



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»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №16-18

По курсу: "Функциональное и Логическое программирование"

Группа ИУ7-63Б (ИУ7и-676) _____

Студент Тэмуужин Я. _____

Преподаватель Толпинская Н.Б. _____

Преподаватель Строганов Ю. В. _____

Лабораторная работа 16. Используя хвостовую рекурсию, разработать программу, позволяющую найти

1. $n!$

2. n -е число Фибоначчи.

Убедиться в правильности результатов.

```
PREDICATES
    fact(integer, integer)
    fact(integer, integer, integer)

    fibo(integer, integer)
    fibo(integer, integer, integer, integer)
CLAUSES
    fact(1, F1, F1) :- !.
    fact(X, F1, F) :-
        Xx = X - 1,
        Ff = F1 * X,
        fact(Xx, Ff, F).

    fact(X, F) :- fact(X, 1, F).

    fibo(0, A, _, F) :- F = A, !.
    fibo(1, _, B, F) :- F = B, !.

    fibo(N, A, B, F) :-
        Nn = N - 1,
        Bb = A + B,
        fibo(Nn, B, Bb, F).

    fibo(N, F) :- fibo(N, 0, 1, F).

GOAL
    %fact(4, F). % 24
    fibo(4, F). % 3
```

Вопрос: fact(4, F)

№	Состояние резольвенты	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	fact(4, F)	fact(4, F) = fact(X, F) Успех X = 4, F = F	Прямой ход Тело правила заносится в резольвенту
2	fact(X, 1, F)	fact(4, 1, F) = fact(1, F1, F1) Неудача	Переход к следующему предложению
3	fact(X, 1, F)	fact(4, 1, F) = fact(X, F1, F) Успех X = 4, F1 = 1, F = F	Прямой ход. Тело правила заносится в резольвенту
4	Xx = X - 1, Ff = F1 * X, fact(Xx, Ff, F)	Xx = 4 - 1 = 3	Прямой ход
5	Ff = F1 * X, fact(Xx, Ff, F)	Ff = 1 * 4 = 4	Прямой ход
6	fact(Xx, Ff, F)	fact(3, 4, F) = fact(1, F1, F1) Неудача	Переход к следующему предложению
7	fact(Xx, Ff, F)	fact(3, 4, F) = fact(X, F1, F) Успех X = 3, F1 = 4, F = F	Прямой ход. Тело правила заносится в резольвенту
8	Xx = X - 1, Ff = F1 * X, fact(Xx, Ff, F)	Xx = 3 - 1 = 2	Прямой ход
9	Ff = F1 * X, fact(Xx, Ff, F)	Ff = 4 * 3 = 12	Прямой ход
10	fact(Xx, Ff, F)	fact(2, 12, F) = fact(1, F1, F1) Неудача	Переход к следующему предложению
11	fact(Xx, Ff, F)	fact(2, 12, F) = fact(X, F1, F) Успех X = 2, F1 = 12, F = F	Прямой ход. Тело правила заносится в резольвенту
12	Xx = X - 1, Ff = F1 * X, fact(Xx, Ff, F)	Xx = 2 - 1 = 1	Прямой ход
13	Ff = F1 * X, fact(Xx, Ff, F)	Ff = 12 * 2 = 24	Прямой ход
14	fact(Xx, Ff, F)	fact(1, 24, F) = fact(1, F1, F1) Успех F1 = 24, F1 = F	Прямой ход. Тело правила заносится в резольвенту
15	!	Успех F = F1 = 24	Прямой ход
			Резольвента пуста. Для fact(Xx, Ff, F) попыток найти другие решения не будет. Откат к 1. Других предложение fact с 2 аргументами нет. Система завершит работу.

Вопрос: fibo(4, F)

