

Segundo trabajo práctico

Polinomios

Taller de Álgebra 1 - Primer cuatrimestre de 2022

En este trabajo vamos a implementar funciones que trabajan con polinomios con coeficientes en \mathbb{R} . Para esto vamos a definir el tipo polinomio como un renombre de tipos:

`type Polinomio = [Float]`

donde $[a_n, a_{n-1}, \dots, a_1, a_0]$ representa al polinomio $P(X) = \sum_{i=0}^n a_i X^i$.

Por ejemplo el polinomio $4X^3 + 3X - 2$ se representa por `[4, 0, 3, (-2)]`.

El invariante de este tipo es que la lista de coeficientes no empieza con cero:

- Se puede suponer que el invariante se cumple en los polinomios que reciben nuestras funciones.
- Se debe garantizar que el invariante se cumple en los polinomios que devuelven nuestras funciones.

Además definimos un renombre de tipos para monomios en $\mathbb{R}[X]$:

`type Monomio = (Float, Int)`

donde el monomio aX^n se representa con `(a, n)`. El invariante de este tipo es $a \neq 0$.

Ejercicios

Se pide resolver en Haskell los siguientes ejercicios:

Ejercicio 1

Escribir la función:

```
crearPolinomio :: [Float] -> Polinomio
```

que dada una lista cualquiera de números reales, le borra los 0s iniciales si los hubiera (luego el resultado cumple el invariante del tipo Polinomio). Por ejemplo:

```
*Main> crearPolinomio [0,0,(-2),0]
[(-2),0]
*Main> crearPolinomio [(-2),0,3,0]
[(-2),0,3,0]
*Main> crearPolinomio [3,0,0,0]
[3,0,0,0]
```

Ejercicio 2

Escribir la función:

```
grado :: Polinomio -> Int
```

que dada un polinomio retorna su grado. En general se usa la convención $\text{grado}(0) = -\infty$, en la implementación se puede dejar `grado []` indefinido. Por ejemplo:

```
*Main> grado [(-2),0,0,0]
3
*Main> grado [1]
0
*Main> grado [3,7,5]
2
```

Ejercicio 3

Escribir la función:

```
evaluar :: Polinomio -> Float -> Float
```

que dados $P \in \mathbb{R}[X]$ y $a \in \mathbb{R}$ calcula $P(a)$. Por ejemplo:

```
*Main> evaluar [(-2),0,0,0] 3
-54
*Main> evaluar [1] (-5)
1
*Main> evaluar [7,5] 2
19
```

Ejercicio 4

Escribir la función:

```
productoPorMonomio :: Monomio -> Polinomio -> Polinomio
```

que calcula el producto de un monomio por un polinomio. Por ejemplo:

```
*Main> productoPorMonomio (5, 2) [2, 3, (-1)]
[10,15.0,-5.0,0.0,0.0]
```

Ejercicio 5

Escribir la función:

```
producto :: Polinomio -> Polinomio -> Polinomio
```

que calcula el producto de dos polinomios. Por ejemplo:

```
*Main> producto [2, 3, (-1)] [1, (-3)]
[2.0,-3.0,-10.0,3.0]
```

Ejercicio 6

Escribir la función:

```
evaluacionMultiple :: [Float] -> Polinomio -> Polinomio -> [Float]
```

que dada una lista de reales $[a_k, \dots, a_1, a_0]$ y dos polinomios P y Q retorna una lista de reales que contiene en la posición i a $P(a_i)$ si i es par, y a $Q(a_i)$ si i es impar, para $0 \leq i \leq k$. Por ejemplo:

```
*Main> evaluacionMultiple [2, 3, (-1)] [(-2),0,0] [1, (-3)]  
[-8.0,0.0,-2.0]
```

Condiciones de entrega

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 poder compilarse solo, y sin llamar a otro módulo. Les pedimos que utilicen como base para escribir el código el archivo `Tp2.hs` que se encuentra en el campus virtual.

El archivo entregado debe tener la forma “apellido1-apellido2-apellido3.hs”, respetando el orden alfabético de los apellidos. Por ejemplo, si el grupo está formado por los estudiantes María López, Ariel Gómez, y Juan Pérez que cursan en el turno de los viernes debe entregar un archivo llamado “Gomez-Lopez-Perez.hs”.

La entrega se debe llevar a cabo a través del campus virtual, subiendo el archivo con el código con el mecanismo disponible dentro del espacio del taller en el campus virtual. Un solo integrante será el encargado de subir el Trabajo Práctico al campus en representación

de todo el grupo.

Se deberá utilizar exclusivamente los conceptos vistos hasta ahora (semana 8 del cronograma de la materia), inclusive. Se evaluará la corrección de las funciones implementadas, la declaratividad y claridad del código, y que las funciones auxiliares (si las hay) tengan nombres apropiados.

Si tienen dudas o consultas respecto del trabajo práctico, pueden enviar un mail a la lista de docentes algebra1-doc (arroba) dc.uba.ar. No hacer consultas a través de la lista de mails de alumnos.

Además del código, deberán rendir un coloquio grupal, en el que se conversará **individualmente con cada integrante del grupo** sobre el trabajo realizado, debiendo responder diversas preguntas sobre lo que entregaron. El día y horario del coloquio lo acordarán por email con el docente que haya corregido su trabajo.

Fecha de entrega: hasta el domingo 12 de junio a las 23:59 hs.