Logique

Laurent Debize

Mathématiques appliquées à l'informatique

Calculs propositionnels

En logique, deux objets sont rencontrés :

- Des propositions (« phrases »mathématiques)
- Des connecteurs logiques (pour manipuler ces « phrases »)

Propositions

Les expressions mathématiques sont composées :

- de **termes** (ou ≪ mots ≫ mathématiques) qui doivent respecter une orthographe
  Les termes représentent des **objets**
  Exemples :
  $$-4\ ;\ \frac{1}{6}\ ;\ x\ ;\ A$$

- d'**énoncés** (ou ≪ phrases ≫ mathématiques) qui doivent respecter une syntaxe (ou ≪ grammaire ≫)
  Les énoncés énoncent des **faits**
  Exemples :
  - $2+2=4$
  - 8 est divisible par 3
  - $y>7$
  - $-x+y=3$ ($a=b$ qu'on lit ≪ a égale b ≫ signifie que a et b représentent la même entité mathématique)

Propositions

Remarque

Certaines expressions n'ont aucun sens mathématique :
Exemples :

$$\frac{1}{0} \ ; \ + = \times \ ; \ \sqrt{-1}$$

Propositions

Valeurs de vérité

- **Logique binaire** : Vrai (V) ou Faux (F)
- « Vrai (V) » et « Faux (F) » sont appelés **valeurs de vérité**
- **Logique de Boole (logique booléenne)** : Vrai $= 1$ et Faux $= 0$

Exemples :

- « $2 + 2 = 4$ » est un énoncé vrai
- « 5 est un nombre impair » est un énoncé vrai
- « pour tout réel x, $x^2 + 1 < 0$ » est un énoncé faux

Propositions

Définition

On appelle **proposition** ou **assertion** un énoncé qu'on peut juger sans ambiguïté Vrai ou Faux.

Notation

Les propositions sont généralement notées par des lettres majuscules (A, B, etc.)

Exemples

- $2 + 2 = 4$ est un énoncé qu'on peut juger, sans ambiguïté, qu'il est vrai donc c'est une proposition vraie.
- « 8 est divisible par 3 » est un énoncé qu'on peut juger, sans ambiguïté, qu'il est faux donc c'est une proposition fausse.
- « Le professeur est sympathique », **énoncé très subjectif**, n'est pas une proposition.
- « $x + 2 = 0$ » n'est pas une proposition car **la valeur de vérité dépend du réel x**.

Exercice 1

Est-ce que les énoncés, ci-dessous, sont des propositions ?

A : « Monsieur Martin est né un 1er janvier »

B : « Monsieur Martin est grand »

C : « Pour x réel, $3x + 4 = 0$ »

D : « Pour x réel, $3x^2 + 4 = 0$ »

Connecteurs logiques

A partir de propositions initiales, on peut définir de nouvelles propositions au moyen de **connecteurs logiques** comme :

- NON
- ET
- OU
- NAND
- NOR
- XOR
- l'implication
- l'équivalence

Ces transformations sont appelées des **opérations logiques**.

Ces opérations logiques peuvent aussi être définies par leurs tables opératoires appelées **tables de vérité**.

NON

Définition

Soit A une proposition.
La proposition « non A » est vraie lorsque A est fausse et vice-versa.

Notation

Cette proposition est notée $\overline{A}$ ou non A.
$\bar{}$ est le connecteur NON.

Table de vérité

A	$\overline{A}$
0	1
1	0

NON

Exemples

- Soit P la proposition « $x > 3$ ».
  Sa négation est la proposition $\overline{P}$: « $x \leq 3$ ».
- En Python, « non » se note « `not` » :

```
>>> A = (2==2)
>>> A
True
>>> not(A)
False
```

ET

Définition

Soient A et B deux propositions.
La proposition « A et B » n'est vraie que si les propositions A et B sont vraies simultanément.

Notation

Cette proposition est notée $A \cdot B$.
$\cdot$ est le connecteur ET.

Table de vérité

A	B	A · B
0	0	0
0	1	0
1	0	0
1	1	1

ET

Exemples

- La proposition « $14 - 3 = 11$ et $2 > 3$ » est fausse
- La proposition « 100 est pair et $100 = 10^2$ » est vraie
- En Python, « et » se note « and »
  et « != » signifie « $\neq$ » :

```
>>> (4==4) and (6!=6)
False
>>> (4<5) and (6<10)
True
```

Les masques de sous-réseau

Convertir cette adresse IPv4 suivante en binaire : 78.42.90.217

Convertir le masque de sous-réseau suivant en binaire : 255.255.240.0

Faire un ET logique bit à bit entre l'adresse IPv4 et le masque de sous-réseau. Que reste-t-il ?

