|  |  |
| --- | --- |
| 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»

**ОТЧЕТ**

*к лабораторной работе №18-20*

*По курсу: «Функциональное и логическое программирование»*

**Темы:** «Рекурсия на Prolog», «Обработка списков на Prolog», «Формирование и модификация списков на Prolog»

Студент: Якуба Д.В.

Группа: ИУ7-63Б

Преподаватели: Толпинская Н. Б.,

Строганов Ю. В.

Москва, 2021 г.

# Практическая часть

# Лабораторная работа 18.

Задание. **Используя хвостовую рекурсию, разработать программу, позволяющую найти:**

1. **n!**,
2. **n-e** 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!

|  |
| --- |
| predicates  factorial(integer, integer).  factorialInner(integer, integer, integer).  fibbo(integer, integer).  fibboInner(integer, integer, integer, integer).  clauses  factorialInner(Number, Answer, NewAnswer) :- Number <= 1, Answer = NewAnswer, !.  factorialInner(Number, Answer, NewAnswer) :-  TempAnswer = NewAnswer \* Number,  TempNumber = Number - 1,  factorialInner(TempNumber, Answer, TempAnswer).  factorial(Number, Answer) :- factorialInner(Number, Answer, 1).  fibboInner(Number, Answer, PrevAnswer, \_) :- Number < 2, Answer = PrevAnswer, !.  fibboInner(Number, Answer, PrevPrevAnswer, PrevAnswer) :-  TempNumber = Number - 1,  NewPrev = PrevPrevAnswer + PrevAnswer,  fibboInner(TempNumber, Answer, PrevAnswer, NewPrev).  fibbo(Number, Answer) :- fibboInner(Number, Answer, 1, 1).  goal  factorial(1, Answer\_); % 1  factorial(2, Answer\_); % 2  factorial(3, Answer\_); % 6  factorial(4, Answer\_); % 24  factorial(5, Answer\_); % 120  factorial(6, Answer\_); % 720  factorial(7, Answer\_); % 5040  factorial(8, Answer\_); % 40320  fibbo(1, Answer); % 1  fibbo(2, Answer); % 1  fibbo(3, Answer); % 2  fibbo(4, Answer); % 3  fibbo(5, Answer); % 5  fibbo(6, Answer); % 8  fibbo(7, Answer); % 13  fibbo(8, Answer) % 21  . |

Вопрос: factorial(3, Answer\_);

