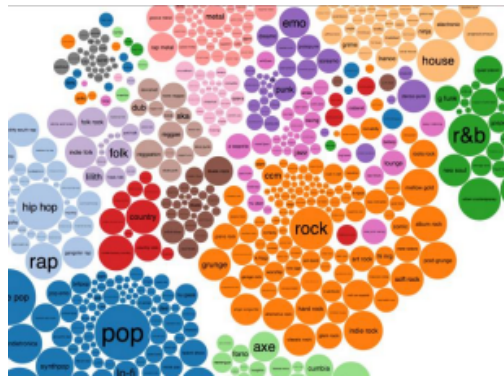


TPE - “Recuperación de Información Musical”

Fecha de entrega: Lunes 16 de Septiembre (hasta las 17hs)

Devolución y coloquio: Lunes 23 de Septiembre

Recuperatorio: Lunes 14 de Octubre



1. INTRODUCCIÓN

La **Recuperación de Información Musical**, conocido como MIR por sus siglas en inglés (*Music Information Retrieval*) es el campo de investigación que se encarga de estudiar los procesos, sistemas y representaciones de conocimiento necesarios para recuperar información de la música. Este campo actualmente está creciendo y puede ser utilizado en aplicaciones que utilizamos a menudo como por ejemplo, los sistemas de recomendación de canciones, la separación de pistas de audio, el reconocimiento de instrumentos, la categorización de música según su género, etc.

Dentro de este campo, es muy común la aplicación de técnicas de procesamiento de señales y en particular, la utilización del formato **WAV**. En este formato, los sonidos son representados como una serie temporal numérica que contiene gran parte de la información que capta nuestro oído. Para entender cómo es posible representar sonidos utilizando números, hay que entender que nuestro oído capta variaciones en la presión del aire que luego nuestro cerebro transforma en sonidos. De manera similar, un micrófono es un dispositivo que captura estas variaciones y las convierte en señales eléctricas que contienen información sobre la presión del aire momento a momento. Luego, es común convertir (con conversores analógico-digital) estos valores continuos en una secuencia de números enteros que indican, momento a momento, el *nivel* de presión en el aire.

En este sistema, las variaciones son representadas con números enteros en un rango acotado. Si los valores posibles se encuentran en el rango $[-2^{(P-1)}, 2^{(P-1)} - 1]$ diremos que la señal tiene una *profundidad* de P bits. Por ejemplo, un audio de 16 bits tiene 65.536 niveles posibles, con 24 bits tenemos 16.777.216 niveles. En la Figura 1 podemos ver un ejemplo de una señal de audio de 4 bits de profundidad en la cual cada punto azul indica cual es la magnitud de la señal en ese instante de tiempo. En este ejemplo, al tener una profundidad de 4 bits, los valores de la secuencia deben pertenecer al rango $[-2^3, 2^3 - 1] = [-8, 7]$.

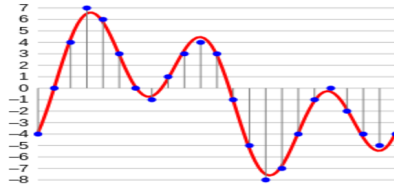


Figura 1: Señal de 4 bits de profundidad

Por último, el formato WAV permite tener más de un *canal* en un mismo audio, entrelazados. Por ejemplo, audio en *estéreo*, donde hay dos canales. Para representar más de un canal, se intercalan los valores de un canal y otro de manera que, en el caso de audio estéreo, la primera muestra corresponde al canal 1, la segunda al canal 2, la tercera al canal 1, la cuarta al 2, etc.

En este trabajo, especificaremos problemas sencillos del área del procesamiento de secuencias temporales con el objetivo de extraer características básicas a partir de ellas.

2. EJERCICIOS

Dados el siguiente renombre de tipos: `type audio = seq<Z>`. Especificar los siguientes problemas:

Ejercicio 1. `proc formatoVálido(in s: seq<Z>, in c: Z, in p: Z, out result: Bool)`

Que dada una secuencia s , compruebe si respeta el formato de un audio de c canales con profundidad p .

Ejercicio 2. `proc replicar(in a: audio, in c: Z, in p: Z, out result: audio)`

Que dado un audio con un solo canal y su profundidad p , lo convierte a un audio con c canales en donde se repita la señal original. Por ejemplo, si $c = 2$ y $audio = \langle 32, 25, -23 \rangle$, el resultado debe ser $\langle 32, 32, 25, 25, -23, -23 \rangle$.

