31HU03MI 14-15
Logique 24 mars 2015

Feuille d'exercices n°4

Calcul propositionnel 3, résolution

Exercice 1. Soit $\mathcal{S}$ un ensemble de clauses (disjonctions de littéraux). Montrer que dans les deux cas suivant $\mathcal{S}$ est satisfaisable :

1. toutes les clauses de $\mathcal{S}$ contiennent une variable propositionnelle niée ;
2. toutes les clauses de $\mathcal{S}$ contiennent une variable propositionnelle non niée.

Exercice 2. Soient les quatre clauses suivantes :

1. $\neg q \vee p \vee r$ **2.** $p \vee q \vee s$ **3.** $\neg q \vee \neg r \vee p$ **4.** $\neg p \vee \neg q \vee \neg r$

Trouver une valuation qui ne satisfait pas cet ensemble de clauses. Trouver toutes les clauses dont il existe une preuve par coupure à partir de ces quatre clauses, et qui ne sont pas des tautologies. Trouver une valuation qui satisfait ces quatre clauses. Combien y-a-t-il de telles valuations ?

Exercice 3. Réfuter par coupure l'ensemble des clauses suivantes :

1. $\neg p \vee \neg q \vee r \vee s$ **3.** $\neg q \vee \neg r \vee s$ **5.** $\neg q \vee \neg s \vee p$ **7.** $p \vee q$
2. $\neg p \vee q \vee s$ **4.** $\neg p \vee \neg s$ **6.** $\neg q \vee p \vee r$

Exercice 4. Pour tout $n \in \mathbb{N}^*$, on appelle $\neg\mathbf{PHP}_n$ (PHP pour *Pigeon Hole Principle*), la formule obtenue par conjonction des clauses suivantes :

— $\varphi = \bigwedge_{i=1}^{n+1} p_{i,1} \vee \ldots \vee p_{i,n}$
— $\psi = \bigwedge_{j=1}^{n} \bigwedge_{1 \leq i < i' \leq n+1} (\neg p_{i,j} \vee \neg p_{i',j})$

En interprétant chaque $p_{i,j}$, pour $1 < j < n$ et $1 < i < n+1$, par *"l'objet i est dans la boite j"*, la formule $\neg\mathbf{PHP}_n$ affirme la négation du principe des tiroirs à savoir : que chaque objet $i \leq n+1$ est dans une boite j, $j \leq n$ (formule φ), et qu'il est possible que toute boite ne contienne pas deux objets (formule ψ).

1. Donner une réfutation de $\neg\mathbf{PHP}_3$ par résolution.

Exercice 5. Donner des contre-exemples montrant que la résolution unitaire n'est pas complète pour la réfutation des formules propositionnelles sous forme clausale.

Exercice 6. Une formule propositionnelle 2-FNC a toutes ses clauses de longueur au plus 2.

1. Soit p une constante propositionnelle et C et D des 2-clauses, telles que $p \in C$ et $\neg p \in D$. Que peut-on dire du résolvant de C et D ?
2. Montrer que toute réfutation d'une formule 2-FNC φ par la méthode de résolution est de longueur au plus $2|\mathsf{var}(\varphi)|^2 + 2|\mathsf{var}(\varphi)| + 1$
3. En déduire un algorithme de test de la satisfaisabilité d'une formule 2-FNC en $O(|\mathsf{var}(\varphi)|^2)$ étapes de résolutions.

Exercice 7. Une formule propositionnelle est dite *affine* si elle est équivalente à une conjonction de formules de la forme :

$$l_1 \oplus l_2 \oplus \ldots \oplus l_k$$

où l_i est un littéral pris sur un ensemble de variable $\mathcal{P}$ et $\oplus$ est le connecteur *ou exclusif* défini par : $0 \oplus 0 = 1 \oplus 1 = 0$, $0 \oplus 1 = 1 \oplus 0 = 1$.

1. Montrer que toute formule affine φ sur un ensemble de variables $\mathcal{P} = \{x_1, ..., x_n\}$ est équivalente à un système d'équations linéaires S_φ sur les mêmes variables au sens suivant :

   pour toute valuation $v : \mathcal{P} \to \{0, 1\}$, $v(\varphi) = 1$ ssi $(v(x_1), \ldots, v(x_n))$ est une solution de S_φ.
2. En déduire un algorithme « rapide » de test de satisfaisabilité pour φ.

Exercice 8 (subsumption). On dit qu'une clause (disjonction de littéraux) C *subsume* une clause C' si tous les littéraux de C apparaissent dans C' et on note $C \subset C'$.

