UNIVERSITÉ TOULOUSE III PAUL SABATIER

Théorie de l'Information

Année 2018 - 2019
L1 Semestre 2

Test d'auto-évaluation pour le Contrôle Partiel

Quelques remarques concernant le contrôle continu et ce test :

— Le CC aura lieu le 18 mars à 7h45 et sera d'une durée de 1h30 (2h pour les tiers-temps).
— Le test est considérablement plus long que l'examen, mais le test contient le genre de questions auxquelles il faut s'attendre dans l'examen.
— Le CC couvre les chapitres du cours 1 (Codage) et 2 (Information selon Shannon et codage optimal), mais pas les deux autres chapitres.
— Aucun document n'est autorisé dans l'examen. Les définitions essentielles seront résumées en annexe. L'annexe est exactement la même que vous trouvez à la fin de ce test.
— Une calculette classique (sans moyen de communication) est autorisée. Tout autre appareil électronique est interdit.

L'examen comportera deux parties :

— une feuille de QCM que vous devez renseigner selon les indications sur cette feuille. Elle est à rendre avec la copie classique. Nous vous recommandons d'effectuer vos calculs sur la copie classique en cas de réclamation.
— une copie classique où vous rédigez le reste des questions.

Le teste fonctionne selon le même principe. Rendez la partie QCM à votre enseignant de cours-TD le plus tôt possible. Nous essayerons alors de scanner la feuille et vous faire un retour avant l'examen.

1 Codage de caractères

La première partie est sur la feuille de QCM.

Exercice 1 Le codage propriétaire Windows 932 code le caractère latin D avec un octet (`44` hexadécimal). Les caractères cyrilliques D et T sont codés avec deux octets, respectivement `8444` et `8484` hex.

1. Supposez que vous cherchez le caractère latin D dans un texte codé avec Windows 932. Quels sont les problèmes qui se posent pour trouver la première position dans le texte ?
2. Montrez, au contraire, que le codage UTF-8 est un codage auto-synchronisant (*"self-synchronizing"*) : pour chaque octet d'un codage, vous savez s'il s'agit du début d'un codage, et si ce n'est pas le cas, vous savez de combien de caractères il faut reculer au maximum pour trouver le début.
3. Supposez qu'un seul octet peut être altéré accidentellement lors d'un transfert d'un message codé. Combien de caractères du message original peuvent être affectés par cette modification sans que le récepteur s'en aperçoive (dans le cas d'un codage en UTF-8 / et en Windows 932) ?

BEGIN SOLUTION

1. Dans une séquence `8484 ... 8444`, il est difficile à savoir si le dernier `44` code le caractère latin D ou `8444` code le cyrillique D. En plus, pour savoir s'il s'agit d'une séquence `TT..lD` ou `TTcD` (où `lD` est un `D` latin et `cD` un `D` cyrillique), il faut réculer arbitrairement dans le texte.
2. En UTF-8, le début d'un codage est marqué par `0` ou par `110`, `1110` ou `11110`. Tout autre octet (commençant avec `10`) se trouve au milieu, il faut réculer d'au plus 3 positions.
3. Une altération du préfixe entraîne une erreur qui est observable (le nombre d'octets commençant avec `10` ne correspond pas aux prédictions de l'octet en tête). Pour être inobservable, l'altération doit se produire à l'intérieur d'un octet (marqué `xxx` dans la table de codage).

END SOLUTION

2 Codes

Exercice 2 On définit les codages c_1, c_2 pour un alphabet $A = \{a, b, c, d, e\}$ selon le tableau suivant :

x	$c_1(x)$	$c_2(x)$
a	10	100
b	111	111
c	01	010
d	110	110
e	101	101

Quels codes sont injectifs / des codes préfixes ? Quels codes sont décodables sans ambiguïté ? Donnez des justifications pour votre réponse.

BEGIN SOLUTION

1. c_1 n'est pas préfixe et en fait pas décodable de manière unique : $c_1(ae) = c_1(ec)$
2. c_2 est préfixe, donc unique et injectif.

END SOLUTION

3 Inégalité de Kraft

Exercice 3 (Inégalité de Kraft) Un collègue vous demande de construire un code préfixe binaire pour les caractères a, b, c, d, e, f respectivement avec des longueurs 2, 2, 3, 3, 3, 4.

1. Vérifiez tout d'abord, à l'aide de l'inégalité de Kraft, qu'un tel code peut exister.
2. Construisez effectivement un tel code.
3. Vous constatez que le code n'est pas optimal, parce que le code d'un caractère pourrait être raccourci. Quelle amélioration pouvez-vous proposer à votre collègue ?

