Nicolas Sendrier

Majeure d’informatique

Introduction la théorie de l’information

Cours n°9

Décodage souple

Canal continu et « log-vraisemblance »

Nous considérons le modèle de canal $(\{0,1\},\mathcal{S},p)$ suivant

– un signal parmi deux possibles est émis (représentant 0 ou 1)

– le signal reçu s est dans un espace de signaux $\mathcal{S}$ selon une certaine loi de transition $p(s \mid b)$, $s \in \mathcal{S}$, $b \in \{0,1\}$

La log-vraisemblance de s est définie par :

$$\ell(s) = \log \frac{p(s \mid 1)}{p(s \mid 0)}$$

où $p(s \mid b)$, $b \in \{0,1\}$, est la (densité de) probabilité de recevoir s lorsque le bit b est émis. La log-vraisemblance est à valeur dans $\mathbf{R}$. C'est une quantité qui ne dépend que de s et des caratéristiques du canal de communication.

Si $\ell(s) > 0$, le signal correspond plutôt à '1', sinon plutôt à '0'.

Vraisemblance maximale et log-vraisemblance

Définition Pour tout $n > 0$, tout $\vec{a} = (a_1, \ldots, a_n) \in \{0,1\}^n$ et tout $\vec{s} = (s_1, \ldots, s_n) \in \mathcal{S}^n$, nous noterons

$$D(\vec{a}, \vec{s}) = \sum_{i=1}^{n} a_i l(s_i)$$

Proposition Soient $n > 0$, $\vec{a} = (a_1, \ldots, a_n) \in \{0,1\}^n$ et $\vec{s} = (s_1, \ldots, s_n) \in \mathcal{S}^n$. Si le canal est symétrique, alors la quantité $P(\vec{a} \mid \vec{s})/P(\vec{a})$ est une fonction croissante de $D(\vec{a}, \vec{s})$.

En particulier, si l'on transmet des mots d'un code binaire de longueur n tirés selon une loi uniforme, le mot de code $(a_1, \ldots, a_n)$ le plus probable est celui qui maximise $\sum_{i=1}^{n} a_i l(s_i)$

Décodage « souple »

