Logique

Propositions, valeurs de vérité

- **Définition**
  Une *proposition* est une expression bien formée, du point de vue d'un certain langage, à laquelle est affectée
  clairement, par un ordinateur ou une communauté de personnes, une valeur de vérité, notée soit V, true ou 1 pour indiquer qu'elle est vraie, soit F, false ou 0 pour indiquer qu'elle est fausse.
- **Exemples**
  "Schtroumph", "il pleuvra demain", "les femmes sont supérieures aux hommes" ne sont pas des propositions françaises (la première n'est pas correctement formée selon les règles de syntaxe et de grammaire, et on ne peut pas donner de valeur de vérité aux deux suivantes, la seconde parce qu'on ne sait pas prédire l'avenir météorologique avec certitude, et je vous laisse le soin de découvrir pourquoi la dernière n'en est pas une.
-
  Par contre, "Nancy est en Meurthe-et-Moselle" est une proposition française dont la valeur de vérité est V.
- "$3 > \pi$" et "$2 + 3 = 5$" sont des propositions mathématiques dont la valeur de vérité est F pour la première, V pour la seconde. Par contre, "$4 + 5$" et "$x = 3$" n'en sont pas : $4 + 5$ est un terme, et $x = 3$ est une équation.

- Les propositions (encore appelées conditions) sont surtout utilisées en informatique à l'intérieur des structures de contrôle suivantes:
-
  - **If** *<condition>* **then** <instruction>;
  - **If** *<condition>* **then** <instruction1> **else** <instruction2>;
  - **While** *<conditionpourcontinuer>* **do** <instruction>;
  - **repeat** <instructions> **until** *<conditionarrêt>*;

Connecteurs logiques

- À partir de propositions *P*, *Q*, *R*... on peut en construire d'autres dont la valeur de vérité ne dépend que de
celles des propositions initiales. On décrit de telles constructions à l'aide de *tables de vérités*, qui donnent, en
fonction des valeurs de vérités des propositions initiales, la valeur de vérité de la construction.

Négation d'une proposition

En mathématiques $\neg P$ se lit "non P" et peut aussi se désigner par $\overline{P}$.

La négation est un connecteur logique unaire défini par la table de vérité:

P	$\neg P$
0	1
1	0

Autrement dit $\neg P$ vaut V ssi P vaut F.

En Pascal, il s'exprime avec l'opérateur "*not*" comme dans l'instruction : "**if not**(i=0) **then** dosomething;".

Remarquons que, pour toute proposition P, $\neg\neg P$ ou encore $\overline{\overline{P}}$ a la même valeur que P :

P	$\neg P$	$\neg(\neg P)$
0	1	0
1	0	1

Conjonction

La conjonction est un connecteur logique binaire défini par la table de vérité :

P	Q	$P \wedge Q$
0	0	0
0	1	0
1	0	0
1	1	1

Autrement dit $P \wedge Q$ vaut V si et seulement si P et Q valent tous les deux V. Remarquons au passage que si on attribue les valeurs 0 et 1 à F et F, $P \wedge Q$ est le *minimum* de P et Q.

En mathématiques $P \wedge Q$ se lit "P et Q".

En Pascal, il s'exprime avec l'opérateur "**and**" comme dans l'instruction :

"**if** (i=0)**and**(j=0) **then** dosomething;".

Attention à ne pas taper "textbfif i=0 **and** j=0 **then** dosomething;" car, les règles de priorité ayant été mal choisies, le compilateur tentera des calculs farfelus et finira par un message d'erreur.

EXERCICES :

- Compléter la table de vérité suivante et vérifier ainsi que $(P \wedge Q) \Leftrightarrow (Q \wedge P)$ est une tautologie signifiant que la conjonction est commutative :

P	Q	$P \wedge Q$	$Q \wedge P$	$(P \wedge Q) \Leftrightarrow (Q \wedge P)$
0	0			
0	1			
1	0			
1	1			

- Comparer les deux instructions Pascal suivantes :

- **If** (nbfacture<>0) **and** (nbimpaye/nbfacture<=0.05) **then** traiterbonclient**;**
- **If** (nbimpaye/nbfacture<=0.05) **and** (nbfacture<>0) **then** traiterbonclient**;**

On suppose que le compilateur Pascal est configuré pour évaluer systématiquement les deux propositions P et Q pour évaluer la proposition P *and* Q. Modifier la deuxième instruction pour qu'elle s'execute correctement.

Disjonction

La disjonction est un connecteur logique binaire défini par la table de vérité :

P	Q	$P \vee Q$
0	0	0
0	1	1
1	0	1
1	1	1

En mathématiques $P \vee Q$ se lit "P ou Q".

