Numérique et science informatique

Lycée Hoche

année scolaire 2025-2026

Contents

1 Valeurs booléennes

En python,il existe des opérateurs relationnels ou de comparaison entre deux objets x et y tel que le résultat de la relation ou de la comparaison soit un booléen comme dans l'exemple ci-dessous:

Pour les entiers $x = 15$ et $y = 8$

opérateur	expression	signification	valeur booléenne
==	$x == y$	x égal	0(Faux)
!=	$x! = y$	non égal	1(Vrai)
>	$x > y$	plus grand	1
<	$x < y$	plus petit	0
>=	$x >= y$	plus grand ou égal	1
<=	$x <= y$	plus petit ou égal	0
is	x is y	est le même objet	0
is not	x is not y	n'est pas le même objet	1

Exercice

Tester dans le *shell* les différents opérateurs relationnels ou de comparaison ci-dessus en les appliquant sur plusieurs types d'objets.

2 Fonction booléenne et table de vérité

fonctions booléennes

Définition

Pour simplifier, une *fonction booléenne* est une fonction qui associe à un ou plusieurs booléens (c'est-à-dire 0 ou 1) un booléen (0 ou 1).

Une fonction booléenne est entièrement déterminée à l'aide d'une table , **table de vérité** de la fonction.

On donne ci-dessous les tables de vérité des fonctions booléennes (ou fonction logiques) *NON,ET* ainsi que la fonction *OU*

x	Non(x)
0	1
1	0

Table 1: NON

x	y	ET(x,y)
0	0	0
0	1	0
1	0	0
1	1	1

Table 2: *fonction ET*

x	y	OU(x,y)
0	0	0
0	1	1
1	0	1
1	1	1

Table 3: *fonction OU*

• En logique booléenne , on écrit $NON(x) = \bar{x}$ (on parle également d'opérateur *unaire*). $ET(x, y) = x\ ET\ y$, $OU(x, y) = x\ OU\ y$. (On parle également d'opérateurs *binaires*).

On écrit également :

$$x.y \quad \text{pour} \quad x\ ET\ y \quad \text{et} \quad x + y \quad \text{pour} \quad x\ OU\ y.$$

Exercice

1. Combien existe-t-il de fonctions unaires définies sur $\mathcal{B} = \{0, 1\}$?
2. Combien existe-t-il de fonctions booléennes définies sur $\mathcal{B} \times \mathcal{B}$?

Les opérateurs *Et* , *OU* (respectivement . et +) possèdent des propriétés algébriques (on parle d'algèbre de Boole).

Propriétés algébriques des opérateurs + et .

Pour tous booléens a, b et c , on a :

- $a + 0 = 0 + a = a$ (on dit que 0 est l'élément neutre de l'ensemble $\mathcal{B} = \{0, 1\}$ pour l'opération +).
- $a.1 = 1.a = a$ (on dit que 1 est l'élément neutre de l'ensemble pour l'opération .).
- $a.0 = 0.a = 0$ et $a + 1 = 1 + a = 1$.
- $a + \bar{a} = 1$ et $a.\bar{a} = 0$
- $a + b = b + a$; $a.b = b.a$ (On dit que les opérations + et . sont commutatives dans $\mathcal{B}$)
- $a.(b.c) = (a.b).c = a.b.c$ et $a + (b + c) = (a + b) + c = a + b + c$ (On dit que les opérations + et . sont associatives dans $\mathcal{B}$)
- $a.(b + c) = a.b + a.c$ $(a + b).c = a.c + b.c$ (On parle de la distributivité de . par rapport à +)
- $a + a = a$ et $a.a = a$ (On parle d'idempotence)

Ces propriétés peuvent être vérifiées en utilisant les tables de vérité des opérateurs *Et* et *OU*

• En Python , les opérateurs booléens correspondant aux fonctions booléennes : *Non* , *OU* , *ET* sont respectivement **not** , **or** et **and**.

Exercice

Tester dans le *shell* les opérateurs booléens en python (en testant par exemple les expressions comme 0 *and* 1 ,1 *or* 1)

Remarque importante

Les trois fonctions *NON* , *ET* , *OU* sont les briques de base à l'aide desquelles on peut exprimer les autres fonctions booléennes.

Considérons l'opérateur binaire **XOR** dont la table de vérité est donnée ci-dessous

x	y	XOR(x,y)
0	0	0
0	1	1
1	0	1
1	1	0

Table 4: *fonction XOR*

Remarquer que cette fonction donne 1 lorsque (x=1 et y=0) ou (x=0 et y=1).On parle alors d'un ***OU exclusif*** .

Exercice

Vérifier à l'aide de la table de vérité de XOR que :

$$\text{xor}(x,y)=(\text{non}(y) \text{ et } x) \text{ ou } (y \text{ et non}(x)).$$

Ce qui peut s'écrire à l'aide des opérations + et . :

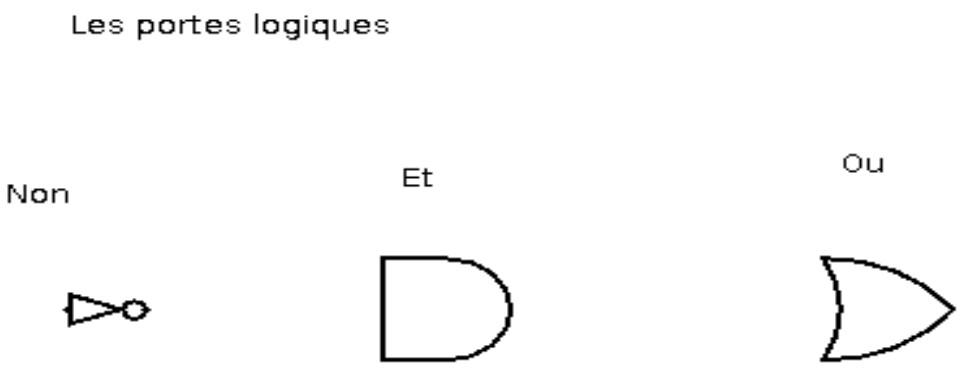
$$\text{xor}(x,y) = \bar{y}.x + y.\bar{x}$$

La formule obtenue dans l'exercice précédent à l'aide des opérateurs ET,OU et NON est dite **forme normale disjonctive** de la fonction XOR.

Pour une fonction quelconque , une méthode élémentaire consiste à faire la somme - c'est-à-dire le OU - des différents cas où la fonction donne 1 (vrai) , chaque cas s'exprimant comme le produit - c'est-à-dire le ET (la conjonction) - de l'ensemble des entrées:
Les entrées à 1 se retrouvent telles quelles tandis que les entrées à 0 sont complémentées dans le produit.

3 opérateurs booléens et circuits logiques

En architecture des ordinateurs , les opérateurs booléens se traduisent par des portes logiques dont les symboles sont les suivants:



Shéma logique du Xor