Calcul Propositionnel

Marc Chevalier
DI ENS

10 septembre 2018

Résumé

Dans ce cours, nous introduisons le calcul propositionnel. Cette logique permet d'écrire des formules à propos de variables atomiques, connectées par des opérateurs logiques comme la disjonction (ou), la conjonction (et), ou la négation. Elle ne comporte pas de quantificateurs (« pour tout », « il existe »). Le but est d'établir des bases pour le raisonnement mathématique.

1 Syntaxe

La syntaxe d'un langage formalise les phrases (ou formules, ou programmes) que nous pouvons écrire. Une syntaxe est généralement donnée par un alphabet de symboles, et des règles de bonne formation pour les phrases ou mots que nous voulons écrire avec ces symboles.

Les phrases (ou formules) du calcul propositionnel sont des suites de symboles qui obéissent à des règles grammaticales. Définissons tout d'abord l'alphabet de ces symboles.

Définition 1 – Alphabet

Un symbole est l'une de ces quatre entités :

— une variable propositionnelle (ou proposition atomique, ou atome) : A, B, C, $A_0, \ldots, A_n$;

— une constante : $\top$, $\bot$;

— un connecteur logique : $\vee$, $\wedge$, $\neg$;

— une paire de séparateurs : (,).

Nous pouvons maintenant donner les règles de bonne formation (ou grammaire) qui définissent les phrases (ou formules) qui sont correctement formées.

C'est une définition par induction. Nous définissons d'abord les formules élémentaires. Puis on s'autorise des règles de construction de formule. Nous en apprendrons plus sur l'induction, ou les récurrences en général dans les prochains cours.

Définition 2 – Formules

L'ensemble des formule est définie par induction par :

- les variables et constantes propositionnelles sont des formules ;
- si φ_1 et φ_2 sont des formules :
  - $\varphi_1 \wedge \varphi_2$ est une formule ;
  - $\varphi_1 \vee \varphi_2$ est une formule ;
  - $\neg\varphi_1$ est une formule ;
  - (φ_1) est une formule.

Exemple 1

- $\top$
- A
- $(A \wedge (\neg A))$
- $((A \vee B) \wedge (\neg A))$
- $((\neg((\neg A) \wedge \bot)) \wedge A)$

sont des formules.

- $()A\vee$
- $A\neg B$

ne sont pas des formules.

Définition 3 – Priorité des opérateurs

Les opérateurs par ordre de priorité décroissante :

- $\neg$;
- $\wedge$;
- $\vee$.

Définition 4 – Taille d'une formule

Étant donné une formule φ, on définit sa taille $|\varphi|$ par :

- 1 si φ est une variable ou constante propositionnelle ;
- $|\varphi_1|$ si $\varphi = (\varphi_1)$;
- $1 + |\varphi_1|$ si $\varphi = \neg\varphi_1$;
- $1 + \max(|\varphi_1|, |\varphi_2|)$ si $\varphi = \varphi_1 \vee \varphi_2$ ou $\varphi = \varphi_1 \wedge \varphi_2$;

où φ_1 et φ_2 sont des formules du calcul propositionnel.

Exemple 2

— $\top$ est de taille 1
— A est de taille 1
— $(A \wedge (\neg A))$ est de taille 3
— $((A \vee B) \wedge (\neg A))$ est de taille 3
— $((\neg((\neg A) \wedge \bot)) \wedge A)$ est de taille 5

Définition 5

On définit l'ensemble des variables par induction sur les formules de la manière suivante :
— $\text{Var}\,(A) = \{A\}$ si A est une variable propositionnelle ;
— $\text{Var}\,(\top) = \emptyset$;
— $\text{Var}\,(\bot) = \emptyset$;
— $\text{Var}\,(\neg\varphi) = \text{Var}\,(\varphi)$ si φ est une formule du calcul propositionnel ;
— $\text{Var}\,(\varphi_1 \wedge \varphi_2) = \text{Var}\,(\varphi_1) \cup \text{Var}\,(\varphi_2)$ si φ_1 et φ_2 sont deux formules du calcul propositionnel ;
— $\text{Var}\,(\varphi_1 \vee \varphi_2) = \text{Var}\,(\varphi_1) \cup \text{Var}\,(\varphi_2)$ si φ_1 et φ_2 sont deux formules du calcul propositionnel.

