**TD1 : Analyse Syntaxique | Expressions régulières**

Exercice1 : **Opérations sur les Langages**

Soit deux ensembles : ***L = {A, B, ..., Z, a, b, ..., z}*** : ensemble des lettres de l'alphabet (majuscules et minuscules). ***C = {0, 1, ..., 9}*** : ensemble des chiffres.

À partir de (L) et (C), réponds aux questions suivantes :

1. Quel est le langage ***L ꓴ C*** ?

2. Quel est le langage ***L ꓵ C*** (concaténation de L et C) ?

3. Décris ***L3*** (puissance du langage **(*L*)** en termes de chaînes générées.

4. Décris ***C+*** (un ou plusieurs chiffres) et donne des exemples de chaînes générées.

5. Décris ***L(L ꓴ C)\**** (lettre suivie d'une chaîne de lettres ou de chiffres ou chaîne vide) et donne quelques exemples.

Exercice 2 : **Opérations sur les langages**

Soit ***L1={a, b}*** et ***L2 = {0, 1}.***

1. Donne les éléments de ***L1 L2*** *(concaténation de* ***L1*** *et* ***L2****)*

2. Donne les éléments de ***L1*** *ꓴ* ***L2***

3. Décris ***L1***\* (fermeture de Kleene) et donne quelques exemples de chaînes générées.

4. Que représente l’expression ***L1***\* ***L2*** ?

Exercice 3 **: Langages et opérations**

Soit les langages suivants définis sur l'alphabet ***Σ= {a, b}*** : Soit ***L1={a,aa,b}*** et ***L2 = {b,bb,ab}.***

Effectue les opérations suivantes et donne les éléments du langage résultant :

1. Donne les éléments de ***L1*** *ꓴ* ***L2***

2. Donne les éléments de ***L1 L2*** *(concaténation de* ***L1*** *et* ***L2****)*

3. Décris ***L1***\* (fermeture de Kleene) et donne quelques exemples de chaînes générées.

Exercice 4 **: Langage et expressions régulières**

Donner une expression régulière pour chacun des langages suivants sur l'alphabet ***L ꓴ C*** (lettres et chiffres) :

1. Les chaînes qui commencent par une lettre majuscule suivie d'une ou plusieurs lettres minuscules.

2. Les chaînes qui contiennent exactement trois chiffres.

3. Les chaînes qui commencent par une lettre, suivie d'une séquence de lettres et de chiffres, et qui se terminent par un chiffre.

Exercice5 : **Lexèmes sur un alphabet**

Donner une expression régulière qui décrit les lexèmes sur l'alphabet `{a, b, c}` qui commencent par `b`.

Exercice 6 : **Expressions régulières**

Écris des expressions régulières pour les langages suivants :

1. Les chaînes contenant au moins un `a`.

2. Les chaînes contenant un nombre pair de `a` (peu importe le nombre de `b`).

3. Les chaînes qui commencent par `a`, contiennent un nombre quelconque de `b`, et finissent par `a`.

Exercice 7 : **Langages et expressions régulières avancées**

Donne une expression régulière pour les langages suivants :

1. Les chaînes sur l'alphabet ***{a, b}*** contenant un nombre pair de `a`.

2. Les chaînes sur l'alphabet ***{a, b}*** contenant exactement deux `b`.

3. Les chaînes sur l'alphabet ***{0, 1}*** qui ne contiennent pas le sous-motif `00`.

Exercice 8 **: Langage de mot de passe**

Écris une expression régulière pour un mot de passe valide respectant les règles suivantes : Contient au moins 8 caractères, doit inclure au moins une lettre majuscule, une lettre minuscule, et un chiffre, ne doit pas contenir d'espace.

Exercice 9 **: Utilisation des ancrages**

**Énoncé** : Tu es en train de développer un compilateur, et tu souhaites valider des identifiants de variables. Ces identifiants doivent :

1. Commencer par une lettre (majuscule ou minuscule) ou un underscore (\_).
2. Être suivis par n'importe quel nombre de lettres, chiffres ou underscores.

Écris une expression régulière utilisant des **ancrages** pour valider les identifiants

Exercice 10 : **Utilisation des assertions**

Tu développes un analyseur lexical qui doit détecter si une variable est suivie d'un opérateur d'affectation (=) sans espace, et renvoyer une erreur si c'est le cas. La variable suit les mêmes règles que dans l'exercice précédent. Écris une expression régulière avec une **assertion négative** qui s'assure que la variable n'est **pas** suivie immédiatement de =.