



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 № 14

*По предмету: «Логическое и Функциональное
Программирование»*

Тема:

Работа программы на Prolog

Студент: Юмаев Артур Русланович
Группа: ИУ7-65Б

Задание

Используя 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.

Владелец может иметь несколько телефонов, автомобилей, вкладов (Факты). В разных городах есть однофамильцы, в одном городе – фамилия уникальна.

Используя **конъюнктивное правило и простой вопрос**, обеспечить возможность поиска:

По Марке и Цвету автомобиля найти Фамилию, Город, Телефон и Банки, в которых владелец автомобиля имеет вклады. Лишней информации не находить и не передавать!!!

Владельцев может быть **несколько** (не более 3-х), **один и ни одного**.

1. Для каждого из трех вариантов **словесно подробно** описать порядок формирования ответа (в виде таблицы). При этом, указать – отметить моменты очередного запуска алгоритма унификации и полный результат его работы. Обосновать следующий шаг работы системы. Выписать унификаторы – подстановки. Указать моменты, причины и результат отката, если он есть.
2. Для случая нескольких владельцев (2-х):

приведите примеры (таблицы) работы системы **при разных порядках** следования в БЗ процедур, и знаний в них: («Телефонный справочник», «Автомобили», «Вкладчики банков», или:

«Автомобили», «Вкладчики банков», «Телефонный справочник»). Сделайте **вывод**:

Одинаковы ли: множество работ и объем работ в разных случаях?

Оформите 2 таблицы, демонстрирующие **порядок работы алгоритма унификации** вопроса и подходящего заголовка правила (для двух случаев из пункта 2) и укажите результаты его работы: ответ и побочный эффект.

Программа – база знаний

domains

```
address=address(symbol city, symbol street, integer house, integer apNum)
sname, bank, account, phoneNum, city, street, brand, color=symbol
price, amount=real
```

predicates

```
record(sname, phoneNum, address)
owns(sname, brand, color, price)
hasBankDeposit(sname, bank, account, amount)
find(brand, color, sname, city, phoneNum, bank)
```

clauses

```
owns(yumaev, toyota, white, 14000).
owns(petrov, toyota, white, 14000).
owns(vetrov, toyota, white, 14000).
owns(vetrov, kia, yellow, 13000).
owns(zhorin, mercedes, blue, 18000).
```

```
record(yumaev, "89036609896", address(moscow, yuzhnaya, 15, 22)).
record(yumaev, "89036600505", address(vologda, yuzhnaya, 15, 22)).
record(petrov, "89068887745", address(moscow, prospekt_mira, 7, 24)).
record(vetrov, "89068886695", address(moscow, vetrovaya, 36, 78)).
record(zhorin, "89068887745", address(ryazan, lubanka, 36, 78)).
record(zhorin, "89068880045", address(ryazan, lubanka, 36, 78)).
```

```
hasBankDeposit(yumaev, rocket_bank, "123456789", 12345).
hasBankDeposit(petrov, rocket_bank, "326856789", 15000).
hasBankDeposit(zhorin, otkritie, "123456000", 45678).
hasBankDeposit(zhorin, alpha_bank, "326556000", 80000).
```

% Rules

```
find(Brand, Color, Sname, City, PhoneNum, Bank) :- owns(Sname, Brand, Color, _),
                                                    record(Sname, PhoneNum, address(City, _, _, _)),
                                                    hasBankDeposit(Sname, Bank, _, _).
```

goal

```
% 1
% find(toyota, white, Sname, City, PhoneNum, Bank).

%2
% find(mercedes, blue, Sname, City, PhoneNum, Bank).

% 3
find(kia, yellow, Sname, City, PhoneNum, Bank).
```

Таблицы унификации

1. find(toyota, white, Sname, City, PhoneNum, Bank).

