|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

**Отчет**

**по лабораторной работе № 13**

**Дисциплина: «Функциональное и логическое программирование»**

Выполнила: Овчинникова А. П.

Группа: ИУ7-65Б

Преподаватель: Толпинская Н. Б.

Строганов Ю. В.

Москва, 2020

**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).

Используя правила, обеспечить возможность поиска:

1. **а)** По № телефона найти: Фамилию, Марку автомобиля, Стоимость автомобиля (может быть несколько),

**в)** Используя сформированное в пункте **а)** правило, по № телефона найти: только Марку автомобиля (автомобилей может быть несколько),

1. 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.

**Для задания1 и задания2**:

для одного из вариантов ответов, и для **а)** и для **в)**, **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.

**Теоретическая часть**

1. Терм – это:

1. Константа:

* Число (целое, вещественное);
* Символьный атом;
* Строка.

1. Переменная:

* Именованная;
* Анонимная.

1. Составной терм:

* f(t1, t2, …,tm), где f - функтор (функциональный символ) , t1, t2, …,tm – термы, в том числе и составные (их называют аргументами).

2. Пусть M – произвольное непустое множество. Тогда n-местным предикатом P(x1, …, xn) называется функция , которая каждому упорядоченному набору элементов из множества M сопоставляет конкретное высказывание, обозначаемое .

Само множество M называется предметным множеством, а аргументы – предметными переменными.

3. Предикат – конструкция логической программы, задающая отношение между объектами предметного поля, описываемого программой. Т. о. предикат фиксирует отношения между некоторыми объектами.

В Prolog существует понятие процедуры. 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. Отношение, определяемое процедурой, называется предикатом.

4. Предложения бывают двух видов:

* факты (например, car\_owner(ivanov, "Ford", "Dark blue", "1000000").);
* правила (например, find\_model\_by\_phone(PHONE, MODEL) :-find\_by\_phone(PHONE, \_, MODEL, \_).).

Третьим специфическим видом предложений Пролога можно считать вопросы (например, find\_model\_by\_phone("89998693211", MODEL).).

Если составные термы, факты, правила и вопросы не содержат переменных, то они называются основными. Составные термы, факты, правила и вопросы в момент фиксации в программе могут содержать переменные, тогда они называются неосновными.

Пример основного факта: car\_owner(ivanov, "Ford", "Dark blue", "1000000").

Пример неосновного правила: find\_model\_by\_phone(PHONE, MODEL) :- find\_by\_phone(PHONE, \_, MODEL, \_).

5. 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.

В более общей форме сформулировано предложение, содержащее переменные.

6. Пусть дан терм: А (X1, X2, …, Xn).

Подстановкой называется множество пар, вида: { xi = ti }, где xi – переменная, а ti – терм.

Пусть Θ = { x1 = t1, x2= t2, … , xn = tn } – подстановка, тогда результат применения подстановки к терму обозначается: АΘ .

7. Терм В называется примером терма А, если существует такая подстановка Θ, что В=АΘ.

Терм С называется общим примером термов А и В, если существуют такие подстановки Θ1 и Θ2, что С = А Θ1 и С = В Θ2.

Пример строится при конкретизации переменной, используя алгоритм унификации. Система строит и хранит sпримеры.

**Практическая часть**

В листинге 1 представлен текст программы.

Листинг 1. Текст программы.

*include "lab13.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\_phone(string, symbol, string, string). % phone, surname, model, cost*

*find\_model\_by\_phone(string, string). % phone, model*

*find\_by\_city\_and\_surname(string, symbol, string, string, string). % city, surname, street, phone, bank*

