Introduction à la logique

Le calcul propositionnel

Lionel Vaux Auclair

I2M, université d'Aix-Marseille

Logique HUGO@Polytech, 2021–2022

Logique ?

La logique [est] l'étude des **règles formelles que doit respecter toute argumentation correcte.**

Logique ?

> La logique [est] l'étude des **règles formelles que doit respecter toute argumentation correcte.**
>
> Article Logique de Wikipédia en français — CC-BY-SA 3.0

- qu'est-ce que ça veut dire ?
- à quoi est-ce que ça sert ?

Les trois âges de la logique

Depuis l'antiquité : comprendre les règles du raisonnement humain
(logique philosophique)

Depuis le 19e siècle : fonder les mathématiques
(logique mathématique)

Depuis le milieu du 20e siècle : outils logiques pour l'informatique
(représentation de l'information et du calcul, IA, démonstration automatique, assistée et/ou certifiée)

Les trois âges de la logique

Depuis l'antiquité : comprendre les règles du raisonnement humain
(logique philosophique)

Depuis le 19e siècle : fonder les mathématiques
(logique mathématique)

Depuis le milieu du 20e siècle : outils logiques pour l'informatique
(représentation de l'information et du calcul, IA, démonstration automatique, assistée et/ou certifiée)

Syntaxe et sémantique

Deux gros mots chargés d'histoire :

- syntaxe : des expressions (les termes, les formules, les programmes)
- sémantique : leurs valeurs (les éléments d'une structure, les valeurs de vérité, les fonctions calculées)

Syntaxe et sémantique

Deux gros mots chargés d'histoire :

- syntaxe : des expressions (les termes, les formules, les programmes)
- sémantique : leurs valeurs (les éléments d'une structure, les valeurs de vérité, les fonctions calculées)

En fait, vous connaissez déjà

Un polynôme c'est une expression avec des $+$ et des $\times$

On fixe un ensemble $\mathcal{V}$ de variables (ou indéterminées).

Une expression polynomiale est une expression construite à partir des variables et des constantes 0 (le polynôme nul) et 1 (le polynôme constant unitaire), par application des constructeurs binaires $+$ et $\times$, et du constructeur unaire $-$.

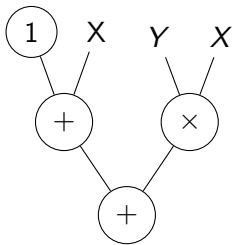
Exemple

$$0 \qquad -(-X) \qquad X+(Y+Z) \qquad (X\times Y)+(X+1)$$

À ce stade, $X \neq -(-X)$ et $X+(Y+Z) \neq (X+Y)+Z$, vu que ça ne s'*écrit* pas pareil.

Un polynôme c'est un arbre avec des $+$ et des $\times$

$(X \times Y) + (X + 1) =$



Valeur d'une expression polynomiale

On peut définir la valeur d'un polynôme, *en fonction* de la valeur de ses variables, prise dans un anneau A quelconque.

Soit $v : \mathcal{V} \to \mathrm{A}$ (une valeur pour chaque variable), on étend v à tous les polynômes en remplaçant chaque variable X par $v(X)$ et en calculant dans A.

Exemple

Avec $P = (X \times Y) + (X + 1)$, $v(X) = 3$ et $v(Y) = 2$ (dans $\mathbf{Z}$) :

$$v(P) = 3 \times 2 + 3 + 1 = 10$$

Fonctions polynomiales

Une fonction polynomiale, c'est une fonction de la forme

$$\begin{aligned} \hat{P} : \mathrm{A}^{\mathcal{V}} &\to \mathrm{A} \\ v &\mapsto v(P) \end{aligned}$$

avec P un polynôme fixé.

- Ça dépend de l'ensemble de coefficients A qu'on considère.
- En général, on peut avoir $\hat{P} = \hat{Q}$ avec $P \neq Q$.

On dit que P et Q sont équivalentes et on note $P \equiv Q$ si $\hat{P} = \hat{Q}$ (dans tout anneau A).

Exemple

$$(X \times Y) + (X + 1) \equiv (X \times (Y + 1)) + 1$$

Calcul propositionnel

On fixe un ensemble infini dénombrable $\mathcal{V}_p$ de variables propositionnelles, notées $X, Y, \ldots$ (ce sont les formules atomiques ou indéterminées).

