PF1 — Principes de Fonctionnement des machines binaires

Jean-Baptiste Yunès
Jean.Baptiste.Yunes@univ-paris-diderot.fr

31/10/2014

Numérisation

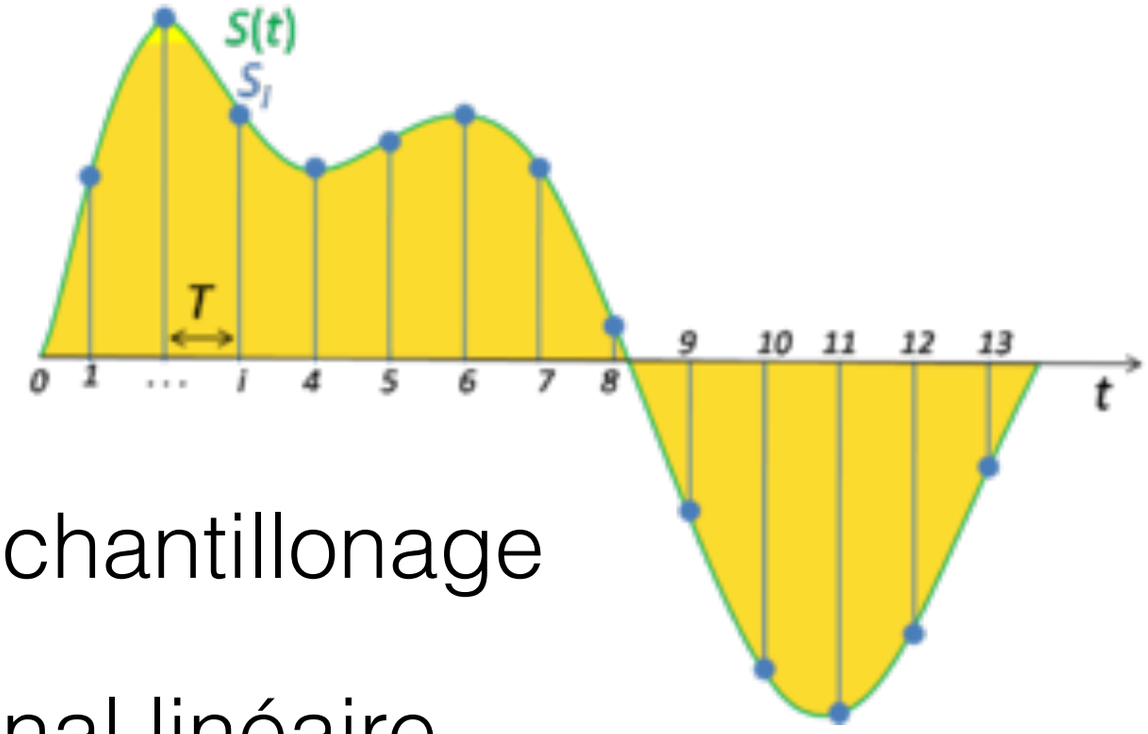
Numérisation

- La **numérisation** désigne le processus consistant à convertir un signal analogique en données numériques

  - Il fait appel à l'**échantillonnage** qui consiste à sélectionner dans un ensemble un nombre fini d'éléments considérés comme représentatifs. L'échantillonnage est une approximation.

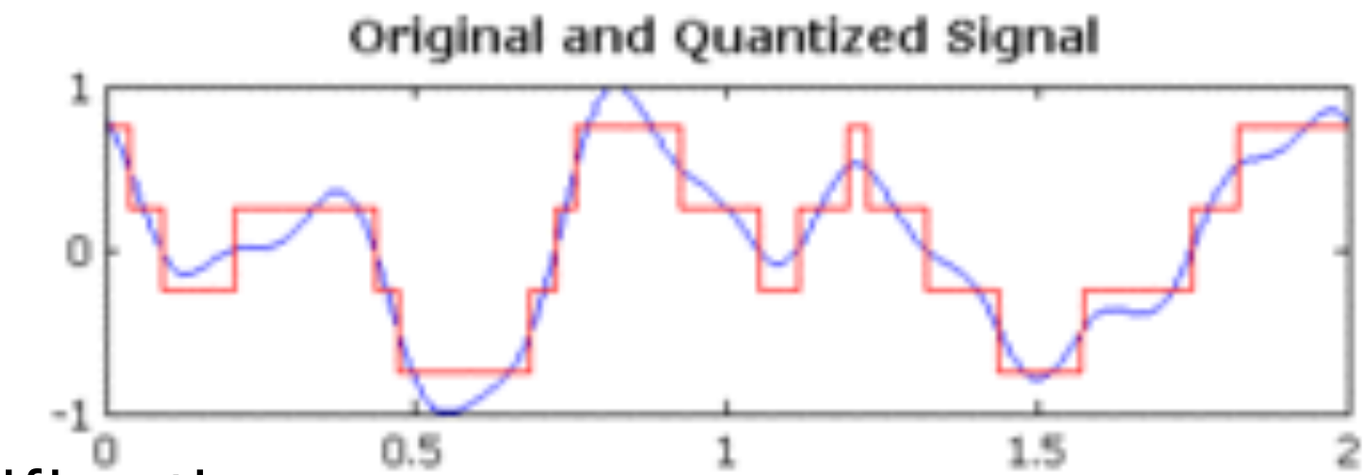
  - Il fait appel à la **quantification** qui consiste à associer à chaque élément prélevé une valeur numérique discrète.

- Ces deux paramètres conditionnent la qualité l'approximation et donc de la reproduction du signal original.

