

Subiectul 1 (6 puncte). Se dă următoarea secvență de declarații în Haskell:

```
type Id = String
data AExp = Lit Int | Var Id | AExp :+: AExp | AExp :<=: AExp
data Stmt = If AExp Stmt Stmt | Id := AExp | Stmt ;: Stmt | Skip
type Mem = [(Id, Int)]
eval :: Stmt -> Mem
```

Să se completeze definiția funcției `eval` pentru a putea calcula starea memoriei (`Mem`) în urma execuției instrucțiunii date ca argument. **Observatie:** Puteți defini oricâte funcții ajutatoare.

If are aceeași semantică ca în C (dacă valoarea expresiei e diferită de 0 se execută prima instrucțiune, dacă nu cea de-a doua). Operatorul `;` este operatorul de compunere secvențială a instrucțiunilor; semantica lui este: se execută prima instrucțiune mai întâi, apoi a doua instrucțiune se execută în starea de după execuția primei instrucțiuni. `Skip` este instrucțiunea vidă. Puteți presupune că toate variabilele au valoarea inițială 0.

Punctaj: 6 puncte din care 3 puncte rezolvarea, + 2 rezolvarea folosind monade + 1 rezolvarea folosind monada **predefinită** cea mai potrivită.

Barem:

0.3 Lit + <= ; skip

0.5 If & Var & :=

1 p. Alegerea monadei State

```
eval s = snd $ runState (evalStmt s) []
(otherwise 0.4)
```

0.4 evalAExp :: AExp -> State Mem Int

0.4 evalStmt :: AExp -> State Mem ()

0.4 Do

0.4 Return

0.4 lookup folosind get / definitia monadei

Subiectul 2 (3 puncte). Se dă următoarea secvență de declarații în Haskell:

```
class Facultate licentiat where
  licenta :: student -> licentiat student
  practica :: (ucenic -> licentiat pregatit) ->
              (licentiat ucenic -> licentiat pregatit)

-- 1 punct
calificare :: (Facultate licentiat) => licentiat (ucenic -> pregatit) ->
              (licentiat ucenic -> licentiat pregatit)
```

-- 2 puncte

```
dubluLicentiat :: (Facultate licentiat) => licentiat (licentiat bunBun) ->  
                                                    licentiat bunBun
```

Să se definească cele două funcții în modul cel mai natural posibil.

Punctaj: jumătate din punctaj pentru orice definiție care se compilează; punctaj întreg pentru o definiție naturală care ia în calcul toate argumentele de intrare.

1 punct din oficiu