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 № 13
Работа программы на Prolog

Студент Лучина Е.Д

Группа ИУ7-61Б

Преподаватель Толпинская Н.Б.

Москва.

апрель 2020 г.

Цель работы – получить навыки построения модели предметной области, разработки и оформления программы на Prolog, изучить принципы, логику формирования программы и отдельные шаги выполнения программы на Prolog.

Задачи работы: приобрести навыки декларативного описания предметной области с использованием фактов и правил. Изучить способы использования термов, переменных, фактов и правил в программе на Prolog, принципы и правила сопоставления и отождествления, порядок унификации.

Задание

Составить программу, т.е. модель предметной области – базу знаний, объединив в ней информацию – знания:

- «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).

1. Используя правила, обеспечить возможность поиска:
   a. По № телефона найти: Фамилию, Марку автомобиля, Стоимость автомобиля (может быть несколько)
   b. Используя сформированное в пункте а) правило, по № телефона найти: только Марку автомобиля (автомобилей может быть несколько)
2. Используя простой, не составной вопрос: по Фамилии (уникальна в городе, но в разных городах есть однофамильцы) и Городу проживания найти: Улицу проживания, Банки, в которых есть вклады и №телефона.

```
domains
      city, street = symbol
      house, flat = integer
      adrdess_type = address(city, street, house, flat)

      lastname = symbol
      phonenumber = string

      marka, color = symbol
      price = integer

      bank = symbol
      account = integer
      sum = real

predicates
      phonebook(lastname, phonenumber, adrdess_type)
```

```
      carinfo(lastname, marka, color, price)
      bankinfo(lastname, bank, account, sum)

      get_name_carinfo_by_phone(phonenumber,lastname, marka, price)
      get_street_bank_phone_by_name_and_city(lastname, city, street, bank,
phonenumber)

clauses
      phonebook(ivanov, "89258200123", address(moscow, tverskaya, 6, 12)).
      phonebook(ivanov, "89128600423", address(tver, moskovskaya, 3, 11)).
      phonebook(sidorov, "89258234128", address(tver, lenina, 13, 7)).
      phonebook(petrov, "89158200124", address(piter, nevskiy, 3, 34)).
      phonebook(ivanova, "89658200113", address(moscow, tverskaya, 6, 12)).
      phonebook(petrova, "89258200654", address(moscow, nikolskaya, 8, 42)).

      carinfo(ivanov, mazda, red, 234000).
      carinfo(ivanov, mazda, blue, 204000).
      carinfo(ivanov, opel, white, 252300).
      carinfo(sidorov, peugeot, white, 183000).
      carinfo(petrov, honda, yellow, 212500).
      carinfo(petrova, opel, white, 242800).
      carinfo(petrova, opel, white, 250000).

      bankinfo(ivanov, sberbank, 12345, 45.54).
      bankinfo(ivanova, alpha, 13545, 40.54).
      bankinfo(petrov, sberbank, 12445, 13.5).
      bankinfo(sidorov, tinkoff, 12345, 45.05).
      bankinfo(petrova, sberbank, 12345, 45.54).

      get_name_carinfo_by_phone(PhoneNumber, LastName, Marka, Price) :-
            phonebook(LastName, PhoneNumber, _),
            carinfo(LastName, Marka, _, Price).

      get_street_bank_phone_by_name_and_city(LastName, City, Street, Bank,
PhoneNumber) :-
            phonebook(LastName, PhoneNumber, address(City, Street, _, _)),
            bankinfo(LastName, Bank, _, _ ).
goal
    phonebook(ivanov, "89258200123", address(moscow, tverskaya, 6, 12)).
```

