Parsing

Homework 8 (LR), due 14 June 2021

Laura Kallmeyer

SS 2021, Heinrich-Heine-Universität Düsseldorf

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

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

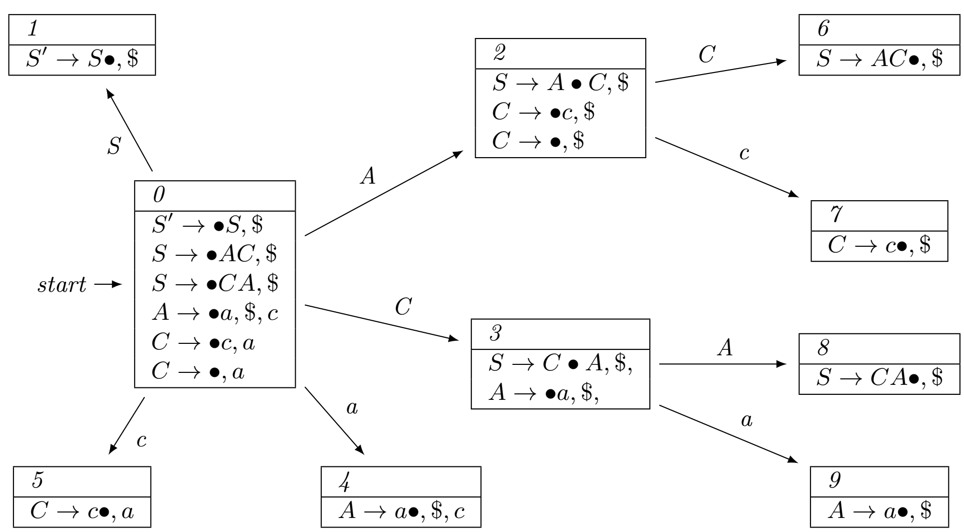
   Consider the canonical LR(1)-construction for this grammar.

   (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*
   $N = \{S, A, C\}$, $T = \{a, c\}$, start symbol S, and productions
   1. $S \to AC$ 2. $S \to CA$ 3. $A \to a$ 4. $C \to c$ 5. $C \to \varepsilon$.

   The LR(1)-automaton is as follows:



   Read off the LR(1) parse table.

Solution:

1. $q_0 =$
   $\{\langle S' \to \bullet S, \$\rangle, \langle S \to \bullet aS, \$\rangle, \langle S \to \bullet aS, a\rangle, \langle S \to \bullet Sa, \$\rangle, \langle S \to \bullet Sa, a\rangle, \langle S \to \bullet, \$\rangle, \langle S \to \bullet, a\rangle\}$

   $q_1 =$
   $\{\langle S \to a \bullet S, \$\rangle, \langle S \to a \bullet S, a\rangle, \langle S \to \bullet aS, \$\rangle, \langle S \to \bullet aS, a\rangle, \langle S \to \bullet Sa, \$\rangle, \langle S \to \bullet Sa, a\rangle, \langle S \to \bullet, \$\rangle\ \langle S \to \bullet, a\rangle\}$

2. Parse table:

	a	c	$	S	A	C
0	s4, r5	s5		1	2	3
1			acc			
2		s7	r5			6
3	s9				8	
4		r3	r3			
5	r4					
6			r1			
7			r4			
8			r2			
9			r3			

Question 2 (Canonical LR(1) parsing)

Consider the CFG $\langle N, T, P, S \rangle$ *with*

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

Its LR(1) parse table with the canonical LR algorithm is as follows:

	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*	

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* $w = aac$*. Number the single configurations and specify for each one, from which other it was obtained, and by which operation. In other words, the start of the table looks as follows:*

	stack	*rem. input*	*operation that was performed*
1.	*0*	*aac$*	–
2.	*0 a 2*	*ac$*	*s2 from 1.*

   If the parser is not LR(1) and the input cannot be parsed deterministically, list all possibilities.
3. *List the rightmost derivations the parser yields, according to its successful configurations. More precisely, give the numbers of the configurations belonging to each derivation and then give the corresponding derivation.*
4. *Which of the configurations (i.e., pairs of stack and remaining input) are not part of a successful parse?*

Solution:

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

2.

	stack	rem. input	operation that was performed
1.	0	aac$	–
2.	0 a 2	ac$	s2 from 1.
3.	0 a 2 a 3	c$	s3 from 2.
4.	0 a 2 a 3 S 6	c$	r3 from 3.
5.	0 a 2 a 3 S 6 c 7	$	s7 from 4.
6.	0 a 2 S 4	c$	r2 from 4.
7.	0 a 2 S 4	$	r1 from 5.
8.	0 a 2 S 4 c 5	$	s5 from 6.
9.	0 S 1	$	r2 from 7.
10.	0 S 1	$	r1 from 8.
11.	–	$	acc from 9,10.

3. There are two sequences of configurations that lead to successful parses, the one ending in 9. and the one ending in 10.

   Configurations 1.,2.,3.,4.,5.,7.,9., the reduces in in reverse order are r2, r1, r3, which gives

   $$S \Rightarrow aS \Rightarrow aaSc \Rightarrow aac$$

   Configurations 1.,2.,3.,4.,6.,8.,10., the reduces in reverse order are r1, r2, r3, which gives

   $$S \Rightarrow aSc \Rightarrow aaSc \Rightarrow aac$$

4. None, as we have just seen. All configurations are part of the two parses of the input.