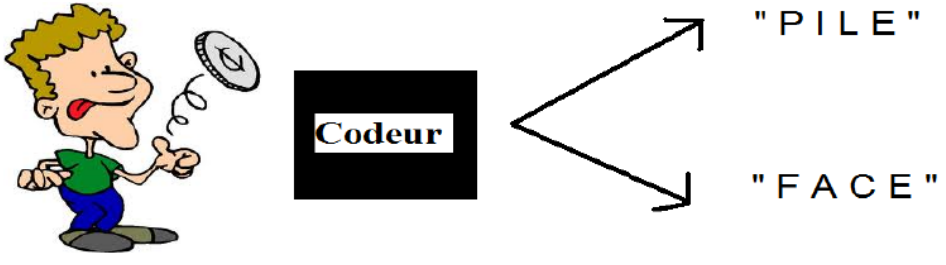
Théorie de l'information

- Transmission/stockage de l'information
  - La source produit de l'information « pure » sous forme abstraite
  - Ne peut pas être transmise ou stockée dans cet état « pur »
  - Doit avoir une représentation physique pour être transportée/stockée
- Codage
  - Processus de transformation de l'information
  - S'adapte au canal de transmission ou moyen de stockage
  - Permet au récepteur de reconvertir (décoder) l'information dans une forme intelligible (même forme qu'à la source)



  - Codage ≠ chiffrement
    - Le codage ne protège pas la confidentialité de l'information

Théorie de l'information

- Compression et codage
  - Le codage peut permettre de faire de la compression
    - Moins de symboles utilisés dans la transmission/stockage que par la source
  - 1er théorème de Shannon (voir plus loin)
    - Établit limite de la compression sans perte d'information *(lossless compression)*
      - Codes de Huffman
      - Lempel-Ziv-Welch (LZW)
    - Ne s'applique pas à la compression avec perte (lossy compression)
      - MP3
      - JPEG
      - MPEG

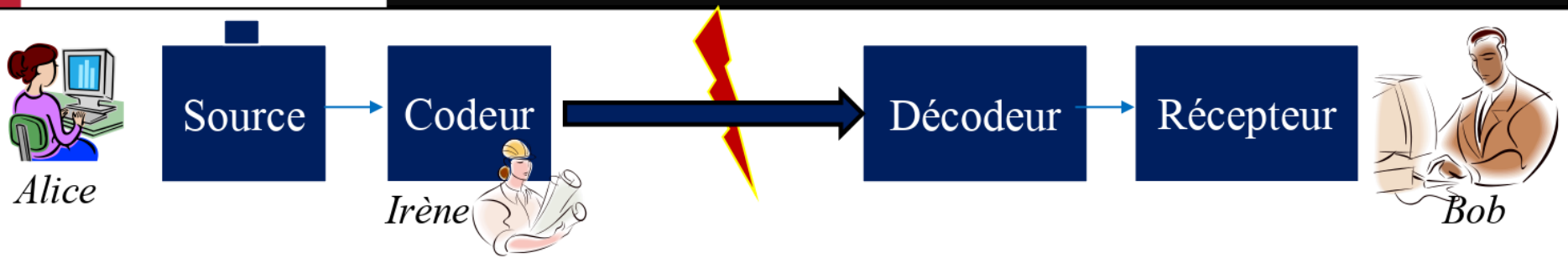
Théorie de l'information

- Les composants que nous avons évoqués sont du côté de la source
  - On fait l'image miroir pour avoir les composants du côté du récepteur
  - Source – codeur $\Rightarrow$ décodeur – récepteur

