Logique et représentation des connaissances

CALCUL PROPOSITIONNEL

Odile PAPINI,

Université de la Méditerranée.

odile.papini@esil.univmed.fr

http://odile.papini.perso.esil.univmed.fr/index.html

Plan

Partie I : La logique propositionnelle

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

Partie II : Raisonnement en logique propositionnelle

- introduction
- tables de vérité
- arbres sémantiques
- résolution

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 :

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

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

proposition (théorème de déduction):

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

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

Quelques théorèmes utiles

proposition :

$\forall P, Q, R \in \mathcal{L}$, 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 :

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

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

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

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

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

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

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) : $\forall P \in \mathcal{L}$, 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` $P \rightarrow Q$ `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

Introduction

3 types de raisonnement

raisonnement déductif

raisonnement abductif

raisonnement inductif

Exemple

$$\text{si} \quad \models A, \quad \text{et} \quad \models A \rightarrow B \quad \text{alors} \quad \models B$$

raisonnement déductif :

connaissant $\models A$ et $\models A \rightarrow B$ on peut inférer $\models B$

raisonnement abductif :

connaissant $\models A \rightarrow B$ et $\models B$ on peut inférer $\models A$

raisonnement inductif :

connaissant $\models A$ et $\models B$ on peut inférer $\models A \rightarrow B$

Raisonnement déductif

lois générales $H_1, H_2, \cdots, H_n \quad \models C$

hypothèses $H_1, H_2, \cdots, H_n$

conclusion C

Raisonnement déductif

définition (rappel)

$\mathcal{F}$ un ensemble de formules de $\mathcal{L}$ et I une interprétation,

$$\mathcal{F} \text{ est insatisfaisable ou incohérent ssi } \forall I \text{ telle que } \exists F \in \mathcal{F},\ I(F) = 0.$$

proposition

$F \in \mathcal{L}, \quad \mathcal{F} \models F$ ssi $\mathcal{F} \cup \{\neg F\}$ est insatisfaisable ou incohérent.

on ramène le problème de conséquence logique à celui de la cohérence ou de la satisfaisabilité.

réfutation : on cherche à montrer l'incohérence

Tables de vérité

p_1	p_2	p_3	$\cdots$	p_n	F
0	0	0	$\cdots$	0	
0	0	0	$\cdots$	1	
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	
1	1	1	$\cdots$	1	

pour n variables 2^n lignes : complexité $O(2^n)$

Tables de vérité

Exercices

- $S = \{p \vee q,\ p \vee r,\ \neg q \vee \neg r,\ \neg p\}$ est-il satisfaisable ?

- En utilisant les tables de vérité, montrer que

$$\{p \rightarrow r,\ q \rightarrow r\} \quad \models \quad (p \vee q) \rightarrow r$$

Arbres sémantiques

S : ensemble de clauses correspondant à la forme CNF de $F \in \mathcal{L}$, et $p_1, p_2, \cdots, p_n \in \mathcal{P}$ apparaissant dans S

principe : construction d'un arbre

- chaque arc est étiqueté par un littéral p_i ou $\neg p_i$

- les littéraux étiquetant les deux arcs issus d'un même sommet sont opposés

- aucune branche ne comporte plus d'une occurrence de chaque atome

Chaque noeud de l'arbre correspond à une interprétation partielle de $p_1, p_2, \cdots, p_n$.

Arbres sémantiques

- un arbre est **complet** ssi chaque feuille de l'arbre correspond à une interprétation de $p_1, p_2, \cdots, p_n$.

- une branche est **fermée** ssi il existe un nœud n tel que l'interprétation partielle correspondante est un contre-modèle d'une clause de S et tel que les interprétations partielles de tout noeud ancêtre de n ne sont pas des contre-modèles d'une clause de S.

- un arbre sémantique est est **fermée** ssi toutes ses branches sont fermées.

Arbres sémantiques

proposition :

S un ensemble de clauses est incohérent ssi pour tout arbre sémantique complet correspondant à S, il existe une arbre fini fermé correspondant à S.

proposition :

S un ensemble de clauses est incohérent ssi il existe un sous-ensemble de clauses S', $S' \subset S$ incohérent.

limites

L'arbre sémantique contient 2^n feuilles : complexité $O(2^n)$

Arbres sémantiques

Exemple

Arbre sémantique pour

$$F = ((p \longrightarrow q) \wedge p) \longrightarrow q$$

Arbres sémantiques

Exercices

- $S = \{p \vee q,\ p \vee r,\ \neg q \vee \neg r,\ \neg p\}$ est-il satisfaisable ?

