|  |  |
| --- | --- |
| 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: Гасанзаде М.А.,

Группа: ИУ7-66Б

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!

# 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?**

Заданный вопрос (goal).

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

Система запускает алгоритм унификации автоматически 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

predicates

factorial(**integer**, **integer**).

factorial(**integer**, **integer**, **integer**).

clauses

factorial(N, -1) :- N < 0, !. % error

factorial(0, 1) :- !.

factorial(N, Res) :- factorial(N, 1, Res).

factorial(1, Res, Res) :- !.

factorial(N, Cur, Res) :-

NewN = N - 1,

NewMult = Cur \* N,

factorial(NewN, NewMult, Res).

goal

%factorial(-10, Res).

%factorial(0, Res).

%factorial(5, Res).

predicates

fib(**integer**, **integer**). %target, res

fib(**integer**, **integer**, **integer**, **integer**, **integer**). %targer, current, prevRes, prevRes2, res

sign(**integer**, **integer**).

clauses

fib(0, 0) :- !.

fib(1, 1) :- !.

fib(N, Res) :-

N < 0,

NormalN = N \* -1,

fib(NormalN, PreRes),

sign(NormalN, Sign),

Res = PreRes \* Sign, !.

fib(N, Res) :- fib(N, 2, 0, 1, Res).

fib(N, N, Prev1, Prev2, Res) :- Res = Prev1 + Prev2, !.

fib(N, CurN, Prev1, Prev2, Res) :-

NewN = CurN + 1,

Next = Prev1 + Prev2,

fib(N, NewN, Prev2, Next, Res).

sign(N, 1) :- (N mod 2) = 1, !.

sign(\_, -1).

goal

%fib(-9, Res). % 34

%fib(-8, Res). % -21

%fib(-2, Res). % -1

%fib(2, Res). % 1

%fib(8, Res). % 21

Эффективность достигнута за счет использования хвостовой рекурсии и использования отсечения.

# ТАБЛИЦА

**Вопрос: 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.  Тело правила заносится в резольвенту. |
| 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, 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 | fib(2, Res) | fib(2, Res) = fib(0, 0)  Неудача | Переход к следующему предложению |
| 2 | fib(2, Res) | fib(2, Res) = fib(1, 1)  Неудача | Переход к следующему предложению |
| 3 | fib(2, Res) | fib(2, Res) = fib(N, Res)  Успех  N = 2 Res = Res | Прямой ход.  Тело правила заносится в резольвенту. |
| 4 | N < 0  NormalN = N \* -1 fib(NormalN, PreRes) sign(NormalN, Sign), Res = PreRes \* Sign ! | Сравнение:  2 < 0  Ложь | Откат к 3.  Переход к следующему предложению |
| 5 | fib(2, Res) | fib(2, Res) = fib(N, Res)  Успех  N = 2 Res = Res | Прямой ход.  Тело правила заносится в резольвенту. |
| 6 | fib(N, 2, 0, 1, Res) | fib(2, 2, 0, 1, Res) = fib(N, N, Prev1, Prev2, Res)  Успех  Prev1 = 0 Prev2 = 1 Res = Res | Прямой ход.  Тело правила заносится в резольвенту. |
| 7 | Res = Prev1 + Prev2 ! | Res = 0 + 1 = 1 | Прямой ход. |
| 8 | ! |  | Найдено решение.  Res = 1  Ввиду отсечения не будет попыток найти другие решения fib(N, 2, 0, 1, Res).  Откат к 5. Конец fib арности 2.  Система завершит свою работу. |