

NSI - Première

Booléens : TD

qkzk

2021/04/19

Objectifs

- Construire la table de vérité d'une expression booléenne relativement simple.
- Évaluer et construire des expressions booléennes en Python

Exercice 1

Construire la table de vérité de l'expression : $a \text{ OU } (\text{NON } b)$

Exercice 2

Construire la table de vérité de l'expression : $\text{NON } a \text{ et } (b \text{ ou } c)$

Exercice 3

Construire la table de vérité de l'expression : $(a \text{ et } \text{NON } b) \text{ ou } (\text{NON } a \text{ et } b)$

Exercice 4

Construire la table de vérité de l'expression : $(a \text{ OU } b) \text{ ET } (a \text{ OU } c)$

Exercice 5

Trouver une expression équivalente à $a \text{ et } b$ construite uniquement à partir des opérateurs NON et ou
On vérifiera à l'aide d'une table de vérité.

Exercice 6

Trouver une expression équivalente à $a \text{ OU } b$ construite uniquement à partir des opérateurs NON et et
On vérifiera à l'aide d'une table de vérité.

Exercice 7

Donner la valeur des expressions booléennes suivantes :

```
(1 > 2) and (3 < 5)
((4 - 7) >= 2) or (2 != 1 + 1)
a = 223
b = 455
a != (b // 2)
```

Exercice 8

insérer le minimum de parenthèses dans les expressions suivantes pour les égalités soient correctes

```
2 + 3 * 5 + 4 == 21
5 + 2 * 3 + 4 == 25
4 + 5 * 2 + 3 == 29
```

Exercice 9

La clé de vérification utilisée en France pour les numéros de sécurité sociale est égal à 97 moins le résidu modulo 97 du nombre formé par les autres chiffres : c'est-à-dire que la clé est l'unique entier entre 01 et 97 tel que la somme de la clé avec le reste du numéro fasse un entier divisible par 97.

Nous allons traduire ceci en Python

1. Voici le numéro de sécurité sociale de JP : 1 81 10 59 340 223. Calculer sa clé de sécurité à l'aide d'une calculatrice.
2. Proposer une expression de la clé de sécurité.
3. Écrire cette expression en Python en supposant que le numéro est noté `numero`

```
numero = 1811059340223
cle = ???
```

4. Proposer une expression booléenne qui soit vraie si `cle` est bien la clé de sécurité de `numero` et fausse sinon.

Exercice 10 - Demi additionneur binaire

Représentation graphique

A chaque porte est associée une représentation graphique. Voici pour les portes ET et XOR :

- porte ET



Figure 1: porte ET

- Porte XOR

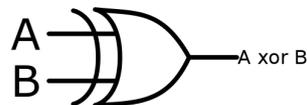


Figure 2: porte XOR

1. Rappeler les tables de vérité de ces deux portes.
2. On considère deux entiers A et B représentés sur un bit.
 - On les additionne et l'éventuelle retenue est perdue. Quels sont les résultats possibles ? Cette opération peut-elle être comparée à une des tables précédentes ?
 - On recommence, mais cette fois on décide de noter sur deux bits le résultat de la somme. Dans quel cas aura-t-on une retenue à écrire ? Quelle opération booléenne sur les bits permet d'obtenir ce résultat ?

Portes logiques

Les opérations logiques évoquées ci-dessus sont mises en oeuvre en électronique sous forme de **portes logiques**.

Les circuits électroniques calculent des fonctions logiques de l'algèbre de Boole.

Pour chacun des opérateurs logiques évoquées ci-dessus (et d'autres) il existe donc des portes logiques appelés *porte ET*, *porte NON*, etc. Les valeurs *vrai* et *faux* sont représentées par deux niveaux de tension, *haut* et *bas*.

Demi additionneur

Un circuit de type *porte ET* dispose donc de deux entrées et une sortie.

La valeur du niveau de tension en sortie est obtenue avec la table de vérité du ET.

Les portes peuvent être connectées entre elles pour réaliser des **circuits logiques** et on peut ainsi réaliser des calculs.

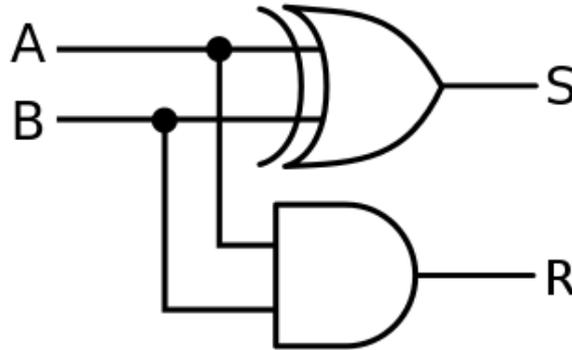


Figure 3: demi-additionneur

Il est appelé *demi-additionneur* car il réalise l'addition de 2 bits (A et B), le résultats de cette somme est représentée par S et la retenue éventuelle par R .

Construisez les tables de vérité de S et R et comparez à celle de l'addition de deux bits A et B .