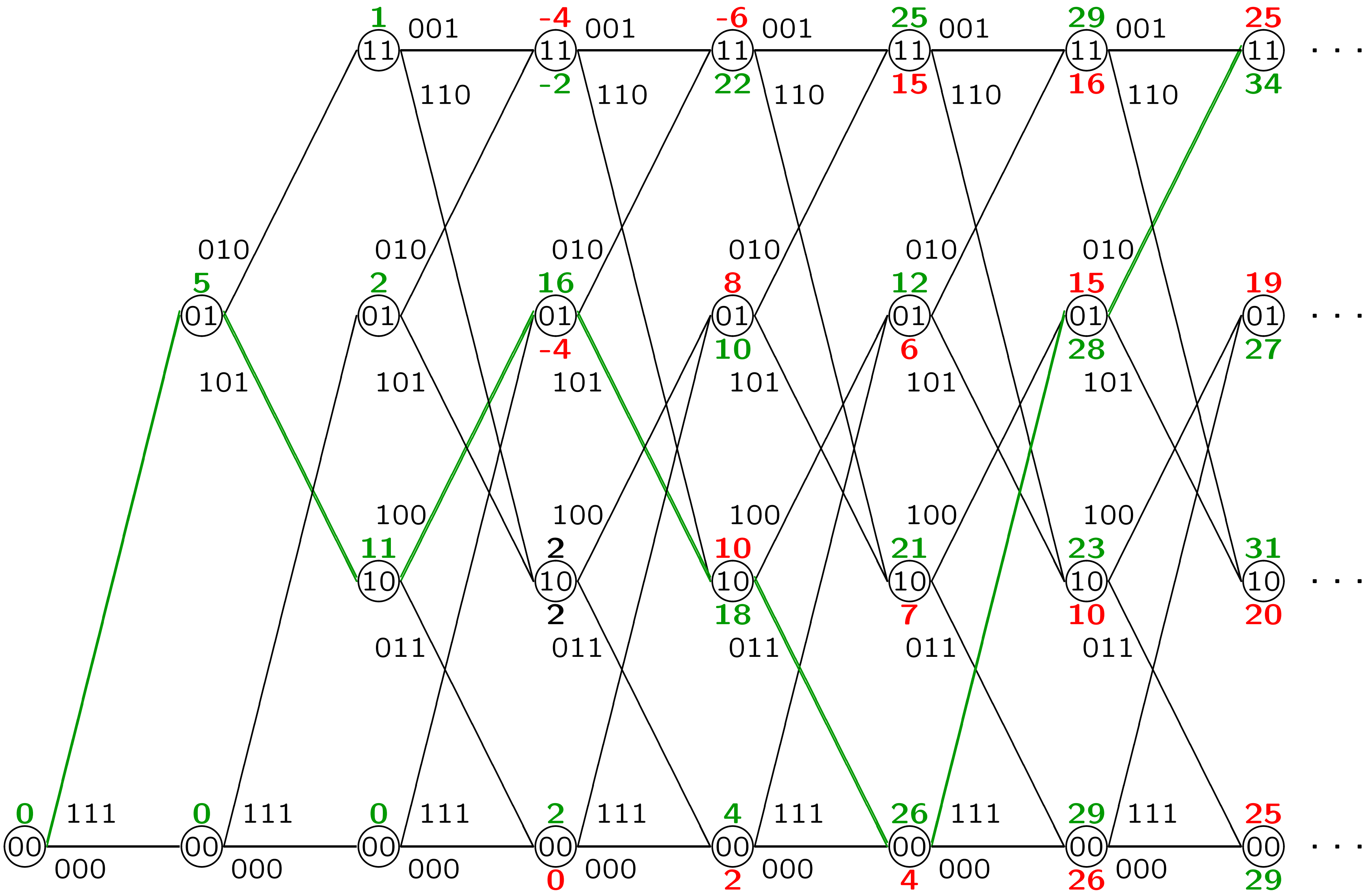
Soit $\mathcal{C}$ un code binaire de longueur n.

Définition Un *algorithme de décodage à entrées souples de* $\mathcal{C}$ est une procédure qui à tout n-uplet de nombres réels associe un mot de $\mathcal{C}$ ou qui échoue.

$$\begin{array}{rccc} \varphi : & \mathbf{R}^n & \longrightarrow & \mathcal{C} \cup \{\text{echec}\} \\ & y & \longmapsto & \varphi(y) \end{array}$$