- En utilisant les les arbres sémantiques, montrer que

$$\{p \rightarrow r,\ q \rightarrow r\} \quad \models \quad (p \vee q) \rightarrow r$$

Méthode de Quine

amélioration de la méthode des arbres sémantiques : élagage

principe :

coupure dans l'arbre sémantique sur un nœud si toutes les branches issues de ce nœud donnent la même valeur de vérité

S : ensemble de clauses (CNF), p : proposition, $S = S_p \cup S_{\neg p} \cup S''$
S_p : ensemble des clauses où p apparaît
$S_{\neg p}$: ensemble des clauses où $\neg p$ apparaît, $S'' = S \backslash (S_p \cup S_{\neg p})$.
S'_p : ensemble des clauses de S_p privées de p
$S'_{\neg p}$: ensemble des clauses $S_{\neg p}$ privées de $\neg p$.

proposition :

S incohérent ssi $S'_p \cup S''$ et $S'_{\neg p} \cup S''$ incohérents.

Algorithme de Quine

quine(S: un ensemble de clauses) : (coherent)
données : S_p, $S_{\neg p}$, S", S'_p, $S'_{\neg p}$, $coherent$
début
si $S = \emptyset$ **alors** $coherent \leftarrow vrai$
  si $S = \{\square\}$ **alors** $coherent \leftarrow faux$
  sinon
  choisir un atome $p \in S$
  calculer S_p, $S_{\neg p}$ et $S" = S \backslash (S_p \cup S_{\neg p})$
  calculer S'_p et $S'_{\neg p}$
  quine($S'_p \cup S"$)
  quine($S'_{\neg p} \cup S"$)
  finsi
finsi
fin

Méthode de Davis et Putnam (DPLL)

DPLL (Davis, Putnam, Davis, Logemann, Loveland) : amélioration de la méthode de Quine

principe :

Tirer parti de la forme CNF pour sélectionner au mieux les propositions afin d'éviter la construction de branches inutiles.

- clause monolittérale : clause contenant seulement p, (respectivement $\neg p$).
- présence d'un seul des littéraux p ou $\neg p$

Méthode de Davis et Putnam

Une proposition p est sélectionnée en priorité si :

- **si S contient une clause monolittérale**,
  S'_p (respectivement $S'_{\neg p}$ contient la clause vide, donc $S'_p \cup S$" (respectivement $S'_{\neg p} \cup S$") est incohérent,
  la recherche d'incohérence se ramène seulement à celle de $S'_{\neg p} \cup S$" (respectivement $S'_p \cup S$").

- **Si un seul des littéraux p ou $\neg p$ apparaît dans S.**
  S_p ou $S_{\neg p}$ est vide,
  la recherche d'incohérence se ramène seulement à celle de $S'_{\neg p} \cup S$" ou à celle de $S'_p \cup S$".

Méthode de Davis et Putnam : Exemple

$$S = \{\neg a \vee b,\ a,\ \neg b\}$$

choix du littéral a :
$S_a = \{a\}$, $S_{\neg a} = \{\neg a \vee b\}$ et $S" = S \backslash (S_a \cup S_{\neg a}) = \{\neg b\}$
$S'_a = \{\Box\}$ et $S'_{\neg a} = \{b\}$

S incohérent ssi $S'_a \cup S" = \{b\}$ et $S'_{\neg a} \cup S" = \{b, \neg b\}$ le sont aussi.

$S'_a \cup S" = \{b, \Box\}$: il est incohérent,

$S'_{\neg a} \cup S" = \{b, \neg b\} = T$
choix du littéral b :
$T_b = \{b\}$, $T_{\neg b} = \{\neg b\}$ et $T" = \emptyset$
$T'_b = \{\Box\}$ et $T'_b \cup T" = \{\Box\}$: incohérent
$T'_{\neg b} = \{\Box\}$ et $T'_{\neg b} \cup T" = \{\Box\}$: incohérent
T incohérent ssi $T'_b \cup T"$ et $T'_{\neg b} \cup T"$ le sont aussi

Méthode de Davis et Putnam

Exercice

- $S = \{p \vee q,\ p \vee r,\ \neg q \vee \neg r,\ \neg p\}$ est-il satisfaisable ?

