



Parser LL(1)



Compiladores – Engenharia Informática (3º ano), 2010-2011
Daniela da Cruz

Tipos de Parser

- ▶ Top-down
 - ▶ Parser recursivo descendente
 - ▶ **LL(1)**
- ▶ Bottom-up
 - ▶ LR(0)



Parser LL(1)

- ▶ É um parser top-down, preditivo e iterativo
- ▶ Um parser LL é chamado LL(k) se usa k tokens de lookahead (todos em avanço) quando faz o reconhecimento de uma frase, para decidir a produção a seguir.



Parser LL(1)

- ▶ O parser LL(1) consiste em:
 1. Uma *stack* para armazenar os terminais e não terminais da gramática que faltam ser reconhecidos
 2. Uma *tabela de parsing* que nos informa qual a produção (se existir alguma) a aplicar tendo em conta o símbolo no topo da stack e o próximo token do texto de entrada



Parser LL(1)

Quando o reconhecimento de uma frase começa, a *stack* contém à partida dois símbolos:

- ▶ O símbolo inicial da gramática
- ▶ O símbolo \$ (terminal especial para indicar o fim da stack e o fim do texto de entrada)



Parser LL(1) – Construção da Tabela de Parsing

Em cada momento, a consulta da tabela de parsing deverá indicar qual a ação a tomar: `ac`, `error`, `skip`, `#p_i`.

A construção da tabela de parsing implica o cálculo dos *lookahead* de cada uma das produções da gramática.



Lookahead

Definition 2 (Lookahead(1)) *The Lookahead(1) set of Terminal symbols of a production $p \in P$ is defined in the following way:*

$$\text{lookahead}(A \rightarrow \alpha) = \text{First}(\alpha) \cup \begin{cases} \emptyset & , \alpha \not\Rightarrow^* \epsilon \\ \text{Follow}(A) & , \alpha \Rightarrow^* \epsilon \end{cases}$$



First e Follow

Definition 3 (First(1)) *The First(1) set of Terminal symbols of a Terminal , Non-terminal symbol or a String (a sequence of symbols) is defined follows:*

1. $First(\epsilon) = \emptyset$

2. $First(t) = \{t\}, t \in T$

3. $First(A) = \bigcup_{A \rightarrow \beta_i} First(\beta_i), A \in N$

4. $First(\alpha) = First(X) \cup \begin{cases} \emptyset & , X \not\Rightarrow^* \epsilon \\ First(\alpha') & , X \Rightarrow^* \epsilon \end{cases} , \alpha = X\alpha'$

Definition 4 (Follow(1)) *The Follow(1) set of Terminal symbols of a Non-Terminal symbol is defined by:*

$$Follow(A) = \bigcup_{Y \rightarrow \alpha A \beta} (First(\beta) \cup \begin{cases} \emptyset & , \beta \not\Rightarrow^* \epsilon \\ Follow(Y) & , \beta \Rightarrow^* \epsilon \end{cases})$$



First e Follow (2)

Em termos informais,

- ▶ $\text{First}(A)$ é o conjunto de terminais que podem aparecer na primeira posição de qualquer frase derivada de A .
- ▶ $\text{Follow}(A)$ é a união dos $\text{First}(\beta)$ onde β é qualquer sequência de símbolos que aparece imediatamente a seguir a A no lado direito de uma produção.



Parser LL(1) – Tabela de Parsing

```
for  $t \in T$ 
  for  $n \in N$ 
     $\text{tab}[n, t] = \text{error}$ 

for  $(A \rightarrow \beta) \in P$ 
  for  $t \in \text{lookahead}(pi: A \rightarrow \beta)$ 
     $\text{tab}[A, t] = pi$ 
```

NUT / δ	t_1	t_2	...	t_n	\$
X_1					
X_2					
...					
X_n					
t_1					
t_2					
...					
t_n					
\$					aceita



Conflito LL(1)

- Uma Gramática G tem um **Conflito LL(1)** se

Para um mesmo símbolo não-terminal X se verificar que os LookAhead de duas produções alternativas de X não é disjunto

$$la(X \rightarrow \alpha_1) \cap la(X \rightarrow \alpha_2) \neq \text{VAZIO}$$

