

## Chapitre II : Logique propositionnelle : sémantique

Goulven GUILLOU

UBO – Département Informatique



1 / 29

## Rappel : syntaxe des formules de la logique des propositions

Soit  $\mathcal{A} = \{p, q, r, s, \dots\}$  l'ensemble des variables propositionnelles, l'ensemble **Prop** est le plus petit ensemble construit par application d'un nombre fini de règles suivantes :

- Si  $x \in \mathcal{A}$  alors  $x \in \mathbf{Prop}$ .
- Si  $p \in \mathbf{Prop}$  alors  $\bar{p} \in \mathbf{Prop}$ .
- Si  $p$  et  $q$  sont dans **Prop** alors  $(p \vee q) \in \mathbf{Prop}$ .
- Si  $p$  et  $q$  sont dans **Prop** alors  $(p \wedge q) \in \mathbf{Prop}$ .
- Si  $p$  et  $q$  sont dans **Prop** alors  $(p \implies q) \in \mathbf{Prop}$ .
- Si  $p$  et  $q$  sont dans **Prop** alors  $(p \iff q) \in \mathbf{Prop}$ .

2 / 29

## Sémantique

Pour l'instant les formules sont de simples constructions syntaxiques. Leur donner une sémantique signifie leur donner un sens. On interprète les formules en termes de *valeurs de vérité* (vrai, faux ou 1, 0).

**Définition** Une *valuation* est une fonction de l'ensemble des variables propositionnelles dans  $\{0, 1\}$ .

Pour connaître la valeur de vérité d'une formule, il suffit de connaître les *tables de vérité* des connecteurs.

3 / 29

## Tables de vérité des connecteurs

$G$	$\neg G$
0	1
1	0

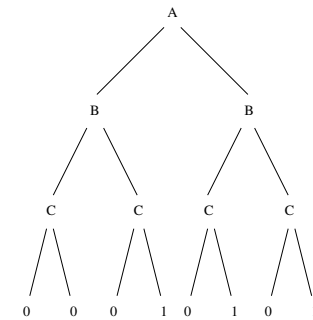
$G$	$H$	$G \wedge H$	$G \vee H$	$G \implies H$	$G \iff H$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

4 / 29

Pour  $(A \vee B) \wedge C$  on a huit interprétations possibles.

A	B	C	$(A \vee B) \wedge C$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Une autre manière de présenter une table de vérité.  
Toujours pour  $(A \vee B) \wedge C$  :

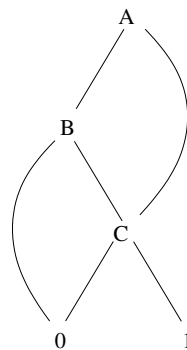


Les variables sont ordonnées.

A partir d'une variable, aller à gauche c'est lui donner la valeur 0 et à droite la valeur 1.

Une feuille donne la valeur de l'interprétation.

Graphe acyclique construit à partir de l'arbre de Shannon.



Plus compact qu'un arbre de Shannon.  
Egalement canonique (modulo l'ordre des variables).  
A la base de **bddc**.

On part de l'arbre de Shannon de la proposition.

- On identifie récursivement les noeuds ayant deux fils identiques à leurs fils.
- On fusionne les noeuds identiques.

On obtient ainsi un graphe enraciné, orienté et acyclique tel que :

- deux noeuds n'ont aucun arc sortant et sont étiquetés par 0 et 1.
- les autres noeuds ont exactement deux arcs sortants.

L'interprétation est la même que pour un arbre de Shannon.  
Lorsqu'une variable n'est pas rencontrée, sa valeur n'a pas d'importance.

- Une formule  $F$  est *satisfaite* ou *satisfaisable* par une valuation  $V$  si sa valeur de vérité pour cette valuation est 1.
- Une *tautologie* est une formule qui est satisfaite par toute valuation.
- Une *antilogie* est une formule qui est fausse quelque soit la valuation choisie.
- Deux formules  $F$  et  $G$  sont dites *équivalentes* si pour toute valuation  $V$  les valeurs de vérités de  $F$  et  $G$  sont identiques. On notera  $F \equiv G$ .

Les formules suivantes sont des tautologies :

$$(p \implies p)$$

$$(p \implies (q \implies p))$$

Vérification pour la dernière :

$p$	$q$	$q \implies p$	$(p \implies (q \implies p))$
0	0	1	1
0	1	0	1
1	0	1	1
1	1	1	1

$$\neg\neg p \equiv p$$

$$(p \implies q) \equiv (\neg p \vee q)$$

$$p \iff q \equiv (p \implies q) \wedge (q \implies p)$$

Exemple de preuve d'équivalence :

$p$	$q$	$p \implies q$	$(\neg p \vee q)$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	1

- Deux formules  $F$  et  $G$  sont équivalentes ssi  $(F \iff G)$  est une tautologie.
- $\equiv$  est une relation d'équivalence sur  $\mathcal{F}$ .

