

– INF01147 –  
Compiladores

Análise Sintática  
Parser LR(0)

Prof. Lucas M. Schnorr  
– Universidade Federal do Rio Grande do Sul –



# Introdução à Análise Ascendente

(revisão)

# Algoritmo Empilha-Reduz – Funcionamento

- Considerando a gramática de operações aritméticas

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow ( E ) \mid id$$

- Ações Empilha-Reduz para **id\*id**

Pilha	Entrada	Ação
\$	$id_1 * id_2$	\$ empilha
\$ $id_1$	$*id_2$	\$ reduz $F \rightarrow id$
\$ $F$	$*id_2$	\$ reduz $T \rightarrow F$
\$ $T$	$*id_2$	\$ empilha
\$ $T*$	$id_2$	\$ empilha
\$ $T * id_2$		\$ reduz $F \rightarrow id$
\$ $T * F$		\$ reduz $T \rightarrow T * F$
\$ $T$		\$ reduz $E \rightarrow T$
\$ $E$		\$ aceita

# Conflitos Empilha-Reduz

- ▶ Duas situações onde ocorrem conflitos
  - ▶ Conflito **Reduz-Reduz**  $\Rightarrow$  mais de uma redução possível
  - ▶ Conflito **Empilha-Reduz**  $\Rightarrow$  gramática ambígua
- ▶ Exemplo

stmt  $\rightarrow$    **if** expr **then** stmt  
                  **if** expr **then** stmt **else** stmt  
                  **other**

- ▶ O que fazer nesta situação?

Pilha	Entrada
\$ ... <b>if</b> expr <b>then</b> stmt	<b>else</b> ... \$

# Análise Ascendente

- ▶ Funcionamento: derivação mais à direita reversa
- ▶ Permite “adiar decisões” na pilha, sendo mais sofisticado
- ▶ Principal método de análise sintática

# Plano da Aula de Hoje

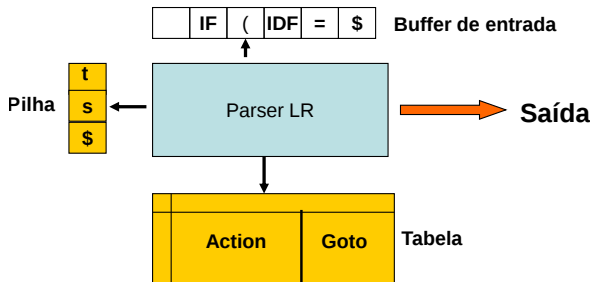
- ▶ Análise LR
- ▶ LR(0)

# Análise LR

- ▶ LR, uma classe de gramáticas onde
  - ▶ Entrada lida da esquerda para a direita
  - ▶ Aplica-se uma derivação mais à direita
- ▶ Usa um **autômato de estados finitos** com pilha
- ▶ Componentes
  - ▶ Pilha contém **estados** (ao invés de símbolos)
  - ▶ Tabela de ações/transições (com terminais e não-terminais)

L de *left-to-right*

R de *rightmost*



# Análise LR – Tabela de Ação/Transição

- ▶ Tabela de ações/transições, a partir de um estado  $s$ 
  - ▶ Ação  $[s, t]$  sendo  $t$  um terminal
  - ▶ Transição  $[s, X]$  sendo  $X$  um não-terminal
- ▶ Ação  $[s, t]$  pode indicar
  - ▶ (Empilha  $e$ ), onde  $e$  é um estado para empilhar
  - ▶ (Reduz  $p$ ), onde  $p$  é a regra de produção para a redução
  - ▶ (Aceita)
- ▶ Transição  $[s, X]$  pode indicar
  - ▶ (Empilha  $e$ ), onde  $e$  é um estado

# Análise LR – Exemplo com entrada **id\*id+id**

(1) $E \rightarrow E + T$	(3) $T \rightarrow T * F$	(5) $F \rightarrow (E)$
(2) $E \rightarrow T$	(4) $T \rightarrow F$	(6) $F \rightarrow id$

