



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 № 8

По дисциплине «Функциональное и логическое программирование»

Студент: Тимонин А. С.

Группа ИУ7-626

Преподаватель Толпинская Н. Б.

Москва.
2020 г.

Практическая часть

Ответить на вопросы:

- Что такое рекурсия? Как организуется хвостовая рекурсия в Prolog? Как организовать выход из рекурсии в Prolog?
- Какое первое состояние резольвенты?
- В каком случае система запускает алгоритм унификации? Каково назначение использования алгоритма унификации? Каков результат работы алгоритма унификации?
- В каких пределах программы переменные уникальны?
- Как применяется подстановка, полученная с помощью алгоритма унификации?
- Как изменяется резольвента?
- В каких случаях запускается механизм отката?

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!

Листинг 1. Реализация программы

```
domains
    num, res = integer.
    n, f = integer.

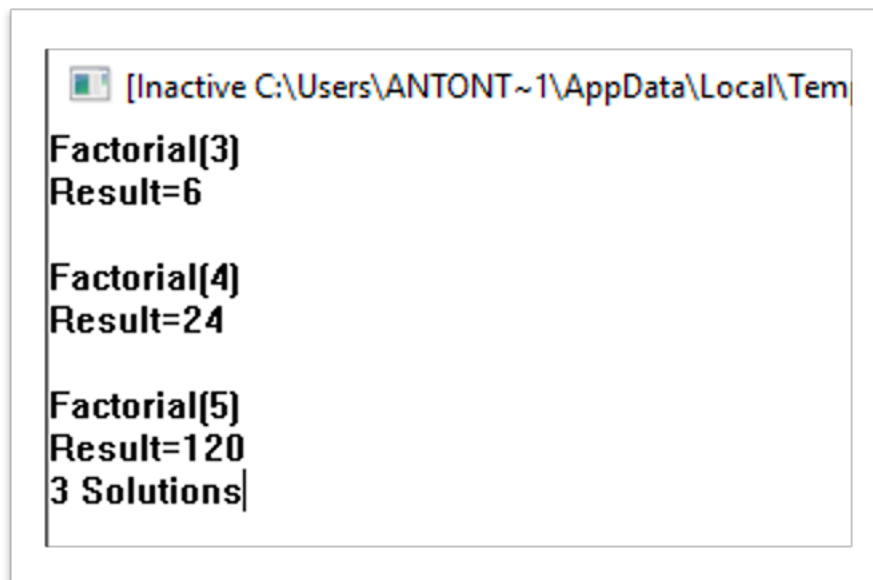
predicates
    nondeterm fact(integer, integer).
    nondeterm fib(integer, integer).
```

clauses

```
fact(0, Result) :- Result = 1, !.  
fact(N, F) :- N1 = N - 1,  
              fact(N1, F1),  
              F = N * F1.  
  
fib(0, Result) :- Result = 0, !.  
fib(1, Result) :- Result = 1, !.  
fib(N, Result) :- N1 = N - 1,  
                  N2 = N - 2,  
                  fib(N1, N1Result),  
                  fib(N2, N2Result),  
                  Result = N1Result + N2Result.
```