- On peut alors créer un modèle mathématique plus formel

  ➔ le modèle de Shannon

Modèle de Shannon



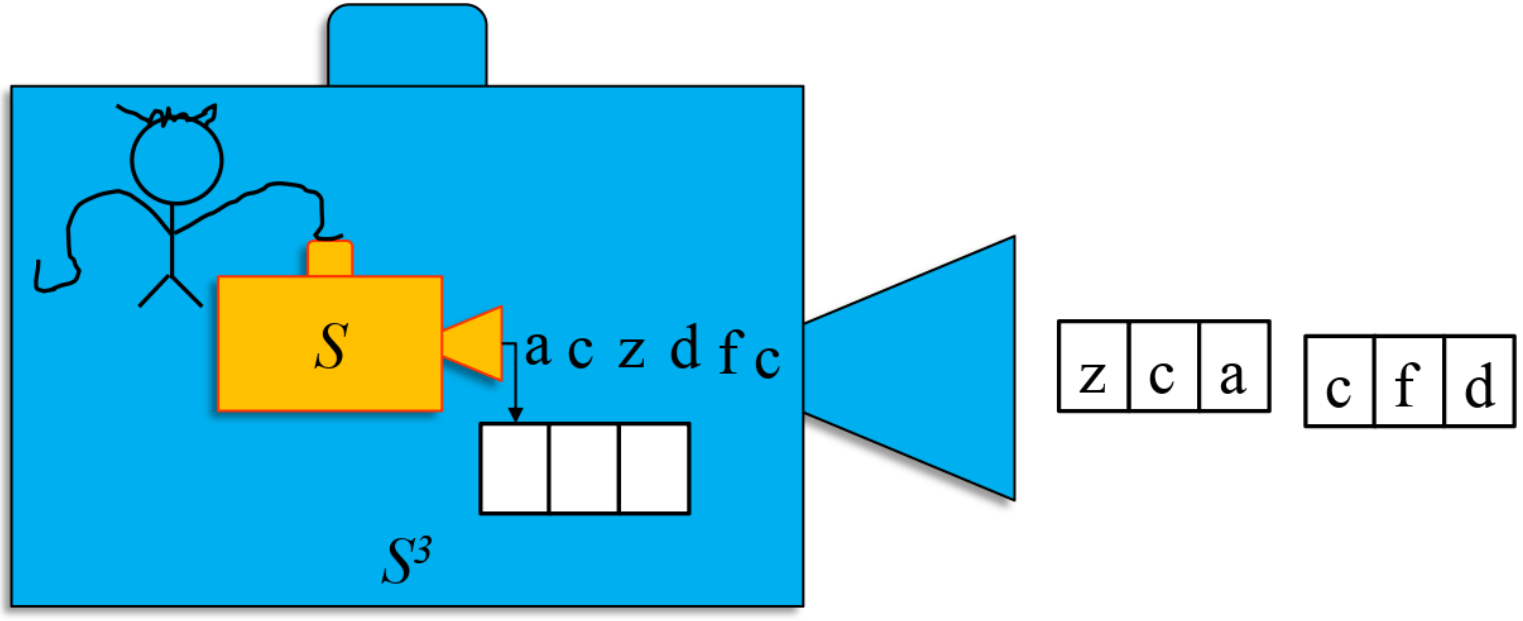
- Source
  - Produit des symboles d'un "alphabet" (∑)
  - Fonctionne "sur demande" (d'où le "bouton")
- Codage
  - Regroupe et transforme les symboles de la source dans un format pouvant être transmis ou sauvegardé
- Canal
  - Peut introduire du bruit
    - symbole reçu ≠ symbole transmis
- Décodage
  - Permet de reconstruire le message original
    - séquence des symboles de source

Source d'informations

- Alphabet
  - Ensemble discret fini $\Sigma = \{\sigma_1, \ldots, \sigma_M\}$
  - Par convention taille de Σ, $|\Sigma| = M$
- Contrôle
  - Un "bouton" qui permet d'obtenir un symbole à la fois
- Principe de la boîte noire
  - Autre que le bouton et un nombre petit d'observations (symboles), on ne peut rien savoir sur le contenu ou fonctionnement de la source (sauf peut-être Alice, mais pas Ève, Irène ou Bob)
- Pourquoi cette abstraction ??
  - Permet de discuter de l'efficacité du codage (théorie de l'information)
  - Permet d'analyser correctement la résistance à certaines menaces
    - Algorithmes de chiffrement
    - Choix de mots de passe et phrases de passe
    - …

