



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 № 18

Название: Формирование эффективных программ на Prolog

Дисциплина: Функциональное и логическое программирование

Студент

ИУ7-65Б

(Группа)

Д.В. Сусликов

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

Н.Б. Толпинская

(Подпись, дата)

(И.О. Фамилия)

Москва, 2021

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

Листинг факториала:

```
1 predicates
2     fact(integer , integer , integer ).
3 clauses
4     fact(1, R, R) :- !.
5     fact(N, Acc, R):-
6         New_N = N - 1,
7         New_Acc = Acc * N,
8         fact(New_N, New_Acc, R).
9 goal
10    fact(3, 1, Result).
```

Листинг фибоначчи:

```
1 predicates
2     fibb(integer ,integer ,integer ,integer )
3 clauses
4     fibb(1, _, Res, Res):- !.
5     fibb(I, Last, Cur, Res):-
6         NI = I - 1,
7         NLast = Cur,
8         NCur = Last + Cur,
9         fibb(NI, NLast, NCur, Res).
10 goal
11    fibb(5, 0, 1, R).
```

Результаты работы:

Result=6
1 Solution|

Пример fac

R=5
1 Solution|

Пример fibb

Приведем таблицу для нахождения факториала.

fact(3, 1, R)

№ шага	Состояние резольвенты и вывод	Для каких термов запускается алгоритм унификации и каков результат	Дальнейшие действия
0	fact(3, 1, R)		
1	fact(3, 1, R)	T1 = fact(3, 1, R) T2 = fact(1, _, R). Неудача. Не унифицируемы	Переход к следующему заголовку БЗ
2	fact(3, 1, R)	T1 = fact(3, 1, R) T2 = fact(N, Acc, R) Успех. Унифицируемы. Подстановка: {N = 3, Acc = 1, R = R}	Замена на тело предложения
3	New_N = 3 - 1, New_Acc = 1 * 3, fact(New_N, New_Acc, R).	New_N = 3 - 1 New_N = 2	Замена на тело предложения (пустое)
4	New_Acc = 1 * 3, fact(2, New_Acc, R).	New_Acc = 1 * 3 New_Acc = 3	Замена на тело предложения (пустое)
5	fact(2, 3, R).	T1 = fact(2, 3, R) T2 = fact(1, _, R). Неудача. Не унифицируемы.	Переход к следующему заголовку БЗ
6	fact(2, 3, R).	T1 = fact(2, 3, R) T2 = fact(N, Acc, R) Успех. Унифицируемы. Подстановка: {N = 2, Acc = 3, R = R}	Замена на тело предложения

7	$New_N = 2 - 1,$ $New_Acc = 3 * 2,$ $fact(New_N, New_Acc, R).$	$New_N = 2 - 1$ $New_N = 1$	Замена на тело предложения (пустое)
8	$New_Acc = 3 * 2,$ $fact(1, New_Acc, R).$	$New_Acc = 3 * 2$ $New_Acc = 6$	Замена на тело предложения (пустое)
9	$fact(1, 6, R)$	$T1 = fact(1, 6, R)$ $T2 = fact(1, _, R)$ Успех. Унифицируемы. Подстановка: $\{1 = 1, R = 6, R = R\}$	Замена на тело предложения
10	!	! Истина	Замена на тело предложения (пустое)
11	Вывод $R = 6$ Резольвента пуста		Откат.
12	!	! Завершение процедуры	Замена на тело предложения (пустое)
13	Резольвента пуста		Завершение работы программы

Приведем таблицу для нахождения числа полследовательности Фибоначчи.

fibb(3, 0, 1, Res)

