

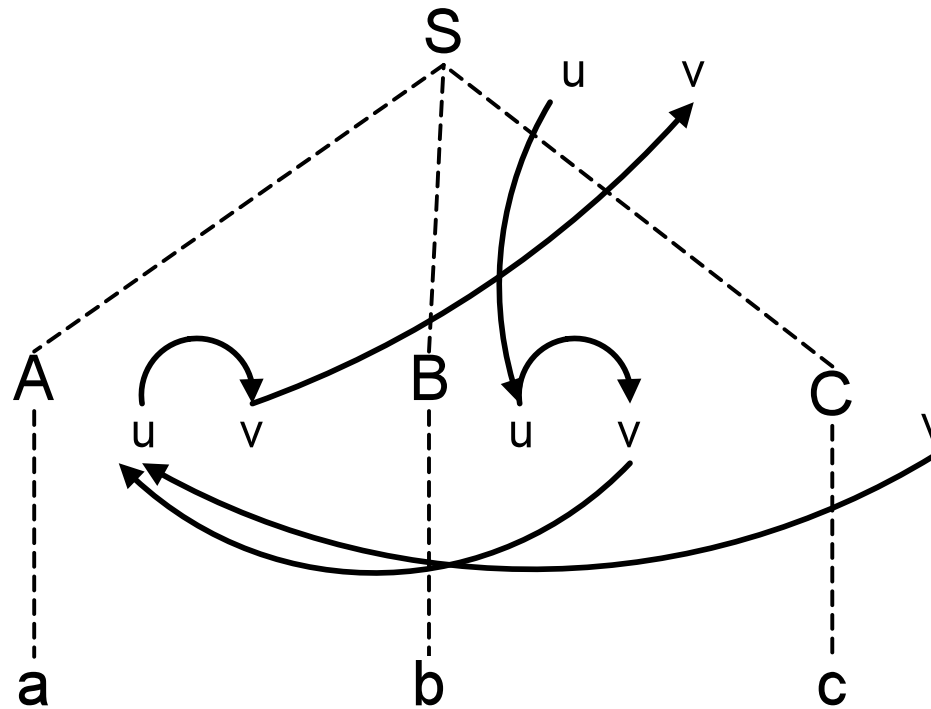
Oppgave 6.5

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

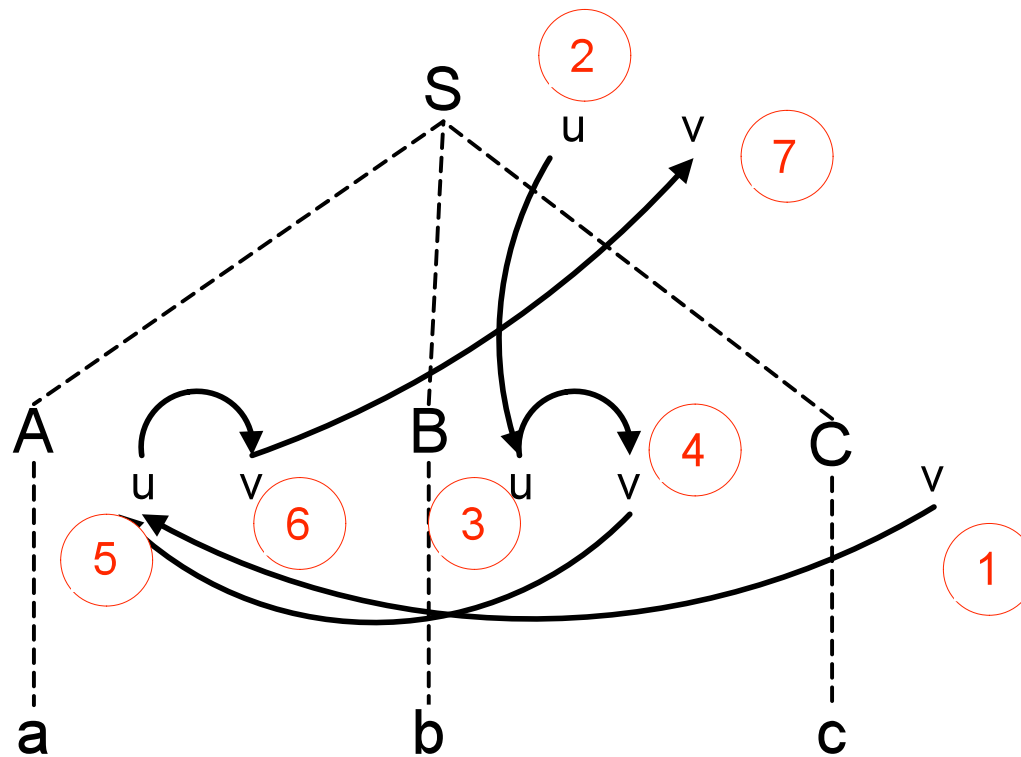
Oppgave 6.7

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

