

TFG

Jorge Marco Esteve

11 de mayo de 2020

1. Antecedentes

Antes de meterte en OMR tienes que hablar de gramáticas musicales, por qué se han codificado, etc... - ¿Te pasé el libro Beyond MIDI: The Handbook of Musical Codes, no?

1.1. Representación Musical

Los códigos musicales han sido utilizados desde sus inicios para facilitar la transcripción de sonidos, desde los neumas que se representaban el canto en monasterios medievales, al solfeo de música pedagógica del siglo actual. Muchos códigos musicales son de uso común, particularmente en la música pedagógica, solfeo. El cual su propósito principal es inculcar en la mente del cantante la manera de relacionar cada tono con sus vecinos y, por lo tanto, proporcionar una base sólida para cantar con precisión a simple vista.

Los números de los dedos en piano, los ideogramas los cuales denominamos tablaturas para guitarra y otros instrumentos con trastes, las notas de notación moderna y muchos símbolos especiales diseñados para transmitir formas particulares de tocar la batería son códigos de un tipo u otro. En este ámbito, el código de lenguaje más utilizado y completo para representar el sonido es la notación musical común o notación musical moderna.

La notación musical moderna o común es la piedra angular en pivota la gran mayoría de sistemas de representación musical actual y de los últimos siglos.

Entre el s.XIV y el s.XX fueron inventados diferentes códigos con el fin de alcanzar objetivos que historiadores musicales, himnógrafos y etnomusicólogos. Pero, desde el siglo XX a la actualidad se ha contemplado la creación de notación para computadores de una manera sólida, ya que debido a su eficiencia y su memoria resulta una oportunidad codiciosa para, por ejemplo, representar largas obras musicales con tantos atributos como queramos codificar.

1.2. Optical Music Recognition (OMR)

Estos últimos años, la disponibilidad de grandes colecciones de partituras digitales ha facilitado tanto la práctica profesional de la música como el acceso del aficionado a fuentes impresas que eran difíciles de obtener en el pasado.

Podemos encontrar ejemplos de estas colecciones en diferentes páginas web que llegan a almacenar cientos de miles de partituras como es el caso de IMSLP <http://www.imslp.org>. Además, las bibliotecas públicas y privadas realizan muchos esfuerzos para publicar sus colecciones online como es en el caso de <https://drm.ccarh.org>. Sin embargo, a demás de esta accesibilidad a contenido musical, las ventajas de tener una imagen digitalizada de cualquier obra no se limita a la facilidad de copiarlas y distribuirlas, dichas obras digitalizadas ofrecen una ausencia de desgaste que el recurso físico no es capaz de ofrecer.

Las grandes posibilidades que las aplicaciones actuales basadas en música pueden ofrecer están restringidas a partituras codificadas simbólicamente

Tienes que añadir alguna sección para explicar qué es una codificación simbólica frente a imagen y audio digital

. El problema de este método es que la transcripción de las partituras se realiza de forma manual

Lleva cuidado que hay demasiado copiado de nuestros papers

. Al hacerse de esta forma, el costo, tanto en tiempo como en recursos, es prohibitivo. Para paliar este inconveniente sería necesario recurrir a un sistema de transcripción automática. El llamado Optical Music Recognition (OMR) se define como la investigación sobre cómo enseñar a las computadoras a leer notación musical, con el fin de poder exportar su contenido al formato deseado de manera automática, similar a la utilidad de una caja negra. A pesar de las grandes ventajas que supone su desarrollo, en estos momentos, OMR está a mitad de camino de ser totalmente confiable, ya que existen problemas de tareas sencillas que obstaculizan de manera importante OMR como la eliminación de las líneas del pentagrama, la localización y clasificación de símbolos, o el ensamblaje de notación musical. Sin embargo, los recientes avances del aprendizaje automático y, específicamente, del aprendizaje profundo, no solo permite resolver estas tareas con facilidad, sino también replantearlas para enfrentar todo el proceso de una manera más elegante y compacta, evitando ciertas reglas que hacen que los sistemas se limiten al tipo de entradas al que están diseñados.

