|  |  |
| --- | --- |
| 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» (ИУ7)**

Направление подготовки: 09.03.04 **Программная инженерия**

**Отчет**

|  |
| --- |
| **по лабораторным работам № 16** |

**Название:** Использование правил в программе на Prolog

**Дисциплина:** Функциональное и логическое программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-62Б |  |  | Д.М.Блохин |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Н.Б.Толпинская |
|  |  |  |  | Ю.В.Строганов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2021

# Задание

**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.**

**Для одного** из вариантов **ВОПРОСА** и конкретной БЗ **составить таблицу**, отражающую конкретный порядок работы системы, с объяснениями:

очередная проблема на каждом шаге и метод ее решения;

каково новое текущее состояние резольвенты, как получено;

какие дальнейшие действия? (Запускается ли алгоритм унификации? Каких термов? Почему этих?) ;

вывод по результатам очередного шага и дальнейшие действия.

Т.к. резольвента хранится в виде стека, то состояние резольвенты требуется отображать в столбик: вершина – сверху! Новый шаг надо начинать с нового состояния резольвенты!

# Вопросы

**В каком случае система запускает алгоритм унификации?**

Система запускает алгоритм унификации автоматически при необходимости что-то доказать, когда цель попадает в резольвенту.

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

Унификация – основной шаг доказательства в системе. Задача – найти нужное знание. Результат – подстановка.

**Какое первое состояние резольвенты?**

Заданный вопрос (goal).

**Как меняется резольвента?**

Преобразования резольвенты выполняются с помощью редукции.

Редукцией цели G с помощью программы P называется замена цели G телом того правила из P, заголовок которого унифицируется с целью.

Новая резольвента образуется в два этапа:

1. в текущей резольвенте выбирается одна из подцелей и для неё выполняется редукция
2. к полученной конъюнкции целей применяется подстановка, полученная как наибольший общий унификатор цели и заголовка сопоставленного с ней правила.

**В каких пределах программы уникальны переменные?**

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

**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?**

Механизм отката запускается в случае неудачи алгоритма унификации и в случае, когда резольвента пустеет, а система не дошла до конца базы знаний.

# Листинг

**domains**

name = symbol

**predicates**

mother(name, name).

father(name, name).

grandfatherF(name, name).

grandfatherM(name, name).

grandmotherF(name, name).

grandmotherM(name, name).

allGrandmothers(name, name).

allGrandfathers(name, name).

allGrandparents(name, name).

grandparentsM(name, name).

**clauses**

mother( maria, ann ).

mother( maria, yulia ).

mother( irina, dima ).

mother( ann, petr ).

father ( ivan , ann ).

father ( ivan , yulia ).

father (maksim , dima ).

father ( dima , petr ).

grandfatherM(X, Y):- mother(Z, Y), father(X, Z).

grandfatherF(X, Y):- father(Z, Y), father(X, Z).

grandmotherM(X, Y):- mother(Z ,Y), mother(X, Z).

grandmotherF(X, Y):- father(Z ,Y), mother(X, Z).

%parent(X, Y, mom) :- mother(X, Y).

%parent(X, Y, dad) :- father(X, Y).

%grandfather((X, Y, Line) :- father(X, Z), parent(Z, Y, Line).

%grandmother((X, Y, Line) :- mother(X, Z), parent(Z, Y, Line).

allGrandmothers(X, Y):- grandmotherM(X, Y).

allGrandmothers(X, Y):- grandmotherF(X, Y).

%grandParents(X, Y, Line) :- grandfather(X, Y, Line).

%grandParents(X, Y, Line) :- grandmother(X, Y, Line).

allGrandfathers(X, Y):- grandfatherM(X, Y).

allGrandfathers(X, Y):- grandfatherF(X, Y).

allGrandparents(X, Y):- allGrandfathers(X, Y).

allGrandparents(X, Y):- allGrandmothers(X, Y).

grandparentsM(X, Y):- grandfatherM (X, Y).

grandparentsM(X, Y):- grandmotherM (X, Y).

**goal**

allGrandMothers(X, petr).

%allGrandFathers(X, petr).

%allGrandParents(X, petr).

%grandParentsM(X, petr).

