

Compiladores

Análise Sintática

Análise Sintática Ascendente

Profa. Dra. Ana Carolina Lorena
UNIFESP

Análise Sintática Ascendente

- Também chamada de **Análise Redutiva**
 - ▮ Tenta construir árvore de derivação a partir das folhas, produzindo derivação mais à direita ao reverso
 - ▮ Parte-se dos símbolos terminais, em direção ao símbolo inicial da gramática

Análise Sintática Ascendente

- Visão Geral:
 - Um analisador sintático ascendente usa uma pilha para efetuar a análise
 - A pilha conterá símbolos terminais, não-terminais e estados de um AFD
 - A pilha está vazia no início da análise e conterá o símbolo inicial ao término de uma análise bem sucedida
 - Um AFD define os estados de um analisador sintático ascendente, e é utilizado para acompanhar os estados do analisador sintático

Análise Sintática Ascendente

- Visão Geral:
 - O analisador ascendente tem duas ações possíveis (além da ação “aceita” ao término da análise bem sucedida):
 - **Empilha** um terminal da cadeia de entrada para o topo da pilha;
 - **Reduz** uma cadeia α do topo da pilha para um símbolo não-terminal A , dada a escolha BNF $A \rightarrow \alpha$
 - Operação de substituição do lado direito de uma produção pelo não-terminal correspondente no lado esquerdo

São chamados de analisadores **empilha-reduz**

Análise Sintática Ascendente

- Visão Geral:
 - Uma característica adicional dos analisadores ascendentes é a gramática aumentada
 - Necessária para garantir um único estado inicial do autômato finito (com pilha)
 - Os analisadores ascendentes são mais poderosos que os descendentes

Análise Sintática Ascendente

- Visão Geral:

- **Início**: fita de entrada contém a sequência a ser analisada seguida de \$ e a pilha contém apenas \$

- **Reconhecimento**: transfere símbolos da fita de entrada para a pilha até que se tenha na pilha um lado direito de produção

- Esse lado é então substituído (reduzido) pelo símbolo do lado esquerdo da regra

- Processo segue até que a entrada seja toda lida

- Se a sentença for válida, a pilha será reduzida ao símbolo inicial

Análise Sintática Ascendente

- Exemplo: considere a gramática para operações de adição:

$$E \rightarrow E + n \mid n$$

Para a entrada $n + n$, as ações do analisador são:

| Pilha | Entrada | Ação |
|-----------|------------|-----------------------------|
| \$ | $n + n \$$ | empilha |
| $\$n$ | $+ n \$$ | reduz $E \rightarrow n$ |
| $\$E$ | $+ n \$$ | empilha |
| $\$E +$ | $n \$$ | empilha |
| $\$E + n$ | $\$$ | reduz $E \rightarrow E + n$ |
| $\$E$ | $\$$ | aceita |

Observar que linhas 3 e 6 contêm ações diferentes com E (porque entradas são diferentes)

Análise Sintática Ascendente

- Exemplo: considere a gramática aumentada para operações de adição:

$$E' \rightarrow E$$

$$E \rightarrow E + n \mid n$$

Para a entrada $n + n$, as ações do analisador são:

| Pilha | Entrada | Ação |
|-----------|------------|-----------------------------|
| \$ | $n + n \$$ | empilha |
| $\$n$ | $+ n \$$ | reduz $E \rightarrow n$ |
| $\$E$ | $+ n \$$ | empilha |
| $\$E +$ | $n \$$ | empilha |
| $\$E + n$ | $\$$ | reduz $E \rightarrow E + n$ |
| $\$E$ | $\$$ | aceita |

Analisador acompanha uma derivação à direita da cadeia de entrada, mas os passos de derivação ocorrem na ordem inversa:

$$E' \Rightarrow E \Rightarrow E + n \Rightarrow n + n$$

Análise Sintática Ascendente

- Exemplo: considere a gramática para parênteses balanceados:

$$S \rightarrow (S)S \mid \varepsilon$$

Para a entrada $()$, as ações do analisador são:

| Pilha | Entrada | Ação |
|------------|---------|-----------------------------------|
| \$ | () \$ | empilha |
| \$ (|) \$ | reduz $S \rightarrow \varepsilon$ |
| \$ (S |) \$ | empilha |
| \$ (S) | \$ | reduz $S \rightarrow \varepsilon$ |
| \$ (S) S | \$ | reduz $S \rightarrow (S) S$ |
| \$ S | \$ | aceita |

