



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»

## Отчёт по лабораторной работе №6

Название: Формирование эффективных программ на Prolog.

Дисциплина: Функциональное и логическое программирование.

Студент

ИУ7-64Б

(Группа)

(Подпись, дата)

Л.Е.Тартыков

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Н.Б.Толпинская

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Ю.В.Строганов

(И.О. Фамилия)

*Москва, 2022*

# 1 Практические задания

## 1.1 Задание

Используя хвостовую рекурсию, разработать программу, позволяющую найти:

1.  $n!$ ;

2.  $n$ -е число Фибоначчи.

Убедиться в правильности результатов. Для одного из вариантов ВОПРОСА и каждого задания составить таблицу, отражающую конкретный порядок работы системы.

Код программы представлен на листинге 1.1.

Листинг 1.1 – Код программы

```
1 predicates
2     factorial(integer , integer).
3     factorial(integer , integer , integer).
4
5     fib(integer , integer).
6     fib(integer , integer , integer , integer).
7 clauses
8     factorial(N, Factorial_N, Factorial_M) :- N > 1,
9                                         Temp_factorial_N = Factorial_M * N,
10                                        M = N - 1, !,
11                                        factorial(M, Factorial_N, Temp_factorial_N).
12     factorial(_, Factorial_M, Factorial_M).
13     factorial(N, Factorial_N) :- factorial(N, Factorial_N, 1).
14
15     fib(N, Fib_N, Last_N, Last_fib) :- N > 3,
16                                         Temp_fib = Last_N + Last_fib ,
17                                         Temp_N = N - 1, !,
18                                         fib(Temp_N, Fib_N, Last_fib , Temp_fib).
19     fib(_, Temp_fib, _, Temp_fib).
20     fib(1, 0).
21     fib(N, Fib_N) :- fib(N, Fib_N, 1, 1).
22 goal
23     %factorial(5, Factorial_N).
24     fib(4, Fib_elem).
```

Ниже на рисунках 1.1 1.2 приведена таблица порядка поиска ответа для нахождения факториала:

№ 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?)
...			
3	factorial(3, Factorial_N)	factorial(0, Factorial_N) = factorial(N, Factorial_N) Результат: унификация успешна. Подстановка: {N = 3}	Прямой ход. Переход к телу правила. Редукция и подстановка в резольвенту.
4	factorial(3, Factorial_N, 1)	factorial(3, Factorial_N, 1) = factorial(N, Factorial_N, Factorial_M) Результат: унификация успешна. Подстановка: {Factorial_M = 1}	Прямой ход. Редукция и подстановка в резольвенту.
5	$3 > 1$ Temp_factorial_N = 1 * 3 M = 3 - 1 ! factorial(M, Factorial_N, Temp_factorial_N)	$3 > 1$ . Результат: да.	Прямой ход. Переход к следующей цели в резольвенте.
6	Temp_factorial_N = 1 * 3 M = 3 - 1 ! factorial(M, Factorial_N, Temp_factorial_N)	Temp_factorial_N = 1 * 3. Результат: унификация успешна Подстановка {Temp_factorial_N = 3}	Прямой ход. Переход к следующей цели в резольвенте.
7	M = 3 - 1 ! factorial(M, Factorial_N, Temp_factorial_N)	M = 3 - 1 Результат: унификация успешна. Подстановка {M = 2}	Прямой ход. Переход к следующей цели в резольвенте.
8	! factorial(2, Factorial_N, 3)	! Результат: да.	Прямой ход. Переход к следующей цели в резольвенте.
9	factorial(2, Factorial_N, 3)	factorial(2, Factorial_N, 3) = factorial(N, Factorial_N, Factorial_M) Результат: унификация успешна. Подстановка: {N = 2, Factorial_M = 3}	Прямой ход. Редукция и подстановка в резольвенту.
10	$2 > 1$ Temp_factorial_N = 3 * 2 M = 2 - 1 ! factorial(M, Factorial_N, Temp_factorial_N)	$3 > 1$ . Результат: да.	Прямой ход. Переход к следующей цели в резольвенте.
...			
13	! factorial(1, Factorial_N, 6)	! Результат: да.	Прямой ход. Переход к следующей цели в резольвенте.
14	factorial(1, Factorial_N, 6)	factorial(1, Factorial_N, 6) = factorial(N, Factorial_N, Factorial_M) Результат: унификация успешна. Подстановка: {N = 1, Factorial_M = 6}	Прямой ход. Переход к следующей цели в резольвенте.
15	$3 > 1$ Temp_factorial_N = 1 * 3 M = 3 - 1 ! factorial(M, Factorial_N, Temp_factorial_N)	$1 > 1$ . Результат: нет.	Обратный ход. Восстановление предыдущего состояния резольвенты. Реконкретизация переменных.

