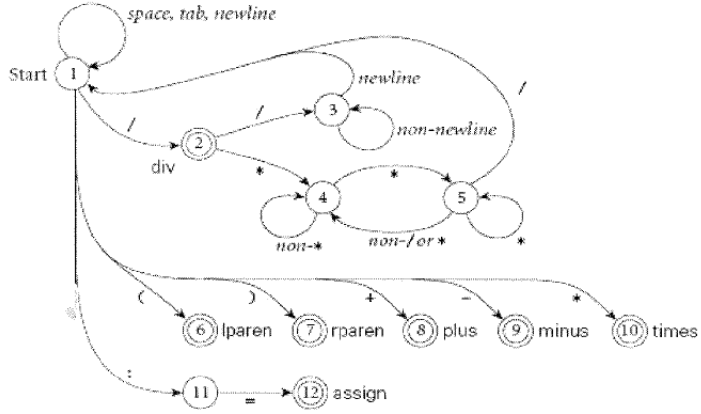


**Concepts of Programming Languages, CSCI 305, Fall 2021  
Homework #4, complete by Oct. 8**

1. Create a scanning table for the following minimal dfa.



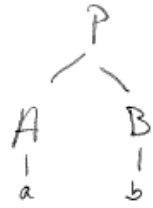
2. Hand execute the table-driven lexical analyzer pseudo code (Figure 2.11, page 66) on the very simple program.

```
sum :=29.57  
write sum+4
```

For each call to the lexical analyzer, list all accesses into the `scan_tab`, in order as the code is executed. End the call by showing what is returned to the parser.

Each “move” action requires two `scan_tab` accesses, one to get the action and a second one to get the `new_state`. Only one access needs to be listed.

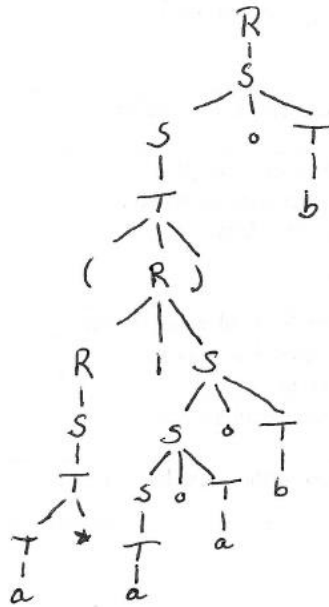
3. Consider the parse tree below.



Give a rightmost derivation of the tree:

Give a leftmost derivation of the tree:

4. Consider the parse tree below.



Give a rightmost derivation of the tree:

5. Create a grammar for the context-free language

$$L = \{x_1w^Rx_2 \#w \mid x_1, x_2, w \in \{a, b\}^*\}$$

where  $w^R$  is the reverse of  $w$

Strings in L	Strings not in L
#	$\epsilon$
aaabaa#b	a
bbaaaa#abb	b
bbabbbb#bbab	bb#aa
	ba#ba
	aaaaba#bab