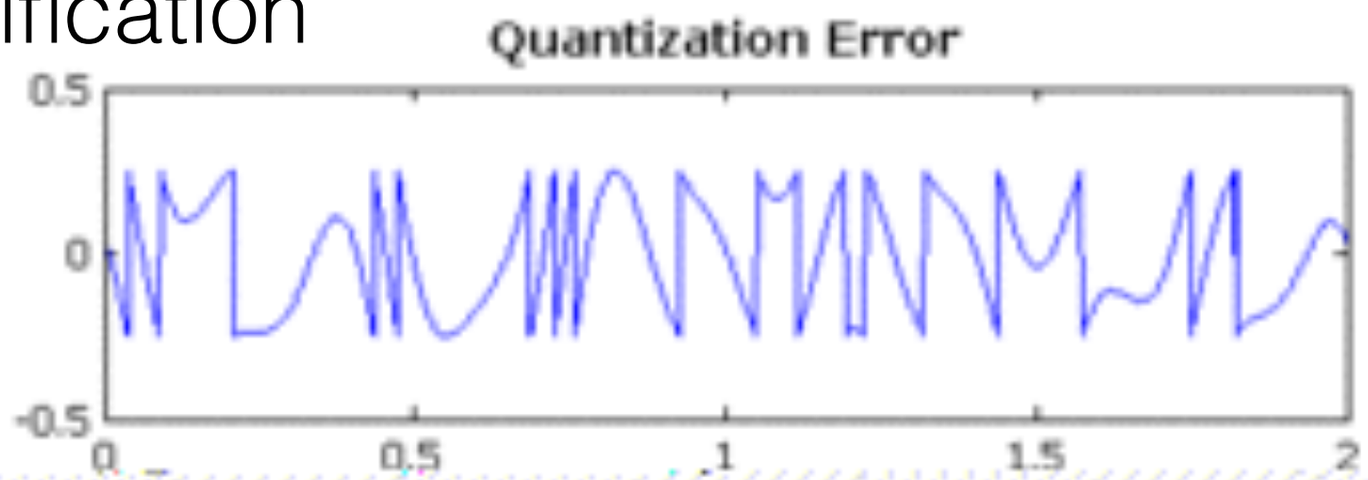


échantillonage

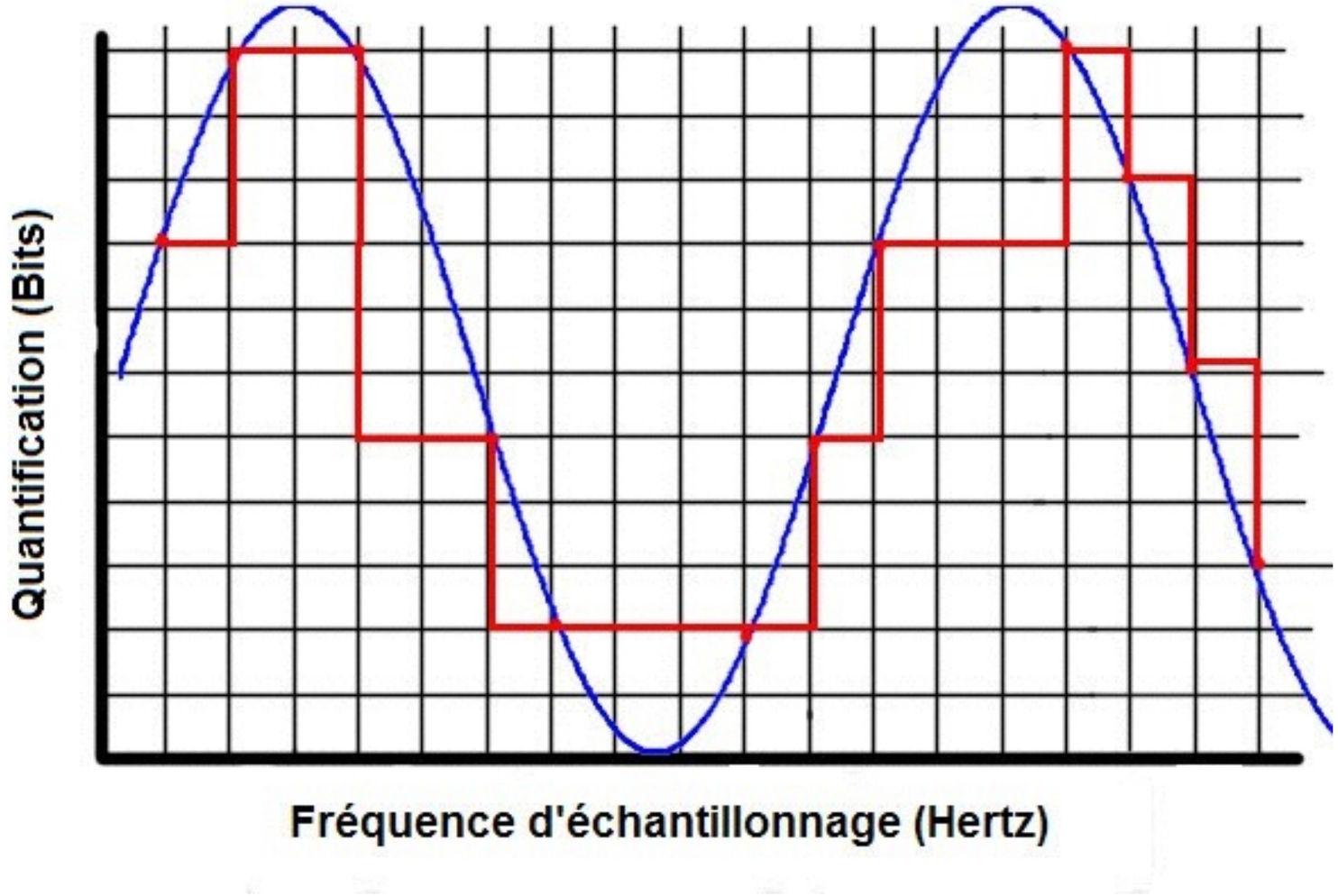
- Example d'un signal linéaire



quantification



- Reproduction du son :
    - avant 1980, reproduction de l'onde sonore par capture de sa forme analogique via un dispositif transformant la pression acoustique en déplacement mécanique et gravage dans un support (cire, plastique). Le dispositif étant facilement réversible (mouvement -> pression)
    - après 1980, numérisation de l'onde et enregistrement ou gravage de son codage



- L’échantillonage est mesuré en Hertz (l’échantillonnage est très généralement effectué à intervalles réguliers - exception : VBR)

- La finesse de la quantification est mesurée en bits

- la reconstitution approximative d'un signal analogique depuis sa forme numérisée nécessite une fréquence d'échantillonnage au moins deux fois plus élevée que la largeur de la bande. (Théorème de Nyquist-Shannon)

    - Par exemple, l'oreille humaine a une bande passante de 20Hz à 20kHz soit approximativement une largeur de bande de 20kHz, la reproduction d'un son destiné à l'oreille humaine nécessite alors une fréquence d'échantillonnage d'au moins 40kHz

        - La numérisation employée pour les CD est de ~44kHz avec une quantification sur 16 bits

        - le choix de 16 bits (14 suffisent) est justifié scientifiquement mais dépasse largement le cadre de ce cours (quantification logarithmique, rapport signal-bruit)

Compression

- Code compresseur
  - il s’agit ici d’obtenir une représentation plus compacte
    - en vue de transmission (économie de bande passante)
    - en vue de stockage (économie d’espace)

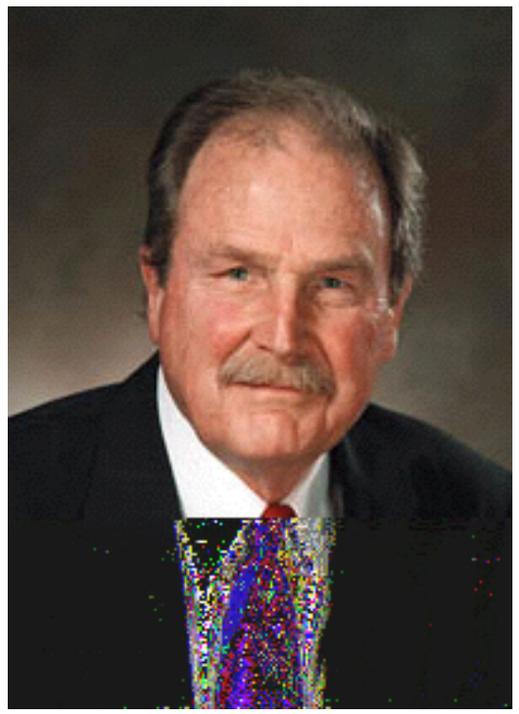
- Deux types de **compression** :
  - **conservative** ou **sans perte** : le message originel peut-être reconstruit à l'identique en inversant la fonction
    - ce type de compression est recherché avec du texte par exemple…
      - je peux vouloir compresser l'œuvre complète de Victor Hugo, mais je souhaite retrouver le texte d'origine!

- Deux types de **compression** :
  - **non conservative** ou **avec perte** : le message originel n'est pas reconstruit à l'identique, mais un message similaire est obtenu à l'inversion
    - souvent le cas des images et des sons
      - il n'est pas nécessaire que des détails quasi-invisibles-sensibles soient conservés dans les photos prises par mon appareil…
    - cette compression permet d'obtenir de très bons taux de compression

- Codage compressé d'un texte…
  - utilisation d'un code à longueur variable (fréquence des lettres), Huffman, Lempel-Ziv
  - création d'un code permettant d'encoder des groupes de lettres en fonction de la fréquence, calcul dynamique de la fréquence…

Code de Huffman

- David Albert Huffman



Source Wikipédia

- Idée : coder avec des mots de petite longueur les lettres les plus fréquentes, coder avec des mots de plus grande longueur les lettres les moins fréquentes

- ex : trois lettres A, B, C avec A très fréquente, B moyennement et C rare

  - on peut utiliser quelque chose comme $\boldsymbol{\tau}$(A)=0, $\boldsymbol{\tau}$(B)=10 et $\boldsymbol{\tau}$(C)=11

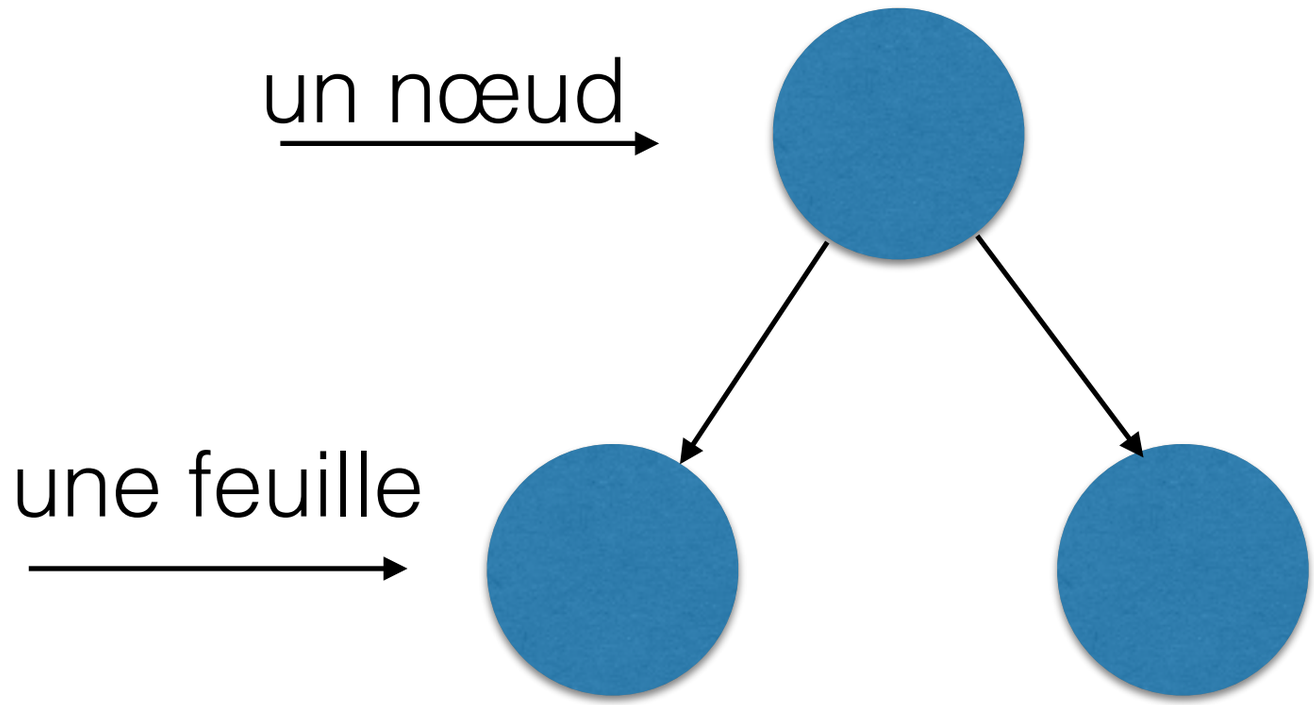
  - soit 1 bit pour A, 2 pour B et C

  - Ainsi le texte AAABAAABBAAAAC serait codé par 00010000101000011 soit 18 bits.

