Cours 2. Entropie. Introduction au codage.

A. Désilles

6 avril 2010

Résumé

1. Rappels
2. Théorie de l'information et cryptographie
3. Codage. Position du problème
4. Codage de source
   - Arbres et codes instantanés

Résumé

1. Rappels
2. Théorie de l'information et cryptographie
3. Codage. Position du problème
4. Codage de source
   - Arbres et codes instantanés

Résumé

Résumé

Résumé

Rappels : Modèle de communication

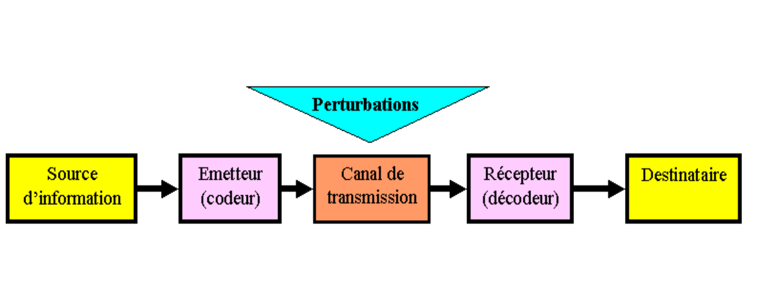


Figure: Paradigme de Shannon

Rappels : Représentation mathématique d'une source

- Une source d'information peut être :
  - un texte
  - un son
  - une image
- Elle est représentée par un ensemble fini d'éléments, appelé alphabet.
- Les éléments de l'alphabet sont appelés symboles (caractères, lettres).
- Toute séquence finie d'éléments d'alphabet est appelée mot (message).

Rappels : Représentation mathématique d'une source

- Une source d'information peut être :
  - un texte
  - un son
  - une image
- Elle est représentée par un ensemble fini d'éléments, appelé alphabet.
- Les éléments de l'alphabet sont appelés symboles (caractères, lettres).
- Toute séquence finie d'éléments d'alphabet est appelée mot (message).

Rappels : Représentation mathématique d'une source

- Une source d'information peut être :
  - un texte
  - un son
  - une image
- Elle est représentée par un ensemble fini d'éléments, appelé alphabet.
- Les éléments de l'alphabet sont appelés symboles (caractères, lettres).
- Toute séquence finie d'éléments d'alphabet est appelée mot (message).

Rappels : Représentation mathématique d'une source

- Une source d'information peut être :
  - un texte
  - un son
  - une image
- Elle est représentée par un ensemble fini d'éléments, appelé alphabet.
- Les éléments de l'alphabet sont appelés symboles (caractères, lettres).
- Toute séquence finie d'éléments d'alphabet est appelée mot (message).

Rappels : Représentation mathématique d'une source

- Une source d'information peut être :
  - un texte
  - un son
  - une image
- Elle est représentée par un ensemble fini d'éléments, appelé alphabet.
- Les éléments de l'alphabet sont appelés symboles (caractères, lettres).
- Toute séquence finie d'éléments d'alphabet est appelée mot (message).

Rappels : Représentation mathématique d'une source

- Une source d'information peut être :
    - un texte
    - un son
    - une image
- Elle est représentée par un ensemble fini d'éléments, appelé alphabet.
- Les éléments de l'alphabet sont appelés symboles (caractères, lettres).
- Toute séquence finie d'éléments d'alphabet est appelée mot (message).

Rappels : Représentation mathématique d'une source

- Une source d'information peut être :
  - un texte
  - un son
  - une image
- Elle est représentée par un ensemble fini d'éléments, appelé alphabet.
- Les éléments de l'alphabet sont appelés symboles (caractères, lettres).
- Toute séquence finie d'éléments d'alphabet est appelée mot (message).

Modèle d'une source d'information

Une source d'information X est décrite par un couple (Ω_X, P_X) où Ω_X est un alphabet fini et P_X est une distribution de probabilités sur Ω_X.

Entropie d'une source

Soient $\Omega_X = \{x_1, \ldots, x_m\}$ l'alphabet fini d'une source et X la variable aléatoire associée t.q. $P[\omega_i] = p_i, \quad i = 1, \ldots, m$. On appelle **entropie** ou encore **quantité moyenne d'information** de la source la quantité

$$H(X) = H(p_1, p_2, \ldots, p_n) = E[h(x)] = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

L'unité de mesure de cette quantité est le "bit par symbole".

Cryptographie. Vocabulaire.

- La **cryptographie** est une science qui étudie les méthodes permettant de transmettre des messages de façon confidentielle.
- **message clair** : un ensemble de données (texte, image,...) que l'on souhaite transmettre.
- Le **chiffrement** est un procédé permettant de transformer le message clair de telle sorte qu'il soit incompréhensible par qui que ce soit d'autre que l'auteur du message et le destinataire.
- Le **chiffré** est le résultat du chiffrement.
- Le **déchiffrement** est le procédé permettant de retrouver le message clair à partir du chiffré.
- La **clé** de chiffrement est un paramètre qui est utilisé dans le procédé de chiffrement ou déchiffrement.

Cryptographie. Vocabulaire.

- La **cryptographie** est une science qui étudie les méthodes permettant de transmettre des messages de façon confidentielle.
- **message clair** : un ensemble de données (texte, image,...) que l'on souhaite transmettre.
- Le **chiffrement** est un procédé permettant de transformer le message clair de telle sorte qu'il soit incompréhensible par qui que ce soit d'autre que l'auteur du message et le destinataire.
- Le **chiffré** est le résultat du chiffrement.
- Le **déchiffrement** est le procédé permettant de retrouver le message clair à partir du chiffré.
- La **clé de chiffrement** est un paramètre qui est utilisé dans le procédé de chiffrement ou déchiffrement.

Cryptographie. Vocabulaire.

- La **cryptographie** est une science qui étudie les méthodes permettant de transmettre des messages de façon confidentielle.
- **message clair** : un ensemble de données (texte, image,...) que l'on souhaite transmettre.
- Le **chiffrement** est un procédé permettant de transformer le message clair de telle sorte qu'il soit incompréhensible par qui que ce soit d'autre que l'auteur du message et le destinataire.
- Le **chiffré** est le résultat du chiffrement.
- Le **déchiffrement** est le procédé permettant de retrouver le message clair à partir du chiffré.
- La **clé** de chiffrement est un paramètre qui est utilisé dans le procédé de chiffrement ou déchiffrement.

Cryptographie. Vocabulaire.

- La **cryptographie** est une science qui étudie les méthodes permettant de transmettre des messages de façon confidentielle.
- **message clair** : un ensemble de données (texte, image,...) que l'on souhaite transmettre.
- Le **chiffrement** est un procédé permettant de transformer le message clair de telle sorte qu'il soit incompréhensible par qui que ce soit d'autre que l'auteur du message et le destinataire.
- Le **chiffré** est le résultat du chiffrement.
- Le **déchiffrement** est le procédé permettant de retrouver le message clair à partir du chiffré.
- La **clé** de chiffrement est un paramètre qui est utilisé dans le procédé de chiffrement ou déchiffrement.

Cryptographie. Vocabulaire.

- La **cryptographie** est une science qui étudie les méthodes permettant de transmettre des messages de façon confidentielle.
- **message clair** : un ensemble de données (texte, image,...) que l'on souhaite transmettre.
- Le **chiffrement** est un procédé permettant de transformer le message clair de telle sorte qu'il soit incompréhensible par qui que ce soit d'autre que l'auteur du message et le destinataire.
- Le **chiffré** est le résultat du chiffrement.
- Le **déchiffrement** est le procédé permettant de retrouver le message clair à partir du chiffré.
- La **clé** de chiffrement est un paramètre qui est utilisé dans le procédé de chiffrement ou déchiffrement.

Cryptographie. Vocabulaire.

- La **cryptographie** est une science qui étudie les méthodes permettant de transmettre des messages de façon confidentielle.
- **message clair** : un ensemble de données (texte, image,...) que l'on souhaite transmettre.
- Le **chiffrement** est un procédé permettant de transformer le message clair de telle sorte qu'il soit incompréhensible par qui que ce soit d'autre que l'auteur du message et le destinataire.
- Le **chiffré** est le résultat du chiffrement.
- Le **déchiffrement** est le procédé permettant de retrouver le message clair à partir du chiffré.
- La **clé** de chiffrement est un paramètre qui est utilisé dans le procédé de chiffrement ou déchiffrement.