*%find\_by\_city\_and\_surname(string, symbol, string).*

*clauses*

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

*abonent(ivanov, "89998693212", address("Vladivostok", "Tamanskaja", 5, 25)).*

*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)).*

*car\_owner(ivanov, "Ford", "Dark blue", "1000000").*

*car\_owner(ivanov, "Mazda", "Red", "500000").*

*car\_owner(petrov, "Lada", "Light gray", "100000").*

*car\_owner(korolev, "Land Rover", "Green", "1020000").*

*investor(ivanov, "Sberbank", "11111111111111111111", "50000").*

*investor(ivanov, "Alpha Bank", "11111111111111111112", "40000").*

*investor(petrov, "Sberbank", "21111111111111111111", "10000").*

*investor(sidorov, "Tochka", "31111111111111111111", "60000").*

*investor(sidorov, "Tinkoff", "31111111111111111112", "5000").*

*investor(ovchinnikova, "Sberbank", "41111111111111111111", "5000").*

*investor(zorina, "Sberbank", "41111111111111111111", "6000").*

*find\_by\_phone(PHONE, SURNAME, MODEL, COST) :-*

*abonent(SURNAME, PHONE, \_), car\_owner(SURNAME, MODEL, \_, COST).*

*find\_model\_by\_phone(PHONE, MODEL) :-*

*find\_by\_phone(PHONE, \_, MODEL, \_).*

*find\_by\_city\_and\_surname(CITY, SURNAME, STREET, PHONE, BANK) :-*

*abonent(SURNAME, PHONE, address(CITY, STREET, \_, \_)), investor(SURNAME, BANK, \_, \_).*

*goal*

*find\_by\_city\_and\_surname("Moscow", ivanov, STREET, PHONE, BANK).*

*%find\_model\_by\_phone("89998693211", MODEL).*

*%find\_by\_phone("89998693211", SURNAME, MODEL, COST).*

*%find\_by\_phone("89998693211", SURNAME, MODEL, COST).*

*%find\_by\_phone("89999999999", SURNAME, MODEL, COST).*

*%investor(SURNAME, BANK, DEPOSIT, AMOUNT).*

*%car\_owner(SURNAME, MODEL, COLOR, COST).*

*%abonent(SURNAME, PHONE, ADDRESS).*

**Примеры возможных вариантов вопросов и ответов**

1. Вопрос: *find\_by\_phone("89998693211", SURNAME, MODEL, COST).*

Ответ:

*SURNAME=ivanov, MODEL=Ford, COST=1000000*

*SURNAME=ivanov, MODEL=Mazda, COST=500000*

*2 Solutions*

1. Вопрос: *find\_by\_phone("89999999999", SURNAME, MODEL, COST).*

Ответ:

*No Solution*

1. Вопрос: *find\_by\_phone("89998695244", SURNAME, MODEL, COST).*

Ответ:

*No Solution*

1. Вопрос: *find\_model\_by\_phone("89998690001", MODEL).*

Ответ:

*MODEL=Land Rover*

*1 Solution*

1. Вопрос: *find\_model\_by\_phone("89999999999", MODEL).*

Ответ:

*No Solution*

1. Вопрос: find\_model\_by\_phone("89998693211", MODEL).

Ответ:

*MODEL=Ford*

*MODEL=Mazda*

*2 Solutions*

1. Вопрос: *find\_by\_city\_and\_surname("Moscow", ivanov, STREET, PHONE, BANK).*

Ответ:

*STREET=Tamanskaja, PHONE=89998693211, BANK=Sberbank*

*STREET=Tamanskaja, PHONE=89998693211, BANK=Alpha Bank*

*2 Solutions*

1. Вопрос: *find\_by\_city\_and\_surname("Moscow", korolev, STREET, PHONE, BANK).*

Ответ:

*No Solution*

1. Вопрос: *find\_by\_city\_and\_surname("Orekhovo-Zuevo", ovchinnikova, STREET, PHONE, BANK).*

Ответ:

*STREET=Volodarskogo, PHONE=89998695230, BANK=Sberbank*

*1 Solution*

**Порядок поиска ответов**

Рассмотрим порядок поиска системой ответа на вопрос *find\_by\_phone("89998693211", SURNAME, MODEL, COST)*.

Правило описывает утверждение, которое истинно, только если выполнены все цели в правой части правила. В правиле *find\_by\_phone(PHONE, SURNAME, MODEL, COST) :- abonent(SURNAME, PHONE, \_), car\_owner(SURNAME, MODEL, \_, COST).* В правой части содержится две подцели. Подцели, образующие составную цель, доказываются последовательно, слева направо. Соответственно система сначала пытается доказать подцель *abonent(SURNAME, PHONE, \_)*.

Для этого система ищет применимое предложение пролог-программы путем последовательного, от начала к концу программы, просмотра всех ее предложений. Предложение C применимо, либо если это факт вида A., и он сопоставим с доказываемой целью, либо же это правило вида A :- B1, …, Bm, и его заголовок-предикат A сопоставим с доказываемой целью. Если применимых предложений в программе нет, то неуспех доказательства, выход из процедуры с ответом no.

Знания (по умолчанию) просматриваются сверху вниз. Первое знание – *abonent(ivanov, "89998693211", address("Moscow", "Tamanskaja", 4, 24)).* . Соответственно система пытается унифицировать термы *abonent(ivanov, "89998693211", address("Moscow", "Tamanskaja", 4, 24)).* и *abonent(SURNAME, PHONE, \_).* .

