Analyseur LL(1)

Un analyseur LL(1) est déterministe et piloté par le sommet de pile :

- si terminal a : lecture de a (ou erreur) ;
- si non terminal X avec a sous la tête de lecture : expansion selon $Table[X, a]$.

Et si $Table[X, a]$ contient plus d'une production ?

Non-déterminisme :

- la grammaire n'est pas LL(1) ;
- on ne peut pas appliquer une analyse LL(1).

Caractérisation d'une grammaire LL(1)

Caractérisation par table d'analyse : une grammaire est LL(1) si chaque case contient exactement une production ou erreur.

Caractérisation « par contre-exemple » : une grammaire n'est pas LL(1) s'il existe 2 productions $X \to \alpha$ et $X \to \beta$ telles que :

1. soit $Premier(\alpha) \cap Premier(\beta) \neq \emptyset$;
   Ex : $S \to aS \mid A,\ A \to a$
2. soit $Eps(\alpha) = vrai$ et $Premier(\beta) \cap Suivant(X) \neq \emptyset$;
   Ex : $S \to aS \mid Ab,\ A \to \epsilon \mid b$
3. soit $Eps(\alpha) = vrai$ et $Eps(\beta) = vrai$ (la grammaire est ambiguë)
   Ex : $S \to A \mid B,\ A \to \epsilon,\ B \to \epsilon$

LL(1) et ambiguïté

Une grammaire LL(1) n'est pas ambiguë.

Une grammaire ambiguë n'est pas LL(1).

Cas classiques non LL(1)

Dans les cas suivants, la grammaire n'est pas LL(1) :

- ambiguïté ;
- récursivité gauche : $A \rightarrow Aa \mid \epsilon$;
  - intuitivement récursivité infinie de `A()`.
- non factorisation gauche : $S \rightarrow aA \mid aB$

Solutions : :

- factorisation à gauche (parfois) ;
- suppression de la récursivité gauche (parfois) ;
- utiliser un générateur de parser plus puissant !