Logique propositionnelle

option informatique

Logiques ?

- On attribue à Aristote l'une des premières tentatives de formalisation du raisonnement à l'aide de ce qu'on appelle la logique des syllogismes.
- L'objet de la **logique** est la formalisation du raisonnement à l'aide de règles.
- La **logique** distingue la manipulation formelle des objets qu'elle manipule (**aspects syntaxiques**) et leur signification ou interprétation (**aspects sémantiques**).
- La **logique** manipule des objets appelés **formules logiques** construites à partir de **propositions atomiques** et de **connecteurs logiques**.

Logiques ?

- La **logique propositionnelle** ne décrit qu'une partie des raisonnements pour lesquels les formules ne peuvent prendre que deux valeurs **vrai** ou **faux**.
- En particulier, elle est incapable d'exprimer l'existence d'un objet ayant une propriété donnée ou encore le fait que plusieurs objets partagent une même propriété.
- La **logique des prédicats** le permet mais n'est pas étudiée dans ce cours.

Syntaxe

Variable propositionnelle

- On suppose donné un ensemble dénombrable $V = \{v_i, i \in \mathbb{N}\}$ dont les éléments sont des objects distincts appelés **variables propositionnelles**.
- Ces variables représentent des énoncés ou des affirmations qui sont soit vrais, soit faux.
- Elles n'ont en réalité aucun caractère variable. Mais la logique a consacré cette dénomination.

Définition 1 (Variable propositionnelle)

Une **variable propositionnelle** (ou **proposition atomique**) est une assertion qui ne peut prendre que deux états possibles appelés **valeurs de vérité**.

Variable propositionnelle

- La **logique propositionnelle** s'intéresse aux énoncés construits à l'aide de plusieurs variables logiques reliées par des **connecteurs**.
- La construction d'énoncés complexes Des règles précises précisent L'écriture d'énoncés complexes et pertinents suivant des règles de construction d'énoncés complexes relève de la **syntaxe** d'une expression logique.
- Les valeurs de vérité sont désignées par **vrai** ou **faux**. selon que chacune des variables prend une **valeur de vérité vrai** ou **faux**.
- Ce point, développé plus loin, relève de la **sémantique** d'une expression logique.

Variable propositionnelle

- Deux variables jouent un rôle particulier en logique et sont parfois désignés sous le nom de **constantes logiques**. Notées $\top$ et $\bot$, elles désignent respectivement une variable **toujours vraie** ou **toujours fausse.**
- Dans la suite de l'exposé, l'ensemble V est supposé contenir ces deux constantes logiques. Les variables propositionnelles seront désignées par une lettre minuscule : p, q, r, etc.

Conneteur logique

Les variables propositionnelles sont les briques élémentaires pour construire des énoncés logiques plus élaborés, le lien étant réalisé par des **connecteurs logiques**.

Définition 2 (Conneteur logique)

Un **connecteur logique** est un symbole modifiant l'état d'une proposition logique (connecteur unaire) ou établissant une liaison entre deux propositions logiques (connecteur binaire).

Conneteur logique

- Un **connecteur logique unaire**, dit de négation et représenté par $\neg$, agit sur une seule variable.
- Deux **connecteurs binaires**[1] dits de **conjonction**, de **disjonction** représentés respectivement par les symboles $\wedge$, $\vee$ agissent sur deux variables.
- Les connecteurs $\wedge$, $\vee$ sont liés aux notions de **et**, **ou** logiques. C'est pourquoi on les lit souvent de ces dernières façons. De même, le connecteur $\neg$ est lu **non**.

1. D'autres connecteurs binaires existent comme l'**implication** $\rightarrow$ et l'**équivalence** $\leftrightarrow$. Liés à l'**implication** et l'**équivalence** entre formules logiques, ils s'expriment à l'aide des connecteurs $\wedge$ et $\vee$.

Formule logique

- On définit l'**alphabet** $\Sigma = V \cup \{\neg, \wedge, \vee,), (\}$.
- Σ est l'ensemble de toutes les variables propositionnelles, des connecteurs logiques et des parenthèses qui permettent l'écriture de **formules logiques** non ambiguës.
- De manière informelle, une **formule logique**[2] est un mot construit sur l'alphabet Σ. Ce mot doit cependant respecter des **règles syntaxiques**. Ce qui mène naturellement à la définition inductive d'une **formule logique**.

