

* Motivations : Problèmes d'emploi du temps, coloriage d'un graphe

I) Syntaxe.

1) le langage de la logique propositionnelle (LP).

def 1 : L'ensemble des variables propositionnelles est un ensemble de symboles $VAR = \{p_1, q_1, r_1, \dots, p_2, p_2, p_3, \dots\}$

def 2 : Le langage \mathcal{F} de la LP est le langage sur l'alphabet $\Sigma = VAR \cup \{ \perp, \neg, \wedge, (,) \}$ défini par induction par :

- $VAR \cup \{ \perp \} \subseteq \mathcal{F}$
- $\forall \varphi \in \mathcal{F}, \neg \varphi \in \mathcal{F}$
- $\forall \varphi, \psi \in \mathcal{F}, (\varphi \wedge \psi) \in \mathcal{F}$

ex 3 : $(\neg p \wedge (q \wedge (\neg \perp \wedge p)))$ est une formule de la LP.

thm 4 (lecture unique) :

Soit $\varphi \in \mathcal{F}$. Alors un et un seul des trois cas suivants se présente :

- (i) $\varphi \in VAR$
- (ii) Il existe $\psi \in \mathcal{F}$ tq $\varphi = \neg \psi$
- (iii) Il existe $(\varphi_1, \varphi_2) \in \mathcal{F}^2$ tq $\varphi = (\varphi_1 \wedge \varphi_2)$

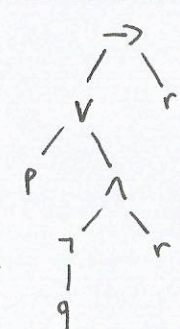
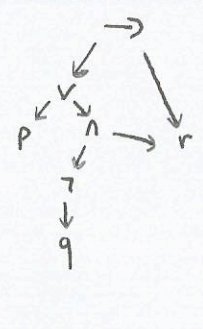
Rq 5 : on s'autorisera, pour tout $\varphi, \psi \in \mathcal{F}$, l'écriture de :

- $(\varphi \vee \psi)$ à la place de $\neg(\neg\varphi \wedge \neg\psi)$
- $(\varphi \rightarrow \psi)$ à la place de $(\neg\varphi \vee \psi)$
- $(\varphi \leftrightarrow \psi)$ à la place de $((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$
- \top à la place de $\neg \perp$

ex 6 : on considérera donc que ceci est une formule de la LP :

$$\varphi_1 = ((p \vee (\neg q \wedge r)) \rightarrow r)$$

2) Représentations

formule (mot)	arbre	DAG
φ_1		

Rq 7 : le DAG économise de l'espace en mémoire.

3) Résolution par réfutation.

def 8 : un littéral est une formule de la forme p ou $\neg p$, avec $p \in VAR$. une clause est une disjonction (\vee) de littéraux.

def 9 : une formule est sous forme normale conjonctive (CNF) si elle s'écrit sous la forme :

$$\varphi = \bigwedge_{i \in I} \bigvee_{j \in J} l_{ij}$$

pour I, J finis
et l_{ij} littéral

def 10 : règles de résolution :

$$\text{coupure} \frac{(\neg p \vee l_2 \vee \dots \vee l_n) \quad (p \vee l'_1 \vee \dots \vee l'_m)}{(l_1 \vee \dots \vee l_n \vee l'_1 \vee \dots \vee l'_m)}$$

$$\text{Redondance} \frac{l \vee l \vee l_1 \vee \dots \vee l_n}{l \vee l_1 \vee \dots \vee l_n}$$

→ Une dérivation d'une formule φ en CNF vers une formule ψ en CNF est une suite de règles de résolutions à partir de l'ensemble des clauses de φ aboutissant à l'ensemble des clauses de ψ .

def 11: Une preuve de réfutation de $\varphi \in \mathcal{F}$ en FNC est une dérivation de φ à \perp .

ex 12: $\varphi_2 = (((p \vee \neg q) \wedge (r \vee \neg p)) \wedge ((q \wedge r) \wedge \neg r))$ [cf Annexe]

II) Sémantique

1) Valuation et table de vérité.

def 13: une valuation est une fonction $v: VAR \rightarrow \{0, 1\}$

prop-def 14: Soit $v \in \{0, 1\}^{VAR}$ une valuation. On étend v de manière unique en une fonction $\bar{v}: \mathcal{F} \rightarrow \{0, 1\}$ par:

- $\bar{v}(p) = p$
- $\bar{v}(\neg \phi) = 1 - \bar{v}(\phi)$
- $\bar{v}(\perp) = 0$
- $\bar{v}(\phi \wedge \psi) = \bar{v}(\phi) \times \bar{v}(\psi)$

Rq 15: On obtient également:

