

CMSC430 Spring 2009 Quiz 2 (Solutions)

1. (20 pts) LR parsing
- a. (10 pts) Given the following ACTION/GOTO table, show the parse of the string “cc”. Describe the stack, remaining input, and action performed at each step.

State	Action			Goto
	a	c	\$	
0	Shift 3	Shift 1	Accept	2
1	Reduce $S \rightarrow c$	Reduce $S \rightarrow c$	Reduce $S \rightarrow c$	3
2	Shift 3	Shift 4	Accept	1
3	Reduce $S \rightarrow Sa$	Reduce $S \rightarrow Sa$	Reduce $S \rightarrow Sa$	4
4	Reduce $S \rightarrow Sc$	Reduce $S \rightarrow Sc$	Reduce $S \rightarrow Sc$	0

Stack	Remaining Input	Action	Comment
\$ 0	c c \$	Shift 1	
\$ 0 c 1	c \$	Reduce $S \rightarrow c$	
\$ 0 S	c \$	Goto 2	Intermediate Step
\$ 0 S 2	c \$	Shift 4	
\$ 0 S 2 c 4	\$	Reduce $S \rightarrow Sc$	
\$ 0 S	\$	Goto 2	Intermediate Step
\$ 0 S 2	\$	Accept	

- b. (6 pts) Consider the following set of LR(1) items in the state of a LR(1) parser. Find all shift/reduce and reduce/reduce errors, and list the LR(1) items and lookaheads causing the conflicts.

$[A \rightarrow b \cdot B, a]$
 $[A \rightarrow \cdot a, b]$
 $[A \rightarrow bb \cdot, c]$
 $[B \rightarrow \cdot b, a]$
 $[B \rightarrow \cdot bbA, a]$
 $[B \rightarrow b \cdot, a]$

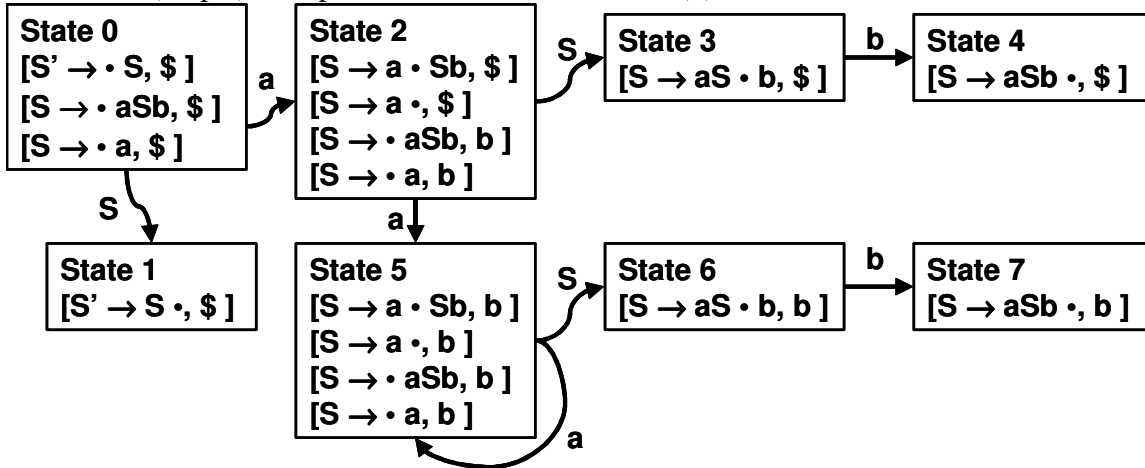
Shift/reduce error for LR(1) items $[A \rightarrow \cdot a, b]$ $[B \rightarrow b \cdot, a]$ on lookahead a.

- c. (4 pts) What is the effect on the associativity of an operator if a shift/reduce conflict for the operator is resolved by always performing shift? Explain why.
- Operator becomes right associative, since (left) operator on stack will be reduced after (right) operator in remaining input.**

2. (40 pts) LR(1) parsing

Consider the following grammar: $S' \rightarrow S$ $S \rightarrow aSb \mid a$

a. (22 pts) Compute the canonical set of LR(1) items



b. (12 pts) Construct the LR(1) ACTION/GOTO table

State	Action			Goto
	a	b	\$	
0	Shift 2			1
1			Accept	
2	Shift 5		Reduce $S \rightarrow a$	3
3		Shift 4		
4			Reduce $S \rightarrow aSb$	
5	Shift 5	Reduce $S \rightarrow a$		6
6		Shift 7		
7		Reduce $S \rightarrow aSb$		

c. (6 pts) Prove the grammar is or is not LALR(1)

Following is set of canonical items produced when merging states with identical cores. No reduce/reduce conflicts added, grammar is LALR(1).