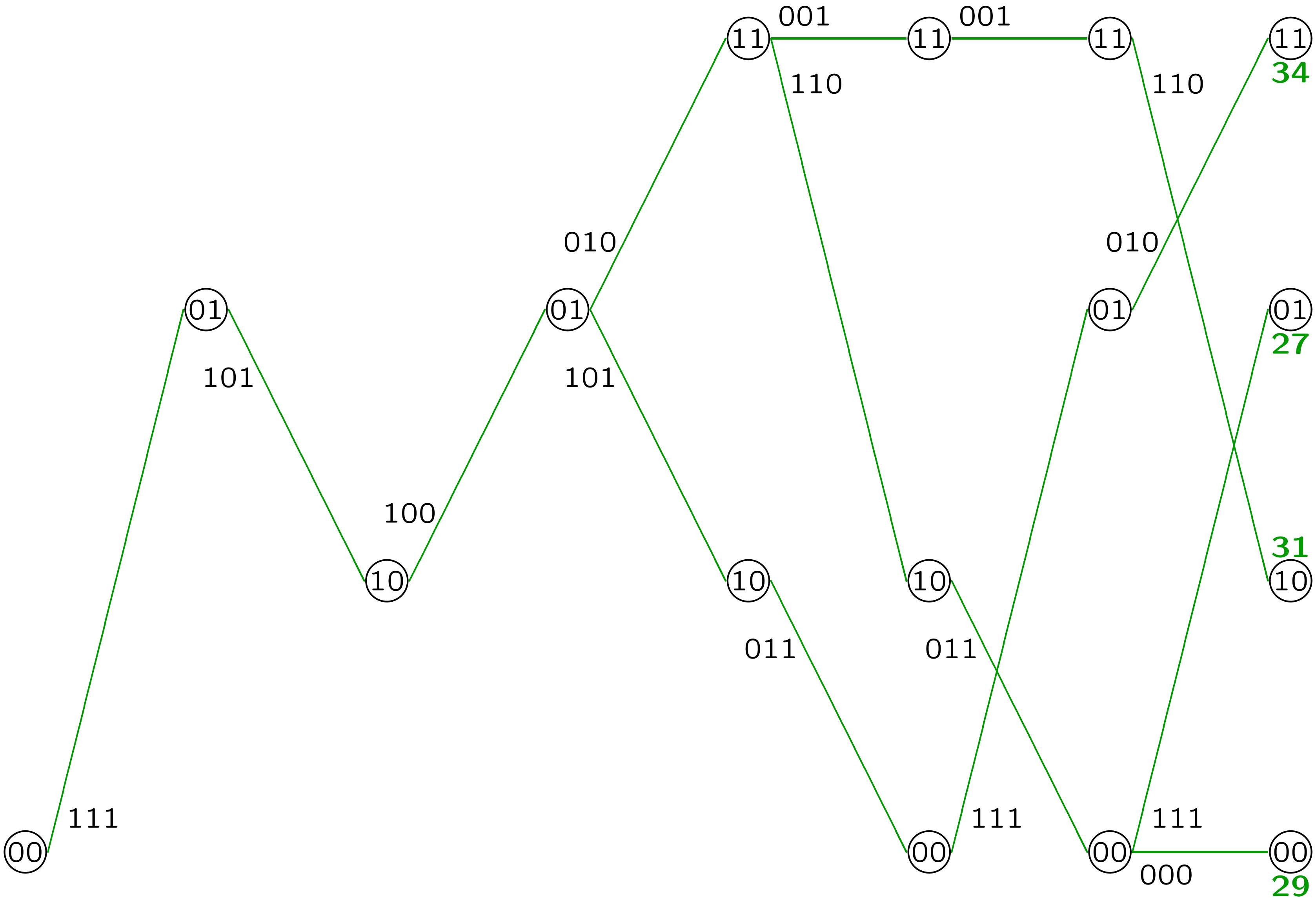
Un tel algorithme sera dit à vraisemblance maximale s'il n'échoue jamais et retourne pour tout $y = (y_1, \ldots, y_n)$ le mot de code le plus probablement émis lorsque les y_i sont les log-vraisemblances des signaux reçus.

Viterbi à sortie souple – Exemple



reçu : $-4, 4, 5$ $\quad 4, -4, 2$ $\quad 5, -4, -5$ $\quad 6, 6, -4$ $\quad -6, 5, 3$ $\quad -6, 4, 4$ $\quad -4, 6, -4$

Viterbi à sortie souple – Exemple



reçu : $-4, 4, 5$ $4, -4, 2$ $5, -4, -5$ $6, 6, -4$ $-6, 5, 3$ $-6, 4, 4$ $-4, 6, -4$

Décodage souple des codes en bloc

Soit $\mathcal{C}$ un code linéaire q-aire de longueur n.

On suppose que l'on possède une information sur la fiabilité des symboles reçus (par exemple la log-vraisemblance dans le cas binaire).

Comment améliorer le décodage ?

Décodage des effacements – Codes poinçonnés

Un effacement est une « erreur localisée ». Pour tout code de distance minimale d, il existe un algorithme de décodage corrigeant $d-1$ effacements. D'une façon générale, un effacement est « deux fois plus facile » à corriger qu'une erreur.

Définition Pour tout code $\mathcal{C}$ de longueur n, le code $\mathcal{C}$ poinçonné en i est défini par

$$\bar{\mathcal{C}}_i = \{(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \mid (x_1, \ldots, x_n) \in \mathcal{C}\}$$

D'une façon générale, pour tout $I \subset \{1, 2, \ldots, n\}$, $\bar{\mathcal{C}}_I$ est le code poinçonné en I.

