Seminar 10

|SLR|

S → simple
L → left-to-right scan of the input
R → right most derivation in reverse

SLR is very similar to LR(0) having one difference: for reductions the prediction of length 1 is considered (FOLLOW) instead of prediction of length 0 used in LR(0)

SLR steps

- canonical collection (same as for LR(0))
- SLR table (similar with LR(0)) with one mention: for terminal symbols action ang goto columns merge together
- use the table for parsing (same process as for LR(0))

Fill in SLR table rules

- if $[A \to \alpha . \beta] \in s_i$ then action(i, u) = shift $\forall u \in \Sigma$ (terminal) and goto$(s_i, u) \neq \emptyset$
- if $[A \to \beta .] \in s_i$ and $A \neq S'$ then action(i, u) = reduce x where x is the production number and $u \in$ FOLLOW(A)
- if $[S' \to S.] \in s_i$ then action$(i, \$)$ = accept
- if goto$(s_i, X) = s_j$ then goto$(i, X) = j$

Example

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

P: $S' \rightarrow E$

$E \rightarrow T$ (1)

$E \rightarrow E+T$ (2)

$T \rightarrow (E)$ (3)

$T \rightarrow id$ (4)

$T \rightarrow const$ (5)

- Canonical collection

$s_0 = closure(\{[S' \rightarrow .E]\}) = \{[S' \rightarrow .E], [E \rightarrow .T], [E \rightarrow .E+T]$
$[T \rightarrow .(E)], [T \rightarrow .id], [T \rightarrow .const]\}$

$s_1 = goto(s_0, E) = closure(\{[S' \rightarrow E.], [E \rightarrow E.+T]\})$

$s_2 = goto(s_0, T) = closure(\{[E \rightarrow T.]\}) = \{[E \rightarrow T.]\}$

$s_3 = goto(s_0, () = closure(\{[T \rightarrow (.E)]\}) = \{[T \rightarrow (.E)],$
$[E \rightarrow .T], [E \rightarrow .E+T], [T \rightarrow .(E)], [T \rightarrow .id],$
$[T \rightarrow .const]\}$

$s_4 = goto(s_0, id) = closure(\{[T \rightarrow id.]\}) = \{[T \rightarrow id.]\}$

$s_5 = goto(s_0, const) = closure(\{[T \rightarrow const.]\}) = \{[T \rightarrow const.]\}$

$s_6 = goto(s_1, +) = closure(\{[E \rightarrow E+.T]\}) = \{[E \rightarrow E+.T],$
$[T \rightarrow .(E)], [T \rightarrow .id], [T \rightarrow .const]\}$

$s_7 = goto(s_3, E) = closure(\{[T \rightarrow (E.)], [E \rightarrow E.+T]\})$
$= \{[T \rightarrow (E.)], [E \rightarrow E.+T]\}$