Una cosa que cabe señalar y que resulta fundamental en este trabajo es que OMR trabaja bajo una representación agnóstica en la que únicamente se tiene en cuenta el punto de vista gráfico a la hora de transcribir las partituras. Este trabajo intenta mejorar esta inteligencia artificial (OMR) por medio de una representación semántica, la cual, codifica los símbolos de acuerdo con su significado musical. Aquí entra en uso la gramática que se va a realizar en este trabajo: Kern y Mens, dado una notación moderna o mensural respectivamente.

Esta frase no está bien. Puedes decir que el OMR nuestro (pones referencias de los papers) permite trabajar con agnóstico y semántico. Que en el semántico - paper PRIMUS - la gramática la diseñamos nosotros. Ahora mismo estamos trabajando con `**kern` pero nos da dificultades. Estamos haciendo .a paños como introducir un código para los multirests, y que tú, en este trabajo vas a hacer una primera formalización (nosotros no paramos y estamos otros cambios que no vas a meter en tu TFG).

1.3. Kern y Mens

Actualmente hay varios sistemas de representación musical como puede ser DARMS[1], SCORE[2], EsAC[3], Kern[4]. Estos sistemas de representación han sido usados para codificar simbólicamente partituras, ya sea para renderizarlas a PDF y obtener escritos de ellas, o para reproducirlas, obteniendo de dichos símbolos una melodía.

Referencias
a todos

Humdrum[5] es un software creado por David Huron en la década de los 80s. Dicho software es empleado con el fin de obtener diferentes recursos musicales a nivel computacional. Junto a su tipo de notación, `**kern`, se puede obtener PDFs de partituras e incluso se puede reproducir la melodía previamente implementada. Esta notación, `**kern`, es una representación secuencial musical de manera vertical en el que cada columna representa una voz de cada melodía con el fin de representar finalmente una partitura. En esta representación podemos implementar cualquier tipo de notación moderna occidental además de poder manipular los siguientes apartados:

cuidado con
los acentos,
estoy corri-
giendo unos
cuantos

- El tono: alteraciones musicales, claves, posición de claves, armadura, fórmula de compás
- La duración: silencios, puntos, ligaduras, ligados.
- Articulaciones y Ornamentos: staccato, tenuto, pizzicato.

Pon lo de antes usando listas

Una variante de la representación `**kern`[4] es `**mens`[6]. Este tipo de representación acoje todo tipo de notación mensural blanca con muchas coincidencias con Kern a la hora de representarse, ya que este tipo de notación también es vertical y se gestiona de una forma muy parecida a la de notación moderna.

El defecto que lleva a esta representación a no estar realmente completa es la falta de gramática que es el tema correspondiente a este trabajo.

1.4. Gramática

Como diría el diccionario, gramática es un conjunto de normas y reglas que se usan para hablar y escribir correctamente una lengua. La finalidad de este trabajo es crear una gramática lo mayormente competente como para poder, a partir de dichas normas y reglas, representar cualquier tipo de notación musical

Vas a tener
un tribunal
de informá-
ticos, debes
hacer una
introducción
más técnica

de la manera más correcta posible. En este caso, la notación Kern y Mens será la adecuada para la creación de la gramática. Obteniendo la primera gramática que comprende los dos tipos de notación: moderna y mensural.

Para implementar nuestra gramática utilizaremos un analizador sintáctico el cual va obteniendo tokens con la finalidad de decidir si un conjunto de estos tokens cumplen las leyes de nuestra gramática. Hay dos tipos de analizadores sintacticos: analizador sintáctico descendente y analizador sintáctico ascendente. El primero parte de una regla inicial y concluye en unos tokens que son generados por las reglas. Mientras el analizador sintactico ascendente parte de un determinado grupo de tokens con el fin de llegar a una regla principal.

