CSE443
Compilers

Dr. Carl Alphonce
alphonce@buffalo.edu
343 Davis Hall

Phases of a compiler

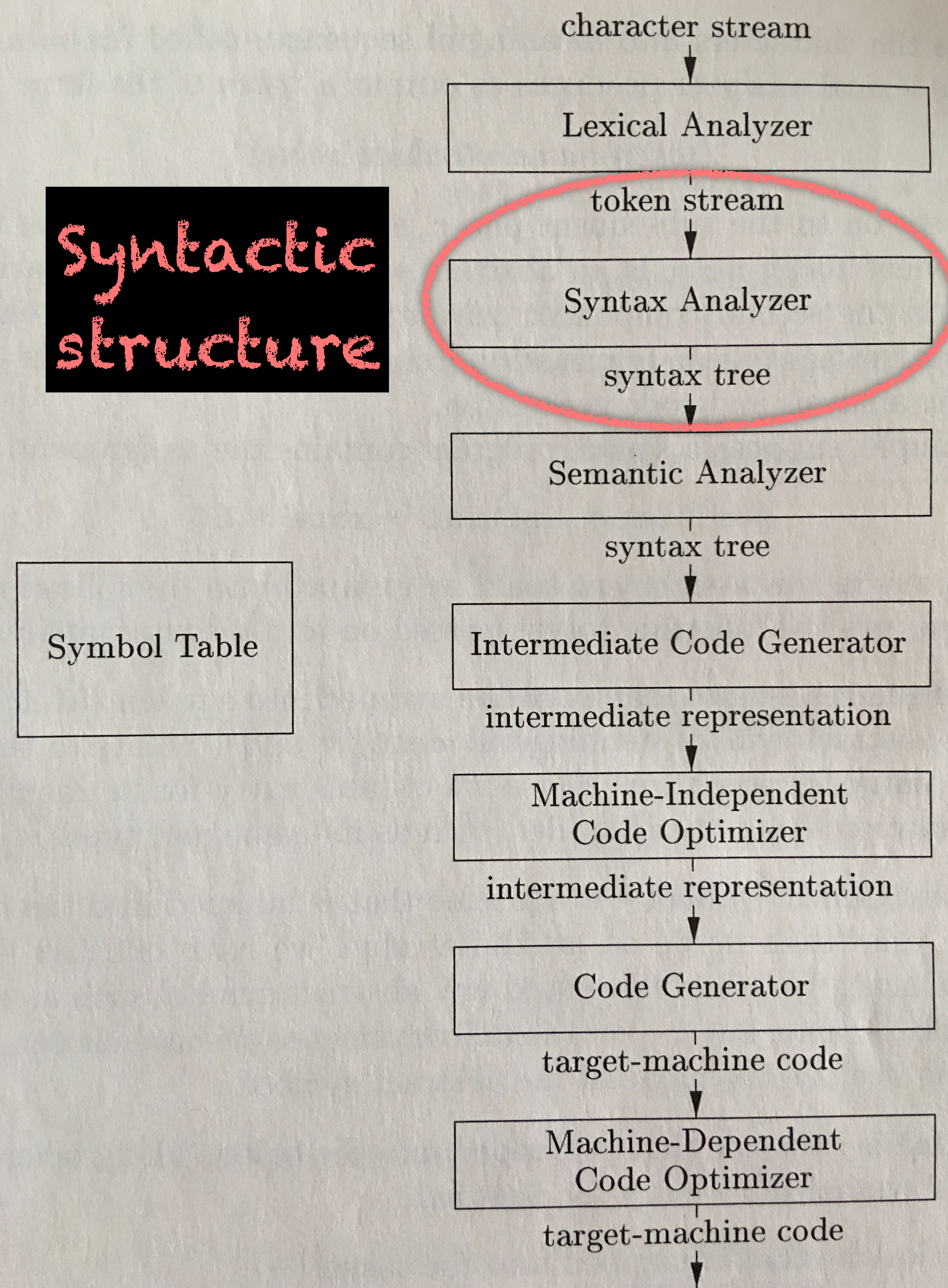

Figure 1.6, page 5 of text

FIRST(X)

- if $X \in T$ then $FIRST(X) = \{ X \}$
- if $X \in N$ and $X \to Y_1\ Y_2 \ldots Y_k \in P$ for $k \geq 1$, then
  - add $a \in T$ to FIRST(X) if $\exists i$ s.t. $a \in FIRST(Y_i)$ and $\varepsilon \in FIRST(Y_j)\ \forall\ j < i$ (i.e. $Y_1\ Y_2 \ldots Y_{i-1} \Rightarrow^* \varepsilon$)
  - if $\varepsilon \in FIRST(Y_j)\ \forall\ j \leq k$ add ε to FIRST(X)
  - if $X \to \varepsilon \in P$, then add ε to FIRST(X)

