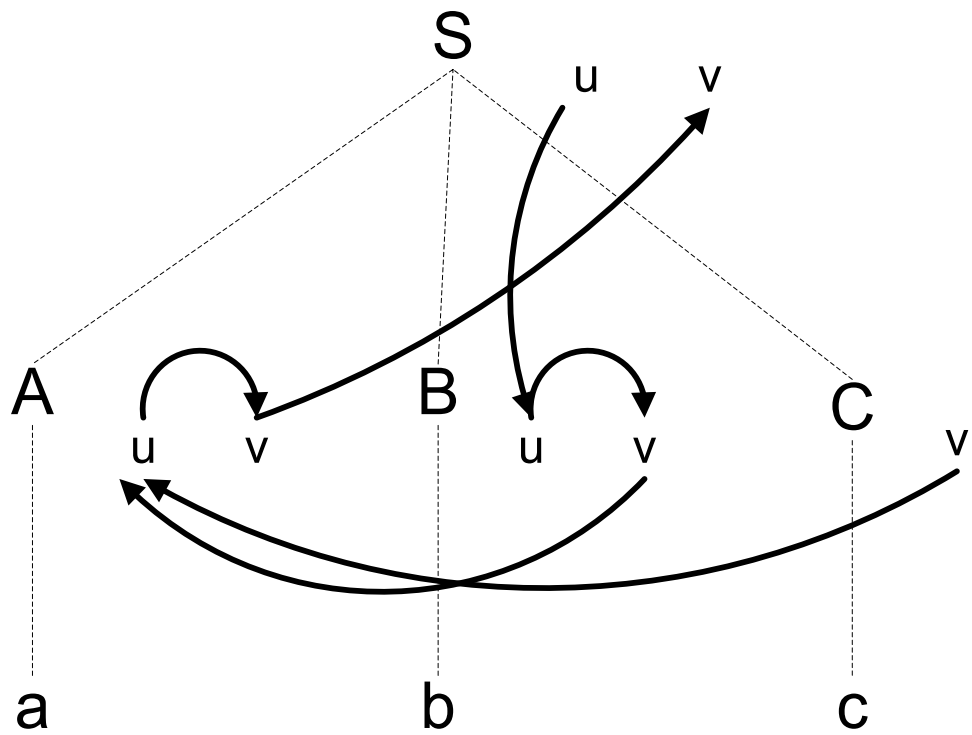
Oppgave 6.5

Grammar Rule	Semantic Rule
$exp_1 \rightarrow exp_2$ **+** *term*	exp_1*.postfix =* exp_2*.postfix || exp_2.postfix || +*
$exp_1 \rightarrow exp_2$ **-** *term*	exp_1*.postfix =* exp_2*.postfix || exp_2.postfix || -*
exp → term	*exp.postfix = term.postfix*
$term_1 \rightarrow term_2$ ***** *factor*	$term_1$*.postfix =* $term_2$*. postfix ||* *factor.postfix ||**
term → factor	*term.postfix = factor.postfix*
factor → **(** *exp* **)**	*factor.postfix = exp.postfix*
factor → **number**	*factor.postfix =* **number***.strval*

