Bases de données
Cours 5 : Logique et bases de données

RAPPEL 1 : CALCUL PROPOSITIONNEL

Odile PAPINI,

ESIL
Université de la méditerranée
Odile.Papini@esil.univ-mrs.fr
http://odile.papini.perso.esil.univmed.fr

Plan

- Introduction
- langage
- syntaxe
- système formel
- sémantique
- formes normales

Le langage du calcul propositionnel : $\mathcal{L}$

Vocabulaire

un ensemble infini dénombrable de variables propositionnelles ou **propositions**

les constantes : 0 (Faux) et 1 (Vrai)

les connecteurs : $\neg \quad \wedge \quad \vee \quad \rightarrow \quad \leftrightarrow$

les parenthèses

Définitions

formules bien formées du calcul propositionnel :

- 0 et 1 sont des formules

- une variable propositionnelle est une formule

- si P et Q sont des formules alors

$$\neg\, P, \quad P \wedge Q, \quad P \vee Q, \quad P \rightarrow Q, \quad P \leftrightarrow Q$$

  sont des formules

Système formel du calcul propositionnel

les axiomes

soit P, Q, R des formules propositionnelles

A1) $(P \rightarrow (Q \rightarrow P))$

A2) $((P \rightarrow (Q \rightarrow R)) \rightarrow ((P \rightarrow Q) \rightarrow (P \rightarrow R)))$

A3) $((\neg\, P \rightarrow \neg\, Q) \rightarrow (Q \rightarrow P))$

règles de déduction

modus ponens

$$\frac{\vdash P, \vdash P \rightarrow Q}{\vdash Q}$$

règle de substitution

remplacer dans un théorème une variable propositionnelle, partout où elle figure, par :

- une autre variable propositionnelle
- ou une formule bien formée

Définitions

soit P et Q des formules propositionnelles :

D1) $P \rightarrow Q =_{def} \neg P \vee Q$

D2) $P \wedge Q =_{def} \neg(\neg P \vee \neg Q)$

D3) $P \leftrightarrow Q =_{def} (\neg P \vee Q) \wedge (P \vee \neg Q)$

Définition de la déduction

une **déduction** à partir d'hypothèses $H_1, H_2, \cdots, H_m$ est une suite de formules bien formées $F_1, F_2, \cdots, F_p$ où chaque F_i est soit :

- une hypothèse
- un axiome
- ou une formule obtenue à partir des règles d'inférence (substitution ou modus ponens) appliquées aux formules placées avant F_i dans la déduction

notation

$$H_1, H_2, \cdots, H_m \vdash F_p$$

théorème déduction sans hypothèse

$$\vdash F$$

proposition :

$$\forall P \in \mathcal{L} \quad \vdash (P \to P)$$

proposition :

$$\forall P_1, \cdots, P_{n-1} \in \mathcal{L} \times \cdots \times \mathcal{L}$$

$$\text{si} \quad P_1, \cdots, P_{n-1} \vdash (P_n \to Q) \quad \text{alors} \quad P_1, \cdots, P_n \vdash Q$$

proposition (théorème de déduction):

Soient $P_1, \cdots, P_n, Q \in \mathcal{L} \times \cdots \times \mathcal{L}$

$$\text{si} \quad P_1, \cdots, P_n \vdash Q \quad \text{alors} \quad P_1, \cdots, P_{n-1} \vdash (P_n \to Q)$$

Quelques théorèmes utiles

proposition :

$\forall P, Q, R \in \mathcal{L} \times \mathcal{L} \times \mathcal{L}$, toutes les formules suivantes sont des théorèmes :

- $\vdash ((P \rightarrow Q) \rightarrow ((Q \rightarrow R) \rightarrow (P \rightarrow R)))$
- $\vdash (P \rightarrow ((P \rightarrow Q) \rightarrow Q))$
- $\vdash (\neg P \rightarrow (P \rightarrow Q))$
- $\vdash (\neg\neg P \rightarrow P)$
- $\vdash (P \rightarrow \neg\neg P)$
- $\vdash ((P \rightarrow Q) \rightarrow (\neg Q \rightarrow \neg P))$
- $\vdash (P \rightarrow (\neg Q \rightarrow \neg(P \rightarrow Q)))$
- $\vdash ((Q \rightarrow P) \rightarrow ((\neg Q \rightarrow P) \rightarrow P))$

sémantique du calcul propositionnel

interprétation :

toute application σ de $\mathcal{P}$ (ensemble des propositions) dans $\{0,1\}$ telle que :

$$\sigma(0) = 0 \text{ et } \sigma(1) = 1$$

$\forall P, Q \in \mathcal{L} \times \mathcal{L}$,

- $\sigma(\neg P) = 1 - \sigma(P)$
- $\sigma(P \vee Q) = max(\sigma(P), \sigma(Q))$
- $\sigma(P \wedge Q) = min(\sigma(P), \sigma(Q))$
- $\sigma(P \rightarrow Q) = max((1 - \sigma(P)), \sigma(Q))$
- $\sigma(P \leftrightarrow Q) = min(max((1 - \sigma(P)), \sigma(Q)), max(\sigma(P), (1 - \sigma(Q))))$.