goal

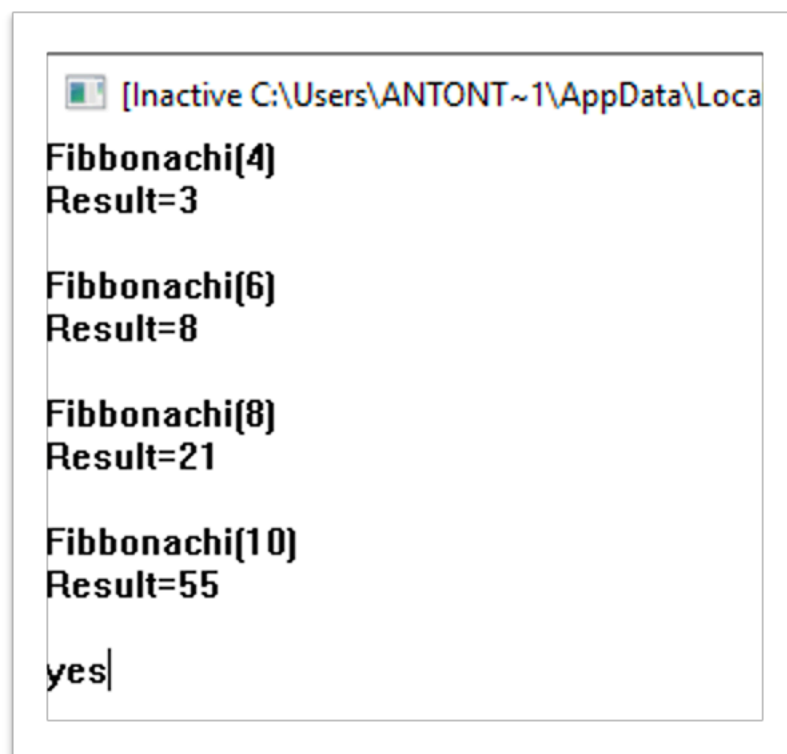
```
%  
% Factorial  
%  
  
write("Factorial(3)\n"),  
fact(3, Result);  
  
write("\nFactorial(4)\n"),  
fact(4, Result);  
  
write("\nFactorial(5)\n"),  
fact(5, Result).  
  
%  
% Fibbonachi  
%  
  
write("Fibbonachi(4)\n"),  
fib(4, Result);  
  
write("\nFibbonachi(6)\n"),  
fib(6, Result);  
  
write("\nFibbonachi(8)\n"),  
fib(8, Result);  
  
write("\nFibbonachi(10)\n"),  
fib(10, Result);  
  
write("\n").
```



A screenshot of a command window with a title bar that reads "[Inactive C:\Users\ANTONT~1\AppData\Local\Tem...". The window contains the following text:

```
Factorial(3)  
Result=6  
  
Factorial(4)  
Result=24  
  
Factorial(5)  
Result=120  
3 Solutions|
```

Рисунок 1. Тестирование fact(integer, integer).



A screenshot of a command window with a title bar that reads "[Inactive C:\Users\ANTONT~1\AppData\Local\Loca...". The window contains the following text:

```
Fibbonachi(4)  
Result=3  
  
Fibbonachi(6)  
Result=8  
  
Fibbonachi(8)  
Result=21  
  
Fibbonachi(10)  
Result=55  
  
yes|
```

Рисунок 2. Тестирование fib(integer, integer).

Формирование ответа

Таблица 1. fact(5, Result)

№ шага	Состояние резольвенты, и вывод: дальнейшие действия (почему?)	Для каких термов запускается алгоритм унификации: T1=T2 и каков результат	Дальнейшие действия: прямой ход или откат
1	fact(5, Result)	Подстановка N = 5, Result = Result	Прямой ход
2	N1 = N - 1, fact(N1, F1), F=N*F1	Подстановка N1 = 4	Прямой ход
3	fact(N1, F1) F=N*F1	Подстановка N1 = 4, F = F1	Прямой ход
4	N1 = N - 1 fact(N1, F1) F=N*F1 F=N*F1	Подстановка N1 = 3	Прямой ход
5	fact(N1, F1) F=N*F1 F=N*F1	Подстановка N1 = 3, F = F1	Прямой ход
6	N1 = N - 1 fact(N1, F1) F=N*F1 F=N*F1 F=N*F1	Подстановка N1 = 2	Прямой ход
7	fact(N1, F1) F=N*F1 F=N*F1 F=N*F1	Подстановка N1 = 2, F = F1	Прямой ход
8	N1 = N - 1 fact(N1, F1) F=N*F1 F=N*F1 F=N*F1 F=N*F1	Подстановка N1 = 1	Прямой ход
9	fact(N1, F1) F=N*F1 F=N*F1 F=N*F1 F=N*F1	Подстановка N1 = 1, F = F1	Прямой ход
10	N1 = N - 1 fact(N1, F1) F=N*F1 F=N*F1 F=N*F1 F=N*F1 F=N*F1	Подстановка N1 = 0	Прямой ход
11	fact(0, F1) F=N*F1 F=N*F1 F=N*F1 F=N*F1 F=N*F1	Подстановка F1 = 1	Прямой ход
12	F=N*F1 F=N*F1 F=N*F1 F=N*F1 F=N*F1	Подстановка F = 1	Прямой ход
13	F=N*F1 F=N*F1 F=N*F1 F=N*F1	Подстановка F = 2	Прямой ход

