



**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

О т ч е т

по лабораторной работе № 14

Дисциплина: «Функциональное и логическое программирование»

Выполнила: Овчинникова А. П.

Группа: ИУ7-65Б

Преподаватель: Толпинская Н. Б.

Строганов Ю. В.

Москва, 2020

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?
3. 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.

Теоретическая часть

1. В какой части правила сформулировано знание? Это знание о чем, с формальной точки зрения?

Правая часть, или тело правила задает условия, при котором верно отношение, записанное в левой части, или заголовке правила. Т. е. знание сформулировано в теле правила. Это знание об отношениях между объектами.

2. Что такое процедура?

Процедурой называется совокупность правил, заголовки которых имеют одно и то же имя и одну и ту же арность (местность), т.е. это совокупность правил, описывающих одно определенное отношение. Предложения с одним и тем же предикатом в заголовке должны идти один за другим.

3. Сколько в БЗ текущего задания процедур?

Три.

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

Терм В называется примером терма А, если существует такая подстановка Θ , что $B=A\Theta$.

5. Что такое наиболее общий пример?

S называется наиболее общим примером T1 и T2, если S такой их общий пример, который является более общим по отношению к любому другому их примеру.

6. Назначение и результат работы алгоритма унификации. Что значит двунаправленная передача параметров при работе алгоритма унификации, поясните на примере одного из случаев пункта 3.

Работа алгоритма унификации заключается в попарном сопоставлении термов и попытке построить для них общий пример, для чего и нужна подстановка. Назначение алгоритма унификации – подбор подходящего правила.

Для того чтобы понять, что заголовок подходит, система строит унификатор – подстановку (побочный эффект работы алгоритма унификации). В результате применения подстановки некоторые переменные конкретизируются значениями, которые (значения) могут и будут далее использованы при доказательстве истинности тела выбранного правила (что тоже выполняется с использованием алгоритма унификации). Т. е. значения переменных переходят на следующий шаг доказательства. Таким образом, с помощью алгоритма унификации происходит двунаправленная передача параметров процедурам.

Унификация – операция, которая позволяет формализовать процесс логического вывода (наряду с правилом резолюции).

7. В каком случае запускается механизм отката?

Унификация - попытка "увидеть одинаковость", может завершаться успехом или тупиковой ситуацией (неудачей), тогда включается механизм отката к предыдущему шагу.

8. Виды и назначение переменных в Prolog. Примеры из задания. Почему использованы те или другие переменные (примеры из задания)?

Факты, правила, и вопросы могут содержать переменные. Имя переменной может состоять из букв латинского алфавита, цифр, знаков подчеркивания и должно начинаться с прописной буквы или знака подчеркивания. Переменные в факты и правила входят только с квантором всеобщности. А в вопросы переменные входят только с квантором существования. Переменная в Прологе обозначает объект, а не некоторую область памяти. Переменные предназначены для передачи значений «во времени и в пространстве». В логическом программировании все переменные рассматриваются как безтиповые.

Переменные могут быть свободными или связанными. Свободная (неконкретизированная) переменная – это переменная, которая еще не получила значения. Переменная, которая получила какое-то значение и оказалась связанной с определенным объектом, называется связанной. Если переменная была конкретизирована каким-то значением и ей сопоставлен некоторый объект, то эта переменная уже не может быть изменена. В логическом программировании поддерживается механизм деструктивной конкретизации переменной. Т. е. используется идея реконкретизации переменной путем «отката» вычислительного процесса и отказа от выполненной ранее конкретизации. Это реализовано для возможности поиска нового значения для именованной переменной.

Областью действия переменной в Прологе является одно предложение. Исключением из правила определения области действия является анонимная переменная, которая обозначается символом подчеркивания «_». Анонимная переменная применяется в случае, когда значение переменной не важно. Каждая анонимная переменная – это отдельный объект.

Именованные переменные уникальны в рамках предложения, а анонимная переменная – любая уникальна. В разных предложениях может использоваться одно имя переменной для обозначения разных объектов.

Практическая часть

В листинге 1 приведен текст программы.

Листинг 1. Текст программы.

```
include "lab14_filp.inc"
```

```
domains
```

```
address = address(string, string, integer, integer). % city, street, house, flat
```

predicates

abonent(symbol, string, address). % surname, phone, address

*car_owner(symbol, string, string, string). % owner surname, model, color,
cost*

investor(symbol, string, string, string). % surname, bank, deposit, amount

*find_by_model_and_color(string, string, symbol, string, string, string). %
model, color, surname, city, phone, bank*

clauses

*abonent(ivanov, "89998693211", address("Moscow",
"Tamanskaja", 4, 24)).*

*abonent(petrov, "89918553214", address("Moscow",
"Babaevskaja", 1, 5)).*

*abonent(sidorov, "89915593215", address("Moscow",
"Korennaja", 31, 51)).*

*abonent(ovchinnikova, "89998695230", address("Orekhovo-
Zuevo", "Volodarskogo", 25, 24)).*

