Lab Exercises - Meta-Logical Predicates

CMPT333N

Problem 1

Write a Prolog predicate `countBT(Tree, Count)` to count the number of nodes in a binary tree that have two children. Use an accumulator. Tree has the structure

```
bt(data, leftTree, rightTree)
```

The empty tree is represented by an uninstantiated variable. The predicate `var(X)` returns true if `X` is an uninstantiated variable and false otherwise.

Problem 2

Write a predicate, `nth(N, TheList, TheItem)`, which is true if TheItem is the `N'th` item in `TheList`. Counting begins at one. `nth(1,Alist,Elem)` is true for the first item in the list.

Problem 3

Write a predicate, `index(Matrix, [I1,I2,...,In], Elem)`, such that `Matrix[I1,I2,...,In] = Elem`, in a multidimentional matrix. Assume index value 1 is the first item in the corresponding dimension.

Problem 4

Write simple Prolog functions such as the following. Take into account lists which are too short.

- remove the N'th item from a list.
- insert as the N'th item.

Problem 5

Write a predicate `diagOf(theMatrix,theDiag)` where `theMatrix` is a square matrix and `theDiag` is the diagonal of the matrix. Use an accumulator.

Problem 6

Assume the prolog predicate `gt(A, B)` is true when `A` is greater than `B`. Use this predicate to define the predicate `addLeaf(Tree,X,NewTree)` which is true if `NewTree` is the `Tree` produced by adding the item `X` in a leaf node. `Tree` and `NewTree` are binary search trees. The empty tree is represented by the atom `nil`.