

Compiladores

Análise sintática (3)

Análise LL(1) com tabela preditiva.

proc First(α : string of symbols)

```
Repeat {  
  // seja  $\alpha \rightarrow X_1 X_2 X_3 \dots X_n$   
  if  $X_1 \in T$   
  then // caso simples onde  $X_1$  é um terminal  
    First( $\alpha$ ) :=  $\{X_1\}$   
  else { //  $X_1$  é um não-terminal  
    First( $\alpha$ ) = First( $X_1$ ) \  $\{\epsilon\}$ ;  
    for (i=1 ; i<=n ; i++) {  
      if  $\epsilon$  is in First( $X_1$ ) and in First( $X_2$ ) and in... First( $X_{i-1}$ )  
        First( $\alpha$ ) := First( $\alpha$ )  $\cup$  First( $X_i$ ) \  $\{\epsilon\}$   
    }  
  }  
  if ( $\alpha \Rightarrow^* \epsilon$ )  
    then First( $\alpha$ ) := First( $\alpha$ )  $\cup$   $\{\epsilon\}$   
} until no change in any First( $\alpha$ )
```

proc Follow($B \in N$)

```
Follow(S) :=  $\{\$$ };  
Repeat  
  foreach  $p \in P$  do {  
    // Varre as produções  
    case  $p == A \rightarrow \alpha B \beta$  {  
      Follow(B) := Follow(B)  $\cup$  First( $\beta$ ) \  $\{\epsilon\}$ ;  
      if  $\epsilon \in$  First( $\beta$ ) then  
        Follow(B) := Follow(B)  $\cup$  Follow(A);  
      end  
    }  
    case  $p == A \rightarrow \alpha B$   
      Follow(B) := Follow(B)  $\cup$  Follow(A);  
  }  
until no change in any Follow(N)
```

Exemplo First/Follow

$S \rightarrow XYZ$

$X \rightarrow aXb \mid \epsilon$

$Y \rightarrow cYZcX \mid d$

$Z \rightarrow eZYe \mid f$

First(X) = {a, ϵ } Follow(X) = {c, d, b, e, f}

First(Y) = {c, d} Follow(Y) = {e, f}

First(Z) = {e, f} Follow(Z) = {\$, c, d}

First(S) = {a, c, d} Follow(S) = {\$}

Observações First/Follow

- Só terminais entram em First e Follow.
- O algoritmo de cálculo de First(α):
 - É trivial quando α é um terminal t.
 - varre as produções $X \rightarrow t\omega$ quando α é um não-terminal X;
 - é chato quando o início de uma derivação de X deriva em ϵ .
 - Inclui ϵ apenas quando X pode derivar em ϵ .
- O algoritmo de cálculo de Follow(X)
 - É reservado aos não-terminais X
 - Inclui o \$ em alguns casos triviais ($X ==$ o start S)
 - Varre as produções onde X aparece à direita ($A \rightarrow \omega X \omega'$)
 - É chato quando X aparece no fim (ou logo antes de algo que deriva em ϵ)
 - NUNCA inclui ϵ

Gramática LL(1)

Condições necessárias:

- sem ambigüidade
- sem recursão a esquerda

Uma gramática G é LL(1) sse

$A \rightarrow \alpha \mid \beta \Rightarrow^* t$

1. First(α) $\cap$ First(β) = $\emptyset$
2. $\alpha \Rightarrow^* \epsilon$ implies $!(\beta \Rightarrow^* \epsilon)$
3. $\alpha \Rightarrow^* \epsilon$ implies First(β) $\cap$ Follow(A) = $\emptyset$

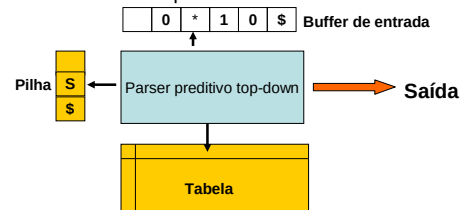
LL(1) = leitura Left -> right + derivação mais a esquerda (Left) + uso de 1 token lookahead.

Análise top-down com tabela preditiva

- Os dois métodos apresentados até agora para fazer análise descendente usam recursividade.
 - Cada não-terminal tem um procedimento associado;
 - Chamadas com ou sem retrocesso.
 - Para gramáticas LL1 não tem retrocesso.
- Chamadas recursivas usam uma pilha implícita
 - A pilha das chamadas!
 - Sobrecusto!
- Idéia: de-recursificar o procedimento:
 - Usa-se uma pilha para armazenar os não-terminais encontrados;
 - Usa-se uma tabela para orientar as derivações.

