|  |  |
| --- | --- |
| 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» (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 Программная инженерия**

**ОТЧЕТ**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 18 |
|  |  |

**Дисциплина:** Функциональное и логическое программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-62Б |  |  | И. В. Козлова |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
| Преподаватель |  |  |  | Н.Б.Толпинская |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |
| Преподаватель |  |  |  | Ю.В.Строганов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Лабораторная работа 18**

**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.

**Код программы:**

|  |
| --- |
| domains  list = integer\*.  predicates  append(list, list, list).  list\_upper\_num(list, integer, list).  list\_odd\_pos(list, list).  delete\_one\_elem(list, integer, list).  delete\_all\_elem(list, integer, list).  members(list, integer).  make\_set(list, list).  clauses  append([H|T], List\_added, [H|Add\_tail]) :-  append(T, List\_added, Add\_tail),  !.  append([], List\_added, List\_added).  list\_upper\_num([H|T], Num, [H|Add\_tail]) :-  H > Num,  !,  list\_upper\_num(T, Num, Add\_tail).  list\_upper\_num([\_|T], Num, Res\_list) :- list\_upper\_num(T, Num, Res\_list), !.  list\_upper\_num([], \_, []).  list\_odd\_pos([\_, H|T], [H|Res\_list]) :-  list\_odd\_pos(T, Res\_list),  !.  list\_odd\_pos(List, List).  delete\_one\_elem([H|T], El, [H|Add\_tail]) :-  El <> H,  !,  delete\_one\_elem(T, El, Add\_tail).  delete\_one\_elem([H|T], H, T) :- !.  delete\_one\_elem(List, \_, List).  delete\_all\_elem([H|T], El, Add\_tail) :-  El = H,  !,  delete\_all\_elem(T, El, Add\_tail).  delete\_all\_elem([H|T], El, [H|Add\_tail]) :-  delete\_all\_elem(T, El, Add\_tail),  !.  delete\_all\_elem(List, \_, List).  members([\_|T], El) :-  members(T, El),  !.  members([H|\_], H).  make\_set([H|T], Add\_tail) :-  members(T, H),  !,  make\_set(T, Add\_tail).  make\_set([H|T], [H|Add\_tail]) :- make\_set(T, Add\_tail), !.  make\_set(List, List).  goal  %make\_set([1, 2, 3, 2, 4, 1, 5, 6, 6, 6, 2, 3], Res\_list\_).  % Res\_list\_=[4,1,5,6,2,3]    %delete\_one\_elem([1, 2, 3, 1, 4, 7, 6], 1, Res\_list\_).  % Res\_list\_=[2,3,1,4,7,6]    %delete\_all\_elem([1, 2, 3, 1, 4, 7, 6], 1, Res\_list\_).  % Res\_list\_=[2,3,4,7,6]    list\_odd\_pos([1, 2, 3, 4, 5, 6], Res\_list\_).  % Res\_list\_=[2,4,6]    %append([1, 2, 3], [4, 5, 6], Res\_list\_).  % Res\_list\_=[1,2,3,4,5,6]    %list\_upper\_num([1, 4, 5], 4, Res\_list\_).  % Res\_list\_=[5] |

Вопрос list\_upper\_num([1, 4, 5], 4, Res\_list\_).