Cryptosystème

Définition

Un **cryptosystème** est un quintuplé $(\mathcal{K}, \mathcal{M}, \mathcal{C}, E, D)$ formé de

- un ensemble de clés $\mathcal{K}$
- un ensemble de messages clairs possibles , $\mathcal{M}$,
- un ensemble de messages chiffrés possibles $\mathcal{C}$
- un algorithme de chiffrage, représenté par une fonction $E : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$,
- et un procédé de déchiffrage , représenté par une fonction $D : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$.

On suppose que pour tout $m \in \mathcal{M}$ il existe une paire de clés de chiffrement et de déchiffrement telles que la relation

$$D(k_D, E(k_E, m)) = m$$

est assurée.

Cryptosystème

Définition

Un **cryptosystème** est un quintuplé $(\mathcal{K}, \mathcal{M}, \mathcal{C}, E, D)$ formé de

- un ensemble de clés $\mathcal{K}$
- un ensemble de messages clairs possibles , $\mathcal{M}$,
- un ensemble de messages chiffrés possibles $\mathcal{C}$
- un algorithme de chiffrage, représenté par une fonction $E : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$,
- et un procédé de déchiffrage , représenté par une fonction $D : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$.

On suppose que pour tout $m \in \mathcal{M}$ il existe une paire de clés de chiffrement et de déchiffrement telles que la relation

$$D(k_D, E(k_E, m)) = m$$

est assurée.

Cryptosystème

Définition

Un **cryptosystème** est un quintuplé $(\mathcal{K}, \mathcal{M}, \mathcal{C}, E, D)$ formé de

- un ensemble de clés $\mathcal{K}$
- un ensemble de messages clairs possibles , $\mathcal{M}$,
- un ensemble de messages chiffrés possibles $\mathcal{C}$
- un algorithme de chiffrage, représenté par une fonction $E : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$,
- et un procédé de déchiffrage , représenté par une fonction $D : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$.

On suppose que pour tout $m \in \mathcal{M}$ il existe une paire de clés de chiffrement et de déchiffrement telles que la relation

$$D(k_D, E(k_E, m)) = m$$

est assurée.

Cryptosystème

Définition

Un **cryptosystème** est un quintuplé $(\mathcal{K}, \mathcal{M}, \mathcal{C}, E, D)$ formé de

- un ensemble de clés $\mathcal{K}$
- un ensemble de messages clairs possibles , $\mathcal{M}$,
- un ensemble de messages chiffrés possibles $\mathcal{C}$
- un algorithme de chiffrage, représenté par une fonction $E : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$,
- et un procédé de déchiffrage , représenté par une fonction $D : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$.

On suppose que pour tout $m \in \mathcal{M}$ il existe une paire de clés de chiffrement et de déchiffrement telles que la relation

$$D(k_D, E(k_E, m)) = m$$

est assurée.

Cryptosystème

Définition

Un **cryptosystème** est un quintuplé $(\mathcal{K}, \mathcal{M}, \mathcal{C}, E, D)$ formé de

- un ensemble de clés $\mathcal{K}$
- un ensemble de messages clairs possibles , $\mathcal{M}$,
- un ensemble de messages chiffrés possibles $\mathcal{C}$
- un algorithme de chiffrage, représenté par une fonction $E : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$,
- et un procédé de déchiffrage , représenté par une fonction $D : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$.

On suppose que pour tout $m \in \mathcal{M}$ il existe une paire de clés de chiffrement et de déchiffrement telles que la relation

$$D(k_D, E(k_E, m)) = m$$

est assurée.

Cryptosystème

Définition

Un **cryptosystème** est un quintuplé $(\mathcal{K}, \mathcal{M}, \mathcal{C}, E, D)$ formé de

- un ensemble de clés $\mathcal{K}$
- un ensemble de messages clairs possibles , $\mathcal{M}$,
- un ensemble de messages chiffrés possibles $\mathcal{C}$
- un algorithme de chiffrage, représenté par une fonction $E : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$,
- et un procédé de déchiffrage , représenté par une fonction $D : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$.

On suppose que pour tout $m \in \mathcal{M}$ il existe une paire de clés de chiffrement et de déchiffrement telles que la relation

$$D(k_D, E(k_E, m)) = m$$

est assurée.

Cryptosystème

Définition

Un **cryptosystème** est un quintuplé $(\mathcal{K}, \mathcal{M}, \mathcal{C}, E, D)$ formé de

- un ensemble de clés $\mathcal{K}$
- un ensemble de messages clairs possibles , $\mathcal{M}$,
- un ensemble de messages chiffrés possibles $\mathcal{C}$
- un algorithme de chiffrage, représenté par une fonction $E : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$,
- et un procédé de déchiffrage , représenté par une fonction $D : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$.

On suppose que pour tout $m \in \mathcal{M}$ il existe une paire de clés de chiffrement et de déchiffrement telles que la relation

$$D(k_D, E(k_E, m)) = m$$

est assurée.

Cryptographie. Un peu d'histoire

- Cryptographie signifie "écriture cachée" (kryptos = caché, graphein= écriture)
- Kàma-Sutra recommandait aux femmes d'apprendre 60 arts, dont les échecs, la reliure, la tapisserie et... l'écriture secrète pour cacher leur liaisons
- L'une des premières méthodes de cryptage par substitution connue est celle de César (50 av. J.C). Chaque lettre de message à transmettre était remplacée par la lettre située dans l'alphabet trois positions plus loin.
- Par exemple, A est remplacé par "D", "Z" par "C".

Cryptographie. Un peu d'histoire

- Cryptographie signifie "écriture cachée" (kryptos = caché, graphein= écriture)
- Kàma-Sutra recommandait aux femmes d'apprendre 60 arts, dont les échecs, la reliure, la tapisserie et... l'écriture secrète pour cacher leur liaisons
- L'une des premières méthodes de cryptage par substitution connue est celle de César (50 av. J.C). Chaque lettre de message à transmettre était remplacée par la lettre située dans l'alphabet trois positions plus loin.
- Par exemple, A est remplacé par "D", "Z" par "C".

Cryptographie. Un peu d'histoire

- Cryptographie signifie "écriture cachée" (kryptos = caché, graphein= écriture)
- Kàma-Sutra recommandait aux femmes d'apprendre 60 arts, dont les échecs, la reliure, la tapisserie et... l'écriture secrète pour cacher leur liaisons
- L'une des premières méthodes de cryptage par substitution connue est celle de César (50 av. J.C). Chaque lettre de message à transmettre était remplacée par la lettre située dans l'alphabet trois positions plus loin.
- Par exemple, A est remplacé par "D", "Z" par "C".

Cryptographie. Un peu d'histoire

- Cryptographie signifie "écriture cachée" (kryptos = caché, graphein= écriture)
- Kàma-Sutra recommandait aux femmes d'apprendre 60 arts, dont les échecs, la reliure, la tapisserie et... l'écriture secrète pour cacher leur liaisons
- L'une des premières méthodes de cryptage par substitution connue est celle de César (50 av. J.C). Chaque lettre de message à transmettre était remplacée par la lettre située dans l'alphabet trois positions plus loin.
- Par exemple, A est remplacé par "D", "Z" par "C".

Cryptographie. Un peu d'histoire.

- Les méthodes de substitution, analogues à celle de César, consistent à apparier les lettres de l'alphabet et à remplacer chaque lettre de message par la lettre de sa paire
- Elles ont été massivement utilisées jusqu'à la fin du 1er millénaire
- IXème siècle : le savant arabe AL-Kindi rédige le premier traité connu sur l'analyse fréquentielle comme technique de cryptanalyse
- Il montre comment retrouver le message en analysant les fréquences d'occurrence de caractères dans une langue donnée
- Le cryptage par substitution devient trop vulnérable

Cryptographie. Un peu d'histoire.

