



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»

Отчёт по лабораторной работе №18 по курсу «Функциональное и логическое программирование»

Тема Формирование и модификация списков на Prolog

Студент Богаченко А. Е.

Группа ИУ7-65Б

Оценка (баллы) _____

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

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).

```

1 domains
2   intlist = integer*
3
4 predicates
5   bigger_than(intlist, integer, intlist)
6   odd_list(intlist, intlist)
7   single_del(intlist, integer, intlist)
8   full_del(intlist, integer, intlist)
9   set(intlist, intlist)
10
11 clauses
12   bigger_than([Head | Tail], N, [Head | ResTail]) :- Head > N, !, bigger_than(Tail,
13     ↪ N, ResTail).
14   bigger_than([_ | Tail], N, Result) :- bigger_than(Tail, N, Result).
15   bigger_than([], _, []).
16
17   odd_list([_, Head | Tail], [Head | ResTail]) :- !, odd_list(Tail, ResTail).
18   odd_list([], []).
19
20   single_del([Head | Tail], N, Tail) :- Head = N, !.
21   single_del([Head | Tail], N, [Head | ResTail]) :- single_del(Tail, N, ResTail), !.
22   single_del([], _, []).
23
24   full_del([Head | Tail], N, [Head | ResTail]) :- Head <> N, !, full_del(Tail, N,
25     ↪ ResTail).
26   full_del([_ | Tail], N, Result) :- full_del(Tail, N, Result), !.
27   full_del([], _, []).
28
29   set([Head | Tail], [Head | Result]) :- full_del(Tail, Head, Nt), !, set(Nt,
30     ↪ Result).
31   set([], []).
32
33 goal
34   %bigger_than([1, 7, 3, 4, 5, 6], 3, Result).
35   %odd_list([1, 2, 3, 4, 5, 6, 7, 8], Result).
36
37   %single_del([1, 2, 3, 1, 2, 3, 1, 2, 3], 1, Result).
38   %full_del([1, 2, 3, 1, 2, 3, 1, 2, 3], 1, Result).
39
40   set([1, 2, 3, 1, 2, 3, 1, 2, 3], Result).

```

Теоретические вопросы

1. Как организуется хвостовая рекурсия в Prolog?

Для организации хвостовой рекурсии рекурсивный вызов должен быть последней подцелью и нужно избавиться от точек возврата с помощью отсечения, чтобы исключить возможные альтернативы.

2. Какое первое состояние резольвенты?

Стек, который содержит конъюнкцию целей, истинность которых система должна доказать, называется резольвентой. Первое состояние резольвенты – вопрос.

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

Для разделения списка используется специальный символ `|`. Список не должен быть пустым

4. Как выделить за один шаг первые два подряд идущих элемента списка? Как выделить 1-й и 3-й элемент за один шаг?

Для выделения нескольких элементов используется специальный символ `|`

5. Как формируется новое состояние резольвенты?

Резольвента меняется в два этапа:

1. редукция – замена подцели телом того правила, с заголовком которого успешно унифицируется данная подцель;
2. применение ко всей резольвенте подстановки.

Резольвента уменьшается, если удаётся унифицировать подцель с фактом. Система отвечает «Да», только когда резольвента становится пустой.

6. Когда останавливается работа системы? Как это определяется на формальном уровне?