Estado	id	+	*	(	)	\$	E	T	F
0	e5			e4			1	2	3
1		e6				a			
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			

LR

# Evolução de Parsers LR

- ▶ Donald Knuth propõem parser LR(1) (em 1965) LR Canônico
  - ▶ Reconhece qualquer gramática livre de contexto determinista
  - ▶ Tempo de análise linear
- ▶ Versões simplificadas (1969) sobre LR(0)
  - ▶ LALR
  - ▶ SLR
- ▶ Diferem em como se calcula o conjunto de símbolos a frente

# LR (Motivação)

- ▶ Como construir a tabela de ações e transições?
- ▶ Como detectar um *handle*? Considerando a gramática

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow ( E ) \mid id$

- ▶ Supondo a análise de  $id * id$

Pilha	Entrada	Ação
\$	$id_1 * id_2$	\$ empilha
\$ $id_1$	$* id_2$	\$ reduz $F \rightarrow id$
\$ $F$	$* id_2$	\$ reduz $T \rightarrow F$
\$ $T$	$* id_2$	\$

- ▶ Devo reduzir  $T \rightarrow E$  ou empilhar o  $*$ ?
- ▶ Manter **estados** para saber onde se encontra na análise
  - ▶ Um estado representa um conjunto de itens

# LR(0) – Item

- ▶ Definição de Item LR(0) de uma gramática
  - ▶ É uma produção com um  $\bullet$  em alguma posição no corpo
  - ▶ Indica o quanto já foi visto de uma produção
- ▶ Supondo a produção  $A \rightarrow \beta\gamma$ , três itens possíveis
  - ▶  $A \rightarrow \bullet\beta\gamma$   
Um item que é uma **possibilidade**
  - ▶  $A \rightarrow \beta \bullet \gamma$   
(Progridimos reconhecendo  $\beta$ )  
Um item **parcialmente completo**
  - ▶  $A \rightarrow \beta\gamma \bullet$   
 $\beta\gamma$  estão empilhados  
Um item **completo**
- ▶ A produção  $A \rightarrow \epsilon$  gera um item  $A \rightarrow \bullet$

## LR(0) – Item (Exemplo)

- Considerando a gramática de parênteses balanceados

$$\begin{aligned}S' &\rightarrow S \\ S &\rightarrow ( S ) S \mid \epsilon\end{aligned}$$

- Esta gramática tem oito itens (possibilidades)

$$\begin{aligned}S' &\rightarrow \bullet S \\ S' &\rightarrow S \bullet \\ S &\rightarrow \bullet ( S ) S \\ S &\rightarrow ( \bullet S ) S \\ S &\rightarrow ( S \bullet ) S \\ S &\rightarrow ( S ) \bullet S \\ S &\rightarrow ( S ) S \bullet \\ S &\rightarrow \bullet\end{aligned}$$

# Autômato Finito para LR(0)

- ▶ Quais são as transições do autômato de itens LR(0)?
  - ▶ Consideramos o item  $A \rightarrow \alpha \bullet \gamma$
  - ▶ Consideramos que  $\gamma$  começa com  $X$ , temos então  $A \rightarrow \alpha \bullet X \eta$
- ▶ Analisando  $X$ 
  - ▶ é terminal  $\Rightarrow$  simboliza uma ação de *empilhar*
  - ▶ é não-terminal
    - ▶ Considerar todas as produções  $X \rightarrow \beta$  e os itens  $X \rightarrow \bullet \beta$

# Autômato Finito para LR(0) – Exemplo

- ▶ Considerando a gramática de parênteses balanceados

$$\begin{aligned} S' &\rightarrow S \\ S &\rightarrow ( S ) S \mid \epsilon \end{aligned}$$

- ▶ Construir o autômato a partir de  $S' \rightarrow \bullet S$
- ▶ Qual o problema deste autômato? E qual a solução?
  - ▶ Indeterminismo
  - ▶ Algoritmo de Subconjuntos
    - ▶ Fechamento- $\epsilon$ (T)
    - ▶ Fechamento- $\epsilon$ (Movimento(T,a))

# LR(0): uma reflexão

- ▶ Partimos da gramática estendida
  - ▶ Criamos o autômato finito não determinístico
  - ▶ Aplicamos o algoritmo de subconjuntos
  - ▶ Construimos a tabela de análise correspondente
- 
- ▶ Será que não há uma forma mais rápida?

