EISTI. **Année scolaire 2008-2009**

DÉPARTEMENT " INFORMATIQUE "

THÉORIE DE L'INFORMATION

Série d'exercices N^{o}3

PARTIE I. ENTROPIE D'UNE SOURCE. VERS LE CODAGE

Exercice 1 (Vers le codage de Shannon.)**.** L'objectif de cet exercice est de comprendre le fonctionnement d'une méthode de codage à partir d'un propriété d'entropie.

Soit X une source d'alphabet $\Omega_X = \{x_1, \ldots, x_n\}$ et de distribution de probabilités $P_X = \{p_1, \ldots, p_n\}$. Soit $1 \leq r < n$. On divise l'alphabet en deux sois-ensembles $A = \{x_1, \ldots, x_r\}$ et $B = \{x_{r+1}, \ldots, x_n\}$ de telle sorte que $\Omega_X = A \cup B$ et $A \cap B = \emptyset$. Soit $p = P(A)$ et $q = 1 - p = P(B)$. Alors on a la relation suivante (propriété du groupe) :

$$H(X) = H(p_1, \ldots, p_n) = H(p, 1-p) + pH\left(\frac{p_1}{p}, \ldots, \frac{p_r}{p}\right) + (1-p)H\left(\frac{p_{r+1}}{1-p}, \ldots, \frac{p_n}{1-p}\right)$$

On remarquera que $P_A = \left\{\frac{p_1}{p}, \ldots, \frac{p_r}{p}\right\}$ et $Q_B = \left\{\frac{p_{r+1}}{1-p}, \ldots, \frac{p_n}{1-p}\right\}$ définissent des distributions de probabilité respectivement sur A et B.

On peut interpréter cette propriété de la façon suivante. Supposons que nous avons une partition de l'ensemble de valeurs possibles de X en deux parties complémentaires, A et B. Alors l'incertitude moyenne (l'entropie) que nous avons sur X est composée de :

1. l'incertitude que nous avons sur le choix de l'une des deux parties A et B ; c'est $H(p, 1-p)$;
2. la moyenne des incertitudes associées à chacune des parties séparément ; c'est $pH\left(\frac{p_1}{p}, \ldots, \frac{p_r}{p}\right) + (1-p)H\left(\frac{p_{r+1}}{1-p}, \ldots, \frac{p_n}{1-p}\right)$.

Soit X une source d'alphabet $\Omega = \{1, 2, 3, 5, 4\}$ de distribution de probabilité $P = \{0.1, 0.2, 0.3, 0.15, 0.25\}$. Supposons que l'on doit deviner le symbole émis par la source et que l'on a droit de poser des questions binaires (réponses possibles "oui" et "non"). On cherche à construire la stratégie qui, en moyenne, permet de trouver la réponse en un nombre minimal de questions.

Remarquons qu'une question binaire induit sur l'ensemble Ω une partition en deux sous-ensembles A et B correspondants aux réponses "oui" et "non". Par exemple, si l'on demande "est que le nombre est impair ?" la partition sera $A = \{1, 3, 5\}$ et $B = \{2, 4\}$. Soit $p = P(A)$ la probabilité de la réponse "oui" à une question donnée.

1. Calculer l'entropie de X.
2. Quelle est l'information moyenne obtenue par la réponse à une question binaire ?

3. Quelle est l'information moyenne maximale ? Et pour quelle valeur de p est est atteinte ?
4. Quel est alors le meilleur choix de première question à poser ? **Indication :** cherchez à diminuer autant que possible votre incertitude, en posant une question.
5. Appliquer le même raisonnement récursivement pour choisir la meilleure deuxième question selon la réponse à la première. Continuer jusqu'à arriver à identifier chaque symbole.
6. Construire un arbre représentant la stratégie obtenue.
7. Calculer le nombre moyen de questions. Comparer à l'entropie de X.

Exercice 2 (Vers le codage de Huffman.)**.** Reprenons la même source que dans l'exercice 2 X d'alphabet $\Omega = \{1, 2, 3, 5, 4\}$ de distribution de probabilité $P = \{0.1, 0.2, 0.3, 0.15, 0.25\}$.

Supposons que l'on doit deviner le symbole émis par la source et que l'on a droit de poser des questions binaires (réponses possibles "oui" et "non"). On cherche à construire la stratégie qui, en moyenne, permet de trouver la réponse en un nombre minimal de questions.

Cette fois, nous recherchons une stratégie dans laquelle à chaque question on "élimine" un symbole. Autrement dit, chaque question est de la forme "Est ce que $X = x_i$? Soit $p_i = P(x_i)$.

1. Quelle est l'information moyenne obtenue par la réponse à une question de type $X = x_i$?
2. Appliquer la propriété de groupe de l'entropie dans ce cas particulier ?
3. Par quel symbole a-t-on intérêt de commencer : le plus probable ou le moins probable ?
4. Construire un arbre représentant la stratégie obtenue.
5. Calculer le nombre moyen de questions. Comparer à l'entropie de X.