Nosotros utilizaremos un analizador sintáctico descendente ya que dada la complejidad que nos puede suponer una gramática musical, uno de las cosas positivas del analizador sintáctico descendente respecto al ascendente es, además de su eficiencia, su sencillez a la hora de implementarlo.

Para implementar nuestro analizador usaremos una herramienta denominada ANTLR4. Con esta herramienta podemos contruir reconocedores , intérpretes, compiladores y traductores de lenguajes a partir de descripciones gramaticales de los mismos.

Debes ser más extenso en esto

referencia al libro

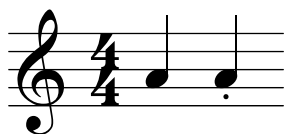
2. Banco de ejemplos

2.1. SKERN

2.1.1. Accidents

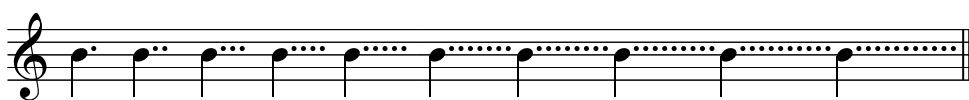


2.1.2. Articulations

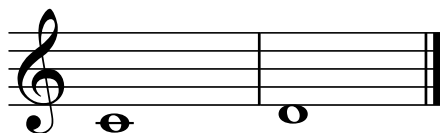




2.1.3. Dots

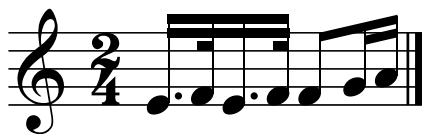
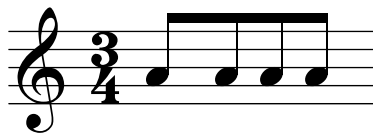
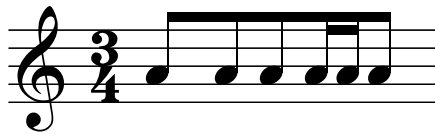
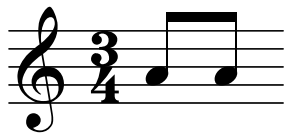


2.1.4. Barlines

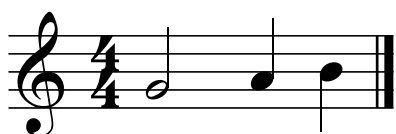
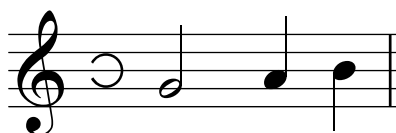
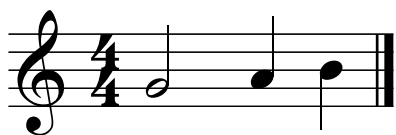
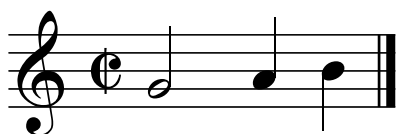
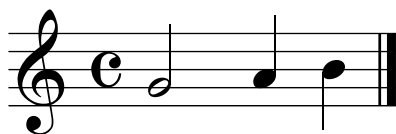


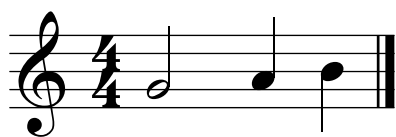
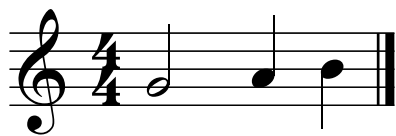


