**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 № 16.**

По курсу «Функциональное и логическое программирование»..

Студент: Власова Е. В.

Группа: ИУ7-64Б.

Преподаватель: Толпинская Н. Б.

*2020 г.*

**Практическая часть**

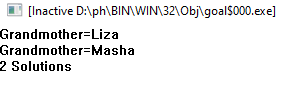
**Код программы:**

|  |
| --- |
| predicates  grandparents(symbol Flag, symbol Grandmother, symbol Grandfather, symbol Child)  parents(symbol Mother, symbol Father, symbol Child)  clauses  parents("Liza", "Oleg", "Nastya").  parents("Masha", "Igor", "Lesha").    parents("Sasha", "Vadim", "Misha").  parents("Nastya", "Lesha", "Ira").    parents("Ira", "Misha", "Dima").    %по линии матери  grandparents("Mum", Grandmother, Grandfather, Child):-  parents(Mother, \_, Child),  parents(Grandmother, Grandfather, Mother).    %по линии отца  grandparents("Dad", Grandmother, Grandfather, Child):  parents(\_, Father, Child),  parents(Grandmother, Grandfather, Father).    goal  grandparents(\_, GrandMother, Grandfather, "Dima"). |

**Примеры работы:**

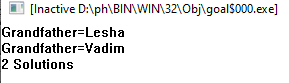
1. По имени субъекта определить всех бабушек:

|  |
| --- |
| Goal  grandparents(\_, Grandmother, \_, "Ira"). |

****

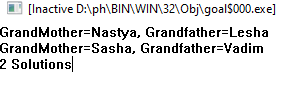
1. По имени определить всех дедушек:

|  |
| --- |
| goal  grandparents(\_, \_, Grandfather, "Dima"). |

****

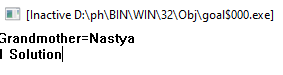
1. По имени определить всех бабушек и дедушек:

|  |
| --- |
| goal  grandparents(\_, GrandMother, Grandfather, "Dima"). |

****

1. По имени определить всех бабушек по материнской линии:

|  |
| --- |
| goal  grandparents("Mum", Grandmother, \_, "Dima"). |

****

1. По имени определить всех бабушек по отцовской линии:

