X - Logique propositionnelle

28 mai 2024

1 Syntaxe du calcul propositionnel

On dispose d'un alphabet constitué :

- d'un ensemble $\mathcal{V}$ appelé ensemble des variables propositionnelles ;
- de deux constantes $\top$ et $\bot$;
- d'un connecteur unaire $\neg$; on dira aussi que $\neg$ est un connecteur logique d'arité 1 ;
- des connecteurs binaires (i.e d'arité 2) $\wedge$, $\vee$, $\rightarrow$ et $\leftrightarrow$.

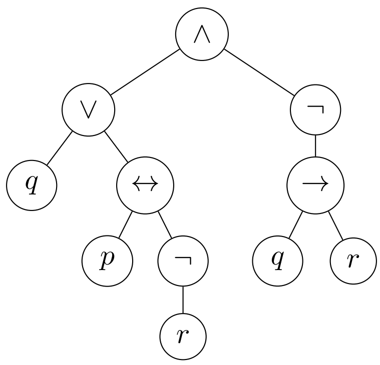
L'ensemble $\mathcal{F}$ des formules du calcul propositionnel, ou formules logiques, est le plus petit ensemble tel que :

- $\top$ et $\bot$ sont des formules logiques
- Les variables propositionnelles (i.e les éléments de $\mathcal{V}$) sont des formules logiques.
- Si φ est une proposition alors $(\neg\varphi)$ est une proposition.
- Si φ et ψ sont deux formules, alors
  - $(\varphi \wedge \psi)$ est une proposition logique.
  - $(\varphi \vee \psi)$ est une proposition logique.
  - $(\varphi \rightarrow \psi)$ est une proposition logique.
  - $(\varphi \leftrightarrow \psi)$ est une proposition logique.

Par exemple, si p, q, r sont des variables propositionnelles alors : p ; $(p \vee (q \rightarrow r))$; $\neg((q \vee (p \leftrightarrow (\neg r))) \wedge p)$ sont des formules logiques.

En revanche $p\neg q \wedge \vee r$ n'est pas une formule logique.

Une formule logique peut se représenter sous forme d'un arbre. Par exemple, $(q \vee (p \leftrightarrow (\neg r))) \wedge (\neg(q \rightarrow r))$ est représenté par :



La *hauteur* d'une formule est la hauteur de l'arbre correspondant. La *taille* de la formule est le nombre de symbole qu'elle contient.

Enfin, une *sous-formule* d'une formule φ représentée par un arbre A est une formule dont l'arbre est un sous-arbre de A.

Par exemple, $p \leftrightarrow (\neg r)$ est une sous-formule de $(q \vee (p \leftrightarrow (\neg r))) \wedge (\neg(q \rightarrow r))$.

L'ensemble $\mathcal{V}$ étant fixé, pour démontrer une propriété P sur les formules, on pourra raisonner

- par récurrence sur la hauteur ou sur la taille des formules
- par **induction structurelle**, en montrant que
  - la propriété P est vérifiée par $\top$, $\perp$ et par les éléments de $\mathcal{V}$
  - Si P est vérifiée par φ alors P est vérifiée par $\neg\varphi$
  - Si P est vérifiée par φ_1 et φ_2, alors P est vérifiée par $\varphi_1 \vee \varphi_2$, par $\varphi_1 \wedge \varphi_2$, par $\varphi_1 \rightarrow \varphi_2$ et par $\varphi_1 \leftrightarrow \varphi_2$.

2 Sémantique du calcul propositionnel

2.1 Évaluation d'une proposition logique

On considère l'ensemble $\{V, F\}$ des valeurs de vérité.

Une *valuation* est une application de l'ensemble $\mathcal{V}$ des variables propositionnelles dans $\{V, F\}$: à chaque variable propositionnelle, on associe une valeur de vérité.