Exemple 3

Nous avons :
— $\text{Var}\,(A) = \{A\}$.
— $\text{Var}\,((A \wedge (\neg A))) = \{A\}$.
— $\text{Var}\,((A \vee B) \wedge (\neg A)) = \{A, B\}$.
— $\text{Var}\,((\neg((\neg A) \wedge \bot)) \wedge A) = \{A\}$.

On prouve que cette définition correspond bien à la notion intuitive de variables d'une formule.

Proposition 1

Soit φ une formule. Les variables propositionnelles qui apparaissent dans φ sont exactement $\text{Var}\,(\varphi)$.

Démonstration. — En effet, A est la seule variable qui apparaît dans la formule A.
— De plus, aucune variable n'apparaît dans les formules $\top$ et $\bot$.
— Par définition, les symboles (,), et $\neg$ ne sont pas des variables. Les variables qui apparaissent dans la formule $\text{Var}\,((\neg\varphi))$ sont donc des variables de $\text{Var}\,(\varphi)$. Réciproquement tous les symboles de φ apparaissent

dans $(\neg\varphi)$, donc toutes les variables qui apparaissent dans φ apparaissent aussi dans $(\neg\varphi)$. Ainsi, les deux ensembles $\mathrm{Var}(\varphi)$ et $\mathrm{Var}((\neg\varphi))$ sont égaux.

— Par définition, les symboles (,), et $\wedge$ ne sont pas des variables. Les variables qui apparaissent dans la formule $(\varphi_1 \wedge \varphi_2)$ sont donc des variables de $\mathrm{Var}(\varphi_1)$ ou de $\mathrm{Var}(\varphi_2)$. Réciproquement tous les symboles de φ_1 ou dans φ_2 apparaissent dans $(\varphi_1 \wedge \varphi_2)$, donc toues les variables qui apparaissent dans φ_1 ou dans φ_2 apparaissent aussi dans $(\varphi_1 \wedge \varphi_2)$.

— Par définition, les symboles (,), et $\vee$ ne sont pas des variables. Les variables qui apparaissent dans la formule $(\varphi_1 \vee \varphi_2)$ sont donc des variables de $\mathrm{Var}(\varphi_1)$ ou de $\mathrm{Var}(\varphi_2)$. Réciproquement tous les symboles de φ_1 ou dans φ_2 apparaissent dans $(\varphi_1 \vee \varphi_2)$, donc toues les variables qui apparaissent dans φ_1 ou dans φ_2 apparaissent aussi dans $(\varphi_1 \vee \varphi_2)$.

$\square$

2 Sémantique

Nous voulons maintenant donner un sens à ces formules. Le but de la sémantique est définir formellement ce sens. Nous commençons par définir les règles de calculs qui donnent un sens aux formules sans variable. Puis nous définissons le sens des formules avec variables comme la valeur qu'elle prenne selon la valeur associée à ses variables. Nous pourrons alors définir si deux formules (différentes) ont la même signification.

Une variable propositionnelle peut prendre deux valeurs : tt (vrai) et ff (faux). Elles sont aussi parfois noté respectivement $\top$ et $\bot$, ou 1 et 0. On note $\mathcal{B}$ l'ensemble de ces deux valeurs. On veut donc définir la valeur de vérité d'une formule propositionnelle. Mais pour cela, il faut définir la valeur de vérité de chaque variable. On introduit donc la notion d'environnement.

Définition 6 – Environnement

Soit V un ensemble de variables propositionnelles. Un environnement sur V est une fonction de $V \to \mathcal{B}$.

Exemple 4

La fonction

$$\begin{aligned} \{A, B\} &\to \mathcal{B} \\ A &\mapsto \mathit{tt} \\ B &\mapsto \mathit{ff} \end{aligned}$$

est un environnement sur $\{A, B\}$.

Nous pouvons maintenant évaluer une formule φ sur un environnement σ en remplaçant toutes les occurrences des variables par le symbole qui leur est associé dans l'environnement et en calculant la sémantique de la formule sans variable obtenue. Nous donnons maintenant une définition plus formelle de l'évaluation d'une formule.

