

Les codes correcteurs, détecteurs de mensonge... mathématiques



Mathilde Herblot, Frédérique Petit

Université Paris Diderot, Sorbonne université

IREM de Paris - 12 décembre 2018



Le groupe des maths fantastiques propose :

- fête de la science et assimilé (ateliers dans les classes, rallye, salon CJM)
- stage PAF
- des fiches pédagogiques
- des supports de présentation

Ces activités se retrouvent sur

<https://www.math.univ-paris-diderot.fr/diffusion/>

Le jeu du « Qui est-ce ? »

Le rallye est décliné sur trois niveaux : jaune, orange, bleu.

Le jeu du « Qui est-ce ? »

Un exemple d'activité présentée lors du rallye de la Fête de la science 2015 : le jeu du « Qui est-ce ? ».

Le jeu du « Qui est-ce ? »

Un exemple d'activité présentée lors du rallye de la Fête de la science 2015 : le jeu du « Qui est-ce ? ».

Les personnages se distinguent par les caractéristiques suivantes :

- des lunettes  ou pas 
- des moustaches  ou pas 
- un chapeau  ou pas 
- des cheveux  ou pas .

Combien de personnages y a-t-il ?

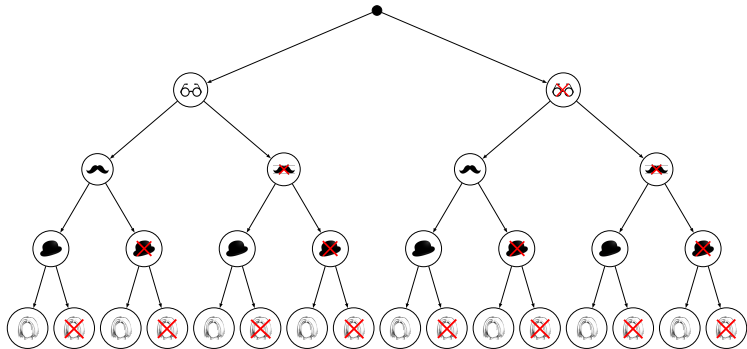
Le rallye

Combien de personnages y a-t-il ?

Réponse. $2^4 = 16$.

Les puissances ne sont vues qu'en Quatrième.

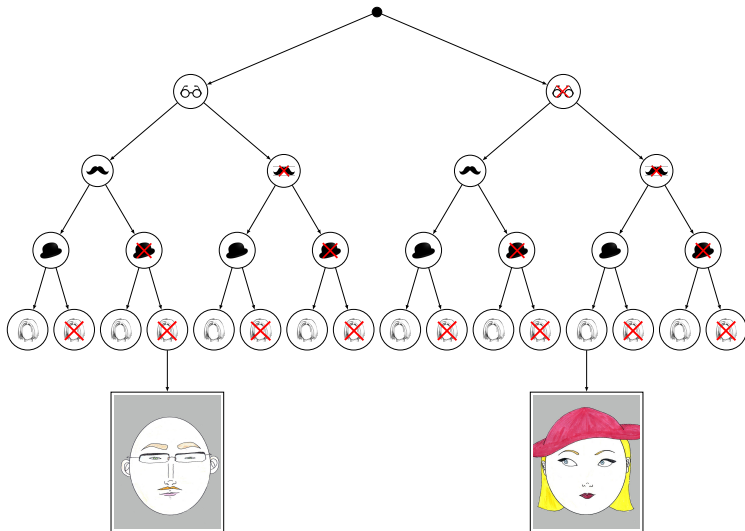
Au niveau jaune, on dira $2 \times 2 \times 2 \times 2$.



Combien faut-il poser de questions pour être sûr de pouvoir trouver ?

Le rallye

On pose autant de questions qu'il y a de critères, soit 4 questions.



