916

Formules du calcul propositionnel : représentation, formes normales, satisfiabilité. Applications.

Motivations:

→ base de théories logiques
→ model checking
→ vérification formelle
→ simplification de circuits logiques
→ encodage de problèmes : planification, coloriages, emploi du temps, n-dames

I Syntaxe

Soit V un ensemble dénombrable de variables, $C = \{\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow\}$ un ensemble fini de connecteurs.

def: L'ensemble F des formules propositionnelles est un sous-ensemble de $(C \cup V \cup \{(,)\})^*$ défini par induction :
- $V \subset F$, • $\forall f \in F$, $\neg f \in F$ • $\forall f, g \in F$, $\forall \alpha \in C \setminus \{\neg\}$, $(f \alpha g) \in F$

ex: $(x \Rightarrow (y \wedge z)) \in F$, $y \Rightarrow \vee x)(\notin F$

prop: $F = \bigcup_{n \geq 0} F_n$ où $F_0 = V$ et $F_{n+1} = F_n \cup \{\neg f, f \in F_n\} \cup \bigcup_{\alpha \in C \setminus \{\neg\}} \{(f \alpha g), f, g \in F_n\}$

def: La hauteur de $f \in F$ est $\inf\{n \geq 0, f \in F_n\}$.

thm (lecture unique): Pour toute formule $f \in F$, un et un seul des cas suivants est vérifié:
- $f \in V$, • $\exists! g \in F$, $f = \neg g$, • $\exists! g_1, g_2 \in F$, $\exists! \alpha \in C \setminus \{\neg\}$, $f = (g_1 \alpha g_2)$

def: Soit $f, g \in F$, $p \in V$. La substitution de p par g dans f, notée $f[g/p]$ est définie par induction:
- si $f = p$, $f[g/p] = g$, • si $f \in V \setminus \{p\}$, $f[g/p] = f$
- si $f = \neg f'$, $f[g/p] = \neg f'[g/p]$ • si $f = (f_1 \alpha f_2)$, $f[g/p] = (f_1[g/p] \alpha f_2[g/p])$

II Sémantique

1) Définitions

def: Une valuation est une application $\varphi : V \to \{0,1\}$.

prop: On peut étendre de manière unique toute valuation φ en $\varphi^* : F \to \{0,1\}$ par induction
- $\varphi^*(f) = \varphi(f)$ si $f \in V$, • $\varphi^*(\neg f) = [\![\neg]\!](\varphi^*(f))$
- $\varphi^*((f \alpha g)) = [\![\alpha]\!](\varphi^*(f), \varphi^*(g))$ où $[\![\alpha]\!] : \{0,1\}^2 \to \{0,1\}$ usuellement représentées par leurs tables de vérité. $[\![\neg]\!] : \{0,1\} \to \{0,1\}$

ex: $[\![\Rightarrow]\!]$ correspond à

a	b	$a \Rightarrow b$
0	0	1
0	1	1
1	0	0
1	1	1

ex: Si $\varphi(x) = \varphi(y) = 1$, $\varphi^*((x \wedge (y \Rightarrow x))) = 1$

def:
- $f \in F$ est satisfaite par φ ssi $\varphi(f) = 1$
- $f \in F$ est une tautologie ssi $\forall \varphi$, $\varphi(f) = 1$
- $f \in F$ est une contradiction ssi $\forall \varphi$, $\varphi(f) = 0$
- $f \in F$ est satisfiable ssi $\exists \varphi$, $\varphi(f) = 1$
- f et $g \in F$ sont équivalentes ssi $\forall \varphi$, $\varphi(f) = \varphi(g)$. On note $f \equiv g$

prop: Si f est une tautologie, toute substitution de f l'est.

prop: $f \in F$ est satisfiable ssi $\neg f$ n'est pas une contradiction.