Définition 7 – Évaluation

Soient φ une formule et V un sur-ensemble fini de $\mathrm{Var}(\varphi)$. Notons $V := \{A_1, \ldots, A_n\}$. Soit σ un environnement sur V.
Nous définissons l'évaluation $[\varphi]_\sigma$ de φ sur l'environnement σ par induction de la manière suivante :

— $[\bot]_\sigma = \mathit{ff}$;
— $[\top]_\sigma = \mathit{tt}$;
— $[A_i]_\sigma = \sigma(A_i)$;
— $[(\varphi)]_\sigma = [\varphi]_\sigma$;
— $[(\neg\varphi)]_\sigma = \begin{cases} \mathit{tt} & \text{si } [\varphi]_\sigma = \mathit{ff}, \\ \mathit{ff} & \text{sinon} ; \end{cases}$
— $[(\varphi_1 \vee \varphi_2)]_\sigma = \begin{cases} \mathit{tt} & \text{si } [\varphi_1]_\sigma = \mathit{tt} \text{ ou si } [\varphi_2]_\sigma = \mathit{tt}, \\ \mathit{ff} & \text{sinon} ; \end{cases}$
— $[(\varphi_1 \wedge \varphi_2)]_\sigma = \begin{cases} \mathit{tt} & \text{si } [\varphi_1]_\sigma = \mathit{tt} \text{ et si } [\varphi_2]_\sigma = \mathit{tt}, \\ \mathit{ff} & \text{sinon.} \end{cases}$

Exemple 5

Soit σ, l'environnement sur $\{A, B\}$

$$A \mapsto \mathit{tt}$$
$$B \mapsto \mathit{ff}$$

— $[\top]_\sigma = \mathit{tt}$
— $[A]_\sigma = \mathit{tt}$
— $[(\neg A)]_\sigma = \mathit{ff}$
— $[(A \vee B)]_\sigma = \mathit{tt}$
— $[(A \wedge (\neg B))]_\sigma = \mathit{tt}$

Pour définir la sémantique des formules avec variables, nous allons regarder quelles valeurs elles prennent lorsque nous donnons une valeur Booléenne leurs variables. Nous remplaçons ainsi toutes les variables de la formule par le symbole $\bot$ ou par le symbole $\top$ et nous regardons la sémantique de la formule sans variables obtenue. Nous considérons toutes les combinaisons possibles pour la valeur des variables.

Définition 8 – Table de vérité

Soient φ une formule et V un sur-ensemble de $\mathrm{Var}(\varphi)$ non vide. La sémantique de la formule φ (paramétrée par V) est une fonction associant chaque environnement σ sur V à la valeur $[\varphi]_\sigma$ prise par φ pour l'environnement σ. Cette fonction est aussi appelée table de vérité. Elle est notée $[\![\varphi]\!]_V$.

Exemple 6

Nous donnons la table de vérité de la formule $(A \vee ((\neg B) \wedge C))$ sur l'ensemble de variables $\{A; B; C\}$.

$[A]_\sigma$	$[B]_\sigma$	$[C]_\sigma$	$[(\neg B)]_\sigma$	$[((\neg B) \wedge C)]_\sigma$	$[(A \vee ((\neg B) \wedge C))]_\sigma$
ff	ff	ff	tt	ff	ff
ff	ff	tt	tt	tt	tt
ff	tt	ff	ff	ff	ff
ff	tt	tt	ff	ff	ff
tt	ff	ff	tt	ff	tt
tt	ff	tt	tt	tt	tt
tt	tt	ff	ff	ff	tt
tt	tt	tt	ff	ff	tt

On remarque qu'il n'y a qu'un nombre fini de possibilités mais qu'il y a un nombre infini de formules avec un ensemble donnée de variables. Par conséquent, certaines formules ont la même table. On parle d'équivalence sémantique.

Définition 9 – Équivalence sémantique

Soit φ_1 et φ_2 deux formules du calcul propositionnel. Nous dirons que φ_1 et φ_2 sont équivalentes sur le point sémantique si et seulement si les deux tables de vérités coïncident ie. $[\![\varphi_1]\!]_{\mathrm{Var}(\varphi_1)\cup\mathrm{Var}(\varphi_2)} = [\![\varphi_2]\!]_{\mathrm{Var}(\varphi_1)\cup\mathrm{Var}(\varphi_2)}$. Dans ce cas, nous noterons $\varphi_1 \equiv \varphi_2$.

Exemple 7

Deux formules équivalentes n'ont pas nécessairement le même ensemble de variables. Ainsi, nous avons :

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

Démonstration. 1. Montrons que ces deux formules n'ont pas le même ensemble de variables.

