



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**  
*к лабораторной работе №17*  
*По курсу: «Функциональное и логическое  
программирование»*  
**Тема: «Обработка списков на Prolog»**

Студент: Зайцева А. А.  
Группа: ИУ7-62Б  
Преподаватели: Толпинская Н. Б.,  
Строганов Ю. В.

Москва, 2022 г.

## Практическая часть

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: Т.к. резольвента хранится в виде стека, то состояние резольвенты требуется 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	list = integer*.
predicates	length(list, integer). lengthInner(list, integer, integer).  sum(list, integer). sumInner(list, integer, integer).  oddSum(list, integer). oddSumInner(list, integer, integer).
clauses	lengthInner([], FinalAnswer, CurAnswer) :- FinalAnswer = CurAnswer, !.  lengthInner([_ Tail], FinalAnswer, CurAnswer) :- NextAnswer = CurAnswer + 1, lengthInner(Tail, FinalAnswer, NextAnswer).  length(List, Length) :- lengthInner(List, Length, 0).  sumInner([], FinalSum, CurSum) :- FinalSum = CurSum, !. sumInner([Head Tail], FinalSum, CurSum) :- NextSum = CurSum + Head, sumInner(Tail, FinalSum, NextSum).  sum(List, Sum) :- sumInner(List, Sum, 0).  % if list has even lenght oddSumInner([], FinalSum, CurSum) :- FinalSum = CurSum, !. % if list has odd lenght oddSumInner([_ []], FinalSum, CurSum) :- FinalSum = CurSum, !. oddSumInner([_ [NextHead NextTail]], FinalSum, CurSum) :- NextSum = CurSum + NextHead, oddSumInner(NextTail, FinalSum, NextSum), !.  oddSum(List, Sum) :- oddSumInner(List, Sum, 0).
goal	% length([], Length_). % Length_ = 0 % 1 Solution % length([1, 2, 3], Length_). % Length = 3

<pre> % 1 Solution  % sum([], Sum_). % Sum_=0 % 1 Solution % sum([-1, 1, 2], Sum_). % Sum_=2 % 1 Solution  % oddSum([], Sum_). % Sum_=0 % 1 Solution  % oddSum([0], Sum_). % Sum_=0 % 1 Solution  % oddSum([0, 1], Sum_). % Sum_=1 % 1 Solution  oddSum([0, 1, 2], Sum_). % Sum_=1 % 1 Solution  % oddSum([0, 1, 2, 3], Sum_). % Sum_=4 % 1 Solution  % oddSum([0, 1, 2, 3, 4], Sum_). % Sum_=4 % 1 Solution </pre>
---

Вопрос: oddSum([0, 1, 2], Sum\_).

№ шага	Текущая резолювента-ТР	ТЦ, выбираемые правила: сравниваемые термы, подстановка	Дальнейшие действия с комментариями
0	oddSum([0, 1, 2], Sum_)		Запуск алгоритма унификации для вопроса, с начала БЗ
1	oddSum([0, 1, 2], Sum_)	oddSum([0, 1, 2], Sum_)= lengthInner([], FinalAnswer, CurAnswer) Сравнение главных функторов: oddSum = lengthInner Унификация неуспешна	Резолювента не меняется (неуспешная унификация с заголовком)  Прямой ход, переход к следующему предложению БЗ
...			
10	oddSum([0, 1, 2], Sum_)	oddSum([0, 1, 2], Sum_)= oddSum(List, Sum)  Унификация успешна  Подстановка: { List=[0, 1, 2]}	Верхняя подцель успешно унифицировалась с заголовком правила. Образование новой резолювенты:  1. Редукция верхней подцели: замена oddSum([0, 1, 2], Sum_) телом найденного правила: oddSumInner(List, Sum, 0)  2. Применение подстановки к полученной конъюнкции целей. Новое состояние резолювенты: oddSumInner([0, 1, 2], Sum, 0) Резолювента непуста, запуск алгоритма унификации для верхней подцели с начала БЗ
...			
17	oddSumInner([0, 1, 2], Sum, 0)	oddSumInner([0, 1, 2], Sum, 0)= oddSumInner([], FinalSum, CurSum) Унификация неуспешна	Прямой ход, переход к следующему предложению БЗ
18	oddSumInner([0, 1, 2], Sum, 0)	oddSumInner([0, 1, 2], Sum, 0)= oddSumInner([_ []], FinalSum, CurSum) Унификация неуспешна	Прямой ход, переход к следующему предложению БЗ
19	oddSumInner([0, 1, 2], Sum, 0)	oddSumInner([0, 1, 2], Sum, 0)= oddSumInner([_ NextHead NextTail]], FinalSum, CurSum)  Унификация успешна  Подстановка: { NextHead=1, NextTail=[2], CurSum=0}	Верхняя подцель успешно унифицировалась с заголовком правила. Образование новой резолювенты:  1. Редукция верхней подцели: замена oddSumInner([0, 1, 2], Sum, 0) телом найденного правила: NextSum = CurSum + NextHead, oddSumInner(NextTail, FinalSum, NextSum), !.  2. Применение подстановки к полученной конъюнкции целей. Новое состояние резолювенты: NextSum = 0 + 1,