# LR(0) - Autômato Finito Determinístico

- ▶ Definir uma gramática estendida
- ▶ Vamos utilizar duas funções
  - ▶ **Fechamento**  $\leadsto$  Ações da tabela LR
  - ▶ **Transição**  $\leadsto$  Transições da tabela LR
- ▶ Gramática estendida
  - ▶ Se  $S$  é o símbolo inicial de uma gramática  $G$   
Então  $G'$  é a gramática estendida, com  $S' \rightarrow S$
  - ▶ Simplifica o processo de aceitação da sentença

# Função de Fechamento

- ▶ Se  $I$  é um conjunto de itens,  $\text{Fechamento}(I)$  é
  - ▶ Todos os itens de  $I$  fazem parte do  $\text{Fechamento}(I)$
  - ▶ Se  $A \rightarrow \alpha \bullet B\beta \in \text{Fechamento}(I)$  e existe a produção  $B \rightarrow \gamma$   
Então  $B \rightarrow \bullet\gamma$  faz parte do  $\text{Fechamento}(I)$
- ▶ Exemplo, considerando a gramática já estendida

$$\begin{aligned}E' &\rightarrow E \\E &\rightarrow E + T \mid T \\T &\rightarrow T * F \mid F \\F &\rightarrow ( E ) \mid \text{id}\end{aligned}$$

- ▶ Defina o  $\text{Fechamento}(\{E' \rightarrow \bullet E\})$ 
  - ▶ Acabamos de definir o **conjunto de itens inicial**  
→ **Estado inicial** do autômato LR(0)
- ▶ **Itens de base** *versus* itens derivados

# Função de Transição

- ▶ Ela define as transições do autômato
- ▶ Transição(I, X)
  - ▶ Um conjunto de itens I
  - ▶ Símbolo X da gramática
- ▶ Retorna um conjunto de itens **já existente ou não**
- ▶ Transição(I, X) é
  - ▶ Fechamento do conjunto dos itens  
 $A \rightarrow \alpha X \bullet \beta$  tais que  $A \rightarrow \alpha \bullet X \beta$  está em I
- ▶ Informalmente
  - ▶ Mover  $\bullet$  para a direita nos itens de I onde  $\bullet$  precede X  
 $A \rightarrow \alpha \bullet X \beta$  em I  
 $A \rightarrow \alpha X \bullet \beta$ , calculando seu fechamento