# Таблица

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Состояние резольвенты, и вывод: дальнейшие действия | Для каких термов запускается алгоритм унификации: Т1=Т2 и каков **результат** | Дальнейшие действия: прямой ход или откат |
|  | allGrandMothers(X, petr). | allGrandMothers(X, petr).= mother( maria, ann ).  Неудача (разные функторы) | Переход к следующему предложению |
|  | … | … | … |
|  | allGrandMothers(X, petr). | allGrandMothers(X, petr).= allGrandmothers(X, Y)  Удача  {X = X’, Y = petr} | Прямой ход.  Тело правила заносится в резольвенту. |
|  | grandmotherM(X’, petr). | grandmotherM(X’, petr).= mother( maria, ann ).  Неудача (разные функторы) | Переход к следующему предложению |
|  | … | … | … |
|  | grandmotherM(X’, petr). | grandmotherM(X’, petr).= grandmotherM(X, Y)  Удача  {X = X’, Y = petr,  X’’ = X’} | Прямой ход.  Тело правила заносится в резольвенту. |
|  | mother(Z , petr),  mother(X’’, Z) | mother(Z ,petr)= mother( maria, ann ).  Неудача (petr <> ann) | Переход к следующему предложению |
|  | mother(Z , petr),  mother(X’’, Z) | mother(Z , petr) = mother( ann, petr ). Удача  {X = X’, Y = petr,  X’ = X’’, Z = ann} | Прямой ход. |
|  | mother(X’’, ann) | mother(X’’, ann) = mother( maria, ann ).  Удача  {X = X’, Y = petr,  X’’ = X’, Z = ann, X’’ = maria}   * X’ = X’’ = maria * X = X’ = maria | Резольвента пуста.  Полученная подстановка:  X = maria  Система не дошла до конца БЗ, происходит откат резольвенты к предыдущему состоянию. (п. 9) |
|  | … | … | … |
|  | mother(X’’, ann) | mother(X’’, ann) = grandparentsM(X, Y)  Неудача (разные функторы) | Система дошла до конца БЗ, откат к п.8 |
|  | … mother(Z , petr),  mother(X’’, Z) | … mother(Z , petr) = grandparentsM(X, Y)  Неудача (разные функторы) | … Система дошла до конца БЗ, откат к п.6 |
|  | … grandmotherM(X’, petr). | … grandmotherM(X’, petr). = grandparentsM(X, Y)  Неудача (разные функторы) | … Система дошла до конца БЗ, откат к п.3 |
|  | allGrandMothers(X, petr). | allGrandMothers(X, petr). =  allGrandmothers(X, Y)  Удача  {X = X’, Y = petr} | Прямой ход.  Тело правила заносится в резольвенту. |
|  | grandmotherF(X’, petr) | grandmotherF(X’, petr) =  mother(name, name).  Неудача (разные функторы) | Переход к следующему предложению |
|  | … | … | … |
|  | grandmotherF(X’, petr) | grandmotherF(X’, petr) =  grandmotherF(X, Y)  Удача  {X = X’, Y = petr,  X’ = X’’} | Прямой ход.  Тело правила заносится в резольвенту. |
|  | father(Z ,petr),  mother(X’’, Z) | father(Z ,petr) = mother( maria, ann ).  Неудача (разные функторы) | Переход к следующему предложению |
|  | … | … | … |
|  | father(Z ,petr),  mother(X’’, Z) | father(Z ,petr) = father ( dima , petr ).  Удача  {X = X’, Y = petr,  X’ = X’’, Z = dima} | Прямой ход. |
|  | mother(X’’, dima) | … mother(X’’, dima) = mother( irina, dima ).  Удача  {X = X’, Y = petr,  X’ = X’’, Z = dima¸ X’’ = irina}   * X’ = X’’ = irina * X = X’ = irina | Резольвента пуста.  Полученная подстановка:  X = irina  Система не дошла до конца БЗ, происходит откат резольвенты к предыдущему состоянию. (п. 21) |
|  | mother(X’’, dima) | … mother(X’’, dima) = grandparentsM(X, Y)  Неудача (разные функторы) | … Система дошла до конца БЗ, откат к п.20 |
|  | father(Z ,petr),  mother(X’’, Z) | … father(Z ,petr) = grandparentsM(X, Y)  Неудача (разные функторы) | … Система дошла до конца БЗ, откат к п.17 |
|  | grandmotherF(X’, petr) | …grandmotherF(X’, petr) =  grandparentsM(X, Y)  Неудача (разные функторы) | … Система дошла до конца БЗ, откат к п.14 |
|  | allGrandMothers(X, petr). | … allGrandMothers(X, petr). =  grandparentsM(X, Y)  Неудача (разные функторы) | … Система дошла до конца БЗ, доказать цель в резольвенте невозможно! Выход из программы |