

TD 9 - Analyse LL

Qu 1. On considère la grammaire $G = (\{a, b, c\}, \{S\}, S, \{S \rightarrow aSa \mid bSb \mid c\})$.

- Donner quatre mots engendrés par la grammaire.
- Quel est le langage engendré par cette grammaire ?
- Construire la table d'analyse $LL(1)$ de la grammaire G .
- Simuler l'analyse prédictive pour les mots `bacab` et `bcba`. En cas de succès, donner l'arbre d'analyse syntaxique et la dérivation gauche résultants.

Qu 2. Soit la grammaire $G = (\{a, b\}, \{S\}, S, \{S \rightarrow aSa \mid bSb \mid \varepsilon\})$.

- Donner quatre mots engendrés par la grammaire.
- Quel est le langage engendré par cette grammaire ?
- En construisant sa table d'analyse, montrer que G n'est pas $LL(1)$.

Qu 3. Soit la grammaire G d'axiome `INSTR`, de terminaux `{ ; , if , (,) , begin , bla , end , else }` et variables `{ INSTR , EXPR , BLOCK , SEQ , REST }`, et dont l'ensemble des règles P est :

$$\left\{ \begin{array}{l} \text{INSTR} \rightarrow \text{EXPR} ; \mid \text{BLOCK} \mid \text{if (EXPR) BLOCK REST} \\ \text{EXPR} \rightarrow \text{bla} \mid \varepsilon \\ \text{BLOCK} \rightarrow \text{begin SEQ end} \\ \text{SEQ} \rightarrow \text{INSTR SEQ} \mid \varepsilon \\ \text{REST} \rightarrow \text{else INSTR} \mid \varepsilon \end{array} \right.$$

- Quelles sont les variables effaçables ?
- Donner la table des ensembles `Premier`.
- On note `$` le terminal spécial qui marque la fin des mots à analyser. On dispose de la table des ensembles `Suivant`

Variable	Suivant
<code>INSTR</code>	<code>\$; if bla begin end</code>
<code>EXPR</code>	<code>;)</code>
<code>BLOCK</code>	<code>\$; if bla begin end else</code>
<code>SEQ</code>	<code>end</code>
<code>REST</code>	<code>\$; if bla begin end</code>

Indiquer comment est déterminé l'ensemble `Suivant(INSTR)` et l'ensemble `Suivant(BLOCK)`.

- Construire la table d'analyse $LL(1)$ de G . Distinguer les entrées qui mettent en jeu les ensembles `Suivant` de celles qui mettent en jeu les ensembles `Premier`.
- Faire l'analyse du mot suivant : `begin bla ; ; end $`
- Peut-on remplacer la production `SEQ → INSTR SEQ` par la production `SEQ → SEQ INSTR` ? Justifier.
- Peut-on supprimer la variable `REST` en remplaçant les productions `REST → else INSTR`, `REST → ε`, `INSTR → if (EXPR) BLOCK REST` par les productions `INSTR → if (EXPR) BLOCK else INSTR` et `INSTR → if (EXPR) BLOCK` ? Justifier.
- Telle qu'elle est écrite, G n'accepte pas les deux mots suivants : `if (bla) bla ; $` et `if (bla) bla ; else bla ; $`.
L'idée est alors de remplacer la production `INSTR → if (EXPR) BLOCK REST` par la production `INSTR → if (EXPR) INSTR REST`.
Que se passe-t-il alors pour G ? Peut-on y remédier ?