|  |  |
| --- | --- |
|  | **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»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ДИСЦИПЛИНА «Функциональное и логическое программирование»

**Лабораторная работа № 16**

**“Формирование эффективных программ на Prolog”**

|  |  |
| --- | --- |
| **Студент**  Чалый А. А.  **Группа** ИУ7 – 62 Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Толпинская Н.Б. |  |

Москва.

2021 г.

**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 вопрпосов. Использовать **конъюнктивные правила и простой вопрос.**

Листинг 1. Программа relatives .pro

domains

name, sex = symbol

predicates

parent(name, sex, name).

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

clauses

parent("Olga", "woman", "Andrey"). % Andrey`s mother is Olga

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

parent("Tamara","woman", "Alexandr"). % Alexandr`s mother is Tamara

parent("Alexandr", "man", "Andrey"). % Andrey`s father is Alexandr

parent("Anatoly", "man", "Alexandr"). % Alexandr`s father is Anatoly

parent("Grigory", "man", "Olga"). % Olga`s father is Grigory

grands(Name, MotherGrandMa, MotherGrandPa, FatherGrandMa, FatherGrandPa):-

parent(Mother, "woman", Name),

parent(MotherGrandMa, "woman", Mother),

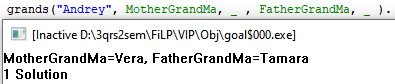
parent(MotherGrandPa, "man", Mother),

parent(Father,"man", Name),

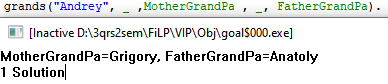
parent(FatherGrandMa,"woman", Father),

parent(FatherGrandPa,"man", Father).

1. по имени субъекта определить всех его бабушек (предки 2-го колена), grands("Andrey", MotherGrandMa, \_ , FatherGrandMa, \_ ).

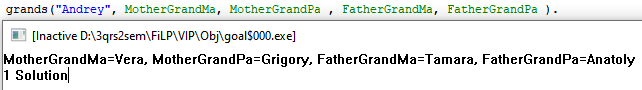


1. по имени субъекта определить всех его дедушек (предки 2-го колена), grands("Andrey ", \_ ,MotherGrandPa , \_, FatherGrandPa).



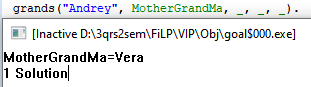
1. по имени субъекта определить всех его бабушек и дедушек (предки 2-го колена),

grands("Andrey ", MotherGrandMa, MotherGrandPa , FatherGrandMa, FatherGrandPa ).



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

grands("Andrey ", MotherGrandMa, \_, \_, \_).



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

grands("Andrey ", MotherGrandMa, MotherGrandPa , \_, \_).

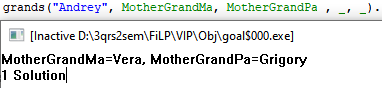


Таблица 1. grands("Andrey ", MotherGrandMa, \_, \_, \_).