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	<p>Сопоставляем: find(toyota, white, Sname, City, PhoneNum, Bank). find(Brand, Color, Sname, City, PhoneNum, Bank)</p> <ol style="list-style-type: none"> 1. Кладем в стек toyota=Brand, white=Color, Sname=Sname, City=City, PhoneNum=PhoneNum, Bank=Bank. 2. Достаем из стека: toyota=Brand – константа и именованная свободная переменная (добавить в результирующую ячейку Brand=toyota), переход к следующему шагу. 3. Достаем из стека: white=Color – константа и именованная свободная переменная (добавить в результирующую ячейку Color=white) переход к следующему шагу. 4. Достаем из стека Sname=Sname – две свободные именованные переменные (связанные переменные). 5. Достаем из стека City=City – две свободные именованные переменные (связанные переменные). 6. Достаем из стека PhoneNum=PhoneNum – две свободные именованные переменные (связанные переменные). 7. Достаем из стека Bank=Bank – две свободные именованные переменные (связанные переменные). 8. Очистка рабочего поля. Неудача=0 <p>Результат: Brand=toyota, Color=white</p>	<p>Унификация выполнена успешно, прямой ход. Переходим к owns (Sname, toyota, white, _),</p>
2	<p>Сопоставляем: owns (Sname, toyota, white, _) owns(yumaev, toyota, white, 14000).</p> <ol style="list-style-type: none"> 1. Кладем в стек Sname=yumaev, toyota=toyota, white=white. 2. Достаем из стека Sname=yumaev – константа и именованная свободная переменная (добавить в результирующую ячейку Sname=yumaev), переход к следующему шагу. 3. Достаем из стека toyota=toyota – равные константы, следующий шаг 4. Достаем из стека white=white – равные константы, следующий шаг 5. Стек пуст, очистка рабочего поля. Неудача=0 <p>Результат: Sname=yumaev</p>	<p>Унификация выполнена успешно, прямой ход. Переходим к record(yumaev, PhoneNum, address(City, _, _)),</p>
3	<p>Сопоставляем: record(yumaev, PhoneNum, address(City, _, _)) record(yumaev, “89036609896”, address(moscow, yuzhnaya, 15, 22)).</p> <ol style="list-style-type: none"> 1. Кладем в стек yumaev=yumaev, PhoneNum=“89036609896”, address(City, _, _)=address(moscow, yuzhnaya, 15, 22). 2. Достаем из стека yumaev=yumaev – равные константы, следующий шаг 3. Достаем из стека PhoneNum=“89036609896” – константа и именованная свободная переменная, (добавить в результирующую ячейку PhoneNum=“89036609896”), переход к следующему шагу 4. Достаем из стека address(City, _, _)=address(moscow, yuzhnaya, 15, 22). – составные термы – кладем в стек moscow=City. 6. Достаем из стека moscow=City – константа и именованная свободная переменная (добавить в результирующую ячейку City=moscow) переход к следующему шагу. 5. Стек пуст, очистить поле. Неудача=0. <p>Результат: PhoneNum=“89036609896”, City=moscow</p>	<p>Унификация выполнена успешно, прямой ход. Переходим к hasBankDeposit(yumaev, Bank, _, _).</p>

4	Сопоставляем: hasBankDeposit(yumaev, Bank, _, _). hasBankDeposit(yumaev, rocket_bank, "123456789", 12345). 1. Кладем в стек yumaev=yumaev, Bank=rocket_bank. 2. Достаем из стека yumaev=yumaev - равные константы, следующий шаг 3. Достаем из стека Bank=rocket_bank - константа и именованная свободная переменная, (добавить в результирующую ячейку Bank=rocket_bank), переход к следующему шагу 4. Стек пустой. Очистка поля. Неудача=0. Результат: Bank=rocket_bank	Унификация выполнена успешно, Sname=yumaev PhoneNum="89036609896", City="Moscow", Brand=toyota, Color=white, Bank=rocket_bank. Обратный ход
---	---	---

Составные термы унифицируемы, если

1. Имеют одинаковые главные функторы,
2. Имеют одинаковую арность.