$goto(s_3, T) = closure(\{[E \rightarrow T.]\}) = s_2$

$goto(s_3, id) = closure(\{[T \to id\circ]\}) = s_4$

$goto(s_3, const) = closure(\{[T \to const\circ]\}) = s_5$

$goto(s_3, () = closure(\{[T \to (\circ E)]\}) = s_3$

$s_8 = goto(s_6, T) = closure(\{[E \to E+T\circ]\}) = \{[E \to E+T\circ]\}$

$goto(s_6, () = closure(\{[T \to (\circ E)]\}) = s_3$

$goto(s_6, id) = closure(\{[T \to id\circ]\}) = s_4$

$goto(s_6, const) = closure(\{[T \to const\circ]\}) = s_5$

$s_9 = goto(s_7,)) = closure(\{[T \to (E)\circ]\}) = \{[T \to (E)\circ]\}$

$goto(s_7, +) = closure(\{[E \to E+\circ T]\}) = s_6$

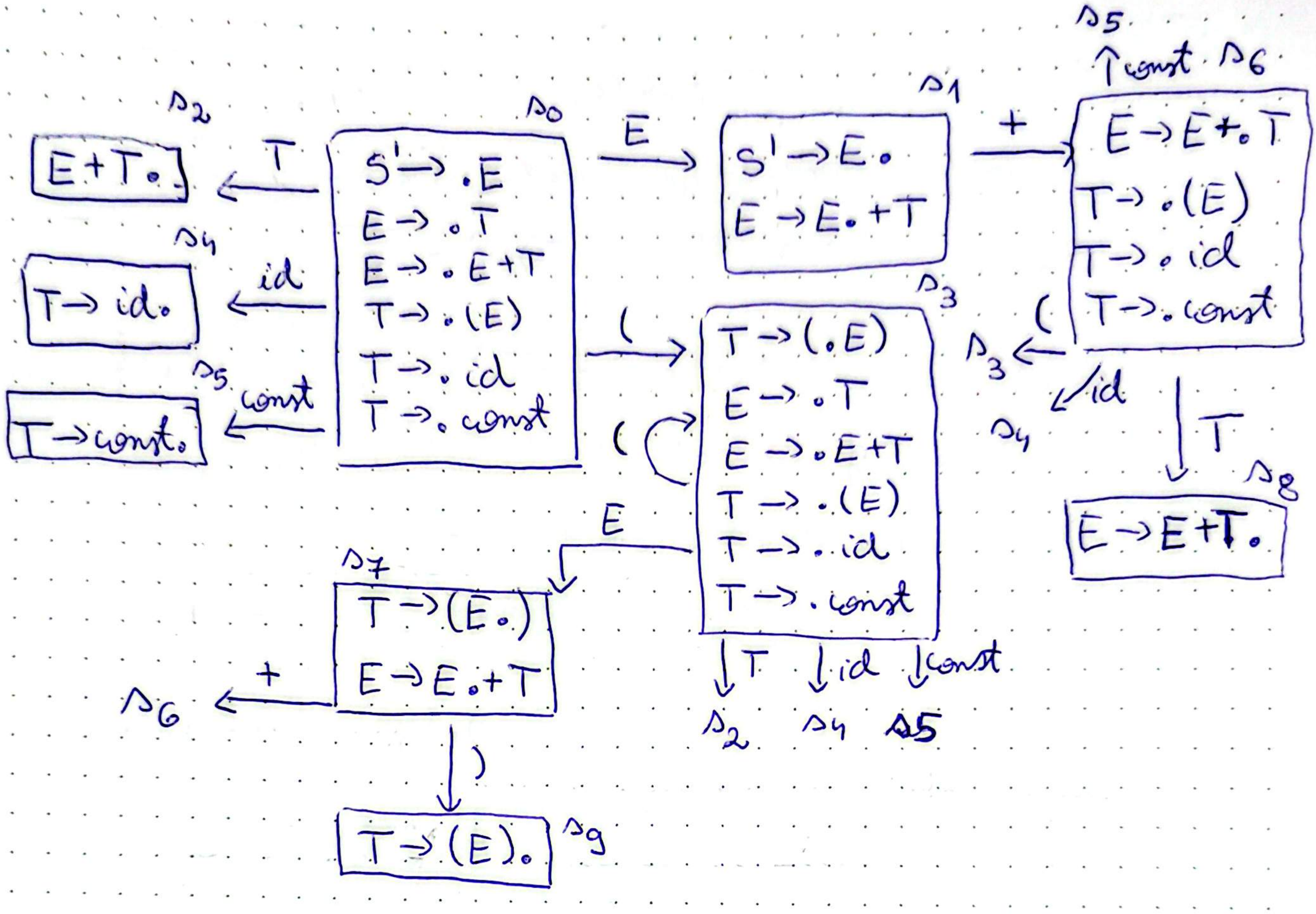
$goto(s_2, X) = \emptyset, \forall X \in N \cup \Sigma$

$goto(s_4, X) = \emptyset, \forall X \in N \cup \Sigma$

$goto(s_5, X) = \emptyset, \forall X \in N \cup \Sigma$

$goto(s_8, X) = \emptyset, \forall X \in N \cup \Sigma$

$goto(s_9, X) = \emptyset, \forall X \in N \cup \Sigma$



$FOLLOW(E) = \{ \varepsilon, +,) \}$

$FOLLOW(T) = \{ \varepsilon, +,) \}$

- SLR TABLE

s	ACTION						GOTO	
	+	(	)	id	const	$	E	T
0		shift 3		shift 4	shift 5		1	2
1	shift 6					acc		
2	reduce 1		reduce 1			reduce 1		
3		shift 3		shift 4	shift 5		7	2
4	reduce 4		reduce 4			reduce 4		
5	reduce 5		reduce 5			reduce 5		
6		shift 3		shift 4	shift 5			8
7	shift 6		shift 9					

	+	(	)	id	const	$	E	T
8	reduce 2		reduce 2			reduce 2		
9	reduce 3		reduce 3			reduce 3		

Parse sequence $w = id + const$

work stack α	input stack β	output band π
$0	id + const $	ε
$0 id 4	+ const $	ε
$0 T 2	+ const $	4
$0 E 1	+ const $	1, 4
$0 E 1 + 6	const $	1, 4
$0 E 1 + 6 const 5	$	1, 4
$0 $\underline{E 1 + 6 T 8}$	$	5, 1, 4
$0 E 1	$	2, 5, 1, 4
accept		2 5 1 4

$w_2 = id + (const + id)$

α	β	π
$0	id + (const + id) $	ε
$0 id 4	+ (const + id) $	ε
$0 T 2	+ (const + id) $	4
$0 E 1	+ (const + id) $	1, 4
$0 E 1 + 6	(const + id) $	1, 4
$0 E 1 + 6 (3	const + id) $	1, 4
$0 E 1 + 6 (3 const 5	+ id) $	1, 4
$0 E 1 + 6 (3 T 2	+ id) $	5 1 4
$0 E 1 + 6 (3 E 7	+ id) $	1 5 1 4
$0 E 1 + 6 (3 E 7 + 6	id) $	1 5 1 4
$0 E 1 + 6 (3 E 7 + 6 id 4	) $	1 5 1 4
$0 E 1 + 6 (3 $\underline{E 7 + 6 T 8}$	) $	1 5 1 4
$0 E 1 + 6 (3 E 7	) $	2 1 5 1 4

