



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 № 18

Дисциплина Функциональное и логическое программирование

Тема Рекурсия на Prolog

Студент Ильясов И. М.

Группа ИУ7-63Б

Оценка (баллы) _____

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

Москва, 2020 г.

Цель работы – изучить рекурсивные способы организации программ на Prolog, методы формирования эффективных рекурсивных программ и порядок реализации таких программ.

Задачи работы:

- приобрести навыки эффективного декларативного описания предметной области с использованием фактов и правил;
- изучить порядок использования фактов и правил в программе на Prolog, принципы и особенности сопоставления и отождествления термов, на основе механизма унификации. Способ формирования и изменения резольвенты. Порядок формирования ответа.

Задание лабораторной работы

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!

Вопрос:...

№ 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...	Комментарий, вывод...
...

Текст программы

predicates

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

clauses

```
fib(1, Res) :- Res = 1, !.  
fib(2, Res) :- Res = 1, !.  
fib(N, Res) :-  
    A = N - 1, B = N - 2,  
    fib(A, A_Res), fib(B, B_Res),  
    Res = A_Res + B_Res.
```

```
fact(0, Res) :- Res = 1, !.  
fact(N, Res) :-  
    Next = N - 1,  
    fact(Next, Next_Res),  
    Res = N * Next_Res.
```

goal

```
write("fib(5) = "),  
fib(5, Res);  
write("3! = "),  
fact(3, Res).
```

Примеры работы программы

1. Поиск $n!$, где $n = 3$.



Рисунок 1. Поиск $3!$

2. Поиск n -ого числа Фибоначчи, где $n = 5$.



Рисунок 2. Поиск 5 числа Фибоначчи.

Задание с таблицей

$\text{fib}(3, \text{Res})$ – поиск n -ого числа Фибоначчи, где $n = 3$.

№ 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	$\text{fib}(3, \text{Res})$	$T1 = \text{fib}(3, \text{Res})$. $T2 = \text{fib}(1, \text{Res})$. Неудача, разные функторы ($3 \neq 1$).	Прямой ход к следующему предложению.
2	$\text{fib}(3, \text{Res})$	$T1 = \text{fib}(3, \text{Res})$. $T2 = \text{fib}(2, \text{Res})$. Неудача, разные функторы ($3 \neq 2$).	Прямой ход к следующему предложению.
3	$\text{fib}(3, \text{Res})$	$T1 = \text{fib}(3, \text{Res})$. $T2 = \text{fib}(N, \text{Res})$. Успех, подстановка $\{N = 3, \text{Res} = \text{Res}\}$.	Прямой ход к следующему предложению.
4	$A = N - 1, B = N - 2,$ $\text{fib}(A, A_Res), \text{fib}(B,$ $B_Res), \text{Res} = A_Res +$ $B_Res.$	$A = 2$	Прямой ход к следующему предложению.
5	$B = N - 2,$ $\text{fib}(A, A_Res), \text{fib}(B,$ $B_Res), \text{Res} = A_Res +$ $B_Res.$	$B = 1$	Прямой ход к следующему предложению.
6	$\text{fib}(A, A_Res), \text{fib}(B,$ $B_Res), \text{Res} = A_Res +$ $B_Res.$	Подстановка $\{N = 2, \text{Res} = A_Res\},$	Прямой ход к следующему предложению.
7	$\text{Res} = 1, \text{fib}(B, B_Res), \text{Res}$ $= A_Res + B_Res.$	Подстановка $\{\text{Res} = 1\}$	Прямой ход к следующему предложению.
8	$\text{fib}(B, B_Res), \text{Res} =$ $A_Res + B_Res.$	Подстановка $\{N = 1, \text{Res} = B_Res\}$	Прямой ход к следующему предложению.
9	$\text{Result} = 1, \text{Res} = A_Res +$ $B_Res.$	Подстановка $\{\text{Res} = 1\}$	Прямой ход к следующему предложению.
10	$\text{Res} = A_Res + B_Res.$	Подстановка $\{\text{Res} = 2\}$	Прямой ход.
11		Результат: $\text{Res} = 3.$	Обратный ход.

fact(3, Res) – поиск $n!$, где $n = 3$.