- Les méthodes de substitution, analogues à celle de César, consistent à apparier les lettres de l'alphabet et à remplacer chaque lettre de message par la lettre de sa paire
- Elles ont été massivement utilisées jusqu'à la fin du 1er millénaire
- IXème siècle : le savant arabe AL-Kindi rédige le premier traité connu sur l'analyse fréquentielle comme technique de cryptanalyse
- Il montre comment retrouver le message en analysant les fréquences d'occurrence de caractères dans une langue donnée
- Le cryptage par substitution devient trop vulnérable

Cryptographie. Un peu d'histoire.

- Les méthodes de substitution, analogues à celle de César, consistent à apparier les lettres de l'alphabet et à remplacer chaque lettre de message par la lettre de sa paire
- Elles ont été massivement utilisées jusqu'à la fin du 1er millénaire
- IXème siècle : le savant arabe AL-Kindi rédige le premier traité connu sur l'analyse fréquentielle comme technique de cryptanalyse
- Il montre comment retrouver le message en analysant les fréquences d'occurrence de caractères dans une langue donnée
- Le cryptage par substitution devient trop vulnérable

Cryptographie. Un peu d'histoire.

- Les méthodes de substitution, analogues à celle de César, consistent à apparier les lettres de l'alphabet et à remplacer chaque lettre de message par la lettre de sa paire
- Elles ont été massivement utilisées jusqu'à la fin du 1er millénaire
- IXème siècle : le savant arabe AL-Kindi rédige le premier traité connu sur l'analyse fréquentielle comme technique de cryptanalyse
- Il montre comment retrouver le message en analysant les fréquences d'occurrence de caractères dans une langue donnée
- Le cryptage par substitution devient trop vulnérable

Cryptographie. Un peu d'histoire.

- Les méthodes de substitution, analogues à celle de César, consistent à apparier les lettres de l'alphabet et à remplacer chaque lettre de message par la lettre de sa paire
- Elles ont été massivement utilisées jusqu'à la fin du 1er millénaire
- IXème siècle : le savant arabe AL-Kindi rédige le premier traité connu sur l'analyse fréquentielle comme technique de cryptanalyse
- Il montre comment retrouver le message en analysant les fréquences d'occurrence de caractères dans une langue donnée
- Le cryptage par substitution devient trop vulnérable

Cryptographie. Notion de sécurité.

Principe de A. Kerckhoffs (fin XIXe)

La sécurité d'un cryptosystème ne doit pas reposer sur la non divulgation de la fonction de cryptage mais uniquement sur la non divulgation de la clé.

Sécurité : approche de Shannon

- 1949. Publication par C. Shannon de l'article "Communication Theory of Secrecy Systems" dans la revue Bell System Technical Journal.
- Le concept d'entropie est utilisé pour analyser et quantifier la sécurité d'un cryptosystème.

Sécurité : approche de Shannon

- 1949. Publication par C. Shannon de l'article "Communication Theory of Secrecy Systems" dans la revue Bell System Technical Journal.
- Le concept d'entropie est utilisé pour analyser et quantifier la sécurité d'un cryptosystème.

Cryptosystème parfaitement sûr

- On associe aux cryptosystème $(\mathcal{K}, \mathcal{M}, \mathcal{C}, E, D)$ trois variables aléatoires :
- $M \in \mathcal{M}$ représente le choix d'un message clair
- $K \in \mathcal{K}$ représente le choix d'une clé
- $C \in \mathcal{C}$ représente le choix d'un chiffré
- on suppose que le message clair et la clé sont choisis de façon indépendante

Cryptosystème parfaitement sûr

- On associe aux cryptosystème $(\mathcal{K}, \mathcal{M}, \mathcal{C}, E, D)$ trois variables aléatoires :
- $M \in \mathcal{M}$ représente le choix d'un message clair
- $K \in \mathcal{K}$ représente le choix d'une clé
- $C \in \mathcal{C}$ représente le choix d'un chiffré
- on suppose que le message clair et la clé sont choisis de façon indépendante

Cryptosystème parfaitement sûr

- On associe aux cryptosystème $(\mathcal{K}, \mathcal{M}, \mathcal{C}, E, D)$ trois variables aléatoires :
- $M \in \mathcal{M}$ représente le choix d'un message clair
- $K \in \mathcal{K}$ représente le choix d'une clé
- $C \in \mathcal{C}$ représente le choix d'un chiffré
- on suppose que le message clair et la clé sont choisis de façon indépendante

Cryptosystème parfaitement sûr

- On associe aux cryptosystème $(\mathcal{K}, \mathcal{M}, \mathcal{C}, E, D)$ trois variables aléatoires :
- $M \in \mathcal{M}$ représente le choix d'un message clair
- $K \in \mathcal{K}$ représente le choix d'une clé
- $C \in \mathcal{C}$ représente le choix d'un chiffré
- on suppose que le message clair et la clé sont choisis de façon indépendante

Cryptosystème parfaitement sûr

- On associe aux cryptosystème $(\mathcal{K}, \mathcal{M}, \mathcal{C}, E, D)$ trois variables aléatoires :
- $M \in \mathcal{M}$ représente le choix d'un message clair
- $K \in \mathcal{K}$ représente le choix d'une clé
- $C \in \mathcal{C}$ représente le choix d'un chiffré
- on suppose que le message clair et la clé sont choisis de façon indépendante

Cryptosystème parfaitement sûr

Définition

Soit un cryptosystème $(\mathcal{K}, \mathcal{M}, \mathcal{C}, E, D)$. Soient M et C les variables aléatoires représentant le choix d'un message claire et d'un chiffré. Le système est dit parfaitement sûr ssi

$$H(M|C) = H(M)$$

La connaissance du chiffré n'apporte aucune information sur le message clair.

Cryptosystème parfaitement sûr

Définition

Soit un cryptosystème $(\mathcal{K}, \mathcal{M}, \mathcal{C}, E, D)$. Soient M et C les variables aléatoires représentant le choix d'un message claire et d'un chiffré. Le système est dit parfaitement sûr ssi

$$H(M|C) = H(M)$$

La connaissance du chiffré n'apporte aucune information sur le message clair.

Exemple de chiffrement parfaitement sûr : le chiffre de Vernam

- Cette méthode a été proposée en 1917.
- Les messages clairs sont des suites de bits de longueur n. L'espace des messages est alors $\mathcal{M} = 0, 1^n$.
- Les clés sont les suites binaires de même longueur que les messages : $\mathcal{K} = \mathcal{M} = 0, 1^n$.
- La fonction de chiffrement : pour tout $m = m_1 \ldots m_n$ et pour tout $k = k_1 \ldots k_n$

$$c = E(k, m) = k \oplus m = m_1 \oplus k_1 \ldots m_n \oplus k_n$$

Exemple de chiffrement parfaitement sûr : le chiffre de Vernam

- Cette méthode a été proposée en 1917.
- Les messages clairs sont des suites de bits de longueur n. L'espace des messages est alors $\mathcal{M} = 0, 1^n$.
- Les clés sont les suites binaires de même longueur que les messages : $\mathcal{K} = \mathcal{M} = 0, 1^n$.
- La fonction de chiffrement : pour tout $m = m_1 \dots m_n$ et pour tout $k = k_1 \dots k_n$

$$c = E(k, m) = k \oplus m = m_1 \oplus k_1 \dots m_n \oplus k_n$$

Exemple de chiffrement parfaitement sûr : le chiffre de Vernam

- Cette méthode a été proposée en 1917.
- Les messages clairs sont des suites de bits de longueur n. L'espace des messages est alors $\mathcal{M} = 0, 1^n$.
- Les clés sont les suites binaires de même longueur que les messages : $\mathcal{K} = \mathcal{M} = 0, 1^n$.
- La fonction de chiffrement : pour tout $m = m_1 \dots m_n$ et pour tout $k = k_1 \dots k_n$

$$c = E(k, m) = k \oplus m = m_1 \oplus k_1 \dots m_n \oplus k_n$$

Exemple de chiffrement parfaitement sûr : le chiffre de Vernam

- Cette méthode a été proposée en 1917.
- Les messages clairs sont des suites de bits de longueur n. L'espace des messages est alors $\mathcal{M} = 0, 1^n$.
- Les clés sont les suites binaires de même longueur que les messages : $\mathcal{K} = \mathcal{M} = 0, 1^n$.
- La fonction de chiffrement : pour tout $m = m_1 \dots m_n$ et pour tout $k = k_1 \dots k_n$

$$c = E(k, m) = k \oplus m = m_1 \oplus k_1 \dots m_n \oplus k_n$$

Exemple de chiffrement parfaitement sûr : le chiffre de Vernam

Proposition

Le chiffrement de Vernam est parfaitement sur

Théorème

Dans un système cryptographique parfaitement sûr on a

$$H(K) \geq H(M)$$

En particulier, si tous les messages et toutes les clés sont équiprobables, les clés sont de longueur au moins égale à celle des messages.

