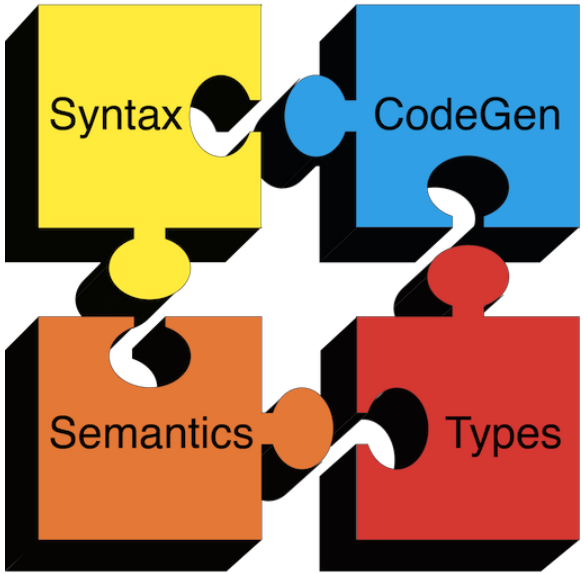


Compilers

LL(1) Parsing Tables

LL(1) Parsing Tables

- Construct a parsing table T for CFG G

- For each production $A \rightarrow \alpha$ in G do:
  - For each terminal $t \in First(\alpha)$ do
    - $T[A, t] = \alpha$
  - If $\varepsilon \in First(\alpha)$, for each $t \in Follow(A)$ do
    - $T[A, t] = \alpha$
  - If $\varepsilon \in First(\alpha)$ and $\$ \in Follow(A)$ do
    - $T[A, \$] = \alpha$

LL(1) Parsing Tables

$E \rightarrow T\ X$

$T \rightarrow (\ E\ ) \mid \text{int}\ Y$

$X \rightarrow +\ E \mid \varepsilon$

$Y \rightarrow *\ T \mid \varepsilon$

Follow(X) = { $,) }

Follow(Y) = { +, $,) }

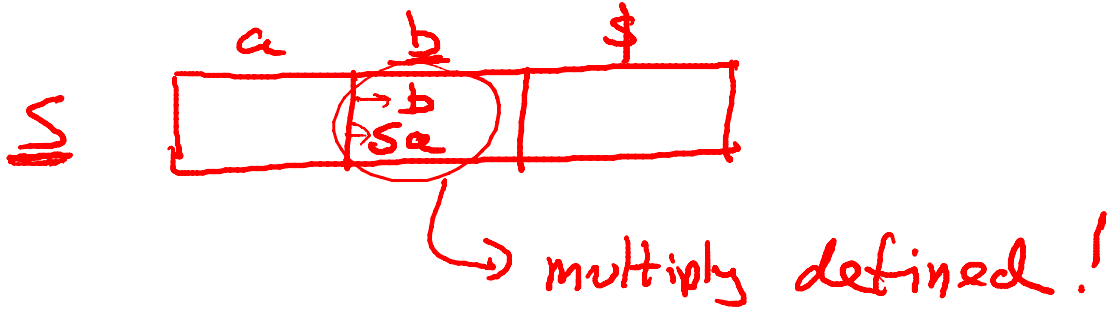
	(	)	+	*	int	$
E	TX				TX	
T	(E)				int Y	
X	error	ε	+E			ε
Y		ε	ε	*T		ε

LL(1) Parsing Tables

$S \rightarrow S\,a \mid b$

First(S) = {b}

Follow(S) = {$, a}



LL(1) Parsing Tables

- If any entry is multiply defined then G is not LL(1)
  - not left factored
  - left recursive
  - ambiguous
  - other grammars are not LL(1)
- Most programming language CFGs are not LL(1)