

Checkin 15

Draw the state of the symbol table betwixt lines 13 and 14

```
1. int a;  
2. bool f;  
3. int m(int arg) {  
4.     int b;  
5.     return arg + 1;  
6. }  
7.  
8. int g() {  
9.     int c;  
10.    int d;  
11.    if (a) {  
12.        int d;  
13.        int f;  
14.        int g;  
15.    }  
16. }
```

Administrivia

- In class check-ins up on Canvas

Flipped Wednesday



Written Work #4

Topics:

- Parser generation

Written Work #4: Question 1

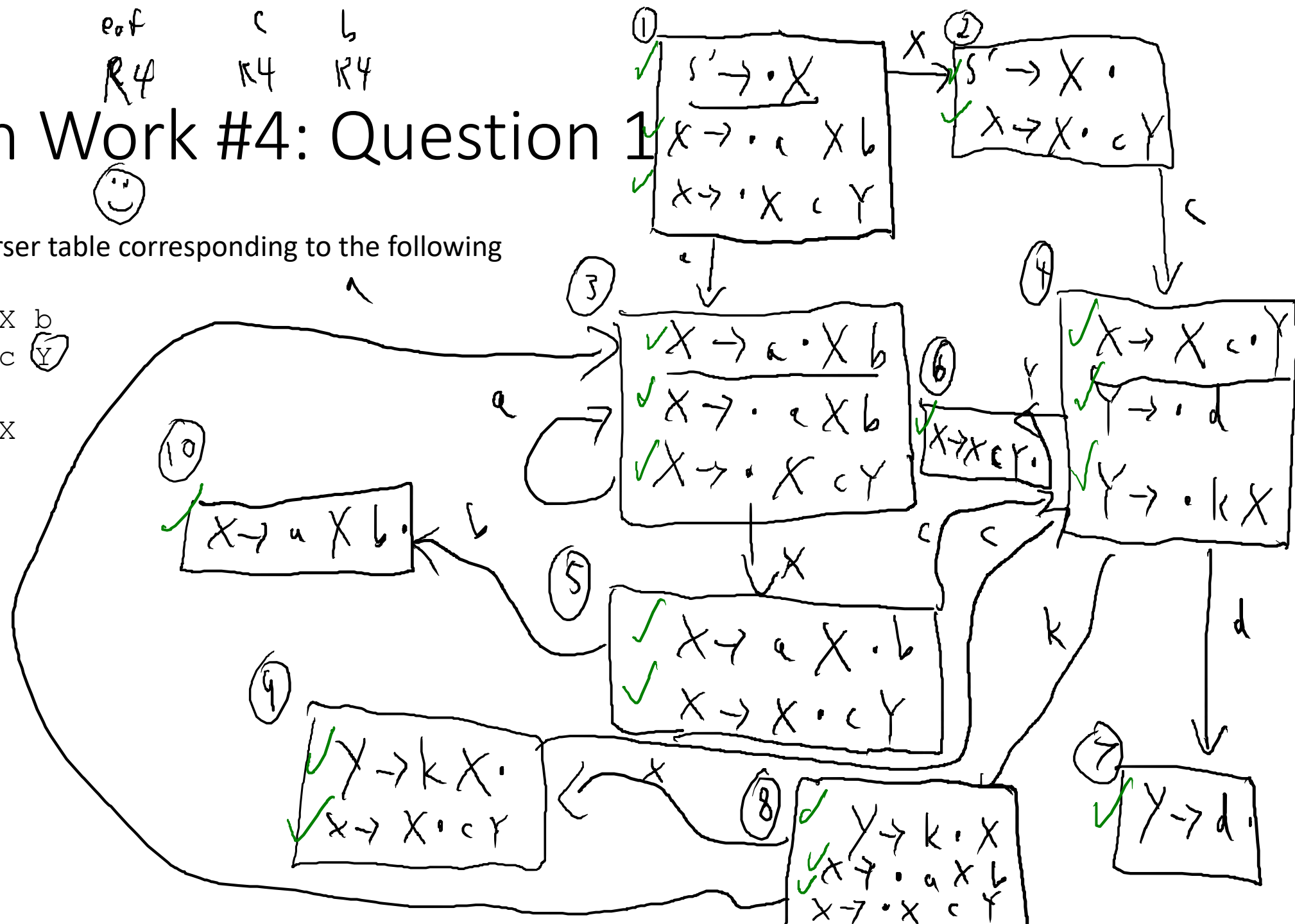
Draw the SLR parser table corresponding to the following grammar

$X ::= a X b$
 $X ::= X c Y$
 $Y ::= d$
 $Y ::= k X$

$I_4, Y \rightarrow I_6$

$I_4, d \rightarrow I_7$

$I_4, k \rightarrow I_8$



Written Work #4: Question 2

Is the prior grammar parseable by an SLR parser? What evidence do you have of that?

