



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 №18 по курсу «Функциональное и логическое программирование»

Тема Формирование и модификация списков на Prolog

Студент Прянишников А.Н.

Группа ИУ7-65Б

Оценка (баллы) _____

Преподаватели Строганов Ю. В., Толпинская Н. Б.

Практическая часть

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).

На листинге 1 представлен код программы:

```
1 bigger_than([Head | Tail], N, [Head | ResTail]) :- Head > N, !, bigger_than(Tail, N,
    ResTail).
2 bigger_than([_ | Tail], N, Result) :- bigger_than(Tail, N, Result).
3 bigger_than([], _, []).
4
5 odd_list([_, Head | Tail], [Head | ResTail]) :- !, odd_list(Tail, ResTail).
6 odd_list([], []).
7
8 single_del([Head | Tail], N, Tail) :- Head is N, !.
9 single_del([Head | Tail], N, [Head | ResTail]) :- single_del(Tail, N, ResTail), !.
10 single_del([], _, []).
11
12 full_del([Head | Tail], N, [Head | ResTail]) :- Head <> N, !, full_del(Tail, N, ResTail).
13 full_del([_ | Tail], N, Result) :- full_del(Tail, N, Result), !.
14 full_del([], _, []).
15
16 set([Head | Tail], [Head | Result]) :- full_del(Tail, Head, Nt), !, set(Nt, Result).
17 set([], []).
18
19 %bigger_than([1, 7, 3, 4, 5, 6], 3, Result).
20 %odd_list([1, 2, 3, 4, 5, 6, 7, 8], Result).
21
22 %single_del([1, 2, 3, 1, 2, 3, 1, 2, 3], 1, Result).
23 %full_del([1, 2, 3, 1, 2, 3, 1, 2, 3], 1, Result).
24
25 %set([1, 2, 3, 1, 2, 3, 1, 2, 3], Result).
```

Теоретические вопросы

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

Рекурсия – это один из способов организации повторных вычислений. Рекурсия – это способ заставить систему использовать многократно одну и ту же процедуру.

Для организации хвостовой рекурсии рекурсивный вызов должен быть последней подцелью и нужно избавиться от точек возврата с помощью отсечения, чтобы исключить возможные альтернативы.

2. Какое первое состояние резольвенты?

Стек, который содержит конъюнкцию целей, истинность которых система должна доказать, называется резольвентой. Первое состояние резольвенты - вопрос.

3. В каком случае система запускает алгоритм унификации? (Как эту необходимость на формальном уровне распознает система?)

Унификация – необходима для того, чтобы определить дальнейший путь поиска решений. Унификация заканчивается конкретизацией части переменных.

4. Каковы назначение и результат использования алгоритма унификации?

Алгоритм унификации – основной шаг с помощью которого система отвечает на вопросы унификации. На вход алгоритм принимает два терма, возвращает флаг успешности унификации, и если успешно, то подстановку.

5. В каких пределах программы уникальны переменные?

Областью действия переменной в Прологе является одно предложение. В разных предложениях может использоваться одно имя переменной для обозначения разных объектов.

6. В каких случаях запускается механизм отката?

Механизм отката, который осуществляет откат программы к той точке, в которой выбирался унифицирующийся с последней подцелью дизъюнкт. Для этого точка, где выбирался один из возможных унифицируемых с подцелью дизъюнктов, запоминается в специальном стеке, для последующего возврата к ней и выбора альтернативы в случае неудачи. При откате все переменные, которые были означены в результате унификации после этой точки, опять становятся свободными.