|  |
| --- |
| goal  grandparents("Dad", Grandmother, \_, "Dima"). |

****

**Таблица для 4 пункта:**

|  |
| --- |
| goal  grandparents("Mum", Grandmother, \_, "Dima"). |

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Состояние резольвенты, и вывод: дальнейшие действия (почему?) | Для каких термов запускается алгоритм унификации: Т1=Т2 и каков **результат** (и подстановка) | Дальнейшие действия: прямой ход или откат (почему и к чему приводит?) |
| 1 | Цель заносится в резольвенту:  grandparents("Mum", Grandmother, \_, "Dima")  Выполнение процесса редукции | Унификация grandparents("Mum", Grandmother, \_, "Dima")  И parents("Liza", "Oleg", "Nastya")  Результат: неудача, разные функторы | Прямой ход, переход к следующему предложению  (т.к. не конец БЗ, резольвента не пуста) |
| 2 | grandparents("Mum", Grandmother, \_, "Dima")  Выполнение процесса редукции | Унификация grandparents("Mum", Grandmother, \_, "Dima")  И parents("Masha", "Igor", "Lesha")  Результат: неудача, разные функторы | Прямой ход, переход к следующему предложению |
| 3 | grandparents("Mum", Grandmother, \_, "Dima")  Выполнение процесса редукции | Унификация grandparents("Mum", Grandmother, \_, "Dima")  И parents("Sasha", "Vadim", "Misha")  Результат: неудача, разные функторы | Прямой ход, переход к следующему предложению |
| 4 | grandparents("Mum", Grandmother, \_, "Dima")  Выполнение процесса редукции | Унификация grandparents("Mum", Grandmother, \_, "Dima")  И parents("Nastya", "Lesha", "Ira")  Результат: неудача, разные функторы | Прямой ход, переход к следующему предложению |
| 5 | grandparents("Mum", Grandmother, \_, "Dima")  Выполнение процесса редукции | Унификация grandparents("Mum", Grandmother, \_, "Dima")  И parents("Ira", "Misha", "Dima")  Результат: неудача, разные функторы | Прямой ход, переход к следующему предложению |
| 6 | grandparents("Mum", Grandmother, \_, "Dima")  Выполнение процесса редукции | Унификация  grandparents("Mum", Grandmother, \_, "Dima")  и  grandparents("Mum", Grandmother, Grandfather, Child)  Результат: успех  Подстановка:  Grandmother = Grandmother  Child = «Dima» | Преобразование резольвенты. (замена текущей цели на тело найденного с помощью унификации факта, применение полученной подстановки)  Прямой ход |
| 7 | parents(Mother, \_, «Dima»)  parents(Grandmother, Grandfather, Mother)  Выбор верхней подцепи резольвенты, запуск редукции | Унификация  parents(Mother, \_, «Dima»)  и  parents("Liza", "Oleg", "Nastya")  Результат: неудача, несовпадающие константы | Прямой ход, переход к следующему предложению |
| 8 | parents(Mother, \_, «Dima»)  parents(Grandmother, Grandfather, Mother)  Выполнение процесса редукции | Унификация  parents(Mother, \_, «Dima»)  и  parents("Masha", "Igor", "Lesha")  Результат: неудача, несовпадающие константы | Прямой ход, переход к следующему предложению |
| … | parents(Mother, \_, «Dima»)  parents(Grandmother, Grandfather, Mother)  Выполнение процесса редукции | Унификация  parents(Mother, \_, «Dima»)  и термов, пока не будет найден подходящий или пока не закончится БЗ | Прямой ход, переход к следующему предложению |
| 11 | parents(Mother, \_, «Dima»)  parents(Grandmother, Grandfather, Mother)  Выполнение процесса редукции | Унификация  parents(Mother, \_, «Dima»)  и parents("Ira", "Misha", "Dima")  Результат: успех  Подстановка: Mother= «Ira» | Преобразование резольвенты. (замена текущей цели на тело найденного с помощью унификации факта, применение полученной подстановки)  Прямой ход |
| 12 | parents(Grandmother, Grandfather, «Ira»)  Выбор верхней подцепи резольвенты, запуск редукции | Унификация  parents(Grandmother, Grandfather, «Ira»)  и  parents("Liza", "Oleg", "Nastya")  Результат: неудача, разные константы | Прямой ход, переход к следующему предложению |
| … | parents(Grandmother, Grandfather, «Ira»)  Выполнение процесса редукции | Унификация  parents(Grandmother, Grandfather, «Ira»)  и термов, пока не будет найден подходящий или пока не закончится БЗ | Прямой ход, переход к следующему предложению |
| 15 | parents(Grandmother, Grandfather, «Ira»)  Выполнение процесса редукции | Унификация  parents(Grandmother, Grandfather, «Ira»)  и parents("Nastya", "Lesha", "Ira")  Результат: успех  Подстановка:  Grandmother="Nastya"  Grandfather="Lesha" | Преобразование резольвенты. (замена текущей цели на тело найденного с помощью унификации факта, применение полученной подстановки) |
| 16 | **Резольвента пуста** |  | **Запуск отката** |
| 17 | parents(Grandmother, Grandfather, «Ira»)  Выполнение процесса редукции | Унификация  parents(Grandmother, Grandfather, «Ira»)  и parents("Ira", "Misha", "Dima")  Результат: неудача, разные константы | Прямой ход |
| 18 | parents(Grandmother, Grandfather, «Ira»)  Выполнение процесса редукции | Унификация  parents(Grandmother, Grandfather, «Ira»)  и grandparents("Mum", Grandmother, Grandfather, Child)  Результат: неудача, разные функторы | Прямой ход |
| 19 | parents(Grandmother, Grandfather, «Ira»)  Выполнение процесса редукции | Унификация  parents(Grandmother, Grandfather, «Ira»)  и grandparents("Dad", Grandmother, Grandfather, Child)  Результат: неудача, разные функторы | Конец БЗ, резольвента не пустая, **откат** |
| 20 | parents(Mother, \_, «Dima»)  parents(Grandmother, Grandfather, Mother)  Выполнение процесса редукции | Унификация parents(Mother, \_, «Dima»)  И grandparents("Mum", Grandmother, Grandfather, Child)  Результат: неудача, разные функторы | Прямой ход |
| 21 | Выполнение редукции | Унификация parents(Mother, \_, «Dima»)  И grandparents("Dad", Grandmother, Grandfather, Child)  Результат: неудача, разные функторы | Конец БЗ, резольвента не пустая, откат |
| 22 | grandparents("Mum", Grandmother, \_, "Dima")  Запуск процесса редукции | Унификация grandparents("Mum", Grandmother, \_, "Dima")  И grandparents("Dad", Grandmother, Grandfather, Child)  Результат: неудача, разные константы | Конец БЗ, резольвента пустая, **конец работы программы** |

**ВОПРОСЫ**

* **В каком случае система запускает алгоритм унификации? (Как эту необходимость на формальном уровне распознает система?)**

Если есть что доказывать (задана цель доказательства).

* **Каковы назначение и результат использования алгоритма унификации?**

Назначение унификации – подобрать знание, которое можно использовать для доказательства истинности вопроса.

