Chapter 5

5.8 Consider the following grammar

declaration→*type var-list*

type →***int | float***

var-list→***identifier***，*var-list* |***identifier***

a. Rewrite it in a form more suitable for bottom-up parsing.

b. Construct the DFA of LR(0) items for the rewritten grammar.

c. Construct the SLR(1) parsing table for the rewritten grammar.

d. Show the parsing stack and the actions of an SLR(1) parser for the input string **int x,y,z** using the table of part(c).

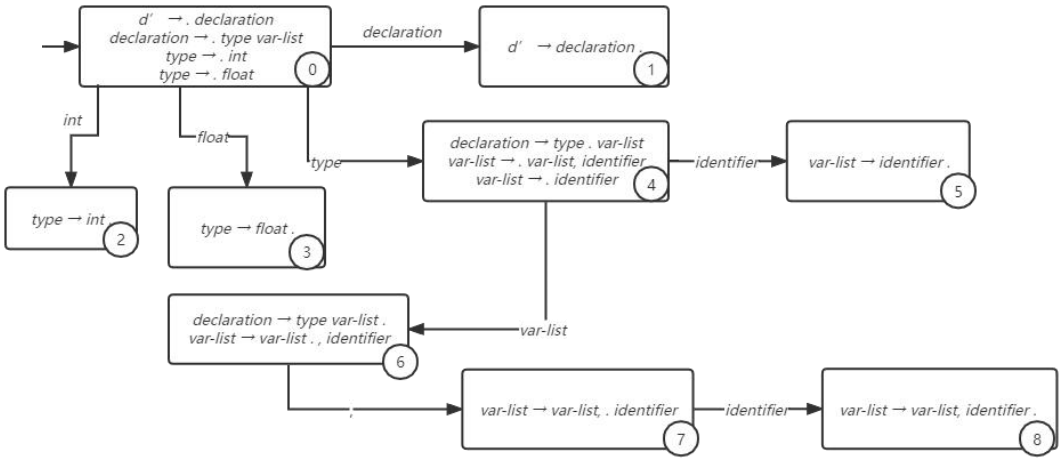
a.

declaration → *type var-list*

type →***int | float***

var-list→ *var-list,* ***identifier*** |***identifier***

b.



c.

Number:

(1) declaration → type var-list

(2) type → ***int***

(3) type → ***float***

(4) var-list → var-list, ***identifier***

(5) var-list → ***identifier***

State	**Input**					**Goto**		
	int	***float***	***identifier***	***,***	***$***	*declaration*	*type*	*var-list*
0	s2	s3				1	4	
1					accept			
2			r(2)					
3			r(3)					
4			s5					6
5				r(5)	r(5)			
6				s7	r(1)			
7			s8		r(4)			
8				r(4)	r(4)			

d.

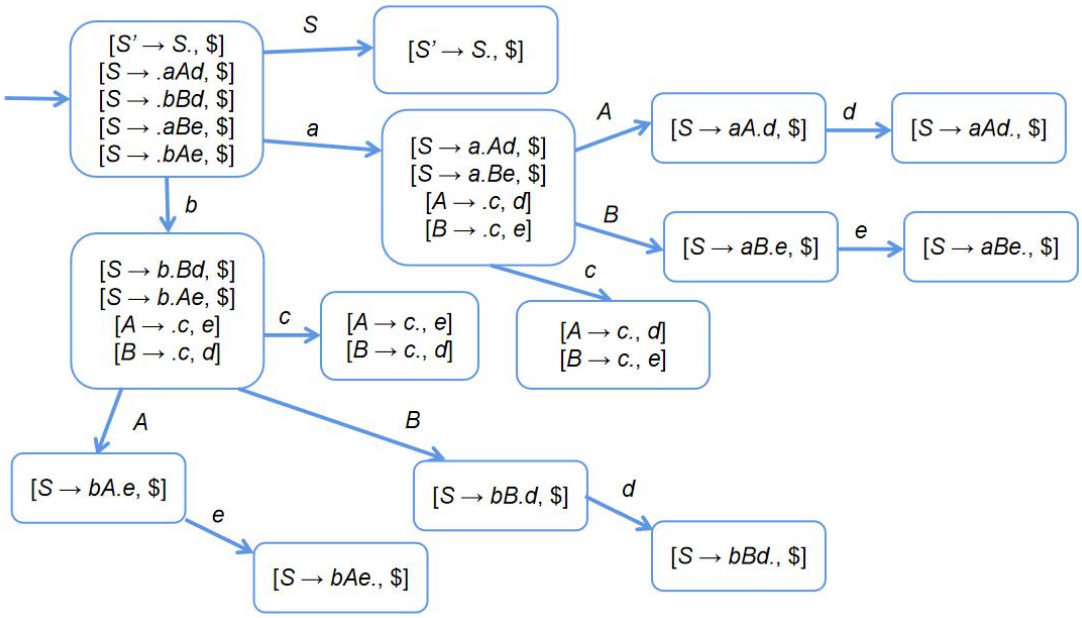
Parsing Stack	**Input**	**Action**
$0	***int x, y, z***$	shift 2
$0 ***int***2	***x, y, z***$	reduce 2
$0 *type*4	***x, y, z***$	shift 5
$0 *type*4 ***identifier***5	***, y, z***$	reduce 5
$0 *type*4 *var-list*6	***, y, z***$	shift 7
$0 *type*4 *var-list*6 ,7	***y, z***$	shift 8
$0 *type*4 *var-list*6 ,7 ***identifier***8	***, z***$	reduce 4
$0 *type*4 *var-list*6	***, z***$	shift 7
$0 *type*4 *var-list*6 ,7	***z***$	shift 8
$0 *type*4 *var-list*6 ,7 ***identifier***8	$	reduce 4
$0 *type*4 *var-list*6	$	reduce 1
$0 *declaration*1	$	accept

5.12 Show that the following grammar is LR(1) but not LALR(1):

$S \rightarrow aAd \mid bBd \mid aBe \mid bAe$

$A \rightarrow c$

$B \rightarrow c$



From the DFA of LR(1) we can see that this grammar is LR(1). The 2 states derived by inputting a c will be combined as one item in the DFA of LALR(1), since they only differ in the lookaheads. So the grammar is LR(1) but not LALR(1).