



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

*к лабораторной работе №16*

*По курсу: «Функциональное и логическое  
программирование»*

Студент ИУ7-64Б  
Лозовский А.А.

Преподаватель  
Толпинская Н.Б

*Москва, 2020 г.*

## 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:

1. очередная проблема на каждом шаге и метод ее решения;
2. каково новое текущее состояние резольвенты, как получено;
3. какие дальнейшие действия? (Запускается ли алгоритм унификации? Каких термов? Почему этих?);
4. вывод по результатам очередного шага и дальнейшие действия.

## Ответы на вопросы

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

Алгоритм унификации запускается автоматически, если есть что доказывать, т.е. если задана цель доказательства.

### 2. Каковы назначение и результат использования алгоритма унификации?

Алгоритм унификации формализует процесс логического вывода. Используется для нахождения «нужного» для доказательства истинности (выводимости) правила в данный момент времени.

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

### 3. Какое первое состояние резольвенты?

На первом шаге в резольвенте находится заданный вопрос (цель).

### 4. Как меняется резольвента?

Для хранения резольвенты используется стек. Меняется она в ходе доказательства. Преобразования выполняются с помощью редукции – замены текущей цели на тело найденного в программе правила (с помощью унификации цели и заголовка правила программы).

Преобразование резольвенты разделено на два этапа:

- 1) Берется верхняя из подцелей резольвенты (по стековому принципу) 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).

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

Именованная переменная уникальна в рамках предложения, в котором она используется. Любая анонимная переменная является уникальной.

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

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

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

Во время работы системы, в случае, если решение не найдено, и из данного состояния невозможен переход в новое состояние (тупиковое состояние), применяется механизм отката.

**predicates**

grandparents(symbol Flag, symbol GrandMother, symbol GrandFather, symbol Child)  
parents(symbol Mother, symbol Father, symbol Child)

**clauses**

*/\* 4th generation \*/*

parents("Nancy", "Tom", "Kolya").  
parents("Nancy", "Tom", "Patricia").  
parents(" \_ ", "Tom", "Will").

*/\* 3rd generation \*/*

parents("Julia", "Howard", "Katty").  
parents("Julia", "Howard", "Josh").  
parents(" \_ ", "Andrew", "Nadya").

*/\* 2nd generation \*/*

parents("Nadya", "Kolya", "Dima").  
parents("Katty", "Will", "Sasha").  
parents("Nadya", "Josh", "Newt").

*/\* 1st generation \*/*

parents("Sasha", "Newt", "Bella").  
parents("Sasha", "Dima", "Vadim").

*/\* mother's line grandparents\*/*

grandparents("Mother's", GrandMa, GrandPa, Child):- parents(Mother, \_, Child),  
parents(GrandMa, GrandPa, Mother).

*/\* father's line grandparents\*/*

grandparents("Father's", GrandMa, GrandPa, Child):- parents(\_, Father, Child),  
parents(GrandMa, GrandPa, Father).

## Примеры целей и результатов работы программы

**Задание:** по имени субъекта определить всех его бабушек (предки 2-го колена)

1. **Goal** grandparents(\_, GrandMa, \_, "Vadim").

**Result** GrandMa=Katty  
GrandMa=Nadya

**Задание** по имени субъекта определить всех его дедушек (предки 2-го колена)

2. **Goal** grandparents(\_, \_, GrandPa, "Vadim").

**Result** GrandPa=Will  
GrandPa=Kolya

**Задание** по имени субъекта определить всех его бабушек и дедушек (предки 2-го колена)

3. **Goal** grandparents(\_, GrandMa, GrandPa, "Bella").

**Result** GrandMa=Katty, GrandPa=Will  
GrandMa=Nadya, GrandPa=Josh

**Задание** по имени субъекта определить его бабушку по материнской линии (предки 2-го колена)

4. **Goal** grandparents("Mother's", GrandMa, \_, "Vadim")

**Result** GrandMa=Katty

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

5. **Goal** grandparents("Mother's", GrandMa, \_, "Vadim").

**Result** GrandPa=Will

## Описание порядка работы системы

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

1. очередная проблема на каждом шаге и метод ее решения;
2. каково новое текущее состояние резольвенты, как получено;
3. какие дальнейшие действия? (Запускается ли алгоритм унификации? Каких термов? Почему этих?);
4. вывод по результатам очередного шага и дальнейшие действия.

**Цель:** grandparents("Mother's", GrandMa, \_, "Vadim")

№ шага	Состояние резольвенты, и вывод: дальнейшие действия (почему?)	Для каких термов запускается алгоритм унификации: T1=T2 и каков <b>результат</b> (и подстановка)	Дальнейшие действия: прямой ход или откат (почему и к чему приводит?)
1	Цель заносится в резольвенту. grandparents("Mother's", GrandMa, _, "Vadim").	Попытка унификации: Grandparents ("Mother's", GrandMa, _, "Vadim")	Прямой ход, переход к следующему предложению.

