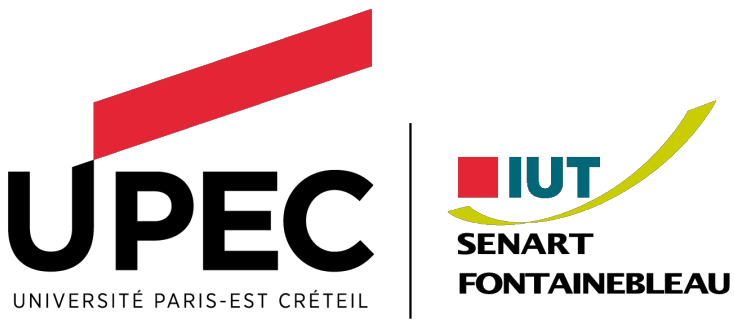
Fondements de l’informatique

Codes Détecteurs et Correcteurs d’erreur

Florent Madelaine

Fondements de l’informatique



Plan

Introduction

L'informatique ne concerne pas seulement le calcul et la programmation, c'est aussi la science de l'information.

Ainsi on s'intéresse au chiffrement et à la signature (cryptographie), à la compression d'information (avec ou sans perte), à l'indexation de données (algo du texte, bdd), *etc*

Ces problématiques engendrent à leur tour la recherche d'algorithmes efficaces et des questionnement en calculabilité ou en complexité.

Nous nous concentrons ici sur les mécanismes permettant de favoriser la transmission l'information sur un canal qui fait des erreurs.

Détection d’erreur

Exemples de la vie courante

comme premier exemple pour la détection d’erreur, on peut considérer les numéros d’identification ayant des clés, par exemple dans un RIB ou un numéro INSEE.

Le numéro est suivi d’une clé dont les chiffres sont calculés à partir du numéro.

On ajoute donc une information redondante, qui permet de détecter une erreur de frappe. Avec plus de redondance on peut dans certains cas corriger (avec une bonne probabilité).

Nobody's perfect

Canaux de transmission et supports de stockage

Ils sont forcément imparfaits !

- Lors du transfert et/ou du stockage d'information, des erreurs surviennent.
- On souhaite détecter les erreurs de transmission et/ou de stockage et les corriger.
- C'est le rôle des codes détecteurs et correcteurs d'erreurs.
- On utilise une stratégie de surcodage
  - Plus efficace que de recommencer trop fréquemment la transmission ou de stocker l'information en la dupliquant.

Canal Binaire symétrique

- Nous utilisons le modèle idéalisé et quelque peu simpliste du canal binaire symétrique paramétré par une probabilité $0 \leqslant p \leqslant 1$ fixée.
- Chaque bit b transmis est mal envoyé, c'est-à-dire transformé en $b + 1 \mod 2$ avec probabilité p.
- Les erreurs sont indépendantes au cours du temps.

Exemples

- probabilité de mal envoyer 2 bits de suite : $p \times p$
  (car les deux événements sont indépendants)
- probabilité de mal envoyer 1 bit, puis envoyer correctement le suivant, puis un autre à nouveau mal : $p \times (1 - p) \times p$

Codage et décodage

Un peu de vocabulaire

Avant émission

- L'information est surcodée par ajout de bits de contrôle
- Le mot binaire obtenu est le message émis

Codage et décodage

Un peu de vocabulaire

Avant émission

- L'information est surcodée par ajout de bits de contrôle
- Le mot binaire obtenu est le message émis

C'est le codage

Codage et décodage

Un peu de vocabulaire

Avant émission

- L'information est surcodée par ajout de bits de contrôle
- Le mot binaire obtenu est le message émis

C'est le codage

Après réception

- À partir du message reçu, en utilisant les bits de contrôle
- On détecte les erreurs
- On corrige le message reçu pour retrouver l'information initiale.

Codage et décodage

Un peu de vocabulaire

Avant émission

- L'information est surcodée par ajout de bits de contrôle
- Le mot binaire obtenu est le message émis

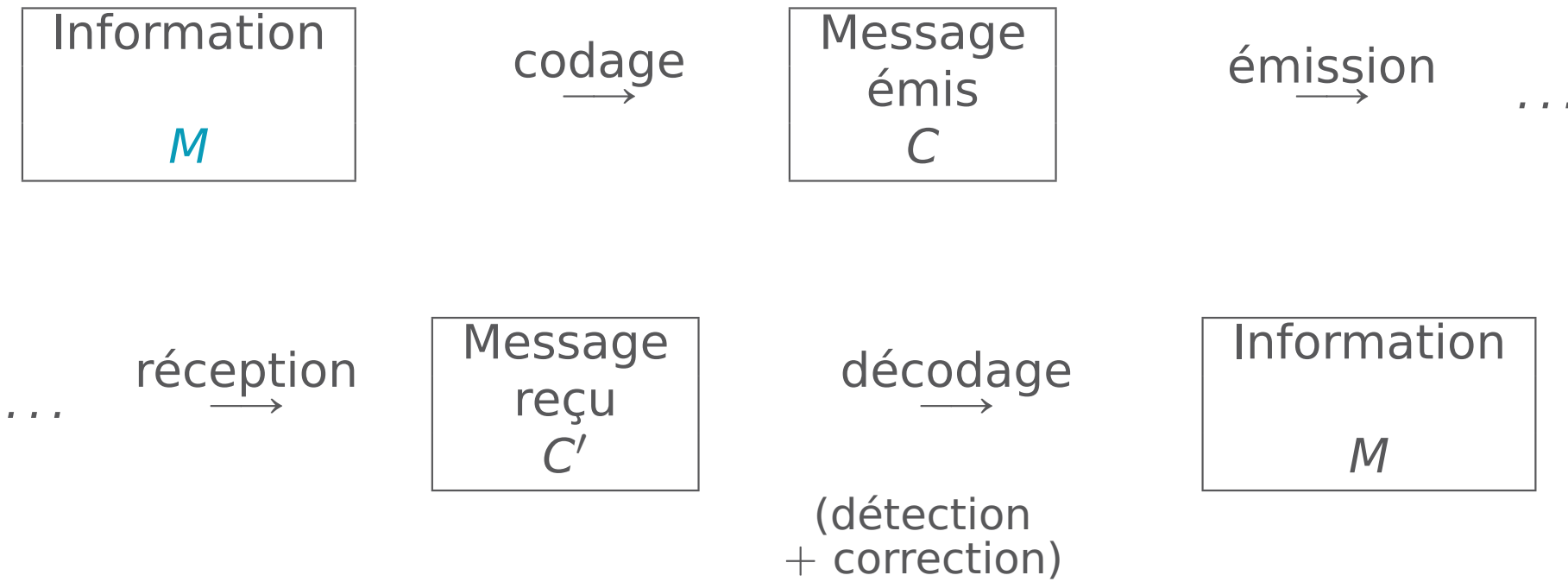
C'est le codage

Après réception

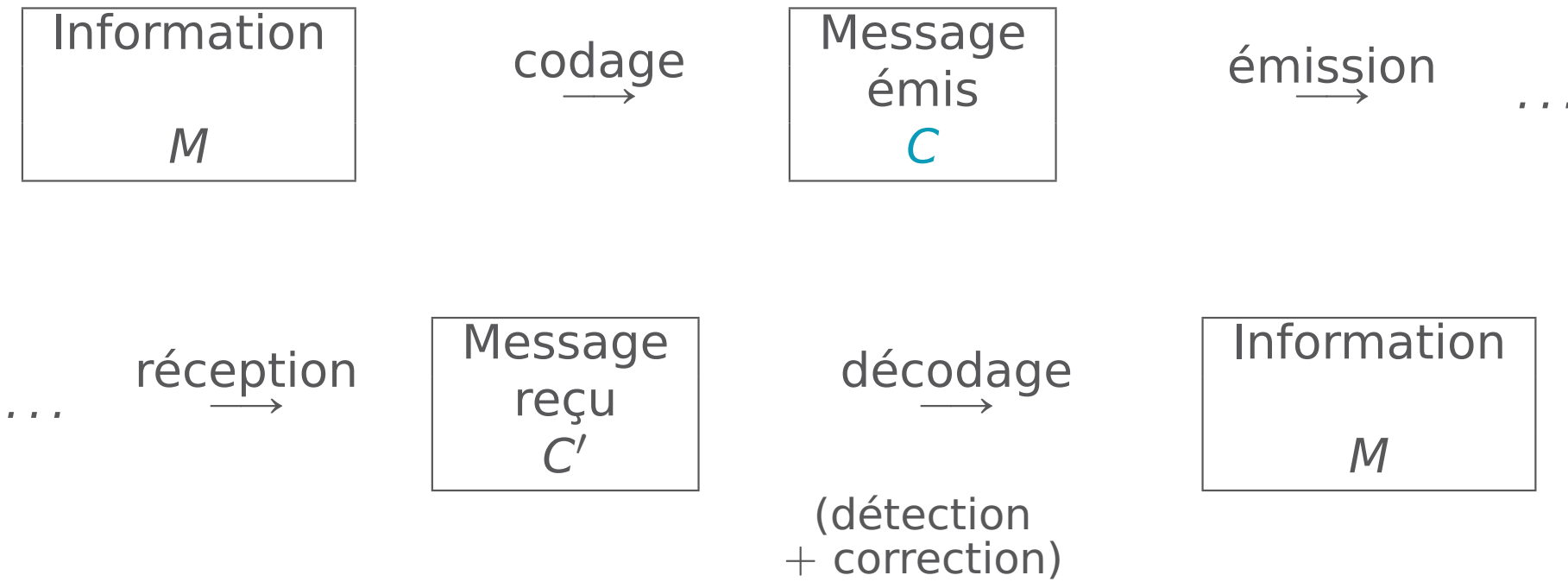
- À partir du message reçu, en utilisant les bits de contrôle
- On détecte les erreurs
- On corrige le message reçu pour retrouver l'information initiale.

