TD10 - Ordres, treillis, manipulation de formules

Exercice 1 *Ordre et équivalence.*
On se donne un ensemble A et une relation R sur A qui est réflexive et transitive. On introduit la relation E définie par $E(x,y) \equiv R(x,y) \wedge R(y,x)$

1. Montrer que E est une relation d'équivalence. On appelle $A_{/E}$ l'ensemble des classes d'équivalence de A pour la relation E. Montrer que pour tout $X \in A_{/E}$ on a $\forall x\, y \in X,\ R(x,y)$.
2. Soit $A = \{a,b,c,d\}$ et R la relation telle que $R(a,a)$, $R(b,b)$, $R(c,c)$, $R(d,d)$, $R(a,b)$, $R(b,a)$, $R(b,c)$, $R(a,c)$. Définir la relation E et donner les classes d'équivalence correspondantes.
3. On introduit la relation R_E sur $A_{/E}$ par $R_E(X,Y) \equiv \exists x \in X,\ \exists y \in Y,\ R(x,y)$.
   (a) Montrer que $R_E(X,Y) \Leftrightarrow \forall x \in X,\ \forall y \in Y,\ R(x,y)$.
   (b) Montrer que R_E est une relation d'ordre sur $A_{/E}$.
   (c) Construire la relation R_E sur l'exemple de relation de la question précédente.

Correction :

1. *Montrons que E est une relation d'équivalence.*
   — *Réflexivité : Pour tout $x \in A$, on a $R(x,x)$, donc $E(x,x)$.*
   — *Symétrie : Soient $x,y \in A$. On suppose $E(x,y)$, montrons que $E(y,x)$. On a $E(x,y) \Leftrightarrow R(x,y) \wedge R(y,x) \Leftrightarrow E(y,x)$.*
   — *Transitivité : Soient $x,y,z \in A$. On suppose $E(x,y)$ et $E(y,z)$, on montre que $E(x,z)$. On a $E(x,y)$ donc $R(x,y)$ et $R(y,x)$. On a également $E(y,z)$ donc $R(y,z)$ et $R(z,y)$. Par transitivité de R, on a $R(x,z)$ et $R(z,x)$, donc $E(x,z)$.*
   Montrons que pour tout $X \in A/E$, on a $\forall x,y \in X, R(x,y)$. Soit $X \in A/E$. Soient $x,y \in X$, on a $E(x,y)$, donc en particulier $R(x,y)$.
2. *La relation E est la restriction de R à sa partie symétrique, on a donc :*
   $E = \{(a,a),(b,b),(c,c),(d,d),(a,b),(b,a)\}$. Les classes d'équivalence sont donc $\{a,b\}$, $\{c\}$ et $\{d\}$.
3. *On a pour $X,Y \in A/E$, $RE(X,Y) \Leftrightarrow \exists x \in X, \exists y \in Y, R(x,y)$.*
   (a) Montrons que $RE(X,Y) \Leftrightarrow \forall x \in X, \forall y \in Y, R(x,y)$. Soient $X,Y \in A/E$, on suppose $RE(X,Y)$. Par définition de RE on sait qu'il existe x et y telles que $x \in X$, $y \in Y$ et $R(x,y)$. Soient $s \in X$ et $t \in Y$, montrons que $R(s,t)$.
   On a $s,x \in X$ donc $R(s,x)$, on a $y,t \in Y$ donc $R(y,t)$, et on sait que $R(x,y)$, donc par transitivité de R, on a $R(s,t)$. L'autre sens est évident.
   (b) Montrons que RE est une relation d'ordre sur A/E.
   — *Réflexivité : Soit $X \in A/E$, pour tout $x \in X$, $R(x,x)$ donc $RE(X,X)$.*
   — *Anti-symétrie : Soient $X,Y \in A/E$. On suppose que $RE(X,Y)$ et $RE(Y,X)$, montrons que $X = Y$. On a $RE(X,Y)$ donc pour tout $x \in X, y \in Y, R(x,y)$. On a $RE(Y,X)$ donc pour tout $y \in Y, x \in X, R(y,x)$. Donc pour tout $x \in X, y \in Y, R(x,y) \wedge R(y,x)$, d'où $E(x,y)$ et X et Y sont la même classe d'équivalence.*
   — *Transitivité : Soient $X,Y,Z \in A/E$. On suppose que $RE(X,Y)$ et $RE(Y,Z)$. Montrons que $RE(X,Z)$. On a $RE(X,Y)$ donc pour tout $x \in X, y \in Y, R(x,y)$. On a $RE(Y,Z)$ donc pour tout $y \in Y, z \in Z, R(y,z)$. Donc pour tout $x \in X, z \in Z, R(x,z)$ par transitivité de R, d'où $RE(X,Z)$.*