|  |  |  |  |
| --- | --- | --- | --- |
| № 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 (к чему приводит?) |
| 1 | factorial(3, Answer\_)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | factorial(3, Answer\_) = factorialInner(Number, Answer, NewAnswer) :- Number <= 0, Answer = NewAnswer, !  Унификация неуспешна (несовпадение функторов) | Откат, переход к следующему предложению |
| 2 | … | … | … |
| 3 | factorial(3, Answer\_)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | factorial(3, Answer\_) = factorial(Number, Answer) :- factorialInner(Number, Answer, 1)  Унификация успешна  Подстановка: {Number = 3, Answer = Answer\_} | Формируется новое состояние резольвенты. Прямой ход |
| 4 | factorialInner(3, Answer\_, 1)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | factorialInner(3, Answer\_, 1) = factorialInner(Number, Answer, NewAnswer) :- Number <= 1, Answer = NewAnswer, !  Унификация успешна  Подстановка:  {Number = 3, Answer = Answer\_, NewAnswer = 1} | Формируется новое состояние резольвенты. Прямой ход |
| 5 | 3 <= 1,  Answer\_ = 1,  !  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | Сравнение констант:  3 <= 1  Результат: ложь | Откат к предыдущему состоянию резольвенты  Новая подстановка: {Number = 3, Answer = Answer\_} |
| 6 | factorialInner(3, Answer\_, 1)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | factorialInner(3, Answer\_, 1) = factorialInner(Number, Answer, NewAnswer) :- TempAnswer = NewAnswer \* Number, TempNumber = Number - 1, factorialInner(TempNumber, Answer, TempAnswer)  Унификация успешна  Подстановка: {Number = 3, Answer = Answer\_, NewAnswer = 1} | Формируется новое состояние резольвенты. Прямой ход |
| 7 | TempAnswer = 1 \* 3,  TempNumber = 3 - 1,  factorialInner(TempNumber, Answer\_, TempAnswer)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | Унификация:  TempAnswer = 1 \* 3  Успех. Подстановка: {TempAnswer = 1 \* 3} | Формируется новое состояние резольвенты. Прямой ход |
| 8 | TempNumber = 3 - 1,  factorialInner(TempNumber, Answer\_, 3)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | Унификация:  TempNumber = 3 – 1  Успех. Подстановка: {TempNumber = 2} | Формируется новое состояние резольвенты. Прямой ход |
| 9 | factorialInner(2, Answer\_, 3)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | factorialInner(2, Answer\_, 3) = factorialInner(Number, Answer, NewAnswer) :- Number <= 1, Answer = NewAnswer, !.  Унификация успешна  Подстановка: {Number = 2, Answer = Answer\_, NewAnswer = 3} | Формируется новое состояние резольвенты. Прямой ход |
| 10 | 2 <= 1,  Answer\_ = 3,  !  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | Сравнение констант:  2 <= 1  Результат: ложь | Откат к предыдущему состоянию резольвенты  Новая подстановка: Подстановка: {Number = 2, Answer = Answer\_, NewAnswer = 3} |
| 11 | factorialInner(2, Answer\_, 3)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | factorialInner(2, Answer\_, 3) = factorialInner(Number, Answer, NewAnswer) :- TempAnswer = NewAnswer \* Number, TempNumber = Number - 1, factorialInner(TempNumber, Answer, TempAnswer)  Унификация успешна  Подстановка: {Number = 2, Answer = Answer\_, NewAnswer = 3} | Формируется новое состояние резольвенты. Прямой ход |
| 12 | TempAnswer = 3 \* 2,  TempNumber = 2 - 1,  factorialInner(TempNumber, Answer, TempAnswer)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | Унификация:  TempAnswer = 3 \* 2  Успех. Подстановка: {TempAnswer = 6} | Формируется новое состояние резольвенты. Прямой ход |
| 13 | TempNumber = 2 - 1,  factorialInner(TempNumber, Answer, 6)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | Унификация:  TempNumber = 2 – 1  Успех. Подстановка: {TempNumber = 1} | Формируется новое состояние резольвенты. Прямой ход |
| 14 | factorialInner(1, Answer\_, 6)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | factorialInner(1, Answer\_, 6) = factorialInner(Number, Answer, NewAnswer) :- Number <= 1, Answer = NewAnswer, !.  Унификация успешна  Подстановка: {Number = 1, Answer = Answer\_, NewAnswer = 6} | Формируется новое состояние резольвенты. Прямой ход |
| 15 | 1 ≤ 1,  Answer\_ = 6,  !  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | Сравнение констант:  1 <= 1  Результат: истина | Формируется новое состояние резольвенты. Прямой ход |
| 16 | Answer\_ = 6,  !  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | Унификация:  Answer\_ = 6  Подстановка: {Answer\_ = 6} | Формируется новое состояние резольвенты. Прямой ход |
| 17 | !  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | Встречен системный предикат отсечения | Резольвента пуста.  Вывод: Answer\_ = 6  Откат с отсечением остаточных предложений процедуры относительно шага 14  Новая подстановка: {Number = 2, Answer = Answer\_, NewAnswer = 3} |
| 18 | factorialInner(1, Answer\_, 6)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | factorialInner(1, Answer\_, 6) = factorial(Number, Answer) :- factorialInner(Number, Answer, 1).  Унификация неуспешна (несовпадение функторов) | Откат, следующее предложение |
| 19-20 | … | … | … |
| 21 | factorialInner(1, Answer\_, 6)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | factorialInner(1, Answer\_, 6) = fibbo(Number, Answer) :- fibboInner(Number, Answer, 1, 1)  Унификация неуспешна (несовпадение функторов) | Конец БЗ достигнут. Откат, следующее предложение относительно пункта 11  Новая подстановка: {Number = 3, Answer = Answer\_, NewAnswer = 1} |
| 22 | factorialInner(2, Answer\_, 3)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | factorialInner(2, Answer\_, 3) = factorial(Number, Answer) :- factorialInner(Number, Answer, 1)  Унификация неуспешна (несовпадение функторов) | Откат, следующее предложение |
| 23-24 | … | … | … |
| 25 | factorialInner(2, Answer\_, 3)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | factorialInner(2, Answer\_, 3) = fibbo(Number, Answer) :- fibboInner(Number, Answer, 1, 1)  Унификация неуспешна (несовпадение функторов) | Конец БЗ достигнут. Откат, следующее предложение относительно пункта 6  Новая подстановка: {Number = 3, Answer = Answer\_, NewAnswer = 1} |
| 26 | factorialInner(3, Answer\_, 1)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | factorialInner(3, Answer\_, 1) = factorial(Number, Answer) :- factorialInner(Number, Answer, 1)  Унификация неуспешна (несовпадение функторов) | Откат, следующее предложение |
| 27-28 | … | … | … |
| 29 | factorialInner(3, Answer\_, 1)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | factorialInner(3, Answer\_, 1) = fibbo(Number, Answer) :- fibboInner(Number, Answer, 1, 1)  Унификация неуспешна (несовпадение функторов) | Конец БЗ достигнут. Откат, следующее предложение относительно пункта 3  Подстановки более нет |
| 30 | factorial(3, Answer\_)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | factorial(3, Answer\_) = fibboInner(Number, Answer, PrevAnswer, \_) :- Number < 2, Answer = PrevAnswer, !  Унификация неуспешна (несовпадение функторов) | Откат, следующее предложение |
| 31 | … | … | … |
| 32 | factorial(3, Answer\_)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | factorial(3, Answer\_) = fibbo(Number, Answer) :- fibboInner(Number, Answer, 1, 1)  Унификация неуспешна (несовпадение функторов) | Конец БЗ достигнут, завершение работы системы. |

