



**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**

## **О т ч е т**

**по лабораторной работе № 16**

**Дисциплина: «Функциональное и логическое программирование»**

Выполнила: Овчинникова А. П.

Группа: ИУ7-65Б

Преподаватель: Толпинская Н. Б.

Строганов Ю. В.

Москва, 2020

## Задание

Ответить на вопросы.

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!

## Теоретическая часть

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

Процесс унификации запускается автоматически, если есть что доказывать, то надо запускать алгоритм унификации. Пользователь имеет право запустить этот процесс вручную, с помощью утверждения  $T1 = T2$ , включенного в текст программы. Если резольвента не пуста – запускается алгоритм унификации (для хранения резольвенты используется стек, соответственно, если стек не пуст – запускается алгоритм унификации).

2. Каковы назначение и результат использования алгоритма унификации?

Назначение унификации – подобрать нужное в данный момент правило. Система подбирает сопоставимые с целью правила с помощью алгоритма унификации. В результате унификации формируются подстановки. Унификация запускается, если резольвента не пуста.

3. Какое первое состояние резольвенты?

Первое состояние резольвенты – заданный вопрос.

4. Как меняется резольвента?

Преобразования резольвенты выполняются с помощью редукции. Новая резольвента образуется в два этапа:

- В текущей резольвенте выбирается одна из подцелей (по стековому принципу – верхняя) и для нее выполняется редукция – замена подцели на тело найденного правила (если удалось найти правило).
- Затем к полученной конъюнкции целей применяется подстановка, полученная как наибольший общий унификатор выбранной цели и заголовка сопоставленного с ней правила.

Если для редукции цели из резольвенты был выбран факт из БЗ, то новая резольвента будет содержать в конъюнкции на одну цель меньше. Если задан

простой вопрос и подобран для редукции факт, то произойдет немедленное его согласование. А если для простого вопроса подобрано правило, то число целей в резольвенте не уменьшится, т. к. цель будет заменена телом подобранного правила.

5. В каких пределах программы уникальны переменные?

Именованные переменные уникальны в рамках предложения, а анонимная переменная – любая уникальна.

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

В случае успешного согласования программы (базы знаний) и вопроса, в качестве побочного эффекта формируется подстановка, которая содержит значения переменных, при которых вопрос является примером программы. Соответствующие переменные конкретизируются полученными значениями.

7. В каких случаях запускается механизм отката?

В ситуации, когда решение не найдено, и из данного состояния невозможен переход в новое состояние, автоматически включается бэктрекинг. Происходит возврат к моменту, где еще можно сделать другой альтернативный выбор, то есть к предыдущему состоянию резольвенты. Бэктрекинг возможен только при наличии альтернативных путей унификации цели.

### **Код программы**

predicates

mother(symbol, symbol). % mother name, child name

father(symbol, symbol). % father name, child name

ancestors(symbol, symbol, symbol, symbol, symbol). % subject, mom`s mom,  
mom`s dad, father`s mom, father`s dad

clauses

```
mother(olga, slava). % slava`s mother is olga
mother(valentina, oleg). %oleg`s mother is valentina
mother(margo, olga). % olga`s mom is margo
```

```
father(oleg, slava). % slava`s father is oleg
father(vitya, oleg). %oleg`s father is vitya
father(andrey, olga). % olga`s father is andrey
```

```
ancestors(NAME, MOMSMOM, MOMSFATHER, FATHERSMOM, FATHERSFATHER)
:-
```

```
    mother(MOM, NAME),
    mother(MOMSMOM, MOM),
    father(MOMSFATHER, MOM),
    father(DAD, NAME),
    mother(FATHERSMOM, DAD),
    father(FATHERSFATHER, DAD).
```

goal

```
%ancestors(slava, GRANDMA1, _, GRANDMA2, _). % all subject`s grandmas
%ancestors(slava, _, GRANDPA1, _, GRANDPA2). % all subject`s grandpas
%ancestors(slava, MOMSMOM, MOMSFATHER, FATHERSMOM,
FATHERSFATHER). % all subject`s grandpas and grandmas
%ancestors(slava, GRANDMA1, _, _, _). % subject`s mom`s mom
ancestors(slava, GRANDMA1, GRANDPA1, _, _). % subject`s mom`s mom and
mom`s dad
```

## Примеры работы программы

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

Вопрос: *ancestors(slava, GRANDMA1, \_, GRANDMA2, \_). % all subject's grandmas*

Ответ:

*GRANDMA1=margo, GRANDMA2=valentina*

*1 Solution*

2. По имени субъекта определить всех его дедушек (предки 2-го колена).