(c) *Sur l'exemple précédent, on a* $RE(\{a,b\},\{a,b\})$*,* $RE(\{c\},\{c\})$*,* $RE(\{d\},\{d\})$ *et* $RE(\{a,b\},\{c\})$*.*

Exercice 2 *Arbres de décision*

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

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

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

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

```
   if1
  /   \
 T    if2
     /   \
    F     T
```

1. Donner des arbres représentant des formules équivalentes respectivement à x_i et à $\neg x_i$.
2. Montrer que l'arbre $\mathtt{if}_i(t,t)$ 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 $\mathtt{not}$ qui transforme un arbre t qui représente la formule P en un arbre $\mathtt{not}(t)$ 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 $\mathtt{tauto}$ qui prend en argument un arbre t et renvoie $\mathtt{vrai}$ si la formule associée à t est une tautologie et $\mathtt{faux}$ sinon.

Correction :

1. *Pour* x_i *:* $if_i(T,F)$*, et pour* $\neg x_i$ *:* $if_i(F,T)$*.*
2. *Formule représentée par l'arbre :* $t' \equiv (x_i \wedge t) \vee (\neg x_i \wedge t)$*. Supposons qu'une interprétation valide* t*, alors quelle que soit la valeur associée à* x_i *elle valide aussi* t'*. Supposons qu'une interprétation valide* t'*, alors elle valide au moins l'une des deux formules* $x_i \wedge t$*,* $\neg x_i \wedge t$*, et donc dans tous les cas elle valide* t*.*
3. *Faire les tables de vérité.*
4. $$\begin{array}{rcl} not(T) &=& F \\ not(F) &=& T \\ not(if_i(t_1,t_2)) &=& if_i(not(t_1), not(t_2)) \end{array}$$

5. *Une tautologie est une formule vraie quelles que soient les valeurs données aux variables propositionnelles. Un critère simple pour cela est d'avoir toutes les feuilles égales à* T.

$$\begin{array}{rcl} tauto(T) & = & true \\ tauto(F) & = & false \\ tauto(if_i(t_1,t_2)) & = & \begin{cases} true & \text{si}\, tauto(t_1) = tauto(t_2) = true \\ false & \text{sinon} \end{cases} \end{array}$$

On peut cependant avoir des cas plus compliqués, par exemple avec l'arbre $if_1(T, if_1(F,T))$. *Le critère simple devient complet si on a des contraintes de forme sur l'arbre, par exemple en demandant que chaque variable n'apparaissent qu'une fois au plus sur tout chemin de la racine à une feuille (ou a fortiori avec le critère usuel des arbres de décision, qui demande que les variables apparaissent en ordre strictement croissant sur ces chemins).*

Exercice 3 *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.
Correction : *L'ensemble N^ω est un treillis borné distributif avec les opérations min pour la borne inférieure et max pour la borne supérieur, un minimum qui est 0 et un maximum ∞ (ce n'est pas une algèbre de Boole car il n'y a pas d'opération de complément). Lorsque $a \geq b$, l'intervalle $[a,b[$ est l'ensemble vide.*

1. $[a_1,b_1[\sqcap[a_2,b_2[=[max(a_1,a_2),min(b_1,b_2)]$ *cette opération correspond à l'intersection des intervalles.*
2. $[a_1,b_1[\sqcup[a_2,b_2[=[min(a_1,a_2),max(b_1,b_2)]$ *cette opération n'est pas l'union, car l'union de deux intervalles n'est pas forcément un intervalle. Cette opération correspond au plus petit intervalle qui contient l'union.*
3. $\bot = [0,0[$ *et* $\top = [0,\infty[$
4. $[a_1,b_1[\sqcap([a_2,b_2[\sqcup[a_3,b_3[) = [max(a_1,min(a_2,a_3)),min(b_1,max(b_2,b_3)[$
$= [min(max(a_1,a_2),max(a_1,a_3)),max(min(b_1,b_2),min(b_1,b_3))[$
$= [max(a_1,a_2),min(b_1,b_2)[\sqcup[max(a_1,a_3),min(b_1,b_3)[$
$= ([a_1,b_1[\sqcap[a_2,b_2[)\sqcup([a_1,b_1[\sqcap[a_3,b_3[)$

Le complément d'un intervalle n'est pas un intervalle, ce n'est donc pas une algèbre de Boole. Pour avoir une algèbre de Boole, il faut considérer les réunions finies d'intervalles.