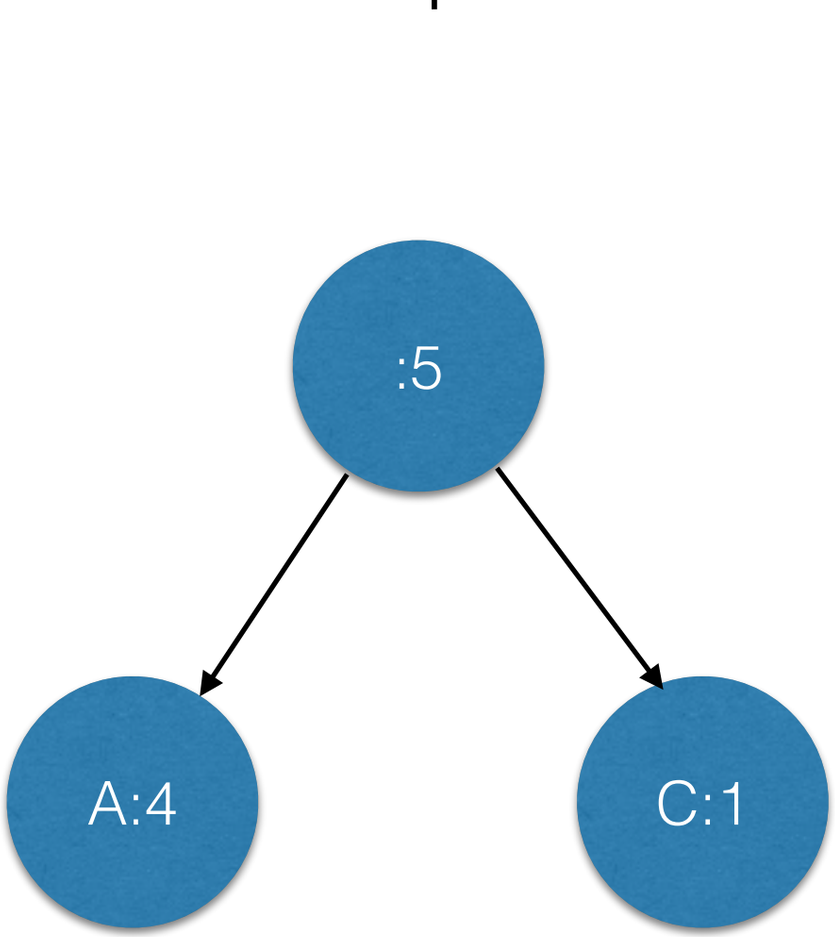
  - Un codage ordinaire sur 2 bits/caractères aurait donné 14*2 soit 28 bits.

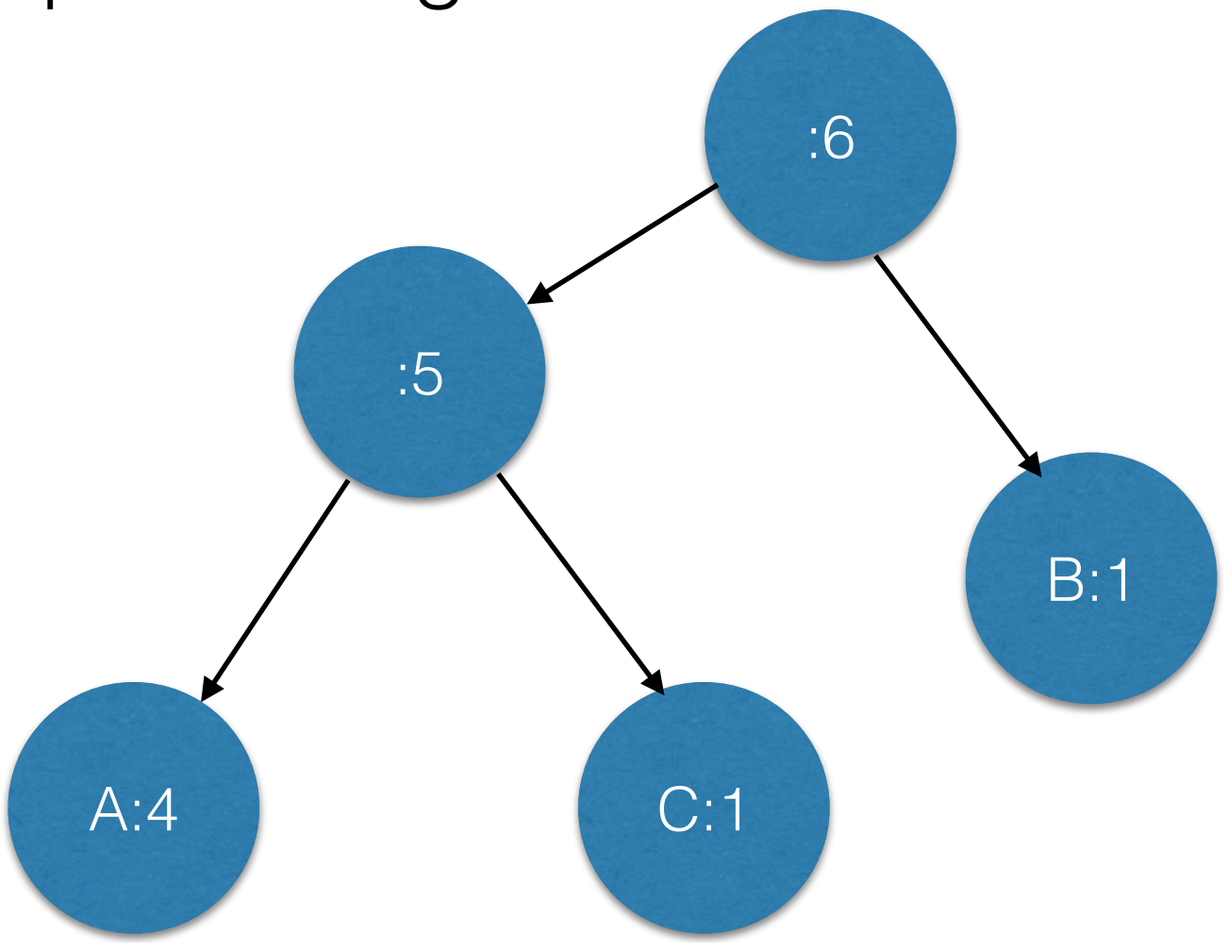
  - On obtient un taux de compression de 18/28 soit ~65%, 2/3…

- Comment obtenir un codage à partir des fréquences des lettres ?
  - on va fabriquer un arbre
    - c'est une structure dans laquelle on trouve des nœuds
      - un **nœud** permet de désigner d'autres nœuds
      - un nœud qui ne désigne rien est une **feuille**
      - un nœud sans ascendant est appelé **racine**



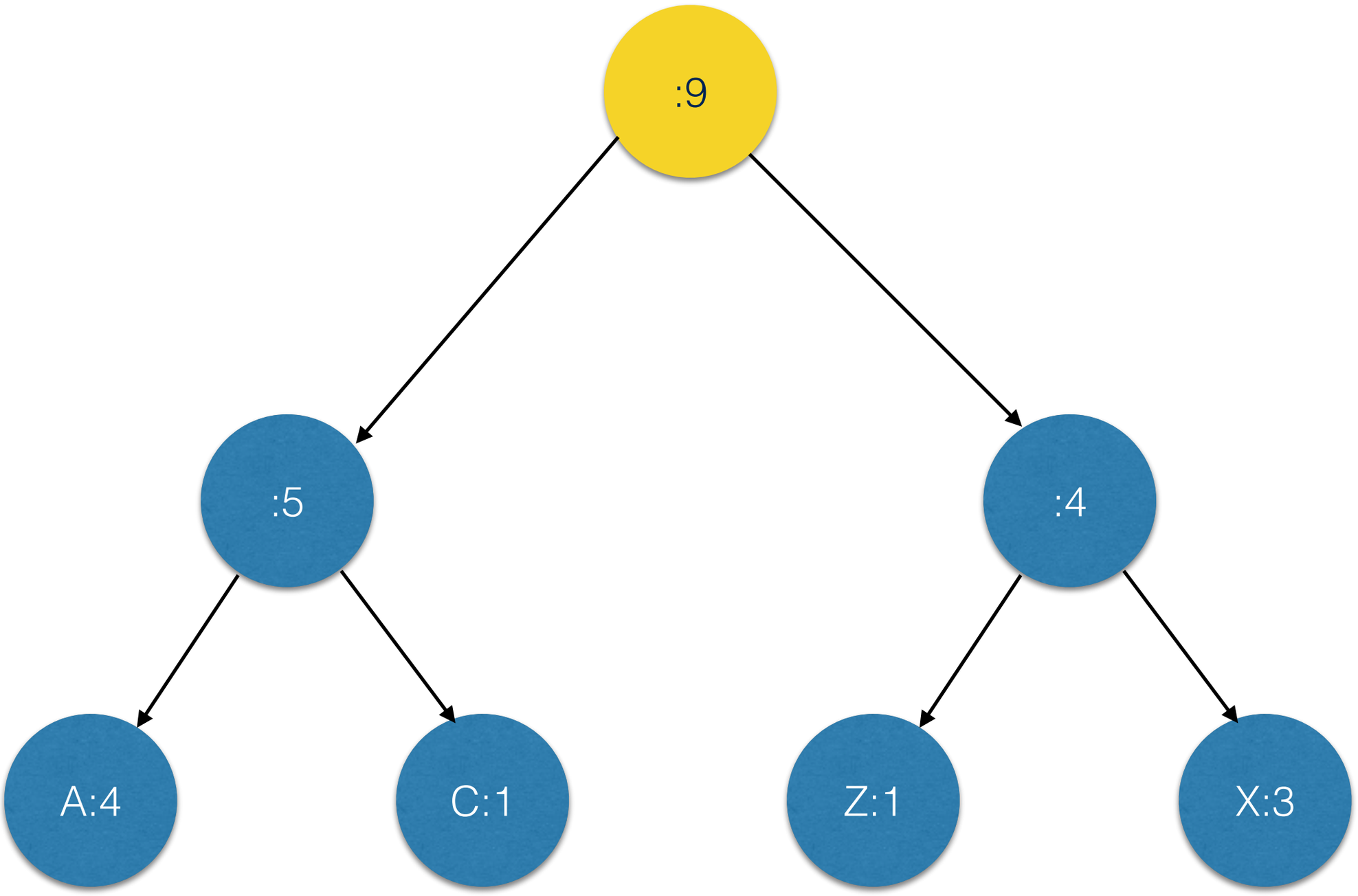
- on va fabriquer un arbre binaire (avec uniquement des nœuds à deux descendants) dans lequel
  - les feuilles représenteront les lettres et leur fréquence associée
  - les nœuds représenteront la somme des fréquences des lettres qu'ils désignent