*abonent(zorina, "89998695233", address("Biisk", "Lenina", 21,
11)).*

*abonent(vasilev, "89998695244", address("Saint-Petersburg",
"Malaja Sadovaja", 1, 2)).*

*abonent(stanovaja, "89998690000", address("Saint-Petersburg",
"Paradnaja", 10, 20)).*

*abonent(korolev, "89998690001", address("Saint-Petersburg",
"Nevsky prospekt", 12, 21)).*

investor(ivanov, "Sberbank", "111111111111111111", "50000").

investor(ivanov, "Alpha Bank", "111111111111111112", "40000").

investor(petrov, "Sberbank", "211111111111111111", "10000").
investor(sidorov, "Tochka", "311111111111111111", "60000").
investor(sidorov, "Tinkoff", "311111111111111112", "5000").
investor(ovchinnikova, "Sberbank", "411111111111111111", "5000").
investor(zorina, "Sberbank", "411111111111111111", "6000").
investor(korolev, "VTB", "511111111111111111", "60000").

car_owner(ivanov, "Ford", "Dark blue", "1000000").
car_owner(ivanov, "Mazda", "Red", "500000").
car_owner(petrov, "Lada", "Light gray", "100000").
car_owner(vasilev, "Peugeot", "White", "700000").
car_owner(korolev, "Land Rover", "Green", "1020000").
car_owner(zorina, "Land Rover", "Green", "1020000").

find_by_model_and_color(MODEL, COLOR, SURNAME, CITY, PHONE, BANK) :-

car_owner(SURNAME, MODEL, COLOR, _), abonent(SURNAME,
PHONE, address(CITY, _, _, _)),
investor(SURNAME, BANK, _, _).

goal

%find_by_model_and_color("Ford", "Dark blue", SURNAME, CITY,
PHONE, BANK). % one owner

%find_by_model_and_color("BMW", "Dark blue", SURNAME, CITY,
PHONE, BANK). % no owners

%find_by_model_and_color("Peugeot", "White", SURNAME, CITY, PHONE,
BANK). % one owner without bank

find_by_model_and_color("Land Rover", "Green", SURNAME, CITY,
PHONE, BANK). % two owners

Примеры возможных вариантов вопросов и ответов

1. Вопрос: *find_by_model_and_color("Ford", "Dark blue", SURNAME, CITY, PHONE, BANK). % one owner*

Ответ:

SURNAME=ivanov, CITY=Moscow, PHONE=89998693211, BANK=Sberbank

SURNAME=ivanov, CITY=Moscow, PHONE=89998693211, BANK=Alpha Bank

2 Solutions

Рассмотрим порядок поиска системой ответа на этот вопрос.

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат
0	<i>find_by_model_and_color("Ford", "Dark blue", SURNAME, CITY, PHONE, BANK).= find_by_model_and_color(MODEL, COLOR, SURNAME, CITY, PHONE, BANK)</i> Неконкретизированные переменные <i>SURNAME, CITY, PHONE, BANK</i> в вопросе и правиле становятся сцепленными, а переменные <i>MODEL</i> и <i>COLOR</i> конкретизируются значениями <i>"Ford"</i> и <i>"Dark blue"</i> соответственно.	Прямой ход. Доказательство первой цели тела правила.
1	<i>car_owner(SURNAME, "Ford", "Dark blue", _) = car_owner(ivanov, "Ford", "Dark blue", "1000000").</i> Унификация успешна. Переменной <i>SURNAME</i> сопоставляется значение <i>ivanov</i> . Подстановка: $\Theta = \{ SURNAME = ivanov, "Ford", "Dark blue", "1000000" \}$ (Других ответов нет).	Прямой ход. Доказательство второй цели тела правила.
2	<i>abonent(ivanov, PHONE, address(CITY, _, _)) = abonent(ivanov, "89998693211", address("Moscow", "Tamanskaja", 4, 24)).</i> Унификация успешна. Переменным <i>PHONE</i> и <i>CITY</i> сопоставляются значения <i>"89998693211"</i> и <i>"Moscow"</i> соответственно. Подстановка: $\Theta = \{ ivanov, PHONE = "89998693211", address(CITY = "Moscow", "Tamanskaja", 4, 24) \}$ (Других ответов нет).	Прямой ход. Доказательство третьей цели тела правила.
3	<i>investor(ivanov, BANK, _, _).= investor(ivanov, "Sberbank", "111111111111111111", "50000").</i> Унификация успешна. Переменной <i>BANK</i> сопоставляется значение <i>"Sberbank"</i> . Подстановка: $\Theta = \{ ivanov, BANK="Sberbank", "111111111111111111", "50000" \}$	Прямой ход. Поиск других решений (с шага 1).