- $\bar{v}(\top) = 1$
- $\bar{v}(\phi \rightarrow \psi) = 1 - \bar{v}(\phi) + \bar{v}(\phi) \bar{v}(\psi)$
- $\bar{v}(\phi \vee \psi) = \bar{v}(\phi) + \bar{v}(\psi) - \bar{v}(\phi) \bar{v}(\psi)$
- $\bar{v}(\phi \leftrightarrow \psi) = 1 - \bar{v}(\phi) - \bar{v}(\psi) + 2 \bar{v}(\phi) \bar{v}(\psi)$

→ Désormais, on note v et on appelle valuation cette extension.

ex 16: si $v(p) = v(q) = v(r) = 0$, $v(\varphi_1) = 1$ et $v(\varphi_2) = 0$.

def 17: une table de vérité est un tableau représentant, en données, l'image de toutes les valuations possibles de formules données.

ex 18: tables de vérité de φ_1 et φ_2 [cf Annexe].

Pb 19: EVAL (φ, v)

{Entrée: une formule $\varphi \in \mathcal{F}$ et une valuation $v \in \{0, 1\}^{VAR}$.

{Sortie: oui si $v(\varphi) = 1$; non sinon.

prop 20: le problème EVAL est décidable en temps linéaire.

2) Validité et satisfiabilité.

def 21: une formule φ est valide si pour toute valuation

$v \in \{0, 1\}^{VAR}$, on a: $v(\varphi) = 1$.

Un ensemble de formules $\{\varphi_i, i \in I\}$ est valide si chaque

φ_i est valide

def 22: une formule φ est satisfiable s'il existe une valuation $v \in \{0, 1\}^{VAR}$ telle que $v(\varphi) = 1$.

Un ensemble de formules $\{\varphi_i, i \in I\}$ est satisfiable s'il existe une valuation v telle que $(\forall i \in I, v(\varphi_i) = 1)$.

On dit insatisfiable dans le cas contraire.

ex 23: φ_1 est satisfiable mais non valide; φ_2 est insatisfiable.

prop 24: $\varphi \in \mathcal{F}$ est valide ssi $\neg \varphi$ est insatisfiable.

def 25: deux formules sont équisatisfiables si elles sont toutes les deux satisfiables ou toutes les deux insatisfiables.

thm 26: (Compacité de la logique propositionnelle)

Un ensemble de formules $E \subseteq \mathcal{F}$ est satisfiable ssi toute partie finie de E est satisfiable.

DEV 1

Application 27: un graphe G non-orienté est k -colorable ssi tout sous-graphe fini $G' \subseteq G$ est k -colorable.

Application 28: Soit (H, V) des règles de pavage sur \mathbb{Z}^2 . Alors

\mathbb{Z}^2 est (H, V) -pavable ssi pour tout $n \geq 0$, $[-n, n]^2$ est (H, V) -pavable.

3) Équivalence sémantique.

def 29: deux formules $\varphi, \psi \in \mathcal{F}$ sont sémantiquement équivalentes si la formule $(\varphi \leftrightarrow \psi)$ est valide. On notera alors $\varphi \sim \psi$.

ex 30: (Lois de DE MORGAN)

$$\neg(p \vee q) \sim (\neg p \wedge \neg q)$$

$$\neg(p \wedge q) \sim (\neg p \vee \neg q)$$

prop 31: deux formules équivalentes sont en particulier équisatisfiables.

4) Lien entre syntaxe et sémantique.

thm 32: (complétude du calcul propositionnel). Soit $\varphi \in \mathcal{F}$ sous CNF.

Il existe une preuve de la réfutation de φ .

← complétude

→ correction

φ est insatisfiable.

III) Satisfiabilité et formes normales.

Pb 33 : SAT (φ)

{ Entrée : une formule $\varphi \in \mathcal{F}$.

{ Sortie : oui si φ est satisfiable (et une valuation v tq $v(\varphi)=1$); non sinon.

thm 34 : (Cook, 1971) le problème SAT est NP-complet. DEV 2

1) Mise sous forme normale conjonctive.

Pb 35 : CNF-SAT (φ)

{ Entrée : une formule $\varphi \in \mathcal{F}$ sous CNF

{ Sortie : oui si φ est satisfiable (et $v \in \{0,1\}^{VAL}$ tq $v(\varphi)=1$); non sinon.

thm 36 : Pour toute formule $\varphi \in \mathcal{F}$, il existe une formule $\tilde{\varphi} \in \mathcal{F}$ en CNF sémantiquement équivalente à φ ($\tilde{\varphi} \sim \varphi$). La transformation est exponentielle.

$$\begin{aligned} \text{ex 37} : (\neg p \rightarrow \neg q) &\rightarrow (p \rightarrow q) \sim \neg(\neg\neg p \vee \neg q) \vee (\neg p \vee q) \\ &\sim (\neg\neg p \wedge \neg\neg q) \vee (\neg p \vee q) \\ &\sim (\neg p \wedge \neg q) \vee (\neg p \vee q) \\ &\sim (\neg p \vee \neg p \vee q) \wedge (q \vee \neg p \vee q) \end{aligned}$$

Cor 38 : le thm 32 est vrai pour toute formule $\varphi \in \mathcal{F}$.

thm 39 (Algorithme de Tseitin) : pour toute formule $\varphi \in \mathcal{F}$, il existe une formule $\tilde{\varphi} \in \mathcal{F}$ en CNF tq φ et $\tilde{\varphi}$ sont équivalentes. De plus, la transformation $tr : \varphi \mapsto \tilde{\varphi}$ est calculable en temps linéaire.

