|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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*

*По предмету: «Функциональное и логическое программирование»*

Преподаватель: Строганов Ю.В.

Студент: Мирзоян С.А.,

Группа: ИУ7-65Б

Москва, 2020 г.

# Задание

Используя хвостовую рекурсию, разработать программу, позволяющую найти

* n!,
* 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!

# Теоретические вопросы

**Что такое рекурсия?**

Рекурсия – это ссылка на описываемый объект при описании объекта.

**Как организуется хвостовая рекурсия в Prolog?**

* Рекурсивный вызов единственен и расположен в конце тела правила
* Не должно быть возможности сделать откат до вычисления рекурсивного вызова

**Как организовать выход из рекурсии в Prolog?**

С помощью отсечения.

**Какое первое состояние резольвенты?**

Исходный вопрос.

**В каком случае система запускает алгоритм унификации?**

Если резольвента не пуста (при необходимости что-то доказать).

**Каково назначение и результат использования алгоритма унификации?**

Унификация – механизм логического вывода. Результат – подстановка.

**В каких пределах программы переменные уникальны?**

Именованная переменная уникальна в рамках предложения, в котором она используется. Анонимные переменные всегда уникальны.

**Как применяется подстановка, полученная с помощью алгоритма унификации?**

Подстановка применяется к целям в резольвенте путем замены текущей переменной на соответствующий терм.

**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. Новая резольвента образуется в два этапа:

1. в текущей резольвенте выбирается одна из подцелей и для неё выполняется редукция;
2. к полученной конъюнкции целей применяется подстановка, полученная как наибольший общий унификатор цели и заголовка сопоставленного с ней правила.

**В каких случаях запускается механизм отката?**

Механизм отката запустится в случае неудачи алгоритма унификации.

# Листинг

Факториал n!

1. predicates
2. factorial(integer, integer).
3. factorial(integer, integer, integer).
5. clauses
6. factorial(0, 1) :**-** !.
7. factorial(N, 0) :**-** N < 0, !. **%** error
8. factorial(N, Res) :**-** factorial(N, 1, Res).
10. factorial(1, Res, Res) :**-** !.
11. factorial(N, I, Res) :**-**
12. CurN **=** N **-** 1,
13. Tmp **=** I **\*** N,
14. factorial(CurN, Tmp, Res).
16. goal
17. **%**factorial(**-**1, Res).
18. **%**factorial(0, Res).
19. **%**factorial(4, Res).
20. %factorial(2, Res).

n-oe исло Фиббоначи

1. predicates
2. fib(integer, integer). **%**target, res
3. fib(integer, integer, integer, integer, integer). **%**targer, current, prevRes, prevRes2, res
4. minus(integer, integer).
6. clauses
7. minus(N, 1) :**-** (N mod 2) **=** 1, !.
8. minus(\_, **-**1).
10. fib(0, 0) :**-** !.
11. fib(1, 1) :**-** !.
12. fib(N, Result) :**-**
13. N < 0,
14. RevN **=** **-**1 **\*** N,
15. fib(RevN, PrevRes),
16. minus(RevN, Minus),
17. Result **=** PrevRes **\*** Minus, !.
18. fib(N, Result) :**-** fib(N, 2, 0, 1, Result).
20. fib(N, N, Prev1, Prev2, Result) :**-** Result **=** Prev1 **+** Prev2, !.
21. fib(N, CurN, Prev1, Prev2, Result) :**-**
22. NewN **=** CurN **+** 1,
23. Next **=** Prev1 **+** Prev2,
24. fib(N, NewN, Prev2, Next, Result).


28. goal
29. **%**fib(**-**4, Result).**%** **=** **-**3
30. **%**fib(**-**3, Result).**%** **=** 2
31. **%**fib(**-**2, Result).**%** **=** **-**1
32. **%**fib(2, Result). **%** **=** 1
33. **%**fib(3, Result). **%** **=** 2
34. **%**fib(10, Result).**%** **=** 55

Эффективность достигнута за счет использования хвостовой рекурсии и использования отсечения.