Tables de vérité

P	Q	$P \vee Q$	$P \wedge Q$	$P \longrightarrow Q$	$P \leftrightarrow Q$
1	1	1	1	1	1
1	0	1	0	0	0
0	1	1	0	1	0
0	0	0	0	1	1

quelques définitions

Soient $P \in \mathcal{L}$, $Q \in \mathcal{L}$ et $\mathcal{F} \subset \mathcal{L}$

P est une **tautologie** , $\models P$, si pour toute interprétation σ, $\sigma(P) = 1$

Q est une **conséquence** de P si pour toute interprétation σ, $\sigma(P) = 1$ alors $\sigma(Q) = 1$, on écrit $P \models Q$

Q est une **conséquence** de $\mathcal{F}$ si pour toute interprétation σ, tq $\forall P \in \mathcal{F}$, $\sigma(P) = 1$ alors $\sigma(Q) = 1$, on écrit $\mathcal{F} \models Q$

P est **satisfaisable** ou **cohérente** s'il existe une interprétation σ tq $\sigma(P) = 1$

$\mathcal{F}$ est **satisfaisable** s'il existe une interprétation σ tq $\forall P \in \mathcal{F}$, $\sigma(P) = 1$

P est **insatisfaisable** ou **incohérente** si pour toute interprétation σ, $\sigma(P) = 0$

$\mathcal{F}$ est **insatisfaisable** si pour toute interprétation σ, $\exists P \in \mathcal{F}$ tq $\sigma(P) = 0$

quelques propriétés

$\forall P, Q \in \mathcal{L} \times \mathcal{L}$,

- $\models (P \rightarrow Q)$ ssi $P \models Q$
- $\models (P \leftrightarrow Q)$ ssi $P \equiv Q$
- si $\models P$ et $\models (P \rightarrow Q)$ alors $\models Q$
- $\models (P \land Q)$ ssi $\models P$ et $\models Q$
- si $\models P$ ou $\models Q$ alors $\models (P \lor Q)$

quelques théorèmes

théorème (d'adéquation) : $\forall P \in \mathcal{L} \quad$ si $\vdash P$ alors $\models P$ (les formules qui sont des théorèmes sont des tautologies)

théorème (de complétude) : Soit $\mathcal{F}$ un ensemble de formules de $\mathcal{L}$, $\quad \forall P \in \mathcal{L} \quad \mathcal{F} \models P$ ssi $\mathcal{F} \vdash P$

théorème (de compacité) : Soit $\mathcal{F}$ un ensemble de formules de $\mathcal{L}$.
Si toute famille finie $\mathcal{F}' \subset \mathcal{F}$ est satisfaisable alors $\mathcal{F}$ est aussi satisfaisable.

théorème (de finitude) : Soit $\mathcal{F}$ un ensemble de formules de $\mathcal{L}$. Soit $Q \in \mathcal{L}$ si $\mathcal{F} \models Q$ alors $\exists \mathcal{F}' \subset \mathcal{F}$ fini tq $\mathcal{F}' \models Q$

théorème (de décidabilité) : $\quad \forall P \in \mathcal{L} \quad$, il existe un programme qui, pour toute formule P, indique en un temps fini si oui ou non $\vdash P$

formes normales

littéral : une proposition ou la négation d'une proposition

clause : disjonction de littéraux

cube : conjonction de littéraux

forme conjonctive normale : une conjonction de clauses

forme disjonctive normale : une disjonction de cubes

théorème (de normalisation) :

- $\forall P \in \mathcal{L}$, P admet une forme conjonctive normale qui lui est équivalente
- $\forall P \in \mathcal{L}$, P admet une forme disjonctive normale qui lui est équivalente

algorithme de normalisation

début

- **élimination des connecteurs $\rightarrow$ et $\leftrightarrow$**

  `remplacer` $P \leftrightarrow Q$ `par` $(P \rightarrow Q) \wedge (Q \rightarrow P)$

  `puis remplacer` $P \leftrightarrow Q$ `par` $\neg P \vee Q$

- **application des lois de Morgan**

  `remplacer` $\neg(P \wedge Q)$ `par` $\neg P \vee \neg Q$ `et` $\neg(P \vee Q)$ `par` $\neg P \wedge \neg Q$

- **élimination des doubles négations**

  `remplacer` $\neg\neg P$ `par` P

- **application des règles de distributivité**

  `remplacer` $P \vee (Q \wedge R)$ `par` $(P \vee Q) \wedge (P \vee R)$

  `et` $(P \wedge Q) \vee R$ `par` $(P \vee R) \wedge (Q \vee R)$

fin