



**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**

Тема Использование правил в программе на Prolog

Студент Сушина А.Д.

Группа ИУ7-616

Оценка (баллы) \_\_\_\_\_

Преподаватель Толпинская Н.Б.

Москва.

2020 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 Задание

#### Ответить на вопросы:

- В каком случае система запускает алгоритм унификации? (Как эту необходимость на формальном уровне распознает система?)
- Каковы назначение и результат использования алгоритма унификации?
- Какое первое состояние резольвенты?
- Как меняется резольвента?
- В каких пределах программы уникальны переменные?
- Как применяется подстановка, полученная с помощью алгоритма унификации?
- В каких случаях запускается механизм отката?

**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 <b>результат</b> (и подстановка)	Дальнейшие действия: прямой ход или откат (почему и к чему приводит?)
1...	...	...	Комментарий, вывод...
...	...	...	...

### Ответы на вопросы:

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

Процесс унификации в случае, когда необходимо доказать какое-то утверждение. Если резольвента не пуста, есть, что доказывать. Также пользователь может запустить этот процесс вручную с помощью утверждения  $T1=T2$ .

#### 2. 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»

#### 3. Какое первое состояние резольвенты?

Первое состояние резольвенты - заданный вопрос.

#### 4. 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.

#### 5. 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.

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

Если алгоритм унификации завершился успешно и найдена подстановка, соответствующие переменные конкретизируются полученными значениями.

## **7. 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.

### **Код программы**

domains

NAME = symbol

predicates

mother(NAME, NAME)

father(NAME, NAME)

all\_grands(NAME, NAME, NAME, NAME, NAME)

clauses

mother(lena, vitya). %vitya's mom is lena

mother(olga, lena). %lena's mom is olga

mother(luba, petya). %petya's mom is luba

father(petya, vitya). %vitya's dad is petya

father(kolya, petya). %petya's dad is kolya

father(dima, lena). %lena's dad is dima

all\_grands(Name, MomGrandMa, MomGrandPa, DadGrandMa, DadGrandPa):-  
mother(Mom, Name),

```

mother(MomGrandMa, Mom),
father(MomGrandPa, Mom),
father(Dad, Name),
mother(DadGrandMa, Dad),
father(DadGrandPa, Dad).

```

goal

```
%all_grands(vitya, MomGrandMa,_,DadGrandMa,_). % all grandmas
```

```
%all_grands(vitya,_,MomGrandPa,_,DadGrandPa). % all grandfas
```

```
%all_grands(vitya, MomGrandMa, MomGrandPa, DadGrandMa, DadGrandPa). % all
grands
```

```
%all_grands(vitya, MomGrandMa,_,_,_). % mother_line grandma
```

```
all_grands(vitya,_,_,DadGrandMa,_). % mother_line grands
```

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

```
all_grands(vitya, MomGrandMa,_,DadGrandMa,_)
```

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

```
all_grands(vitya,_,MomGrandPa,_,DadGrandPa)
```

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

```
all_grands(vitya, MomGrandMa, MomGrandPa, DadGrandMa, DadGrandPa)
```

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

```
all_grands(vitya, MomGrandMa,_,_,_)
```

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

```
all_grands(vitya,_,_,DadGrandMa,_)
```

№ 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?)
1-6	Резольвента: <b>all_grands(vitya, MomGrandMa,_,_,_)</b>	<b>all_grands(vitya, MomGrandMa,_,_,_)</b> сравнивается с каждым термом из mother(lena, vitya).	Переход к следующему

	<p>Первое состояние резольвенты — сам вопрос. Далее запускается унификация.</p>	<p>mother(olga, lena).  mother(luba, petya).  father(petya, vitya).  father(kolya, petya).  father(dima, lena).  На каждом шаге производится попытка унификации. Унификация не успешна, факторы не совпадают.</p>	предложению
7	Резольвента: <b>all_grands(vitya, MomGrandMa, _, _, _)</b>	<p>T1 = all_grands(vitya, MomGrandMa, _, _, _)  T2 = all_grands(Name, MomGrandMa, MomGrandPa, DadGrandMa, DadGrandPa)  Попытка унификации. Результат: Успех.  Подстановка:  { Name = vitya,  MomGrandMa = MomGrandMa,  MomGrandPa = MomGrandPa,  DadGrandMa = DadGrandMa,  DadGrandPa = DadGrandPa}</p>	<p>Прямой ход.  Содержимое резольвенты заменяется телом найденного правила. К резольвенте применяется подстановка.</p>
8	Резольвента: mother(Mom, vitya), mother(MomGrandMa, Mom), father(MomGrandPa, Mom), father(Dad, vitya), mother(DadGrandMa, Dad), father(DadGrandPa, Dad).	<p>T1 = mother(Mom, vitya)  T2 = mother(lena, vitya)  Попытка унификации. Результат: успех.  Подстановка:  { Mom = lena  vitya = vitya}</p>	<p>Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции.</p>
9	Резольвента: mother(MomGrandMa, lena), father(MomGrandPa, lena), father(Dad, vitya), mother(DadGrandMa, Dad), father(DadGrandPa, Dad).	<p>T1 = mother(MomGrandMa, lena)  T2 = mother(lena, vitya)  Попытка унификации. Результат: неудача  lena != vitya</p>	Переход к следующему предложению
10	Резольвента: mother(MomGrandMa, lena), father(MomGrandPa, lena),	<p>T1 = mother(MomGrandMa, lena)  T2 = mother(olga, lena)  Попытка унификации. Результат: успех.</p>	Прямой ход. К резольвенте применяется