# Таблицы

factorial(2, Res).

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Состояние резольвенты, и вывод: дальнейшие действия (почему?) | Для каких термов запускается алгоритм унификации: Т1=Т2 и каков **результат** (и подстановка) | Дальнейшие действия: прямой ход или откат (почему и к чему приводит?) |
| 1 | factorial(2, Res). | factorial(2, Res) = factorial(N, -1)  Успех  N = 2 Res = -1 | Прямой ход.  В резольвенту заносится тело правила. |
| 2 | N < 0 ! | Сравнение:  2 < 0 Ложь | Откат к 1.  Переход к следующему предложению |
| 3 | factorial(2, Res). | factorial(2, Res) = factorial(0, 1)  Неудача | Переход к следующему предложению |
| 4 | factorial(2, Res). | factorial(2, Res) = factorial(N, Res)  Успех  N = 2 Res = Res | Прямой ход.  Тело правила заносится в резольвенту. |
| 5 | factorial(N, 1, Res) | factorial(2, 1, Res) = factorial(1, Res, Res)  Неудача | Переход к следующему предложению |
| 6 | factorial(N, 1, Res) | factorial(2, 1, Res) = factorial(N, Cur, Res)  Успех  N = 2 Cur = 1 Res = Res | Прямой ход.  Тело правила заносится в резольвенту. |
| 7 | NewN = N – 1 NewMult = Cur \* N  factorial(NewN, NewMult, Res) | NewN = 2 – 1 = 1 | Прямой ход. |
| 8 | NewMult = Cur \* N  factorial(NewN, NewMult, Res) | NewMult = 1 \* 2 = 2 | Прямой ход. |
| 9 | factorial(NewN, NewMult, Res) | factorial(1, 2, Res) = factorial(1, Res, Res)  Успех  Res = Res = 2 | Прямой ход. |
| 10 |  |  | Резольвента пуста.  Res = 2  Откат к 4. Конец процедуры factorial арности 2.  Система завершает работу. |

**Вопрос: fib(2, Result).**

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Состояние резольвенты, и вывод: дальнейшие действия (почему?) | Для каких термов запускается алгоритм унификации: Т1=Т2 и каков **результат** (и подстановка) | Дальнейшие действия: прямой ход или откат (почему и к чему приводит?) |
| 1 | fib(2, Result) | fib(2, Result) = fib(0, 0)  Неудача | Переход к следующему предложению |
| 2 | fib(2, Result) | fib(2, Result) = fib(1, 1)  Неудача | Переход к следующему предложению |
| 3 | fib(2, Result) | fib(2, Result) = fib(N, Result)  Успех  N = 2 Res = Res | Прямой ход.  Тело правила заносится в резольвенту. |
| 4 | N < 0  RevN = -1 \* N fib(RevN, PrevRes) sign(RevN, Sign), Res = PreRes \* Sign ! | Сравнение:  2 < 0  Ложь | Откат к 3.  Переход к следующему предложению |
| 5 | fib(2, Result) | fib(2, Result) = fib(N, Result)  Успех  N = 2 Result = Result | Прямой ход.  Тело правила заносится в резольвенту. |
| 6 | fib(N, 2, 0, 1, Result) | fib(2, 2, 0, 1, Result) = fib(N, N, Prev1, Prev2, Result)  Успех  Prev1 = 0 Prev2 = 1 Result = Result | Прямой ход.  Тело правила заносится в резольвенту. |
| 7 | Res = Prev1 + Prev2 ! | Result = 0 + 1 = 1 | Прямой ход. |
| 8 | ! |  | Найдено решение.  Result = 1  Отсечение нивелирует попытки найти другие решения fib(N, 2, 0, 1, Result).  Откат к 5. Конец fib арности равной 2.  Система завершает свою работу. |