|  |  |
| --- | --- |
| 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 **ПРОГРАМНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.03 ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

**Отчет**

|  |  |
| --- | --- |
| **По лабораторной работе №** | 1 16 |

**Дисциплина:** Функциональное и логическое программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-62Б |  |  | Н.А. Гарасев |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Н.Б.Толпинская |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2021

**Цель работы** –изучить использование правил в программе: структуру, особенности оформления, а также, способ и принципы выполнения таких программ на Prolog.

**Задачи работы**: приобрести навыки эффективного декларативного описания предметной области с использованием фактов и правил.

Изучить порядок использования фактов и правил в программе на Prolog, принципы и особенности сопоставления и отождествления термов, на основе механизма унификации. Способ формирования и изменения резольвенты. Порядок формирования ответа.

**Ход работы**

**Создать базу знаний: «ПРЕДКИ»**, позволяющую **наиболее эффективным** способом (за меньшее количество шагов, что обеспечивается меньшим количеством предложений БЗ - правил), используя разные варианты (примеры) **одного вопроса**, определить (указать: какой вопрос для какого варианта):

1. по имени субъекта определить всех его бабушек (предки 2-го колена),
2. по имени субъекта определить всех его дедушек (предки 2-го колена),
3. по имени субъекта определить всех его бабушек и дедушек (предки 2-го колена),
4. по имени субъекта определить его бабушку по материнской линии (предки 2-го колена),
5. по имени субъекта определить его бабушку и дедушку по материнской линии (предки 2-го колена).

Минимизировать количество правил и количество вариантов вопросов. Использовать **конъюнктивные правила и простой вопрос.**

**Код программы**

domains

name, sex = symbol

predicates

parent(name, sex, name).

grands(name, name, name, name, name).

clauses

parent("Irina", "woman", "Nikita"). % Nikita`s mother is Irina

parent("Raya", "woman", "Irina"). % Irina`s mother is Raya

parent("Vera", "woman", "Alex"). % Alex`s mother is Vera

parent("Alex", "man", "Nikita"). % Nikita`s mother is Alex

parent("Anatoly", "man", "Alex"). % Alex`s mother is Anatoly

parent("Vitaliy", "man", "Irina"). % Vitaliya`s mother is Irina

grands(Name, MotherMother, MotherFather, FatherMother, FatherFather):-

parent(Mother, "woman", Name),

parent(MotherMother, "woman", Mother),

parent(MotherFather, "man", Mother),

parent(Father,"man", Name),

parent(FatherMother,"woman", Father),

parent(FatherFather,"man", Father).

1. по имени субъекта определить всех его бабушек (предки 2-го колена),

grands("Nikita", MotherMother, \_ , FatherMother, \_ ).



1. по имени субъекта определить всех его дедушек (предки 2-го колена),

grands("Nikita", \_ , MotherFather, \_, FatherFather).



1. по имени субъекта определить всех его бабушек и дедушек (предки 2-го колена),

grands("Nikita", MotherMother, MotherFather, FatherMother, FatherFather).



1. по имени субъекта определить его бабушку по материнской линии (предки 2-го колена),

grands("Nikita", MotherMother, \_, \_, \_).



1. по имени субъекта определить его бабушку и дедушку по материнской линии (предки 2-го колена).

grands("Nikita", MotherMother, MotherFather, \_, \_).



grands("Nikita", MotherMother, \_, \_, \_).