Вопрос: *ancestors(slava, \_, GRANDPA1, \_, GRANDPA2). % all subject's grandpas*

Ответ:

*GRANDPA1=andrey, GRANDPA2=vitya*

*1 Solution*

3. По имени субъекта определить всех его бабушек и дедушек (предки 2-го колена).

Вопрос: *ancestors(slava, MOMSMOM, MOMSFATHER, FATHERSMOM, FATHERSFATHER). % all subject's grandpas and grandmas*

Ответ:

*MOMSMOM = margo, MOMSFATHER = andrey, FATHERSMOM = valentina, FATHERSFATHER = vitya*

*1 Solution*

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

Вопрос: *ancestors(slava, GRANDMA1, \_, \_, \_). % subject's mom's mom*

Ответ:

*GRANDMA1=margo*

### 1 Solution

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

Вопрос: *ancestors(slava, GRANDMA1, GRANDPA1, \_, \_)*. % *subject`s mom`s mom and mom`s dad*

Ответ:

*GRANDMA1=margo, GRANDPA1=andrey*

### 1 Solution

Порядок работы системы для вопроса *ancestors(slava, GRANDMA1, \_, \_)*. .

№ 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	Резольвента: <i>ancestors(slava, GRANDMA1, _, _)</i> .	Попытка унификации: T1= <i>ancestors(slava, GRANDMA1, _, _)</i> . T2 = <i>mother(olga, slava)</i> . Результат: неудача. Разные функторы.	Откат, переход к следующему предложению
2	Резольвента: <i>ancestors(slava, GRANDMA1, _, _)</i> .	Попытка унификации: T1= <i>ancestors(slava, GRANDMA1, _, _)</i> . T2 = <i>mother(valentina, oleg)</i> . Результат: неудача. Разные функторы.	Откат, переход к следующему предложению
3	Резольвента: <i>ancestors(slava, GRANDMA1, _, _)</i> .	Попытка унификации: T1= <i>ancestors(slava, GRANDMA1, _, _)</i> . T2 = <i>mother(margo, olga)</i> . Результат: неудача. Разные функторы.	Откат, переход к следующему предложению
4	Резольвента: <i>ancestors(slava, GRANDMA1, _, _)</i> .	Попытка унификации: T1= <i>ancestors(slava, GRANDMA1, _, _)</i> . T2 = <i>father(oleg, slava)</i> . Результат: неудача. Разные функторы.	Откат, переход к следующему предложению
5	Резольвента: <i>ancestors(slava, GRANDMA1, _, _)</i> .	Попытка унификации: T1= <i>ancestors(slava, GRANDMA1, _, _)</i> . T2 = <i>father(vitya, oleg)</i> . Результат: неудача. Разные функторы.	Откат, переход к следующему предложению

