

Facultad de Ciencias, UNAM.

Programación declarativa.

Proyecto.

Emiliano Galeana Araujo
Juan Alfonso Garduño Solís

1. Introducción

A grandes nuestro proyecto es un programa que a partir de un archivo fuente con notas musicales genera ondas de sonido correspondientes a dichas notas para después reproducirlas con un programa externo. Propusimos este tema de proyecto por curiosidad y gusto; curiosidad porque no teníamos mucha idea de cómo generar sonidos con algún lenguaje de programación y mucho menos con un lenguaje de programación declarativo, pero sabíamos que seguramente era posible y queríamos intentarlo, y gusto porque a quién no le gusta la música. Naturalmente elegimos Haskell porque si nos íbamos a meter en un terreno desconocido, al menos queríamos conocer un poco el lenguaje, y honestamente fué una buena decisión. A continuación explicamos las ideas fundamentales para desarrollar el proyecto, como ejecutar el programa resultante y las conclusiones que pudimos obtener a partir de esta experiencia.

2. Justificación del diseño

Para explicar el desarrollo se deben tener en mente ciertos conceptos de sonido y música. El sonido es una vibración que se propaga, naturalmente en forma de onda y según la forma que tenga la onda es el sonido que podemos percibir, por ejemplo la intensidad del sonido está en función de la amplitud de la onda y el tono del sonido está en función de la frecuencia.

Si tomamos una onda, por ejemplo sinusoidal, y modificamos su frecuencia a 440Hz decimos que la nota que genera es un La, y como no estamos generando sus armónicos decimos que es una nota pura.

Aquí viene la parte difícil, entiendo el código y todo el desmadre pero ya no supe como redactarlo, pero sale hoy

3. Ejecución

Para ejecutar correctamente el programa se debe de tener el software **FFmpeg** instalado en la computadora ya que este es el encargado de reproducir las notas generadas por nuestro programa a través de la herramienta **ffplay**, la instalación no debería ser complicada. Para el desarrollo del proyecto se utilizó **GHC**, así que se recomienda que el modo interactivo que se utilice para cargar el proyecto sea **GHCi**, no garantizamos el correcto funcionamiento del proyecto si se utiliza un compilador distinto.

Además de tener el archivo de entrada (la canción), el cuál debe estar construido siguiendo las siguientes reglas:

- El archivo puede tener comentarios, las líneas del archivo que sean un comentario deben de comenzar con el carácter #.

Ejemplo: `#Este es un comentario`

- La primer linea del archivo que no sea un comentario debe de ser un número (no necesariamente entero), este número representa el tiempo en segundos que todas las notas van a durar.

Ejemplo: `0.25`

Ejemplo: `1`

- Las lineas siguientes al número del tiempo deben ser comentarios o notas en la notación anglosajona musical separadas por un espacio.

Ejemplo: `C C G G A A G`

Este ejemplo representaría las notas Do Do Sol Sol La La Sol

El archivo `estrellita.je` es un ejemplo de un archivo de de entrada bien construido y como su nombre lo indica, codifica la cancion estrellita.

Una vez que cumplidos estos requisitos se debe abrir una terminal en un directorio que contenga los tres módulos del proyecto **AQUI VAN LOS NOMBRES FINALES DE LOS MÓDULOS** y el archivo que contiene la canción a ejecutar, vamos a decir que este archivo es `estrellita.je`. En la terminal vamos a ingresar al modo interactivo de GHC con el comando `gchi Main.hs`, ya que estamos dentro con el comando `interactive "estrellita.je"` se ejecuta el programa y se reproduce la cancion al cabo de unos segundos.

4. Conclusión

Normalmente como programadores no salimos de ciertas zonas de confort (paradigmas) porque son con las que nos instruyeron o son las maneras más naturales de pensar una solución a los problemas que se nos pueden presentar, sin embargo hay formas de solucionar los mismos problemas con distintos enfoques. Para realizar un ejercicio como el presentado en este proyecto lo último que alguien normalmente pensaría sería programarlo en Haskell o cualquier otro lenguaje declarativo porque es difícil ver el camino a seguir, pero es posible y tal vez hasta es más fácil y rápido que el camino habitual. Si bien el proyecto honestamente no tenga mucha utilidad, es el fruto del uso de practicamente todo lo revisado en el curso, desde utilizar elementos que nos brinda el lenguaje y el paradigma como listas por comprensión, caza de patrones, mónadas u operadores fold, hasta estrategias (mañas) para solucionar problemas con cierto patron. Claro que además de la capacidad de modelar el problema en la computadora, en este caso particular modelar las ondas de sonido, dicha capacidad adquirida a lo largo de la carrera.

Entonces, en conclusión podemos decir que.... **No se me ocurrió nada más jajaja**

```
translate (x:xs) =
  if isAlpha x
  then
    let
      posible_note = Map.lookup x map1
    in
      case posible_note of
        Just (note, ord) -> [(ord + 40)] ++ translate xs
        Nothing -> [] -- omite las letras que no estn.
  else translate xs
```

5. Bibliografía