

– INF01147 –
Compiladores

Análise Sintática
Gramáticas LL(1) e Análise LL(1)

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



Conjuntos Primeiro e Sequência

(revisão da aula anterior)

Sumário sobre Conjuntos Primeiro e Sequência

- ▶ Compostos somente por **terminais**
e também ϵ no caso dos conjuntos Primeiro
- ▶ Algoritmo de cálculo do **Primeiro(α)**
 - ▶ Trivial quando α é um terminal
 - ▶ Se α é um não-terminal
 - ▶ Varre todas as produções $\alpha \rightarrow \dots$
 - ▶ Complica se o início de uma produção deriva em ϵ
 - ▶ Contém ϵ somente se $\alpha \Rightarrow^* \epsilon$
- ▶ Algoritmo de cálculo do **Sequência(α)**
 - ▶ Inclui o \$ em alguns casos triviais (não-terminal inicial)
 - ▶ Pois o que vem depois do símbolo inicial é o fim da entrada
 - ▶ Varre as produções onde α aparece no corpo ($X \rightarrow Y \alpha Z$)
 - ▶ Complica se $Z \Rightarrow^* \epsilon$
 - ▶ Complica quando α aparece no fim de uma regra ($X \rightarrow Y \alpha$)
 - ▶ Nunca contém o ϵ

Exemplo de Revisão

- Considerando a gramática

$$E \rightarrow T E'$$

$$E' \rightarrow + T E' \mid \epsilon$$

$$T \rightarrow F T'$$

$$T' \rightarrow * F T' \mid \epsilon$$

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

- Calcule os conjuntos Primeiro e Sequência

Recursivo Preditivo – Usando Primeiro e Sequência

- Considerando a seguinte gramática

$S \rightarrow ACE$

$A \rightarrow a \mid b \mid \epsilon$

$C \rightarrow c \mid d \mid \epsilon$

$E \rightarrow e$

- Implementação possível para o não-terminal A desta gramática

```
A() {  
    switch (lookahead_token) {  
        //Tratando os símbolos em Primeiro(A)  
        case 'a': Consome('a'); break;  
        case 'b': Consome('b'); break;  
  
        //Utilizando o Sequência(A) para tratar da produção vazia de A  
        case 'c':  
        case 'd':  
        case 'e': break;  
  
        //Erro  
        default: abortar ("Erro Sintático");  
    }  
}
```

Análise Sintática LL(1)

Visão Geral

Ações da Análise LL(1)

Pilha, Entrada e Ação

- ▶ Utiliza uma **pilha** explícita (ao invés de ativações recursivas)
 - ▶ Com três tipos de ações: empilhamento, casamento e aceita
- ▶ Sendo a gramática que gera cadeias de parênteses balanceados

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

- ▶ Análise da entrada ()

Pilha de análise sintática	Entrada	Ação
\$ S	() \$	$S \rightarrow (S) S$
\$ S) S (() \$	casamento
\$ S) S) \$	$S \rightarrow \epsilon$
\$ S)) \$	casamento
\$ S	\$	$S \rightarrow \epsilon$
\$	\$	aceita

- ▶ Relação com os passos de uma **derivação à esquerda** da cadeia
- ▶ Decisões guiadas por uma **Tabela de Análise Sintática LL(1)**

Tabela de Análise Sintática LL(1)

- ▶ Correlaciona cada não-terminal com cada terminal $\Rightarrow M[N, T]$
 - ▶ Controla as ações de empilhamento (e substituição para o ϵ)
- ▶ Sendo a gramática que gera cadeias de parênteses balanceados

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

- ▶ Tabela de Análise Sintática LL(1) para esta gramática

$M[N, T]$	()	\$
S	$S \rightarrow (S) S$	$S \rightarrow \epsilon$	$S \rightarrow \epsilon$

- ▶ A tabela guia as ações de empilhamento da análise LL(1)
 - ▶ Exemplo com a entrada $()()$

Gramáticas LL(1)

Gramáticas LL(1)

- ▶ LL(1), uma classe de gramáticas onde
 - ▶ A entrada é lida da **esquerda para a direita**
L de *left-to-right*
 - ▶ Aplica-se uma **derivação mais à esquerda**
L de *leftmost*
 - ▶ Tomada de decisão precisa somente um token na entrada
1
- ▶ **Definição**
Uma gramática é LL(1) se a tabela de análise sintática LL(1) associada tiver no máximo uma produção em cada célula
- ▶ **Condições necessárias** para ser LL(1) (reflexos da definição)
 - ▶ Sem recursão à esquerda
 - ▶ Fatorada à esquerda
 - ▶ **Sem ambiguidade**
- ▶ Suficiente para reconhecer a maioria das construções

