

Check-In C15

Review: LR Parsers

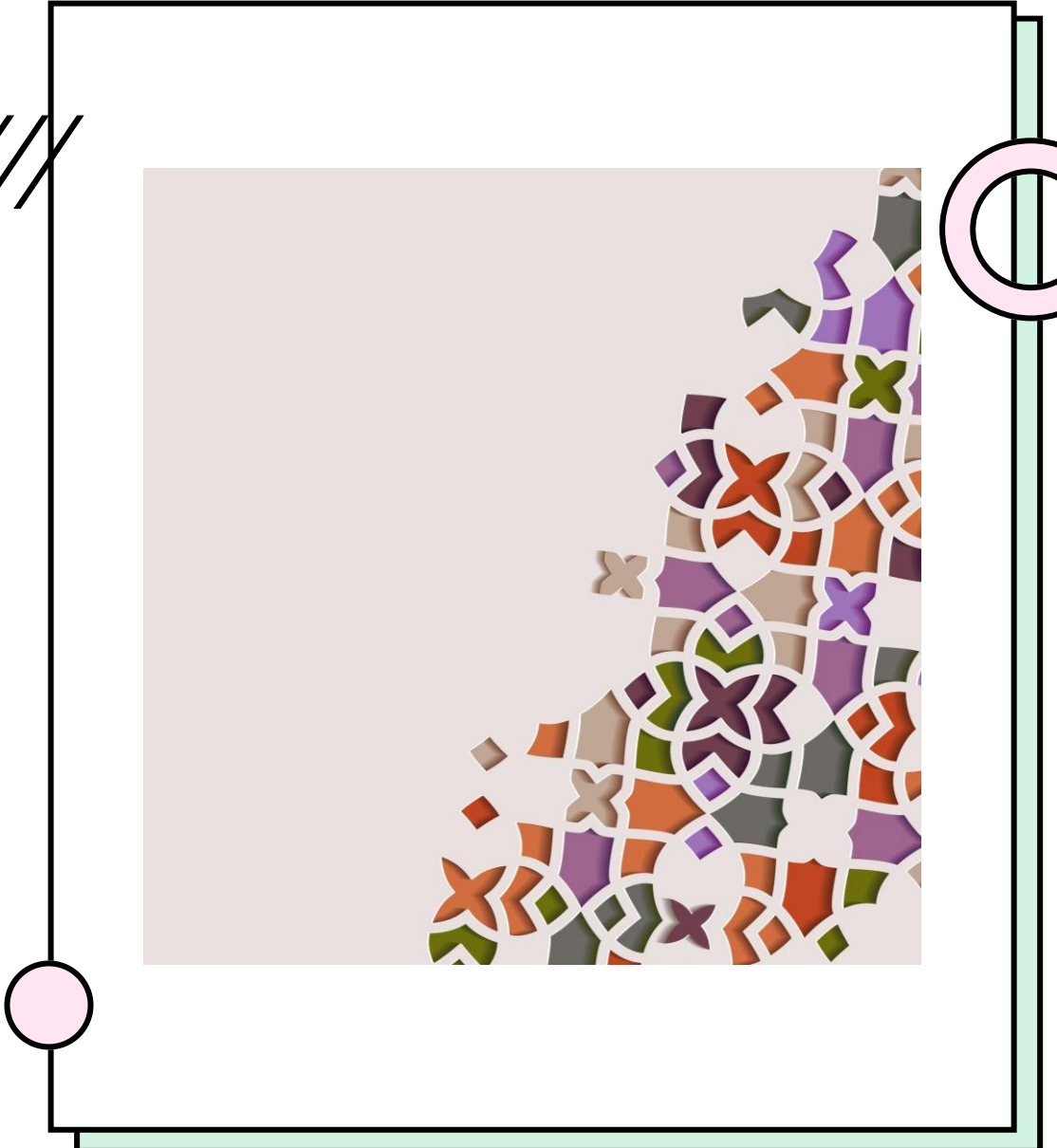
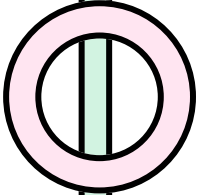
Draw the symbol table between line 12 and 13

```
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. }
```

Announcements

Housekeeping

**FLIPPED
WEDNESDAY**



○ Written Work #4

Topics:

