

– INF01147 –
Compiladores

Análise Sintática
Parser SLR(1)
Introdução ao Parser LR(1)

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



Autômato para Parser LR(0)

- ▶ Gramática de expressões aritméticas rudimentares

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

- ▶ Construa o autômato determinístico LR(0) e a tabela LR(0)
- ▶ Regras de construção da tabela LR(0) – Lembrete
 - ▶ Se $[A \rightarrow \alpha \bullet a \beta] \in E0$ e $\text{Transição}(E0, a) = E1$
 - ▶ defina $\text{Ação}(E0, a)$ como “Empilha j”
 - ▶ Se $[A \rightarrow \alpha \bullet] \in E0$
 - ▶ defina $\text{Ação}(E0, a) \forall \text{ terminal } a$ como “Reduz $A \rightarrow \alpha$ ”
 - ▶ Se $[S' \rightarrow S \bullet] \in E0$
 - ▶ defina $\text{Ação}(E0, \$)$ como “Aceita”
 - ▶ Para todos os não-terminais A , se $\text{Transição}(E0, A) = E1$
 - ▶ defina $\text{Transição}(E0, A) = E1$

Plano da Aula de Hoje

- ▶ Gramáticas LR(0)
- ▶ Parser SLR(1)
- ▶ Introdução ao Parser LR(1)

Gramáticas LR(0)

- ▶ Gramáticas LR(0) \Rightarrow reconhecidas pela Análise LR(0)
- ▶ Algoritmo para Análise LR(0)
 - ▶ Se um estado contiver $[A \rightarrow \alpha \bullet X \beta]$ (X sendo um terminal)
 - ▶ A ação é de empilhar o estado que contém $[A \rightarrow \alpha X \bullet \beta]$
 - ▶ Se um estado contiver $[A \rightarrow \gamma \bullet]$
 - ▶ A ação é de reduzir pela regra $A \rightarrow \gamma$
- ▶ Uma gramática é LR(0) se não houver ambiguidades
 - ▶ Autômato LR(0) \Rightarrow sem conflitos
 - ▶ Tabela LR(0) \Rightarrow cada célula **com uma única ação**

Gramáticas LR(0) – Conflitos

- Empilha-Reduz - um estado contiver os seguintes itens

$$A \rightarrow \alpha \bullet$$

$$A \rightarrow \alpha \bullet X \beta$$

- Reduz-Reduz - um estado contiver os seguintes itens

$$A \rightarrow \alpha \bullet$$

$$B \rightarrow \beta \bullet$$

Gramáticas LR(0) – Exemplos e contra-exemplos

- ▶ Exemplo de gramáticas LR(0)

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

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

- ▶ Exemplos de gramáticas que não são LR(0)

- ▶ Gramática de expressões aritméticas rudimentares

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

- ▶ Gramática de parênteses balanceados

$$S \rightarrow (S)S \mid \epsilon$$

- ▶ Gramática do **else** opcional

$$S \rightarrow IF \mid a$$

$$IF \rightarrow \text{if } S \mid \text{if } S \text{ else } S$$

Parser SLR(1)

Parser SLR(1) – Introdução

- ▶ Funciona sobre o autômato LR(0)
- ▶ Utiliza um token de *lookahead* de duas formas
 - ▶ Garantir a existência de uma transição válida no autômato
 - ▶ Utiliza o conjunto Sequência(A) para decidir sobre as reduções
- ▶ Algoritmo para Análise SLR(1)
 - ▶ Se um estado contiver $[A \rightarrow \alpha \bullet X \beta]$ e *lookahead* = X
 - ▶ A ação é de empilhar o estado que contém $[A \rightarrow \alpha X \bullet \beta]$
 - ▶ Se um estado contiver $[A \rightarrow \gamma \bullet]$ e *lookahead* \in Sequência(A)
 - ▶ A ação é de reduzir pela regra $A \rightarrow \gamma$
- ▶ Gramática é SLR(1) se não houver ambiguidades

Tabela SLR(1)

- ▶ Tabela do autômato LR(0) com alterações SLR(1)
- ▶ Regras de construção da parte Ação
 - ▶ Se $[A \rightarrow \alpha \bullet a \beta] \in E0$ e $\text{Transição}(E0, a) = E1$
 - ▶ defina $\text{Ação}(E0, a)$ como “Empilha j”
 - ▶ Se $[A \rightarrow \alpha \bullet] \in E0$
 - ▶ defina $\text{Ação}(E0, a)$ como “Reduz $A \rightarrow \alpha$ ” $\forall a \in \text{Sequencia}(A)$
 - ▶ Se $[S' \rightarrow S \bullet] \in E0$
 - ▶ defina $\text{Ação}(E0, \$)$ como “Aceita”
 - ▶ Para todos os não-terminais A, se $\text{Transição}(E0, A) = E1$
 - ▶ defina $\text{Transição}(E0, A) = E1$
- ▶ Estado inicial é aquele construído a partir de $[S' \rightarrow \bullet S]$

Parser SLR(1) – Exemplo

- ▶ Gramática de expressões aritméticas rudimentares

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

- ▶ Utilizando as funções de fechamento e transição
 - ▶ Construa o autômato finito determinístico LR(0)
 - ▶ Defina a tabela SLR(1)
- ▶ Analise $n + n + n$ com a tabela recém criada
- ▶ Tabela SLR(1) correspondente

	n	+	\$	E
0	e2			1
1		e3	aceita	
2		$r(E \rightarrow n)$	$r(E \rightarrow n)$	
3	e4			
4				
5		$r(E \rightarrow E + n)$	$r(E \rightarrow E + n)$	