Si σ est une valuation, on définit inductivement l'évaluation $v_\sigma(\varphi)$ d'une proposition φ pour la valuation σ par :

- $v_\sigma(\top) = V$ et $v_\sigma(\perp) = F$.
- Si $\varphi = v \in \mathcal{V}$, alors $v_\sigma(\varphi) = \sigma(v)$
- Si $\varphi = \neg\psi$, alors $v_\sigma(\varphi) = \begin{cases} F, & \text{si } v_\sigma(\psi) = V \\ V, & \text{sinon} \end{cases}$
- Si $\varphi = \varphi_1 \wedge \varphi_2$, alors $v_\sigma(\varphi) = \begin{cases} V, & \text{si } v_\sigma(\varphi_1) = V \text{ et } v_\sigma(\varphi_2) = V \\ F, & \text{sinon} \end{cases}$
- Si $\varphi = \varphi_1 \vee \varphi_2$, alors $v_\sigma(\varphi) = \begin{cases} V, & \text{si } v_\sigma(\varphi_1) = V \text{ ou } v_\sigma(\varphi_2) = V \\ F, & \text{sinon} \end{cases}$
- Si $\varphi = \varphi_1 \rightarrow \varphi_2$, alors $v_\sigma(\varphi) = \begin{cases} F, & \text{si } v_\sigma(\varphi_1) = V \text{ et } v_\sigma(\varphi_2) = F \\ V, & \text{sinon} \end{cases}$
- Si $\varphi = \varphi_1 \leftrightarrow \varphi_2$, alors $v_\sigma(\varphi) = \begin{cases} V, & \text{si } v_\sigma(\varphi_1) = v_\sigma(\varphi_2) \\ F, & \text{sinon} \end{cases}$

2.2 Table de vérité

Si une proposition φ s'exprime à l'aide de n variables propositionnelles $(v_1, \ldots, v_n)$, il y a 2^n valeurs possibles pour le n-uplet $(\sigma(v_1), \ldots, \sigma(v_n))$.

On peut alors résumer l'évaluation de φ en fonction des valeurs de ses variables dans une table de vérité à 2^n lignes.

Par exemple, soit $\varphi = p \vee (\neg p \wedge q)$. Sa table de vérité est :

p	q	$\neg p$	$\neg p \wedge q$	φ
V	V	F	F	V
V	F	F	F	V

p	q	$\neg p$	$\neg p \wedge q$	φ
F	V	V	V	V
F	F	V	F	F

2.3 Tautologies et satisfiabilité

- On dit qu'une formule φ est *satisfiable* lorsqu'il existe une valuation σ telle que $v_\sigma(\varphi) = V$. On dit alors que σ est un *modèle* pour φ.
- On dit qu'une formule φ est une *tautologie* lorsque pour toute valuation σ, $v_\sigma(\varphi) = V$, autrement dit lorsque toutes les valuations sont des modèles pour φ.
- On dit qu'une formule φ est une *antologie* lorsque pour toute valuation σ, $v_\sigma(\varphi) = F$, autrement dit lorsque φ n'est pas satisfiable.

Par exemple, la formule $(p \Rightarrow q) \leftrightarrow (\neg p \vee q)$ est une tautologie ; en effet, la table de vérité associée à cette formule est :

p	q	$p \Rightarrow q$	$\neg p$	$\neg p \vee q$	$(p \to q) \leftrightarrow (\neg p \vee q)$
V	V	V	F	V	V
V	F	F	F	F	V
F	V	V	V	V	V
F	F	V	V	V	V

Remarques : - Une formule φ est satistiable si et seulement si l'ensemble de ses modèles est non vide. - Une formule φ est non satisfiable si et seulement si $\neg\varphi$ est une tautologie.

Par conséquent, savoir tester si une formule est satisfiable et savoir tester si une formule est une tautologie sont des problèmes étroitement liés.

- On peut résoudre le problème de la satisfiabilité d'une formule φ en construisant sa table de vérité et en examinant toutes les lignes. Si φ comporte n variables propositionnelles, le temps de calcul pour cette méthode est clairement proportionnelle à 2^n (on dira qu'on peut résoudre le problème en temps $O(2^n)$.
- En revanche, étant donnée une valuation σ, il est facile de vérifier si $v_\sigma(\varphi) = V$ (en temps polynomial). On ne connaît pas d'algorithme réellement plus performant ; ce problème fait partie d'une classe de problèmes d'optimisation difficiles.