C'est le décodage

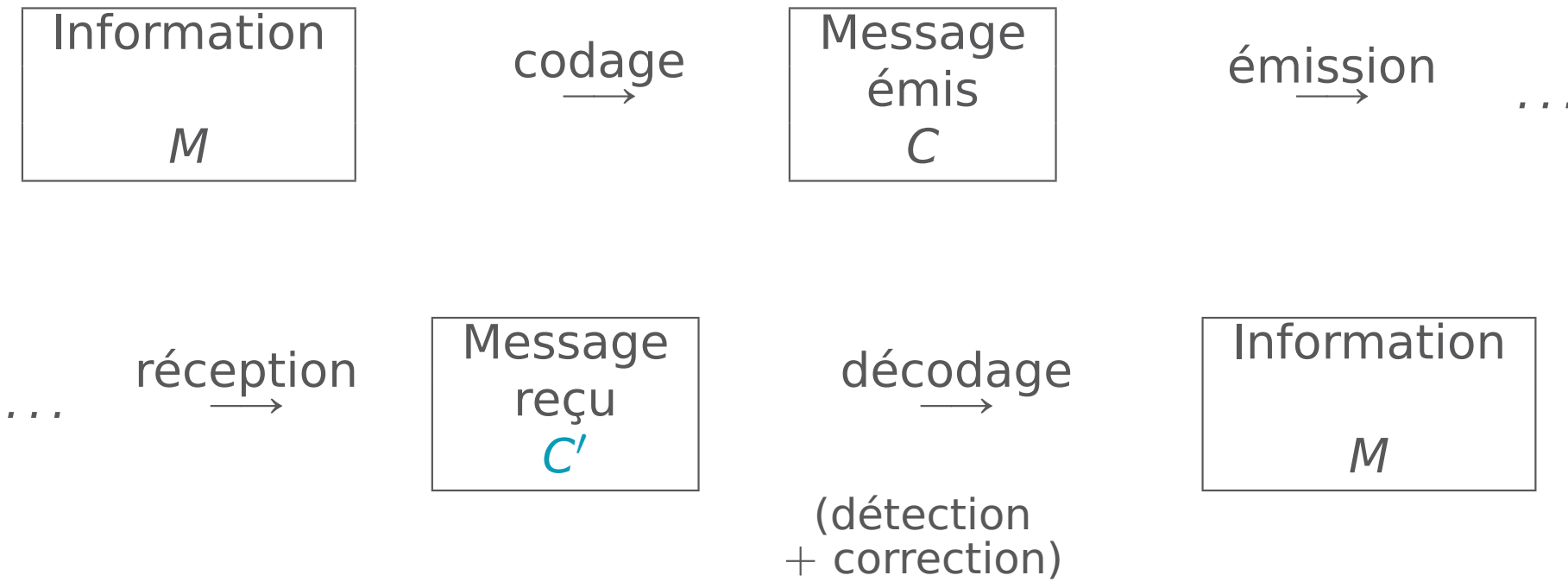
Résumé


Information
M
codage
Message émis
C
émission
...
...
réception
Message reçu
C'
décodage
(détection + correction)
Information
M

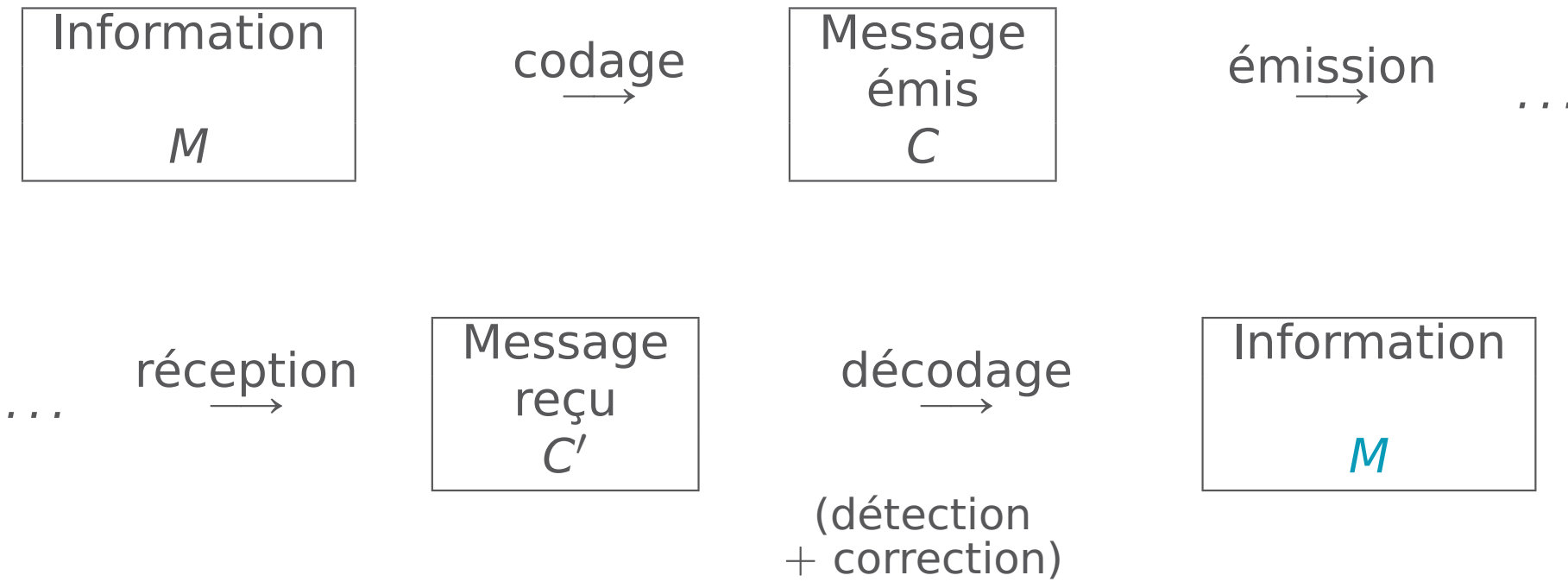
Résumé


Information
M
codage
Message émis
C
émission
...
...
réception
Message reçu
C'
décodage
(détection + correction)
Information
M

Résumé


Information
M
codage
Message émis
C
émission
...
...
réception
Message reçu
C'
décodage
(détection + correction)
Information
M

Résumé


Information
M
codage
Message
émis
C
émission
. . .
. . .
réception
Message
reçu
C′
décodage
(détection
+ correction)
Information
M

Code de type $\mathcal{C}_{n,m}$

- L'information est découpée en blocs de longueur fixe :

codage

mot d'information
m bits

$\longrightarrow$

mot de code
n bits
=
m bits d'information
+
r bits de contrôle

- Redondance : $r = n - m =$ nombre de bits de contrôle
- Rendement : $\rho = \frac{m}{n} = \frac{m}{m+r}$

Code de type $\mathcal{C}_{n,m}$

- L'information est découpée en blocs de longueur fixe :

mot d'information	codage	mot de code
m bits	$\longrightarrow$	n bits = m bits d'information + r bits de contrôle

- Redondance : $r = n - m =$ nombre de bits de contrôle
- Rendement : $\rho = \frac{m}{n} = \frac{m}{m+r}$

Un code pour détecter mais pas corriger

Le code de parité

- On ajoute 1 seul bit de contrôle qui est la somme des autres bits modulo 2
- Si il y a 1 erreur (et plus généralement un nombre impair d'erreurs) alors on peut le détecter au décodage
- Il suffit de calculer la somme des bits reçus modulo 2 : si on obtient 1, alors il y a nécessairement une erreur.

Dans ce cas, $r = 1$

Par exemple pour $m = 4$.

Un code pour détecter mais pas corriger

Le code de parité

- On ajoute 1 seul bit de contrôle qui est la somme des autres bits modulo 2
- Si il y a 1 erreur (et plus généralement un nombre impair d'erreurs) alors on peut le détecter au décodage
- Il suffit de calculer la somme des bits reçus modulo 2 : si on obtient 1, alors il y a nécessairement une erreur.

Dans ce cas, $r = 1$

Par exemple pour $m = 4$.
Info. 1001
Code. 10010
Reçu. 10110

Un code pour détecter mais pas corriger

Le code de parité

- On ajoute 1 seul bit de contrôle qui est la somme des autres bits modulo 2
- Si il y a 1 erreur (et plus généralement un nombre impair d'erreurs) alors on peut le détecter au décodage
- Il suffit de calculer la somme des bits reçus modulo 2 : si on obtient 1, alors il y a nécessairement une erreur.

Dans ce cas, $r = 1$

Par exemple pour $m = 4$.
Info. 1001
Code. 10010
Reçu. 10110
Somme non nulle modulo 2 : erreur détectée.