4	<code>investor(ivanov, BANK, _, _).= investor(ivanov, "Alpha Bank", "11111111111111111112", "40000").</code> Унификация успешна. Переменной BANK сопоставляется значение "Alpha Bank". Подстановка: $\Theta = \{ \text{ivanov, BANK="Alpha Bank", "11111111111111111112", "40000"} \}$ (Других решений нет).	Прямой ход. Все цели доказаны. Правило доказано. Одно решение найдено. Продолжается поиск решений.
---	---	--

2. Вопрос: *find_by_model_and_color("BMW", "Dark blue", SURNAME, CITY, PHONE, BANK). % no owners*

Ответ:

No Solution

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат
0	<code>find_by_model_and_color("BMW", "Dark blue", SURNAME, CITY, PHONE, BANK).= find_by_model_and_color(MODEL, COLOR, SURNAME, CITY, PHONE, BANK)</code> Неконкретизированные переменные SURNAME, CITY, PHONE, BANK в вопросе и правиле становятся сцепленными, а переменные MODEL и COLOR конкретизируются значениями "BMW" и "Dark blue" соответственно.	Прямой ход. Доказательство первой цели тела правила.
1	<code>car_owner(SURNAME, MODEL, COLOR, _)= car_owner(ivanov, "Ford", "Dark blue", "1000000").</code> Унификация не пройдена.	Откат. Переход к следующему по порядку применимому предложению.
2-5
6	<code>car_owner(SURNAME, MODEL, COLOR, _)= car_owner(korolev, "Land Rover", "Green", "1020000").</code> Унификация не пройдена.	Применимых предложений больше нет. Неуспех доказательства. Выход из процедуры с ответом no.

3. Вопрос: *find_by_model_and_color("Land Rover", "Green", SURNAME, CITY, PHONE, BANK).*

Ответ:

SURNAME=korolev, CITY=Saint-Petersburg, PHONE=89998690001, BANK=VTB

SURNAME=zorina, CITY=Biisk, PHONE=89998695233, BANK=Sberbank

2 Solutions

№ шага	Сравниваемые термы; результат; подстановка, если есть	Дальнейшие действия: прямой ход или откат
0	$find_by_model_and_color("Land\ Rover", "Green", SURNAME, CITY, PHONE, BANK). = find_by_model_and_color(MODEL, COLOR, SURNAME, CITY, PHONE, BANK)$ Неконкретизированные переменные <i>SURNAME, CITY, PHONE, BANK</i> в вопросе и правиле становятся сцепленными, а переменные <i>MODEL</i> и <i>COLOR</i> конкретизируются значениями <i>"Land Rover"</i> и <i>"Green"</i> соответственно.	Прямой ход. Доказательство первой цели тела правила.
1	$car_owner(ivanov, "Ford", "Dark\ blue", "1000000"). = car_owner(SURNAME, "Land\ Rover", "Green", _)$ Унификация не пройдена.	Откат. Переход к следующему по порядку применимому предложению.
2-4
5	$car_owner(zorina, "Land\ Rover", "Green", "1020000"). = car_owner(SURNAME, "Land\ Rover", "Green", _)$ Унификация успешна. Переменной <i>SURNAME</i> сопоставляется значение <i>zornia</i> . Подстановка: $\Theta = \{ SURNAME = zornia, "Land\ Rover", "Green", "1020000" \}$	Прямой ход. Переход к следующему по порядку применимому предложению.
4	$car_owner(korolev, "Land\ Rover", "Green", "1020000"). = car_owner(SURNAME, "Land\ Rover", "Green", _)$ Унификация успешна. Переменной <i>SURNAME</i> сопоставляется значение <i>korolev</i> . Подстановка: $\Theta = \{ SURNAME = korolev, "Land\ Rover", "Green", "1020000" \}$	Прямой ход. Применимых приложений больше нет. Доказательство второй цели тела правила.
5	$abonent(zorina, PHONE, address(CITY, _ _ _)) = abonent(ivanov, "89998693211", address("Moscow", "Tamanskaja", 4, 24)).$ Унификация не пройдена.	Откат. Переход к следующему по порядку применимому предложению.
6-8
9	$abonent(zorina, PHONE, address(CITY, _ _ _)) = abonent(zorina, "89998695233", address("Biisk", "Lenina", 21, 11)).$ Унификация успешна. Переменным <i>PHONE, CITY</i> сопоставляются значения <i>"89998695233"</i> и <i>"Biisk"</i> соответственно. Подстановка: $\Theta = \{ zorina, PHONE = "89998695233", address(CITY = "Biisk", "Lenina", 21, 11) \}$ (Других решений нет).	Прямой ход. Доказательство третьей цели тела правила.
10	$investor(ivanov, "Sberbank", "111111111111111111", "50000"). = investor(zorina, BANK, _ _).$ Унификация не пройдена.	Откат. Переход к следующему по порядку применимому предложению.