Ejercicio 3. `proc revertirAudio(in a: audio, in c: Z, in p: Z, out invertido: audio)`

Que dado un audio, lo revierte de manera que se escuche de atrás para adelante. Por ejemplo, si $canales = 1$ y $audio = \langle 32, 25, -23, -9, 17 \rangle$, el resultado debe ser $\langle 17, -9, -23, 25, 32 \rangle$.

Ejercicio 4. `proc magnitudAbsolutaMáxima(in a: audio, in c: Z, in p: Z, out máximos: seq<Z>, out posicionesMaximo: seq<Z>)`

Que dado un audio con profundidad p y c canales calcule, para cada canal, la máxima magnitud absoluta y la posición en la que se encuentra.

Ejercicio 5. `proc redirigir(in a: audio, in c: Z, in p: Z, out invertido: audio)`

Es muy común en la música grabada en estéreo (de dos canales) que la voz de los cantantes aparezca tanto en un canal como en el otro. Para eliminar la voz de uno de los canales y que sólo se escuche en uno de los dos, puede utilizarse la técnica de *invertir la polaridad* para contrarestar. Esta técnica consiste en multiplicar por (-1) uno de los canales y luego sumarlo al otro. De esta manera, las señales compartidas entre los dos canales se oponen quedando la información duplicada (la voz en este caso) sólo en uno de los dos canales.

Especificar el problema que, dado un audio de profundidad p con dos canales, aplica la técnica para que los sonidos duplicados sólo se escuchen en el canal c .

Ejercicio 6. `proc bajarCalidad(inout a: seq<audio>, in p: Z, in p2: Z)`

Al disminuir la profundidad de un audio se disminuye la cantidad de niveles posibles para representar la intensidad del sonido y por lo tanto disminuye la calidad del mismo.¹

¹Ejemplo de audio a 16 bits en comparación de audio a 8 bits: <https://www.youtube.com/watch?v=n0Dn5HJGtAY>

Especificar el problema que, dada una secuencia de audios de un canal con profundidad p , modifica cada audio para que su profundidad pase a ser $p2$ (con $p2 < p$) manteniendo lo más posible la forma del audio original. Como se puede observar en la Figura 2, si un audio

$$a = \langle 0, -8, 0, 8, 14, 12, 6, 0, -2, 2, 6, 8, 6, -2, -10, -16, -13, -8, -2, 0 \rangle$$

con profundidad $p = 5$ y $p2 = 4$, la señal de audio resultante mantiene su forma

$$a = \langle 0, -4, 0, 4, 7, 6, 3, 0, -1, 1, 3, 4, 3, -1, -5, -8, -6, -4, -1, 0 \rangle$$

ya que todos sus valores están en el rango $[-8, 7]$. Notar que en caso de que la magnitud resultante no sea entera, se debe redondear para abajo.

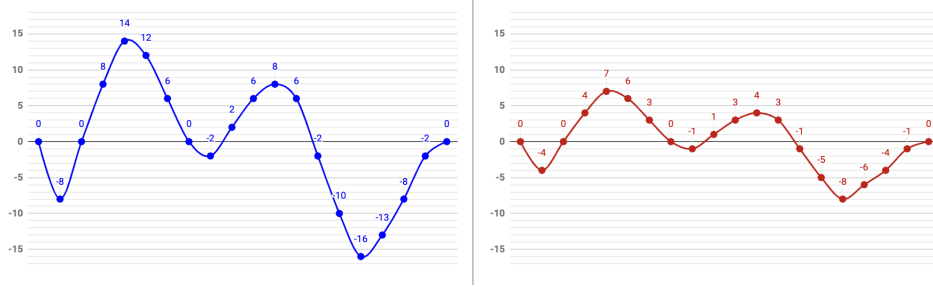


Figura 2: Ejemplo de un audio normalizado

Ejercicio 7. `proc audiosSoftYHard(in sa: seq<audio>, in p: ℤ, in long: ℤ, in umbral: ℤ, out soft : seq<ℤ>, out hard : seq<ℤ>):`

Dada una secuencia de audios de profundidad p y un canal, calcule cuáles son *soft* y cuáles *hard*. Diremos que un audio es *soft* si no posee un conjunto de magnitudes consecutivas de longitud mayor a *long* tal que todas sus magnitudes superen el valor *umbral*. Los audios que no son *soft*, son *hard*.

