UNIVERSITÉ TOULOUSE III PAUL SABATIER	**Théorie de l'Information**	Année 2018 - 2019 L1 Semestre 2

TD 1 : Codes et Codage de caractères

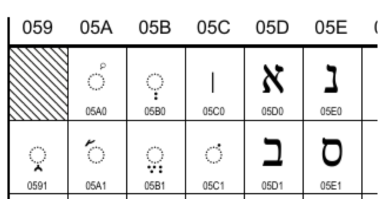
1 Codage de caractères

Exercice 1 Convertissez
- les nombres $(17)_{10}$, $(42)_{10}$, $(555)_{10}$ en base 16 et 2
- les nombres $(3A)_{16}$ et $(DEC)_{16}$ en base 10 et 2

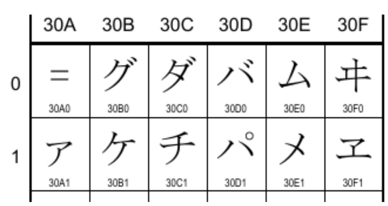
Exercice 2 Quelle partie de l'espace de code est utilisée par UTF-32 ?

Exercice 3 Quelle est la taille (en octets) d'un texte avec n caractères ASCII codé en format
1. UTF-8
2. UTF-16
3. UTF-32

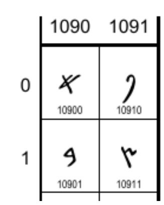
Exercice 4 Voici des extraits de la table de codage Unicode pour l'alphabet hébreu, japonais et phénicien.



(a) Hebrew



(b) Katakana



(c) Phoenician

Codez les caractères Alef, Bet et Nun (05D0, 05D1, 05E0), les caractères Gu et We (30B0, 30F1) et Alf, Bet (10900, 10901) en UTF-8, UTF-16, UTF-32.

2 Codes et codages

Exercice 5 Cochez les cases où $m_1 \preceq m_2$:

m_1 / m_2	01	010	110
01			
101			
10			
11			

Exercice 6 On dit qu'une relation R est un ordre si elle est
- réflexive : $\forall x.R(x,x)$
- antisymétrique : $\forall xy.R(x,y) \wedge R(y,x) \rightarrow x = y$
- transitive : $\forall xy.R(x,y) \wedge R(y,z) \rightarrow R(x,z)$

Pour la relation $\preceq$, on utilise une notation *infixe*, c.à.d. on écrit $x \preceq y$ au lieu de $\preceq (x,y)$.
Montrez que la relation $\preceq$ définie par $m \preceq m' =_{def} \exists r.m' = m \cdot r$ est un ordre.

Exercice 7 On définit trois codes c_1, c_2, c_3 pour un alphabet $A = \{a, b, c, d\}$ selon le tableau suivant :

x	$c_1(x)$	$c_2(x)$	$c_3(x)$
a	0	10	0
b	010	00	10
c	01	11	110
d	10	110	111

Montrez que :
— c_1 est injectif, mais son extension homomorphe c_1^* ne l'est pas.
— c_2 n'est pas un code préfixe, mais son extension homomorphe c_2^* est unique
— c_3 est un code préfixe

Exercice 8 Montrez formellement que tout codage c^* qui est l'extension homomorphe d'un code préfixe c est injectif.

Exercice 9 Est-ce que le code Morse est injectif / un code préfixe ?

Exercice 10 Appliquez l'algorithme `arbre_dec` aux codages c_1 et c_2 de la table suivante.

x	$c_1(x)$	$c_2(x)$	$c_3(x)$
a	00	01	0
b	01	11	10
c	10	00	110
d	11	001	1110

Si la construction de l'arbre échoue, identifiez les causes. Est-ce que vous pouvez proposer des codages qui évitent le problème ?

Exercice 11 Appliquez l'algorithme `tab_cod` aux arbres de décodage obtenus dans l'exercice 10 et vérifiez que vous obtenez bien les tables d'origine.

Exercice 12 Pourquoi est-ce que l'algorithme `tab_cod` termine ?

Exercice 13 ***(Devoir maison)***
Analyse de l'algorithme `arbre_dec` :

1. Quels problèmes se poseraient pour un algorithme de décodage si l'arbre n'était pas un arbre binaire (mais si un noeud intérieur pouvait avoir un seul successeur) ?
2. Un invariant de `arbre_dec` est qu'il prend la représentation d'une table `tab` d'un codage préfixe. Démontrez que cet invariant est maintenu par les appels récursifs, donc, que

$$\{(c, m) | (c, 0 \cdot m) \in tab\}$$

représente bien une table d'un codage préfixe (et pareil pour $1 \cdot m$).
3. Démontrez que si $(c, []) \in tab$, alors il n'est pas possible d'avoir un $(d, m) \in tab$, pour un $c \neq d$.
4. Démontrez que l'algorithme termine.

Exercice 14

1. Utilisez l'inégalité de Kraft pour déterminer s'il est possible de construire un code préfixe pour les caractères $a \dots d$ avec les longueurs de code suivants :

caractère	a	b	c	d
longueur	1	2	2	3

2. Quelle serait votre réponse si on admet un code de longueur 3 pour le caractère b ? Proposez effectivement un code.

Exercice 15

1. Une entreprise veut installer un système téléphonique interne où les 5 membres du directoire ont un numéro à un seul chiffre (de 0 à 9) et les 80 autres employés un nombre à deux chiffres. Est-ce possible ?
2. Serait-il possible d'avoir des numéros à deux chiffres pour les membres du directoire et de trois chiffres pour les autres employés ? Faites une proposition concrète.

A noter : L'inégalité de Kraft se généralise d'un code binaire à un code n-aire (alphabet à n chiffres) comme suit : Il existe un code préfixe n-aire avec k codes $u_1 \dots u_k$ si et seulement si

$$\sum_{i=1}^{k} n^{-|u_i|} \leq 1$$

où $|u_i|$ est la longueur du code u_i.