

Teoria da Computação

Aula Teórica 20:
Analisador sintáctico LL(1)

António Ravara

Departamento de Informática

27 de Maio de 2019

Analizador sintático LL(1)

- ▶ Pretende-se agora definir como verificar se dada palavra pertence à linguagem de uma gramática LL(1).
- ▶ A ideia não é gerar a palavra por derivação (o código de um programa pode ser bastante complexo e ter milhares de linhas), mas sim lê-la da esquerda para a direita e ir verificando que há uma derivação que gera a sequência de símbolos da palavra.
- ▶ É necessário associar as variáveis da gramática com os símbolos que geram. É o propósito das relações *FIRST* e *FOLLOW*.
- ▶ Porquê “LL(1)”? Lê da esquerda para a direita (L) e reproduz uma derivação *leftmost* (L), antecipando um símbolo (com *FIRST* e *FOLLOW*).

Analizador sintático LL(1) – função de transição

- ▶ Pretende-se relacionar cada par variável/símbolo com a regra que permite gerar o símbolo a partir da variável.
- ▶ Se tal não é possível, marca-se o par como um erro.
- ▶ A ideia é então definir um DFA para cada variável, sendo as palavras aceites por esse DFA as geradas pela variável na gramática.
- ▶ O analisador sintático recorre a uma função de transição que é a união das dos DFAs.

Considere-se uma gramática $G = \langle V, T, P, S \rangle$ e seja $a \in T$.

Relação de transição $\delta \subseteq V \times (\{\epsilon\} \cup T) \rightarrow (P \cup \{\text{SYNTAXERROR}\})$

$\delta(X, \epsilon) = X \rightarrow w$ se $X \rightarrow w \in P \wedge w \Rightarrow^* \epsilon$

$\delta(X, a) = X \rightarrow w$ se

$X \rightarrow w \in P \wedge (a \in \text{FIRST}(w) \vee (w \Rightarrow^* \epsilon \wedge a \in \text{FOLLOW}(X)))$

Analizador sintático LL(1) – intuição

Consiste em:

1. Uma variável de entrada, que contém a palavra a processar.
2. Uma pilha onde se colocam as palavras que são lados direitos das produções que se têm que tratar.
3. Uma tabela de *parsing*, que representa a função de transição definida atrás.

A ideia:

- ▶ Se
 - ▶ na entrada está a w_1 ,
 - ▶ no topo da pilha está a w_2 ;remove-se a da entrada e da palavra no topo da pilha.
- ▶ Se
 - ▶ na entrada está a w_1 ,
 - ▶ no topo da pilha está X w_2 ,
 - ▶ na tabela tem-se $\delta(X, a) = w_3$;substitui-se no topo da pilha X w_2 por w_3 w_2 .

Analizador sintático LL(1) – definição

Algoritmo

1. Coloca-se na pilha vazia o símbolo inicial.
2. Em cada passo (até a pilha estar vazia):
 - 2.1 lê-se o primeiro símbolo da palavra que está no topo da pilha;
 - 2.2 lê-se o primeiro símbolo da palavra que está na variável de entrada;
 - 2.3 se coincidem, apagam-se ambos;
 - 2.4 se não, e:
 - 2.4.1 são ambos terminais, o algoritmo termina;
 - 2.4.2 o que veio da pilha é uma variável, procura-se na tabela a regra correspondente:
 - se existe, aplica-se a regra à variável e re-coloca-se a palavra resultante na pilha;
 - se não, o algoritmo termina;

Se no fim de correr o algoritmo a pilha e a variável de entrada estão vazias, a palavra foi reconhecida (pertence à linguagem).

Análise sintáctica LL(1) – um exemplo

Seja $G = \langle \{I, F\}, \{ (,), *, x \}, I, P \rangle$, com P contendo as seguintes regras:

1. $I \rightarrow (I * F)$
2. $I \rightarrow F$
3. $F \rightarrow x$

Para se construir a tabela de *parsing*, é necessário obter os conjuntos *FIRST* e *FOLLOW* para cada variável.

