IENAC-S-96 Vendredi 10 octobre 1997

Corrigé de logique et calculabilité

1 Logique

1. Soit $\mathcal{F}^2$ l'ensemble des formules propositionnelles sur n atomes $\{p_1, ..., p_n\}$ s'écrivant comme une conjonction de clauses à 2 littéraux :

$$\bigwedge_{i=1}^{k} (l_i^1 \vee l_i^2)$$

avec $l_i^1 = p_j$ ou $l_i^1 = \neg p_j$ (idem pour l_i^2).

(a) Pour écrire une clause $(l_i^1 \vee l_i^2)$ sur n variables, il y a $2n$ choix pour l_i^1 et $2n$ choix pour l_i^2, ce qui fait $4n^2$ clauses. La disjonction étant commutative ($a \vee b \equiv b \vee a$), il y a donc $2n^2$ clauses à deux littéraux distinctes.

(b) On remarque qu'une résolution entre deux clauses à deux littéraux produit une clause également à deux littéraux. D'où l'algorithme consistant à appliquer le principe de résolution jusqu'à
– obtenir la clause vide (la formule est insatisfiable) ;
– ne pas pouvoir produire une clause déjà existante (la formule est satisfiable).
Toutes les clauses manipulées ont deux littéraux donc il y en a moins de $2n^2$ donc l'algorithme est de complexité polynômiale.

2. On peut axiomatiser en calcul des propositions avec le codage suivant :

e le membre est écossais ;

r le membre porte des chaussettes rouges ;

m le membre est marié ;

d le membre sort le dimanche ;

k le membre porte un kilt ;

pour lequel les règles du club s'écrivent :

$$\neg e \rightarrow r \tag{1}$$
$$k \vee \neg r \tag{2}$$
$$m \rightarrow \neg d \tag{3}$$
$$d \leftrightarrow e$$
$$\equiv \quad (d \rightarrow e) \tag{4}$$
$$\wedge \quad (e \rightarrow d) \tag{5}$$
$$k \rightarrow m \wedge e$$
$$\equiv \quad (k \rightarrow m) \tag{6}$$
$$\wedge \quad (k \rightarrow e) \tag{7}$$
$$e \rightarrow k \tag{8}$$

Le club ne possède pas de membres si et seulement si l'ensemble des règles est contradictoire, en particulier si on peut obtenir la clause vide par résolution :

$$\begin{aligned} 1,2 &\rightsquigarrow e \vee k && (9)\\ 9,8 &\rightsquigarrow k && (10)\\ 10,7 &\rightsquigarrow e && (11)\\ 10,6 &\rightsquigarrow m && (12)\\ 12,3 &\rightsquigarrow \neg d && (13)\\ 13,5 &\rightsquigarrow \neg e && (14)\\ 14,11 &\rightsquigarrow \square && (15) \end{aligned}$$

3. (a) Soient les constantes a, b et c pour les objets, les prédicats
   - $v(x)$ pour « x est vert » ;
   - $b(x)$ pour « x est bleu » ;
   - $s(x, y)$ pour « x est posé sur y » ;

   Les faits sont exprimés par les clauses suivantes :

$$\begin{aligned} b(c) && (16)\\ v(a) && (17)\\ s(b, c) && (18)\\ s(a, b) && (19)\\ \forall x\, b(x) \rightarrow \neg v(x) && (20) \end{aligned}$$

   (b) Pour montrer $\exists x \exists y\, (s(x, y) \wedge v(x) \wedge \neg v(y))$, on procède par résolution en réfutant l'ensemble des hypothèse et la négation de la conclusion :

$$\forall x \forall y\, \neg s(x, y) \vee \neg v(x) \vee v(y) \qquad (21)$$

   D'où la suite de résolutions :

$$\begin{aligned} 21[x \leftarrow b, y \leftarrow c], 18 &\rightsquigarrow \neg v(b) \vee v(c) && (22)\\ 22, 20[x \leftarrow c] &\rightsquigarrow \neg v(b) \vee \neg b(c) && (23)\\ 23, 16 &\rightsquigarrow \neg v(b) && (24)\\ 24, 21[y \leftarrow b] &\rightsquigarrow \neg s(x, b) \vee \neg v(x) && (25)\\ 25[x \leftarrow a], 19 &\rightsquigarrow \neg v(a) && (26)\\ 26, 17 &\rightsquigarrow \square && (27) \end{aligned}$$

   Donc on a montré que

$$16.17, 18, 19, 20 \models \exists x \exists y\, (s(x, y) \wedge v(x) \wedge \neg v(y))$$

   (c) La démonstration n'a pas exhibé les deux objets x et y de la conclusion (ils ont été substitués respectivement par b et c puis a et b). On qualifie la démonstration de non *constructive*.

4. (a) Il n'est pas possible de produire un arbre de preuve dont la racine est $\forall x\,(p(x) \vee \neg p(x))$ donc ce n'est pas une tautologie. En effet, il faudrait nécessairement appliquer la règle $\forall_D$ puis $\vee_D^1$ ou $\vee_D^2$. Dans les deux cas on arrive à un « cul de sac ».

   (b) La formule

$$v(b) \vee \neg v(b)$$

   est nécessaire à la démonstration. Elle correspond aux deux « cas » à examiner.

2 Calculabilité

1. Soit f fonction décroissante dans $\mathbb{N}$. On remarque que l'ensemble $\{n | f(n+1) \neq f(n)\}$ est fini puisque la fonction est décroissante est minorée. Notons le $\{n_1, n_2, ..., n_k\}$. Alors f est définie par :

$$f(n) = \begin{cases} f(0) & \text{si } n \leq n_1 \\ f(n_i) & \text{si } n_{i-1} < n \leq n_i \\ f(n_k + 1) & \text{si } n > n_k \end{cases}$$

   Cette description est finie, donc implémentable en C (par exemple). Donc f est calculable.

2. On remarque le le nombre de configurations possibles pour une machine de Turing à bande finie est fini : pour une machine à z états et n cases sur un alphabet à k symboles, il y a

$$\underbrace{z}_{\text{état courant}} \times \underbrace{k^n}_{\text{bande}} \times \underbrace{n}_{\text{tête}}$$

   configurations possibles.

   Pour savoir si la machine va s'arrêter, il suffit donc de la faire « marcher » est de mémoriser toutes les configurations atteintes. Soit la machine atteint un état terminal (elle s'arrête), soit elle boucle en retrouvant une configuration déjà vue. Donc le problème de l'arrêt est décidable dans ce cas.