# Função de Transição (Exemplo)

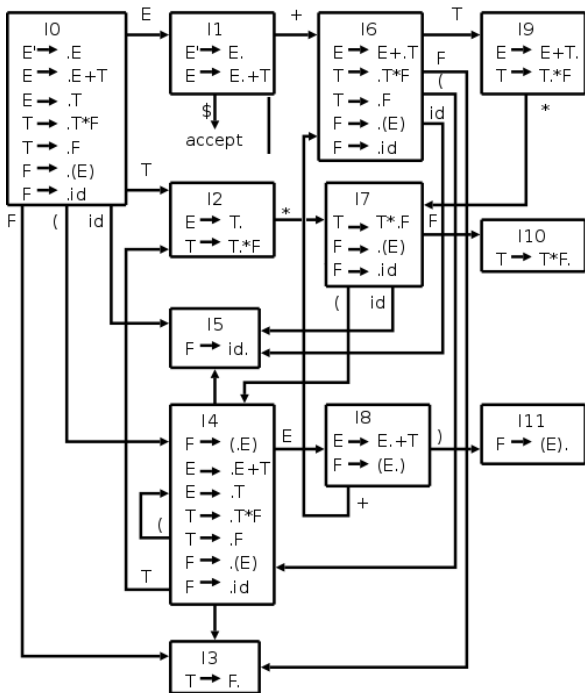
- ▶ Transição( $I, X$ ) é
  - ▶ Fechamento do conjunto dos items  
 $A \rightarrow \alpha X \bullet \beta$  tais que  $A \rightarrow \alpha \bullet X \beta$  está em  $I$
  - ▶ Examinar somente produções onde  $X$  está logo depois de  $\bullet$
- ▶ Exemplo, considerando a gramática já estendida
$$\begin{aligned} E' &\rightarrow E \\ E &\rightarrow E + T \mid T \\ T &\rightarrow T * F \mid F \\ F &\rightarrow ( E ) \mid id \end{aligned}$$
- ▶ Se  $I = \{ [E' \rightarrow E\bullet], [E \rightarrow E\bullet + T] \}$ 
  - ▶ Qual o conjunto de transição( $I, +$ ) ?
  - ▶ Qual o conjunto de transição( $I, \$$ ) ?

# Função de Transição (Resposta do Exemplo)

- ▶ Se  $I = \{ [E' \rightarrow E\bullet], [E \rightarrow E\bullet + T] \}$ 
  - ▶ Transição( $I, +$ )

E	$\rightarrow$	E + $\bullet$ T
T	$\rightarrow$	$\bullet$ T * F
T	$\rightarrow$	$\bullet$ F
F	$\rightarrow$	$\bullet$ ( E )
F	$\rightarrow$	$\bullet$ id

- ▶ Transição( $I, \$$ )  $\leadsto$  **Aceita**



# Construindo um exemplo completo

- ▶ Gramática de parênteses balanceados com um **a** no meio

$$A \rightarrow ( A ) \mid a$$

- ▶ Utilizando as funções de fechamento e transição
  - ▶ Construa o autômato finito determinístico LR(0)

# Tabela LR(0)

- ▶ Representação tabular do autômato LR(0)
- ▶ Regras de construção da parte Ação
  - ▶ Se  $[A \rightarrow \alpha \bullet a \beta] \in I_i$  e  $\text{Transição}(I_i, a) = I_j$   
defina  $\text{Ação}(i, a)$  como “Empilha j”
  - ▶ Se  $[A \rightarrow \alpha \bullet] \in I_i$   
defina  $\text{Ação}(i, a)$  como “Reduz  $A \rightarrow \alpha$ ”
  - ▶ Se  $[S' \rightarrow S \bullet] \in I_i$  (Ação de aceitação)  
defina  $\text{Ação}(i, \$)$  como “Aceita”
- ▶ Regras de construção da parte Transição
  - ▶ Para todos os não-terminais A  
Se  $\text{Transição}(I_i, A) = I_j$ , então  $\text{Transição}(i, A) = j$
- ▶ Estado inicial é aquele construído a partir de  $[S' \rightarrow \bullet S]$

# Construindo a tabela (Exemplo)

- Considerando o autômato LR(0) para a gramática de parênteses balanceados com um **a** no meio

$$\begin{array}{ll} 0 & A' \rightarrow A \\ 1 & A \rightarrow (A) \\ 2 & A \rightarrow a \end{array}$$

- Resposta

Estado	(	a	)	\$	A
0	e3	e2			1
1	r0	r0	r0	a	
2	r2	r2	r2		
3	e3	e2			4
4				e5	
5	r1	r1	r1		

- Analisando a entrada **((a))**

# Conclusão

- ▶ Leituras Recomendadas
  - ▶ Livro do Dragão
  
- ▶ Próxima Aula
  - ▶ SLR(1) e LR(1)