Chapitre 2 La codification et contrôle des données

1 Modèle entité association :

Le modèle entité/association a été proposé par *Chen*, en 1976 pour la modélisation des données et les liens existants entres elles, avec des concepts simples et efficaces. C'est une représentation naturelle du monde réel du SI à étudier. Il est bâti autour de trois concepts : entité, association et propriétés.

1.1. Définitions des concepts de base :

1.1.1. L'entité :

Une entité est une représentation, dans un SI, d'un objet matériel ou immatériel pourvu d'une existence propre.

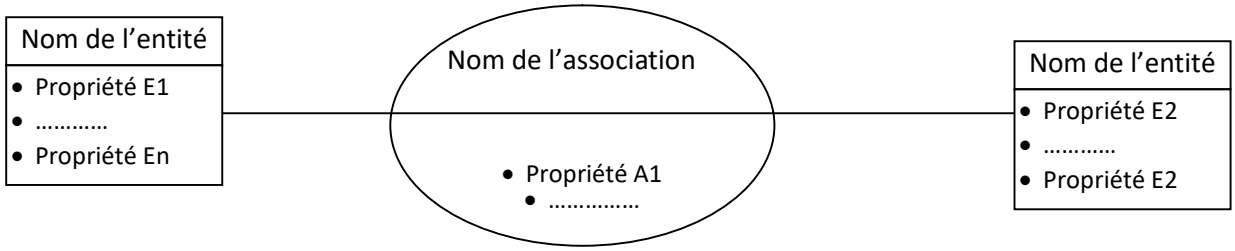
1.1.2. L'association :

L'association représente un lien entre les entités. Elle est dépourvue d'existence propre. Son existence des entités qu'elle met en interaction.

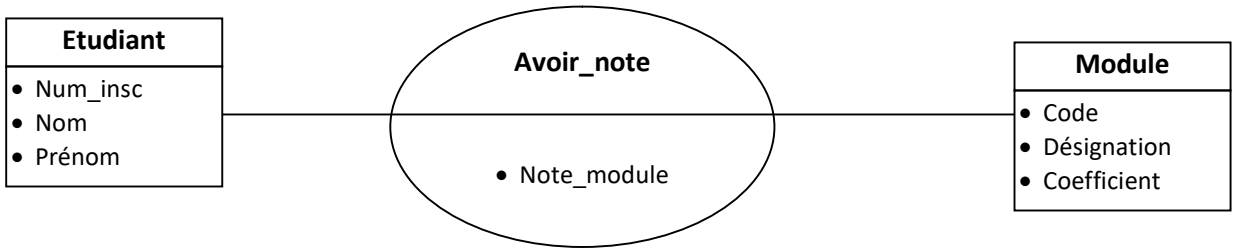
1.1.3. Propriété

Une propriété est une donnée élémentaire qui caractérise une entité ou une association.

Formalisme graphique :



Exemple :



1.1.4. Dimension d'une association :

La dimension d'une association désigne le nombre d'entités qui participent à cette association.

Exemple : La dimension de l'association *avoir_note* est deux.

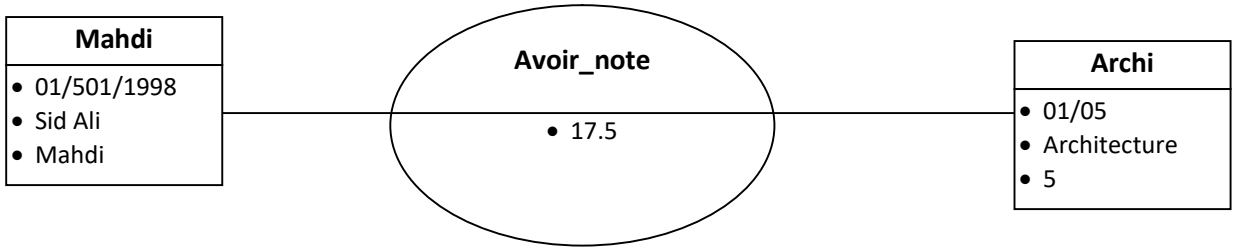
1.1.5. Occurrence d'une entité :

Une occurrence d'une entité est un élément individualisé, appartenant à cette entité. Elle est par l'attribution de valeurs aux différentes propriétés qui caractérisent un objet particulier, appartenant à cette entité.

1.1.6. Occurrence d'une association :

Une Occurrence d'une association est une association individualisée entre une et une seule occurrence de chaque entité participant à l'association.

