Chapitre 1 : Calcul propositionnel

La logique propositionnel est appelé aussi booléenne est la plus simple logique, elle ne comporte que des variables et des connecteurs logiques. Comme toute logique, elle est caractérisé par :

- Une syntaxe : la façon de construire les formules ;
- Une sémantique : la signification des formules ;
- Un système de preuve : comment démontrer les formules.

I- Syntaxe

1-Définition d'une proposition

Une proposition est un énoncé déclaratif qui est soit vrai soit faut mais pas les deux à la fois (C'est le principe du **tiers-exclu**).

Exemple1 :

1. Alger est la capitale d'Algérie.
2. 1 + 1 = 2
3. Un carré a 4 angles droits
4. il pleut
5. 2 + 2 = 3.
6. $3 > \pi$
7. Il existe un réel x tel que $x^2 < 0$

Toutes les phrases déclaratives précédentes sont des propositions : Les trois premières phrases sont des propositions dont la valeur de vérité est vraie alors que les trois dernières sont des propositions mathématiques dont la valeur de vérité est fausse.

Exemple2 :

1. Quelle heure est-il?
2. Lisez-le attentivement.
3. Qu'elle est belle !
4. x=3
5. x + 1 = 2.
6. Il pleuvra demain
7. 4 + 5

- Les phrases précédentes ne sont pas des propositions.

2-Variables propositionnelles

On appelle une variable propositionnelle, une variable qui représente une **proposition atomique**
On utilise dans ce cours les lettres pour designer les variables propositionnelles auxquelles on peut ajouter un indice entier (p, q , r, p_0, p_1, p_2, . . .).

Exemple :
p : il pleut
q : la route est mouillée
r : il neige
On désignera donc par $\mathcal{P}$ l'ensemble non vide des variables propositionnelles tel que $\mathcal{P}$ = {p, q, r...........}

De nouvelles propositions, appelées **propositions composées**, sont formés à partir des propositions existantes en utilisant des connecteurs logiques.

3-Connecteurs logiques

Il existe dans la littérature différentes manières de les représenter, et celle que nous donnons ci-dessous n'en est qu'une parmi d'autres. On donne donc les 5 connecteurs suivants : $\neg$ (unaire), $\vee$, $\wedge$, $\longrightarrow$, $\longleftrightarrow$ (opérateurs binaires)

1-¬ est appelé symbole de négation, on le prononce “non”.

exp :

La négation de la proposition **p** : ‘ 24 est un multiple de 2’ est : ’24 **n’**est **pas** un multiple de 2’ et elle s’écrit : **¬p**

nb : si p est vrai alors **¬p** est fausse et vise vers ca

2- ∧ est appelé symbole de conjonction, on le prononce “et”.

exp :

Considérons les deux propositions p et q suivantes : **p** : « 10 est divisible par 2 », **q** : « 10 est divisible par 3 ». 10 est divisible par 2 **et** 10 est divisible par 3 s’écrit : **p ∧ q**

nb : p ∧ q est vrai ssi les deux propositions sont vraies simultanément et faux dans tous les autres cas.

3- ∨ est appelé symbole de disjonction, on le prononce “ou”.

exp :

10 est divisible par 2 **ou** 10 est divisible par 3 s’écrit : **p ∨ q**

nb : p ∨ q est vraie ssi une ou les deux propositions sont vraies.

4- ⟶ est appelé symbole d’implication (si.. alors), on le prononce “implique”.

exp :

S’il pleut **alors** la route est mouillée s’écrit : **p ⟶ q**

nb :

- On dit que p est une **condition suffisante** pour q.
- q ⟶p s’appelle **l’implication réciproque** de p ⟶q.
- p ⟶q est vraie ssi les deux propositions sont vraies et lorsque p est fausse (peut importe la valeur de vérité de q)

5- ⟷ est appelé symbole d’équivalence, ou encore symbole de double implication, on le prononce “équivaut à” ou encore “si et seulement si”.

exp :

s : n est impair **t** : n^2 est impair n est impair **si et seulement si** n^2 est impair s’écrit : **s⟷t**

nb : s⟷t est vraie si et seulement si s et sont toutes les deux vraies ou toutes les deux fausses.

