

Análise Sintática

(Cap. 04)

Análise Sintática Ascendente - *Bottom-Up*

# Análise Sintática

- Análise sintática ascendente
  - Constrói a árvore de derivação de baixo para cima, das folhas para raíz
  - Produz uma derivação mais à direita para uma cadeia de entrada

# Análise Sintática

- Análise Sintática Ascendente
  - Processo de redução de uma string  $w$  para o símbolo inicial da gramática
  - Em cada passo da redução, uma substring é associada com o corpo de uma produção e obtém-se a “cabeça” dessa produção
  - Decisões chave: quando reduzir e quais produções utilizar
  - Por definição, redução é o inverso da derivação.
    - Obs: bottom-up parsing constrói uma derivação em modo reverso.

# Análise Sintática

- Análise Sintática Ascendente - Poda do Handle
  - Handle é uma subcadeia que “casa” com o corpo de uma produção e cuja redução representa um passo da derivação mais à direita ao inverso.
  - Uma derivação mais à direita em ordem reversa pode ser obtida pela “poda do handle”
    - Iniciar com uma string de terminais  $w$ . Se  $w$  é uma sentença da gramática dada, então  $w=y_n$  onde  $y_n$  é a  $n$ -ésima forma sentencial mais à direita de alguma derivação mais à direita desconhecida
    - Para reconstruir a derivação, localizar handle  $B_n$  em  $y_n$  e substituir  $B_n$  pelo lado esquerdo da produção  $A_n \rightarrow B_n$  para obter a forma sentencial anterior  $y_{n-1}$
    - Repetimos esse processo até alcançar o símbolo inicial da gramática, nesse caso, temos um sucesso. Caso contrário, um mensagem de erro deve ser gerada

# Análise Sintática

- Analisadores shift-reduce
  - Forma de análise ascendente onde uma pilha contém símbolos da gramática e um buffer de entrada contém o restante da cadeia a ser reconhecida sintaticamente.
  - Usa-se o símbolo \$ para marcar o fim da pilha e o último caracter da entrada
    - Obs: Nos exemplos a seguir, topo da pilha será apresentado à direita. Inicialmente, a pilha está vazia e a cadeia  $w$  representa a entrada

**PILHA**

\$

**ENTRADA**

$w$ \$

# Análise Sintática

- Analisadores shift-reduce
  - Durante o parsing (da esquerda para direita), o analisador desloca (shift) zero ou mais símbolos de entrada na pilha, até que está pronto para reduzir (reduce) uma string B (no topo da pilha) para a “cabeça” da produção apropriada
  - O analisador sintático repete esse ciclo até encontrar um erro (nenhuma produção é encontrada) ou até que a pilha contém o símbolo de início e a entrada está vazia

# Análise Sintática

- Analisadores shift-reduce - Exemplo
  - Considere a string  $w=id*id$  e a gramática:
    - $E \rightarrow E + T \mid T$
    - $T \rightarrow T * F \mid F$
    - $F \rightarrow (E) \mid id$

# Análise Sintática

<b>• PILHA</b>	<b>ENTRADA</b>	<b>AÇÃO</b>
\$	id1*id2\$	deslocar
\$id1	*id2\$	reduzir para $F \rightarrow id$
\$F	*id2\$	reduzir para $T \rightarrow F$
\$T	*id2\$	deslocar
\$T*	id2\$	deslocar
\$T*id2	\$	reduzir para $F \rightarrow id$
\$T*F	\$	reduzir para $T \rightarrow T*F$
\$T	\$	reduzir para $E \rightarrow T$
\$E	\$	aceitar



# Análise Sintática

- Observe que:
  - Deslocar (shift): deslocar o próximo símbolo da entrada no topo da pilha
  - Reduzir (reduce): a parte (mais à direita) da string a ser reduzida está no topo da pilha. Decidir com qual não-terminal substituir a string
  - Aceitar: analisador sintático finalizou com sucesso
  - Error: erro sintático

# Análise Sintática

- Conflitos durante execução do analisador shift-reduce
  - Quando o analisador não consegue decidir se deve realizar um shift ou um reduce, temos um conflito shift/reduce
  - Quando o analisador não consegue determinar qual redução fazer, temos um conflito reduce/reduce

# Análise Sintática

- Exemplo:
  - Dada a gramática:
    - s- $\rightarrow$  if E then S | if E then S else S | other
  - E a configuração:

**PILHA**

...if E then S

**ENTRADA**

else .... \$

# Análise Sintática

- Exemplo:

- Dada a gramática:

- $s \rightarrow id ( \text{parametros} )$

- $s \rightarrow e := e$

- $\text{parametros} \rightarrow \text{parametros}, \text{parametro}$

- $\text{parametros} \rightarrow \text{parametro}$

- $\text{parametro} \rightarrow id$

- $e \rightarrow id ( \text{lista\_e} )$

- $e \rightarrow id$

- E a configuração:

**PILHA**

... id ( id

**ENTRADA**

, id ) ...