Exemple de chiffrement parfaitement sûr : le chiffre de Vernam

Proposition

Le chiffrement de Vernam est parfaitement sur

Théorème

Dans un système cryptographique parfaitement sûr on a

$$H(K) \geq H(M)$$

En particulier, si tous les messages et toutes les clés sont équiprobables, les clés sont de longueur au moins égale à celle des messages.

Attention ! Un chiffrement parfaitement sûr n'est pas invulnérable !

- Dans le cas de chiffrement de Vernam il suffit d'un bloc de message clair pour découvrir la clé :

$$k = c \oplus m$$

- Il est alors nécessaire de changer de clé à chaque bloc ; d'où le nom de "masque jetable" ;
- En pratique, cette procédure et très lourde : les clés ont la même longueur que les messages
- Ce chiffrement a été utilisé pour le téléphone rouge : la clé a été envoyé par porteur

Attention ! Un chiffrement parfaitement sûr n'est pas invulnérable !

- Dans le cas de chiffrement de Vernam il suffit d'un bloc de message clair pour découvrir la clé :

$$k = c \oplus m$$

- Il est alors nécessaire de changer de clé à chaque bloc ; d'où le nom de "masque jetable" ;
- En pratique, cette procédure et très lourde : les clés ont la même longueur que les messages
- Ce chiffrement a été utilisé pour le téléphone rouge : la clé a été envoyé par porteur

Attention ! Un chiffrement parfaitement sûr n'est pas invulnérable !

- Dans le cas de chiffrement de Vernam il suffit d'un bloc de message clair pour découvrir la clé :

$$k = c \oplus m$$

- Il est alors nécessaire de changer de clé à chaque bloc ; d'où le nom de "masque jetable" ;
- En pratique, cette procédure et très lourde : les clés ont la même longueur que les messages
- Ce chiffrement a été utilisé pour le téléphone rouge : la clé a été envoyé par porteur

Attention ! Un chiffrement parfaitement sûr n'est pas invulnérable !

- Dans le cas de chiffrement de Vernam il suffit d'un bloc de message clair pour découvrir la clé :

$$k = c \oplus m$$

- Il est alors nécessaire de changer de clé à chaque bloc ; d'où le nom de "masque jetable" ;
- En pratique, cette procédure et très lourde : les clés ont la même longueur que les messages
- Ce chiffrement a été utilisé pour le téléphone rouge : la clé a été envoyé par porteur

Le codage

D'une manière générale le codage peut être vu comme une transformation de symboles d'un alphabet donné $\Omega_1 = \{s_1, \ldots, s_n\}$ en suites de symboles d'un autre alphabet $\Omega_2 = \{c_1, \ldots, c_d\}$.

Deux problèmes de codage

1. **Codage de source ou encore codage sans bruit.** Sous cette hypothèse le meilleur code sera celui qui permettra la transmission la plus rapide possible. **Le premier théorème de Shannon** donne la solution à ce problème.
2. **Codage de canal ou encore codage en présence de bruit.** On cherchera une méthode de codage permettant une transmission aussi rapide que possible tout en minimisant la probabilité des erreurs. **Le second théorème de Shannon** donne la solution à ce problème.

Deux problèmes de codage

1. **Codage de source ou encore codage sans bruit.** Sous cette hypothèse le meilleur code sera celui qui permettra la transmission la plus rapide possible. **Le premier théorème de Shannon** donne la solution à ce problème.
2. **Codage de canal ou encore codage en présence de bruit.** On cherchera une méthode de codage permettant une transmission aussi rapide que possible tout en minimisant la probabilité des erreurs. **Le second théorème de Shannon** donne la solution à ce problème.

Vocabulaire

- **Lettre, symbole ou caractère** Tout élément d'un alphabet donné ;
- **Message ou mot** Une séquence finie m de caractères d'un alphabet donné ;
- **Longueur de mot** Le nombre $l(m)$ de caractères d'un mot m ;

Vocabulaire

- **Lettre, symbole ou caractère** Tout élément d'un alphabet donné ;
- **Message ou mot** Une séquence finie m de caractères d'un alphabet donné ;
- **Longueur de mot** Le nombre $l(m)$ de caractères d'un mot m ;

Vocabulaire

- **Lettre, symbole ou caractère** Tout élément d'un alphabet donné ;
- **Message ou mot** Une séquence finie m de caractères d'un alphabet donné ;
- **Longueur de mot** Le nombre $l(m)$ de caractères d'un mot m ;

Un code

- Soit une source S d'alphabet $\Omega_S = \{s_1, \ldots, s_n\}$ et de distribution de probabilité $P_S = \{p_1, \ldots, p_n\}$
- Un code est un ensemble $\{m_1, m_2, \ldots, m_n\}$ de n mots codes correspondants chacun à un symbole de l'alphabet de la source : $\forall\ i = 1, \ldots, n,\ \ m_i = m(s_i)$.
- Soit $l_i = l(m_i)$ les longueurs des mots m_i du code. On définit alors la longueur moyenne du code par

$$\bar{L} = E[L] = \sum_{i=1}^{n} p_i l_i$$

Un code

- Soit une source S d'alphabet $\Omega_S = \{s_1, \dots, s_n\}$ et de distribution de probabilité $P_S = \{p_1, \dots, p_n\}$
- Un code est un ensemble $\{m_1, m_2, \dots, m_n\}$ de n mots codes correspondants chacun à un symbole de l'alphabet de la source : $\forall\ i = 1, \dots, n,\ \ m_i = m(s_i)$.
- Soit $l_i = l(m_i)$ les longueurs des mots m_i du code. On définit alors la longueur moyenne du code par

$$\bar{L} = E[L] = \sum_{i=1}^{n} p_i l_i$$

Un code

- Soit une source S d'alphabet $\Omega_S = \{s_1, \ldots, s_n\}$ et de distribution de probabilité $P_S = \{p_1, \ldots, p_n\}$
- Un code est un ensemble $\{m_1, m_2, \ldots, m_n\}$ de n mots codes correspondants chacun à un symbole de l'alphabet de la source : $\forall\ i = 1, \ldots, n,\ \ m_i = m(s_i)$.
- Soit $l_i = l(m_i)$ les longueurs des mots m_i du code. On définit alors la longueur moyenne du code par

$$\bar{L} = E[L] = \sum_{i=1}^{n} p_i l_i$$

Propriétés d'un code

- **Régularité** Un code $\{m_1, m_2, \ldots, m_n\}$ est dit régulier si tous les mots qui le composent sont distincts : $m_i \neq m_k, \quad \forall i \neq k$. Un code qui n'est pas régulier est dit **singulier ou irréversible**.
- **Déchiffrabilité** Un code régulier est dit déchiffrable (ou encore à décodage unique) si pour toute suite de mots de code $m^1 m^2 \ldots m^k$ il est possible de distinguer sans ambiguïté tous les mots et donc identifier les symboles $s^j, \ j = 1, \ldots, k$ composant le message.

Propriétés d'un code

- **Régularité** Un code $\{m_1, m_2, \ldots, m_n\}$ est dit régulier si tous les mots qui le composent sont distincts : $m_i \neq m_k, \quad \forall i \neq k$. Un code qui n'est pas régulier est dit **singulier ou irréversible**.
- **Déchiffrabilité** Un code régulier est dit déchiffrable (ou encore à décodage unique) si pour toute suite de mots de code $m^1 m^2 \ldots m^k$ il est possible de distinguer sans ambiguïté tous les mots et donc identifier les symboles $s^j, \ j = 1, \ldots, k$ composant le message.