Un code pour détecter mais pas corriger

Le code de parité

- On ajoute 1 seul bit de contrôle qui est la somme des autres bits modulo 2
- Si il y a 1 erreur (et plus généralement un nombre impair d'erreurs) alors on peut le détecter au décodage
- Il suffit de calculer la somme des bits reçus modulo 2 : si on obtient 1, alors il y a nécessairement une erreur.

Dans ce cas, $r = 1$

Par exemple pour $m = 4$.
Info. 1001
Code. 10010
Reçu. 10111

Un code pour détecter mais pas corriger

Le code de parité

- On ajoute 1 seul bit de contrôle qui est la somme des autres bits modulo 2
- Si il y a 1 erreur (et plus généralement un nombre impair d'erreurs) alors on peut le détecter au décodage
- Il suffit de calculer la somme des bits reçus modulo 2 : si on obtient 1, alors il y a nécessairement une erreur.

Dans ce cas, $r = 1$

Par exemple pour $m = 4$.
Info. 1001
Code. 10010
Reçu. 10111
Somme nulle modulo 2 : 2 erreurs mais pas détectées.

Un code pour détecter et corriger

Le code par répétition

- On répète chaque bit du message k fois (k impair).
- Si il y a strictement moins de k erreurs alors on peut le détecter puisque tous les bits reçus devraient être les mêmes et que ce n'est pas le cas.
- Pour la correction, on corrige au bit le plus fréquent (toujours possible puisque k est impair)

Dans ce cas, $r = (k-1)m$

Pour $m = 1$ et $k = 3$ répétitions, soit $n = 3$ et $r = 2$.
Info. 1
Code. 111
Reçu. 110

Un code pour détecter et corriger

Le code par répétition

- On répète chaque bit du message k fois (k impair).
- Si il y a strictement moins de k erreurs alors on peut le détecter puisque tous les bits reçus devraient être les mêmes et que ce n'est pas le cas.
- Pour la correction, on corrige au bit le plus fréquent (toujours possible puisque k est impair)

Dans ce cas, $r = (k-1)m$

Pour $m = 1$ et $k = 3$ répétitions, soit $n = 3$ et $r = 2$.
Info. 1
Code. 111
Reçu. 110
Pas tous égaux : erreur détectée.
On corrige à 1 (bit le plus fréquent).

Un code pour détecter et corriger

Le code par répétition

- On répète chaque bit du message k fois (k impair).
- Si il y a strictement moins de k erreurs alors on peut le détecter puisque tous les bits reçus devraient être les mêmes et que ce n'est pas le cas.
- Pour la correction, on corrige au bit le plus fréquent (toujours possible puisque k est impair)

Dans ce cas, $r = (k-1)m$

Pour $m = 1$ et $k = 3$ répétitions, soit $n = 3$ et $r = 2$.
Info. 1
Code. 111
Reçu. 010

Un code pour détecter et corriger

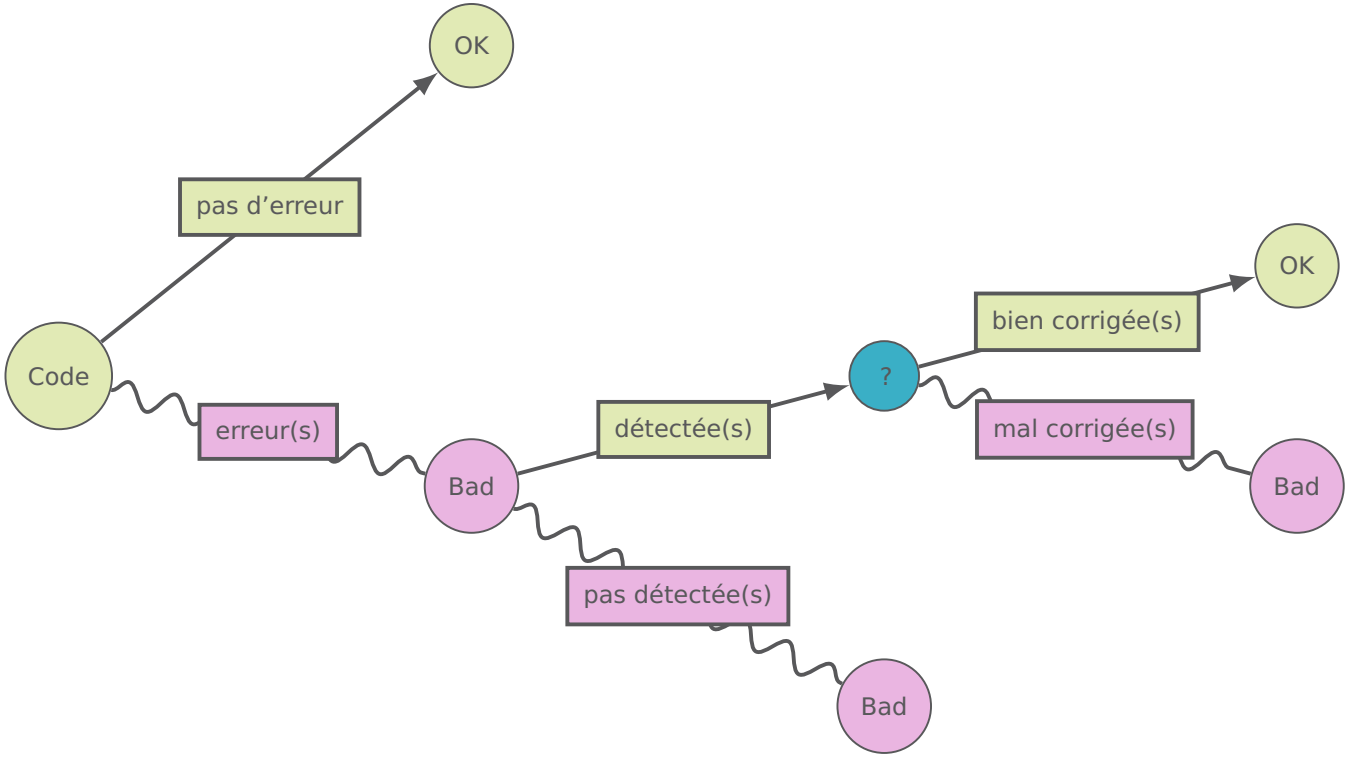
Le code par répétition

- On répète chaque bit du message k fois (k impair).
- Si il y a strictement moins de k erreurs alors on peut le détecter puisque tous les bits reçus devraient être les mêmes et que ce n'est pas le cas.
- Pour la correction, on corrige au bit le plus fréquent (toujours possible puisque k est impair)

Dans ce cas, $r = (k-1)m$

Pour $m = 1$ et $k = 3$ répétitions, soit $n = 3$ et $r = 2$.
Info. 1
Code. 111
Reçu. 010
Pas tous égaux : erreur détectée.
On corrige à 0 (bit le plus fréquent).

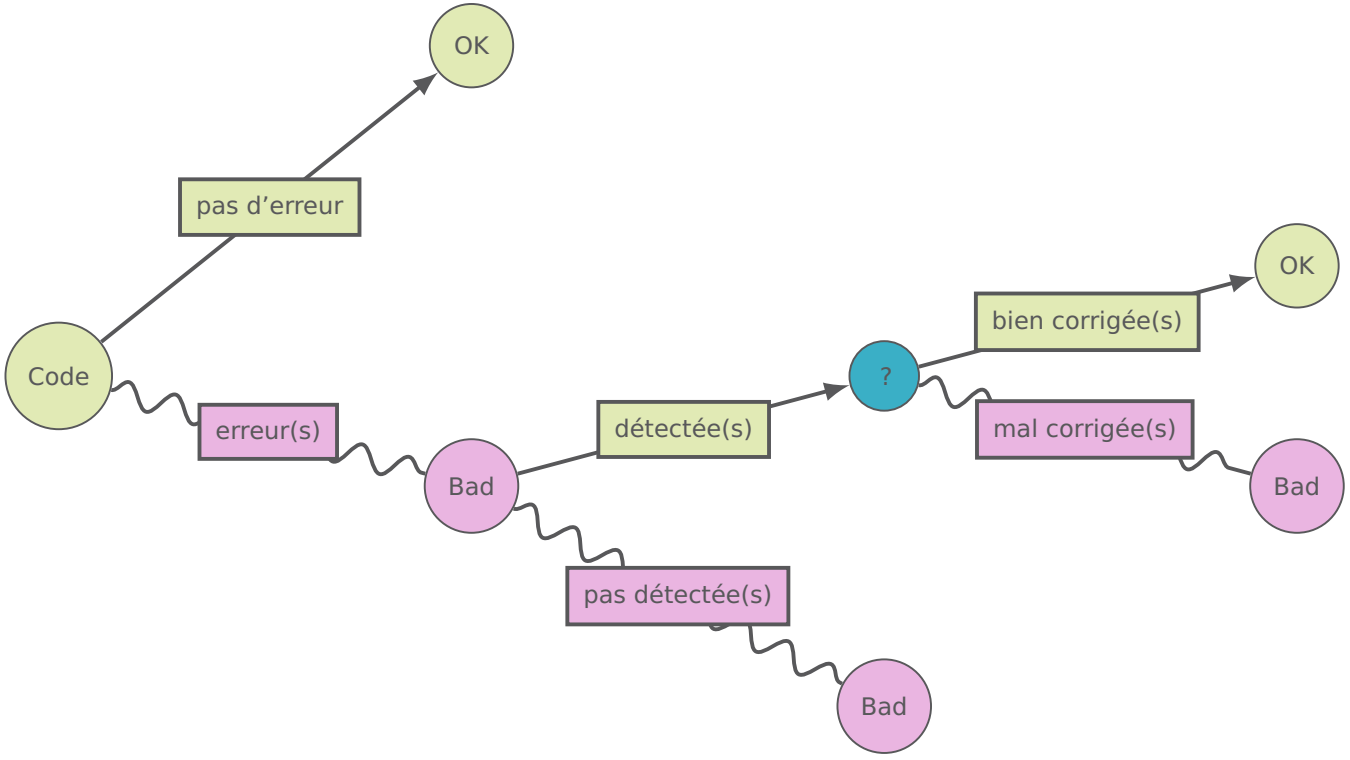
La perfection n’est pas possible



Quel est l’intérêt ?