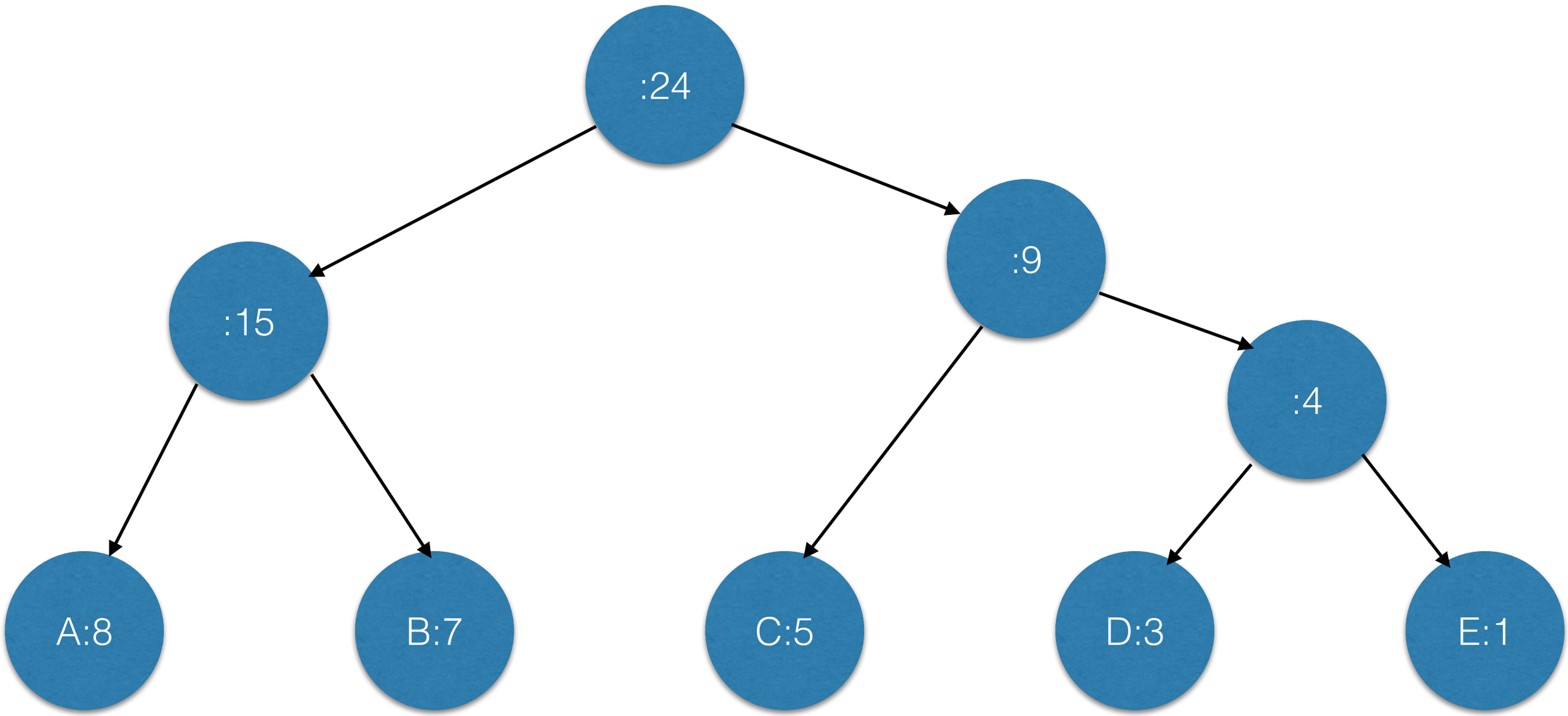


- Attention : algorithme! (enfin presque)
- Pour fabriquer cet arbre on part des arbres réduits aux simples lettres
- À chaque étape, on sélectionne deux arbres dont les fréquences des racines sont les plus petites et on fabrique un arbre dont le nœud racine pointera vers les deux arbres sélectionnés et dont la fréquence sera simplement la somme des fréquence

