

Guía de Ejercicios

Paradigma Funcional

PREGUNTA 1

Se provee de una implementación parcial del TDA matriz dispersa que permite almacenar sólo aquellos elementos cuyos valores son distintos de cero (en este contexto se interpretan como inexistentes). A saber, su representación consiste en una lista de listas. La lista principal o Matriz contiene N listas que representan las columnas. Cada una de estas listas está compuesta por un número entero seguido múltiples pares. El número entero representa el número de columna C, mientras que los pares sucesivos representan las filas en esa columna. Las filas son expresadas como pares de números enteros donde el primer elemento representa el número de fila F y el segundo elemento corresponde al ítem de la matriz contenido en la posición (C,F). El siguiente ejemplo ilustra la estructura antes descrita:

`'(#columna '(#fila . elemento) ... '(#fila . elemento)) ... (#columna '(#fila . elemento) ... '(#fila . elemento)))`

De esta forma, la matriz:

	1	2	3	4
1	0	0	2	0
2	0	3	8	0
3	0	7	0	0
4	0	0	0	9

Queda expresada como:

`'((2 '(2 . 3) '(3 . 7)) '(3 '(1 . 2) '(2 . 8)) '(4 '(4 . 9)))`

La implementación parcial del TDA cuenta con los constructores `MatrizDispersa`, `Columna` y `Fila` para la construcción de matrices dispersas, columnas y filas respectivamente. De esta forma es posible crear la matriz anterior de la siguiente forma

`(MatrizDispersa (Columna 2 (Fila 2 3) (Fila 3 7)) (Columna 3 (Fila 1 2) (Fila 2 8)) (Columna 4 (Fila 4 9)))`

Los constructores realizan las validaciones correspondientes para respetar la estructura, tipos de datos y evitar duplicidad de filas y columnas dentro de una misma matriz.

Además de constructores se cuenta con funciones de pertenencia `MatrizDispersa?`, `Columna?` y `Fila?` Para verificar que un valor es una matriz dispersa, columna y fila respectivamente.

Su labor es implementar en Scheme los siguientes selectores indicando la firma (signature) de cada función (esto es dominio recorrido) y el tipo de recursión empleada (si aplica).

1. **(getElemento nCol nFil matriz) (3 pts) ;retorna el elemento contenido en la posición (nCol,nFil) de la matriz.**
2. **(getColumna nCol matriz) (3 pts) ;retorna la columna nCol de la matriz**
3. **(getNumCol col) (1 pto) ;retorna el número de la columna col**
4. **(getFila nFila col) (3 pts) ;retorna la fila contenida en la columna col**
5. **(getFilas col) (2 pto) ;retorna las filas donde existen elementos en la columna col**
6. **(getNumFila fil) (1 pto) ;retorna el número de la fila fil**
7. **(getElementoFila fil) (1pto) ;retorna el elemento contenido en la fila fil**

PREGUNTA 2

Utilizando el mismo enunciado anterior, pero esta vez centrado en MODIFICADORES, y además implementar funciones de pertenencia

1. **(setElementoFila fil elemento) (2 pts) ;cambia el elemento contenido en la fila fil**
2. **(setNumFila fil nFil) (2 pts) ;cambia el numero de la fila fil por nFil**
3. **(setNumCol col nCol) (2 pts) ;cambia el numero de la columna col por nCol**

PREGUNTA 3

Se desea implementar el TDA hora. Implementar al menos una función por cada nivel del TDA (a excepción de los selectores que den ser implementados en su totalidad) e indicar la representación escogida. Como función complementaria del TDA puede implementar una función que use estratégicamente los elementos subyacentes del TDA (ej: siguienteHora, siguienteMinuto, siguienteSegundo). Procurar hacer validaciones correspondientes.

