



Lógica Computacional, 2017-2

Práctica 7: Introducción a Coq

Víctor Zamora Gutiérrez

Manuel Soto Romero

Fecha de inicio: 2 de mayo de 2017

Fecha de término: 12 de mayo de 2017



Instrucciones generales

- Para tener derecho a calificación, la práctica debe ejecutarse sin errores ni advertencias. No está permitido usar las tácticas `intuition`, `auto`, `tauto`, `omega` u otra táctica que dependa directamente o indirectamente de `auto`. Tampoco está permitido importar bibliotecas externas.
- La entrega es por **equipos de 3 a 4 integrantes**. Seguir los lineamientos especificados en: <http://sites.ciencias.unam.mx/logica-computacional-2017-2/laboratorio/lineamientos>.

Ejercicios

Dadas las definiciones de `stack`, `pop`, `top`, `concatenate` y `reverse` anexos a este archivo PDF, demostrar las siguientes proposiciones:

1. `pop(pop(pop(empty))) = empty`
2. Para todo natural `x`, `pop(push x (pop (push x empty))) = empty`.
3. Si un `stack` tiene algo en el tope, entonces no es vacío:
`forall (s:stack) (x:nat) , top s = Some x -> s <> empty`.
4. Las “operaciones” `push` y `pop` son inversas:
`forall (s:stack) (x:nat), pop (push x s) = s`.
Nota: Tomen en cuenta que en realidad, `pop` es un constructor de `stack` y `push` es una función.
5. El `stack` vacío es el elemento neutro por la derecha en la concatenación:
`forall s:stack, s ++ empty = s`.
6. La reversa de un `stack` que contiene un solo elemento es sí mismo:
`forall n:nat, reverse(push n empty) = push n empty`.
7. La concatenación es asociativa
`forall s1 s2 s3:stack, (s1 ++ s2) ++ s3 = s1 ++ (s2 ++ s3)`.
8. La reversa de la concatenación de dos `stacks` es equivalente a sacar la reversa de ambos y concatenarlos en orden inverso:
`forall s1 s2:stack, reverse (concatenate s1 s2) = concatenate (reverse s2) (reverse s1)`.

By using the insignificant item, you had a very fruitful experience that cannot be understood by someone who does not use something insignificant.