

2008 Paper 4 Question 8

Prolog

The Prolog predicate `perm(+In,-Out)` generates all permutations of the input list `In`. A programmer implements `perm/2` as follows:

```
perm([], []).  
perm(L, [H|T]) :- take(L,H,R), perm(R,T).
```

The predicate `take(+L,-E,-R)` removes one element (`E`) from the input list `L` and unifies `R` with the remainder of `L`. Thus, the list `R` has one element fewer than `L`.

(a) Consider the `perm/2` predicate:

- (i) Explain briefly in words the operation of the `perm/2` predicate. [3 marks]
- (ii) Provide an implementation of the `take/3` predicate. [4 marks]
- (iii) Give the complete sequence of answers (in the correct order) generated by `perm([1,2,3],A)`. [3 marks]

(b) A student attempts to invoke the query `perm(A, [1,2,3])`.

- (i) Explain what happens and why. [5 marks]
- (ii) Implement a predicate `sameLength/2` which is true if the two parameters are lists of the same length. [2 marks]
- (iii) Using `sameLength/2`, or otherwise, provide an implementation of `safePerm/2` which generates permutations regardless of the order in which the parameters are provided: both `safePerm(+In,-Out)` and `safePerm(-Out,+In)` should generate all permutations of `In`. The order in which these permutations are generated is not important. [3 marks]