<u>**Table de vérité des connecteurs**</u>

On résume ceci dans la table de vérité suivante:

p	¬p
0	1
1	0

p	q	p∧q	p∨q	p⟶q	p⟷q
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

4- Alphabet du calcul propositionnel

L'alphabet du CP est composé :

- des variables propositionnelles (symboles de propositions, propositions élémentaires);
- des connecteurs : ¬, ∧, ∨, ⟶,⟷ ;
- des séparateurs : les parenthèses : il s'agit simplement de la parenthèse ouvrante **(** et de la parenthèse fermante **)**

$\mathcal{A} = \mathcal{P} \cup \{\neg, \vee, \wedge, \longrightarrow, \leftrightarrow\} \cup \{), (\}$

$= \{p, q, r \ldots\} \cup \{\neg, \vee, \wedge, \longrightarrow, \leftrightarrow\} \cup \{), (\}$

$= \{p, q, r, \ldots, \neg, \vee, \wedge, \longrightarrow, \leftrightarrow,), (\}$

À partir de l'alphabet, on peut construire plusieurs **mots** certain d'entre eux sont **des formules propositionnelles** et d'autres ne sont pas.

5- L'ensemble des formules propositionnelles

a- Première définition (Définition descendante ou par le haut) : l'ensemble $\mathcal{F}$ des formules propositionnelles construites sur $\mathcal{P}$ est le plus petit sous ensemble de A^* qui :

- (1) Contient $\mathcal{P}$;
- (2) À chaque fois qu'il contient F, il contient aussi ¬F ;
- (3) À chaque fois qu'il contient F et G, il contient aussi (F*G) ; $* \in \{\vee, \wedge, \longrightarrow, \leftrightarrow\}$

L'ensemble $\mathcal{F}$ des formules propositionnelles est donc l'intersection de toutes les parties de A^* qui vérifient les propriétés (1), (2) et (3).
Cette intersection est non vide car A^* lui-même vérifie ces propriétés

$$\mathcal{F} = \bigcap_{\substack{Y \subseteq A^* \\ Y \text{ vérifie } (1),(2) et\ (3)}} Y$$

Exemple: soit p, q, r, s et t sont des éléments de $\mathcal{P}$:

Etudions le mot : F= (((p ^ q) ⟶s) ∨ ¬(s ⟷ t))
Ligne 1 : p, q, s et t sont des formules d'après la règle 1
Ligne 2 : (p ^ q) est une formule d'après la ligne 1 et la règle 3
Ligne 3 : ((p ^ q) ⟶s) est une formule d'après les lignes 1 et 2 et la règle 3
Ligne 4 : (s ⟷ t) est une formule d'après la ligne 1 et la règle 3
Ligne 5 : (¬(s ⟷ t) est une formule d'après la ligne 4 et la règle 2
Ligne 6 : F est une formule d'après les lignes 3 et 5 et la règle 3

Voici d'autres exemples de formules :

p
¬¬p
((¬p ∧ q) ⟶ r)
¬ (p ∨ q)
((¬ (p ∨ q) ⟶ ¬¬q) ^ r)
(((p ∧ (¬q⟶¬p)) ∧(¬q ∨ ¬r)) → (r⟶¬p))

Et voici des exemples qui ne sont pas des formules :

p ∧ q
¬(p)
pq
∧p¬q
→q ∨r
(p⟶q ∨ r)

(p⟶q,r)

(p ∧ q ∧ r)

∀p (¬p ∨ p)

((p ∧ (q⟶r)) ∨ (¬p⟶ (q ∧r)) ∧ (¬p ∨ q))

-Longueur d'une formule : la longueur d'une formule est sa longueur en tant que mot. On la note ℓ(F) i.e.

ℓ (F) est le nombre de lettres dans F.

NB : Il n'y pas de formule de longueur 0 (car la plus petite formule est la proposition élémentaire qui a une longueur égale à 1)

Exemples :

1) F= p ; ℓ (F)=1
2) F= ¬(p↔ q) ; ℓ (F)=6

Remarque :

En appliquant strictement la définition ci-dessus de « Formules », nous dirions que p ↔ q → r n'est pas une formule bien formée. En effet, le parenthèsage est absent. On ne sait pas s'il s'agit de la formule ((p ↔ q) → r) ou s'il s'agit de (p ↔ (q → r)). Les parenthèses sont un moyen de lever l'ambiguïté.