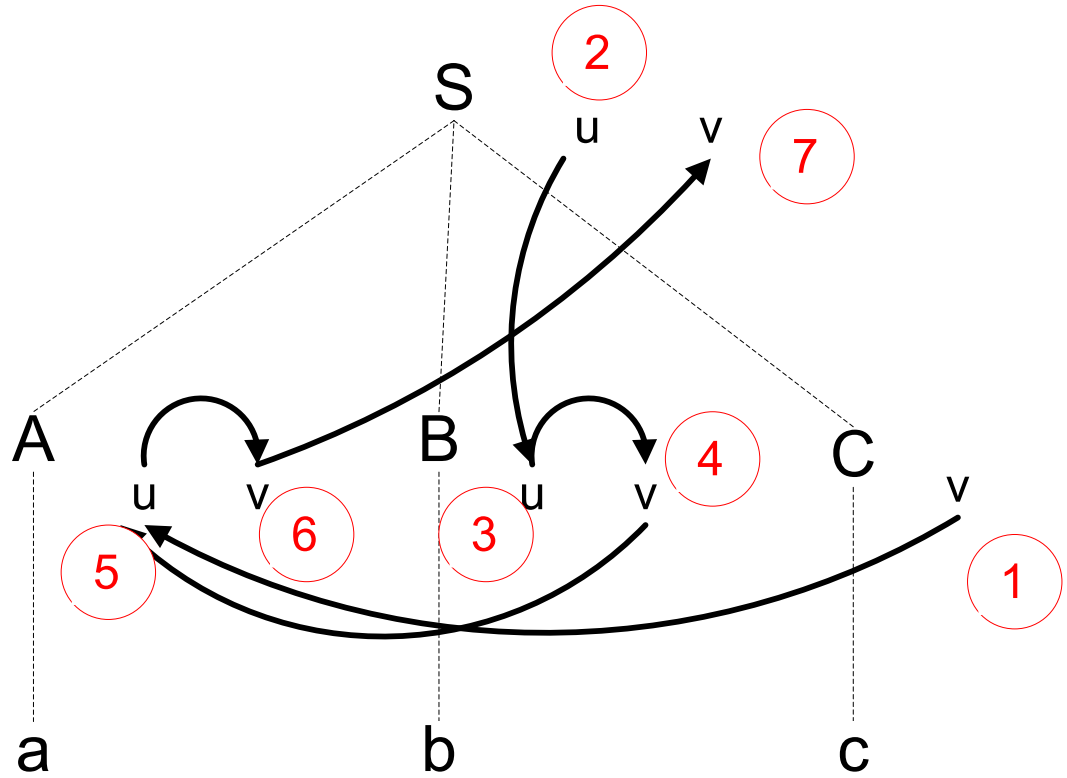
Oppgave 6.7

Grammar Rule	Semantic Rule
decl → *var-list* : *type*	*var-list.dtype* = *type.dtype*
$var\text{-}list_1$ → $var\text{-}list_2$, **id**	$var\text{-}list_2.dtype = var\text{-}list_1.dtype$ **id**.*dtype* = $var\text{-}list_1.dtype$
var-list → **id**	**id**.*dtype* = *var-list.dtype*
type → **int**	*type.dtype* = *integer*
type → **real**	*type.dtype* = *real*

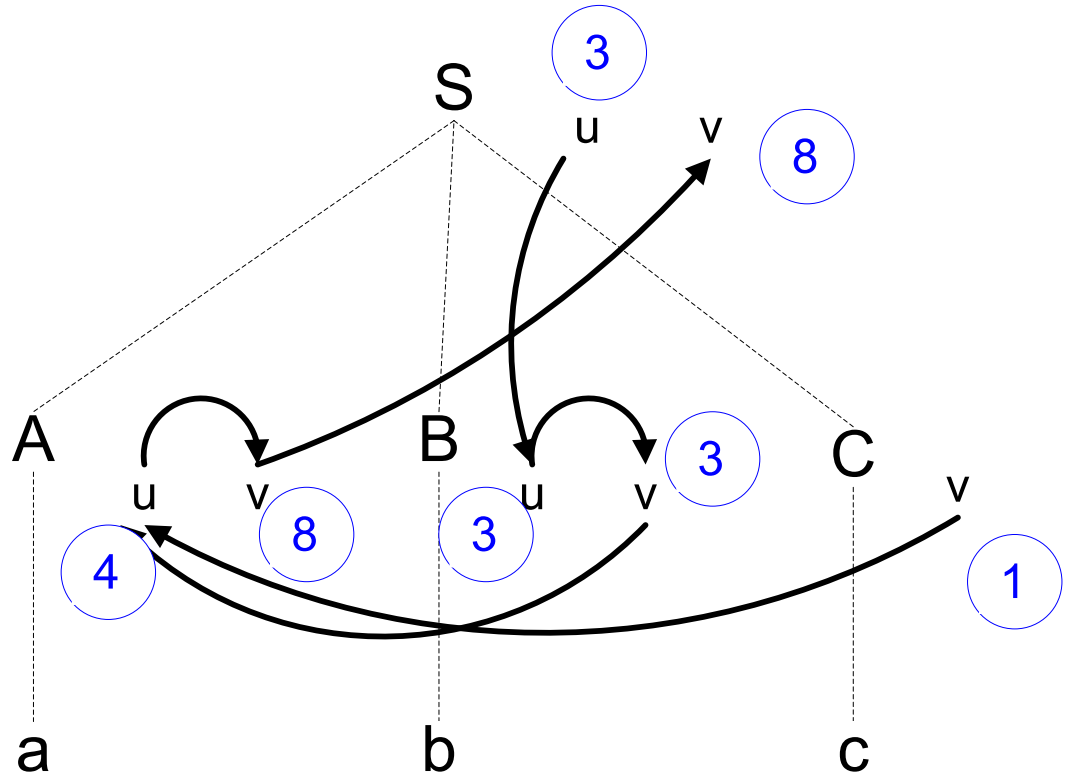
Oppgave 6.13 a 1)