DEV 3

Corollaire 40 : le problème CNF-SAT est NP-complet.

Pb 41 : 3CNF-SAT (φ)

{ Entrée : $\varphi \in \mathcal{F}$ en CNF avec au plus 3 littéraux par clause.

{ Sortie : oui si φ est satisfiable (\forall tq $v(\varphi)=1$); non sinon.

thm 42 : Il existe une réduction polynomiale de CNF-SAT à 3CNF-SAT.

Corollaire 43 : le problème 3CNF-SAT est NP-complet.

Pb 44 : 2CNF-SAT

{ Entrée : $\varphi \in \mathcal{F}$ sous CNF avec au plus deux littéraux par clause.

{ Sortie : oui si φ est satisfiable (\forall tq $v(\varphi)=1$); non sinon.

thm 45 : le problème 2CNF-SAT est décidable en temps polynomial.

Corollaire 46 : Il n'existe pas de réduction polynomiale du problème 3CNF-SAT au problème 2CNF-SAT.

2) D'autres formes normales à la rescousse!

def 47 : une formule est sous forme normale disjonctive (DNF) si elle s'écrit sous la forme :

$$\varphi = \bigvee_{i \in I} \bigwedge_{j \in J} p_{i,j}$$

pour I, J finis
et $p_{i,j}$ littéraux

Pb 48 : DNF-SAT (φ)

{ Entrée : $\varphi \in \mathcal{F}$ sous DNF

{ Sortie : oui si φ est satisfiable (\forall tq $v(\varphi)=1$); non sinon.

thm 49 : le problème DNF-SAT est calculable en temps linéaire.

prop 50 : Toute formule $\varphi \in \mathcal{F}$ est sémantiquement équivalente à une formule $\tilde{\varphi} \in \mathcal{F}$ sous DNF (il suffit de lire sa table de vérité).

Rq 51 : D'après le théorème de Cook, cette transformation ne peut pas être polynomiale. De plus, le thm 49 assure qu'il n'existe pas d'algorithme calculant en temps polynomial une transformation $tr : \varphi \mapsto \tilde{\varphi}$ tq φ et $\tilde{\varphi}$ soient équivalentes et $\tilde{\varphi}$ soit sous DNF.

def 52 : une clause de Horn est une clause $l_1 \vee \dots \vee l_n$ où apparaît au plus un littéral positif $l_i = p \in \text{VAR}$ ($\forall j \neq i, l_j = \neg q, q \in \text{VAR}$)

Pb 53 : HORN-SAT (φ)

{ Entrée : $\varphi \in \mathcal{F}$ sous CNF avec uniquement des clauses de Horn.

{ Sortie : oui si φ est satisfiable (\forall tq $v(\varphi)=1$); non sinon.

thm 54 : le problème HORN-SAT est décidable en temps linéaire.

ex 55 : la formule suivante est satisfiable pour la valuation

$$\varphi_3 = \underbrace{(p \vee \neg q \vee \neg r)}_{\text{clause de Horn stricte / positive / négative}} \wedge \underbrace{(q)}_{\text{clause de Horn stricte / positive / négative}} \wedge \underbrace{(\neg p \vee \neg q)}_{\text{clause de Horn stricte / positive / négative}}$$

$$\begin{cases} p \rightarrow 0 \\ q \rightarrow 1 \\ r \rightarrow 0 \end{cases}$$

ANNEXE

ex 12: $\varphi_2 = (((p \vee \neg q) \wedge (r \vee \neg p)) \wedge ((q \vee r) \wedge \neg r))$

$$\begin{array}{r}
 \text{(F)} \frac{p \vee \neg q \quad r \vee \neg p}{\text{(F)} \frac{\neg q \vee r \quad q \vee r}{\text{(R)} \frac{r \vee r}{r} \quad \neg r}} \\
 \perp
 \end{array}$$

ex 18:

p	q	r	$\varphi_1 = ((p \vee (\neg q \wedge r)) \rightarrow r)$	$\varphi_2 = (((p \vee \neg q) \wedge (r \vee \neg p)) \wedge ((q \vee r) \wedge \neg r))$
0	0	0	1	0
0	0	1	1	0
0	1	0	0	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	0
1	1	0	0	0
1	1	1	1	0