FST Mulhouse.

Licence 1 Maths-Info

Mathématiques : ARITHMETIQUE-ALGEBRE

Elisabeth REMM

Chapitre 3

Systèmes de numération. Codages

Table des matières

1. Systèmes de numération décimal et binaire 1
1.1. Le système décimal 1
1.2. Le système binaire 2
2. Système de numération en base b 3
2.1. Ecriture d'un nombre en base b 3
2.2. Le système octal 4
2.3. Cas où la base b est plus grande que 10 : un exemple, le système bibibinaire 5
2.4. Retour au binaire. Code correcteur 6
3. Arithmétique en binaire 6
3.1. L'addition 6
3.2. La multiplication 7
4. L'exponentiation modulaire 7
4.1. Calculer le reste modulo n de a^p, avec n grand. 7
4.2. Application : le système de codage R.S.A 8

1. Systèmes de numération décimal et binaire

1.1. **Le système décimal.** La numération est un moyen de représenter un nombre à l'aide de symboles. Cette manière a varié au cours des temps. Le système en vigueur aujourd'hui, dans la vie courante, est basé sur dix symboles, $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$. On l'appelle donc le système décimal. Dans les paragraphes suivants, nous présenterons des systèmes utilisés de nos jours mais dans des domaines orientés vers l'informatique, le système binaire qui repose sur 2 symboles, le 0 et le 1, et le système bibinaire qui repose sur seize symboles (il faudra donc leur trouver une calligraphie).

Considérons donc un nombre entier N positif. Il s'écrit, en écriture décimale

$$N = a_p 10^p + a_{p-1} 10^{p-1} + \cdots + a_1 10 + a_0$$

a_0 étant le chiffre des unités, a_1 celui des dizaines, etc (voir le cours de CM1). Les "chiffres" $a_0, a_1, \cdots, a_p$ appartient à $\{0, 1, \cdots, 9\}$ et sont parfois appelés les digits (langage informatique). Cette écriture s'étend également aux nombres décimaux (cf cours CM2), l'écriture sera alors donnée de la façon suivante

$$N = \sum_{i=-m}^{i=p} a_i 10^i$$

où m et p sont des entiers positifs ou nuls.

1.2. **Le système binaire.** Dans le système binaire, qui est le système fondamental en informatique, seuls deux symboles sont utilisés communément appelés en langage informatique bits. Ces symboles sont notés, car aucune confusion n'est possible avec les symboles qui ont la même calligraphie dans le système décimal, 0 et 1. L'écriture d'un nombre dans ce système est basé sur la relation suivante : tout nombre entier s'écrit de manière unique sous la forme

$$N = u_p 2^p + u_{p-1} 2^{p-1} + \cdots + u_1 2 + u_0$$

où les nombres entiers u_i appartiennent à $\{0, 1\}$. Si dans l'écriture décimale on convient d'écrire un nombre de droite à gauche dans l'ordre unité, dizaine , centaine etc, il en sera de même en écriture binaire. On écrira le nombre N sous la forme

$$\overline{u_p u_{p-1} \cdots u_1 u_0}^{(2)}.$$

Si aucune confusion n'est possible, s'il est précisé auparavant que le système de numération utilisé est le binaire, alors dans ce cas, nous simplifierons l'écriture en écrivant

$$u_p u_{p-1} \cdots u_1 u_0.$$

Ainsi la suite des nombres ordonnée dans le système binaire commence ainsi

$$\begin{cases} 0 \\ 1 \\ 10 \\ 11 \\ 100 \\ 101 \\ 110 \\ 111 \\ 1000 \\ 1001 \\ \cdots \end{cases}$$

Chacun de ces nombres est représenté dans le système décimal par $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \cdots$.

Comment passer du système décimal au système binaire ? Cette opération s'appelle le codage. Une méthode algorithmique efficace, basée sur l'écriture $N = u_p 2^p + u_{p-1} 2^{p-1} + \cdots +$

$u_1 2 + u_0$ du nombre donné et utilisant la division euclidienne par 2. On écrit la suite suivante de division euclidienne

$$\begin{aligned} N &= 2N_1 + \mathbf{r_0} \\ N_1 &= 2N_2 + \mathbf{r_1} \\ N_2 &= 2N_3 + \mathbf{r_2} \\ &\cdots \\ N_{p-1} &= 2N_p + \mathbf{r_{p-1}} \\ N_p &= 2 \cdot 0 + \mathbf{r_p} \end{aligned}$$

L'écriture du nombre N en base 2 (ou écriture binaire) est alors

$$\mathbf{r_p r_{p-1} \cdots r_2 r_1 r_0}.$$

(On fera très attention au fait que le premier reste trouvé est celui des "unités", le deuxième celui des "deuxaines",...) Il est clair que cet algorithme s'arrête lorsque l'un des quotients est plus petit que 2. Un exercice intéressant est de programmer cet algorithme sur PYTHON.