2.1.5. Beaming



2.1.6. Meter

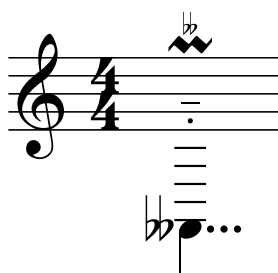




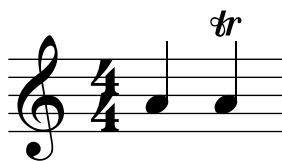
2.1.7. Keysignature

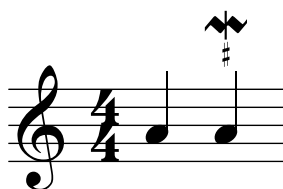
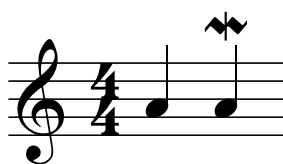
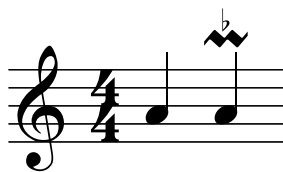
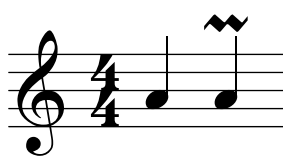
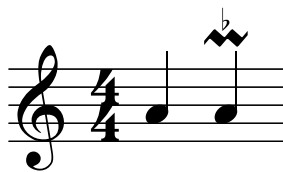


2.1.8. Notes



2.1.9. Ornaments

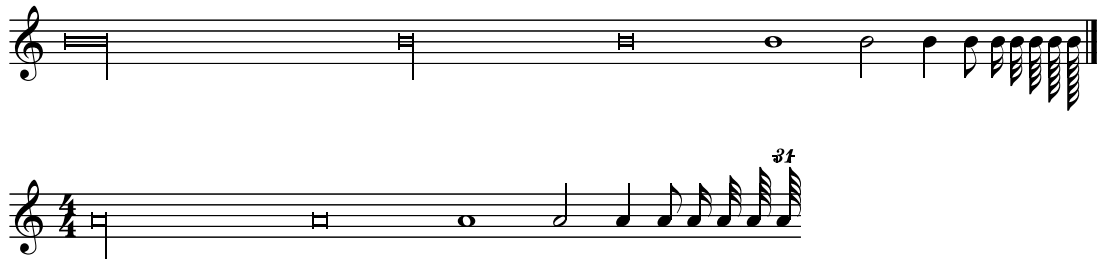




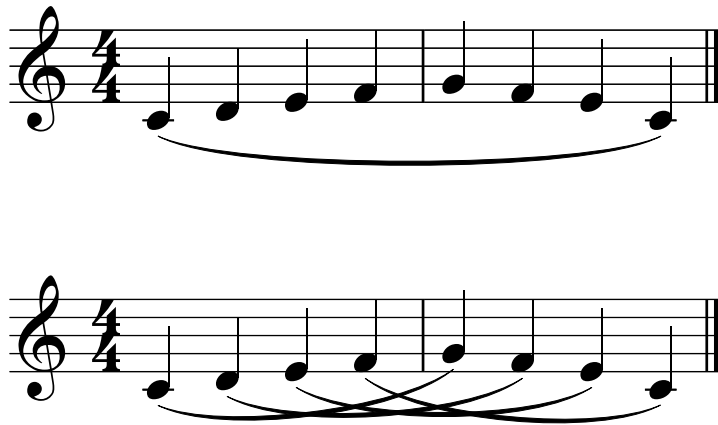
2.1.10. Rests



2.1.11. Rhythm



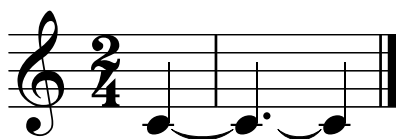
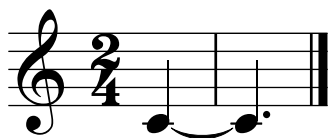
2.1.12. Slurs



2.1.13. Stem Direction

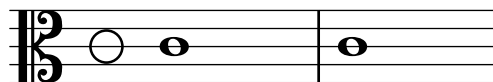
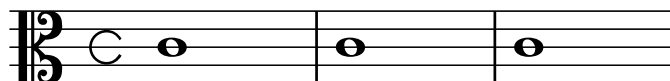