|  |  |  |  |
| --- | --- | --- | --- |
| № 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?) |
| 0 | Резольвента:  grands("Nikita", MotherMother, \_, \_, \_).  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Резольвента не пуста система запускает алгоритм унификации. | grands("Nikita", MotherMother, \_, \_, \_).  =parent("Irina", "woman", "Nikita").  Унификация не успешна, функторы не совпадают. | Переход к следующему предложению |
| … | … | … | … |
| 6 | Резольвента:  grands("Nikita", MotherMother, \_, \_, \_). | grands("Nikita", MotherMother, \_, \_, \_).  =grands(Name, MotherMother, MotherFather, FatherMother, FatherFather).  Унификация успешна  Θ = {"Nikita" = Name, MotherMother = MotherMother } | Прямой ход.  1.Преобразование резольвенты с помощью редукции. Замена цели телом того правила заголовок, которого унифицируется с целью.  2. Применяется подстановка. |
| 8 | Резольвента:  parent(Mother,"woman", "Nikita"),  parent(MotherMother,"woman", Mother),  parent(MotherFather, "man", Mother),  parent(Father,"man", "Nikita"),  parent(FatherMother,"woman", Father),  parent(FatherFather,"man", Father). | parent(Mother, "woman", "Nikita")=parent("Irina", "woman", "Nikita")  Унификация успешна.  Подстановка:  Θ = { Mother = "Irina", "woman" = "woman", "Nikita" = "Nikita"} | Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции. |
| 9 | Резольвента:  parent(MotherMother,"woman", "Irina"),  parent(MotherFather, "man", "Irina"),  parent(Father,"man", "Nikita"),  parent(FatherMother,"woman", Father),  parent(FatherFather,"man", Father). | parent(MotherMother, "woman", "Irina")=parent("Irina", "woman", "Nikita")  Унификация не успешна.  "Irina" != "Nikita" | Переход к следующему предложению |
| 10 | Резольвента:  parent(MotherMother,"woman", "Irina"),  parent(MotherFather, "man", "Irina"),  parent(Father,"man", "Nikita"),  parent(FatherMother,"woman", Father),  parent(FatherFather,"man", Father). | parent(MotherMother, "woman", "Irina")=parent("Raya", "woman", "Irina")  Унификация успешна.  Подстановка:  Θ = { MotherMother = "Raya", "woman" = "woman", "Irina" = "Irina"} | Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции. |
| 11 | Резольвента:  parent(MotherFather, "man", "Irina"),  parent(Father,"man", "Nikita"),  parent(FatherMother,"woman", Father),  parent(FatherFather,"man", Father). | parent(MotherFather, "man", "Irina") = parent("Irina", "woman", "Nikita")    Унификация не успешна "man" != "woman" | Переход к следующему предложению |
| … | … | … | … |
| 13 | Резольвента:  parent(MotherFather, "man", "Irina"),  parent(Father,"man", "Nikita"),  parent(FatherMother,"woman", Father),  parent(FatherFather,"man", Father). | parent(MotherFather, "man", "Irina") = parent("Alex", "man", "Nikita")    Унификация не успешна "Irina" != "Nikita" | Переход к следующему предложению |
| … | … | … | … |
| 15 | Резольвента:  parent(MotherFather, "man", "Irina"),  parent(Father,"man", "Nikita"),  parent(FatherMother,"woman", Father),  parent(FatherFather,"man", Father). | parent(MotherFather, "man", "Irina") = parent("Vitaliy", "man", "Irina")    Унификация успешна.  Подстановка:  Θ = { MotherFather = "Vitaliy", "man" = "man", "Irina" = "Irina"} | Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции. |
| 16 | Резольвента:  parent(Father,"man", "Nikita"),  parent(FatherMother,"woman", Father),  parent(FatherFather,"man", Father). | parent(Father,"man","Nikita")= parent("Alex","man", "Nikita")  Унификация успешна Подстановка:  { Father = "Alex", "man" = "man",  "Nikita" = "Nikita"} | Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции. |
| 17 | Резольвента:  parent(FatherMother,"woman", "Alex"),  parent(FatherFather,"man", "Alex"). | parent(FatherMother,"woman", "Alex")=parent("Irina", "woman", "Nikita")  Унификация не успешна "Alex" != "Nikita" | Переход к следующему предложению |
| … | … | … | … |
| 19 | Резольвента:  parent(FatherMother,"woman", "Alex"),  parent(FatherFather,"man", "Alex"). | parent(FatherMother,"woman", "Alex")=parent("Vera", "woman", "Alex")  Унификация успешна Подстановка:  { FatherMother = "Vera", "woman" = "woman","Alex" = "Alex"} | Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции. |
| 20 | Резольвента:  parent(FatherFather,"man", "Alex"). | parent(FatherMother,"man","Alex")=parent("Irina", "woman", "Nikita")  Унификация не успешна "man" != "woman" | Переход к следующему предложению |
| … | … | … | … |
| 23 | Резольвента:  parent(FatherFather,"man", "Alex"). | parent(FatherMother,"man","Alex")=parent("Alex", "man", "Nikita")  Унификация не успешна "Alex" != "Nikita" | Переход к следующему предложению |