Exemple. On veut coder en binaire le nombre 11. On a

$$\begin{aligned} 11 &= 2 \times 5 + \mathbf{1} \\ 5 &= 2 \times 2 + \mathbf{1} \\ 2 &= 2 \times 1 + \mathbf{0} \\ 1 &= 2 \times 0 + \mathbf{1} \end{aligned}$$

et donc 11 s'écrit en binaire 1011.

Comment passer du système binaire au système décimal ? Cette opération s'appelle le décodage. Elle est particulièrement simple. Il suffit d'utiliser la décomposition en bits. Si on considère un nombre $N = (u_p i_{p-1} \cdots u_1 u_0)_{(2)}$ écrit en binaire, il s'écrit en décimal $N = u_p 2^p + u_{p-1} 2^{p-1} + \cdots + u_1 2 + u_0$. Par exemple, considérons $N = \overline{1011}^{(2)}$, alors

$$N = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 2 + 1 = 11.$$

Exemple.

2. Système de numération en base b

Considérons un entier $b \geq 2$. Cet entier sera appelé la base du système de numération. Pour le système décimal, $b = 10$, pour le système binaire, $b = 2$.

2.1. Ecriture d'un nombre en base b.

Théorème 1. *Tout entier N positif s'écrit de manière unique sous la forme*

$$N = a_p b^p + a_{p-1} b^{p-1} + \cdots + a_1 b + a_0$$

où les a_i sont des entiers vérifiant $0 \leq a_i \leq b - 1$.

Démonstration. L'unicité se prouve assez facilement : supposés que l'on ait deux décompositions

$$N = a_p b^p + a_{p-1} b^{p-1} + \cdots + a_1 b + a_0 = b(a_p b^{p-1} + a_{p-1} b^{p-2} + \cdots + a_1) + a_0.$$

Ainsi a_0 qui vérifie $0 \leq a_0 \leq b - 1$ est le reste de a division euclidienne de N par b. Il est donc unique. On considère ensuite l'entier $(N - a_0)b^{-1}$ et on réitère le raisonnement que nous avons fait pour a_0. Reste à prouver l'existence de cette décomposition. Cette preuve nous donne en fait la procédure du codage d'un nombre en base b. Nous la développons ci-dessous.

Comment passer du système décimal au système en base b ? Cette opération s'appelle toujours le codage. La méthode algorithmique efficace, identique à celle présentée pour $b = 2$ est basée sur l'écriture $N = u_p b^p + u_{p-1} b^{p-1} + \cdots + u_1 b + u_0$ du nombre donné et utilisant la division euclidienne par b. On écrit la suite suivante de division euclidienne

$$\begin{aligned} N &= bN_1 + \mathbf{r_0} \\ N_1 &= bN_2 + \mathbf{r_1} \\ N_2 &= bN_3 + \mathbf{r_2} \\ &\cdots \\ N_{p-1} &= bN_p + \mathbf{r_{p-1}} \\ N_p &= b \cdot 0 + \mathbf{r_p} \end{aligned}$$

L'écriture du nombre N en base b (ou écriture binaire) est alors

$$\overline{r_p r_{p-1} \cdots r_2 r_1 r_0}^{(b)}.$$

Exemple. Ecrire en base 9 le nombre 1237. Bien entendu 1237 est l'écriture décimale, ce que nous conviendrons lorsqu'aucune base n'est précisée. On a

$$\begin{aligned} 1237 &= 9 \times 137 + \mathbf{4} \\ 137 &= 9 \times 15 + \mathbf{2} \\ 15 &= 9 \times 1 + \mathbf{6} \\ 1 &= 9 \times 0 + \mathbf{1} \end{aligned}$$

Ainsi 1237 s'écrit en base 9 :

$$\overline{1624}^{(9)}.$$

Notons que le décodage, c'est-à-dire le passage à l'écriture décimale, se fait, comme en base 2 en utilisant le développement en puissance de b.

