



UNIVERSITÉ LIBRE DE BRUXELLES

INFO-F302
INFORMATIQUE FONDAMENTALE

Synthèse

Étudiants :
Hugo CALLENS

Enseignants :
E. FILIOT

29 septembre 2023



Contents

1	Logique propositionnelle	2
1.1	Construction de formules	2
1.2	Sémantique	2
1.3	Validité et Stabilité	2
1.3.1	Définitions	2
1.3.2	Conséquence logique	3
1.3.3	Equivalence	3
1.3.4	Lien entre satisfaisabilité et validité	3
1.3.5	Tableaux sémantiques	4

1 Logique propositionnelle

1.1 Construction de formules

Le vocabulaire du langage de la logique propositionnelle est composé de :

1. de propositions x, y, z, \dots ; ou X, Y, Z, \dots ;
2. de deux constantes vrai (\top ou 1) et faux (\perp ou 0);
3. d'un ensemble de connecteurs logiques : $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$.
4. de parenthèses ().

1.2 Sémantique



La sémantique d'une formule est la valeur de vérité de cette formule. La valeur de vérité d'une formule Φ formée à partir de propositions d'un ensemble X , évaluée avec la fonction d'interprétation V , est notée $\llbracket \Phi \rrbracket_V$. La fonction $\llbracket \Phi \rrbracket_V$ est définie par induction sur la syntaxe de Φ de la façon suivante :

- $\llbracket \top \rrbracket_V = 1$; $\llbracket \perp \rrbracket_V = 0$; $\llbracket x \rrbracket_V = V(x)$
- $\llbracket \neg \Phi \rrbracket_V = 1 - \llbracket \Phi \rrbracket_V$
- $\llbracket \Phi_1 \vee \Phi_2 \rrbracket_V = \max(\llbracket \Phi_1 \rrbracket_V, \llbracket \Phi_2 \rrbracket_V)$
- $\llbracket \Phi_1 \wedge \Phi_2 \rrbracket_V = \min(\llbracket \Phi_1 \rrbracket_V, \llbracket \Phi_2 \rrbracket_V)$
- $\llbracket \Phi_1 \leftarrow \Phi_2 \rrbracket_V = \max(1 - \llbracket \Phi_1 \rrbracket_V, \llbracket \Phi_2 \rrbracket_V)$
- $\llbracket \Phi_1 \leftrightarrow \Phi_2 \rrbracket_V = \min(\llbracket \Phi_1 \rightarrow \Phi_2 \rrbracket_V, \llbracket \Phi_2 \rightarrow \Phi_1 \rrbracket_V)$

Nous notons $V \models \Phi \Leftrightarrow \llbracket \Phi \rrbracket_V = 1$ soit " V satisfait Φ ."

L'information contenue dans la définition est souvent représentée sous forme de table de vérité :

Φ_1	Φ_2	$\Phi_1 \vee \Phi_2$	$\Phi_1 \wedge \Phi_2$	$\Phi_1 \rightarrow \Phi_2$	$\Phi_1 \leftrightarrow \Phi_2$
0	0	0	0	1	1
0	1	1	0	1	0
1	0	1	0	0	0
1	1	1	1	1	1



Dans l'implication suivante : $\Phi_1 \rightarrow \Phi_2$, la cas où Φ_1 est faux ne nous intéresse pas. Dans ce cas, l'implication est toujours vraie.

1.3 Validité et Stabilité

1.3.1 Définitions



Une formule propositionnelle Φ est **satisfaisable** \Leftrightarrow il existe une fonction d'interprétation V pour les propositions de Φ , telle que $V \models \Phi$.



Une formule propositionnelle Φ est **valide** \Leftrightarrow pour toute fonction d'interprétation V pour les propositions de Φ , $V \models \Phi$.

1.3.2 Conséquence logique



Soit $\Phi_1, \dots, \Phi_n, \Phi$ des formules. On dira que Φ est une **conséquence logique** de Φ_1, \dots, Φ_n , noté $\Phi_1, \dots, \Phi_n \models \Phi$, si $(\Phi_1 \wedge \dots \wedge \Phi_n) \rightarrow \Phi$ est valide.

