



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 по курсу «Функциональное и логическое программирование»

Тема Обработка списков на Prolog

Студент Богаченко А. Е.

Группа ИУ7-65Б

Оценка (баллы) _____

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

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);

```
1 domains
2   intlist = integer*
3
4 predicates
5   rec_length(integer, integer, intlist)
6   length(integer, intlist)
7
8   rec_sum(integer, integer, intlist)
9   sum(integer, intlist)
10
11  rec_oddsum(integer, integer, intlist)
12  oddsum(integer, intlist)
13
14 clauses
15  rec_length(Res, Len, [_ | Tail]) :- NewLen = Len + 1, !, rec_length(Res, NewLen,
16    ↪ Tail).
17  rec_length(Res, Len, []) :- Res = Len.
18  length(Res, List) :- rec_length(Res, 0, List).
19
20  rec_sum(Res, Sum, [Head | Tail]) :- NewSum = Sum + Head, !, rec_sum(Res, NewSum,
21    ↪ Tail).
22  rec_sum(Res, Sum, []) :- Res = Sum.
23  sum(Res, List) :- rec_sum(Res, 0, List).
24
25  rec_oddsum(Res, Sum, [_ , Head | Tail]) :- NewSum = Sum + Head, !, rec_oddsum(Res,
26    ↪ NewSum, Tail).
27  rec_oddsum(Res, Sum, []) :- Res = Sum.
28  oddsum(Res, List) :- rec_oddsum(Res, 0, List).
29
30 goal
31   %length(Res, [1, 2, 3, 4]).
32   %sum(Res, [1, 2, 3, 4]).
33   oddsum(Res, [1, 2, 3, 4]).
```

Теоретические вопросы

1. Что такое рекурсия? Как организуется хвостовая рекурсия в Prolog? Как можно 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 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.

4. В какой момент, и каким способом системе удастся получить доступ к голове списка?

Для разделение списка на «голову» и «хвост» используется специальный символ |

5-6. 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.

7. Как формируется новое состояние резольвенты?

Резольвента меняется в два этапа:

1. редукция – замена подцели телом того правила, с заголовком которого успешно унифицируется данная подцель;
2. применение ко всей резольвенте подстановки.

Резольвента уменьшается, если удаётся унифицировать подцель с фактом. Система отвечает «Да», только когда резольвента становится пустой.

8. Как применяется подстановка, полученная с помощью алгоритма унификации – как глубоко?

Пусть дан терм: (X_1, X_2, \dots, X_n) , подстановка – множество пар, вида: $X_i = t_i$, где X_i – переменная, а t_i – терм. В ходе выполнения программы выполняется связывание переменных с различными объектами, этот процесс называется конкретизацией. Это относится только к именованным переменным. Анонимные переменные не могут быть связаны со значением.

9. 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.

10. Когда останавливается работа системы? Как это определяется на формальном уровне?