Примеры работы программы при запуске разных goal (целей)

```
phonebook(ivanov, "89258200123", address(moscow, tverskaya, 6, 12)).
```

```
yes
```

```
phonebook(ivanov, "89258000123", address(moscow, tverskaya, 6, 12)).
```

```
no
```

```
phonebook(ivanov, PhoneNumber, address(moscow, tverskaya, 6, 12)).
```

```
PhoneNumber=89258200123
1 Solution
```

```
phonebook(LastName, "89258200123", address(moscow, tverskaya, 6, 12)).
```

```
LastName=ivanov
1 Solution
```

```
phonebook(ivanov, "89258200123", address(City, _, _, _)).
```

```
City=moscow
1 Solution
```

```
phonebook(ivanov, "89258200123", Address).
```

```
Address=address("moscow","tverskaya",6,12)
1 Solution
```

```
phonebook(ivanov, _, Address).
```

```
Address=address("moscow","tverskaya",6,12)
Address=address("tver","moskovskaya",3,11)
2 Solutions
```

```
get_name_carinfo_by_phone("89258200123", LastName, Marka, Price).
```

```
LastName=ivanov, Marka=mazda, Price=234000
LastName=ivanov, Marka=mazda, Price=204000
LastName=ivanov, Marka=opel, Price=252300
3 Solutions
```

```
get_name_carinfo_by_phone("89258234128", LastName, Marka, Price).
```

```
LastName=sidorov, Marka=peugeot, Price=183000
1 Solution
```

```
get_name_carinfo_by_phone("89658200113", LastName, Marka, Price).
/* у п е т р о в о й  н е т  м а ш и н ы */
```

```
No Solution
```

```
get_name_carinfo_by_phone("89258200654", _, Marka, _).
```

```
Marka=opel
Marka=opel
2 Solutions
```

```
get_name_carinfo_by_phone("89158200124", _, Marka, _).
```

```
Marka=honda
1 Solution
```

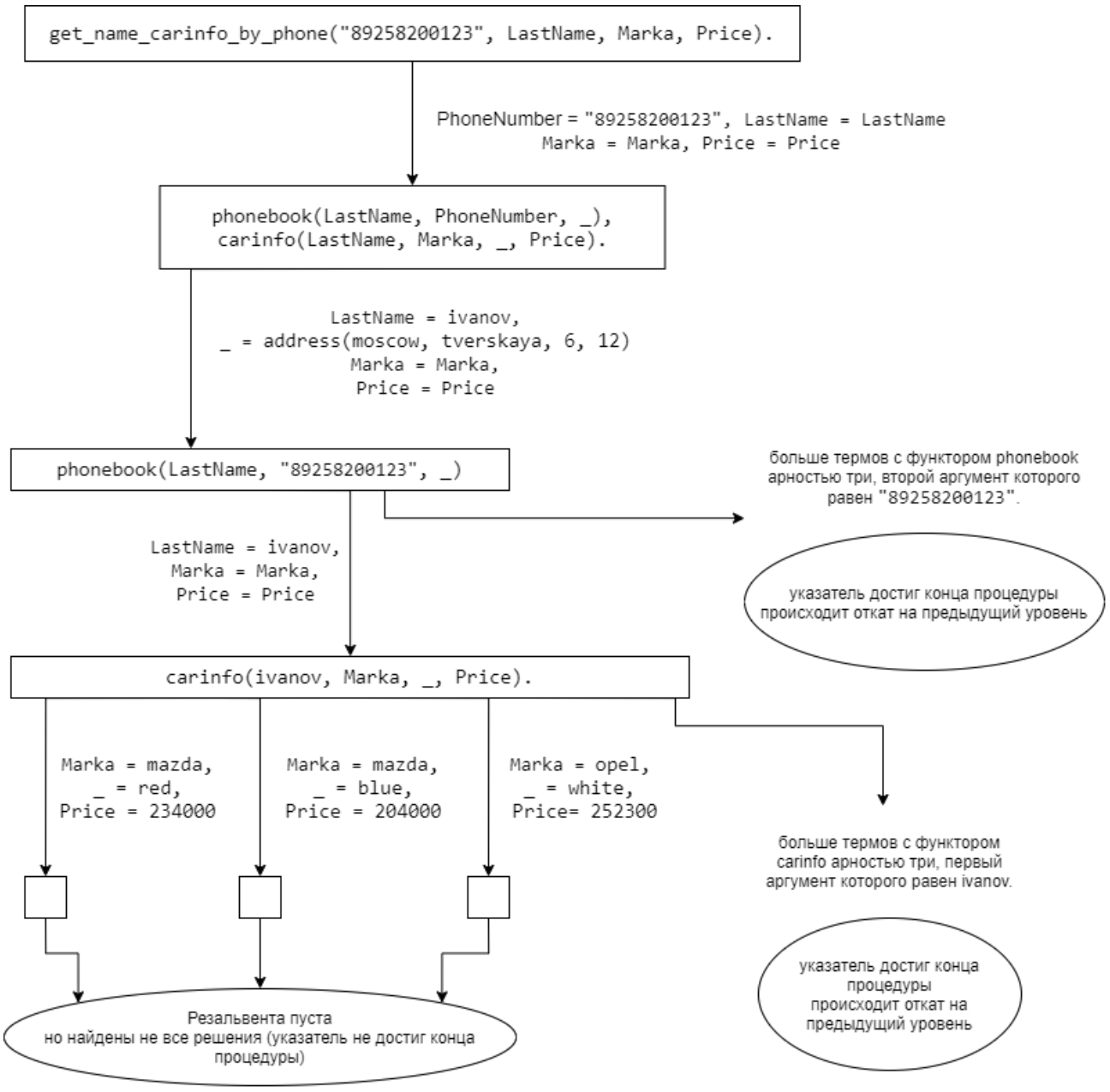
```
get_street_bank_phone_by_name_and_city(ivanov, moscow, Street, Bank,
PhoneNumber).
```