2.2. **Le système octal.** Il correspond au système numérique en base 8. On peut passer facilement du système binaire au système octal (en fait on sait passer facilement du système binaire à un système de base 2^k). La règle générale de conversion est la suivante :

(1) En allant de droite à gauche , on scinde l'expression binaire en paquets de 3 chiffres (3 est l'exposant de 2 dans $b = 8 = 2^3$).

(2) Comme en base 2, le nombre 7 s'écrit $\overline{111}^{(2)}$, un paquet de 3 chiffre composés de 0 et de 1 représente un nombre plus petit que 7.

(3) On remplace chacun des groupes, correspondant à l'écriture en base 2 d'un nombre plus petit ou égal à 7 par ce nombre.

Exemple. Soit le nombre qui s'écrit en base 2

$$\overline{100110110101}^{(2)}.$$

Cherchons son expression en base 8 en suivant les étapes suivantes

$$\begin{aligned} &\overline{100110110101}^{(2)} \\ &100; 110; 110; 101 \\ &4; 6; 6; 5 \end{aligned}$$

l'écriture cherchée est donc

$$\overline{4665}^{(8)}.$$

2.3. **Cas où la base b est plus grande que 10 : un exemple, le système bibibinaire.** Lorsque $b \geq 10$, l'écriture d'un nombre sous la forme

$$N = a_p b^p + a_{p-1} b^{p-1} + \cdots + a_1 b + a_0$$

fait apparaître des composantes a_i qui sont inférieures ou égales à $b-1$. Mais nous ne pouvons plus utiliser comme composantes les nombres $10, 11, 12, \cdots, b-1$ car ils représentent déjà des codes en système décimal. Il est alors d'usage d'utiliser des lettres majuscules, $A, B, C, \cdots,$. Comme nous pouvons supposer que b n'est pas trop grand, l'alphabet devrait suffire pour satisfaire ce besoin.

Un exemple intéressant, utilisé en informatique, est le système hexadécimal correspondant à $b = 16$. Le symboles de base sont alors

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.$$

Il est largement utilisé en informatique car, comme nous l'avons vu ci-dessus, il est très facile de passer du système binaire au système hexadécimal. En effet $16 = 2^4$, on découpe alors le nombre écrit en base 2 en paquet de 4 en partant de la droite et on remplace chaque paquet par le nombre correspondant en écriture décimale qui sera nécessairement un nombre inférieur ou égal à 15, en prenant soin de remplacer 10 par A, 11 par B, 12 par C, 13 par D, 14 par E et 15 par F. Par exemple le nombre qui s'écrit en base 2

$$\overline{100110110101}^{(2)}$$

s'écrira, en suivant la règle de conversion

$$\begin{array}{l} \overline{100110110101}^{(2)} \\ 1001; 1011; 0101 \\ 9; 11; 5 \end{array}$$

l'écriture cherchée est donc

$$\overline{9B5}^{(16)}.$$

Un peu d'histoire. Le système hexadécimal a été étudié pour la première fois par le chanteur Boby Lapointe dans sa thèse de Doctorat soutenue à l'Université de Montpellier en 1968. Pour situer un peu ce chanteur, voici le refrain d'une de ses célèbres chanson, intitulée Ta Katie t'a quitté :

Tic-tac tic-tac

Ta Katie t'a quitté

Tic-tac tic-tac

Ta Katie t'a quitté

Tic-tac tic-tac

T'es cocu, qu'attends-tu ?

Cuite-toi, t'es cocu

T'as qu'à, t'as qu'à t' cuiter

Et quitter ton quartier

Ta tactique était toc

Ta tactique était toc

Revenons au système hexadécimal de Boby Lapointe. Il en a décrit toute l'arithmétique et montrer son importance comme outil numérique. Il ne l'appelait pas hexadécimal mais bibibinaire car $16 = 2^{2^2}$. Il avait utilisé 16 symboles pour décrire les composantes d'un nombre à la place de $0, 1, \cdots, F$ utilisés de nos jours. Pour pouvoir lire ces symboles, comme on lit *zéro* pour 0, un pour 1, etc, il utilisait 4 voyelles et 4 consonnes. Plus précisément la prononciations des symboles $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F$ était dans l'ordre

HO, HA, HE, HI, BO, BA, BE, BI, KO, KA, KE, KI, DO, DA, DE, DI.

Par exemple le nombre 2000 se lit en bibibinaire $BIDAHO$.