			<p>oddSumInner([2], FinalSum, NextSum), !.</p> <p>Резольвента непуста, запуск алгоритма унификации для верхней подцели</p>
20	<p>NextSum = 0 + 1, oddSumInner([2], FinalSum, NextSum), !.</p>	<p>NextSum = 0 + 1</p> <p>Унификация успешна</p> <p>Подстановка: { NextSum=1}</p>	<p>Образование новой резольвенты:</p> <p>1. Редукция верхней подцели: удаление NextSum = 0 + 1</p> <p>2. Применение подстановки к полученной конъюнкции целей. Новое состояние резольвенты:</p> <p>oddSumInner([2], FinalSum, 1), !.</p> <p>Резольвента непуста, запуск алгоритма унификации для верхней подцели с начала БЗ</p>
...			
27	<p>oddSumInner([2], FinalSum, 1), !.</p>	<p>oddSumInner([2], FinalSum, 1)= oddSumInner([], FinalSum, CurSum)</p> <p>Унификация неуспешна</p>	<p>Прямой ход, переход к следующему предложению БЗ</p>
28	<p>oddSumInner([2], FinalSum, 1), !.</p>	<p>oddSumInner([2], FinalSum, 1)= oddSumInner([], FinalSum, CurSum)</p> <p>Унификация успешна</p> <p>Подстановка: { CurSum=1 }</p>	<p>Верхняя подцель успешно унифицировалась с заголовком правила. Образование новой резольвенты:</p> <p>1. Редукция верхней подцели: замена oddSumInner([2], FinalSum, 1) телом найденного правила: FinalSum = CurSum, !</p> <p>2. Применение подстановки к полученной конъюнкции целей. Новое состояние резольвенты: FinalSum = 1, !, !.</p> <p>Резольвента непуста, запуск алгоритма унификации для верхней подцели</p>
29	<p>FinalSum = 1, !, !.</p>	<p>FinalSum = 1</p> <p>Унификация успешна</p> <p>Подстановка: { FinalSum = 1}</p>	<p>Образование новой резольвенты:</p> <p>1. Редукция верхней подцели: удаление FinalSum = 1</p> <p>2. Применение подстановки к полученной конъюнкции целей. Новое состояние резольвенты:</p> <p>!, !.</p> <p>Резольвента непуста, запуск алгоритма унификации для верхней подцели</p>
30	<p>!, !.</p>	<p>!</p> <p>Истина</p>	<p>Встречен системный предикат отсечения, который на обратном ходе запрещает использование других правил процедуры oddSumInner. Редукция верхней подцели</p>
31	<p>!</p>	<p>!</p> <p>Истина</p>	<p>Встречен системный предикат отсечения, который на обратном ходе запрещает использование других правил процедуры oddSumInner. Редукция верхней подцели</p>
32			<p>Резольвента пуста. Решение найдено: формируется подстановка { Sum_=1} в качестве побочного эффекта.</p> <p>Восстановление предыдущего состояния резольвенты, дважды ! запрещает использование других правил процедуры oddSumInner, новое состояние резольвенты oddSum([0, 1, 2], Sum_). Конец БЗ. Завершение работы.</p>