PARTIE II. CODES SANS PRÉFIXE. PRÉPARATION THÉORIQUE

Nous allons considérer une source d'alphabet

$$\Omega = \{a, b, c, d, e, f, g, h, i, j\}$$

et un canal d'alphabet binaire $\{0, 1\}$.

Voici les définitions importantes.

Code. Un code binaire pour l'alphabet donné Ω de taille n est un ensemble de n mots binaires $\{m_1, \ldots, m_n\}$. Un code est régulier si les mots correspondants aux différents caractères de l'alphabet sont différents. Dans la suite nous étudions uniquement les codes réguliers.

Exemple. Pour notre alphabet un code régulier peut être

s_i	a	b	c	d	e	f	g	h	i	j
m_i	0	1	00	01	11	10	100	101	110	111

Code déchiffrable Un code binaire est déchiffrable si toute séquence de bits peut être décodée de façon unique.

Exemple. Le code ci-dessus n'est pas déchiffrable. En effet, la séquence de bits "110" peut être interprétée comme "i" ou comme "bf" ou même comme "ea".

Code sans préfixe. Un code est sans préfixe ou instantané si aucun mot code n'est le préfixe d'un autre. Le code ci-dessus n'est pas un code sans préfixe. En effet, le mot code 0 est le début des mots du code 00, 01. Le code suivant est sans préfixe :

s_i	a	b	c	d	e	f	g	h	i	j
m_i	1111	1110	110	1011	1010	100	011	000	010	001

Décodage pas à pas d'un code sans préfixe. Le principe de décodage est simple. Il suffit de lire la séquence codée de gauche à droite jusqu'à ce qu'on trouve un mot du code. On est alors certain que ce mot correspond sans ambiguïté à un seul caractère de l'alphabet. On enregistre le caractère et on recommence la lecture.

Exercice 3. Appliquez la procédure de décodage pas à pas à la séquence 10110000101010100.

PARTIE III. CODES SANS PRÉFIXE. PRÉPARATION ALGORITHMIQUE

Pour décrire le problème de codage et décodage nous allons créer une classe, appelée "Code". Vous trouverez sur AREL le fichier Code.java qui contient les premiers éléments de cette classe.

Un code sera défini par la donnée d'un entier positif n, la taille de l'alphabet, d'un tableau de caractères $alphabet$ et d'un tableau de chaînes de caractères, $motsCode$.

```
  public class Code
  {
private int n;                // taile de l'alphabet
private char alphabet[];      // tableau de symboles de l'alphabet
private String motsCode[];    //tableau de suites binaires, mots du code

    public Code(int k, char tab1[], String tab2[])       //constructeur
   {
n=k;
alphabet=tab1;
motsCode=tab2;
   }
...
  }
```

Deux méthodes sont déjà définies :

```
    public String CoderSymbole(char caract)
/* cette méthode recherche un caractère donné , caract, dans l'alphabet
 * et renvoie son mot code ou null si le caractère n'est pas dans l'alphabet
*/
{
   int i=0;
while(alphabet[i]!=caract & i<n)
```

```
{
i++;
}
 if(i<n)
{return motsCode[i];
}
 else return null;
}
public String DecoderMot(String mot)
/* cette méthode recherche un mot binaire donné, mot,
      *dans le  tableau des mots de  code,  et renvoie le symbole de l'alphabet
 *  correspondant ou null si le mot n'est pas trouvé
*/
{
int i=0;
while( i<n && !motsCode[i].equals(mot))
{

i++;
}
 if(i<n)
{
 return String.valueOf(alphabet[i]);
}
 else return null;
}
```

La première, CoderSymbole, permet de trouver le mot de code correspondant à un caractère de l'alphabet donné. La seconde, DecoderMot, recherche une séquence de bits donnée dans le tableau des mots du code. Elle renvoie le symbole correspondant au mot du code trouvé ou "null" au cas où la séquence ne correspond à aucun mot du code. Ces deux méthodes seront utiles pour la construction des algorithmes de codage et décodage des messages.

Votre objectif dans le projet de Pôle sera de compléter le classe Code avec deux méthodes : "Coder" et "Decoder".

Exercice 4. Ecrire **le pseudo-code** de l'algorithme de codage d'un message donné quelconque.

La méthode correspondante aura pour argument une chaîne de caractères, représentant le message formé à partir de l'alphabet. Elle doit renvoyer une chaîne de caractères contenant la suite de bits correspondant à la séquence codée.

Exercice 5. Ecrire l'algorithme de décodage d'une séquence de bits donnée **sous forme d'un pseudo-code**. La méthode correspondante aura pour argument une chaîne de caractères, représentant la séquence codée. Elle doit renvoyer une chaîne de caractères contenant le message décodé.