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат |
| 1 | *find\_by\_phone("89998693211", SURNAME, MODEL, COST). = find\_by\_phone(PHONE, SURNAME, MODEL, COST).*  Неконкретизированные переменные SURNAME, MODEL, COST в вопросе и правиле становятся сцепленными, а переменная PHONE конкретизируется значением *"89998693211"*. Унификация успешна. | Прямой ход. Доказательство первой цели тела правила. |
| 2 | *abonent(ivanov,"89998693211",address("Moscow","Tamanskaja", 4, 24)). = abonent(SURNAME, "89998693211", \_).*  Первая пара термов (константа и неконкретизированная переменная) унифицируется успешно. Вторая пара термов (две одинаковые константы) унифицируется успешно. Третья пара термов (составной терм, содержащий константы, и анонимная переменная) унифицируется успешно. Унификация успешна. Переменной *SURNAME* сопоставляется значение i*vanov*.  Подстановка: Θ = { *SURNAME*=ivanov, *"89998693211", address("Moscow","Tamanskaja", 4, 24)* }.  Пример терма: *abonent(Ivanov, "89998693211", address("Moscow","Tamanskaja", 4, 24)).* | Прямой ход. Доказательство второй цели правила. |
| 3 | *car\_owner(SURNAME,MODEL,\_,COST).=car\_owner(ivanov, "Ford", "Dark blue", "1000000").*  Первая пара термов (две одинаковые константы – *ivanov*) унифицируется успешно. Вторая пара термов (неконкретизированная переменная и константа) унифицируется успешно. Третья пара термов (анонимная переменная и константа) унифицируется успешно. Четвертая пара термов (неконкретизированная переменная и константа) унифицируется успешно. Унификация успешна. Переменным MODEL и COST сопоставляются значения «Ford» и «1000000» соответственно.  Подстановка: Θ = *(Ivanov, MODEL=”Ford”, “Dark blue”, COST=”1000000”).*  Пример терма: *car\_owner(Ivanov, ”Ford”, “Dark blue”, ”1000000”).* | Вторая цель доказана. Правило доказано. Одно решение найдено. Продолжается поиск решений. |

Аналогичным образом продолжается поиск остальных решений, если они есть. Если унификация происходит не успешно, то пролог-система производит бэктрекинг и начинает новую попытку доказательства, уже с помощью следующего применимого предложения программы.

В результате на данный вопрос система находит два ответа:

*SURNAME=ivanov, MODEL=Ford, COST=1000000*

*SURNAME=ivanov, MODEL=Mazda, COST=500000*

*2 Solutions*

Рассмотрим порядок поиска системой ответа на вопрос *find\_model\_by\_phone("89998690001", MODEL).*

Правило *find\_model\_by\_phone(PHONE, MODEL ):-find\_by\_phone (PHONE, \_, MODEL, \_).* содержит одну поддцель. Соответственно система пытается доказать подцель *find\_by\_phone (PHONE, \_, MODEL, \_).* .