On fait le pari que les bons cas seront beaucoup plus fréquents que les mauvais, et que en moyenne utiliser un code détecteurs et correcteurs d’erreurs sera mieux que de ne rien faire.

La perfection n’est pas possible



Quel est l’intérêt ?

On fait le pari que les bons cas seront beaucoup plus fréquents que les mauvais, et que en moyenne utiliser un code détecteurs et correcteurs d’erreurs sera mieux que de ne rien faire.

Pourquoi ça marche ?

Hypothèse de travail

- Pour $0 \leqslant k < k' \leqslant n$, il est beaucoup plus probable de recevoir un message avec k erreurs qu'avec k' erreurs
- Quand on ne détecte aucune erreur dans le message reçu, il est très probable qu'il ne contient effectivement aucune erreur
- Quand on corrige un message erroné par le mot de code le plus proche, il est très probable qu'il soit corrigé correctement

Remarque

- Pour tous les exemples réalistes, ceci est vrai.
- On verra que c'est presque toujours le cas dans notre cadre de travail théorique (le canal binaire symétrique).

Vecteur de différence

C_1 et C_2 mots de n bits

- $C_1 = 01010110$
- $C_2 = 11010010$

Vecteur de différence entre C_1 et C_2

- Ou exclusif
- $D = C_1 \oplus C_2 = 10000100$
- Position des bits différents

Rappel

- On a $D = C_1 \oplus C_2$
- Mais aussi $C_1 = C_2 \oplus D$ (propriété du "ou exclusif")

Distance de Hamming

C_1 et C_2 mots de n bits

- $C_1 = 01010110$
- $C_2 = 11010010$

Distance de Hamming entre C_1 et C_2

- Nombre de bits dont ils diffèrent, ici 2
- Poids (nombre de 1) du vecteur de différence
- $w(C_1 \oplus C_2) = w(10000100) = 2$

Cas des codes

Vecteur d'erreur et Nombre de bits erronés

$C_1 = 01010110$ message émis
$C_2 = 11010010$ message reçu

- Vecteur d'erreur $E = C_1 \oplus C_2 = 10000100$
  - Position des bits erronés
- Distance de Hamming $w(C_1 \oplus C_2) = w(E) = 2$
  - Nombre de bits erronés

Remarque

- On a dit que quand on détecte un message erroné, on le corrige en le remplaçant par le mot de code "le plus proche".
- On est maintenant en mesure de préciser : "le plus proche" selon la distance de Hamming
- C'est-à-dire avec le moins de bits à modifier

Distance minimale du code

Définition

La distance minimale du code (notée d)

- Plus petite distance de Hamming entre deux mots de code différents
- Plus petit nombre de bits dont diffèrent deux mots de code différents

Exemples

- Code de parité $\mathcal{C}_{4,1}$: tous les mots qui ont un
  - nombre pair de 1 sont des mots de codes et seulement ceux là.
  - $d = 2$
- Le code par répétition $\mathcal{C}_{3,1}$
  - Les seuls mots de code sont 000 et 111
  - $d = 3$

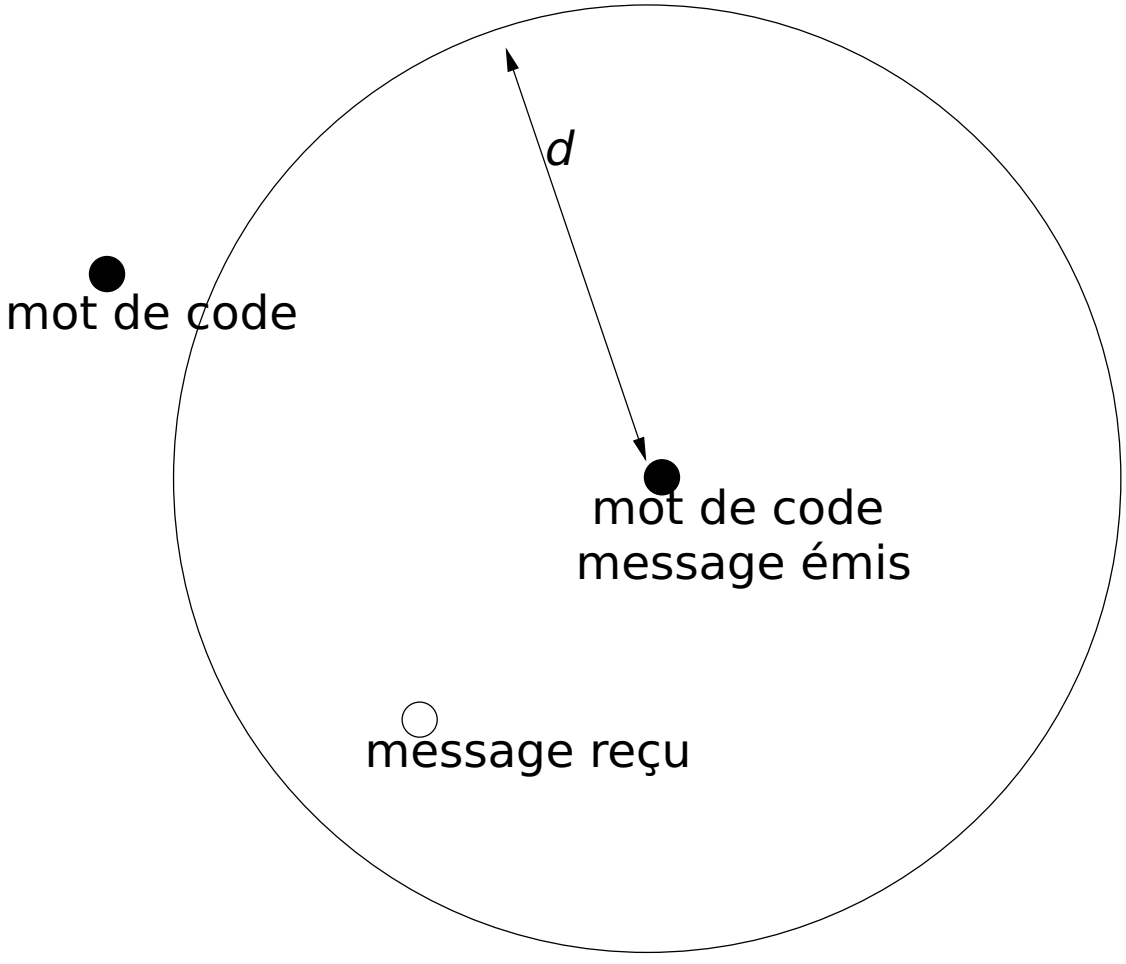
Cas général

- Code $\mathcal{C}_{n,m}$: il faudrait prendre toutes les paires de mots de code différents, calculer à chaque fois leur distance de Hamming, et garder la plus petite valeur obtenue... Il y a 2^m mots de code...

