



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 № 20

Дисциплина Функциональное и логическое программирование

Тема Работа программы на Prolog

Студент Куприй А. А.

Группа ИУ7-63Б

Оценка (баллы) \_\_\_\_\_

Преподаватель Толпинская Н. Б., Строганов Ю. В.

Москва, 2020 г.

**Цель работы** – получить навыки построения модели предметной области, разработки и оформления программы на Prolog, изучить принципы, логику формирования программы и отдельные шаги выполнения программы на Prolog.

**Задачи работы:** приобрести навыки декларативного описания предметной области с использованием фактов и правил.

Изучить способы использования термов, переменных, фактов и правил в программе на Prolog, принципы и правила сопоставления и отождествления, порядок унификации.

**Задание:**

Используя хвостовую рекурсию, разработать, комментируя аргументы, эффективную программу, позволяющую:

1. Сформировать список из элементов числового списка, больших заданного значения;
2. Сформировать список из элементов, стоящих на нечетных позициях исходного списка (нумерация от 0);
3. Удалить заданный элемент из списка (один или все вхождения);
4. Преобразовать список в множество (можно использовать ранее разработанные процедуры).

## **Вопросы:**

### **Как организуется хвостовая рекурсия в Prolog?**

- Рекурсивный вызов единственен и расположен в конце тела правила
- Не должно быть возможности сделать откат до вычисления рекурсивного вызова

### **Как организовать выход из рекурсии в Prolog?**

С помощью отсечения.

### **Какое первое состояние резольвенты?**

Заданный вопрос (goal).

### **Каким способом можно разделить список на части, какие, требования к частям?**

Получить голову или хвост списка можно при унификации списка с  $[H|T]$ ,  $H$  – голова списка,  $T$  – хвост списка (является списком).

### **Как выделить за один шаг первые два подряд идущих элемента списка?**

$[H1|[H2|_]]$

### **Как выделить 1-й и 3-й элемент за один шаг?**

$[H1|_|[H3|_]]$

### **Как формируется новое состояние резольвенты?**

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.

### **Когда останавливается работа системы?**

Работа системы останавливается, когда найдены все возможные ответы на вопрос.

### **Как это определяется на формальном уровне?**

Когда в резольвенте находится исходный вопрос, для которого пройдена вся БЗ.

## Текст программы

### domains

**list = integer\*.**

### predicates

**biggerThan(list, integer, list).  
oddElements(list, list).  
deleteElement(list, integer, list).  
createSet(list, list).**

### clauses

**biggerThan([], \_, []).  
biggerThan([H|T], Min, [H|ResultTail]) :-  
    H > Min,  
    biggerThan(T, Min, ResultTail).  
biggerThan([H|T], Min, ResultTail) :-  
    H <= Min,  
    biggerThan(T, Min, ResultTail).**

**oddElements([], []).  
oddElements([\_], []).  
oddElements([\_|[H|T]], [H|ResultTail]) :-  
    oddElements(T, ResultTail).**

**deleteElement([], \_, []).  
deleteElement([El|T], El, ResultTail) :-  
    deleteElement(T, El, ResultTail).  
deleteElement([H|T], El, [H|ResultTail]) :-  
    H <> El,  
    deleteElement(T, El, ResultTail).**

**createSet([], []).  
createSet([H|T], [H|ResultTail]) :-  
    deleteElement(T, H, TmpResult),  
    createSet(TmpResult, ResultTail).**

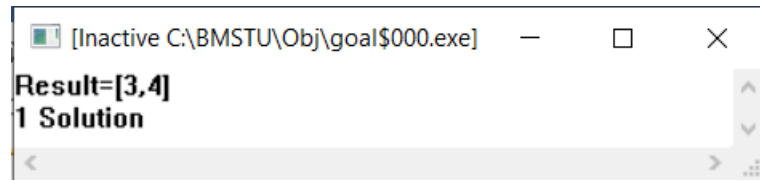
### goal

**biggerThan([1, 2, 3, 4, 2, 1], 2, Result).  
%oddElements([1, 2, 3], Result).  
%deleteElement([1, 2, 3, 3, 1, 3, 4], 3, Result).  
%createSet([1, 2, 3, 3, 1, 3, 4], Result).**

