

Homework 8
SNU 4190.310, 2010 가을
이 광근
Due: 12/03(Fri), 24:00

Exercise 1 (30pts) “Algebraic Data Type”

Let-다형타입시스템을 갖춘 M 은 재귀적으로 정의되는 데이터(inductive data)를 다루는 함수들을 받아들이지 못한다.

예를 들어, 리스트(list)나 이진 트리(binary tree)는 다음과 같이 프로그램 하게 될 것이지만, 우리의 let-다형 타입 시스템은 이 프로그램들을 받아들이지 못한다:

```
(* program with list *)

let val nil = 0
    val isNil = fn list => list = 0
    val link = fn element => fn list => (element, list)
    val first = fn list => list.1
    val rest = fn list => list.2

in
  let
    val list = link 2 (link 1 (link 0 nil))
    rec sum = fn list => if isNil list then 0
                       else first list + sum (rest list)
  in
    sum list
  end
end

(* program with binary tree *)
```

```

let val emptyTree = 0
    val leaf = fn n => (n, (emptyTree, emptyTree))
    val node = fn n => fn ltree => fn rtree => (n, (ltree, rtree))
    val value = fn tree => tree.1
    val leftTree = fn tree => tree.2.1
    val rightTree = fn tree => tree.2.2
    val isEmpty = fn tree => tree = emptyTree
    val isLeaf = fn tree => isEmpty(leftTree tree) and isEmpty(rightTree tree)
in
  let
    rec traverse =
      fn tree => if isEmpty tree then 0
                 else if isLeaf tree then value tree
                 else value tree + traverse (leftTree tree)
                   + traverse (rightTree tree)
    val tree = node 8 (leaf 7) (node 10 (leaf 9) emptyTree)
  in
    traverse tree
  end
end

```

어떻게 언어를 확장하고, 타입 시스템은 어떻게 하면 될 런지, 논의하라.
리포트를 PDF 양식으로 만들어서 제출한다. □