Le rallye

Voici tous les personnages du jeu :



Maintenant,
on rajoute des éléments sur certains de ces personnages :

- des boucles d'oreille



- une barbe



- un nœud papillon



Maintenant,
on rajoute des éléments sur certains de ces personnages :

- des boucles d'oreille



- une barbe



- un nœud papillon



Combien de questions faut-il poser pour être sûr de pouvoir
trouver le personnage ?

Combien de questions faut-il poser pour être sûr de pouvoir trouver le personnage ?

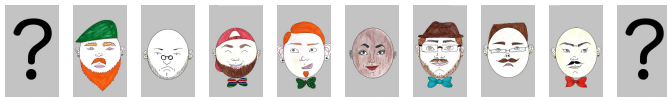
Réponse. *Cela ne change rien... toujours les 4 mêmes questions !*

On joue à ce jeu avec les éléments supplémentaires et en rajoutant la règle suivante : le joueur qui a choisi le personnage secret a le droit de mentir au plus une fois en répondant aux questions.

Ces questions sont, dans l'ordre

- (1) A-t-il des lunettes ?
- (2) A-t-il une moustache ?
- (3) A-t-il un chapeau ?
- (4) A-t-il des cheveux ?
- (5) A-t-il des boucles d'oreille ?
- (6) A-t-il une barbe ?
- (7) A-t-il un nœud papillon ?

Lançons une partie sur l'ordinateur !

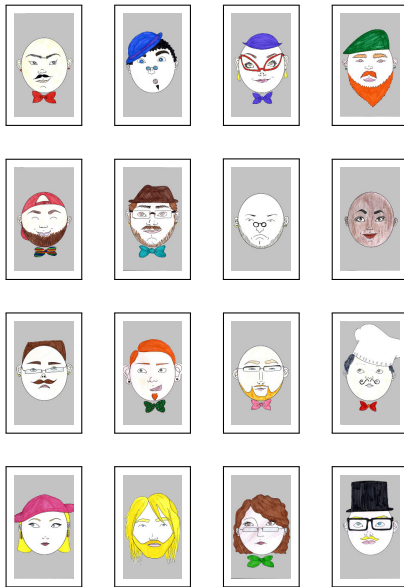


Lancer le jeu



Quel est le nombre minimal de différences entre deux personnages distincts du jeu ?

Le rallye



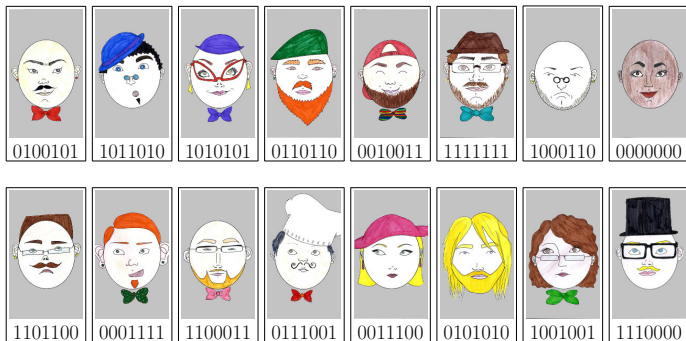
Réponse. *Il y en a 3. On ne va pas demander de vérifier tous les couples de personnages mais l'admettre. Ce qui est important c'est plutôt de comprendre la question suivante.*

Pourquoi cela permet-il de deviner le personnage secret, même si le joueur a menti une fois ?

Le niveau jaune s'arrête là, mais si l'enfant a envie de continuer, on peut lui proposer la méthode de codage d'un personnage avec des 0 et des 1 présentée dans le niveau orange.

