Université Libre de Bruxelles

INFO-F302
Informatique Fondamentale

Synthèse

Étudiants :
Hugo Callens

Enseignants :
E. Filiot

10 octobre 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 3

2 Déduction naturelle **4**

2.1 Règles 4

2.1.1 Règles pour la conjonction 4

2.1.2 Règles pour la double négation 4

2.1.3 Elimination de l'implication : Modus Ponens 4

2.1.4 Règle pour l'introduction de l'implication 5

2.1.5 Règle pour l'ouverture et la fermeture d'hypothèses 5

2.1.6 Règle pour l'introduction de la disjonction 5

2.1.7 Elimination de la disjonction 5

2.2 Théorèmes 5

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, ... ; ou X, Y, Z, ... ;
2. de deux constantes vrai ($\top$ ou 1) et faux ($\bot$ ou 0) ;
3. d'un ensemble de connecteurs logiques : $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.
4. de paranthèses ( ).

1.2 Sémantique

Définition 1.1. Sémantique

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

- $[\![\top]\!]_V = 1$; $[\![\bot]\!]_V = 0$; $[\![x]\!]_V = V(x)$
- $[\![\neg\Phi]\!]_V = 1 - [\![\Phi]\!]_V$
- $[\![\Phi_1 \vee \Phi_2]\!]_V = \max([\![\Phi_1]\!]_V, [\![\Phi_2]\!]_V)$
- $[\![\Phi_1 \wedge \Phi_2]\!]_V = \min([\![\Phi_1]\!]_V, [\![\Phi_2]\!]_V)$
- $[\![\Phi_1 \leftarrow \Phi_2]\!]_V = \max(1 - [\![\Phi_1]\!]_V, [\![\Phi_2]\!]_V)$
- $[\![\Phi_1 \leftrightarrow \Phi_2]\!]_V = \min([\![\Phi_1 \rightarrow \Phi_2]\!]_V, [\![\Phi_2 \rightarrow \Phi_1]\!]_V)$

Nous notons $V \vDash \Phi \Leftrightarrow [\![\Phi]\!]_V = 1$ soit "V satisfait Φ."

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

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

Définition 1.2. Formule propositionnelle satisfaisable

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

Définition 1.3. Formule propositionnelle valide

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

Définition 1.4. Conséquence Logique

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

1.3.3 Equivalence

Définition 1.5. Formules équivalentes

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

Théorème 1.1. Lien entre satisfaisabilité et validité

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

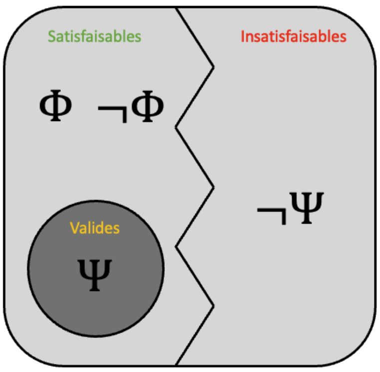

FIGURE 1.1 – Lien entre satisfaisabilité et validité

1.3.5 Tableaux sémantiques

Définition 1.6. Littéral

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.

Remarque: algorithme généré par copilot.

2 Déduction naturelle

2.1 Règles

2.1.1 Règles pour la conjonction

- Règle d'introduction :

$$\frac{\Phi \quad \Psi}{\Phi \wedge \Psi} \bigwedge i$$

- Règle d'élimination :

$$\frac{\Phi \wedge \Psi}{\Phi} \bigwedge e_1 \qquad \frac{\Phi \wedge \Psi}{\Psi} \bigwedge e_2$$

2.1.2 Règles pour la double négation

- Règle d'introduction :

$$\frac{\Phi}{\neg\neg\Phi} \neg\neg i$$

- Règle d'élimination :

$$\frac{\neg\neg\Phi}{\Phi} \neg\neg e$$

2.1.3 Elimination de l'implication : Modus Ponens

Règle d'élimination :

$$\frac{\Phi \quad \Phi \rightarrow \Psi}{\Psi} \rightarrow_{\text{MP}} (\text{ou} \rightarrow_e)$$

En contraposition, nous avons le Modus Tollens :

$$\frac{\Phi \rightarrow \Psi \quad \neg\Psi}{\neg\Phi} \rightarrow_{\text{MT}}$$

2.1.4 Règle pour l'introduction de l'implication

$$\begin{array}{l}\Phi\text{hyp.}\\ \dots\\ \dfrac{\Psi\text{fin hyp.}}{\Phi \to \Psi} \to_i\end{array}$$

2.1.5 Règle pour l'ouverture et la fermeture d'hypothèses

- toute hypothèse introduite doit être fermée.
- on ne peut jamais fermer deux hypothèses en même temps.
- Une fois une hypothèse fermée, on ne peut pas utilsier les formules déduites entre l'ouverture et la fermeture de cette hypothèse.

2.1.6 Règle pour l'introduction de la disjonction

$$\frac{\Phi}{\Phi \vee \Psi} \vee_{i_2} \qquad \frac{\Psi}{\Phi \vee \Psi} \vee_{i_2}$$

2.1.7 Elimination de la disjonction

$$\begin{array}{l}\Psi_1\text{hyp.} \quad \Psi_2\text{hyp.}\\ \dots\\ \dfrac{\Psi_1 \vee \Psi_2 \quad \Phi\text{fin hyp.} \quad \Phi\text{fin hyp.}}{\Phi} \vee_e\end{array}$$

Remarque: ... Il y a encore tout un tas de règles dérivées etc, mais je pense qu'elles sont principalement d'orde pratique. De même je pense que ce chapitre de cours demande plutôt de l'entrainement que de la théorie.

2.2 Théorèmes

Définition 2.1. adéquation

Pour tout $\Psi_1, ..., \Psi_n, \Psi$, si $\Psi_1, ..., \Psi_n \vdash \Psi$ alors $\Psi_1, ..., \Psi_n \models \Psi$.

Définition 2.2. complétude

Pour tout $\Psi_1, ..., \Psi_n, \Psi$, si $\Psi_1, ..., \Psi_n \models \Psi$ alors $\Psi_1, ..., \Psi_n \vdash \Psi$.