Reconhecedor preditivo com Pilha

- Tem um buffer em entrada;
 - \$ marca seu fim.
- Tem um fluxo de saída;
- Uma pilha cujo fundo é marcado por \$
 - Inicializada com S (Start)
- Uma tabela sintática preditiva M



Funcionamento do parser

- Seja X o símbolo no topo da pilha
- Seja a o símbolo de entrada (terminal!) a analisar
 - Se $X = \$$ e $a = \$$: para e reconheceu uma sentença.
 - Se $X = a$ e $a \neq \$$: desempilha X e avança de um símbolo na entrada.
 - Se X é não-terminal: Consulta a tabela $M(X, a)$
 - Se for vazia: ERRO
 - Se contém $X \rightarrow UVW$, então substitui na pilha X por UVW (U no topo).

Algoritmo para construir a tabela

- Re-escrever a gramática para satisfazer condições de LL(1)
- Calcular os conjuntos First e Follow
- Para cada produção $A \rightarrow \alpha$
 - Para cada $a \in \text{First}(\alpha)$
 - incluir a produção $A \rightarrow \alpha$ em $M[A, a]$
 - Se $\epsilon \in \text{First}(\alpha)$
 - incluir a produção $A \rightarrow \alpha$ em $M[A, b]$ para cada b em $\text{Follow}(A)$
 - Se $\epsilon \in \text{First}(\alpha)$ e $\$ \in \text{Follow}(A)$
 - incluir $A \rightarrow \alpha$ to $M[A, \$]$
- Todas entradas não definidas são erros

A tabela preditiva M(X, t)

- Tabela Bi-dimensional:
 - Dimensão 1: Não-terminal X
 - Dimensão 2: Caractere da entrada (terminal) t
 - A entrada (X,t) contém a regra de produção a aplicar

	a	b	c	\$
S				
A				
B				

$S \rightarrow cAa$

$A \rightarrow cB \mid B$

$B \rightarrow bcB \mid \epsilon$

$\text{First}(A) = \{b, c, \epsilon\}$

$\text{First}(B) = \{b, \epsilon\}$

$\text{First}(S) = \{c\}$

$\text{Follow}(S) = \{\$\}$

$\text{Follow}(A) = \{a\}$

$\text{Follow}(B) = \{a\}$

Exemplo de tabela M(X, t)

- Tabela Bi-dimensional:
 - Dimensão 1: Não-terminal X
 - Dimensão 2: Caractere da entrada (terminal) t
 - A entrada (X,t) contém a regra de produção a aplicar

	a	b	c	\$
S			$S \rightarrow cAa$	
A				
B				

$S \rightarrow cAa$

$A \rightarrow cB \mid B$

$B \rightarrow bcB \mid \epsilon$

$\text{First}(A) = \{b, c, \epsilon\}$

$\text{First}(B) = \{b, \epsilon\}$

$\text{First}(S) = \{c\}$

$\text{Follow}(S) = \{\$\}$

$\text{Follow}(A) = \{a\}$

$\text{Follow}(B) = \{a\}$

Exemplo de tabela M(X, t)

- Tabela Bi-dimensional:
 - Dimensão 1: Não-terminal X
 - Dimensão 2: Caractere da entrada (terminal) t
 - A entrada (X,t) contém a regra de produção a aplicar

	a	b	c	\$
S			S → cAa	
A	A → B	A → B		
B				

$S \rightarrow cAa$
 $A \rightarrow cB \mid B$
 $B \rightarrow bcB \mid \epsilon$

$\text{First}(A) = \{b, c, \epsilon\}$
 $\text{First}(B) = \{b, \epsilon\}$
 $\text{First}(S) = \{c\}$

$\text{Follow}(S) = \{\$ \}$
 $\text{Follow}(A) = \{a\}$
 $\text{Follow}(B) = \{a\}$

Exemplo de tabela M(X, t)

- Tabela Bi-dimensional:
 - Dimensão 1: Não-terminal X
 - Dimensão 2: Caractere da entrada (terminal) t
 - A entrada (X,t) contém a regra de produção a aplicar

	a	b	c	\$
S			S → cAa	
A	A → B	A → B	A → cB	
B				

$S \rightarrow cAa$
 $A \rightarrow cB \mid B$
 $B \rightarrow bcB \mid \epsilon$

$\text{First}(A) = \{b, c, \epsilon\}$
 $\text{First}(B) = \{b, \epsilon\}$
 $\text{First}(S) = \{c\}$

$\text{Follow}(S) = \{\$ \}$
 $\text{Follow}(A) = \{a\}$
 $\text{Follow}(B) = \{a\}$

Exemplo de tabela M(X, t)