№ 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)	$T1 = \text{fact}(3, \text{Res}).$ $T2 = \text{fib}(1, \text{Res}).$ Неудача, разные функторы.	Прямой ход к следующему предложению.
2	fact(3, Res)	$T1 = \text{fact}(3, \text{Res}).$ $T2 = \text{fib}(2, \text{Res}).$ Неудача, разные функторы.	Прямой ход к следующему предложению.
3	fact(3, Res)	$T1 = \text{fact}(3, \text{Res}).$ $T2 = \text{fib}(N, \text{Res}).$ Неудача, разные функторы.	Прямой ход к следующему предложению.
4	fact(3, Res)	$T1 = \text{fact}(3, \text{Res}).$ $T2 = \text{fact}(0, \text{Res}).$ Неудача, разные функторы.	Прямой ход к следующему предложению.
5	fact(3, Res)	$T1 = \text{fact}(3, \text{Res}).$ $T2 = \text{fact}(N, \text{Res}).$ Успех, подстановка $\{N = 3, \text{Res} = \text{Res}\}$	Прямой ход к следующему предложению.
6	$\text{Next} = N - 1,$ $\text{fact}(\text{Next}, \text{Next_Res}),$ $\text{Res} = N * \text{Next_Res}.$	Подстановка $\{\text{Next} = 2\}$	Прямой ход к следующему предложению.
7	$\text{fact}(\text{Next}, \text{Next_Res}),$ $\text{Res} = N * \text{Next_Res}.$	Подстановка $\{N = 2, \text{Res} = \text{Next_Res}\}$	Прямой ход к следующему предложению.
8	$\text{Next} = N - 1,$ $\text{fact}(\text{Next}, \text{Next_Res}),$ $\text{Res} = N * \text{Next_Res},$ $\text{Res} = N * \text{Next_Res}.$	Подстановка $\{\text{Next} = 1\}$	Прямой ход к следующему предложению.
6	$\text{fact}(\text{Next}, \text{Next_Res}),$ $\text{Res} = N * \text{Next_Res},$ $\text{Res} = N * \text{Next_Res}.$	Подстановка $\{N = 1, \text{Res} = \text{Next_Res}\}$	Прямой ход к следующему предложению.
7	$\text{Next} = N - 1,$ $\text{fact}(\text{Next}, \text{Next_Res}),$ $\text{Res} = N * \text{Next_Res},$ $\text{Res} = N * \text{Next_Res},$ $\text{Res} = N * \text{Next_Res}.$	Подстановка $\{\text{Next} = 1\}$	Прямой ход к следующему предложению.
8	$\text{fact}(\text{Next}, \text{Next_Res}),$ $\text{Res} = N * \text{Next_Res},$ $\text{Res} = N * \text{Next_Res},$ $\text{Res} = N * \text{Next_Res}.$	Подстановка $\{N = 0, \text{Res} = \text{Next_Res}\}$	Прямой ход.

9	Res = 1, Res = N * Next_Res, Res = N * Next_Res, Res = N * Next_Res.	Подстановка {Res = 1}	Прямой ход к следующему предложению.
10	Res = N * Next_Res, Res = N * Next_Res, Res = N * Next_Res.	Подстановка {Res = 1}	Прямой ход.
11	Res = N * Next_Res, Res = N * Next_Res.	Подстановка {Res = 2}	Прямой ход.
12	Res = N * Next_Res.	Подстановка {Res = 6}	Прямой ход.
13		Результат: Res = 6	Обратный ход.

Ответы на вопросы

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

Рекурсия – это ссылка на определяемый объект во время его определения. В Prolog рекурсия организуется правилом, в котором есть обращение к тому же правилу. Организовать выход из рекурсии можно при помощи отсечения.

2) Какое первое состояние резольвенты?

Если задан простой вопрос, то сначала он попадает в резольвенту.

3) В каком случае система запускает алгоритм унификации? (Как эту необходимость на формальном уровне распознает система?) Каково назначение использования алгоритма унификации? Каков результат использования алгоритма унификации?

Если есть что доказывать (цель), то процесс унификации запускается автоматически. Формально: если резольвента не пуста – запускается алгоритм унификации.

Назначение алгоритма унификации заключается в попарном сопоставлении термов и попытке построить для них общий пример.

Унификация может завершаться успехом или тупиковой ситуацией (неудачей).

4) В каких пределах программы уникальны переменные?

Переменные уникальны в пределах предложения, т.е. в рамках предложения одно и то же имя принадлежит одной и той же переменной. Исключение – анонимные переменные (обозначаются символом нижнего подчеркивания «_») – каждая такая переменная является отдельной сущностью и применяется, когда ее значение неважно для данного предложения.

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

Применение подстановки $\{X1=T1, ..., Xn=Тn\}$ заключается в замене каждого вхождения переменной Xi на соответствующий терм Ti .

6) Как меняется резольвента?

Изменение резольвенты происходит в 2 этапа:

- 1) из стека выбирается подцель (верхняя, т.к. стек) и для нее выполняется редукция, т.е. замена подцели на тело найденного правила;
- 2) к полученной конъюнкции целей применяется подстановка (наибольший общий унификатор выбранной цели и заголовка сопоставленного с этой целью правила).

7) В каких случаях запускается механизм отката?

Механизм отката запускается в 2 случаях:

1. Если алгоритм попал в тупиковую ситуацию.
2. Если резольвента не пуста и решение найдено, но в базе знания остались не отмеченные предложения.