**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 по лабораторной работе №18 по курсу

**«Функциональное и логическое программирование»**

**Тема** Формирование и модификация списков на Prolog

**Студент** Климов И.С.

**Группа** ИУ7-62Б

**Оценка (баллы)**

**Преподаватели** Толпинская Н.Б., Строганов Ю.В.

Москва — 2022 г.

## **Задание 1**

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

**Решение**

**domains**

list = integer\*.

**predicates**

add\_more\_number(integer, integer, list, list).

greater\_help(list, integer, list, list).

greater(list, integer, list).

add\_odd\_pos(integer, unsigned, list, list)

odd\_pos\_help(list, list, unsigned, list).

odd\_pos(list, list).

delete\_equal(integer, integer, list, list).

delete\_from\_list\_help(list, integer, list, list).

delete\_from\_list(list, integer, list).

member(list, integer).

add\_member(integer, list, list).

list\_to\_set\_help(list, list, list).

list\_to\_set(list, list).

**clauses**

add\_more\_number(NumFromList, Number, List, ResultList) :-

NumFromList > Number,

ResultList = [NumFromList | List],

!.

add\_more\_number(\_, \_, List, List) :- !.

greater\_help([], \_, TmpList, TmpList) :- !.

greater\_help([H|T], Number, ResultList, TmpList) :-

add\_more\_number(H, Number, TmpList, UpdTmpList),

greater\_help(T, Number, ResultList, UpdTmpList).

greater(List, Number, ResultList) :-

greater\_help(List, Number, ResultList, []).

add\_odd\_pos(NumFromList, Position, List, UpdTmpList) :-

Position mod 2 = 1,

UpdTmpList = [NumFromList | List],

!.

add\_odd\_pos(\_, \_, List, List) :- !.

odd\_pos\_help([], TmpList, \_, TmpList) :- !.

odd\_pos\_help([H|T], ResultList, Position, TmpList) :-

add\_odd\_pos(H, Position, TmpList, UpdTmpList),

IncPosition = Position + 1,

odd\_pos\_help(T, ResultList, IncPosition, UpdTmpList).

odd\_pos(List, Result) :- odd\_pos\_help(List, Result, 0, []).

delete\_equal(NumFromList, Number, TmpList, UpdTmpList) :-

not(NumFromList = Number),

UpdTmpList = [NumFromList | TmpList],

!.

delete\_equal(\_, \_, TmpList, TmpList) :- !.

delete\_from\_list\_help([], \_, TmpList, TmpList) :- !.

delete\_from\_list\_help([H|T], Number, ResultList, TmpList) :-

delete\_equal(H, Number, TmpList, UpdTmpList),

delete\_from\_list\_help(T, Number, ResultList, UpdTmpList).

delete\_from\_list(List, Number, ResultList) :-

delete\_from\_list\_help(List, Number, ResultList, []).

member([H|\_], Number) :- H = Number, !.

member([\_|T], Number) :- member(T, Number).

add\_member(NumFromList, TmpSet, UpdTmpSet) :-

not(member(TmpSet, NumFromList)),

UpdTmpSet = [NumFromList | TmpSet],

!.

add\_member(\_, TmpSet, TmpSet) :- !.

list\_to\_set\_help([], TmpSet, TmpSet) :- !.

list\_to\_set\_help([H|T], Set, TmpSet) :-

add\_member(H, TmpSet, UpdTmpSet),

list\_to\_set\_help(T, Set, UpdTmpSet).

list\_to\_set(List, Set) :- list\_to\_set\_help(List, Set, []).

**goal**

write("Elements of list [ 1, 2, 3 ] greate than 1 : "),

greater([1, 2, 3], 1, Result);

write("\nElements in odd positions of list [ 1, 2, 3 ] : "),

odd\_pos([1, 2, 3], Result);

write("\nList with deleted 2 from list [ 1, 2, 3, 2] : "),

delete\_from\_list([1, 2, 3, 2], 2, Result);

write("\nSet from list [ 1, 2, 3, 2, 1 ] : "),

list\_to\_set([1, 2, 3, 3, 2, 1], Result).