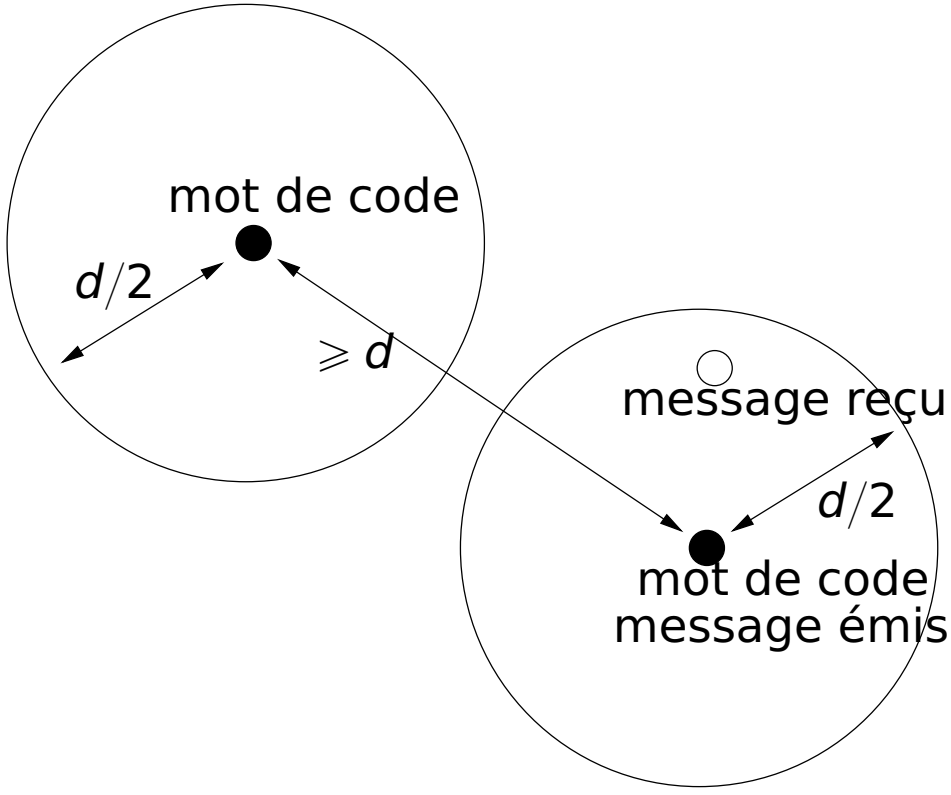
  C'est long

- On verra qu'on peut aller plus vite pour les codes linéaires

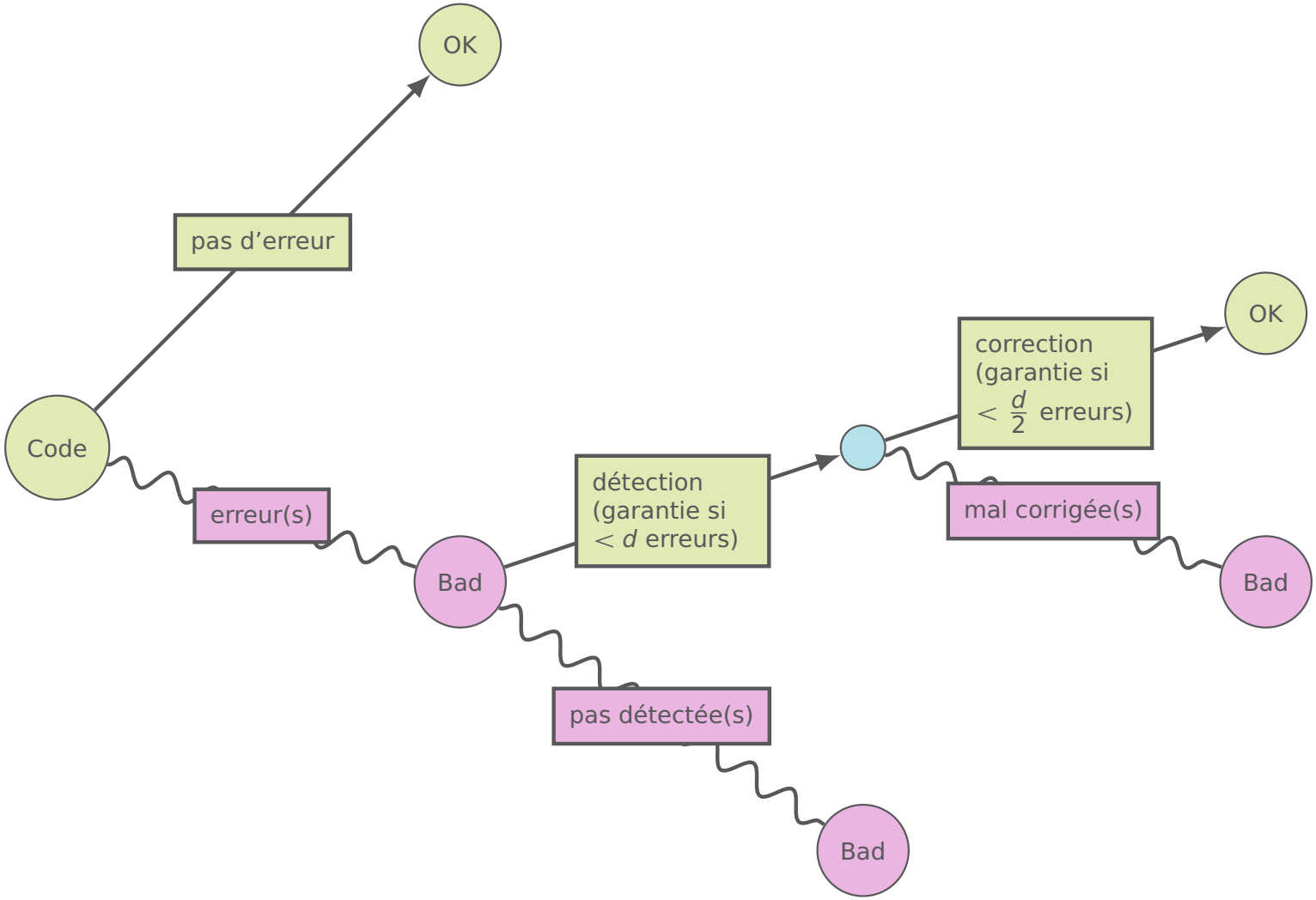
Messages erronés avec strictement moins de d erreurs : TOUS DÉTECTÉS



Messages erronés avec strictement moins de $d/2$ erreurs : TOUS BIEN CORRIGÉS



La distance minimale mesure l'efficacité du code



Notion essentielle

Mesurer l'efficacité du code

- Messages erronés avec

  Strictement moins de d erreurs : TOUS DÉTECTÉS

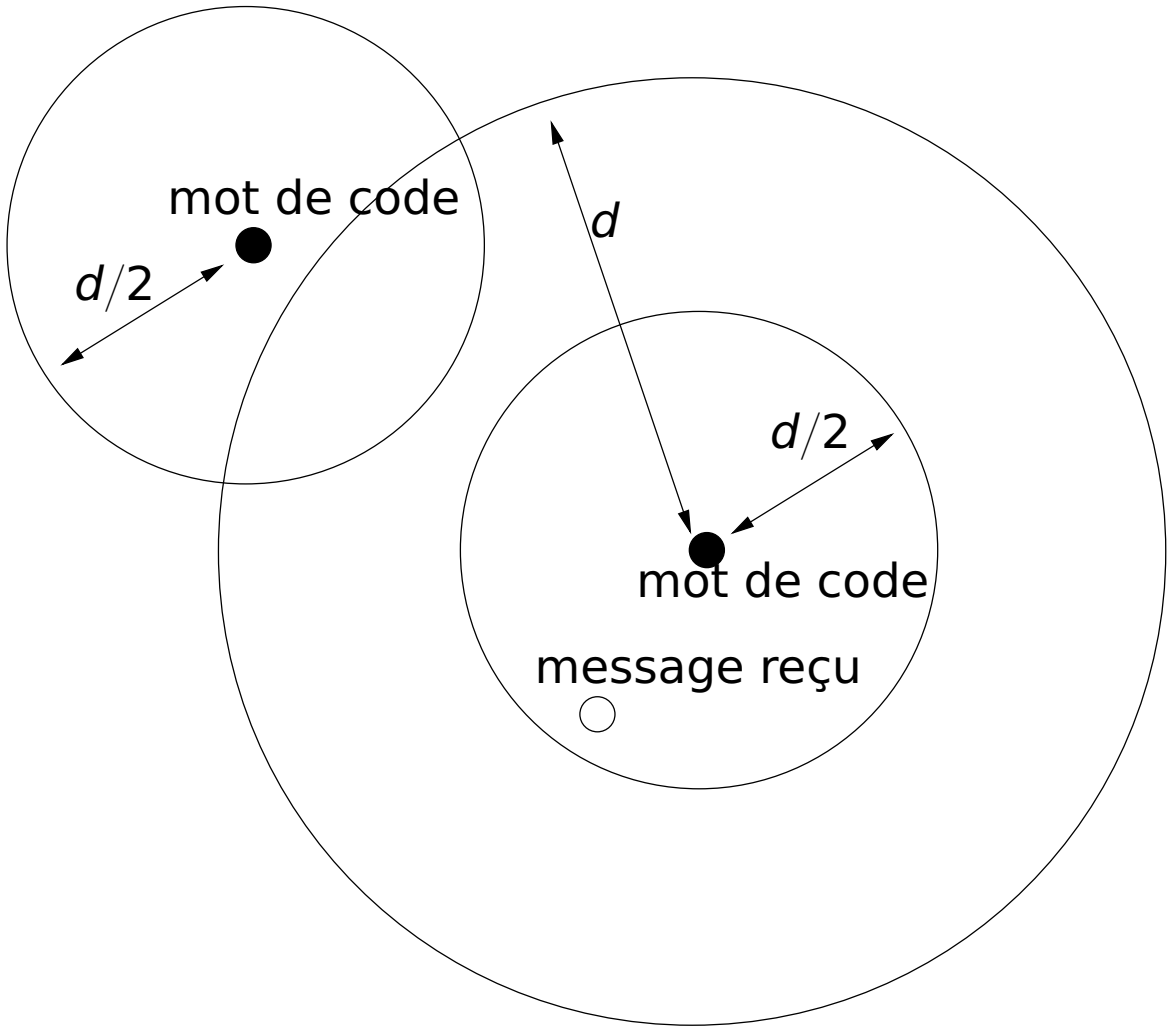
  Les autres : certains pas détectés

- Messages erronés avec

  Strictement moins de $d/2$ erreurs : TOUS BIEN CORRIGÉS

  Les autres : certains mal corrigés

Résumé



Retour sur le type des codes

La distance minimale du code joue un rôle tellement important en théorie et en pratique qu'on l'introduit dans la description du type des codes :

Code de type $\mathcal{C}_{n,m,d}$

- Code de parité : $\mathcal{C}_{8,7,2}$
- Code par répétition : $\mathcal{C}_{3,1,3}$

Pourquoi choisir un code linéaire ?