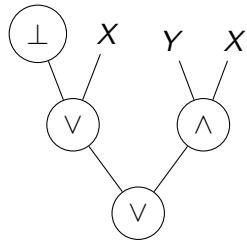
Définition

L'ensemble $\mathcal{F}_p$ des formules propositionnelles est l'ensemble des expressions de la forme :

$$A, B, \ldots ::= X \mid A \wedge B \mid A \vee B \mid \neg A \mid A \Rightarrow B \mid A \Leftrightarrow B \mid \top \mid \bot$$

Exemple

$A \coloneqq (X \wedge Y) \vee (X \vee \bot) =$



Les tables de la véritable vérité vraie (100% garanti)

La validité d'une formule propositionnelle se *calcule* dans $\mathbf{B} = \{0, 1\}$:

A	B	$A \wedge B$	$A \vee B$	$\neg A$	$A \Rightarrow B$	$A \Leftrightarrow B$	$\top$	$\bot$
0	0	0	0	1	1	1	1	0
0	1	0	1	1	1	0	1	0
1	0	0	1	0	0	0	1	0
1	1	1	1	0	1	1	1	0

- Chaque connecteur est interprété comme une opération booléenne.
- La valeur d'une expression se déduit de celle de ses sous-expressions.

Les tables de la véritable vérité vraie (100% garanti)

La validité d'une formule propositionnelle se *calcule* dans $\mathbf{B} = \{0, 1\}$:

A	B	$A \wedge B$	$A \vee B$	$\neg A$	$A \Rightarrow B$	$A \Leftrightarrow B$	$\top$	$\bot$	X
0	0	0	0	1	1	1	1	0	?
0	1	0	1	1	1	0	1	0	?
1	0	0	1	0	0	0	1	0	?
1	1	1	1	0	1	1	1	0	?

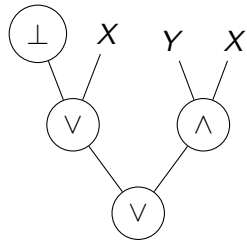
- Chaque connecteur est interprété comme une opération booléenne.
- La valeur d'une expression se déduit de celle de ses sous-expressions.

$\rightsquigarrow$ Elle dépend donc du contexte, qui est donné par une distribution de valeurs de vérités $d : \mathcal{V}_p \to \mathbf{B}$:
d s'étend en une valuation de toutes les formules $A \in \mathcal{F}_p \mapsto d(A) \in \mathbf{B}$.

- On note $d \models A$ (d satisfait A) si $d(A) = 1$.

Exemple

$A := (X \wedge Y) \vee (X \vee \bot) =$



- $d \models A$ ssi $d(X) = 1$

Valide = toujours vrai

- A est valide si $d \models A$ pour toute dvv d, et alors on note $\models A$
  (on dit aussi que A est une tautologie)
- A est contradictoire si $d \not\models A$ pour toute dvv d, c'est-à-dire si $\models \neg A$
  (on dit aussi que A est une antilogie)
- A est satisfaisable s'il existe une dvv d telle que $d \models A$ pour toute dvv d.

Valide = toujours vrai

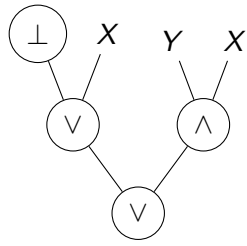
- A est valide si $d \models A$ pour toute dvv d, et alors on note $\models A$
  (on dit aussi que A est une tautologie)
- A est contradictoire si $d \not\models A$ pour toute dvv d, c'est-à-dire si $\models \neg A$
  (on dit aussi que A est une antilogie)
- A est satisfaisable s'il existe une dvv d telle que $d \models A$ pour toute dvv d.