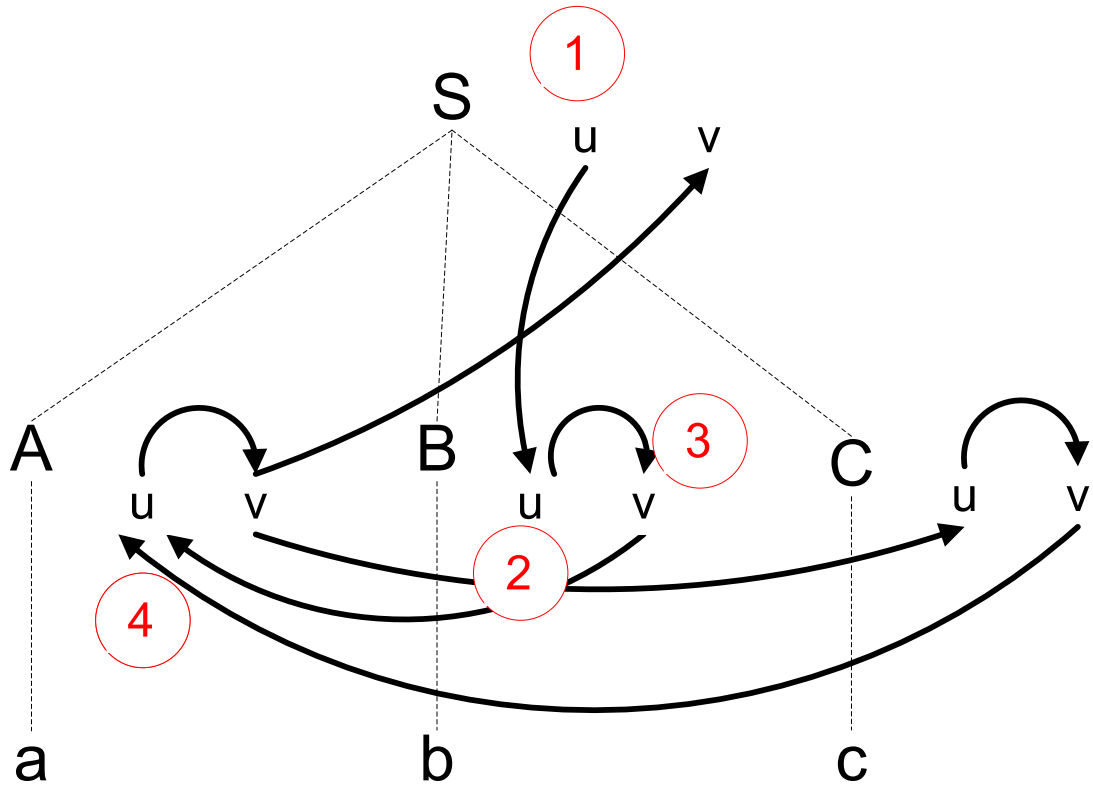
Oppgave 6.13 a 2)



Oppgave 6.13 b)



Oppgave 6.13 c



Grammar Rule	Semantic Rule
class → ***class name {*** *decls* ***}***	*decls.enclosingClassName* = ***name***.name
$decls_1$ → $decls_2$; *decl*	$decls_2$*.enclosingClassName* = $decls_1$*.enclosingClassName* *decl.enclosingClassName* = $decls_1$*.enclosingClassName*
decls → *decl*	*decl.enclosingClassName* = *decls.enclosingClassName*
decl → *variable-decl*	
decl → *method-decl*	*method-decl.enclosingClassName* = *decl.enclosingClassName*
method-decl → *type* ***name (*** *params* ***)*** *body*	***if*** *(****name***.name = *method-decl.enclosingClassName)* ***then if*** *(****not****(type.*type = void*))****then*** *error(“constructor not of type void”)* eller ***if*** *(****name***.name = *method-decl.enclosingClassName)* ***and*** *(****not****(type.*type = void*))****then*** *error(“constructor not of type void”)*
type → **int**	*type.*type = int
type → **bool**	*type.*type = bool
type → **void**	*type.*type = void