Gramáticas LL(1) – Definição Formal

- ▶ Uma gramática G é LL(1) se e somente se
- ▶ Considerando $A \rightarrow \alpha \mid \beta$
- ▶ Para um terminal a , somente uma produção começa por a
 - ▶ $\text{Primeiro}(\alpha) \cap \text{Primeiro}(\beta) = \emptyset$
- ▶ Somente uma produção pode derivar para ϵ
- ▶ Se uma produção deriva pra vazio, a outra não deve iniciar com o que vem depois de A , ou seja
 - ▶ Se $\alpha \Rightarrow^* \epsilon$, $\text{Primeiro}(\beta) \cap \text{Sequência}(A) = \emptyset$
 - ▶ Se $\beta \Rightarrow^* \epsilon$, $\text{Primeiro}(\alpha) \cap \text{Sequência}(A) = \emptyset$
- ▶ Esta gramática é LL(1)?

$$\begin{array}{ll} S & \rightarrow \text{ iEtSS}' \mid a \\ S' & \rightarrow \text{ eS} \mid \epsilon \\ E & \rightarrow \text{ b} \end{array}$$

Gramáticas LL(1) – Reconhecimento

- ▶ Gramáticas LL(1) podem ser reconhecidas por analisadores sintáticos **descendentes preditivos**
- ▶ Exemplo

Única produção possível para cada terminal em Primeiro(stmt)

```
stmt  →  if ( expr ) stmt else stmt
        |  while ( expr ) stmt
        |  { stmt_list }
```

- ▶ Implementação
 - ▶ **Tabela de Análise Sintática LL(1)**
 - ▶ Correlaciona não-terminais e terminais (tokens) da entrada
 - ▶ Cada posição guarda uma única regra de produção

Tabela Preditiva

Tabela Preditiva – Algoritmo de Construção

- ▶ Para cada produção $A \rightarrow \alpha$ da gramática, faça
 - ▶ Para cada terminal a em $\text{Primeiro}(\alpha)$, inclua $(A \rightarrow \alpha)$ em $M[A,a]$
 - ▶ Se $\epsilon \in \text{Primeiro}(\alpha)$, inclua $(A \rightarrow \alpha)$ em $M[A,b]$
 - ▶ Para cada terminal b em $\text{Sequência}(A)$
 - ▶ Se $\epsilon \in \text{Primeiro}(\alpha)$ e $\$ \in \text{Sequência}(A)$, inclua $(A \rightarrow \alpha)$ em $M[A,\$]$
- ▶ Ao fim do algoritmo
 - ▶ Células vazias na tabela são consideradas erros

Tabela Preditiva – Exemplo 1

- Considerando a gramática

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \epsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid \epsilon$$

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

	Primeiro	Sequência
E	(id) \$
E'	+ ϵ) \$
T	(id	+) \$
T'	* ϵ	+) \$
F	(id	+ *) \$

- Tabela Preditiva

	id	+	*	()	\$
E						
E'						
T						
T'						
F						

Tabela Preditiva – Solução do Exemplo 1

	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Tabela Preditiva – Exemplo 2

► Gramática do **else** opcional

$S \rightarrow iEtSS' \mid a$

$S' \rightarrow eS \mid \epsilon$

$E \rightarrow b$

	Primeiro	Sequência
S	i a	e \$
S'	e ϵ	e \$
E	b	t

► Tabela Preditiva

	a	b	e	i	t	\$
S						
S'						
E						

Tabela Preditiva – Solução do Exemplo 2

- Notem como a ambiguidade se manifesta
 - Reconhecendo S' e temos um **else**, o que fazer?
 - Devemos escolher $S' \rightarrow eS$
(casar o **else** com o **then** mais próximo)

	a	b	e	i	t	\$
S	$S \rightarrow a$			$S \rightarrow iEtSS'$		
S'			$S' \rightarrow eS$ $S' \rightarrow \epsilon$			$S' \rightarrow \epsilon$
E		$E \rightarrow b$				

