

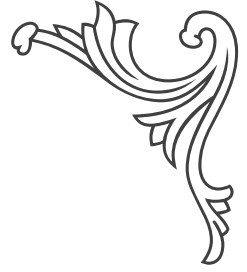


---

# Quiz 2

---

*EECS665 - Compiler Construction  
2020, Fall*



## QUIZ COPY

This quiz originally appeared on Blackboard and has been copied here for the purpose of review.

Total Questions: 5

Total Pages: 6

Total Points: 50

---



Student ID: \_\_\_\_\_

### QUESTION 1 (2 POINTS)

Describe one difference between a parse tree and an abstract syntax tree

QUESTION 2 (12 POINTS)

Student ID: \_\_\_\_\_

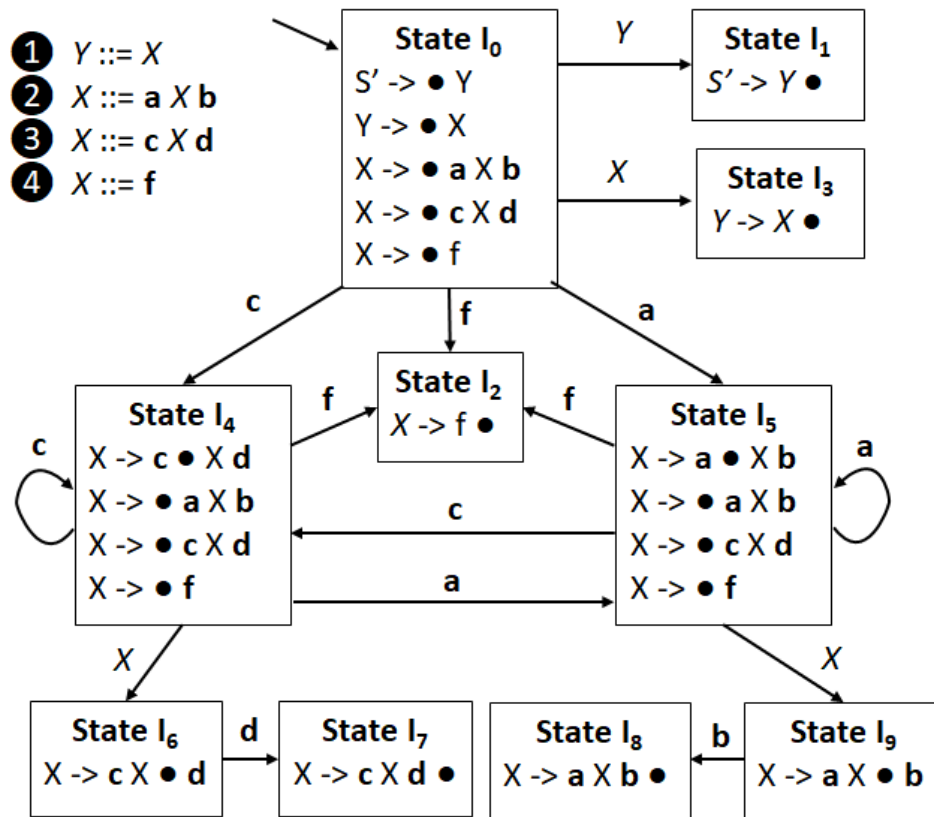
PART I: GIVE A GRAMMAR  $G$  THAT IS NOT  $LL(1)$

PART II: SHOW HOW  $G$  CAN BE TRANSFORMED INTO A GRAMMAR  $H$  THAT IS  $LL(1)$   
AND RECOGNIZES THE SAME LANGUAGE AS  $G$

# QUESTION 3 (12 POINTS)

Student ID: \_\_\_\_\_

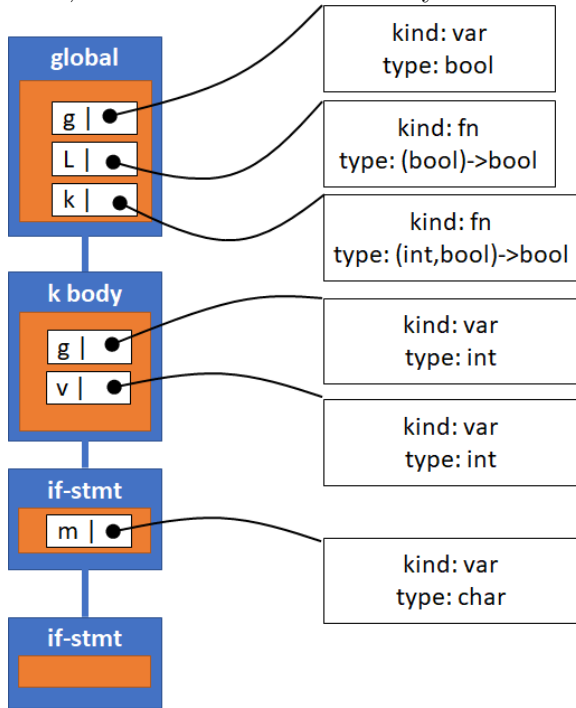
Create the LR Parsing table for this parser automaton



# QUESTION 4 (12 POINTS)

Student ID: \_\_\_\_\_

Give a possible source code program in our language and a line of that program that would produce this Symbol Table. Indicate any name analysis errors. If no such program is possible, indicate the reason the symbol table cannot be constructed.



## QUESTION 5 (12 POINTS)

Student ID: \_\_\_\_\_

Consider the following grammar

$Y ::= X$

$X ::= a X b$

$X ::= c X d$

$X ::= f$

Consider the SDT goal to translate a string in the above language to true if the input has more a tokens than d tokens and false otherwise.

### PART I

Write out the LL(1) semantic stack actions to meet the above goal. Recall that the semantic stack actions should push and pop the semantic stack and leave the result on the stack

### PART II

Show the LL(1) selector table translator with actions embedded. If you can't get the action numbers embedded, show the selector table without action numbers for partial credit