SLR parser

Ex. $G = (\{S', E, T\},\ \{+, id, const, (,)\},\ P,\ S')$

P: $S' \rightarrow E$

(1) $E \rightarrow T$

(2) $E \rightarrow E + T$

(3) $T \rightarrow (E)$

(4) $T \rightarrow id$

(5) $T \rightarrow const$

$w = id + const$

1. Compute the canonical collection

// [redacted]

S0 = closure({[S' -> .E]}) = {[S' -> .E], [E->.T], [E-> .E + T], [T -> .(E)], [T -> .id], [T -> .const]}
S1 = goto(s0, E) = closure({[S' -> E.],[E -> E.+T]}) = {[S' -> E.],[E -> E.+T]}
S2 = goto(s0, T) = closure({[E -> T.]}) = {[E -> T.]}
S3 = goto(s0, () = closure({[T -> (.E)]}) = {[T -> (.E)], [E -> .T], [E->.E+T], [T->.(E)], [T->.id], [T->.const]}
S4 = goto(s0, id) = closure({[T -> id.]}) = {[T -> id.]}
S5 = goto(s0, const) = closure({[T -> const.]}) = {[T -> const.]}
S6 = goto(s1, +) = closure({[E -> E+.T]}) = {[E -> E+.T], [T -> .(E)], [T -> .id], [T -> .const]}
S7 = goto(s3, E) = closure({[T -> (E.)], [E -> E.+T]}) = {[T -> (E.)], [E -> E.+T]}
goto(s3, T) = closure({[E-> T.]}) = S2
goto(s3, id) = closure({[T -> id.]}) = S4
goto(s3, const) = closure({[T -> const.]}) = S5
goto(s3, () = closure({[T -> (.E)]}) = S3
S8 = goto(s6, T) = closure({[E -> E+T.]})
goto(s6, () = closure({[T->(.E)]}) = s3
goto(s6, id) = closure({[T -> id.]}) = s4
goto(s6, const) = closure({[T -> const.]}) = s5
S9 = goto(s7,)) = closure({[T -> (E).]}) = {[T -> (E).]}
goto(s7, +) = closure({[E -> E+.T]}) = s6

FOLLOW(E) = { ε, +,)}
FOLLOW(T) = { ε, +,)}

2. Fill the SLR table
//[redacted]

	ACTION						GOTO	
	+	**(**	**)**	**id**	**const**	**$**	**E**	**T**
0		Shift 3		Shift 4	Shift 5		1	2
1	Shift 6					acc		
2	Reduce1		Reduce1			Reduce1		
3		Shift 3		Shift 4	Shift 5		7	2
4	Reduce4		Reduce4			Reduce4		
5	Reduce 5		Reduce 5			Reduce 5		
6		Shift3		Shift4	Shift5			8
7	Shift6		Shift9					
8	Reduce 2		Reduce 2			Reduce 2		
9	Reduce 3		Reduce 3			Reduce 3		

3. Parse the sequence
// [redacted]

Work stack	**Input stack**	**Output band**
$0 $0id4 $0T2 $0E1 $0E1+6 $0E1+6const5 $0E1+6T8 $0E1 accept	id+const$ +const$ +const$ +const$ const$ $ $ $	ε ε 4 14 14 14 514 2514

