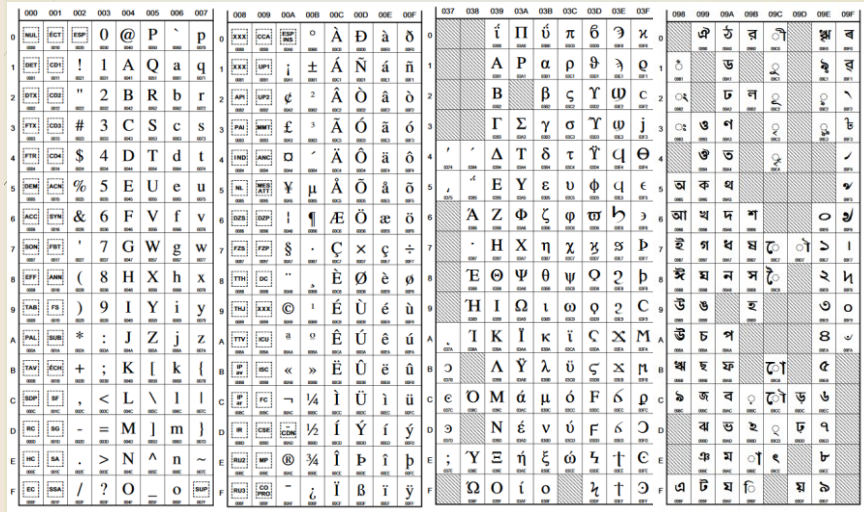
JFA - 51

Table de Codage de L'UNICODE

Exemples de tables de codage :



https://fr.wikipedia.org/wiki/Table_des_caract%C3%A8res_Unicode_(0000-0FFF)

(caractères latins) (latins accentués) (caractères grec et copte) (caractères bengali)

JFA - 52

Code UTF 8, UTF 16, UTF 32

- ***UTF :*** *Universal Character Set Transformation Format*
- Pour résoudre le problème du doublement de la taille des fichiers et de l'incohérence des représentations, Ken Thompson, a inventé le 2 septembre 1992 sur un coin de table, le codage UTF-8, une autre manière de coder les caractères Unicode.
- L'idée a été de coder les caractères :
  - les plus utilisés : de 0 à 127 sur un octet avec un 0 en MSB,
  - de coder les caractères de 128 à 1023 sur 2 octets (**UTF-8**) avec 110_2 en début du 1er octet et 10_2 comme début de 2ème octet,
  - et au-dessus de 1023 sur 1 ou 2 mots de 16 bits (**UTF-16**),
    Attention : 2 codages sont possibles en :
    - Big Endian (**UTF-16BE**)
    - Little Endian (**UTF-16LE**),
  - et ce jusqu'à 1 mot de 32 bits (4 octets) (**UTF-32**).
- Le codage UTF-8 est devenu le standard de l'Internet, donc de l'informatique en général. Par exemple, l'UTF-8 est le codage par défaut des documents en XML (OpenOffice) donc en XHTML.

JFA - 53

Code UTF 8, UTF 16, UTF 32

- Ces caractéristiques sont intéressantes à plusieurs titres :
  - Un texte écrit en US-ASCII reste inchangé. Le passage à l'UTF-8 est invisible pour les Américains. En revanche un texte écrit en ISO-8859-1 est modifié pour les lettres accentuées représentées en UTF-8 sur 2 caractères. La taille d'un texte avec des accents est donc seulement augmentée de quelques % .
  - Le jeu de caractères est "sans-état" [stateless], la signification d'un caractère ne dépendant pas d'un marqueur dans une séquence. Un caractère manquant dans la séquence détruit la représentation du caractère, mais pas plus, car on peut se resynchroniser au caractère suivant. C'est donc un code très robuste.
- Ces caractéristiques en font donc le codage à préférer pour toutes les applications