Convertir cette adresse à nouveau en base 10.

Voilà comment on récupère l'adresse du sous-réseau !

OU

Définition

Soient A et B deux propositions.
La proposition « A ou B » est vraie si au moins une de deux propositions A et B est vraie.

Notation

Cette proposition est notée $A + B$.
$+$ est le connecteur OU.

Table de vérité

A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

OU

Remarque

Le connecteur $+$ utilisé en logique n'est pas le OU du langage courant :
choisir « fromage ou dessert » est le plus souvent interprété comme un choix exclusif (ou l'un ou l'autre mais pas les deux).

OU

Exemples

- La proposition « $14 - 3 = 11$ ou $2 > 3$ » est vraie
- La proposition « 100 est pair ou $100 = 10^2$ » est vraie
- En Python, « ou » se note « or »

```
>>> (2!=2) or (5<7)
True
>>> (7<3) or (6==10)
False
```

Exercice 2

On note P et Q les affirmations suivantes :

- P = « Paul aime le foot »
- Q = « Paul aime les maths »

Représenter les affirmations suivantes sous forme symbolique en utilisant P, Q et des connecteurs logiques.

- A = « Paul aime le foot mais pas les maths »
- B = « Paul n'aime ni le foot ni les maths »
- C = « Paul aime le foot ou il aime les maths et pas le foot »
- D = « Paul aime les maths et le foot ou il aime les maths mais pas le foot »

Exercice 3

Quelles sont les valeurs de vérité des propositions suivantes :

- $A = \ll \pi = 5$ et $2 + 3 = 5 \gg$
- $B = \ll \pi = 5$ ou $2 + 3 = 5 \gg$
- $C = \ll 11 > 0$ et $3 < 2 \gg$
- $D = \ll 3 > 6$ ou $6 > 20 \gg$

XOR

Définition

Soient A et B deux propositions.
La proposition ≪ A XOR B ≫ (A OU EXCLUSIF B) est vraie si une seule de deux propositions A ou B est vraie.

Notation

Cette proposition est notée A xor B.

Table de vérité

A	B	A *xor* B
0	0	0
0	1	1
1	0	1
1	1	0

Remarque

C'est le connecteur utilisé dans l'expression ≪ fromage ou dessert ≫ (l'un ou l'autre mais pas les deux).

Exercice 4

Soient A et B deux propositions.
Compléter la table de vérité suivante :

A	B	A xor B	(A xor B) xor B
0	0		
0	1		
1	0		
1	1		

A quoi est égal (A xor B) xor B ?

Dégager un intérêt pour la cryptographie.

NAND

Définition

Soient A et B deux propositions.
La proposition « A NAND B » est en fait la proposition « NON (A ET B) ».

Notation

Cette proposition est notée $\overline{A \cdot B}$.

Table de vérité

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

NOR

Définition

Soient A et B deux propositions.
La proposition ≪ A NOR B ≫ est en fait la proposition ≪ NON (A OU B) ≫.

Notation

Cette proposition est notée $\overline{A + B}$.

Table de vérité

A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

Equivalence

Définition

Soient A et B deux propositions.
L'équivalence des propositions A et B est la proposition « A si et seulement si B ».
Elle est vraie si les deux propositions A et B ont la même valeur de vérité.

Notation

Cette proposition est notée « $A \Leftrightarrow B$ »
$\Leftrightarrow$ est le connecteur ...SI ET SEULEMENT SI...

Table de vérité

A	B	$A \Leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

Equivalence

Exemples

- La proposition « $2+3=5 \Leftrightarrow 5<6$ » est vraie
- La proposition « $8 \geq 10 \Leftrightarrow 4=3$ » est vraie
- La proposition « $5+7=12 \Leftrightarrow 2=3$ » est fausse
- En Python, l'équivalence se note aussi « == »

```
>>> (5<6) == (1==4)
False
>>> (6<5) == (1==4)
True
```

Implication

Définition

Soient A et B deux propositions.
L'implication des propositions A et B est la proposition ≪ A implique B ≫.
Elle est vraie si les deux propositions A et B sont vraies, ou si A est fausse.