- On peut coder, détecter, corriger en utilisant des matrices.
- Au niveau calcul, cela veut dire que le programme utilisé stocke peu d'information et qu'il fonctionne rapidement.

Détour ou retour sur le produit matriciel

Les matrices sont des objets mathématiques assez versatiles qu'on utilise dans de nombreux domaines scientifiques. En informatique, ils sont très utiles en imagerie, central dans l'algorithme pagerank de google et importants quand on s'intéresse à certains graphes issus du monde réel (graphe sociaux, graphes de protéines).

Une matrice est tout simplement un tableau de nombres.

- On peut ajouter deux matrices semblables, en ajoutant les nombres qui sont au même positions.
- On peut multiplier chaque nombre de la matrice par un nombre fixé.

Ces deux opérations généralisent celles que vous connaissez sur des vecteurs, qui sont des matrices avec une dimension égale à 1.

Algèbre linéaire

L'étude des matrices correspond à une branche des mathématiques qu'on appelle l'algèbre linéaire.

La base de cette théorie se penche sur la résolution de systèmes d'équations linéaires à plusieurs inconnues, par exemple avec la méthode du pivot de Gauss.

Chaque opération sur les matrices à une interprétation dans cette théorie.

Le produit matriciel

Une matrice M_f représente dans notre contexte une fonction linéaire f (sur des variables $x_1, x_2, \ldots$) qu'on applique à un vecteur v.

Le produit de la matrice M_f par un vecteur v correspond à une substitution (remplacer x_1 par v_1, x_2 par v_2 etc) c'à.d. au calcul de $f(v)$.

Si la matrice de droite est composée de plusieurs vecteurs colonnes v_1, v_2, etc, le résultat du produit matriciel par M_f sera composé des colonnes $f(v_1), f(v2)$ etc

Une nouvelle opération $\star$

On note $\star$ l'opération de substitution des variables dans le système, que l'on voit comme une opération binaire prenant

- en entrée la matrice des coefficients et un vecteur de valeurs pour les variables,
- et donnant en sortie le résultat de la substitution.

Exemple

$$\begin{pmatrix} 0 & 0 & -1 & 4 \\ 4 & -8 & -1 & 0 \\ -1 & 2 & 0 & -1 \end{pmatrix} \star \begin{pmatrix} -\frac{1}{4} + 2a \\ a \\ -6 \\ -\frac{3}{4} \end{pmatrix}$$

$$= \begin{pmatrix} & & -1(-6) & +4(-\frac{3}{4}) \\ 4(-\frac{1}{4} + 2a) & -8a & -1(-6) & \\ -1(-\frac{1}{4} + 2a) & +2a & & -1(-\frac{3}{4}) \end{pmatrix}$$

matrice $\star$ vecteur

Définition

Soient $A = \left(a_{ij}\right)_{\substack{1 \leqslant i \leqslant m \\ 1 \leqslant j \leqslant n}}$ une matrice de m lignes et n colonnes et $v = (v_1, \ldots, v_n)$ un vecteur de $\mathbb{R}^n$. Le résultat de l'opération $A \star v$ est le vecteur $w = (w_1, \ldots, w_m)$ de $\mathbb{R}^m$ tel que $w_k = \sum_{j=1}^{n} a_{kj}.v_j$ pour tout $1 \leqslant k \leqslant m$

Exemple

Le produit de la matrice de coefficients à 3 lignes et 4 colonnes par le vecteur de $\mathbb{R}^4$

$$\begin{pmatrix} 0 & 0 & -1 & 4 \\ 4 & -8 & -1 & 0 \\ -1 & 2 & 0 & -1 \end{pmatrix} \star \begin{pmatrix} -\frac{1}{4} + 2a \\ a \\ -6 \\ -\frac{3}{4} \end{pmatrix}$$

a pour résultat le vecteur de $\mathbb{R}^3$

$$\begin{pmatrix} -1(-6) + 4(-\frac{3}{4}) \\ 4(-\frac{1}{4} + 2a) - 8a + (-1)(-6) \\ -1(-\frac{1}{4} + 2a) + 2a + (-1)(-\frac{3}{4}) \end{pmatrix}$$

Intérêt : vérification résolution système

La vérification (qu'un choix de valeur satisfait bien un système d'équations linéaires) correspond à effectuer une opération $\star$ et un test d'égalité entre deux vecteurs.

Slogan

Matrice coefficients $\star$ vecteur valeurs = vecteur des constantes

Remarque

C'est normal, l'opération $\star$ est une opération de substitution. On l'a introduite pour ça.

Intérêt : vérification résolution système

La vérification (qu'un choix de valeur satisfait bien un système d'équations linéaires) correspond à effectuer une opération $\star$ et un test d'égalité entre deux vecteurs.

Slogan

Matrice coefficients $\star$ vecteur valeurs = vecteur des constantes

Remarque

C'est normal, l'opération $\star$ est une opération de substitution. On l'a introduite pour ça.

Intérêt : vérification résolution système

La vérification (qu'un choix de valeur satisfait bien un système d'équations linéaires) correspond à effectuer une opération $\star$ et un test d'égalité entre deux vecteurs.

Slogan

Matrice coefficients $\star$ vecteur valeurs = vecteur des constantes

Remarque

C'est normal, l'opération $\star$ est une opération de substitution. On l'a introduite pour ça.

Autres exemples

Exemple

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 2 \\ 3 & 0 & 3 \\ 10 & 20 & 30 \end{pmatrix} \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = ?$$

Exemple

$$\begin{pmatrix} 1 & 0 \\ 2 & 2 \\ 3 & 4 \end{pmatrix} \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = ?$$

Autres exemples

Exemple

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 2 \\ 3 & 0 & 3 \\ 10 & 20 & 30 \end{pmatrix} \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} a & +b \\ & \\ & \\ & \end{pmatrix}$$

Exemple

$$\begin{pmatrix} 1 & 0 \\ 2 & 2 \\ 3 & 4 \end{pmatrix} \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = ?$$

Autres exemples

Exemple

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 2 \\ 3 & 0 & 3 \\ 10 & 20 & 30 \end{pmatrix} \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} a & +b & \\ & 2b & +2c \\ & & \end{pmatrix}$$

Exemple

$$\begin{pmatrix} 1 & 0 \\ 2 & 2 \\ 3 & 4 \end{pmatrix} \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = ?$$

Autres exemples

Exemple

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 2 \\ 3 & 0 & 3 \\ 10 & 20 & 30 \end{pmatrix} \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} a & +b & \\ & 2b & +2c \\ 3a & & +3c \\ & & \end{pmatrix}$$

Exemple

$$\begin{pmatrix} 1 & 0 \\ 2 & 2 \\ 3 & 4 \end{pmatrix} \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = ?$$

Autres exemples

Exemple

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 2 \\ 3 & 0 & 3 \\ 10 & 20 & 30 \end{pmatrix} \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} a & +b & \\ & 2b & +2c \\ 3a & & +3c \\ 10a & +20b & +30c \end{pmatrix}$$

Exemple

$$\begin{pmatrix} 1 & 0 \\ 2 & 2 \\ 3 & 4 \end{pmatrix} \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = ?$$

Autres exemples

Exemple

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 2 \\ 3 & 0 & 3 \\ 10 & 20 & 30 \end{pmatrix} \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} a+b \\ 2b+2c \\ 3a+3c \\ 10a+20b+30c \end{pmatrix}$$

Exemple

$$\begin{pmatrix} 1 & 0 \\ 2 & 2 \\ 3 & 4 \end{pmatrix} \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = ?$$

Autres exemples

Exemple

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 2 \\ 3 & 0 & 3 \\ 10 & 20 & 30 \end{pmatrix} \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} a+b \\ 2b+2c \\ 3a+3c \\ 10a+20b+30c \end{pmatrix}$$

Contre-exemple

$$\begin{pmatrix} 1 & 0 \\ 2 & 2 \\ 3 & 4 \end{pmatrix} \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \text{Pas défini !}$$

Autres exemples

Exemple

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 2 \\ 3 & 0 & 3 \\ 10 & 20 & 30 \end{pmatrix} \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} a+b \\ 2b+2c \\ 3a+3c \\ 10a+20b+30c \end{pmatrix}$$

Contre-exemple

$$\begin{pmatrix} 1 & 0 \\ 2 & 2 \\ 3 & 4 \end{pmatrix} \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \text{Pas défini !}$$

car matrice à 2 colonnes fois vecteur de $\mathbb{R}^3$
intuition : on cherche à substituer 3 valeurs alors qu'on a 2 variables seulement !

