Proyecto 9 - TAD's Funcionales - Polinomios

Algoritmos y Estructuras de Datos I
Laboratorio

20 de noviembre de 2006

En este proyecto hay que completar la implementación del TAD Poly (implementación de polinomios con grado máximo) y utilizarlo para programar dos funciones sobre polinomios.

La implementación incompleta (archivo `Poly.hs`) se encuentra en la página de la materia. En este archivo se implementan los polinomios con arreglos en haskell. La implementación tiene como restricción que el grado máximo de los polinomios es 10.

El tipo nuevo creado es `Poly a` el cual denota los polinomios con coeficientes de tipo `a`. Como se puede ver este es un tipo polimórfico (como "listas de `a`"). Al ser `a` el tipo de los coeficientes, se pide que pertenezca a la clase `Num`.

Para construir los polinomios se utiliza el constructor de términos:

```
(*^) :: Num a => a -> Int -> Poly a
```

el mismo permite construir los términos separados de un polinomio. Por ejemplo, el término $3*x^2$ se construye como `3 *^2`. A partir de estos términos se puede construir cualquier polinomio utilizando la suma (`Poly a` pertenece a la clase `Num`, ver `Poly.hs`). Por ejemplo, el polinomio $3 * x^5 + x + 2$ puede ser construido simplemente escribiendo `3 *^5 + 1 *^1 + 2`.

En el módulo también están definidas funciones para obtener el grado y los coeficientes de un polinomio.

La única función que hace falta implementar en este TAD es:

```
val :: Num a => Poly a -> a -> a
```

la misma toma un polinomio y un valor y evalúa el mismo en ese valor.

Antes de comenzar el proyecto lea detenidamente el archivo `Poly.hs`. Si no entiende algo, pregunte en la clase del teórico del laboratorio o a los ayudantes del taller. Observe todas las clases a las que pertenece el tipo `Poly a`.

Ejercicios del proyecto:

1. Programe y agregue en el archivo `Poly.hs` la función `val`.

2. En otro archivo, importe el módulo y programe la función

```
mayor :: (Num a, Ord a) => Poly a -> Poly a -> Bool
```

La cual calcula si el primer polinomio es mayor que el segundo. El orden que se debe usar es el usual. O sea un polinomio es mayor que otro si puede encontrarse un valor desde el cual toda valuación del primero es mayor que el segundo. Esto es

$$mayor\,.p1.p2 \;\equiv\; \langle\, \exists\, x \;::\; \langle\, \forall\, y \;:\; y > x :\; p1(y) > p2(y) \,\rangle \,\rangle$$

Hint: Revise los libros de álgebra para ver como se puede computar esto.

3. En el mismo archivo programa la función

```
fstDif :: Poly Integer -> Poly Integer -> Integer
```

la cual compara los dos polinomios (con la función anterior) y calcula el primer valor mayor o igual a cero donde el mayor polinomio es mayor que el otro. En caso que sean iguales devuelve -1. Esto es

$$fstDif\,.p1.p2 = \begin{cases} \langle \textbf{Min}\; x \;:\; x \geq 0 \;\wedge\; p1(x) > p2(x) :\; x \,\rangle & \text{, si } mayor\,.p1.p2 \\ \langle \textbf{Min}\; x \;:\; x \geq 0 \;\wedge\; p1(x) < p2(x) :\; x \,\rangle & \text{, si } mayor\,.p2.p1 \\ -1 & \text{, si } p1 = p2 \end{cases}$$