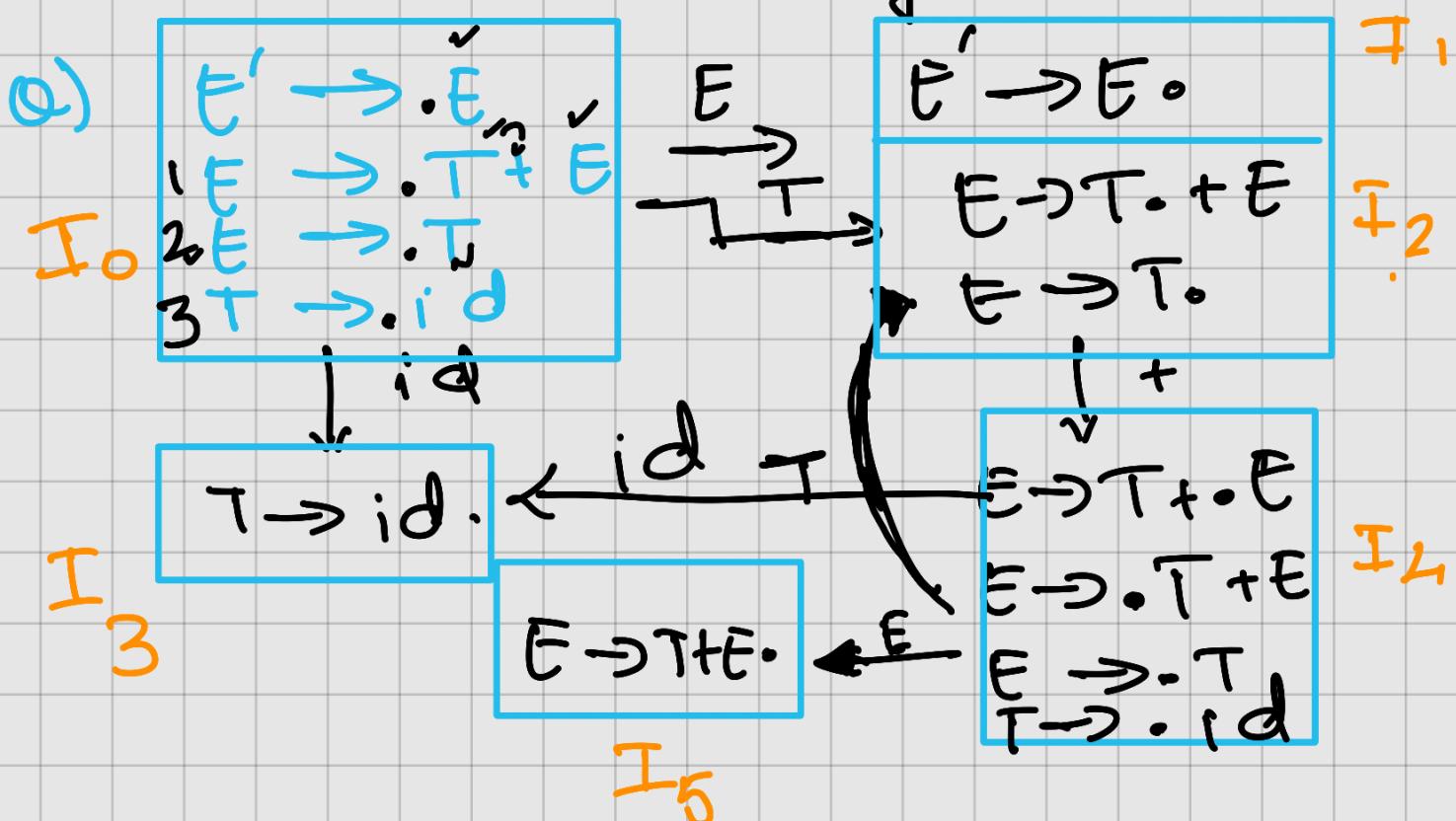


## SLR Parsers

- SLR(1) → look ahead of 1
  - ↳ Simple
  - ↳ left to right
  - ↳ reverse rightmost
- The only difference is where reduce symbols are "put".

In SLR(1) "Y" are put only in look aheads(NT) of NT.



$$\text{first}(E') = \text{first}(E) \cup \{id\}$$

$$\text{first}(E) = \text{first}(T) = \{id\}$$

$$\text{first}(T) = \{id\}$$

follow(E') = { \$ }

follow(E) = follow(E') = { \$ }

follow(T) = follow(E) ∪ { + }  
= { \$, + }

Give table:

	id	+	\$	E	T
0	3			1	2
1			accept		
2		4	$r_2$		
3		$r_3$	$r_3$		
4	3			5	2
5			$r_1$		

⇒ blanks in the table are pause errors & lead to reject!

- We now reduce only when input is a member of the follow!
- The preceding of string remaining the same: w = id + id.

\$0 id + id \$  
 \$0 id 3 + id \$  $\stackrel{r_3 \Rightarrow T \rightarrow id}{\hookrightarrow}$   
 \$0 T 2 + id \$  
 \$0 T 2 + 4 id \$  $\stackrel{r_3 \Rightarrow T \rightarrow id}{\hookrightarrow}$ .  
 \$0 T 2 + 4 id 3  
 \$0 T 2 + 4 T 2 \$  $\stackrel{r_2 \Rightarrow E \rightarrow T}{\hookrightarrow}$   
 \$0 T 2 + 4 E 5 \$  $\stackrel{r_1 \Rightarrow E \rightarrow T + E}{\hookrightarrow}$   
 \$0 E 1 \$  $\rightarrow$  Accept!