Propriétés d'un code. Exemples

Soit $\Omega_S = \{a, b, c, d\}$ de distribution de probabilité $P_S = \{0.4, 0.3, 0.2, 0.1\}$. L'entropie de cette source est $H(S) \simeq 1.85$.

S	Proba	Code 1	Code 2	Code 3	Code 4	Code 5	Code 6
a	0.4	1	00	0	0	0	0
b	0.3	0	01	10	01	10	11
c	0.2	1	10	01	011	110	100
d	0.1	0	11	010	0111	1110	101
	Long. Moy.	1	2	1.7	2	2	1.9

Le code 1 n'est pas régulier. Le code 2 est un code **de longueur fixe** .

Propriétés d'un code. Exemples

Soit $\Omega_S = \{a, b, c, d\}$ de distribution de probabilité $P_S = \{0.4, 0.3, 0.2, 0.1\}$. L'entropie de cette source est $H(S) \simeq 1.85$.

S	Proba	Code 1	Code 2	Code 3	Code 4	Code 5	Code 6
a	0.4	1	00	0	0	0	0
b	0.3	0	01	10	01	10	11
c	0.2	1	10	01	011	110	100
d	0.1	0	11	010	0111	1110	101
	Long. Moy.	1	2	1.7	2	2	1.9

Le code 1 n'est pas régulier. Le code 2 est un code **de longueur fixe** .

Propriétés d'un code. Exemples

Soit $\Omega_S = \{a, b, c, d\}$ de distribution de probabilité $P_S = \{0.4, 0.3, 0.2, 0.1\}$. L'entropie de cette source est $H(S) \simeq 1.85$.

S	Proba	Code 1	Code 2	Code 3	Code 4	Code 5	Code 6
a	0.4	1	00	0	0	0	0
b	0.3	0	01	10	01	10	11
c	0.2	1	10	01	011	110	100
d	0.1	0	11	010	0111	1110	101
	Long. Moy.	1	2	1.7	2	2	1.9

Le code 1 n'est pas régulier. Le code 2 est un code **de longueur fixe** .

Propriétés d'un code. Exemples

Soit $\Omega_S = \{a, b, c, d\}$ de distribution de probabilité $P_S = \{0.4, 0.3, 0.2, 0.1\}$. L'entropie de cette source est $H(S) \simeq 1.85$.

S	Proba	Code 1	Code 2	Code 3	Code 4	Code 5	Code 6
a	0.4	1	00	0	0	0	0
b	0.3	0	01	10	01	10	11
c	0.2	1	10	01	011	110	100
d	0.1	0	11	010	0111	1110	101
	Long. Moy.	1	2	1.7	2	2	1.9

Le code 1 n'est pas régulier. Le code 2 est un code **de longueur fixe** .

Un code indéchiffrable

S	Proba	Code 1	Code 2	Code 3	Code 4	Code 5	Code 6
a	0.4	1	00	0	0	0	0
b	0.3	0	01	10	01	10	11
c	0.2	1	10	01	011	110	100
d	0.1	0	11	010	0111	1110	101
	Long. Moy.	1	2	1.7	2	2	1.9

Le code 3 défini par $\{0, 10, 01, 010\}$ est régulier mais pas déchiffrable.
La séquence 010 correspond à la fois à trois messages différents : "d", "ca" et "ab".

Un code indéchiffrable

S	Proba	Code 1	Code 2	Code 3	Code 4	Code 5	Code 6
a	0.4	1	00	0	0	0	0
b	0.3	0	01	10	01	10	11
c	0.2	1	10	01	011	110	100
d	0.1	0	11	010	0111	1110	101
	Long. Moy.	1	2	1.7	2	2	1.9

Le code 3 défini par $\{0, 10, 01, 010\}$ est régulier mais pas déchiffrable.
La séquence 010 correspond à la fois à trois messages différents : "d", "ca" et "ab".

Un code indéchiffrable

S	Proba	Code 1	Code 2	Code 3	Code 4	Code 5	Code 6
a	0.4	1	00	0	0	0	0
b	0.3	0	01	10	01	10	11
c	0.2	1	10	01	011	110	100
d	0.1	0	11	010	0111	1110	101
	Long. Moy.	1	2	1.7	2	2	1.9

Le code 3 défini par $\{0, 10, 01, 010\}$ est régulier mais pas déchiffrable.
La séquence 010 correspond à la fois à trois messages différents : "d", "ca" et "ab".

Décodage unique : Solution 1.

Codes de longueur fixe

Un code régulier de longueur fixe peut toujours être décodé sans ambiguïté.
Désavantage : la longueur moyenne n'est pas optimale.

S	Proba	Code 1	Code 2	Code 3	Code 4	Code 5	Code 6
a	0.4	1	00	0	0	0	0
b	0.3	0	01	10	01	10	11
c	0.2	1	10	01	011	110	100
d	0.1	0	11	010	0111	1110	101
	Long. Moy.	1	2	1.7	2	2	1.9

Décodage unique : Solution 1.

Codes de longueur fixe

Un code régulier de longueur fixe peut toujours être décodé sans ambiguïté.

Désavantage : la longueur moyenne n'est pas optimale.

S	Proba	Code 1	Code 2	Code 3	Code 4	Code 5	Code 6
a	0.4	1	00	0	0	0	0
b	0.3	0	01	10	01	10	11
c	0.2	1	10	01	011	110	100
d	0.1	0	11	010	0111	1110	101
	Long. Moy.	1	2	1.7	2	2	1.9

Décodage unique : Solution 1.

Codes de longueur fixe
Un code régulier de longueur fixe peut toujours être décodé sans ambiguïté.
Désavantage : la longueur moyenne n'est pas optimale.

S	Proba	Code 1	Code 2	Code 3	Code 4	Code 5	Code 6
a	0.4	1	00	0	0	0	0
b	0.3	0	01	10	01	10	11
c	0.2	1	10	01	011	110	100
d	0.1	0	11	010	0111	1110	101
	Long. Moy.	1	2	1.7	2	2	1.9

Décodage unique : Solution 1.

Codes de longueur fixe

Un code régulier de longueur fixe peut toujours être décodé sans ambiguïté.

Désavantage : la longueur moyenne n'est pas optimale.

S	Proba	Code 1	Code 2	Code 3	Code 4	Code 5	Code 6
a	0.4	1	00	0	0	0	0
b	0.3	0	01	10	01	10	11
c	0.2	1	10	01	011	110	100
d	0.1	0	11	010	0111	1110	101
	Long. Moy.	1	2	**1.7**	2	2	**1.9**

Codes de longueur fixe. Exemple. Encodage de texte

Les formats courants d'encodage de caractères d'un texte sont les codes à longueur fixe

- Le code ASCII (American Standard Code for Information Interchange) utilise 8 bits (1 octet) dont un bit de parité pour chaque caractère. Il est possible de représenter 128 (2^7) caractères.
- La norme ISO 8859-1 représente 191 caractères, chacun sur 1 octet.
- Les standards du consortium UNICODE, utilisent une représentation des caractères sur 2 octets (16 bits). Le format UTF-16 (Universal Transformation Format, 16 bits) utilise 2 octets pour chaque caractère. UTF-8 utilise un nombre variable d'octets (de 2 à 4) en fonction du numéro de caractère.

Codes de longueur fixe. Exemple. Encodage de texte

Les formats courants d'encodage de caractères d'un texte sont les codes à longueur fixe

- Le code ASCII (American Standard Code for Information Interchange) utilise 8 bits (1 octet) dont un bit de parité pour chaque caractère. Il est possible de représenter 128 (2^7) caractères.
- La norme ISO 8859-1 représente 191 caractères, chacun sur 1 octet.
- Les standards du consortium UNICODE, utilisent une représentation des caractères sur 2 octets (16 bits). Le format UTF-16 (Universal Transformation Format, 16 bits) utilise 2 octets pour chaque caractère. UTF-8 utilise un nombre variable d'octets (de 2 à 4) en fonction du numéro de caractère.

