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» (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 Программная инженерия**

О Т Ч Е Т

по лабораторной работе № 1 9

Дисциплина: Функциональное и логическое программирование

Студент ИУ7-62Б _____ Е.В. Брянская
(Группа) (Подпись, дата) (И.О. Фамилия)

Преподаватель _____ Н.Б.Толпинская
Ю.В.Строганов
(Подпись, дата) (И.О. Фамилия)

Москва, 2021

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)

Убедиться в правильности результатов.

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

Так как резольвента хранится в виде стека, то состояние резольвенты следует отображать в столбик: вершина – сверху! Новый шаг надо начинать с нового состояния резольвенты! Для каждого запуска алгоритма унификации, требуется указать № выбранного правила и дальнейшие действия – и почему.

```
domains
        lst = integer*.

predicates
        len(lst, integer).
        len(lst, integer, integer).

        sum(lst, integer).
        sum(lst, integer, integer).

        sum_odd_pos(lst, integer).
        sum_odd_pos(lst, integer, integer).

clauses
        % list's length
        len([_|T], Len_temp, Len) :-            % T - tail
                Temp = Len_temp + 1,            % Len_temp - temporary length
                len(T, Temp, Len),              % Len - length
                !.                              % Temp - temporary var
        len([], Len, Len) :- !.                 % Lst - currect list
        len(Lst, Len) :- len(Lst, 0, Len), !.

        % sum of elements
        sum([X|T], Sum_temp, Sum) :-            % X - 1st element
                Temp = Sum_temp + X,            % T - tail
                sum(T, Temp, Sum).              % Sum_temp - temporary sum, Sum - sum
        sum([], Sum, Sum) :- !.                 % Temp - temporary var
        sum(Lst, Sum) :- sum(Lst, 0, Sum), !.   % Lst - currect list

        % sum of elements in odd positions
        sum_odd_pos([_, X|T], Sum_temp, Sum) :-         % X - 2nd element from head = 1st index
                Temp = Sum_temp + X,                    % T - tail
                sum_odd_pos(T, Temp, Sum).              % Sum_temp - temporary sum, Sum - sum
        sum_odd_pos([_], Sum, Sum) :- !.        % Temp - temporary var
        sum_odd_pos([], Sum, Sum) :- !.                 % Lst - currect list, result
```

```
        sum_odd_pos(Lst, Sum) :- sum_odd_pos(Lst, 0, Sum), !.

goal
        %len([5], Len).
        %sum([-5, 0, 5, -9], Sum).
        %sum_odd_pos([-5, 1, 3], Sum).
```

Текст процедуры:

```
len([_|T], Len_temp, Len) :-        % T - tail
        Temp = Len_temp + 1,        % Len_temp - temporary length
        len(T, Temp, Len),          % Len - length
        !.                          % Temp - temporary var
len([], Len, Len) :- !.             % Lst - currect list
len(Lst, Len) :- len(Lst, 0, Len), !.
```

Вопрос: len([5, -2, 0], Len).

№	Текущая резольвента - TP	ТЦ, выбираемые правила: сравниваемые термы, подстановка	Дальнейшие действия
0.	len([5, -2, 0], Len)		
1.	len([5, -2, 0], Len)	len([5, -2, 0], Len) = len([_|T], Len_temp, Len) (1) **Неудача** (разная арность)	Прямой ход, переход к следующему правилу.
	len([5, -2, 0], Len)	len([5, -2, 0], Len) = len([], Len, Len) (2) **Неудача** (разная арность)	Прямой ход, переход к следующему правилу.
	len([5, -2, 0], 0, Len_1), !	len([5, -2, 0], Len) = len(Lst_1, Len_1) (3) **Удача** **Подстановка:** {Lst_1=[5, -2, 0], Len=Len_1}	Прямой ход **Изменение резольвенты**: 1. применение редукции 2. применение подстановки
2.	Temp_2 = 0 + 1, len([-2,0], Temp_2, Len_2), !, !	len([5, -2, 0], 0, Len_1) = len([_|T_2], Len_temp_2, Len_2) (1) **Удача** **Подстановка:** {Lst_1=[5, -2, 0], Len=Len_1, T_2=[-2,0], Len_temp_2=0, Len_1=Len_2}	Прямой ход **Изменение резольвенты**: 1. применение редукции 2. применение подстановки