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат |
| 1 | *find\_model\_by\_phone(PHONE,MODEL)=* *find\_model\_by\_phone("89998690001", MODEL).*  Переменная *PHONE* конкретизируется значением *"89998690001",* а переменная MODEL в вопросе и правиле становится сцепленной. | Прямой ход. Доказательство первой цели тела правила. |
| 2 | *find\_by\_phone("89998690001",\_,MODEL,\_).= find\_by\_phone("89998690001",SURNAME, MODEL, COST).*  Переменная MODEL в заголовке и теле правила становится сцепленной. | Прямой ход. Доказательство первой цели тела правила. |
| 3 | *abonent(SURNAME,"89998690001",\_)=abonent(ivanov,"89998693211", address("Moscow", "Tamanskaja", 4, 24)).*  Первая пара термов (неконкретизированная переменная и константа) унифицируются успешно. Вторая пара термов (две разные константы) не унифицируются успешно. Унификация не пройдена. | Откат. Переход к следующему по порядку применимому предложению. |
| 4-10 | Унификация не пройдена. | Откат. Переход к следующему по порядку применимому предложению. |
| 11 | *abonent(SURNAME,"89998690001",\_)=abonent(korolev,"89998690001",address("Saint-Petersburg","Nevsky prospekt", 12, 21)).*  Первая пара термов (неконкретизированная переменная и константа) унифицируются успешно. Вторая пара термов (две одинаковых константы) унифицируются успешно. Третья пара термов (анонимная переменная и константа) унифицируются успешно.  Переменная SURNAME конкретизируется значением *korolev.*  Подстановка: Θ = {SURNAME=korolev,*"89998690001”, address("Saint-Petersburg","Nevsky prospekt", 12, 21)*}  Пример терма: *abonent(korolev,"89998690001”, address("Saint-Petersburg","Nevsky prospekt", 12, 21).* | Шаг вперед. Доказательство второй цели тела правила. |
| 12 | *car\_owner(korolev, MODEL, \_, COST).= car\_owner(ivanov, "Ford", "Dark blue", "1000000").*  Первая пара термов (разные константы) не унифицируема. Унификация не пройдена. | Откат. Переход к следующему по порядку применимому предложению. |
| 13-14 | Унификация не пройдена. | Откат. Переход к следующему по порядку применимому предложению. |
| 15 | *car\_owner(korolev, MODEL, \_, COST).=* *car\_owner(korolev, "Land Rover", "Green", "1020000").*  Первая пара термов (две одинаковые константы) унифицируются успешно. Вторая пара термов (неконкретизированная переменная и константа) унифицируются успешно. Третья пара термов унифицируется успешно. Четвертая пара термов унифицируется успешно. Унификация успешна.  Переменные MODEL и COST конкретизируются значениями *"Land Rover"* и *"1020000"* соответственно.  Подстановка: Θ = {*korolev* , *MODEL="Land Rover", “Green”, COST=”1020000”*}  Пример терма: *car\_owner(korolev, "Land Rover", “Green”,”1020000”).* | Вторая цель доказана. Правило доказано. |
| 16 | *find\_by\_phone("89998690001",\_,MODEL,\_).= find\_by\_phone("89998690001",korolev, “Land Rover”, “1020000”).*  Унификация успешна.  Переменная MODEL конкретизируется значением *“Land Rover”.*  Подстановка: Θ = {*"89998690001",korolev, MODEL=“Land Rover”, “1020000”*}  Пример терма: *find\_by\_phone("89998690001", korolev,”Land Rover”,”1020000”).* | Шаг вперед. |
| … | Поиск других решений для правила *find\_by\_phone(PHONE, SURNAME, MODEL, COST)* | … |
| 17 | *find\_model\_by\_phone("89998690001",”Land Rover”)=* *find\_model\_by\_phone("89998690001", MODEL).*  Унификация успешна. | Шаг вперед. Правило доказано. Продолжается поиск решений. |
| … | Поиск других решений для правила *find\_model\_by\_phone(PHONE, MODEL)* | … |

Рассмотрим порядок поиска системой ответа на вопрос *find\_by\_city\_and\_surname("Moscow", korolev, STREET, PHONE, BANK).*

Правило *find\_by\_city\_and\_surname(CITY, SURNAME, STREET, PHONE, BANK):-abonent(SURNAME, PHONE, address(CITY, STREET, \_, \_)), investor(SURNAME, BANK, \_, \_).* содержит две подцели. Система доказывает их по порядку (слева направо).

|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат |
| 1 | *find\_by\_city\_and\_surname("Moscow", korolev, STREET, PHONE, BANK).= find\_by\_city\_and\_surname(CITY, SURNAME, STREET, PHONE, BANK).*  Переменные CITY и SURNAME конкретизируются значениями *“Moscow”* и *korolev* соответственно. Переменные STREET, PHONE и BANK в вопросе и теле правила становятся сцепленными. | Прямой ход. Доказательство первой цели тела правила. |
| 2 | *abonent(korolev,PHONE,address(“Moscow”, STREET,\_,\_)).=abonent(ivanov,"89998693211", address("Moscow", "Tamanskaja", 4, 24)).*  Унификация не пройдена. | Откат. Переход к следующему по порядку применимому предложению. |
| 3-9 | Унификация не пройдена. | Откат. Переход к следующему по порядку применимому предложению. |
| 10 | *abonent(korolev,PHONE,address(“Moscow”, STREET,\_,\_)).=* *abonent(korolev, "89998690001", address("Saint-Petersburg", "Nevsky prospekt", 12, 21)).*  Унификация не пройдена. | Применимых предложений больше нет. Неуспех доказательства. Выход из процедуры с ответом no. |

В этом примере система не находит ответов, поэтому выводит сообщение «No solutions».