2.4 Formules logiquement équivalentes

- On dit que deux formules logiques φ et ψ sont *équivalentes*, et on note $\varphi \equiv \psi$, lorsque, pour toute valuation σ, $v_\sigma(\varphi) = v_\sigma(\psi)$.
- On dit que ψ est une *conséquence logique* de φ, et on note $\varphi \models \psi$, lorsque tout modèle de φ est un modèle de ψ, autrement dit lorsque pour toute valuation σ, si $v_\sigma(\varphi) = V$ alors $v_\sigma(\psi) = V$.

Remarque : La relation $\equiv$ est une relation d'équivalence.

Pour vérifier que deux formules sont logiquement équivalentes, on examine leurs tables de vérité. Par exemple, $p \vee (\neg p \wedge q) \equiv p \vee q$.

2.5 Quelques équivalences fondamentales

Soient φ_1, φ_2 et φ_3 trois formules. Alors :

— $\varphi_1 \wedge \varphi_2 \equiv \varphi_2 \wedge \varphi_1$ et $\varphi_1 \vee \varphi_2 \equiv \varphi_2 \vee \varphi_1$ *(commutativité de $\wedge$ et de $\vee$).*
— $(\varphi_1 \wedge \varphi_2) \wedge \varphi_3 \equiv \varphi_1 \wedge (\varphi_2 \wedge \varphi_3)$ et $(\varphi_1 \vee \varphi_2) \vee \varphi_3 \equiv \varphi_1 \vee (\varphi_2 \vee \varphi_3)$ *(associativité de $\wedge$ et de $\vee$).*
— $(\varphi_1 \wedge \varphi_2) \vee \varphi_3 \equiv (\varphi_1 \vee \varphi_3) \wedge (\varphi_2 \vee \varphi_3)$ et $(\varphi_1 \vee \varphi_2) \wedge \varphi_3 \equiv (\varphi_1 \wedge \varphi_3) \vee (\varphi_2 \wedge \varphi_3)$ *(distributivité de $\vee$ par rapport à $\wedge$ et de $\wedge$ par rapport à $\vee$).*

Remarque : l'associativité permettra de faire l'économie de quelques parenthèses : on notera $\varphi_1 \vee \varphi_2 \vee \cdots \vee \varphi_3$ au lieu $((\ldots(\varphi_1 \vee \varphi_2) \vee \ldots) \vee \varphi_3)$; de même pour $\wedge$.

Soient φ_1, φ_2 deux formules. Alors :

— $\varphi_1 \vee (\varphi_1 \wedge \varphi_2) \equiv \varphi_1$
— $\varphi_1 \wedge (\varphi_1 \vee \varphi_2) \equiv \varphi_1$
— **Lois de Morgan** : $\neg(\varphi_1 \vee \varphi_2) \equiv (\neg\varphi_1) \wedge (\neg\varphi_2)$ et $\neg(\varphi_1 \wedge \varphi_2) \equiv (\neg\varphi_1) \vee (\neg\varphi_2)$
— **Tiers-exclu** : $\varphi_1 \vee \neg\varphi_1 \equiv \top$, autrement dit $\varphi_1 \vee \neg\varphi_1$ est une tautologie
— **Décomposition de l'implication** : $\varphi_1 \to \varphi_2 \equiv (\neg\varphi_1) \vee \varphi_2$
— **Décomposition de l'équivalence** : $\varphi_1 \leftrightarrow \varphi_2 \equiv (\varphi_1 \to \varphi_2) \wedge (\varphi_2 \to \varphi_1)$
— **Contraposition** : $(\varphi_1 \to \varphi_2) \equiv (\neg\varphi_2 \to \neg\varphi_1)$

2.6 Ensemble de formules

On peut généraliser les définitions à un ensemble Γ de formules :

— Un *modèle* pour un ensemble de formules Γ est une valuation σ telle que

$$\forall\varphi \in \Gamma,\ v_\sigma(\varphi) = V$$

— Une formule ψ est une *conséquence logique* de Γ lorsque tout modèle de Γ est un modèle de ψ. On note alors $\Gamma \models \psi$.

Par exemple, $\{p, p \to q\} \models q$.

3 Formes normales

Définitions

— On appelle *littéral* toute proposition logique de la forme v ou $\neg v$, où v est une variable propositionnelle.
— Une *conjonction élémentaire* est une forme de la forme $l_1 \wedge l_2 \wedge \cdots \wedge l_{n-1} \wedge l_n$ où chaque l_i est un littéral (ou de la forme $\top$).
— Une *clause*, ou *disjonction élémentaire*, est une formule de la forme $l_1 \vee l_2 \vee \cdots \vee l_{m-1} \vee l_m$ où chaque l_i est un littéral (ou de la forme $\bot$).

