



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 №11 по дисциплине "Функциональное и логическое программирование"

Тема Формирование, обработка и модификация списков на Prolog

Студент Егорова П.А.

Группа ИУ7-64Б

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

Москва — 2023 г.

Постановка задачи

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);
4. Сформировать список из элементов числового списка, больших заданного значения;
5. Удалить заданный элемент из списка (один или все вхождения);
6. Объединить два списка.

Решение

```
1 domains
2   list = integer*
3
4 predicates
5   rec_length(integer, integer, list)
6   length(integer, list)
7
8   rec_sum(integer, integer, list)
9   sum(integer, list)
10
11  rec_oddsum(integer, integer, list)
12  oddsum(integer, list)
13
14  rm_all(list, integer, list)
15
16  rm_one(list, integer, list)
17
18  bigger(list, integer, list)
19
20  lists_join(list, list, list)
21
22 clauses
23  rec_length(Res, Len, [_ | Tail]) :- NewLen = Len + 1, rec_length(Res,
24    NewLen, Tail).
25  rec_length(Res, Len, []) :- Res = Len, !.
26  length(Res, List) :- rec_length(Res, 0, List).
27
28  rec_sum(Res, Sum, [Head | Tail]) :- NewSum = Sum + Head, rec_sum(Res,
29    NewSum, Tail).
30  rec_sum(Res, Sum, []) :- Res = Sum, !.
```

```

29 sum(Res, List) :- rec_sum(Res, 0, List).
30
31 rec_oddsum(Res, Sum, [_ | Tail]) :- NewSum = Sum + Head, rec_oddsum(
    Res, NewSum, Tail).
32 rec_oddsum(Res, Sum, []) :- Res = Sum, !.
33 oddsum(Res, List) :- rec_oddsum(Res, 0, List).
34
35 bigger([Head | Tail], N, [Head | ResTail]) :- Head > N, !, bigger(Tail, N,
    ResTail).
36 bigger([_ | Tail], N, Result) :- bigger(Tail, N, Result).
37 bigger([], _, []).
38
39 rm_one([Head | Tail], N, Tail) :- Head = N, !.
40 rm_one([Head | Tail], N, [Head | ResList]) :- !, rm_one(Tail, N, ResList).
41 rm_one([], _, []).
42
43 rm_all([Head | Tail], N, [Head | ResList]) :- Head <> N, !, rm_all(Tail, N
    , ResList).
44 rm_all([_ | Tail], N, ResList) :- rm_all(Tail, N, ResList), !.
45 rm_all([], _, []).
46
47 lists_join([Head | Tail], [A | B], [Head | ResTail]) :- !, lists_join(Tail
    , [A | B], ResTail).
48 lists_join([], [Head | Tail], [Head | ResTail]) :- !, lists_join([], Tail,
    ResTail).
49 lists_join([], [], []).
50
51 goal
52 %length(Res, [1, 2, 3, 4]).
53 %sum(Res, [1, 2, 3, 4]).
54 %oddsum(Res, [1, 2, 3, 4]).
55 %bigger([1, 7, 3, 4, 5, 6, 2], 3, Result).
56 %rm_one([1, 2, 3, 1, 2, 3, 1, 2, 3], 3, Result).
57 %rm_all([1, 2, 3, 1, 2, 3, 1, 2, 3], 1, Result).
58 %lists_join([1, 7, 3, 4, 5, 6, 2], [1, 2, 3], Result).

```

Вопрос

Вопрос: `odd_list([1, 2, 3, 4], Result)`

No шага	Состояние резольвенты, и вывод: дальнейшие действия (почему?)	Для каких термов запускается алгоритм унификации: $T1=T2$ и каков результат (и подстановка)	Дальнейшие действия: прямой ход или откат (почему и к чему приводит?)
1	<code>odd_list([1, 2, 3, 4], Result)</code>	<code>odd_list([_, Head Tail], [Head ResTail])</code> и <code>odd_list([1, 2, 3, 4], Result)</code> Результат: успех + подстановка $\{_ = 1, \text{Head} = 2, \text{Tail} = [3, 4], \text{Result} = [2 \text{Restail}]\}$	Заголовок правила заменяется его телом с учетом подстановки
2	<code>odd_list([3, 4], ResTail)</code>	<code>odd_list([_, Head Tail], [Head ResTail])</code> и <code>odd_list([3, 4], Result)</code> Результат: успех + подстановка $\{_ = 3, \text{Head} = 4, \text{Tail} = [], \text{Result} = [4 \text{Restail}]\}$	Заголовок правила заменяется его телом с учетом подстановки
3	<code>odd_list([], ResTail)</code>	<code>odd_list([_, Head Tail], [Head ResTail])</code> и <code>odd_list([3, 4], Result)</code> Результат: неудача, термы не унифицируемы	Переход к следующему терму
4	<code>odd_list([], ResTail)</code>	<code>odd_list([], [])</code> и <code>odd_list([], ResTail)</code> Результат: успех + подстановка $\{\text{Restail} = []\}$	Заголовок правила заменяется его телом с учетом подстановки
5	!	успех	Переход к следующему терму
6			Решение найдено, возврат результата <code>Restail = []</code>

Рис. 1: Таблица к вопросу