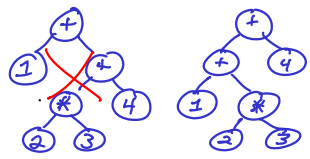


# LR Parsing

left-to-right consumption  
left-to-right construction (top-down)

# LR Parsing

bottom-up construction

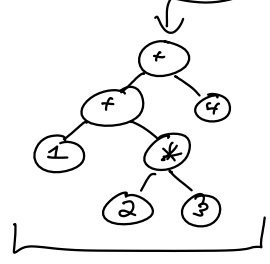


"shift" b/c  $prec(*) > prec(+)$   
 "reduce" b/c  $prec(+)<prec(*)$   
 "reduce" b/c  $prec(+)=prec(+)$  and + is left assoc.

input (tokens)	1+2*3+4	+2*3+4	+2*3+4	2*3+4	*3+4	*3+4	3+4	+4	+4	+4	+4	4	4	4	4
stack (tokens and ASTs)		1	(1)	(1) +	(1) + 2	(1) + (2)	(1) + (2) *	(1) + (2) * 3	(1) + (2) * 3 +	(1) + (2) * 3 + 4	(1) + (2) * 3 + 4 +	(1) + (2) * 3 + 4 + 4	(1) + (2) * 3 + 4 + 4	(1) + (2) * 3 + 4 + 4	(1) + (2) * 3 + 4 + 4

$\langle expr \rangle ::= \langle expr \rangle + \langle expr \rangle$   
 $| \langle expr \rangle * \langle expr \rangle$   
 $| \epsilon$

+ is left assoc  
 \* is left assoc  
 prec: +, \*



yet another compiler compiler



(is a)  
 Domain-specific language (DSL)