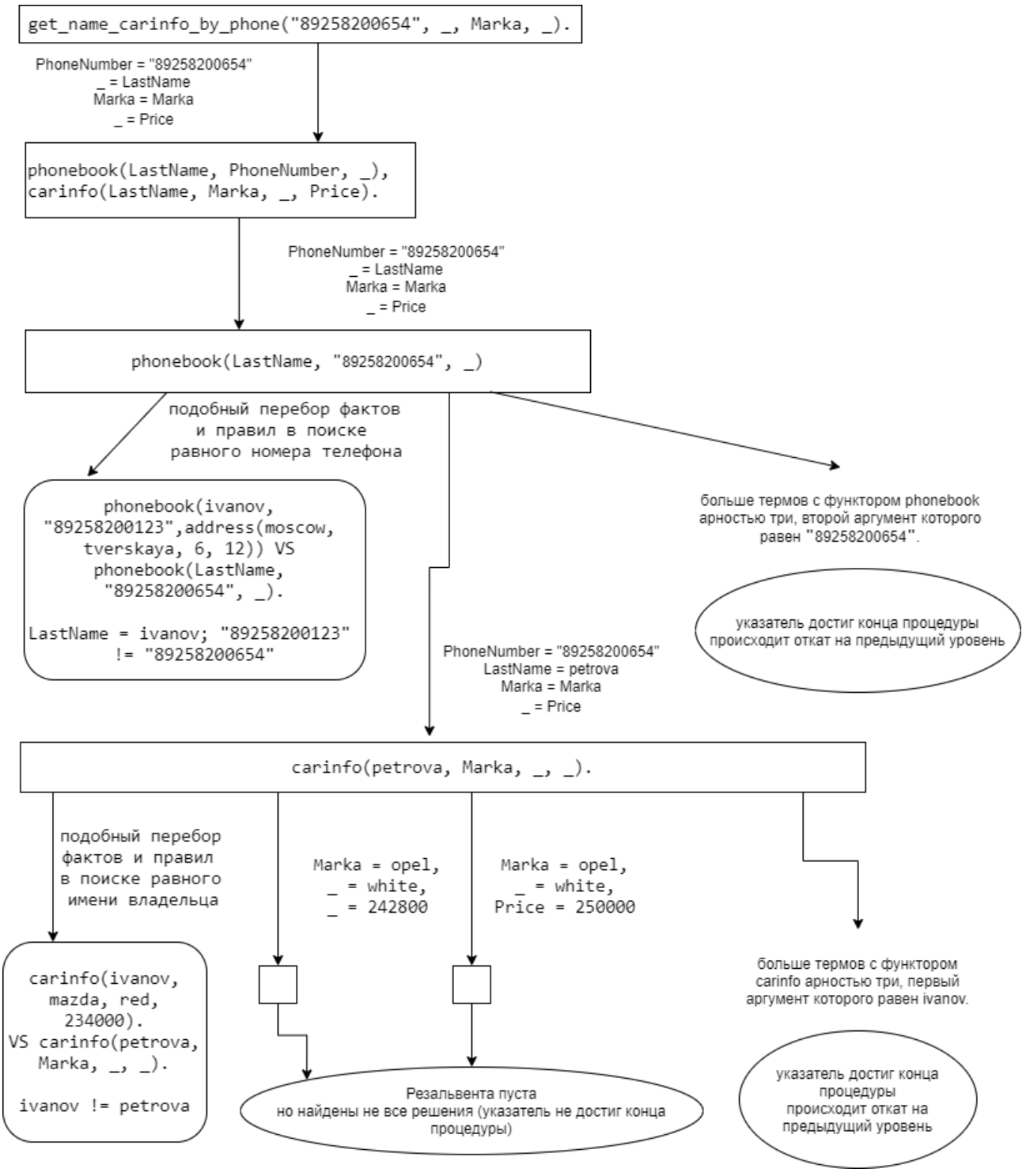
```
Street=tverskaya, Bank=sberbank, PhoneNumber=89258200123
```

