



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»

---

## Отчёт

### по лабораторной работе №17

Название «Обработка списков на Prolog»

---

Дисциплина «Функциональное и логическое программирование»

---

Студент ИУ7-65Б

---

(подпись, дата)

Клименко А.К.

---

(Фамилия И.О.)

Преподаватель

---

(подпись, дата)

Толпинская Н.Б.

---

(Фамилия И.О.)

Москва, 2022

## Введение

**Цель работы** – изучить способы организации, представления и обработки списков в программах на Prolog, методы создания эффективных рекурсивных программ обработки списков и порядок их реализации.

Для достижения поставленной цели необходимо решить следующие задачи:

- приобрести навыки использования рекурсии на Prolog, эффективного способа ее организации и порядка работы соответствующей программы.
- Изучить особенность использования переменных при обработке списков.
- Изучить способ формирования и изменения резольвенты в этом случае и порядок формирования ответа.

## Теоритические вопросы

1) *Что такое рекурсия? Как организуется хвостовая рекурсия в Prolog? Как организовать выход из рекурсии в Prolog?*

**Ответ:** Рекурсия – это один из способов организации повторных вычислений. Т.к. логическое программирование – не операторное, то рекурсия – это способ заставить систему использовать многократно одну и ту же процедуру. Хвостовая рекурсия в прологе организуется с помощью использования в определении знания того же знания.

2) *Какое первое состояние резольвенты?*

**Ответ:** первое состояние резольвенты представляет собой вопрос.

3) *В каком случае система запускает алгоритм унификации? Каково назначение использования алгоритма унификации? Каков результат работы алгоритма унификации?*

**Ответ:** алгоритм унификации запускается системой в случае необходимости проверить, подходит ли текущее правило в базе знаний для доказательства текущей цели. Результат алгоритма унификации представляет ответ да или нет. При ответе да результатом также является подстановка, сформированная в процессе работы алгоритма.

4) *В каких пределах программы уникальны переменные?*

**Ответ:** именованные переменные уникальны в пределах предложения, а анонимные переменные уникальны всегда.

5) *Как применяется подстановка, полученная с помощью алгоритма унификации?*

**Ответ:** все переменные, содержащиеся в постановке и в термах резольвенты, заменяются в резольvente на соответствующие значения для этих переменных.

6) *Как изменяется резольвента?*

**Ответ:** при нахождении подходящего правила для первого терма резольвенты он заменяется на тело правила.

7) *В каких случаях запускается механизм отката?*

**Ответ:** механизм отката запускается в случае, когда система попадает в тупиковое состояние – резольвента не пуста, но вся база знаний уже была просмотрена с целью подбора знания для текущей цели доказательства.

## **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 каждого задания составить таблицу, отражающую конкретный порядок работы системы.

## Текст программы

```
1 domains
2     num = integer
3     elemList = num*
4
5 predicates
6     length(num, elemList).
7     sum(num, elemList).
8     sum_odd_id(num, elemList).
9
10 clauses
11     length(0, []) :- !.
12     length(R, [_|T]) :-
13         length(R_1, T),
14         R = R_1 + 1.
15
16     sum(0, []) :- !.
17     sum(R, [R]) :- !.
18     sum(R, [H|T]) :-
19         sum(R1, T),
20         R = H + R1.
21
22     sum_odd_id(0, []) :- !.
23     sum_odd_id(0, [_]) :- !.
24     sum_odd_id(R, [_, R]) :- !.
25     sum_odd_id(R, [_, H|T]) :-
26         sum_odd_id(R1, T),
27         R = R1 + H.
28
29 goal
30     sum_odd_id(R, [1, 2, 3, 4, 5]).
```

## Порядок поиска ответа для вопросов

Порядок поиска ответа для вопроса:  $\text{length}(R, [2, 1, 5])$

№ шага	Сравниваемые термы; результат; подстановка	Дальнейшие действия	Резольвента
1	$T1 = \text{length}(R, [2, 1, 5])$ $T2 = \text{length}(0, [])$ не унифицированы	Переход к следующему правилу в базе знаний.	$\text{length}(R, [2, 1, 5])$
2	$T1 = \text{length}(R, [2, 1, 5])$ $T2 = \text{length}(R, [_ T])$ унифицированы $\theta = \{R=R, T=[1, 5]\}$	Прямой ход. Замена терма резольвенты на тело правила.	$\text{length}(R_1, [1, 5])$ $R = R_1 + 1$
3	$T1 = \text{length}(R_1, [1, 5])$ $T2 = \text{length}(0, [])$ не унифицированы	Переход к следующему правилу в базе знаний.	$\text{length}(R_1, [1, 5])$ $R = R_1 + 1$
4	$T1 = \text{length}(R_1, [1, 5])$ $T2 = \text{length}(R', [_ T])$ унифицированы $\theta = \{R'=R_1, T=[5]\}$	Прямой ход. Замена терма резольвенты на тело правила.	$\text{length}(R_1', [5])$ $R_1 = R_1' + 1$ $R = R_1 + 1$
5	$T1 = \text{length}(R_1', [5])$ $T2 = \text{length}(0, [])$ не унифицированы	Переход к следующему правилу в базе знаний.	$\text{length}(R_1', [5])$ $R_1 = R_1' + 1$ $R = R_1 + 1$
6	$T1 = \text{length}(R_1', [5])$ $T2 = \text{length}(R'', [_ T])$ унифицированы $\theta = \{R''=R_1', T=[]\}$	Прямой ход. Замена терма резольвенты на тело правила.	$\text{length}(R_1'', [])$ $R_1' = R_1'' + 1$ $R_1 = R_1' + 1$ $R = R_1 + 1$
7	$T1 = \text{length}(R_1'', [])$ $T2 = \text{length}(0, [])$ унифицированы $\theta = \{R_1''=0\}$	Подобран факт. Удаление терма резольвенты, переход к доказательству следующего терма.	$R_1' = 0 + 1$ $R_1 = R_1' + 1$ $R = R_1 + 1$
8	$R_1' = 0 + 1$ $\theta = \{R_1'=1\}$	Удаление терма резольвенты. Конкретизация значения переменной $R_1'$	$R_1 = 1 + 1$ $R = R_1 + 1$
9	$R_1 = 1 + 1$ $\theta = \{R_1=2\}$	Удаление терма резольвенты. Конкретизация значения переменной $R_1$	$R = 2 + 1$
10	$R = 2 + 1$ $\theta = \{R=3\}$	Резольвента пуста. Вывод результата на экран. Остановка алгоритма.	

## **Заключение**

В ходе работы были приобретены знания о способах организации эффективных программ на Prolog. Были изучены особенности использования системных предикатов и порядок выполнения программ с их использованием.

Эффективность программ написанных на языке Prolog может быть достигнута за счет изменения порядка следования фактов и правил в базе знаний, а также за счёт использования системных предикатов, которые позволяют сократить количество достигаемых в процессе поиска решения тупиковых состояний.