11-15
16	$investor(zorina, "Sberbank", "41111111111111111111", "6000").=investor(zorina, BANK, _ _).$ Унификация успешна. Переменной <i>BANK</i> сопоставляется значения “ <i>Sberbank</i> ”. Подстановка: $\Theta = \{ zorina, BANK="Sberbank", "41111111111111111111", "6000" \}$ (Других решений нет).	Прямой ход. Все цели доказаны. Правило доказано. Одно решение найдено. Аналогично с шага 10 для SURNAME=korolev.
...
17	$investor(korolev, "VTB", "51111111111111111111", "60000").=investor(korolev, BANK, _ _).$ Унификация успешна. Переменной <i>BANK</i> сопоставляется значения “ <i>VTB</i> ”. Подстановка: $\Theta = \{ korolev, BANK="VTB", "51111111111111111111", "60000" \}$ (Других решений нет).	Прямой ход. Все цели доказаны. Правило доказано. Второе решение найдено. Продолжается поиск решений (с шага 1).

4. Вопрос: *find_by_model_and_color("Peugeot", "White", SURNAME, CITY, PHONE, BANK). % one owner without bank*

Ответ:

No Solution

Примеры работы системы при разных порядках следования процедур в БЗ

Далее будут рассмотрены ответы системы на вопрос *find_by_model_and_color("Land Rover", "Green", SURNAME, CITY, PHONE, BANK).* при разном порядке следования процедур в БЗ.

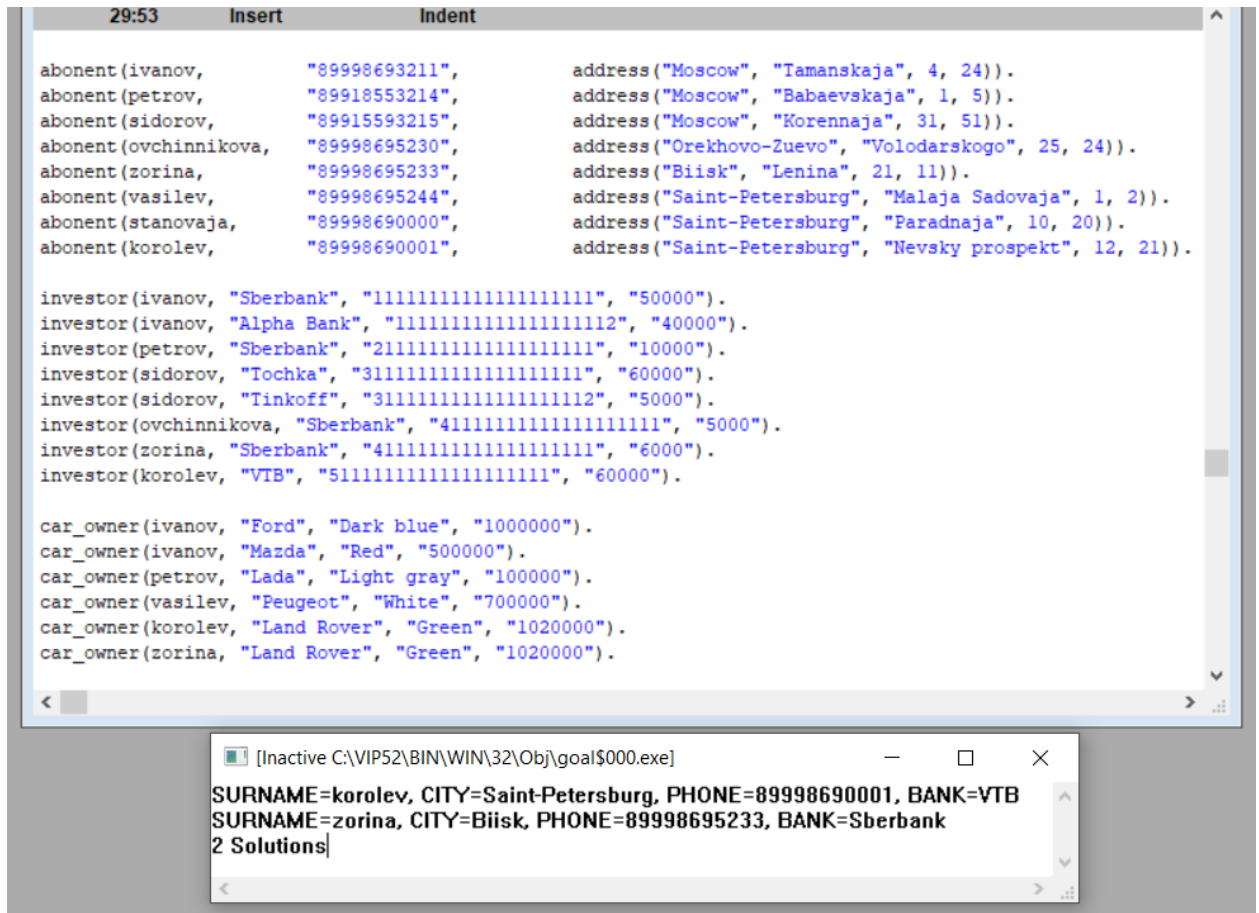
На рисунках 1- представлен порядок следования процедур в БЗ и знаний в них и ответ на вопрос.