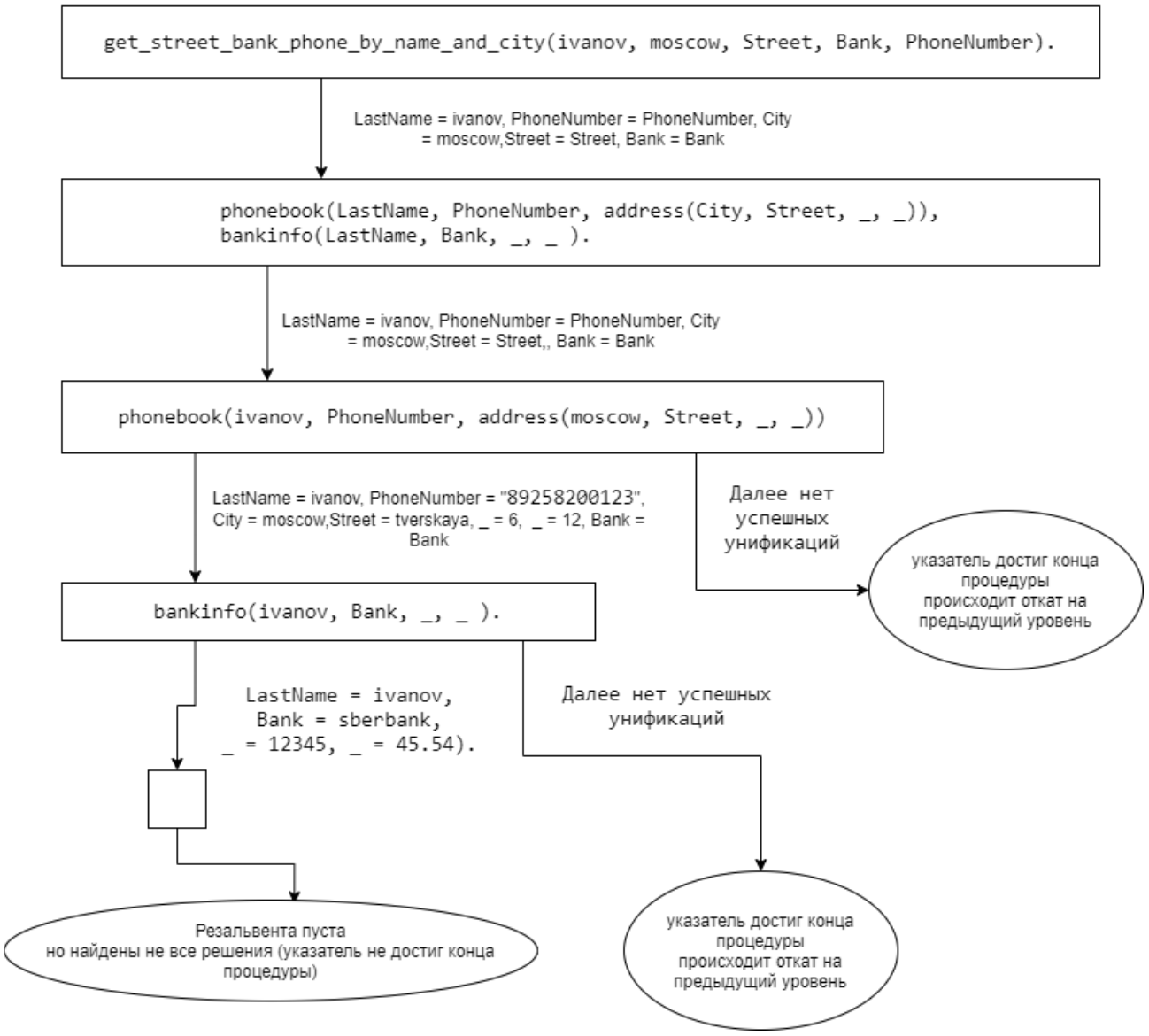
1 Solution
get_street_bank_phone_by_name_and_city(ivanov, tver, Street, Bank, PhoneNumber).
Street=moskovskaya, Bank=sberbank, PhoneNumber=89128600423 1 Solution
get_street_bank_phone_by_name_and_city(sidorov, tver, Street, Bank, PhoneNumber).
Street=lenina, Bank=tinkoff, PhoneNumber=89258234128 1 Solution