14	$F=N \cdot F1$ $F=N \cdot F1$ $F=N \cdot F1$	Подстановка $F = 6$	Прямой ход
15	$F=N \cdot F1$ $F=N \cdot F1$	Подстановка $F = 24$	Прямой ход
16	$F=N \cdot F1$	Подстановка $F = 120$	Прямой ход
17	Пусто	Результат $Result = 120$	Откат

Таблица 2. $fib(4, Result)$

№ шага	Состояние резольвенты, и вывод: дальнейшие действия (почему?)	Для каких термов запускается алгоритм унификации: $T1=T2$ и каков результат	Дальнейшие действия: прямой ход или откат
1	$fib(4, Result)$	Подстановка $N = 4, Result = Result$	Прямой ход
2	$N1 = N-1$ $N2 = N-2$ $fib(N1, N1Result)$ $fib(N2, N2Result)$ $Result = N1Result+N2Result$	Подстановка $N1 = 3$	Прямой ход
3	$N2 = N-2$ $fib(N1, N1Result)$ $fib(N2, N2Result)$ $Result = N1Result+N2Result$	Подстановка $N2 = 2$	Прямой ход
4	$fib(N1, N1Result)$ $fib(N2, N2Result)$ $Result = N1Result+N2Result$	Подстановка $N = 3, Result = N1Result$	Прямой ход
5	$N1 = N-1$ $N2 = N-2$ $fib(N1, N1Result)$ $fib(N2, N2Result)$ $Result = N1Result+N2Result$ $fib(N2, N2Result)$ $Result = N1Result+N2Result$	Подстановка $N1 = 2$	Прямой ход
6	$N2 = N-2$ $fib(N1, N1Result)$ $fib(N2, N2Result)$ $Result = N1Result+N2Result$ $fib(N2, N2Result)$ $Result = N1Result+N2Result$	Подстановка $N2 = 1$	Прямой ход
7	$fib(N1, N1Result)$ $fib(N2, N2Result)$ $Result = N1Result+N2Result$ $fib(N2, N2Result)$ $Result = N1Result+N2Result$	Подстановка $N = 2, Result = N1Result$	Прямой ход
8	$N1 = N-1$ $N2 = N-2$ $fib(N1, N1Result)$ $fib(N2, N2Result)$ $Result = N1Result+N2Result$ $fib(N2, N2Result)$ $Result = N1Result+N2Result$ $fib(N2, N2Result)$ $Result = N1Result+N2Result$	Подстановка $N1 = 1$	Прямой ход

9	N2 = N-2 fib(N1, N1Result) fib(N2, N2Result) Result = N1Result+N2Result fib(N2, N2Result) Result = N1Result+N2Result fib(N2, N2Result) Result = N1Result+N2Result	Подстановка N2 = 0	Прямой ход
10	fib(N1, N1Result) fib(N2, N2Result) Result = N1Result+N2Result fib(N2, N2Result) Result = N1Result+N2Result fib(N2, N2Result) Result = N1Result+N2Result	Подстановка N1Result = 1	Прямой ход
11	fib(N2, N2Result) Result = N1Result+N2Result fib(N2, N2Result) Result = N1Result+N2Result fib(N2, N2Result) Result = N1Result+N2Result	Подстановка N2Result = 0	Прямой ход
12	Result = N1Result+N2Result fib(N2, N2Result) Result = N1Result+N2Result fib(N2, N2Result) Result = N1Result+N2Result	Подстановка Result = 1	Прямой ход
13	fib(N2, N2Result) Result = N1Result+N2Result fib(N2, N2Result) Result = N1Result+N2Result	Подстановка N = 2, Result = N2Result	Прямой ход
14	N1 = N-1 N2 = N-2 fib(N1, N1Result) fib(N2, N2Result) Result = N1Result+N2Result Result = N1Result+N2Result fib(N2, N2Result) Result = N1Result+N2Result	Подстановка N1 = 1	Прямой ход
15	N2 = N-2 fib(N1, N1Result) fib(N2, N2Result) Result = N1Result+N2Result Result = N1Result+N2Result fib(N2, N2Result) Result = N1Result+N2Result	Подстановка N2 = 0	Прямой ход
16	fib(N1, N1Result) fib(N2, N2Result) Result = N1Result+N2Result Result = N1Result+N2Result fib(N2, N2Result) Result = N1Result+N2Result	Подстановка N1Result = 1	Прямой ход
17	fib(N2, N2Result) Result = N1Result+N2Result Result = N1Result+N2Result fib(N2, N2Result) Result = N1Result+N2Result	Подстановка N2Result = 0	Прямой ход
18	Result = N1Result+N2Result Result = N1Result+N2Result fib(N2, N2Result) Result = N1Result+N2Result	Подстановка Result = 1	Прямой ход
19	Result = N1Result+N2Result fib(N2, N2Result) Result = N1Result+N2Result	Подстановка Result = 2	Прямой ход
20	fib(N2, N2Result) Result = N1Result+N2Result	Подстановка Result = 0	Прямой ход

