TD 1 : Représentation de l'information

Codage

Useful to travel through the Matrix.

Dans ce qui suit nous notons x_i le chiffre de rang i en base β.

Étant donné un ensemble B de mots de n bits et un ensemble I d'informations différentes, on appelle codage de I sur n bits toute fonction de B dans I. Le codage est *total* si la fonction est surjective.

1. Quel est le nombre minimum de bits nécessaire pour un codage total de n informations ?
2. Donnez les principes du codage des entiers positifs en numération simple de position en base β. Combien d'entiers naturels positifs différents peut-on exprimer avec n positions (chiffres) en base β ?
3. Convertissez le nombre $(101000101)_2$ en base 5, 8 et 16. Quelle est la conclusion ?
4. Faites quelques additions d'entiers positifs sur 8 bits pour vous faire la main.

Les entiers

Don't be always negative !

Les négatifs - premières tentatives

1. Proposez une manière naïve de coder les entiers relatifs. Quel est l'intervalle des entiers signés représentés par un mot de n bits ?
2. Faites les additions suivantes $16 + 64$ et $42 + (-17)$. Et si on doit faire $17 + (-42)$? Quels sont les problèmes ?
3. Trouvez une autre manière de coder les entiers relatifs qui soit plus pratique. Astuce : pensez au complément à la base.
4. Donnez la formule de calcul de N (en décimal) à partir de cette représentation ?
5. Quel est l'intervalle des entiers codés en complément à 1, représentés par un mot de n bits ? Quel est le problème ?

Les négatifs - c'est bon !

1. Dans la représentation en *complément à 2*, le nombre N est obtenu en rajoutant $+1$ à sa représentation en complément à 1. Donnez la formule de calcul de N (en décimal) à partir de sa représentation en complément à 2 ?
2. Quels sont le plus grand et le plus petit nombre représentables sur 6 bits en complément à 2 ? Quel est l'intervalle des entiers signés représentés par n bits ?
3. Faites quelques additions et quelques soustractions d'entiers relatifs codés en complément à 2 sur 8 bits.
4. Posez quelques multiplications de nombres de 4 bits en binaire.

Les réels

Shall I represent them all ?

La norme IEEE 754 est un standard pour la représentation des réels en binaire. Elle comprend 1 bit de signe, e bits d'exposant et m bits de mantisse. L'exposant est un entier relatif codé – non pas avec un complément à 2 – mais avec un biais permettant de ne manipuler que des entiers non signés. Ce biais est $2^{e-1}-1$.

Soient les nombres binaires v_s, v_e, v_m représentant respectivement le signe, l'exposant et la mantisse. La valeur décimale d'un nombre flottant est définit par $N = (-1)^{v_s} * 2^{v_e - bias} * (1 + v_m)$ où $0 \leq v_m < 1$. En effet, $v_m = \sum_{i=0}^{m-1} x_i 2^{i-m}$ avec x_i bit de position i (bit de poids fort à gauche).

Nous considèrons ici le format simple précision comprenant 1 bit de signe, 8 bits d'exposant et 23 bits de mantisse.

- Un nombre est dit normalisé lorsque l'exposant avec biais est non nul et différent de $2^e - 1$.
- Un nombre est dit dénormalisé lorsque l'exposant avec biais est nul.
- Les zeros sont représentés par une mantisse et un exposant avec biais nuls.
- Les infinis sont représentés par un exposant de $2^e - 1$ et une mantisse nulle.
- *Not A Number* (NaN) est réprésenté par un exposant de $2^e - 1$ et une mantisse non-nulle.

Attention, pour un nombre dénormalisé, $N = (-1)^{v_s} * 2^{v_e - bias} * (0 + v_m)$

1. Quels nombres réels représentent les mots de 32 bits suivants : 0x41380000 et 0xBF800001 ?
2. Donnez la représentation de 1, de 3.625, de 2^{20} et de 0.
3. Donnez la valeur du plus grand positif et de son prédécesseur, indiquez leur écart.
4. Donnez la valeur du plus petit positif normalisé et du plus petit positif dénormalisé non-nul.
5. Quel est le résultat de l'addition $2^{20} + 2^{-10}$ dans un additionneur flottant ?

Détection et correction d'erreurs

Chuck Norris doesn't make errors !

Généralités

Étant donnée une source émettant une information codée sur n bits reçue par un récepteur, on définit les termes suivants :

- un codage est auto-vérificateur s'il permet de déceler une erreur de transmission, et
- un codage est auto-correcteur s'il permet de reconstituer l'information.

1. Trouvez des exemples simples de codages auto-vérificateurs ou auto-correcteurs.

   Le controle de parité utilise $n-1$ bits pour l'information et un bit de controle dit bit de parité. Celui-ci est positionné de telle sorte que le nombre total de bits à 1 de la configuration soit pair (parité paire) ou impair (parité impaire). On retrouve cette méthode dans le protocole de communication du lien série (UART/RS-232) par exemple.
2. Quelles sont les possibilités et les limites d'un tel codage ?

(Bonus) Compression

tar -cfz useless.tar.gz /dev/null

Dans le premier exercice de la partie 1, nous avons vu une borne inférieure théorique sur le nombre de bits nécessaires pour coder n informations (par exemple les caractères ASCII). Cependant, il existe des techniques d'encodage qui sont plus efficaces en pratique.

Un code préfixe est un code où aucun mot ne préfixe un autre mot de ce code. L'intérêt d'un tel code est qu'il possède un décodage unique sans nécessiter de séparateur. A noter que les codes de taille fixe sont tous des codes préfixes.

1. Trouver un code préfixe de longueur variable.

   Un algorithme de compression représente une information sur moins de bits que sa représentation d'origine. Il peut être avec pertes de données comme le format jpeg, ou sans.

   Supposons un fichier contenant une suite de caractères ASCII.

2. En utilisant ce code préfixe, proposez un algorithme de compression sans pertes de données.