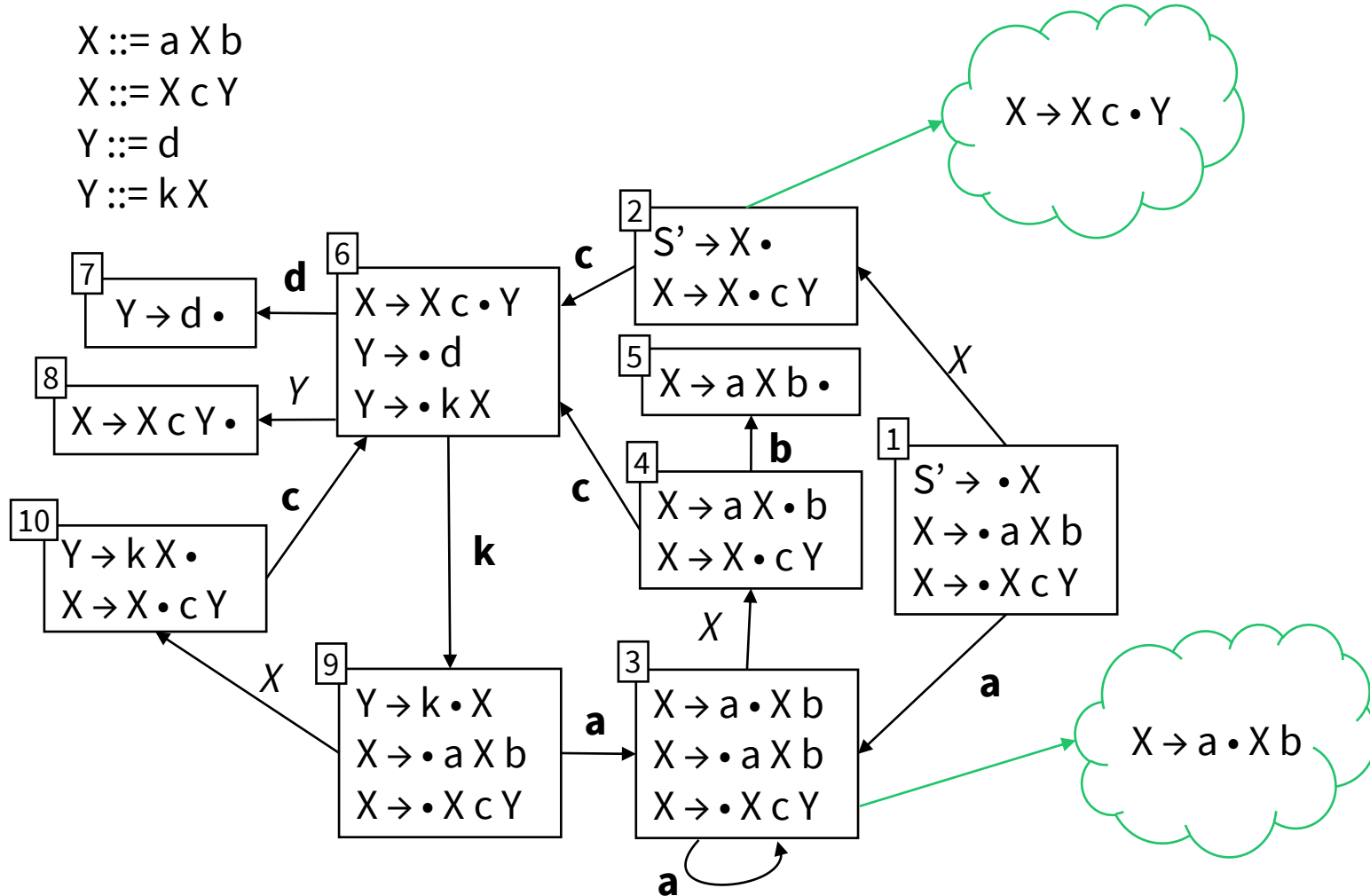
- Top-Down SDT
- LR Parsers
- SLR Parsers