Exemple d'opération linéaire

Exemple

L'opération f_M de $\mathbb{R}^3$ dans $\mathbb{R}^3$ qui mélange ses arguments en échangeant le premier et le dernier, c'est-à-dire telle que

$$f_M(a, b, c) = (c, b, a)$$

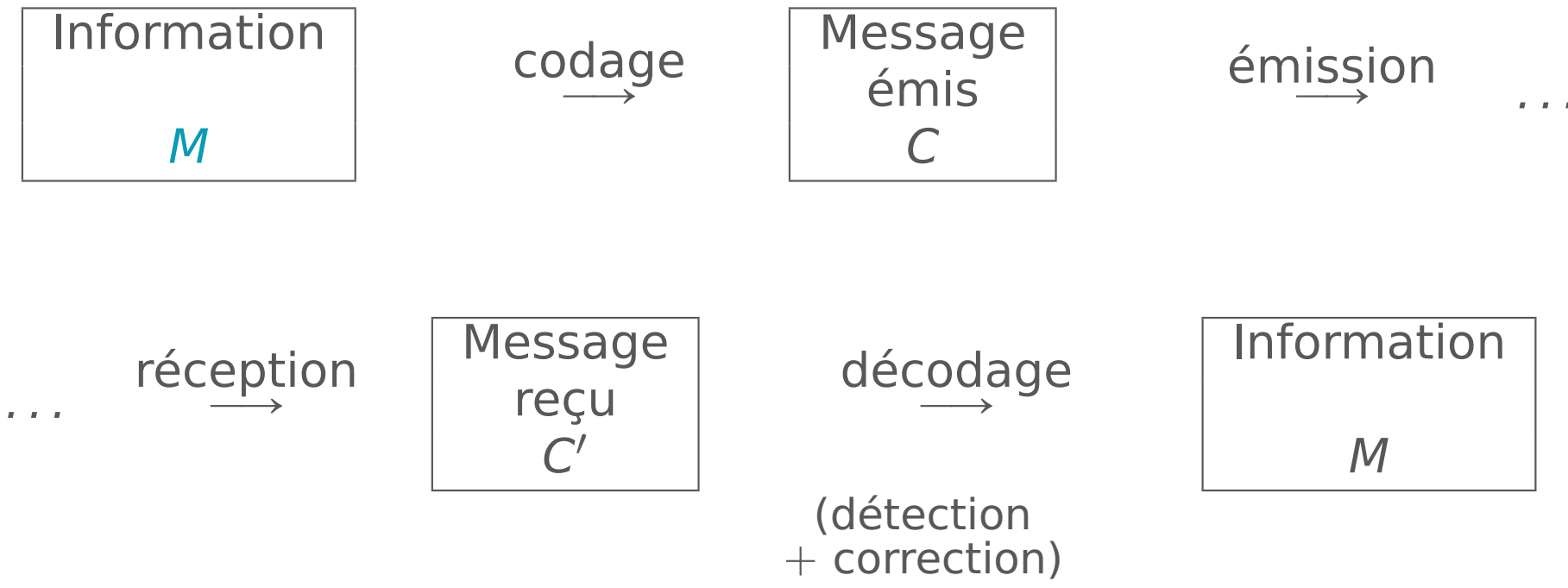
a pour matrice

$$M = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

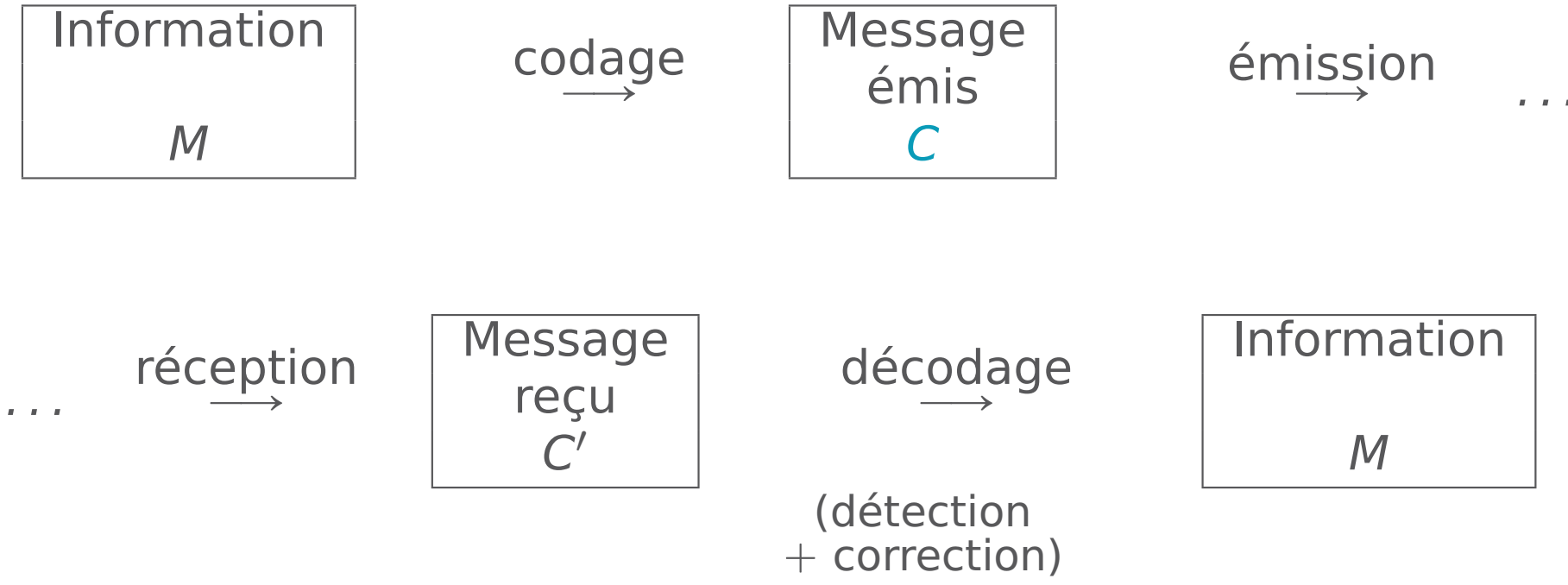
on a bien

$$M \star \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} c \\ b \\ a \end{pmatrix}$$

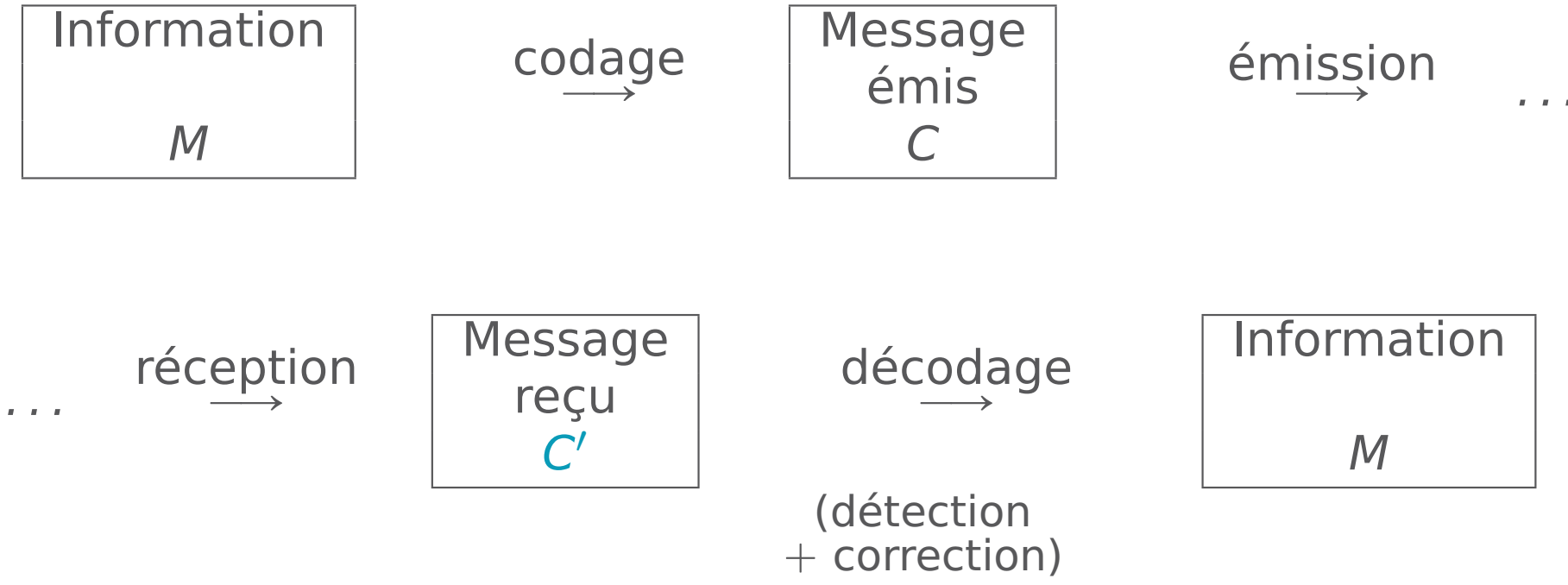
Résumé


Information
M
codage
Message
émis
C
émission
...
...
réception
Message
reçu
C'
décodage
(détection
+ correction)
Information
M

