INTRODUCTION

Alice et Bob veulent communiquer, mais leurs appareils (téléphone, cellulaire, radio, ordinateur...) sont imparfaites. Est-ce qúils peuvent communiquer sans compromettre leur conversation? Par exemple, ils veulent se parler au téléphone, mais la ligne est imparfaite, donnant parfois du "bruit". Ou bien Alice veut transmettre à Bob ses photos de vacances sur son cellulaire, sachant que quelques bits seront renverser par l'interférence. C'est un modèle très général: si "Alice" est un disque compacte et "Bob" est un haut-parleur, alors on cherche à jouer correctement la musique même si le disque est endommagé.

Toutes ces situations se comprennent comme un désir de transmettre un message, sachant qu'il serait modifiée de façon inconnue, mais le transmettre d'une manière à ce que le récepteur peut encore correctement comprendre.

Ce n'est pas un problème nouveau: on pourrait dire que c'est le problème fondamental de communication entre humains. Les langues modernes représentent une solution: vuos pouvvait encore comprenrde le sens malgrè des ereurs, car la langue française inclut suffisamment de redondance à pouvoir détecter et même corriger ces erreurs.

La solution c'est donc simple, en principe: la redondance. Le problème c'est qu'on veut aussi que le message soit aussi petit que possible (e.g.,la transmission d'images digitales).

MOTS, DISTANCE, BOULES

On écrit les messages avec un ALPHABET de q symboles, typiquement $\{0, 1, 2, \cdots, q-1\}$. Par exemple, si $q = 2$, alors chaque symbole est un "bit". On forme des MOTS avec n symboles. Par exemple, si $q = 2$ un mot de $n = 8$ bits serait un "byte".

Les mots possibles sont tous les mots de longueur n avec q symboles; il y a donc au total q^n mots possibles. Un CODE C est un sous-ensemble des mots possibles.

La *distance* entre deux mots est le nombre de positions dans lesquelles ils diffèrent: on écrit $d(x, y)$ pour la distance entre les deux mots x et y. La DISTANCE MINIMALE du code est le minimum de la distance entre tout pair de mots dans le code; on le dénote par δ.

Pour chaque mot x du code et chaque entier positif t, on définit la BOULE de rayon t autour de x comme l'ensemble de tous les mots à distance au plus t de x. Autrement dit, c'est l'ensemble de tous les mots qui sont à t erreurs ou moins de x. Formellement, $B_t(x) = \{z \mid d(x, z) \leq t\}$.

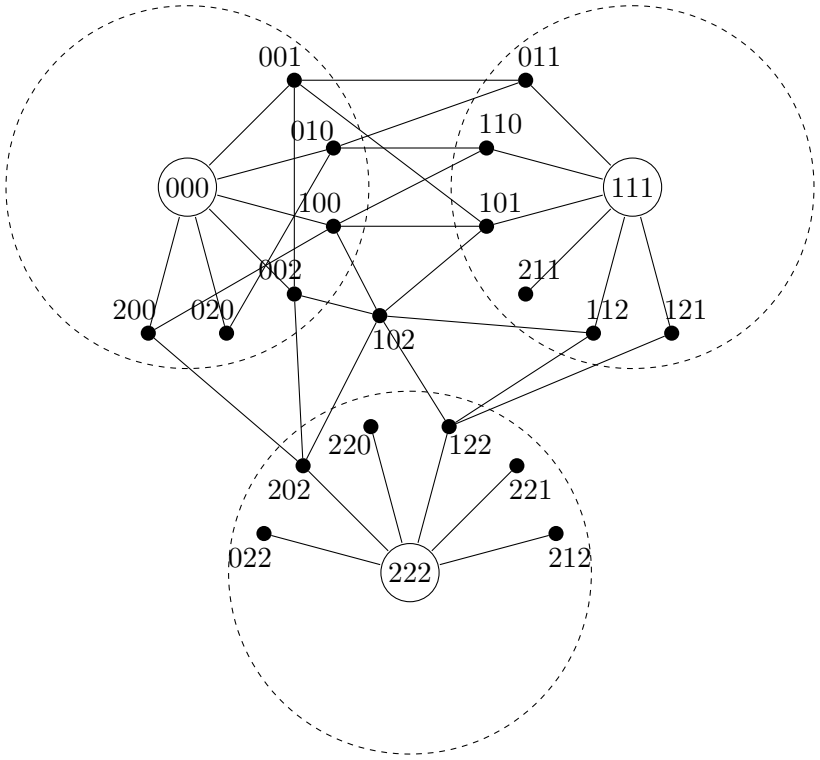
Exemple 13.1. Considérons le code suivant. On décrira ses paramètres.

$$C = \{000, 111, 222\}$$

On a $q = 3$, et $n = 3$: c'est un code de longueur 3 avec 3 symboles. On a $d(000, 222) = 3$, car ces deux mots diffèrent en trois positions. Aussi $d(000, 222) = d(111, 222) = 3$. La distance minimale est $\delta = 3$. Ici, on a $d(x, y) = 3$ pour tout mot $x \neq y$!