Nous avons :

$$\mathrm{Var}\,(((A \wedge (\neg A)) \vee B)) = \{A, B\}$$

et :

$$\mathrm{Var}\,(B) = \{B\}.$$

Donc $A \in \mathrm{Var}\,(((A \wedge (\neg A)) \vee B))$ et $A \notin \mathrm{Var}\,(B)$. Puis $\mathrm{Var}\,(((A \wedge (\neg A)) \vee B)) \neq \mathrm{Var}\,(B)$.

2. Montrons que ces deux formules sont sémantiquement équivalentes.

$[A]_\sigma$	$[B]_\sigma$	$[(\neq A)]_\sigma$	$[(A \wedge (\neg A))]_\sigma$	$[((A \wedge (\neg A)) \vee B]_\sigma$	$[B]_\sigma$
ff	*ff*	*tt*	*ff*	*ff*	*ff*
ff	*tt*	*tt*	*ff*	*tt*	*tt*
tt	*ff*	*ff*	*ff*	*ff*	*ff*
tt	*tt*	*ff*	*ff*	*tt*	*tt*

D'où, $((A \wedge (\neg A)) \vee B) \equiv B$.

Ainsi les formules $((A \wedge (\neg A)) \vee B)$ et B sont sémantiquement équivalentes mais non pas le même ensemble de variables. □

Définition 10 – Tautologie

Soit φ une formule. Nous dirons que φ est une tautologie si et seulement si $\varphi \equiv \top$.

Exemple 8

La formule $\top$ est une tautologie.

Démonstration. Donnons la table de vérité de la formule $\top$ sur l'ensemble de variables $\{A\}$.

$[A]_\sigma$	$[\top]_\sigma$
ff	*tt*
tt	*tt*

□

Exemple 9

La formule $(A \vee (\neg A))$ est une tautologie.

Démonstration. Donnons la table de vérité de la formule $(A \vee (\neg A))$ sur l'ensemble de variables $\{A\}$.

$[A]_\sigma$	$[(\neg A)]_\sigma$	$[(A \vee (\neg A))]_\sigma$
ff	tt	tt
tt	ff	tt

□

Exemple 10

La formule $(\neg(A \wedge (\neg A)))$ est une tautologie

Démonstration. Donnons la table de vérité de la formule $(\neg(A \wedge (\neg A)))$ sur l'ensemble de variables $\{A\}$.

$[A]_\sigma$	$[(\neg A)]_\sigma$	$[(A \wedge (\neg A))]_\sigma$	$[(\neg(A \wedge (\neg A)))]_\sigma$
ff	tt	ff	tt
tt	ff	ff	tt

□

Définition 11 – Contradiction

Soit φ une formule. Nous dirons que φ est une contradiction si et seulement si $\varphi \equiv \perp$.

Exemple 11

La formule $\perp$ est une contradiction.

Démonstration. Donnons la table de vérité de la formule $\perp$ sur l'ensemble de variables $\{A\}$.

$[A]_\sigma$	$[\perp]_\sigma$
ff	ff
tt	ff

□

Exemple 12

La formule $(A \wedge (\neg A))$ est une contradiction.

Démonstration. Donnons la table de vérité de la formule $(A \vee (\neg A))$ sur l'ensemble de variables $\{A\}$.

$[A]_\sigma$	$[(\neg A)]_\sigma$	$[(A \wedge (\neg A))]_\sigma$
ff	tt	ff
tt	ff	ff

□

Exemple 13

La formule $(\neg(A \vee (\neg A)))$ est une contradiction.

Démonstration. Donnons la table de vérité de la formule $(\neg(A \vee (\neg A)))$ sur l'ensemble de variables $\{A\}$.

$[A]_\sigma$	$[(\neg A)]_\sigma$	$[(A \vee (\neg A))]_\sigma$	$[(\neg(A \vee (\neg A)))]_\sigma$
ff	tt	tt	ff
tt	ff	tt	ff

□

3 Autres connecteurs logiques

En théorie, il peut y avoir 4 (2^{2^1}) connecteurs unaires avec des sémantiques différentes, 16 (2^{2^2}) connecteurs binaires avec des sémantiques différentes, (puis 2^{2^3} connecteurs ternaires). Certains opérateurs ont une importance particulière, comme l'implication ou l'équivalence, puisqu'ils sont couramment utilisés dans le langage naturel. Les autres sont une curiosité qu'on ne développera pas ici.

On introduit un nouveau connecteur dans la construction des formules : l'implication.