PREGUNTA 4

Implementar dos funciones en Scheme que permitan filtrar listas de elementos (de cualquier tipo). La primera función debe hacerlo empleando recursión lineal, mientras que la segunda lo debe hacer por medio de recursión de cola. En ambos casos se debe encapsular el tipo de implementación (descuentos hasta 2 ptos. por no hacerlo). Además en sus implementaciones procure indicar el tipo de recursión y el o los elementos distintivos de cada tipo de recursión (2 ptos). Las funciones deben recibir como parámetros una lista de los elementos a filtrar y un filtro apropiado para operar sobre los elementos contenidos en la lista. El resultado de la función es una lista con lo elementos filtrados independientemente del orden. La correspondencia entre el tipo de los elementos de la lista y los filtros proporcionados es responsabilidad del usuario, por lo tanto no realizar validaciones a este nivel. Para la construcción de listas solo puede usar la función cons. Finalmente demuestre el uso de sus implementaciones (llamado y resultado) usando como lista de entrada (list 1 2

3 4 5 6) y un criterio que permita seleccionar los elementos mayores que 3. Este filtro debe ser pasado como una función anónima.

PREGUNTA 5

(A) Implementar una función en pseudoScheme que reciba una lista de strings LP y retorne una lista de valores booleanos. El objetivo del procedimiento es evaluar si los elementos de LP son palíndromos y el resultado de cada evaluación (valor booleano) se debe almacenar en la posición correspondiente de la lista resultante. (B) Debe expresar en un diagrama simple la descomposición del problema. (C) Indicar el o los tipo de recursión empleadas junto a cada función que haga uso de este recurso de programación. (D) Finalmente, mostrar cómo se utiliza la función implementada en (A) a través de UN ejemplo e indicar cuál sería el resultado para la entrada empleada. Nota: Si lo requiere puede hacer uso de las siguientes funciones:

- a. remainder: resto de la división entera
- b. quotient: división entera
- c. string>list: convierte un string a lista de caracteres
- d. eqv?: determina si dos elementos son equivalentes
- e. length: entrega el largo de una lista
- f. car, cdr, cons, null?, list?

Cualquier otra función requerida debe ser implementada.

PREGUNTA 6

Un arreglo de números decimales es “palíndromo” considerando solo el valor absoluto de la parte entera de sus elementos; esto quiere decir, que para un arreglo de ‘n’ números, la parte entera del i-esimo elemento del arreglo es igual a la parte entera del (n-i)-esimo elemento, esto sin considerar el signo.

Por ejemplo los siguientes arreglos cumplen con lo descrito anteriormente:

4.1	-55	1	2	-2.9	1.1	-55	4.2
-----	-----	---	---	------	-----	-----	-----

1	20	-9	20	-1
---	----	----	----	----

3	-3.99
---	-------

a) Establezca una representación adecuada en Scheme para expresar la estructura del arreglo de números decimales que será procesado para determinar si es palindromo o no. Exprese el arreglo de números decimales del ejemplo de la pregunta anterior de acuerdo a la representación escogida.

- b) Realice una adecuada descomposición en subproblemas según las restricciones descritas anteriormente. Indique cuáles son estos subproblemas.
- c) Desarrolle una función en pseudo-Scheme que permita saber si un arreglo de números decimales es "palíndromo". Demuestre el uso de recursión lineal y de cola en la implementación de su solución. Procure indicar dónde y cuándo usa cada tipo de recursión.
- d) Muestre dos ejemplos del uso de su función. Uno para un caso verdadero y otro para un caso falso.

PREGUNTA 7

Se requiere implementar un programa bajo el paradigma funcional para operar con números complejos en su representación binómica.

1. Implementar solución en pseudo-scheme para soportar las 4 operaciones básicas sobre números complejos (considere que la aridad de estas funciones es 2). Procure especificar y explicar una representación apropiada para estos números. Además implementar constructor y selectores naturales.
2. Implementar una función en pseudo-scheme que reciba dos listas de números complejos y una operación para números complejos. La función debe entregar una lista donde cada uno de los elemento corresponde al resultado de operar los elementos correspondientes en cada una de las listas de entrada. Especificar y explicar el tipo de recursión utilizada.

Ej: $((a + bi) (c + di)) + ((e + fi) (g + hi)) = ((a+b) + (e + f)i) [(c + g) (d + h)i]$

Recordatorio:

- Número complejo:
 $a + bi$
- Suma:
 $(a + bi) + (c + di) = (a + c) + (b + d)i$
- Resta:
 $(a + bi) - (c + di) = (a - c) + (b - d)i$
- Multiplicación:
 $(a + bi) * (c + di) = (ac + bd) + (ad + bc)i$
- Division:

$$\frac{(a + bi)}{(c + di)} = \frac{(ac + bd)}{c^2 + d^2} + \frac{(bc - ad)}{c^2 + d^2} i$$