2. On parle également de **formule propositionnelle**, de **proposition logique** ou d'**expression logique**.

Formule logique

Définition 3 (Formule logique)

Soit $\mathcal{L}_0 = V$ et pour tout entier naturel n :

$$\mathcal{L}_{n+1} = \mathcal{L}_n \cup \{\neg\varphi \mid \varphi \in \mathcal{L}_n\} \cup \{(\varphi \wedge \psi), (\varphi \vee \psi) \mid \varphi, \psi \in \mathcal{L}_n\}$$

Une **formule logique** est un mot du langage $\mathcal{L} = \bigcup_{n \in \mathbb{N}} \mathcal{L}_n$ défini sur l'alphabet Σ.

Cette définition fixe un cadre pour écrire des formules logiques qu'on qualifie de **syntaxiquement correctes**. Toute expression comportant des variables propositionnelles, des parenthèses et des connecteurs logiques qui ne satisfait pas les conditions de cette définition sera déclarée incorrecte d'un point de vue syntaxique.

Exemple

Considérons trois variables propositionnelles.

v_1 : Il fait jour. v_2 : Il fait nuit. v_3 : Le soleil brille.

et l'alphabet Σ défini par :

$$\Sigma = \{v_1, v_2, v_3,), (, \neg, \wedge, \vee\}$$

Exemple

- L'expression $(v_1 \vee v_2)$, **syntaxiquement correcte**, est une **formule logique**.
  Elle contient les deux variables propositionnelles v_1 et v_2 reliées par le connecteur binaire $\vee$, le tout entouré des parenthèses (et) dans cet ordre.
- Les expressions $)v_1 \vee v_2($, $)v_1 \vee v_2)$, $(v_1 \vee v_2($, $v_1 \vee v_2)$, **syntaxiquement incorrectes**, ne sont pas des formules logiques.

Exemple

- La présence des parenthèses peut sembler superflue. Pour des expressions plus compliquées, elles sont indispensables pour éviter toute erreur syntaxique ou toute ambiguité.
- Par exemple, comment lire l'expression $v_1 \land v_3 \lor v_2$? L'absence de parenthèses rend la lecture impossible sauf à définir des règles de priorité sur les connecteurs logiques.
- Les expressions $((v_1 \land v_3) \lor v_2)$ et $(v_1 \land (v_3 \lor v_2))$ sont syntaxiquement correctes. Elles comportent toutes les parenthèses qui respectent la définition, même les parenthèses externes.

Exemple

À ce niveau de l'exposé, on peut s'interroger sur le sens à donner à formules logiques syntaxiquement correctes comme $((v_1 \wedge v_3) \vee v_2)$ et $(v_1 \wedge (v_3 \vee v_2))$. Mais l'objet de cette partie n'est justement pas le sens des expressions mais seulement leur syntaxe. Si l'affirmation :

(Il fait jour. et Le soleil brille.) ou Il fait nuit.

semble porteuse de sens dans le langage commun, l'expression suivante est plus délicate à comprendre !

Il fait jour. et (Le soleil brille. ou Il fait nuit.)

En outre, le **ou** est-il inclusif ou exclusif ? En langage courant, il peut être tantôt l'un, tantôt l'autre suivant le contexte ! Ces points sont discutés dans la partie consacrée à la sémantique des **formules logiques**.

Fermeture

Théorème 4

$\mathcal{L}$ est le plus petit langage sur l'alphabet Σ vérifiant :

1. *toute variable propositionnelle est une formule ;*
2. *si φ est une formule alors $\neg\varphi$ est une formule ;*
3. *si φ et ψ sont des formules alors pour tout connecteur binaire $\square$, $\varphi \square \psi$ est une formule.*