def: Un ensemble de formules Σ est dit
- satisfiable ssi $\exists \varphi$ tel que $\forall f \in \Sigma$, $\varphi(f) = 1$
- finiment satisfiable ssi toute partie finie de Σ est satisfiable

def: $f \in F$ est conséquence de $\Sigma \subset F$ ssi pour tout φ satisfaisant Σ, $\varphi(f) = 1$. On note $\Sigma \models f$.

thm (Compacité) Σ satisfiable $\Leftrightarrow$ Σ finiment satisfiable (DEV)

applications:
- Un graphe est 3-coloriable ssi tout sous-graphe fini est 3-coloriable
- Tout ordre partiel sur un ensemble dénombrable peut être étendu en un ordre linéaire

def: $f, g \in F$ sont équisatisfiables ssi (f satisfiable $\Leftrightarrow$ g satisfiable)

2) Formes équivalentes

def:
- On dit que $f \in F$ est sous forme normale disjonctive (FND) ssi $f = B_1 \vee \dots \vee B_p$ où $B_i = C_{i,1} \wedge \dots \wedge C_{i,s_i}$ et $C_{i,j} \in V \cup \neg V$.
- On dit que $f \in F$ est sous forme normale conjonctive (FNC) ssi $f = B_1 \wedge \dots \wedge B_p$ où $B_i = C_{i,1} \vee \dots \vee C_{i,s_i}$ et $C_{i,j} \in V \cup \neg V$

prop:
- $\forall f \in F$, $\exists f', f''$, f' sous FNC, f'' sous FND et $f \equiv f' \equiv f''$
- $\forall f \in F$, $\exists g$ sous FNC telle que f et g sont équisatisfiables

def:
- On note $\varphi \to \psi_1, \psi_0$ pour $(\varphi \wedge \psi_1) \vee (\neg \varphi \wedge \psi_0)$
- L'expansion de Shannon de φ sur x est $\varphi \uparrow^x = x \to \varphi[1/x], \varphi[0/x]$.

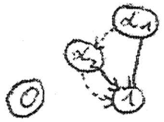
def : on dit que φ est sous forme if-then-else ssi :
- $\varphi = 0$ ou $\varphi = 1$, • $\varphi = x \to \psi_1, \psi_2$ où $\begin{cases} \psi_1, \psi_2 \text{ sont sous forme if-then-else} \\ x \in V \text{ n'apparait pas dans } \psi_1, \psi_2 \end{cases}$

thm : $\forall f \in F, \exists g$ sous forme if then else tel que $f \equiv g$. Si on fixe un ordre sur V, g est unique.

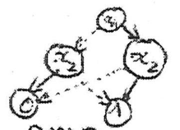
def : Un diagramme de décision binaire (BDD) est un graphe orienté acyclique ayant une racine u et
- deux noeuds terminaux de degré sortant 0, notés 0 et 1
- tel que tous les autres noeuds ont un degré sortant 2 et sont étiquetés par des éléments de V

def :
- Un BDD est dit ordonné ssi il existe un ordre $x_1 < \dots < x_n$ sur ses variables tel que tout chemin $u \rightsquigarrow 0$ ou $u \rightsquigarrow 1$ rencontre les x_i dans un ordre croissant.
- Un BDD est dit réduit ssi $\begin{cases} \text{tout noeud interne a deux fils distincts} \\ \text{deux noeuds ayant même étiquette et mêmes fils sont égaux} \end{cases}$

ex :



OBDD non réduit



ROBDD

3) Le problème SAT

def :
- On appelle SAT le problème consistant à déterminer si $f \in F$ est satisfiable
- n-SAT est la restriction de SAT aux formules du type
$$C_1 \wedge \dots \wedge C_k \text{ et } C_i = x_{i,1} \vee \dots \vee x_{i,j_i} \text{ où } j_i \leq n, \; x_{i,j} \in V \cup \neg V$$

thm (Cook) : SAT est NP-complet si V est infini (DEV).

rmq : Si V est fini, il suffit de tester les $2^{|V|}$ valuations possibles donc on a un algorithme en $\mathcal{O}(1)$.

