



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 №11 по дисциплине "Функциональное и логическое программирование"

Тема Рекурсия на Prolog

Студент Светличная А.А.

Группа ИУ7-63Б

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

Москва — 2023 г.

1 Задание

Используя хвостовую рекурсию, разработать (комментируя назначение аргументов) эффективную программу, 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.

Для одного из вариантов вопроса составить таблицу, отражающую конкретный порядок работы системы.

2 Листинги

Листинг 2.1 – Выполнение задания

```
1 DOMAINS
2
3     list = integer*
4
5 PREDICATES
6
7     len(list, integer).
8     len_rec(list, integer, integer).
9
10    sum(list, integer).
11    sum_rec(list, integer, integer).
12
13    sum_odd_pos(list, integer).
14    sum_odd_pos_rec(list, integer, integer).
15
16
17    el_larger(list, integer, list).
18    el_larger_rec(list, integer, list, list).
19
20    rm_el(list, integer, list).
21    rm_el_rec(list, integer, list, list).
22
23    merge(list, list, list).
24    merge_rec(list, list, list, list).
25
26    append(list, list, list).
27
28 CLAUSES
29
30    %1
31    len_rec([], Res, Res) :- !.
32
33    len_rec([_|T], Res, Cur) :-
34        Cur_res = Cur + 1,
35        len_rec(T, Res, Cur_res).
36
37    len(List, Res) :-
38        len_rec(List, Res, 0).
39
40    % 2
41    sum_rec([], Res, Res) :- !.
42
43    sum_rec([H|T], Res, Cur) :-
44        Cur_res = Cur + H,
```

```

45         sum_rec(T, Res, Cur_res).
46
47 sum(List, Res) :-
48     sum_rec(List, Res, 0).
49
50 %3
51 sum_odd_pos_rec([], Res, Res) :- !.
52
53 sum_odd_pos_rec([_|[H|T]], Res, Cur) :-
54     Cur_res = Cur + H,
55     sum_odd_pos_rec(T, Res, Cur_res), !.
56
57 sum_odd_pos_rec([_|[H|_]], Res, Cur) :-
58     Res = Cur + H.
59
60 sum_odd_pos(List, Res) :-
61     sum_odd_pos_rec(List, Res, 0).
62
63 %4
64 el_larger_rec([], _, Res, Res) :- !.
65
66 el_larger_rec([H|T], Num, Res, Cur_res) :-
67     H > Num, !,
68     el_larger_rec(T, Num, Res, [H|Cur_res]).
69
70 el_larger_rec([_|T], Num, Res, Cur_res) :-
71     el_larger_rec(T, Num, Res, Cur_res).
72
73 el_larger(List, Num, Res) :- el_larger_rec(List, Num, Res, []).
74
75 %5
76 rm_el_rec([], _, Res, Res) :- !.
77
78 rm_el_rec([H|T], Num, Res, Cur_res) :-
79     H <> Num, !,
80     rm_el_rec(T, Num, Res, [H|Cur_res]).
81
82 rm_el_rec([_|T], Num, Res, Cur_res) :-
83     rm_el_rec(T, Num, Res, Cur_res).
84
85 rm_el(List, Num, Res) :- rm_el_rec(List, Num, Res, []).
86
87 %6.1
88 merge_rec([], [], Res, Res) :- !.
89 merge_rec([H|T1], List2, Res, Cur_res) :-
90     !, merge_rec(T1, List2, Res, [H|Cur_res]).
91 merge_rec(List1, [H|T2], Res, Cur_res) :-
92     !, merge_rec(List1, T2, Res, [H|Cur_res]).

```

```
93
94     merge(List1, List2, Res) :- merge_rec(List1, List2, Res, []).
95
96     %6.2
97     append([], L2, L2).
98     append([H|T1], L2, [H|T3]) :- append(T1, L2, T3).
99
100 GOAL
101     %len([1, 2, 3, 4], Res).
102     %sum([1, 2, 3, 4], Res).
103     %sum_odd_pos([1, 2, 3, 4, 5], Res).
104
105     %el_larger([1, 2, 3, 4], 2, Res).
106     %rm_el([3, 1, 3, 2, 3], 1, Res).
107     merge([1, 2, 3], [4, 5], Res).
```

2 Таблицы для Лабораторной работы №11

Вопрос: $\text{sum}([1, 2, 3], \text{Res})$.

