Programmation Logique

Sup Galilée-INFO2

Céline Rouveirol

2010-2011

Plan

1. Chapitre 0 : Généralités
2. Ch. 1 : Résolution en logique propositionnelle
   - Rappels logique propositionnelle
   - Résolution en logique propositionnelle

Objectifs et plan du cours

- Maîtrise des concepts de :
  - la Programmation Logique
  - PROLOG
- Introduction à l'Intelligence Artificielle

Sources du cours - Bibliographie

- J.-M. Alliot, T. Schiex, P. Brisset, F. Garcia, Intelligence Artificielle et Informatique Théorique, 2ème ed., 2002
- Poly de cours de logique, S. Cerrito `http://www.lami.univ-evry.fr/~serena/`
- P. Flach, Simply Logical : Intelligent Reasoning by Example John Wiley 1994
- M. R. Genesereth et N. J. Nilsson, Logical Foundations of Artificial Intelligence, Morgan Kaufmann, 1987
- J. Lloyd, Foundations of Logic Programming, Springer, 1987
- L. Sterling et E. Shapiro, The Art of Prolog : Advanced Programming Techniques, 2nd ed, 1994

Syntaxe logique propositionnelle

atomes ou variable propositionnelle : des énoncés dont nous ne cherchons pas pas connaître la structure interne. Notés p, q, r, s, ...

connecteurs : les connecteurs $\rightarrow$, $\leftrightarrow$, $\neg$, $\vee$, $\wedge$, ...

formule
- un atome
- si A est une formule, $\neg A$ est une formule
- si A et B sont des formules $A \leftrightarrow B$, $A \rightarrow B$, $A \vee B$ et $A \wedge B$ sont des formules
- formule vide notée $\Box$, et définie par $(a \wedge \neg a)$

littéral : positif = atome et littéral négatif = négation d'un atome

Sémantique

- Valuation : à chaque variable propositionnelle de la formule, on associe une valeur de vérité $\{t, f\}$.
- Interprétation d'une formule : une valuation pour les variables prop. de la formule. Par abus de language, on liste parfois pour décrire une interprétation I uniquement les variables prop. qui ont pour valeur t dans I. Notation $I(p)$ (resp. $I(F)$) : la valeur de vérité de la variable p (resp. de la formule F) dans l'interprétation I

Tables de vérité

On associe à chaque connecteur (unaire, binaire) une fonction booléenne qui permet de calculer la v. v. d'une formule dont ce connecteur est le connecteur principal, à partir de la v.v. de ses sous-formules A et B.

A	B	$\neg A$	$A \vee B$	$A \wedge B$	$A \rightarrow B$	$A \leftrightarrow B$
t	t	f	t	t	t	t
t	f	f	t	f	f	f
f	t	t	t	f	t	f
f	f	t	f	f	t	t

Sémantique

- Satisfiabilité ($\models$). Soit F une formule et I une interprétation de F. On définit la relation de satisfiabilité ($I \models F$) par :
  - F est une variable V : $I \models F$ ssi $i(V) = t$
  - F est une formule de la forme $\neg G$: $I \models F$ ssi $i(G) = f$
  - F est une formule de la forme $G \vee H$: $I \models F$ ssi $i(G) = t$ ou $i(H) = t$
  - F est une formule de la forme $G \wedge H$, $I \models F$ ssi $I(G) = t$ et $I(H) = t$

Sémantique

modèle I est un modèle pour une formule si $I \models F$. I est un modèle pour un ensemble de formules $F_i, 1 \leq i \leq n$ si pour tout i $I \models F_i$

conséquence logique La formule F est conséquence logique de la formule G (noté $G \models F$) si pour tout I tel que $I \models G$ alors $I \models F$

satisfiabilité
- Une formule est *satisfiable* si il existe au moins une interprétation I telle $I \models F$. Une formule est dite *insatisfiable* s'il n'existe aucun I telle que $I \models F$.
- Une formule qui est vraie pour toute interprétation I est une *tautologie*. On dit aussi qu'elle est *valide*. Notation : $\models F$.