- Conflicts in SLR

Conflicts are possible in  
 SLR parser

### a) Left recursion

$E \rightarrow E + T \mid T$   
 $T \rightarrow * F \mid F$   
 $F \rightarrow id \mid num \mid (E)$

$E' \rightarrow TE' \mid \lambda$        $T' \rightarrow * FT' \mid \lambda$   
 $E' \rightarrow TT' \mid \lambda$        $F \rightarrow id \mid num \dots$   
 $T \rightarrow FT'$

Q) LL(0) for:  $w = (a, a)$   
 $S \Rightarrow a^L (L)$   $\Rightarrow S \Rightarrow a^L (L)$   
 $L \Rightarrow L, S^1 S$   $\Rightarrow L \Rightarrow L, S^1 a^L (L)$

$S \Rightarrow a^L (L)$   
 $L \Rightarrow a^{L'} | (L) L'$   
 $L' \Rightarrow , S^{L'} | \lambda$

$$\text{first}(S) = \{a, c\}$$

$$\text{first}(L) = \text{first}(S) = \{a, c\}$$

$QS \not\in \text{first}(S)$

don't need to

$$\text{first}(L') = \{\lambda\} \quad \text{consider } L'$$

$$\begin{aligned} \text{follow}(S) &= \{\$\} \cup \text{first}(L') - \lambda \\ &\cup \text{follow}(L') \\ &\cup \text{follow}(L') \end{aligned}$$

$= \{\$, a, c\}$

$$\text{follow}(L) = \{\lambda\}$$

$$\text{follow}(L') = \text{follow}(L) = \{\lambda\}$$

Table:

a	c	)	,	\$
S $\rightarrow$ a	$S \rightarrow (L)$	;	;	;
L $\rightarrow$ aL'	$L \rightarrow (L)L'$	;	;	;
L'		$L \rightarrow \lambda$	$L' \rightarrow SL'$	

S \$	(a, a) \$			
(L) \$	(a, a) \$			for LL(1)
L \$	(a, a) \$			
q(L') \$	(a, a) \$			Every entry in
L' \$	,	(a) \$		the table will
SL' \$	x (a) \$			Only have 1
SL' \$	a) \$			entry.
qL' \$	(a) \$			
L' \$	.) \$			
x \$	x \$			Accept!

error 'sync' are put in follow(v)

{  
     $\lambda \notin \text{first}(v)$

if  $M[Nt, i] = \text{Sync}$

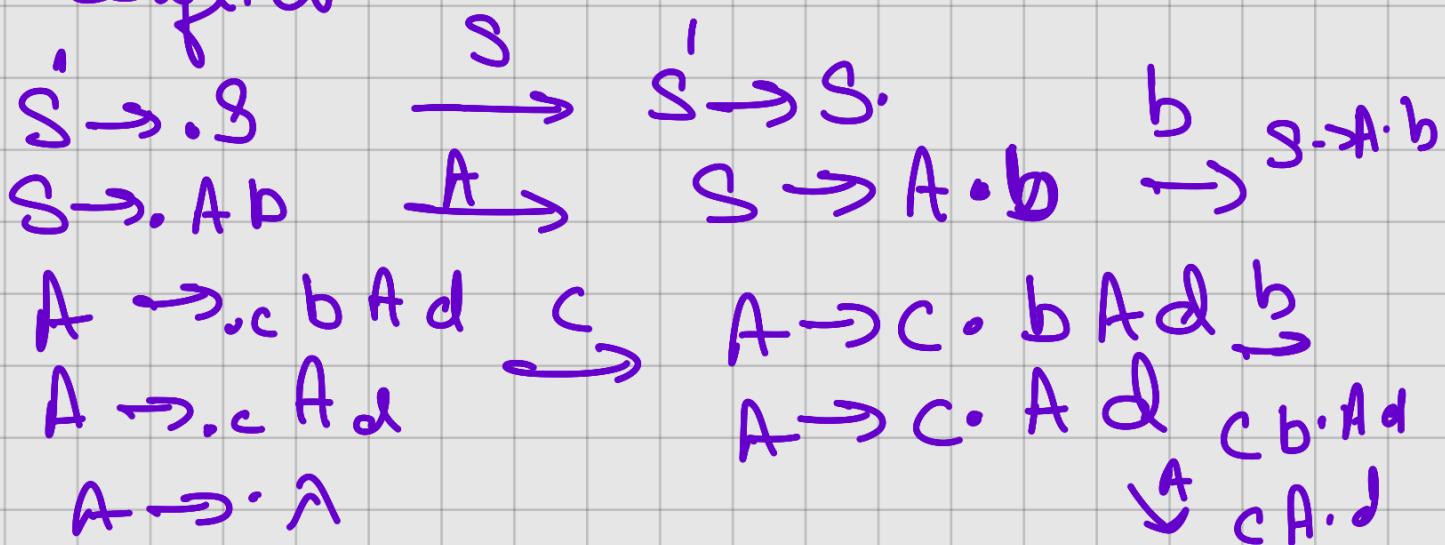
:]  
     $Nt$  the only symbol on

Stack top  $\epsilon$  from input

if not, pop  $Nt$  from stack &  
keep input as is

• for LR(0) search for Sync or r/v  
conflicts in the augmented grammar

(1) find shortest RHS to introduce SR  
conflict.



$S' \rightarrow S$   
 $S \rightarrow L = R$   
 $S \rightarrow R$   
 $L \rightarrow \text{or } R$   
 $L \rightarrow \text{id}$   
 naive:  $R \rightarrow L$   
 $\text{org}: R \rightarrow L$

$\text{first}(R) = \text{first}(L)$   
 $\text{first}(L) = \{^*, \text{id}\}$

