

Paradigmas de Programación - Examen Final

7 de Julio de 2006

Instrucciones: Entregar los ejercicios {1,2,3} y {4,5,6} en dos partes separadas. Los libres deben entregar los ejercicios marcados con Libre. Además, deberán primero aprobar el examen de regulares para que los ejercicios extras sean corregidos.

1. (2.5 puntos) Escriba una función (SwitchMap Xs F G) que tome una lista Xs y dos funciones unarias F y G. Devuelve una lista donde los elementos en posiciones pares son obtenidos de los elementos de Xs en posiciones pares a los cuales se les aplicó F y a los impares se les aplicó G. Por ejemplo, con las definiciones fun (Inc N) N+1 end fun (Dec N) N-1 end la llamada (SwitchMap [1 1 2 2] Inc Dec) devuelve [2 0 3 1]. La función debe ser recursiva a la cola y no se permite usar ninguna función auxiliares.

2. (2.5 puntos)

Usando la función de alto orden (FoldR L F U) definir la función (LCMlist Xs) que calcula el *mínimo común múltiplo* de una lista Xs de números positivos. Se puede asumir que ya está implementada la función (GCD A B) que computa el *máximo común divisor* entre dos números positivos.

3. Libres Se puede definir un árbol binario de la siguiente manera:

```
<BTree> ::= leaf | tree(<Int> <BTree>1 <BTree>2)
```

Implementar la función (Count T) que cuente cuantos nodos interiores tiene el árbol binario T.

Dicho de otra manera, se están contando cuantos enteros almacena el árbol.

Esta función debe ser recursiva a la cola.

4. (2.5 puntos) La función funQ A B ... end calcula iterativamente la suma $\sum_{i=A}^B i$. Puede asumir que inicialmente $A > 0, B > 0$ y $B > A$.

a) Implemente Q de una manera tal que solo use estado implícito para calcular la suma. Una solución analítica no es aceptada.

b) Implemente Q de una manera tal que solo use estado explícito para calcular la suma. Una solución analítica no es aceptada.

5. (2.5 puntos) En la programación orientada a objetos, la definición de una clase se representa con un record. Nuestra representación y nuestra función New no soportan variables estáticas. Modifique la representación de la clase en el ejemplo de la Figura 7.27 y la función New (Figura 7.28) de manera que el atributo val sea estático.

6. (Libres) De una definición "lazy" de (LazyMap Xs F), donde Xs es una lista y F una función unaria. Considere además las dos siguientes funciones:

```
fun (EagerScale Ns M)
  (LazyMap Ns fun ($ N) N*M end)
end
fun lazy (LazyScale Ns M)
  (LazyMap Ns fun ($ N) N*M end)
end
```

Ambas devuelven una lista de números. Cuántos elementos son computados por

```
{EagerScale [1 2 3] 3}
```

y cuantos por

```
{LazyScale [1 2 3] 3}
```