\$0E1+6(3E7	)\$	21514
\$0E1+6(3E7)9	\$	21514
\$0E1+6T8	\$	321514
\$0E1	\$	2321514
accept		2321514

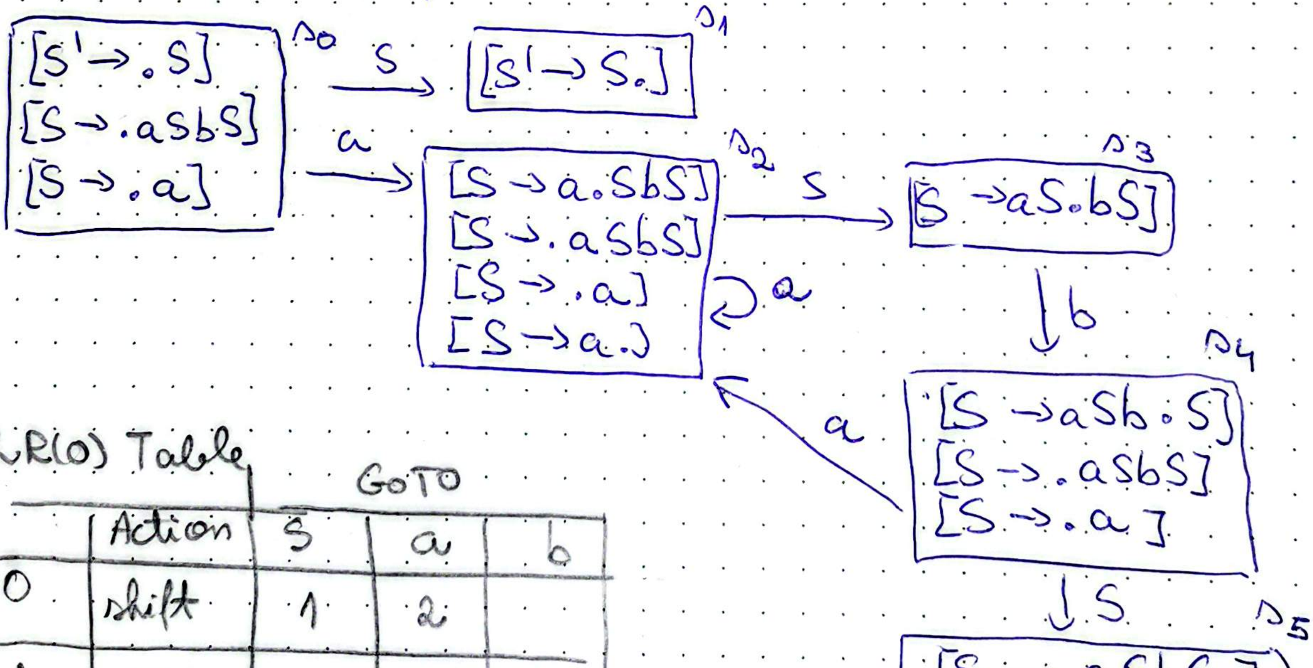
← prev. line

Ex 2

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

P : $S' \to S$
$S \to aSbS$ (1)
$S \to a$ (2)

Is the grammar LR(0)? But SLR? Use the proper parser to verify if w = aabaaba ∈ L(G)



• LR(0) Table

	Action	GOTO		
		S	a	b
0	shift	1	2	
1	acc.			
2	reduce 2 shift			
3				
4				
5				

→ conflict shift reduce

⇒ G is NOT a LR(0) grammar

- SLR table

state	Action			GOTO
	a	b	$	S
0	shift 2			1
1			acc	
2	shift 2	reduce 2	reduce 2	3
3		shift 4		
4	shift 2			5
5		reduce 1	reduce 1	

FOLLOW(S) = {ε, b}

No conflicts in this table ⇒
⇒ G is a SLR grammar
⇒ we can use SLR parser to parse it

α	β	π
$0	aabaaba$	ε
$0a2	abaaba$	ε
$0a2a2	baaba$	ε
$0a2S3	baaba$	2
$0a2S3b4	aaba$	2
$0a2S3b4a2	aba$	2
$0a2S3b4a2a2	ba$	2
$0a2S3b4a2S3	ba$	22
$0a2S3b4a2S3b4	a$	22
$0a2S3b4a2S3b4a2	$	22
$0a2S3b4a2S3b4S5	$	222
$0a2S3b4S5	$	1222
$0S1	$	11222
accept		11222