AP - Balade dans $\{0;1\}$

1 Système de numération binaire

Définition 1. *Le système binaire, ou base 2, représente les chiffres avec uniquement 0 et 1.*

Exercice 1 : Compter jusqu'à 20 en binaire.

Exercice 2 : Sur des exemples, calculer l'addition de deux nombres en binaire. Idem pour la multiplication et la division.

2 Particularité de l'ensemble $\{0;1\}$

Définition 2. *On définit dans $\{0;1\}$ une addition et une multiplication, dont on donne les tables ci-dessous. Munies de ces deux lois, l'ensemble $\{0;1\}$ est noté $\mathbb{F}_2$: c'est le corps à deux éléments.*

$+$	*0*	*1*
0	*0*	*1*
1	*1*	*0*

$\times$	*0*	*1*
0	*0*	*0*
1	*0*	*1*

Exercice 3 : Soient $a, b \in \mathbb{F}_2$. Montrer que $(a+b)^2 = a^2 + b^2$. Cette relation vous choque-t-elle ?

Exercice 4 : (vraiment dur, à faire à la fin) De la même manière, construire le corps à 3 éléments. Peut-on construire facilement un corps à 4 éléments ?

3 Opérations logiques

Définition 3. *Soient P et Q deux propositions. Alors :*

- *La négation de P s'écrit $\neg P$ (NOT P)*
- *P et Q s'écrit $P \wedge Q$ (P AND Q)*
- *P ou Q s'écrit $P \vee Q$ (P OR Q)*

Exercice 5 : Soit P=$\{$*Je suis allé au cinéma ce week-end*$\}$ et Q=$\{$*Il pleuvait ce week-end*$\}$. Exprimer $\neg P$, $P \wedge Q$ et $P \vee Q$.

3.1 Arithmétique Booléenne

A partir de maintenant, on note 1 si une proposition est vérifiée, et 0 sinon. Lorsqu'ils représentent des propositions vraies ou fausses, 0 et 1 sont appelés des *booléens*.

Exercice 6 : Soient x et y deux booléens. Montrer que $x \wedge y = x \times y = \min(x,y)$; $x \vee y = x + y - x \times y = max(x,y)$ et $\neg x = 1 - x$

Définition 4. *On définit l'opération NAND (ou NOT AND) par :*

$$x \uparrow y = \neg(x \wedge y) \tag{1}$$

Exercice 7 : Montrer que $x \uparrow y = (\neg x) \vee (\neg y)$

4 Classical gates

A classical (deterministic) computer evaluates a function: given n-bits of input it produces m-bits of output that are uniquely determined by the input; that is, it finds the value of

$$f : \{0;1\}^n \to \{0;1\} \tag{2}$$

Question 4.1. *Combien y-a-t-il de fonctions prenant n-bits en entrée et renvoyant un bit ?*

The evaluation of any such function can be reduced to a sequence of elementary logical operations. Let us divide the possible values of the input

$$x = x_1 x_2 \dots x_n \tag{3}$$

into one set of values for which f(x) = 1, and a complementary set for which f(x) = 0. For each $x^{(a)}$ such that $f(x^{(a)}) = 1$, consider the function $f^{(a)}$ such that

$$f^{(a)} = \begin{cases} 1 & \text{si} \quad x = x^{(a)} \\ 0 & \text{sinon} \end{cases} \tag{4}$$

Then

$$f(x) = f^{(1)}(x) \vee f^{(2)}(x) \vee \dots \tag{5}$$

Now consider the evaluation of $f^{(a)}$. In the case where $x^{(a)} = 111...1$, we may write

$$f^{(a)}(x) = x_1 \wedge x_2 \wedge \dots \wedge x_n \tag{6}$$

For any other $x^{(a)}$, $f^{(a)}$ is again obtained as the AND of n bits, but where the NOT ($\neg$) operation is first applied to each xi such that $x_i^{(a)} = 0$; for example

$$f^{(a)}(x) = (\neg x1) \wedge x_2 \wedge x_3 \wedge (\neg x4) \wedge \dots \tag{7}$$

$$x^{(a)} = 0110 \dots \tag{8}$$

We have now constructed the function f(x) from three elementary logical connectives: NOT, AND, OR. The expression we obtained is called the "disjunctive normal form" of f(x).