2. Поиск унифицируемых термов

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	Сопоставляем: find(toyota, white, Sname, City, PhoneNum, Bank). record(yumaev, "89036609896", address(moscow, yuzhnaya, 15, 22)). Кладем в стек find(...) = record(...). Вытаскиваем из стека find(...) = record(...). Термы составные: главные функторы и арность не совпадают – унификация невозможна.	Прямой ход, переход к следующему предложению. (по аналогии, до конца процедуры record)
2	Сопоставляем: find(toyota, white, Sname, City, PhoneNum, Bank). owns(yumaev, toyota, white, 14000). Кладем в стек find(...) = owns(...). Вытаскиваем из стека find(...) = owns(...). Термы составные: главные функторы и арность не совпадают – унификация невозможна.	Прямой ход, переход к следующему предложению. (по аналогии, до конца процедуры owns)
3	Сопоставляем: find(toyota, white, Sname, City, PhoneNum, Bank). hasBankDeposit(yumaev, rocket_bank, "123456789", 12345). Кладем в стек find(...) = hasBankDeposit(...). Вытаскиваем из стека find(...) = hasBankDeposit(...). Термы составные: главные функторы и арность не совпадают – унификация невозможна.	Прямой ход, переход к следующему предложению. (по аналогии, до конца процедуры hasBankDeposit)
4	Сопоставляем: find(toyota, white, Sname, City, PhoneNum, Bank). find(Brand, Color, Sname, City, PhoneNum, Bank) кладем в стек: find(...) = find(...). Вытаскиваем из стека: find(...) = find(...). Составные термы: имена главных функторов совпали, арность совпала – термы унифицируемы. В стек заносятся: Brand=toyota, Color=white, Sname=Sname, City=City, PhoneNum=PhoneNum, Bank= Bank. Результат унификации: успех Наиболее общий унификатор: Brand=toyota, Color=white.	Прямой ход. Переход в тело правила.

3. Таблица для одного автомобиля

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
1	Сопоставляем: owns(Sname, toyota, white, _), с термами аналогично терму find (см. Таблица 2), пока не будет найден терм удовлетворяющий условиям унификации.	Прямой ход, переход к следующему предложению. (по аналогии, до конца процедуры record)
2	Сопоставляем: owns(Sname, toyota, white, _), owns(yumaev, toyota, white, 14000). Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: Sname=yumaev.	Прямой ход, переход к следующей подцели правила.
3	Сопоставляем: record(yumaev, PhoneNum, address(City, _, _)) record(yumaev, “89036609896”, address(moscow, yuzhnaya, 15, 22)). Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: PhoneNum=“89036609896”, City=moscow.	Прямой ход, переход к следующей подцели правила.
4	Сопоставляем: hasBankDeposit(yumaev, Bank, _, _). с термами аналогично терму find (см. Таблица 2), пока не будет найден терм удовлетворяющий условиям унификации.	Прямой ход, переход к следующему предложению. (по аналогии, до конца процедур record, owns)
5	Сопоставляем: hasBankDeposit(yumaev, Bank, _, _). hasBankDeposit(yumaev, rocket_bank, “123456789”, 12345). Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: Bank=rocket_bank.	Истинное тело найдено. Sname=yumaev, PhoneNum=“89036609896”, City=moscow, Bank=rocket_bank. Откат, переход к следующему предложению.
6	Сопоставляем: hasBankDeposit(yumaev, Bank, _, _). hasBankDeposit(petrov, rocket_bank, “326856789”, 15000). Составные термы, унифицируемые. Результат унификации: неудача, константы yumaev и petrov не совпадают.	Унификация завершилась неудачей, прямой ход, переход к следующему предложению.
7	Сопоставляем: hasBankDeposit(yumaev, Bank, _, _). с термами в процедуре hasBankDeposit. Все термы унифицируемы, но унификация завершается неудачей.	Унификация завершилась неудачей. Откат
8	Сопоставляем: record(yumaev, PhoneNum, address(City, _, _)) record(yumaev, “89036600505”, address(vologda, yuzhnaya, 15, 22)) Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: PhoneNum=“89036600505”, City=vologda.	Прямой ход, переход к следующей подцели правила