Análise Preditiva Tabular

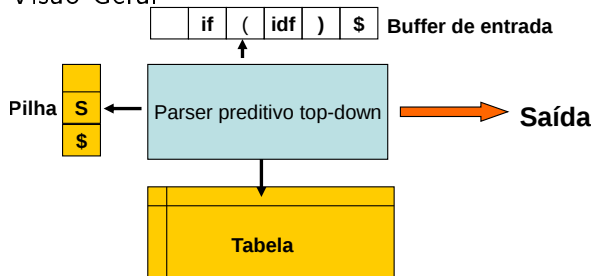
Análise Preditiva Tabular

- ▶ Vimos até agora dois métodos
 - ▶ Recursivo com Retrocesso
 - ▶ Recursivo Preditivo – para Gramáticas LL(1)
- ▶ Implementação
 - ▶ Cada não-terminal tem um procedimento associado
 - ▶ Realiza chamadas **recursivas** quando apropriado
Exemplo: $S \rightarrow iEtSS'$
 - ▶ Utilizam uma pilha implícita de chamadas
- ▶ **Desvantagens** desta abordagem
 - ▶ Relativo sobrecusto por chamada
 - ▶ Pilha tem tamanho limitado pela memória
- ▶ **Análise Preditiva Tabular**
 - ▶ Tem uma pilha explícita para armazenar os não-terminais
 - ▶ Usa a **Tabela Preditiva** para guiar derivações

Análise Preditiva Tabular – Visão Geral

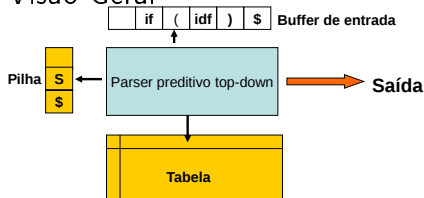
- ▶ Buffer de tokens na entrada (o token \$ marca o fim)
- ▶ Pilha com fundo marcado por \$
 - ▶ Inicializada com o símbolo não-terminal S
- ▶ Tabela preditiva

- ▶ Visão Geral



Análise Preditiva Tabular – Funcionamento

- ▶ Seja
 - ▶ X um símbolo no topo da pilha
 - ▶ a um token no buffer de entrada
- ▶ Etapas
 - ▶ Se $X == \$$ e $a == \$ \leadsto$ Reconheceu a sentença
 - ▶ Se $X == a$ e $a != \$ \leadsto$ Desempilha e avança
 - ▶ Se X é um não-terminal \leadsto Consulta a tabela $M[X,a]$
 - ▶ Se contém $X \rightarrow UVW \leadsto$ Empilha WVU
 - ▶ Se for vazia \leadsto tratamento de erros
- ▶ Visão Geral



Análise Preditiva Tabular – Exemplo 1

- Reconhecer **id+id*id**
- Considerando a tabela preditiva
- A pilha começa com **E**

	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Análise Preditiva Tabular – Exemplo 2

- ▶ Reconhecer **ibtibtaeaea**
- ▶ Considerando a tabela preditiva
- ▶ A pilha começa com **S**
- ▶ Devemos tratar a **ambiguidade** em $M[S', e]$
 - ▶ Faça desconsiderando $S' \rightarrow \epsilon$
 - ▶ O que acontece se desconsiderarmos $S' \rightarrow eS$?

	a	b	e	i	t	\$
S	$S \rightarrow a$			$S \rightarrow iEtSS'$		
S'			$S' \rightarrow eS$ $S' \rightarrow \epsilon$			$S' \rightarrow \epsilon$
E		$E \rightarrow b$				

Gerenciamento de Erros

- ▶ Modo pânico
 - ▶ Descarte até achar um **token de sincronismo**
 - ▶ Existe uma série de heurísticas
- ▶ Heurística: utilizar o conjunto Sequência(A)
- ▶ Exemplo para a gramática de operadores aritméticos

	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	synch	synch
E'		$E' \rightarrow +TE'$			$E' \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$	synch		$T \rightarrow FT'$	synch	synch
T'		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow id$	synch	synch	$F \rightarrow (E)$	synch	synch

- ▶ Funcionamento
 - ▶ Se $M[A,a]$ é branco \leadsto ignore o token na entrada
 - ▶ Se $M[A,a]$ é **synch** \leadsto desempilhe
 - ▶ Se terminal na pilha for diferente da entrada \leadsto desempilhe

- ▶ Teste as entradas erradas:

+ id * + id

) id * + id

Sumário da Aula

- ▶ Análise descendente para gramáticas LL(1)
 - ▶ Eficiente e simples
 - ▶ Implementação preditiva com tabela
 - ▶ Baseada nos cálculos dos conjuntos Primeiro/Sequência
- ▶ Limitação
 - ▶ O que fazer se a gramática não é LL(1)?

Conclusão

- ▶ Leituras Recomendadas
 - ▶ Livro do Dragão
 - ▶ Seções 4.4.3 até 4.4.6
 - ▶ Série Didática
 - ▶ Seção 3.2.2 e 3.2.3
- ▶ Próxima Aula
 - ▶ Apresentação da Etapa 2
 - ▶ Sala 104 do Prédio 67 (Turma A)
 - ▶ Sala 101 do Prédio 67 (Turma B)