Алгоритм унификации может завершиться успехом (тогда формируется подстановка, содержащая значения переменных, при которых вопрос станет примером программы), а может завершиться неудачей (тупиковая ситуация).

* **Какое первое состояние резольвенты?**

Первое состояние - заданный вопрос

* **Как меняется резольвента?**

Резольвента меняется в ходе доказательства. Она хранится в стеке.

Преобразования выполняются с помощью редукции – замены текущей цели на тело найденного в программе правила.

Преобразование резольвенты разделено на два этапа:

1. Берётся верхняя из подцелей резольвенты и заменяется на тело правила, найденного в программе;
2. Затем к полученной конъюнкции целей применяется подстановка (наибольший общий унификатор цели и сопоставленного с ней правила).

* **В каких пределах программы уникальны переменные?**

Именованная переменная уникальна в рамках предложения, в котором она используется.

Анонимные переменные являются уникальными.

* **Как применяется подстановка, полученная с помощью алгоритма унификации?**

Полученная с помощью алгоритма унификации подстановка применяется к целям в резольвенте.

* **В каких случаях запускается механизм отката?**

В случае, если решение не найдено, и из данного состояния невозможен переход в новое состояние.

**Исправления л.р.14**

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

Знание сформулировано в заголовке правила. Заголовок содержит отдельное знание о предметной области.

1. **Сколько в БЗ текущего задания процедур?**

4 процедуры.

1. **Назначение и результат работы алгоритма унификации.**

Назначение - поиск содержательного ответа на поставленный вопрос, с помощью имеющейся базы знаний. Может завершиться успехом или тупиковой ситуацией(неудачей). Результат работы – наиболее общий унификатор.

**Программа:**

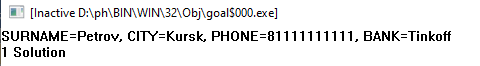
|  |
| --- |
| domains  home = address(CITY, STREET, HOUSE, FLAT)  SURNAME = string  PHONE = string  CITY, STREET = string  HOUSE, FLAT = integer  MARKA, COLOR = string  PRICE = integer  BANK = string  NUMBER, MONEY = integer  predicates  info(SURNAME, PHONE, home)  cars(SURNAME, CITY, MARKA, COLOR, PRICE)  banks(SURNAME, CITY, BANK, NUMBER, MONEY)  find\_by\_car(SURNAME, CITY, PHONE, MARKA, COLOR, BANK)  clauses  info("Petrov", "88885353535", address("Moscow", "Saykin street", 10, 143)).  info("Petrov", "81111111111", address("Kursk", "Non street", 1, 10)).  info("Krot", "89995353566", address("Moscow", "Nwot street", 5, 60)).  %2 owners  cars("Petrov", "Moscow", "TOYOTA", "RED", 256000). %PERSON 1  cars("Krot", "Moscow", "TOYOTA", "RED", 256000). %PERSON 3  %1 owner  cars("Petrov", "Kursk", "LADA", "GREEN", 100000). %PERSON 2  %0 owners  cars("Krotova", "Moscow", "Merc", "BLUE", 100000).    banks("Petrov", "Moscow", "Sberbank", 356, 50000). %PERSON 1  banks("Petrov", "Kursk", "Tinkoff", 357, 9000). %PERSON 2  banks("Krot", "Moscow", "Rocket-bank", 20, 2000). %PERSON 3    find\_by\_car(SURNAME, CITY, PHONE, MARKA, COLOR, BANK):- cars(SURNAME, CITY, …………..MARKA, COLOR, \_),  info(SURNAME, PHONE, address(CITY, \_, \_, \_)),  banks(SURNAME, CITY, BANK, \_, \_).  goal  find\_by\_car(SURNAME, CITY, PHONE, "TOYOTA", "RED", BANK). |

**Порядок работы алгоритма унификации**

На 1-м шаге стек - не полный ИСПРАВИТЬ!!!!

1. У машины один владелец:

|  |
| --- |
| Goal  find\_by\_car(SURNAME, CITY, PHONE, "LADA", "GREEN", BANK). |