Attention

- $d \not\models A$ ssi $d \models \neg A$
- $\models \neg A$ ssi A n'est pas satisfaisable
- $\models A$ ssi $\neg A$ n'est pas satisfaisable
- on peut avoir $\not\models A$ et $\not\models \neg A$: c'est-à-dire que A et $\neg A$ sont satisfaisables

Exemple

$A \coloneqq (X \wedge Y) \vee (X \vee \bot) =$



- $d \models A$ ssi $d(X) = 1$
- A est donc satisfaisable, mais n'est ni valide ni contradictoire

Priorités et associations

On donne les priorités

$$\{\neg\} > \{\wedge, \vee\} > \{\Rightarrow\} > \{\Leftrightarrow\}$$

donc on écrit par exemple

$$\neg A \vee B \Leftrightarrow A \Rightarrow B \quad = \quad ((\neg A) \vee B) \Leftrightarrow (A \Rightarrow B).$$

On note

$$A \wedge B \wedge C = (A \wedge B) \wedge C$$
$$A \vee B \vee C = (A \vee B) \vee C$$
$$A \Rightarrow B \Rightarrow C = A \Rightarrow (B \Rightarrow C)$$

donc on écrit par exemple

$$A \wedge B \wedge C \Rightarrow A \wedge B \Rightarrow B \wedge C \quad = \quad (A \wedge B) \wedge C \Rightarrow (A \wedge B \Rightarrow B \wedge C).$$

Exerçons-nous

⇝ Exercices 1 et 2

Équivalence

On dit que A et B sont équivalentes ssi, pour toute dvv d, $d(A) = d(B)$:
on note $A \equiv B$.

Lemme

On a $A \equiv B$ ssi $\models A \Leftrightarrow B$.

Démonstration: On a $d(A \Leftrightarrow B) = 1$ ssi $d(A) = d(B)$. □

Remarque

- Chaque formule A définit une fonction booléenne $\hat{A} : d \mapsto d(A)$.
- On a $A \equiv B$ ssi $\hat{A} = \hat{B}$.

Équivalences classiques

Le quotient $\mathcal{F}_p/\equiv$ est un treillis ...

$$A \wedge B \equiv B \wedge A \quad (A \wedge B) \wedge C \equiv A \wedge (B \wedge C) \quad A \wedge (A \vee B) \equiv A$$
$$A \vee B \equiv B \vee A \quad (A \vee B) \vee C \equiv A \vee (B \vee C) \quad A \vee (A \wedge B) \equiv A$$
$$\text{(et donc : } \quad A \wedge A \equiv A \quad A \vee A \equiv A)$$

... distributif ...

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

... borné et complémenté...

$$A \wedge \top \equiv A \quad A \vee \bot \equiv A \quad A \wedge \neg A \equiv \bot \quad A \vee \neg A \equiv \top$$
$$\text{(et donc : } \quad A \wedge \bot \equiv \bot \quad A \vee \top \equiv \top)$$

$\rightsquigarrow$ c'est une algèbre de Boole

Équivalences classiques

Lois de De Morgan :

$$\neg\neg A \equiv A \quad \neg(A \wedge B) \equiv \neg A \vee \neg B \quad \neg(A \vee B) \equiv \neg A \wedge \neg B$$
$$\neg\top \equiv \bot \quad \neg\bot \equiv \top$$

Autour de l'implication :

$$A \Rightarrow B \equiv \neg A \vee B \equiv \neg B \Rightarrow \neg A$$
$$A \wedge B \Rightarrow C \equiv A \Rightarrow B \Rightarrow C$$
$$\neg(A \Rightarrow B) \equiv A \wedge \neg B$$
$$\neg A \equiv A \Rightarrow \bot$$

Équivalences classiques

Lois de De Morgan :

$$\neg\neg A \equiv A \quad \neg(A \wedge B) \equiv \neg A \vee \neg B \quad \neg(A \vee B) \equiv \neg A \wedge \neg B$$
$$\neg\top \equiv \bot \quad \neg\bot \equiv \top$$

Autour de l'implication :

$$A \Rightarrow B \equiv \neg A \vee B \equiv \neg B \Rightarrow \neg A$$
$$A \wedge B \Rightarrow C \equiv A \Rightarrow B \Rightarrow C$$
$$\neg(A \Rightarrow B) \equiv A \wedge \neg B$$
$$\neg A \equiv A \Rightarrow \bot$$

$\rightsquigarrow$ Vérifiez

Jeux de connecteurs complets

Théorème

Toute formule propositionnelle est équivalente à une formule n'utilisant que les connecteurs $\wedge$ et $\neg$ (ou $\vee$ et $\neg$, ou encore $\Rightarrow$ et $\neg$, ou ...).

