Construcción Formal de Programas en Teoría de Tipos
Primer Parcial – Mayo de 2024

Problema 1. Pruebe el siguiente lema usando lógica clásica:

Variables P R : Prop.
Lemma lema1_1 : (P -> R) \/ (R -> P).

Problema 2. Considere el siguiente problema: Un zoológico tiene las siguientes reglas para sus animales:

Regla 1: Los animales cuadrúpedos no nadan.
Regla 2: Los animales herbívoros vuelan.
Regla 3: Los animales son herbívoros o toman leche.
Regla 4: Los animales vuelan o no toman leche.
Regla 5: Los animales que vuelan son herbívoros y cuadrúpedos.
Regla 6: Los animales nadan si y sólo si son herbívoros.

Demuestre que las reglas son contradictorias, o sea que ningún animal puede estar en ese zoológico, sin usar tácticas automáticas ni lógica clásica.

Problema 3. Considere:

Variable C : Set.
Variables T U : C -> C -> Prop.

Demuestre sin usar tácticas automáticas ni lógica clásica:

Lemma L2: (exists x1:C, forall x2:C, U x1 x2 -> ~T x1 x2) -> forall x3:C, exists x4:C, T x4 x3 -> ~U x4 x3.

Problema 4. Considere las siguientes declaraciones:

Variable Bool: Set.
Variable TRUE : Bool.
Variable FALSE : Bool.
Variable Not : Bool -> Bool.
Variable Imp : Bool -> Bool -> Bool.
Variable Xor : Bool -> Bool -> Bool.

Axiom Disc : ~ (FALSE = TRUE).
Axiom BoolVal : forall b : Bool, b = TRUE \/ b = FALSE.
Axiom NotTrue : Not TRUE = FALSE.
Axiom NotFalse : Not FALSE = TRUE.
Axiom ImpFalse : forall b : Bool, Imp FALSE b = TRUE.
Axiom ImpTrue : forall b : Bool, Imp TRUE b = b.
Axiom XorTrue : forall b : Bool, Xor TRUE b = Not b.
Axiom XorFalse : forall b : Bool, Xor FALSE b = b.

Bajo el contexto previo, demuestre en Coq los siguientes lemas:

a) *Lemma L41 : forall b: Bool, Xor b b = FALSE.*
b) *Lemma L42: forall b1 b2: Bool, Imp b1 b2 = FALSE -> b1 = TRUE /\ b2 = FALSE.*
c) *Lemma L43: forall b1 b2: Bool, Xor b1 b2 = TRUE -> Imp b1 (Not b2) = TRUE.*

Problema 5. Considere el constructor de tipos *ABnat* de árboles binarios no vacíos de *n* nodos, de números naturales: `Parameter ABnat : forall n : nat, Set.`
a) Defina el tipo del operador *null* que permita representar un árbol binario de naturales vacío (sin nodos).
b) Defina el tipo del operador *add* que permita generar árboles binarios no vacíos de naturales. Este operador junto con *null* deberían permitir crear cualquier árbol de naturales de *n* nodos.
c) Construya utilizando los operadores *null* y *add* un árbol binario de naturales que tenga 3 nodos y sea de altura 3, cuyos elementos sean 5, 6 y 7, donde 5 esté en la raíz y 7 sea el de mayor profundidad en el árbol.
d) Generalice las definiciones *ABnat*, *null* y *add* para trabajar con elementos de un tipo genérico (en vez de *nat*).
NOTA: en todo el problema use el tipo nat predefinido de Coq con O y +, como el cero y la suma de naturales.