Codes de longueur fixe. Exemple. Encodage de texte

Les formats courants d'encodage de caractères d'un texte sont les codes à longueur fixe

- Le code ASCII (American Standard Code for Information Interchange) utilise 8 bits (1 octet) dont un bit de parité pour chaque caractère. Il est possible de représenter 128 (2^7) caractères.
- La norme ISO 8859-1 représente 191 caractères, chacun sur 1 octet.
- Les standards du consortium UNICODE, utilisent une représentation des caractères sur 2 octets (16 bits). Le format UTF-16 (Universal Transformation Format, 16 bits) utilise 2 octets pour chaque caractère. UTF-8 utilise un nombre variable d'octets (de 2 à 4) en fonction du numéro de caractère.

Codes de longueur fixe. Exemple. La norme ISO-8859-1

ISO/CEI 8859-1																
	x0	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
0x	*caractères de contrôle et divers non imprimables*															
1x																
2x		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	$\vert$	}	~	
8x	*caractères de contrôle et divers non imprimables*															
9x																
Ax		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
Bx	°	±	^{2}	^{3}	´	µ	¶	·	¸	^{1}	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Décodage unique : Solution 2.

Utilisation d'un séparateur

Pour un canal binaire, on peut coder le i-ème symbole de la source s_i à l'aide de i caractères "1" et utiliser "0" comme séparateur.

S	Proba	Code 1
a	0.4	10
b	0.3	110
c	0.2	1110
d	0.1	11110
	Long. Moy.	3

et la séquence "abc" donnerait "101101110"

On constate que la longueur moyenne qui tient compte du séparateur est plus élevée que tous les autres codes.

Décodage unique : Solution 2.

Utilisation d'un séparateur

Pour un canal binaire, on peut coder le i-ème symbole de la source s_i à l'aide de i caractères "1" et utiliser "0" comme séparateur.

S	Proba	Code 1
a	0.4	10
b	0.3	110
c	0.2	1110
d	0.1	11110
	Long. Moy.	3

et la séquence "abc" donnerait "101101110"

On constate que la longueur moyenne qui tient compte du séparateur est plus élevée que tous les autres codes.

Décodage unique : Solution 2.

Utilisation d'un séparateur

Pour un canal binaire, on peut coder le i-ème symbole de la source s_i à l'aide de i caractères "1" et utiliser "0" comme séparateur.

S	Proba	Code 1
a	0.4	10
b	0.3	110
c	0.2	1110
d	0.1	11110
	Long. Moy.	3

et la séquence "abc" donnerait "101101110"

On constate que la longueur moyenne qui tient compte du séparateur est plus élevée que tous les autres codes.

Décodage unique : Solution 2.

Utilisation d'un séparateur

Pour un canal binaire, on peut coder le i-ème symbole de la source s_i à l'aide de i caractères "1" et utiliser "0" comme séparateur.

S	Proba	Code 1
a	0.4	10
b	0.3	110
c	0.2	1110
d	0.1	11110
	Long. Moy.	3

et la séquence "abc" donnerait "101101110"

On constate que la longueur moyenne qui tient compte du séparateur est plus élevée que tous les autres codes.

Décodage unique : Solution 3

Préfixe

On dit qu'un mot W est **un préfixe** d'un autre mot V s'il existe un mot U tel que $V = WU$. Autrement dit, le mot V commence par le mot W.

Un code instantané ou sans préfixe

On dit qu'un code donné est **sans préfixe** ou **instantané** si aucun mot du code n'est un préfixe d'un autre.

Le code 6 défini par $\{0, 11, 100, 101\}$ est sans préfixe.

Décodage unique : Solution 3

Préfixe

On dit qu'un mot W est **un préfixe** d'un autre mot V s'il existe un mot U tel que $V = WU$. Autrement dit, le mot V commence par le mot W.

Un code instantané ou sans préfixe

On dit qu'un code donné est **sans préfixe** ou **instantané** si aucun mot du code n'est un préfixe d'un autre.

Le code 6 défini par $\{0, 11, 100, 101\}$ est sans préfixe.

Décodage unique : Solution 3

Préfixe

On dit qu'un mot W est **un préfixe** d'un autre mot V s'il existe un mot U tel que $V = WU$. Autrement dit, le mot V commence par le mot W.

Un code instantané ou sans préfixe

On dit qu'un code donné est **sans préfixe** ou **instantané** si aucun mot du code n'est un préfixe d'un autre.

Le code 6 défini par $\{0, 11, 100, 101\}$ est sans préfixe.

Décodage d'un code sans préfixe

Prenons une séquence $W = 011010011101$ du code 6 donné par $\{0, 11, 100, 101\}$.

1. **Pas 1** Le premier mot du code formé en lisant de gauche à droite est $m^1 =$ "0". Donc le premier symbole est $s^1 = a$. On sépare le mot m^1 de la séquence. On obtient la nouvelle séquence $W_1 = 11010011101$.
2. **Pas 2** $m^2 =$ "11" $\Rightarrow$ $s^2 = b$ et $W_2 = 010011101$.
3. **Pas 3** $m^3 =$ "0" $\Rightarrow$ $s^3 = a$ et $W_3 = 10011101$.
4. **Pas 4** $m^4 =$ "100" $\Rightarrow$ $s^4 = c$ et $W_2 = 11101$.
5. **Pas 5** $m^5 =$ "11" $\Rightarrow$ $s^5 = b$ et $W_2 = 101$.
6. **Pas 6** $m^6 =$ "101" $\Rightarrow$ $s^6 = d$ et $W_6 = \emptyset$.

On obtient en symboles de l'alphabet de la source : *abacbd*.

Décodage d'un code sans préfixe

Prenons une séquence $W = 011010011101$ du code 6 donné par $\{0, 11, 100, 101\}$.

1. **Pas 1** Le premier mot du code formé en lisant de gauche à droite est $m^1 = "0"$. Donc le premier symbole est $s^1 = a$. On sépare le mot m^1 de la séquence. On obtient la nouvelle séquence $W_1 = 11010011101$.
2. **Pas 2** $m^2 = "11" \Rightarrow s^2 = b$ et $W_2 = 010011101$.
3. **Pas 3** $m^3 = "0" \Rightarrow s^3 = a$ et $W_3 = 10011101$.
4. **Pas 4** $m^4 = "100" \Rightarrow s^4 = c$ et $W_2 = 11101$.
5. **Pas 5** $m^5 = "11" \Rightarrow s^5 = b$ et $W_2 = 101$.
6. **Pas 6** $m^6 = "101" \Rightarrow s^6 = d$ et $W_6 = \emptyset$.

On obtient en symboles de l'alphabet de la source : *abacbd*.

Décodage d'un code sans préfixe

Prenons une séquence $W = 011010011101$ du code 6 donné par $\{0, 11, 100, 101\}$.

1. **Pas 1** Le premier mot du code formé en lisant de gauche à droite est $m^1 = "0"$. Donc le premier symbole est $s^1 = a$. On sépare le mot m^1 de la séquence. On obtient la nouvelle séquence $W_1 = 11010011101$.
2. **Pas 2** $m^2 = "11" \quad \Rightarrow \quad s^2 = b$ et $W_2 = 010011101$.
3. **Pas 3** $m^3 = "0" \quad \Rightarrow \quad s^3 = a$ et $W_3 = 10011101$.
4. **Pas 4** $m^4 = "100" \quad \Rightarrow \quad s^4 = c$ et $W_2 = 11101$.
5. **Pas 5** $m^5 = "11" \quad \Rightarrow \quad s^5 = b$ et $W_2 = 101$.
6. **Pas 6** $m^6 = "101" \quad \Rightarrow \quad s^6 = d$ et $W_6 = \emptyset$.

On obtient en symboles de l'alphabet de la source : *abacbd*.

Décodage d'un code sans préfixe

Prenons une séquence $W = 011010011101$ du code 6 donné par $\{0, 11, 100, 101\}$.

