

6.5 Rewrite the attribute grammar of Table 6.2 to compute a *postfix* string attribute instead of *val*, containing the postfix form for the simple integer expression. For example, the *postfix* attribute for **(34-3)*42** is “34 3 - 42 + *.” You may assume a concatenation operator `||` and the existence of a ***number.strval*** attribute.

6.7 Consider the following grammar for simple Pascal-style declarations:

$$\begin{aligned} decl &\rightarrow var\text{-list} : type \\ var\text{-list} &\rightarrow var\text{-list} , id \mid id \\ type &\rightarrow \mathbf{integer} \mid \mathbf{real} \end{aligned}$$

Write an attribute grammar for the type of a variable.

6.13 Consider the following attribute grammar:

Grammar Rule	Semantic Rules
$S \rightarrow A B C$	$B.u = S.u$ $A.u = B.v + C.v$ $S.v = A.v$
$A \rightarrow a$	$A.v = 2 * A.u$
$B \rightarrow b$	$B.v = B.u$
$C \rightarrow c$	$C.v = 1$

- Draw the parse tree for the string abc (the only string in the language), and draw the dependency graph for the associated attributes. Describe a correct order for the evaluation of the attributes.
- Suppose that $S.u$ is assigned the value 3 before attribute evaluation begins. What is the value of $S.v$ when evaluation has finished?

c. Suppose the attribute equations are modified as follows:

Grammar Rule	Semantic Rules
$S \rightarrow A B C$	$B.u = S.u$ $C.u = A.v$ $A.u = B.v + C.v$ $S.v = A.v$
$A \rightarrow a$	$A.v = 2 * A.u$
$B \rightarrow b$	$B.v = B.u$
$C \rightarrow c$	$C.v = C.u - 2$

What value does $S.v$ have after attribute evaluation, if $S.u = 3$ before evaluation begins?

```
class → class name superclass { decls }  
decls → decls ; decl | decl  
decl → variable-decl  
decl → method-decl  
method-decl → type name ( params ) body  
type → int | bool | void  
superclass → name
```

Ord i kursiv er non-terminaler, ord og tegn i **fet skrift** er terminal-symboler, mens **name** representerer et navn som scanneren leverer. Det kan antas at **name** har attributtet 'name'.

Metoder med samme navn som klassen er 'konstruktører', og det gjelder følgende regel: Konstruktører må være spesifisert med typen **void**.

Lag semantiske regler for denne regel i følgende fragment av en attributtgrammatikk.

Grammar Rule

Semantic Rule

class → *class name { decls }*

decls → *decls ; decl*

decls → *decl*

decl → *variable-decl*

Skal ikke fylles ut

decl → *method-decl*

method-decl →
type name (params) body

type → *int*

type → *bool*

type → *void*