Notation

Cette proposition est notée ≪ $A \Rightarrow B$ ≫, ≪ Si A alors B ≫, ≪ A entraîne B ≫
$\Rightarrow$ est le connecteur SI... ALORS...

Table de vérité

A	B	$A \Rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

Implication

Remarque

Les deux dernières lignes de la table de vérité de « $A \Rightarrow B$ » montrent que le sens que les mathématiques donnent aux mots « implique » et « entraîne » est plus général que le langage courant.

Exemple :

- « $2+2=5$ » $\Rightarrow$ « La France a gagné la coupe du monde en 1998 » **est vraie**
- « La France a gagné la coupe du monde en 2014 » $\Rightarrow$ « $2+2=5$ » **est vraie**

Implication

Exemples

- La proposition « $5 < 5 \Rightarrow 5 = 5$ » est vraie
- La proposition « $5 = 5 \Rightarrow 5 < 5$ » est fausse
- En Python, l'implication se note « `<=` ».

```
>>> (3==2) <= (5<7)
True
>>> (2<4) <= (3>5)
False
```

Exercice 5

Quelles sont les valeurs de vérité des propositions suivantes :

- $A = \ll \pi \simeq 3,14 \Rightarrow 5 + 6 = 11 \gg$
- $B = \ll \pi = 5 \Rightarrow 2 + 3 = 5 \gg$
- $C = \ll 5 + 5 = 10 \Leftrightarrow \pi = 11 \gg$
- $D = \ll 3 < 2 \Rightarrow 5 = 5 \gg$

Propriétés des opérateurs logiques

Propriétés du connecteur NON

- $\overline{V} = F$ et $\overline{F} = V$
- $\overline{\overline{A}} = A$

Exercice 6

Complétez la table de vérité ci-dessous :

A	B	C	$A+B$	$(A+B)+C$	$B+C$	$A+(B+C)$
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Que concluez-vous, en observant les 5ᵉ et 7ᵉ colonnes ?

Propriétés des opérateurs logiques

Propriétés du connecteur OU

- idempotence : $A + A = A$
- commutativité : $A + B = B + A$
- associativité : $(A + B) + C = A + (B + C)$ que l'on peut donc noter $A + B + C$
- $A + F = A$ et $A + V = V$

Exercice 7

Complétez la table de vérité ci-dessous :

A	B	C	$A \cdot B$	$(A \cdot B) \cdot C$	$B \cdot C$	$A \cdot (B \cdot C)$
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Que concluez-vous, en observant les 5^e et 7^e colonnes ?

Propriétés des opérateurs logiques

Propriétés du connecteur ET

- idempotence : $A \cdot A = A$
- commutativité : $A \cdot B = B \cdot A$
- associativité : $(A \cdot B) \cdot C = A \cdot (B \cdot C)$ que l'on peut donc noter $A \cdot B \cdot C$
- $A \cdot F = F$ et $A \cdot V = A$

Exercice 8

Complétez la table de vérité ci-dessous :

A	B	C	$B+C$	$A\cdot(B+C)$	$A\cdot B$	$A\cdot C$	$(A\cdot B)+(A\cdot C)$
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

Que concluez-vous, en observant les 5^{e} et 8^{e} colonnes ?

Exercice 9

Complétez la table de vérité ci-dessous :

A	B	C	$B \cdot C$	$A + (B \cdot C)$	$A + B$	$A + C$	$(A + B) \cdot (A + C)$
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

Que concluez-vous, en observant les 5^e^ et 8^e^ colonnes ?

Propriétés des opérateurs logiques

Distributivité

- $A + (B \cdot C) = (A + B) \cdot (A + C)$
- $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$

Exemple

Soit a un réel. Soient les propositions :

- A : « $a > 0$ »
- B : « $a < -3$ »
- C : « $3 < a$ »

$A \cdot (B + C)$ est la proposition $(a > 0) \cdot (a < -3 + 3 < a)$
c'est-à-dire $(a > 0 \cdot a < -3) + (a > 0 \cdot 3 < a)$
soit $F + (a > 0 \cdot 3 < a)$
donc $A \cdot (B + C) = 3 < a$

Exercice 10

Complétez la table de vérité ci-dessous :

A	B	$\overline{A}$	$\overline{B}$	$A+B$	$\overline{A+B}$	$\overline{A}\cdot\overline{B}$
0	0					
0	1					
1	0					
1	1					

Que concluez-vous, en observant les 6ᵉ et 7ᵉ colonnes ?

