Eléments de logique formelle et
du raisonnement mathématique

Eléments de la logique propositionnelle (1)

2018-2019

Plan

1. Langage propositionnel
2. Redondance de formules et la formule normale disjonctive (FND)
3. Expressivité de connecteurs
4. Exemple : le connecteur *XOR* et ses applications

Langage propositionnel

Un langage propositionnel est un ensemble de formules bien formées (par exemple, une formule peut être vérifiée s'elle est bien formée par l'arbre de décomposition).

Les symboles :

- $\top, \bot$: vrai, faux
- variables propositionnelles : p,q,r,…
- connecteurs : $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
- parenthèses : ()

Les formules :

1. $\top, \bot$ et les variables propositionnelles sont des formules (formules atomiques);
2. Le résultat de l'application d'un connecteur à une ou plusieurs formules est encore une formule (formule composée);
3. Il n'existe pas d'autres formules dans le langage propositionnel que celles obtenues à partir de 1 et 2.

Langage fini : le nombre de variables propositionnelles est fini.

Langage propositionnel

Les notions de *formules équivalentes*, *tautologie*, *contradiction* ainsi que *table de vérité* et *arbre de décomposition* recouvrent des notions identiques à celles introduites au chapitre ***Langage et Logique***.

Redondance de formules et forme normale disjonctive

Question 1 :

Combien de formules non équivalentes existe-t-il dans un langage propositionnel construit sur n variables?

Remarquons que, même le langage propositionnel le plus simple, construit sur une seule variable propositionnelle p, noté L(p), comporte une infinité de formules:

- P, ¬P, ¬(¬P),…

Aussi on s'intéresse aux Forme Normale Disjonctive (FND) pour étudier cette redondance.

Forme normale disjonctive (FND)

Exemple :

Considérons un langage L(p,q,r) construit de trois variables propostionnelles p,q,r.

Etant donné une table de vérité, construire une formule A en FND correspondante à cette table.

p	q	r	A
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Forme normale disjonctive (FND)

On procède de la manière suivante :

1. On choisit la première ligne où A est censée prendre la valeur 1 :

   Ligne (3) : 0,1,0, on écrit $\neg p \wedge q \wedge \neg r$.

2. On réitère maintenant le procédé pour toutes les lignes où A prend la valeur 1 :

   Ligne (5) : 1,0,0, on écrit $p \wedge \neg q \wedge \neg r$.

   Ligne (8) : 1,1,1, on écrit $p \wedge q \wedge r$.

3. On forme enfin la disjonction de 3 formules obtenues en 1, 2 :

   $A = (\neg p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge \neg r) \vee (p \wedge q \wedge r)$

p	q	r	A
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Forme normale disjonctive (FND)

Une formule construite ainsi est une formule en FND (La définition complète de FND est donnée au chapitre 5).

Toute formule peut être convertie en une telle formule, et deux formules en FND distinctes ne sont pas équivalentes.

Nombre de formules non équivalentes

Exemple :

Dans le langage L(p,q,r), la table de vérité d'une formule A est donnée par une colonne de 8 chiffres, chacun d'eux étant égal à 0 ou 1.

Il y a donc 2^8 = 256 tables de vérité, comme une table correspond à une formule FND, donc il y a 256 formules non équivalents.

Dans le cas général, il existe 2^(2^n) formules non équivalentes dans un langage construit sur n variables propositionnelles.

p	q	r	A
0	0	0	*
0	0	1	*
0	1	0	*
0	1	1	*
1	0	0	*
1	0	1	*
1	1	0	*
1	1	1	*

Nombre de formules non équivalentes

Exemple :

Considérons L(p), quelles sont les formules non équivalentes?

La table de vérité d'une telle formule est donnée par 2 chiffres, 0 ou 1.

Il y a donc 2^2 = 4 formules :

A1 = $\neg p \wedge p$ ($\bot$)

A2 = p

A3 = $\neg p$

A4 = $\neg p \vee p$ ($\top$)

p	A1	A2	A3	A4
0	0	0	1	1
1	0	1	0	1

Expressivité de connecteurs

Dans la construction d'un langage propositionnel, le choix des connecteurs dépend de l'usage auquel est destiné ce langage.

Dans le but de faciliter l'expression, on se donne un ensemble de connecteurs $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$.

Question 2 :

Est-ce que $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ est un ensemble minimal de connecteurs pour exprimer toute formule du langage propositionnel?

Réponse :

Non.

Expressivité de connecteurs

On a :

$p \wedge q = \neg(\neg p \vee \neg q)$

$p \rightarrow q = \neg p \vee q$

$p \leftrightarrow q = (p \rightarrow q) \wedge (q \rightarrow p)$

Il s'ensuit qu'on peut construire toute formule en utilisant seulement $\{\neg, \vee\}$.