6	Резольвента: ancestors(slava, GRANDMA1, _, _ _).	Попытка унификации: T1=ancestors(slava,GRANDMA1, _, _, _). T2 = father(andrey, olga). Результат: неудача. Разные функторы.	Откат, переход к следующему предложени ю
7	Резольвента: ancestors(slava, GRANDMA1, _, _ _).	Попытка унификации: T1=ancestors(slava,GRANDMA1, _, _, _). T2= ancestors(NAME, MOMSMOM, MOMSFATHER, FATHERSMOM, FATHERSFATHER) Результат: успех. Подстановка: {NAME=slava, MOMSMOM=GRANDMA1, MOMSFATHER=_, FATHERSMOM=_, FATHERSFATHER= }	Прямой ход. Содержимое резольвенты заменяется телом найденного правила. К резольвенте применяется подстановка.
8	Резольвента: mother(MOM, slava), mother(GRANDMA1, MOM), father(_, MOM), father(DAD, slava), mother(_, DAD), father(_, DAD).	Попытка унификации: T1= mother(MOM, slava) T2= mother(olga, slava). Результат: успех. Подстановка: {MOM=olga, slava=slava}	Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуетс я с помощью редукции.
9	Резольвента: mother(GRANDMA1, olga), father(_, olga), father(DAD, slava), mother(_, DAD), father(_, DAD).	Попытка унификации: T1= mother(GRANDMA1, olga) T2= mother(olga, slava). Результат: неудача, slava != olga	Откат, переход к следующему предложени ю.
10	Резольвента: mother(GRANDMA1, olga), father(_, olga), father(DAD, slava), mother(_, DAD), father( , DAD).	Попытка унификации: T1= mother(GRANDMA1, olga) T2= mother(valentina, oleg). Результат: неудача, oleg != olga	Откат, переход к следующему предложени ю.
11	Резольвента: mother(GRANDMA1, olga), father(_, olga), father(DAD, slava), mother(_, DAD), father(_, DAD).	Попытка унификации: T1= mother(GRANDMA1, olga) T2= mother(margo, olga). Результат: успех. Подстановка: {GRANDMA1=margo, olga=olga}	Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуетс я с помощью редукции.
12	Резольвента: father(_, olga), father(DAD, slava), mother(_, DAD), father(_, DAD).	Попытка унификации: T1= father(_, olga) T2= mother(olga, slava). Результат: неудача. Разные функторы.	Откат, переход к следующему предложени ю.
13	Резольвента:	Попытка унификации:	Откат,



	father(_, olga), father(DAD, slava), mother(_, DAD), father(_, DAD).	T1= father(_, olga) T2= mother(valentina, oleg). Результат: неудача. Разные функторы.	переход к следующему предложени ю.
14	Резольвента: father(_, olga), father(DAD, slava), mother(_, DAD), father(_, DAD).	Попытка унификации: T1= father(_, olga) T2= mother(margo, olga). Результат: неудача. Разные функторы.	Откат, переход к следующему предложени ю.
15	Резольвента: father(_, olga), father(DAD, slava), mother(_, DAD), father(_, DAD).	Попытка унификации: T1= father(_, olga) T2= father(oleg, slava). Результат: неудача, olga != slava	Откат, переход к следующему предложени ю.
16	Резольвента: father(_, olga), father(DAD, slava), mother(_, DAD), father(_, DAD).	Попытка унификации: T1= father(_, olga) T2= father(vitya, oleg). Результат: неудача, olga != oleg	Откат, переход к следующему предложени ю.
17	Резольвента: father(_, olga), father(DAD, slava), mother(_, DAD), father(_, DAD).	Попытка унификации: T1= father(_, olga) T2= father(andrey, olga). Результат: успех. Подстановка: {_=andrey, olga=olga}	Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуетс я с помощью редукции.
18	Резольвента: father(DAD, slava), mother(_, DAD), father(_, DAD).	Попытка унификации: T1= father(DAD, slava) T2= mother(olga, slava). Результат: неудача. Разные функторы.	Откат, переход к следующему предложени ю.
19	Резольвента: father(DAD, slava), mother(_, DAD), father(_, DAD).	Попытка унификации: T1= father(DAD, slava) T2= mother(valentina, oleg). Результат: неудача. Разные функторы.	Откат, переход к следующему предложени ю.
20	Резольвента: father(DAD, slava), mother(_, DAD), father(_, DAD).	Попытка унификации: T1= father(DAD, slava) T2= mother(margo, olga). Результат: неудача. Разные функторы.	Откат, переход к следующему предложени ю.
21	Резольвента: father(DAD, slava), mother(_, DAD), father(_, DAD).	Попытка унификации: T1= father(DAD, slava) T2= father(oleg, slava). Результат: успех. Подстановка: {DAD=oleg}	Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуетс я с помощью редукции.
22	Резольвента: mother(_, oleg), father(_, oleg).	Попытка унификации: T1= mother(_, oleg) T2= mother(olga, slava).	Откат, переход к следующему