cor : 3-SAT est NP-complet.

exemples d'algorithmes pour SAT :
- algorithme naïf $\mathcal{O}(2^n)$ où n est le nombre de variables de la formule en entrée.
- mise sous FND : quand la FND est trouvée, $\mathcal{O}(1)$ mais la FND a une taille exponentielle en la taille de la formule de départ.
- ROBDD : choisir un ordre sur V, construire l'unique ROBDD associé à la formule d'entrée puis vérifier que le ROBDD n'est pas réduit à 0
L'algorithme est polynômial mais le nombre de noeuds du BDD varie exponentiellement selon l'ordre choisi sur V.

applications : On utilise la NP-complétude de SAT pour prouver celle d'autres problèmes :
- sac à dos • chemin hamiltonien • existence de clique
- couverture des sommets

thm : 2-SAT est résoluble en temps polynomial via l'algorithme de Kosaraju

thm : La restriction de SAT aux formules sous la forme
$$x_1 \vee \dots \vee x_n \text{ où } x_i \in V \cup \neg V \text{ et au plus un des } x_i \text{ appartient à } V$$
est résoluble en temps polynômial.

application : Pour résoudre des problèmes, on les encode par des formules et on applique un algorithme pour SAT.

ex : problème des n dames : on prend une variable par case de l'échiquier et on encode chaque contrainte "deux dames ne sont pas sur la même ligne". Trouver une valuation satisfaisant les contraintes donne une configuration des n dames

III Systèmes de déduction

1) Coupure

def : Une clause est une formule de la forme
$$p_1 \vee \dots \vee p_n \vee \neg q_1 \vee \dots \vee \neg q_m \text{ avec } p_i, q_i \in V.$$
Les p_i sont les variables positives, q_i les variables négatives.
On note $\Delta = \{p_1, \dots, p_n\}$, $\Gamma = \{q_1, \dots, q_m\}$

prop : toute formule est équivalente (et même équisatisfiable) à la conjonction d'un nombre fini de clauses (mise sous FNC)

règle de coupure : Soit $C_1 = (\Gamma_1, \Delta_1)$, $C_2 = (\Gamma_2, \Delta_2)$ telles que $p \in \Gamma_1 \cap \Delta_2$.
On dit que $C = (\Gamma, \Delta)$ est obtenue par coupure sur p de C_1 et C_2 ssi $\Gamma = (\Gamma_1 \setminus \{p\}) \cup \Gamma_2$, $\Delta = \Delta_1 \cup (\Delta_2 \setminus \{p\})$. On note $\dfrac{C_1 \quad C_2}{C}$

ex :
$$\frac{(\{p, q\}, \{r\}) \qquad (\{q\}, \{p\})}{(\{q\}, \{r\})} \text{ par coupure sur } p.$$

def :
- Une preuve par coupure est l'application d'une suite finie de coupures à partir d'un ensemble de clauses
- Une réfutation par coupure est la preuve par coupure de la clause vide $\square = (\{\}, \{\})$

def : On note $S \vdash C$ si on peut prouver C par coupure à partir de S.

thm (correction) : Si $S \vdash C$, $S \models C$. Autrement dit, toute formule prouvable par coupure à partir de S est conséquence sémantique de S.

déf : Soit S un ensemble de clauses, $p \in V$ apparaissant positivement et négativement dans S. On note S_p l'ensemble des clauses où p apparaît.
Le résolvant $Res(S_p)$ est l'ensemble des clauses obtenues par coupure sur p de deux éléments de S.

prop : S est satisfiable ssi $(S \setminus S_p) \cup Res(S_p)$ l'est

thm (complétude) : si $S \models C$, $S \vdash C$. Autrement dit, si C est conséquence sémantique de S, il existe une preuve de C à partir de S

app : la méthode de résolution pour les formules du premier ordre est complète.