- ▶ $FIRST(F) = FIRST(x) = \{x\}$
- ▶ $FIRST(I) = FIRST((I * F)) \cup FIRST(F) = \{(\cup \{x\} = \{(, x\}$
- ▶ $FOLLOW(I) = FIRST(*F)) \cup \emptyset = \{*\}$
- ▶ $FOLLOW(F) = (FIRST()) \cup \emptyset \cup (\emptyset \cup FOLLOW(I)) = \{ \} \cup \{*\}$

Análise sintáctica LL(1) – um exemplo

Seja $G = \langle \{I, F\}, \{ (,), *, x \}, I, P \rangle$, com P com as regras:

1. $I \rightarrow (I * F)$
2. $I \rightarrow F$
3. $F \rightarrow x$

Obtém-se agora a função de transição, onde $w \in (V \cup T)^*$.

- ▶ $\delta(I, () = I \rightarrow (I * F)$, pois $(\in \text{FIRST}((I * F))$
- ▶ $\delta(I,)) = \text{SYNTAXERROR}$, pois $) \notin \text{FIRST}((I * F)) \wedge (I * F) \not\Rightarrow^* \epsilon$ e $) \notin \text{FIRST}(F) \wedge F \not\Rightarrow^* \epsilon$
- ▶ $\delta(I, *) = \text{SYNTAXERROR}$, pois $* \notin \text{FIRST}((I * F)) \wedge (I * F) \not\Rightarrow^* \epsilon$ e $* \notin \text{FIRST}(F) \wedge F \not\Rightarrow^* \epsilon$
- ▶ $\delta(I, x) = I \rightarrow F$, pois $x \in \text{FIRST}(F)$
- ▶ $\delta(F, () = \text{SYNTAXERROR}$, pois $(\notin \text{FIRST}(x) \wedge x \not\Rightarrow^* \epsilon$
- ▶ $\delta(F,)) = \text{SYNTAXERROR}$, pois $) \notin \text{FIRST}(x) \wedge x \not\Rightarrow^* \epsilon$
- ▶ $\delta(F, *) = \text{SYNTAXERROR}$, pois $* \notin \text{FIRST}(x) \wedge x \not\Rightarrow^* \epsilon$
- ▶ $\delta(F, x) = F \rightarrow x$, pois $x \in \text{FIRST}(x)$

Análise sintáctica LL(1) – um exemplo

Considerando então as regras:

1. $I \rightarrow (I * F)$

2. $I \rightarrow F$

3. $F \rightarrow x$

constroi-se a tabela de *parsing*:

δ	(	)	*	x
I	1	SE	SE	2
F	SE	SE	SE	3

O analisador processa uma palavra (e.g., $(x * x)$) da seguinte forma:

Entrada	Pilha	Regra
$(x * x)$	I	$\delta(I, () = 1$
$(x * x)$	$(I * F)$	
$x * x)$	$I * F)$	$\delta(I, x) = 2$
$x * x)$	$F * F)$	$\delta(F, x) = 3$
$x * x)$	$x * F)$	
$*x)$	$*F)$	
$x)$	$F)$	$\delta(F, x) = 3$
$x)$	$x)$	
$)$	$)$	

Análise sintáctica LL(1) – um exemplo

Mais exemplos de processamento de palavras.

Entrada	Pilha	Regra
(x*	I	$\delta(I, () = 1$
(x*	(I * F)	
x*	I * F)	$\delta(I, x) = 2$
x*	F * F)	$\delta(F, x) = 3$
x*	x * F)	
*	*F)	
	F)	

Como a entrada está vazia mas a pilha não (e não há regra $F \rightarrow \epsilon$), a palavra não pertence à linguagem.

Entrada	Pilha	Regra
x * x	I	$\delta(I, x) = 2$
x * x	F	$\delta(F, x) = 3$
x * x	x	
*x		

Como a pilha está vazia mas a entrada não, a palavra não pertence à linguagem.

Analise sintáctica LL(1) – um exemplo

Mais exemplos de processamento de palavras.

Entrada	Pilha	Regra
(xx)	I	$\delta(I, () = 1$
(xx)	(I * F)	
xx)	I * F)	$\delta(I, x) = 2$
xx)	F * F)	$\delta(F, x) = 3$
xx)	x * F)	
x)	*F)	

Como o primeiro símbolo da palavra na entrada não coincide com o símbolo não terminal no topo da pilha, a palavra não pertence à linguagem.

Entrada	Pilha	Regra
(x*)	I	$\delta(I, () = 1$
(x*)	(I * F)	
x*)	I * F)	$\delta(I, x) = 2$
x*)	F * F)	$\delta(F, x) = 3$
x*)	x * F)	
*)	*F)	
)	F)	$\delta(F,)) = \text{SE}$

Como se encontrou um erro sintático, a palavra não pertence à linguagem.