1. Vérifier que $C \subset C'$ si et seulement si $C \to C'$ est une tautologie.
2. Soit $\mathcal{S}$ un ensemble de clauses, et $C \subset C'$. Montrer (sans faire appel à la complétude) que s'il existe une dérivation (par coupures) de longueur n à partir de $\mathcal{S} \cup \{C'\}$ de la clause D, alors il existe une dérivation de longueur $\leq n$ à partir de $\mathcal{S} \cup \{C\}$, d'une clause D_0 telle $D_0 \subset D$.
3. En déduire que si $C \in \mathcal{S}$, $C \subset C'$ et $\mathcal{S} \cup \{C'\}$ est réfutable par coupures, alors $\mathcal{S} \cup \{C\}$ est réfutable par coupures.
   Ce résultat justifie que dans une stratégie de réfutation par application de la règle de coupures, dès qu'une clause est subsumée par une clause dérivée, elle peut être éliminée (des clauses avec lesquelles on cherche à construire la réfutation).
4. Une clause est une tautologie si elle contient une variable propositionnelle et sa négation. Montrer (sans faire appel à la complétude) que si C est une tautologie et $\mathcal{S} \cup \{C\}$ est réfutable par coupures, $\mathcal{S}$ est réfutable par coupures.
   Dans une stratégie de réfutation par coupures on peut donc éliminer les tautologies.

Exercice 9 (complétude). On propose dans cet exercice une autre preuve de complétude différente de celle du cours, puis généralisée à un ensemble infini dénombrable de clauses. Les résultats de l'exercice précédent, qui ont été démontrés sans le théorème de complétude, peuvent donc être utilisés.

On suppose donné un système fini $\mathcal{S}$ de clauses qui n'est pas réfutable, et on va montrer l'existence d'une valuation qui le satisfait.

Ce système de clauses utilise un nombre fini de variables propositionnelles soient $p_1, \dots, p_n$.

On construit par récurrence sur n un arbre $\mathcal{A}_n$ de réfutation (d'un ensemble de clauses qui n'est pas déterminé a priori) « par le bas » :

— $\mathcal{A}_0$ est réduit à la clause vide (c'est l'arbre de réfutation de la clause vide !) ;
— $\mathcal{A}_{k+1}$ est l'arbre obtenu à partir de $\mathcal{A}_k$ en prolongeant $\mathcal{A}_k$ en toutes les feuilles qui ne sont pas des clauses subsumées par une clause de $\mathcal{S}$ par coupure sur la variable p_{k+1}.

Par exemple, si $\mathcal{S}$ ne contient pas la clause vide, l'arbre $\mathcal{A}_1$ est :

$$\frac{\neg p_1 \quad p_1}{\square}$$

et si $\mathcal{S}$ ne contient de plus ni la clause $\neg p_1$ ni la clause p_1, l'arbre $\mathcal{A}_2$ est :

$$\dfrac{\dfrac{\neg p_1, \neg p_2 \quad \neg p_1, p_2}{\neg p_1} \qquad \dfrac{p_1, \neg p_2 \quad p_1, p_2}{p_1}}{\square}$$

si $\mathcal{S}$ contient la clause $\neg p_1$ mais pas la clause p_1 ni la clause vide, $\mathcal{A}_2$ est :

$$\dfrac{\neg p_1 \qquad \dfrac{p_1, \neg p_2 \quad p_1, p_2}{p_1}}{\square}$$

1. Montrer par récurrence sur $k \leq n$ que si $\mathcal{S}$ n'est pas réfutable, l'arbre $\mathcal{A}_k$ possède toujours une clause feuille qui n'est pas subsumée par une clause de $\mathcal{S}$.
2. En déduire une valuation qui satisfait $\mathcal{S}$.
3. Généraliser ce résultat à un ensemble infini dénombrable de clauses $\mathcal{S}$. Un ensemble (non nécessairement fini) de formules propositionnelles est dit satisfaisable quand il existe une valuation qui satisfait toutes ses formules.
   Un ensemble dénombrable de clauses utilise un nombre au plus dénombrable de variables, soit $(p_n)_{n \in \mathbb{N}^*}$ ces variables. Une réfutation de $\mathcal{S}$ est par définition n'importe quelle réfutation d'un sous-ensemble fini de clauses de $\mathcal{S}$. On construira de la même façon que ci-dessus, la suite d'abres finis $\mathcal{A}_k$ telle que A_{k+1} prolonge A_k (qui forme donc un arbre potentiellement infini). On montrera qu'il existe une suite infinie de clauses $(C_k)_{k \in \mathbb{N}^*}$, telle que C_k n'utilise que les variables propositionnelles $p_1, \dots, p_k$, $C_k \subset C_{k+1}$, et aucune des clauses C_k n'est subsumée par une clause de $\mathcal{S}$ (remarque : la démonstration de l'existence de cette suite de clauses utilise une forme faible de l'axiome du choix). On en déduira la valuation cherchée.
4. Déduire du résultat précédent le *théorème de compacité* du calcul propositionnel pour un ensemble dénombrable de formules : un ensemble dénombrable $\mathcal{S}$ de formules propositionnelles est satisfaisable si et seulement si tous les sous-ensembles finis de $\mathcal{S}$ sont satisfaisables (le démontrer d'abord pour des clauses).