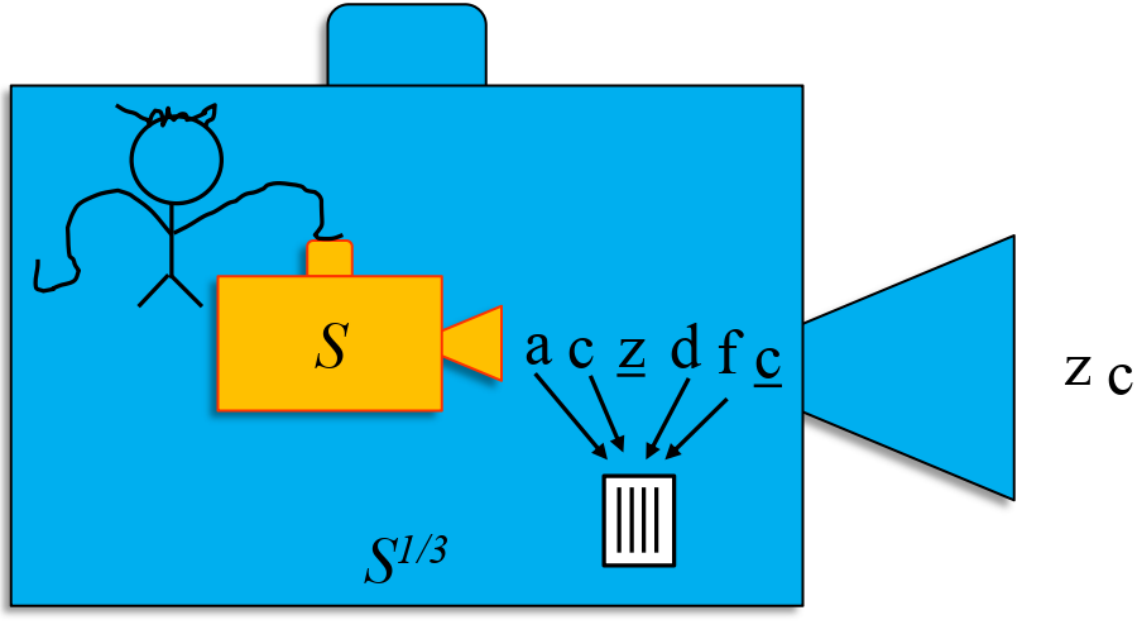
Sources dérivées

- Source par bloc
  - Étant donné une source S, et un entier positif b
  - S^b représente la source obtenue en encapsulant S par une boîte
    - qui mets b symboles de S dans un tampon (« buffer ») avant de les sortir
  - Noter que l'alphabet de S^b est maintenant Σ^b



Sources dérivées

- Source par échantillonnage
  - Étant donné une source *S*, et un entier positif *b*,
  - $S^{1/b}$ représente la source obtenue en encapsulant *S* par une boîte
    - qui émet seulement le 1er symbole de chaque *b* symboles sortie de *S*
  - L'alphabet de $S^{1/b}$ est le même que *S*, soit Σ



Types de source d'information

- Déterministe
  - La boîte « connaît » à l'avance toute la séquence de symboles (potentiellement infinie…)
- Probabiliste
  - La boîte choisit les symboles au fur et à mesure selon une distribution de probabilité
    - **Processus markovien ou "sans mémoire"**
      - p_i = Prob (S => " σ_i"), $\forall$ 1< i < M
      - e.g. Prob (S^b => " σ_i , σ_j") = $p_i\, p_j$
    - **Processus non-markovien**
      - Les probabilités de symboles peuvent dépendre des symboles antérieurs sortis de la source…

