



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

QUESTION 1 (2 POINTS)

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

Parse trees have leaves that explicitly represent every token of the input stream.

ASTs may discard some tokens that only exist for grouping

Char stream
1 + 2

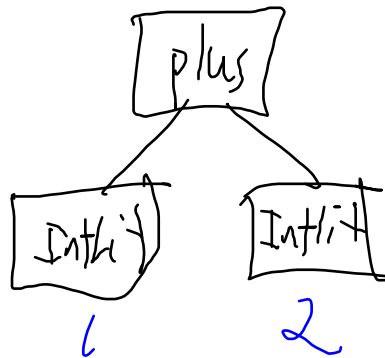
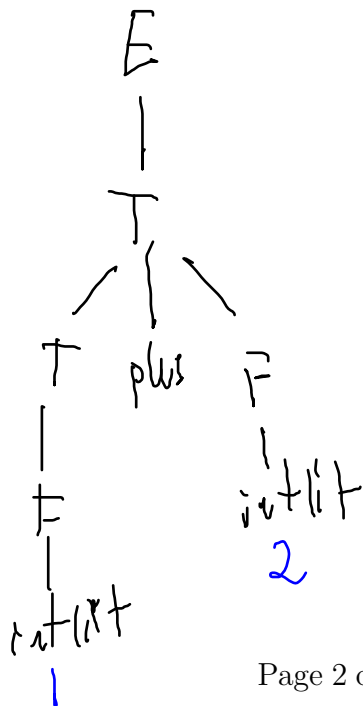
tokens
intlit plus intlit
1 2

Grammar

$$E ::= E - T$$

$$T ::= T \text{ plus } F$$

$$F ::= \text{intlit}$$



QUESTION 2 (12 POINTS)

Student ID: _____

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

G is

$$\begin{aligned} X &::= a b \\ &| a c \end{aligned}$$

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

H is

$$\begin{aligned} X &::= a X' \\ X' &::= b \\ &| c \end{aligned}$$

QUESTION 3 (12 POINTS)

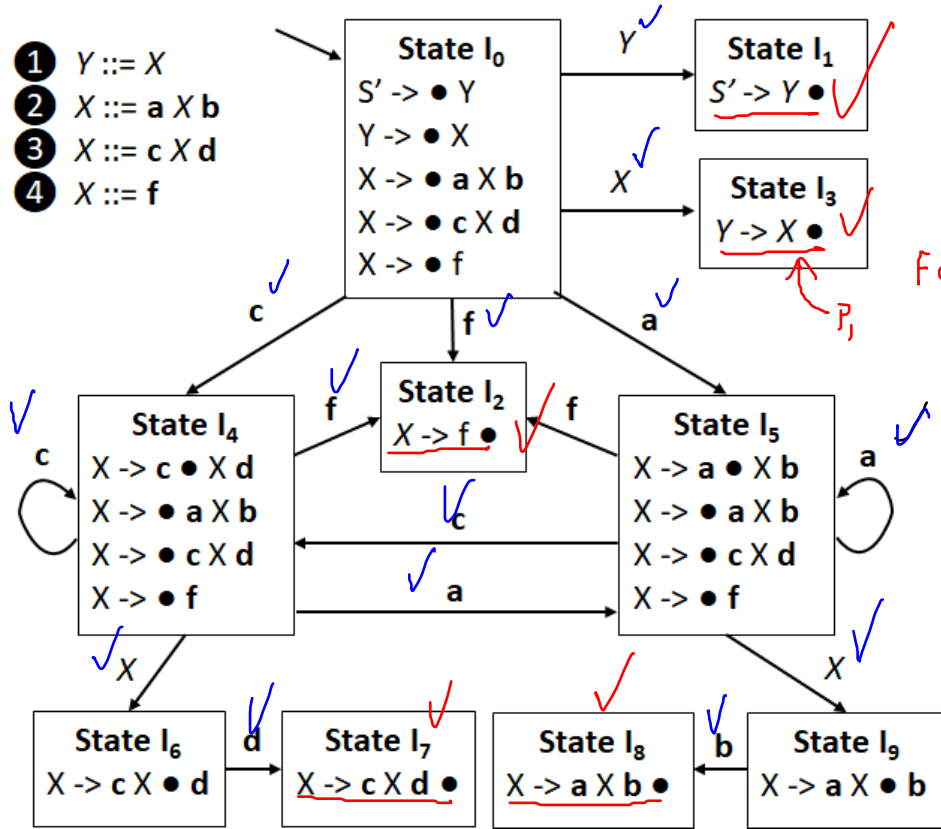
Student ID: _____

Create the LR Parsing table for this parser automaton

FOLLOW(Y)
{eof}

FOLLOW(X)
{eof, b, d}

- 1 $Y ::= X$
- 2 $X ::= a X b$
- 3 $X ::= c X d$
- 4 $X ::= f$



For $[I_3]$ [i in follow LHS]