Exemple :



1.1.7. Cardinalité d'une entité par rapport à une association :

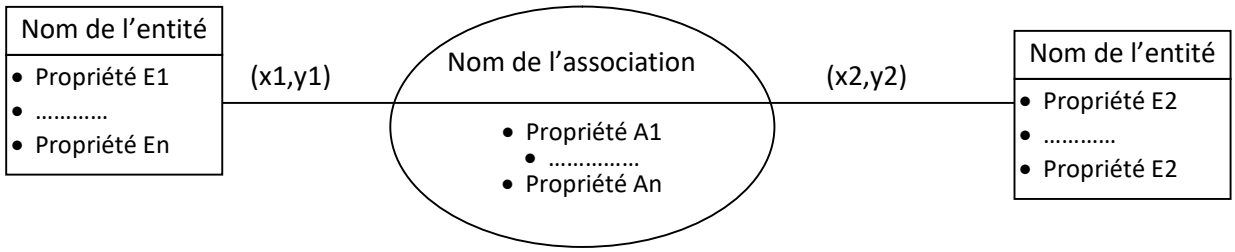
La cardinalité d'une entité X par rapport à une association avec une entité Y exprime le nombre d'occurrences de Y que l'on peut associer à une occurrence de l'entité X.

La cardinalité est exprimée par un couple de valeurs (x, y) tel que :

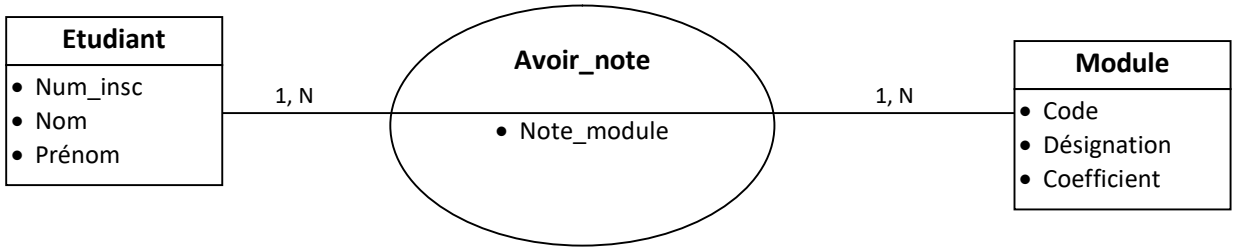
- x est le nombre minimum d'occurrences de Y que l'on peut associer à une occurrence de X
- y est le nombre maximum d'occurrences de Y que l'on peut associer à une occurrence de X

Les cas possibles de cardinalités sont (0,1), (1,1), (0,n), (1,n).

Formalisme graphique :



Exemple :



1.1.8. Identifiant d'une entité :

L'identifiant d'une entité est une propriété particulière (ou ensemble de propriétés) de l'entité qui permet d'identifier chaque occurrence de cette entité, de manière unique.

Formalisme graphique :

Nom de l'entité
Identifiant • Propriété E1 • …… • Propriété En

Exemple :

Etudiant
• **Num_insc** • Nom • Prénom

1.1.9. Identifiant d'une association :

L'identifiant d'une association est obtenu par la concaténation des identifiants des entités qui participent à cette association.

Exemple :

L'identifiant de l'association *avoir_note* est le coupe (num_insc, code).

2.1. Définition :

Les tables de décisions sont une représentation sous forme d'un tableau à plusieurs entrées. C'est outil qui permet de présenter de façon claire les règles de logique à utiliser pour décider des actions à exécuter en tenant compte des conditions à satisfaire.

Elles permettent de clarifier les textes souvent très difficiles à comprendre et surtout de déceler les anomalies et les oublis.

2.2. Structure des tables de décision

Une table de décision se présente sous forme d'un tableau divisé en deux parties :
La souche : elle comprend à son tour deux parties qui serrent à décrire :
Les conditions : ce sont les propositions à tester.
Les actions qui doivent être exécutées pour chaque ensemble de conditions.
Le corps : il correspond également à deux parties :
Les valeurs des conditions (entrées des conditions) : ce sont les valeurs que prendre une condition et construisent une règle.
Les valeurs des actions (entrées et actions) : ce sont les actions qui doivent être exécutées si la règle correspondante est satisfaite.