Avec $q = 3$ symboles et des mots de longueur $n = 3$, il y a $q^n = 3^3 = 27$ mots possibles. Il y a seulement 3 mots "permis": le code C. La boule de rayon 1 autour de 000 est $B_1(000) = \{001, 010, 100, 011, 101, 110\}$: tous les mots qui sont à une erreur proche du mot 000. □

On peut représenter un code par un graphe. Ici on indique les mots du code par un gros cercle, et les autres mots par des points. Deux mots sont reliés si leur distance est 1. On montre les trois boules $B_1(000)$, $B_1(111)$ et $B_1(222)$ comme cercles pointillés. NB: le dessin n'est *pas* complet, pour ne pas le rendre trop compliqué!



Exercice 13.2. Pour le graphe ci-haut, il manque certains mots et arrêts. Ajouter quelques-uns. Combien de mots sont à l'intérieur de chaque boule de rayon 1? Combien de mots ne sont contenus dans aucune boule de rayon 1? □

On peut comprendre ce graphe comme suit. Certains mots sont "correctes": ce sont les mots du code: 000, 111 et 222. Certains sont incorrectes, mais avec une erreur: par exemple 001. Certains sont incorrectes, mais avec deux erreurs: par exemple 102.

Le graphe montre que pour le code de l'exemple 13.1 les trois boules $B_1(000)$, $B_1(111)$ et $B_1(222)$ sont disjointes. Si on transmet un mot du code, et on sait que la transmission est assez fiable pour avoir au plus une erreur, alors on sait que le résultat, même corrompu, serait encore dans la bonne boule. On pourra alors "corriger" l'erreur: un message corrompu peut être compris parfaitement par le récepteur!

On peut généraliser ces observations.

Théorème 13.3. *Si un code à une distance minimale $\delta \geq 2t+1$, alors les boules de rayon t sont toutes disjointes. Un tel code peut alors détecter et corriger jusqu'à t erreurs par mot.* □

Le code de l'exemple 13.1 a $\delta = 3$, et $3 \geq 2(1)+1$, donc ce code peut corriger 1 erreur.

Pour le code de l'exemple 13.1 il y a certains mots qui sont à distance deux de chaque mot du code (par exemple 102). Si on reçoit ce mot, on sait qu'il y a eu au moins deux erreurs. On peut, dans ce cas, détecter le fait qu'il y a eu (au moins) deux erreurs. Par contre, on ne

peut pas toujours détecter deux erreurs: il se peut que 100 a soumis deux erreurs, mais on ne pourra savoir, car ce mot incorrecte et plus proche à 000 (une erreur).

On peut généraliser.

Théorème 13.4. *Si un code a une distance minimale $\delta \geq 2t$, alors les boules de rayon t sont toutes disjointes sauf possiblement pour les mots à distance t du code. Un tel code peut alors détecter (mais pas nécessairement corriger) jusqu'à t erreurs par mot.* □

Le code de l'exemple 13.1 a $\delta = 3$, et $3 \geq 2(1)$, donc ce code peut détecter 1 erreur.

Exemple 13.5. Soit le code $C = \{00, 11, 22\}$, avec $n = 2$ et $q = 3$.

La distance minimale est $\delta = 2$: ceci se voit directement en calculant la liste de toutes les distances dans le code:

$$d(00, 11) = 2 \qquad d(00, 22) = 2 \qquad d(11, 22) = 2$$

Les boules de rayon 1 ne sont pas disjointes, car $B_1(00) = \{00, 01, 10\}$ et $B_1(11) = \{11, 01, 10\}$. L'intersection est non vide: les mots $\{01, 10\}$ sont dans les deux boules.