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>f</i>	<i>eof</i>	<i>Y</i>	<i>X</i>
I_0	$\rightarrow I_5$		$\rightarrow I_4$		$\rightarrow I_2$		$\rightarrow I_1$	$\rightarrow I_3$
I_1						$\rightarrow I_1$		
I_2						$\rightarrow I_2$		
I_3								
I_4	$\rightarrow I_5$		$\rightarrow I_4$		$\rightarrow I_2$			
I_5	$\rightarrow I_5$		$\rightarrow I_4$					
I_6				$\rightarrow I_7$				
I_7								
I_8								
I_9								

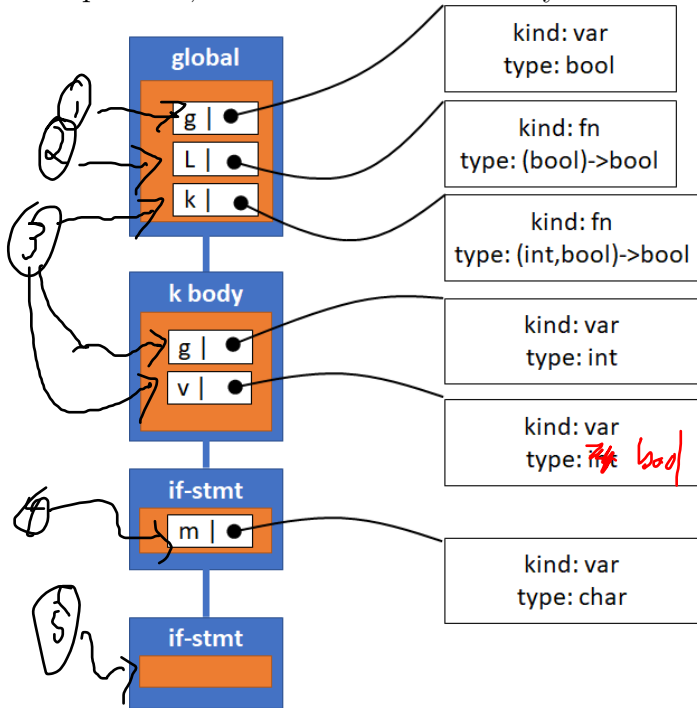
Handwritten notes in red:

- $R(P_1)$ in I_2 for *f*
- $R(P_2)$ in I_4 for *a*
- $R(P_3)$ in I_4 for *c*
- $R(P_4)$ in I_5 for *a*
- $R(P_5)$ in I_5 for *c*
- $R(P_6)$ in I_5 for *f*
- $R(P_7)$ in I_6 for *d*
- $R(P_8)$ in I_7 for *f*
- $R(P_9)$ in I_8 for *f*
- $R(P_{10})$ in I_9 for *b*
- $R(P_{11})$ in I_9 for *f*

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.



No such program is possible because there is no symbol for the 2nd formal of k in k's body scope

```

1 bool g;
2 bool L (bool b) { return true; }
3 bool k (int g, bool v) {
4     if (v) {
5         char m;
6         if (true) {
7             g = 4;
8         }
9     }
10 }
    
```

QUESTION 5 (12 POINTS)

Student ID: _____

- Consider the following grammar
- ① $Y ::= X$ *← translate the a lead to a boolean*
 - ② $X ::= a X b$ *← $t + a$ lead*
 - ③ $X ::= c X d$ *← $--$ a lead*
 - ④ $X ::= f$ *← a lead to 0*
- $\$1 = \$1 > 0$
 $\$2 = \$2 + 1$
 $\$3 = \$2 - 1$
 $\$4 = 0$

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

- $Y ::= X \#1$
 - $X ::= a X b \#2$
 - $X ::= c X d \#3$
 - $X ::= f \#4$
- $\#1: t = \text{semStack.pop}(); \text{semStack.push}(t > 0);$
 $\#2: t = \text{semStack.pop}(); \text{semStack.push}(t + 1);$
 $\#3: \text{semStack.push}(\text{semStack.pop}() - 1);$
 $\#4: \text{semStack.push}(0);$

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

	a	b	c	d	f	eof
Y	P_1 $X \#1$		P_1 $X \#1$	P_1 $X \#1$		
X	P_2 $a X b \#2$		P_3 $c X d \#3$		P_4 $f \#4$	