German University in Cairo
Department of Computer Science
Assoc. Prof. Haythem O. Ismail

CSEN 1003 Compiler, Spring Term 2019
Practice Assignment 9

Discussion: 10.04.18 - 15.04.18

Exercise 9-1

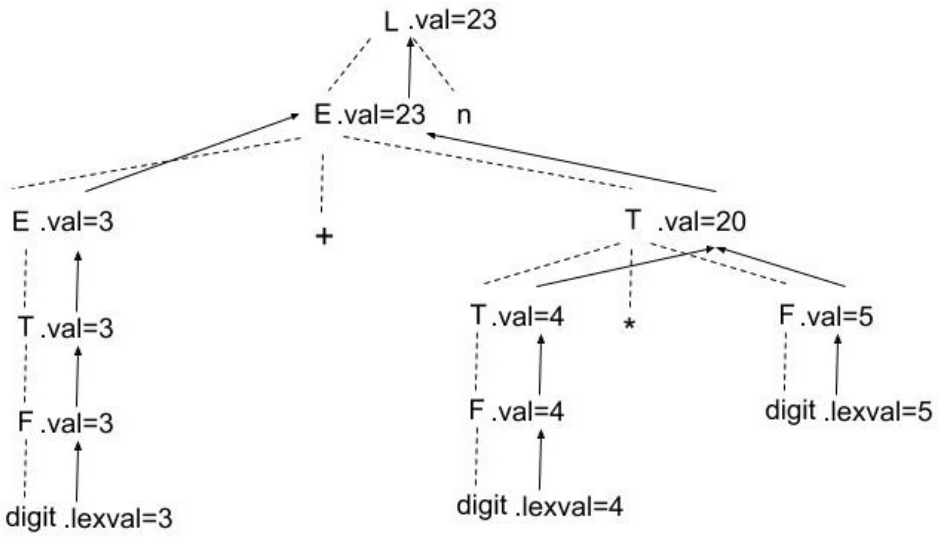
Consider the following SDD:

Production				Semantic Rule
1)	L	$\rightarrow$	E `n`	$L.val = E.val$
2)	E	$\rightarrow$	E_1 `+` T	$E.val = E_1.val + T.val$
3)	E	$\rightarrow$	T	$E.val = T.val$
4)	T	$\rightarrow$	T_1 `*` F	$T.val = T_1.val * F.val$
5)	T	$\rightarrow$	F	$T.val = F.val$
6)	F	$\rightarrow$	`(` E `)`	$F.val = E.val$
7)	F	$\rightarrow$	**digit**	$F.val =$ **digit**.lexval

Give the dependency graphs of the annotated parse trees for the following expressions