9	Сопоставляем: hasBankDeposit(yumaev, Bank, _, _). hasBankDeposit(yumaev, rocket_bank, "123456789", 12345). Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: Bank=rocket_bank.	Истинное тело найдено. Sname=yumaev, PhoneNum="89036600505", City=vologda, Bank=rocket_bank. Откат, переход к следующему предложению.
10	Сопоставляем: hasBankDeposit(yumaev, Bank, _, _). с термами в процедуре hasBankDeposit. Все термы унифицируемы, но унификация завершается неудачей.	Унификация завершилась неудачей. Откат
11	Сопоставляем: record(yumaev, PhoneNum, address(City, _, _, _)) с термами в процедуре record. Все термы унифицируемы, но унификация завершается неудачей.	Унификация завершилась неудачей. Откат
12	Сопоставляем: owns(Sname, toyota, white, _), owns(petrov, toyota, white, 14000). Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: Sname=petrov.	Прямой ход, переход к следующей подцели правила.
13	Сопоставляем: record(petrov, PhoneNum, address(City, _, _, _)) с термами в процедуре record, пока не найдем успешно унифицируемый терм, либо пока не закончится БЗ.	Прямой ход, переход к следующему предложению.
14	Сопоставляем: record(petrov, PhoneNum, address(City, _, _, _)) record(petrov, "89068887745", address(moscow, prospekt_mira, 7, 24)) Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: PhoneNum="89068887745", City=moscow.	Прямой ход, переход к следующей подцели правила.
15	Сопоставляем: hasBankDeposit(petrov, Bank, _, _) с термами в процедуре hasBankDeposit, пока не найдем успешно унифицируемый терм, либо пока не закончится БЗ.	Прямой ход, переход к следующему предложению.
16	Сопоставляем: hasBankDeposit(petrov, Bank, _, _). hasBankDeposit(petrov, rocket_bank, "326856789", 15000). Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: Bank=rocket_bank.	Истинное тело найдено. Sname=petrov, PhoneNum="89068887745", City=moscow, Bank=rocket_bank. Откат, переход к следующему предложению
17	Сопоставляем: hasBankDeposit(yumaev, Bank, _, _). с термами в процедуре hasBankDeposit. Все термы унифицируемы, но унификация завершается неудачей.	Унификация завершилась неудачей. Откат

18	Сопоставляем: record(yumaev, PhoneNum, address(City, _, _)) с термами в процедуре record. Все термы процедуры унифицируемы, но унификация завершается неудачей.	Унификация завершилась неудачей. Откат
19	Сопоставляем: owns(Sname, toyota, white, _) с термами в процедуре record. Все термы процедуры унифицируемы, но унификация завершается неудачей. Далее до конца БЗ термы не унифицируемы.	Унификация завершилась неудачей. Откат
20	Сопоставляем: find(toyota, white, Sname, City, PhoneNum, Bank), с термами до конца БЗ, унифицируемых термов не найдено.	Конец

Цель: find(mercedes, blue, Sname, City, PhoneNum, Bank).

Таблица 4. Пример работы программы на цели для 1 владельца автомобиля.

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
0	Сопоставляем: find(mercedes, blue, Sname, City, PhoneNum, Bank). find(Brand, Color, Sname, City, PhoneNum, Bank) кладем в стек: find(...) = find(...). Вытаскиваем из стека: find(...) = find(...). Составные термы: имена главных функторов совпали, аргументы совпали – термы унифицируемы. В стек заносятся: Brand=mercedes, Color=blue, Sname= Sname, City=City, PhoneNum=PhoneNum, Bank= Bank. Результат унификации: успех Наиболее общий унификатор: Brand=mercedes, Color=blue.	Прямой ход. Переход в тело правила.
1	Сопоставляем: owns(Sname, mercedes, blue, _, _) с термами аналогично терму find (см. Таблица 2), пока не будет найден терм удовлетворяющий условиям унификации.	Прямой ход, переход к следующему предложению. (по аналогии, до конца процедуры record)
2	Сопоставляем: owns(Sname, mercedes, blue, _, _), owns(zhorin, mercedes, blue, 18000) Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: Sname=zhorin.	Прямой ход, переход к следующей подцели правила.
3	Сопоставляем: record(zhorin, PhoneNum, address(City, _, _)) record(zhorin, “89068880045”, address(ryazan, lubanka, 36, 78)). Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: PhoneNum=“89068880045”, City=ryazan.	Прямой ход, переход к следующей подцели правила.

4	Сопоставляем: hasBankDeposit(zhorin, Bank, _, _). с термами аналогично терму find (см. Таблица 2), пока не будет найден терм удовлетворяющий условиям унификации.	Прямой ход, переход к следующему предложению. (по аналогии, до конца процедур record, owns)
5	Сопоставляем: hasBankDeposit(zhorin, Bank, _, _). hasBankDeposit(zhorin, otkritie, "123456000", 45678). Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: Bank=otkritie.	Истинное тело найдено. Sname=zhorin, PhoneNum="89068880045", City=gyazan, Bank=otkritie. Откат, переход к следующему предложению.
6	Сопоставляем: hasBankDeposit(zhorin, Bank, _, _). с термами в процедуре hasBankDeposit. Все термы унифицируемы, но унификация завершается неудачей.	Унификация завершилась неудачей. Откат
7	Сопоставляем: record(zhorin, PhoneNum, address(City, _, _)) с термами в процедуре record. Все термы процедуры унифицируемы, но унификация завершается неудачей.	Унификация завершилась неудачей. Откат
8	Сопоставляем: owns(Sname, mercedes, blue, ..., ...), с термами в процедуре record. Все термы процедуры унифицируемы, но унификация завершается неудачей. Далее до конца БЗ термы не унифицируемы.	Унификация завершилась неудачей. Откат
9	Сопоставляем: find(mercedes, blue, Sname, City, PhoneNum, Bank), с термами до конца БЗ, унифицируемых термов не найдено.	Конец