- En utilisant la méthode de Davis et Putnam, montrer que

$$\{p \rightarrow r,\ q \rightarrow r\} \quad \models \quad (p \vee q) \rightarrow r$$

Problème SAT

Etant donne une formule booléenne mise sous forme CNF , existe t-il une affectation de valeurs (0 ou 1) des variables rendant la formule vraie ?

En informatique théorique : complexité
SAT est un problème NP-Complet de référence [Cook 1971]

En pratique : solveurs SAT très efficaces pour résoudre des problèmes combinatoires
http://www.SATlive.org
http://www.satcompetition.org/

Résolution en calcul propositionnel

principe de la résolution J. -A. Robinson

Soit A, B, X des formules propositionnelles

On suppose que $A \vee X$ et $B \vee \neg X$ sont cohérentes

- si X est cohérente alors B est cohérente
- $\neg X$ est cohérente alors A est cohérente

dans les 2 cas $A \vee B$ est cohérente

plus formellement : $\{A \vee X, B \vee \neg X\} \models A \vee B$

proposition (résolution)

S : un ensemble de clauses, $c_1,\ c_2\ \in S$

si l est un littéral tel que :

$c_1 = l_{i_1} \vee l_{i_2} \vee \cdots \vee l_{i_n} \vee l$ et $c_2 = l_{j_1} \vee l_{j_2} \vee \cdots \vee l_{j_m} \vee \neg l$

alors $S \models r$,

avec $r = l_{i_1} \vee l_{i_2} \vee \cdots \vee l_{i_n} \vee l_{j_1} \vee l_{j_2} \vee \cdots \vee l_{j_m}$

On appelle r la **résolvante**

principe de la résolution

S : une ensemble de clauses (CNF correspondant à $\mathcal{F}$ ensemble de formules propositionnelles)

proposition : S est incohérent ssi $S \models \Box$

Algorithme de résolution

<u>`début`</u>

<u>`tant que`</u> $\Box \notin S$ <u>`faire`</u>

`choisir` l, c_1, c_2 `tels que` $l \in c_1$ `et` $\neg l \in c_2$

`calculer la résolvante` r

`remplacer` S `par` $S \cup \{r\}$

<u>`fin tant que`</u>

<u>`fin`</u>

Résolution en calcul propositionnel

Exercice

- $S = \{p \vee q,\ p \vee r,\ \neg q \vee \neg r,\ \neg p\}$ est-il satisfaisable ?

- En utilisant la résolution, montrer que

$$\{p \rightarrow r,\ q \rightarrow r\} \quad \models \quad (p \vee q) \rightarrow r$$

propriétés de la résolution

proposition :

S : un ensemble de clauses, S' : ensemble de clauses après résolution

si S est satisfaisable alors S' est satisfaisable

proposition :

F : une formule propositionnelle

si F a une preuve par résolution alors F est une tautologie

théorème de complétude :

F : une formule propositionnelle

si F est une tautologie alors F a une preuve par résolution

Graphes de résolution

principe : résolution graphique par un graphe acyclique orienté (DAG en anglais) construire un graphe :

- **chaque noeud est soit une racine soit a deux parents,**
  - **un noeud racine est étiqueté par une clause initiale**
  - **un noeud qui a 2 parents est étiqueté par la résolvante produite à partir des 2 parents**

C_i est un ancêtre de C_j si il existe un chemin direct de C_i à C_j dans le graphe

Strategies de résolution

Résolution linéaire

principe : résolution à partir de C_i et C_j ssi dans le graphe de résolution si l'une des clauses est

- **soit une racine du graphe (clause initiale)**
- **soit C_i est un ancêtre de C_j dans le graphe**

Strategies de résolution

Résolution monolittérale

clause **monolittérale :** clause ne contenant qu'un littéral

principe : résolution à partir de C_i et C_j ssi l'une des clauses est une clause monolittérale

adaptée pour **les clauses de Horn :** contenant **au plus un littéral positif**

- **résolution complète pour les clauses de Horn**
- **efficacité, compexité en temps : linéaire en taille de la forme clausale**
- **opération de simplification**