

Logique et formules propositionnelles

I) Formules propositionnelles

I.1) Syntaxe

Définition 1 [Formule propositionnelle]

On se donne X un ensemble de *variables*. On définit inductivement les *formules propositionnelles* comme :

- (cas de base) v pour $v \in X$
- (induction) si f_1 et f_2 sont des formules,
 - $\neg f_1$ la négation de f_1
 - $f_1 \wedge f_2$ la conjonction
 - $f_1 \vee f_2$ la disjonction

\neg , \wedge et \vee sont des *connecteurs logiques* d'arité 1, 2 et 2 respectivement.

Définition 2 []

On utilise parfois des connecteurs logiques supplémentaires :

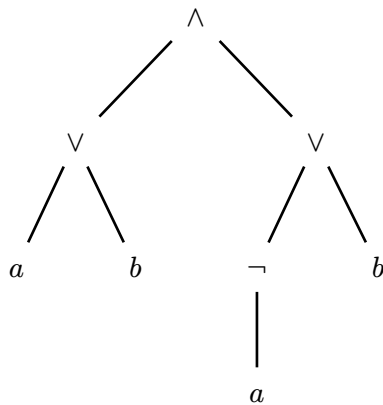
- \Leftrightarrow ou \leftrightarrow pour l'équivalence
- \Rightarrow ou \rightarrow pour l'implication
- \perp , ou F pour le "faux"
- \top , ou V pour le "vrai"

Ces symboles pourront être exprimés à partir des symboles de base.

Exemple 3

$(a \vee b) \wedge (\neg a \vee c)$ est une formule propositionnelle.

Les formules peuvent être représentées sous forme d'arbre :



Définition 4 [hauteur et taille]

On définit

- la *taille* d'une formule comme le nombre de symboles nécessaires pour l'écrire.
- la *hauteur* d'une formule comme la hauteur de l'arbre correspondant.

Exemple 5

$(a \vee b) \wedge (\neg a \vee c)$ est de taille 8 et de hauteur 3.

I.2) Sémantique

On va maintenant donner un *sens* à nos formules.

Définition 6 [valuation]

On appelle *valuation* une application $\nu : X \rightarrow \{F, V\}$.

Définition 7 [valeur de vérité]

Pour *nu* une valuation, on définit par induction la valeur de vérité $[f]_\nu$ d'une formule f :

- pour $x \in X$, $[x]_\nu = \nu(x)$
- si f_1, f_2 sont des formules avec une valeur de vérité $[f_1]_\nu$ et $[f_2]_\nu$, alors

$$\begin{aligned}[\neg f_1]_\nu &= \neg [f_1]_\nu \\ [f_1 \wedge f_2]_\nu &= [f_1]_\nu \wedge [f_2]_\nu \\ [f_1 \vee f_2]_\nu &= [f_1]_\nu \vee [f_2]_\nu\end{aligned}$$

avec le “non”, “et” et “ou” usuels.

Si $[f]_\nu = V$, on dit que ν est un *modèle* de f .

Définition 8 [Satisfiabilité]

On dit que une formules f est :

- satisfiable si elle a un modèle.
- une tautologie si $\forall \nu : X \rightarrow \mathbb{B}$, ν est un modèle de f .

Définition 9 [Equivalence de deux formules]

Deux formules f_1 et f_2 sont *équivalentes* si

$$\forall \nu : X \rightarrow \mathbb{B}, [f_1]_\nu = [f_2]_\nu$$

On note alors $f_1 \equiv f_2$.

I.3) Simplification de formules

Grâce à la notion d'équivalence, on peut simplifier ou reformuler les formules, à l'aide d'opérations.

Propriétés 10 [Opérations sur les formules]

1. *distributivité* \vee et \wedge sont distributives l'une sur l'autre

- $f_1 \wedge (f_2 \vee f_3) \equiv (f_1 \wedge f_2) \vee (f_1 \wedge f_3)$
- $f_1 \vee (f_2 \wedge f_3) \equiv (f_1 \vee f_2) \wedge (f_1 \vee f_3)$

2. *tiers exclu*

- $f \vee (\neg f) \equiv \top$ (c'est une tautologie).