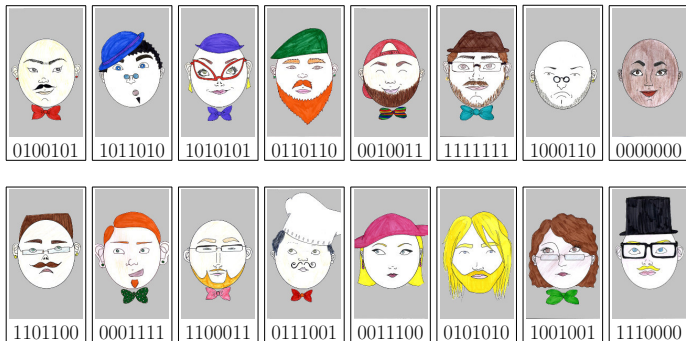
Le rallye

On représente chaque personnage par une suite de sept chiffres, 0 ou 1. Le deuxième personnage, celui qui a des lunettes, pas de moustache, un chapeau, des cheveux, pas de boucle d'oreille, une barbe, pas de nœud papillon, sera ainsi représenté par 1011010.



Le rallye

On code les réponses du joueur aux questions de la même façon, par exemple si j'ai choisi le personnage numéro 2 et que je mens à la question « a-t-il des boucles d'oreille ? », ma réponse sera 1011110.



Voyons maintenant comment jouer à ce jeu avec mensonge, sans l'aide de l'ordinateur !

On reçoit une réponse. Pour savoir si elle est correcte ou mensongère, on va calculer trois nombres, les « syndromes » qui valent chacun 0 ou 1.

Quelques explications

- s_1 : on garde un chiffre sur deux $\square \times \square \times \square \times \square$ on en fait la somme. s_1 vaut 0 si la somme est paire, 1 si elle est impaire.
- s_2 : $\times \square \square \times \square \square \times$, on fait la somme, s_2 vaut 0 si elle est paire, 1 si elle est impaire.
- s_3 : $\times \times \times \square \square \square \square$ (que les quatre derniers chiffres), on fait la somme, s_3 vaut 0 si elle est paire, 1 si elle est impaire.

Quelques explications

- s_1 : on garde un chiffre sur deux $\square \times \square \times \square \times \square$ on en fait la somme. s_1 vaut 0 si la somme est paire, 1 si elle est impaire.
- s_2 : $\times \square \square \times \square \square \times$, on fait la somme, s_2 vaut 0 si elle est paire, 1 si elle est impaire.
- s_3 : $\times \times \times \square \square \square \square$ (que les quatre derniers chiffres), on fait la somme, s_3 vaut 0 si elle est paire, 1 si elle est impaire.

Exemple. Si la réponse obtenue est 0011110, on a

$$s_1 = 0, s_2 = 1, s_3 = 1.$$

Le joueur répond sans mensonge.

Pour chaque personnage, on calcule, s_1 , s_2 et s_3 .

Le joueur répond sans mensonge.

Pour chaque personnage, on calcule, s_1, s_2 et s_3 .

*On répartit la tâche dans le groupe d'élèves,
chacun en calcule 1 ou 2 !*

Dans tous les cas, on trouve $s_1 = s_2 = s_3 = 0$.

Quelques explications

Le joueur répond sans mensonge.

Pour chaque personnage, on calcule, s_1, s_2 et s_3 .

*On répartit la tâche dans le groupe d'élèves,
chacun en calcule 1 ou 2 !*

Dans tous les cas, on trouve $s_1 = s_2 = s_3 = 0$.

Que valent s_1, s_2 et s_3 si on ment à la première question et pas aux autres ?

Que valent s_1 , s_2 et s_3 si on ment à la première question et pas aux autres ?

Il est important de réussir à faire comprendre que la réponse à cette question ne dépend pas du personnage choisi.

Pour le calcul de s_1 , on regarde $\square \times \square \times \square \times \square$. s_1 vaut donc 1.

Quelques explications

Pour le calcul de s_1 , on regarde $\square \times \square \times \square \times \square$. s_1 vaut donc 1.

Pour le calcul de s_2 , $\times \square \square \times \square \square \times$. s_2 qui vaut toujours 0.

Quelques explications

Pour le calcul de s_1 , on regarde $\square \times \square \times \square \times \square$. s_1 vaut donc 1.