Montrer de la même façon que $\overline{A\cdot B}=\overline{A}+\overline{B}$

Propriétés des opérateurs logiques

Lois de De Morgan

- $\overline{A + B} = \overline{A} \cdot \overline{B}$
- $\overline{A \cdot B} = \overline{A} + \overline{B}$

Exemple

Soit a un réel. Soient les propositions :

- A : « $a < -3$ »
- B : « $3 < a$ »

$$
\begin{array}{lll}
A + B & \text{est la proposition} & (a < -3) + (3 < a) \\
\overline{A + B} & \text{est la proposition} & \overline{(a < -3) + 3 < a}) \\
 & \text{c'est-à-dire} & (a \geq -3) \cdot (3 \geq a) \\
 & \text{soit} & -3 \leq a \leq 3
\end{array}
$$

Exercice 11

Au moment de demander le jour de naissance d'un utilisateur, on veut le forcer à saisir un nombre entre 1 et 31.

1. Écrire l'algorithme qui demande à l'utilisateur de recommencer la saisie tant qu'il se trompe.
2. Implémenter cet algorithme dans un langage que vous connaissez.

Exercice 12

Universalité du NAND

1. Déterminer la proposition A NAND A. En déduire l'expression à l'aide du seul connecteur NAND la propostion $\overline{A}$
2. Donner la définition de A NAND B. En considérant que $A \cdot B = \overline{\overline{A \cdot B}}$, exprimer $A \cdot B$ à l'aide du seul connecteur NAND.
3. A l'aide d'une loi de De Morgan, exprimer $\overline{A + B}$. En considérant que $A + B = \overline{\overline{A + B}}$, exprimer $A + B$ à l'aide du seul connecteur NAND.
4. Pourquoi dit-on que le connecteur NAND est **universel** ?
5. *Facultatif :* faire de même mais avec le seul connecteur NOR.

Minterme

Définition

Un **minterme** de n variables booléennes est un produit de ces n variables ou de leurs complémentaires

Exemples

- Soient deux variables booléennes a et b, alors $a \cdot b$ et $a \cdot \overline{b}$ sont deux mintermes
- Soient trois variables booléennes a, b et c, alors $a\overline{b}c$ est un minterme, par contre $a\overline{b}$ ne l'est pas.

Forme canonique disjonctive

Définition

On appelle **forme canonique disjonctive** de la fonction f son écriture sous forme de somme de mintermes (**cette décomposition est unique**).

Exemples

Soient deux variables booléennes a et b

- $f(a, b) = ab + \overline{a}\overline{b}$ est sous forme canonique disjonctive
- $g(a, b) = a + \overline{a}b$ ne l'est pas.
  Sa forme canonique disjonctive est $g(a, b) = ab + a\overline{b} + \overline{a}b$

Représentation des fonctions booléennes à 2 variables

Définition : tableau de Karnaugh

Les fonctions booléennes de 2 variables a et b sont représentées par le tableau ci-dessous appelé **tableau de Karnaugh**. Chaque case représente un produit des variables a, b ou de leur complémentaire et chacun de ces produits est appelé un **minterme**.
Pour une fonction de 2 variables ils sont au nombre de 4.

a \ b	0	1
0	$\bar{a}\bar{b}$	$\bar{a}b$
1	$a\bar{b}$	ab

Représentation des fonctions booléennes à 2 variables

Pour représenter une fonction f, on met en évidence les mintermes composant f en notant 1 dans les cases pour lesquelles $f = 1$.

Exemples

- Pour $f(a,b) = ab + \overline{a}\overline{b}$, on a :

$a \backslash b$	0	1
0	1	
1		1

- Pour $g(a,b) = a + \overline{a}b$, on a :

$a \backslash b$	0	1
0		1
1	1	1

a est représenté par 2 cases adjacentes ($a\overline{b}$ et ab).

Représentation des fonctions booléennes à 2 variables

Remarque

Lorsque l'on passe d'une case à une autre case du tableau, et si une des variables seulement change d'état, les cases correspondantes sont dites **adjacentes**.