№ шага	Сравнение термы, результат, подстановка, если есть	Дальнейшие действия, прямой ход или откат (к чему приводит?)
0		Состояние резольвенты: $\text{sum}([1, 2, 3], \text{Res})$
1	Сравнение: $\text{sum}([1, 2, 3], \text{Res}) == \text{len_rec}([], \text{Res}, \text{Res})$ Унификация: неуспешно (несовпадение функторов)	Прямой ход Переход к следующему предложению
2-5	—//—	—//—
6	Сравнение: $\text{sum}([1, 2, 3], \text{Res}) == \text{sum}([List], \text{Res})$ Унификация: успешно Подстановка: $\{List = [1, 2, 3], \text{Res} = \text{Res}\}$	Образование новой резольвенты: 1. Редукция верхней подцели: замена $\text{sum}([1, 2, 3], \text{Res})$ телом найденного правила Получена конъюнкция целей: $\text{sum_rec}(List, \text{Res}, 0)$. 2. Применение подстановки к полученной конъюнкции целей. Новое состояние резольвенты: $\text{sum_rec}([1, 2, 3], \text{Res}, 0)$.
7	Сравнение: $\text{sum_rec}([1, 2, 3], \text{Res}, 0) == \text{len_rec}([], \text{Res}, Cur)$ Унификация: неуспешно (несовпадение функторов)	Прямой ход Переход к следующему предложению
8-10	—//—	—//—
11	Сравнение: $\text{sum_rec}([1, 2, 3], \text{Res}, 0) == \text{sum_rec}([H/T], \text{Res}, Cur)$ Унификация: успешно Подстановка: $\{H = 1, T = [2, 3], \text{Res} = \text{Res}, Cur = 0\}$	Образование новой резольвенты: 1. Редукция верхней подцели: замена $\text{sum_rec}([1, 2, 3], \text{Res}, 0)$ телом найденного правила Получена конъюнкция целей: $Cur_res = Cur + H,$ $\text{sum_rec}(T, \text{Res}, Cur_res)$. 2. Применение подстановки к полученной конъюнкции целей. Новое состояние резольвенты: $Cur_res = 0 + 1,$

		$sum_rec([2, 3], Res, Cur_res).$
12	<p>Сравнение: $Cur_res = 0 + 1$</p> <p>Результат: успех</p> <p>Подстановка: $\{Cur_res = 1\}$</p>	<p>Образование новой резольвенты:</p> <p>1. Редукция верхней подцели: удаление $Cur_res = 0 + 1$, так как результат – истина.</p> <p>Получена конъюнкция целей: $sum_rec([2, 3], Res, Cur_res).$</p> <p>2. Применение подстановки к полученной конъюнкции целей.</p> <p>Новое состояние резольвенты: $sum_rec([2, 3], Res, 1).$</p>
13-27
28	<p>Сравнение: $sum_rec([], Res, 6) == sum_rec([], Res, Res)$</p> <p>Унификация: успешно</p> <p>Подстановка: $\{Res = 6\}$</p>	<p>Образование новой резольвенты:</p> <p>1. Редукция верхней подцели: замена $sum_rec([], 6, 6)$ телом найденного правила</p> <p>Получена конъюнкция целей: $!$</p> <p>2. Применение подстановки к полученной конъюнкции целей.</p> <p>Новое состояние резольвенты: $!$</p>
29	!	<p>Решение найдено: формирование подстановки в качестве побочного эффекта: $\{Res = 6\}$</p> <p>Встречен системный предикат отсечения, откат с отсечением остаточных предложений процедуры относительно шага 28</p>
30	<p>Сравнение: $sum_rec([], 6, 6) == sum(List, Res)$</p> <p>Унификация: неуспешно (несовпадение функторов)</p>	<p>Прямой ход</p> <p>Переход к следующему предложению</p>
32-35	—//—	—//—
36		<p>Конец БЗ</p> <p>Обратная трассировка:</p>

		<p>1) Отмена крайней редукции (шаг 24)</p> <p>2) Восстановление предыдущего состояния резольвенты (шаг 23): $\text{sum_rec}([3], Res, 3)$.</p> <p>3) Реконкретизация переменных с шага 24: $\{H = 3, T = [], Res = Res, Cur = 6\}$</p> <p>Переход к следующему предложению относительно шага 24.</p>
Решения далее найдены не будут, в итоге система через несколько восстановлений восстановит резольвенту до шага 0		
53		<p>Конец БЗ</p> <p>Обратная трассировка:</p> <p>1) Отмена крайней редукции (шаг 0)</p> <p>2) Восстановление предыдущего состояния резольвенты: резольвента пуста.</p> <p>Завершение работы.</p> <p>На вопрос удалось ответить утвердительно, поэтому в качестве побочного эффекта была возвращена 1 подстановка.</p>