

Autores/Estudiantes:

Lucas Godoy

Lic. Ciencias de la Computación

Rodrigo Mancera

Lic. Ciencias de la Computación

Miguel Maldonado

Lic. Ciencias de la Computación

Introducción

Este artículo refleja la idea de lo que un grupo de estudiantes de Ciencias de la Computación y entusiastas por la programación realiza para solidificar los conceptos en la escritura correcta de programas y con el fin de poner en práctica lo aprendido en el transcurso de la carrera. Formularemos problemas, los resolveremos y expondremos su solución, introduciendonos en el lenguaje de programación Python conforme avance.

1. Derivación de Programas Funcionales

El objetivo es poner en práctica y ejemplificar el tema, utilizando las herramientas de cálculo proposicional, cálculo de predicados, expresiones cuantificadas, su formalización y técnicas elementales de la programación para construcción de programas.

1.1. Ejercicio Propuesto

Dada una lista de números, se debe determinar si algún elemento de la lista es divisible por 2.

1. **Formalización.** Utilizando las herramientas que nos brinda el cálculo de predicados podemos escribir lo siguiente:

$$\langle \exists i : 0 \leq i < \#xs : xs.i \bmod 2 \rangle$$

2. **Análisis de Tipado.** Utilizaremos por convención Haskell para la tipificación.

$$div2.xs :: [int] \rightarrow Bool$$

3. **Definición de la Función.** Definición recursiva de la función.

$$\begin{aligned} \text{div2}[\] &\doteq \text{False} \\ \text{div2}(x \triangleright xs) &\doteq x \bmod 2 = 0 \vee \text{div2}.xs \end{aligned}$$

4. **Hipótesis..**

$$\langle \exists i : 0 \leq i < \#xs : xs.i \bmod 2 = 0 \rangle \equiv \text{div2}.xs$$

5. **Demostración de Caso Base.**

- a) $\langle \exists i : 0 \leq i < \#[\] : [\].i \bmod 2 = 0 \rangle \equiv \text{div2}[\]$
- b) $\langle \exists i : 0 \leq i < 0 : [\].i \bmod 2 = 0 \rangle \equiv \text{div2}[\]$
- c) $\langle \exists i : \text{False} : [\].i \bmod 2 = 0 \rangle \equiv \text{div2}[\]$
- d) $\text{False} \equiv \text{div2}[\]$

6. **Demostración para $x \triangleright xs$**

- a) $\langle \exists i : 0 \leq i < \#(x \triangleright xs) : (x \triangleright xs).i \bmod 2 = 0 \rangle \equiv \text{div2}.(x \triangleright xs)$
- b) $\langle \exists i : 0 = i \vee 1 \leq i < \#(x \triangleright xs) : (x \triangleright xs).i \bmod 2 = 0 \rangle \equiv \text{div2}.(x \triangleright xs)$
- c) $\langle \exists i : 0 = i : (x \triangleright xs).i \bmod 2 = 0 \rangle \vee \langle \exists i : 1 \leq i < \#(x \triangleright xs) : (x \triangleright xs).i \bmod 2 = 0 \rangle \equiv \text{div2}.(x \triangleright xs)$
- d) $\langle \exists i : 0 = i : (x \triangleright xs).i \bmod 2 = 0 \rangle \vee \langle \exists i : 1 \leq i < \#(x \triangleright xs) - 1 : (x \triangleright xs).i + 1 \bmod 2 = 0 \rangle \equiv \text{div2}.(x \triangleright xs)$
- e) $\langle \exists i : 0 = i : (x \triangleright xs).i \bmod 2 = 0 \rangle \vee \langle \exists i : 1 \leq i < \#xs : xs.i \bmod 2 = 0 \rangle \equiv \text{div2}.(x \triangleright xs)$
- f) $(x \triangleright xs).0 \bmod 2 = 0 \vee \text{div2}.xs \equiv \text{div2}.(x \triangleright xs)$
- g) $x \bmod 2 = 0 \vee \text{div2}.xs \equiv \text{div2}.(x \triangleright xs)$

7. **Codificación.** Como habíamos mencionado anteriormente adoptaremos como lenguaje de programación Python. Código fuente.

Bibliografía Consultada.

- 1. **Cálculo de Programas.**
Autores: Javier Blanco, Silvina Smith y Damián Barsotti
- 2. **How to Think Like a Computer Scientist**
Autores: Allen Downey, Jeffrey Elkner y Chris Meyers