Proposition Si $\mathcal{C}$ a pour distance minimale $\rho = |I| < d$, alors $|\bar{\mathcal{C}}_I| = |\mathcal{C}|$ et la distance minimale de $\bar{\mathcal{C}}_I$ est au moins égale à $d - \rho$.

Proposition Pour tout code de distance minimale d, il existe un algorithme de décodage corrigeant ν erreurs et ρ effacements ssi

$$2\nu + \rho < d$$

Algorithme

Soit $\mathcal{C}$ un code de longueur n et de distance minimale d (supposée impaire). Pour tout $I \subset \{1, 2, \ldots, n\}$, $|I| < d$, soit φ_I un décodeur pour $\mathcal{C}_I$ borné par $(d - |I| - 1)/2$.

L'algorithme suivant

entrée : $y \in \mathbf{F}_q^n$

$L \leftarrow \{\}$; $I \leftarrow \{\}$

tant que $|I| < d$

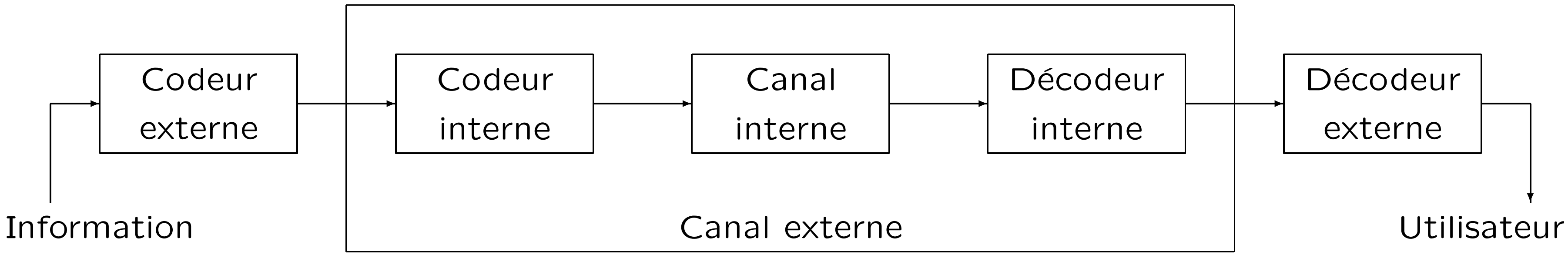
$L \leftarrow L \cup \{\varphi_I(y)\}$ *// les positions de y dans I sont ignorées*

$I \leftarrow I \cup \{$les 2 positions les moins fiables non encore effacées$\}$

retourner $L \setminus \{$echec$\}$

retourne un ensemble (éventuellement vide) d'au plus $d/2$ mots de codes, le mot décodé sera celui qui maximise un critère de vraisemblance (par exemple la log-vraisemblance dans le cas binaire).