		Результат: неудача, oleg != slava	предложени ю.
23	Резольвента: mother(_,oleg), father(_,oleg).	Попытка унификации: T1= mother(_,oleg) T2= mother(valentina, oleg). Результат: успех. Подстановка: {_=valentina}	Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуетс я с помощью редукции.
24	Резольвента: father(_,oleg).	Попытка унификации: T1= father(_,oleg). T2= mother(olga, slava). Результат: неудача. Разные функторы.	Откат, переход к следующему предложени ю.
25	Резольвента: father(_,oleg).	Попытка унификации: T1= father(_,oleg). T2= mother(valentina, oleg). Результат: неудача. Разные функторы.	Откат, переход к следующему предложени ю.
26	Резольвента: father(_,oleg).	Попытка унификации: T1= father(_,oleg). T2= mother(margo, olga). Результат: неудача. Разные функторы.	Откат, переход к следующему предложени ю.
27	Резольвента: father(_,oleg).	Попытка унификации: T1= father(_,oleg). T2= father(oleg, slava). Результат: неудача, oleg != slava	Откат, переход к следующему предложени ю.
28	Резольвента: father(_,oleg).	Попытка унификации: T1= father(_,oleg). T2= father(vitya, oleg). Результат: успех. Подстановка: {_=vitya, oleg=oleg}	Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуетс я с помощью редукции.
29	Резольвента пуста. Все переменные связаны.		Вывод результата. Поиск альтернатив ного решения. Откат к предыдущем у состоянию резольвенты.
30	Резольвента: father(_,oleg).	Попытки унификации подцели из резольвенты и еще неотмеченных фактов из БЗ: father(andrey, olga). ancestors(NAME, MOMSMOM, MOMSFATHER, FATHERSMOM,	Откат к предыдущем у состоянию резольвенты.

		FATHERSFATHER). Все попытки унификации завершаются неудачей: oleg != olga в первом случае и разные функторы во втором.	
31	Резольвента: mother(_, oleg), father(_, oleg).	Попытки унификации верхней подцели из резольвенты и еще неотмеченных фактов из БЗ: mother(margo, olga). father(oleg, slava). father(vitya, oleg). father(andrey, olga). ancestors(NAME, MOMSMOM, MOMSFATHER, FATHERSMOM, FATHERSFATHER) Все попытки унификации завершаются неудачей.	Откат к предыдущему состоянию резольвенты.
32	Резольвента: father(DAD, slava), mother(_, DAD), father(_, DAD).	Попытки унификации верхней подцели из резольвенты и еще неотмеченных фактов из БЗ: father(vitya, oleg). father(andrey, olga). ancestors(NAME, MOMSMOM, MOMSFATHER, FATHERSMOM, FATHERSFATHER) Все попытки унификации завершаются неудачей.	Откат к предыдущему состоянию резольвенты.
33	Резольвента: father(_, olga), father(DAD, slava), mother(_, DAD), father(_, DAD).	Попытки унификации верхней подцели из резольвенты и еще неотмеченных фактов из БЗ: ancestors(NAME, MOMSMOM, MOMSFATHER, FATHERSMOM, FATHERSFATHER) Все попытки унификации завершаются неудачей.	Откат к предыдущему состоянию резольвенты.
34	Резольвента: mother(GRANDMA1, olga), father(_, olga), father(DAD, slava), mother(_, DAD), father(_, DAD).	Попытки унификации верхней подцели из резольвенты и еще неотмеченных фактов из БЗ: father(oleg, slava). father(vitya, oleg). father(andrey, olga). ancestors(NAME, MOMSMOM, MOMSFATHER, FATHERSMOM, FATHERSFATHER) Все попытки унификации завершаются неудачей.	Откат к предыдущему состоянию резольвенты.
35	Резольвента: mother(MOM, slava), mother(GRANDMA1, MOM), father(_, MOM), father(DAD, slava), mother(_, DAD), father(_, DAD).	Попытки унификации верхней подцели из резольвенты и еще неотмеченных фактов из БЗ: mother(valentina, oleg). mother(margo, olga). father(oleg, slava). father(vitya, oleg).	Откат к предыдущему состоянию резольвенты.

		<p>father(andrey, olga).  ancestors(NAME, MOMSMOM,  MOMSFATHER,  FATHERSMOM,  FATHERSFATHER)</p> <p>Все попытки унификации  завершаются неудачей.</p>	
36	Резольвента: ancestors(slava, GRANDMA1, _, _, _).	<p>Попытки унификации верхней  подцели из резольвенты и еще  неотмеченных фактов из БЗ:  Все факты в БЗ отмечены.</p>	Откат к предыдущем у состоянию резольвенты.
37	Резольвента пуста.		Успешное завершение программы.