№	Состояние резольвенты	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	fibo(4, F)	fibo(4, F) = fibo(N, F) Успех N = 4, F = F	Прямой ход. Тело правила заносится в резольвенту
2	fibo(N, 0, 1, F)	fibo(4, 0, 1, F) = fibo(0, A, _, F) Неудача	Переход к следующему предложению
3	fibo(N, 0, 1, F)	fibo(4, 0, 1, F) = fibo(1, _, B, F) Неудача	Переход к следующему предложению
4	fibo(N, 0, 1, F)	fibo(4, 0, 1, F) = fibo(N, A, B, F) Успех N = 4, A = 0, B = 1, F = F	Прямой ход. Тело правила заносится в резольвенту
5	Nn = N - 1, Bb = A + B, fibo(Nn, B, Bb, F)	Nn = 4 - 1 = 3	Прямой ход
6	Bb = A + B, fibo(Nn, B, Bb, F)	Bb = 0 + 1 = 1	Прямой ход
7	fibo(Nn, B, Bb, F)	fibo(3, 1, 1, F) = fibo(0, A, _, F) Неудача	Переход к следующему предложению
8	fibo(Nn, B, Bb, F)	fibo(3, 1, 1, F) = fibo(1, _, B, F) Неудача	Переход к следующему предложению
9	fibo(Nn, B, Bb, F)	fibo(3, 1, 1, F) = fibo(N, A, B, F) Успех N = 3, A = 1, B = 1, F = F	Прямой ход. Тело правила заносится в резольвенту
10	Nn = N - 1, Bb = A + B, fibo(Nn, B, Bb, F)	Nn = 3 - 1 = 2	Прямой ход
11	Bb = A + B, fibo(Nn, B, Bb, F)	Bb = 1 + 1 = 2	Прямой ход
12	fibo(Nn, B, Bb, F)	fibo(2, 1, 2, F) = fibo(0, A, _, F) Неудача	Переход к следующему предложению
13	fibo(Nn, B, Bb, F)	fibo(2, 1, 2, F) = fibo(1, _, B, F) Неудача	Переход к следующему предложению
14	fibo(Nn, B, Bb, F)	fibo(2, 1, 2, F) = fibo(N, A, B, F) Успех N = 2, A = 1, B = 2, F = F	Прямой ход. Тело правила заносится в резольвенту
15	Nn = N - 1, Bb = A + B, fibo(Nn, B, Bb, F)	Nn = 2 - 1 = 1	Прямой ход
16	Bb = A + B, fibo(Nn, B, Bb, F)	Bb = 1 + 2 = 3	Прямой ход
17	fibo(Nn, B, Bb, F)	fibo(1, 2, 3, F) = fibo(0, A, _, F) Неудача	Переход к следующему предложению
18	fibo(Nn, B, Bb, F)	fibo(1, 2, 3, F) = fibo(1, _, B, F) Успех B = 3, F = F	Прямой ход. Тело правила заносится в резольвенту
19	F = B, !	F = 3	Прямой ход
20	!	Успех	Прямой ход
			Резольвента пуста. Для fibo(Nn, B, Bb, F) попыток найти другие решения не будет. Откат к 1. Других предложение fibo с 2 аргументами нет. Система завершит работу.

Лабораторная работа 17-18. Используя хвостовую рекурсию, разработать программу, позволяющую найти

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).

Убедиться в правильности результатов.

```
DOMAINS
    intlist = integer*
PREDICATES
    len(intlist, integer)
    len(intlist, integer, integer)
    sum(intlist, integer)
    sum(intlist, integer, integer)
    odd_pos_sum(intlist, integer)
    odd_pos_sum(intlist, integer, integer)
    lst_more_than(intlist, integer, intlist)
    odd_pos_lst(intlist, intlist)
    del_lst_elems(intlist, integer, intlist)
    del_lst_elem(intlist, integer, intlist)
    set_from_lst(intlist, intlist)
CLAUSES
    len([_ | T], Tmp, L) :-
        Tmpp = Tmp + 1,
        len(T, Tmpp, L).
    len([], Tmp, Tmp).
    len(List, L) :- len(List, 0, L).

    % sum of elements
    sum([H | T], Tmp, S) :-
        Tmpp = Tmp + H,
        sum(T, Tmpp, S).
    sum([], Tmp, Tmp).
    sum(List, S) :- sum(List, 0, S).

    % sum of elements on odd position
    odd_pos_sum([_, H | T], Tmp, S) :-
        Tmpp = Tmp + H,
        odd_pos_sum(T, Tmpp, S).
    odd_pos_sum([_], S, S).
    odd_pos_sum([], S, S).
    odd_pos_sum(List, S) :- odd_pos_sum(List, 0, S).

    % create new list from lst with elements more than Num
    lst_more_than([H | T], Num, [H | Res]) :-
        H > Num, !,
        lst_more_than(T, Num, Res).

    lst_more_than([_ | T], Num, Res) :-
        lst_more_than(T, Num, Res).

    lst_more_than([], _, []).
```

```

% create new list from lst with elements on odd position
odd_pos_lst([_, H | T], [H | Res]) :-
    odd_pos_lst(T, Res).
odd_pos_lst([], []).
odd_pos_lst([], []).

% delete all elements of list that equal Elem
del_lst_elems([Elem | T], Elem, Res) :- !,
    del_lst_elems(T, Elem, Res).
del_lst_elems([H | T], Elem, [H | Res]) :- !,
    del_lst_elems(T, Elem, Res).
del_lst_elems([], _, []).

% delete first instance of Elem in list
del_lst_elem([Elem | T], Elem, T) :- !.
del_lst_elem([_ | T], Elem, Res) :-
    del_lst_elem(T, Elem, Res).
del_lst_elem([], _, []).

% create set from list using del_lst_elem
set_from_lst([H | T], [H | Res]) :-
    del_lst_elems(T, H, CleanT),
    set_from_lst(CleanT, Res).
set_from_lst([], []).

```

GOAL

```

%len([1, 2, 3], L).
%sum([], S).
%odd_pos_sum([], S).
%lst_more_than([8, 9, 3, 4, 7], 5, Res).    % [8, 9, 7]
%odd_pos_lst([9, 8, 7, 6], Res).
%del_lst_elems([1, 2, 3, 1, 1, 7], 1, Res). % [2, 3, 7]
%del_lst_elem([1, 2, 3, 1, 1, 7], 1, Res).  % [2, 3, 1, 1, 7]
set_from_lst2([1, 2, 3, 1, 1, 7], Set).    % [1, 2, 3, 7]

```