|  |  |  |  |
| --- | --- | --- | --- |
| № 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 | list\_upper\_num([1, 4, 5], 4, Res\_list\_)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | list\_upper\_num([1, 4, 5], 4, Res\_list\_) = append([H|T], List\_added, [H|Add\_tail]) :- append(T, List\_added, Add\_tail), !  Унификация неуспешна (несовпадение функторов) | Откат, переход к следующему предложению |
| 2 | … | … | … |
| 3 | list\_upper\_num([1, 4, 5], 4, Res\_list\_)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | list\_upper\_num([1, 4, 5], 4, Res\_list\_) = list\_upper\_num([H|T], Num, [H|Add\_tail]) :- H > Num, !, list\_upper\_num(T, Num, Add\_tail)  Унификация успешна  Подстановка: {H = 1, T = [4, 5], Num = 4, [H|Add\_tail] = Res\_list\_} | Формируется новое состояние резольвенты, прямой ход |
| 4 | 1 > 4,  !,  list\_upper\_num([4, 5], 4, Add\_tail)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | Унификация:  1 > 4  Результат: ложь | Откат, возврат к предыдущему состоянию резольвенты, переход к следующему предложению |
| 5 | list\_upper\_num([1, 4, 5], 4, Res\_list\_)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | list\_upper\_num([1, 4, 5], 4, Res\_list\_) = list\_upper\_num([\_|T], Num, Res\_list) :- list\_upper\_num(T, Num, Res\_list), !  Унификация успешна  Подстановка: {T = [4, 5], Num = 4, Res\_list = Res\_list\_} | Формируется новое состояние резольвенты, прямой ход |
| 6 | list\_upper\_num([4, 5], 4, Res\_list\_),  !  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | list\_upper\_num([4, 5], 4, Res\_list\_) = append([H|T], List\_added, [H|Add\_tail]) :- append(T, List\_added, Add\_tail), !  Унификация неуспешна (несовпадение функторов) | Откат, переход к следующему предложению |
| 7 | … | … | … |
| 8 | list\_upper\_num([4, 5], 4, Res\_list\_),  !  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | list\_upper\_num([4, 5], 4, Res\_list\_) = list\_upper\_num([H|T], Num, [H|Add\_tail]) :- H > Num, !, list\_upper\_num(T, Num, Add\_tail)  Унификация успешна  Подстановка: {H = 4, T = [5], Num = 4, [4|Add\_tail] = Res\_list\_} | Формируется новое состояние резольвенты, прямой ход |
| 9 | 4 > 4,  !,  list\_upper\_num([5], 4, Add\_tail)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | Унификация:  1 < 4  Результат: ложь | Откат, возврат к предыдущему состоянию резольвенты, переход к следующему предложению |
| 10 | list\_upper\_num([4, 5], 4, Res\_list\_),  !  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | list\_upper\_num([4, 5], 4, Res\_list\_) = list\_upper\_num([\_|T], Num, Res\_list) :- list\_upper\_num(T, Num, Res\_list), !  Унификация успешна  Подстановка: {T = [5], Num = 4, Res\_list = Res\_list\_} | Формируется новое состояние резольвенты, прямой ход |
| 11 | list\_upper\_num([5], 4, Res\_list\_),  !  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | list\_upper\_num([5], 4, Res\_list\_) = append([H|T], List\_added, [H|Add\_tail]) :- append(T, List\_added, Add\_tail), !  Унификация неуспешна (несовпадение функторов) | Откат, переход к следующему предложению |
| 12 | … | … | … |
| 13 | list\_upper\_num([5], 4, Res\_list\_),  !  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | list\_upper\_num([5], 4, Res\_list\_) = list\_upper\_num([H|T], Num, [H|Add\_tail]) :- H > Num, !, list\_upper\_num(T, Num, Add\_tail)  Унификация успешна  Подстановка: {H = 5, T = [], Num = 4, [5|Add\_tail] = Res\_list\_} | Формируется новое состояние резольвенты, прямой ход |
| 14 | 5 > 4,  !,  list\_upper\_num([], 4, Add\_tail)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | Унификация:  5 > 4  Результат: истина | Формируется новое состояние резольвенты, прямой ход |
| 15 | !,  list\_upper\_num([], 4, Add\_tail)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | Встречен системный предикат отсечения | Дальнейшие предложения процедуры рассматриваться не будут. Формируется новое состояние резольвенты, прямой ход |
| 16 | list\_upper\_num([], 4, Add\_tail)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | list\_upper\_num([], 4, Add\_tail) = append([H|T], List\_added, [H|Add\_tail]) :- append(T, List\_added, Add\_tail), !  Унификация неуспешна (несовпадение функторов) | Откат, переход к следующему предложению |
| 17 | … | … | … |
| 18 | list\_upper\_num([], 4, Add\_tail)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | list\_upper\_num([], 4, Add\_tail) = list\_upper\_num([H|T], Num, [H|Add\_tail]) :- H > Num, !, list\_upper\_num(T, Num, Add\_tail)  Унификация неуспешна (первые аргументы не унифицируемы) | Откат, переход к следующему предложению |
| 19 | list\_upper\_num([], 4, Add\_tail)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | list\_upper\_num([], 4, Add\_tail) = list\_upper\_num([\_|T], Num, Res\_list) :- list\_upper\_num(T, Num, Res\_list), !  Унификация неуспешна (первые аргументы не унифицируемы) | Откат, переход к следующему предложению |
| 20 | list\_upper\_num([], 4, Add\_tail)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | list\_upper\_num([], 4, Add\_tail) = list\_upper\_num([], \_, [])  Унификация успешна  Подстановка: {Add\_tail = [], [5 | Add\_tail] = Res\_list\_} | Резольвента пуста.  Вывод: Res\_list\_ = [5 | []] (или [5])  Откат, переход к следующему предложению |
| 33 | list\_upper\_num([], 4, Add\_tail)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | list\_upper\_num([], 4, Add\_tail) = make\_set(List, List)  Унификация неуспешна (несовпадение функторов) | Конец БЗ достигнут. Откат с отсечением остаточных предложений процедуры относительно шага 13 |
| 46 | list\_upper\_num([5], 4, Res\_list\_),  !  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | list\_upper\_num([5], 4, Res\_list\_) = make\_set(List, List)  Унификация неуспешна (несовпадение функторов) | Конец БЗ достигнут. Формируется новое состояние резольвенты |
| 47 | !  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | Встречен системный предикат отсечения | Дальнейшие предложения процедуры рассматриваться не будут. Резольвента пуста, восстановление резольвенты относительно шага 10 |
| 60 | list\_upper\_num([4, 5], 4, Res\_list\_),  !  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | list\_upper\_num([4, 5], 4, Res\_list\_) = make\_set(List, List)  Унификация неуспешна (несовпадение функторов) | Конец БЗ достигнут. Формируется новое состояние резольвенты |
| 61 | !  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | Встречен системный предикат отсечения | Дальнейшие предложения процедуры рассматриваться не будут. Резольвента пуста, восстановление резольвенты относительно шага 5 |
| 74 | list\_upper\_num([1, 4, 5], 4, Res\_list\_)  Резольвента непуста, запуск алгоритма унификации для подцели, извлекаемой из стека | list\_upper\_num([1, 4, 5], 4, Res\_list\_) = makeSet(List, List)  Унификация неуспешна (несовпадение функторов) | Конец БЗ достигнут. Резольвента пуста, завершение работы системы |