Primer trabajo práctico
La Conjetura de Lothar Collatz

Taller de Álgebra 1 - Primer cuatrimestre de 2022

¿Escucharon alguna vez hablar de Lothar Collatz? Se trata de un matemático alemán que nació en 1910 y fue una de las grandes figuras de la disciplina en su país y el mundo en el siglo XX. Destacado alumno, Collatz fue una persona muy inquieta y con una enorme vitalidad, falleció en 1990 cuando se encontraba en un congreso en Varna (Bulgaria) en el que iba a dar una conferencia. Luego de finalizar su doctorado sobre ecuaciones diferenciales lineales en la Universidad de Berlín en 1935, trabajó con las iteraciones sobre los números enteros, que representaba mediante grafos. En 1937 plantea por primera vez un problema matemático que lo iba a ocupar por el resto de su vida.

Para entender este problema necesitamos primero definir la secuencia de Collatz de un número. El primer término es el propio número y la definición general de la secuencia viene dada de manera recursiva: si un término es par, lo dividimos por dos y, si es impar, lo multiplicamos por 3 y le sumamos 1; es decir, que tenemos la siguiente definición:

$$a_{k+1} = \begin{cases} \frac{a_k}{2} & \text{si } a_k \text{ es par} \\ 3a_k + 1 & \text{si } a_k \text{ es impar} \end{cases}$$

Conocido lo que es la secuencia de Collatz de un número, la conjetura enunciada por Lothar Collatz establece lo siguiente:

Conjetura. *Si tomamos un número natural cualquiera, su secuencia de Collatz siempre llega al número 1.*

Por ejemplo, tomemos el número 13, entonces si $a_1 = 13$, obtenemos la siguiente secuencia que llega al 1: $13 \to 40 \to 20 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$ (9 iteraciones o pasos). En la Figura 1 se puede ver la secuencia de Collatz pensada como un grafo de números.

Se han probado ya con varios miles de millones de números, e indefectiblemente siempre acabamos en el 1. Pero, ¿garantiza eso que siempre es así para cualquier número por enorme que sea? Pues en verdad no, y por eso para la comunidad científica, hasta que no haya una demostración formal que lo pruebe o un contraejemplo que lo niegue, esta afirmación hecha por Collatz seguirá siendo una conjetura.

Collatz discutió sobre su conjetura con varios matemáticos de la época y planteó el problema en varias conferencias, la primera de ellas impartida en la Universidad

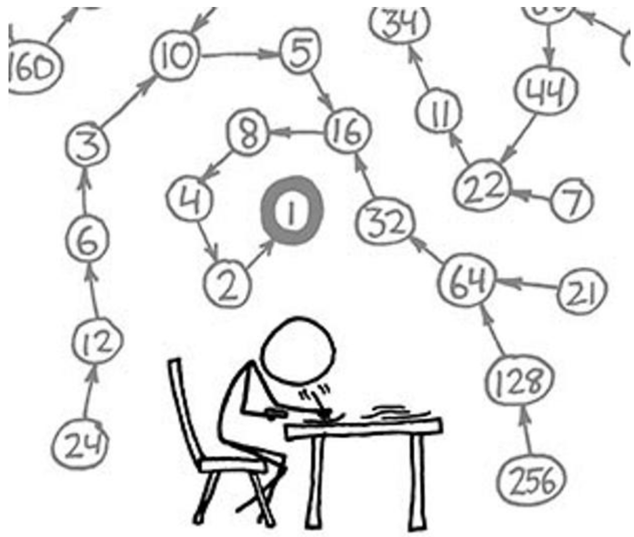

Figura 1: Viñeta del historietista y físico estadounidense Randall Munroe que muestra a Lothar Collatz pensando su conjetura como un grafo de números.

de Siracusa (Nueva York), razón por la que el problema adopta también este nombre, conjetura de Siracusa. Paralelamente, el matemático polaco Stanislaw Ulam (1909-1984), uno de los participantes en el proyecto Manhattan, trabajó sobre ella en el Laboratorio Nacional de Los Álamos en EEUU. En la década de 1960, el problema es asimismo difundido y trabajado por el matemático japonés Shizuo Kakutani (1911-2004) en las Universidades de Yale y Chicago. Kakutani es autor de un teorema de punto fijo que generaliza al de Brouwer, y que permitió la prueba de la existencia del Equilibrio de Nash en teoría de juegos. En esos años sesenta, en plena guerra fría, se corrió el rumor de que la conjetura de Collatz era parte de un complot soviético para frenar otras investigaciones de mayor envergadura de los EE.UU.