- Tabela Bi-dimensional:
 - Dimensão 1: Não-terminal X
 - Dimensão 2: Caractere da entrada (terminal) t
 - A entrada (X,t) contém a regra de produção a aplicar

	a	b	c	\$
S			S → cAa	
A	A → B	A → B	A → cB	
B	B → ε			

$S \rightarrow cAa$
 $A \rightarrow cB \mid B$
 $B \rightarrow bcB \mid \epsilon$

$\text{First}(A) = \{b, c, \epsilon\}$
 $\text{First}(B) = \{b, \epsilon\}$
 $\text{First}(S) = \{c\}$

$\text{Follow}(S) = \{\$ \}$
 $\text{Follow}(A) = \{a\}$
 $\text{Follow}(B) = \{a\}$

Exemplo de tabela M(X, t)

- Tabela Bi-dimensional:
 - Dimensão 1: Não-terminal X
 - Dimensão 2: Caractere da entrada (terminal) t
 - A entrada (X,t) contém a regra de produção a aplicar

	a	b	c	\$
S			S → cAa	
A	A → B	A → B	A → cB	
B	B → ε	B → bcB		

$S \rightarrow cAa$
 $A \rightarrow cB \mid B$
 $B \rightarrow bcB \mid \epsilon$

$\text{First}(A) = \{b, c, \epsilon\}$
 $\text{First}(B) = \{b, \epsilon\}$
 $\text{First}(S) = \{c\}$

$\text{Follow}(S) = \{\$ \}$
 $\text{Follow}(A) = \{a\}$
 $\text{Follow}(B) = \{a\}$

Exemplo de tabela M(X, t)

- Tabela Bi-dimensional:
 - Dimensão 1: Não-terminal X
 - Dimensão 2: Caractere da entrada (terminal) t
 - A entrada (X,t) contém a regra de produção a aplicar

	a	b	c	\$
S	ERRO	ERRO	S → cAa	ERRO
A	A → B	A → B	A → cB	ERRO
B	B → ε	B → bcB	ERRO	ERRO

$S \rightarrow cAa$
 $A \rightarrow cB \mid B$
 $B \rightarrow bcB \mid \epsilon$

$\text{First}(A) = \{b, c, \epsilon\}$
 $\text{First}(B) = \{b, \epsilon\}$
 $\text{First}(S) = \{c\}$

$\text{Follow}(S) = \{\$ \}$
 $\text{Follow}(A) = \{a\}$
 $\text{Follow}(B) = \{a\}$

Usando a tabela

- String: "cbca"

Pilha	Entrada	Ação
S\$	cbca\$	S->cAa
cAa\$	cbca\$	casar c
Aa\$	bca\$	A->B
Ba\$	bca\$	B->bcB
bcBa\$	bca\$	casar b
cBa\$	ca\$	casar c
Ba\$	a\$	B->ε
a\$	a\$	casar a
\$	\$	casar \$, sucesso

Mais um exemplo...

$E \rightarrow TE'$
 $E' \rightarrow +TE' \mid \epsilon$
 $T \rightarrow FT'$
 $T' \rightarrow *FT' \mid \epsilon$
 $F \rightarrow (E) \mid \text{id}$

Símbolo	First	Follow
E	{(, id}	{\$,)}
E'	{+, ε}	{\$,)}
T	{(, id}	{+, \$,)}
T'	{*, ε}	{+, \$,)}
F	{(, id}	{*, +, \$,)}

Exemplo LL(1)

$S \rightarrow XYZ$

$X \rightarrow aXb \mid \epsilon$

$Y \rightarrow cYZcX \mid d$

$Z \rightarrow eZYe \mid f$

Construir Tabela
Analisar abcdcf

$\text{First}(X) = \{a, \epsilon\}$ $\text{Follow}(X) = \{c, d, b, e, f\}$

$\text{First}(Y) = \{c, d\}$ $\text{Follow}(Y) = \{e, f\}$

$\text{First}(Z) = \{e, f\}$ $\text{Follow}(Z) = \{\$, c, d\}$

$\text{First}(S) = \{a, c, d\}$ $\text{Follow}(S) = \{\$\}$

Observação sobre a Tabela

- A tabela indica se há ambigüidade!
 - Mais de uma regra numa entrada!
- Soluções?
 - Tornar a gramática LL(1)
 - Eliminar ambigüidade, recursividade... Ver os slides seguintes.
 - Usar uma heurística para desempatar as regras
 - Qual?
 - Usar outros algoritmos do que os top-down!
- Exemplo total: if... Then... Else:

$S \rightarrow i E t S \mid i E t S e S \mid a$
 $E \rightarrow b$

Transformações de GLCs

- Eliminação de produções vazias
- Eliminação de recursividade à esquerda:
 - recursão direta
 - recursão indireta
- Fatoração de uma gramática

