Université Larbi Ben Mhidi

Niveau : 1ère année Master IA

Département MI

Module : TLN

Année : 2024-2025

Série de TP 4 : Analyse syntaxique

Exercice 01 :

1. Définissez une grammaire formelle simple pour la génération des phrases dans un langage fictif :

- **Exemple de grammaire :**
  - S → NP VP
  - NP → Det N
  - VP → V NP
  - Det → "le" | "la"
  - N → "chat" | "souris"
  - V → "mange" | "chasse"

2. Écrivez un script pour générer toutes les phrases possibles à partir de cette grammaire.
3. Quelle est la taille de l'espace des phrases générées si on ajoute un adjectif (Adj) avant chaque nom ?

Exercice 02

Implémenter un analyseur syntaxique descendant (top-down parsing) pour la grammaire suivante :

S → NP VP

NP → Det N

VP → V NP | V NP PP

PP → P NP

Det → "un" | "le"

N → "programme" | "code" | "fichier"

V → "compile" | "exécute" | "lit"

P → "dans" | "depuis"

Exercice 3 : Analyse syntaxique probabiliste

Dans le domaine informatique, les commandes ou les instructions suivent souvent des structures grammaticales prévisibles. Par exemple :

- **"le programme exécute un script"**
- **"un ancien script ouvre le programme dans un dossier"**

Vous allez modéliser une grammaire simplifiée qui permet de représenter ces types de phrases, en attribuant des probabilités aux différentes règles pour refléter la fréquence d'usage.

Les phrases suivent une structure basique : **Sujet (NP) + Verbe (VP)**.

1. Ajoutez des détails tels que des adjectifs (Adj), des prépositions (P), et des compléments (PP).
2. Attribuez des probabilités à chaque règle.
3. Implémentez une grammaire PCFG dans Python en utilisant la bibliothèque **NLTK**.