2.4. **Retour au binaire. Code correcteur.** Le but d'un réseau est de transmettre des informations d'un ordinateur à un autre. Passons sur les divers moyens techniques pour assurer cette transmission, et restons sur l'échange de données qui est basée sur un codage de l'information. Supposons que ce codage soit supporté par un langage binaire. Une étape importante et fondamentale au cours de ces échanges est celle du contrôle des erreurs : est-ce qu'une information codée émise est correctement interprétée par le destinataire. Les erreurs doivent impérativement être détectées dès leur apparition. Par exemple le mot envoyé est codé en binaire par 10101101. Le destinataire reçoit ce code est doit s'assurer que c'est bien le code envoyé par l'expéditeur.

Première méthode : Introduction d'un bit de parité. On rajoute au mot binaire envoyé un bit supplémentaire On choisit par exemple le 1 si la somme des 1 du mots codés est impaire et 0 sinon. A la réception, on vérifie en lisant le bit de contrôle, situé en fin de mot, que la parité est bien respectée. Il est clair que ce test peut être lui aussi erroné mais il est simple à mettre en place et renseigne si le test est faux (un peu comme la règle de 3 pour la multiplication). Il ne renseigne pas sur la position dans le mot de l'erreur.

Deuxième méthode. Introduction d'un mot de parité. Cette méthode permet de localiser presque à coup sûr l'erreur. Supposons par exemple que l'on veuille tester un nombre binaire comprenant 4 bits, c'est-à-dire $\overline{a_1a_2a_3a_4}^{(2)}$. On rajoutera à ce code 3 bits supplémentaires de parité, le premier pour le code $\overline{a_1a_2a_3}^{(2)}$, le deuxième pour $\overline{a_1a_2a_4}^{(2)}$ le troisième pour $\overline{a_1a_3a_4}^{(2)}$. Cela est suffisantt pour localiser une erreur chez le destinataire.

3. Arithmétique en binaire

3.1. **L'addition.** La technique d'addition est la même que celle en base 10, on reporte les retenues sur la colonne de gauche immédiate, ou si cette retenue a plusieurs bits, sur les colonnes de gauche en partant de celle qui jouxte. Par exemple

$$\begin{array}{cccccc} & 1 & 0 & 1 & 1 \\ + & 1 & 1 & 0 & 1 \\ \hline 1 & 1 & 0 & 0 & 0 \end{array}$$

3.2. **La multiplication.** Pour cette opération aussi la technique est le même, avec un gros avantage, les seules tables de multiplication à connaître sont celles de 0 et de 1 (vieux traumatisme du CE1-CE2). Par exemple

$$\begin{array}{cccccccc} & & & 1 & 0 & 1 & 0 \\ & & \times & 1 & 1 & 0 & 0 \\ \hline & & & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 & \\ & 1 & 0 & 1 & 0 & & \\ 1 & 0 & 1 & 0 & & & \\ \hline 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{array}$$

4. L'exponentiation modulaire

Etant donné un entier n "grand", on souhaiterait calculer le reste modulo n d'une puissance a^p d'un entier donné. Nous donnerons une application de ce problème concernant le cryptage.

4.1. **Calculer le reste modulo n de a^p, avec n grand.** L'exponentiation modulaire (ou puissance modulo) est le résultat du calcul a^p modulo n. Elle est utilisée en informatique et en cryptographie. Par exemple, quel est le reste modulo 56 de 12^{34} ?.

Première méthode : la méthode naïve. On calcule le reste de a modulo n. Soit r_1 ce reste. On calcule ensuite le reste de $a \cdot r_1$ modulo n. Soit r_2 ce reste. Puis le reste de $a \cdot r_2$. On continue ainsi b fois, et le dernier reste obtenu sera le résultat cherché. Dans notre exemple, nous aurons

$$12 = 12\ (56), 144 = 32\ (56), 12 \times 32 = 384 = 48\ (56), 48 \times 12 = 576 = 16\ (56) = \cdots$$

Ce calcul est long. Il nécessite $b = 34$ opérations.

Deuxième méthode. On décompose l'exposant b en somme de puissance de 2. Rappelons ces puissances :

$2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16, 2^5 = 32, 2^6 = 64, 2^7 = 128, 2^8 = 256, 2^9 = 512, 2^{10} = 1024, \cdots$

On aura donc

$$b = 2^{k_1} + 2^{k_2} + \cdots + 2^{k_r}$$

avec $k_1 > k_2 \cdots > k_r \geq 0$. On en déduit

$$a^b = a^{2^{k_1}+2^{k_2}+\cdots+2^{k_r}} = a^{2^{k_1}} \cdot a^{2^{k_2}} \cdots a^{2^{k_r}}.$$