E => E + T => E + const => T + const => id + const
2 5 1 4

LR(1) parser

Ex. $G = (\{S', S, A\},\ \{a, b\},\ P,\ S')$

P: $S' \rightarrow S$

(1) $S \rightarrow AA$

(2) $A \rightarrow aA$

(3) $A \rightarrow b$

w =

LR(1) item $[A \rightarrow \alpha.\beta,\ a]$

FIRST(S) = {a,b}
FIRST(A) = {a,b}

1. Canonical collection
   //[redacted]

   S0 = closure({[S' -> .S, $]}) = {[S' -> .S, $], [S->.AA, $], [A->.aA, a], [A->.aA, b], [A->.b, a], [A->.b, b] }
   S1 = goto(S0, S) = closure({[S' -> S., $]}) = {[S' -> S., $]}
   S2 = goto(S0, A) = closure({[S->A.A, $]}) = {[S->A.A, $], [A -> .aA, $], [A -> .b, $]}
   S3 = goto(S0, a) = closure({[A-> a.A, a], [A->a.A, b]}) = {[A -> a.A, a], [A -> a.A, b], [A -> .aA, a], [A -> .b, a], [A -> .aA, b], [A -> .b, b]}
   S4 = goto(S0, b) = closure({ [A-> b., a], [A-> b., b] }) = { [A -> b., a], [A -> b., b] }
   S5 = goto(S2, A) = closure({[S->AA., $]}) = {[S->AA., $]}
   S6 = goto(S2, a) = closure({[A -> a.A, $]}) = { [A -> a.A, $], [A -> .aA, $], [A -> .b, $] }
   S7 = goto(S2, b) = closure({[A -> b., $]}) = { [A -> b., $] }
   S8 = goto(S3, A) = closure({[A -> aA., a], [A -> aA., b]}) = { [A -> aA., a], [A -> aA., b] }
   goto(S3, a) = closure({ [A -> a.A, a], [A -> a.A, b]}) = S3
   goto(S3, b) = closure({[A -> b., a], [A -> b., b]}) = S4
   S9 = goto(S6, A) = closure({ [A -> aA., $] }) = { [A -> aA., $] }
   goto(S6, a) = closure({ [A -> a.A, $] }) = S6
   goto(S6, b) = closure({ [A -> b., $] }) = S7

2. Fill the LR(1) table
   //[redacted]

	ACTION			GOTO	
	a	**b**	**$**	**S**	**A**
0	**Shift 3**	**shift4**		1	2
1			**accept**		
2	**shift6**	**shift7**			5
3	**shift3**	**shift4**			8
4	**reduce3**	**reduce3**			
5			**reduce1**		
6	**shift6**	**shift7**			9
7			**reduce3**		
8	**reduce2**	**reduce2**			
9			**reduce2**		

3. Syntactical Analysis
W = abab
/[redacted]

Work stack	**Input stack**	**Output band**
$0 $0a3 $0a3b4 $0a3A8 $0A2 $0A2a6 $0A2a6b7 $0A2a6A9 $0A2A5 $0S1 AC	abab$ bab$ ab$ ab$ ab$ b$ $ $ $ $	- - - 3 23 23 23 323 2323 x

LALR(1) parser

Ex. $G = (\{S', S, A\},\ \{a, b\},\ P,\ S')$

P: $S' \rightarrow S$

(1) $S \rightarrow AA$

(2) $A \rightarrow aA$

(3) $A \rightarrow b$

W = aaab

1. Canonical collection

$s_0 = \{[S' \rightarrow .S,\ \$],\ [S \rightarrow .AA,\ \$], [A \rightarrow .aA, a], [A \rightarrow .aA, b],\ , [A \rightarrow .b,\ a],\ [A \rightarrow .b,\ b]\}$

$s_1 = \{[S' \rightarrow S.,\ \$]\}$

$s_2 = \{[S \rightarrow A.A,\ \$],\ [A \rightarrow .aA, \$], [A \rightarrow .b, \$]\}$

$s_{36} = \{[A \rightarrow a.A, a/b/\$],\ [A \rightarrow .aA, a/b/\$],\ [A \rightarrow .b, a/b/\$]\}$

$s_{47} = \{[A \rightarrow b.,\ a/b/\$]\}$

$s_5 = \{[S \rightarrow AA.,\ \$]\}$

$s_{89} = \{[A \rightarrow aA.,\ a/b/\$]\}$

2. LALR(1) table

	ACTION			GOTO	
	a	**b**	**$**	**S**	**A**
s0	Shift s36	Shift s47		s1	s2
s1			accept		
s2	Shift s36	Shift s47			s5
s36	Shift s36	Shift s47			s89
s47	Reduce 3	Reduce 3	Reduce 3		
s5			Reduce 1		
s89	Reduce 2	Reduce 2	Reduce 2		

3. Parse the sequence

Work stack	Input stack	Output band

$ s0
$ s0 a s36
$ s0 a s36 a s36
$ s0 a s36 a s36 a s36
$ s0 a s36 a s36 a s36 b s47
$ s0 a s36 a s36 a s36 A s89
$ s0 a s36 a s36 A s89
$ s0 a s36 A s89
$ s0 A s2
	a a a b $
	a a b $
	a b $
	b $
	$
	$
	$
	$
	$
		Eps
		Eps
		Eps
		Eps
		Eps
		3
		23
		223
		2223