IUT GRAND OUEST NORMANDIE iNFO R 1.03

JFA - 54

Exemple de Codage UTF 8 sur 2 octets

- Méthode pour coder un caractère Unicode en UTF-8 :
  - Prendre le point de code du caractère à convertir :
    - Ex : U+0416 : Ж : CYRILLIC CAPITAL LETTER ZHE
  - Convertir le code du plan (0416) en Binaire sur 12 bits :
    - 0100 0001 0110
  - Retirer le MSB
    - 100 0001 0110
  - Mettre 110, puis les 5 bits de gauche pour le 1er octet :
    - 110 10000
  - Mettre 10, puis les 6 bits restants pour le 2ème octet :
    - 10010110
  - Recoder en hexadécimal :
    - 11010000 10010110 => D0 96
- Donc :

Unicode : U+0416 : Ж : CYRILLIC CAPITAL LETTER ZHE => UTF-8 : 0xD0 0x96

IUT GRAND OUEST NORMANDIE iNFO R 1.03

JFA - 55

Synthèse :

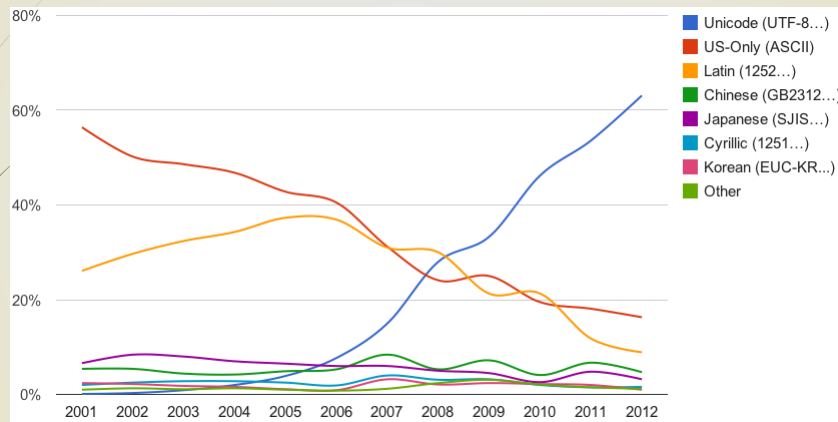
- ***Exemples de différents codages :***
  - *Texte :*
    Ça αβ豈Ⅴ。米
  - *Numéro du caractère en Unicode (Hexa)*
    10 C7 61 20 3B1 3B2 F900 2123 2D30 2D65
  - *UTF-8 : (Hexa)*
    10 C3 87 61 20 CE B1 CE B2 EF A4 80 E2 84 A3 E2 B4 B0 E2 B5 A5
  - *UTF-16 : (Hexa)*
    0010 00C7 0061 0020 03B1 03B2 F900 2123 2D30 2D65

http://hapax.qc.ca/conversion.fr.html

JFA - 56

Comparaisons des codages :

- ***Comparaison des pages Web avec différents encodages :***

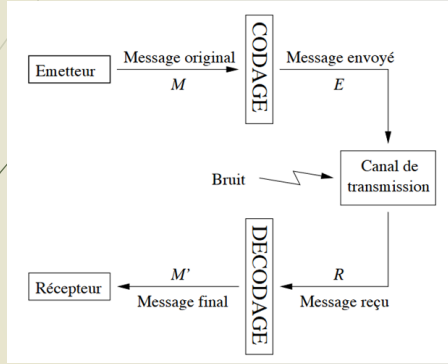


http://i1-news.softpedia-static.com/images/news2/Unicode-Now-Used-by-More-than-60-Percent-of-the-Web-2.png

JFA - 57

Transmissions de données

- Lors de transferts, les données peuvent être altérées par des parasites, des problèmes de contacts, de tensions trop faible, … . Il a fallu trouver une méthode pour vérifier si les données avaient été altérées lors de cet échange.



- Le message à transmettre M est d'abord codé en un message E. Le codage introduit un supplément d'information (redondances).
- Le message E est envoyé sur le canal de transmission.
- A cause de l'imperfection du canal, le message reçu R n'est pas forcément identique au message envoyé E : il peut contenir des erreurs.
- Le message reçu R est ensuite décodé : des erreurs éventuelles peuvent être détectées et corrigées.
- Le message décodé M" est identique au message d'origine M si il n'y a pas eu d'erreurs dans la transmission.