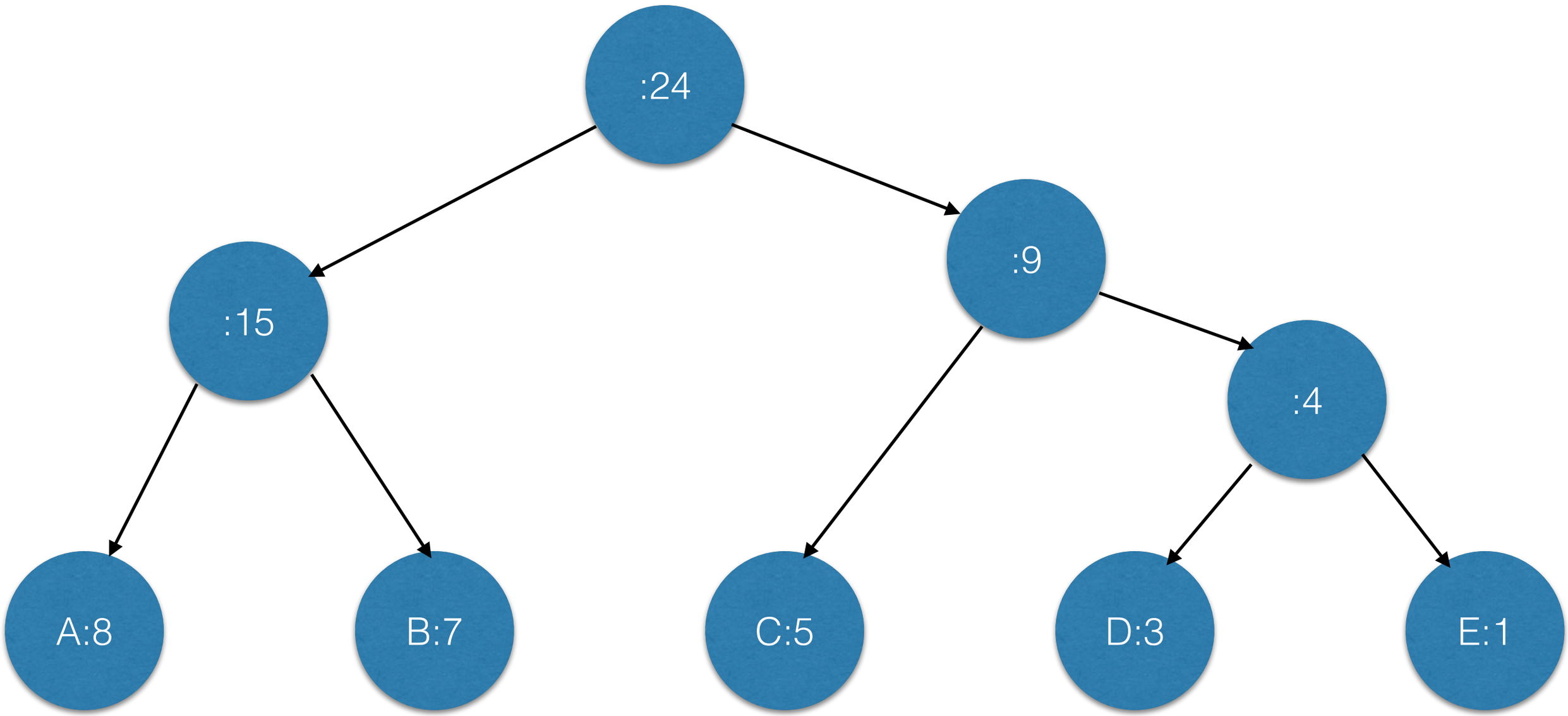
- Exemple de la fusion de deux arbres…



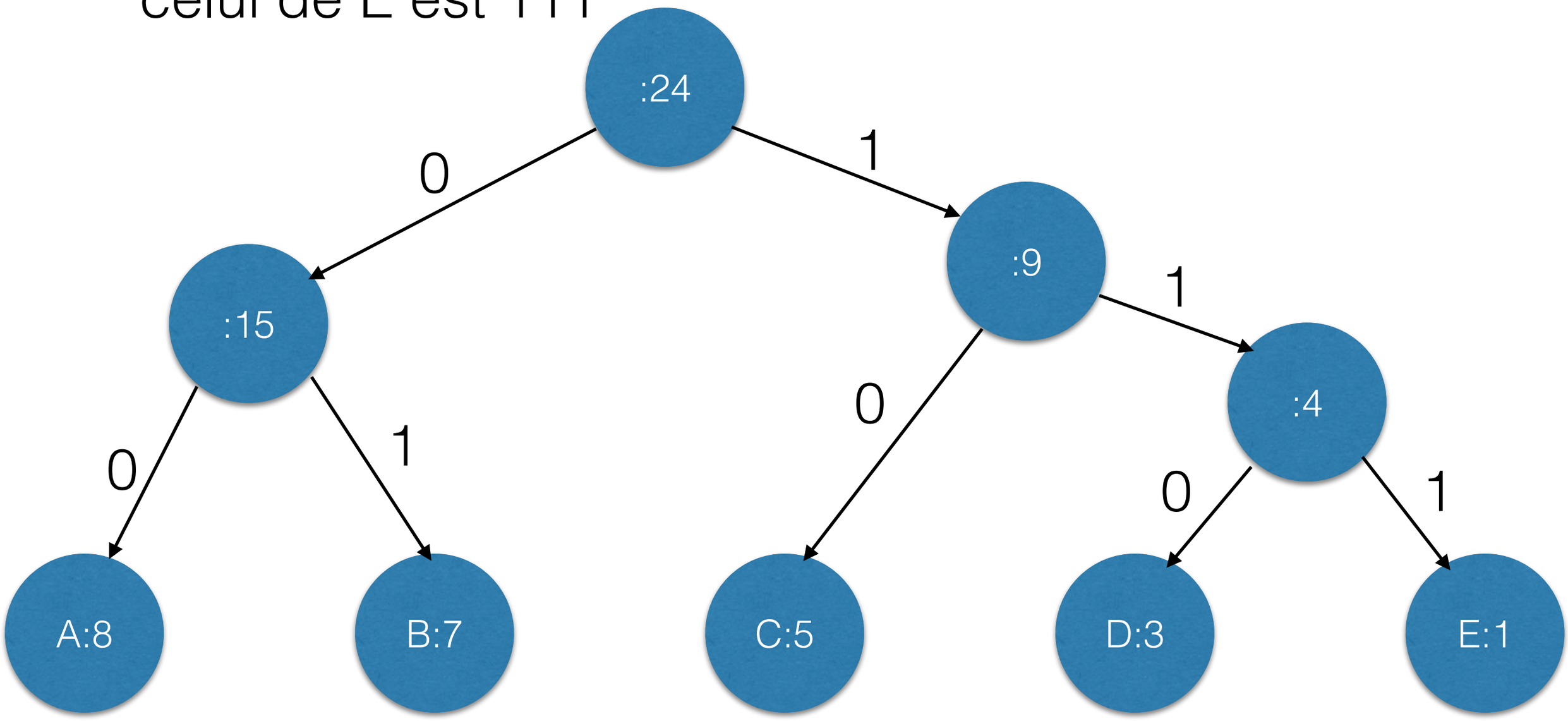
- Prenons un exemple avec les lettres des fréquences suivantes :
  - A:8, B:7, C:5, D:3, E:1



- Ce que nous avons obtenu est l'**arbre de Huffman**, on va l'utiliser pour coder les lettres!



- Les flèches qui descendent à gauche coderont 0
- Les flèches qui descendent à droite coderont 1
- Le codage de A est donc 00, le codage de B est 01, le codage de C est 10, celui de D est 110 et celui de E est 111



- Un codage ordinaire aurait conduit à utiliser 3 bits par lettre.

- Le codage de A est donc 00, le codage de B est 01, le codage de C est 10, celui de D est 110 et celui de E est 111

  - Ici les lettres les plus fréquentes sont codées sur 2 bits et les plus rares sur 3

- Essayons de coder EABDACABBCABABD

- 11100011100010000101100001000111O

- 45 bits vs 33 bits! 33/45 ~ 75%

- Fréquence des lettres en Français (source Wikipédia)

	F		F
A	8,25 %	N	7,25 %
B	1,25 %	O	5,75 %
C	3,25 %	P	3,75 %
D	3,75 %	Q	1,25 %
E	17,75 %	R	7,25 %
F	1,25 %	S	8,25 %
G	1,25 %	T	7,25 %
H	1,25 %	U	6,25 %
I	7,25 %	V	1,75 %
J	0,75 %	W	0,00 %
K	0,00 %	X	0,00 %
L	5,75 %	Y	0,75 %
M	3,25 %	Z	0,00 %

- Fréquence des lettres en Français (source Wikipédia)

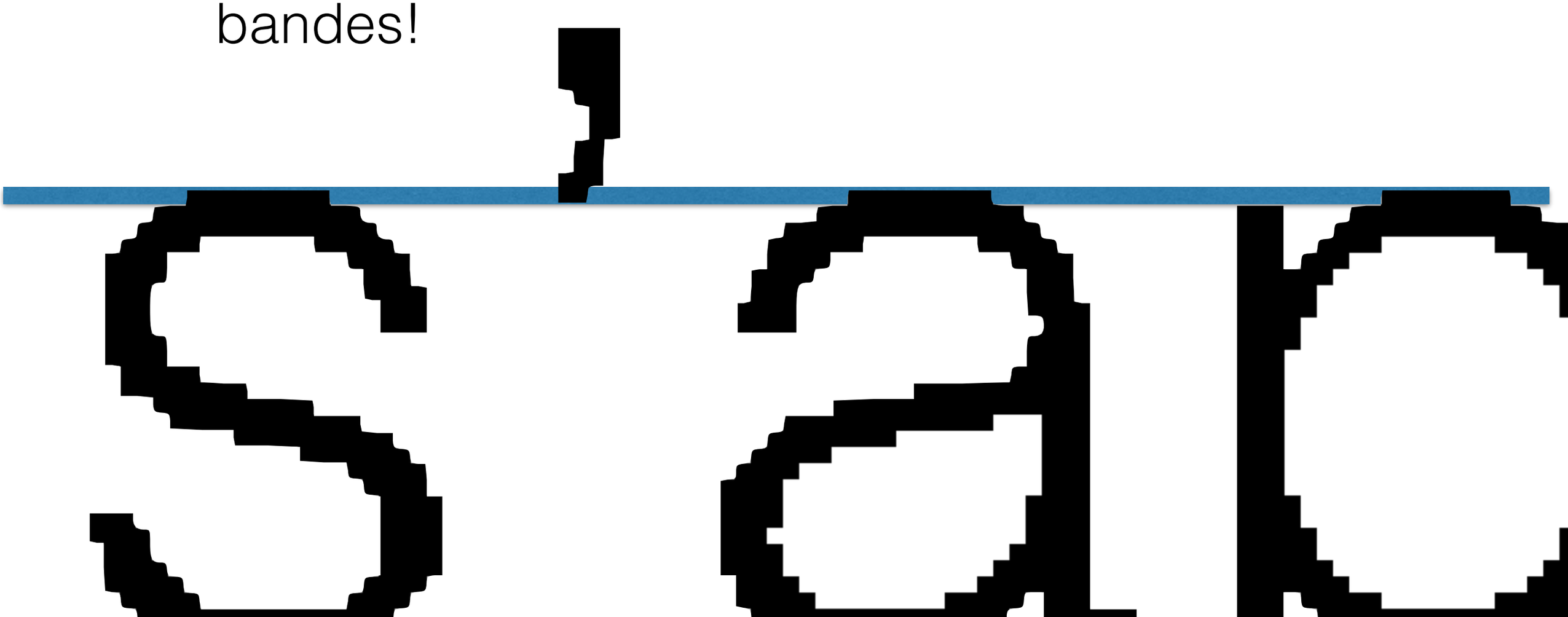
	F		F
E	17,75 %	M	3,25 %
A	8,25 %	V	1,75 %
S	8,25 %	B	1,25 %
I	7,25 %	F	1,25 %
N	7,25 %	G	1,25 %
R	7,25 %	H	1,25 %
T	7,25 %	Q	1,25 %
U	6,25 %	J	0,75 %
L	5,75 %	Y	0,75 %
O	5,75 %	K	0,00 %
D	3,75 %	W	0,00 %
P	3,75 %	X	0,00 %
C	3,25 %	Z	0,00 %

Compression d’image

- La compression d’image recouvre de nombreuses techniques très différentes…
- Les meilleures compressions sont avec perte
- mais sur certaines images on peut vraiment gagner même sans perte…

- Prenons les images fabriquées avec un ordinateur (non numérisées)
  - beaucoup d'entre elles ont de larges bandes uniformes
    - on peut exploiter cette propriété

- Le codage RLE (Run-Length Encoding) s'applique assez bien aux images noires et blanc
  - dans l'image ci-dessous on a pour chaque ligne une alternance de bandes de pixels noirs et pixels blancs
  - on va donc coder simplement la longueur de ces bandes!



- Au lieu de coder le mot de 40 bits directement 0000001111111000000000001111111000000000
- on va coder la suite des longueurs 6 6 12 7 9
- pour cela il nous faut 4 bits par nombre et donc
- 0110 0110 1100 0111 1001
- soit 20 bits seulement!