Как видно из рисунков 1-3, при изменении только порядка процедур ответ не меняется. Если же мы изменим порядок знаний в процедуре *car_owner*, то ответ изменится (изменится не содержание ответа, а порядок, в котором система находит подходящие знания), как на рисунке 4.

Для доказательства каждой подцели пролог-система ищет применимое предложение путем последовательного, от начала к концу программы, просмотра всех ее предложений. Предложение применимо, либо если это факт вида *A.*, и он сопоставим с доказываемой подцелью, либо же это правило вида

А: В1, ..., Вm. и его заголовок-предикат А сопоставим с доказываемой подцелью.

Таким образом, объем работы при изменении порядка следования процедур и знаний в них не изменяется.



The screenshot shows a Prolog interpreter window with the following code:

```
29:53      Insert      Indent

abonent(ivanov,      "89998693211",      address("Moscow", "Tamanskaja", 4, 24)).
abonent(ivanov,      "89918553214",      address("Moscow", "Babaevskaja", 1, 5)).
abonent(sidorov,     "89915593215",      address("Moscow", "Korennaja", 31, 51)).
abonent(ovchinnikova, "89998695230",      address("Orekhovo-Zuevo", "Volodarskogo", 25, 24)).
abonent(zorina,      "89998695233",      address("Biisk", "Lenina", 21, 11)).
abonent(vasilev,     "89998695244",      address("Saint-Petersburg", "Malaja Sadovaja", 1, 2)).
abonent(stanovaja,   "89998690000",      address("Saint-Petersburg", "Paradnaja", 10, 20)).
abonent(korolev,     "89998690001",      address("Saint-Petersburg", "Nevsky prospekt", 12, 21)).

investor(ivanov, "Sberbank", "1111111111111111", "50000").
investor(ivanov, "Alpha Bank", "11111111111111112", "40000").
investor(ivanov, "Sberbank", "2111111111111111", "10000").
investor(sidorov, "Tochka", "3111111111111111", "60000").
investor(sidorov, "Tinkoff", "31111111111111112", "5000").
investor(ovchinnikova, "Sberbank", "4111111111111111", "5000").
investor(zorina, "Sberbank", "4111111111111111", "6000").
investor(korolev, "VTB", "5111111111111111", "60000").

car_owner(ivanov, "Ford", "Dark blue", "1000000").
car_owner(ivanov, "Mazda", "Red", "500000").
car_owner(ivanov, "Lada", "Light gray", "100000").
car_owner(ivanov, "Peugeot", "White", "700000").
car_owner(korolev, "Land Rover", "Green", "1020000").
car_owner(zorina, "Land Rover", "Green", "1020000").
```

Below the code, a query window is open with the following text:

```
[Inactive C:\VIP52\BIN\WIN\32\Obj\goal$000.exe]
SURNAME=korolev, CITY=Saint-Petersburg, PHONE=89998690001, BANK=VTB
SURNAME=zorina, CITY=Biisk, PHONE=89998695233, BANK=Sberbank
2 Solutions|
```

Рисунок 1. Работа системы при разном порядке следования процедур (ч. 1).