Codage

- Translittération
  - Un codage traduit les symboles de source vers un autre « alphabet » $T = \{ \tau_1, \ldots, \tau_N \}$, *(Tau majuscule)*
- Fonction de codage
  - $F : \Sigma \rightarrow T$,
    - $\tau = F(\sigma)$, représente comment le symbole σ devra être transmis
- Fonction de décodage
  - $F^{-1}: T \rightarrow \Sigma$
    - $\sigma' = F^{-1}(\tau')$,
      - Si $\tau' \neq \tau$ alors $\sigma' \neq \sigma$

        il y a eu erreur de transmission (bruit dans le canal)
      - Si $\tau' = \tau$ alors $\sigma' = \sigma$

        transmission sans erreur

        Bob reçoit ce que Alice (source) a émis

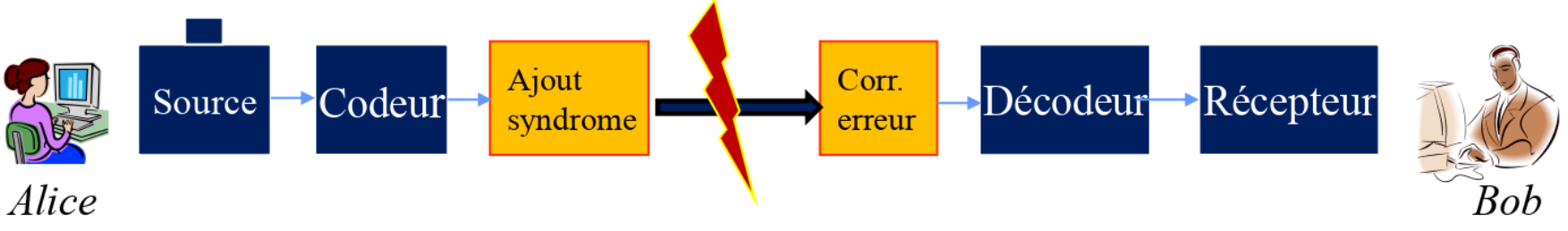
    ➔ F est nécessairement une injection

Correction d'erreur

- Code correcteur d'erreur
  - L'introduction de bruit dans le canal est compensé en utilisant un code correcteur d'erreur dans le codage t.q.
    Prob ($F^{-1}(\tau') = \tau$) → 1, où τ' est le symbole reçu via le canal
- L'efficacité du code correcteur d'erreur
  - Dépend du niveau de bruit introduit par le canal
    → celui-ci peut être mesuré avec l'entropie de Shannon
  - Se mesure également en nombre de bits nécessaires par symbole de source, pour un code qui corrige « presque toutes les erreurs »
- 2e Théorème de Shannon
  - Établit le lien entre l'efficacité du code correcteur d'erreur et le niveau de bruit du canal

Correction d'erreur

- En pratique, la correction d'erreur est souvent Une étape distincte et séparée du codage
  - Chez Alice
    - Codage supplémentaire après codage initial
    - Ajout d'information supplémentaire (*syndrome*)
  - Chez Bob
    - Décodage initial avant décodage final
    - Analyse du syndrome et du message
      - permet de corriger les erreurs (avec haute probabilité)