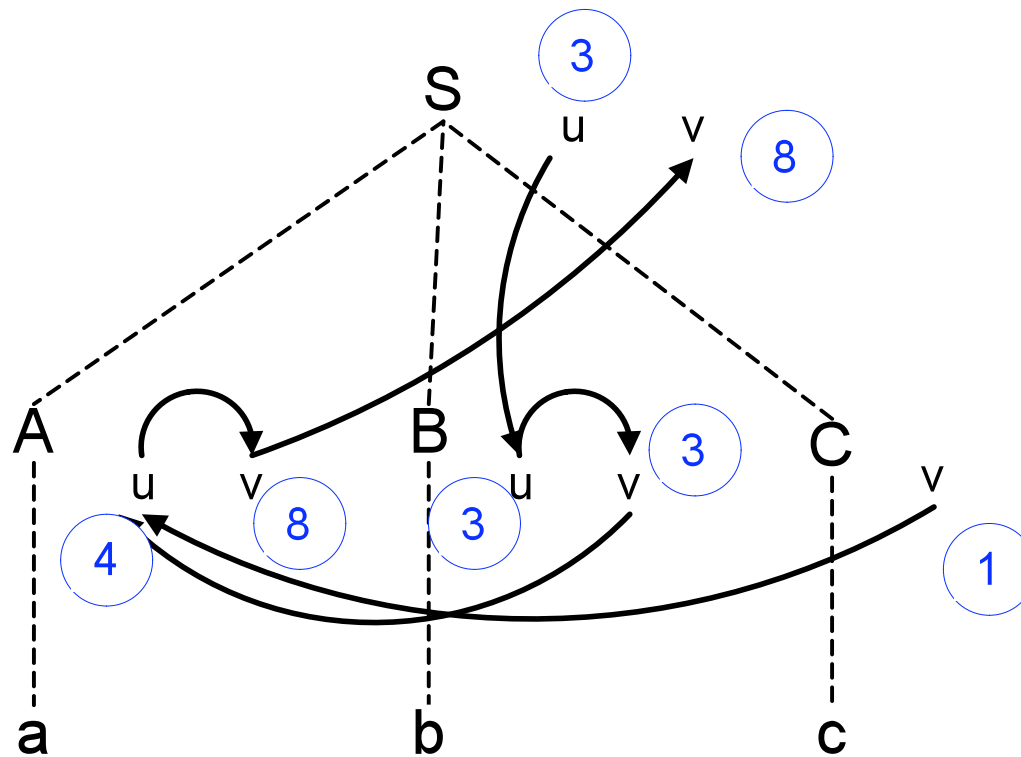
Oppgave 6.13 a 1)



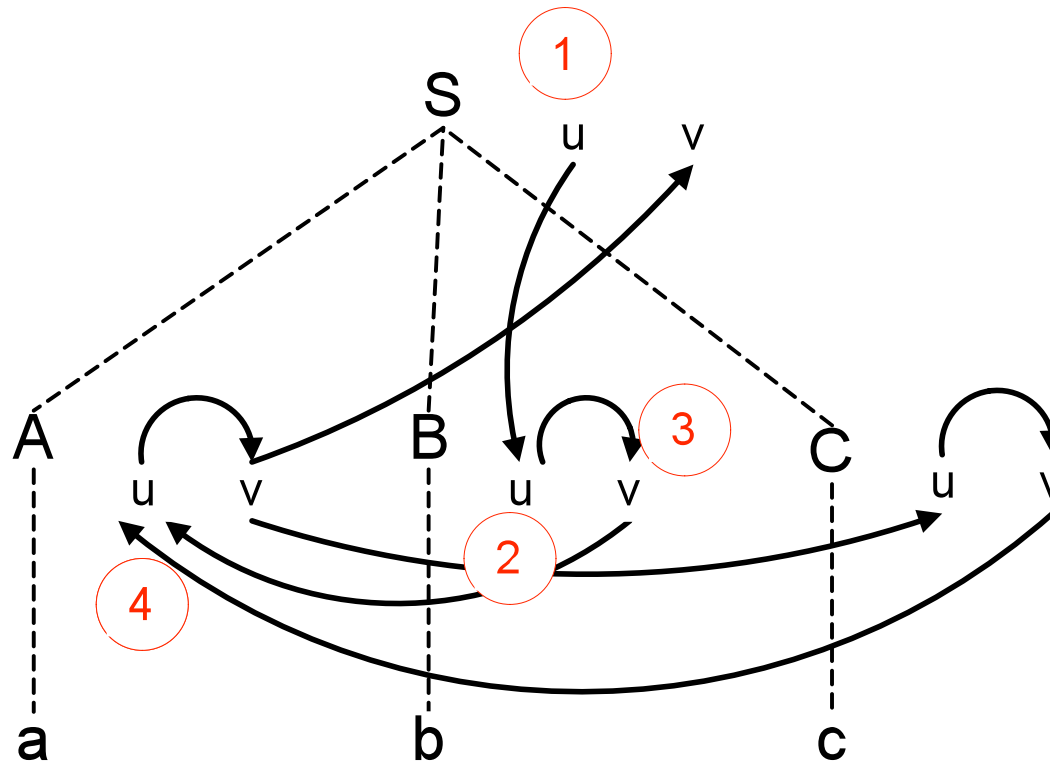
Oppgave 6.13 a 2)



Oppgave 6.13 b)



Oppgave 6.13 c



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