

# Construcción Formal de Programas en Teoría de Tipos

## Primer Parcial - Setiembre de 2023

**Problema 1.** Formalice y demuestre la validez del siguiente razonamiento en Coq, sin usar tácticas automáticas ni lógica clásica:

**Hipótesis 1:** Si una base de datos está normalizada entonces no presenta redundancia de información.

**Hipótesis 2:** Una base de datos es consistente o no es útil.

**Tesis:** Si una base de datos no es consistente o presenta redundancia de información, entonces la base de datos no está normalizada o no es útil

**Problema 2.** Sean T y U predicados binarios sobre elementos de un conjunto C. Pruebe en Coq sin usar tácticas automáticas ni lógica clásica el siguiente lema:

$$(\forall x y.(U(x, y) \rightarrow \sim T(x, y)) \wedge \exists z.T(z, z)) \rightarrow \exists w.\sim U(w, w)$$

**Problema 3.** Considere las siguientes declaraciones:

Section Problema3.

```
Variable Bool: Set.
Variable TRUE : Bool.
Variable FALSE : Bool.
Variable Not : Bool -> Bool.
Variable Or : Bool -> Bool -> Bool.
Variable And : Bool -> Bool -> Bool.

Axiom BoolVal : forall b : Bool, b = TRUE \/ b = FALSE.
Axiom NotTrue : Not TRUE = FALSE.
Axiom NotFalse : Not FALSE = TRUE.
Axiom AndTrue : forall b : Bool, And TRUE b = b.
Axiom AndFalse : forall b : Bool, And FALSE b = FALSE.
Axiom OrTrue : forall b : Bool, Or TRUE b = TRUE.
Axiom OrFalse : forall b : Bool, Or FALSE b = b.
```

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

```
Lemma lema3_1 : forall b : Bool, Not (Not b) = b.
Lemma lema3_2 : forall b1 b2 : Bool, Not (Or b1 b2) = And (Not b1) (Not b2).
Lemma lema3_3 : forall b1 : Bool, exists b2 : Bool, And b1 b2 = Or b1 b2.
```

End Problema3.

**Problema 4.** Considere el constructor de tipos `Array`, paramétrico en el largo y de elementos de un tipo genérico:

```
Parameter Array : Set -> nat -> Set.
```

- Declare el tipo del operador `empty` que permita representar un array vacío de elementos de un tipo genérico.
- Declare el tipo del operador `add` que permita generar arreglos no vacíos de elementos de un tipo genérico, donde se agregan dos elementos a un arreglo dado. Mediante los operadores `empty` y `add` se podrían generar arreglos de largo par.
- Construya utilizando los operadores `empty` y `add` un arreglo de enteros de largo 6, que contenga los números del 1 al 6 (en algún orden).
- ¿Qué operador se podría agregar para que junto con el operador `add` definido en la parte c) se pudieran generar arreglos de elementos de un tipo genérico de largo impar (solo usando `add` y el operador que se pide agregar). Defínalo y construya luego usando este operador y `add` un arreglo de enteros de largo 3, con elementos del 7 al 9 (en algún orden).

**Problema 5.** Considere las declaraciones:

```
Variable S : Set.
Variable e1 e2 : S.
Variable P1 P2 : S -> S -> Prop.
Variable P3 P4 : S -> Prop.
```

a) Complete la siguiente prueba en Coq en un sólo renglón utilizando únicamente la táctica `exact` con el término asociado.

```
Lemma lema5_1 : (forall x y : S, P1 x y -> P2 y x) -> (P1 e1 e2) -> P2 e2 e1.
```

b) Demuestre el siguiente lema en un sólo renglón utilizando compositores (estructuradores) de tácticas:  $((\forall x \in S. P3(x)) \vee (\forall y \in S. P4(y))) \rightarrow (\forall z \in S. (P3(z) \vee P4(z)))$ , donde P3 y P4 son predicados unarios sobre S.