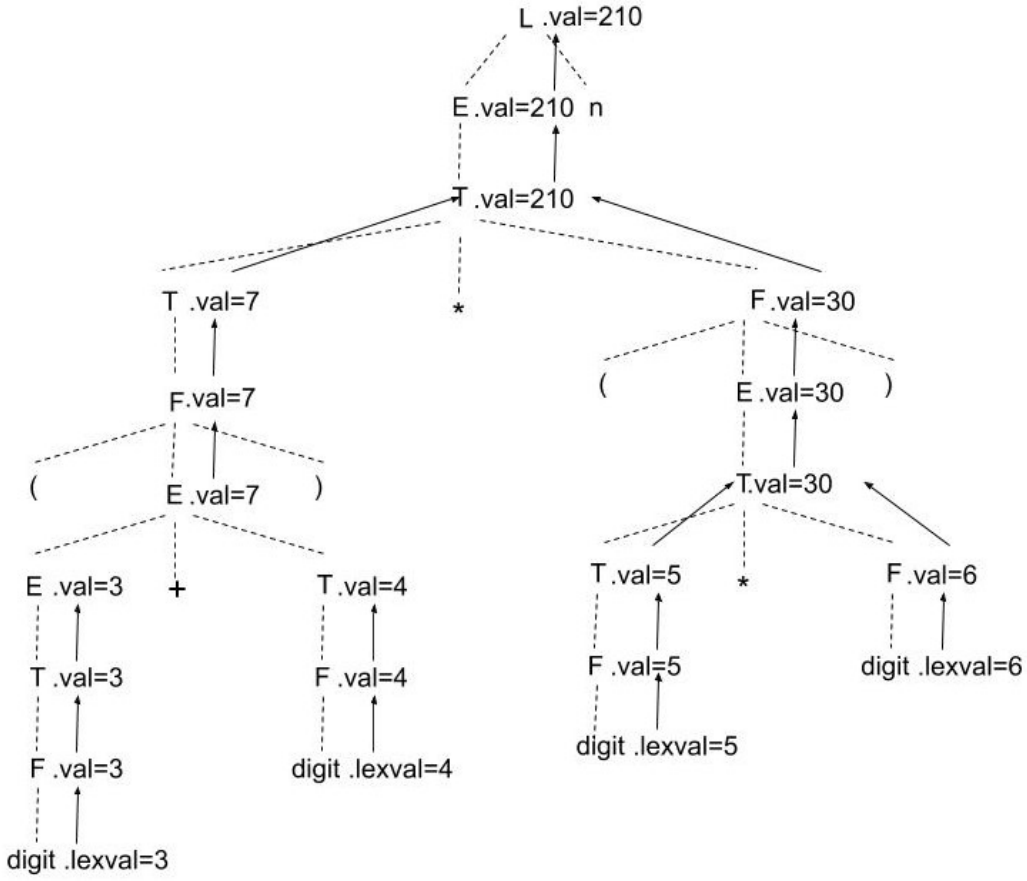
a) `3 + 4 * 5 n`

Solution:



b) `(3 + 4) * ( 5 * 6) n`

Solution:



Exercise 9-2

Extend the following SDD to handle expressions as in SDD given in Exercise 9-1.

Production				Semantic Rule
1)	T	$\rightarrow$	$F\ T'$	$T'.inh = F.val$
				$T.val = T'.syn$
2)	T'	$\rightarrow$	$*\ F\ T'_1$	$T'_1.inh = T'.inh\ *\ F.val$
				$T'.syn = T'_1.syn$
3)	T'	$\rightarrow$	ε	$T'.syn = T'.inh$
4)	F	$\rightarrow$	**digit**	$F.val =$ **digit**.lexval

Solution:

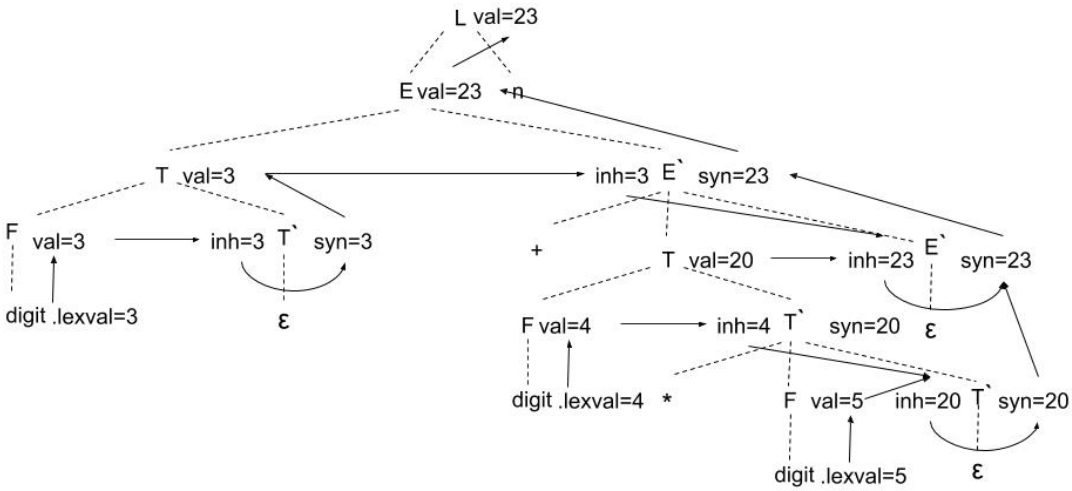
Note: *.syn* and *.val* are different names for the same synthesized attributes of the grammer variables.

Production				Semantic Rule
1)	L	$\rightarrow$	E `n`	$L.val = E.syn$
2)	E	$\rightarrow$	$T\ E'$	$E.syn = E'.syn$
				$E'.inh = T.val$
3)	E'	$\rightarrow$	`+` $T\ E'_1$	$E'_1.inh = E.inh + T.val$
				$E.syn = E'_1.syn$
4)	E'	$\rightarrow$	ε	$E'.syn = E'.inh$
5)	T	$\rightarrow$	$F\ T'$	$T'.inh = F.val$
				$T.val = T'.syn$
6)	T'	$\rightarrow$	`*` $F\ T'_1$	$T'_1.inh = T'.inh * F.val$
				$T'.syn = T'_1.syn$
7)	T'	$\rightarrow$	ε	$T'.syn = T'.inh$
8)	F	$\rightarrow$	`(` E `)`	$F.val = E.val$
9)	F	$\rightarrow$	**digit**	$F.val =$ **digit**.lexval

Give the dependency graphs of the annotated parse trees for the following expressions

a) `3 + 4 * 5 n`

Solution:



b) `(3 + 4) * ( 5 * 6) n`

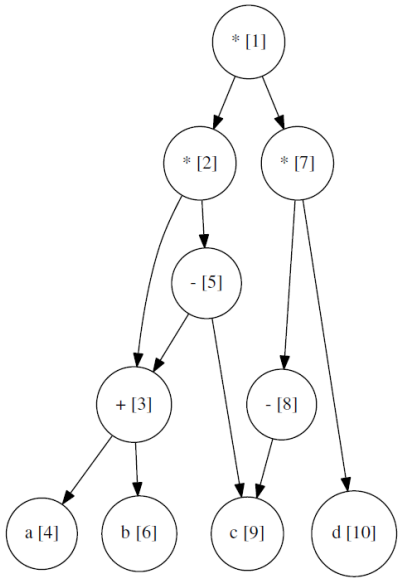
Solution:

Do it yourself.