BEGIN SOLUTION

1. **(1pt)** $2 * 2^{-2} + 3 * 2^{-3} + 1 * 2^{-4} = \frac{15}{16} < 1$
2. **(0.5pt)** Par exemple :

a	b	c	d	e	f
00	01	100	101	110	1110

3. **(0.5pt)** Le code 1111 n'est pas utilisé. On pourrait raccourcir le code pour le caractère f et lui attribuer le code 111.

END SOLUTION

Exercice 4 Le protocole d'internet (IP) utilise des *adresses IP* pour identifier des ordinateurs, par exemple pour acheminer un message électronique. L'adresse IP est donc l'analogue d'un numéro de téléphone pour ordinateurs.

Dans la version toujours actuelle du protocole (IPv4), l'adresse est uniformément composée de quatre octes (donc 32 bits). Jusqu'au milieu des années 1990, l'espace des adresses IP était sous-divisé en "classes", un mécanisme qui a depuis été abandonné.

Les classes sont les suivantes :

— *classe A :* adresses commençant avec le bit `0`. Le premier octet de l'adresse (y compris le `0`) identifie un *réseau*, les trois octets restants un *ordinateur* dans le réseau.

— *classe B :* adresses commençant avec la séquence de bits `10`, les deux premiers octets pour identifier le réseau, les deux autres pour l'ordinateur dans le réseau.
— *classe C :* adresses commençant avec la séquence de bits `110`, les trois premiers octets pour identifier le réseau, l'octet restant pour l'ordinateur dans le réseau.
— *classes D et E :* adresses commençant avec `1110` respectivement `1111`, pour des usages spécifiques.

Note : Un réseau correspond par exemple à l'ensemble des ordinateurs administrés par une entreprise et permet une gestion plus flexible des adresses au sein de celle-ci.

1. Pour les adresses IP suivantes, indiquez leur classe (A, B ou C). Quelle partie de l'adresse est l'identifiant du réseau, quelle partie est l'identifiant de l'ordinateur dans le réseau ?
   (a) `10111001.11010010.11110110.01011101`
   (b) `00110011.10110101.10110101.11110110`
   (c) `11010101.10110110.10001110.00111101`
2. Quel est le nombre de réseaux de classe A (B / C) ? N'oubliez pas que les préfixes identifiant les classes (à savoir, 0, 10 etc.) sont fixes et réduisent le nombre de réseaux.
3. Quel est le nombre maximal d'ordinateurs dans un réseau de classe A (B / C) ?
4. Un problème du mécanisme des classes était qu'il était trop rigide et ne permettait pas d'identifier un nombre suffisant de réseaux de grande et moyenne taille. Utilisez l'inégalité de Kraft pour montrer le suivant : il n'est pas possible de concevoir un schéma d'adressage préfixe permettant d'identifier en même temps : 2^7 réseaux différents avec un octet, 2^{15} réseaux différents avec deux octets et 2^{10} réseaux différents avec trois octets.

N.B. : Effectuez vos calculs avec des expressions de la forme 2^n, ne convertissez pas en décimal !

BEGIN SOLUTION

1. En gras, l'identifiant du réseau :
   (a) **`10111001.11010010`**`.11110110.01011101` classe B
   (b) **`00110011`**`.10110101.10110101.11110110` classe A
   (c) `110`**`10101.10110110.10001110`**`.00111101` classe C
2. A : 2^7, B : 2^{14}, C : 2^{21} (par exemple C : 24 bits pour identifier le réseau, moins 3 bits préfixe `110`)
3. A : 2^{24}, B : 2^{16}, C : 2^8
4. Pour qu'un tel schéma d'adressage existe, il faudrait que $s \leq 1$, pour $s = 2^7 * 2^{-8} + 2^{15} * 2^{-16} + 2^{10} * 2^{-24} = 1 + 2^{-14}$