	father(Dad, vitya), mother(DadGrandMa,Dad), father(DadGrandPa, Dad).	Подстановка: { MomGrandMa = olga lena=lena}	подстановка. Резольвента преобразуется с помощью редукции.
11	Резольвента: father(MomGrandPa,lena), father(Dad, vitya), mother(DadGrandMa,Dad), father(DadGrandPa, Dad).	<b>father(MomGrandPa,lena)</b> сравнивается с каждым термом из mother(lena, vitya). mother(olga, lena). mother(luba, petya). На каждом шаге производится попытка унификации. Унификация не успешна, функторы не совпадают.	Переход к следующему предложению
12	Резольвента: father(MomGrandPa,lena), father(Dad, vitya), mother(DadGrandMa,Dad), father(DadGrandPa, Dad).	T1 = father(MomGrandPa,lena) T2 = father(petya, vitya) Попытка унификации. Результат: неудача lena != vitya	Переход к следующему предложению
13	Резольвента: father(MomGrandPa,lena), father(Dad, vitya), mother(DadGrandMa,Dad), father(DadGrandPa, Dad).	T1 = father(MomGrandPa,lena) T2 = father(kolya, petya) Попытка унификации. Результат: неудача lena != petya	Переход к следующему предложению
14	Резольвента: father(MomGrandPa,lena), father(Dad, vitya), mother(DadGrandMa,Dad), father(DadGrandPa, Dad).	T1 = father(MomGrandPa,lena) T2 = father(dima, lena) Попытка унификации. Результат: успех. Подстановка: { MomGrandPa = Dima lena=lena}	Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции.
15	Резольвента: father(Dad, vitya), mother(DadGrandMa,Dad), father(DadGrandPa, Dad).	<b>father(Dad, vitya),</b> сравнивается с каждым термом из mother(lena, vitya). mother(olga, lena). mother(luba, petya). На каждом шаге производится попытка	Переход к следующему предложению

		унификации. Унификация не успешна, функторы не совпадают.	
16	Резольвента: father(Dad, vitya), mother(DadGrandMa,Dad), father(DadGrandPa, Dad).	T1 = father(Dad, vitya) T2 = father(petya, vitya) Попытка унификации. Результат: успех. Подстановка: { Dad = petya vitya=vitya}	Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции.
17	Резольвента: mother(DadGrandMa,petya), father(DadGrandPa, petya).	T1 = mother(DadGrandMa,petya) T2 = mother(lena, vitya) Попытка унификации. Результат: неудача vitya != petya	Переход к следующему предложению
18	Резольвента: mother(DadGrandMa,petya), father(DadGrandPa, petya).	T1 = mother(DadGrandMa,petya) T2 = mother(olga, lena) Попытка унификации. Результат: неудача petya != lena	Переход к следующему предложению
19	Резольвента: mother(DadGrandMa,petya), father(DadGrandPa, petya).	T1 = mother(DadGrandMa,petya) T2 = mother(luba, petya) Попытка унификации. Результат: успех. Подстановка: { DadGrandMa = luba petya = petya}	Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции.
20	Резольвента: father(DadGrandPa, petya).	<b>father(DadGrandPa, petya).</b> сравнивается с каждым термом из mother(lena, vitya). mother(olga, lena). mother(luba, petya). На каждом шаге производится попытка унификации. Унификация не успешна, функторы не совпадают.	Переход к следующему предложению



21	Резольвента: father(DadGrandPa, petya).	T1 = father(DadGrandPa, petya). T2 = father(petya, vitya) Попытка унификации. Результат: неудача petya != vitya	Переход к следующему предложению
22	Резольвента: father(DadGrandPa, petya).	T1 = father(DadGrandPa, petya). T2 = father(kolya, petya) Попытка унификации. Результат: успех. Подстановка: { DadGrandPa = kolya petya = petya }	Прямой ход. К резольвенте применяется подстановка. Резольвента преобразуется с помощью редукции.
23	Резольвента пуста. Все переменные связаны		Вывод результата. Поиск альтернативного решения. Откат к предыдущему состоянию резольвенты.
24	Возврат предыдущего состояния резольвенты. Резольвента: father(DadGrandPa, petya).	Происходят унификации подцели из резольвенты и еще неотмеченных фактов из базы знаний. father(dima, lena) all_grands(Name, MomGrandMa, MomGrandPa, DadGrandMa, DadGrandPa) Все унификации завершаются неудачей petya != lena в первом случае и разный функторы во втором.	Откат к предыдущему состоянию резольвенты
25	Возврат предыдущего состояния резольвенты. Резольвента: mother(DadGrandMa, petya), father(DadGrandPa, petya).	Происходят унификации подцели из резольвенты и еще неотмеченных фактов из базы знаний. Все унификации завершаются неудачей	Откат к предыдущему состоянию резольвенты
26	Возврат предыдущего	Происходят унификации подцели из	Откат к

