
PROCESAMIENTO DE SEÑALES DE AUDIO Y SUS FRECUENCIAS

202200007 – Natalia Mariel Calderón Echeverría

Resumen

El presente proyecto se concentró en el análisis y procesamiento de señales de audio, expresadas como matrices de frecuencia en un archivo en formato xml, el procesamiento de las señales de audio permitió la determinación de patrones dentro de las mismas señales, dichos patrones dieron paso a la creación de las matrices reducidas de las señales de audio originales.

Las necesidades del Centro de Investigación de la Facultad de Ingeniería requerían de la posibilidad de graficar dichas matrices de frecuencia, tanto las matrices originales como las obtenidas luego de que estas se procesaran en base los patrones encontrados (Matrices reducidas). La relevancia de esta solución software se encuentra en su importancia para el Centro de Investigación de la Facultad de Ingeniería y en la metodología usada para llegar a ella, ya que esta basada en la programación orientada a objeto haciendo uso únicamente de datos abstractos y las librerías necesarias para graficar las señales procesadas.

Palabras clave

Señales, matriz, listas enlazadas, datos abstractos, frecuencia.

Abstract

The present project focused on the analysis and processing of audio signals, expressed as frequency matrices in an XML file. The processing of the audio signals allowed the determination of patterns within the same signals. These patterns led to the creation of reduced matrices of the original audio signals.

The needs of the Engineering Faculty Research Center required the ability to graph these frequency matrices, both the original matrices and those obtained after they were processed based on the patterns found (reduced matrices). The importance of this software solution lies in its importance for the Engineering Faculty Research Center and in the methodology used to achieve it, as it is based on object-oriented programming using only abstract data and the libraries necessary to graph the processed signals.

Keywords

Signals, matrix, linked lists, abstract data, frequency.

Introducción

El procesamiento de señales de audio y sus frecuencias realizado en este proyecto, se llevó a cabo a través del análisis de 2 parámetros propios de las ondas de sonidos: Frecuencia y Amplitud, esto debido a que dichos parámetros son capaces de describir la señal de audio en función del tiempo.

El análisis de dichas señales se llevó a cabo a través de la lectura y correcta interpretación de un archivo XML, que permitió contener la información de cada señal dentro de listas enlazadas. Las listas enlazadas facilitaron la creación de la matriz de patrones y posteriormente la matriz reducida correspondiente a cada señal, a lo largo del proyecto fue necesario la creación de varias listas enlazadas y en ocasiones anidadas una dentro de otra dependiendo de la información que cada una necesitara almacenar.

Los grafos específicos a cada una de las señales fueron creados con ayuda de la librería autorizada: graphviz, dicha librería, de la mano con la correcta aplicación de las listas enlazadas, permitieron la correcta creación de las gráficas de las señales, tanto las señales originales como las señales obtenidas luego de su procesamiento (matrices reducidas)

Desarrollo del tema

El enunciado establecido para este proyecto establece la necesidad de trabajar únicamente con listas enlazadas, prohibiendo así el uso de arreglos, diccionarios y librerías que podrían ser útil para el manejo de información. El problema se plantea desde el punto de vista de listas anidadas, una lista enlazada señales contiene dentro una lista de datos, única para cada señal.

A partir de esta idea es que inicia la creación de listas y nodos, dependiendo de los datos necesarios para el correcto procesamiento de los datos en cada una de las etapas.

Con el objetivo de maximizar el aprovechamiento del tiempo y minimizar al máximo la complejidad, se dividió el proyecto en etapas.

a. Etapa 1 – Lista de señales

Esta etapa inicia con la creación de la lista de señales, tomando en cuenta la idea propuesta de listas anidadas, es decir es necesario crear dos listas enlazadas, una que almacenara señales y la otra que almacenara los datos de cada señal. Partiendo de esto se procede a determinar la información relevante para cada una de estas dos listas siendo esta:

- Lista de datos: tiempo, Amplitud, Valor
- Lista de señales: nombre de la señal, tiempos, Amplitudes

La lista de señales guarda los tiempos y las amplitudes de cada señal, haciendo esto referencia al número de filas y columnas de nuestra matriz de señales, mientras que la lista de datos guarda el tiempo y la amplitud, que hace referencia a la posición del valor dentro de la señal.

Luego de tener las primeras listas del proyecto es necesario establecer la manera en la que se va a leer el archivo xml para luego poder llegar a las listas de la información correcta – tomando en cuenta la necesidad de los datos de estar ordenados se crean las verificaciones necesarias para:

1. Ordenar los datos en la lista

2. En caso de que faltara un dato este será cero

La segunda verificación, esta establecida en las especificaciones necesarias del proyecto.