Par exemple, les cases correspondant aux mintermes ab et $\overline{a}b$ sont adjacentes mais les cases correspondant aux mintermes ab et $\overline{a}\,\overline{b}$ ne le sont pas.

$a \backslash b$	0	1
0		
1		

Cases adjacentes

$a \backslash b$	0	1
0		
1		

Cases non adjacentes

Méthode graphique pour les fonctions booléennes à 2 variables

Méthode :

Pour simplifier une expression booléenne, on remplace deux cases adjacentes par une seule variable.

Exemple :

Pour g définie par $g(a;b) = a + \overline{a}b$, on a :

$a \backslash b$	0	1
0		1
1	1	1

$g(a;b) = {\color{blue}a} + {\color{red}b}$

Exercice 13 : mot de passe

La connexion à un site Internet nécessite la saisie d'un mot de passe. Ces caractères peuvent être des lettres majuscules de l'alphabet français, ou des chiffres.
Un mot de passe est valide si l'une au moins des trois conditions suivantes est réalisée :

- Il comporte au moins trois chiffres
- Il comporte au moins cinq lettres
- Il comporte moins de trois chiffres mais au moins cinq lettres

Partie A - Reconnaître si un mot de passe est valide

1. Parmi les mots de passe suivants, quels sont ceux qui sont valides ?

   H3EXZK5   LUC230598   1ZRMK4

2. Alice veut créer un mot de passe avec quatre lettres et quatre chiffres. Ce mot de passe sera-t-il accepté ? Et un mot de passe de huit lettres ?

Partie B - Écriture d'une expression booléenne

On définit deux variables booléennes a et b de la façon suivante :

- $a = 1$ si le mot de passe contient au moins trois chiffres, sinon $a = 0$.
- $b = 1$ si le mot de passe contient au moins cinq lettres, sinon $b = 0$.

ainsi que la variable A telle que $A = 1$ si le mot de passe est valide, $A = 0$ sinon.

1. Traduire chacune des trois conditions de validité d'un mot de passe à l'aide des variables a, b. En déduire l'expression de A.
2. Représenter A avec un tableau de Karnaugh. En déduire une expression simplifiée de A.
3. Écrire l'algorithme qui teste si un mot de passe est valide.
4. Pour des raisons pratiques, on préfère tester quand le mot de passe est non valide pour redemander sa saisie. Donner l'expression de $\overline{A}$.
5. Écrire l'algorithme qui teste si un mot de passe est non valide.

Représentation des fonctions booléennes à 3 variables

Représentation de mintermes

Représentation de $\overline{a}\,\overline{b}\,\overline{c}$

$a \backslash bc$	00	01	11	10
0	1			
1				

Représentation de $a\overline{b}c$

$a \backslash bc$	00	01	11	10
0				
1		1		

Représentation des fonctions booléennes à 3 variables

Représentation d'une fonction booléenne quelconque de 3 variables

La représentation à l'aide d'un tableau de Karnaugh d'une fonction booléenne quelconque s'effectue en combinant les représentations de chaque minterme.

Pour $f(a,b,c) = abc + \overline{a}bc + \overline{a}b\overline{c}$

a \ bc	00	01	11	10
0			1	1
1			1	

Représentation des fonctions booléennes à 3 variables

Représentation des produits de deux variables ou de leurs complémentaires

Représentation de ab

a \ bc	00	01	11	10
0				
1			1	1

Représentation de $\overline{b}c$

a \ bc	00	01	11	10
0		1		
1		1		

Représentation des fonctions booléennes à 3 variables

Représentation d'une variable

Représentation de a :

$a \backslash bc$	00	01	11	10
0				
1	1	1	1	1

Représentation de $\overline{b}$:

$a \backslash bc$	00	01	11	10
0	1	1		
1	1	1		

Représentation des fonctions booléennes à 3 variables

Exercice :

Pour $g(a, b, c) = ab + \overline{a}\,\overline{c} + abc$

a \ bc	00	01	11	10
0				
1				

Pour $h(a, b, c) = abc + \overline{c} + ab$

a \ bc	00	01	11	10
0				
1				

Représentation des fonctions booléennes à 3 variables

Remarque

Pour 3 variables booléennes :

- 4 cases du tableau de Karnaugh sont dites adjacentes si elles peuvent s'écrire à l'aide d'une seule variable
- 2 cases sont dites adjacentes si elles peuvent s'écrire à l'aide d'un produit de 2 variables.

