TD 1 – Syntaxe de la logique des propositions

Exercice 1 – Pour chacune des formules suivantes construites sur l'ensemble de symboles propositionnels S={p,q,r,s}, les connecteurs {¬,∧,∨,→,↔}, les constantes {⊥,⊤} et les parenthèses {(,)}, vous donnerez :

a) son statut vis-à-vis de la logique des propositions : sont-elles ou non des propositions (i.e. des formules bien formées du langage propositionnel) ?
b) l'arborescence associée à la formule (si la formule n'est pas bien formée vous proposerez une correction) ;
c) une expression fonctionnelle de la formule ;
d) une expression infixée débarrassée des parenthèses superflues en considérant les priorités entre connecteurs vues en cours :

¬((q→r)∨p)	(((p∧q)¬r)→p)
((¬p→(¬r∨q))→p)	(¬¬(q→r)∨(¬q→¬r))
(p∨(¬q∧(r∧s)))	(⊥∧(¬p∨⊤))

Exercice 2 - Donner tous les arborescences pouvant correspondre aux formules suivantes sans tenir compte des conventions de priorité entre connecteurs, puis préciser quel arborescence correspond aux conventions d'écriture infixée données dans le cours :

p∧q∨r   p∧¬q   ¬p∧q   ¬p→¬q∧r   p→q→r→s

Rem : dans la suite on supposera que l'on tient compte des conventions de priorité du cours.

Exercice 3 – Que faudrait-il modifier dans la définition inductive du langage des propositions pour prendre en compte un connecteur binaire « ou exclusif » noté ⊕ ? Même question avec un connecteur binaire « non et » noté *nand* et avec un connecteur ternaire « Si Alors Sinon » noté *?;* (exemple : *(p?q;r)*).

Exercice 4 – Soit S l'ensemble des symboles propositionnels et *PROP(S)* l'ensemble des fbf construites à partir de S et soit A et B les fbf suivantes :

A = (¬p→(p → (r ∨ p)))
B = (((p ∧ ⊥) ∨ (¬r → p)) ↔ r)

a) Dessiner les arborescences associées à A et B
b) Calculer les ensembles *SP(A)* et *SP(B)* où SP est l'application vue en cours qui définit l'ensemble des symboles propositionnels d'une formule.
c) Soit l'application *Prof* de *PROP(S)* dans l'ensemble des entiers naturels qui, à toute fbf *P*, associe la profondeur de l'arborescence syntaxique associée à *P* (c'est-à-dire le nombre maximum d'arêtes d'un chemin dans cette arborescence). Donner une définition par induction structurelle de l'application Prof.
d) Calculer les entiers *Prof(A)* et *Prof(B)*

Exercice 5 – Une sous-formule d'une fbf A est une fbf B qu'il est nécessaire de produire lors de la construction de A par le processus d'induction.

a) Dites si les formules suivantes sont des sous-formules de la formule ((¬p→q∧r)↔((s→¬p∧r)↔¬q∧s)) :
   ¬p, p→q, p→q∧r, ¬p→q, ¬p→q∧r, ¬p∧r, (s→¬p∧r)↔¬q∧s
b) Quel est l'ensemble des sous-formules de la formule ((¬p∧r) →¬p)
c) Donnez une définition par induction de l'application *Sub* qui associe à une proposition l'ensemble des sous-formules d'une formule.

Exercice 6 – Soit l'application *Sub* précédente et l'application *nbc* de *PROP* dans $\mathbb{N}$ qui, à toute fbf *P*, associe le nombre de connecteurs de *P*, et soit la fbf A = ((p ∧ q) → (¬r ∨ p)).

a) Donner *Sub(A)* et *nbc(A)*, puis vérifier que $|Sub(A)| \leq 2.nbc(A) + 1$.
b) Donner une définition par induction de l'application *nbc*.
c) Montrer par induction structurelle qu'une fbf *P* ayant *n* occurrences de connecteurs a au plus *2n + 1* sous-fbf (autrement dit, que pour toute fbf *P*, $|Sub(P)| \leq 2.nbc(P) + 1$).