

Proyecto 2

Recursión y Listas

Algoritmos y Estructuras de Datos II
Laboratorio

3 de septiembre de 2006

Este proyecto tiene dos partes.

1. Productoria.

Derivar en el formalismo básico la función $prod : [Num] \mapsto Num$ especificada como:

$$prod. xs = \langle \prod i : 0 \leq i < \#xs : xs.i \rangle$$

Obtener el resultado de la derivación en el formalismo básico con pattern matching y análisis por casos. Una vez obtenidos ambos resultados, traducirlos a haskell en dos programas.

Ejecutar ambos programas con distintos valores en hugs y comparar los resultados en base a la cantidad de reducciones y uso de memoria que utilicen los programas.

2. Fibonacci.

Programar en haskell la función fibonacci dada en el teórico respetando el tipo:

```
fib :: Integer -> Integer
```

Una vez hecho esto, derivar en el formalismo básico una versión más eficiente de la función utilizando tuplas. A partir de este resultado programar en haskell esta función *gfib* y programar otra que la utilice, respetando el tipo:

```
fib2 :: Integer -> Integer
```

Ejecutar ambos programas con distintos valores en hugs y comparar los resultados en base a la cantidad de reducciones y uso de memoria que utilicen los programas.

Programar todas las funciones utilizando pattern matching.