Licence STS, semestre 4 — 2015–16
Mathématiques pour l'Informatique 2 — 5 avril 2016
`http://www.lri.fr/~paulin/MathInfo2`

TD8 - Ordres bien fondés, treillis, calcul booléen

Exercice 1 *Ordre et induction bien fondés*

Soient A et B des ensembles, B muni d'une relation d'ordre $\leq_B$ et $<_B$ l'ordre strict associé. Soit f une application de A dans B. On définit une relation sur A par $x <_A y \stackrel{\text{def}}{=} f(x) <_B f(y)$.

1. Montrer que $x <_A y$ est un ordre strict.
2. Montrer que si $<_B$ est un ordre bien fondé alors il en est de même de $x <_A y$.
3. On se donne un ensemble ordonné total $(A, \leq)$. On étudie une fonction de tri sur les mots de A^* qui s'inspire de l'algorithme quicksort. On se donne une fonction `sep`(a,l) qui renvoie deux mots (l_1, l_2) avec l_1 qui ne contient que des lettres b de l telle que $b \leq a$ et l_2 qui ne contient que des lettres b de l telle que $a < b$.
   (a) Donner des équations récursives structurelles sur le mot l qui définissent la fonction `sep`.
   (b) On veut définir une fonction `tri` qui vérifie les équations suivantes avec $a \in A$ et $l \in A^*$.

$$\texttt{tri}(\epsilon) = \epsilon \qquad \texttt{tri}(al) = \texttt{tri}(l_1)a\texttt{tri}(l_2) \quad \text{si } \texttt{sep}(a, l) = (l_1, l_2)$$

   Donner des règles d'inférence pour définir la relation `Tri` binaire correspondant aux équations (c'est-à-dire que $\texttt{Tri}(l, m)$ représente le fait que $\texttt{tri}(l)$ a pour valeur m) et justifier que cette relation est totale (c'est-à-dire que pour tout mot l, il existe un mot m tel que $\texttt{Tri}(l, m)$ est vérifié).

Exercice 2 *Algèbre de Boole*

On considère l'ensemble $N^\omega = \mathbb{N} \cup \{\infty\}$. Montrer comment munir l'ensemble des intervalles $[a, b[$ avec $a \in \mathbb{N}$ et $b \in N^\omega$ d'une structure de treillis borné distributif.

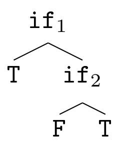
Exercice 3 *Calcul booléen, exam 2014*

On utilise des arbres binaires pour représenter des formules booléennes sur des variables $x_1, \ldots, x_n$. Pour cela on introduit deux constantes `T` et `F` et pour chaque variable x_i, on introduit un constructeur $\texttt{if}_i$ binaire. On appelle `bdt` l'ensemble des termes ainsi construits.

Chaque terme représente une formule booléenne. La constante `T` représente la formule $\top$ (vrai), la constante `F` représente la forume $\perp$ (faux), et si le terme t représente la formule P et le terme u représente la formule Q, alors le terme $\texttt{if}_i(t, u)$ représente la formule $(x_i \wedge P) \vee (\neg x_i \wedge Q)$. On introduit une fonction `val` qui associe une formule booléenne à chaque terme de `bdt` et qui est définie de manière récursive par les équations suivantes :

$$\texttt{val}(\texttt{T}) = \top \qquad \texttt{val}(\texttt{F}) = \perp \qquad \texttt{val}(\texttt{if}_i(t, u)) = (x_i \wedge \texttt{val}(t)) \vee (\neg x_i \wedge \texttt{val}(u))$$

L'arbre ci dessous représente donc la formule $(x_1 \wedge \top) \vee (\neg x_1 \wedge ((x_2 \wedge \perp) \vee (\neg x_2 \wedge \top)))$ (qui est équivalente à $x_1 \vee (\neg x_1 \wedge \neg x_2)$)



1. Construire un arbre qui représente une formule équivalente à x_i, même question pour $\neg x_i$.
2. Montrer que l'arbre $\texttt{if}_i(t, t)$ représente une formule équivalente à la formule représentée par t.
3. Montrer que les formules $\neg((x \wedge P) \vee (\neg x \wedge Q))$ et $(x \wedge \neg P) \vee (\neg x \wedge \neg Q)$ ont la même valeur de vérité quelles que soient les valeurs de x, P et Q.

4. Déduire de la question précédente des équations récursives pour définir une fonction `not` qui transforme un arbre t qui représente la formule P en un arbre $\texttt{not}(t)$ qui représente une formule équivalente à $\neg P$.
5. A quelle condition sur l'arbre reconnait-on que la formule associée est une tautologie ? Donner les équations récursives pour définir une fonction `tauto` qui prend en argument un arbre t et renvoie `vrai` si la formule associée à t est une tautologie et `faux` sinon.

Exercice 4 *Ordre sur les mots*

Soit $\mathbb{B}$ l'ensemble des booléens $\{0,1\}$. On note $\mathbb{B}^n$ l'ensemble des mots de longueur n sur l'alphabet $\mathbb{B}$ et $\mathbb{B}^*$ l'ensemble des mots finis (de longueur quelconque).

On introduit une relation binaire $\prec$ sur $\mathbb{B}^*$ par le système d'inférence suivant ($x \in \mathbb{B}$ et $m, m_1, m_2 \in \mathbb{B}^*$) :

$$\frac{}{\epsilon \prec xm} \qquad \frac{}{0m_1 \prec 1m_2} \qquad \frac{m_1 \prec m_2}{xm_1 \prec xm_2}$$

On admettra sans le démontrer que cette relation est un ordre strict sur les mots.

1. Montrer que $100 \prec 11$.
2. Comparer les mots 0000, 00 et 1.
3. Définir par des équations récursives une fonction $\texttt{test} \in \mathbb{B}^* \times \mathbb{B}^* \to \mathbb{B}$ telle que $\texttt{test}(m_1, m_2)$ est vrai exactement lorsque $m_1 \prec m_2$. On pourra introduire des équations pour les quatre cas :

$$\texttt{test}(\epsilon,\epsilon) = \dots \quad \texttt{test}(\epsilon, xm) = \dots \quad \texttt{test}(xm,\epsilon) = \dots \quad \texttt{test}(xm_1, ym_2) = \dots$$

4. Montrer que si $\texttt{test}(m_1, m_2)$ est faux alors soit $m_1 = m_2$, soit $m_2 \prec m_1$, (ce qui implique que la relation $\prec$ est un ordre total).
5. Si $x \in \mathbb{B}$, on note x^n le mot de longueur n qui ne contient que des x. On peut définir ce mot par des équations récursives sur n :

$$x^0 = \epsilon \qquad x^{n+1} = x(x^n)$$

   Montrer par récurrence sur n les deux propriétés suivantes :
   (a) $\forall n \in \mathbb{N}, x^n \prec x^{n+1}$
   (b) $\forall n \in \mathbb{N}, 0^{n+1}1 \prec 0^n 1$
6. La relation $\prec$ est-elle un ordre bien fondé ? justifier votre réponse.