## ■ Commutativité :

$$(F \wedge G) \equiv (G \wedge F)$$

$$(F \vee G) \equiv (G \vee F)$$

## ■ Associativité :

$$(F \wedge (G \wedge H)) \equiv ((F \wedge G) \wedge H)$$

$$(F \vee (G \vee H)) \equiv ((F \vee G) \vee H)$$

## ■ Idempotence :

$$(F \wedge F) \equiv F$$

$$(F \vee F) \equiv F$$

## ■ Règles de De Morgan :

$$\neg(F \wedge G) \equiv (\neg F \vee \neg G)$$

$$\neg(F \vee G) \equiv (\neg F \wedge \neg G)$$

## ■ Distributivité :

$$(F \wedge (G \vee H)) \equiv ((F \wedge G) \vee (F \wedge H))$$

$$(F \vee (G \wedge H)) \equiv ((F \vee G) \wedge (F \vee H))$$

## ■ Absorption :

$$(F \wedge (F \vee G)) \equiv F$$

$$(F \vee (F \wedge G)) \equiv F$$

## 1 et 0 (notation) :

$$\blacksquare F \vee \neg F \equiv 1$$

$$\blacksquare F \wedge \neg F \equiv 0$$

$$\blacksquare F \vee 0 \equiv F$$

$$\blacksquare F \vee 1 \equiv 1$$

$$\blacksquare F \wedge 0 \equiv 0$$

$$\blacksquare F \wedge 1 \equiv F$$

Prouvons par équivalence que

$$(A \implies (B \wedge C)) \implies (A \implies B) \wedge (A \implies C)$$

est une tautologie.

$$\begin{aligned} & (A \implies (B \wedge C)) \implies (A \implies B) \wedge (A \implies C) \\ & \equiv \neg(A \implies (B \wedge C)) \vee (A \implies B) \wedge (A \implies C) \\ & \equiv \neg(A \implies (B \wedge C)) \vee (\neg A \vee B) \wedge (\neg A \vee C) \\ & \equiv \neg(A \implies (B \wedge C)) \vee (\neg A \vee (B \wedge C)) \\ & \equiv \neg(A \implies (B \wedge C)) \vee (A \implies (B \wedge C)) \\ & \equiv 1 \end{aligned}$$

Comme on l'a déjà vu, la notion de connecteur se généralise à celle de fonction booléenne.

**Définition** une *fonction booléenne* est une fonction de  $\{0, 1\}^n$  dans  $\{0, 1\}$  avec  $n \geq 1$ .

**Définition** un ensemble  $\mathcal{C}$  de connecteurs est *complet* si pour toute fonction booléenne  $f$  il existe une formule  $A$  contenant uniquement des connecteurs de  $\mathcal{C}$  telle que  $f$  réalise  $A$  (i.e.  $f$  et  $A$  prennent les mêmes valeurs de vérité pour les mêmes distributions).

**Théorème** l'ensemble  $\{\neg, \wedge, \vee\}$  est complet.

**Définition**

$$nor(a, b) = \neg(a \vee b)$$

$$nand(a, b) = \neg(a \wedge b)$$

**Théorème** l'ensemble  $\{nor\}$  est complet ainsi que l'ensemble  $\{nand\}$ .

**Définition** On appelle *littéral* une variable propositionnelle ou une variable propositionnelle précédée d'une négation.

**Définition**

Une forme normale disjonctive est :

- soit une *disjonction*  $(F_1 \vee F_2 \vee \dots \vee F_k)$  où chaque formule  $F_i$  ( $i \in 1, 2, \dots, k$ ) est de la forme  $(G_1^i \wedge G_2^i \wedge \dots \wedge G_{l_i}^i)$  chaque  $G_j^i$  étant un littéral.
- soit réduite à une formule  $F_1$ .

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

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

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

**Définition**

Une forme normale conjonctive est :

- soit une *conjonction*  $(F_1 \wedge F_2 \wedge \dots \wedge F_k)$  où chaque formule  $F_i$  ( $i \in 1, 2, \dots, k$ ) est de la forme  $(G_1^i \vee G_2^i \vee \dots \vee G_{l_i}^i)$  chaque  $G_j^i$  étant un littéral.
- soit réduite à une formule  $F_1$ .

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

$$(p \vee \neg q \vee r) \wedge (\neg p \vee r) \wedge (\neg r \vee \neg s)$$

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

Pour toute formule du calcul propositionnel il est possible de construire une forme normale conjonctive équivalente.

Pour toute formule du calcul propositionnel il est possible de construire une forme normale disjonctive équivalente.