Цель: find(kia, yellow, Sname, City, PhoneNum, Bank).

Таблица 5. Пример работы программы на цели для машины с 1 владельцем, без вклада.

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
0	Сопоставляем: find(kia, yellow, Sname, City, PhoneNum, Bank) find(Brand, Color, Sname, City, PhoneNum, Bank) кладем в стек: find(...) = find(...). Вытаскиваем из стека: find(...) = find(...). Составные термы: имена главных функторов совпали, аргументы совпали – термы унифицируемы. В стек заносятся: Brand=kia, Color=yellow, Sname= Sname, City=City, PhoneNum=PhoneNum, Bank= Bank. Результат унификации: успех Наиболее общий унификатор: Brand=kia, Color=yellow.	Прямой ход. Переход в тело правила.

1	Сопоставляем: owns(Sname, kia, yellow, ...), с термами аналогично терму find (см. Таблица 2), пока не будет найден терм удовлетворяющий условиям унификации.	Прямой ход, переход к следующему предложению. (по аналогии, до конца процедуры record)
2	Сопоставляем: owns(Sname, kia, yellow, _), owns(vetrov, kia, yellow, 13000). Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: Sname=vetrov.	Прямой ход, переход к следующей подцели правила.
3	Сопоставляем: record(vetrov, PhoneNum, address(City, _, _, _)) record(vetrov, "89068886695", address(moscow, vetrovaya, 36, 78)). Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: PhoneNum= "89068886695", City=moscow.	Прямой ход, переход к следующей подцели правила.
4	Сопоставляем: hasBankDeposit(vetrov, Bank, _, _). с термами пока не будет найден подходящий терм, либо пока не закончится БЗ. БЗ закончилась. Унифицируемых термов найдено не было.	Откат
5	Сопоставляем: record(vetrov, PhoneNum, address(City, _, _, _)) с термами в процедуре record. Все термы процедуры унифицируемы, но унификация завершается неудачей.	Унификация завершилась неудачей. Откат
6	Сопоставляем: owns(Sname, kia, yellow, _), с термами в процедуре record. Все термы процедуры унифицируемы, но унификация завершается неудачей. Далее до конца БЗ термы не унифицируемы.	Унификация завершилась неудачей. Откат
7	Сопоставляем: find(kia, yellow, Sname, City, PhoneNum, Bank), с термами до конца БЗ, унифицируемых термов не найдено.	0 Solutions

Задание 2

1) «Телефоны» «Автомобили» «Инвесторы»

Таблица 6.

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
0	Сопоставляем: find(mercedes, blue, Sname, City, PhoneNum, Bank). find(Brand, Color, Sname, City, PhoneNum, Bank) кладем в стек: find(...) = find(...).	Прямой ход. Переход в тело правила.