№ шага	Состояние резольвенты и вывод	Для каких термов запускается алгоритм унификации и каков результат	Дальнейшие действия
0	fibb(3, 0, 1, Res)		
1	fibb(3, 0, 1, Res)	T1 = fibb(3, 0, 1, Res) T2 = fibb(1, _, Res, Res) Неудача. Не унифицируемы	Переход к следующему заголовку БЗ
2	fibb(3, 0, 1, Res)	T1 = fibb(3, 0, 1, Res) T2 = fibb(I, Last, Cur, Res) Успех. Унифицируемы. Подстановка: {I = 3, Last = 0, Cur = 1, Res = Res}	Замена на тело предложения
3	NI = 3 - 1, NLast = 1, NCur = 0 + 1, fibb(NI, NLast, NCur, Res).	NI = 3 - 1 NI = 2	Замена на тело предложения (пустое)
4	NLast = 1, NCur = 0 + 1, fibb(2, NLast, NCur, Res)	NLast = 1	Замена на тело предложения (пустое)
5	NCur = 0 + 1, fibb(2, 1, NCur, Res)	NCur = 0 + 1 NCur = 1	Замена на тело предложения (пустое)
6	fibb(2, 1, 1, Res)	T1 = fibb(2, 1, 1, Res) T2 = fibb(1, _, Res, Res) Неудача. Не унифицируемы.	Переход к следующему заголовку БЗ

7	fibb(2, 1, 1, Res)	$T1 = \text{fibb}(2, 1, 1, \text{Res})$ $T2 = \text{fibb}(I, \text{Last}, \text{Cur}, \text{Res})$ Успех. Унифицируемы. Подстановка: $\{I = 2, \text{Last} = 1, \text{Cur} = 1, \text{Res} = \text{Res}\}$	Замена на тело предложения
8	$NI = 2 - 1,$ $N\text{Last} = 1,$ $N\text{Cur} = 1 + 1,$ $\text{fibb}(NI, N\text{Last}, N\text{Cur}, \text{Res}).$	$NI = 2 - 1$ $NI = 1$	Замена на тело предложения (пустое)
9	$N\text{Last} = 1,$ $N\text{Cur} = 1 + 1,$ $\text{fibb}(1, N\text{Last}, N\text{Cur}, \text{Res}).$	$N\text{Last} = 1$	Замена на тело предложения (пустое)
10	$N\text{Cur} = 1 + 1,$ $\text{fibb}(1, 1, N\text{Cur}, \text{Res}).$	$N\text{Cur} = 1 + 1$ $N\text{Cur} = 2$	Замена на тело предложения (пустое)
11	fibb(1, 1, 2, Res)	$T1 = \text{fib}(1, 1, 2, \text{Res})$ $T2 = \text{fib}(1, _, \text{Res}, \text{Res})$ Успех. Унифицируемы. Подстановка: $\{1 = 1, _ = 1, \text{Res} = 2, \text{Res} = \text{Res}\}$	Замена на тело предложения
12	!	! Истина	Замена на тело предложения (пустое)
13	Вывод $\text{Res} = 2$ Резольвента пуста		Откат
14	!	! Завершение процедуры	Замена на тело предложения (пустое)
15	Резольвента пуста		Завершение работы программы

Вывод

Эффективность работы системы может быть достигнута за счет хвостовой рекурсии и использования отсечения в тех случаях, где заведомо известна единственность ответа на вопрос.

Ответы на вопросы

- 1) Что такое рекурсия? Как организуется хвостовая рекурсия в Prolog? Как организовать выход из рекурсии в Prolog?

Рекурсия – определение объекта через ссылку на самого себя. Один из способов организации повторных вычислений. Для организации хвостовой рекурсии необходимо, чтобы рекурсивный вызов был последним в теле рекурсивного правила, и не оставалось других точек выбора. Выход из рекурсии осуществляется либо достижением базиса рекурсии, либо условием в теле правила.

- 2) Какое первое состояние резольвенты?

Исходная резольвента содержит вопрос.

- 3) В каком случае система запускает алгоритм унификации? (Как эту необходимость на формальном уровне распознает система?)

Система запускает алгоритм унификации, когда резольвента не пуста.

- 4) В каких пределах программы уникальны переменные?

Именованные переменные уникальны в рамках предложения, анонимные - уникальны везде.

- 5) Как применяется подстановка, полученная с помощью алгоритма унификации?

В результате подстановки связываются переменные, которые еще не были связаны. После связывания всех утверждений, будет напечатано значение связанных переменных.

- 6) Как формируется новое состояние резольвенты?

Резольвента меняется в 2 этапа:

- Редукция (замена вопроса на тело правила, заголовок которого был успешно унифицирован);
- Применение подстановки.

7) В каких случаях запускается механизм отката?

В случае, когда унификация на текущем шаге завершается тупиковой ситуацией, или был получен ответ «да».