Sémantique (suite)

- Théorème de déduction : $F_1, \ldots, F_n \models G$ ssi $(F_1 \wedge \ldots F_n \rightarrow G)$ est valide
- Soit G une formule quelconque. Si E est un ensemble insatisfiable de formules, alors G est conséquence logique de E
- Si G est valide, G est conséquence logique d'un ensemble quelconque de formules
- Soit E un ensemble de formules et G une formule, $E \models G$ ssi $E, \neg G$ est insatisfiable.

Equivalence logique

- Soit F et G deux formules quelconques. F est logiquement équivalente à G, noté $F \equiv G$ ssi $F \models G$ et $G \models F$.
- $F \equiv G$ ssi $\models F \leftrightarrow G$
- Quelques équivalences utiles
  1. $F \leftrightarrow G \equiv F \rightarrow G \wedge G \rightarrow F$
  2. $F \rightarrow G \equiv \neg F \vee G$
  3. $\neg(F \wedge G) \equiv \neg F \vee \neg G$ et $\neg(F \vee G) \equiv \neg F \wedge \neg G$ (loi de Morgan)
  4. $\neg\neg F \equiv F$ (loi de double négation)
  5. $F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$ et $F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$ (distributivité de $\vee$ sur $\wedge$ et vice versa)

Axiomatique du calcul propositionnel

- Méthodes algorithmiques pour décider automatiquement et aussi efficacement que possible de la satisfiabilité d'un ensemble de formules.
- Un système de preuve est défini par un ensemble de règles (dites *règles d'inférence*) à appliquer "mécaniquement", en ne tenant compte que de la syntaxe des formules traitées.
- Notation :

$$\frac{P_1 \qquad P_2}{C}$$

où P_1 et P_2 sont les prémisses du séquent et C sa conclusion.
- Règle du modus ponens :

$$\frac{A \rightarrow B \qquad A}{B}$$

Propriétés d'un système d'inférence

On note classiquement $\vdash$ la notion de dérivation par un système d'inférence.

Correction Un système d'inférence est correct si, pour toute règle du système, la formule conclusion est conséquence logique de sa formule prémisse (de la conjonction de ses formules prémisses) : $\vdash$ est correcte si .

Complétude Un système d'inférence est complet s'il suffit à prouver toute formule valide : pour toute formule F telle que $\models F$, alors $\vdash F$.

Formes normales

Littéral : formule atomique ou négation d'une formule atomique

Clause : formule de la forme $l_1 \vee \ldots \vee l_n$ où chaque l_i est un littéral

FNC : formule de la forme $D_1 \wedge \ldots \wedge D_k$ où $k > 0$ et chaque D_i est une clause

Conjonction élémentaire : formule de la forme $l_1 \wedge \ldots \wedge l_n$ où chaque l_i est un littéral

FND : formule de la forme $C_1 \vee \ldots \vee C_k$ où $k > 0$ et chaque C_i est une conjonction élémentaires

Toute formule admet une forme normale conjonctive et disjonctive qui lui est logiquement équivalente

Algorithme de mise sous FNC

Utilisation des équivalences vues au transparent 11.

1. Eliminer les $\rightarrow$ et les $\leftrightarrow$: application des équivalences 1 et 2
2. Limiter la portée des négations (faire "descendre" les négations) : application des équivalences 3 et 4
3. Mettre sous forme clausale : application des équivalences 5

Notes :

- Pour une formule f donnée, il existe plusieurs f' telles que $f \equiv f'$ et f' en FNC.
- La taille de f' peut croître (exponentiellement)