Démonstration

Soit $\mathcal{M}$ un langage sur A vérifiant 1, 2 et 3. On a $\mathcal{L}_0 = V \subset \mathcal{M}$ par le point 1. Soit n un entier naturel et supposons $\mathcal{L}_n \subseteq \mathcal{M}$. Par les propriétés de stabilité 2 et 3, on a immédiatement $\mathcal{L}_{n+1} \subseteq \mathcal{M}$. On a donc montré par récurrence que pour tout entier naturel n, $\mathcal{L}_n \subseteq \mathcal{M}$. Ainsi, $\mathcal{L} \subseteq \mathcal{M}$.
Montrons maintenant que $\mathcal{L}$ vérifie 1, 2 et 3. La propriété 1 est clairement vérifiée. Montrons par exemple 3 (2 se traite de façon identique) : soient $\varphi, \psi \in \mathcal{L}$. Soit $\square \in C_2$. Il existe deux entiers m et n tels que $\varphi \in \mathcal{L}_m$ et $\psi \in \mathcal{L}_n$. Prenons par exemple $m \leqslant n$. La suite de langages $(\mathcal{L}_n)_{n\in\mathbb{N}}$ est croissante au sens de l'inclusion, donc $\varphi, \psi \in \mathcal{L}_n$ et ainsi $(\varphi \square \psi) \in \mathcal{L}_{n+1} \subset \mathcal{L}$.

Fermeture

- Les propriétés 1, 2 et 3 expriment qu'avec des formules des variables et des connecteurs on ne peut fabriquer que des formules.
- La propriété de minimalité de $\mathcal{L}$ indique que toute formule est composée, selon des règles précises, à partir de connecteurs et d'autres formules.
- Ce résultat constitute le **théorème de lecture unique des formules**.

Lecture unique

Théorème 5 (Lecture unique des formules)

Toute formule φ vérifie une seule des trois situations suivantes.

- *φ est une variable propositionnelle.*
- *Il existe une unique formule ψ telle que $\varphi = \neg\psi$.*
- *Il existe une unique formule ψ_1, une unique formule ψ_2 et un unique connecteur binaire $\square$ tels que $\varphi = (\psi_1 \square \psi_2)$.*

Ainsi, étant donnée une formule logique, il n'en existe qu'une seule décomposition en formules **plus élémentaires**, où ce que l'on appelle **élémentaire** mériterait quelques précisions[3].

3. Le livre de René Cori et Daniel Lascar, **Logique mathématique 1 - Calcul propositionnel, algèbre de Boole, calcul des prédicats** chez Dunod (Ed. 2003), complète cet exposé. Une démonstration du théorème peut y être trouvée.

Longueur d'une formule

En fait, à chaque formule on peut associer des nombres qui mesurent son degré de simplicité. Plus ces nombres sont petits, plus la formule est simple. Nous allons nous concentrer sur deux exemples de tels nombres : la **longueur** et la **hauteur** d'une formule.

Définition 6

La longueur d'une formule φ de $\mathcal{L}$, notée $|\varphi|$, est le nombre de lettres qui composent le mot φ.

Exemple

- Il n'existe aucune formule de longueur 0.
- Une formule est de longueur 1 si et seulement si c'est une variable propositionnelle.
- Si v_1, v_2 et v_3 sont des variables propositionneles, la formule $\neg((v_1 \wedge v_3) \vee v_2)$ est de longueur 10.

Hauteur d'une formule

Définition 7

La hauteur d'une formule φ de $\mathcal{L}$, notée $h(\varphi)$, est le plus petit entier naturel n tel que $\varphi \in \mathcal{L}_n$.

Si φ est une variable propositionnelle, $h(\varphi) = 0$. Si φ n'est pas une variable propositionnelle, $h(\varphi) > 0$. Par minimalité, $h(\varphi)$ est le seul entier vérifiant $\varphi \notin \mathcal{L}_{h(\varphi)-1}$ et $\varphi \in \mathcal{L}_{h(\varphi)}$. En outre, par croissance de la suite des $\mathcal{L}_k$, pour tout entier k plus grand que h, on a $\varphi \in \mathcal{L}_k$.

Exemple

Si x et y sont des variables propositionnelles, la formule $\varphi = ((x \wedge (x \rightarrow y)) \rightarrow y$ est de hauteur 3. Noter le rôle du théorème de lecture unique dans l'établissement de ce résultat. Sans sa connaissance, on ne pourrait écrire que $h(\varphi) \leqslant 3$.