	<p>Вытаскиваем из стека: find(...) = find(...).</p> <p>Составные термы: имена главных функторов совпали, аргументы совпали – термы унифицируемы. В стек заносятся: Brand=mercedes, Color=blue, Sname= Sname, City=City, PhoneNum=PhoneNum, Bank= Bank.</p> <p>Результат унификации: успех</p> <p>Наиболее общий унификатор: Brand=mercedes, Color=blue.</p>	
1	<p>Сопоставляем:</p> <p>owns(Sname, mercedes, blue, _, _), с термами аналогично терму find (см. Таблица 2), пока не будет найден терм удовлетворяющий условиям унификации.</p>	<p>Прямой ход, переход к следующему предложению. (по аналогии, до конца процедуры record)</p>
2	<p>Сопоставляем:</p> <p>owns(Sname, mercedes, blue, _, _),</p> <p>owns(zhorin, mercedes, blue, 18000)</p> <p>Составные термы, унифицируемые.</p> <p>Результат унификации: успех</p> <p>Наиболее общий унификатор: Sname=zhorin.</p> <p>Для поиска было проведено 10 сопоставлений</p>	<p>Прямой ход, переход к следующей подцели правила.</p>
3	<p>Сопоставляем:</p> <p>record(zhorin, PhoneNum, address(City, _, _, _))</p> <p>record(zhorin, “89068880045”, address(ryazan, lubanka, 36, 78)).</p> <p>Составные термы, унифицируемые.</p> <p>Результат унификации: успех</p> <p>Наиболее общий унификатор: PhoneNum=“89068880045”, City=ryazan. Для поиска было проведено 6 сопоставлений.</p>	<p>Прямой ход, переход к следующей подцели правила.</p>
4	<p>Сопоставляем:</p> <p>hasBankDeposit(zhorin, Bank, _, _). с термами аналогично терму find (см. Таблица 2), пока не будет найден терм удовлетворяющий условиям унификации.</p>	<p>Прямой ход, переход к следующему предложению. (по аналогии, до конца процедур record, owns)</p>
5	<p>Сопоставляем:</p> <p>hasBankDeposit(zhorin, Bank, _, _).</p> <p>hasBankDeposit(zhorin, otkritie, “123456000”, 45678).</p> <p>Составные термы, унифицируемые.</p> <p>Результат унификации: успех</p> <p>Наиболее общий унификатор: Bank=otkritie.</p> <p>Для поиска было проведено 14 сопоставлений</p>	<p>Истинное тело найдено.</p> <p>Sname=zhorin,</p> <p>PhoneNum=“89068880045”,</p> <p>City=ryazan,</p> <p>Bank=otkritie.</p> <p>Откат, переход к следующему предложению.</p>
6	<p>Сопоставляем:</p> <p>hasBankDeposit(zhorin, Bank, _, _). с термами в процедуре hasBankDeposit. Все термы унифицируемы, но унификация завершается неудачей.</p> <p>Было проведено 2 сопоставления</p>	<p>Унификация завершилась неудачей. Откат</p>
7	<p>Сопоставляем:</p> <p>record(zhorin, PhoneNum, address(City, _, _, _)) с термами в процедуре record. Все термы процедуры унифицируемы, но унификация завершается неудачей.</p> <p>Было проведено 10 сопоставлений</p>	<p>Унификация завершилась неудачей. Откат</p>
8	<p>Сопоставляем:</p> <p>owns(Sname, mercedes, blue, _, _), с термами в процедуре record. Все термы процедуры унифицируемы, но унификация завершается неудачей. Далее до конца БЗ термы не унифицируемы.</p> <p>Было проведено 6 сопоставлений</p>	<p>Унификация завершилась неудачей. Откат</p>

9	Сопоставляем: find(mercedes, blue, Sname, City, PhoneNum, Bank), с термами до конца БЗ, унифицируемых термов не найдено.	Конец
---	--	-------

Итого было проведено 48 сопоставлений.

2) «Автомобили» «Телефоны» «Инвесторы»

Таблица 6.

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат (к чему приводит?)
0	Сопоставляем: find(mercedes, blue, Sname, City, PhoneNum, Bank). find(Brand, Color, Sname, City, PhoneNum, Bank) кладем в стек: find(...) = find(...). Вытаскиваем из стека: find(...) = find(...). Составные термы: имена главных функторов совпали, аргументы совпали – термы унифицируемы. В стек заносятся: Brand=mercedes, Color=blue, Sname= Sname, City=City, PhoneNum=PhoneNum, Bank= Bank. Результат унификации: успех Наиболее общий унификатор: Brand=mercedes, Color=blue. Для поиска было проведено 16 сопоставлений.	Прямой ход. Переход в тело правила.
1	Сопоставляем: owns(Sname, mercedes, blue, _, _), с термами аналогично терму find (см. Таблица 2), пока не будет найден терм удовлетворяющий условиям унификации.	Прямой ход, переход к следующему предложению. (по аналогии, до конца процедуры record)
2	Сопоставляем: owns(Sname, mercedes, blue, _, _), owns(zhorin, mercedes, blue, 18000) Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: Sname=zhorin. Для поиска было проведено 4 сопоставлений	Прямой ход, переход к следующей подцели правила.
3	Сопоставляем: record(zhorin, PhoneNum, address(City, _, _, _)) record(zhorin, “89068880045”, address(ryazan, lubanka, 36, 78)). Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: PhoneNum=“89068880045”, City=ryazan. Для поиска было проведено 11 сопоставлений.	Прямой ход, переход к следующей подцели правила.
4	Сопоставляем: hasBankDeposit(zhorin, Bank, _, _). с термами аналогично терму find (см. Таблица 2), пока не будет найден терм удовлетворяющий условиям унификации.	Прямой ход, переход к следующему предложению. (по аналогии, до конца процедур record, owns)