$P \vee Q$ vaut V si et seulement si l'une au moins des propositions P ou Q vaut V. Remarquons au passage que si on attribue les valeurs 0 et 1 à F et F, $P \wedge Q$ est le *minimum* de P et Q.

Attention au fait qu'il s'agit du "ou" inclusif car il n'est pas interdit que les deux propositions soient vraies. En français le "ou" peut-être inclusif comme dans la phrase : "Une entreprise recherche un stagiaire parlant anglais ou espagnol ". Il est parfois exclusif comme dans la phrase : "ce soir je vais au cinéma ou au théâtre". Il peut avoir aussi le sens d'une implication, comme dans la phrase : "mange ta soupe ou tu seras privé de dessert".

Exercices :

- Compléter la table de vérité suivante et vérifier ainsi que $(P \vee Q) \Leftrightarrow (Q \vee P)$ est une tautologie signifiant que la disjonction est commutative :

P	Q	$P \vee Q$	$Q \vee P$	$(P \vee Q) \Leftrightarrow (Q \vee P)$
0	0			
0	1			
1	0			
1	1			

- Comparer les deux instructions Pascal suivantes:

- **If** (nbfacture=0)**or** (nbimpaye/nbfacture>0.05) **then** traitermauvaisclient **;**
- **If** (nbimpaye/nbfacture>0.05) or (nbfacture=0) **then** traitermauvaisclient**;**

On suppose que le compilateur Pascal est configuré pour évaluer systématiquement les deux propositions P et Q pour évaluer la proposition P or Q. Modifier la première instruction pour qu'elle s'exécute correctement.

- Compléter la table de vérité suivante et vérifier ainsi que $P \wedge (Q \vee R) \Leftrightarrow (P \wedge Q) \vee (P \wedge R)$ est une tautologie signifiant que la conjonction est distributive par rapport à la disjonction :

P	Q	R	$Q \vee R$	$P \wedge (Q \vee R)$	$P \wedge Q$	$P \wedge R$	$(P \wedge Q) \vee (P \wedge R)$	$P \wedge (Q \vee R) \Leftrightarrow (P \wedge Q) \vee (P \wedge R)$
0	0							
0	1							
1	0							
1	1							

- Construire une table de vérité pour vérifier que $P \vee (Q \wedge R) \Leftrightarrow (P \vee Q) \wedge (P \vee R)$ est une tautologie signifiant que la disjonction est distributive par rapport à la conjonction.

Équivalence de deux propositions

En mathématiques $P \Leftrightarrow Q$ se lit "P équivaut à Q".

L'équivalence est un connecteur logique binaire défini par la table de vérité :

P	Q	$P \Leftrightarrow Q$	
0	0	1	1
0	1	0	0
1	0	0	0
1	1	1	1

Autrement dit $P \Leftrightarrow Q$ vaut V si et seulement si P et Q ont la même valeur.

D'après le paragraphe précédent, pour toute proposition P, $(\neg\neg P \Leftrightarrow P)$ vaut vrai.

Une proposition qui est vraie quelle que soit la valeur de vérité de ses composants est une *tautologie*.

En Pascal, l'équivalence s'exprime avec l'opérateur "=" comme dans l'instruction :

"**if** (i=0)=(j=0) **then** dosomething;"

mais attention à ne pas confondre cette égalité avec l'égalité numérique[3] !

EXERCICE : Quels sont, en fonction du contenu des variables i et j, les cas où l'instruction "dosomething" est exécutée ?

Implication

L'implication est un connecteur logique binaire défini par la table de vérité :

P	Q	$P \Rightarrow Q$
0	0	1
0	1	1
1	0	0
1	1	1

En mathématiques $P \Rightarrow Q$ se lit "P implique Q".

Autrement dit $P \Rightarrow Q$ vaut V si et seulement si Q vaut V *lorsque P vaut V*. Remarquez que la valeur de Q n'a pas d'influence lorsque P est faux. Par exemple la phrase "Quand les poules auront des dents je serai ministre de l'Éducation" est parfaitement vraie car pour me contredire il faudrait que les poules aient des dents et que je ne sois pas ministre !

De la même façon, il faut bien faire attention au fait que l'implication $P \Longrightarrow Q$ *ne dit absolument rien sur la valeur de vérité de P*. En particulier, si P est fausse, alors l'implication $P \Longrightarrow Q$ est vraie, quelle que soit la valeur de vérité de Q. Ainsi, "si $4 < 0$, alors $1 = 2$" est une implication vraie, de même que "si $1 = 2$, alors $4 > 0$". Cela conduit à la constatation suivante : du faux, on peut déduire n'importe quoi.