Il en existe un autre qui consiste à donner à chaque connecteur **un ordre de priorité**. **Les connecteurs sont traditionnellement classés de la façon suivante (par priorité décroissante des connecteurs) :**

¬, ∧, ∨, →, ↔.

Dans le cas où deux connecteurs ont même priorité, et en l'absence de parenthèses, l'associativité se fait de gauche à droite. On peut ainsi se permettre d'omettre les parenthèses. Par exemple, la formule p → q ↔ ¬r doit se lire ((p → q) ↔ ¬r).

A l'avenir, lorsque nous parlerons de formules bien formées, nous inclurons également les formules dont le parenthèsage est partiellement ou complètement implicite. L'ensemble des **formules bien formées** ainsi défini forme **le langage** de la logique propositionnelle.

b- Deuxième définition (Définition ascendante ou par le bas) :

Il est possible de donner de l'ensemble $\mathcal{F}$ une description plus explicite, pour cela on va définir par récurrence une suite $(F_n)_{n\in N}$ de parties de A^*.

On pose :

$$\begin{cases} F_0 = P \\ et\ pour\ chaque\ entier\ n;\ F_{n+1} = F_n \cup \{\neg F, F \in F_n\} \cup \left\{ \begin{array}{c} (F * G), \\ F, G \in F_n, * \in \{\vee, \wedge, \longrightarrow, \leftrightarrow\} \end{array} \right. \end{cases}$$

$$\mathcal{F} = \bigcup_{n \in N} F_n$$

On remarquera que la suite $(F_n)_{n\in N}$ est croissante (pour n≤ m, on a $F_n \subseteq F_m$)

Cette définition va nous permettre de définir la hauteur d'une formule.

- Hauteur d'une formule : la hauteur d'une formule F est le plus petit des entiers n tel que $F \in F_n$.

h(F)= min {n / F∈ $\mathcal{F}_n$}

Exemples :

1) F_1=p

$\mathcal{F}_0$= {p} ; h[F_1]=0

2) F_2= ((p∧¬q) ⟶¬r)

$\mathcal{F}_0$= {p, q, r}

$\mathcal{F}_1$= $\mathcal{F}_0$ ∪ {¬q, ¬r}

$\mathcal{F}_2$= $\mathcal{F}_1$ ∪ {(p∧¬q)}

$\mathcal{F}_3 = \mathcal{F}_2 \cup \{((p \wedge \neg q) \longrightarrow \neg r)\}$; $h[F_2]=3$

<u>6- Sous formules</u>

A chaque formule $F \in \mathcal{F}$, on associe un sous-ensemble SF(F) de F, appelé ensemble des sous-formules de F, défini par induction de la façon suivante :

- si $F \in \mathcal{P}$, $SF(F) = \{ F \}$;
- si $F = \neg G$, $SF(F) = \{ F \} \cup SF(G)$;
- si $F = (G * H)$ tel que $* \in \{\wedge, \vee, \longrightarrow, \leftrightarrow\}$, $SF(F) = \{ F \} \cup SF\{G\} \cup SF\{ H \}$

Exemple : Considérons la formule :

$F = (\neg p \wedge \neg\neg(p \wedge (q \longrightarrow p)))$.

- ❖ Les sous-formules de F sont (intuitivement, on commence par les variables propositionnelles jusqu'à arrivé à F):

$SF(F) = \{\ p, q, \neg p, (q \longrightarrow p), (p \wedge (q \longrightarrow p)), \neg(p \wedge (q \longrightarrow p)), \neg\neg(p \wedge (q \longrightarrow p)), (\neg p \wedge \neg\neg(p \wedge (q \longrightarrow p)))\}$

- ❖ En appliquant la définition, les sous-formules de F sont:

$SF(F) = \{F\} \cup SF\{\neg p\} \cup SF\{\neg\neg(p \wedge (q \longrightarrow p))\}$

$= \{F\} \cup \{\neg p\} \cup SF\{p\} \cup \{\neg\neg(p \wedge (q \longrightarrow p))\} \cup SF\{\neg(p \wedge (q \longrightarrow p))\}$

