой лабораторной работы

### Замечание

*tatiana = tatiana константы в подстановку не заносятся!*

Исправленная таблица для вопроса: «По имени субъекта определить его бабушку и дедушку по материнской линии (предки 2-го колена)»

*Текст программы:*

predicates

```
first_ancestors(symbol Child, symbol Father, symbol Mother)
```

```
second_ancestors(symbol Child, symbol Grandfather_f,  
symbol Grandmother_f, symbol Grandfather_m, symbol Grandmother_m)
```

clauses

```
first_ancestors(ilya, kirill, tatiana).  
first_ancestors(kirill, sergei, nadya).  
first_ancestors(tatiana, tolya, luba).
```

```
first_ancestors(vasilisa, yura, sveta).  
first_ancestors(yura, andrei, zina).  
first_ancestors(sveta, kolya, gerda).
```

```
second_ancestors(Child, GF_f, GM_f, GF_m, GM_m) :-  
first_ancestors(Child, Father, Mother),  
first_ancestors(Father, GF_f, GM_f),  
first_ancestors(Mother, GF_m, GM_m).
```

goal

```
second_ancestors(ilya, _, _, Grandfather_m, Grandmother_m).
```

*%Вывод:*

```
Grandfather_m=tolya, Grandmother_m=luba  
1 Solution
```

*Порядок работы системы:*

№ ша-га	Состояние резольвенты, и вывод: дальнейшие действия (почему?)	Для каких термов запускается алгоритм унификации: T1=T2 и каков <b>результат</b> (и подстановка)	Дальнейшие действия: прямой ход или откат (почему и к чему приводит?)

1	second_ancestors (ilya, _, _, Grandfather_m, Grandmother_m)	T1 = second_ancestors (ilya, _, _, Grandfather_m, Grandmother_m); T2 = first_ancestors (ilya, kirill, tatiana). Неудача (функто- ры second_ancestors и first_ancestors не равны)	Прямой ход к следую- щему предложению. Аналогичная ситуация в следующих 5 пред- ложениях. Переход к следующему предложе- нию.
7	second_ancestors (ilya, _, _, Grandfather_m, Grandmother_m)	T1 = second_ancestors (ilya, _, _, Grandfather_m, Grandmother_m); T2 = second_ancestors (Child, GF_f, GM_f, GF_m, GM_m). Успех. Подстанов- ка {ilya = Child, Grandfather_m = GF_m, Grandmother_m = GM_m}.	Прямой ход: first_ancestors(ilya, Father, Mother). Поиск с начала предложений.
8	first_ancestors (ilya, Father, Mother), first_ancestors (Father, _, _), first_ancestors (Mother, GF_m, GM_m).	T1 = first_ancestors (ilya, Father, Mother), T2 = first_ancestors (ilya, kirill, tatiana). Успех. Подстанов- ка {Father=kirill, Mother=tatiana}.	Прямой ход к first_ancestors (kirill, _, _). Прямой ход к first_ancestors (tatiana, GF_m, GM_m). Поиск с начала предложений.
9	first_ancestors (tatiana, GF_m, GM_m).	T1 = first_ancestors (tatiana, GF_m, GM_m), T2 = first_ancestors (ilya, kirill, tatiana). Неудача, tatiana не равно ilya.	Прямой ход к следую- щему предложению.





15	first_ancestors (tatiana, GF_m, GM_m).	T1 = first_ancestors (tatiana, GF_m, GM_m), T2 = second_ancestors (Child, GF_f, GM_f, GF_m, GM_m). Неудача (функции first_ancestors и second_ancestors не равны).	Откат, переход к предыдущему состоя- нию резольвенты (шаг 8), реконкретизация Father, Mother.
16	first_ancestors (ilya, Father, Mother), first_ancestors (Father, _, _), first_ancestors (Mother, GF_m, GM_m).	T1 = first_ancestors (ilya, Father, Mother), T2 = first_ancestors (kirill, sergei, nadya). Неудача, ilya не равно kirill.	Прямой ход к следую- щему предложению.
17	first_ancestors (ilya, Father, Mother), first_ancestors (Father, _, _), first_ancestors (Mother, GF_m, GM_m).	T1 = first_ancestors (ilya, Father, Mother), T2 = first_ancestors (tatiana, tolya, luba). Неудача, ilya не равно tatiana.	Прямой ход к следую- щему предложению.
18	first_ancestors (ilya, Father, Mother), first_ancestors (Father, _, _), first_ancestors (Mother, GF_m, GM_m).	T1 = first_ancestors (ilya, Father, Mother), T2 = first_ancestors (vasilisa, yura, sveta). Неудача, ilya не равно vasilisa.	Прямой ход к следую- щему предложению.