On a $\delta = 2 \geq 2(1)$, donc ce code peut détecter une erreur. Par contre on a $\delta = 2 \geq 2(0) + 1$, donc se peut corriger 0 erreur. On pourra savoir qúun message est corrompu, mais on pourra savoir comment l'interpréter. C'est consistant avec les observations sur les boules: un mot comme 01 est visiblement en erreur, mais on ne peut pas décider *quelle* est l'erreur. □

Exercice 13.6. Faire un graphe du code de l'exemple précédant. Inclure tous les mots possibles et les boules de rayon 1. □

Exercice 13.7. Soit le code $C = \{0000, 1111\}$, avec $n = 4$ et $q = 2$. Déterminer la distance minimale. Faire un graphe des mots possibles, incluant les boules de rayon 1. (Si le graphe est trop grand, donner une partie seulement.) Est-ce que les boules de rayon 1 sont disjointes? De rayon 2? Combien d'erreurs est-ce que ce code peut détecter? Combien d'erreurs est-ce que ce code peut corriger? □

Exercice 13.8. Soit le code $C = \{0000, 0011, 1122\}$, avec $n = 4$ et $q = 3$. Déterminer la distance minimale. Faire un graphe des mots possibles, incluant les boules de rayon 1. (Si le graphe est trop grand, donner une partie seulement.) Est-ce que les boules de rayon 1 sont disjointes? De rayon 2? Combien d'erreurs est-ce que ce code peut détecter? Combien d'erreurs est-ce que ce code peut corriger? □

Exercice 13.9. Donner un exemple d'un code de distance minimale 5. Préciser n et q pour votre exemple. Faire un graphique (partiel au besoin!) des mots possibles. Inclure les boules de rayon t, ou t est aussi grand que possible pour que les boules soient disjointes. Combien d'erreurs est-ce que ce code peut détecter? Combien d'erreurs est-ce que ce code peut corriger? □

Exercice 13.10. Soit un code de distance minimale 5. Combien d'erreurs peut-il détecter? Combien d'erreurs peut-il corriger?

Si un code peut détecter 3 erreurs, mais peut corriger seulement 2 erreurs, alors quelle est sa distance minimale? □

Dans le code C de l'exemple 13.1, la distance minimale est $\delta = 3$ qui donne que les boules de rayon 1 sont disjointes. Chaque boule contient 7 mots (un mot du code et 6 mots à une erreur proche). Donc il y a au moins $3 \times 7 = 21$ mots au total (en réalité il y a 27 mots).

Considérons un code C de longueur n avec q symboles. Pour chaque mot du code il y a $n(q-1)$ mots qui sont à une erreur (n positions qui pourront changer à $q-1$ symboles différents). De plus il y a $\binom{n}{2}(q-1)^2$ mots qui sont à une erreur ($\binom{n}{2}$ combinaisons de 2 positions qui pourront chacun changer à $q-1$ symboles différents). En générale, le nombre de mots dans une boule est

$$|B_t(x)| = 1 + n(q-1) + \binom{n}{2}(q-1)^2 + \cdots + \binom{n}{t}(q-1)^t$$

Si toutes les boules sont disjointes, alors la somme de $|B_t(x)|$ pour chaque x dans le code est au plus q^n. Le nombre de boules est exactement égal au nombre de mots dans le code, $|C|$.

Théorème 13.11. *Soit C un code de longueur n avec q symboles et de distance minimale δ. Soit $t = \lfloor(\delta-1)/2\rfloor$. Alors*

$$|C| \leq \frac{q^n}{1 + n(q-1) + \binom{n}{2}(q-1)^2 + \cdots + \binom{n}{t}(q-1)^t}$$

Preuve. Si un code a une distance minimale δ, alors les boules de rayon t seront disjointes, où $t = \lfloor(\delta-1)/2\rfloor$. Le dénominateur est donc exactement le nombre de mots dans une boule. Le numérateur est le nombre de mots possibles. Le nombre de boules, multiplié par le nombre de mots dans une boule, ne peut pas dépasser le nombre de mots possibles. □

Un code tel que l'inégalité dans théorème 13.11 est une égalité est un CODE PARFAIT.

Exemple 13.12. Considérons le code de l'exemple 13.1. Ici on a $\delta = 3$, donc avec $t = 1$ on a les boules de rayon 1 disjointes. De plus, $|B_1(x)| = 1 + 3(3-1) = 7$. Donc

$$|C| \leq \frac{q^n}{|B_1(x)|} \qquad \rightarrow \qquad 3 \leq \frac{3^3}{7} = \frac{27}{7}$$

Puisque $3 < 27/7$ ce code n'est pas parfait. □

Si un code n'est pas parfait, alors il existe des mots ambigus: on ne sait pas comment les interpréter. Dans le code de l'exemple 13.1, le récepteur ne sait pas comment interpréter le mot 012: évidemment il y a eu au moins deux erreurs, mais si on accepte la possibilité de deux erreurs dans un mot alors le mot 001 devient ambigu aussi.

Exercice 13.13. Pour les codes des exercices ci-haut, lesquels sont parfaits? □

Exercice 13.14. Donner un exemple d'un code parfait de distance minimale 3, et un code parfait de distance minimale 5. (indice: $q = 2$) □