



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 №17-18
по дисциплине
"Функциональное и логическое программирование"**

Тема Списки на Prolog

Студент Малышев И. А.

Группа ИУ7-61Б

Оценка (баллы) _____

Преподаватель: Толпинская Н. Б.

Москва — 2022 г.

Лабораторная работа №17

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);

Для одного из вариантов ВОПРОСА и одного из заданий составить таблицу, отражающую конкретный порядок работы системы.

Решение

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

Порядок работы

| Шаг | Сравниваемые термы, результаты | Дальнейшие действия | Резольвента | Подстановка |
|-----|---|--|---|--|
| 1 | Запуск алгоритма унификации для $\text{length}(\text{Res}, [4, 3, 2, 1])$ и $\text{rec_length}(\text{Res}, \text{Len}, _ \text{Tail})$. Унификация неуспешна | Прямой ход, переход к следующему предложению | $\text{length}(\text{Res}, [4, 3, 2, 1])$ | Пустая |
| ... | ... | ... | ... | ... |
| 3 | Запуск алгоритма унификации для $\text{length}(\text{Res}, [1, 2, 3, 4])$ и $\text{length}(\text{Res}, \text{List})$. Успешная унификация. | Прямой ход (след. цель резольвенты) | $\text{rec_length}(\text{Res}, 0, [4, 3, 2, 1])$! | $\text{Res} = \text{Res}$ $\text{List} = [4, 3, 2, 1]$ |
| 4 | Запуск алгоритма унификации для $\text{rec_length}(\text{Res}, 0, [4, 3, 2, 1])$ и $\text{rec_length}(\text{Res}, \text{Len}, _ \text{Tail})$. Успешная унификация. | Прямой ход (след. цель резольвенты) | $\text{NewLen} = 0 + 1$! $\text{rec_length}(\text{Res}, \text{NewLen}, [3, 2, 1])$! | $\text{Res} = \text{Res}$ $\text{List} = [4, 3, 2, 1]$ $\text{Len} = 0$ $\text{Res} = \text{Res}$ $\text{Tail} = [3, 2, 1]$ |
| 5 | $\text{NewLen} = 0 + 1$ | Прямой ход (след. цель резольвенты) | $\text{rec_length}(\text{Res}, 1, [3, 2, 1])$! | $\text{Res} = \text{Res}$ $\text{List} = [4, 3, 2, 1]$ $\text{Len} = 0$ $\text{Res} = \text{Res}$ $\text{Tail} = [3, 2, 1]$ $\text{NewLen} = 1$ |
| 5 | !, отсечение 4, 5 | Прямой ход (след. цель резольвенты) | $\text{rec_length}(\text{Res}, 1, [3, 2, 1])$! | $\text{Res} = \text{Res}$ $\text{List} = [4, 3, 2, 1]$ $\text{Len} = 0$ $\text{Res} = \text{Res}$ $\text{Tail} = [3, 2, 1]$ $\text{NewLen} = 1$ |
| 6 | Запуск алгоритма унификации для $\text{rec_length}(\text{Res}, 1, [3, 2, 1])$ и $\text{rec_length}(\text{Res}, \text{Len}, _ \text{Tail})$. Успешная унификация. | Прямой ход (след. цель резольвенты) | $\text{NewLen} = 1 + 1$! $\text{rec_length}(\text{Res}, \text{NewLen}, [2, 1])$! | ... $\text{NewLen} = 1$ $\text{Res} = \text{Res}$ $\text{Len} = 1$ $\text{Tail} = [2, 1]$ |
| 7 | $\text{NewLen} = 1 + 1$ | Прямой ход (след. цель резольвенты) | $\text{rec_length}(\text{Res}, 2, [2, 1])$! | ... $\text{NewLen} = 1$ $\text{Res} = \text{Res}$ $\text{Len} = 1$ $\text{Tail} = [2, 1]$ $\text{NewLen} = 2$ |
| 8 | !, правда, отсечение 6, 7 | Прямой ход (след. цель резольвенты) | $\text{rec_length}(\text{Res}, 2, [2, 1])$! | ... $\text{NewLen} = 1$ $\text{Res} = \text{Res}$ $\text{Len} = 1$ $\text{Tail} = [2, 1]$ $\text{NewLen} = 2$ |

