

Lenguajes de Programación

Auxiliar N^o 1

Profesor: Éric Tanter
Auxiliar: Ismael Figueroa

13 de Agosto de 2010

1. Implemente la función *fibs* que calcule el k-ésimo número de Fibonacci.
2. Implemente la función *rango* que genera una lista con los números enteros dado su valor inicial y final, ambos incluidos. Asuma que el valor inicial es menor o igual al valor final. En caso contrario retorne una lista vacía.
3. Implemente la función *lista-fibs* que retorne la lista de los primeros k números de Fibonacci
4. Recuerde la función *proc* vista en clases. Scheme la incorpora con el nombre *map*. Utilice *map* para reimplementar la función *lista-fibs*
5. Implemente la función *intercalar* que dadas dos listas, retorne una lista con sus valores intercalados, comenzando por la primera lista. Realice el intercalado sólo hasta consumir la lista más pequeña.
6. Implemente la función *suma-lista* que sume los elementos de una lista. Asuma que la lista tendrá sólo números. En caso de lista vacía el valor será cero.
7. Implemente la función *suma-fibs-cuadrado*, *suma-fibs-cubo*, *suma-fibs-mitad* para los k números de Fibonacci. En caso de lista vacía el valor será cero.
8. Recuerde la función *process* vista en clase. Scheme la incorpora con el nombre *foldl* (la l es porque procesa comenzando por la izquierda, asimismo *foldr* comienza por la derecha)¹. Utilice *foldl* para reimplementar las funciones de los puntos 7 y 8
9. Defina *map* utilizando *foldr*. ¿Qué ocurre si utiliza *foldl*?. Hint: la función que se le da como parámetro a *foldr* debe aplicar la función dada a *map*, y además realizar un *cons* de elementos.

¹ Vea http://en.wikipedia.org/wiki/Fold_%28higher-order_function%29 para una buena explicación gráfica