Pour le calcul de s_2 , $\times \square \square \times \square \square \times$. s_2 qui vaut toujours 0.

Pour s_3 , comme pour s_2 , une modification du premier bit ne change rien, on a toujours $s_3 = 0$.

Et si on ment seulement à la cinquième question ?

s_1 : ☐X☐X☐X☐

s_2 : X☐☐X☐☐X

s_3 : XXX☐☐☐☐

Quelques explications

Et si on ment seulement à la cinquième question ?

s_1 : ☐ ☒ ☐ ☒ ☐ ☒ ☐

s_2 : ☒ ☐ ☐ ☒ ☐ ☐ ☒

s_3 : ☒ ☒ ☒ ☐ ☐ ☐ ☐

Dans ce cas, (s_1, s_2, s_3) est égal à la cinquième colonne du tableau

	1	2	3	4	5	6	7
s_1	1	0	1	0	1	0	1
s_2	0	1	1	0	1	1	0
s_3	0	0	0	1	1	1	1

On peut donner une explication du même type qu'à la question précédente, pour trouver $s_1 = 1$, $s_2 = 1$ et $s_3 = 1$.

Comment découvrir facilement à quelle question le joueur a menti ?

Comment découvrir facilement à quelle question le joueur a menti ?
On calcule les syndromes. S'ils ne valent pas tous 0, on regarde le numéro de la colonne correspondant, il donne le numéro de la question à laquelle on a menti. Si les syndromes sont tous les trois nuls, le joueur n'a pas menti.
On a construit une sorte de détecteur de mensonge !

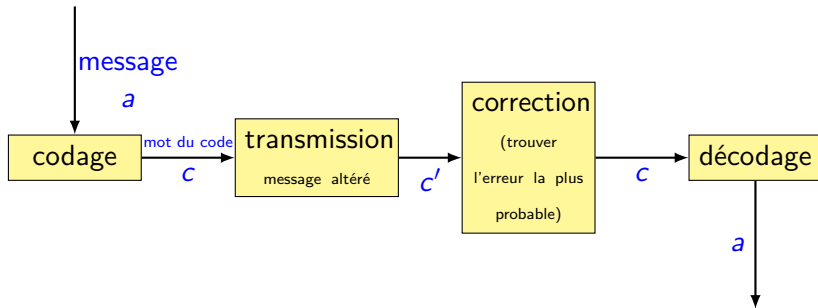
Quelques explications

Comment découvrir facilement à quelle question le joueur a menti ?
On calcule les syndromes. S'ils ne valent pas tous 0, on regarde le numéro de la colonne correspondant, il donne le numéro de la question à laquelle on a menti. Si les syndromes sont tous les trois nuls, le joueur n'a pas menti.
On a construit une sorte de détecteur de mensonge !

	1	2	3	4	5	6	7
s_1	1	0	1	0	1	0	1
s_2	0	1	1	0	1	1	0
s_3	0	0	0	1	1	1	1

Le principe sur lequel repose ce jeu est en fait à la base d'applications très utiles, pour transmettre ou stocker des données.

Principe des codes correcteurs d'erreurs



Exemples d'utilisation



Exemples d'utilisation



Distance de Hamming

Alphabet $\mathcal{A}$, ensemble fini de cardinal q . Un code correcteur $\mathcal{C}$ de longueur n sur $\mathcal{A}$ est un sous-ensemble de $\mathcal{A}^n$ (de cardinal au moins 2).

Distance de Hamming sur $\mathcal{A}^n$:

pour tous $x = (x_1, \dots, x_n) \in \mathcal{A}^n$ et $y = (y_1, \dots, y_n) \in \mathcal{A}^n$,

$$d(x, y) = \text{Card}\{j \in \{1, \dots, n\} \mid x_j \neq y_j\}$$

Distance de Hamming

Alphabet $\mathcal{A}$, ensemble fini de cardinal q . Un code correcteur $\mathcal{C}$ de longueur n sur $\mathcal{A}$ est un sous-ensemble de $\mathcal{A}^n$ (de cardinal au moins 2).