## **Задание 2**

**Для одного** из вариантов **ВОПРОСА** и **1-ого 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 запуска алгоритма унификации, требуется указать № выбранного правила и соответствующий вывод: успех или нет – и почему.

**Решение**

**Вопрос:** greater([1, 2, 3], 1, Result).

|  |  |  |  |
| --- | --- | --- | --- |
| **Шаг** | **Состояние резольвенты, и вывод** | **Сравниваемые термы; результат, подстановка, если есть** | **Дальнейшие действия: прямой ход или откат** |
| 1 | greater([1, 2, 3], 1, Result) | greater([1, 2, 3], 1, Result)  и  add\_more\_number(NumFromList, Number, List, ResultList)  Неудача (разные главные функторы) | Прямой ход |
| 2-4 | … | … | … |
| 5 | greater([1, 2, 3], 1, Result) | greater([1, 2, 3], 1, Result)  и  greater(List, Number, ResultList)  Успех:  List = [1, 2, 3], Number = 1. | Прямой ход |
| 6 | greater\_help(  [1, 2, 3], 1,  ResultList, []) | greater\_help([1, 2, 3], 1, ResultList, [])  и  add\_more\_number(NumFromList, Number, List, ResultList)  Неудача (разные главные функторы) | Прямой ход |
| 7 | ... | … | … |

|  |  |  |  |
| --- | --- | --- | --- |
| 8 | greater\_help(  [1, 2, 3], 1,  ResultList, []) | greater\_help([1, 2, 3], 1, ResultList, [])  и  greater\_help([], \_, TmpList, TmpList)  Неудача (неуспешно унифицируется первая пара соответствующих компонентов) | Прямой ход |
| 9 | greater\_help(  [1, 2, 3], 1,  ResultList, []) | greater\_help([1, 2, 3], 1, ResultList, [])  и  greater\_help([H|T], Number, ResultList, TmpList)  Успех:  H = 1, T = [2, 3], Number = 1, TmpList = [] | Прямой ход |
| 10 | add\_more\_number(1, 1, [], UpdTmpList)  greater\_help([2, 3], 1, ResultList, UpdTmpList) | add\_more\_number(1, 1, [], UpdTmpList)  и  add\_more\_number(NumFromList, Number, List, ResultList)  Успех:  NumFromList = 1, Number = 1, List = [] | Прямой ход |
| 11 | 1 > 1    ResultList = [1 | []]  !  greater\_help([2, 3], 1, ResultList, UpdTmpList) | Неудача (1 > 1 – неверно) | Откат |
| 12 | add\_more\_number(1, 1, [], UpdTmpList)  greater\_help([2, 3], 1, ResultList, UpdTmpList) | add\_more\_number(1, 1, [], UpdTmpList)  и  add\_more\_number(\_, \_, List, List)  Успех:  \_ = 1, \_ = 1, List = [], UpdTmpList = [] | Прямой ход |

|  |  |  |  |
| --- | --- | --- | --- |
| 13 | !  greater\_help([2, 3], 1, ResultList, UpdTmpList) | Отсечение | Прямой ход |
| 14 | greater\_help([2, 3], 1, ResultList, []) | greater\_help([1, 2, 3], 1, ResultList, [])  и  add\_more\_number(NumFromList, Number, List, ResultList)  Неудача (разные главные функторы) | Прямой ход |
| 15 | … | … | … |
| 16 | greater\_help([2, 3], 1, ResultList, []) | greater\_help([2, 3], 1, ResultList, [])  и  greater\_help([], \_, TmpList, TmpList)  Неудача (неуспешно унифицируется первая пара соответствующих компонентов) | Прямой ход |
| 17 | greater\_help([2, 3], 1, ResultList, []) | greater\_help([2, 3], 1, ResultList, [])  и  greater\_help([H|T], Number, ResultList, TmpList)  Успех:  H = 2, T = [3], Number = 1, TmpList = [] | Прямой ход |
| 18 | add\_more\_number(2, 1, [], UpdTmpList)  greater\_help([3], 1, ResultList, UpdTmpList) | add\_more\_number(2, 1, [], UpdTmpList)  и  add\_more\_number(NumFromList, Number, List, ResultList)  Успех:  NumFromList = 2, Number = 1, List = [] | Прямой ход |