Les ***règles***

Conditions

Désignation des conditions	Entrées des conditions
Désignation des actions	Entrées des actions

Actions

Souche ***Corps***

Exemple : les activités de jeudi de monsieur X dépendront de deux conditions : le temps et la qualité des programmes de la télévision, ou le cinéma. Il ira à la pèche s'il fait beau, et ce, que les programmes de la télévision soient intéressants ou pas. Il regardera la télévision s'il ne fait pas beau et si les programmes sont intéressants. Il ira au cinéma s'il ne fait pas beau et si les programmes de la télévision ne l'intéressent pas.

La table de décision suivante traduit la logique de décision de monsieur X concernant ses activités de Jeudi.

			R1	R2	R3	R4
Conditions	C1	Fait-il bien ?	O	O	N	N
	C2	Les programmes de la télévision sont-ils intéressants ?	O	N	O	N
Actions	A1	Monsieur X va à la pèche	X	X		
	A2	Monsieur X regarde la télévision			X	
	A3	Monsieur X va au cinéma				X

Légende

O veut dire oui

N veut dire NON

X veut dire action à exécuter

2.3. Les types de tables de décision

Il existe deux types de tables de décision :

2.3.1. Les tables de décision à entrées limitées :

Ce sont des tables où

Les valeurs des conditions ne peuvent être que des oui, non ou des signes =. Le signe = placé dans une case signifie que la condition située en face de cette case n'intervient pas, que la réponse soit oui ou non, la décision est toujours exécutée.

		R1	R2	R3
C1	Fait-il bien ?	O	N	N
C2	Les programmes de la télévision sont-ils intéressants ?	=	O	N
A1	Monsieur X va à la pèche	X		
A2	Monsieur X regarde la télévision		X	
A3	Monsieur X va au cinéma			X

La règle R1 ne fait pas intervenir les valeurs de la condition C2.

2.3.2. Les tables de décision à entrées étendues :

Les valeurs des conditions représentent des précisions quantitatives concernant les conditions.

Les valeurs des actions représentent des précisions quantitatives concernant les actions.

Temps ?	Beau	Beau	Pas Beau	Pas Beau
Les programmes de la télévision ?	Intéressants	Pas Intéressants	Intéressants	Pas Intéressants
Activités du jeudi de Monsieur X	Pèche	Pèche	Télévision	Cinéma

2.3.3. L'enchaînement des tables de décision

Dans le cas d'un problème complexe dont la résolution revient à résoudre plusieurs problèmes simples, il est préférable de remplacer une grande table de décision regroupant beaucoup de conditions et d'actions indépendantes en plusieurs petites tables plus faciles à comprendre et à tester tel que chaque table de décision correspond à un sous problème.

Résoudre le problème revient à passer d'une table à une autre selon certaine logique de résolution. On parlera alors de tables de décision chaînées.

Le chaînage entre les tables se fait en attribuant à chacune d'elles un numéro séquentiel l'identifiant.

Identifiant de la table	Désignation des conditions	Entrées des conditions
	Désignation des actions	Entrées des actions
	Aller à la table « num »	

Exemple :

Table 1	Fait-il bien ?	O	N
	Monsieur X va à la pèche	X	
	Aller à la table 2		X

Table 2	Les programmes de la télévision sont il intéressants ?	O	N
	Monsieur X regarde la télévision	X	
	Monsieur X va au cinéma		X

3. La codification

Pour être traitées par l'ordinateur, les informations ont besoin d'être structurées. Cette structuration passe, obligatoirement, par l'association des codes aux différentes informations et concepts manipulés par le système d'information.

Ces codes vont permettre de désigner chaque information de manière claire et unique.

Exemple :

Soit le document suivant :

Bon de commande

Numéro commande :…….

Date commande :……

Numéro client :…….

Nom client :……..

Adresse client :…….

Référence	Désignation	Quantité

Vous voyez que les désignations des données du document sont trop longues et donc très lourdes à manipuler. Le mieux, serait de les abréger sans perdre leurs significations.

Exemple :

Numéro commande	→ Num_C	Date commande	→ Date_C
Numéro client	→ Num_Cl	Nom client	→ Nom_Cl

3.1. Définitions :

Un code :

Est un nom abrégé ou une représentation de l'information permettant de désigner un objet ou un concept de manière claire et unique.

La codification :

Est l'opération qui consiste à remplacer une information, sous sa forme, naturelle par un code clair qui serait mieux adapté aux besoins de l'utilisateur de l'information.