Distance de Hamming sur $\mathcal{A}^n$:

pour tous $x = (x_1, \dots, x_n) \in \mathcal{A}^n$ et $y = (y_1, \dots, y_n) \in \mathcal{A}^n$,

$$d(x, y) = \text{Card}\{j \in \{1, \dots, n\} \mid x_j \neq y_j\}$$

- $d(x, y) = 0 \Leftrightarrow x = y$
- $d(x, y) = d(y, x)$
- $d(x, y) \leq d(x, z) + d(z, y)$

Distance de Hamming

Alphabet $\mathcal{A}$, ensemble fini de cardinal q . Un code correcteur $\mathcal{C}$ de longueur n sur $\mathcal{A}$ est un sous-ensemble de $\mathcal{A}^n$ (de cardinal au moins 2).

Distance de Hamming sur $\mathcal{A}^n$:

pour tous $x = (x_1, \dots, x_n) \in \mathcal{A}^n$ et $y = (y_1, \dots, y_n) \in \mathcal{A}^n$,

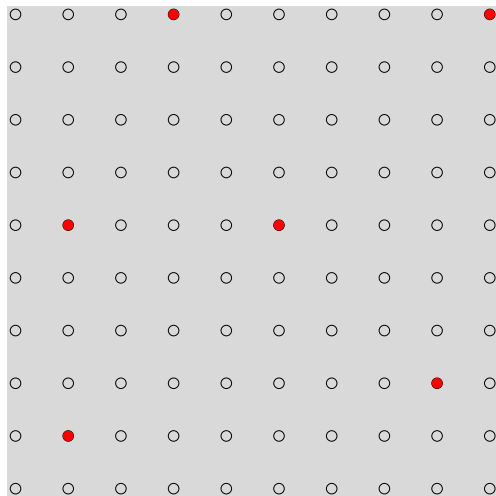
$$d(x, y) = \text{Card}\{j \in \{1, \dots, n\} \mid x_j \neq y_j\}$$

Distance minimale d'un code $\mathcal{C} \subset \mathcal{A}^n$:

$$d = d(\mathcal{C}) = \min_{x, y \in \mathcal{C}, x \neq y} d(x, y).$$

Capacité de détection et de correction

On peut détecter $d - 1$ erreurs.

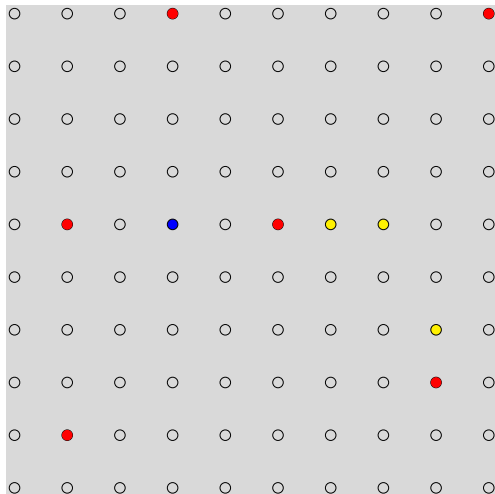


- mots du code
- mots de $\mathcal{A}^n$
hors du code

Capacité de détection et de correction

On peut détecter $d - 1$ erreurs.

On peut corriger $\left\lfloor \frac{d-1}{2} \right\rfloor$ erreurs.