Por ejemplo, dada la secuencia de audios

$$sa = \langle \langle 10, 12, 5, 9, 35 \rangle, \langle 12, 15, 11, 13, 10, 1, 10 \rangle, \langle 10, 12, 5, 9 \rangle, \langle -5, 2, 10, 12, 3, 4 \rangle \rangle$$

y los valores $long = 4$, $umbral = 5$, se debe devolver

$$soft = \langle \langle 10, 12, 5, 9, 35 \rangle, \langle 10, 12, 5, 9 \rangle, \langle -5, 2, 10, 12, 3, 4 \rangle \rangle$$

$$hard = \langle \langle 12, 15, 11, 13, 10, 1, 10 \rangle \rangle$$

Ejercicio 8. `proc reemplazarSubAudio(in p: ℤ, inout a: audio, in a1: audio, in a2: audio)`

Que dados tres audios de un canal y profundidad p , reemplaza la aparición de a_1 en a con el contenido de a_2 . El audio a_1 puede aparecer a lo sumo una vez en a . Por ejemplo: $a = \langle 25, -23, -9, 17, 93, 10 \rangle$, $a_1 = \langle -23, -9, 17 \rangle$ y $a_2 = \langle -20, -2 \rangle$ entonces a se convierte en $\langle 25, -20, -2, 93, 10 \rangle$

Ejercicio 9. `proc máximosTemporales(in a: audio, in tiempos: seq<ℤ>, out máximos : seq<ℝ>, out intervalos: seq<ℤ × ℤ>)`

Este problema consiste en analizar los valores máximos de audios de un canal en distintos intervalos del mismo. La secuencia *tiempos* indica la longitud de los intervalos a considerar. Por ejemplo, si $tiempos = \langle 2, 3 \rangle$, y el audio a tiene longitud 6, se deberán analizar 5 intervalos (tres de longitud 2 y dos de longitud 3).

El parámetro *intervalos* describirá los índices (cerrado - cerrado) que describen cada uno de los intervalos analizados. En caso de no coincidir el final del intervalo con el final del audio, deberá extenderse el audio con el último valor tantas veces como sea necesario.

El parámetro *máximos* contendrá los valores máximos del audio en cada uno de los intervalos. Notar que las secuencias *intervalos* y *máximos* deben estar en el mismo orden. Es decir, el primer elemento de *máximos*

contendrá el máximo valor del audio en el intervalo definido por $intervalos[0]$, el segundo elemento contendrá el máximo valor del audio en el intervalo definido por $intervalos[1]$, etc.

Tener en cuenta que al ser posible extender los audios, la lista de intervalos puede contener intervalos de índices que no existan en el audio original. Por ejemplo, si $a = \langle 33, 25, -1, 3, 1 \rangle$ y $tiempos = \langle 3, 2 \rangle$, una solución válida es $intervalos = \langle (2, 3), (3, 5), (0, 2), (4, 5), (0, 1) \rangle$ y $máximos = \langle 3, 3, 33, 1, 33 \rangle$

Ejercicio 10. `proc limpiarAudio(inout a: audio, out atípicos : seq(Z)):`

A menudo los audios poseen algunas magnitudes que son numéricamente distantes del resto, los cuales se denominan *outliers*. Estos datos erróneos pueden ser generados debido a un problema al momento de grabarlos y pueden deteriorar la calidad del sonido. Diremos que una magnitud es un outlier si supera al *percentil-95*² de los datos.

Dado un audio de un canal, queremos detectar sus outliers y reemplazarlos por el promedio de las dos magnitudes, no outliers, que se encuentran a su derecha y a su izquierda (en el caso de que el resultado no sea un entero, redondear para abajo). Además, para saber cuales fueron los outliers reemplazados, colocaremos en la lista *atípicos* su posición en el audio.

Términos y condiciones

El trabajo práctico se realiza de manera grupal, pero su aprobación será individual. Para aprobar el trabajo se necesita:

- Que todos los ejercicios estén resueltos.
- Que las soluciones sean correctas.
- Que el lenguaje de especificación esté bien utilizado.
- Que las soluciones sean prolijas: evitar repetir especificaciones innecesariamente y usar adecuadamente las funciones y predicados auxiliares.
- Que no haya casos de sub-especificación ni sobre-especificación.
- Que demuestren en el coloquio que entienden la solución de cualquiera de los ejercicios y puedan explicarlas con sus palabras.

Pautas de Entrega

Fecha de entrega: Lunes 16 de Septiembre (hasta las 17hs)

Devolución y coloquio: Lunes 23 de Septiembre

Recuperatorio: Lunes 14 de Octubre

²<https://en.wikipedia.org/wiki/Percentile>