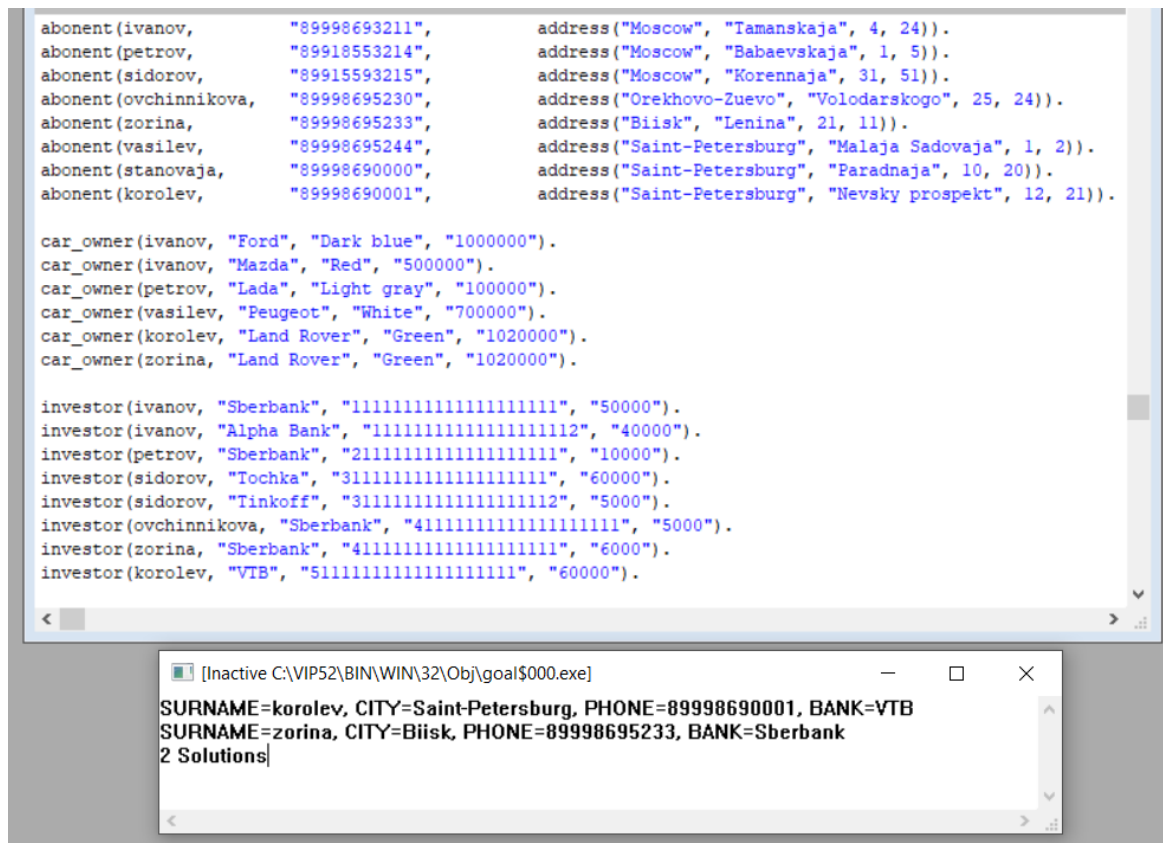


Рисунок 2. Работа системы при разном порядке следования процедур (ч. 2).

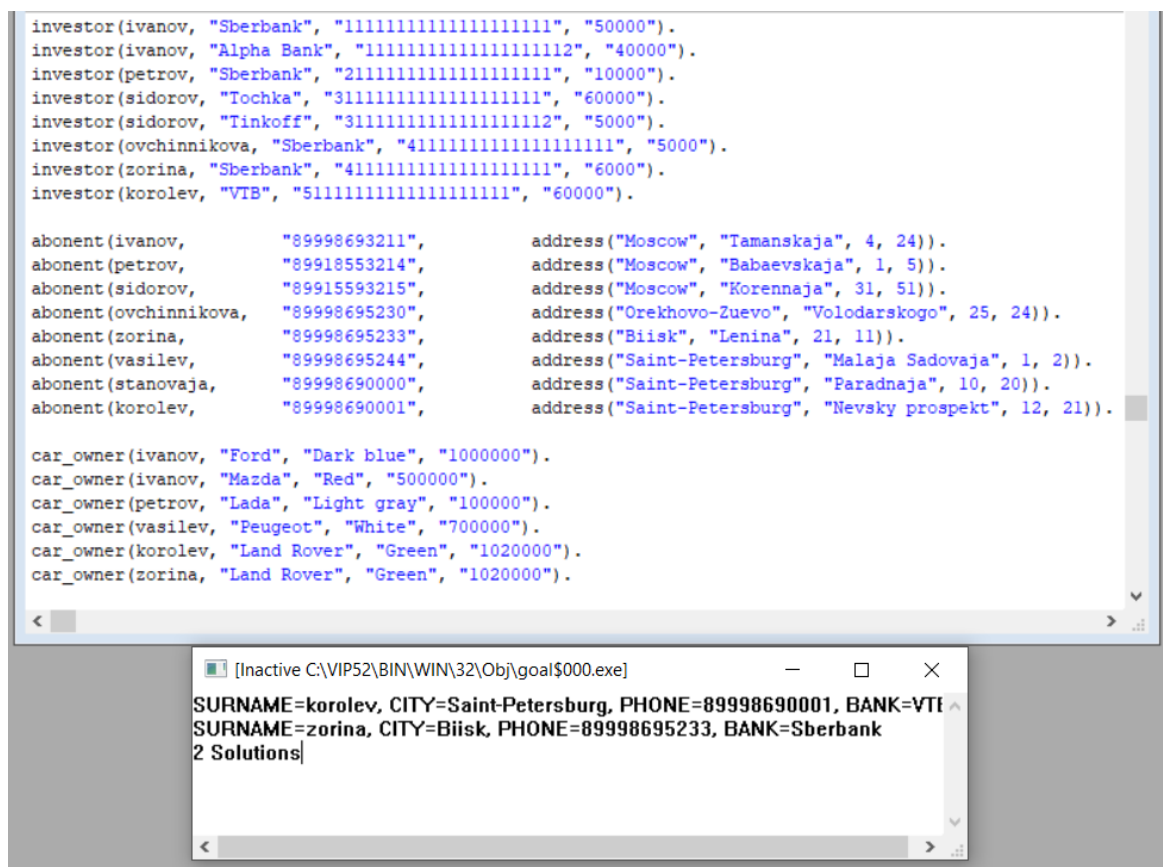


Рисунок 3. Работа системы при разном порядке следования процедур (ч. 3).