- Transmissions inter-ordinateurs (liaisons Ethernet, wifi, internet, ...) Transmissions intra-ordinateurs (disque dur, mémoire vive, processeur, …) Transmissions hertziennes (radio, téléphones portables, satellites, sondes spatiales, ...), mais aussi la transmission de données via des supports : disquettes, CD, DVD, lettres, livres, ...

https://www.math.u-psud.fr/~montcouq/Enseignements/Codage/generalites.pdf

JFA - 58

Codes détecteurs d'erreurs

- Les codes **détecteurs** d'erreurs permettent la détection d'une erreur au moins de transmission :

  - Voici différents types de codes correcteurs :
    - Bit de Parité
      - Si lors de la transmission une erreur survient, la parité du message reçu est incorrecte et le récepteur décèle l'existence d'une erreur sans pour autant être capable de la localiser. Il demande alors une nouvelle émission du message.
      - Si lors de la transmission deux erreurs surviennent, la parité du message reçu est correcte, les erreurs ne sont pas décelées...

    - CRC (Cycle Redundancy Check)

      Le code CRC permet de vérifier :

      - 100% des erreurs de 1 ou 2 bits,
      - 100% des erreurs impaires,
      - 100% des paquets de 16 erreurs ou moins
      - 99.99% des erreurs de longueur égale ou supérieure à 17,

      Si le choix des polynômes générateurs est bien choisi !

JFA - 59

Code avec bit de parité

- C'est un des codes détecteurs d'erreurs les plus simples, et un des plus utilisés. L'intérêt particulier des contrôles de parité est de vérifier qu'aucune erreur ne se produit lors d'un transfert de données. Il y a 2 styles de parité :
  - La parité paire,
  - La parité impaire.
- On rajoute un bit de parité à la donnée.
- Le nombre de 1 dans la donnée, avec le bit de parité doit être :
  - Paire pour une parité Paire,
  - Impaire pour une parité Impaire.

Il faut donc mettre le bit de parité en conséquence !

Ex : 1011010
⇒ 01011010 en parité paire
⇒ 11011010 en parité impaire

Si on a des erreurs de transmission (en parité paire) :
Reçu avec 1 erreur :
01011000 le nombre de 1 est impair donc erreur !
Reçu avec 2 erreurs :
01011100 le nombre de 1 est pair donc pas d'erreur !
Reçu avec 3 erreurs :
01010100 le nombre de 1 est impair donc erreur !

On ne peut détecter qu'un nombre impair d'erreurs de bits !

iNFO IUT GRAND OUEST NORMANDIE R 1.03

JFA - 60

Exemples de Code avec bit de parité

- ***Exemple : En ASCII sur 7 bits, on peut utiliser le 8ème bit comme bit de parité :***

En écrivant JFAnne en ASCII nous obtenons :

J		F		A		n		n		e	
4	A	4	6	4	1	6	E	6	E	6	5
100	1010	100	0110	100	0001	110	1110	110	1110	110	0101

Pour une parité paire, on a le résultat est le suivant :

J		F		A		n		n		e	
4	A	4	6	4	1	6	E	6	E	6	5
1100	1010	1100	0110	0100	0001	1110	1110	1110	1110	0110	0101

Pour une parité impaire, on a le résultat est le suivant :

J		F		A		n		n		e	
4	A	4	6	4	1	6	E	6	E	6	5
0100	1010	0100	0110	1100	0001	0110	1110	0110	1110	1110	0101

iNFO IUT GRAND OUEST NORMANDIE R 1.03

JFA - 61

Code CRC : définitions

Le code CRC (Code de redondance cyclique) fait partie des codes polynomiaux, c'est-à-dire qu'il utilise l'arithmétique des polynômes.