| 24 | Резольвента:  parent(FatherFather,"man", "Alex"). | parent(FatherMother,"man","Alex")=parent("Anatoly", "man", "Alex")  Унификация успешна Подстановка:  { FatherMother = "Anatoly", "man" = "man","Alex" = "Alex"} | Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции. |
| 25 | Резольвента пуста.  Все переменные связаны |  | Вывод результата. Поиск альтернативного решения. Откат к предыдущему состоянию резольвенты. |
| 26 | Возврат предыдущего состояния резольвенты.  Резольвента:  parent(FatherFather,"man", "Alex"). | parent(FatherMother,"man","Alex")=parent("Vitaliy", "man", "Irina")  Унификация не успешна "Alex" != "Irina" | Переход к следующему предложению |
| 27 | Резольвента:  parent(FatherFather,"man", "Alex"). | parent(FatherMother,"man","Alex")=grands(Name,MotherMother, MotherFather,FatherMother, FatherFather)  Унификация не успешна. Разные функторы. | Откат к предыдущему состоянию резольвенты |
| 28-32 | Возврат предыдущего состояния резольвенты.  Резольвента:  parent(FatherMother,"woman", "Alex"),  parent(FatherFather,"man", "Alex"). | Происходят унификации подцели из резольвенты и еще неотмеченных фактов из базы знаний. Все унификации завершаются неудачей | Откат к предыдущему состоянию резольвенты |
| 32-35 | Возврат предыдущего состояния резольвенты.  parent(Father,"man", "Nikita"),  parent(FatherMother,"woman", Father),  parent(FatherFather,"man", Father). | Происходят унификации подцели из резольвенты и еще неотмеченных фактов из базы знаний. Все унификации завершаются неудачей | Откат к предыдущему состоянию резольвенты |
| 36 | Возврат предыдущего состояния резольвенты.  Резольвента:  parent(MotherFather, "man", "Irina"),  parent(Father,"man", "Nikita"),  parent(FatherMother,"woman", Father),  parent(FatherFather,"man", Father). | Происходят унификации подцели из резольвенты и еще неотмеченных фактов из базы знаний. Все унификации завершаются неудачей | Откат к предыдущему состоянию резольвенты |
| 37-42 | Возврат предыдущего состояния резольвенты.  Резольвента:  parent(MotherMother,"woman", "Irina"),  parent(MotherFather, "man", "Irina"),  parent(Father,"man", "Nikita"),  parent(FatherMother,"woman", Father),  parent(FatherFather,"man", Father). | Происходят унификации подцели из резольвенты и еще неотмеченных фактов из базы знаний. Все унификации завершаются неудачей | Откат к предыдущему состоянию резольвенты |
| 43-49 | Возврат предыдущего состояния резольвенты.  Резольвента:  parent(Mother,"woman", "Nikita"),  parent(MotherMother,"woman", "Irina"),  parent(MotherFather, "man", "Irina"),  parent(Father,"man", "Nikita"),  parent(FatherMother,"woman", Father),  parent(FatherFather,"man", Father). | Происходят унификации подцели из резольвенты и еще неотмеченных фактов из базы знаний. Все унификации завершаются неудачей | Откат к предыдущему состоянию резольвенты |
| 50 | Резольвента:  grands("Nikita", MotherMother, \_, \_, \_).  Пуста. | … | Завершение выполнения программы. |

**Ответы на вопросы:**

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

Резольвента не пуста, система запускает алгоритм унификации.

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

Унификация позволяет формализовать процесс логического вывода. Назначение - поиск знания, которое является ответом на конкретный вопрос. Результат унификации — ответ «да» или «нет».

1. **Какое первое состояние резольвенты?**

Первое состояние резольвенты - заданный вопрос.

1. **Как меняется резольвента?**

В процессе доказательства утверждений в резольвента меняется. При изменении строится новая резольвента. По стековому принципу берется верхняя подцель и заменяется на тело подходящего правила. Затем применяется найденная на текущем этапе подстановка. Успешное завершение работы программы достигается, когда резольвента пуста.

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

Именованные переменные уникальны в рамках одного предложения, т. е. в разных предложениях одно и то же имя переменной может использоваться для обозначения разных объектов. Анонимные переменные уникальны везде.

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

Если алгоритм унификации завершился успешно и найдена подстановка, соответствующие переменные конкретизируются полученными значениями.

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

Механизм отката к предыдущему шагу выполняется в случае, когда унификация завершается тупиковой ситуацией(неудачей). Кроме того, механизм используется для того, чтобы получить все возможные ответы.