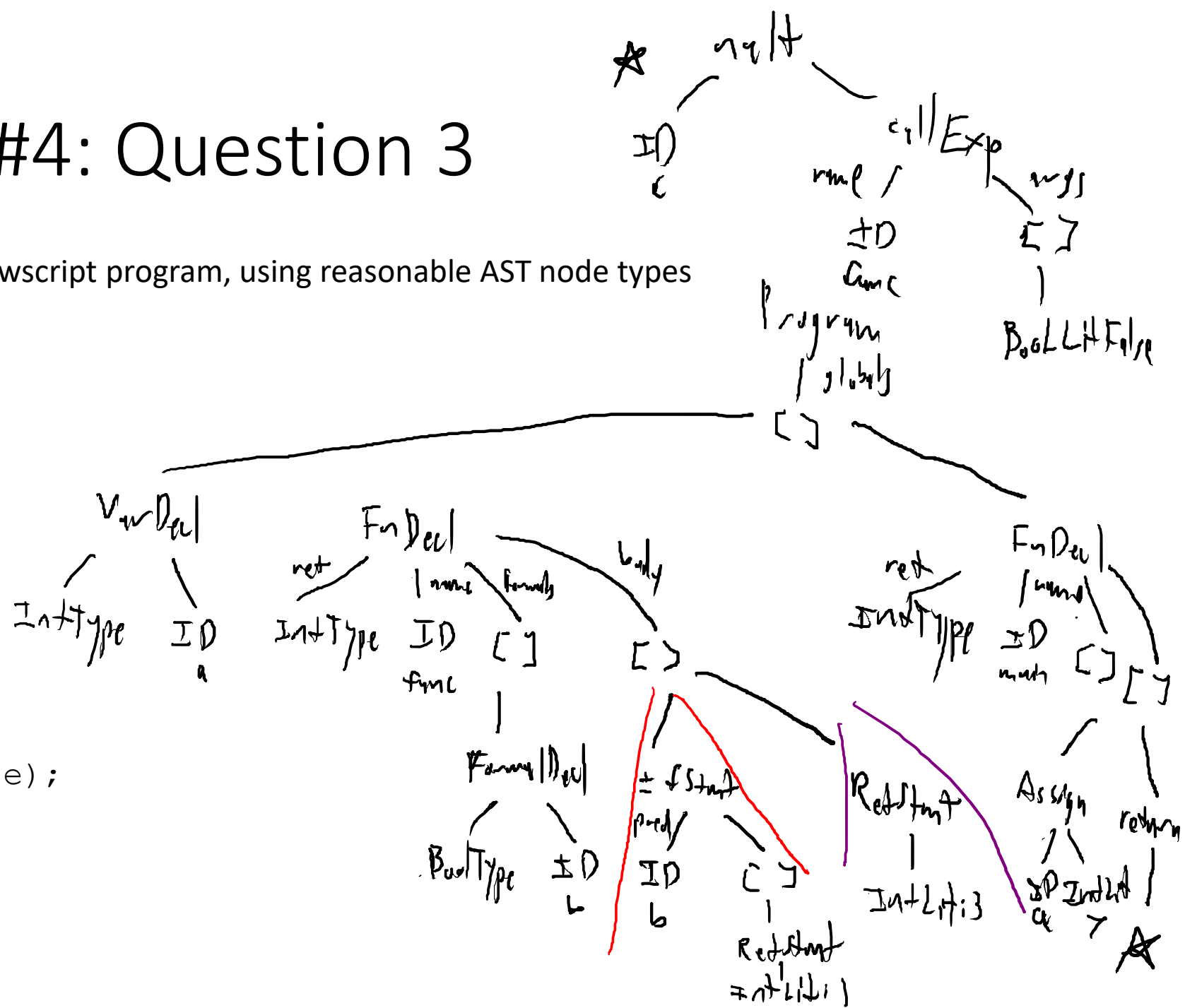
No, not SLR parseable, shift/reduce conflict at
state 19

Written Work #4: Question 3

Draw the AST for the following Drewscript program, using reasonable AST node types

```
int a;
int func(bool b) {
    if (b) {
        return 1;
    }
    return 3;
}

int main () {
    a = 7;
    return a * func(false);
}
```



Written Work #4: Question 4

List the FIRST and FOLLOW sets for the following grammar (do not transform the grammar):

$L ::= v \ m \ L$

$L ::= \epsilon$

$L ::= C$

$L ::= D$

$C ::= \epsilon$

$C ::= C \ k$

$D ::= C$

$C ::= m \ L$

$FIRST(L)$
 v, ϵ, m, k

$FIRST(C)$
 ϵ, k, m

$FIRST(D)$
 ϵ, k, m

$FOLLOW(L) = \text{eof}, k$

$FOLLOW(C) = k, \text{eof}$

$FOLLOW(D) = k, \text{eof}$