Para la correcta introducción de los datos, se crea la función “datos_ordenados”, que traba con cada uno de los nodos de la lista de datos, es decir los datos y sus posiciones. Una vez, establecidas las verificaciones previas se procede a crear un método que imprima la lista de señales, esto para asegurar que la lista haya sido llegada de la manera correcta.

b. Etapa 2 – Matriz de patrones

La matriz de patrones es la que posteriormente nos permitirá crear la matriz reducida de frecuencias que es lo que se espera de este proyecto, por lo que su elaboración es una de las partes fundamentales del proyecto.

Para su correcta elaboración se creó la función “procesar_bi”, que trabaja con la información dentro de la lista de señales y la lista datos. La función recorre cada nodo de datos, dentro de cada señal y se encarga de identificar si el valor es igual o distinto a 0, si es igual a 0 el valor se queda igual mientras que si es otro número distinto a cero este se vuelve 1. Esta función crea una lista de ceros y unos que nos permite visualizar de mejor forma los patrones y al igual que con la lista de señales, esta lista se imprime para verificar su correcta elaboración.

c. Etapa 3 – Lista de Comparación

Esta etapa es el segundo paso que se llevo a cabo para conseguir la matriz reducida, y consiste en comparar

cada una de las filas de datos dentro de las señales en la lista de patrones previamente elaborada.

Con el objetivo de facilitar la comparación, se crea otra lista enlazada, llamada lista de comparación en donde todos los datos con el mismo valor t dentro de una señal se convierten en un “String”, esto creara una fila definida y permitirá encontrar las filas que posteriormente deberán ser sumadas para obtener la matriz reducida.

d. Etapa 4 – Lista de Repetidos

Esta lista se creo con el objetivo de mantener un orden específico en cuanto a los grupos repetidos dentro de una misma señal, se tomó la decisión de crear una lista únicamente para almacenar los datos correspondientes a los valores de t de los grupos repetidos debido al orden y requerimientos específicos. Los datos propios de esta lista son:

- Lista de repetidos: nombre y t

Debido a que un dato representa un grupo de una señal específica y los grupos pueden estar formados por varios valores de t , fue necesario establecer una segunda lista anidada dentro de los nodos de la lista de repetidos. Esta lista será la encargada de almacenar los valores de t que forman parte de un mismo grupo y posteriormente se tendrán que sumar, es por eso que los nodos de la lista repetidos quedaron definidos con los siguientes datos:

- Nodo de repetidos: nombre y t
Siendo t una lista enlazada anidada, bajo el nombre de ListaT

Esta lista nos permite almacenar los datos de las determinadas filas que debemos sumar, siendo una señal totalmente independiente de la otra.

La señal compara los elementos de la lista de comparación dentro de una señal que contienen el mismo patrón, y al encontrarlo agrupa los valores de t como la variable t y el nombre de la lista.

De esta forma logra devolver correctamente los grupos de patrones iguales dentro de una misma señal. Este análisis se hace con todas las filas y señales previamente cargadas y analizadas. Una vez, terminado ya es posible avanzar a la parte etapa de análisis final que correspondería a la creación de la matriz de señales reducidas.

e. Etapa 5 – Lista de Reducida

La matriz de señales reducida se obtiene a través de la creación de una última lista llamada, Lista Resultado. Los datos definidos para esta lista son:

- Lista Resultado: nombre, t , A , dato

Esta lista trabaja con la información que contiene la Lista Repetidos y la información original de la Lista de Señales. Para lograr obtener los datos reducidos de las señales, se creó una función encargada de recorrer la lista de repetidos de principio a fin y obteniendo los datos de la lista señales que corresponden a el nombre de la señal y la fila que se desea sumar.

Una vez identificada la señal y las filas por sumar, se procede a la suma de los datos que contienen la misma A dentro de las señales, cuando los datos de las señales. Insertando estos mismo a la lista resultado, manteniendo el nombre, los valores de t , el

valor de A de los datos que se sumaron y el dato obtenido luego de la suma.

Esta función nos permite sumar los datos con patrones iguales, dando paso a la creación de la matriz de señales reducidas.

Debido a la naturaleza de la lista reducida y a la anidación de la lista T , los grupos quedan definidos por la unión de los valores de las filas repetidos, siendo un ejemplo:

- Grupo 1 ($t=1,3$)

La comprobación de la correcta suma y agrupación de valores se realiza a través del método impresión creado para la lista reducida, en donde se puede comprobar la suma correspondiente de cada valor de A dentro de los grupos específicos.

Una vez creada la lista reducida, es posible proceder a uno de los requisitos establecidos por el Centro de Investigación de la Facultad de Ingeniería, que es la creación de un archivo XML que contenga la información obtenida de las señales reducida. Esto se logró a través de la creación de la función “xml_de_resultado” que sigue las reglas y condiciones necesarias para la creación de un archivo XML y hace uso de los datos de la lista de resultados (lista reducida).