Exemples de cases adjacentes :

a \ bc	00	01	11	10
0				
1				

a \ bc	00	01	11	10
0				
1				

a \ bc	00	01	11	10
0				
1				

a \ bc	00	01	11	10
0				
1				

Représentation des fonctions booléennes à 3 variables

Exemples de cases non adjacentes :

a \ bc	00	01	11	10
0		■		
1			■	

a \ bc	00	01	11	10
0	■			
1				■

Elles ne peuvent pas s'écrire à l'aide d'un produit de 2 variables ou avec 1 variable.

Méthode graphique pour les fonctions booléennes à 3 variables

Méthode :

Pour simplifier l'écriture d'une fonction à l'aide d'un tableau de Karnaugh, on regroupe les cases adjacentes :

- par 4 et on écrit la variable correspondante
- par 2 et on écrit le produit de deux variables correspondant

Règle :

- Il faut regrouper le plus de cases adjacentes possibles à la fois (faire un « rectangle » le plus grand possible)
- Il faut que chaque case contenant un « 1 » soit au moins dans un regroupement
- Une case peut être dans plusieurs regroupements

Méthode graphique pour les fonctions booléennes à 3 variables

Exemple 1 :

Pour $f(a,b,c) = abc + \bar{a}b\bar{c} + \bar{a}bc$, on a :

a \ bc	00	01	11	10
0			1	1
1			1	

$f(a,b,c) = \textcolor{blue}{\bar{a}b} + \textcolor{red}{bc}$

Méthode graphique pour les fonctions booléennes à 3 variables

Exemple 2 :

Pour $f(a, b, c) = ab + bc + \overline{a}\,\overline{c}$, on a :

a \ bc	00	01	11	10
0	1		1	1
1			1	1

$f(a, b, c) = \color{blue}{b} + \color{red}{\overline{a}\,\overline{c}}$

Méthode graphique pour les fonctions booléennes à 3 variables

Remarque :

L'écriture de la forme simplifiée d'une fonction booléenne obtenue à l'aide d'un tableau de Karnaugh **n'est pas unique**.

Par exemple, pour la fonction f définie par le tableau de Karnaugh suivant :

a \ bc	00	01	11	10
0	1	1		1
1		1	1	

$f(a, b, c) =$
$f(a, b, c) =$

Exercice 14 : mot de passe plus complexe

La connexion à un site Internet nécessite la saisie d'un mot de passe comportant de 8 à 12 caractères. Ces caractères peuvent être des lettres majuscules de l'alphabet français, ou des chiffres, ou des caractères spéciaux (tels que &, *, /, @ etc).
Un mot de passe est valide si l'une au moins des trois conditions suivantes est réalisée :

- Il comporte au moins trois chiffres et trois caractères spéciaux
- Il comporte au moins cinq lettres
- Il comporte moins de trois chiffres mais au moins cinq lettres et trois caractères spéciaux

Partie A - Reconnaître si un mot de passe est valide

1. Parmi les mots de passe suivants, quels sont ceux qui sont valides ?

   H32EXZ&K5= LUC230598** 123(M*K#4

2. Alice veut créer un mot de passe avec quatre lettres, quatre chiffres et quatre caractères spéciaux. Ce mot de passe sera-t-il accepté ? Et un mot de passe de huit lettres ?

Partie B - Écriture d'une expression booléenne

On définit trois variables booléennes a, b et c de la façon suivante :

- $a = 1$ si le mot de passe contient au moins trois chiffres, sinon $a = 0$.
- $b = 1$ si le mot de passe contient au moins cinq lettres, sinon $b = 0$.
- $c = 1$ si le mot de passe contient au moins trois caractères spéciaux, sinon $c = 0$.

ainsi que la variable A telle que $A = 1$ si le mot de passe est valide, $A = 0$ sinon.

1. Traduire chacune des trois conditions de validité d'un mot de passe à l'aide des variables a, b et c. En déduire l'expression de A.
2. Représenter A avec un tableau de Karnaugh. En déduire une expression simplifiée de A.
3. Écrire l'algorithme qui teste si un mot de passe est valide.
4. Pour des raisons pratiques, on préfère tester quand le mot de passe est non valide pour redemander sa saisie. Donner l'expression de $\overline{A}$.
5. Écrire l'algorithme qui teste si un mot de passe est non valide.