19	first_ancestors (ilya, Father, Mother), first_ancestors (Father, _, _), first_ancestors (Mother, GF_m, GM_m).	T1 = first_ancestors (ilya, Father, Mother), T2 = first_ancestors (yura, andrei, zina). Неудача, ilya не равно yura.	Прямой ход к следующему предложению.
20	first_ancestors (ilya, Father, Mother), first_ancestors (Father, _, _), first_ancestors (Mother, GF_m, GM_m).	T1 = first_ancestors (ilya, Father, Mother), T2 = first_ancestors (sveta, kolya, gerda). Неудача, ilya не равно sveta.	Прямой ход к следующему предложению.
21	first_ancestors (ilya, Father, Mother), first_ancestors (Father, _, _), first_ancestors (Mother, GF_m, GM_m).	T1 = first_ancestors (ilya, Father, Mother), T2 = second_ancestors (Child, GF_f, GM_f, GF_m, GM_m). Неудача (функции first_ancestors и second_ancestors не равны).	Откат, переход к предыдущему состоянию резольвенты (шаг 7). Реконкретизация Grandfather_m, Grandmother_m.
22	second_ancestors (ilya, _, _, Grandfather_m, Grandmother_m); конец clauses; опустошение резольвенты; завершение работы.		

## Замечание

*Каковы назначение и результат использования алгоритма унификации?*

*Назначение алгоритма унификации заключается в попарном сопоставлении термов и попытке построить для них общий пример. Ради чего?*

### Исправление

Назначение использования алгоритма унификации двух термов состоит в том, чтобы подобрать нужное в данный момент правило.

### Замечание

*Результатом использования алгоритма унификации может быть успех или тупиковая ситуация (неудача). И все???*

### Исправление

Результатом использования алгоритма унификации может быть успех согласования базы знаний и вопроса, в качестве побочного эффекта формируется подстановка; или может быть тупиковая ситуация (неудача).

## Исправление ошибок 14-ой лабораторной работы

### Замечание

*Ниже приведена исправленная таблица:*

*Сравниваются два терма! Откуда берется на шаге 1 в стеке?:*

*auto(Sername, bugatti, gold, \_, \_) = auto(samkov, bugatti, gold, \_, \_),  
auto(Sername, bugatti, gold, \_, \_) = auto(ryazanova, bugatti, gold, \_, \_),  
bank\_list(Sername, Bank, \_, \_), phone\_list(Sername, Phone, \_, \_). И т.д.???*

**Исправленная таблица, демонстрирующая порядок работы алгоритма унификации для вопроса и подходящего заголовка правила**

*Вопрос:*

goal

get\_info\_by\_model\_color(bugatti, gold, Sername, Phone, Bank)

*Заголовок правила:*

get\_info\_by\_model\_color(Model, Color, Sername, Phone, Bank)

№ ша- га	Результирующая ячейка	Рабочее поле	П. алг.	Стек
0			1	get_info_by_model_color(bugatti, gold, Surname, Phone, Bank)=get_info_by_model_color(Model, Color, Surname, Phone, Bank).
1		get_info_by_model_color(bugatti, gold, Surname, Phone, Bank)=get_info_by_model_color(Model, Color, Surname, Phone, Bank).	e	bugatti = Model gold = Color Surname = Surname Phone = Phone Bank = Bank
2	bugatti = Model	bugatti = Model	r	gold = Color Surname = Surname Phone = Phone Bank = Bank
3	bugatti = Model, gold = Color	gold = Color	r	Surname = Surname Phone = Phone Bank = Bank
4	bugatti = Model, gold = Color Surname = Surname	Surname = Surname	r	Phone = Phone Bank = Bank
5	bugatti = Model, gold = Color Surname = Surname Phone = Phone	Phone = Phone	r	Bank = Bank

6	bugatti = Bank Model, gold = Color Sername = Sername Phone = Phone Bank = Bank		г	
Вы-подстановка вод:		Т.к. стек пуст – успех и в рез. ячейке подстанов- ка		