Example

E -> T E'

E' -> + T E' | ε

T -> F T'

T' -> * F T' | ε

F -> (E) | id

- if X ∈ T then FIRST(X) = { X }
- if X ∈ N and X -> Y_1 Y_2 ... Y_k ∈ P for k≥1, then
  - add a ∈ T to FIRST(X) if ∃i s.t. a ∈ FIRST(Y_i) and ε ∈ FIRST(Y_j) ∀ j < i (i.e. Y_1 Y_2 ... Y_k ⇒* ε)
  - if ε ∈ FIRST(Y_j) ∀ j < k add ε to FIRST(X)
  - if X -> ε ∈ P, then add ε to FIRST(X)

FIRST SETS

E -> T E'
E' -> + T E' | ε
T -> F T'
T' -> * F T' | ε
F -> (E) | id

- FIRST(F) = { (, id }
- FIRST(T) = FIRST(F) = { (, id }
- FIRST(E) = FIRST(T) = FIRST(F) = { (, id }
- FIRST(E') = { + , ε }
- FIRST(T') = { * , ε }

- if $X \in T$ then FIRST(X) = { X }
- if $X \in N$ and $X \rightarrow Y_1\, Y_2 \ldots Y_k \in P$ for $k \geq 1$, then
  - add $a \in T$ to FIRST(X) if $\exists i$ s.t. $a \in \text{FIRST}(Y_i)$ and $\varepsilon \in \text{FIRST}(Y_j)\ \forall\, j < i$ (i.e. $Y_1\, Y_2 \ldots Y_j \Rightarrow^* \varepsilon$)
  - if $\varepsilon \in \text{FIRST}(Y_j)\ \forall\, j < k$ add ε to FIRST(X)

FOLLOW(X)

- Place \$ in FOLLOW(S), where S is the start symbol (\$ is an end marker)
- if $A \rightarrow \alpha B \beta \in P$, then FIRST($\beta$) - $\{\varepsilon\}$ is in FOLLOW(B)
- if $A \rightarrow \alpha B \in P$ or $A \rightarrow \alpha B \beta \in P$ where $\varepsilon \in$ FIRST(β), then everything in FOLLOW(A) is in FOLLOW(B)

Example

E -> T E'

E' -> + T E' | ε

T -> F T'

T' -> * F T' | ε

F -> (E) | id

- Place $ in FOLLOW(S), where S is the start symbol ($ is an end marker)
- if A -> $\alpha B \beta \in P$, then FIRST(β) – {ε} is in FOLLOW(B)
- if A -> $\alpha B \in P$ or A -> $\alpha B \beta \in P$ where $\varepsilon \in$ FIRST(β), then everything in FOLLOW(A) is in FOLLOW(B)

FOLLOW SETS

$E \to T E'$
$E' \to + T E' \mid \varepsilon$
$T \to F T'$
$T' \to * F T' \mid \varepsilon$
$F \to (E) \mid id$

- FOLLOW(E) = {) , $ }
- FOLLOW(E') = FOLLOW(E) = {) , $ }
- FOLLOW(T) = { + ,) , $ }
- FOLLOW(T') = FOLLOW(T) = { + ,) , $ }
- FOLLOW(F) = { + , * ,) , $ }

- Place $ in FOLLOW(S), where S is the start symbol ($ is an end marker)
- if $A \to \alpha B \beta \in P$, then $FIRST(\beta) - \{\varepsilon\}$ is in FOLLOW(B)
- if $A \to \alpha B \in P$ or $A \to \alpha B \beta \in P$ where $\varepsilon \in FIRST(\beta)$, then everything in FOLLOW(A) is in FOLLOW(B)

Table-driven predictive parsing Algorithm 4.32 (p. 224)

- INPUT: Grammar G = (N,T,P,S)
- OUTPUT: Parsing table M
- For each production A -> α of G:

1. For each terminal a $\in$ FIRST(α), add A -> α to M[A,a]
2. If $\varepsilon \in$ FIRST(α), then for each terminal b in FOLLOW(A), add A -> α to M[A,b]
3. If $\varepsilon \in$ FIRST(α) and \$ $\in$ FOLLOW(A), add A -> α to M[A,\$]