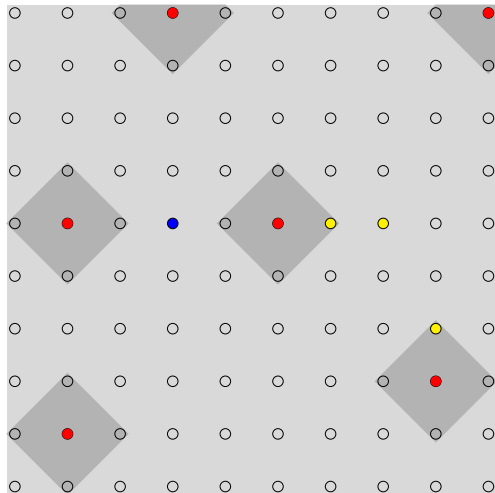


- mots du code
- mots de $\mathcal{A}^n$ hors du code
- quelques mots que l'on peut corriger
- un mot qu'on ne peut pas corriger

Capacité de détection et de correction

On peut détecter $d - 1$ erreurs.

On peut corriger $\left\lfloor \frac{d-1}{2} \right\rfloor$ erreurs.



- mots du code
- mots de $\mathcal{A}^n$ hors du code
- quelques mots que l'on peut corriger
- un mot qu'on ne peut pas corriger

Théorème (borne de Hamming)

Soit $\mathcal{C}$ un code de longueur n sur un alphabet de cardinal q , et soit d sa distance minimale. Posons $t = \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor$ la capacité de correction de $\mathcal{C}$. Alors :

$$\text{Card}(\mathcal{C}) \sum_{j=0}^t \binom{n}{j} (q-1)^j \leq q^n.$$

Théorème (borne de Hamming)

Soit $\mathcal{C}$ un code de longueur n sur un alphabet de cardinal q , et soit d sa distance minimale. Posons $t = \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor$ la capacité de correction de $\mathcal{C}$. Alors :

$$\text{Card}(\mathcal{C}) \sum_{j=0}^t \binom{n}{j} (q-1)^j \leq q^n.$$

Démonstration.

Nombres de mots à distance exactement j d'un mot donné :

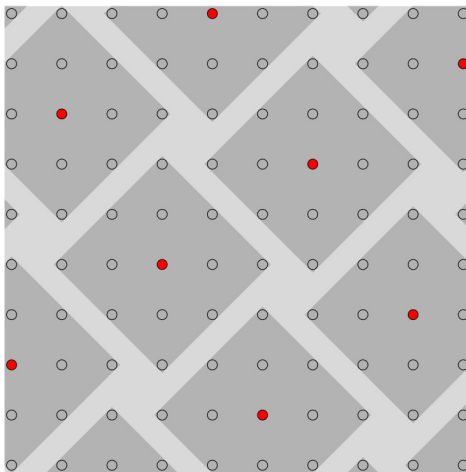
$$\binom{n}{j} (q-1)^j.$$

Il y a $\text{Card}(\mathcal{C})$ boules de rayon t centrées en des mots du code, deux à deux disjointes.



Codes parfaits

Un code est dit *parfait* si la borne de Hamming est atteinte.



Difficulté algorithmique : comment trouver le mot du code le plus proche ?

→ Plus de structures algébriques.

Alphabet : corps fini F (par exemple, $F = \mathbf{Z}/p\mathbf{Z}$)

Code $\mathcal{C}$ de longueur n sur F : sous-espace vectoriel $\mathcal{C}$ de F^n .

$k = \dim \mathcal{C}$ et $\text{Card} \mathcal{C} = (\text{Card} F)^k$

Alphabet : corps fini F (par exemple, $F = \mathbf{Z}/p\mathbf{Z}$)

Code $\mathcal{C}$ de longueur n sur F : sous-espace vectoriel $\mathcal{C}$ de F^n .

$k = \dim \mathcal{C}$ et $\text{Card} \mathcal{C} = (\text{Card} F)^k$

$$\forall x, y \in F^n, d(x, y) = d(x - y, 0)$$

Distance minimale d'un code linéaire $\mathcal{C}$:

$$d(\mathcal{C}) = \min_{x, y \in \mathcal{C}, x \neq y} d(x, y) = \min_{z \in \mathcal{C}, z \neq 0} d(z, 0) = \min_{z \in \mathcal{C}, z \neq 0} \omega(z).$$

Codes linéaires

Alphabet : corps fini F (par exemple, $F = \mathbf{Z}/p\mathbf{Z}$)

Code $\mathcal{C}$ de longueur n sur F : sous-espace vectoriel $\mathcal{C}$ de F^n .

