Cours logique - Mémo n°1

Logique propositionnelle

Emmanuel Coquery

1 Syntaxe

Définition 1 (Formules logiques) *Soient les deux constantes* $\top$ *et* $\bot$*, un ensemble* $\mathcal{P}$ *de* variables propositionnelles*, notées* $p, q, r, \ldots$*, et les* connecteurs logiques $\{\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow\}$.

- $\top$, $\bot$, p *où* p *est une variable propositionnelle sont des formules.*
- *Si* A *est une formule, alors* $(\neg A)$ *est une formule.*
- *Si* A *et* B *sont des formules alors* $(A \vee B)$, $(A \wedge B)$, $(A \Rightarrow B)$ *et* $(A \Leftrightarrow B)$ *sont des formules.*

On pourra se passer des parenthèses en utilisant les règles suivantes : $\neg$ est plus prioritaire que les autres opérateurs et $\vee$ et $\wedge$ sont plus prioritaires que $\Rightarrow$ et $\Leftrightarrow$.

Définition 2 (Formules atomiques) *Une formule est* atomique *si c'est* $\top$, $\bot$ *ou si c'est une variable propositionnelle.*

Définition 3 (Sous-formules) *Étant donnée une formule logique* A*, l'ensemble de ses* sous-formules *est donné par la fonction* $sf(A)$*, définie inductivement comme suit :*

- *Si* A *est atomique,* $sf(A) = \{A\}$.
- *Si* A *est de la forme* $\neg B$*, alors* $sf(A) = \{A\} \cup sf(B)$.
- *Si* A *est de la forme* $B \vee C$, $B \wedge C$, $B \Rightarrow C$ *ou* $B \Leftrightarrow C$*, alors* $sf(A) = \{A\} \cup sf(B) \cup sf(C)$.

Définition 4 (Arbre de syntaxe abstraite) *L'*arbre de syntaxe abstraite $ASA(A)$ *associé à une formule* A *est un arbre dont les noeud sont étiqueté par des connecteurs logiques ou des variables propositionnelles. Il est inductivement défini comme suit :*

- *Si* A *est atomique, alors* $ASA(A)$ *est une feuille étiquetée par* A.
- *Si* $A = \neg B$*, alors* $ASA(A)$ *est un arbre dont la racine est étiquetée par* $\neg$ *et qui a un seul fils :* $ASA(B)$.
- *Si* $A = B \vee C$*, alors* $ASA(A)$ *est un arbre dont la racine est étiquetée par* $\vee$ *et qui a deux fils. Le fils gauche est* $ASA(B)$ *et le fils droit est* $ASA(C)$.
- *Si* $A = B \wedge C$*, alors* $ASA(A)$ *est un arbre dont la racine est étiquetée par* $\wedge$ *et qui a deux fils. Le fils gauche est* $ASA(B)$ *et le fils droit est* $ASA(C)$.
- *Si* $A = B \Rightarrow C$*, alors* $ASA(A)$ *est un arbre dont la racine est étiquetée par* $\Rightarrow$ *et qui a deux fils. Le fils gauche est* $ASA(B)$ *et le fils droit est* $ASA(C)$.
- *Si* $A = B \Leftrightarrow C$*, alors* $ASA(A)$ *est un arbre dont la racine est étiquetée par* $\Leftrightarrow$ *et qui a deux fils. Le fils gauche est* $ASA(B)$ *et le fils droit est* $ASA(C)$.

2 Sémantique

Définition 5 *L'ensemble des* booléens*, noté* $\mathcal{B}$*, est l'ensemble* $\{0, 1\}$. 0 *dénote la valeur* faux *et* 1 *la valeur* vrai.

Définition 6 *Une* fonction booléenne *à* n *arguments est une fonction* $\mathcal{B}^n \to \mathcal{B}$.

Définition 7 *La fonction booléenne $f_\neg$ associée au connecteur $\neg$ est donnée par la table de vérité suivante :*

x	$f_\neg(x)$
1	0
0	1

Les fonction booléennes $f_\vee$, $f_\wedge$, $f_\Rightarrow$ et $f_\Leftrightarrow$ associées aux connecteurs $\vee$, $\wedge$, $\Rightarrow$ et $\Leftrightarrow$ sont données par la table de vérité suivante :

x	y	$f_\vee(x,y)$	$f_\wedge(x,y)$	$f_\Rightarrow(x,y)$	$f_\Leftrightarrow(x,y)$
1	1	1	1	1	1
1	0	1	0	0	0
0	1	1	0	1	0
0	0	0	0	1	1

Définition 8 *Une* interprétation *pour un ensemble de variable propositionnelles P est une fonction $I : P \to \mathcal{B}$.*