**Définition** une **clause** est une formule qui est une disjonction de littéraux. La clause vide est notée 0 (l'élément neutre de la disjonction).

$A \vee B \vee \neg C \vee D$  et  $\neg A$  sont des clauses.

Comment obtenir un ensemble de clauses à partir d'une formule ?

Il faut calculer sa forme normale conjonctive et ensuite éliminer les connecteurs  $\wedge$ .

L'ensemble des clauses de

$$((p \vee \neg q \vee r) \wedge (\neg p \vee r) \wedge (\neg r \vee \neg s))$$

est

$$\{p \vee \neg q \vee r, \neg p \vee r, \neg r \vee \neg s\}$$

Si  $C_1$  et  $C_2$  sont deux clauses et si  $L_1 = \neg L_2$  et  $L_1$  est dans  $C_1$  et  $L_2$  est dans  $C_2$ ,  $C$  est la disjonction des clauses restantes après suppression des littéraux  $L_1$  et  $L_2$ .  $C$  est appelée *clause résolvente* ou *résolvant* de  $C_1$  et  $C_2$ .

### Exemples

$$C_1 = A \vee B$$

$$C_2 = \neg B \vee C$$

$$C = A \vee C$$

$$C_1 = P$$

$$C_2 = \neg P$$

$$C = 0 \text{ la clause vide}$$

### Propriété

Le résolvant de deux clauses  $C_1$  et  $C_2$  est une conséquence logique de  $C_1$  et  $C_2$  :  $C_1 \wedge C_2 \implies C$

### Propriété

Soit  $\Delta$  un ensemble de clause et  $\Delta_1 = \Delta \cup C$  avec  $C$  clause résolvente de  $\Delta$  alors la conjonction des clauses de  $\Delta$  est satisfaite ssi (la conjonction des clauses de)  $\Delta_1$  est satisfaite.

Consiste à construire une suite d'ensemble de clauses à partir d'un ensemble de clauses  $\Delta$  telle que  $\Delta \subset \Delta_1 \subset \Delta_2 \subset \dots \Delta_n \dots$  et  $\Delta_{i+1}$  satisfaite ssi  $\Delta_i$  l'est.

### Etape élémentaire :

Si  $\Delta_i$  contient deux clauses  $D \vee p$  et  $C \vee \neg p$  et  $D \vee C \notin \Delta_i$  alors  $\Delta_{i+1} = \Delta_i \cup \{D \vee C\}$ .

On peut simplifier les clauses de la forme  $C \vee p \vee p$  directement en  $C \vee p$ .

Soit  $\Delta = \{p \vee r \vee s, r \vee \neg s, \neg r\}$

On peut construire :

$\Delta_1 = \{p \vee r \vee s, r \vee \neg s, \neg r, p \vee r \vee r\}$  (résolvant simplifiable en  $p \vee r$ )

$\Delta_2 = \{p \vee r \vee s, r \vee \neg s, \neg r, p \vee r, p\}$

$\Delta_3 = \{p \vee r \vee s, r \vee \neg s, \neg r, p \vee r, p, p \vee s\}$

$\Delta_4 = \{p \vee r \vee s, r \vee \neg s, \neg r, p \vee r, p, p \vee s, \neg s\}$

Pour prouver qu'une formule est une antilogie il suffit d'obtenir par dérivation la clause vide :

$\Delta = \{p \vee r \vee s, r \vee \neg s, \neg r, \neg p\}$

$\Delta_1 = \{p \vee r \vee s, r \vee \neg s, \neg r, \neg p, p \vee r\}$

$\Delta_2 = \{p \vee r \vee s, r \vee \neg s, \neg r, \neg p, p \vee r, p\}$

$\Delta_3 = \{p \vee r \vee s, r \vee \neg s, \neg r, \neg p, p \vee r, p, 0\}$  (le résolvant de  $p$  et  $\neg p$  est 0)

Donc  $(p \vee r \vee s) \wedge (r \vee \neg s) \wedge \neg r \wedge \neg p$  est une antilogie.

Pour prouver qu'une proposition est une tautologie il suffit de la nier et d'en dériver la clause vide.

Montrons que  $(P \implies Q) \vee (Q \implies R)$  est une tautologie.

$$\neg((P \implies Q) \vee (Q \implies R)) \equiv \neg((\neg P \vee Q) \vee (\neg Q \vee R)) \equiv P \wedge \neg Q \wedge Q \wedge \neg R$$

L'ensemble de clauses  $C$  est :

$$C = \{P, \neg Q, Q, \neg R\}$$

avec  $\neg Q$  et  $Q$  on obtient la clause vide.

Ainsi  $(P \implies Q) \vee (Q \implies R)$  est une tautologie.