1.3.3 Equivalence



Deux formules, Φ et Ψ , sont **équivalentes** si la formule $\Phi \leftrightarrow \Psi$ est valide. On notera $\Phi \equiv \Psi$.

Pour toutes formules Φ_1, Φ_2, Φ_3 :

- $\neg\neg\Phi_1 \equiv \Phi_1$
- $\neg(\Phi_1 \wedge \Phi_2) \equiv (\neg\Phi_1 \vee \neg\Phi_2)$
- $\neg(\Phi_1 \vee \Phi_2) \equiv (\neg\Phi_1 \wedge \neg\Phi_2)$
- $\Phi_1 \wedge (\Phi_2 \vee \Phi_3) \equiv (\Phi_1 \wedge \Phi_2) \vee (\Phi_1 \wedge \Phi_3)$
- $\Phi_1 \vee (\Phi_2 \wedge \Phi_3) \equiv (\Phi_1 \vee \Phi_2) \wedge (\Phi_1 \vee \Phi_3)$
- $\Phi_1 \rightarrow \Phi_2 \equiv (\neg\Phi_1 \vee \Phi_2)$

1.3.4 Lien entre satisfaisabilité et validité



Une formule Φ est valide $\Leftrightarrow \neg\Phi$ est insatisfaisable.

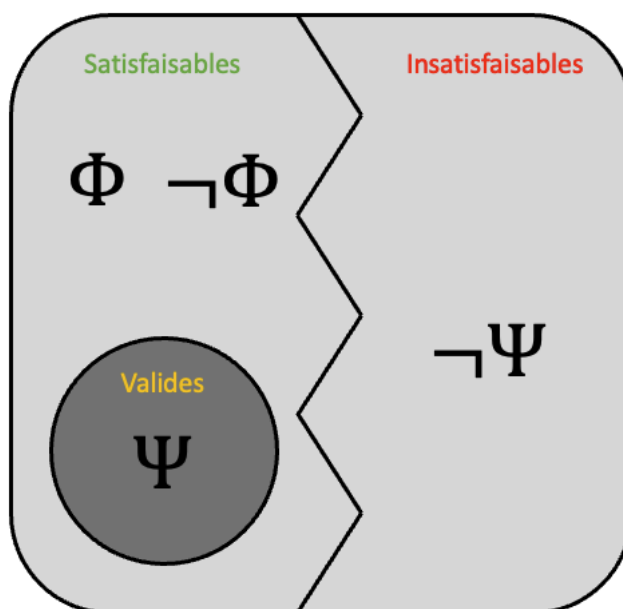


FIGURE 1.1 – Lien entre satisfaisabilité et validité

1.3.5 Tableaux sémantiques



Un littéral est une proposition x ou la négation d'une proposition $\neg x$.

La méthode des tableaux sémantiques est un algorithme pour tester la satisfaisabilité d'une formule. Elle consiste à construire un arbre dont les noeuds sont des formules et les feuilles sont des littéraux. On construit l'arbre de la façon suivante :

- On place la formule à tester à la racine de l'arbre.
- On applique les règles suivantes jusqu'à ce que l'arbre soit complet :
 - Si la formule à tester est une constante, on arrête.
 - Si la formule à tester est une conjonction, on ajoute les deux conjoncteurs comme fils de la formule à tester.
 - Si la formule à tester est une disjonction, on ajoute un fils avec le premier disjoncteur et un autre fils avec le deuxième disjoncteur.
 - Si la formule à tester est une implication, on ajoute un fils avec la négation de l'antécédent et un autre fils avec le conséquent.
 - Si la formule à tester est une équivalence, on ajoute un fils avec la négation de la première formule et un autre fils avec la deuxième formule.
 - Si la formule à tester est une négation, on ajoute un fils avec la négation de la formule à tester.