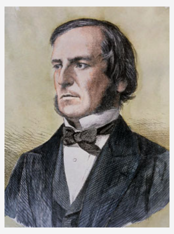
Notions de base

2. Le calcul propositionnel

En algorithmique et en informatique de manière générale, nous avons besoin d'indiquer au programme ce q'il doit faire. Nous utilisons pour cela la notion de proposition.

Nous allons ici nous intéresser à l'algèbre de Boole, le calcul booléen.


Georges Boole (1815-1864)

La proposition

Une proposition est une affirmation qui prend toujours la même valeur de vérité.

Il existe 2 valeurs: vrai et faux Une affirmation est soit vraie, soit fausse mais ne peut pas être en partie vraie et en partie fausse.

Par exemple, l'affirmation *10 est un nombre pair* est une proposition.

La négation

La négation d'une proposition est la proposition obtenue en affirmant son contraire.

Soit une proposition P, sa négation se dit alors *non P*.

Par exemple, L'affirmation *la chaise a quatre pieds* a pour négation *la chaise n'a pas quatre pieds*

Suivant les cas, la négation peut s'exprimer de différentes manières. Par exemple pour les notions exclusives, on peut simplement exprimer l'autre notion.

Exemple d'un nombre pair :

n est un nombre entier pair a pour négation *n n'est pas un nombre entier pair* mais également *n est un nombre entier impair*. En effet un nombre est soit pair, soit impair.

Il est donc possible d'utiliser soit la négation, soit directement l'inverse de la proposition quand cela est possible. Cela permet de gagner en lisibilité sans avoir à devoir interpréter la négation lors de la lecture.

Exemple de comparaison entre les nombres

a est plus grand que b a pour négation *a n'est pas plus grand que b*.

Mais si a n'est pas plus grand que b, que peut-on en déduire au juste ? Il ne reste que 2 possibilités :

- a est plus petit que b
- a est égal à b

Peut-on dans ce cas exprimer l'inverse de la proposition (sans utuiliser la négation) ?

OUI, grâce aux connecteurs...

Connecteurs

Un connecteur en logique mathématique est un outil qui permet de créer une proposition à partir d'une ou plusieurs proposition initiales.

Le connecteur ET (conjonction)

Une proposition P est composée de plusieurs propositions connectées à l'aide du connecteur ET.

P sera alors considérée comme vraie, uniquement si toutes les propositions connectées sont vraies. Dans tous les autres cas, P sera fausse.

Par exemple, la proposion *la chaise est verte ET le mur est bleu* est composée de 2 proposions et ne sera vraie que si les 2 propositions sont vraies. La chaise doit donc être verte et le mur bleu.

Si on reprend la proposition *a est plus grand que b*, peut-on utiliser ce connecteur pour exprimer l'inverse ? Que donnerait *a est plus petit que b ET a est égal à b* ? Cette proposition ne pourra jamais être vraie car a ne pourra jamais être plus petit que b ET égal à b.

Ce connecteur ne nous permet donc pas d'exprimer l'inverse de notre proposition mais il existe un autre connecteur.

Le connecteur OU (disjonction)

Une proposition P est composée de plusieurs propositions connectées à l'aide du connecteur OU.

P sera alors considérée comme vraie, si au moins une des propositions connectées est vraie. P sera donc fausse uniquement si toutes les propositions connectées sont fausses.

Par exemple, la proposition *la chaise est verte OU le mur est bleu* est composée de 2 proposions et sera vraie -si la chaise est verte quelque soit la couleur du mur -si le mur est bleu quelque soit la couleur de la chaise

Elle ne sera donc fausse que si le mur n'est pas bleu et que la chaise n'est pas verte

Négation dans le cas des connecteurs

Comme nous venons de le voir, il existe pour une proposition 2 valeurs de vérité VRAI et FAUX et il est possible de passer de l'un a l'autre en utilisant

- soit la négation
- soit en inversant la proposition quand c'est possible

Dans le cas de propositions avec connecteur, il est tout à fait possible de changer de valeur de vérité de la même façon :

Si on considère la proposition P composée de A ET B :

P = A ET B

NON P = NON (A ET B)

NON (A ET B) = (NON A) OU (NON B)

NON P = (NON A) OU (NON B)

Si on considère la proposition P composée de A OU B :

P = A OU B

NON P = NON (A OU B)

NON (A OU B) = (NON A) ET (NON B)

NON P = (NON A) ET (NON B)

Ce sont les lois de Morgan.

Si on souhaite faire la démonstration, on peut passer par les tables de vérité. On considère alors que 2 propositions sont équivalente car elles ont la meme table de vérité.

Les tables de vérité NON, ET, OU

NON

P	NON P
VRAI	FAUX
FAUX	VRAI

ET

A	B	A ET B
VRAI	VRAI	VRAI
VRAI	FAUX	FAUX
FAUX	VRAI	FAUX
FAUX	FAUX	FAUX

OU

A	B	A OU B
VRAI	VRAI	VRAI
VRAI	FAUX	VRAI
FAUX	VRAI	VRAI
FAUX	FAUX	FAUX

L'algèbre de Boole, ou plus précisément la logique booléenne est la base du raisonnement algorithmique.

Basé sur des propositions et des connecteurs, il est ainsi possible de modéliser des raisonnements logiques, en exprimant un « état » en fonction de conditions.

Exemple :
Communication = Émetteur ET Récepteur
ou pour un téléphone, Décrocher = (Sonnerie ET Décision de répondre) OU Décision d'appeler

Il est ainsi possible de modéliser tout état logique et nous allons voir par la suite comment cela va nous aider a construire nos algorithmes de programmation.