- On utilise la représentation polynomiale d'un nombre binaire :
  Soit un mot binaire (b_{n-1} b_{n-2} ... b_1 b_0) de longueur n. On représente ce nombre par un polynôme P(X) de variable X et de degré (n-1) dont les coefficients binaires sont :

$$P(x) = b_0 + b_1 X + b_2 X^2 \dots b_{n-1} X^{n-1}$$

  Exemple : $10111_2 \Leftrightarrow 1*X^4+0*X^3+1*X^2+1*X^1+1*X^0$

- Pour avoir une bonne capacité de détection, des codes ont été étudiés et normalisés :

  - ***Polynômes normalisés :***
    - ***CRC-8*** $G(x) = X^8 + X^2 + X + 1$
    - ***CRC-10*** $G(x) = X^{10} + X^9 + X^5 + X^4 + X + 1$
    - ***CRC-12*** $G(x) = X^{12} + X^{11} + X^3 + X^2 + X + 1$
    - ***CRC-16*** $G(x) = X^{16} + X^{15} + X^2 + 1$
    - ***CRC-CCITT*** $G(x) = X^{16} + X^{12} + X^5 + 1$
    - ***CRC-32*** $G(x) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{26} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

JFA - 62

Code CRC : Polynômes

- Pour effectuer le calcul du code CRC, nous allons avoir besoin de plusieurs polynômes :
  - M(x) : polynôme du message binaire de m bits à transmettre (degré m-1),
  - M'(x) : polynôme du message binaire de m bits reçus (degré m-1),
  - G(x) : polynôme générateur de g bits (degré g-1),
  - R(x) : polynôme de contrôle de r bits (degré r >= g),
  - T(x) : polynôme du message binaire transmis avec le CRC (degré m+g-1),
  - T'(x) : polynôme du message binaire reçu avec le CRC (degré m+g-1).

JFA - 63

Code CRC : Méthode de calcul

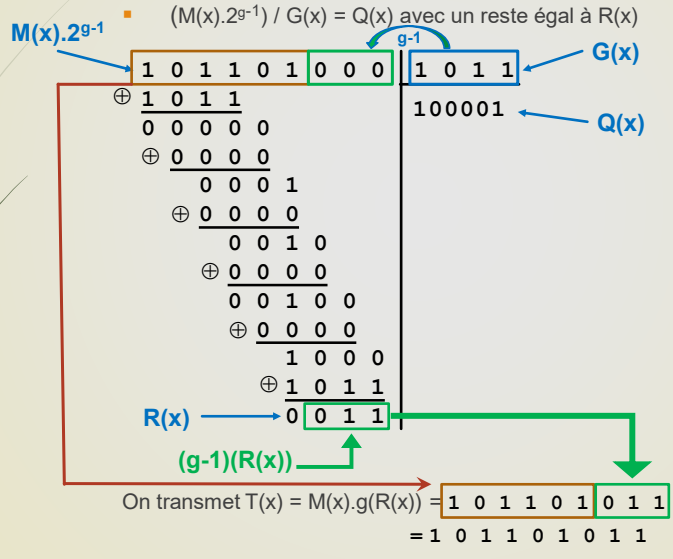
- ***Méthode :***
  - Choisir avec minutie le polynôme générateur G(X) (g chiffres) car c'est lui qui définit les caractéristiques du code.
  - En fin du message de M(x) ajouter g-1 zéros : $M(x)*2^{(g-1)}$
  - Effectuer la division euclidienne (Ou exclusif) de $M(x)*2^{(g-1)}$/ G(x),
  - On obtient Q(x) le quotient, et R(x) le reste.
  - Ajouter alors g bits de R(X) à la fin de M(x), comme bits de contrôle.
  - On obtient alors notre message T(x),
  - Après transmission du message, on obtient T'(x) (qui est identique à T(x), si il n'y a pas eu d'erreurs de transmission),
  - Pour vérifier le message, effectuer la division euclidienne de T'(x) / G(x),
  - On obtient Q'(x) le quotient, et R'(x) le reste.
  - Seul le reste R'(x) nous intéresse :
    - Si R'(x) = 0 alors le message est identique à celui qui a été émis,
    - Si R'(x) ≠ 0 alors le message contient une ou des erreurs, il faut alors demander la réémission du message.
  - Pour obtenir M'(x), il faut enlever g-1 bits à droite de T'(x)

iNFO IUT GRAND OUEST NORMANDIE R 1.03

JFA - 64

Code CRC : Émission

- **Pour l'émission**, on utilise l'équation :
  - $(M(x).2^{g-1}) / G(x) = Q(x)$ avec un reste égal à R(x)