|  |  |  |  |
| --- | --- | --- | --- |
| 19 | 2 > 1    ResultList = [2 | []]  !  greater\_help([3], 1, ResultList, UpdTmpList) | Успех (2 > 1 – верно)  ResultList = [2] | Прямой ход |
| 20 | !  greater\_help([3], 1, ResultList, UpdTmpList) | Отсечение  UpdTmpList = ResultList = [2] | Прямой ход |
| 21 | greater\_help([3], 1, ResultList, [2]) | greater\_help([3], 1, ResultList,[2])  и  add\_more\_number(NumFromList, Number, List, ResultList)  Неудача (разные главные функторы) | Прямой ход |
| 22 | … | … | … |
| 23 | greater\_help([3], 1, ResultList, [2]) | greater\_help([3], 1, ResultList,[2])  и  greater\_help([], \_, TmpList, TmpList)  Неудача (неуспешно унифицируется первая пара соответствующих компонентов) | Прямой ход |
| 24 | greater\_help([3], 1, ResultList, [2]) | greater\_help([3], 1, ResultList,[2])  и  greater\_help([H|T], Number, ResultList, TmpList)  Успех:  H = 3, T = [], Number = 1, TmpList = [2] | Прямой ход |
| 25 | add\_more\_number(3, 1, [2], UpdTmpList)  greater\_help([], 1, ResultList, UpdTmpList) | add\_more\_number(3, 1, [2], UpdTmpList)  и  add\_more\_number(NumFromList, Number, List, ResultList)  Успех:  NumFromList = 3, Number = 1, List = [2] | Прямой ход |
| 26 | 3 > 1    ResultList =  [3 | [2]]  !  greater\_help([], 1, ResultList, UpdTmpList) | Успех (3 > 1 – верно)  ResultList = [3, 2] | Прямой ход |
| 27 | !  greater\_help([], 1, ResultList, UpdTmpList) | Отсечение  UpdTmpList = ResultList = [3, 2] | Прямой ход |
| 28 | greater\_help([], 1, ResultList, [3, 2]) | greater\_help([], 1, ResultList,  [3, 2])  и  add\_more\_number(NumFromList, Number, List, ResultList)  Неудача (разные главные функторы) | Прямой ход |
| 29 | greater\_help([], 1, ResultList, [3, 2]) | greater\_help([], 1, ResultList,  [3, 2])  и  greater\_help([], \_, TmpList, TmpList)  Успех:  \_ = 1, TmpList = [3, 2],  ResultList = [3, 2] | Решение найдено:  Result = ResultList = [3, 2]  Откат |
| 30 | greater([1, 2, 3], 1, Result) | greater([1, 2, 3], 1, Result)  и  add\_odd\_pos(NumFromList, Position, List, UpdTmpList)  Неудача (разные главные функторы) | Прямой ход |
| 31-45 | … | … | … |

|  |  |  |  |
| --- | --- | --- | --- |
| 46 | greater([1, 2, 3], 1, Result) | greater([1, 2, 3], 1, Result)  и  list\_to\_set(List, Set)  Неудача (разные главные функторы) | Прямой ход  Откат  (конец БЗ) |
| 47 | Резольвента пуста.  Найденные решения:  Result = [3, 2] |  |  |