3. *lois de De Morgan* :

- $\neg(f_1 \vee f_2) \equiv \neg f_1 \wedge \neg f_2$
- $\neg(f_1 \wedge f_2) \equiv \neg f_1 \vee \neg f_2$

Propriétés 11 [Réécriture de l'implication et de l'équivalence]

On peut réécrire l'implication et l'équivalence avec les autres opérateurs :

- $f_1 \rightarrow f_2 \equiv \neg f_1 \vee f_2$
- $f_1 \leftrightarrow f_2 \equiv (f_1 \wedge f_2) \vee (\neg f_1 \wedge \neg f_2)$

Exemple 12

Simplifier au maximum la formule suivante :

$$\neg a \vee \neg b \vee (b \wedge \neg a)$$

$$(a \wedge b) \vee (b \wedge \neg a)$$

$$(a \vee \neg a) \wedge b$$

$$b$$

I.4) Conséquence logique

Définition 13 [Ensemble de formule]

Soit Γ un ensemble de formule. On dit qu'une valuation ν est un modèle de Γ si elle est un modèle de chacune de ses formules.

Définition 14 [Conséquence logique d'un ensemble de formules]

Soit Γ un ensemble de formules. Une formule f est une *conséquence logique* de Γ si toute modèle de Γ est un modèle de f .

I.5) Formes normales

Définition 15 [Littéral]

Un littéral est soit une variable, soit la négation d'une variable.

Définition 16 [Clause]

Une clause conjonctive (resp disjonctive) est une conjonction (resp disjonction) de littéraux.

Conjonction = "et", Disjonction = "ou"

Définition 17 [Formes normales]

Une formule f est :

1. en *forme normale conjonctive* est une conjonction de clauses disjonctives, c'est à dire qu'elle est de la forme :

$$\bigwedge_i C_i = \bigwedge_i \bigvee_j l_{ij}$$

1. en *forme normale disjonctive* est une disjonction de clauses conjonctive, c'est à dire qu'elle est de la forme :

$$\bigvee_i C_i = \bigvee_i \bigwedge_j l_{ij}$$

La forme la plus courante est la forme normale conjonctive, et donc on dit parfois simplement "clause" au lieu de "clause disjonctive".

Propriétés 18 [Lien avec la table de vérité]

Une formule est en *forme normale disjonctive complète* si chaque clause contient toutes les variables.

Dans ce cas, pour remplir la table de vérité, il suffit de mettre un "1" correspondant à chaque clause, et un 0 sur le reste.

Exemple 19

$$(a \wedge b) \vee (\neg a \wedge \neg b)$$

I.6) Le problème SAT

On a plusieurs problèmes très célèbres en informatique : la famille des problèmes SAT.

Définition 20 [Problèmes SAT]

- *SAT* : étant donné une formule, est-elle satisfiable ?
- *CNF-SAT* : étant donné une formule en forme normale conjonctive, est-elle satisfiable ?
- *k-SAT* : étant donné une formule en forme normale conjonctive avec au plus k littéraux par clause, est-elle satisfiable ?

Exemple 21

Exemple d'instances de ces problèmes :

- *SAT* : $a \vee (b \wedge (c \wedge \neg d \wedge d))$ est-elle satisfiable ?
- *2-SAT* : $(a \vee \neg b) \wedge (\neg a \vee b)$ est-elle satisfiable ?

Théorème 22 [hors programme]

3-SAT est NP-complet. Il en découle que SAT, CNF-SAT, et k-SAT pour $k \geq 3$ le sont aussi.

Remarque 23

Globalement, un problème NP-complet est un problème qui ne peut à priori pas être résolu en temps polynomial, donc c'est un problème (très) difficile.

Cela vaut si $P \neq NP$, ce qui n'a pas été démontré.

Le problème SAT est important car il est assez facile d'"encoder" plein d'autres problèmes avec, et de montrer que ceux-ci sont également NP-complets.