5	Сопоставляем: hasBankDeposit(zhorin, Bank, _, _). hasBankDeposit(zhorin, otkritie, "123456000", 45678). Составные термы, унифицируемые. Результат унификации: успех Наиболее общий унификатор: Bank=otkritie. Для поиска было проведено 14 сопоставлений	Истинное тело найдено. Sname=zhorin, PhoneNum="89068880045", City=ryazan, Bank=otkritie. Откат, переход к следующему предложению.
6	Сопоставляем: hasBankDeposit(zhorin, Bank, _, _) с термами в процедуре hasBankDeposit. Все термы унифицируемы, но унификация завершается неудачей. Было проведено 2 сопоставления	Унификация завершилась неудачей. Откат
7	Сопоставляем: record(zhorin, PhoneNum, address(City, _, _)) с термами в процедуре record. Все термы процедуры унифицируемы, но унификация завершается неудачей. Было проведено 5 сопоставлений	Унификация завершилась неудачей. Откат
8	Сопоставляем: owns(Sname, mercedes, blue, _, _) с термами в процедуре record. Все термы процедуры унифицируемы, но унификация завершается неудачей. Далее до конца БЗ термы не унифицируемы. Было проведено 11 сопоставлений	Унификация завершилась неудачей. Откат
9	Сопоставляем: find(mercedes, blue, Sname, City, PhoneNum, Bank), с термами до конца БЗ, унифицируемых термов не найдено.	Конец

Итого было проведено 48 сопоставлений.

При перестановках количество сопоставлений сохранилось.

Вывод: Пролог проходит по всем предложениям последовательно сравнивая текущий терм со всеми в разделе clauses. Поэтому в какой бы последовательности не шли предикаты, число проходов – объем работы останется тем же.

Задание 3

Работа алгоритма расписана в таблице 1 для получения первого набора данных.

Унификация терма find(mercedes, blue, Sname, City, PhoneNum, Bank)

и

find(Brand, Color, Sname, City, PhoneNum, Bank)

шаг унификации	результатирующая ячейка	рабочее поле	пункт Алг-ма	стек
0			1.	find (mercedes, blue,...) = find (Brand, Color,...)
1		find (mercedes, blue,...) = find (Brand, Color ,...)	е	Brand=mercedes Color=blue
2	Brand=mercedes	Brand=mercedes	г	Color=blue
3	Color=blue	Color=blue	г	Стек пуст.
4		Очистка поля	3.	

Флаг неудача = 0.

Наиболее общий унификатор: Brand=mercedes, Color=blue

Унификация терма owns(Sname, mercedes, blue _), и owns(zhorin, mercedes, blue, 18000).

шаг унификации	результатирующая ячейка	рабочее поле	пункт Алг-ма	стек
0			1.	owns(Sname, mercedes, blue _) = owns(zhorin, mercedes, blue, 18000).
1		owns(Sname, mercedes, blue _) = owns(zhorin, mercedes, blue, 18000).	е	Sname = zhorin mercedes=mercedes blue=blue
2	Sname = zhorin	Sname = zhorin	г	
3	Sname = zhorin	mercedes=mercedes	б	Color=blue
4	Sname = zhorin	blue=blue	б	Стек пуст.
5		Очистка поля	3.	

Флаг неудача = 0.

Наиболее общий унификатор: Sname = zhorin.

