

LL(1) Parsing

Daniela da Cruz Pedro Rangel Henriques

{danieladacruz,prh}@di.uminho.pt
Departamento de Informática, CCTC
Universidade do Minho

LL(k) - "Left to right, Leftmost derivation with **k** lookahead symbols"

An LL parser is called an LL(k) parser if it uses **k** tokens of lookahead when parsing a sentence. If such a parser exists for a certain grammar and it can parse sentences of this grammar without backtracking then it is called an LL(k) grammar. Of these grammars, LL(1) grammars, although fairly restrictive, are very popular because the corresponding LL parsers only need to look at the next token to make their parsing decisions. We consider, in this document, $k=1$.

1 LL1 Definitions

Definition 1 (LL(1) Condition) A grammar $G = (T, N, S, P)$ satisfies LL(1) condition if and only if:

$$\forall A \rightarrow \alpha_1, A \rightarrow \alpha_2 : \text{lookahead}(A \rightarrow \alpha_1) \cap \text{lookahead}(A \rightarrow \alpha_2) = \emptyset$$

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

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

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. $\text{First}(\epsilon) = \emptyset$
2. $\text{First}(t) = \{t\}$, $t \in T$
3. $\text{First}(A) = \bigcup_{A \rightarrow \beta_i} \text{First}(\beta_i)$, $A \in N$
4. $\text{First}(\alpha) = \text{First}(X) \cup \begin{cases} \emptyset & , X \not\Rightarrow^* \epsilon \\ \text{First}(\alpha') & , X \Rightarrow^* \epsilon \end{cases}$, $\alpha = X\alpha'$

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

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

2 Algorithm to construct the table LL(1)

function build_PT($\alpha \rightarrow$ Parsing Table)

$$\left\{ \begin{array}{ll} \text{for } t \in T & \text{for } n \in N \\ & \text{end_for} \\ \text{end_for} & \alpha'[n, t] \leftarrow \text{error} \\ \text{for } (A \rightarrow \beta) \in P & \text{for } t \in \text{lookahead}(A \rightarrow \beta) \\ & \text{end_for} \\ \text{end_for} & \alpha'[A, t] \leftarrow (A \rightarrow \beta) \end{array} \right.$$