21	$\text{Result} = \text{N1Result} + \text{N2Result}$	Подстановка $\text{Result} = 1 + 2 = 3$	Прямой ход
22	Пусто	Результат $\text{Result} = 3$	Откат

Теоретическая часть

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

Рекурсия – это один из способов организации повторных вычислений. Т.к. логическое программирование – не операторное, то рекурсия – это способ заставить систему использовать многократно одну и ту же процедуру. Но этот процесс рано или поздно надо остановить. Поэтому в рекурсивных процедурах должна быть предусмотрена возможность выхода из рекурсии – специальное предложение процедуры. Эффективный способ организации рекурсии – это хвостовая рекурсия. Кроме этого, повысить эффективность рекурсивной процедуры можно отсекая неперспективные пути поиска решения. В целях выхода из рекурсии используется предикат отсечения («!»), который, при необходимости, включают в тело некоторых правил.

Какое первое состояние резольвенты?

Простой вопрос

В каком случае система запускает алгоритм унификации? Каково назначение использования алгоритма унификации? Каков результат работы алгоритма унификации?

Если резольвента не пуста – запускается алгоритм унификации, а если – пуста, это значит, что получен один, однократный ответ «Да» на поставленный вопрос, после чего включается механизм отката, в попытке

найти другое решение с помощью другого знания. При этом, БЗ просматривается сверху вниз.

Алгоритм унификации связывает переменные из вопроса со значениями, с параметрами, которые содержатся в правилах и фактах.

Унификация – операция, которая позволяет формализовать процесс логического вывода. Алгоритм унификации сопоставляет подцель с заданной переменной.

```
1 goal
2     a = man("Anton", Surname).
3     a = man(Name, "Timonin")
```

Например, если в строке 2, в переменной a Name сопоставится с «Anton», а в строке 3, переменная Surname сопоставится с «Timonin». Если бы в строке 2 мы поставили бы фамилию, например, «Konin», тогда строка 3 выдала бы ошибку, так как «Konin» не равно «Timonin». Это связано с тем, что когда параметр в какой-либо переменной занят, он перестает сопоставлять переменные, а начинает их сравнивать.

В каких пределах программы переменные уникальны?

Именованные переменные уникальны в рамках предложения, а анонимная переменная – любая уникальна.

Как применяется подстановка, полученная с помощью алгоритма унификации?

В ходе алгоритма унификации может быть два варианта: унифицируемые термы успешны или нет. Если они успешны:

1. T1 и T2 - одинаковые константы;
2. T1 – не конкретизированная переменная, а T2 - константа или составной терм, не содержащий в качестве аргумента T1. Тогда унификация успешна, причем T1 конкретизируется значением T2.

Тогда применяется подстановка и переменные связываются со значениями.

Как изменяется резольвента?

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

В каких случаях запускается механизм отката?

Когда в программе возможен выбор нескольких вариантов, Пролог заносит в стек точку возврата, для последующего отката по этой точке возврата.

Пролог унифицирует выбранный вариант, если унификация прошла успешно, тогда пролог подготавливает ответ, и далее по точке возврата происходит унификация с другими вариантами. Если пролог не видит дальнейшие варианты, которые он мог бы проунифицировать, тогда по точке возврата программа возвращается на еще более раннюю стадию.