Décodage des codes concaténés

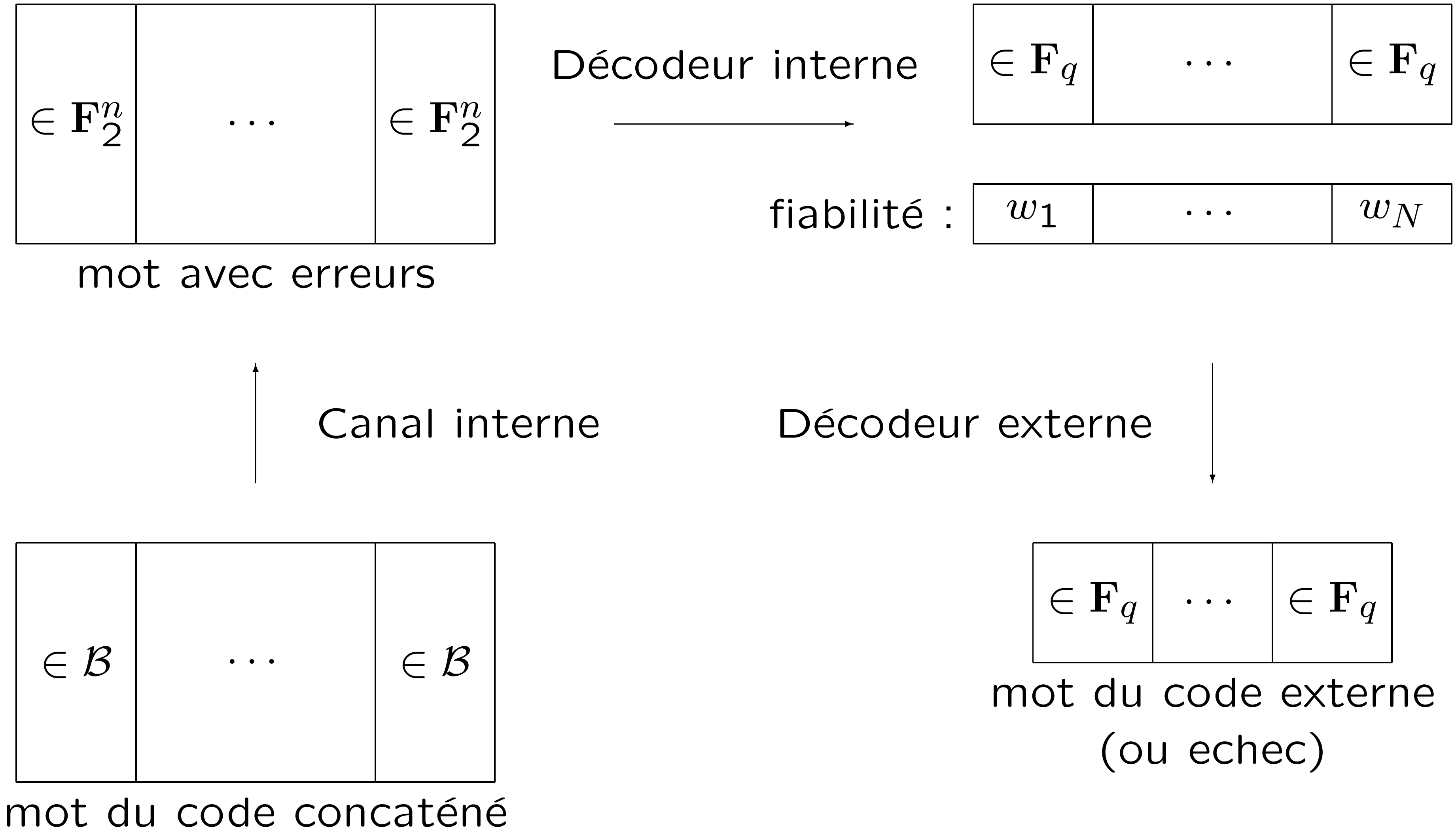


L'idée est d'obtenir une information de fiabilité sur la sortie du décodeur interne, donc sur l'entrée du décodeur externe.

Pour un code concaténé avec un code interne en bloc, la fiabilité sera inversement proportionnelle au nombre d'erreurs corrigées.

Décodage des codes concaténés (2)

code interne binaire $\mathcal{B}[n,k]$, code externe $\mathcal{E}[N,K]$ sur $\mathbf{F}_q$, $q = 2^k$



Entrée et sortie souples

On peut définir un décodeur à entrée et sortie souples

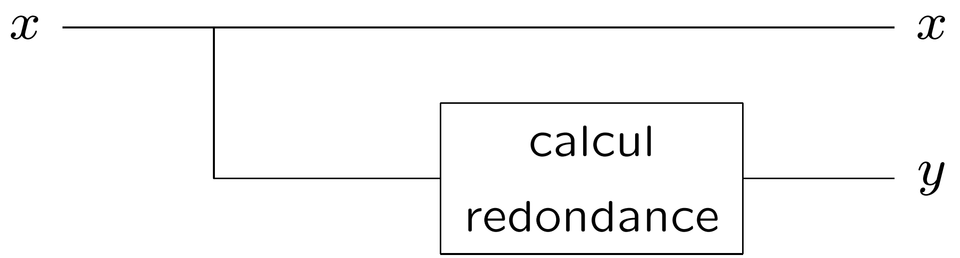
$$\begin{array}{rccc} \varphi : & \mathbf{R}^n & \longrightarrow & \mathbf{R}^n \\ & y & \longmapsto & \varphi(y) = (z_1, \ldots, z_n) \end{array}$$

où $z_i = \log(P_{1,i}/P_{0,i})$ avec $P_{0,i}$ et $P_{1,i}$ des *estimations* des probabilités d'avoir émis 0 ou 1 « sachant tout ce qu'on sait » (canal, code, *toutes* les sorties).