Cryptographie

- écriture cachée selon les Grecs
    - des cas extrêmement simples
        - le chiffre de César, ROT13, Vigénère, le masque jetable de Vernam (brrr)
    - des cas plus compliqués
        - LFSR (linear feedback shift register - registre à décalage à rétroaction linéaire), DES, RSA

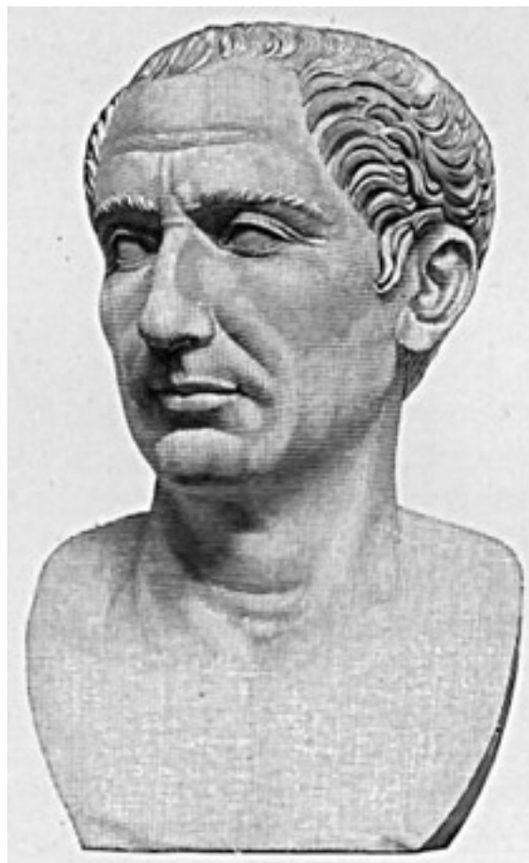
- Idée : utiliser une fonction difficilement inversible pour coder un texte. Le secret est justement la fonction inverse…

Le chiffre de César

- Attribué à l'empereur César (Caius Iulius Caesar IV -100 — -44)

- il l'aurait utilisé pour masquer certaines correspondances

- c'est un chiffrement monoalphabétique par substitution



source Wikipédia

- Il repose sur une permutation circulaire de l'alphabet
  - les lettres sont décalées de p (pour un p choisi) rangs dans l'ordre alphabétique
    - p=3, A->D, B->E, etc
      - On peut encore utiliser l'arithmétique modulaire (décidément) pour le définir…
      - Q: son inverse est aussi un chiffre de César. Lequel ?
- Le codage ROT13 est le chiffre de César pour p=13

- L’utilisation du chiffre de César ne peut pas bluffer quelqu’un très longtemps ?
    - Force brute (seulement 25 chiffres possibles!)
        - vous seul avec un (petit) poil de courage
        - 25 esclaves
        - un ordinateur
- Analyse de fréquences pour casser le code…

Chiffre de Vigénère

- Créé par Blaise de Vigénère (1523-1596)
- Il a fallu attendre environ 300 ans avant de trouver la méthode permettant de casser ce code
    - Merci Friedrich Wilhem Kasiski

source Wikipédia

- Comment ça marche ?

- On va utiliser différents chiffres de César pour les lettres du message

  - pour César la clé est p, le décalage

  - pour Vigénère on utilise n clés p_i, $0 \leq i < n$, en fait la clé est elle-même un texte (secret) en général plus court que le texte à encoder

	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
A	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
B	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	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	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

- Clé secrète : `INFORMATIQUE`
- Message : `JADORELINFORMATIQUE`

- Message codé : `RNICIQLBVVIVUNYWHGE`
  `INFORMATIQUEINFORMATIQUE`
  `JADORELINFORMATIQUE`

- Casser ce code n’est pas très difficile, si le message est assez long
    - On peut y repérer des répétitions et deviner la longueur de la clé, sinon on peut essayer diverses longueurs de clés
    - Puis faire des analyses fréquentielles
    - Un bon ordinateur (ou beaucoup de patience) et le tour est joué…

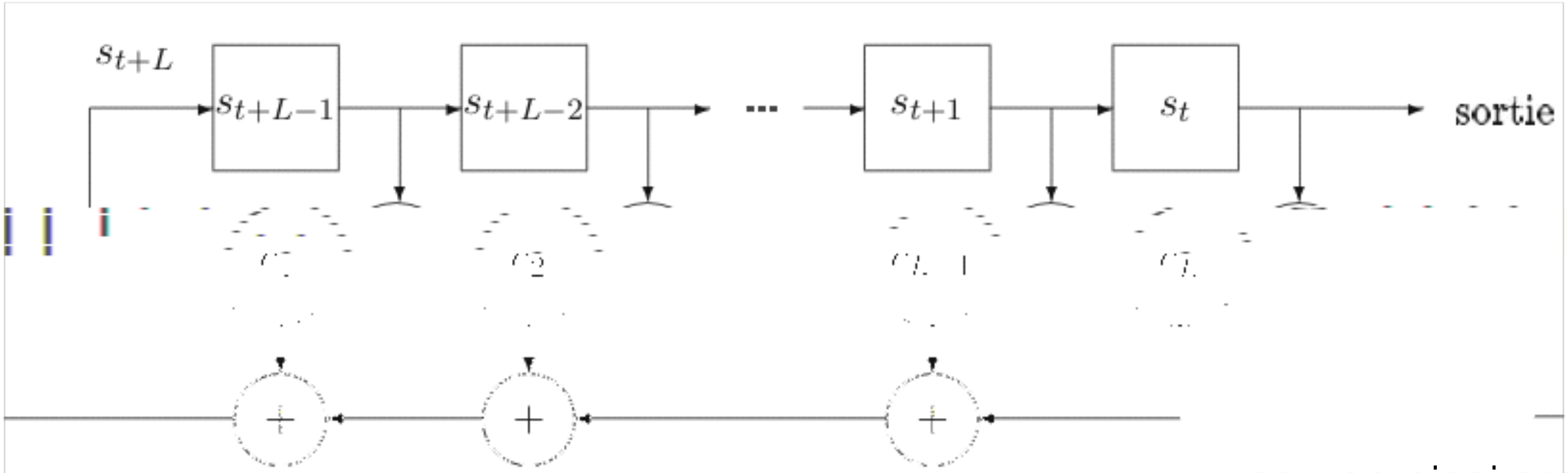
- Q: Vous avez reçu le message `RNNAVFRHXBYWUNYVJ`
- êtes-vous d’accord ?

Chiffre de Vernam/Mauborgne ou masque jetable (one-time pad)

- Un chiffre de Vigénère pour lequel :
  - la clé est aussi longue que le texte
  - la clé est obtenue par distribution aléatoire
  - la clé ne doit être employé qu'une seule fois
- Si les conditions sont réunies, le chiffre est inviolable
  - il a probablement été utilisé pour sécuriser le « téléphone rouge »

- Aujourd'hui on utilise des systèmes plus solides…
  - symétriques ou à clé secrète
  - par blocs (par exemple le DES)
  - ou par flot
  - les plus standards ne sont pas considérés comme très solides
    - le DES peut être cassé en quelques jours avec quelques dizaines d'ordinateurs
    - mais c'est utile tout de même si vos messages n'ont pas un caractère vital ou si la durée de vie du contenu du message n'est pas très longue

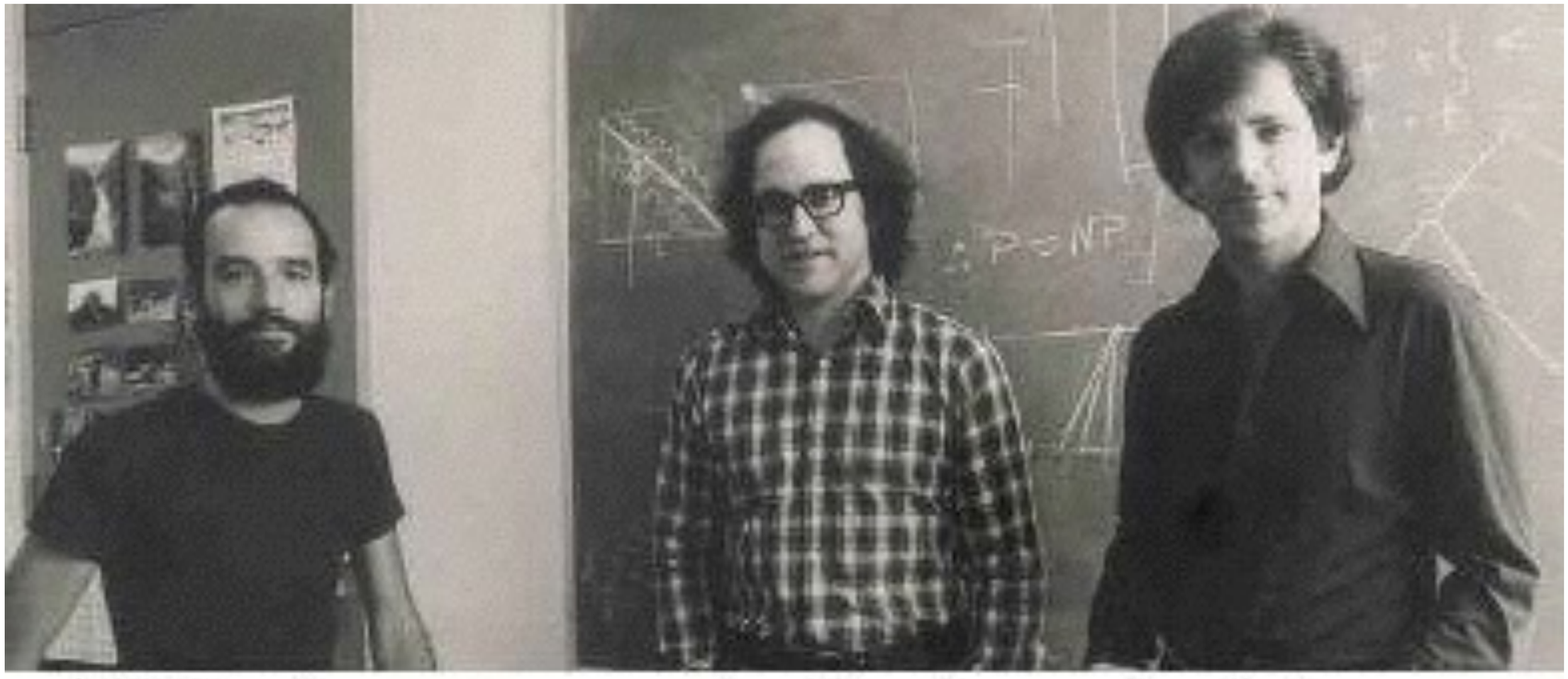
LFSR



source picsi.org

- permet d'obtenir une suite pseudo-aléatoire…
  - utile, par exemple, pour obtenir un masque jetable

RSA



Adi Shamir Ron Rivest Len Adleman

- Aujourd'hui on utilise des systèmes plus solides que les chiffrement par flot pour les échanges très secrets…
    - asymétriques ou à clés publiques
        - le RSA (Rivest Shamir Adleman)
- On utilise ces systèmes pour transmettre des clés secrètes de chiffrement par flots.