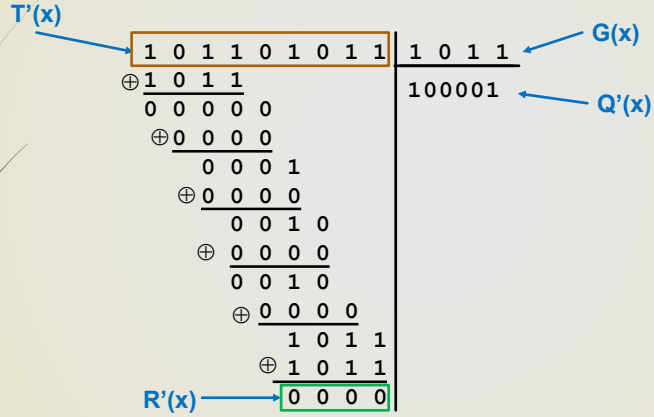
- $M(x) = X^5 + X^3 + X^2 + 1$
- $G(x) = X^3 + X + 1$
- $Q(x) = X^5 + 1$
- $R(x) = X + 1$

iNFO IUT GRAND OUEST NORMANDIE R 1.03

JFA - 65

Code CRC : Réception

- **À la réception**, pour vérifier la bonne transmission, on utilise l'équation :
  - T'(x) / G(x) = Q'(x) avec un reste égal à R'(x)

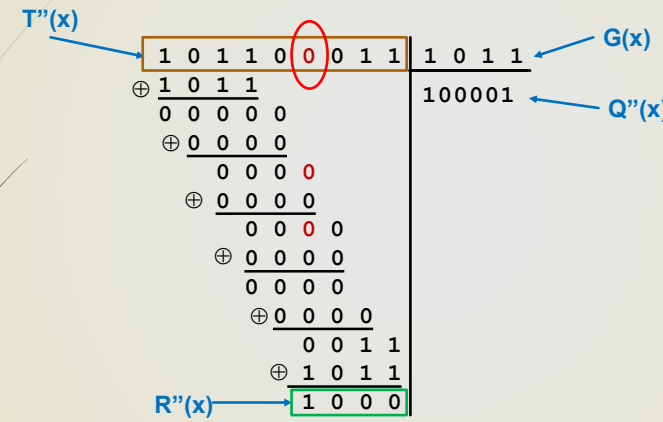


- Si la transmission s'est bien déroulée : R'(x) = 0

JFA - 66

Code CRC : Réception

- ***Si on a eu une erreur de transmission :***



- R"(x) ≠ 0 : la transmission ne s'est pas bien déroulée : il y a au moins une erreur de transmission, il faut donc demander la réémission du message.

JFA - 67

Codes correcteurs d'erreurs

- Un **code correcteur** est une technique de codage basée sur la redondance. Elle est destinée à corriger les erreurs de transmission d'une information (plus souvent appelée message) sur une voie de communication peu fiable.

Il a été développé différentes familles de codes :

- ***Les codes en bloc*** (linéaires, cycliques ou non) : le codage/décodage d'un bloc dépend uniquement des informations de ce bloc.
- Les codes en treillis (convolutifs, récursifs ou non) : le codage/décodage d'un bloc dépend des informations d'autres blocs (généralement de blocs précédemment transmis).
  - Un code convolutif s'applique sur une suite infinie de symboles et produit une suite infinie.

- Voici différents types de codes correcteur d'erreurs :
  - Code de Hamming
  - Code de Golay
  - Code de Reed-Müller (NASA Photos de MARS)
  - Code de Goppa (cryptographie)
  - Code de Xing
  - Code de Reed-Solomon (CD, DVD, Satellite DVB, ADSL, ADSL2, ADSL 2+)

JFA - 68

Exemple de code correcteur d'erreur

Un exemple de code correcteur est obtenu en ajoutant à une suite de trois nombres (l'information à transmettre) 2 autres nombres (le code de contrôle de l'information). L'ensemble des 5 nombres permet alors de détecter et de corriger une erreur qui se serait produite lors de la transmission.