1. **Pas 1** Le premier mot du code formé en lisant de gauche à droite est $m^1 = "0"$. Donc le premier symbole est $s^1 = a$. On sépare le mot m^1 de la séquence. On obtient la nouvelle séquence $W_1 = 11010011101$.
2. **Pas 2** $m^2 = "11" \quad \Rightarrow \quad s^2 = b$ et $W_2 = 010011101$.
3. **Pas 3** $m^3 = "0" \quad \Rightarrow \quad s^3 = a$ et $W_3 = 10011101$.
4. **Pas 4** $m^4 = "100" \quad \Rightarrow \quad s^4 = c$ et $W_2 = 11101$.
5. **Pas 5** $m^5 = "11" \quad \Rightarrow \quad s^5 = b$ et $W_2 = 101$.
6. **Pas 6** $m^6 = "101" \quad \Rightarrow \quad s^6 = d$ et $W_6 = \emptyset$.

On obtient en symboles de l'alphabet de la source : *abacbd*.

Décodage d'un code sans préfixe

Prenons une séquence $W = 01101001 1101$ du code 6 donné par $\{0, 11, 100, 101\}$.

1. **Pas 1** Le premier mot du code formé en lisant de gauche à droite est $m^1 =$ "0". Donc le premier symbole est $s^1 = a$. On sépare le mot m^1 de la séquence. On obtient la nouvelle séquence $W_1 = 11010011101$.
2. **Pas 2** $m^2 = "11" \Rightarrow s^2 = b$ et $W_2 = 010011101$.
3. **Pas 3** $m^3 = "0" \Rightarrow s^3 = a$ et $W_3 = 10011101$.
4. **Pas 4** $m^4 = "100" \Rightarrow s^4 = c$ et $W_2 = 11101$.
5. **Pas 5** $m^5 = "11" \Rightarrow s^5 = b$ et $W_2 = 101$.
6. **Pas 6** $m^6 = "101" \Rightarrow s^6 = d$ et $W_6 = \emptyset$.

On obtient en symboles de l'alphabet de la source : *abacbd*.

Décodage d'un code sans préfixe

Prenons une séquence $W = 011010011101$ du code 6 donné par $\{0, 11, 100, 101\}$.

1. **Pas 1** Le premier mot du code formé en lisant de gauche à droite est $m^1 =$ "0". Donc le premier symbole est $s^1 = a$. On sépare le mot m^1 de la séquence. On obtient la nouvelle séquence $W_1 = 11010011101$.
2. **Pas 2** $m^2 = "11" \quad \Rightarrow \quad s^2 = b$ et $W_2 = 010011101$.
3. **Pas 3** $m^3 = "0" \quad \Rightarrow \quad s^3 = a$ et $W_3 = 10011101$.
4. **Pas 4** $m^4 = "100" \quad \Rightarrow \quad s^4 = c$ et $W_2 = 11101$.
5. **Pas 5** $m^5 = "11" \quad \Rightarrow \quad s^5 = b$ et $W_2 = 101$.
6. **Pas 6** $m^6 = "101" \quad \Rightarrow \quad s^6 = d$ et $W_6 = \emptyset$.

On obtient en symboles de l'alphabet de la source : *abacbd*.

Décodage d'un code sans préfixe

Prenons une séquence $W = 011010011101$ du code 6 donné par $\{0, 11, 100, 101\}$.

1. **Pas 1** Le premier mot du code formé en lisant de gauche à droite est $m^1 = "0"$. Donc le premier symbole est $s^1 = a$. On sépare le mot m^1 de la séquence. On obtient la nouvelle séquence $W_1 = 11010011101$.
2. **Pas 2** $m^2 = "11" \quad \Rightarrow \quad s^2 = b$ et $W_2 = 010011101$.
3. **Pas 3** $m^3 = "0" \quad \Rightarrow \quad s^3 = a$ et $W_3 = 10011101$.
4. **Pas 4** $m^4 = "100" \quad \Rightarrow \quad s^4 = c$ et $W_2 = 11101$.
5. **Pas 5** $m^5 = "11" \quad \Rightarrow \quad s^5 = b$ et $W_2 = 101$.
6. **Pas 6** $m^6 = "101" \quad \Rightarrow \quad s^6 = d$ et $W_6 = \emptyset$.

On obtient en symboles de l'alphabet de la source : *abacbd*.

Décodage d'un code sans préfixe

Prenons une séquence $W = 011010011101$ du code 6 donné par $\{0, 11, 100, 101\}$.

1. **Pas 1** Le premier mot du code formé en lisant de gauche à droite est $m^1 = "0"$. Donc le premier symbole est $s^1 = a$. On sépare le mot m^1 de la séquence. On obtient la nouvelle séquence $W_1 = 11010011101$.
2. **Pas 2** $m^2 = "11" \quad \Rightarrow \quad s^2 = b$ et $W_2 = 010011101$.
3. **Pas 3** $m^3 = "0" \quad \Rightarrow \quad s^3 = a$ et $W_3 = 10011101$.
4. **Pas 4** $m^4 = "100" \quad \Rightarrow \quad s^4 = c$ et $W_2 = 11101$.
5. **Pas 5** $m^5 = "11" \quad \Rightarrow \quad s^5 = b$ et $W_2 = 101$.
6. **Pas 6** $m^6 = "101" \quad \Rightarrow \quad s^6 = d$ et $W_6 = \emptyset$.

On obtient en symboles de l'alphabet de la source : *abacbd*.

Décodage d'un code sans préfixe

Prenons une séquence $W = 011010011101$ du code 6 donné par $\{0, 11, 100, 101\}$.

1. **Pas 1** Le premier mot du code formé en lisant de gauche à droite est $m^1 = "0"$. Donc le premier symbole est $s^1 = a$. On sépare le mot m^1 de la séquence. On obtient la nouvelle séquence $W_1 = 11010011101$.
2. **Pas 2** $m^2 = "11" \quad \Rightarrow \quad s^2 = b$ et $W_2 = 010011101$.
3. **Pas 3** $m^3 = "0" \quad \Rightarrow \quad s^3 = a$ et $W_3 = 10011101$.
4. **Pas 4** $m^4 = "100" \quad \Rightarrow \quad s^4 = c$ et $W_2 = 11101$.
5. **Pas 5** $m^5 = "11" \quad \Rightarrow \quad s^5 = b$ et $W_2 = 101$.
6. **Pas 6** $m^6 = "101" \quad \Rightarrow \quad s^6 = d$ et $W_6 = \emptyset$.

On obtient en symboles de l'alphabet de la source : *abacbd*.

Codes instantanés : existence

Problème

Soient l'alphabet de la source $\Omega_S = \{s_1, \ldots, s_n\}$ de taille n et et l'alphabet du canal $\Omega_C = \{c_1, \ldots, c_d\}$ de taille d. Étant donnés n nombres entiers positifs $(l_1, l_2, \ldots, l_n) \in \mathbb{Z}_+^*$ existe-t-il un code régulier instantané de n mots $\{m_1, \ldots, m_n\}$ tel que chaque nombre l_i soit la longueur du mot de code m_i ?

Inégalité de Kraft

Théorème

Un code instantané de longueurs de mots données $l_1, \ldots, l_n$ existe si et seulement si

$$\sum_{i=1}^{n} d^{-l_i} \leq 1$$

où d est la taille de l'alphabet du canal.

Application aux codes binaires

L'alphabet du canal est $\Omega_C = \{0, 1\}$ de taille $d = 2$. Alors un code instantané de longueurs de mots données $l_1, \ldots, l_n$ existe si et seulement si

$$\sum_{i=1}^{n} 2^{-l_i} \leq 1.$$

Inégalité de Kraft

Théorème

Un code instantané de longueurs de mots données $l_1, \ldots, l_n$ existe si et seulement si

$$\sum_{i=1}^{n} d^{-l_i} \leq 1$$

où d est la taille de l'alphabet du canal.

