



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»

Лабораторная работа №16

*По предмету: «Функциональное и логическое
программирование»*

Преподаватель: Строганов Ю.В.

Студент: Гасанзаде М.А.,

Группа: ИУ7-66Б

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:

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!

№ шага	Состояние резольвенты, и вывод: дальнейшие действия (почему?)	Для каких термов запускается алгоритм унификации: $T1=T2$ и каков результат (и подстановка)	Дальнейшие действия: прямой ход или откат (почему и к чему приводит?)
1...	Комментарий, вывод...
...

Текст программы

predicates

```
family(symbol Child, symbol Father, symbol Mother).  
grand(symbol Child, symbol F_f, symbol M_f, symbol F_m, symbol M_m).
```

clauses

```
family(alexey, vladimir, elena).  
family(inna, egor, kseniya).  
family(vladimir, alexey, inna).
```

```
family(oleg, kirill, nadezhda).  
family(olga, andrey, anna).  
family(ivan, oleg, olga).
```

```
grand(Child, F_f, M_f, F_m, M_m) :- family(Child, Father, Mother),  
family(Father, F_f, M_f), family(Mother, F_m, M_m).
```

goal

```
% grand(ivan, _, M_f, _, M_m). %task_1  
% grand(ivan, F_f, _, F_m, _). %task_2  
% grand(ivan, F_f, M_f, F_m, M_m). %task_3  
% grand(ivan, _, _, _, M_m). %task_4  
% grand(ivan, _, _, F_m, M_m). %task_5
```

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

goal

```
grand(ivan, _, Grandma_f, _, Grandma_m).
```

Output:

```
Grandma_f=Nadezhda, Grandma_m=Anna
```

1 Solution

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

goal

```
grand(ivan, Grandpa_f, _, Grandpa_m, _).
```

Output:

```
Grandpa_f=kirill, Grandpa_m=andrey
```

1 Solution

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

goal

```
grand(ivan, Grandpa_f, Grandma_f, Grandpa_m, Grandpa_m).
```

Output:

F_f=kirill, M_f=nadezhda, F_m=andrey, M_m=anna

1 Solution

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

goal

```
grand(ivan, _, _, _, Grandma_m).
```

Output:

Grandma_m=anna

1 Solution

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

goal

```
grand(ivan, _, _, Grandpa_m, Grandma_m).
```

Output:

Grandpa_m=andrey, Grandma_m=anna

1 Solution

Описание порядка работы системы

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

goal

`grand(ivan, Grandpa_f, _, Grandpa_m, _).`

Output:

Grandpa_f=kirill, Grandpa_m=andrey

1 Solution

№ шага	Состояния резольвенты, и вывод: дальнейшие действия (почему?)	Для каких термов запускается алгоритм унификации: T1=T2 и каков результат (и подстановка)	Дальнейшие действия: прямой ход или откат (почему и к чему приводит?)
1	<code>grand(ivan, Grandpa_f, _, Grandpa_m, _)</code>	<p>Попытка унификации:</p> $\text{grand(ivan, Grandpa_f, _, Grandpa_m, _)} = \text{family(alexey, vladimir, elena)}$ <p>Результат: Неудача (разные функторы)</p>	Прямой ход, переход к следующему предложению.
2	<code>grand(ivan, Grandpa_f, _, Grandpa_m, _)</code>	<p>Попытка унификации:</p> $\text{grand(ivan, Grandpa_f, _, Grandpa_m, _)} = \text{family(inna, egor, kseniya)}$ <p>Результат: Неудача (разные функторы)</p>	Прямой ход, переход к следующему предложению.
3	<code>grand(ivan, Grandpa_f, _, Grandpa_m, _)</code>	<p>Попытка унификации:</p> $\text{grand(ivan, F_f, _, F_m, _)} = \text{family(vladimir, alexey, inna)}$ <p>Результат: Неудача (разные функторы)</p>	Прямой ход, переход к следующему предложению
4	<code>grand(ivan, Grandpa_f, _, Grandpa_m, _)</code>	<p>Попытка унификации:</p> $\text{grand(ivan, F_f, _, F_m, _)} = \text{family(oleg, kirill, nadezhda)}$ <p>Результат: Неудача (разные функторы)</p>	Прямой ход, переход к следующему предложению
5	<code>grand(ivan, Grandpa_f, _, Grandpa_m, _)</code>	<p>Попытка унификации:</p> $\text{grand(ivan, F_f, _, F_m, _)} = \text{family(olga, andrey, anna)}$	Прямой ход, переход к следующему предложению