La codification porte sur le nom de l'information à codifier, mais, aussi, sur sa valeur.

Exemple :

Nom de la variable	Num_Cl = C003 Num_Cl = E015	Valeurs de la variable

3.2. Principales caractéristiques d'une codification :

3.2.1. La nom ambiguïté :

Une codification ne doit pas être ambiguë, c'est-à-dire, qu'elle doit associer un code et un seul à chaque information à codifier et chaque code doit être attribué à une et une seule information.

3.2.2. Facilité d'utilisation :

Cela se traduit par le fait que les fonctions de codification et de décodification doivent être faciles et simples à réaliser par l'utilisateur. Autrement dit, il doit être possible à l'utilisateur de rajouter, tout seul, de nouveaux codes et qu'il puisse interpréter les codes pré établis.

3.2.3. Possibilité d'extension et d'insertion :

L'extension exprime le fait que l'ensemble des informations codifiées puisse s'accroître.

L'insertion exprime le fait qu'un nouveau code puisse s'insérer entre deux codes déjà existants.

La codification doit permettre l'extension et l'insertion de nouveaux objets sans remettre en cause la codification choisie, c'est-à-dire que la codification établie doit être aussi stable que possible.

3.2.4. La concision :

La concision traduit le fait qu'un code doit être claire et court, sans pour autant négliger la possibilité de l'évolution de l'ensemble des informations à codifier.

3.3. Les différents types de codification :

3.3.1. La codification séquentielle :

Cela consiste à attribuer à chaque information à codifier, un numéro de sorte que les numéros associés soient consécutifs (1, 2, 3, …).

Avantages :

- Non ambiguë
- Simple à mettre en œuvre
- Extension possible

Inconvénients :

- Insertion impossible
- Non significative

3.3.2. La codification par tranches :

Cela consiste à attribuer une tranche de codes à chaque catégorie d'objets à codifier.

Les codes contenus dans une tranche sont séquentiels.

Exemple :

Dans une bibliothèque, les ouvrages sont classés par catégories, comme suit : technologie, littérature, sociologie, médecine et culture générale.

La codification des ouvrages peut se faire comme suit :

- De 001 à 100 : Technologie
- De 101 à 200 : Littérature
- De 201 à 300 : Sociologie
- De 301 à 400 : Médecine
- De 401 à 500 : Culture générale

Avantages :

- Non ambiguë
- Simple à mettre en œuvre
- Insertion possible : si le nombre d'information à codifier ne dépasse pas l'intervalle des codes prévus pour cette tranche.

- Extension possible

Inconvénients :

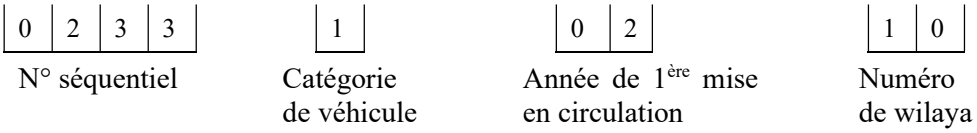
- Non significative
- Le nombre de code dans une tranche est, quelques fois, difficile à fixer
- La répartition des objets en catégories n'est pas toujours évidente
- Insertion impossible : si le nombre d'informations à codifier dépasse l'intervalle prévu

3.3.3. La codification articulée :

Cela consiste à attribuer des codes découpés en zones, chaque zone est appelée descripteur, chaque descripteur a une signification particulière relative à l'objet codifié.

Exemple :

L'exemple type est le code attribué à l'immatriculation d'un véhicule :



Avantages :

Non ambiguë

Insertion et extension possibles

Possibilité de regrouper les objets selon un critère donné

Codification très utilisée

Possibilité de contrôle

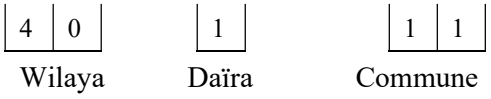
Inconvénients :

- Codes trop longs, donc lourds à manipuler
- Possibilité de saturation d'un descripteur
- Instabilité : si un descripteur change, c'est tout le code qui va être changé

4.3.4. La codification par niveau :

C'est cas particulier de la codification articulée. Les descripteurs sont des niveaux.

