Syntactic Transformation To Monadic Form

- ***Expressions:***

------ exp -------

$desugar_{<exp>}$ *:: Exp → Exp*

$desugar_{<exp>}$ exp = $desugar_{<lexp>}$ exp

------- lexp -------

$desugar_{<lexp>}$ *:: Exp → Exp*

-------exp: fexp -------

$desugar_{<lexp>}$ fexp = $desugar_{<fexp>}$ fexp

------- fexp -------

$desugar_{<fexp>}$ aexp = return $ $desugar_{<aexp>}$ aexp

$desugar_{<fexp>}$ (fexp (exp))= $desugar_{<aexp>}$ (exp) >>= \x → ($desugar_{<fexp>}$ fexp) x

$desugar_{<fexp>}$ (fexp (exp1,…, exp_k))= $desugar_{<aexp>}$ (exp1,…, exp_k) >>= \tuple →
($desugar_{<fexp>}$ fexp) tuple

$desugar_{<fexp>}$ (fexp [exp1,…, exp_k])= $desugar_{<aexp>}$ [exp1,…, exp_k] >>= \list →
($desugar_{<fexp>}$ fexp) list

------- aexp -------

$desugar_{<aexp>}$ literal = literal

$desugar_{<aexp>}$ qvar = qvar

$desugar_{<aexp>}$ gcon = gcon

$desugar_{<aexp>}$ (exp) = ($desugar_{<lexp>}$ exp)

$desugar_{<aexp>}$ (exp1,…, exp_k) = ($desugar_{<lexp>}$ exp_1, … , $desugar_{<lexp>}$ exp_k)

$desugar_{<aexp>}$ [exp1,…, exp_k] = [$desugar_{<lexp>}$ exp_1, … , $desugar_{<lexp>}$ exp_k]

-------lexp: let decls in exp -------

$desugar_{<lexp>}$ (let decls in exp) = $desugar_{<dclrs>}$ decls (return $desugar_{<lexp>}$ exp)

- ***Declarations***

```
-------  dclrs -------
desugar<dclrs> (dclr1; … ; dclrn) =  desugar<exp> exp1,… ,desugar<exp> expn) >>= \x1..xk →
                        | (;)  = \_ →
  desugar<dclrs> (dclr1; … ; dclrn) = desugar<dclr> declr1 $ …  $ desugar<dclr> declrn
                          |  (;) = \_ ->


  -------  dclr -------

    desugar<dclr> :: Dclr → Stmt

    desugar<dclr> (funlhs | pat) rhs =  (funlhs | pat) desugar<rhs> rhs


  -------  rhs  -------

    desugar<rhs> (= exp) =  desugar<exp> exp >>= \x ->
```