2) Déduction naturelle

a) Logique minimale (NM)

déf : L'ensemble des preuves est défini inductivement comme l'ensemble des couples (Γ, A) tels que :

- $A \in \Gamma$ (on note $\overline{\Gamma, A \vdash A}^{\,ax}$)
- A est obtenu par des règles d'inférence du type $\dfrac{\text{hypothèses}}{\text{conclusions}}$

Les règles d'inférences sont données pour chaque élément de C. une ou plusieurs règles d'introduction et une ou plusieurs règles d'élimination.

ex :

$$\dfrac{\Gamma \vdash A \qquad \Delta \vdash A \Rightarrow B}{\Gamma, \Delta \vdash B}\ \text{elim} \Rightarrow$$

$$\dfrac{\Gamma \vdash A \vee B \qquad \Delta, A \vdash C \qquad \Delta', B \vdash C}{\Gamma, \Delta, \Delta' \vdash C}\ \text{elim} \vee$$

b) Logique intuitionniste (NJ) (on peut donner une sémantique et ce serait complet mais pas la même sémantique qu'en logique (classique?))

On ajoute les axiomes concernant $\neg$:

$$\dfrac{\Gamma, A \vdash \neg B \qquad \Delta, A \vdash B}{\Gamma, \Delta \vdash \neg A}\ \text{intro} \neg \qquad \dfrac{\Gamma \vdash \neg A \qquad \Delta \vdash A}{\Gamma, \Delta \vdash B}\ \text{elim} \neg$$

prop : • intro $\neg$ est prouvable dans NM
• NJ est plus forte que NM

c) Logique classique (NK)

NJ n'est pas complet, on ajoute l'axiome du tiers exclu sous trois formes équivalentes :

$$\overline{\Gamma \vdash A \vee \neg A}^{\,ax}, \qquad \dfrac{\Gamma \vdash \neg\neg A}{\Gamma \vdash A} \quad \text{ou} \quad \dfrac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A}$$

où $\bot$ désigne la clause contradictoire.

rmq : on peut voir $\neg A$ comme un raccourci pour $A \Rightarrow \bot$

prop : NK est strictement plus forte que NJ

ex : la loi de Peirce $(((p \Rightarrow q) \Rightarrow p) \Rightarrow p)$ est prouvable dans NK mais pas dans NJ

thm : Le système de déduction $\vdash_{NK}$ est complet (& correct).

Réf : Cory
Lassaigne

Questions BDD → prendre une formule et explique les BDD à partir de ça

: $x \wedge (\neg y \vee z) \equiv x \rightarrow (y \rightarrow z, 1), 0$

mise sous forme de Shanon, ordre sur les variables $x < y < z$ → ordre dans lequel on croise les variables.

$x \rightarrow \underbrace{\neg y \vee z}^{x=1}, \overbrace{0}^{x=0}$

$y \rightarrow \underset{y=1}{z}, \underset{y=0}{1}$

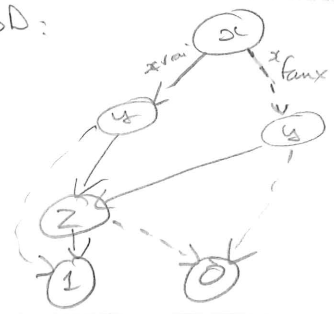
autre exemple:

$(x \vee y) \wedge (\neg y \vee z)$

$\equiv x \rightarrow (y \rightarrow (z \rightarrow 1, 0), 1), (y \rightarrow (z \rightarrow 1, 0), 0)$

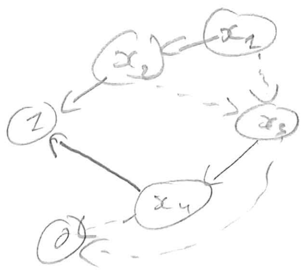
→ forme If then else

BDD:



L'ordre des variables est important.

ex: $(x_1 \wedge x_2) \vee (x_3 \wedge x_4)$



$x_1 < x_2 < x_3 < x_4$

pour $x_1 < x_3 < x_2 < x_4$