Semaine 2
Logique booléenne

La logique est une branche des mathématiques et de l'informatique qui permet d'évaluer la qualité d'un raisonnement. Il existe plusieurs types de logique plus ou moins complexes. Nous étudierons cette année la **logique propositionnelle**, également appelée **logique booléenne** (prononcer« bouléenne ») en hommage à George Boole, fondateur de la logique moderne (voir Note 2.1).

Comme son nom l'indique, la logique propositionnelle étudie les raisonnements basés sur des propositions logiques. Ces propositions, également appelées **prédicats**, peuvent uniquement prendre les valeurs VRAI ou FAUX, fréquemment représentées à l'aide des nombres 0 (pour faux) et 1 (pour vrai). Il est ainsi possible de noter "le train est rouge" = 1 pour signifier que le prédicat "le train est rouge" est vrai.

Calcul booléen

Opérateurs booléens

En arithmétique, il est possible de combiner plusieurs nombres à l'aide d'opérateurs afin d'obtenir de nouveaux nombres. Par exemple, il est possible d'utiliser l'opérateur de l'addition « + » au sein de l'expression suivante : 1 + 1 = 2.

En logique propositionnelle, ces opérateurs ne s'appliquent pas à des nombres, mais aux prédicats eux-mêmes. On parle alors d'opérateur logiques, ou d'**opérateurs booléens**.

La négation (« NON »)

La négation est le plus simple des opérateurs booléens. L'opérateur NON (*NOT* en anglais) est noté « ¬ », et possède une signification plutôt intuitive : la négation renverse la valeur d'un prédicat. Par exemple, si`"IlFaitBeau" =1`, alors sa négation `¬("IlFaitBeau")= 0`.La négation ne s'appliquant qu'à un prédicat, il s'agit d'un opérateur **unaire**.

Il est possible de résumer l'effet d'un opérateur booléen au sein d'une **table de vérité**. Par exemple, la table de vérité de la négation est la suivante :

A	¬A
VRAI	FAUX
FAUX	VRAI

*Tableau 1,*Table de vérité de la négation.

Cette table indique que si le prédicat `"A"` vaut VRAI, alors sa négation `"¬A"` vaut FAUX.

La conjonction (« AND »)

La conjonction est également un opérateur assez intuitif. Si la négation ne s'applique qu'à un prédicat, l'opérateur ET (*AND*) s'applique à deux prédicats, il s'agit donc d'un **opérateur binaire**. Il est noté soit « & », soit « ∧ », et possède la table de vérité suivante :

A	B	A ∧ B
VRAI	VRAI	VRAI
VRAI	FAUX	FAUX
FAUX	VRAI	FAUX
FAUX	FAUX	FAUX

Tableau 2, Table de vérité de la conjonction.

On notera que le prédicat A∧B n'est vrai que lorsque les deux prédicats A et B sont vrais.

La disjonction (« OR »)

Très semblable à la conjonction, l'opérateur OU (*OR*)est également binaire. Il est noté « ∨ », et possède la table de vérité suivante :

A	B	A ∨ B
VRAI	VRAI	VRAI
VRAI	FAUX	VRAI
FAUX	VRAI	VRAI
FAUX	FAUX	FAUX

Tableau 3, Table de vérité de la disjonction.

On notera que le prédicat A ∨ B n'est faux que lorsque les deux prédicats A et B sont faux. Contrairement au « ou » français, qui est souvent synonyme d'exclusivité, le « ou » logique est vrai même si les deux prédicats A, B sont vrais. Ainsi, en logique, la phrase « Vous voulez du thé ou du café ? » vous permet d'avoir les deux boissons chaudes en répondant simplement « oui ».

Note 2.1

Invention du calcul booléen (ou algèbre de Boole)

L'algèbre de Boole fut établie par le mathématicien anglais George Boole aux alentours des années 1850. Similaire à l'algèbre classique, l'algèbre de Boole permet de formaliser les opérations logiques. Cette formalisation a été fondamentale dans l'établissement de l'informatique.

La disjonction exclusive (« XOR »)

Il existe une disjonction exclusive dénommée XOR (de l'anglais *exclusive* OR). Elle est traditionnellement notée $\oplus$ et possède la table de vérité suivante :

A	B	A $\oplus$ B
VRAI	VRAI	FAUX
VRAI	FAUX	VRAI
FAUX	VRAI	VRAI
FAUX	FAUX	FAUX

Tableau 4, Table de vérité de la disjonction exclusive.