Définitions

— Une proposition φ est dite sous forme normale disjonctive si elle est de la forme $C_1 \vee C_2 \vee \cdots \vee C_k$ où chaque C_i est une conjonction élémentaire.

$$\varphi = \bigvee_i \left(\bigwedge_j l_{i,j} \right)$$

— Une proposition φ est dite sous forme normale conjonctive si elle est de la forme $D_1 \wedge D_2 \wedge \cdots \wedge D_q$ où chaque D_i est une clause.

$$\varphi = \bigwedge_i \left(\bigvee_j l_{i,j} \right)$$

Proposition :

— Si φ est une conjonction alors $\neg\varphi$ est équivalente à une disjonction.

— Si φ est une disjonction alors $\neg\varphi$ est équivalente à une conjonction.

Théorème : Soit φ une formule logique quelconque. Alors - il existe une formule logique φ' sous forme normale disjonctive logiquement équivalente à φ. - il existe une formule logique φ'' sous forme normale conjonctive logiquement équivalente à φ.

3.1 Table de vérités et forme normale disjonctive

Lemme : Soient $v_1, \ldots, v_n$ des variables propositionnelles et σ une valuation sur $\{v_1, \ldots, v_n\}$. On pose $l_i = v_i$ si $\sigma(v_i) = 1$ et $l_i = \neg v_i$ si $\sigma(v_i) = 0$. Soit $C_\sigma = l_1 \wedge \cdots \wedge l_n$.
Alors pour toute valuation τ, $V_\tau(C_\sigma) = \begin{cases} 1 & \text{si } \tau = \sigma \\ 0 & \text{sinon} \end{cases}$

Théorème : Soient φ une formule logique, $v_1, \ldots, v_n$ les variables propositionnelles intervenant dans φ. Pour toute distribution de vérité σ, on définit la conjonction C_σ comme dans le lemme.

Soit Σ l'ensemble des affectations σ telles que $V_\sigma(P) = 1$. Alors $\varphi \equiv \bigvee_{\sigma \in \Sigma} C_\sigma$.

Soit $\varphi = (p \vee (\neg q)) \wedge (q \leftrightarrow (\neg p \vee r))$. Sa table de vérité est :

p	q	r	$p \vee (\neg q)$	$\neg p \vee r$	$q \leftrightarrow (\neg p \vee r)$	φ
V	V	V	V	V	V	V
V	V	F	V	F	F	F
V	F	V	V	V	F	F
V	F	F	V	F	V	V
F	V	V	F	V	V	F
F	V	F	F	V	V	F
F	F	V	V	V	F	F
F	F	F	V	V	F	F

Donc $\varphi \equiv (p \wedge q \wedge r) \vee (p \wedge \neg q \wedge \neg r)$

3.2 Problème SAT

Le problème SAT est le problème consistant à déterminer si une formule propositionnelle est satisfiable. Pour une formule propositionnelle comportant n variables, on peut toujours répondre à ce problème en examinant les 2^n valuations possibles, ce qui donne un algorithme exponentiel.

Par ailleurs, pour un entier non nul k fixé, le problème k-SAT consiste à déterminer si une formule φ sous forme normale conjonctive dont chaque clause comporte k littéraux est satisfiable.

Le problème 1-SAT peut se résoudre en temps linéaire : une instance φ de 1-SAT est satisfiable si et seulement si φ ne contient pas à la fois une variable et sa négation.

Le problème 2-SAT peut se résoudre grâce à un graphe. On construit un graphe orienté dont les sommets sont étiquetés par les variables propositions et leur négation. Comme toute clause $\ell_1 \vee \ell_2$ est équivalente à $(\neg \ell_1 \rightarrow \ell_2) \wedge (\neg \ell_2 \rightarrow \ell_1)$, on construit une arête de $\neg\ell$_1 vers ℓ_2 et de $\neg \ell_2$ vers ℓ_1.