END SOLUTION

4 Théorie de l'information et algorithme de Huffman

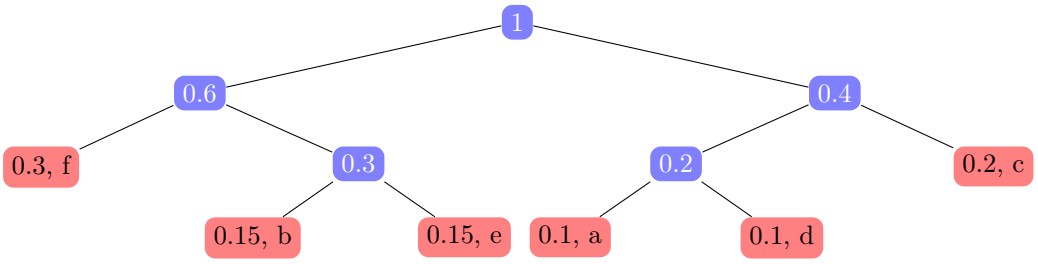
Exercice 5 (Arbres de Huffman) Une source d'information émet les caractères `a` ... `f` selon la distribution de probabilité suivante :

a	b	c	d	e	f
0.1	0.15	0.2	0.1	0.15	0.3

1. Calculez l'entropie de cette source d'information.
2. Construisez l'arbre de Huffman pour trouver un codage optimal des caractères.
3. Calculez la longueur moyenne du code ainsi obtenu et comparez avec l'entropie.

BEGIN SOLUTION

1. **(1pt)** $-(0.1 * log_2(0.1) + ...0.3 * log_2(0.3)) = 2.47$
2. **(1pt)** voir fig
3. **(1pt)** $0.1 * 3(a) + 0.15 * 3(b) + 0.2 * 2(c) + 0.1 * 3(d) + 0.15 * 3(e) + 0.3 * 2(f) = 2.5 > 2.47$



END SOLUTION

Exercice 6 Pour prendre en compte les difficultés d'une société de plus en plus analphabète, il a été décidé de limiter les numéros de téléphone à des séquences des chiffres 0, 1, 2, 3. Comme avant, la séquence 00 est le préfixe pour un appel international, qui est suivi du code national (actuellement 33 pour la France) composé uniquement des chiffres 1, 2, 3.

Étant donné la table suivante de quelques pays et leur population (en million d'habitants), concevez un système de codes nationaux qui est optimal au sens qu'il attribue des codes plus courts aux pays plus peuplés.

CN	IN	US	BR	PK	RU	DE	FR	UK	IT	ES
1380	1330	314	194	188	146	82	67	65	61	46

BEGIN SOLUTION

On construit un arbre ternaire, en regroupant à chaque fois les trois arbres avec les poids les plus faibles. Un code possible est alors :

CN	IN	US	BR	PK	RU	DE	FR	UK	IT	ES
1	2	32	311	312	331	332	333	3131	3132	3133

END SOLUTION

A Définitions du cours

A.1 Unicode

UTF-32 Chaque caractère représenté par un mot de 32 bits

UTF-16 Un ou deux mots de 16 bits, construits comme suit :

1. U+0000 ... U+D7FF et U+E000 ... U+FFFF :
   représentés par *un* mot de 16 bits avec la même valeur
2. U+D800 ... U+DFFF : ne sont pas des code points valides
3. U+10000 ... U+10FFFF : deux mots.
   Algorithme pour conversion de U+x_{16} :
   (a) Calculer $(x')_{16} = (x)_{16} - (10000)_{16}$
   (b) Représenter $(x')_{16}$ en binaire $(b')_2$ avec 20 chiffres : $(x')_{16} = (b')_2$
   (c) Scinder $(b')_2$ en deux mots v et w de 10 bits
   (d) Résultat du codage :
   — Premier mot : $(110110v)_2$
   — Deuxième mot : $(110111w)_2$

UTF-8 Codage entre 1 et 4 octets, selon la table :

Intervalle	*Octet 1*	*Octet 2*	*Octet 3*	*Octet 4*
U+0 ... U+7F	0xxxxxxx			
U+80 ... U+7FF	110xxxxx	10xxxxxx		
U+800 ... U+FFFF	1110xxxx	10xxxxxx	10xxxxxx	
U+10000 ... U+10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

A.2 Théorie de l'information

Entropie (selon Shannon) d'une source d'information avec des événements $\{x_i | i \in I\}$:

$$H =_{def} \sum_{i \in I} (-\log_2(P(x_i))) * P(x_i)$$

Longueur moyenne d'un ensemble (mot $\times$ probabilté) : $lnm(E) = \sum_{(m,p) \in E} |m| * p$

Notions de logarithme

— Définition du logarithme en base b : $\log_b(x) = y$ si et seulement si $x = b^y$.
— Calcul du logarithme binaire à l'aide du logarithme décimal : $\log_2(x) = \frac{\log_{10}(x)}{\log_{10}(2)}$