La qualité du décodeur dépend de la qualité de l'estimation.

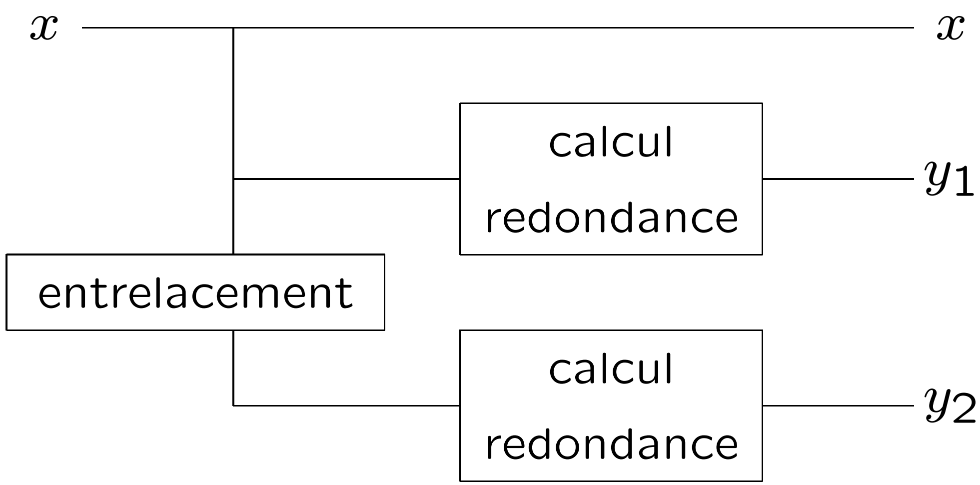
Turbo codes

Codeur convolutif
systématique



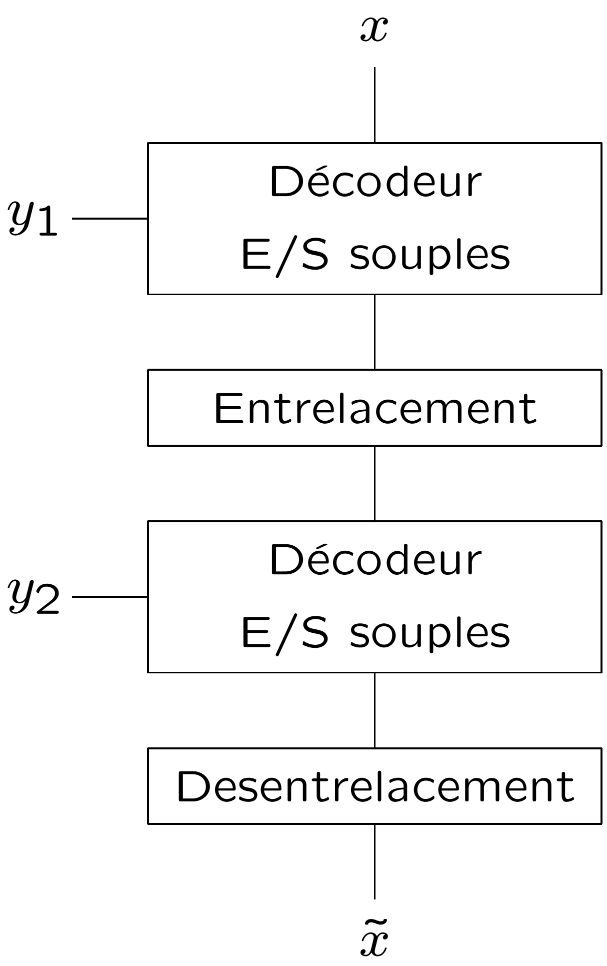
Turbo codeur

l'entrelacement est une permutation d'un bloc de qq centaines à qq milliers de bits



Décodage des turbo codes

Étage du décodeur



Décodeur complet