Exemple : Code postale



Avantages :

- Mêmes avantages que pour la codification articulée
- Facilité de recherche due à la hiérarchisation des niveaux

Inconvénients :

- Mêmes inconvénients que pour la codification articulée

3.4. Choix d'une codification :

Pour choisir le type de codification, il faut savoir :

- De quelle manière sera utilisé le code ?
- Quel est le nombre d'information à codifier ?
- L'ensemble des informations est-il évolutif ?

En conclusion, on dira que pour établir une codification, la consultation des utilisateurs s'impose, vu que ce sont eux qui vont l'utiliser et qui connaissent mieux que nous, les objets à codifier et la manière dont seront utilisé les codes associés.

4. Les contrôles

4.1. Définition :

Contrôler une information, c'est vérifier sa justesse et sa conformité à la réalité de l'organisation.

4.2. Principaux types de contrôles

4.2.1 Les contrôles directs :

Il s'agit des contrôles qui s'effectuent sur l'information elle-même sans tenir compte des autres informations existant dans le système.

Les principaux contrôles directs qu'on peut effectuer sur une information sont :

A. ***Contrôle de présence ou non présence :*** consiste à vérifier l'existence ou non d'une information sur le support où elle devrait se trouver. Le support peut être un document ou un fichier.
B. ***Contrôle de type :*** il s'agit de vérifier que le type d'une information correspond à ce qu'il devrait être.
C. ***Contrôle de cadrage :*** le cadrage désigne la position d'une information dans une zone de saisie ou remplissage. Les informations numériques sont cadrées à droite.

4.2.2 Les contrôles indirects :

Un contrôle indirect consiste à vérifier la conformité d'une information par rapport à d'autres informations, ce qui sous-entend qu'il y a comparaison entre les informations. Les principaux contrôles indirects sont :

A. ***Contrôle de cohérence interne :*** ce type de contrôle s'applique, généralement à la codification articulée. Il s'agit de vérifier l'exactitude d'une partie de l'information par rapport à d'autres parties de la même information.
B. ***Contrôle de cohérence externe :*** cela consiste à vérifier la conformité d'une information par rapport à d'autres informations.
C. ***Contrôle de vraisemblance :*** il s'agit de s'assure que l'information est vraisemblable, c'est-à-dire possible et concevable en fonction de son sens.

5. la compression des données :

La compression est l'action utilisée pour réduire la taille physique d'un bloc d'information. En compressant des données, on peut placer plus d'informations dans le même espace de stockage, ou utiliser moins de temps pour le transfert au travers d'un réseau téléinformatique.

Les données une fois compressées ne sont plus directement accessibles, en tant que données cohérentes ; pour les récupérer, il suffit de les décompresser par un algorithme inverse de compression.

5.1. L'évaluation de la compression :

Un algorithme de compression est évalué par son degré de réduction des données. Il prend le nom de ***quotient*** de compression. Ce quotient compare le volume des données comprimées à celui des

données initiales. *Quotient de compression =taille de données compressées / taille de données originales*

5.2. Les méthodes de compression

5.2.1 Suppression de blancs :

Le but est de scruter un ensemble d'information et de rechercher des suites de blancs. Lorsqu'une séquence de blancs est rencontrée, elle est remplacée par un couple de caractères ordonné : le premier est un code spécial indiquant qu'il s'agit d'une répétition, le second est un compteur du nombre de blancs.

```
Début : algorithme de compression
        Compteur = 0
       CS /*caractère spécial de répétition
       Tant que pas fin des données faire
                Lire (caractère)
               Si caractère = blancs alors
                  Compteur = compteur + 1      /*codage du nombre de répétition sur un octet
              Sinon
                    Si compteur > 2 alors
                      Ecrire (CS)
                     Ecrire (compteur)
                    Sino
                        Si compteur = 2 alors écrire (blancs, blancs) fin si
                       Si compteur =1 alors écrire (blancs) fin si
                   Fin si
            Ecrire (caractère)
           Fin si
    Fin tant que
Fin
```

Des gains de 30 à 50% ont été enregistrés pour cette méthode de compression.

Un inconvénient major réside dans le choix du caractère spécial ; il ne faut pas qu'il fasse partie des données initiales sinon les données restituées n'auraient plus aucun sens.

Dans la phase de décompression, il suffit de lire les caractères jusqu'à ce qu'on trouve un caractère spécial. Dans ce cas, la lecture du code suivant indique le nombre de blancs à insérer dans le texte final.