Des transformations similaires peuvent être montrées pour $\{\neg, \wedge\}$, $\{\neg, \rightarrow\}$.

Expressivité de connecteurs

Propositions 1 :

{¬, ∨}, {¬, ∧}, {¬, →} sont les uniques sous-ensembles minimaux d’être capable d'exprimer toute formule.

Proposition 2 :

Les connecteurs {↓} (NOR) et {↑} (NAND) seuls permettent d'exprimer toute formule.

Exemple : le connecteur *XOR* et ses applications

Le connecteur *XOR*

Le connecteur *XOR (ou exclusif*, noté $\oplus$) est défini : $p \oplus q$ est vraie si une et une seule des proposition p et q est vraie (soit p est vraie, soit q est vraie).

Par exemple :

Le menu propose fromage ou dessert.

1. Ecrire la table de vérité du $p \oplus q$ et montrer que

$$p \oplus q = (p \wedge \neg q) \vee (\neg p \wedge q)$$

2. Montrer que

$$p \oplus p = 0$$

$$p \oplus 0 = p$$

$$p \oplus q = q \oplus p$$

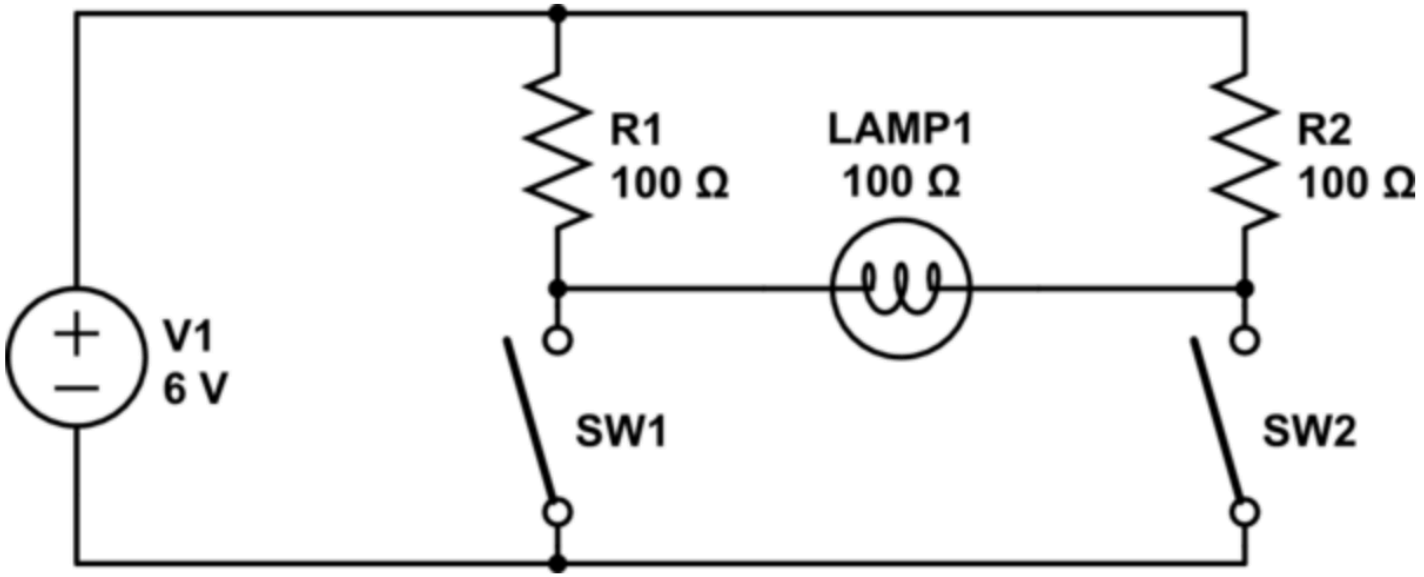
$$p \oplus (q \oplus r) = (p \oplus q) \oplus r$$

p	q	p⊕q
0	0	0
0	1	1
1	0	1
1	1	0

Exemple : le connecteur *XOR* et ses applications

Application 1 :

La lampe s'allume si l'on appuie sur « SW1 » ou « SW2 » seulement, mais pas si l'on appuie sur « SW1 » et « SW2 » simultanément. Cette application peut être exprimé par la porte logique XOR.