Eliminação de produções vazias (1)

- Objetivo:**
 - eliminar produções da forma $A \rightarrow \epsilon$.
- Algoritmo:** seja $G = (N, T, P, S)$ uma GLC
 - Etapa 1:**
 - construir N_ϵ , o conjunto de não-terminais que geram a palavra vazia:

$N_\epsilon = \{A \mid A \rightarrow \epsilon\}$;
 - Repita

$N_\epsilon = N_\epsilon \cup \{X \mid X \rightarrow X_1 \dots X_n \in P \text{ tq } X_1, \dots, X_n \in N_\epsilon\}$
 - Até que o cardinal de N_ϵ não aumente.

Eliminação de produções vazias (2)

- Etapa 2:**
 - construir o conjunto de produções sem produções vazias:

gera $G_1 = (N, T, P_1, S)$, onde P_1 é construído como segue:
- $P_1 = \{A \rightarrow \alpha \mid \alpha \neq \epsilon\}$;
 Repita
 Para toda $A \rightarrow \alpha \in P_1$ e $X \in N_\epsilon$ tal que
 $\alpha = \alpha_1 X \alpha_2$ e $\alpha_1 \alpha_2 \neq \epsilon$
 Faça $P_1 = P_1 \cup \{A \rightarrow \alpha_1 \alpha_2\}$
 Até que o cardinal de P_1 não aumente

Eliminação de produções vazias (3)

– Etapa 3:

- incluir geração da palavra vazia, se necessário.
- Se a palavra vazia pertence à linguagem, então a gramática resultante é
- $G_2 = (N, T, P_2, S)$, onde $P_2 = P_1 \cup \{S \rightarrow \epsilon\}$

Eliminação de recursividade à esquerda

• Exemplo

$A \rightarrow Aa \mid b$

Com a palavra vazia

$A \rightarrow bX$
 $X \rightarrow aX \mid \epsilon$

Sem a palavra vazia

$A \rightarrow b \mid bX$
 $X \rightarrow a \mid aX$

Exemplo

- $E \rightarrow E+T \mid T$
- $T \rightarrow T * F \mid F$
- $F \rightarrow (E) \mid \text{Id}$

Convertendo recursão a esquerda para recursão a direita

$X \rightarrow Xb \mid Xc \mid AB \mid C$

• Introduzir Y

$X \rightarrow ABY \mid CY$
 $Y \rightarrow bY \mid cY \mid \epsilon$

Fatoração de uma gramática

- Elimina indecisão de qual produção aplicar quando duas ou mais produções iniciam com a mesma forma sentencial

$A \rightarrow \alpha\beta_1 \mid \alpha\beta_2$

Se torna:

$A \rightarrow \alpha X$

$X \rightarrow \beta_1 \mid \beta_2$

Exemplo de Fatoração a Esquerda

$\text{Cmd} \rightarrow \text{if Expr then Cmd else Cmd}$

$\text{Cmd} \rightarrow \text{if Expr then Cmd}$

$\text{Cmd} \rightarrow \text{Outro}$

• Fatorando a esquerda:

$\text{Cmd} \rightarrow \text{if Expr then Cmd ElseOpc}$

$\text{Cmd} \rightarrow \text{Outro}$

$\text{ElseOpc} \rightarrow \text{else Cmd} \mid \epsilon$

Gerenciamento de Erros

- Relatar erros & recuperar
 - Relatar erros assim que possível
 - Mensagens de erro adequadas
 - Continua após o erro
 - Evitar a cascata de erros
- Nível-Frase (local) x Modo-Pânico
 - Nível frase: tenta-se alterar UM símbolo para recuperar.
 - Modo pânico: pula x tokens de entrada até poder voltar a fazer a análise.
 - = até encontrar um token de sincronização.

Recuperação em Modo Pânico

- Pula tokens até que um “conjunto de sincronização” é encontrado
 - Follow(A) (A sendo no topo da pilha)
 - Símbolos de alta hierarquia na gramática
 - { , for, while, if...
 - First(A)
 - Epsilon produção
 - Pop/Insert um terminal no topo da pilha.
- Adicione ações de sincronia para a tabela

Sumário

- Análise top-down possibilita o reconhecimento eficiente e simples de gramáticas LL(1):
 - Implementação preditiva com tabela.
 - Baseada nos cálculos dos conjuntos First/Follow
 - Obs: têm casos que não foram tratados (e.g. o '+')
- Limitação:
 - Quando a gramática não é LL(1) !
- Por isso: usa-se também análise ascendente (bottom-up).