Análise Sintática Ascendente

- Exemplo: considere a gramática aumentada para parênteses balanceados:

$$S' \rightarrow S$$

$$S \rightarrow (S)S \mid \varepsilon$$

Para a entrada $()$, as ações do analisador são:

| Pilha | Entrada | Ação |
|------------|---------|-----------------------------------|
| \$ | () \$ | empilha |
| \$ (|) \$ | reduz $S \rightarrow \varepsilon$ |
| \$ (S |) \$ | empilha |
| \$ (S) | \$ | reduz $S \rightarrow \varepsilon$ |
| \$ (S) S | \$ | reduz $S \rightarrow (S) S$ |
| \$ S | \$ | aceita |

Analisador acompanha uma derivação à direita da cadeia de entrada, mas os passos de derivação ocorrem na ordem inversa:

$$S' \Rightarrow S \Rightarrow (S)S \Rightarrow (S) \Rightarrow ()$$

Análise Sintática Ascendente

- O analisador empilha símbolos da entrada até ter na pilha uma sequência de símbolos que corresponde à definição de algum não-terminal
 - Essa sequência define o **handle**
- Redução consiste em substituir o handle na pilha pelo não-terminal que o corresponde
- Uso da sequência correta de handles deve levar ao símbolo inicial da gramática

Análise Sintática Ascendente

- Os **handles** são as sequências de símbolos que são lados direitos de produção, tais que suas reduções levam, no final, ao símbolo inicial da gramática
 - ▮ Pelo reverso de uma derivação mais à direita
 - ▮ Se uma gramática é não-ambígua, então toda forma sentencial gerada por ela tem exatamente um handle

Análise Sintática Ascendente

- Assim, as ações de um reconhecedor empilha-reduz são:
 - ▮ **Empilha**: coloca no topo da pilha o símbolo que está sendo lido (e estado de AFD) e avança a leitura
 - ▮ **Reduz**: substitui o handle do topo da pilha pelo não-terminal correspondente
 - ▮ **Aceita**: reconhece a sentença de entrada
 - ▮ **Erro**: acusa erro

Análise Sintática Ascendente

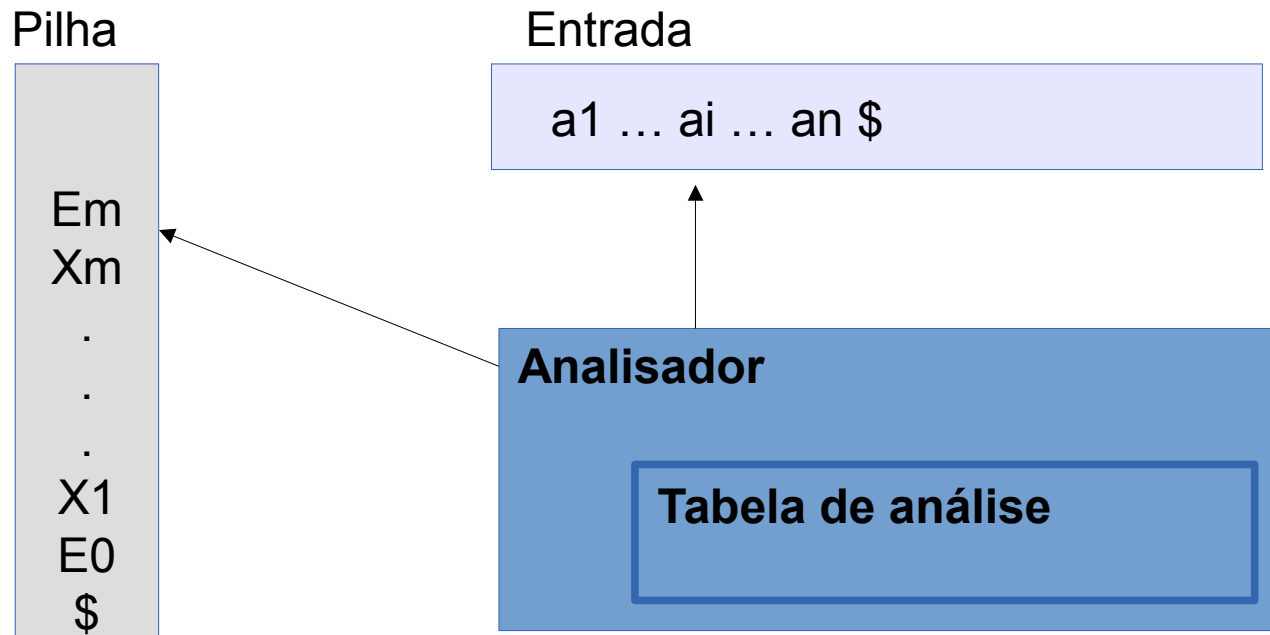
- Há alguns tipos de analisadores ascendentes:
 - ▮ Mais geral: LR(1)
 - ▮ L: entrada é processada da esquerda para a direita
 - ▮ R: derivação à direita é produzida
 - ▮ 1: um símbolo de verificação à frente é usado
 - ▮ LR(0): sem verificação à frente
 - ▮ Marca é examinada após ter aparecido na pilha
 - ▮ SLR(1): LR(1) simples
 - ▮ Usa algumas estruturas da LR(0)
 - ▮ LALR(1): *Look Ahead* LR(1)
 - ▮ Simplificação do LR(1) geral