	Запуск процесса редукции.	= parents("Nancy", "Tom", "Kolya"). Результат: неудача, разные главные функторы.	
...	grandparents("Mother's", GrandMa, _, "Vadim"). Выполнение процесса редукции.	Попытка унификации: Grandparents ("Mother's", GrandMa, _, "Vadim") И термов БЗ, пока не будет найден подходящий терм или не будет «просмотрена» вся БЗ.	Прямой ход (резольвента не пуста, не конец БЗ), переход к следующему предложению.
12	grandparents("Mother's", GrandMa, _, "Vadim"). Выполнение процесса редукции	Попытка унификации: Grandparents ("Mother's", GrandMa, _, "Vadim") = grandparents("Mother's", GrandMa, GrandPa, Child) результат: успех подстановка: { GrandMa = GrandMa , Child ="Vadim" }	(возможно, установка точки возврата), Преобразование резольвенты (замена текущей цели на тело найденного с помощью унификации правила, применение полученной подстановки), Прямой ход
13	parents(Mother, _, "Vadim"), parents(GrandMa, GrandPa, Mother). Выбор верхней подцели резольвенты, запуск редукции.	Попытка унификации: parents(Mother, _, "Vadim") = parents("Nancy", "Tom", "Kolya"). Результат: неудача, несовпадающие константы.	Прямой ход, переход к следующему предложению.
...	parents(Mother, _, "Vadim"), parents(GrandMa, GrandPa, Mother). Выполнение процесса редукции	Попытка унификации: parents(Mother, _, "Vadim") и термов, пока не будет найдено подходящее правило или не закончится БЗ.	Прямой ход, переход к следующему предложению.
24	parents(Mother, _, "Vadim"), parents(GrandMa, GrandPa, Mother). Выполнение процесса редукции	Попытка унификации: parents(Mother, _, "Vadim") = parents("Sasha", "Dima", "Vadim") результат: успех подстановка: {Mother="Sasha"}	(Возможно, установка точки возврата), Преобразование резольвенты (замена текущей цели на <u>пустое</u> тело найденного с помощью унификации факта, применение полученной подстановки), Прямой ход
...	parents(GrandMa, GrandPa, "Sasha"). Выбор верхней подцели резольвенты, запуск редукции.	Попытка унификации: parents(GrandMa, GrandPa, "Sasha") и термов, пока не будет найдено подходящее правило или не закончится БЗ.	Прямой ход, переход к следующему предложению.
32	parents(GrandMa, GrandPa, "Sasha"). Выполнение процесса редукции	Попытка унификации: parents(GrandMa, GrandPa, "Sasha") parents("Katty", "Will", "Sasha"). Результат: успех Подстановка: { GrandMa="Katty", GrandPa="Will" }.	Преобразование резольвенты (замена текущей цели на <u>пустое</u> тело найденного с помощью унификации факта, применение полученной подстановки)
33	<b>Резольвента пуста.</b>		Вывод запуск отката
34	parents(GrandMa, GrandPa, "Sasha").	Попытка унификации: parents(GrandMa, GrandPa, "Sasha") =	Прямой ход, переход к следующему предложению.

	Выбор верхней подцели резольвенты, запуск редукции	parents("Nadya", "Josh", "Newt") Результат: неудача, несовпадающие константы	
...	parents(GrandMa, GrandPa, "Sasha"). Выполнение редукции.	Попытка унификации: parents(GrandMa, GrandPa, "Sasha") и термов, пока не будет найдено подходящее правило или не закончится БЗ.	Прямой ход, переход к следующему предложению
38	parents(GrandMa, GrandPa, "Sasha"). Выполнение редукции.	Попытка унификации: parents(GrandMa, GrandPa, "Sasha") = grandparents("Father's", GrandMa, GrandPa, Child) Результат: неудача, разные главные функторы.	Конец БЗ, резольвента не пустая, откат.
39	parents(Mother, _, "Vadim"), parents(GrandMa, GrandPa, Mother). Выбор верхней подцели резольвенты, запуск редукции	Попытка унификации: parents(Mother, _, "Vadim") = grandparents("Mother's", GrandMa, GrandPa, Child) Результат: неудача, разные главные функторы.	Прямой ход, переход к следующему предложению
40	parents(Mother, _, "Vadim"), parents(GrandMa, GrandPa, Mother). Выполнение редукции	Попытка унификации: parents(Mother, _, "Vadim") = grandparents("Father's", GrandMa, GrandPa, Child) Результат: неудача, разные главные функторы.	Конец БЗ, резольвента не пустая, откат.
41	grandparents("Mother's", GrandMa, _, "Vadim"). Запуск процесса редукции.	Попытка унификации: Grandparents ("Mother's", GrandMa, _, "Vadim") = grandparents("Father's", GrandMa, GrandPa, Child) Результат: неудача, разные константы.	Конец БЗ. Конец работы программы.