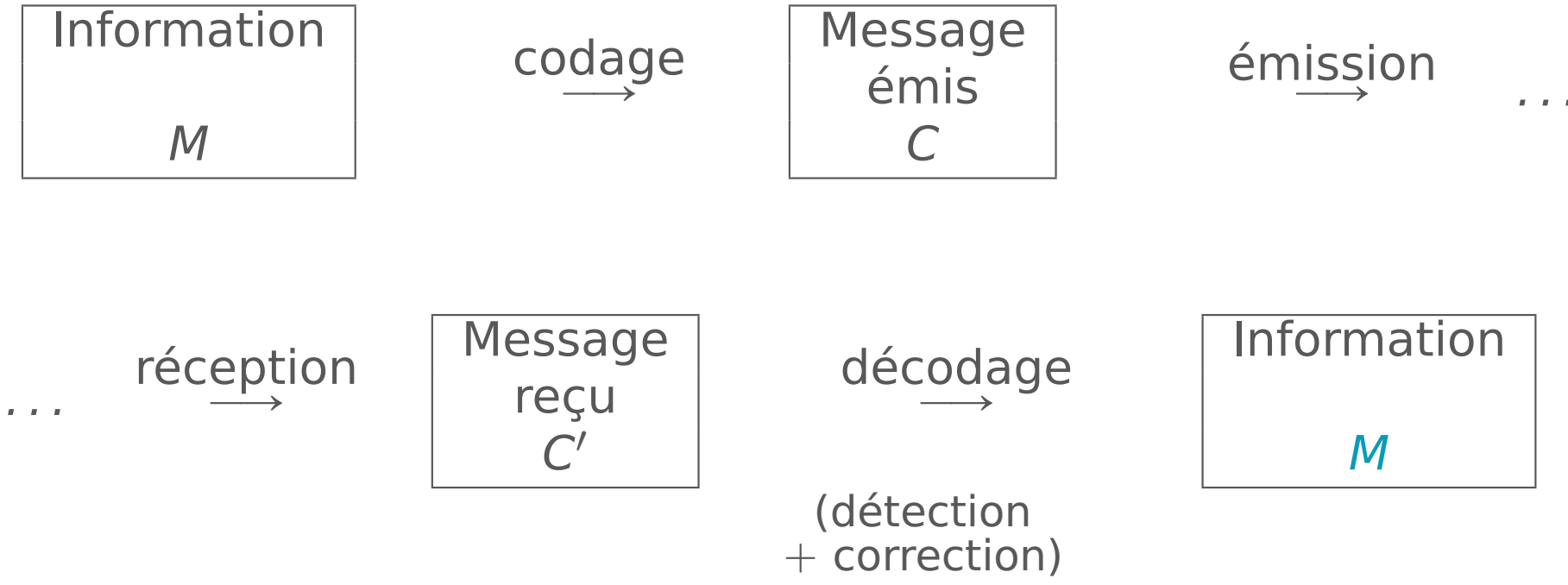
Résumé


Information
M
codage
Message émis
C
émission
...
...
réception
Message reçu
C'
décodage
(détection + correction)
Information
M

Résumé


Information
M
codage
Message
émis
C
émission
. . .
. . .
réception
Message
reçu
C′
décodage
(détection
+ correction)
Information
M

Résumé


Information
M
codage
Message
émis
C
émission
. . .
. . .
réception
Message
reçu
C′
décodage
(détection
+ correction)
Information
M

2 matrices importantes

- La matrice de codage G.

  On fait $G \star M$ pour obtenir le mot de code C correspondant à M
- La matrice de contrôle H.

  On fait $H \star C'$ pour détecter si C' est un mot de code

La seule différence avec le calcul matriciel classique est qu'on va utiliser le calcul modulo 2 sur les entiers. Ça ne change pas les propriétés du calcul matriciel.

Exemple pour le code de parité

Pour le code de parité avec 8 bits d'information et 1 bit de contrôle, la matrice G est

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \star \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Vocabulaire

Aujourd'hui nous ne verrons que des codes qui recopient l'information avant d'y ajouter des bits de contrôle : on parle dans ce cas de code systématique. Lorsqu'au contraire, on mélange information et contrôle, on parle de code entrelacé.

Vecteurs élémentaires

- Il s'agit des vecteurs ayant un seul 1.
- Pour $\mathbb{R}^n$, il y a en a donc exactement n.
- Ce sont les colonnes de la matrice identité I_n
- On note ces vecteurs $e_1, e_2, \ldots, e_n$.
- Le vecteur e_i a un 1 en $i$ème position.

Base canonique

- Tout vecteur de $\mathbb{R}^n$ s'écrit comme combinaison linéaire de ces vecteurs.
- C'est ce que nous faisons sans le savoir en écrivant un vecteur de $\mathbb{R}^n$ sous forme de tuple.

Comment déterminer la matrice de codage d'un code linéaire ?

La méthode

- Pour une fonction de codage f d'un code linéaire
  - Il suffit de connaître les mots de code des vecteurs élémentaires : $f(e_1), f(e_2) \ldots, f(e_m)$
  - qui forment les colonnes de la matrice G.

Pourquoi ?

C'est une propriété de toutes les applications linéaires, c'est-à-dire de toutes les fonctions qu'on peut représenter par une matrice. Nous verrons plus de détails à ce sujet ultérieurement.

Matrice génératrice du code ou matrice de codage

Si

vecteurs élémentaires		mot de code
1000	$\longrightarrow$	1000101
0100	$\longrightarrow$	0100111
0010	$\longrightarrow$	0010110
0001	$\longrightarrow$	0001011

Alors

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

Matrice génératrice du code ou matrice de codage

vecteurs élémentaires		mot de code
1000	$\longrightarrow$	1000101

$$G = \begin{pmatrix} 1 & \\ 0 & \\ 0 & \\ 0 & \\ 1 & \\ 0 & \\ 1 & \end{pmatrix}$$

Matrice génératrice du code ou matrice de codage

vecteurs élémentaires		mot de code
1000	$\longrightarrow$	1000101
0100	$\longrightarrow$	0100111

$$G = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

Matrice génératrice du code ou matrice de codage

vecteurs élémentaires		mot de code
1000	$\longrightarrow$	1000101
0100	$\longrightarrow$	0100111
0010	$\longrightarrow$	0010100
0001	$\longrightarrow$	0001011

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

Utilisation de la matrice génératrice

$GM = C$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \star \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

matrice génératrice mot d'info mot de code

$$G \star M = C$$

Détails du calcul des mots de code

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \star \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

- On calcule le mot de code C en faisant la somme $\oplus$ des colonnes de la matrice génératrice G qui correspondent aux lignes des 1 du mot d'information M.

  (C'est parce qu'on fait les calculs modulo 2)

Structure de la matrice génératrice G

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

- Code de type $\mathcal{C}_{7,4}$
  - Longueur des mots de code $n = 7$ bits
  - Longueur des mots d'info $m = 4$ bits
- Matrice génératrice G
  - Nombre de lignes $n = 7$
  - Nombre de colonnes $m = 4$

$$I_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Matrice identité I_m
  - Nombre de lignes $m = 4$
  - Nombre de colonnes $m = 4$

$$Q = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

- Matrice de surcodage Q
  - Nombre de lignes $r = n - m = 3$
  - Nombre de colonnes $m = 4$

Rôle des matrices I_m et Q

- La matrice identité I_m sert à recopier les bits d'information

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

- La matrice de surcodage Q sert à calculer les bits de contrôle

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Rappel de l'exemple

mot de la base		mot de code
1000	$\longrightarrow$	1000101
0100	$\longrightarrow$	0100111
0010	$\longrightarrow$	0010110
0001	$\longrightarrow$	0001011

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

Le mot d'information 1101 est codé par le mot de code 1101001

Matrice de contrôle H

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Matrice de contrôle H

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$$Q = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \qquad I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Matrice de contrôle H

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$$Q = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \qquad I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

$$I_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Q = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

Utilisation de la matrice de contrôle

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \\ \\ \end{pmatrix}$$

matrice de contrôle — message reçu — *syndrome*

H — C — $= S$

Utilisation de la matrice de contrôle

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

matrice de contrôle — message reçu — *syndrome*

$$H \qquad C = S$$

Syndrome d'un message $HC = S$

- Le message reçu est un mot de code : syndrome nul

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

- Le message reçu n'est pas un mot de code : syndrome non nul

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

Structure de la matrice de contrôle H

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

- Code de type $\mathcal{C}_{7,4}$
  - Longueur des mots de code $n = 7$ bits
  - Longueur des mots d'info $m = 4$ bits
- Matrice de contrôle H
  - Nombre de lignes $r = n - m = 3$
  - Nombre de colonnes $n = 7$

$$Q = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

- Matrice de surcodage Q
  - Nombre de lignes $r = 3$
  - Nombre de colonnes $m = 4$

$$I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Matrice identité I_r
  - Nombre de lignes $r = 3$
  - Nombre de colonnes $r = 3$

Décomposition du produit $HC = S$

$$
\begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}
\begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}
$$

$$
= \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}
\begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}
\oplus
\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}
\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}
$$

$$
= \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}
$$

$$
= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}
$$

Rôle des matrices Q et I_r

- La matrice de surcodage Q agit sur les bits d'info reçus et sert à recalculer les bits de contrôle

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

- La matrice identité I_r agit sur les bits de contrôle reçus et sert à les recopier

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

- La somme $\oplus$ permet de tester si les deux colonnes obtenues sont identiques

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Rôle des matrices Q et I_r

- La matrice de surcodage Q agit sur les bits d'info reçus et sert à recalculer les bits de contrôle

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

- La matrice identité I_r agit sur les bits de contrôle reçus et sert à les recopier

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

- La somme $\oplus$ permet de tester si les deux colonnes obtenues sont identiques

$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

Détection des erreurs

Propriété

C est un mot de code ssi son syndrome $S = HC = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$

Message reçu	Syndrome	Contrôle
C	$HC = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$	OK
C'	$HC' \neq \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$	Anomalie