Arbre de la syntaxe abstraite

À une formule logique peut être associé un arbre défini inductivement :

- ses nœuds sont étiquetés par des connecteurs logiques ou des variables propositionnelles ;
- dont les feuilles sont des variables propositionnelles, et les noeuds, soit unaires soit binaires, sont des connecteurs. D'après le théorème de lecture unique, cet arbre est unique, appelé **arbre syntaxique**.

Arbre de la syntaxe abstraite

Définition 8

L'**arbre de la syntaxe abstraite** associé à une formule est

- l'arbre de la syntaxe abstraite associé à une variable propositionnelle possède une unique feuille, étiquetée par cette variable ;
- pour toute formule φ, l'arbre de la syntaxe abstraite associé à $\neg\varphi$ a une racine étiquetée par $\neg$ et un unique enfant qui est l'arbre syntaxique associé à φ ;
- pour toutes formules φ et ψ et tout connecteur binaire $\square \in \{\wedge, \vee\}$, l'arbre de la syntaxe abstraite associé à $(\varphi \square \psi)$ a une racine étiquetée par $\square$, un sous-arbre gauche qui est l'arbre de la syntaxe abstraite associé à φ et un sous-arbre droit qui est l'arbre de la syntaxe abstraite associé à ψ.

SÉMANTIQUE

Sémantique ?

- En linguistique, la **syntaxe** désigne le **signifiant** d'un énoncé. La **sémantique** désigne son **signifié**.
- Il existe entre la **sémantique** et la **syntaxe** le même rapport qu'entre la **forme** et le **fond**.
- La **syntaxe** est le support de la **sémantique**.

Sémantique ?

En logique propositionnelle, la **sémantique** s'attache à définir la **valeur de vérité** d'une **formule syntaxiquement correcte**, à savoir son caractère **vrai** ou **faux**. Pour ce faire, il convient :

- d'attribuer une valeur de vérité à chaque variable propositionnelle ;
- de définir les règles d'interprétation d'un connecteur ;
- de définir la valeur de vérité de la formule.

Distribution de vérité

- Dans la suite de l'exposé, on appelle **ensemble des booléens** $\mathcal{B}$ l'ensemble $\{0, 1\}$.
- L'usage a consacré les notations 0 et 1 pour désigner les éléments de $\mathcal{B}$ mais d'autres notations sont possibles comme **F** et **V**, ou encore `false` et `true`.
- Le booléen 0 est associé à la notion de **faux** alors que le booléen 1 est associé à celle de **vrai**.

Définition 9 (Distribution de vérité)

On appelle distribution de vérité [a] toute fonction $\gamma : V \to \mathcal{B}$.

a. On dit également **environnement**, **valuation** ou encore **contexte**.

On note Γ l'ensemble des distributions de vérité, c'est-à-dire $\mathcal{B}^V$.

Fonctions booléennes

Définition 10 (Fonction booléenne)

On appelle **fonction booléenne à n arguments** toute fonction de $\mathcal{B}^n$ dans $\mathcal{B}$.

- La fonction $F_\neg : \mathcal{B} \to \mathcal{B}$ est définie par :

$$F_\neg(0) = 1 \qquad F_\neg(1) = 0$$

- La fonction $F_\wedge : \mathcal{B}^2 \to \mathcal{B}$ est définie par :

$$F_\wedge(x,y) = \begin{cases} 1 & \text{si et seulement si } x = 1 \text{ et } y = 1 \\ 0 & \text{sinon} \end{cases}$$

- La fonction $F_\vee : \mathcal{B}^2 \to \mathcal{B}$ est définie par :

$$F_\vee(x,y) = \begin{cases} 0 & \text{si et seulement si } x = 0 \text{ et } y = 0 \\ 1 & \text{sinon} \end{cases}$$

Fonctions booléennes

Il est possible d'utiliser les opérations sur les entiers pour évaluer ces fonctions de manière moins verbeuse. Pour tout couple $(x, y) \in \mathcal{B}^2$, on peut écrire :

$$\begin{cases} F_\neg(x) = 1 - x \\ F_\wedge(x, y) = xy \\ F_\vee(x, y) = x + y - xy = 1 - (1 - x)(1 - y) = \max(x, y) \end{cases}$$

Évaluation d'une formule

La **valeur de vérité d'une formule** peut être définie inductivement à partir de celle de ses variables propositionnelles et des fonctions booléennes précédentes.

