|  |  |
| --- | --- |
|  | **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»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ДИСЦИПЛИНА «Функциональное и логическое программирование»

**Лабораторная работа № 20**

**“Формирование и модификация списков на Prolog”**

|  |  |
| --- | --- |
| **Студент**  Чалый А. А.  **Группа** ИУ7 – 62 Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Толпинская Н.Б. |  |

Москва.

2021 г.

**Цель работы:**

Изучить способы формирования и модификации списков в Prolog, эффективные методы обработки списков и порядок реализации рекурсивных программ.

**Задачи работы**:

Приобрести навыки формирования и модификации списков на Prolog, эффективного способа их обработки, организации и порядка работы соответствующих программ.

Изучить особенность использования переменных при обработке списков. Способ формирования и изменения резольвенты в этом случае и порядок формирования ответа.

**Задание:**

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

1. Сформировать список из элементов числового списка, больших заданного значения;
2. Сформировать список из элементов, стоящих на нечетных позициях исходного списка (нумерация от 0);
3. Удалить заданный элемент из списка (один или все вхождения);
4. 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.

Листинг 1. list\_20.pro

domains

list = integer\*.

predicates

bigger(list, integer, list)

oddlist(list, list)

delete(list, integer, list)

createSet(list, list)

clauses

bigger([], \_ ,[]):-!.

bigger([H|T], Num, Res) :- H <= Num, bigger(T, Num, Res).

bigger([H|T], Num, [H|Res]) :- H > Num, bigger(T, Num, Res).

oddlist([], []):- !.

oddlist([\_], []):- !.

oddlist([\_|[H|T]], [H|Res]):- oddlist(T, Res).

delete([], \_ , []):- !.

delete([H|T], Num, [H|Res]) :- H <> Num, delete(T, Num, Res).

delete([H|T], H, Res):- delete(T,H, Res).

createSet([], []):- !.

createSet([H|T], [H| Res]):- delete(T, H, Tmp), createSet(Tmp, Res).

Примеры работы:

Сформировать список из элементов числового списка, больших заданного значения;

bigger([5, 4, 3, 2, 1], 2, Res). → [5, 4, 3]

Сформировать список из элементов, стоящих на нечетных позициях исходного списка (нумерация от 0);

oddlist([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], Res). → [1, 3, 5, 7, 9]

oddlist([0, 1, 2, 3, 4, 5, 6, 7, 8], Res). → [1, 3, 5, 7]

Удалить заданный элемент из списка (один или все вхождения);

delete([1, 2, 2, 2, 2, 2], 2, Res). → [1]

delete([2, 2, 2, 2, 2], 2, Res). → []

Преобразовать список в множество (можно использовать ранее разработанные процедуры).

createSet([1, 2, 3, 3, 2, 1], Res). → [1, 2, 3]

Таблица 1. bigger([1,3], 2, Res).

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Текущая резольвента – ТР | ТЦ, выбираемые правила: сравниваемые термы,  подстановка | Дальнейшие действия с комментариями |
| 1 | bigger([1,3], 2, Res). | ТЦ: bigger([1,3], 2, Res). | Поиск знания с начала базы знаний. |
|  | bigger([1,3], 2, Res). | ПР1:  [] = [1,3]  \_ = 2  Res = []  Неудача | Метка переносится ниже |
|  | bigger([1,3], 2, Res). | ПР2:  [H|T] = [1,3]  Num = 2  Res = Res  Успех  Подстановка: {  Н = 1, T = [3], Num = 2, Res= Res  } | Тело ПР2 заменяет цель в резольвенте |
| 2 | 1 <= 2  bigger([3], 2, Res) | 1 <= 2  успех | Переход к следующей подцели |
| 3 | bigger([3], 2, Res) | ТЦ: Bigger([3], 2, Res) | Поиск знания с начала базы знаний. |
|  | bigger([3], 2, Res) | ПР1:  [] = [3]  \_ = 2  Res = []  Неудача | Метка переносится ниже |
|  | bigger([3], 2, Res) | ПР2:  [H|T] = [3]  Num = 2  Res = Res  Успех  Подстановка: {  Н = 3, T = [], Num = 2, Res= Res  } | Тело ПР2 заменяет цель в резольвенте |
| 4 | 3 <= 2  bigger([], 2, Res). | 3 <= 2  Ложь | Откат к шагу 3.  Метка переносится ниже. |
| 5 | bigger([3], 2, Res) | ПР3:  [H|T] = [3]  Num = 2  [H|Res] = Res  Успех  Подстановка: {  Н = 3, T = [], Num = 2, Res=[3 | Res]  } | Тело ПР3 заменяет цель в резольвенте |
| 6 | 3 > 2  bigger([], 2, Res). | 3 > 2  успех | Переход к следующей подцели |
| 7 | bigger([], 2, Res). | ТЦ : bigger([], 2, Res). | Поиск знания с начала базы знаний. |
|  | bigger([], 2, Res). | ПР1:  [] = []  \_ = 2  Res = []  Успех  Подстановка:  {Res = []} | Тело ПР1 заменяет цель в резольвенте |
| 8 | ! |  | Так как встречен знак отсечения не будет попыток найти другие решения.  Система завершает работу.  Найдено решение  Res = [3|[]] = [3] |

**Вывод:**

Эффективность работы достигнута за счет использования хвостовой рекурсии и отсечений. Также разбиения на части и проверки выполняются в заголовке правила. Все это позволяет уменьшить количество шагов, необходимых для достижения результата.

**Контрольные 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

**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.