Для одного из вариантов ответов (и для 1а, и для 1б, и 2) описать словесно порядок поиска ответа на вопрос, указав, как выбираются знания, и, при этом, для каждого этапа унификации, выписать подстановку – наибольший общий унификатор, и соответствующие примеры термов.

В процессе выполнения программы система, используя встроенный алгоритм унификации, пытается обосновать возможность истинности вопроса, строя подстановки и примеры термов. Алгоритм унификации работает следующим образом: последовательно выполняет “сравнение” знания (терма) из базы знаний и поставленный вопрос. Ищется терм с тем же функтором, имеющий ту же арность и типы аргументов. Выполняется подстановка - связывание переменных со значениями. В случае доказательства истинности вопроса, программа выводит найденное решение - именованные переменные и связанные с ними значения. Если указатель, просматривающий в процессе унификации базу знаний, не достиг конца, то это означает возможность существования еще решений. Тогда происходит откат для дальнейшего поиска подходящих знаний.


get_name_carinfo_by_phone("89258200123", LastName, Marka, Price).
PhoneNumber = "89258200123", LastName = LastName
Marka = Marka, Price = Price
phonebook(LastName, PhoneNumber, _),
carinfo(LastName, Marka, _, Price).
LastName = ivanov,
_ = address(moscow, tverskaya, 6, 12)
Marka = Marka,
Price = Price
phonebook(LastName, "89258200123", _)
больше термов с функтором phonebook арностью три, второй аргумент которого равен "89258200123".
LastName = ivanov,
Marka = Marka,
Price = Price
указатель достиг конца процедуры
происходит откат на предыдущий уровень
carinfo(ivanov, Marka, _, Price).
Marka = mazda,
_ = red,
Price = 234000
Marka = mazda,
_ = blue,
Price = 204000
Marka = opel,
_ = white,
Price= 252300
больше термов с функтором carinfo арностью три, первый аргумент которого равен ivanov.
указатель достиг конца процедуры
происходит откат на предыдущий уровень
Резальвента пуста
но найдены не все решения (указатель не достиг конца процедуры)


get_name_carinfo_by_phone("89258200654", _, Marka, _).
PhoneNumber = "89258200654"
_ = LastName
Marka = Marka
_ = Price
phonebook(LastName, PhoneNumber, _),
carinfo(LastName, Marka, _, Price).
PhoneNumber = "89258200654"
_ = LastName
Marka = Marka
_ = Price
phonebook(LastName, "89258200654", _)
подобный перебор фактов и правил в поиске равного номера телефона
phonebook(ivanov, "89258200123",address(moscow, tverskaya, 6, 12)) VS phonebook(LastName, "89258200654", _).
LastName = ivanov; "89258200123" != "89258200654"
больше термов с функтором phonebook арностью три, второй аргумент которого равен "89258200654".
указатель достиг конца процедуры происходит откат на предыдущий уровень
PhoneNumber = "89258200654"
LastName = petrova
Marka = Marka
_ = Price
carinfo(petrova, Marka, _, _).
подобный перебор фактов и правил в поиске равного имени владельца
Marka = opel,
_ = white,
_ = 242800
Marka = opel,
_ = white,
Price = 250000
carinfo(ivanov, mazda, red, 234000). VS carinfo(petrova, Marka, _, _).
ivanov != petrova
Резальвента пуста
но найдены не все решения (указатель не достиг конца процедуры)
больше термов с функтором carinfo арностью три, первый аргумент которого равен ivanov.
указатель достиг конца процедуры происходит откат на предыдущий уровень



Вопросы:

1. Что такое терм?