$k = \dim \mathcal{C}$ et $\text{Card} \mathcal{C} = (\text{Card} F)^k$

$$d(\mathcal{C}) = \min_{z \in \mathcal{C}, z \neq 0} \omega(z).$$



Exemple du « qui est-ce ? »



$F = \mathbf{Z}/2\mathbf{Z}$, $n = 7$, $\text{Card}(\mathcal{C}) = 16$, $k = 4$,

$\mathcal{C} = \{ \begin{array}{l} 0100101, 1011010, 1010101, 0110110, 0010011, 1111111, \\ 1000110, 0000000, 1101100, 0001111, 1100011, 0111001, \\ 0011100, 0101010, 1001001, 1110000 \end{array} \}$

$d = 3$, $t = 1$

Borne de Hamming : $\frac{q^n}{\sum_{j=0}^t \binom{n}{j} (q-1)^j} = \frac{2^7}{1+7} = 2^4$, ce code est parfait !

Juhani Virtakallio a publié en 1947 un code correcteur sur $\mathbf{Z}/3\mathbf{Z}$ de $3^6 = 729$ mots de longueur 11, de distance minimale 5 et qui est donc parfait, dans la revue Veikkaaja...magazine de football finlandais !

Matrice génératrice G :
une matrice de $\mathcal{M}_{k,n}(F)$ dont les lignes sont une base de $\mathcal{C}$.



Exemple du « qui est-ce ? »



$$\mathcal{C} = \{ \begin{array}{l} 0100101, 1011010, 1010101, 0110110, 0010011, 1111111, \\ 1000110, 0000000, 1101100, 0001111, 1100011, 0111001, \\ 0011100, 0101010, 1001001, 1110000 \end{array} \}$$

Matrices génératrices

Matrice génératrice G :

une matrice de $\mathcal{M}_{k,n}(F)$ dont les lignes sont une base de $\mathcal{C}$.



Exemple du « qui est-ce ? »



$\mathcal{C} = \{$ 0100101, 1011010, 1010101, 0110110, 0010011, 1111111,
1000110, 0000000, 1101100, 0001111, 1100011, 0111001,
0011100, 0101010, 1001001, 1110000 $\}$

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

Matrice vérificatrice H :

une matrice de $\mathcal{M}_{n-k,n}(F)$ dont le noyau est $\mathcal{C}$ (de rang max telle que $HG = 0$).



Exemple du « qui est-ce ? »



rallye

s_1 : ☐X☐X☐X☐

s_2 : X☐X☐X☐X

s_3 : XXX☐X☐X☐

Matrices vérificatrices

Matrice vérificatrice H :

une matrice de $\mathcal{M}_{n-k,n}(F)$ dont le noyau est $\mathcal{C}$ (de rang max telle que $HG = 0$).



Exemple du « qui est-ce ? »



rallye

algèbre linéaire

s_1 : $\square \times \square \times \square \times \square$

s_2 : $\times \square \square \times \square \square \times$

s_3 : $\times \times \times \square \square \square \square$

$$\begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}}_{=H} \begin{pmatrix} x_1 \\ \vdots \\ x_7 \end{pmatrix}$$