CRYPTOGRAPHIE I – THÉORIE DE L'INFORMATION – ENTROPIE

POLYTECHNIQUE MONTRÉAL

UNIVERSITÉ D'INGÉNIERIE

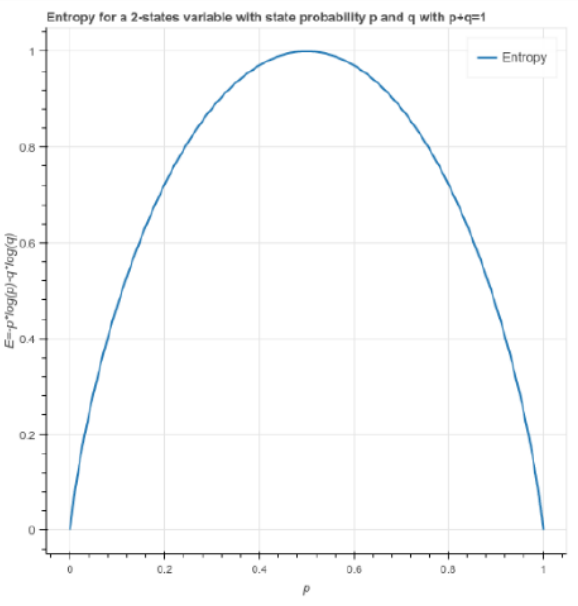
Efficacité du codage - Compression

- Compression
  - Dans certaines circonstances, on voudrait pouvoir coder en utilisant moins de bande passante, p.ex. tel que $N < M$
  - Efficacité du code
    - est mesurée en bits transmis par chaque symbole de source émis
  - 1er Théorème de Shannon
    - Efficacité maximum d'un code compresseur est approximativement égale à $H(S)$
    - Il existe un code compresseur (sans erreur) avec efficacité $H(S) + 1$
  - Qu'est-ce « $H(S)$ » → L'entropie de la source S

Entropie de Shannon

- Définitions
  - $H(S) = \Sigma_i \; p_i \log_2 1/p_i$

- Propriétés
  - Fonction convexe
  - $\Sigma = \{0,1\}$
    - Prob (S="0") = p, Prob (S="1") = q = 1-p
    - Valeur minimale
      - Prob (S="0") = 1; Prob (S="1") = 0
        - ➔ $H(S)$ = 0 **bit**
    - Valeur maximale
      - Prob (S="0") = Prob (S="1") = ½
        - ➔ $H(S)$ = 1 **bit**
  - Σ arbitraire, $|\Sigma| = N$
    - Valeur minimale
      - Prob ($S = \sigma$)=1 pour un σ donné, Prob ($S = \sigma$) = 0 pour tous les autres
        - ➔ $H(S)$ = 0 **bit**
    - Valeur maximale
      - Prob ($S = \sigma_i$) = Prob ($S = \sigma_j$), $\forall\ \sigma_i , \sigma_j \in \Sigma$
        - ➔ $H(S) = \log_2 N$ **bit**



Entropie

- Exemple de calcul d'entropie
  - Pile ou face
    - Alphabet {`pile`, `face`}
    - Probabilité d'occurrence des symboles (p_i): chaque symbole équiprobable avec une probabilité de ½
    - $H(S) = \Sigma_i \; p_i \log_2 1/p_i$
      - `pile` : ½ $\log_2$ (1 / ½) = ½ $\log_2$ 2 = ½ * 1 = ½
      - `face` : ½ $\log_2$ (1 / ½) = ½ $\log_2$ 2 = ½ * 1 = ½
      - $H(S)$ = ½ + ½ = 1 *bit*
  - Alphabet équiprobable
    - Alphabet = {`a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z`}
    - Probabilité d'occurrence des symboles (p_i):
      - chaque symbole équiprobable avec une probabilité de 1/26
    - $H(S) = \Sigma$ 1/26 $\log_2$ 1/(1/26)= 26*1/26 * $\log_{10}(26)/\log_{10} 2$= $= \log_{10}(26)/\log_{10} 2 = 4.7$ *bits (on peut vérifier :* $2^{4.7} = 25.99$*)*

Analyse fréquentielle vs. entropie

- Problème
  - L'entropie de Shannon
    - est définie à partir de *probabilités*
    - s'applique seulement aux sources markoviennes
  - Comment calculer/utiliser l'entropie sur
    - des sources non-markoviennes ?
    - des textes/séquences finies de symboles ?
- « Solution »
  - Fréquence de symbole
    - Soit $S_N = s_1, s_2, \ldots s_N$, $s_i \in \Sigma$, une séquence d'une source S, on définit:

$$f_i(S_N) = \frac{|\{j \mid s_j = \sigma_i\}|}{N}$$

  - Pseudo-entropie
    - Définie/calculée à partir des fréquences (au lieu de probabilités)

