



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 №10 по дисциплине "Функциональное и логическое программирование"

Тема Рекурсия в Prolog

Студент Егорова П.А.

Группа ИУ7-64Б

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

Москва — 2023 г.

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:

Т.к. резольвента хранится в виде стека, то состояние резольвенты требуется отображать в столбик: вершина – сверху! Новый шаг надо начинать с нового состояния резольвенты!

Код программы

```
1 domains
2   num = integer
3
4 predicates
5   fact(num, num)
6   rec_fact(num, num, num)
7
8   fib(num, num)
9   rec_fib(num, num, num, num)
10
11 clauses
12   rec_fact(N, Res, Acc) :- N > 1,
13                           NewN = N - 1,
14                           NewAcc = Acc * N,
15                           rec_fact(NewN, Res, NewAcc), !.
16   rec_fact(_, Res, Acc) :- Res = Acc.
17   fact(N, Res) :- rec_fact(N, Res, 1), !.
18
19   rec_fib(N, F1, F2, Res) :- N > 2,
20                             NewF1 = F2,
21                             NewF2 = F1 + F2,
22                             NewN = N - 1,
23                             rec_fib(NewN, NewF1, NewF2, Res), !.
24   rec_fib(_, _, B, Res) :- Res = B.
25   fib(N, Res) :- rec_fib(N, 1, 1, Res).
26
27 goal
28   %fact(5, Res).
29   %fib(7, Res).
```

Вопрос №1

Но 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	fact(3, Res)	rec_fact(N, Res, Acc) и fact(3, Res) Результат: неудача, термы не унифицируемы	Прямой ход. Переход к следующему предложению.
2	fact(3, Res)	rec_fact(_, Res, Acc) и fact(3, Res) Результат: неудача, термы не унифицируемы	Прямой ход. Переход к следующему предложению.
3	fact(3, Res), !	fact(N, Res) и fact(3, Res) Результат: успех + подстановка {N = 3, Res = Res}	Заголовок правила заменяется его телом с учетом подстановки: rec_fact(3, Res, 1)
4	rec_fact(3, Res, 1)	rec_fact(N, Res, Acc) и rec_fact(3, Res, 1) Результат: успех + подстановка {N = 3, Res = Res, Acc = 1}	Заголовок правила заменяется его телом с учетом подстановки: $3 > 1$, NewN = 3 - 1, NewAcc = 1 * 3, rec_fact(NewN, Res, NewAcc), !
5	$3 > 1$, NewN = 3 - 1, NewAcc = 1 * 3, rec_fact(NewN, Res, NewAcc), !	$3 > 1$ Результат: успех	Переход к следующему терму
6	NewN = 3 - 1, NewAcc = 1 * 3, rec_fact(NewN, Res, NewAcc), !	NewN = 3 - 1 Результат: успех + {NewN = 2}	Переход к следующему терму с учётом подстановки NewN = 2
7	NewAcc = 1 * 3, rec_fact(2, Res, NewAcc), !	NewAcc = 1 * 3 Результат: успех	Переход к следующему терму с учётом подстановки NewAcc = 3
8	rec_fact(2, Res, 3), !	rec_fact(2, Res, 3) и rec_fact(N, Res, Acc) Результат: успех + подстановка {N = 2, Res = Res, Acc = 3}	Заголовок правила заменяется его телом с учетом подстановки: $2 > 1$, NewN = 2 - 1, NewAcc = 3 * 2, rec_fact(NewN, Res, NewAcc), !
9	$2 > 1$, NewN = 2 - 1, NewAcc = 3 * 2, rec_fact(NewN, Res, NewAcc), !, !	$2 > 1$ Результат: успех	Переход к следующему терму
6	NewN = 2 - 1, NewAcc = 3 * 2, rec_fact(NewN, Res, NewAcc), !, !	NewN = 2 - 1 Результат: успех + {NewN = 1}	Переход к следующему терму с учётом подстановки NewN = 1

Рис. 1: Таблица к вопросу № 1.

7	NewAcc = 3 * 2, rec_fact(1, Res, NewAcc), !, !	NewAcc = 3 * 2 Результат: успех	Переход к следующему терму с учётом подстановки NewAcc = 6
8	rec_fact(1, Res, 6), !, !	rec_fact(1, Res, 6) и rec_fact(N, Res, Acc) Результат: успех + подстановка {N = 1, Res = Res, Acc = 6}	Заголовок правила заменяется его телом с учетом подстановки: 1 > 1, NewN = 1 - 1, NewAcc = 6 * 1, rec_fact(NewN, Res, NewAcc), !
9	1 > 1, NewN = 1 - 1, NewAcc = 6 * 1, rec_fact(NewN, Res, NewAcc), !, !, !	1 > 1 Результат: неудача	Откат к 8
10	rec_fact(1, Res, 6), !, !	rec_fact(_, Res, Acc) и rec_fact(1, Res, 6) Результат: успех + подстановка {_ = 1, Res = Res, Acc = 6}	Заголовок правила заменяется его телом с учетом подстановки: Res = 6
11	Res = 6, !, !	Res = 6 Результат: успех	Переход к следующему терму
12	!, !	! Результат: успех	Переход к следующему терму
13	!	! Результат: успех	Переход к следующему терму
14			Найдено решение Res=6 Отсечение => система больше не будет искать решение в процедуре rec_fact Откат к 3

Рис. 2: Таблица к вопросу № 1.