5.2.2 Run Length encoding (RLE) :

Le principe consiste à détecter une donnée ayant un nombre d'apparition consécutive qui dépasse un seuil fixé. Puis à remplacer cette séquence par trois informations : un code spécial indiquant qu'il s'agit d'une répétition, un chiffre indiquant le nombre de répétition et enfin l'information à répéter.

```
Début : algorithme de compression
        NbrRep = 1
       CodeRep /*caractère spécial de répétition*/
       Lire (AncCar)
       Tant que pas fin des données faire
                Lire (caractère)
               Si caractère = AncCar alors
                  NbrRep = NbrRep + 1
              Sinon
                    Si NbrRep > 3 alors
```

```
                    Ecrire (Coderep)
                   Ecrire (NbrRep)
                  Ecrire (AncCar)
                 Sino
                    Pour i allant de 1 à NbrRep faire
                        Ecrire (AncCar)
                    Fin pour
                Fin si
               NbrRep = 1
              AncCar = caractère
          Fin si
    Fin tant que
Fin
```

```
Début : algorithme dedé compression
        CodeRep /*caractère spécial de répétition*/
        Tant que pas fin des données faire
                Lire (caractère)
               Si caractère =Coderep alors
                 Lire ( NbrRep)
                Lire (codeARepeter)
                Pour i allant de 1 à NbrRep faire
                     Ecrire (codeARepeter)
                 Fin pour
              Sino
                   Ecrire ( caractère)
              Fin si
    Fin tant que
Fin
```

Cette méthode est surtout utilisée pour la compression de images.

Exemple : Source : 000001111111111111011

Compression : #60#121011

5.2.3. Méthode de Huffman :

Mise en œuvre par D.Huffman dès l'année 1952, elle fut révolutionnaire dans le sens où les caractères n'étaient plus codés sur un même nombre de bits, mais plutôt sur un nombre variable de bits, en fonction de leurs fréquences d'apparition dans le texte.

La méthode consiste à construire un arbre binaire en utilisant deux passes :

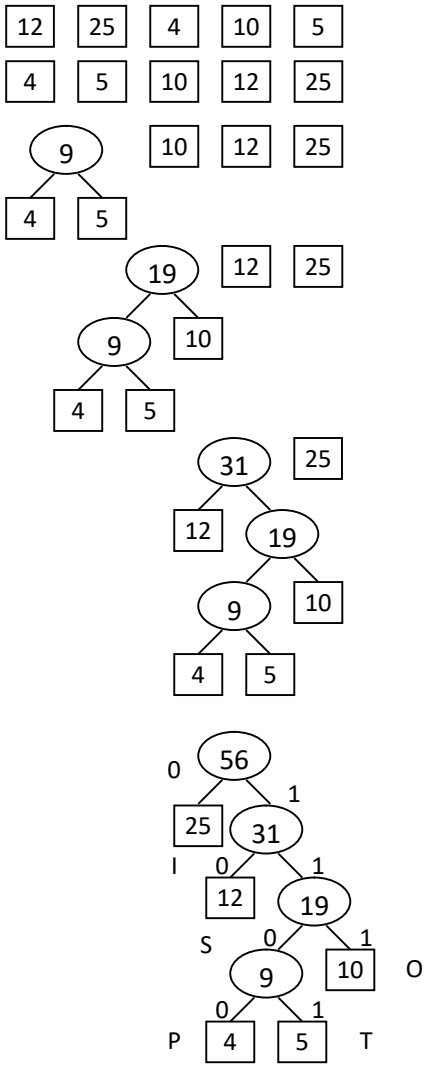
1. La première passe est effectuée sur le texte à coder en vue de déterminer les fréquences des caractères présents. La table des fréquences est alors triée selon un ordre décroissant.
2. L'arbre Huffman est alors construit d'une manière ascendante ; à chaque pas dans la construction de l'arbre, les deux plus petites fréquences sont remplacées par un nœud intermédiaire constituant leur père et ayant un poids égal à leur somme. Ce processus est répété avec la nouvelle table jusqu' à générer la racine de l'arbre et donc jusqu'à ce qu'il n'y ait plus qu'une valeur dans la table.
3. Une nouvelle passe est ensuite effectuée sur le texte pour substituer les codes aux caractères originaux. Pour chaque caractère le mot code correspondant est obtenu en suivant le chemin à

partir de la racine de l'arbre vers la feuille, le représentant en mémorisant 0 chaque fois qu'on emprunte la branche de gauche et 1 chaque fois qu'on emprunte la branche de droite.