Рисунок 1.1 – Таблица порядка поиска ответов для нахождения факториала.

№ 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?)
16	factorial(1, Factorial_N, 6)	factorial(1, Factorial_N, 6) = factorial(_, Factorial_M, Factorial_M) Результат: унификация успешна. Подстановка: {Factorial_M = 6, Factorial_N = 6}	Прямой ход. Переход к следующему предложению.
...			
21	factorial(1, 6, 6)	БЗ пуста.	Обратный ход (тело пусто, но резольвента нет). Восстановление предыдущего состояния резольвенты.
22	! factorial(1, 6, 6)	! Результат: нет.	Запрет унификации терма factorial(1, 6, 6). Обратный ход (тело пусто, но резольвента нет). Восстановление предыдущего состояния резольвенты из шага 8.
23	! factorial(2, 6, 3)	! Результат: нет.	Запрет унификации терма factorial(2, 6, 3). Обратный ход (тело пусто, но резольвента нет). Восстановление предыдущего состояния резольвенты из шага 4.
24	factorial(3, 6, 1)	Тело пусто. Добавление в рез. ячейку {Factorial_N = 6}. Реконкретизация Factorial_N.	Обратный ход. Переход к следующему предложению.
...			
30	factorial(3, Factorial_N, 1)	factorial(3, Factorial_N, 1) = fib(N, Fib_N). Результат: унификация неуспешна.	Обратный ход. Реконкретизация переменных. Восстановление предыдущего состояния резольвенты.
...			
35	factorial(3, Factorial_N)	factorial(3, Factorial_N) = fib(N, Fib_N). Результат: унификация неуспешна.	Обратный ход. Резольвента пуста. БЗ просмотрена полностью. Вывод на экран подстановки {Factorial_N = 6}

Рисунок 1.2 – Таблица порядка поиска ответов для нахождения факториала (продолжение).

Ниже на рисунке 1.3 и 1.4 приведена таблица порядка поиска ответа для нахождения значения N-ого числа Фибоначчи:

№ 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?)
...			
7	fib(4, Fib_elem)	fib(4, Fib_elem) = fib(N, Fib_N) Результат: унификация успешна. Подстановка: {N = 4}	Прямой ход. Переход к телу правила. Редукция и подстановка в резольвенту.
...			
11	fib(4, Fib_N, 1, 1)	fib(4, Fib_N, 1, 1) = fib(N, Fib_N, Last_N, Last_fib) Результат: унификация успешна. Подстановка: {Last_N = 1, Last_fib = 1}	Прямой ход. Редукция и подстановка в резольвенту.
12	4 > 3 Temp_fib = 1 + 1 Temp_N = 4 - 1 ! fib(Temp_N, Fib_N, 1, Temp_fib)	4 > 3. Результат: унификация успешна.	Прямой ход. Переход к следующей цели в резольвенте.
13	Temp_fib = 1 + 1 Temp_N = 4 - 1 ! fib(Temp_N, Fib_N, 1, Temp_fib)	Temp_fib = 1 + 1 Результат: унификация успешна. Подстановка: {Temp_Fib = 2}	
14	Temp_N = 4 - 1 ! fib(Temp_N, Fib_N, 1, 2)	Temp_N = 4 - 1 Подстановка: {Temp_N = 3}	Прямой ход. Переход к следующей цели в резольвенте.
15	! fib(3, Fib_N, 1, 2)	! Результат: да.	Прямой ход. Переход к следующей цели в резольвенте.
16	fib(3, Fib_N, 1, 2)	! Результат: да.	Прямой ход. Переход к следующей цели в резольвенте.
...			
20	пусто	fib(3, Fib_N, 1, 2) = fib(N, Fib_N, Last_N, Last_fib) Результат: унификация успешна. Подстановка: {N = 3, Last_N = 1, Last_fib = 2}	Переход к телу правила. Редукция и подстановка в резольвенту.
21	3 > 3 Temp_fib = 1 + 2 Temp_N = 3 - 1 ! fib(Temp_N, Fib_N, 2, Temp_fib)	3 > 3. Результат: нет.	Обратный ход. Восстановление предыдущего состояния резольвенты (шаг 16). Реконкретизация переменных.
22	fib(3, Fib_N, 1, 2)	fib(3, Fib_N, 1, 2) = fib(_, Temp_fib, _, Temp_fib) Результат: унификация успешна. Подстановка: {Temp_fib = 2, Fib_N = 2}	Сохранение подстановки {Fib_N = 2} в памяти. Обратный ход. Реконкретизация переменных.
...			
24	fib(3, Fib_N, 1, 2)	fib(3, Fib_N, 1, 2) = fib(N, Fib_N) Результат: унификация неуспешна.	Обратный ход. Восстановление предыдущего состояния резольвенты.

Рисунок 1.3 – Таблица порядка поиска ответов для нахождения значения N-ого числа Фибоначчи.

№ 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?)
25	! fib(3, Fib_N, 1, 2)	! Результат: нет	Запрет унификации терма резольвенты. Резольвента пуста. Обратный ход. Восстановление предыдущего состояния резольвенты (шаг 11).
26	fib(4, Fib_N, 1, 1) (резольвента пуста)	Тело пустое	Обратный ход. Восстановление предыдущего состояния резольвенты (шаг 7)
27	fib(4, Fib_elem) (резольвента пуста)	БЗ пуста.	Обратный ход. Резольвента пуста. БЗ просмотрена полностью. Вывод на экран подстановки {Fib_N = 2}

Рисунок 1.4 – Таблица порядка поиска ответов для нахождения значения N-ого числа Фибоначчи (продолжение).

## 2 Контрольные вопросы

1. **Что такое рекурсия? Как организуется хвостовая рекурсия в Prolog? Как организовать выход из рекурсии?**

Рекурсия – ссылка на описываемый объект при описании объекта. Хвостовая рекурсия организовывается следующим образом: сначала выполняются необходимые вычисления, и последним шагом такой «функции» является вызов того же самого объекта. При этом вычисления собираются по мере выхода из рекурсии. Для того, чтобы система не выполняла лишних действий и при этом правильно отработывала, необходимо ставить условия выхода из рекурсии вначале.

2. **Какое первое состояние резольвенты?**

В резольвенте изначально хранится конъюнкция вопросов.

3. **В каком случае система запускает алгоритм унификации? Каково назначение использования алгоритма унификации? Каков результат алгоритма унификации?**

Назначение алгоритма унификации – подбор знаний. Результатом её работы является ответ «да» или «нет», т.е. удалось ли ей подобрать знание или нет.

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

Именованные переменные уникальны в пределах одного предложения, анонимные переменные – уникальные всегда.

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

Подстановка применяется путем конкретизации переменных.

6. **Как изменяется резольвента?**

Резольвента меняется в два этапа.

- (a) Новое состояние приобретается в результате алгоритма редукции.
- (b) К полученному состоянию применяется подстановка.

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

Механизм отката применяется в случае тупиковой ситуации – в ситуации, когда нельзя перейти из данного состояния в новое, и при этом резольвента не пуста.