Exercise 9-3

Topological sort of a dependency graph

Give a topological sort for the following dependency graph:



Solution:

Post-order traversal: 4, 6, 3, 9, 5, 2, 8, 10, 7, 1

Topological sort: 1, 7, 10, 8, 2, 5, 3, 6, 4, 9

Exercise 9-4

Suppose that we have a production $A \rightarrow BCD$. Each of the four nonterminals A,B,C,and D have two attributes: s is a synthesized attribute, and i is an inherited attribute. For each of the sets of rules below, tell whether (i) the rules are consistent with an S-attributed definition (ii) the rules are consistent with an L-attributed definition, and (iii) whether the rules are consistent with any evaluation order at all?

a) $A.s = B.i + C.s$.

b) $A.s = B.i + C.s$ and $D.i = A.i + B.s$.

c) $A.s = B.s + D.s$.

d) $A.s = D.i$, $B.i = A.s + C.s$, $C.i = B.s$, and $D.i = B.i + C.i$.

Solution:

a) Consistent with an L-attributed definition, since the value of the synthesized attribute $A.s$ of the node is computed from that of the inheritred attribute $(B.i)$ and synthesized attribute $(C.s)$ of its children nodes.

b) Consistent with an L-attributed definition, since the value of the inherited attribute $D.i$ is computed from the attributes of a parent node $(A.i)$ and a sibling node that appears to its left $(B.s)$.

c) Consistent with an S-attributed definition, since the value of the synthesized attribute ($A.s$) of the node is computed from that of the attributes of its children nodes $C.s$ and $D.s$.

d) Neither, since the inherited attribute ($B.i$) is computed from the attribute $C.s$ of a sibling node that appears to its right in the production, which is inconsistent with the L-attributed definition. Moreover, the value of the inherited attribute of B ($B.i$) is dependents on a synthesized attribute of the parent which is also inconsistent with the L-attribute definition.

Exercise 9-5

The following grammar generates binary numbers with a "decimal" point:

$$\begin{aligned} S &\rightarrow L.L \mid L \\ L &\rightarrow LB \mid B \\ B &\rightarrow 0 \mid 1 \end{aligned}$$

Design an L-attributed SDD to compute $S.val$, the decimal-number value of an input string. For example, the translation of string 101.101 should be the decimal number 5.625.

Solution:

S	$\longrightarrow$	$L_1.L_2$	$L_1.f = 1; L_1.p = 1; L_2.f = 0.5; L_2.p = 0.5$
			$S.val = L_1.val \ + \ L_2.val$
S	$\longrightarrow$	L	$L.f = 1; L.p = 1; S.val = L.val$
L	$\longrightarrow$	L_1B	$L_1.f = L.f \times L.p; L_1.p = L.p$
			$L.min = L_1.min; L.val = 2 \times L_1.val + L.min \times B.val$
L	$\longrightarrow$	B	$L.min = L.f; L.val = L.min \times B.val$
B	$\longrightarrow$	`0`	$B.val = 0$
B	$\longrightarrow$	`1`	$B.val = 1$

Exercise 9-6

Convert your SDD from Exercise 9-2 to an SDT.

Solution:

Production			
1)	L	$\rightarrow$	E `n` $\{L.val = E.syn\}$
2)	E	$\rightarrow$	T $\{E'.inh = T.val\}$ E' $\{E.syn = E'.syn\}$
3)	E'	$\rightarrow$	`+` T $\{E'_1.inh = E.inh + T.val\}$ E'_1 $\{E.syn = E'_1.syn\}$
4)	E'	$\rightarrow$	$\varepsilon$$\{$E'.syn=E'.inh$\}$
5)	T	$\rightarrow$	F $\{T'.inh = F.val\}$ T' $\{T.val = T'.syn\}$
6)	T'	$\rightarrow$	`*` F $\{T'_1.inh = T.inh * F.val\}$ T'_1 $\{T'.syn = T'_1.syn\}$
7)	T'	$\rightarrow$	$\varepsilon$$\{$T'.syn=T'.inh$\}$
8)	F	$\rightarrow$	`(` E `)` $\{F.val = E.val\}$
9)	F	$\rightarrow$	**digit** $\{F.val =$ **digit**.lexval$\}$