Exemple

Soit un texte composé des lettres S, I, P, O et T.

Soit la table des fréquences (12, 25, 4, 10, 5).



Lettre	Fréquence	Code
S	12	10
I	25	0
P	4	1100
O	10	111
T	5	1101

Sans codage on aurait 56 octets et avec codage en aurait : 12*2+25*1+4*4+10*3+5*4=115 bits soit 14 octets.

Donc nous avons un taux de compression = 14/56 = 25%

D'où un gain de 1 - 0.25 = 75%

Pour encoder une chaîne, on remplace chaque lettre par le code correspondant pris dans la table des codes. Des exemples seraient :

Chaîne initiale	Chaîne encodée
TOP	1101111100
TOITS	1101110110110
STOP	1011011111100
SOIT	1011101101

Pour décoder une chaîne encodée, on utilise les bits successifs pour trouver un chemin à partir de la racine sachant que 1 signifié tourner à droite et que 0 signifie tourner à gauche.

Chaque fois qu'une feuille est atteinte, on sort la lettre contenue dans cette feuille et on reprend à partir de la racine.

Par exemple : soit la chaîne suivante : 101100111110110 sera décodée en SPOTS.

Malgré son efficacité, cet algorithme comporte quelques inconvénients. Premièrement, afin de constituer la table des fréquences et le codage de l'information. Il faut lire une première fois l'ensemble des données sources. En fonction de sa dimension, on perd plus ou moins de temps. Deuxièmement, lors de la transmission des informations. Il faut copier cette table de décodage pour que le décodeur restitue l'ensemble de données sources.

6. La cryptographie

6.1. Définitions :

6.1.1. La confidentialité : Propriété d'une information qui n'est accessible qu'aux seules personnes autorisées.

6.1.2. La sécurité : Techniques et outils visant à préserver la confidentialité, l'intégrité et la disponibilité des données et des systèmes informatiques.

6.1.3. Cryptographie : science visant à assurer la *confidentialité* et la *sécurité* de l'information.

6.2. Éléments clés de la cryptographie

La cryptographie comporte quatre éléments essentiels :

- Le chiffrement : processus qui consiste à encoder les données en transformant le texte lisible ou le message original en « texte chiffré », inintelligible.
- Le déchiffrement : processus qui consiste à décoder les données en rendant le texte chiffré lisible ou en rétablissant le message d'origine, le rendant de nouveau compréhensible.
- Un algorithme : formule mathématique appliquée au message qui permet de chiffrer et de déchiffrer les données.
- Une clé : code qui, appliqué à un algorithme, permet de chiffrer et de déchiffrer les données de façon à ce que celles-ci soient associées à une personne ou à une entreprise en particulier.

6.3. Les types de chiffrement :

6.3.1. Chiffrement à clé secrète : Un système de chiffrement à clé secrète, dit aussi symétrique, repose sur le partage entre deux personnes en communication, d'une même clé secrète utilisée à la fois pour le chiffrement des données et pour son déchiffrement (AES).

6.3.2. Chiffrement à clé publique :

Le chiffrement à clé publique, aussi appelé le chiffrement asymétrique, implique une paire de clés : une clé publique et une clé privée. La clé publique est connue de tous alors que la clé privée n'est connue que de la personne à qui la paire appartient.

Pour envoyer un texte chiffré à une personne, il faut utiliser la clé publique de cette personne et chiffrer le texte. Une fois reçu, le destinataire utilise sa clé privée correspondante pour retrouver le message clair (RSA).

6.4. Les problèmes de base de la cryptographie sont :

Confidentialité : capacité à communiquer sur un canal non sécurisé tout en préservant le secret de l'information transmise.

Intégrité des données : empêcher un adversaire actif de modifier les données transmises. Se réalise en général au moyen d'empreintes digitales (fonctions de hachage cryptographiques, cf. md5sum).

Authentification : s'assurer de l'identité d'un interlocuteur en vérifiant qu'il connaît bien un secret donné.

Non-Répudiation : est la garantie qu'aucun des correspondants ne pourra nier la transaction.

6.5. Exemple d'un algorithme de cryptage par substitution :

ROT13 : est une méthode de chiffrement qui consiste à remplacer chacun des 13 premières lettres de l'alphabet par chacun des 13 derniers. "a" devient "n" et vice versa, "b" devient "m", etc... Il suffit donc d'appliquer rot13 sur un texte chiffré en rot13 pour le déchiffrer.