	состояния резольвенты. Резольвента: father(Dad, vitya), mother(DadGrandMa,Dad), father(DadGrandPa, Dad).	резольвенты и еще неотмеченных фактов из базы знаний. Все унификации завершаются неудачей	предыдущему состоянию резольвенты
27	Возврат предыдущего состояния резольвенты. Резольвента: father(MomGrandPa,lena), father(Dad, vitya), mother(DadGrandMa,Dad), father(DadGrandPa, Dad).	Происходят унификации подцели из резольвенты и еще неотмеченных фактов из базы знаний. Все унификации завершаются неудачей	Откат к предыдущему состоянию резольвенты
28	Возврат предыдущего состояния резольвенты. Резольвента: mother(MomGrandMa, lena), father(MomGrandPa,lena), father(Dad, vitya), mother(DadGrandMa,Dad), father(DadGrandPa, Dad).	Происходят унификации подцели из резольвенты и еще неотмеченных фактов из базы знаний. Все унификации завершаются неудачей	Откат к предыдущему состоянию резольвенты
29	Возврат предыдущего состояния резольвенты. Резольвента: mother(Mom, vitya), mother(MomGrandMa, Mom), father(MomGrandPa,Mom), father(Dad, vitya), mother(DadGrandMa,Dad), father(DadGrandPa, Dad).	Происходят унификации подцели из резольвенты и еще неотмеченных фактов из базы знаний. Все унификации завершаются неудачей	Откат к предыдущему состоянию резольвенты
30	Резольвента: <b>all_grands(vitya, MomGrandMa, _ _ _)</b>	Происходят унификации подцели из резольвенты и еще неотмеченных фактов из базы знаний. Все унификации завершаются неудачей	Унификации безуспешны. Завершение выполнения программы.

### Замечания к лр 14

ИУ7-616 Сушина А.Д.

Задание 3. А шаги 1-14 это работа того же запуска алгоритма унификации? А он много раз запускается?, А когда останов? А 23-й шаг – продолжает??? Работать алгоритм??

На 1-14 шагах запускаются разные алгоритмы унификации для термина `byModelColorFindPerson(volvo, red, LASTNAME, CITY, PHONE, BANK)` и других термов из БЗ. Алгоритм работы одного алгоритма унификации представлен на шагах 15-22. На шаге 23 запускается другой алгоритм унификации для других термов.

Алгоритм унификации на шагах 15-22 завершается успехом. Побочный эффект — подстановка

{

MODEL = `volvo`,

COLOR = `red`,

LASTNAME=LASTNAME,

CITY=CITY,

PHONE=PHONE

}

Шаг унификации	Результирующая ячейка	Рабочее поле	Стек
1-14		...	<code>byModelColorFindPerson(volvo, red, LASTNAME, CITY, PHONE, BANK) =</code> ...
15			<code>byModelColorFindPerson(volvo, red, LASTNAME, CITY, PHONE, BANK) =</code> <code>byModelColorFindPerson(MODEL, COLOR, LASTNAME, CITY, PHONE, BANK)</code>
16		<code>byModelColorFindPerson(volvo, red, LASTNAME, CITY, PHONE, BANK) =</code>	MODEL = <code>volvo</code> COLOR = <code>red</code> LASTNAME=LASTNAME

		byModelColorFindPerson(MODEL, COLOR, LASTNAME, CITY, PHONE, BANK)	CITY=CITY PHONE=PHONE BANK=BANK
17		MODEL = volvo	COLOR = red LASTNAME=LASTNAME CITY=CITY PHONE=PHONE BANK=BANK
18	MODEL = volvo	COLOR = red	LASTNAME=LASTNAME CITY=CITY PHONE=PHONE BANK=BANK
19	MODEL = volvo COLOR = red	LASLNAME=LASTNAME	CITY=CITY PHONE=PHONE BANK=BANK
20	MODEL = volvo COLOR = red LASTNAME=LASTNAME	CITY=CITY	PHONE=PHONE BANK=BANK
21	MODEL = volvo COLOR = red LASTNAME=LASTNAME CITY=CITY	PHONE=PHONE	BANK=BANK
22	MODEL = volvo COLOR = red LASTNAME=LASTNAME CITY=CITY PHONE=PHONE	BANK=BANK	
23	MODEL = volvo COLOR = red LASTNAME=LASTNAME CITY=CITY PHONE=PHONE BANK=BANK		car(LASTNAME, volvo, _, red, CITY)= person(petrov, "1", adress(moscow, lenina, 34, 12))