TELECOM Nancy

Test 1 (1H30)

MI : Mathématiques Appliquées à l'Informatique

UNIVERSITÉ DE LORRAINE

Théorie des langages

La notation tiendra compte de la présentation et de la clarté de la rédaction.

Partie I - Généralités

★ Exercice 1: Code-O-Scopie

On rappelle qu'un langage non vide $C \subset A^+$ est un code si tout mot de C^* se décompose de manière unique comme produit (concaténation) de mots de C. On a les propriétés suivantes :

1. Tout sous-ensemble d'un code est un code.
2. Si C est un code, $\varepsilon \notin C$.
3. Si C est un code, C et $\bigcup_{n\geq 2} C^n$ sont deux ensembles disjoints.
4. Si tous les mots de C sont de même longueur non nulle, alors C est un code. On parle de code uniforme.
5. Si aucun mot de C n'est préfixe (resp. suffixe) d'autre mot de C alors C est un code. Dans ce cas C est qualifié de code préfixe (resp. suffixe).

Lorsque ces propriétés ne sont pas suffisantes pour montrer qu'un langage L est ou n'est pas un code, on a recours à l'algorithme de Sardinas Patterson suivant :

1. Initialisation : $U_0 = L^{-1}.L \setminus \{\varepsilon\}$
2. Itération : $U_{n+1} = U_n^{-1}.L \cup L^{-1}.U_n$
3. Condition d'arrêt : $\begin{cases} \varepsilon \in U_n & \Rightarrow \quad L \textit{ n'est pas un code} \\ \exists (i,j) \in \mathbb{N}^2 \ i \neq j \textit{ et } U_i = U_j & \Rightarrow \quad L \textit{ est un code} \end{cases}$

Parmi les langages suivants déterminer ceux qui sont des codes[1].

1. $L_1 = \{aa, ba, ab\}$ sur l'alphabet $A = \{a, b\}$
2. $L_2 = 0^*1^*$ sur l'alphabet $A = \{0, 1\}$
3. $L_3 = \{10110, 1101, 011, 01101\}$ sur l'alphabet $A = \{0, 1\}$
4. $L_4 = b^+a^+$ sur l'alphabet $A = \{a, b\}$

Partie II - Expressions régulières

★ Exercice 2: Les régulières d'Alexandre

Soit $A = \{a, b\}$ un alphabet. Ecrire des expressions régulières (rationnelles) qui dénotent les langages suivants :

1. $L_1 =$ l'ensemble des mots de A^* qui ne contiennent pas le mot bab
2. $L_2 =$ l'ensemble des mots de A^* contenant exactement deux a

1. On demande une démonstration pour chaque langage. Il est recommandé d'utiliser prioritairement les propriétés.