|  |  |
| --- | --- |
|  | **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 №20*

*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](#_Toc40024849)

[Вопросы 4](#_Toc40024850)

[Листинг 5](#_Toc40024851)

[Таблица 7](#_Toc40024852)

# Задание

Используя хвостовую рекурсию, разработать, комментируя аргументы, эффективную программу, позволяющую:

* Сформировать список из элементов числового списка, больших заданного значения;
* Сформировать список из элементов, стоящих на нечетных позициях исходного списка (нумерация от 0);
* Удалить заданный элемент из списка (один или все вхождения);
* Преобразовать список в множество (можно использовать ранее разработанные процедуры).

Убедиться в правильности результатов

Для одного из вариантов ВОПРОСА и 1-ого задания 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 соответствующий вывод: успех или нет –и почему.

**Текст процедуры …; Вопрос:…..**

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

# Вопросы

**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).

**Каким способом можно разделить список на части, какие, требования к частям?**

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

**Как выделить за один шаг первые два подряд идущих элемента списка?**

[H1|[H2|\_]]

**Как выделить 1-й и 3-й элемент за один шаг?**

[H1|[\_|[H3|\_]]]

**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.

**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**

1: is\_Bigger\_Than([], \_, []).

2: is\_Bigger\_Than([H|T], Min, [H|Res\_T]) :-

H > Min,

is\_Bigger\_Than(T, Min, Res\_T).

3: is\_Bigger\_Than([\_|T], Min, Res\_T) :-

is\_Bigger\_Than(T, Min, Res\_T).

**Вопрос:** is\_Bigger\_Than **([1], 2, Res)**

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Текущая резольвента – ТР | ТЦ, выбираемые правила: сравниваемые термы,  подстановка | Дальнейшие действия с комментариями |
| 1 | is\_Bigger\_Than([1], 2, Res) | ТЦ: is\_Bigger\_Than([1], 2, Res) | Поиск знания с начала БЗ |
|  | is\_Bigger\_Than([1], 2, Res) | ПР1:  [] = [1] \_ = 2 [] = Res  Неудача | Метка переносится ниже |
|  | is\_Bigger\_Than([1], 2, Res) | ПР2:  [H1|T1] = [1] Min1 = 2 [H1|Res\_T] = Res  Успех  H1 = 1 T1 = [] Min1 = 2 Res = [1|Res\_T] | Тело ПР2 заменяет цель в резольвенте |
| 2 | 1 > 2 is\_Bigger\_Than([], 2, Res\_T1) ! | Сравнение:  1 > 2  Ложь | Откат к 1.  Метка переносится ниже. |
| 3 | is\_Bigger\_Than([1], 2, Res) | ПР3:  [\_|T3] = [1] Min3 = 2 Res\_T3 = Res  Успех  T3 = [] Min3 = 2 Res = Res\_T3 | Тело ПР3 заменяет цель в резольвенте |
| 4 | is\_Bigger\_Than([], 2, Res\_T3) | ТЦ: is\_Bigger\_Than([], 2, Res\_T3) | Поиск знания с начала БЗ |
|  | is\_Bigger\_Than([], 2, Res\_T3) | ПР1: [] = [] \_ = 2 [] = Res\_T3  Успех  Res\_T3 = 0 | Пустое тело заменяет цель в резольвенте |
|  | Пусто |  | Успех.  Res = Res\_T3 = []  Возврат к предыдущему состоянию резольвенты |
| 5 | is\_Bigger\_Than([], 2, Res\_T3) | ПР2: [H5|T5] = [] Min5 = 2 [H5|ResTail5] = Res\_T3  Неудача | Метка переносится ниже. |
|  | is\_Bigger\_Than([], 2, Res\_T3) | ПР3:  [\_|T3] = [] Min5 = 2 Res\_T5 = Res  Неудача | Необходимо включить откат, но метки в конце процедур – других альтернатив нет.  Система завершает работу. |