$= \{F\} \cup \{\neg p\} \cup \{p\} \cup \{\neg\neg(p \wedge (q \longrightarrow p))\} \cup \{\neg(p \wedge (q \longrightarrow p))\} \cup SF\{(p \wedge (q \longrightarrow p)\}$

$= \{F\} \cup \{\neg p\} \cup \{p\} \cup \{\neg\neg(p \wedge (q \longrightarrow p))\} \cup \{\neg(p \wedge (q \longrightarrow p))\} \cup \{(p \wedge (q \longrightarrow p)\} \cup SF\{p\} \cup SF\{(q \longrightarrow p)\}$

$= \{F\} \cup \{\neg p\} \cup \{p\} \cup \{\neg\neg(p \wedge (q \longrightarrow p))\} \cup \{\neg(p \wedge (q \longrightarrow p))\} \cup \{(p \wedge (q \longrightarrow p)\} \cup \{p\} \cup \{(q \longrightarrow p)\} \cup SF\{q\} \cup SF\{p\}$

$= \{F\} \cup \{\neg p\} \cup \{p\} \cup \{\neg\neg(p \wedge (q \longrightarrow p))\} \cup \{\neg(p \wedge (q \longrightarrow p))\} \cup \{(p \wedge (q \longrightarrow p)\} \cup \{p\} \cup \{(q \longrightarrow p)\} \cup \{q\} \cup \{p\}$

$= \{p, q, \neg p, (q \longrightarrow p), (p \wedge (q \longrightarrow p), \neg(p \wedge (q \longrightarrow p)), \neg\neg(p \wedge (q \longrightarrow p)), (\neg p \wedge \neg\neg(p \wedge (q \longrightarrow p)))\}$

<u>-Sous formule strict :</u>

On dit que G est une sous-formule stricte de F si G est une sous-formule de F qui n'est pas F i.e. toutes les sous-formules de F sont des sous-formules strictes sauf F.

7- <u>Arbre de décomposition</u> (représentation arborescente des formules)

(A)Procédure de décomposition

1- On décompose la formule propositionnelle F au niveau du connecteur principal. Deux cas peuvent se présenter :
- ➢ (a) soit $F = \neg G$ et dans ce cas il part de F une seule branche qui aboutit immédiatement au niveau inferieur au noud G.
- ➢ (b) ou bien $F = (G * H)$ tel que $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ et dans ce cas il part de F deux branches qui aboutissent au niveau inferieur au deux nœuds G et H.

2- On répète **(1)** pour les nouveaux nœuds obtenus jusqu'à avoir des variables propositionnelles.

Ou bien

(B)

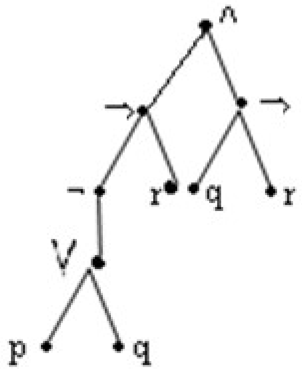
L'arbre A_F de la formule F est défini par récurrence sur F :

- si $F=p$, A_F : •p
- si $F = \neg G$, A_F : ¬

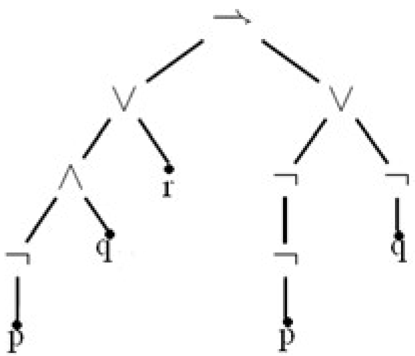
  A_G
- si $F = (G*H)$, A_F : * $\quad * \in \{\vee, \wedge, \longrightarrow, \leftrightarrow\}$

  $A_G \quad A_H$

Exemple1 : F_1= ((¬ (p∨q) ⟶r) ∧(q⟶r))



Exemple2 :F2= ((¬p∧q)∨r)⟶(¬¬p∨¬q))



b-Hauteur d'un arbre : la hauteur d'un arbre est le nombre max de branches partant de la racine jusqu'aux extrémités de l'arbre.

$h(A_{F1})$ =max(4,4,2,2,2)=4

$h(A_{F2})$=max(4,3,2,4,3)=4

Remarques :

1-Il est important de noter que toute formule bien formée a un et un seul arbre de décomposition.

