

2023 NCKU CSIE Compiler Quiz 2

1. (25 %) Build the CFSM for the following grammar:

1	<i>Prog</i>	\rightarrow	<i>StmtList</i> \$
2	<i>StmtList</i>	\rightarrow	<i>StmtList Stmt</i>
3	<i>Stmt</i>	\rightarrow	<i>Var assign Expr</i>
4	<i>Var</i>	\rightarrow	<i>id</i>
5			<i>id lb Expr rb</i>
6	<i>Expr</i>	\rightarrow	<i>Expr plus T</i>
7			<i>T</i>
8	<i>T</i>	\rightarrow	<i>Var</i>
9			<i>lp Expr rp</i>

2. (25 %) Show the SLR(1) parse table for the CFSM constructed in the above problem.
3. (25 %) A grammar for a simplified language that allows conditional statements follows:

1	<i>Start</i>	\rightarrow	<i>Stmt</i> \$
2	<i>Stmt</i>	\rightarrow	<i>if e then Stmt else Stmt</i>
3			<i>if e then Stmt</i>
4			<i>other</i>

Explain why the grammar is or is not LALR(1).

4. (25 %) Let S_1 , S_2 , S_3 , and S_4 denote the classes of grammars that can be parsed by LR(0), LR(1), LALR(1), and SLR(1) parsers, respectively. Please use a Venn diagram to illustrate the relationships among these four sets.

The diagram illustrates the LR(0) item set diagram for the grammar. It shows 17 states (State 0 to State 17) arranged in a complex flow. Each state is a table with LR(0) items and their transitions. Transitions are labeled with grammar symbols: \$, StmtList, Stmt, Var, id, lb, Expr, T, plus, rp, and Goto. The diagram illustrates the state transitions for the given grammar, including the initial State 0 and the final State 17.

States and their LR(0) items:

- State 0:**
 - Prog -> • StmtList \$
 - 1
 - 1
- State 1:**
 - Prog -> StmtList • \$
 - 2
 - 3
 - 4
 - 5
 - 5
- State 2:**
 - Prog -> StmtList \$ •
 - Goto
- State 3:**
 - StmtList -> StmtList Stmt •
 - Goto
- State 4:**
 - Stmt -> Var • assign Expr
 - Goto
 - 6
- State 5:**
 - Var -> id •
 - Var -> id lb Expr rb
 - 11
- State 6:**
 - Stmt -> Var assign • Expr
 - Goto
 - 7
- State 7:**
 - Expr -> • Expr plus T
 - Expr -> • T
 - T -> • Var
 - T -> • lp Expr rp
 - Var -> • id
 - Var -> • id lb Expr rb
 - 5
 - 5
- State 8:**
 - Expr -> T •
 - Goto
- State 9:**
 - T -> Var •
 - Goto
- State 10:**
 - T -> lp • Expr rp
 - 13
 - 8
 - 9
 - 10
 - 5
 - 5
- State 11:**
 - Var -> id lb Expr rb •
 - Goto
- State 12:**
 - Expr -> Expr plus • T
 - Goto
 - 15
 - 9
 - 10
 - 5
 - 5
- State 13:**
 - Expr -> • Expr plus T
 - Expr -> • T
 - T -> • Var
 - T -> • lp Expr rp
 - Var -> • id
 - Var -> • id lb Expr rb
 - 5
 - 5
- State 14:**
 - Var -> id lb Expr • rb
 - 17
 - 12
- State 15:**
 - Expr -> Expr plus T •
 - Goto
- State 16:**
 - T -> lp Expr rp •
 - Goto
- State 17:**
 - Var -> id lb Expr rb •
 - Goto

Transitions:

- State 0 to State 1:** StmtList
- State 1 to State 2:** \$
- State 1 to State 3:** Stmt
- State 1 to State 4:** Var
- State 1 to State 5:** id
- State 4 to State 6:** assign
- State 5 to State 11:** lb
- State 5 to State 8:** id
- State 5 to State 10:** T
- State 5 to State 13:** Var
- State 5 to State 14:** lp
- State 6 to State 7:** Expr
- State 7 to State 8:** T
- State 7 to State 9:** Var
- State 7 to State 10:** lp
- State 7 to State 12:** Expr
- State 7 to State 13:** plus
- State 7 to State 15:** T
- State 10 to State 13:** Expr
- State 10 to State 16:** rp
- State 12 to State 15:** T
- State 13 to State 16:** rp
- State 14 to State 17:** rb

2.

State	assign	id	lb	rb	plus	lp	rp	\$	Prog	StmtList	Stmt	Var	Expr	T
0									accept	1				
1		5						2			3	4		
2								1						
3		2						2						
4	6													
5	4	4	11	4	4		4	4						
6		5				10						9	7	8
7		3			12			3						
8		7		7	7		7	7						
9		8		8	8		8	8						
10		5				10						9	13	8
11		5				10						9	14	8
12		5				10						9		15
13					12		16							
14				17	12									
15		6		6	6		6	6						
16		9		9	9		9	9						
17	5	5		5	5		5	5						

3. Rule 2和3的ItemFollow皆存在else。
因此存在shift-reduce conflict, 本題文法並非LALR(1)。

4.