		Результат: Неудача (разные функторы)	
6	<code>grand(ivan, Grandpa_f, _, Grandpa_m, _)</code>	Попытка унификации: <code>grand(ivan, F_f, _, F_m, _)</code> = <code>family(ivan, oleg, olga)</code> Результат: Неудача (разные функторы)	Прямой ход, переход к следующему предложению
7	<code>grand(ivan, Grandpa_f, _, Grandpa_m, _)</code>	Попытка унификации: <code>grand(ivan, F_f, _, F_m, _)</code> = <code>grand(Child, F_f, M_f, F_m, M_m)</code> Результат: Успех (подстановка) {Child = ivan, F_f = F_f, F_m = F_m}	Редукция цели, прямой ход к сопоставлению <code>family(Child, Father, Mother)</code> , поиск с начала предложений.
8	<code>family(ivan, Father, Mother), family(Father, F_f, _), family(Mother, F_m, _)</code>	Попытка унификации: <code>family(ivan, Father, Mother)</code> = <code>family(alexey, vladimir, elena)</code> Результат: Неудача (<code>ivan ≠ alexey</code>)	Прямой ход, переход к следующему предложению
9	<code>family(ivan, Father, Mother), family(Father, F_f, _), family(Mother, F_m, _)</code>	Попытка унификации: <code>family(ivan, Father, Mother)</code> = <code>family(inna, egor, kseniya)</code> Результат: Неудача (<code>ivan ≠ inna</code>)	Прямой ход, переход к следующему предложению
10	<code>family(ivan, Father, Mother), family(Father, F_f, _), family(Mother, F_m, _)</code>	Попытка унификации: <code>family(ivan, Father, Mother)</code> = <code>family(vladimir, alexey, inna)</code> Результат: Неудача (<code>ivan ≠ vladimir</code>)	Прямой ход, переход к следующему предложению
11	<code>family(ivan, Father, Mother), family(Father, F_f, _), family(Mother, F_m, _)</code>	Попытка унификации: <code>family(ivan, Father, Mother)</code> = <code>family(oleg, kirill, nadezhda)</code> Результат: Неудача (<code>ivan ≠ oleg</code>)	Прямой ход, переход к следующему предложению
12	<code>family(ivan, Father, Mother), family(Father, F_f, _), family(Mother, F_m, _)</code>	Попытка унификации: <code>family(ivan, Father, Mother)</code> = <code>family(olga, andrey, anna)</code> Результат: Неудача (<code>ivan ≠ olga</code>)	Прямой ход, переход к следующему предложению

13	<pre>family(ivan, Father, Mother), family(Father, F_f, _), family(Mother, F_m, _)</pre>	<p>Попытка унификации:</p> <pre>family(ivan, Father, Mother) = family(ivan, oleg, olga)</pre> <p>Результат: Успех (подстановка) {Father = oleg, Mother = olga}</p>	<p>Прямой ход к сопоставлению <code>family(oleg, F_f, _)</code>, поиск с начала предложений.</p>
14	<pre>family(Father, F_f, _), family(Mother, F_m, _)</pre>	<p>Попытка унификации:</p> <pre>family(oleg, F_f, _) = family(alexeу, vladimir, elena)</pre> <p>Результат: Неудача (oleg ≠ alexeу)</p>	<p>Прямой ход, переход к следующему предложению</p>
15	<pre>family(Father, F_f, _), family(Mother, F_m, _)</pre>	<p>Попытка унификации:</p> <pre>family(oleg, F_f, _) = family(inna, egor, kseniya)</pre> <p>Результат: Неудача (oleg ≠ inna)</p>	<p>Прямой ход, переход к следующему предложению</p>
16	<pre>family(Father, F_f, _), family(Mother, F_m, _)</pre>	<p>Попытка унификации:</p> <pre>family(oleg, F_f, _) = family(vladimir, alexeу, inna)</pre> <p>Результат: Неудача (oleg ≠ vladimir)</p>	<p>Прямой ход, переход к следующему предложению</p>
17	<pre>family(Father, F_f, _), family(Mother, F_m, _)</pre>	<p>Попытка унификации:</p> <pre>family(oleg, F_f, _) = family(oleg, kirill, nadezhda)</pre> <p>Результат: Успех (подстановка) {F_f = kirill}</p>	<p>Прямой ход к сопоставлению <code>family(olga, F_m, _)</code>, поиск с начала предложений.</p>
18	<pre>family(olga, F_m, _)</pre>	<p>Попытка унификации:</p> <pre>family(olga, F_m, _) = family(alexeу, vladimir, elena)</pre> <p>Результат: Неудача (olga ≠ alexeу)</p>	<p>Прямой ход, переход к следующему предложению</p>
19	<pre>family(olga, F_m, _)</pre>	<p>Попытка унификации:</p> <pre>family(olga, F_m, _) = family(inna, egor, kseniya)</pre> <p>Результат: Неудача (olga ≠ inna)</p>	<p>Прямой ход, переход к следующему предложению</p>

