Verificación formal de arboles rojinegros en Haskell con Coq

David Felipe Hernández Chiapa

Facultad de Ciencias
Universidad Nacional Autónoma de México

28 de noviembre de 2018

1. Verificación formal en Haskell

Haskell es un lenguaje con una base muy grande de desarrolladores que constantemente están generando mas programas escritos en el.

1. Verificación formal en Haskell

Una de las cosas que se dice de Haskell es que la verificación de su código es bastante sencilla.

1. Verificación formal en Haskell

¿Pero que tan cierto y escalable es esto?

2. Verificación formal en Coq.

A diferencia de Haskell, Coq es un asistente de pruebas, con el cual tu puedes escribir un programa en el y después verificarlo formalmente.

2. Verificación formal en Coq.

Una de las diferencias mas grandes entre la escritura de programas entre Haskell y Coq, es que Coq solo acepta funciones totales.

3. Problema.

Nos gustaría una manera de traducir módulos de Haskell con funciones totales a Coq para poder verificarlas formalmente de una manera mas sencilla, ordenada y escalable.

4. hs-to-coq

Es una herramienta en desarrollo por un equipo de la Universidad de Pensilvania.
En esta herramienta ya existen bibliotecas de Haskell traducidas a Coq y tambien te da la facilidad de traducir tus propios programas de Haskell.

4. hs-to-coq

Esta herrmienta es creada para facilitar la verificación, siguiendo los siguientes pasos:

4. hs-to-coq

1. Escribir un modulo de Haskell, digamos un modulo de Arboles Rojinegros.

4. hs-to-coq

1. Escribir un modulo de Haskell, digamos un modulo de Arboles Rojinegros.
2. Probar ese codigo en Haskell, generar ejemplos.

4. hs-to-coq

1. Escribir un modulo de Haskell, digamos un modulo de Arboles Rojinegros.
2. Probar ese codigo en Haskell, generar ejemplos.
3. Utilizar hs-to-coq para traducir el codigo a Coq.

4. hs-to-coq

1. Escribir un modulo de Haskell, digamos un modulo de Arboles Rojinegros.
2. Probar ese codigo en Haskell, generar ejemplos.
3. Utilizar hs-to-coq para traducir el codigo a Coq.
4. ¡A verificar!

4. hs-to-coq

Esto simplifica mucho la verificación en varios frentes:

4. hs-to-coq

- La traducción no se hace a mano.

4. hs-to-coq

- La traducción no se hace a mano.
- La cooperación en un equipo de trabajo se hace mas sencilla.

5. Ejemplos

404 not found