# Лабораторная работа 19.

Задание. **Используя хвостовую рекурсию, разработать эффективную программу (*комментируя назначение аргументов*), позволяющую:**

1. Найти длину списка (по верхнему уровню);
2. Найти сумму элементов числового списка;
3. Найти сумму элементов числового списка, стоящих на нечетных позициях исходного списка (нумерация от 0);

Убедиться в правильности результатов.

Для одного из вариантов ВОПРОСА и одного из заданий 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! Для каждого запуска алгоритма унификации, требуется указать № выбранного правила и дальнейшие действия – и почему.

|  |
| --- |
|  |

Текст процедуры:

Вопрос:

|  |  |  |  |
| --- | --- | --- | --- |
| № 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 (к чему приводит?) |
| 1 | maxOfThree(1.1, 1.2, 1.3, Max)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | maxOfThree(1.1, 1.2, 1.3, Max)  = maxOfTwo(FVar, SVar, FVar)  Унификация неуспешна (несовпадение функторов) | Откат, переход к следующему предложению |

# Лабораторная работа 20.

Задание. **Используя хвостовую рекурсию, разработать, комментируя аргументы, эффективную программу, позволяющую:**

1. Сформировать список из элементов числового списка, больших заданного значения;
2. Сформировать список из элементов, стоящих на нечетных позициях исходного списка (нумерация от 0):
3. Удалить заданный элемент из списка (один или все вхождения);
4. Преобразовать список в множество (можно использовать ранее разработанные процедуры).

Убедиться в правильности результатов.

Для одного из вариантов ВОПРОСА и 1-го 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! Для каждого хапуска алгоритма унификации, требуется указать № выбранного правила и соответствующий вывод: успех или нет – и почему.

Текст процедуры:

Вопрос:

|  |  |  |  |
| --- | --- | --- | --- |
| № 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 (к чему приводит?) |
| 1 | maxOfThree(1.1, 1.2, 1.3, Max)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | maxOfThree(1.1, 1.2, 1.3, Max)  = maxOfTwo(FVar, SVar, FVar)  Унификация неуспешна (несовпадение функторов) | Откат, переход к следующему предложению |