|  |  |
| --- | --- |
| 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 №20*

*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, комментируя аргументы, эффективную программу, позволяющую:

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

Убедиться в правильности результатов

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.

# Теоретические вопросы

**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

**Каким способом можно разделить список на части, какие, требования к частям?**

Получить голову или хвост списка можно при унификации списка с [Head|Tail], Head – голова списка, Tail – хвост списка (является списком).

**Как выделить за один шаг первые два подряд идущих элемента списка?**

[Head|[Head|\_]]

**Как выделить 1-й и 3-й элемент за один шаг?**

[Head | [ \_ | [Head | \_ ]]]

**Как формируется новое состояние резольвенты?**

C 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.

**Когда останавливается работа системы?**

Когда найдены все возможные ответы на вопрос.

**Как это определяется на формальном уровне?**

Когда в резольвенте находится исходный вопрос, для которого пройдена вся БЗ.

# Листинг

1. domains
2. lst = integer\*.
4. predicates
5. big(lst, integer, lst).     % original, num, result
6. odds(lst, lst).         % original, result
7. delete(lst, integer, lst).  % original, num, result
8. createSet(lst, lst).        % original, result
10. clauses
11. %1
12. big([], \_, []).
13. big([H|T], Num, [H|RT]) :- Num < H, big(T, Num, RT), !.
14. big([\_|T], Num, RT) :- big(T, Num, RT).
16. %2
17. odds([], []).
18. odds([\_], []).
19. odds([\_|[H|T]],[H|RT]) :- odds(T, RT).
21. %3
22. delete([], \_, []).
23. delete([Elem|T], Elem, RT) :- delete(T, Elem, RT), !.
24. delete([H|T], Elem, [H|RT]) :- delete(T, Elem, RT).
26. %4
27. createSet([], []).
28. createSet([H|T], [H|RT]) :- delete(T, H, Tmp), createSet(Tmp, RT).
30. goal
31. %big([1, 2, 3, 4, 2, 1],5, Res).
32. %odds([1, 2, 3, 4], Res).
33. %delete([1, 2, 3, 3, 1, 3, 4], 3, Res).
34. %createSet([1, 2, 3, 3, 1, 3, 4], Res).
35. %createSet([1, 1, 1], Res).

# Таблицы

**Вопрос: big([0], 2, Res)**

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Текущая резольвента – ТР | ТЦ, выбираемые правила: сравниваемые термы,  подстановка | Дальнейшие действия с комментариями |
| 1 | big([0], 2, Res) | ТЦ: big([0], 2, Res) | Поиск знания с начала БЗ |
|  | big([0], 2, Res) | ПР1:  [] = [0] \_ = 2 [] = Res  Неудача | Метка переносится ниже |
|  | big([0], 2, Res) | ПР2:  [H|T] = [1] Min = 2 [H1|RT1] = Res  Успех  H = 0 T1 = [] Min1 = 2 Res = [1|RT1] | Тело ПР2 заменяет цель в резольвенте |
| 2 | 0 > 2 big([], 2, RT1) ! | Сравнение:  0 > 2  Ложь | Откат к 1.  Метка переносится ниже. |
| 3 | big([0], 2, Res) | ПР3:  [\_|T] = [1] Min = 2 RT = Res  Успех  T = [] Min = 2 Res = RT3 | Тело ПР3 заменяет цель в резольвенте |
| 4 | big([], 2, RT3) | ТЦ: big([], 2, RT3) | Поиск знания с начала БЗ |
|  | big([], 2, RT3) | ПР1: [] = [] \_ = 2 [] = RT  Успех  RT3 = 0 | Пустое тело заменяет цель в резольвенте |
|  | Пусто |  | Успех.  Res = RT = []  Возврат к предыдущему состоянию резольвенты |
| 5 | big([], 2, RT) | ПР2: [H|T] = [] Min = 2 [H|RT] = RT  Неудача | Метка переносится ниже. |
|  | big([], 2, RT) | ПР3:  [\_|T] = [] Min = 2 RT = Res  Неудача | Необходимо включить откат, но метки в конце процедур – других альтернатив нет.  Система завершает работу. |