Exercício de aplicação do método SLR(1)

- ▶ Gramática de parênteses balanceados

$$(1) S \rightarrow (S) S$$

$$(2) S \rightarrow \epsilon$$

- ▶ Utilizando as funções de fechamento e transição
 - ▶ Construa o autômato finito determinístico LR(0)
 - ▶ Defina a tabela SLR(1)
- ▶ Analise $()()$ com a tabela recém criada
- ▶ Tabela SLR(1) correspondente

	()	\$	S
0	e2	r2	r2	1
1			aceita	
2	e2	r2	r2	3
3		e4		
4	e2	r2	r2	5
5		r1	r1	

Gramáticas LR(0) *versus* Gramáticas SLR(1)

► Gramáticas LR(0)

- Se um estado contém $[A \rightarrow \alpha \bullet]$, não pode conter outros itens
- Traduzindo: ou o estado é de empilhar, ou de reduzir

► Gramáticas SLR(1)

- Se um estado contém $[A \rightarrow \alpha \bullet X \beta]$ (X é terminal)
 - Não há nesse estado um item $[C \rightarrow \gamma \bullet] \mid X \in \text{Sequência}(C)$
- Se um estado contém dois itens $[A \rightarrow \alpha \bullet]$ e $[B \rightarrow \beta \bullet]$
 - $\text{Sequência}(A) \cap \text{Sequência}(B) = \epsilon$

Parser SLR(1) – Conflitos

- ▶ Empilha-Reduz

- ▶ Estado com os itens abaixo e $X \in \text{Sequência}(B)$

$$A \rightarrow \alpha \bullet X \beta$$

$$B \rightarrow \alpha \bullet$$

- ▶ Reduz-Reduz

- ▶ Estado com os itens abaixo e $\text{Sequência}(A) \cap \text{Sequência}(B) \neq \epsilon$

$$A \rightarrow \alpha \bullet$$

$$B \rightarrow \beta \bullet$$

Parser SLR(1) – Conflito em evidência

- ▶ Gramática do **else** opcional

$$S \rightarrow IF \mid a$$
$$IF \rightarrow \text{if } S \mid \text{if } S \text{ else } S$$

- ▶ Utilizando as funções de fechamento e transição
 - ▶ Construa o autômato finito determinístico LR(0)
 - ▶ Defina a tabela SLR(1)

Parser SLR(1) – Limitações

- ▶ SLR(1) é simples e eficaz, mas com limitações
- ▶ Gramática de declarações

declaração	→	ativação atribuição
ativação	→	identificador
atribuição	→	variável = expressão
variável	→	variável [expressão] identificador
expressão	→	variável número

- ▶ Simplificando esta gramática

S	→	id V = E
V	→	id
E	→	V n

- ▶ Veja o estado inicial do autômato LR(0) e transição com **id**
- ▶ Perceber o conflito Reduz-Reduz
 - ▶ A redução por $V \rightarrow \text{id}$ *nunca* deveria ser efetuada com **\$**
 - ▶ V *nunca* pode ocorrer no final antes de = ser visto e empilhado

Parser LR(1)

Introdução

Parser LR(1)

- ▶ Donald Knuth propõem parser LR(1) (em 1965) LR Canônico
 - ▶ Primeiro algoritmo ascendente
- ▶ Resolve a limitação da análise SLR(1)
- ▶ Custo: aumento substancial da complexidade
 - ▶ Não é utilizada na prática
- ▶ Parser LALR(1)
 - ▶ Preserva as vantagens da análise LR(1)
 - ▶ Com a eficiência do método SLR(1)

Itens LR(0) Lembrete

- ▶ O ponto \bullet indica a posição em uma produção
 - ▶ Diferencia o que já foi lido do que ainda é esperado
- ▶ Supondo a produção $A \rightarrow \beta\gamma$
Três itens são possíveis
 - ▶ $A \rightarrow \bullet\beta\gamma$
Um item que é uma **possibilidade**
 - ▶ $A \rightarrow \beta \bullet \gamma$
(Progredimos reconhecendo β)
Um item **parcialmente completo**
 - ▶ $A \rightarrow \beta\gamma\bullet$
 $\beta\gamma$ estão empilhados
Um item **completo**

Itens LR(1)

- ▶ Analisadores LR(1) necessitam de um token de *lookahead*
- ▶ Item LR(1)
 - ▶ Produção com o ponto \bullet
 - ▶ Símbolo de *lookahead*
- ▶ Supondo a produção $A \rightarrow \beta\gamma$ e um token a de *lookahead*
Três itens são possíveis
 - ▶ $[A \rightarrow \bullet\beta\gamma, a]$
 - ▶ $[A \rightarrow \beta \bullet \gamma, a]$
 - ▶ $[A \rightarrow \beta\gamma \bullet, a]$
 $\beta\gamma$ estão empilhados, podemos reduzir se $a \in \text{Follow}(A)$

Itens LR(1) Importantes

- Considerando a gramática exemplo

Objetivo	→	Lista
Lista	→	Lista Par
Lista	→	Par
Par	→	(Par)
Par	→	()

- [Objetivo → • Lista, \$]
Representa o **estado inicial** do analisador
- [Objetivo → Lista •, \$]
Representa o **estado final** desejado

Conclusão

- ▶ Leituras Recomendadas
 - ▶ Livro do Dragão
 - ▶ Seções 4.6 até 4.8
 - ▶ Série Didática
 - ▶ Seções 3.3 (3.3.3 contém SLR) e 3.4
 - ▶ Keith Cooper
 - ▶ Seções 3.4 até 3.4.2 (somente SLR)

- ▶ Próxima Aula
Análise Sintática