Définition 11 (Évaluation d'une formule)

Pour toute distibution de vérité γ, la **fonction d'évaluation d'une formule** $\gamma^* : \mathcal{L} \to \mathcal{B}$ se définit par induction structurelle.

- Pour toute variable propositionnelle v : $\gamma^*(v) = \gamma(v)$.
- Pour toute formule φ : $\gamma^*(\neg\varphi) = F_\neg(\gamma^*(\varphi))$.
- Pour toutes formules φ, ψ et pour tout connecteur binaire $\square \in \{\wedge, \vee\}$: $\gamma^*(\varphi \,\square\, \psi) = F_\square(\gamma^*(\varphi), \gamma^*(\psi))$.

Évaluation d'une formule

- Pour une formule réduite à une variable propositionnelle, γ^* s'identifie à γ ; elle en est une extension.
- Par abus de notation, on confond parfois le deux notations. Pis encore, on identifie même une variable propositionnelle avec sa valeur de vérité ! Ce qui allège considérablement les notations mais attention à la rigueur.

- arrtes@gmail.com

Tables de vérité

- Étant donné une distribution de vérité γ et une formule φ, on peut montrer que la **valeur de $\gamma^*(\varphi)$** ne **dépend** que **de la valeur de γ en les variables propositionnelles** ayant une occurrence dans φ.
- Si les variables propositionnelles intervenant dans φ sont $v_1, v_2, \ldots, v_n$, il suffit de considérer les distributions de vérité restreintes à $\{v_1, v_2, \ldots, v_n\}$ pour connaître toutes les évaluations possibles de φ.
- Il existe exactement 2^n **restrictions de distributions de vérité** à cet ensemble de cardinal n. Toutes les évaluations possibles peuvent être résumées dans un tableau de $2n$ lignes, **la** table de vérité de la formule φ.

Tables de vérité

Cette table n'étant pas unique, nous adoptons les conventions suivantes.

- Dans les premières colonnes, les variables propositionnelles rangée dans un ordre logique, alphabétique par exemple.
- Dans les colonnes suivantes, les sous-formules de φ utiles à la lecture et à la compréhension de la table.
- Dans la dernière colonne, la formule φ.
- Sur chaque ligne, un environnement. Les lignes seront elles-aussi rangées dans un ordre logique, par exemple l'ordre d'une énumération en binaire dans l'ordre croissant pour les colonnes relatives aux variables.

Tables de vérité

x	$F_{\neg}(x)$
0	1
1	0

x	y	$F_{\vee}(x,y)$	$F_{\wedge}(x,y)$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Tables de vérité de $\neg$, $\wedge$, $\vee$

SATISFIABILITÉ

Satisfiabilité

Soit γ un distribution de vérité. On dit qu'une formule φ est satisfaite par γ lorsque $\gamma^*(\varphi) = 1$. On dit aussi que γ satisfait φ

Définition 12 (Satisfiabilité)

Une formule φ est **satisfiable** s'il existe une distribution de vérité γ qui la satisfait. On note alors $\gamma \models \varphi$.

- Une formule qui n'est pas satisfiable est dite **inconsistante**. On dit également que la formule est une **contradiction**.
- Une formule satisfaite pour toutes les distributions de vérités possibles est appelée une **tautologie**. On note alors $\models \varphi$.

Équivalence

Définition 13 (Équivalence)

Deux formules φ et ψ sont dites **équivalentes** lorsque pour toute distribution de vérité γ, on a $\gamma^*(\varphi) = \gamma^*(\psi)$. On note :

$$\varphi \equiv \psi$$

Deux formules sont équivalentes si elles ont la même table de vérité.

Remarque

L'équivalence $\varphi \equiv \psi$ n'est pas une formule. C'est seulement un jugement porté sur les formules φ et ψ qui exprime que d'un point de vue sémantique, les deux formules sont indiscernables.

Problème SAT

Le **problème SAT** consiste à déterminer si une formule est satisfiable.

- $SAT(\varphi)$ = `true` si F est satisfiable.
- $SAT(\varphi)$ = `false` si F est inconsistante.