- Comment ça marche ?
    - Encore des histoires de nombres et calculs modulaires

1. on prend 2 nombres p,q
2. n = p.q, ϕ(n) = (p-1)(q-1)
3. on choisit e premier avec ϕ(n) et plus petit que ϕ(n)
4. on calcule d l'inverse de e module ϕ(n), c'est-à-dire tel que e.d ≡1 (mod ϕ(n))
5. (n,e) est la clé publique que tout le monde peut connaître et utiliser pour coder, (n,d) est la clé privée et qui sert à décoder

- Pour chiffrer le message M :
  - on calcule $C \equiv M^e \pmod n$
- Pour déchiffrer C :
  - on calcule $M \equiv C^d \pmod n$
- La difficulté repose sur le fait que connaissant n et e il est très difficile de trouver d car il faut pour cela connaître p et q c'est-à-dire décomposer le nombre n en facteurs premiers…
  - Attention p,q,d,e sont normalement de très très très grand nombres!

- Allons-y
- on prend p=7 et q=11
- n = 7*11 = 77
- ϕ(n) = 6*10 = 60
- prenons e=7 qui est premier avec 60
- on calcule d tel que 7*d ≡ 1 (mod 60), d=43 fonctionne car 43*7=301=5*60+1
- clé publique (7,77), clé privée (43,77)
- prenons BONJOUR, soit en ascii 42 4F 4E 4A 4F 55 52, on prend la valeur décimale 18664546135659858, puis on découpe en une suite de nombres plus petits que n=77 donc 18 66 45 46 13 56 59 8 58

- 18 66 45 46 13 56 59 8 58

- on chiffre
  - 18^7 mod 77 = 39
  - 66^7 mod 77 = 66
  - 45^7 mod 77 = 45
  - 46^7 mod 77 = 18
  - 13^7 mod 77 = 62
  - 56^7 mod 77 = 56
  - 59^7 mod 77 = 38
  - 8^7 mod 77 = 57
  - 58^7 mod 77 = 9
- Soit le message codé 39 66 45 18 62 56 38 57 9

- 39 66 45 18 62 56 38 57 9
- on décode
  - 39^43 mod 77 = 18
  - 66^43 mod 77 = 66
  - 45^43 mod 77 = 45
  - 18^43 mod 77 = 46
  - 62^43 mod 77 = 13
  - 56^43 mod 77 = 56
  - 38^43 mod 77 = 59
  - 57^43 mod 77 = 8
  - 9^43 mod 77 = 58
- Soit 18 66 45 46 13 56 59 8 58 donc 186645461356598 58 et en hexa 424F4E4A4F5552
- BONJOUR

- Vous pouvez toujours vous demander comment calculer une puissance modulo avec des grands nombres ?
  - par exemple 3^{259} (mod 127) ?
  - Il n'est pas nécessaire de calculer 3^{259}
  - on sait que :
    - 3^1 = 3 (mod 127)
    - 3^2 = 9 (mod 127)
    - 3^4 = 81 (mod 127)
    - 3^8 = 84 (mod 127)
    - 3^16 = 71 (mod 127)
    - 3^32 = 88 (mod 127)
    - 3^64 = 124 (mod 127)
    - 3^128 = 9 (mod 127)
    - 3^256 = 81 (mod 127)
  - or $3^{259} = 3^{256}.3^2.3$ donc $3^{259} \equiv 81.9.3 \equiv 28$ (mod 127)

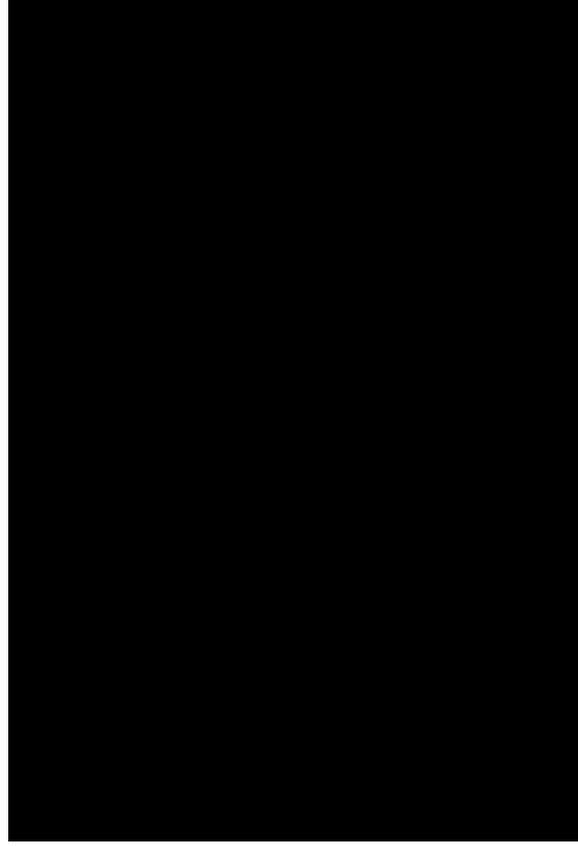
Contrôle d’erreur

- CRC
- Hamming

- Le principe est de fournir de la redondance
  - doubler/tripler le message
- un autre exemple : l'alphabet radio international
  - PF1 : Papa, Foxtrot, One
  - on rajoute de l'information permettant d'assurer la bonne lisibilité du message en cas de bruit

- Le CRC était employé dans la transmission du code ASCII, en ajoutant un 8-ième bit de contrôle
    - de sorte que la somme des huit bits soit toujours paire
    - permet de détecter si **une** erreur s'est produite, mais pas où
    - ne permet pas de détecter deux erreurs…

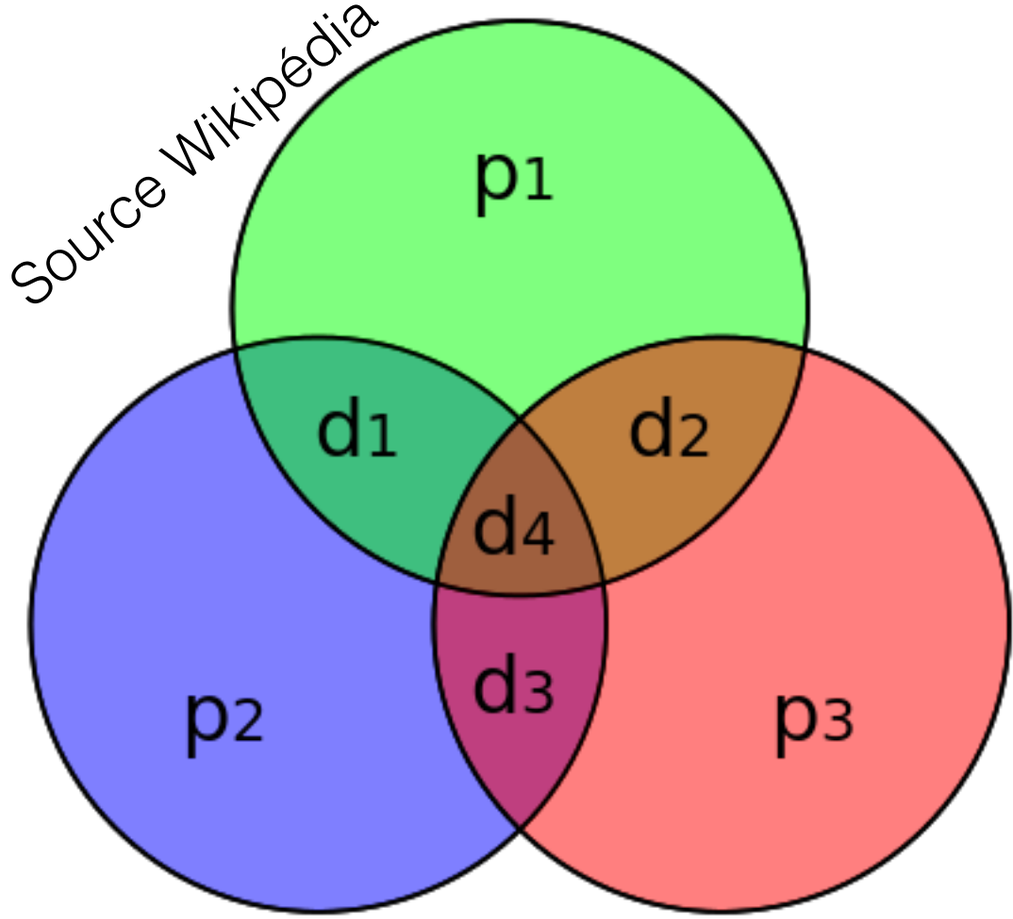
- Le code de (Richard) Hamming
- une famille de codes
- permet de
  - détecter
  - et corriger