3.	len([-2,0], 1, Len_2), !, !	Temp_2 = 0 + 1 **Удача** **Подстановка:** {Lst_1=[5, -2, 0], Len=Len_1, T_2=[-2,0], Len_temp_2=0, Len_1=Len_2, Temp_2=1}	Прямой ход **Изменение резольвенты**: 1. применение редукции 2. применение подстановки
4.	Temp_4 = 1 + 1, len([0], Temp_4, Len_4), !, !, !	len([-2,0], 1, Len_2) = len([_|T_4], Len_temp_4, Len_4) (1) **Удача** **Подстановка:** {Lst_1=[5, -2, 0], Len=Len_1, T_2=[-2,0], Len_temp_2=0, Len_1=Len_2, Temp_2=1, T_4=[0], Len_temp_4=1, Len_2=Len_4}	Прямой ход **Изменение резольвенты**: 1. применение редукции 2. применение подстановки
5.	len([0], 2, Len_4), !, !, !	Temp_4 = 1 + 1 **Удача** **Подстановка:** {Lst_1=[5, -2, 0], Len=Len_1, T_2=[-2,0], Len_temp_2=0, Len_1=Len_2, Temp_2=1, T_4=[0], Len_temp_4=1, Len_2=Len_4, Temp_4=2}	Прямой ход **Изменение резольвенты**: 1. применение редукции 2. применение подстановки
6.	Temp_6 = 2 + 1, len([], Temp_6, Len_6), !, !, !, !	len([0], 2, Len_4) = len([_|T_6], Len_temp_6, Len_6) (1) **Удача** **Подстановка:** {Lst_1=[5, -2, 0], Len=Len_1, T_2=[-2,0], Len_temp_2=0, Len_1=Len_2, Temp_2=1, T_4=[0], Len_temp_4=1, Len_2=Len_4, Temp_4=2, T_6=[], Len_temp_6=2, Len_4=Len_6}	Прямой ход **Изменение резольвенты**: 1. применение редукции 2. применение подстановки
7.	len([], 3, Len_6), !, !, !, !	Temp_6 = 2 + 1 **Удача** **Подстановка:** {Lst_1=[5, -2, 0], Len=Len_1, T_2=[-2,0], Len_temp_2=0, Len_1=Len_2,	Прямой ход **Изменение резольвенты**: 1. применение редукции 2. применение подстановки

		Temp_2=1, T_4=[0], Len_temp_4=1, Len_2=Len_4, Temp_4=2, T_6=[], Len_temp_6=2, Len_4=Len_6, Temp_6=3}	
8.	len([], 3, Len_6), !, !, !, !	len([], 3, Len) = len([_|T], Len_temp, Len) (1) **Неудача** (пустой список)	Прямой ход, переход к следующему правилу.
	!, !, !, !, !	len([], 3, Len_6) = len([], Len_8, Len_8) (2) **Удача** **Подстановка:** {Lst_1=[5, -2, 0], Len=Len_1, T_2=[-2,0], Len_temp_2=0, Len_1=Len_2, Temp_2=1, T_4=[0], Len_temp_4=1, Len_2=Len_4, Temp_4=2, T_6=[], Len_temp_6=2, Len_4=Len_6, Temp_6=3, Len_8=3, Len_6=Len_8}	Прямой ход **Изменение резольвенты**: 1. применение редукции 2. применение подстановки
9.	!, !, !, !	! **Удача** **Подстановка:** без изменений	**Отсечение** (системный предикат отсечения) Прямой ход **Изменение резольвенты**: 1. применение редукции 2. применение подстановки
10.	!, !, !	! **Удача** **Подстановка:** без изменений	**Отсечение** (системный предикат отсечения) Прямой ход **Изменение резольвенты**: 1. применение редукции 2. применение подстановки
11.	!, !	! **Удача** **Подстановка:** без изменений	**Отсечение** (системный предикат отсечения) Прямой ход **Изменение резольвенты**: 1. применение редукции 2. применение подстановки
12.	!	! **Удача** **Подстановка:** без изменений	**Отсечение** (системный предикат отсечения) Прямой ход **Изменение резольвенты**:

			1. применение редукции 2. применение подстановки
13.	**Резольвента пуста**	! **Удача** **Подстановка:** без изменений	**Отсечение** (системный предикат отсечения) Прямой ход **Изменение резольвенты**: 1. применение редукции 2. применение подстановки **Вывод:** Len = 3 **Откат** (пустая резольвента)
14.	len([], 3, Len_6), !, !, !, !	**Подстановка:** {Lst_1=[5, -2, 0], Len=Len_1, T_2=[-2,0], Len_temp_2=0, Len_1=Len_2, Temp_2=1, T_4=[0], Len_temp_4=1, Len_2=Len_4, Temp_4=2, T_6=[], Len_temp_6=2, Len_4=Len_6, Temp_6=3}	**Откат** (отсечение)
15.	Temp_6 = 2 + 1, len([], Temp_6, Len_6), !, !, !, !	**Подстановка:** {Lst_1=[5, -2, 0], Len=Len_1, T_2=[-2,0], Len_temp_2=0, Len_1=Len_2, Temp_2=1, T_4=[0], Len_temp_4=1, Len_2=Len_4, Temp_4=2, T_6=[], Len_temp_6=2, Len_4=Len_6}	**Откат** (унификация с константой)
16.	len([0], 2, Len_4), !, !, !	**Подстановка:** {Lst_1=[5, -2, 0], Len=Len_1, T_2=[-2,0], Len_temp_2=0, Len_1=Len_2, Temp_2=1, T_4=[0], Len_temp_4=1, Len_2=Len_4, Temp_4=2}	**Откат** (отсечение)
17.	Temp_4 = 1 + 1, len([0], Temp_4, Len_4), !, !, !	**Подстановка:** {Lst_1=[5, -2, 0], Len=Len_1, T_2=[-2,0], Len_temp_2=0, Len_1=Len_2, Temp_2=1, T_4=[0], Len_temp_4=1, Len_2=Len_4}	**Откат** (унификация с константой)
18.	len([-2,0], 1, Len_2), !, !	**Подстановка:** {Lst_1=[5, -2, 0], Len=Len_1, T_2=[-2,0], Len_temp_2=0, Len_1=Len_2, Temp_2=1}	**Откат** (отсечение)
19.	Temp_2 = 0 + 1, len([-2,0], Temp_2, Len_2), !, !	**Подстановка:** {Lst_1=[5, -2, 0], Len=Len_1, T_2=[-2,0], Len_temp_2=0, Len_1=Len_2}	**Откат** (унификация с константой)

20.	len([5, -2, 0], 0, Len_1), !	**Подстановка:** {Lst_1=[5, -2, 0], Len=Len_1}	**Откат** (отсечение)
21.	len([5, -2, 0], Len)	**Подстановка:** {}	Завершение работы

Вопросы

1. Что такое рекурсия? Как организуется хвостовая рекурсия в Prolog? Как можно организовать выход из рекурсии в Prolog?

Рекурсия – это ссылка на описываемый объект в процессе его описания. При хвостовой рекурсии все действия сделаны до момента выхода из неё, вызов единственен. Выход из рекурсии организуется с помощью отсечения.

2. Какое первое состояние резольвенты?

Начальное состояние резольвенты – вопрос.

3. В каких пределах программы переменные уникальны?

Именованные переменные уникальны в пределах предложения. Анонимные переменные уникальны всегда.

4. В какой момент, и каким способом системе удаётся получить доступ к голове списка?

Получить голову можно при унификации списка ([H|T], где H – голова, T – хвост).

5. Каково назначение использования алгоритма унификации?

Алгоритм унификации используется для доказательства очередной цели.

6. Каков результат работы алгоритма унификации?

Алгоритм унификации делает вывод о том, унифицируемы два терма или нет, и если да, то строит наиболее общий унификатор.

7. Как формируется новое состояние резольвенты?

Резольвента меняется в два этапа:

1. В текущей резольвенте выбирается одна из целей, для неё выполняется редукция
2. Затем к резольвенте применяется подстановка, полученная, как наибольший общий унификатор цели и заголовка сопоставимого с ней правила.

8. Как применяется подстановка, полученная с помощью алгоритма унификации, как глубоко?

Подстановка - это множество пар вида {Xi = ti}. Применить подстановку, значит, найти все вхождения в резольвенте и результирующей ячейке Xi и заменить на соответствующее значение ti.Применяется на любую вложенность.

9. В каких случаях запускается механизм отката?

Механизм отката запускается в случаях, если резольвента оказалась пустой (то есть, будет воспроизведена попытка найти следующее подходящее знание), либо возникла тупиковая ситуация (просмотрена вся БЗ). В обоих случаях происходит откат к предыдущему состоянию резольвенты.

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

На формальном уровне это определяется тем, что в резольвенте находится исходный вопрос, для которого вся БЗ просмотрена. То есть система завершает работу в случае, когда все возможные ответы рассмотрены.