Démonstration: Faisons le avec $\wedge$ et $\neg$. On a :

$$A \vee B \equiv \neg(\neg A \wedge \neg B)$$
$$A \Rightarrow B \equiv \neg A \vee B \equiv \neg(A \wedge \neg B)$$
$$A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A) \equiv \neg(A \wedge \neg B) \wedge \neg(B \wedge \neg A)$$
$$\bot \equiv X \wedge \neg X$$
$$\top \equiv \neg(X \wedge \neg X)$$

On montre le résultat par récurrence sur la taille des formules : pour les variables propositionnelles c'est direct ; dans les autres cas, on conclut par l'hypothèse de récurrence, en appliquant les transformations ci-dessus pour les connecteurs autres que $\wedge$ et $\neg$. □

Donc, à équivalence près, on peut raisonner sur $\mathcal{F}_{\wedge,\neg} ::= X \mid A \wedge B \mid \neg A$.

Expressions booléennes

On retrouve *partout* en informatique les expressions booléennes :

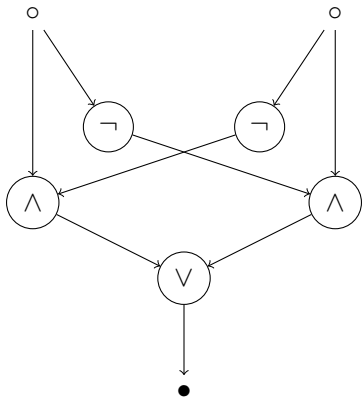
$$\mathcal{F}_b ::= X \mid A \wedge B \mid A \vee B \mid \neg A \mid \top \mid \bot.$$

- en programmation (conditionnelles, boucles)
- matérialisées dans des circuits (portes = connecteurs)
- dans les bases de données relationnelles (SELECT)
- pour la résolution de contraintes (on représente un problème par une formule : une solution = une dvv qui satisfait la formule)

Circuits booléens

Un circuit booléen est un graphe dirigé acyclique, dont les nœuds sont des entrées, des sorties, et des connecteurs (en respectant les degrés d'entrée).

Exemple



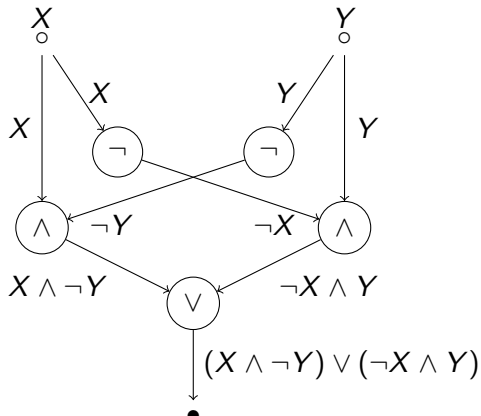
Circuits booléens

Un circuit booléen est un graphe dirigé acyclique, dont les nœuds sont des entrées, des sorties, et des connecteurs (en respectant les degrés d'entrée).
Si on associe une variable à chaque entrée, chaque arc dans un circuit booléen définit une expression booléenne :

- l'arc issu d'un nœud d'entrée X reçoit l'expression X ;
- l'arc issu d'un nœud $\top$ reçoit l'expression $\top$,
- l'arc issu d'un nœud $\bot$ reçoit l'expression $\bot$,
- l'arc issu d'un nœud $\wedge$ reçoit l'expression $A \wedge B$, si A et B sont les expressions des arcs entrants ;
- l'arc issu d'un nœud $\vee$ reçoit l'expression $A \vee B$, si A et B sont les expressions des arcs entrants ;
- l'arc issu d'un nœud $\neg$ reçoit l'expression $\neg A$, si $\neg A$ est l'expression de l'arc entrant.

Chaque sortie reçoit donc une expression booléenne, calculée par le circuit.

Exemple



Quelques questions pertinentes

- une expression booléenne est-elle tautologique ? ou alors admet-elle une expression équivalente plus simple ?
- pour une fonction booléenne donnée, quelle est l'expression booléenne la plus simple ?
- comment construire un circuit qui réalise une fonction booléenne donnée ? comment obtenir le circuit le plus petit ?
- comment vérifier automatiquement et efficacement si une formule est valide/satisfaisable ?
- comment certifier la validité d'une formule ? ou l'équivalence entre deux formules ?

On explorera certaines de ces pistes par la suite.

Exerçons-nous

⇝ Reste des exercices