On peut alors montrer qu"une instance de 2-SAT est satisfiable si et seulement si aucune composante fortement connexe du graphe ne contient à la fois une variable propositionnelle et sa négation, et qu'il est possible de le vérifier en temps polynomial.

Tout instance de k-SAT pour $k \geqslant 3$ peut se réduire à une instance de 3-SAT en ajoutant des variables propositionnelles.

On ne connaît aucun algorithme polynomial pour résoudre 3-SAT.

Exercice

Vous avez été sélectionné(e) pour participer au jeu “Cherchez les Clés du Paradis (CCP)". Le jeu se déroule en trois épreuves, au cours desquelles vous devez collecter des clés vertes. A l'issue de chacune d'entre elles, vous passez à l'épreuve suivante en cas de succès et êtes éliminé(e) en cas d'échec.

Première épreuve

Jean-Pierre Pendule, le célèbre animateur, vous accueille pour la première épreuve. Il vous explique la règle du jeu. Devant vous, deux boîtes et sur chacune d'entre elles une inscription. Chacune des boîtes contient soit une clé verte, soit une clé rouge. Vous devez choisir l'une des boîtes : si le résultat est une clé rouge, alors vous quittez le jeu, si c'est une clé verte vous êtes qualifié(e) pour l'épreuve suivante.

Jean-Pierre Pendule dévoile les inscriptions sur chacune des boîtes et vous affirme qu'elles sont soit vraies toutes les deux, soit fausses toutes les deux :

— sur la boîte 1, il est écrit “Une au moins des deux boîtes contient une clé verte" ;

— sur la boîte 2, il est écrit “Il y a une clé rouge dans l'autre boîte".

Dans toute cette partie, on note P_i la proposition affirmant qu'il y a une clé verte dans la boîte i.

Q.1 Donner une formule de la logique des propositions représentant la phrase écrite sur la boîte 1.

Q.2 Donner ude même ne formule de la logique des propositions pour l'inscription de la boîte 2.

Q.3 Donner une formule représentant l'affirmation de l'animateur. Simplifier cette formule de sorte à n'obtenir qu'une seule occurrence de chaque P_i.

Q.4 Quel choix devez-vous faire pour continuer le jeu à coup sûr ?

Deuxième épreuve

Bravo, vous avez obtenu la première clé verte. Jean-PierrePendule vous félicité et vous annonce que cette première épreuve n'était qu'une mise en jambe. Avec les mêmes règles du jeu, l'animateur vous propose alors deux nouvelles boîtes portant les inscriptions suivantes :

— sur la boîte 1, il est écrit “Il y a une clé rouge dans cette boîte, ou bien il y a une clé verte dans la boîte 2" ;

— sur la boîte 2, il est écrit “Il y a une clé verte dans la boîte 1".

Q.5 Donner une formule de la logique des propositions pour chaque affirmation.

Q.6 Sachant qu'encore une fois les deux affirmations sont soit vraies toutes les deux, soit fausses toutes les deux, donner le contenu de chaqu boîte. En déduire votre choix pour remporter la deuxième clé verte.

Troisième épreuve

Le suspens est à son comble, vous voici arrivé(e) à la dernière épreuve. A votre grande surprise, Jean-Pierre Pendule vous dévoile une troisième boîte et vous explique les règles du jeu. Dans une des boîtes se cache la clé qui vous permet de remporter la victoire finale. Dans une autre boîte se cache une clé rouge qui vous fait tout perdre. La dernière boîte est vide. Encore une fois, chacune des boîtes porte une inscription :

— sur la boîte 1, il est écrit “La boîte 3 est vide" ;

— sur la boîte 2, il est écrit “La clé rouge est dans la boîte 1" ;

— sur la boîte 3, il est écrit “Cette boîte est vide".

L'animateur affirme que l'inscription portée sur la boîte contenant la clé verte est vraie, celle portée par la boîte contenant la clé rouge est fausse. L'inscription affichée sur la boîte vide est aussi vraie.

Q.7 Donner une formule de la logique des propositions pour chaque inscription.

Q.8 Donner une formule de la logique des propositions synthétisant l'information que vous apportée l'animateur.

Q.9 En supposant que la clé verte est dans la boîte 2, montrer par l'absurde que l'on aboutit à une incohérence.

Q.10 Donner alors la composition des trois boîtes.