Syntactic Transformation To Monadic Form

- ***Expressions:***

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

$desugar_{<exp>} :: Exp \rightarrow Exp$

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

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

$desugar_{<lexp>} :: Exp \rightarrow Exp$

-------lexp: fexp -------

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

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

$desugar_{<fexp>}$ aexp = $desugar_{<aexp>}$ aexp >>= \h -> return h

$desugar_{<fexp>}$ (fexp literal) = $desugar_{<lexp>}$ fexp literal

$desugar_{<fexp>}$ (fexp qvar) = (fexp qvar) >>= \g → ($desugar_{<lexp>}$ fexp) g

$desugar_{<fexp>}$ (fexp gcon) = $desugar_{<lexp>}$ fexp gcon

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

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

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

$desugar_{<dclrs>}$ ($dclr_1$; … ; $dclr_n$) = $desugar_{<exp>}$ exp_1,… ,$desugar_{<exp>}$ exp_n) >>= \$x_{1..}x_k$ →

| (;) = _ →

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 ->