La résolution naïve de ce problème énumère toutes les distributions de vérité possibles d'une formule (ou d'un ensemble de formules) pour vérifier si l'une de ces distributions de vérité satisfait la formule. Si φ est une formule qui comporte n variables propositionnelles, il existe 2^n interprétations possibles pour φ. À ce jour, il n'existe pas d'algorithme efficaces pour résoudre un tel problème. Le problème SAT est dit NP-complet.

Littéral et clause

Définition 14

Littéral Un **littéral** est un variable propositionnelle ou la négation d'une variable propositionnelle.

Par exemple : p, $\neg p$.

Définition 15

Clauses Une **clause disjonctive** (resp. **conjonctive)** est une disjonction (resp. conjonction) de littéraux.

Par exemple : $p \vee q \vee \neg r$, $p \wedge q \wedge \neg r$.

Formes normales

Définition 16 (Formes normales)

Une **forme normale disjonctive** est une disjonction de clauses conjonctives $C_1 \vee C_2 \vee \cdots \vee C_N$ avec $C_i = l_{i,1} \wedge \cdots \wedge l_{i,n_i}$ et $l_{i,j}$ un littéral.
Une **forme normale conjonctive** est une conjonction de clauses disjonctives $D_1 \wedge D_2 \wedge \cdots \wedge D_N$ avec $D_i = l_{i,1} \vee \cdots \vee l_{i,n_i}$ et $l_{i,j}$ un littéral.

Par exemple :

$$\text{FND} : \underbrace{\neg x}_{\text{clause}} \vee \underbrace{(y \wedge \neg z)}_{\text{clause}} \vee \underbrace{(\neg y \wedge z)}_{\text{clause}}$$

$$\text{FNC} : \underbrace{\neg x}_{\text{clause}} \wedge \underbrace{(y \vee \neg z)}_{\text{clause}} \wedge \underbrace{(\neg y \vee z)}_{\text{clause}}$$

Construction à partir d'une formule logique

- Éliminer les $\rightarrow$ en remplaçant $\phi \rightarrow \psi$ par $\neg\phi \vee \psi$.
- Propager les négations à l'aide des relations de De Morgan.

$$\neg(\phi \wedge \psi) \equiv \neg\phi \vee \neg\psi \qquad \neg(\phi \vee \psi) \equiv \neg\phi \wedge \neg\psi$$

- Simplifier les double-négations : $\neg\neg\phi \equiv \phi$.
- Distribuer.

$$\phi \wedge (\psi \vee \omega) \equiv (\phi \wedge \psi) \vee (\phi \wedge \omega)$$

$$\phi \vee (\psi \wedge \omega) \equiv (\phi \vee \psi) \wedge (\phi \vee \omega)$$

Construction à partir d'une table de vérité

- Construction d'une FND
  - Identifier les 1 dans la colonne de la formule ϕ.
  - Construire une disjonction de conjonctions de ϕ à partir des propositions de ces lignes.
- Construction d'une FNC
  - Identifier les 1 dans la colonne de la formule ϕ.
  - Construire la table de vérité de $\neg\phi$.
  - Construire une disjonction de conjonctions de $\neg\phi$ à partir des propositions de ces lignes.
  - En déduire une FNC par négation de la FND précédente.

Pourquoi les FND ou les FNC ?

Théorème 17

Toute proposition logique comportant un nombre fini de variables propositionnelles est équivalente à une proposition sous FND ou sous FNC. Cette décomposition est unique à l'ordre près dès lors que les littéraux prennent une forme minimale.

Ce résultat justifie l'intérêt porté aux FND puisque toute FND est satisfiable si et seulement si chacune de ses clauses a au moins un littéral vrai. Ainsi, l'ensemble des clauses constitue un ensemble de contraintes à satisfaire.

Pourquoi les FND ou les FNC ?

En outre, SAT est utile pour :

- encoder un problème en logique propositionnelle ;
- retranscrire le modèle pour trouver une solution au problème ;
- utiliser un solveur SAT !

Parmi les applications de SAT, on peut citer la vérification de modèles, la modélisation de circuits, le coloriage de cartes (graphes) ou encore les démonstrations automatiques.