Parsing

Homework 9 (LR), due 22 June 2020

Laura Kallmeyer

SS 2020, Heinrich-Heine-Universität Düsseldorf

Question 1 (Canonical LR(1) construction and parse table)

1. *Consider the CFG G with non-terminals $N = \{S\}$, terminals $T = \{a, b\}$, start symbol S and productions $S \to Sa \,|\, aSb \,|\, \varepsilon$.*

   Consider the canonical LR(1)-construction for this grammar.

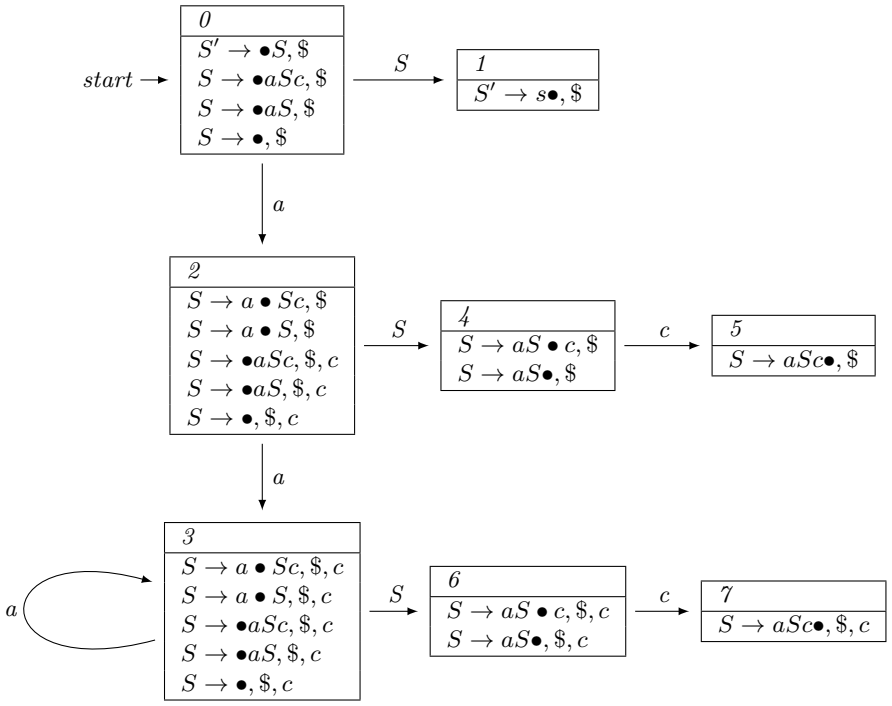
   (a) *How does the state $q_0 = closure(\{S' \to \bullet S\})$ look like?*

   (b) *Give the state $q_1 = goto - state(q_0, a)$.*

2. *Now consider the CFG $\langle N, T, P, S\rangle$ with*

   $N = \{S\}$, $T = \{a, c\}$, start symbol S, and productions $1.S \to aSc$, $2.S \to aS$, $3.S \to \varepsilon$.

   The LR(1)-automaton is as follows:



   Read off the LR(1) parse table.

Solution:

1. $q_0 =$

   $\{\langle S' \to \bullet S, \$\rangle, \langle S \to \bullet Sa, \$\rangle, \langle S \to \bullet aSb, \$\rangle, \langle S \to \bullet, \$\rangle, \langle S \to \bullet Sa, a\rangle, \langle S \to \bullet aSb, a\rangle, \langle S \to \bullet, a\rangle\}$

   $q_1 =$

   $\{\langle S \to a \bullet Sb, \$\rangle, \langle S \to a \bullet Sb, a\rangle, \langle S \to \bullet Sa, b\rangle, \langle S \to \bullet aSb, b\rangle, \langle S \to \bullet, b\rangle, \langle S \to \bullet Sa, a\rangle, \langle S \to \bullet aSb, a\rangle, \langle S \to \bullet, a\rangle\}$

2. Parse table:

	a	c	$	S
0	s2		r3	1
1			acc	
2	s3	r3	r3	4
3	s3	r3	r3	6
4		s5	r2	
5			r1	
6		s7, r2	r2	
7		r1	r1	

Question 2 (Canonical LR(1) parsing)

Consider the CFG $G_4 = \langle N, T, P, S\rangle$ with $N = \{S, A, B, C\}$, $T = \{a, b, c\}$ and productions 1.$S \to ABC$, 2.$A \to a$, 3.$A \to aC$, 4.$B \to b$, 5.$B \to bC$, 6.$C \to c$.

It's LR(1) parse table with the canonical LR algorithm is as follows:

	a	*b*	*c*	*$*	*A*	*B*	*C*	*S*
0	*s1*				*2*			*3*
1		*r2*	*s11*				*10*	
2		*s5*				*4*		
3				*acc*				
4			*s7*				*6*	
5			*s8, r4*				*9*	
6				*r1*				
7				*r6*				
8			*r6*					
9			*r5*					
10		*r3*						
11		*r6*						

1. *Is this grammar LR(1)?*
2. *Give the shift-reduce parsing trace (i.e., the sequence of pairs of stack and remaining input) that we obtain for the input $w = acbc$.*

   If the parser is not LR(1) and the input cannot be parsed deterministically, list all possibilities.

Solution:

1. No, because there are two entries in the field of state 5 and lookahead symbol *c*.
2.

stack	rem. input	
0	acbc$	
0 a 1	cbc$	
0 a 1 c 11	bc$	
0 a 1 C 10	bc$	
0 A 2	bc$	
0 A 2 b 5	c$	two options: s8, r4
0 A 2 b 5 c 8	$	results from s8, no further operations possible
0 A 2 B 4	c$	results from r4
0 A 2 B 4 c 7	$	
0 A 2 B 4 C 6	$	
0 S 3	$	acc