

1. Текст упражнения.

Упражнение: проанализируйте работу одного из основных предикатов обработки списков, выполнив его в режиме трассировки. Какие задачи можно решать, используя данный предикат? Придумайте цели для решения каждой задачи и начертите деревья поиска решений.

Предикат соединения двух списков:

`conc([],L,L).`

`conc([H|T],L,[H|T1]):-`

`conc(T,L,T1).`

Цель 1 - соединить два списка

`conc([1,2,3],[4,5],X)`

Ответ: $X = [1, 2, 3, 4, 5]$

Цель 2 - проверить является ли список результатом соединения других

`conc([1, 2, 3], [4, 5], [1, 2, 5]).`

Ответ: NO

2. Текст индивидуального задания.

Написать программу, которая вставляет:

а) в список L новый элемент X перед всеми вхождениями элемента Y , если Y входит в L ;

б) в непустой список L пару новых элементов $X1$ и $X2$ перед его последним элементом;

в) в непустой список L , элементы которого упорядочены по возрастанию, новый элемент X так, чтобы сохранилась упорядоченность,

3. Текст программы.

domains

list=integer*

predicates

paste_a(list, integer, integer , list)

paste_b(list, integer, integer , list)

paste_c(list,list,integer)

clauses

paste_a([],_,_,[]).

paste_a([H|T],H,Y,[Y,H|T1]):-paste_a(T,H,Y,T1),!.

paste_a([H|T],X,Y,[H|T1]):-paste_a(T,X,Y,T1).

paste_b([H],X,Y,[X,Y,H]):- !.

paste_b([H|T],X,Y,[H|T1]):- paste_b(T,X,Y,T1).

paste_c([],[X|[]],X).

paste_c([H|T],[X,H|T],X):-X<H,!.

paste_c([H|T],[H|T1],X):-paste_c(T,T1,X).

4. Описание задач, решаемых данной программой.

а) вставить в список L новый элемент X перед всеми вхождениями элемента Y, если Y входит в L;

goal

paste_a([1,2,3,4,1,1,3],1,5,L).

L=[5,1,2,3,4,5,1,5,1,3]

1 Solution

б) в непустой список L вставить пару новых элементов X1 и X2 перед его последним элементом;

goal

`paste_b([1,2,3,4,1,1,3],4,5,L).`

`L=[5,1,2,3,4,5,1,5,1,3]`

1 Solution

в) в непустой список L, элементы которого упорядочены по возрастанию, вставить новый элемент X так, чтобы сохранилась упорядоченность

goal

`paste_c([1,2,3,5],L,5).`

`L=[1,2,3,4,5]`

1 Solution