| | | | | |
|----|--|--|--|---|
| 9 | Запуск алгоритма унификации для <code>rec_length(Res, 2, [2, 1])</code> и <code>rec_length(Res, Len, [_ Tail])</code> . Успешная унификация. | Прямой ход (след. цель резолювенты) | <code>NewLen = 2 + 1</code> ! <code>rec_length(Res, NewLen, [1])</code> ! | <code>...</code> <code>NewLen = 2</code> <code>Res = Res</code> <code>Len = 2</code> <code>Tail = [1]</code> |
| 10 | <code>NewLen = 2 + 1</code> | Прямой ход (след. цель резолювенты) | <code>rec_length(Res, 3, [1])</code> ! | <code>...</code> <code>NewLen = 2</code> <code>Res = Res</code> <code>Len = 2</code> <code>Tail = [1]</code> <code>NewLen = 3</code> |
| 11 | !, отсечение 9, 10 | Прямой ход (след. цель резолювенты) | <code>rec_length(Res, 3, [4])</code> ! | <code>...</code> <code>NewLen = 2</code> <code>Res = Res</code> <code>Len = 2</code> <code>Tail = [1]</code> <code>NewLen = 3</code> |
| 12 | Запуск алгоритма унификации для <code>rec_length(Res, 2, [1])</code> и <code>rec_length(Res, Len, [_ Tail])</code> . Успешная унификация. | Прямой ход (след. цель резолювенты) | <code>NewLen = 3 + 1</code> ! <code>rec_length(Res, 4, [])</code> ! | <code>...</code> <code>NewLen = 3</code> <code>Res = Res</code> <code>Len = 3</code> <code>Tail = []</code> |
| 13 | <code>NewLen = 3 + 1</code> | Прямой ход (след. цель резолювенты) | <code>rec_length(Res, 4, [])</code> ! | <code>...</code> <code>Len = 3</code> <code>Res = Res</code> <code>Tail = []</code> <code>NewLen = 4</code> |
| 14 | !, отсечение 12, 13 | Прямой ход (след. цель резолювенты) | <code>rec_length(Res, 4, [])</code> ! | <code>...</code> <code>NewLen = 3</code> <code>Res = Res</code> <code>Len = 3</code> <code>Tail = []</code> <code>NewLen = 4</code> |
| 15 | Запуск алгоритма унификации для <code>rec_length(Res, 4, [])</code> и <code>rec_length(Res, Len, [_ Tail])</code> . Успешная унификация. | Прямой ход. Переход к след. предложению | <code>rec_length(Res, 4, [])</code> ! | <code>...</code> <code>NewLen = 3</code> <code>Res = Res</code> <code>Len = 3</code> <code>Tail = []</code> <code>NewLen = 4</code> |
| 16 | Запуск алгоритма унификации для <code>rec_length(Res, 4, [])</code> и <code>rec_length(Res, Len, [_ Tail])</code> . Успешная унификация. | Прямой ход (след. цель резолювенты) | <code>Res = 4</code> ! ! | <code>...</code> <code>Tail = []</code> <code>NewLen = 4</code> <code>Res = Res</code> <code>Len = 4</code> |
| 17 | <code>Res = 4</code> | Прямой ход (след. цель резолювенты) | ! ! | <code>...</code> <code>Tail = []</code> <code>NewLen = 4</code> <code>Res = Res</code> <code>Len = 4</code> <code>Res = 4</code> |
| 18 | !, отсечение 16, 17 | Прямой ход (след. цель резолювенты) | ! | <code>...</code> <code>Tail = []</code> <code>NewLen = 4</code> <code>Res = Res</code> <code>Len = 4</code> <code>Res = 4</code> |
| 19 | !, отсечение 3 | Результат (<code>Res = 4</code>). Завершение работы. | Пустая | <code>...</code> <code>Tail = []</code> <code>NewLen = 4</code> <code>Res = Res</code> <code>Len = 4</code> <code>Res = 4</code> |

Лабораторная работа №18

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).

Для одного из вариантов ВОПРОСА и 1-ого задания составить таблицу, отражающую конкретный порядок работы системы.

Решение

```
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, !,
13     bigger_than(Tail, 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     !.
23   single_del([], _, []).
24
25   full_del([Head | Tail], N, [Head | ResTail]) :- Head <> N, !, full_del(Tail, N,
26     ResTail).
27   full_del([_ | Tail], N, Result) :- full_del(Tail, N, Result), !.
28   full_del([], _, []).
29
30   set([Head | Tail], [Head | Result]) :- full_del(Tail, Head, Nt), !, set(Nt,
31     Result).
32   set([], []).
33
34 goal
35   bigger_than([1, 7, 3, 4, 5, 6], 3, Result).
36   odd_list([1, 2, 3, 4, 5, 6, 7, 8], Result).
37
38   single_del([1, 2, 3, 1, 2, 3, 1, 2, 3], 1, Result).
39   full_del([1, 2, 3, 1, 2, 3, 1, 2, 3], 1, Result).
40
41   set([1, 2, 3, 1, 2, 3, 1, 2, 3], Result).
```