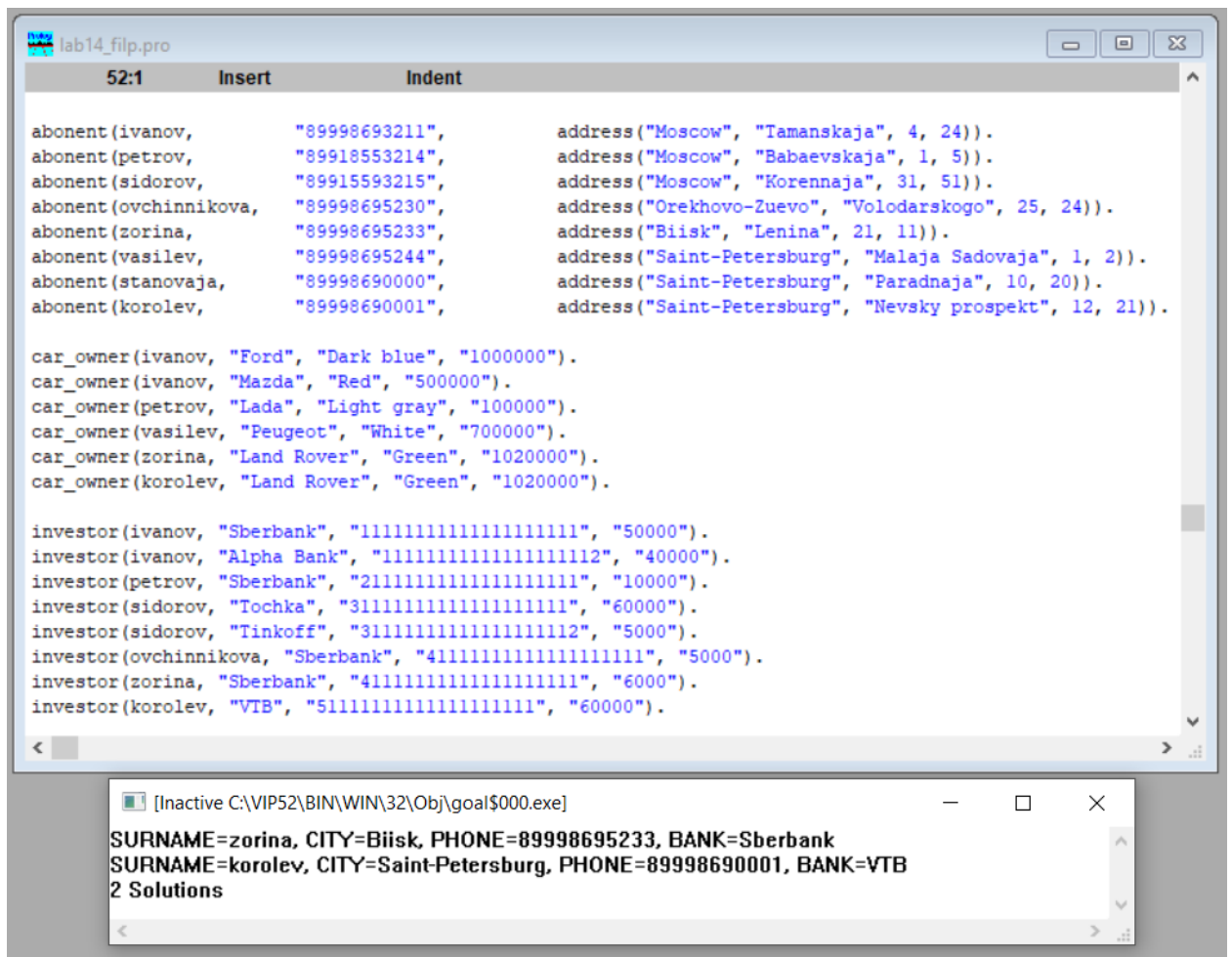


Рисунок 4. Работа системы при разном порядке следования процедур (ч.4).

Порядок работы алгоритма унификации

Пример работы алгоритма унификации двух термов:
 $find_by_model_and_color("Land\ Rover", "Green", SURNAME, CITY, PHONE, BANK).$ и $find_by_model_and_color(MODEL, COLOR, SURNAME, CITY, PHONE, BANK) :- car_owner(SURNAME, MODEL, COLOR, _), abonent(SURNAME, PHONE, address(CITY, _, _)), investor(SURNAME, BANK, _, _).$ Обозначим первый терм как S, а второй как T.

Шаг униф.	Результирующая ячейка	рабочее поле	пункт алгоритма	стек
0			1	$car_owner(SURNAME, "Land\ Rover", "Green", _) = car_owner(ivanov, "Ford", "Dark\ blue", "1000000").$

1		car_owner(SURNAME,"Land Rover","Green",_)=car_owner(ivanov,"Ford", "Dark blue", "1000000").	e	SURNAME=Ivanov "Land Rover"="Ford" "Green"="Dark blue" _="1000000"
2	SURNAME=Ivanov	SURNAME=Ivanov	г	"Land Rover"="Ford" "Green"="Dark blue" _="1000000"
3	SURNAME=Ivanov	"Land Rover"="Ford"	г	"Green"="Dark blue" _="1000000"
4			1	car_owner(SURNAME,"Land Rover","Green",_)=car_owner(ivanov,"Mazda", "Red", "500000").