Application aux codes binaires

L'alphabet du canal est $\Omega_C = \{0, 1\}$ de taille $d = 2$. Alors un code instantané de longueurs de mots données $l_1, \ldots, l_n$ existe si et seulement si

$$\sum_{i=1}^{n} 2^{-l_i} \leq 1.$$

Arbres binaires

1. Un arbre binaire est un graphe orienté (N, R) où N est un ensemble de nœuds et $R \subset N \times N$ est un ensemble d'arcs.
2. Chaque nœud a au plus deux fils et chaque nœud sauf la racine a exactement un père.
3. Les nœuds qui n'ont pas de fils s'appellent feuilles de l'arbre.
4. On dit qu'un nœud est de **niveau** n si le chemin qui le relie à la racine est de longueur n.
5. On appelle **profondeur** d'un arbre la longueur du plus long chemin partant de la racine.
6. **Arbre binaire complet de profondeur** n : tous les nœuds sauf les feuilles ont **exactement** 2 fils.
7. **Arbre binaire incomplet** : tous les nœuds ont 2 ou 0 fils.

Arbres binaires

1. Un arbre binaire est un graphe orienté (N, R) où N est un ensemble de nœuds et $R \subset N \times N$ est un ensemble d'arcs.
2. Chaque nœud a au plus deux fils et chaque nœud sauf la racine a exactement un père.
3. Les nœuds qui n'ont pas de fils s'appellent feuilles de l'arbre.
4. On dit qu'un nœud est de **niveau** n si le chemin qui le relie à la racine est de longueur n.
5. On appelle **profondeur** d'un arbre la longueur du plus long chemin partant de la racine.
6. **Arbre binaire complet de profondeur** n : tous les nœuds sauf les feuilles ont **exactement** 2 fils.
7. **Arbre binaire incomplet** : tous les nœuds ont 2 ou 0 fils.

Arbres binaires

1. Un arbre binaire est un graphe orienté (N, R) où N est un ensemble de nœuds et $R \subset N \times N$ est un ensemble d'arcs.
2. Chaque nœud a au plus deux fils et chaque nœud sauf la racine a exactement un père.
3. Les nœuds qui n'ont pas de fils s'appellent feuilles de l'arbre.
4. On dit qu'un nœud est de **niveau** n si le chemin qui le relie à la racine est de longueur n.
5. On appelle **profondeur** d'un arbre la longueur du plus long chemin partant de la racine.
6. **Arbre binaire complet de profondeur** n **:** tous les nœuds sauf les feuilles ont exactement 2 fils.
7. **Arbre binaire incomplet** : tous les nœuds ont 2 ou 0 fils.

Arbres binaires

1. Un arbre binaire est un graphe orienté (N, R) où N est un ensemble de nœuds et $R \subset N \times N$ est un ensemble d'arcs.
2. Chaque nœud a au plus deux fils et chaque nœud sauf la racine a exactement un père.
3. Les nœuds qui n'ont pas de fils s'appellent feuilles de l'arbre.
4. On dit qu'un nœud est de **niveau** n si le chemin qui le relie à la racine est de longueur n.
5. On appelle **profondeur** d'un arbre la longueur du plus long chemin partant de la racine.
6. **Arbre binaire complet de profondeur** n **:** tous les nœuds sauf les feuilles ont exactement 2 fils.
7. **Arbre binaire incomplet** : tous les nœuds ont 2 ou 0 fils.

Arbres binaires

1. Un arbre binaire est un graphe orienté (N, R) où N est un ensemble de nœuds et $R \subset N \times N$ est un ensemble d'arcs.
2. Chaque nœud a au plus deux fils et chaque nœud sauf la racine a exactement un père.
3. Les nœuds qui n'ont pas de fils s'appellent feuilles de l'arbre.
4. On dit qu'un nœud est de **niveau** n si le chemin qui le relie à la racine est de longueur n.
5. On appelle **profondeur** d'un arbre la longueur du plus long chemin partant de la racine.
6. **Arbre binaire complet de profondeur** n **:** tous les nœuds sauf les feuilles ont exactement 2 fils.
7. **Arbre binaire incomplet** : tous les nœuds ont 2 ou 0 fils.

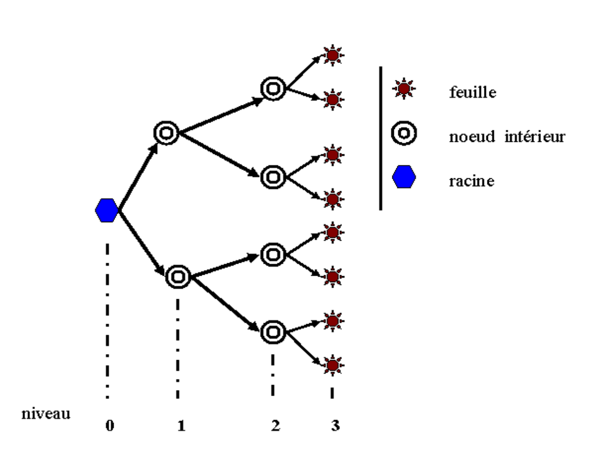
Arbres binaires

1. Un arbre binaire est un graphe orienté (N, R) où N est un ensemble de nœuds et $R \subset N \times N$ est un ensemble d'arcs.
2. Chaque nœud a au plus deux fils et chaque nœud sauf la racine a exactement un père.
3. Les nœuds qui n'ont pas de fils s'appellent feuilles de l'arbre.
4. On dit qu'un nœud est de **niveau** n si le chemin qui le relie à la racine est de longueur n.
5. On appelle **profondeur** d'un arbre la longueur du plus long chemin partant de la racine.
6. **Arbre binaire complet de profondeur** n **:** tous les nœuds sauf les feuilles ont **exactement** 2 fils.
7. **Arbre binaire incomplet** : tous les nœuds ont 2 ou 0 fils.

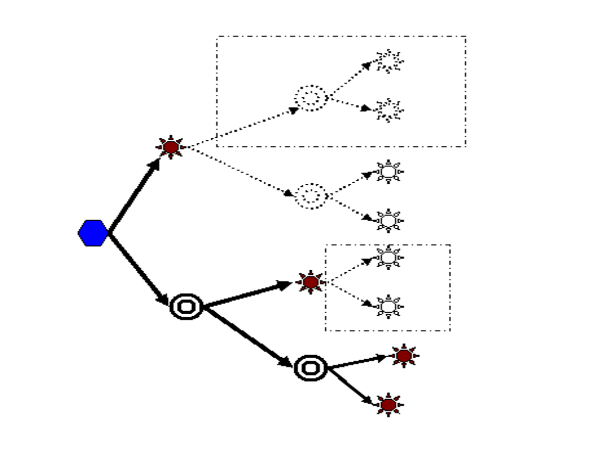
Arbres binaires

1. Un arbre binaire est un graphe orienté (N, R) où N est un ensemble de nœuds et $R \subset N \times N$ est un ensemble d'arcs.
2. Chaque nœud a au plus deux fils et chaque nœud sauf la racine a exactement un père.
3. Les nœuds qui n'ont pas de fils s'appellent feuilles de l'arbre.
4. On dit qu'un nœud est de **niveau** n si le chemin qui le relie à la racine est de longueur n.
5. On appelle **profondeur** d'un arbre la longueur du plus long chemin partant de la racine.
6. **Arbre binaire complet de profondeur n :** tous les nœuds sauf les feuilles ont **exactement** 2 fils.
7. **Arbre binaire incomplet** : tous les nœuds ont 2 ou 0 fils.

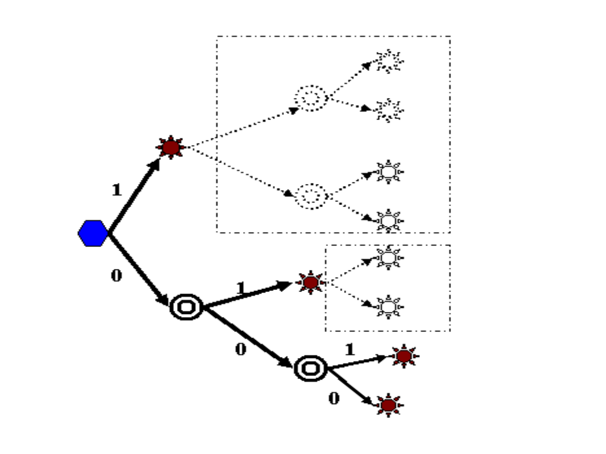
Exemple : arbre complet



Exemple : arbre incomplet



Exemple : associer un code à un arbre incomplet



Le code associé est $\{1, 01, 001, 000\}$.