Exemple de mise sous FNC

$$(p \rightarrow (q \leftrightarrow \neg r)) \wedge (r \leftrightarrow \neg s) \equiv$$
$$\neg p \vee ((q \rightarrow \neg r) \wedge (\neg r \rightarrow q)) \wedge (r \leftrightarrow \neg s) \equiv$$
$$\neg p \vee ((q \rightarrow \neg r) \wedge (\neg r \rightarrow q)) \wedge (r \rightarrow \neg s) \wedge (\neg s \rightarrow r) \equiv$$
$$\neg p \vee ((\neg q \vee \neg r) \wedge (r \vee q)) \wedge (r \rightarrow \neg s) \wedge (\neg s \rightarrow r) \equiv$$
$$\neg p \vee ((\neg q \wedge r) \vee (\neg r \wedge q)) \wedge (r \rightarrow \neg s) \wedge (\neg s \rightarrow r) \equiv$$
$$(\neg p \vee \neg q \vee \neg r) \wedge (\neg p \vee r \vee q) \wedge (\neg r \vee \neg s) \wedge (s \vee r)$$

Résolution en calcul propositionnel

Méthode de démonstration par réfutation.

- Ensemble de formules insatisfiable : un ensemble de formules $\{F_i\}, 1 \leq i \leq n$ est insatifiable ssi il n'existe aucun modèle M tel que pour tout i, $M \models F_i$.
- Principe de déduction : une formule G est conséquence logique d'un ensemble de formules $F = \{F_i, 1 \leq i \leq n\}$ ssi $F \cup \neg G$ est insatisfiable.

Résolution en calcul propositionnel

Deux règles d'inférence :

- Résolution. Soient C_1 et C_2 deux clauses, soit p un littéral.

$$\frac{C_1 \vee p \qquad C_2 \vee \neg p}{R = C_1 \vee C_2}$$

  On appelle R la résolvante de $C_1 \vee p$ et $C_2 \vee \neg p$
- Factorisation. Soient C une clause et p une littéral

$$\frac{C \vee p \vee p}{C \vee p}$$

Dérivation par résolution propositionnelle

Soient E un ensemble de clauses et c une clause. Une *dérivation* par résolution de c à partir des *hypothèses* E est une suite de clause $r_1, \dots, r_n$ telles que pour tout i, $1 \leq i \leq n$:

- $r_i \in E$
- il existe un $j \leq i$ tel que $\dfrac{r_j}{r_i}$ par factorisation
- il existe $j, k \leq i$ tels que $\dfrac{r_j \qquad r_k}{r_i}$ par résolution (r_i est appelé résolvante de r_j et r_k
- Une *réfutation* de E est une dérivation par résolution de $\square$ à partir des hypothèses de E

Exemple de réfutation

$S = \{p \vee q \vee r; \neg p \vee q \vee r; \neg q \vee r; \neg r\}$

$$\cfrac{\cfrac{\cfrac{\neg q \vee r \qquad \neg r}{\neg q} \qquad \cfrac{\cfrac{p \vee q \vee r \qquad \neg p \vee q \vee r}{q \vee q \vee r}}{q \vee r}}{r} \qquad \neg r}{\square}$$

Propriétés de la résolution propositionnelle

Correction Soient A, B et C des clauses. $\{A \vee C, B \vee \neg C\} \models A \vee B$

Non complétude Il existe une théorie clausale F et une clause C telle que $F \models C$, mais il n'existe pas de dérivation par résolution de C à partir de F.

Complétude pour la réfutation Un ensemble de formules F est insatisfiable ssi on peut dériver la clause vide $\Box$ de F par résolution. Si $F \models C$, il existe une réfutation à partir de $F \cup \neg C$.

Résolution et clause de Horn

Clauses de Horn une clause comprenant au plus un littéral positif.

Résolution unitaire au moins un des deux parents est unitaire

$$(\ell) : \frac{\ell \qquad L \vee \neg \ell}{L}$$

Stratégie de résolution

1. Si $\Box$ est dans F alors F est insatisfiable
2. Sinon, choisir une clause C et p une clause unitaire positive et C contient $\neg p$
3. Calculer la résolvante R de p et C
4. Remplacer dans F C par R

Résolution et clause de Horn

- La résolution unitaire n'est pas complète par réfutation pour les théories clausales quelconques. Exemple : $\{a \vee b; \neg a \vee b; a \vee \neg b; \neg a \vee \neg b\}$
- La résolution unitaire est complète par réfutation pour les ensembles de clauses de Horn. Exemple : $\{a; c; \neg a \vee b; \neg b \vee c\}$