Written Work #4: Question 1

Draw the SLR parser table corresponding to the following grammar

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

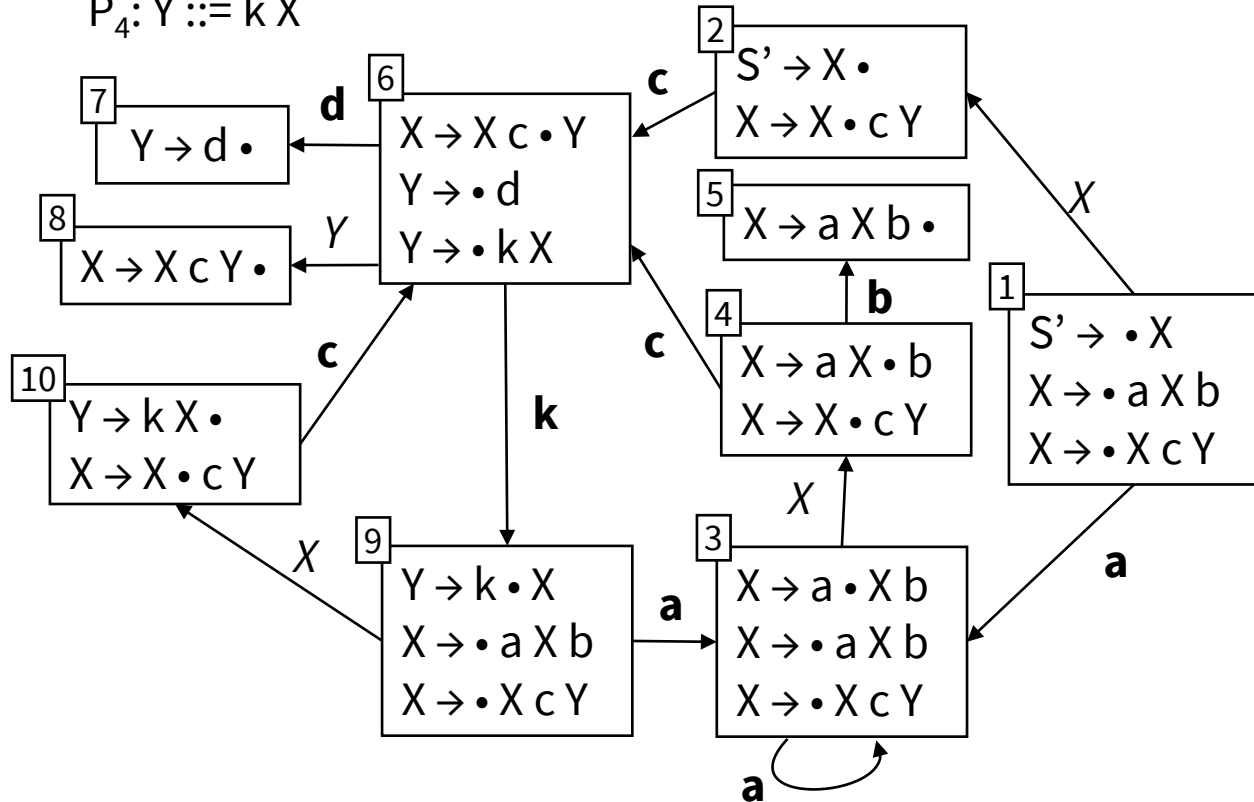


Written Work #4: Question 1

Draw the SLR parser table corresponding to the following grammar

- $P_1: X ::= a X b$
- $P_2: X ::= X c Y$
- $P_3: Y ::= d$
- $P_4: Y ::= k X$

$FOLLOW(X) = \{ eof, c, b \}$
 $FOLLOW(Y) = \{ b, eof, c \}$



	a	b	c	d	k	eof	X	Y
1	3						2	
2			6			☺		
3	3						4	
4		5	6					
5		R1	R1			R1		
6				7	9			8
7		R3	R3			R3		
8		R2	R2			R2		
9	3						10	
10		R4	6			R4		