|  |  |  |
| --- | --- | --- |
| № шага | Сравниваемые термы; результат; подстановка, если есть | Дальнейшие действия: прямой ход или откат |
| 0 | find\_by\_car(SURNAME, CITY, PHONE, MARKA, COLOR, BANK)  =  find\_by\_car(SURNAME, CITY, PHONE, «LADA», «GREEN», BANK) | Заносится в стек  find\_by\_car(SURNAME, CITY, PHONE, MARKA, COLOR, BANK)  =  find\_by\_car(SURNAME, CITY, PHONE, «LADA», «GREEN», BANK) |
| 1 | Раскрывается правило, то есть надо доказать каждое утверждение в теле правила (последовательно)  cars(SURNAME, CITY, MARKA, COLOR, PRICE)  =  cars(SURNAME, CITY, «LADA», «GREEN», PRICE) | Прямой ход  Заносится в стек  cars(SURNAME, CITY, «LADA», «GREEN», PRICE) |
| 2 | cars(SURNAME, CITY, «LADA», «GREEN», PRICE)  По функтору и списку аргументов ищется определение отношения  cars("Petrov", "Moscow", "TOYOTA", "RED", 256000) | Определение отношения найдено |
| 3 | Унификация  cars(SURNAME, CITY, «LADA», «GREEN», PRICE) и cars("Petrov", "Moscow", "TOYOTA", "RED", 256000) | Результат сравнения термов false, прямой ход |
| 4 | Унификация  cars(SURNAME, CITY, «LADA», «GREEN», PRICE) и  cars("Krot", "Moscow", "TOYOTA", "RED", 256000) | Результат сравнения термов false, прямой ход |
| 5 | Унификация cars(SURNAME, CITY, «LADA», «GREEN», PRICE)  и  cars("Petrov", "Kursk", "LADA", "GREEN", 100000) | Результат сравнения термов true  Подстановка:  SURNAME = "Petrov"  CITY = "Kursk"  PRICE = 100000  Переход к следующему утверждению в теле правила (прямой ход) |
| 6 | Следующее утверждение  info(SURNAME, PHONE, address(CITY, \_, \_, \_)) |  |
| 7 | Раскрывается правило  info(SURNAME, PHONE, address(CITY, \_, \_, \_))  ->  info("Petrov", PHONE, address("Kursk", \_, \_, \_))  По функтору и списку аргументов ищется определение отношения (с самого начала) | Определение отношения найдено  Заносится в стек  info("Petrov", PHONE, address("Kursk", \_, \_, \_)) |
| 8 | Унификация  info("Petrov", PHONE, address("Kursk", \_, \_, \_)) и  info("Petrov", "88885353535", address("Moscow", "Saykin street", 10, 143)) | Результат сравнения термов false, прямой ход |
| 9 | Унификация  info("Petrov", PHONE, address("Kursk", \_, \_, \_)) и  info("Petrov", "81111111111", address("Kursk", "Non street", 1, 10) | Результат сравнения термов true  Подстановка:  PHONE = "81111111111"  Анонимные переменные не связываются со значением  Переход к следующему утверждению в теле правила (прямой ход) |
| 10 | Следующее утверждение  banks(SURNAME, CITY, BANK, \_, \_) |  |
| 11 | Раскрывается правило  banks(SURNAME, CITY, BANK, \_, \_)  ->  banks("Petrov", "Kursk", BANK, \_, \_)  По функтору и списку аргументов ищется определение отношения | Определение отношения найдено  Заносится в стек  banks("Petrov", "Kursk", BANK, \_, \_) |
| 12 | Унификация  banks("Petrov", "Kursk", BANK, \_, \_)  и  banks("Petrov", "Moscow", "Sberbank", 356, 50000) | Результат сравнения термов false, прямой ход |
| 13 | Унификация  banks("Petrov", "Kursk", BANK, \_, \_)  и  banks("Petrov", "Kursk", "Tinkoff", 357, 9000) | Результат сравнения термов true  Подстановка:  BANK = "Tinkoff"  Вывод результата, переход к следующей строке |
| 14 | Унификация  banks("Petrov", "Kursk", BANK, \_, \_)  и  banks("Krot", "Moscow", "Rocket-bank", 20, 2000). | Результат сравнения термов false. В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся banks("Petrov", "Kursk", BANK, \_, \_) |
| 15 | Унификация  info("Petrov", PHONE, address("Kursk", \_, \_, \_)) и  info("Krot", "89995353566", address("Moscow", "Nwot street", 5, 60)) | Результат сравнения термов false. В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся info("Petrov", PHONE, address("Kursk", \_, \_, \_)) |
| 16 | Унификация  cars(SURNAME, CITY, «LADA», «GREEN», PRICE) и  cars("Krotova", "Moscow", "Merc", "BLUE", 100000) | Результат сравнения термов false. В базе знаний больше нет утверждений с заданным именем. Откат, из стека достаётся cars(SURNAME, CITY, «LADA», «GREEN», PRICE) |
| 17 | Следующее утверждение  find\_by\_car(SURNAME, CITY, PHONE, «LADA», «GREEN», BANK)  Сравнение оставшихся термов false, т.к. в базе знаний больше нет подходящих утверждений | Откат, из стека достаётся  find\_by\_car(SURNAME, CITY, PHONE, «LADA», «GREEN», BANK) |
| 18 | Стек пуст | Стек пуст, завершение программы |