2- Il est facile de vérifier que les sous formules d'une formule sont exactement celles qui figurent aux nœuds de son arbre de décomposition (Sous Formules de F= Sous Arbres de F).

3- La hauteur d'une formule est égale à la hauteur de son arbre de décomposition.

8- <u>Démonstration par induction sur l'ensemble des formules</u>

Soit Q une propriété qu'on veut démontrer qu'elle est vérifiée par toute formule F∈ $\mathcal{F}$. On montre que :

1-Q(F) est vérifiée, ∀F∈ $\mathcal{P}$

2-Si Q(F) est vérifiée alors Q(¬F) est vérifiée

3-si Q(F) est vérifiée et Q(G) est vérifiée alors Q(F*G) est vérifiée *∈{∧,∨,⟶,⟷}

Exercice :

Démontrez **par induction** que le nombre de parenthèses ouvrantes dans une formule est égal au nombre de parenthèses fermantes :

Solution :

Posons :

o : le nombre de parenthèses ouvrantes dans F

f : le nombre de parenthèses fermantes dans F

Q : o[F]=f[F]

1) ∀F∈ $\mathcal{P}$, o[F]=0 =f[F]

donc ∀F∈ $\mathcal{P}$, o[F]=f[F]=0 par conséquent Q est vérifiée

2) $o[F]=f[F] \Longrightarrow o[\neg F]=f[\neg F]$
   $\forall F\in \mathcal{F}$, $o[\neg F]= o[F]= f[F] =f[\neg F]$
   donc $\forall F\in \mathcal{F}$, $o[\neg F]=f[\neg F]$ par conséquent Q est vérifiée
3) $o[F]=f[F]$ et $o[G]=f[G] \Longrightarrow o[(F*G)]=f[(F*G)]$
   $\forall F, G\in \mathcal{F}$, $o[(F*G)]=1+ o[F]+ o[G]= 1+f[F]+ f[G]= f[(F*G)]$
   donc $\forall F\in \mathcal{F}$, $o[(F*G)]=f[(F*G)]$ par conséquent Q est vérifiée

Ainsi, $\forall F\in \mathcal{F}$, $o[F]= f[F]$

9 Substitution d'une formule

a- Substitution simple :

-une occurrence de la variable p dans une formule F est la donnée de cette variable et d'une place oû elle apparait dans F.

-Soient F, G deux formules propositionnelles et p une variable propositionnelle de F. F [G / p] (ou bien F [p ← G]) est la formule obtenue en substituant (remplaçant) toutes les occurrences de la variable p par la formule G dans F.

La formule F[G/p] est définie par induction sur la formule F :
(B)- Si F est la variable propositionnelle p, alors F[G/p] est la formule G;
(B)- Si F est une variable propositionnelle q, avec $q\neq p$, alors F[G/p] est la formule F ;
(I)- Si F est de la forme ¬H, alors F[G/p] est la formule ¬H[G/p] ;
(I)- Si F est de la forme (F1*F2), alors F(G/p) est la formule (F1[G/p]* F2[G/p]) ;$*\in\{\wedge,\vee,\longrightarrow,\longleftrightarrow\}$

Exemple

F : $(p\longrightarrow(q\vee p))$ // La variable p a 2 occurrences dans F, q a une seule occurrence dans F
G : $(q\longrightarrow r)$

$$F(G / p) = F[p\leftarrow G] = (G \longrightarrow(q\vee G)) \\ =((q \longrightarrow r)\longrightarrow(q \vee(q\longrightarrow r)))$$

b- substitution simultanée

On peut étendre la substitution à plusieurs formules.

Etant donnée une formule F[p1,p2,…..pn,,q1,q2,……qm] et n formules G1,…….,Gn

F [G1/ p1, G2/ p2, …, Gn/ pn] est la formule obtenue en substituant respectivement les formules G1, G2 , …, Gn à toutes les occurrences des variables p1, p2, …, pn dans F.