A partir de las listas de señales y la lista de resultado (la matriz reducida) es posible crear una representación gráfica de las misma. Para la correcta representación de las señales se creó una función capaz de buscar una señal y sus elementos de acuerdo a su nombre, luego de verificar la existencia de dicha señal se procede a graficarla, haciendo uso de la librería graphviz, que nos permite representar cada dato como un nodo.

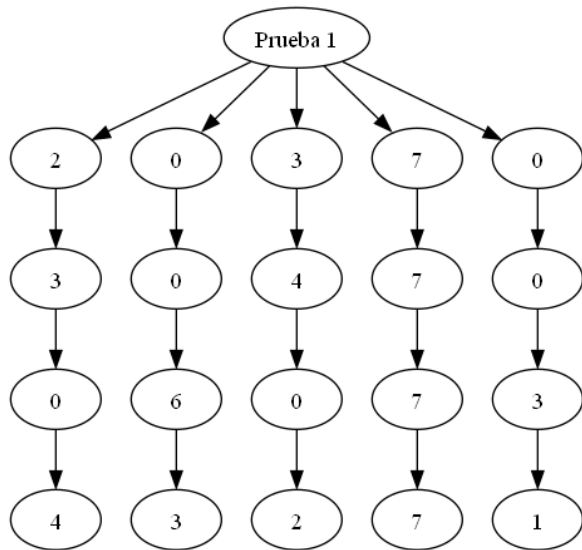


Figura 1. Prueba 1.

Fuente: elaboración propia, a través del proyecto [graphviz]

Para el caso de la lista reducida, se mantuvieron los valores de t debido a la necesidad de poder determinar los grupos y los valores de t usadas para su creación.

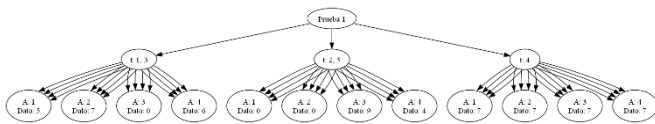


Figura 2. Prueba 1_REDUCIDA.

Fuente: elaboración propia, a través del proyecto [graphviz]

Conclusiones

A pesar de su complejidad, las listas enlazadas suponen una gran ayuda al momento de trabajar con distintos tipos de datos ya que permiten la personalización de las mismas de acuerdo a las necesidades del problema. Logrando de esta manera

un control total y entendimiento mas profundo de los datos a manejar y la solución necesaria.

Un obstáculo que pueden presentar las listas enlazadas es la necesidad de definir los datos que se van a almacenar, esto si bien es cierto supone un planteamiento del problema y posibles soluciones profundo que puede llevar bastante tiempo, este tiempo se ve luego recuperado al momento de comprender el problema y las estructuras necesarias para abordarlo.

Referencias bibliográficas

freeCodeCamp.org. (2020). Data Structures 101: Linked lists. freeCodeCamp.org.

<https://www.freecodecamp.org/news/data-structures-101-linked-lists-254c82cf5883/>

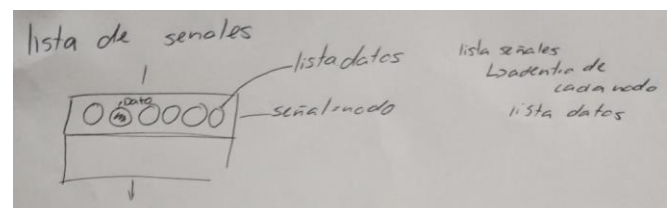
Polygons. (2021, 10 mayo). Graphviz.

<https://graphviz.org/Gallery/directed/crazy.html>

xml.etree.ElementTree — The ElementTree XML API. (s. f.). Python documentation.

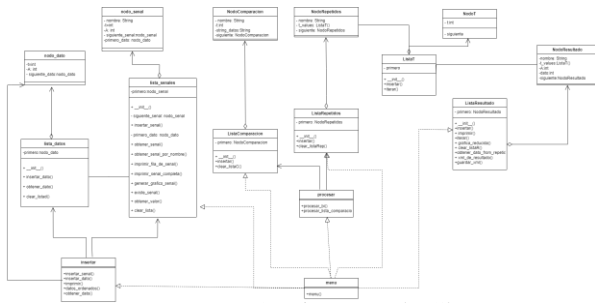
<https://docs.python.org/3/library/xml.etree.elementtree.html>

Anexos



Anexo 1. Primer esquema de idea sobre la aplicación de las listas enlazadas.

Fuente: elaboración propia.



Anexo 2. Diagrama de Clases

Fuente: elaboración propia.