Pseudo-entropie

- Pour une séquence finie *SN* $\Psi(S_N) = \sum_i f_i(S_N) \log \frac{1}{f_i(S_N)}$
- Pour une séquence *S* $\Psi(S) = \lim_{N \to \infty} \Psi(S_N)$
- Pour une source d'information quelconque *S*
  - A chaque fois qu'on utilise la source « N fois »
  - On obtient une séquence SN différente de longueur N
  - On calcule la pseudo entropie $\Psi(S_N)$ de cette séquence, qui est elle-même une variable aléatoire
  - Sa valeur espérée $\overline{\Psi(S_N)}$ représente une pseudo-entropie de la source sur des séquences de longueur N
- On considère alors la pseudo-entropie de la source comme étant la limite de cette valeur espérée

$$\Psi(S) = \lim_{N \to \infty} \overline{\Psi(S_N)}$$

Entropie vs. pseudo-entropie

- Pour les sources markoviennes
  - La pseudo-entropie d'une séquence générée par la source va s'approcher de l'entropie
  - Cette convergence est bonne lorsque la taille de la sous-séquence est grande, parce que les fréquences f_i s'approchent des probabilités p_i (loi des grands nombres)
    - Quand $N \rightarrow \infty$, alors $\Psi(S_N) \rightarrow H(S)$
    - Si N est trop petit, alors
      - déductions faites à partir des f_i non valable statistiquement
        ➔ cryptanalyse difficile (voir TP 1)
- Pour les sources non-markoviennes
  - L'entropie $H(S)$ n'est pas vraiment définie,
    - On utilise $\Psi(S)$ à la place (outil de calcul d'entropie TP1)
    - On écrira dans le reste du cours « H(S) »,
      mais on veut vraiment dire $\Psi(S)$ …

Interprétation de l'entropie d'une source

- Interprétation de $H(S)$
  - 1er théorème : Chaque symbole émit par S peut être codé individuellement avec en moyenne $H(S)$ bits
  - Et si on permet que le codage regroupe 2 lettres à la fois ?
    - ➔ Par 1er théorème on peut coder chaque digramme (2 symboles) avec $H(S^2)$ bits, soit $H(S^2)/2$ bits par symbole
    - ➔ Mais si $H(S^2)/2 \leq H(S)$, donc on peut avoir un gain en compression
- Taux de compression
  - Sans compression
    - ➔ log N bits par symbole, dans le pire cas (entropie maximale)
  - Avec compression par bloc de b symbole
    - ➔ $H(S^b)/b$ bits par symbole
  - Taux de compression = $\frac{H(S^b)/b}{\log N}$

Source markovienne vs. non markoviennes

- Source markovienne
  - Si S est markovienne, alors $H(S^b) = b*H(S)$
    - Conséquence: Aucun gain de compression en codant par bloc
    - Intuition: Il n'existe pas de corrélation entre les symboles (distribution de probabilité indépendante), et chaque symbole doit être codé individuellement
- Source non markovienne
  - En général $H(S^b) \leq b*H(S)$
    - Conséquence1: Il y a en général un gain de compression en codant par bloc
    - Intuition:
      - Les probabilités des symboles dépendent des symboles antérieurs
      - Cette « dépendance » statistique peut être exploité par le codage pour réduire le nombre de symboles ou le nombre de bits dans leur codage

        P.ex. en français
        - la lettre « u » suit (presque toujours) la lettre « q »
        - Un sujet est suivi d'un verbe, p.ex. « Je_ » doit être suivi d'un verbe conjugué à la 1e personne du singulier
      - Le gain de compression devrait augmenter en considérant des tailles de blocs plus grandes

Entropie du langage de la source

- En théorie,
  - plus la taille de bloc *b* est grande,
    plus le taux de compression est élevé (jusqu'à une certaine limite)
- Langage associé à une source *S*
  - ensembles de chaines finies générées par *S*
- L'entropie H_L du *langage* associé à la source *S*,

$$H_L(S) = \lim_{b \to \infty} \frac{H(S^b)}{b}$$

  - est le minimum de bits nécessaires (en moyenne) pour coder chaque symbole de chaînes émises par S, même si on permet de coder avec des tailles de blocs arbitraires
  - représente la limite ultime de compression