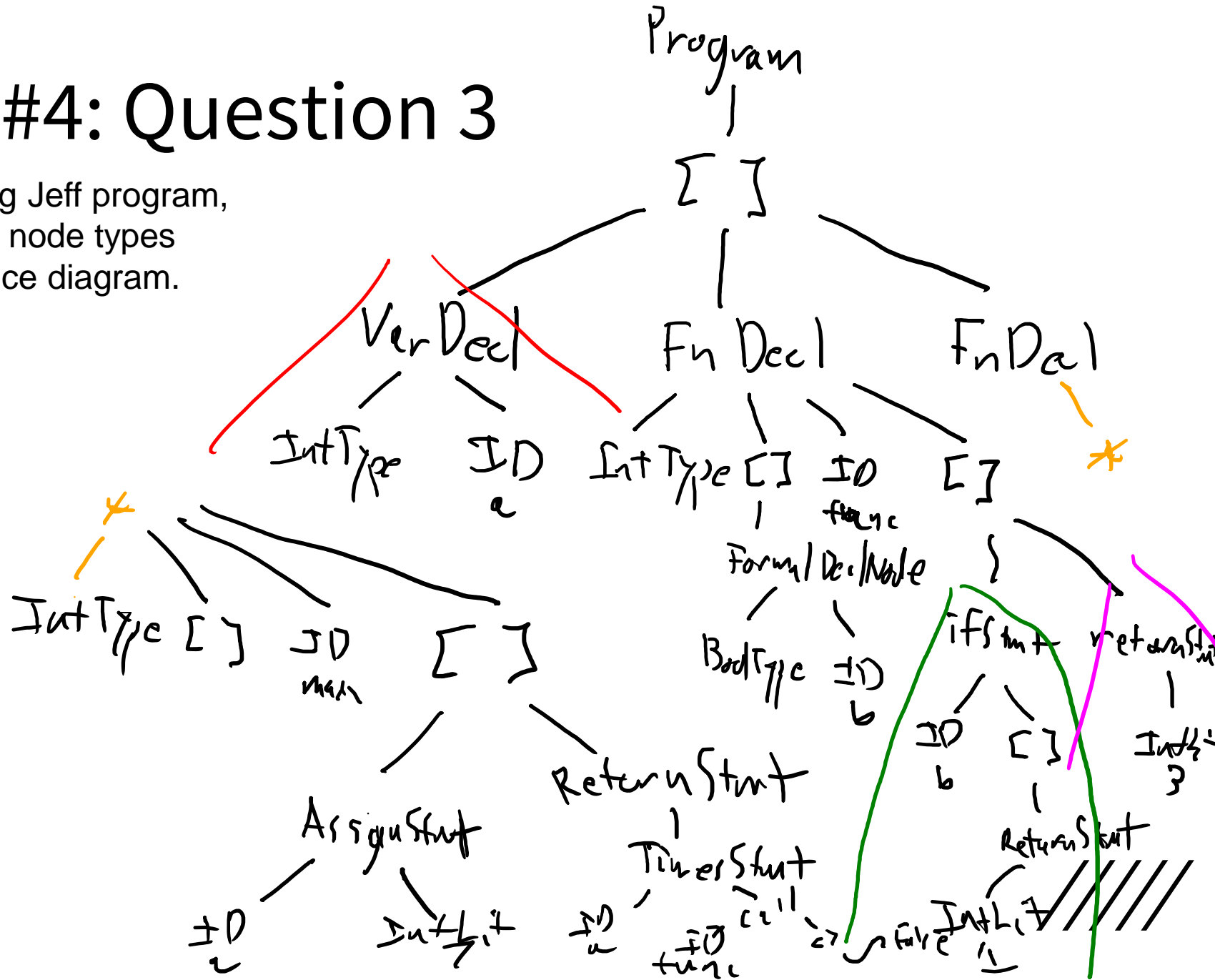
R4



Written Work #4: Question 3

Draw the AST for the following Jeff program, using the recommended AST node types suggested in the P3 inheritance diagram.

```
int a;  
fn : (bool b) int func {  
  if (b){  
    return 1;  
  }  
  return 3;  
}  
fn : () int main {  
  a = 7;  
  return a * func(false);  
}
```



Written Work #4: Question 4

List the FIRST and FOLLOW sets for the following grammar

$L ::= v m L$

$L ::= \epsilon$

$L ::= C$

$L ::= D$

$C ::= \epsilon$

$C ::= C k$

$D ::= C$

$C ::= m L$

FIRST(L) FOLLOW(L)
v, ϵ , k, m eof, k

FIRST(C) FOLLOW(C)
 ϵ , k, m eof, k

FIRST(D) FOLLOW(D)
k, m, ϵ eof, k

FIRST(v m L)
v

FIRST(C k)
k, m

FIRST(m L)
m



○ Auxiliary Question

Write an SLR Parser for the following grammar

$L ::= \epsilon$

$L ::= K$

$K ::= a \text{ plus } K$

$K ::= a$



○ Auxiliary Question

Create an LL(1) parser with semantic actions for the following grammar

```
L ::= ε           { LHS.trans = [ ]; }  
L ::= K           { LHS.trans = K.trans; }  
K ::= a plus K   { LHS.trans = [a.value].concat(K.trans); }  
K ::= a           { LHS.trans = [a.value] }
```



Written Work #4: Question 2

Is the Question 1 grammar parseable by an SLR parser? What evidence do you have of that?

	a	b	c	d	k	eof	X	Y
1	3						2	
2			6			☺		
3	3						4	
4		5	6					
5		R1	R1			R1		
6				7	9			8
7		R3	R3			R3		
8		R2	R2			R2		
9	3						10	
10		R4	6			R4		

R4

No, there is a collision in the parser table at $[w][c]$