-Si F est une variable propositionnelle alors F[G1/p1 ,G2/p2 …Gn/pn] est égale à la formule Gk si F=pk (1<=k<=n) et à F si $F\notin$[p1,p2…..pn]
-Si F est de la forme ¬H, alors F[G1/p1 ,G2/p2 …Gn/pn] est la formule ¬H [G1/p1 ,G2/p2 …Gn/pn];
-Si F est de la forme (F1*F2), alors F[G1/p1 ,G2/p2 …Gn/pn] est la formule (F1[G1/p1 ,G2/p2 …Gn/pn] * F2[G1/p1 ,G2/p2 …Gn/pn]) ;$*\in\{\wedge,\vee,\longrightarrow,\longleftrightarrow\}$

NB : Les pi sont différents : F[G1/P, G2/P] n'est pas une substitution

Exemples

F: $(p \wedge q)$, G : $(q \vee r)$, H : $(p \wedge r)$

- F [G/p, H/q] = $(G \wedge H) = ((q \vee r) \wedge (p \wedge r))$
- F [H/r] = F= $(p\wedge q)$

Remarque :

La substitution simultanée (remplacement en parallèle) est différente de la substitution séquentielle (remplacement en série)

F [G1/p1 ,G2/p2] $\neq$ (F [G1/p1]) [G2/p2]

substitution simultanée substitution séquentielle

Exemples :

F : $(p \wedge q)$ G : $(p \vee q)$ H : $(p \longrightarrow q)$

• $F[G/p, H/q] = ((p \vee q) \wedge (p \rightarrow q))$

• $(F[G/p])[H/q] = ((p \vee q) \wedge q)[H/q]$
$= ((p \vee (p \rightarrow q)) \wedge (p \rightarrow q))$

• $F[H/q, G/p] = ((p \vee q) \wedge (p \rightarrow q))$

• $(F[H/q])[G/p] = (p \wedge (p \rightarrow q))[G/p]$
$= ((p \vee q) \wedge ((p \vee q) \rightarrow q))$

• Pour la substitution simultanée l'ordre n'est pas important
$F[G/p, H/q] = F[H/q, G/p]$
• Pour la substitution séquentielle l'ordre est important
$(F[G/p])[H/q]] \neq (F[H/q])[G/p]$

II- SEMANTIQUE

Il faut maintenant un moyen de préciser le sens d'une formule. Le sens d'une formule est sa valeur de vérité (i.e. déterminer si une formule est vraie ou fausse).
$\{0,1\}$ représente l'ensemble des valeurs de vérité : La valeur 0 désigne 'faux' et la valeur 1 désigne 'vrai'.
En logique propositionnelle, la sémantique a deux principes :

- On a deux valeurs de vérité 0 et 1.
- La valeur de vérité d'une formule est déduite à partir des valeurs de vérité de ses variables propositionnelles et des connecteurs qui relient ces variables. Ces valeurs sont données par une valuation

1-<u>valuation :</u> (distribution de valeur, évaluation)

Une valuation est une application de l'ensemble des variables propositionnelles $\mathcal{P}$ vers l'ensemble $\{0,1\}$.
$V : \mathcal{P} \rightarrow \{0,1\}$: V affecte à chaque VP une valeur de vérité 1 ou 0.
Si $\mathcal{P}$ est de cardinalité n, on a 2^n valuations possibles.

<u>2-Interprétation:</u>

Pour toute valuation V, il existe une application unique $\bar{V}$ définie sur $\mathcal{F}$ qui **coïncide avec V sur** $\mathcal{P}$ (qui prolonge V à l'ensemble $\mathcal{P}$: si F=p alors $\bar{V}[p]=V[p]$) et vérifie les propriétés suivantes :

1) $\bar{V}[\neg F]=1$ ssi $\bar{V}[F]=0$, $\forall F \in \mathcal{F}$
2) $\bar{V}[(F \wedge G)]=1$ ssi $\bar{V}[F]=\bar{V}[G]=1$, $\forall F,G \in \mathcal{F}$
3) $\bar{V}[(F \vee G)]=0$ ssi $\bar{V}[F]=\bar{V}[G]=0$, $\forall F,G \in \mathcal{F}$
4) $\bar{V}[(F \rightarrow G)]=0$ ssi $\bar{V}[F]=1$ et $\bar{V}[G]=0$, $\forall F,G \in \mathcal{F}$
5) $\bar{V}[(F \leftrightarrow G)]=1$ ssi $\bar{V}[F]=\bar{V}[G]$, $\forall F,G \in \mathcal{F}$