Примеры вопросов и ответов

```
find(toyota, white, Sname, City, PhoneNum, Bank).
```

```
Sname=yumaev, City=moscow, PhoneNum=89036609896, Bank=rocket_bank  
Sname=yumaev, City=vologda, PhoneNum=89036600505, Bank=rocket_bank  
Sname=petrov, City=moscow, PhoneNum=89068887745, Bank=rocket_bank  
3 Solutions
```

```
find(mercedes, blue, Sname, City, PhoneNum, Bank).
```

```
Sname=zhorin, City=ryazan, PhoneNum=89068887745, Bank=otkritie  
Sname=zhorin, City=ryazan, PhoneNum=89068887745, Bank=alpha_bank  
Sname=zhorin, City=ryazan, PhoneNum=89068880045, Bank=otkritie  
Sname=zhorin, City=ryazan, PhoneNum=89068880045, Bank=alpha_bank  
4 Solutions
```

```
find(kia, yellow, Sname, City, PhoneNum, Bank).
```

```
No solutions
```

Ответы на вопросы

1. **В какой части правила сформулировано знание? Это знание о чем, с формальной точки зрения?**
Знание сформулировано в заголовке, как в составном терме $f(t_1, \dots, t_n)$. Оно содержит знание о том, что между аргументами t_1, \dots, t_n существует отношение. «f» - название этого отношения.
2. **Что такое процедура?**
Процедурой называется совокупность правил, заголовки которых имеют одно и то же имя и одну и ту же аргументность (местность), т.е. это совокупность правил, описывающих одно определенное отношение.
3. **Сколько в БЗ текущего задания процедур?**
Конкретно в моей программе 6 процедур – phonebook, investors, automobile и 3 правила search_by.
4. **Что такое пример терма, это частный случай терма, пример? Как строится пример?**
Примером терма В называется такой терм А, если существует такая подстановка в терм А, которая в результате будет эквивалентна терму В. Например, есть терм $\text{cook}(X, Y)$ и есть $\text{cook}(\text{“Яблоки”}, \text{“Груши”})$, тогда при подстановке $\text{cook}(X, Y)$ и $\text{cook}(\text{“Яблоки”}, \text{“Груши”})$ получится $\text{cook}(\text{“Яблоки”}, \text{“Груши”})$, являющийся примером терма $\text{cook}(\text{“Яблоки”}, \text{“Груши”})$. Пример терма строится при сопоставлении (конкретизации) более общего терма с возможными подстановками.
5. **Что такое наиболее общий пример?**
Терм С называется общим примером термов А и В, если существуют такие подстановки θ_1 и θ_2 , что $C = A\theta_1$ и $C = B\theta_2$, т.е С является общим примером для В и А, причем, А и В являются примерами С, а С для них не является.
6. **Назначение и результат работы алгоритма унификации. Что значит двунаправленная передача параметров при работе алгоритма унификации, поясните на примере одного из случаев пункта 3.**
Алгоритм унификации - основной вычислительный шаг работы программы. Предназначен для «сопоставления термов», формализует процесс логического вывода. Завершается либо «успехом», либо «неудачей». Результат работы программы – наиболее общий унификатор.
Двунаправленная передача параметров – после унификации, переменные могут быть конкретизированы, в этом случае, конкретизированные переменные переходят на следующий шаг доказательства (в подцели, например).
7. **В каком случае запускается механизм отката?**
Механизм отката запускается, в случае, если во алгоритм унификации завершился тупиковой ситуацией, неудачей.
8. **Виды и назначение переменных в Prolog. Примеры из задания. Почему использованы те или другие переменные (примеры из задания)?**
Переменные предназначены для передачи значений «во времени и в пространстве». Они служат частью процесса сопоставления, и не являются «хранилищем» информации.
Переменные бывают:
 - Именованные – обозначается комбинацией символов латинского алфавита, цифр и символа подчеркивания, начинающейся с прописной буквы или символа подчеркивания. Уникальны в рамках одного предложения
 - Анонимные – обозначаются символом подчеркивания «_». Любая анонимная переменная уникальна.Во время вычисления, именованные переменные могут конкретизироваться (связываться с различными объектами), причем, она может быть ре-конкретизирована, путем «отката» вычислительного процесса и отмены ранее проведенной конкретизации для нахождения новых решений.
Анонимные переменные не могут быть связаны со значениями.

Пример: `search_by(yumaev, moscow, PhoneNum, Street, Bank)`. - используются именованные свободные переменные, для получения набора значений при вычислении цели.

Пример: `search_by(PhoneNum, Brand) :- search_by(_, PhoneNum, Brand, _)`. – анонимные переменные, используются для игнорирования некоторых значений программой.