Définition 12 – Implication

On introduit le connecteur $\Rightarrow$: $(\varphi_1 \Rightarrow \varphi_2)$ est une formule propositionnelle. On lui donne la sémantique :

$$[(\varphi_1 \Rightarrow \varphi_2)]_\sigma = \begin{cases} \mathit{tt} & \text{si } [\varphi_1]_\sigma = \mathit{ff} \text{ ou } [\varphi_2]_\sigma = \mathit{tt} \\ \mathit{ff} & \text{sinon}; \end{cases}$$

Proposition 2

Nous avons :

$$(\varphi_1 \Rightarrow \varphi_2) \equiv ((\neg\varphi_1) \vee \varphi_2).$$

Démonstration. Nous vérifions que pour tout environnement σ que, quel que soit l'évaluation des formules φ_1 et φ_2 sur l'environnement σ, l'évaluation des formules $(\varphi_1 \Rightarrow \varphi_2)$ et $((\neg\varphi_1) \vee \varphi_2)$ sur l'environnement σ donne toujours la même valeur.

$[\varphi_1]_\sigma$	$[\varphi_2]_\sigma$	$[(\neg\varphi_1)]_\sigma$	$[((\neg\varphi_1) \vee \varphi_2)]_\sigma$	$[(\varphi_1 \Rightarrow \varphi_2)]_\sigma$
ff	*ff*	*tt*	*tt*	*tt*
ff	*tt*	*tt*	*tt*	*tt*
tt	*ff*	*ff*	*ff*	*ff*
tt	*tt*	*ff*	*tt*	*tt*

□

Proposition 3

Nous remarquons que $(\perp \Rightarrow \varphi_1)$ est une tautologie

Démonstration. Nous vérifions que pour tout environnement σ que, quel que soit l'évaluation de la formule φ_1 sur l'environnement σ, l'évaluation de la formule $(\perp \Rightarrow \varphi_1)$ donne toujours la valeur *tt*.

$[\varphi_1]_\sigma$	$[\perp]_\sigma$	$[\perp \Rightarrow \varphi_1]_\sigma$
ff	*ff*	*tt*
tt	*ff*	*tt*

□

Définition 13 – Équivalence

On introduit le connecteur $\Leftrightarrow$: $(\varphi_1 \Leftrightarrow \varphi_2)$ est une formule propositionnelle. On lui donne la sémantique :

$$[(\varphi_1 \Leftrightarrow \varphi_2)]_\sigma = \begin{cases} tt & \text{si } [\varphi_1]_\sigma = [\varphi_2]_\sigma \\ ff & \text{sinon}; \end{cases}$$

Proposition 4

Nous avons :

$$(\varphi_1 \Leftrightarrow \varphi_2) \equiv ((\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1)).$$

Démonstration. Nous vérifions que pour tout environnement σ que, quel que soit l'évaluation des formules φ_1 et φ_2 sur l'environnement σ, l'évaluation des formules $(\varphi_1 \Leftrightarrow \varphi_2)$ et $((\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1))$ sur l'environnement σ donne toujours la même valeur.

$[\varphi_1]_\sigma$	$[\varphi_2]_\sigma$	$[(\varphi_1 \Rightarrow \varphi_2)]_\sigma$	$[(\varphi_2 \Rightarrow \varphi_1)]_\sigma$	$[((\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1))]_\sigma$	$[(\varphi_1 \Leftrightarrow \varphi_2)]_\sigma$
ff	*ff*	*tt*	*tt*	*tt*	*tt*
ff	*tt*	*tt*	*ff*	*ff*	*ff*
tt	*ff*	*ff*	*tt*	*ff*	*ff*
tt	*tt*	*tt*	*tt*	*tt*	*tt*

$\square$

4 Les règles de calcul

Proposition 5 – De Morgan

$$(\neg(\varphi_1 \vee \varphi_2)) \equiv ((\neg\varphi_1) \wedge (\neg\varphi_2))$$
$$(\neg(\varphi_1 \wedge \varphi_2)) \equiv ((\neg\varphi_1) \vee (\neg\varphi_2))$$

Démonstration. Nous vérifions que pour tout environnement σ que, quel que soit l'évaluation des formules φ_1 et φ_2 sur l'environnement σ, l'évaluation des formules $(\neg(\varphi_1 \vee \varphi_2))$ et $((\neg\varphi_1) \wedge (\neg\varphi_2))$ d'une part, et de $(\neg(\varphi_1 \wedge \varphi_2))$ et $\neg\varphi_1) \vee (\neg\varphi_2))$, d'autre part, sur l'environnement σ donne toujours la même valeur.