15			1	car_owner(SURNAME,"Land Rover","Green",_)=car_owner(zornia, "Land Rover", "Green", "1020000").
16		car_owner(SURNAME,"Land Rover","Green",_)=car_owner(zornia, "Land Rover", "Green", "1020000").	e	SURNAME=zornia "Land Rover"="Land Rover" "Green"="Green" _="1020000"
17	SURNAME=zornia	SURNAME=zornia	г	"Land Rover"="Land Rover" "Green"="Green" _="1020000"
18	SURNAME=zornia "Land Rover"="Land Rover"	"Land Rover"="Land Rover"	г	"Green"="Green" _="1020000"
19	SURNAME=zornia "Land Rover"="Land Rover" "Green"="Green"	"Green"="Green"	г	_="1020000"
20	SURNAME=zornia "Land Rover"="Land Rover" "Green"="Green"	_="1020000"	г	
Вывод:	Подстановка	Стек пуст, подцель доказана. В рез. ячейке подстановка. Аналогичный поиск других решений не дает других решений. Переход к доказательству следующей цели.		
21			1	abonent(zornia,PHONE,address(CITY,_,_)).=abonent(ivanov,"89998693211",address("Moscow", "Tamanskaja", 4, 24)).
21		abonent(SURNAME,PHONE,address(CITY,_,_)).=abonent(ivanov,"89998693211",address("Moscow", "Tamanskaja", 4, 24)).	e	zornia=ivanov PHONE="89998693211" CITY="Moscow" _="Tamanskaja" _=4 _=24
22		zornia=ivanov	г	PHONE="89998693211" CITY="Moscow" _="Tamanskaja" _=4 _=24

23			1	abonent(zornia,PHONE,address(CITY,_,_)).=abonent(petrov,"89918553214",address("Moscow", "Babaevskaja", 1, 5)).
24-30
31			1	abonent(zornia,PHONE,address(CITY,_,_)).=abonent(zorina,"89998695233",address("Biisk", "Lenina", 21, 11)).
32		abonent(zornia,PHONE,address(CITY,_,_)).=abonent(zorina,"89998695233",address("Biisk", "Lenina", 21, 11)).	e	zornia= zornia PHONE="89998695233" CITY="Biisk" _= "Lenina" _=21 _=11
33	zornia= zornia	zornia= zornia	г	PHONE="89998695233" CITY="Biisk" _= "Lenina" _=21 _=11
34	zornia= zornia PHONE="89998695233"	PHONE="89998695233"	Г	CITY="Biisk" _= "Lenina" _=21 _=11
35	CITY="Biisk" zornia= zornia PHONE="89998695233"	CITY=" Biisk "	г	_= "Lenina" _=21 _=11
36	CITY="Biisk" zornia= zornia PHONE="89998695233"	_= "Lenina"		_=21 _=11
37	CITY="Biisk" zornia= zornia PHONE="89998695233"	_=21		_=11
38	CITY="Biisk" zornia= zornia PHONE="89998695233"	_=11		
Вывод:	Подстановка	Стек пуст, подцель доказана. В рез. ячейке подстановка. Аналогичный поиск других решений не дает других решений. Переход к доказательству следующей цели.		
39			1	investor(zornia, BANK, _, _).=investor(ivanov, "Sberbank", "1111111111111111111", "50000").
40		investor(zornia,BANK,_,_).=investor(ivanov,"Sberbank","1111111111111111111", "50000").	e	zornia= ivanov BANK=" Sberbank" _= "1111111111111111111" _= "50000"
41		zornia= ivanov	г	BANK=" Sberbank" _= "1111111111111111111" _= "50000"
42			1	investor(zornia, BANK, _, _).=investor(ivanov, "Alpha Bank", "1111111111111111112", "40000").
42-55
56			1	investor(zornia, BANK, _, _).=investor(zorina, "Sberbank",

				"411111111111111111", "6000").
57		investor(zornia,BANK,_,_).=investor(zor ina,"Sberbank","411111111111111111 1", "6000").	e	zornia= zornia BANK="Sberbank" _="411111111111111111" _="6000"
58	zornia= zornia	zornia= zornia	г	BANK="Sberbank" _="411111111111111111" _="6000"
59	BANK="Sber bank" zornia= zornia	BANK="Sberbank"	г	_="411111111111111111" _="6000"
60	BANK="Sber bank" zornia= zornia	_="411111111111111111"	г	_="6000"
61	BANK="Sber bank" zornia= zornia	_="6000"	г	
Выво д:	Подстановка	Стек пуст, подцель доказана. В рез. ячейке подстановка. Аналогичный поиск других решений не дает других решений. Цель доказана. Поиск других доказательств для правила, аналогично начиная с шага 0. (Будет на)		

Если поменять местами знания в процедурах, то объем работы системы не изменится, потому что ей необходимо найти все ответы. Поэтому любой другой пример поиска информации о двух владельцах машины будет работать аналогично.