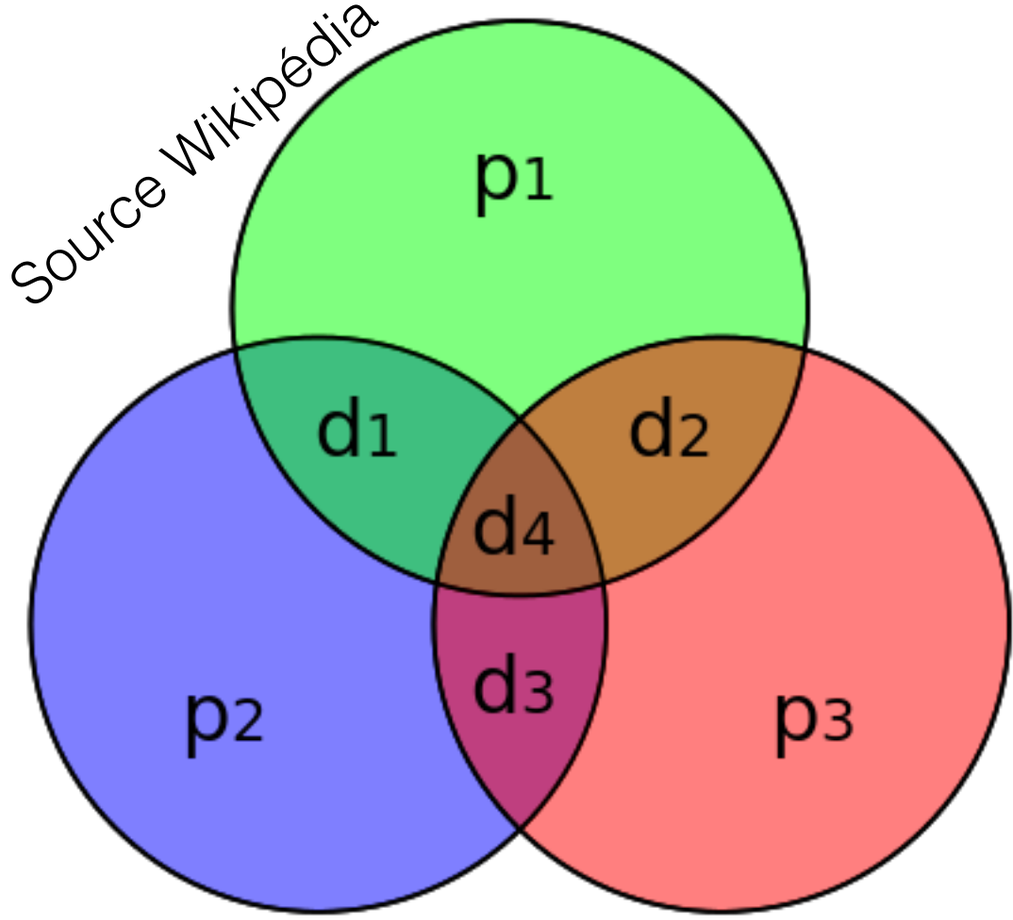
source acm.org

- son principe est de calculer plusieurs parités sur différentes parties du mot de sorte qu'il soit possible de détecter si erreur il y a et où.
  - le code de Hamming le plus simple est le [7,4] (ou [7,4,3])
  - il code des mots de 4 bits sur 7 bits (c'est le prix à payer)

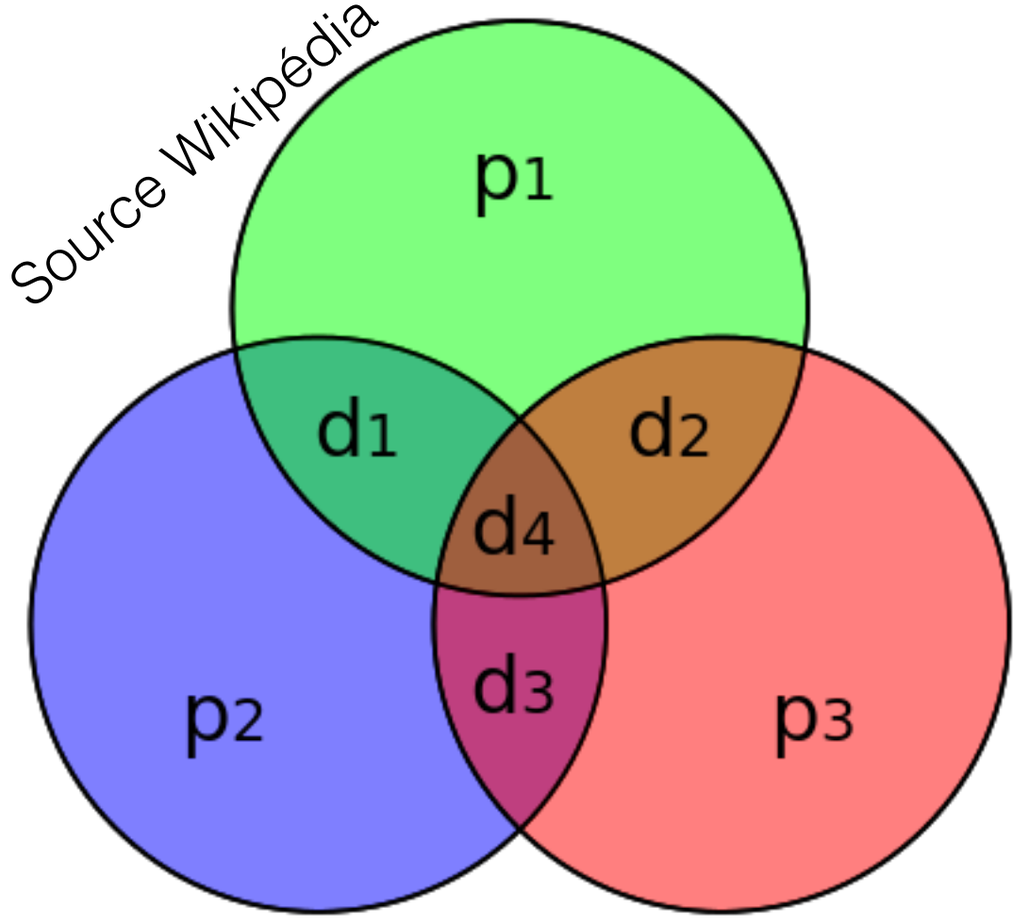
- Le message est $d_1d_2d_3d_4$



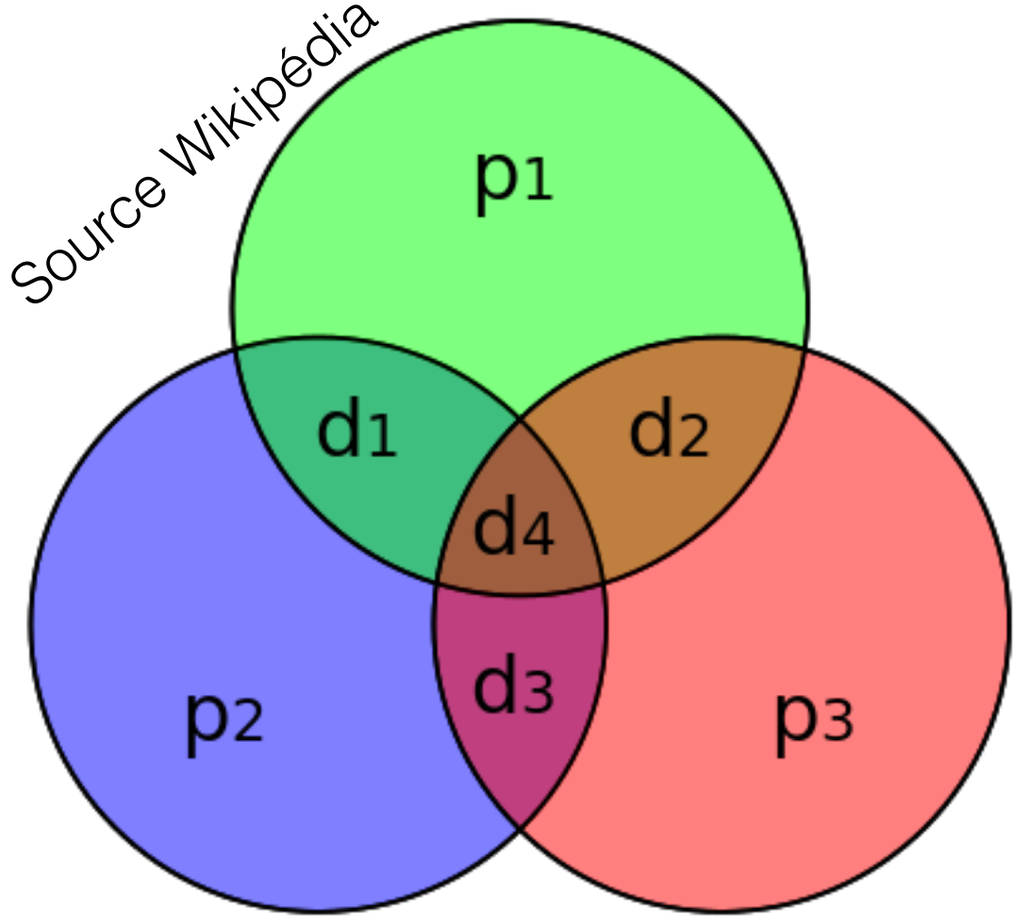
- On y rajoute 3 bits $p_1p_2p_3$ de sorte que
- p_1 est la somme de contrôle de $d_1d_4d_2$
- p_2 la somme de contrôle de $d_1d_4d_3$
- p_3 la somme de contrôle de $d_2d_4d_3$



- Le message est 0101
- p_1=0, p_2=1, p_3=0
- Le message codé est 010**0101**



- Le message est 0101
- p_1=0, p_2=1, p_3=0
- Le message codé est 010**0101**
- Si deux contrôles sont corrects c'est que le message est correct et le troisième contrôle faux
  - Si je reçois **1**100101, je sais que p_1 est faux!



- Le message est 0101
- p_1=0, p_2=1, p_3=0
- Le message codé est 010**0101**
- Si un contrôle est correct, c'est que le message est faux et l'erreur est sur le bit qui n'est pas couvert pas le contrôle correct
  - Si je reçois 010**1**101, je sais que d_1 est faux