Définition 9 *La* valeur de vérité *d'une formule A par rapport à une interprétation I, notée $[A]_I$ est définie inductivement de la manière suivante :*
- *Si p est une variable propositionnelle, $[p]_I = I(p)$.*
- *$[\top]_I = 1$ et $[\bot]_I = 0$.*
- *Si A est une formule logique alors $[\neg A]_I = f_\neg([A]_I)$.*
- *Si A et B sont deux formules logiques alors :*
  - $[A \vee B]_I = f_\vee([A]_I, [B]_I)$
  - $[A \wedge B]_I = f_\wedge([A]_I, [B]_I)$
  - $[A \Rightarrow B]_I = f_\Rightarrow([A]_I, [B]_I)$
  - $[A \Leftrightarrow B]_I = f_\Leftrightarrow([A]_I, [B]_I)$

Définition 10 *Une interprétation I est un* modèle *d'une formule A, noté $I \models A$, si et seulement si $[A]_I = 1$.*

Une interprétation I est un modèle d'un ensemble de formules $E = \{A_1, \ldots, A_n\}$, noté $I \models E$, si pour toute formule $A_i \in E$, $I \models A_i$.

Définition 11 *Une formule A est dite* satisfiable *si il existe une interprétation I telle que $I \models A$.*

Une ensemble de formules $E = \{A_1, \ldots, A_n\}$ est satisfiable si il existe une interprétation I qui est un modèle de E.

Une formule (resp. un ensemble de formules) qui n'est pas satisfiable est dit insatisfiable.

Définition 12 *Le problème* SAT *consiste à déterminer si une formule est satisfiable. Autrement dit, $SAT(A) = true$ si A est satisfiable. Sinon $SAT(A) = false$.*

L'algorithme naïf pour résoudre ce problème consiste à énumérer toutes les interprétations possibles de A pour vérifier si une de ces interprétations est un modèle de A. On peut remarquer que si A possède n variables, il existe 2^n interprétations possibles pour A. Il n'existe cependant pas d'algorithme connu permettant de faire mieux dans tous les cas (le problème est dit NP-Complet).

Définition 13 *Une formule A est* valide, *si toute interprétation est un modèle de A. On note $\models A$ le fait que A soit valide. On dit également que A est une* tautologie.

Un ensemble de formules est valide si toutes ses formules sont valides.

Propriété 1 *Une formule A est valide si et seulement si $\neg A$ est insatisfiable.*

Preuve: (Intérêt de la preuve : comprendre les interactions entre les définitions 7, 9, 10, 11 et 13.)

A est valide si et seulement si (noté *ssi*) pour toute interprétation I, $I \models A$, i.e. ssi $[A]_I = 1$. D'après les définitions 7 et 9 $[A]_I = 1$ ssi $[\neg A]_I = 0$. Donc A est valide ssi pour toute interprétation I, $I \not\models \neg A$, i.e. si $\neg A$ est insatisfiable. □

Définition 14 *Soit $E = \{A_1, \ldots, A_n\}$ une ensemble de formules et B une formule. B est une conséquence logique de E, noté $E \models B$ si tout modèle de E est un modèle de B.*

Propriété 2
- *$\{A_1, \ldots, A_n\} \models B$ si et seulement si $(A_1 \wedge \ldots \wedge A_n) \Rightarrow B$ est valide.*
- *Si E est insatisfiable, alors pour toute formule B, $E \models B$.*
- *Si B est valide, alors pour tout E, $E \models B$.*
- *$E \models B$ si et seulement si $E \cup \neg B$ est insatisfiable.*

Preuve: A faire en TD □

3 Equivalence de formules

Définition 15 *Deux formules A et B sont dites équivalentes, noté $A \equiv B$, si $A \models B$ et $B \models A$.*

Propriété 3 (Principe de substitution) *Soient A, B et C des formules telles que $A \equiv B$ et A est une sous-formule de C. Toute formule C' obtenue en remplaçant une occurrence de A par B dans C est équivalente à C.*

Preuve: A faire en TD. □

Définition 16 *Soit f une fonction booléenne à n arguments. Soit A une formule ayant n variables $p_1, \ldots, p_n$. Si, pour toute interprétation I de A, on a $f(I(p_1), \ldots, I(p_n)) = [A]_I$, alors ont dit que f est* réalisée par *A.*

Définition 17 *Soit C un ensemble de connecteurs. C est dit* fonctionnellement complet *si pour toute fonction booléenne f il existe une formule A ne contenant que des connecteurs de C et telle que f soit réalisée par A.*

Propriété 4 *$\{\neg, \vee, \wedge\}$, $\{\neg, \wedge\}$, $\{\neg, \vee\}$ et $\{\neg, \Rightarrow\}$ sont fonctionnellement complets.*

Preuve: (Intérêt de la preuve : voir une preuve par récurrence, preuve constructive : on peut en faire un programme.)