Se han propuesto diferentes intentos de demostraciones, entre ellas una famosa de Peter Schorer en 2009, pero por el momento ninguna ha sido aceptada como satisfactoria por la comunidad matemática. En este artículo[1] se describe cómo se encuentra la investigación actualmente, por si alguien se anima (hay un premio en dólares)...

[1] Ren, Wei. "A new approach on proving Collatz conjecture", *Journal of Mathematics*, 2019. Disponible on-line en: `https://www.hindawi.com/journals/jmath/2019/6129836/`

Ejercicios

Se pide resolver en Haskell los siguientes ejercicios:

Ejercicio 1

Escribir la función:

```
satisfaceCollatz :: Integer -> Integer -> Bool
```

que dados dos números naturales n y m devuelve `True` si el número n satisface la conjetura de Collatz en menos de m pasos o iteraciones y `False` en caso contrario. Por ejemplo:

```
*Main> satisfaceCollatz 13 10
True
*Main> satisfaceCollatz 1121 35
False
*Main> satisfaceCollatz 1 2
True
```

Ejercicio 2

Escribir la función:

```
satisfaceCollatzHasta :: Integer -> Integer -> Bool
```

que dados dos naturales n y m devuelve `True` si todos los números naturales desde 1 hasta n satisfacen la conjetura de Collatz en menos de m pasos o iteraciones y `False` en caso contrario. Por ejemplo:

```
*Main> satisfaceCollatzHasta 88 50
False
*Main> satisfaceCollatzHasta 26 100
True
*Main> satisfaceCollatzHasta 1 2
True
```

Ejercicio 3

Escribir la función

```
cantidadTerminosPares :: Integer -> Integer
```

que dado un natural n que satisface la conjectura de Collatz devuelve la cantidad de términos pares que tiene la secuencia de Collatz desde $a_1 = n$ hasta llegar a 1. Por ejemplo:

```
*Main> cantidadTerminosPares 13
7
*Main> cantidadTerminosPares 88
13
*Main> cantidadTerminosPares 1121
31
```

Ejercicio 4

Escribir la función

```
largoSecuencia :: Integer -> Integer
```

que dado un natural n que satisface la conjectura de Collatz devuelve la cantidad de pasos desde $a_1 = n$ hasta llegar a 1. Por ejemplo:

```
*Main> largoSecuencia 13
9
*Main> largoSecuencia 5
5
*Main> largoSecuencia 27
111
*Main> largoSecuencia 1121
44
```

Ejercicio 5

Escribir la función

```
secuenciaMasLargaHasta :: Integer -> Integer
```

que dado un natural n que verifica que para todo $1 \leq k \leq n$ el número k satisface la conjetura de Collatz, devuelve el mínimo valor m, $1 \leq m \leq n$, que genera la secuencia más larga de pasos desde $a_1 = m$ hasta llegar a 1. Por ejemplo:

```
*Main> secuenciaMasLargaHasta 13
9
*Main> secuenciaMasLargaHasta 30
27
*Main> secuenciaMasLargaHasta 88
73
*Main> secuenciaMasLargaHasta 1121
871
```

Condiciones de entrega

El Trabajo Práctico se debe realizar en grupo de 3 estudiantes (ni más ni menos). Deberán previamente completar el formulario *"Inscripción del grupo"* que se encuentra en el campus virtual. Aquellas personas que no hayan podido formar grupo deberán completar el formulario *"No tengo grupo"* y se les asignará un equipo de trabajo. La fecha límite para completar estos formularios es el miércoles 20 de abril a las 23:59 hs. Pasada esta fecha, se considerará que los y las estudiantes que no integran un grupo y no completaron el formulario para informar que no tienen grupo, no van a entregar el Trabajo Práctico en esta instancia y deberán hacerlo en fecha de recuperatorio.

Está prohibido subir el código que implementen a repositorios públicos, así como también usar total o parcialmente soluciones implementadas por otros grupos. No está permitida la interacción con otros grupos para discutir las soluciones de los ejercicios propuestos.

No evaluaremos la eficiencia de los algoritmos que propongan para resolver los ejercicios (y por ende, el tiempo en que tardan en ejecutarse las funciones) pero sí la correctitud del código. Algunas funciones que implementen pueden demorar minutos en arrojar el resultado, no se preocupen por el tiempo sino porque devuelvan el valor correcto.

La entrega consiste en un único archivo `.hs` con las funciones de los ejercicios implementadas, junto con todas las funciones auxiliares que sean necesarias para ejecutarlas. Las funciones deben respetar la signatura (nombres y parámetros) especificados en cada ejercicio, dado que serán testeadas automáticamente. El archivo que entregan tiene que