## Calcul des propositions et algèbre de Boole

Un processeur est composé de transistors permettant de réaliser des fonctions sur des signaux numériques. Ces transistors, assemblés entre eux forment des composants permettant de réaliser des fonctions très simples. A partir de ces composants il est possible de créer des circuits réalisant des opérations très complexes. L'algèbre de Boole est un moyen d'arriver à créer de tels circuits

L'algèbre de Boole est une algèbre se proposant de traduire des signaux en expressions mathématiques. Des méthodes (table de vérité) permettent de définir les opérations que l'on désire réaliser, et à transcrire le résultat en une expression algébrique. Grâce à des règles appelées lois de composition, ces expressions peuvent être simplifiées. Cela va permettre de représenter grâce à des symboles un circuit logique, c'est-à-dire un circuit qui schématise l'agencement des composants de base sans se préoccuper de la réalisation au moyen de transistors.

Un ordinateur ne manipule que des données binaires, on appelle donc variable logique une donnée binaire, c'est-à-dire une donnée ayant deux états possibles : 0 ou 1. L’algèbre de Boole permet de donner un résultat à une combinaison de variables.

L’algèbre de Boole est composé des éléments suivants :

* Des variables booléennes, 1 ou 0, respectivement VRAI et FAUX.
* Un ensemble E.

Les opérateurs logiques :

* Des opérations binaires : + et \* en algèbre de Boole, respectivement **OU** (il correspond à la réunion de deux conditions, il peut aussi être symbolisé par Or dans les conditions ou V en maths) et **ET** (il correspond à l’intersection de deux conditions, il peut aussi être symbolisé par AND ou && dans les conditions et en maths par ∧.
* Une opération unaire : https://img-19.ccm2.net/rRYNRCmcBsgDczDMS5yiugENiYM=/e22677d8850d44908c010cbd08029b95/ccm-encyclopedia/logic-images-non.gif, correspondant à la négation logique. Il correspond aussi au complément à 1 d’une condition. Il est représenté dans les conditions par ! et en maths par ¬
* L’opérateur XOR aussi appelé fonction OU EXCLUSIF est représenté par un plus encerclé : https://img-19.ccm2.net/ygH5LyyJbKHESdozAvCccnxjIrQ=/e68279b0b269409e8e50cc538b20d3f5/ccm-encyclopedia/logic-images-xor.gif. Il correspond à l’intersection de deux conditions sans réunion (exemple : je prends l’avion OU EXCLUSIF le train).
* L’opérateur NXOR (NON EXCLUSIF), il n’a pas de symbolisation propre on le note juste NXOR.
* L’opérateur NAND (NON ET), c’est-à-dire la l’enchainement de la fonction NON et ET symbolisé par ↑.
* L’opérateur NOR (NON OU), l’enchainement de la fonction OU et NON symbolisé par ↓.

Les règles de simplifications :

Associativité :

Comme avec les opérations habituelles, certaines parenthèses sont inutiles :

(a + b) + c = a + (b + c) = a + b + c

(a·b)·c = a·(b·c) = a·b·c

Commutativité :

L'ordre est sans importance :

a + b = b + a

a·b = b·a

Distributivité

Comme avec les opérations mathématiques habituelles, il est possible de distribuer :

a·(b + c) = a·b + a·c

Attention : comportement différent par rapport aux opérateurs + et · habituels : a + (b·c) = (a + b)·(a + c)

Élément neutre

a + 0 = a

a·1 = a

Élément nul 0·a = 0

1 + a = 1

Idempotence

a + a + a + [...] + a = a

a·a·a·[...]·a = a

Complémentarité

a = ¬(¬a)

a + = 1

a · = 0

Lois de De Morgan





Ci-dessous la table de vérité, les opérations booléennes et la représentation symbolique de la porte logique :