Порядок работы

| Шаг | Сравниваемые термы, результаты | Дальнейшие действия | Резольвента | Подстановка |
|-----|--|---|---|---|
| 1 | Запуск алгоритма унификации для $\text{odd_list}([1, 2, 3, 4, 5, 6], \text{Result})$ и $\text{odd_list}(_, \text{Head} \mid \text{Tail}), [\text{Head} \mid \text{ResTail}]$:- $!$, $\text{odd_list}(\text{Tail}, \text{ResTail})$. Унификация успешна. | Прямой ход (след. цель резольвенты) | $!$ $\text{odd_list}([3, 4, 5, 6], \text{ResTail})$ | $\text{Head} = 2$ $\text{Result} = [2 \mid \text{ResTail}]$ $\text{Tail} = [3, 4, 5, 6]$ |
| 2 | $!$, отсечение шага 1 | Прямой ход (след. цель резольвенты) | $\text{odd_list}([3, 4, 5, 6], \text{ResTail})$ | $\text{Head} = 2$ $\text{Result} = [2 \mid \text{ResTail}]$ $\text{Tail} = [3, 4, 5, 6]$ |
| 3 | Запуск алгоритма унификации для $\text{odd_list}([3, 4, 5, 6], \text{Res})$ и $\text{odd_list}(_, \text{Head} \mid \text{Tail}), [\text{Head} \mid \text{ResTail}]$ унификация успешна. | Прямой ход (след. цель резольвенты) | $!$ $\text{odd_list}([5, 6], \text{ResTail})$ | $\text{Head} = 2$ $\text{Result} = [2 \mid \text{ResTail}]$ $\text{Tail} = [3, 4, 5, 6]$ $\text{Head} = 4$ $\text{ResTail} = [4 \mid \text{ResTail}]$ $\text{Tail} = [5, 6]$ |
| 4 | $!$, отсечение шага 3 | Прямой ход (след. цель резольвенты) | $\text{odd_list}([5, 6], \text{ResTail})$ | $\text{Head} = 2$ $\text{Result} = [2 \mid \text{ResTail}]$ $\text{Tail} = [3, 4, 5, 6]$ $\text{Head} = 4$ $\text{ResTail} = [4 \mid \text{ResTail}]$ $\text{Tail} = [5, 6]$ |
| 5 | Запуск алгоритма унификации для $\text{odd_list}([5, 6], \text{Res})$ и $\text{odd_list}(_, \text{Head} \mid \text{Tail}), [\text{Head} \mid \text{ResTail}]$. Унификация успешна. | Прямой ход (след. цель резольвенты) | $!$ $\text{odd_list}([], \text{ResTail})$ | $\text{Head} = 2$ $\text{Result} = [2 \mid \text{ResTail}]$ $\text{Tail} = [3, 4, 5, 6]$ $\text{Head} = 4$ $\text{ResTail} = [4 \mid \text{ResTail}]$ $\text{Tail} = [5, 6]$ $\text{Head} = 6$ $\text{ResTail} = [6 \mid \text{ResTail}]$ $\text{Tail} = []$ |
| 6 | $!$, отсечение шага 5 | Прямой ход (след. цель резольвенты) | $\text{odd_list}([], \text{ResTail})$ | $\text{Head} = 2$ $\text{Result} = [2 \mid \text{ResTail}]$ $\text{Tail} = [3, 4, 5, 6]$ $\text{Head} = 4$ $\text{ResTail} = [4 \mid \text{ResTail}]$ $\text{Tail} = [5, 6]$ $\text{Head} = 6$ $\text{ResTail} = [6 \mid \text{ResTail}]$ $\text{Tail} = []$ |
| 7 | Запуск алгоритма унификации для $\text{odd_list}([], \text{Res})$ и $\text{odd_list}(_, \text{Head} \mid \text{Tail}), [\text{Head} \mid \text{ResTail}]$. Унификация неуспешна. | Прямой ход. Переход к следу. предложению | $\text{only_odd}([], \text{ResTail})$ | $\text{Head} = 2$ $\text{Result} = [2 \mid \text{ResTail}]$ $\text{Tail} = [3, 4, 5, 6]$ $\text{Head} = 4$ $\text{ResTail} = [4 \mid \text{ResTail}]$ $\text{Tail} = [5, 6]$ $\text{Head} = 6$ $\text{ResTail} = [6 \mid \text{ResTail}]$ $\text{Tail} = []$ |
| 8 | Запуск алгоритма унификации для $\text{odd_list}([], \text{Res})$ и $\text{odd_list}([], [])$. Унификация успешна. | Прямой ход (след. цель резольвенты) | $!$ | $\text{Head} = 2$ $\text{Result} = [2 \mid \text{ResTail}]$ $\text{Tail} = [3, 4, 5, 6]$ $\text{Head} = 4$ $\text{ResTail} = [4 \mid \text{ResTail}]$ $\text{Tail} = [5, 6]$ $\text{Head} = 6$ $\text{ResTail} = [6 \mid \text{ResTail}]$ $\text{Tail} = []$ $\text{ResTail} = []$ |
| 9 | $!$, отсечение шага 8 | Результат ($\text{Result} = [2, 4, 6]$). Завершение работы | Пустая | $\text{Head} = 2$ $\text{Result} = [2 \mid \text{ResTail}]$ $\text{Tail} = [3, 4, 5, 6]$ $\text{Head} = 4$ $\text{ResTail} = [4 \mid \text{ResTail}]$ $\text{Tail} = [5, 6]$ $\text{Head} = 6$ $\text{ResTail} = [6 \mid \text{ResTail}]$ $\text{Tail} = []$ $\text{ResTail} = []$ |