|  |  |  |  |
| --- | --- | --- | --- |
| № 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("Andrey", MotherGrandMa, \_, \_, \_).  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Резольвента не пуста система запускает алгоритм унификации. | grands("Andrey", MotherGrandMa, \_, \_, \_).  =parent("Olga", "woman", "Andrey").  Унификация не успешна, функторы не совпадают. | Переход к следующему предложению |
| … | … | … | … |
| 6 | Резольвента:  grands("Andrey", MotherGrandMa, \_, \_, \_). | grands("Andrey", MotherGrandMa, \_, \_, \_).  =grands(Name, MotherGrandMa, MotherGrandPa, FatherGrandMa, FatherGrandPa).  Унификация успешна  Θ = {"Andrey" = Name, MotherGrandMa = MotherGrandMa } | 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", "Andrey"),  parent(MotherGrandMa,"woman", Mother),  parent(MotherGrandPa, "man", Mother),  parent(Father,"man", "Andrey"),  parent(FatherGrandMa,"woman", Father),  parent(FatherGrandPa,"man", Father). | parent(Mother, "woman", "Andrey")=parent("Olga", "woman", "Andrey")  Унификация успешна.  Подстановка:  Θ = { Mother = "Olga", "woman" = "woman", "Andrey" = "Andrey"} | Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции. |
| 9 | Резольвента:  parent(MotherGrandMa,"woman", "Olga"),  parent(MotherGrandPa, "man", "Olga"),  parent(Father,"man", "Andrey"),  parent(FatherGrandMa,"woman", Father),  parent(FatherGrandPa,"man", Father). | parent(MotherGrandMa, "woman", "Olga")=parent("Olga", "woman", "Andrey")  Унификация не успешна.  "Olga" != "Andrey" | Переход к следующему предложению |
| 10 | Резольвента:  parent(MotherGrandMa,"woman", "Olga"),  parent(MotherGrandPa, "man", "Olga"),  parent(Father,"man", "Andrey"),  parent(FatherGrandMa,"woman", Father),  parent(FatherGrandPa,"man", Father). | parent(MotherGrandMa, "woman", "Olga")=parent("Vera", "woman", "Olga")  Унификация успешна.  Подстановка:  Θ = { MotherGrandMa = "Vera", "woman" = "woman", "Olga" = "Olga"} | Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции. |
| 11 | Резольвента:  parent(MotherGrandPa, "man", "Olga"),  parent(Father,"man", "Andrey"),  parent(FatherGrandMa,"woman", Father),  parent(FatherGrandPa,"man", Father). | parent(MotherGrandPa, "man", "Olga") = parent("Olga", "woman", "Andrey")    Унификация не успешна "man" != "woman" | Переход к следующему предложению |
| … | … | … | … |
| 13 | Резольвента:  parent(MotherGrandPa, "man", "Olga"),  parent(Father,"man", "Andrey"),  parent(FatherGrandMa,"woman", Father),  parent(FatherGrandPa,"man", Father). | parent(MotherGrandPa, "man", "Olga") = parent("Alexandr", "man", "Andrey")    Унификация не успешна "Olga" != "Andrey" | Переход к следующему предложению |
| … | … | … | … |
| 15 | Резольвента:  parent(MotherGrandPa, "man", "Olga"),  parent(Father,"man", "Andrey"),  parent(FatherGrandMa,"woman", Father),  parent(FatherGrandPa,"man", Father). | parent(MotherGrandPa, "man", "Olga") = parent("Anatoly", "man", "Olga")    Унификация успешна.  Подстановка:  Θ = { MotherGrandPa = "Anatoly", "man" = "man", "Olga" = "Olga"} | Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции. |
| 16 | Резольвента:  parent(Father,"man", "Andrey"),  parent(FatherGrandMa,"woman", Father),  parent(FatherGrandPa,"man", Father). | parent(Father,"man","Andrey")= parent("Alexandr","man", "Andrey")  Унификация успешна Подстановка:  { Father = "Alexandr", "man" = "man",  "Andrey" = "Andrey"} | Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции. |
| 17 | Резольвента:  parent(FatherGrandMa,"woman", "Alexandr"),  parent(FatherGrandPa,"man", "Alexandr"). | parent(FatherGrandMa,"woman", "Alexandr")=parent("Olga", "woman", "Andrey")  Унификация не успешна "Alexandr" != "Andrey" | Переход к следующему предложению |
| … | … | … | … |
| 19 | Резольвента:  parent(FatherGrandMa,"woman", "Alexandr"),  parent(FatherGrandPa,"man", "Alexandr"). | parent(FatherGrandMa,"woman", "Alexandr")=parent("Tamara", "woman", "Alexandr")  Унификация успешна Подстановка:  { FatherGrandMa = "Tamara", "woman" = "woman","Alexandr" = "Alexandr"} | Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции. |
| 20 | Резольвента:  parent(FatherGrandPa,"man", "Alexandr"). | parent(FatherGrandMa,"man","Alexandr")=parent("Olga", "woman", "Andrey")  Унификация не успешна "man" != "woman" | Переход к следующему предложению |
| … | … | … | … |
| 23 | Резольвента:  parent(FatherGrandPa,"man", "Alexandr"). | parent(FatherGrandMa,"man","Alexandr")=parent("Alexandr", "man", "Andrey")  Унификация не успешна "Alexandr" != "Andrey" | Переход к следующему предложению |
| 24 | Резольвента:  parent(FatherGrandPa,"man", "Alexandr"). | parent(FatherGrandMa,"man","Alexandr")=parent("Anatoly", "man", "Alexandr")  Унификация успешна Подстановка:  { FatherGrandMa = "Anatoly", "man" = "man","Alexandr" = "Alexandr"} | 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(FatherGrandPa,"man", "Alexandr"). | parent(FatherGrandMa,"man","Alexandr")=parent("Grigory", "man", "Olga")  Унификация не успешна "Alexandr" != "Olga" | Переход к следующему предложению |
| 27 | Резольвента:  parent(FatherGrandPa,"man", "Alexandr"). | parent(FatherGrandMa,"man","Alexandr")=grands(Name,MotherGrandMa, MotherGrandPa,FatherGrandMa, FatherGrandPa)  Унификация не успешна. Разные функторы. | Откат к предыдущему состоянию резольвенты |
| 28-32 | Возврат предыдущего состояния резольвенты.  Резольвента:  parent(FatherGrandMa,"woman", "Alexandr"),  parent(FatherGrandPa,"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", "Andrey"),  parent(FatherGrandMa,"woman", Father),  parent(FatherGrandPa,"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(MotherGrandPa, "man", "Olga"),  parent(Father,"man", "Andrey"),  parent(FatherGrandMa,"woman", Father),  parent(FatherGrandPa,"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(MotherGrandMa,"woman", "Olga"),  parent(MotherGrandPa, "man", "Olga"),  parent(Father,"man", "Andrey"),  parent(FatherGrandMa,"woman", Father),  parent(FatherGrandPa,"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", "Andrey"),  parent(MotherGrandMa,"woman", "Olga"),  parent(MotherGrandPa, "man", "Olga"),  parent(Father,"man", "Andrey"),  parent(FatherGrandMa,"woman", Father),  parent(FatherGrandPa,"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("Andrey", MotherGrandMa, \_, \_, \_).  Пуста. | … | Завершение выполнения программы. |

**Контрольные 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.