

Résumé — Analyse Syntaxique

Techniques de Compilation – Chapitre 2 (2023-2024)

1. Objectif et contexte

- **But** : transformer le flot d'unités lexicales en *arbre abstrait* :contentReference[oaicite :1]index=1.
- Deux grandes familles d'analyse : *descendante* (top-down) et *ascendante* (bottom-up) :contentReference[oaicite :2]index=2.
- Convention : lecture de gauche à droite, un symbole à la fois (notation **LR**).

Rôle de l'analyseur — vérifier la conformité, signaler les erreurs, produire l'arbre et alimenter la table des symboles

2. Analyse descendante

2.1 Avec rebroussement (backtracking)

Développement récursif + retour arrière possible ; peu utilisé car coûteux en temps :contentReference[oaicite :3]index=3.

2.2 Sans rebroussement (analyse prédictive)

Nécessite une grammaire :

- *non ambiguë*,
- *factorisée à gauche*,
- *non récursive à gauche*.

Grammaire LL(1) — table d'analyse sans entrées multiples (Left-to-right, Leftmost derivation, 1 look-ahead)

3. Transformations indispensables

3.1 Factorisation à gauche

$$A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \dots \implies A \rightarrow \alpha A', A' \rightarrow \beta_1 \mid \beta_2 \mid \dots$$

3.2 Suppression de la récursivité gauche

$$A \rightarrow A\alpha \mid \beta \implies A \rightarrow \beta A', A' \rightarrow \alpha A' \mid \varepsilon$$

3.3 Algorithme généralisé (récursivité cachée)

Ordre des non-terminaux A_1, \dots, A_n puis double boucle i, j pour substituer et éliminer :contentReference[oaicite :4]index=4.

4. Ensembles *FIRST* et *FOLLOW*

- **FIRST**(X) : terminaux pouvant ouvrir une dérivation de X (éventuellement ε).
- **FOLLOW**(A) : terminaux pouvant suivre A dans une dérivation.
- Utiles pour remplir la table $M[A, a]$ de l'analyseur prédictif.

Règle	Impact sur <i>FIRST</i> / <i>FOLLOW</i>
$A \rightarrow \alpha B \beta$	$\text{FOLLOW}(B) \supseteq \text{FIRST}(\beta) \setminus \{\varepsilon\}$
$A \rightarrow \alpha B$ ou $(\varepsilon \in \text{FIRST}(\beta))$	$\text{FOLLOW}(B) \supseteq \text{FOLLOW}(A)$

5. Table d'analyse prédictive

$$M[A, a] = \begin{cases} A \rightarrow \alpha & \text{si } a \in \text{FIRST}(\alpha) \\ A \rightarrow \alpha & \text{si } \varepsilon \in \text{FIRST}(\alpha) \wedge a \in \text{FOLLOW}(A) \\ \text{erreur} & \text{sinon} \end{cases}$$

Algorithme non récursif : pile initiale $E \$$, boucle sur le sommet X et le symbole d'entrée a : contentReference[oaicite :5]index=5.

6. Exemple complet « expression arithmétique »

$$\begin{array}{ll} E \rightarrow TE' & E' \rightarrow +TE' \mid \varepsilon \\ T \rightarrow FT' & T' \rightarrow *FT' \mid \varepsilon \\ F \rightarrow (E) \mid \text{id} & \end{array}$$

- *FIRST/FOLLOW* déjà calculés dans les diapos 30-31.
- Table LL(1) sans conflit \Rightarrow grammaire LL(1) valide.
- Traçage de la pile pour l'entrée $\text{id} + (\text{id} * \text{id}) \$$ (cf. diapos 36-39).

7. Checklist finale

- Expliquer rôle et interface de l'analyseursyntaxique avec le lexer.
- Appliquer factorisation suppression récursivité gauche.
- Calculer *FIRST* et *FOLLOW*, construire la table LL(1).
- Simuler l'algorithme prédictif non récursif sur un mot donné.
- Connaître les critères d'une grammaire LL(1).