## Результаты работы программы:

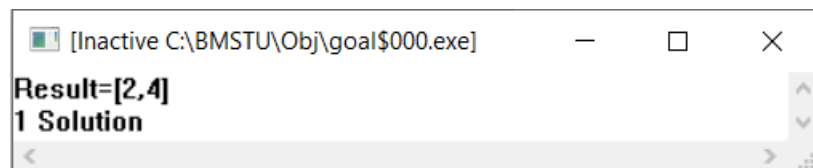
**goal**

**biggerThan([1, 2, 3, 4, 2, 1], 2, Result).**



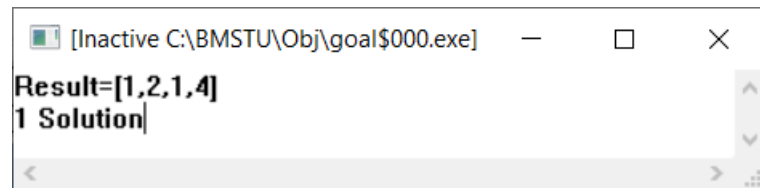
**goal**

**oddElements([1, 2, 3, 4], Result).**



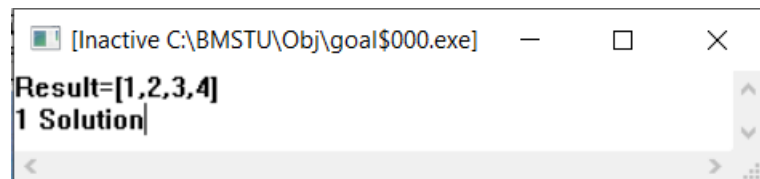
**goal**

**deleteElement([1, 2, 3, 3, 1, 3, 4], 3, Result).**



**goal**

**createSet([1, 2, 3, 3, 1, 3, 4], Result).**



## Таблицы

Цель:

goal

**biggerThan([1], 2, Result).**

№ 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	biggerThan([1], 2, Result)	ТЦ: biggerThan([1], 2, Result)	Поиск знания с начала БЗ
	biggerThan([1], 2, Result)	ПР1: [] = [1] _ = 2 [] = Result Неудача	Метка переносится ниже
	biggerThan([1], 2, Result)	ПР2: [H1 T1] = [1] Min1 = 2 [H1 ResultTail1] = Result Успех H1 = 1 T1 = [] Min1 = 2 Result = [1 ResultTail1]	Тело ПР2 заменяет цель в резольвенте
2	1 > 2 biggerThan([], 2, ResultTail1) !	Сравнение: 1 > 2 Ложь	Откат к 1. Метка переносится ниже.
3	biggerThan([1], 2, Result)	ПР3: [_ T3] = [1] Min3 = 2 ResultTail3 = Result Успех T3 = [] Min3 = 2 Result = ResultTail3	Тело ПР3 заменяет цель в резольвенте
4	biggerThan([], 2, ResultTail3)	ТЦ: biggerThan([], 2, ResultTail3)	Поиск знания с начала БЗ
	biggerThan([], 2, ResultTail3)	ПР1: [] = [] _ = 2 [] = ResultTail3 Успех ResultTail3 = 0	Пустое тело заменяет цель в резольвенте

	Пусто		Успех. Result = ResultTail3 = [] Возврат к предыдущему состоянию резольвенты
5	biggerThan([], 2, ResultTail3)	ПР2: [H5 T5] = [] Min5 = 2 [H5 ResultTail5] = ResultTail3 Неудача	Метка переносится ниже.
	biggerThan([], 2, ResultTail3)	ПР3: [_ T3] = [] Min5 = 2 ResultTail5 = Result Неудача	Необходимо включить откат, но метки в конце процедур – альтернатив нет. Система завершает работу.