Codes linéaires, correction

$x = (x_1, \dots, x_n) \in F^n$, syndrome de x :

$$s(x) = H \cdot {}^t x$$

$$s(x) = 0 \Leftrightarrow x \in \mathcal{C}$$

Pour tout $c \in \mathcal{C}$, pour tout $e \in F$,

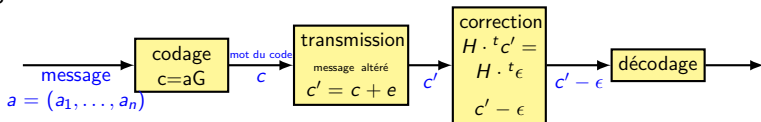
$$s(c + e) = H \cdot {}^t(c + e) = H \cdot {}^t e = s(e).$$

Principe de la correction

- On précalcule les syndromes de tous les mots de poids inférieur à $t = \lfloor \frac{d-1}{2} \rfloor$.
- On reçoit un mot x , on calcule $s(x)$.
- Si $s(x) = 0$, x est un mot du code.
- Sinon, on cherche dans la table des syndromes un e tel que $s(e) = s(x)$.
- On corrige x en $c = x - e$.

Codage, correction, décodage

Code linéaire de longueur n et dimension k , de matrice génératrice G et matrice vérificatrice H



Exemple du « qui est-ce ? »



message initial : $a = (a_1, a_2, a_3, a_4)$

codage : $c = aG = (a_1, a_2, a_3, a_4, c_5, c_6, c_7)$,

où $c_5 = a_1 + a_2 + a_4$, $c_6 = a_1 + a_3 + a_4$, $c_7 = a_2 + a_3 + a_4$

message reçu : $c' = c$ ou $c' = c + e_i$

$H \cdot {}^t c' = 0$ ou $H \cdot {}^t c' = t$ i-ème colonne de H .

On corrige si nécessaire $c = c' + e_i$

Décodage : (c_1, c_2, c_3, c_4) .

« Qui est-ce ? », making of

Matrice H dont les colonnes $H_1, \dots, H_7$ sont les 7 vecteurs non nuls de $(\mathbf{Z}/2\mathbf{Z})^3$. La distance minimale du code est 3 :

- au plus 3 par la borne de Hamming.
- Si $x \in \ker(H)$, $\sum_{i=1}^7 x_i H_i = 0$ et H n'a pas deux colonnes liées.

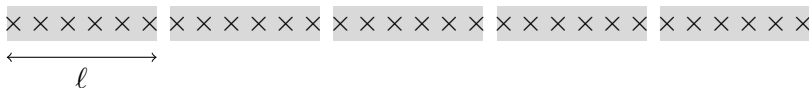
Les personnages du jeu sont ceux du noyau.

Correction de salves d'erreurs

Pour certaines utilisations, les erreurs ont tendance à être groupées. On envoie un message formé de bits 0, 1. On les groupe par blocs de ℓ .

$(\mathbf{Z}/2\mathbf{Z})^\ell$ est isomorphe comme $\mathbf{Z}/2\mathbf{Z}$ -espace vectoriel à $\mathbf{F}_{2^\ell}$.

On utilise un code correcteur sur $\mathbf{F}_{2^\ell}$ capable de corriger t erreurs.



Correction de salves d'erreurs

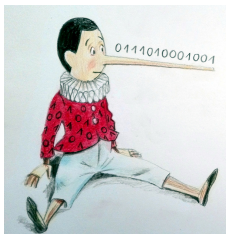
Pour certaines utilisations, les erreurs ont tendance à être groupées. On envoie un message formé de bits 0, 1. On les groupe par blocs de ℓ .

$(\mathbf{Z}/2\mathbf{Z})^\ell$ est isomorphe comme $\mathbf{Z}/2\mathbf{Z}$ -espace vectoriel à $\mathbf{F}_{2^\ell}$.

On utilise un code correcteur sur $\mathbf{F}_{2^\ell}$ capable de corriger t erreurs.



On peut corriger toute salve de $(t - 1)\ell + 1$ erreurs consécutives.



Mcrei puor vtore anteiottn !

Fiches de questions et de réponses disponibles sur :
<https://www.math.univ-paris-diderot.fr/diffusion/rallye>