

---

**Exercice 1. Théorie des langages : rappels**

---

- Définitions
  - qu'est-ce qu'un alphabet ? comment peut-il être défini ?
  - qu'est-ce qu'un langage ? comment peut-il être défini ?
  - qu'est-ce qu'une grammaire ? comment peut-elle être définie ?
- La disparition. Soit l'alphabet  $\Sigma = \{a, \dots, d, f, \dots, z\}$  précisez pour chacun des mots suivants si il est dans le langage  $\Sigma^*$ , et s'il ne l'est pas, complétez l'alphabet en précisant les symboles nécessaires afin que le mot soit dans  $\Sigma^*$  : Bonjour, €, bonjour, message, voici un message.
- Soit la grammaire  $G$  (irrespectueuse de la langue française) suivante :
  - Phrase  $\rightarrow$  SujetVerbe Complément | Verbe
  - SujetVerbe  $\rightarrow$  SujetVerbe SujetVerbe | SujetVerbe Adverbe | SujetVerbe Verbe | tkt | g | €
  - Verbe  $\rightarrow$  Adverbe Verbe | Verbe Nom | fé
  - Adverbe  $\rightarrow$  po
  - Complément  $\rightarrow$  Conjonction Pronom | Proposition Pronom
  - Conjonction  $\rightarrow$  kom
  - Pronom  $\rightarrow$  toa | moa
  - Proposition  $\rightarrow$  ac
  - Nom  $\rightarrow$  gaf
  - Donnez si possible les arbres de dérivation des phrases *tkt g po fé kom toa* et *fé gaf*.

---

**Exercice 2. Compilation**

---

- Rappelez les différentes phases d'analyse d'un compilateur.
- Donnez une spécification d'analyse lexicale d'une phrase du langage naturel, où les lexèmes sont soit des suites de caractères (mots), soit des caractères de ponctuation (., ,, ;, !, ?).
- Donnez une spécification d'analyse syntaxique de la phrase pour laquelle celle-ci doit nécessairement se terminer par ., ! ou ?.
- Donnez une spécification d'analyse lexicale d'un micro-programme en langage C dont les lexèmes sont les identifiants des variables, les types de bases, les opérateurs arithmétiques (+, -, x, / et %) et d'affectation (=), les nombres (entiers et flottants).
- Donnez une spécification d'analyse syntaxique du micro-programme en langage C comprenant : déclarations de variables de types de base (int, float), affectations, expressions arithmétiques (avec les opérateurs précédents).

Un exemple de micro-programme à analyser en vue de sa compilation est le suivant :

```
int result;
float a;
float b;
a=0.2;
b=0.4;
result=(0.2+0.4)/2;
```

---

**Exercice 3. Analyse lexicale avec flex**

---

- Implémentez en langage C la spécification d'analyse lexicale pour la phrase simple précédente et générez-la avec flex.
- Implémentez en langage C la spécification d'analyse lexicale pour le micro-programme écrit en C et générez-la.

---

**Exercice 4. Analyse syntaxique avec bison**

---

- Implémentez en langage C la spécification d'analyse syntaxique pour une phrase simple en langage naturel et générez-la avec bison.
- Implémentez en langage C la spécification d'analyse syntaxique pour le micro-programme et générez-la.