20	<code>family(olga, F_m, _)</code>	<p>Попытка унификации:</p> $\text{family}(\text{olga}, F_m, _) = \text{family}(\text{vladimir}, \text{alexey}, \text{inna})$ <p>Результат: Неудача ($\text{olga} \neq \text{vladimir}$)</p>	Прямой ход, переход к следующему предложению
21	<code>family(olga, F_m, _)</code>	<p>Попытка унификации:</p> $\text{family}(\text{olga}, F_m, _) = \text{family}(\text{oleg}, \text{kirill}, \text{nadezhda})$ <p>Результат: Неудача ($\text{olga} \neq \text{oleg}$)</p>	Прямой ход, переход к следующему предложению
22	<code>family(olga, F_m, _)</code>	<p>Попытка унификации:</p> $\text{family}(\text{olga}, F_m, _) = \text{family}(\text{olga}, \text{andrey}, \text{anna})$ <p>Результат: Успех (подстановка) $\{F_m = \text{andrey}\}$</p>	<p>Вывод: $F_f = \text{kirill},$ $F_m = \text{andrey}.$</p> <p>Опустошение резольвенты вследствие успешной унификации.</p> <p>Откат, переход к предыдущему состоянию резольвенты. Реконкретизация F_m.</p>
23	<code>family(olga, F_m, _)</code>	<p>Попытка унификации:</p> $\text{family}(\text{olga}, F_m, _) = \text{family}(\text{ivan}, \text{oleg}, \text{olga})$ <p>Результат: Неудача ($\text{olga} \neq \text{ivan}$)</p>	Прямой ход к следующему предложению.
24	<code>family(olga, F_m, _)</code>	<p>Попытка унификации:</p> $\text{family}(\text{olga}, F_m, _) = \text{grand}(\text{Child}, F_f, M_f, F_m, M_m)$ <p>Результат: Неудача (разные функторы)</p>	Конец БЗ, откат, переход к предыдущему состоянию резольвенты (шаг 17), реконкретизация F_f .
25	<code>family(Father, F_f, _), family(Mother, F_m, _)</code>	<p>Попытка унификации:</p> $\text{family}(\text{oleg}, F_m, _) = \text{family}(\text{olga}, \text{andrey}, \text{anna})$ <p>Результат: Неудача ($\text{oleg} \neq \text{olga}$)</p>	Прямой ход к следующему предложению.
26	<code>family(Father, F_f, _), family(Mother, F_m, _)</code>	<p>Попытка унификации:</p> $\text{family}(\text{oleg}, F_m, _) = \text{family}(\text{ivan}, \text{oleg}, \text{olga})$ <p>Результат: Неудача ($\text{oleg} \neq \text{ivan}$)</p>	Прямой ход к следующему предложению.

27	<code>family(Father, F_f, _),</code> <code>family(Mother, F_m, _)</code>	Попытка унификации: <code>family(oleg, F_m, _)</code> <code>=</code> <code>grand(Child, F_f, M_f, F_m, M_m)</code> Результат: Неудача (разные функторы)	Конец БЗ, откат, переход к предыдущему состоянию резольвенты (шаг 13), реконкретизация Father и Mother.
28	<code>family(ivan, Father, Mother),</code> <code>family(Father, F_f, _),</code> <code>family(Mother, F_m, _)</code>	Попытка унификации: <code>family(ivan, Father, Mother)</code> <code>=</code> <code>grand(Child, F_f, M_f, F_m, M_m)</code> Результат: Неудача (разные функторы)	Откат, переход к предыдущему состоянию резольвенты (шаг 7).
29	<code>grand(ivan, Grandpa_f, _, Grandpa_m, _)</code> конец clauses; опустошение резольвенты; завершение работы.		

Теоретические вопросы.

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

Если есть что доказывать (цель), то процесс унификации запускается автоматически. Формально: если резольвента не пуста – запускается алгоритм унификации.

2. Каковы назначение и результат использования алгоритма унификации?

Назначение алгоритма унификации заключается в попарном сопоставлении термов и попытке построить для них общий пример. Унификация может завершаться успехом или тупиковой ситуацией (неудачей).

3. Какое первое состояние резольвенты?

Если задан простой вопрос, то сначала он попадает в резольвенту.

4. Как меняется резольвента?

Изменение резольвенты происходит в 2 этапа:

1) из стека выбирается подцель (верхняя, т.к. стек) и для нее выполняется редукция, т.е. замена подцели на тело найденного правила;

2) к полученной конъюнкции целей применяется подстановка (наибольший общий унификатор выбранной цели и заголовка сопоставленного с этой целью правила).

5. В каких пределах программы уникальны переменные?

Переменные уникальны в пределах предложения, т.е. в рамках предложения одно и то же имя принадлежит одной и той же переменной. Исключение – анонимные переменные (обозначаются символом нижнего подчеркивания «_») – каждая такая переменная является отдельной сущностью и применяется, когда ее значение неважно для данного предложения.

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

Применение подстановки $\{x_1=t_1, \dots, x_n=t_n\}$ заключается в замене каждого вхождения переменной x_i на соответствующий терм t_i .

7. В каких случаях запускается механизм отката?

Механизм отката запускается, если возникла тупиковая ситуация (достигнут конец БЗ) либо резольвента пуста. В таких случаях происходит откат к предыдущему состоянию резольвенты.