Análise Sintática Ascendente

- **Vantagem** analisadores LR(k):
 - São capazes de reconhecer praticamente todas as estruturas sintáticas definidas por GLC
 - São capazes de descobrir erros mais cedo
- **Desvantagem** analisadores LR(k):
 - Dificuldade de implementação

Análise Sintática Ascendente

- Esquema geral analisadores LR(1):



A pilha armazena símbolos da gramática (X_i) intercalados com estados do analisador (do AFD)

Analisador é dirigido por tabela de análise

Análise Sintática Ascendente

- **Tabela de análise:** tabela de transição de estados formada por duas partes
 - **Ação:** contém ações associadas aos terminais da entrada
 - **Transição:** contém transições de estados com relação aos não-terminais

| | AÇÃO | TRANSIÇÃO |
|---------------------------------|-----------|---------------|
| | TERMINAIS | NÃO-TERMINAIS |
| E S T A D O S | empilha | estados |
| | reduz | |
| | aceita | |
| | erro | |
| | | |

Análise Sintática Ascendente

- **Funcionamento**: seja E_m o estado no topo da pilha e a_i a entrada corrente
 - Analizador consulta $Ação[E_m, a_i]$, que pode ser:
 - Empilha E_x** : empilha $a_i E_x$
 - Reduz n ($n = \text{número de produção } A \rightarrow \beta$)**: desempilha $2r$ símbolos, em que $r = |\beta|$, e o empilhamento de $A E_y$, onde E_y resulta em consulta a $Transição[E_m, A]$
 - Aceita**: analisador reconhece entrada como válida
 - Erro**: analisador identifica erro sintático

Análise Sintática Ascendente

- Exemplo
- 1) $E \rightarrow E \vee T$
- 2) $E \rightarrow T$
- 3) $T \rightarrow T \& F$
- 4) $T \rightarrow F$
- 5) $F \rightarrow (E)$
- 6) $F \rightarrow \text{id}$

| | id | v | & | (|) | \$ | E | T | F |
|----|----|----|----|----|-----|----|---|---|----|
| 0 | e5 | | | e4 | | | 1 | 2 | 3 |
| 1 | | e6 | | | | AC | | | |
| 2 | | r2 | e7 | | r2 | r2 | | | |
| 3 | | r4 | r4 | | r4 | r4 | | | |
| 4 | e5 | | | e4 | | | 8 | 2 | 3 |
| 5 | | r6 | r6 | | r6 | r6 | | | |
| 6 | e5 | | | e4 | | | | 9 | 3 |
| 7 | e5 | | | e4 | | | | | 10 |
| 8 | | e6 | | | e11 | | | | |
| 9 | | r1 | e7 | | r1 | r1 | | | |
| 10 | | r3 | r3 | | r3 | r3 | | | |
| 11 | | r5 | r5 | | r5 | r5 | | | |

Análise Sintática Ascendente

- Bibliografia consultada

LOUDEN, K. C. **Compiladores: princípios e práticas**. São Paulo: Pioneira Thompson Learning, 2004. (Cap. 5)

RICARTE, I. **Introdução à Compilação**. Rio de Janeiro: Editora Campus/Elsevier, 2008. (Cap. 4)

PRICE, A. M. A. e TOSCANI, S. S. **Implementação de Linguagens de Programação: Compiladores**. Bookman, 2008 (Cap. 3)