On commence par prouver que $\{\neg, \vee, \wedge\}$ est fonctionnellement complet.

Soit $f : \mathcal{B}^n \to \mathcal{B}$. On montre par récurrence qu'il existe une formule A_f ayant $p_1, \ldots, p_n$ comme variables telle que pour tout $(b_1, \ldots, b_n) \in \mathcal{B}^n$, en posant $I(p_1) = b_1, \ldots, I(p_n) = b_n$, on a $[A_f]_I = f(b_1, \ldots, b_n)$.

Considérons le cas de base $n = 1$.

Il n'y a que 4 fonctions booléennes à 1 argument, décrites dans le tableau ci-dessous :

	$f(0)$	$f(1)$
f_1	1	1
f_2	1	0
f_3	0	1
f_4	0	0

Le tableau ci-dessous donne pour chaque f_i la formule A_i qui la réalise (i.e. telle que pour tout $b_1 \in \{0,1\}$, en posant $I(p_1) = b_1$, on a $[A_i]_I = f_i(b_1)$:

A_1	$p_1 \vee \neg p_1$
A_2	$\neg p_1$
A_3	p_1
A_4	$p_1 \wedge \neg p_1$

Considérons le cas où f possède $n+1$ arguments et supposons la propriété démontrée pour tout f ayant n arguments ou moins.

Posons $f_0(b_1, \ldots, b_n) = f(b_1, \ldots, b_n, 0)$ et $f_1(b_1, \ldots, b_n) = f(b_1, \ldots, b_n, 1)$.

Alors il existe A_0 ayant pour variables $p_1, \ldots, p_n$ et réalisant f_0 et A_1 ayant pour variables $p_1, \ldots, p_n$ et réalisant f_1.

Posons $A_f = ((\neg p_{n+1}) \wedge A_0) \vee (p_{n+1} \wedge A_1)$.

Montrons que A_f réalise f. Soit $(b_1, \ldots n_{n+1}) \in B^{n+1}$ et I tq $I(p_k) = b_k$ pour $1 \leq k \leq n+1$.

- si $b_{n+1} = 0$, alors $[p_{n+1}]_I = 0$ et donc $[p_{n+1} \wedge A_1]_I = 0$, d'où
  $[A_f]_I = [(\neg p_{n+1}) \wedge A_0]_I = [A_0]_I = f_0(b_1, \ldots, b_n) = f(b_1, \ldots, b_n, b_{n+1})$.
- si $b_{n+1} = 1$, alors $[p_{n+1}]_I = 1$ et donc $[\neg p_{n+1} \wedge A_0]_I = 0$, d'où
  $[A_f]_I = [p_{n+1} \wedge A_1]_I = [A_1]_I = f_1(b_1, \ldots, b_n) = f(b_1, \ldots, b_n, b_{n+1})$.

$\square$

Quelques équivalences remarquables :

$\neg\top \equiv \bot$

$\neg\bot \equiv \top$

$\top \wedge A \equiv A$

$\bot \vee A \equiv A$

$\top \vee A \equiv \top$

$\bot \wedge A \equiv \bot$

$A \vee \neg A \equiv \top$

$A \wedge \neg A \equiv \bot$

$\neg\neg A \equiv A$

$A \vee A \equiv A \equiv A \wedge A$

$A \vee B \equiv B \vee A$

$A \wedge B \equiv B \wedge A$

$A \vee (B \vee C) \equiv (A \vee B) \vee C$
$\equiv A \vee B \vee C$

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

$A \wedge (A \vee B) \equiv A$

$A \vee (A \wedge B) \equiv A$

$\neg(A \wedge B) \equiv \neg A \vee \neg B$

$\neg(A \vee B) \equiv \neg A \wedge \neg B$

$A \Rightarrow B \equiv \neg A \vee B$

$A \Rightarrow B \equiv \neg B \Rightarrow \neg A$

$A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A)$

$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$

$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$

Quelques tautologies remarquables :

$A \Rightarrow A$ (identité)

$((A \Rightarrow B) \Rightarrow A) \Rightarrow A$ (loi de Pierce)

$A \vee \neg A$ (tiers exclu)

$(A \Rightarrow B) \wedge A \Rightarrow B$ (modus ponens)

$(A \Rightarrow B) \wedge \neg B \Rightarrow \neg A$ (modus tollens)

$(A \Rightarrow B) \wedge (B \Rightarrow C) \Rightarrow (A \Rightarrow C)$ (modus barbara)

4 Substitutions en calcul propositionnel

Attention : ne pas confondre avec le principe de substitution.

Définition 18 *Une substitution est une fonction σ d'un ensemble fini de variables propositionnelles dans les formules. On notera $dom(\sigma)$ son domaine de définition.*

Notation: on notera $[{}^{A_1}/_{p_1}, \ldots, {}^{A_n}/_{p_n}]$ la substitution σ qui, pour $1 \leq i \leq n$, associe la formule A_i à la variable p_i.