On confond souvent l'implication avec une relation de causalité : le fait que $P \Longrightarrow Q$ est vraie serait compris comme entraînant que *Q découle de P*. Il n'en est rien en logique pure. Par contre, une sorte de réciproque de ce raisonnement est vraie : on l'appelle **"règle d'inférence"**, ou "modus ponens" : pour toutes propositions P et Q, on a

$$(P \wedge (P \Longrightarrow Q)) \Longrightarrow Q$$

Propriétés des connecteurs logiques

1-Commutativité et associativité de $\lor$ et $\land$

Commutativité

Les phrases “il fait beau et chaud” et “il fait chaud et beau” sont équivalentes, de même que les phrases “il fait beau ou chaud” et “il fait chaud ou beau”.

Plus généralement, pour toutes propositions P et Q,

$$\boxed{(P \vee Q) \iff (Q \vee P)} \quad \text{et} \quad \boxed{(P \wedge Q) \iff (Q \wedge P)}$$

La démonstration de ces deux tautologies se fait simplement en constatant que les tables de vérité ne changent pas lorsqu'on permute leurs deux premières colonnes.

Associativité

Lorsqu'on rencontre une expression de la forme $P \wedge Q \wedge R$, qu'on peut interpréter d'au moins deux façons différentes : on peut d'abord calculer $U = P \wedge Q$, puis calculer $U \wedge R$, ou bien calculer d'abord $V = Q \wedge R$, puis calculer $P \wedge V$.

L'associativité des relations $\wedge$ et $\vee$ nous apprend que l'ordre n'est pas important : plus précisément, pour toutes propositions P, Q et R,

$$\boxed{(P \vee Q) \vee R \iff P \vee (Q \vee R)} \quad \text{et} \quad \boxed{(P \wedge Q) \wedge R \iff P \wedge (Q \wedge R)}$$

Ces deux propriétés permettent de calculer une expression de la forme $P_1 \wedge P_2 \wedge \cdots \wedge P_n$ de n'importe quelle façon : échanger la position des propositions, et commencer par n'importe quel connecteur.

Double distributivité

Si on vous "pour entrer dans le château, ouvrez la porte en bois, et la porte de gauche ou celle de droite", vous avez deux façons d'entrer dans le château : ouvrir la porte en bois et la porte de gauche, ou bien ouvrir la porte en bois et la porte de droite.

Ceci provient des tautologies suivantes : pour toutes propositions P, Q et R :

$$P \vee (Q \wedge R) \iff (P \vee Q) \wedge (P \vee R) \quad \text{et} \quad P \wedge (Q \vee R) \iff (P \wedge Q) \vee (P \wedge R)$$

Ces relations peuvent être vues comme des relations de distributivité de $\vee$ par rapport à $\wedge$, et de $\wedge$ par rapport à $\vee$, de la même façon que la multiplication est distributive par rapport à l'addition.

Élément neutre

Soit $\mathcal{V}$ une proposition vraie, et $\mathcal{F}$ une proposition fausse. Alors, pour toute proposition P :

$$\boxed{P \wedge \mathcal{V} = \mathcal{P}} \qquad \boxed{P \wedge \mathcal{F} = \mathcal{F}} \qquad \boxed{P \vee \mathcal{V} = \mathcal{V}} \qquad \boxed{P \vee \mathcal{F} = \mathcal{P}}$$

On dit que $\mathcal{V}$ est *l'élément neutre* du connecteur $\wedge$, et *l'élément absorbant* du connecteur $\vee$.

Exercices :

- Énoncez une phrase similaire pour $\mathcal{F}$.
- Démontrez ces affirmations.

Loi de De Morgan

La négation de “il faut beau et chaud” n’est pas “il faut moche et froid”[5] mais “il faut moche **ou** froid”.

Les lois de De Morgan établissent ces tautologies :

$$\boxed{\neg(P \wedge Q) \iff (\neg P \vee \neg Q)} \quad \text{et} \quad \boxed{\neg(PQ) \vee \iff (\neg P \wedge \neg Q)}$$

EXERCICES :

- Vérifiez ces tautologies à l’aide de tables de vérité.
- Quel est la négation de la phrase “il est beau, riche et intelligent” ?
- Expliquer comment on peut éliminer toutes les conjonctions d’une formule logique. On peut ainsi ne l’écrire qu’avec deux connecteurs : la négation et la disjonction.

Principe de dualité

Les lois de De Morgan ont une conséquence bien pratique : le principe de dualité. Ce principe permet, à partir d'une identité logique (une tautologie), d'en construire une autre. Pour cela, il suffit d'échanger les rôles de $\vee$ et $\wedge$ d'une part, et de V et F d'autre part.

