|  |  |
| --- | --- |
|  | **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 №19*

*По курсу: «Функциональное и логическое программирование»*

Студент ИУ7-62Б

Блохин Д.М.

Преподаватели

Толпинская Н.Б.

Строганов Ю.В.

*Москва, 2021 г.*

Оглавление

[Задание 3](#_Toc39943141)

[Вопросы 3](#_Toc39943142)

[Листинг 6](#_Toc39943143)

[Таблица 8](#_Toc39943144)

# Задание

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

* Найти длину списка (по верхнему уровню);
* Найти сумму элементов числового списка
* 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.

**Текст процедуры, Вопрос:…..**

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Текущая резольвента – ТР | ТЦ, выбираемые правила: сравниваемые термы,  подстановка | Дальнейшие действия с комментариями |
| шаг1 | … | … | … |
| … | … | … | … |

# Вопросы

**Что такое рекурсия?**

Рекурсия – это ссылка на описываемый объект при описании объекта.

**Как организуется хвостовая рекурсия в Prolog?**

* Рекурсивный вызов единственен и расположен в конце тела правила
* Не должно быть возможности сделать откат до вычисления рекурсивного вызова

**Как организовать выход из рекурсии в Prolog?**

С помощью отсечения.

**Какое первое состояние резольвенты?**

Заданный вопрос (goal).

**В какой момент, и каким способом системе удается получить доступ к голове списка?**

Получить голову или хвост списка можно при унификации списка с [H|T], H – голова списка, T – хвост списка.

**Каково назначение и результат использования алгоритма унификации?**

Унификация – механизм логического вывода. Результат – подстановка.

**В каких пределах программы переменные уникальны?**

Именованная переменная уникальна в рамках предложения, в котором она используется. Анонимные переменные всегда уникальны.

**Как формируется новое состояние резольвенты?**

Преобразования резольвенты выполняются с помощью редукции. Редукцией цели G с помощью программы P называется замена цели G телом того правила из P, заголовок которого унифицируется с целью. Новая резольвента образуется в два этапа:

1. в текущей резольвенте выбирается одна из подцелей и для неё выполняется редукция;
2. к полученной конъюнкции целей применяется подстановка, полученная как наибольший общий унификатор цели и заголовка сопоставленного с ней правила.

**Как применяется подстановка, полученная с помощью алгоритма унификации? Как глубоко?**

Подстановка применяется к целям в резольвенте путем замены текущей переменной на соответствующий терм. В результате применения подстановки некоторые переменные конкретизируются значениями, которые (значения) могут и будут далее использованы при доказательстве истинности тела выбранного правила.

**В каких случаях запускается механизм отката?**

Механизм отката запустится в случае неудачи алгоритма унификации.

**Когда останавливается работа системы?**

Работа системы останавливается, когда найдены все возможные ответы на вопрос.

**Как это определяется на формальном уровне?**

Когда в резольвенте находится исходный вопрос, для которого пройдена вся БЗ.

# Листинг

# 

Эффективность достигнута за счет использования хвостовой рекурсии и использования отсечения.

# Таблица

**Текст процедуры**

List\_Len(List, Len) :- list\_Len(List, 0, Len).

List\_Len([], Len, Len) :- !.

List\_Len([\_|T], Cur\_Len, Len) :-

New\_Len = Cur\_Len + 1,

List\_Len(T, New\_Len, Len).

**Вопрос: list\_Len([1], Len)**

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Текущая резольвента – ТР | ТЦ, выбираемые правила: сравниваемые термы,  подстановка | Дальнейшие действия с комментариями |
| 1 | **list\_Len([1,], Len)** | ТЦ: **list\_Len([1,], Len)** | Поиск знания с начала БЗ |
|  | List\_Len([1], Len) | ПР1:list\_Len([1], Len) =  list\_Len(List, Len)  Успех  List = [1] Len = Len | Тело ПР1 заменяет цель в резольвенте |
| 2 | List\_Len([1], 0, Len) | ПР2:List\_Len([1], 0, Len) = list\_Len([], Len, Len)  Неудача | Метка переноситься ниже |
| 3 | List\_Len([1], 0, Len) | ПР3:list\_Len([1], 0, Len) = list\_Len([\_|T], Cur\_Len, Len)  Успех  T = [] Cur\_Len = 0 Len = Len | Прямой ход.  Тело правила заносится в резольвенту. |
| 4 | New\_Len = Cur\_Len +1 list\_Len(T, New\_Len, Len) | New\_Len = 0 +1 = 1 | Прямой ход. |
| 5 | List\_Len(T, New\_Len, Len) | ПР1:List\_Len([], 1, Len) = list\_Len([], Len, Len)  Успех  Len = Len = 1 | Прямой ход.  Тело правила заносится в резольвенту. |
| 6 | ! |  | Резольвента пуста.  Найдено решение:  Len = 1  Ввиду отсечения не будет попыток найти другие решения list\_Len([], 1, Len)  Откат к 1. Конец list\_Len арности 2.  Система завершает работу |