Définition 19 *L'application d'une substitution σ à une formule A, notée $A\sigma$, est définie inductivement par :*
- $\top\sigma = \top$
- $\bot\sigma = \bot$
- $p\sigma = \sigma(p)$ *si p est une variable propositionnelle appartenant à $dom(\sigma)$*
- $p\sigma = p$ *si p est une variable propositionnelle n'appartenant pas à $dom(\sigma)$*
- $(\neg A)\sigma = \neg(A\sigma)$
- $(A \vee B)\sigma = A\sigma \vee B\sigma$
- $(A \wedge B)\sigma = A\sigma \wedge B\sigma$
- $(A \Rightarrow B)\sigma = A\sigma \Rightarrow B\sigma$
- $(A \Leftrightarrow B)\sigma = A\sigma \Leftrightarrow B\sigma$

Propriété 5 *Soit A une formule et σ une substitution quelconques. Si A est valide, alors $A\sigma$ est valide.*

Preuve: Soit $\sigma = [{}^{A_1}/_{p_1}, \ldots, {}^{A_n}/_{p_n}]$ une substitution quelconque. Il faut montrer pour toute interprétation I, $[A\sigma]_I = 1$. Soit I une interprétation quelconque.

On introduit une interprétation auxiliaire I^σ telle que I^σ est identique à I sauf pour les variables substituées par σ. Pour une telle variable p_i, la valeur de I^σ est donnée par la valeur de vérité dans I de la formule A_i qui remplace p_i. Plus formellement, I^σ est définie par :
- $I^\sigma(p) = I(p)$ si p n'est pas dans $dom(\sigma)$
- $I^\sigma(p) = [\sigma(p)]_I$ si p est dans $dom(\sigma)$
  (c'est-à-dire $I^\sigma(p_i) = [A_i]_I$ pour $1 \leq i \leq n$).

On montre à présent par induction que pour toute formule B, $[B\sigma]_I = [B]_{I^\sigma}$. On procède par cas en fonction de la forme de B :
- $B = \top$: on a $B\sigma = \top$ et $[B\sigma]_I = [\top]_I = 1 = [\top]_{I^\sigma} = [B]_{I^\sigma}$.
- $B = \bot$: on a $B\sigma = \bot$ et $[B\sigma]_I = [\bot]_I = 0 = [\bot]_{I^\sigma} = [B]_{I^\sigma}$.
- $B = p$ avec p qui n'est pas dans $dom(\sigma)$: on a $B\sigma = p$
  et $[B\sigma]_I = [p]_I = I(p) = I^\sigma(p) = [p]_{I^\sigma} = [B]_{I^\sigma}$.
- $B = p$ avec p qui est dans $dom(\sigma)$: on a $B\sigma = \sigma(p)$
  et $[B\sigma]_I = [\sigma(p)]_I = I^\sigma(p) = [p]_{I^\sigma} = [B]_{I^\sigma}$.
- $B = \neg C$: on a $B\sigma = \neg(C\sigma)$ et, par hypothèse d'induction, $[C\sigma]_I = [C]_{I^\sigma}$.
  On en déduit : $[B\sigma]_I = [\neg(C\sigma)]_I = f_\neg([C\sigma]_I) = f_\neg([C]_{I^\sigma}) = [\neg C]_{I^\sigma} = [B]_{I^\sigma}$.
- $B = C \vee D$: on a $B\sigma = C\sigma \vee D\sigma$ et par hypothèse d'induction, $[C\sigma]_I = [C]_{I^\sigma}$ et $[D\sigma]_I = [D]_{I^\sigma}$.
  On en déduit : $[B\sigma]_I = [C\sigma \vee D\sigma]_I = f_\vee([C\sigma]_I, [D\sigma]_I) = f_\vee([C]_{I^\sigma}, [D]_I) = [C \vee D]_{I^\sigma} = [B]_{I^\sigma}$
- La démonstration des cas $\wedge$, $\Rightarrow$ et $\Leftrightarrow$ est similaire au cas précédent.

Ce résultat est en particulier vrai pour A, c'est-à-dire que $[A\sigma]_I = [A]_{I^\sigma}$. Comme A est valide, $[A]_{I^\sigma} = 1$. Comme I est quelconque, ce résultat est vrai pour tout I. Donc pour toute interprétation I, $[A\sigma]_I = 1$, donc $A\sigma$ est valide. □

Corollaire 1 *Si $A \equiv B$ alors pour toute substitution $A\sigma \equiv B\sigma$*

En particulier, pour démontrer une équivalence remarquable, il suffit de la démontrer lorsque les formules sont des variables (par exemple en utilisant les tables de vérités), le corollaire précédent permettant d'étendre le résultat à toutes les formules.