Par exemple, à partir de la distributivité de $\vee$ par rapport à $\wedge$:

$$P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$$

on déduit l'*identité duale* :

$$P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$$

qui exprime la distributivité de $\wedge$ par rapport à $\vee$. Bien pratique pour limiter le travail de démonstration !

Algèbre booléenne

$a + a b = a$

On considère un ensemble comportant deux éléments (il s'agit donc d'une paire) que l'on note **0** et **1** (il sont sans rapport avec les nombres 0 et 1).
On définit dans cet ensemble une addition et une multiplication comme suit :

a	b	$a+b$	$a.b$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

$a.(a+b)$

$a.a + a.b$

$a + ab = a$

$a+1=1$ $a.0=0$ abso	$a+b=b+a$ $a.b=b.a$ comutati
$a+a=a$ $a.a=a$ idempotence	$a+0=a$ $a.1=a$ neutre
Morgan $\overline{a+b}=\bar{a}.\bar{b}$ $\overline{a.b}=\bar{a}+\bar{b}$	$a+b.c=(a+b).(a+c)$ $a.(b+c)=a.b+a.c$ dist
absorbtion $a+a.b=a$ $a.(a+b)=a$	$a+\bar{a}=1$ $a.\bar{a}=0$ complimant
associativ $(a+b)+c=a+(b+c)$ $(a.b).c=a.(b.c)$	$\bar{\bar{a}}=a$ négation

a	b	a.b	a+b
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Formes canoniques d'une fonction booléenne

Soit $(E;+;\times;^{-})$ une algèbre de Boole.

Une fonction booléenne *f* de *n* variables booléennes de E^n dans E est une application qui à tout *n*-uplet de E fait correspondre un élément de E construit à l'aide des opérations booléennes.

Exemples :

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

$$g(a,b,c,d) = abc + bc\bar{d} + ad + b$$

Un « minterme » de *n* variables booléennes est un produit comportant *n* facteurs, chaque facteur correspondant à une variable donnée ou à son complémentaire.

Un « maxterme » de *n* variables booléennes est une somme comportant *n* termes, chaque terme correspondant à une variable donnée ou à son complémentaire.

Formes canoniques disjonctive et conjonctive d'une fonction booléenne

Soit f une fonction booléenne de E^n dans E.
Il est possible d'écrire f de façon unique sous la forme d'une somme de mintermes.
Cette somme est appelée « forme disjonctive » de f.
De façon analogue, il est possible d'écrire f sous la forme d'un produit de maxtermes.
Ce produit est appelé « forme conjonctive » de f.

Exemple.
Considérons la fonction booléenne : $f(a,b,c) = a\overline{b}$.
On a :

$$\begin{aligned}
f(a,b,c) &= a\overline{b}1 && (1 \text{ est élément neutre de la multiplication}) \\
&= a\overline{b}(c+\overline{c}) && (\text{principe du tiers exclus}) \\
&= a\overline{b}c + a\overline{b}\overline{c} && (\text{distributivité})
\end{aligned}$$

$a\overline{b}c + a\overline{b}\overline{c}$ est la forme canonique disjonctive de la fonction f.

Par ailleurs, on a :

$$\begin{aligned}
f(a,b,c) &= a\overline{b} + c\overline{c} \\
&= (a\overline{b}+c)(a\overline{b}+\overline{c}) \\
&= (a+c)(\overline{b}+c)(a+\overline{c})(\overline{b}+\overline{c}) \\
&= ((a+c)+b\overline{b})((\overline{b}+c)+a\overline{a})((a+\overline{c})+b\overline{b})((\overline{b}+\overline{c})+a\overline{a}) \\
&= (a+c+b)(a+c+\overline{b})(\overline{b}+c+a)(\overline{b}+c+\overline{a})(a+\overline{c}+b)(a+\overline{c}+\overline{b})(\overline{b}+\overline{c}+a)(\overline{b}+\overline{c}+\overline{a}) \\
&= (a+b+c)(a+b+\overline{c})(a+\overline{b}+c)(a+\overline{b}+\overline{c})(\overline{a}+\overline{b}+c)(\overline{a}+\overline{b}+\overline{c})
\end{aligned}$$

Tables de Karnaugh

- Une table de Karnaugh est une méthode inventée par Maurice Karnaugh en 1954 et qui sert à simplifier des équations logiques ou à trouver l'équation logique correspondant à une table de vérité. La méthode utilisée est graphique. Elle fonctionne très bien avec 3 ou 4 variables, beaucoup moins bien avec 5 ou 6 variables, et plus du tout au-delà !
- Soit la table de vérité de S suivante avec les variables A, B, C et D :

A	B	C	D	S
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1

La table de Karnaugh correspondante se présente ainsi :

S CD AB	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	0	1	1	1
10	1	0	1	1