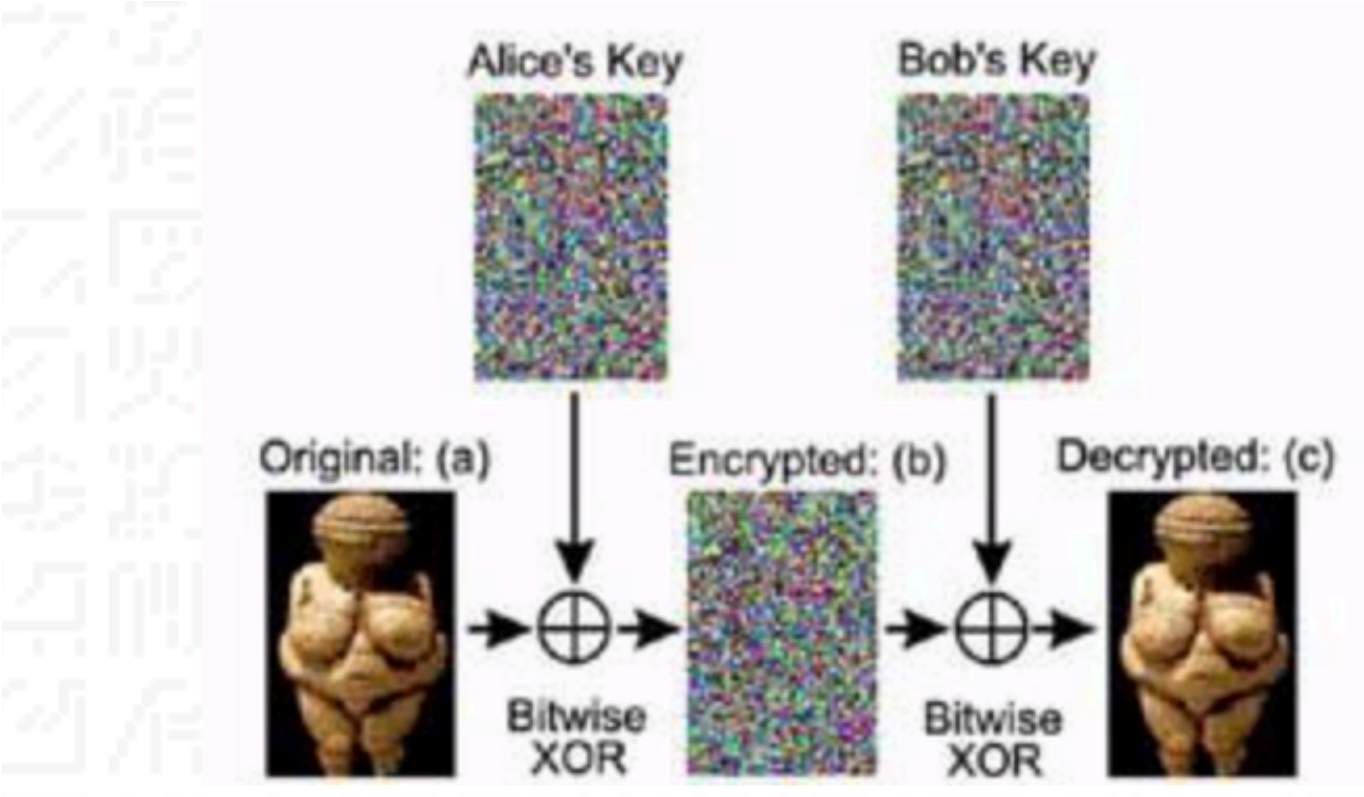


https://electronics.stackexchange.com/questions/86106/how-to-build-an-interesting-xor-circuit-with-my-childs-electronics-kit

Exemple : le connecteur *XOR* et ses applications

Application 2 : Images cryptées

Alice veut transmettre l'image du Vénus de Willendorf (Autriche, 30'000 à 25'000 ans) à Bob de façon en sécurité.



https://www.apprendre-en-ligne.net/crypto/images/index.html

Exemple : le connecteur *XOR* et ses applications

Application 2 : Images cryptées

Une image digitalisée peut être décrite comme une suite de pixels (des points lumineux), chaque pixel ayant une couleur représenté par un nombre entier entre 0 et 255 (0:noir, 255:blanc). Ces nombres sont codés en binaire sur 8 bits (par ex.: 01110011=115).

Par exemple, si le premier pixel de l'image originale est 01110011 et le premier pixel de l'image-clef 10100101, Alice crypte l'image originale avec cette clé en appliquant le connecteur XOR :

1er pixel de l'image originale	0	1	1	1	0	0	1	1 (A)
1er pixel de l'image-clef	1	0	1	0	0	1	0	1 (B)
1er pixel de l'image brouillée	1	1	0	1	0	1	1	0 (C=A⊕B)

Quand Bob reçoit l'image cryptée, il la décrypte avec la même clé en appliquant le connecteur XOR et retrouve l'image originale :

1er pixel de l'image brouillée	1	1	0	1	0	1	1	0 (C)
1er pixel de l'image-clef	1	0	1	0	0	1	0	1 (B)
1er pixel de l'image originale	0	1	1	1	0	0	1	1 (A = C⊕B=A⊕B⊕B)