FIRST SETS

- FIRST(F) = { (, id }
- FIRST(T) = FIRST(F) = { (, id }
- FIRST(E) = FIRST(T) = FIRST(F) = { (, id }
- FIRST(E') = { + , ε }
- FIRST(T') = { * , ε }

E -> T E'
E' -> + T E' | ε
T -> F T'
T' -> * F T' | ε
F -> (E) | id

- if $X \in T$ then FIRST(X) = { X }
- if $X \in N$ and $X \rightarrow Y_1 Y_2 \ldots Y_k \in P$ for $k \geq 1$, then
  - add $a \in T$ to FIRST(X) if $\exists i$ s.t. $a \in$ FIRST(Y_i) and $\varepsilon \in$ FIRST(Y_j) $\forall j < i$ (i.e. $Y_1 Y_2 \ldots Y_j \Rightarrow^* \varepsilon$)
  - if $\varepsilon \in$ FIRST(Y_j) $\forall j < k$ add ε to FIRST(X)

For each production A -> α of G:

- For each terminal $a \in$ FIRST(α), add A -> α to M[A,a]
- If $\varepsilon \in$ FIRST(α), then for each terminal b in FOLLOW(A), add A -> α to M[A,b]
- If $\varepsilon \in$ FIRST(α) and $\$ \in$ FOLLOW(A), add A -> α to M[A,$]

FOLLOW SETS

E -> T E'
E' -> + T E' | ε
T -> F T'
T' -> * F T' | ε
F -> (E) | id

- FOLLOW(E) = {) , $ }
- FOLLOW(E') = FOLLOW(E) = {) , $ }
- FOLLOW(T) = { + ,) , $ }
- FOLLOW(T') = FOLLOW(T) = { + ,) , $ }
- FOLLOW(F) = { + , * ,) , $ }

- Place $ in FOLLOW(S), where S is the start symbol ($ is an end marker)
- if A -> $\alpha B \beta \in$ P, then FIRST(β) – {ε} is in FOLLOW(B)
- if A -> $\alpha B \in$ P or A -> $\alpha B \beta \in$ P where $\varepsilon \in$ FIRST(β), then everything in FOLLOW(A) is in FOLLOW(B)

For each production A -> α of G:

- For each terminal a $\in$ FIRST(α), add A -> α to M[A,a]
- If $\varepsilon \in$ FIRST(α), then for each terminal b in FOLLOW(A), add A -> α to M[A,b]
- If $\varepsilon \in$ FIRST(α) and $ $\in$ FOLLOW(A), add A -> α to M[A,$]

EXERCISE:
fill in the parse table

(see next slide)

Parse-table M

NON TERMINALS	id	+	*	(	)	$
E						
E'						
T						
T'						
F						

FIRST(E) = FIRST(T) = FIRST(F) = { (, id }
FIRST(E') = { + , ε }
FIRST(T') = { * , ε }

FOLLOW(E') = FOLLOW(E) = {) , $ }
FOLLOW(T') = FOLLOW(T) = { + ,) , $ }
FOLLOW(F) = { + , * ,) , $ }

E -> T E'
E' -> + T E' | ε
T -> F T'
T' -> * F T' | ε
F -> (E) | id

For each production A -> α of G:

- For each terminal a ∈ FIRST(α), add A -> α to M[A,a]
- If ε ∈ FIRST(α), then for each terminal b in FOLLOW(A), add A -> α to M[A,b]
- If ε ∈ FIRST(α) and $ ∈ FOLLOW(A), add A -> α to M[A,$]

Parse-table M

NON TERMINALS	id	+	*	(	)	$
E	E -> T E'			E -> T E'		
E'						
T						
T'						
F						

FIRST(E) = FIRST(T) = FIRST(F) = { (, id }

FIRST(E') = { + , ε }

FIRST(T') = { * , ε }

FOLLOW(E') = FOLLOW(E) = {) , $ }

FOLLOW(T') = FOLLOW(T) = { + ,) , $ }

FOLLOW(F) = { + , * ,) , $ }

E -> T E'

E' -> + T E' | ε

T -> F T'

T' -> * F T' | ε

F -> (E) | id

For each production A -> α of G:

- For each terminal a ∈ FIRST(α), add A -> α to M[A,a]
- If ε ∈ FIRST(α), then for each terminal b in FOLLOW(A), add A -> α to M[A,b]
- If ε ∈ FIRST(α) and $ ∈ FOLLOW(A), add A -> α to M[A,$]