Dans notre exemple, la décomposition de $b = 34$ en somme de puissance de 2 est

$$34 = 32 + 2 = 2^5 + 2^1.$$

Ainsi

$$12^{34} = 12^{2^5} \cdot 12^2.$$

Revenons au cas général. La décomposition de b en somme de puissance de 2 ramène notre calcul à trouver les restes modulo n de a élevé à une puissance de 2. On va donc calculer

(1) le reste r_1 modulo n de a
(2) le reste r_2 modulo n de $r_1 \times a$
(3) le reste r_3 modulo n de $r_2 \times a$
(4) $\cdots$
(5) enfin le reste r_{k_1} modulo n de $r_{k_1-1} \times a$.

Notons que certaine étapes peuvent être éviter suivant les valeurs des exposants.

Reprenons notre exemple

(1) $a = 12 \equiv 12$ modulo 56
(2) $r_1 \times 12 = 12^2 \equiv 32$ modulo 56 ceci correspondant à a^2
(3) $r_2 \times 12 = 384 \equiv 48$ modulo 56 ceci correspondant à a^3
(4) $r_3 \times 12 = 576 \equiv 16$ modulo 56 ceci correspondant à a^4
(5) $r_4 \times a = 192 \equiv 24$ modulo 56
(6) $r_5 = 16$.

Ainsi

$$12^{34} = 12^{2^5} \cdot 12^2 = 24 \times 32 \equiv 40 \text{ modulo } 56.$$

Ici on aurait pu se limiter aux seules opérations

(1) $a = 12 \equiv 12$ modulo 56
(2) $r_1 \times 12 = 12^2 \equiv 32$ modulo 56 ceci correspondant à a^2
(3) $r_2^2 = 1024 \equiv 16$ modulo 56 ceci correspondant à a^4
(4) $16 \times 12 = 192 \equiv 24$ modulo 56
(5) $r_5 = 16$.

4.2. **Application : le système de codage R.S.A.** Le système R.S.A est un système de codage basé sur deux clés publiques (c'est-à-dire connu par l'émetteur du message codé) et ces clés publiques appartient au receveur. Par exemple, l'émetteur du message voulant envoyer son message à un recteur précis trouvera ces clés dans un site public . Comment je constitue mes clés publiques personnelles :

(1) Je me donne deux entiers p et q premiers entre eux grands (on en connaît beaucoup) mais ces deux nombres je les garde secrets.
(2) Je fais le produits $n = pq$ et le produit $m = (p-1)(q-1)$.
(3) Je considère un entier e premier avec m

Les clés publiques sont n et e.

Considérons un message M que nous devons codé (M est un message chiffré, par exemple un code PIN). Nous voulons l'envoyer à un destinataire. Je vais rechercher sur un annuaire public les clés personnelles de ce destinataire. Soient n et e ces clés.

On considère maintenant le code chiffré

$$C \equiv M^e \bmod n.$$

A la reception, il faudra décodé C mais le receveur connait n, p, q, m, e. Soit d tel que

$$ed \equiv 1 \bmod m.$$

Alors

$$M \equiv C^d \bmod n.$$

Ce problème de déchiffrement se traite par l'exponentiation modulaire.

Remarque. Pour éviter toute tentative de pirater ce code, au lieu de coder directement m, on découpe M en paquets égaux de chiffres (plus grand que 2) et on code chaque paquets.

Exemple. Nicolas veut envoyer à Paul le message MATHS pour lui rappeler l'importance de ce cours. Il commence à l'écrire numériquement par exemple en utilisant la place des lettres dans l'alphabet. Ce mot devient

$$13\ 1\ 20\ 8\ 19.$$

On découpe ce nombre par paquets de 3 :

$$013\ 120\ 819.$$

Nicolas va chercher dans l'annuaire les clés de Paul. Il trouve $n = 5141$ et $e = 7$. En fait, en secret Paul avait choisit $p = 53, q = 97$ et donc $n = 5141, m = 4992$ et $e = 7$ est premier avec 4992. Nicolas va donc calculer

$$C_1 \equiv 013^7 \bmod 5141,\ C_2 = 120^7 \bmod 5141,\ C_3 = 819^7 \bmod 5141.$$

La méthode de l'exponentiation modulaire nous donne

$$C_1 = 2646,\ C_2 = \ C_3 = .$$

Paul doit décoder. Pour cela il utilise sa clé secrète d qui est donnée par $ed = \equiv 1 \bmod m$ soit comme $e = 7$ et $m = 5140 \times 4992$