Equivalence entre opérateurs

Il existe de nombreux autres opérateurs booléens, cependant ces derniers ne seront pas présentés dans le cadre de ce cours. Ces autres opérateurs sont le plus souvent obtenus en combinant les opérateurs présentés précédemment. Par exemple, il est possible d'obtenir l'opérateur ∨ en combinant l'opérateur ¬ avec l'opérateur ∧. Pour s'en assurer, étudions la table de vérité de l'expression « ¬(¬A ∧¬B) » :

A	B	¬A	¬B	¬A ∧¬B	¬(¬A ∧¬B)
VRAI	VRAI	FAUX	FAUX	FAUX	VRAI
VRAI	FAUX	FAUX	VRAI	FAUX	VRAI
FAUX	VRAI	VRAI	FAUX	FAUX	VRAI
FAUX	FAUX	VRAI	VRAI	VRAI	FAUX

Tableau 5, Obtention d'une disjonction à partir de négations et d'une conjonction.

Tout comme en arithmétique, il existe une priorité entre les opérateurs. L'écriture de parenthèses influence par exemple le sens d'une expression logique. Ainsi, comparons la table de vérité des expressions « ¬(A ∧B) » et « ¬A ∧ B » :

A	B	¬(A ∧ B)	¬A ∧ B
VRAI	VRAI	FAUX	FAUX
VRAI	FAUX	FAUX	FAUX
FAUX	VRAI	FAUX	VRAI
FAUX	FAUX	VRAI	FAUX

Tableau 6, Influence de la priorité des opérateurs sur la vérité d'une expression.

Circuits logiques

Introduction au binaire

En informatique, l'information est portée par des nombres binaires, c'est-à-dire des nombres écrits en base 2. Dans cette base, la plus petite information est codée sur un « chiffre binaire » plus communément appelé **bit** (contraction de l'anglais *binary digit*) qui peut prendre les valeurs 0 ou 1. Tout comme pour les nombres décimaux, il est possible de former des nombres binaires en écrivant une succession de chiffes binaires (et donc de bits).Nous étudierons au prochain cours (***Semaine 3, Les bases numériques***) comment convertir les nombres décimaux en nombres binaires, mais voici pour l'instant quelques exemples :

- Les chiffres décimaux 0 et 1 s'écrivent de la même façon en binaire.
- A partir de 2, les décimaux sont convertis en binaire en utilisant plusieurs bits :
  - 2 s'écrit 10, 3 s'écrit 11, 4 s'écrit 100…
  - Toute série de bit (par exemple 10010) peut donc représenter un nombre décimal (ici 18).

Utiliser l'algèbre de Boole pour réaliser des opérations arithmétiques

Comme vous l'avez peut-être déjà remarqué, il est possible de représenter la valeur d'une expression booléenne à l'aide d'un seul bit, 0 représentant une expression fausse et 1 une expression vraie.

Il est également possible d'appliquer les opérateurs booléens sur des nombres binaires plus longs en réalisant des **opérations bit à bit**. Par exemple :

$$\begin{array}{rll} & 10010 & \text{(décimal : 18)} \\ \wedge & 01010 & \text{(décimal : 10)} \\ \hline & 00010 & \text{(décimal : 2)} \end{array}$$

Les opérations arithmétiques telles que l'addition peuvent également être représentées à l'aide d'opérateurs booléens. Il est par exemple possible de réaliser un(demi-)additionneur à l'aide des opérateurs booléens XOR et AND :

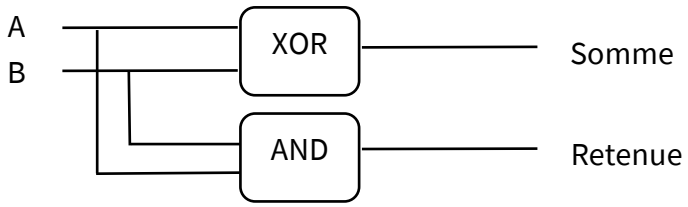

Figure 1, Demi-additionneur.

Etudions les tables de vérités de cet additionneur :

A	B	Somme (soit A $\oplus$B)	Retenue (soit A $\wedge$ B)
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

Si l'on observe la somme de cet additionneur, on peut remarquer que 0 + 0 = 0, que1 + 0 = 1 et que 0 + 1 = 1. Lorsque l'on effectue 1 + 1, on fait alors intervenir la retenue (comme dans une addition décimale) et l'on obtient 1 + 1 = 10 (soit 2 en binaire).

Pour aller plus loin : Le vidéaste e-penser a réalisé de très bonnes vidéos explicatives sur le demi-additionneur (et sur sa variante plus complexe, l'additionneur complet).

Exercices non à soumettre

Exercice 1

Etablissez les tables de vérité des opérations binaires suivantes :

1. $A \wedge \neg A$
2. $\neg A \vee B$
3. $(A \wedge B) \vee C$

Exercice 2

Réaliser les opérations bit à bit suivantes :

1. $\neg 0110$
2. $10011 \wedge \neg 01100$
3. $(1011 \oplus 0110) \vee 0010$

Envoyer le devoir à soumettre n°1