$[\varphi_1]_\sigma$	$[\varphi_2]_\sigma$	$[(\varphi_1 \vee \varphi_2)]_\sigma$	$[(\varphi_2 \wedge \varphi_1)]_\sigma$	$[(\neg(\varphi_1 \vee \varphi_2))]_\sigma$	$[(\neg(\varphi_1 \wedge \varphi_2))]_\sigma$
ff	*ff*	*ff*	*ff*	*tt*	*tt*
ff	*tt*	*tt*	*ff*	*ff*	*tt*
tt	*ff*	*tt*	*ff*	*ff*	*tt*
tt	*tt*	*tt*	*tt*	*ff*	*ff*

$[\varphi_1]_\sigma$	$[\varphi_2]_\sigma$	$[(\neg\varphi_1)]_\sigma$	$[(\neg\varphi_2)]_\sigma$	$((\neg\varphi_1) \vee (\neg\varphi_2))$	$((\neg\varphi_1) \wedge (\neg\varphi_2))$
ff	*ff*	*tt*	*tt*	*tt*	*tt*
ff	*tt*	*tt*	*ff*	*tt*	*ff*
tt	*ff*	*ff*	*tt*	*tt*	*ff*
tt	*tt*	*ff*	*ff*	*ff*	*ff*

□

Proposition 6 – Tiers exclu

$$(\neg(\neg\varphi_1)) \equiv \varphi_1$$

Démonstration. Nous vérifions que pour tout environnement σ que, quel que soit l'évaluation de la formule φ_1 sur l'environnement σ, l'évaluation des formules φ_1 et $(\neg(\neg\varphi_1))$ sur l'environnement σ donne toujours la même valeur.

$[\varphi_1]_\sigma$	$[(\neg\varphi_1)]_\sigma$	$[(\neg(\neg\varphi_1))]_\sigma$
ff	*tt*	*ff*
tt	*ff*	*tt*

□

Proposition 7

La relation $\equiv$ est une relation d'équivalence, c'est à dire :
- $\varphi_1 \equiv \varphi_1$ (réflexivité) ;
- si $\varphi_1 \equiv \varphi_2$ et $\varphi_2 \equiv \varphi_3$ alors $\varphi_1 \equiv \varphi_3$ (transitivité) ;
- si $\varphi_1 \equiv \varphi_2$ alors $\varphi_2 \equiv \varphi_1$ (symétrie).

Démonstration. On se donne φ_1, φ_2 et φ_3 des formules propositionnelles et V un sur ensemble des variables de ces formules.

Réflexivité : pour tout environnement σ sur V, $[\varphi_1]_\sigma = [\varphi_1]_\sigma$

Transitivité : on suppose $\varphi_1 \equiv \varphi_2$ et $\varphi_2 \equiv \varphi_3$. Par conséquent, pour tout environnement σ sur V, on a $[\varphi_1]_\sigma = [\varphi_2]_\sigma$ et $[\varphi_2]_\sigma = [\varphi_3]_\sigma$. On en déduit donc $[\varphi_1]_\sigma = [\varphi_3]_\sigma$. Par conséquent, $\varphi_1 \equiv \varphi_3$.

Symétrie : on suppose $\varphi_1 \equiv \varphi_2$. Par conséquent, pour tout environnement σ sur V, on a $[\varphi_1]_\sigma = [\varphi_2]_\sigma$. On en déduit $[\varphi_2]_\sigma = [\varphi_1]_\sigma$ donc $\varphi_2 \equiv \varphi_1$. □

Corollaire 1

$$(\neg(\varphi_1 \Rightarrow \varphi_2)) \equiv (\varphi_1 \wedge (\neg\varphi_2))$$

Démonstration. On peut donner les tables comme avant, ou simplement utiliser les propositions précédentes pour obtenir le résultat.

$$\begin{aligned}(\neg(\varphi_1 \Rightarrow \varphi_2)) &\equiv (\neg((\neg\varphi_1) \vee \varphi_2)) \\ &\equiv ((\neg(\neg\varphi_1)) \wedge (\neg\varphi_2)) \\ &\equiv (\varphi_1 \wedge (\neg\varphi_2))\end{aligned}$$

La conclusion s'obtient par transitivité de $\equiv$ □