

SLR Parsing

- (1) $S' \rightarrow S$
- (2) $S \rightarrow bX * S$
- (3) $S \rightarrow b$
- (4) $X \rightarrow + bX$
- (5) $X \rightarrow + b$

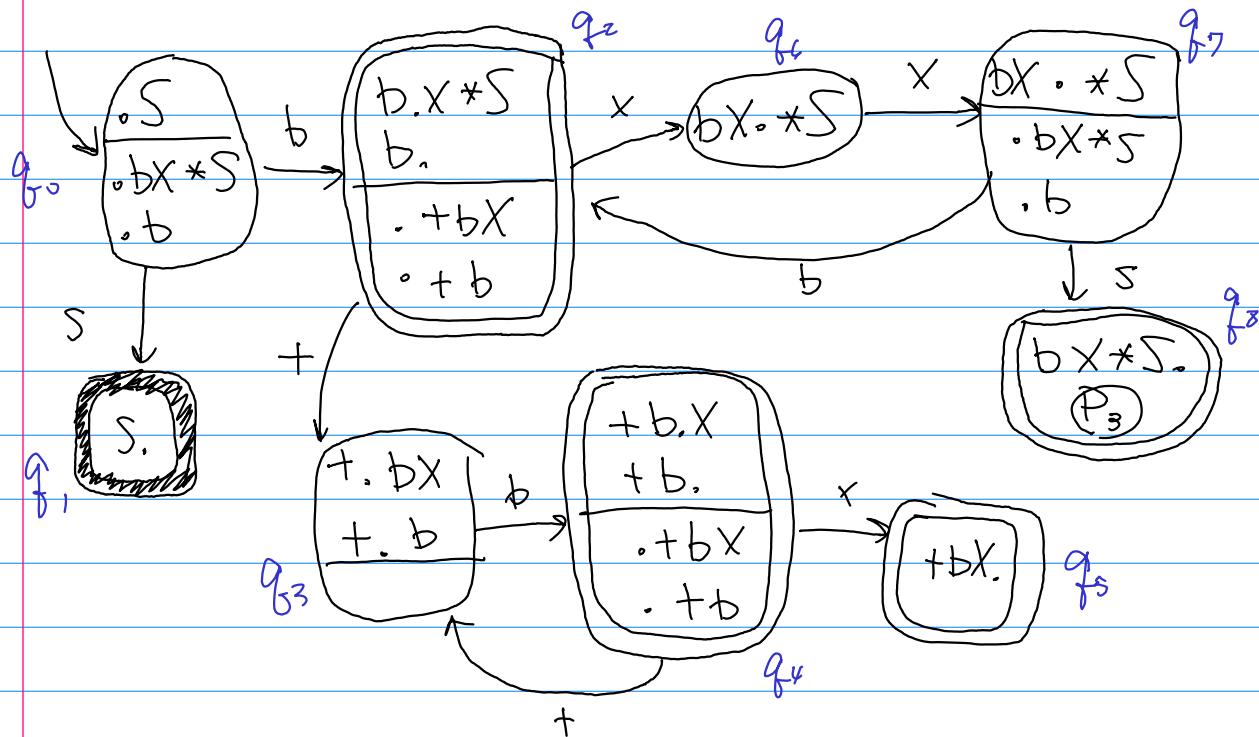
① Augment $S' \rightarrow S$

② LR(0) Items \Rightarrow

$.S$	$.bX * S$	$.+bX$	$.+b$
$S.$	$b.X * S$	$+.bX$	$+.b$
$bX.*S$		$+b.X$	$+b.$
$bX*.S$		$+bX.$	
$bX*S$			

Informally, the "dot" marks the position of parser in a parse.

③ Construct Automata



④ Parsing Algorithm

④ Parsing Algorithm

Action : Shift to state q_K

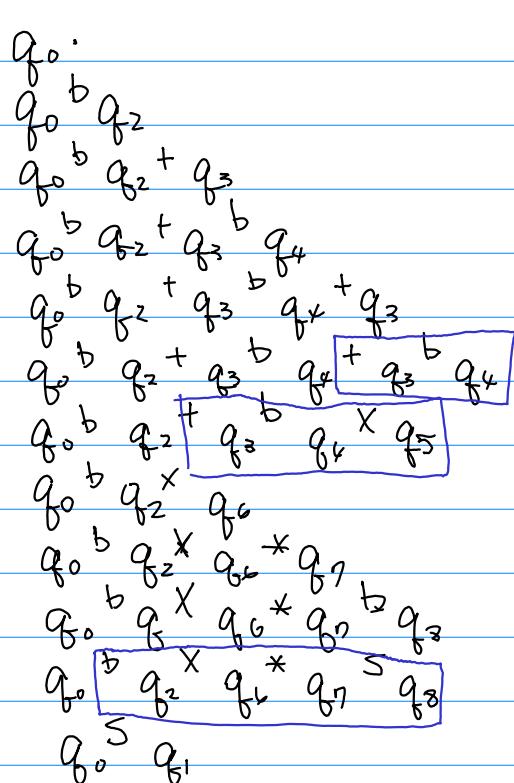
Reduce by production P

Accept

Error

Start: push q₀

Example : $b + b + b * b$



$b + b + b * b ^ 3$
 $+ b + b * b ^ 3$
 $b + b * b ^ 3$
 $+ b * b ^ 3$
 $b * b ^ 3$
 $* b ^ 3$
 $* b ^ 3$
 $b ^ 3$
 $b ^ 3$

shift q_0
shift q_2
shift q_x
shift q_3
shift q_4
reduce P5
reduce P4
shift q_7

bottom-up derivation (rightmost)

$$b + b + b * b$$

$$b + bX \square + b \checkmark$$

$$b^{\diamond} + b + b * T$$

$b \times \square * b$

$$d + \diamond d + d \neq d$$

b x * □ b

b + b[◇] + b * b

b x * b \diamond ✓

$$b + \varphi + \square b * b$$

bx * s ✓

$$\checkmark b + b + b \diamond * b$$

✓

$$\checkmark b + b + b \square * b$$

S ✓