6.17 forslag

Grammar Rule	Semantic Rule
exp_1 → ***let*** *dec-list* ***in*** exp_2	*dec-list.intab = exp_1.symtab* *dec-list.locintab = exp_1.symtab* *dec-list.outtab =* *exp_1.symtab + dec-list.locouttab* *exp_2.symtab = dec-list.outtab*
$dec\text{-}list_1$ → $dec\text{-}list_2$ **,** *decl*	*$decl_2$.intab = $dec\text{-}list_1$.intab* *decl.intab = $dec\text{-}list_1$.intab* *$dec\text{-}list_2$.locintab = $dec\text{-}list_1$.locintab* *decl.locintab = $decl\text{-}list_2$.locouttab* *$dec\text{-}list_1$.locouttab = decl.locouttab*
dec-list → *decl*	*decl.intab = decl-list.intab* *decl.locintab = dec-list.locintab* *dec-list.locouttab = decl.locouttab*
decl → ***id*** = *exp*	*decl.locouttab = ...* *insert(decl.locintab, ...)*

6.18 forslag

Grammar Rule	Semantic Rule
exp$_1$ → *exp*$_2$ + *exp*$_3$	*exp*$_1$*.val =* **if** *(exp*$_2$*.val =* **error***)* **or** *(exp*$_3$*.val =* **error***)* **then error** **else** *exp*$_2$*.val + exp*$_3$*.val*
exp$_1$ → *(exp*$_2$*)*	*exp*$_1$*.val = exp*$_2$*.val*
exp → ***id***	*exp.val = lookupVal(exp.symtab, id.name)*
exp → ***num***	*exp.val =* ***num****.val*

exp$_1$ → ***let*** *dec-list* ***in*** *exp*$_2$	*exp*$_1$*.val =*
	if *(decl-list.outtab = errtab)*
	then error
	else *exp*$_2$*.val*

decl → ***id*** = *exp*	
	```
	decl.outtab =
	if  (decl.intab = errtab)
	then errtab
	else
	   if
	      (lookupLevel(
	          decl.intab,id.name) =
	      decl.nestlevel)
	   then errtab
	   else
	    insert(decl.intab,id.name,
	        decl.nestlevel, exp.val)
	```