2011 Paper 3 Question 8

Prolog

(*a*) Give the result and any variable bindings that occur from making each of the following (independent) queries:

(*i*) `A=3` [1 mark]

(*ii*) `A is 4` [1 mark]

(*iii*) `A<5` [1 mark]

(*iv*) `A=6,not(A=6)` [1 mark]

(*b*) Consider the following clauses:

```
a(1).
a(a).
b(3).
b(a).
c(A,B) :- b(B),!,a(A).
c(X,_) :- a(X),b(X).
```

(*i*) List all the solutions to the query `c(A,B)` in order, giving any binding of variables that occurs. [2 marks]

(*ii*) List all the solutions to the query `c(X,1)` in order, giving any binding of variables that occurs. [2 marks]

(*c*) A binary tree has nodes whose values are non-empty lists. A tree node is represented using a term that takes the form `Left-SomeList+Right` or `nil` (note that parentheses can be used to enforce precedence). For example, the following term would be a valid tree:

```
nil-[root,1,2]+(nil-[child,4]+nil)
```

(*i*) Write a predicate `preorder(+Tree,-ValueList)` that unifies `ValueList` with a list containing all of the tree nodes' values from a pre-order tree walk (i.e. emit node value, then left subtree, then right subtree). The predicate should fail if any of the tree nodes' values are not of the correct form. For the above example tree, `preorder/2` would unify `ValueList` with `[[root,1,2],[child,4]]`. You may assume that `append/3` has been defined already. [6 marks]

(*ii*) Now write a predicate `preorderdl/2` that behaves exactly like your `preorder/2` predicate, but in its implementation makes use of difference lists, instead of querying `append/3`. [6 marks]