Терм является основным элементом языка Prolog. Терм - это

- Константа:
  - Число (целое, вещественное),
  - Символьный атом (комбинация символов латинского алфавита, цифр и символа подчеркивания, начинающаяся со строчной буквы: aA, ab_2),
  - Строка: последовательность символов, заключенных в кавычки,
- Переменная:
  - Именованная – обозначается комбинацией символов латинского алфавита, цифр и символа подчеркивания, начинающейся с прописной буквы или символа подчеркивания (X, A21, _X),
  - Анонимная - обозначается символом подчеркивания (_),
- Составной терм:
  - Это средство организации группы отдельных элементов знаний в

единый объект, синтаксически представляется: f(t1, t2, …,tm), где f - функтор (функциональный символ) , t1, t2, …, tm – термы (их называют аргументами).

2. Что такое предикат в матлогике (математике)?

Предикат (n-местный или n-арный) - функция, определенная на множестве $M = M_1, M_2, \ldots, M_n$, которая для каждого набора элементов этого множества принимает одно из двух значений: 1 - истина, 0 - ложь.

3. Что описывает предикат в Prolog?

Предикаты в prolog описывают отношение (связь) между объектами.

4. Назовите виды предложений в программе и приведите примеры таких предложений из Вашей программы. Какие предложения являются основными, а какие – не основными? Каковы: синтаксис и семантика (формальный смысл) этих предложений (основных и неосновных)?

<ПРОЛОГ-предложение> ::= <правило> | <факт> | <запрос>
<правило> ::= <заголовок> ‘:-’ <тело>

```
get_name_carinfo_by_phone(PhoneNumber, LastName, Marka, Price) :-
      phonebook(LastName, PhoneNumber, _),
      carinfo(LastName, Marka, _, Price).
```

<факт> ::= <заголовок> ‘.’

```
carinfo(ivanov, mazda, red, 234000).
```

<запрос> ::= <тело>‘.’

```
phonebook(ivanov, _, Address).
```

Если предложение содержит переменную оно называется неосновным, иначе - основным.

В процессе унификации именные переменные связываются со значениями, таким образом мы можем получить содержательный ответ, а не только установить факт выводимости (или не выводимости) этого утверждения. Если эти данные нам неизвестны и при этом не важны (чтобы не передавать и не находить не нужную нам информацию), используем анонимную переменную, которая унифицируется с любым значением.

5. Каковы назначение, виды и особенности использования переменных в программе на Prolog? Какое предложение БЗ сформулировано в более общей – абстрактной форме: содержащее или не содержащее переменных?

Переменные бывают именованные и анонимные. Именованные переменные уникальны в рамках предложения, а любая анонимная переменная уникальна само по себе. В процессе выполнения программы именные переменные могут

связываться с различными объектами – конкретизироваться. Предложение, содержащее переменную (неосновное), сформулировано в более общей форме.

6. Что такое подстановка?

Пусть дан терм: A(X 1, X 2, X n). Подстановкой называется множество пар, вида: { x i = ti } , где x i – переменная, a ti – терм.

Пусть $\Theta = \{x_1 = t_1, \ldots, x_n = t_n\}$ – подстановка, тогда результат применения подстановки к терму обозначается: AΘ . Применение подстановки заключается в замене каждого вхождения переменной xi на соответствующий терм.

7. Что такое пример терма? Как и когда строится? Как Вы думаете, система строит и хранит примеры?

Терм B называется примером терма A, если существует такая подстановка Θ, что B=AΘ. Терм C называется общим примером термов A и B, если существуют такие подстановки Θ1 и Θ2, что C = A Θ1 и C=B Θ2.

Примеры терма строятся например в правилах, когда мы последовательно конкретизируем переменные. Конкретизированные раннее переменные подстанавливаются в терм. Подстановка хранится как переменные и связанные с ними значения. При унификации термов происходит сравнение с этими значениями (или конкретизация еще несвязанных переменных).