Soit donc un bloc de 3 nombres que l'on souhaite transmettre : 02 09 12
Ajoutons 2 nombres de contrôle de l'information.
Le premier est la somme des 3 nombres : 02 + 09 + 12 = 23
Le second est la somme pondérée des 3 nombres, chacun est multiplié par son rang : 02×1 + 09×2 + 12×3 = 56
À la sortie du codeur, le bloc à transmettre est : 02 09 12 23 56

À la suite d'une perturbation, le récepteur reçoit : 02 **13** 12 23 56
À partir des données reçues, le décodeur calcule :
Sa somme simple : 02 + **13** + 12 = **27**
Sa somme pondérée : 02×1 + **13**×2 + 12×3 = **64**
La différence entre la somme simple calculée (**27**) et celle reçue (23) indique la valeur de l'erreur : 4 (**27**-23 = 4)
La différence entre la somme pondérée calculée (**64**) et celle reçue (56), elle-même divisée par la valeur de l'erreur indique la position où l'erreur se trouve : 2 ((**64**-56) / 4 = 2).
Il faut donc retirer 4 au nombre du rang 2.
Le bloc original est donc 02 (**13**-4=09) 12 23 56
Lors d'une transmission sans perturbation, les différences des sommes simples et des sommes pondérées sont nulles.

https://fr.wikipedia.org/wiki/Code_correcteur

JFA - 69

Codes en blocs

- Dans le cas général, les messages à transmettre n'ont pas de longueur fixe. En revanche, il est plus simple de développer un code correcteur pour des messages d'une longueur fixe.
- La solution utilisée consiste donc à segmenter le message. Dans un premier temps est traité le cas d'un message de longueur fixe. Pour le cas général, une solution simple consiste à concaténer une suite de blocs. La méthode la plus répandue, car la plus efficace est celle du code convolutif.
- On préfère généralement le codage par bloc dans les applications téléinformatiques classiques :
  le codage/décodage est plus simple et plus rapide

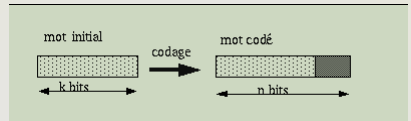
Un code en bloc est un code correcteur traitant des messages de longueur fixe.

IUT GRAND OUEST NORMANDIE iNFO R 1.03

JFA - 70

Codes en blocs : définition

- Un code (k, n) code k bits d'information (longueur initiale des mots) en un bloc de n bits (longueur du code). Le code introduit une redondance puisque n >= k.



- Le poids de Hamming d'un mot w(N) est le nombre de bits à 1 qu'il contient.
- La distance de Hamming d(N1, N2) entre deux mots de même longueur est définie par le nombre de bits qui diffèrent entre ces deux mots. On l'obtient par le poids de Hamming du ou exclusif binaire (⊕) des 2 mots.

$$d(N1,N2) = w(N1 \oplus N2)$$

La distance de Hamming ***d'un code*** est la distance minimum entre tous les mots du code : $d_C = \min\{d(Ni,Nj) | i!=j\}$

- Un code C ayant une distance minimale de d est noté [n, k, d]. Cette notation signifie que l'on code des mots de k symboles en mots de code de n symboles étant tous distants les uns des autres de d symboles au minimum,
- ***Exemple :***
  Poids_de_Hamming de (01001100) = 3
  Distance_de_Hamming de (01001100, 01010101) = (01001100 ⊕ 01010101)

```
  01001100
⊕ 01010101
  00011001   => 3
```

IUT GRAND OUEST NORMANDIE iNFO R 1.03

JFA - 71

Codes en blocs : définition

- La capacité de détection (de correction) d'un code est définie par les erreurs qu'il est capable de détecter (corriger). Une erreur simple (resp. double, ou d'ordre p) affecte une seule (resp. 2, ou p) position(s) binaire(s) d'un mot. Pour qu'un code ait une capacité de détection (resp. correction) des erreurs d'ordre e, il faut que sa distance de Hamming soit supérieure à 1+e (resp. 1 + 2e).

Exemple :

Si on a une distance de Hamming = 3
=> capacité de ***détection*** =< 2,
capacité de ***correction*** =< 1.

Plus la distance de Hamming est grande, meilleur est le code !

IUT GRAND OUEST NORMANDIE
INFO
R 1.03