2.1.14. Ties

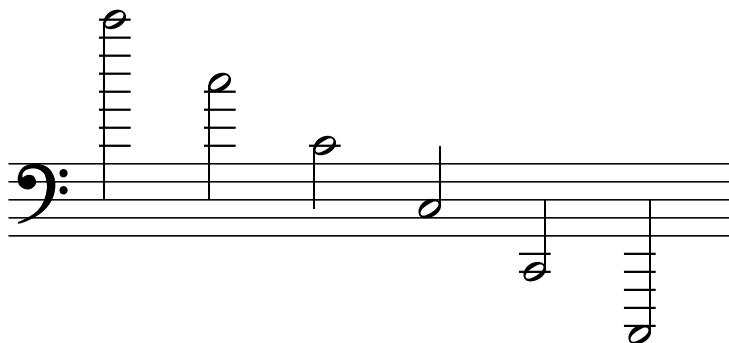




2.1.15. Time Signature



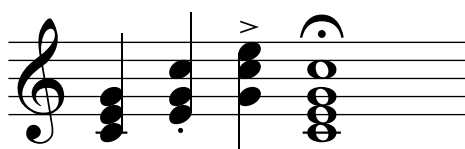
2.1.16. Octaves



2.1.17. Chage configuration

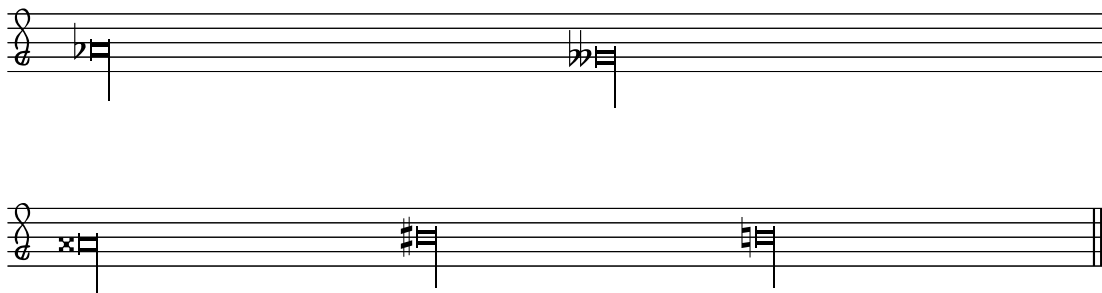


2.1.18. Chords

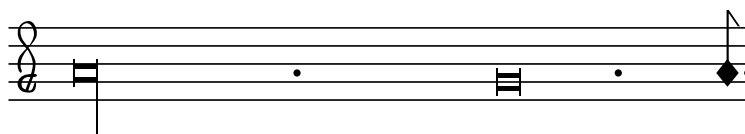


2.2. SMENS

2.2.1. Barlines



2.2.2. Dot

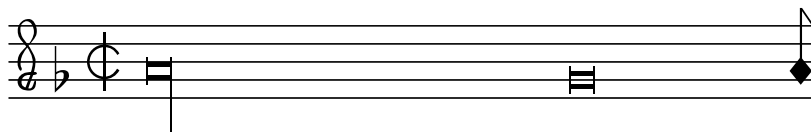
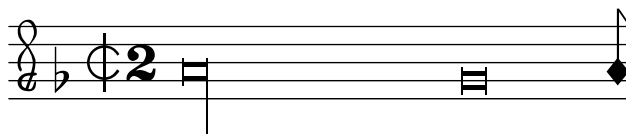
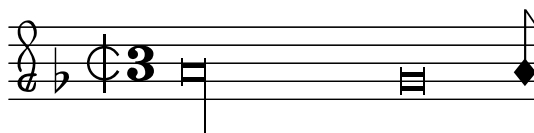
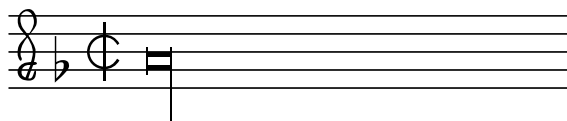
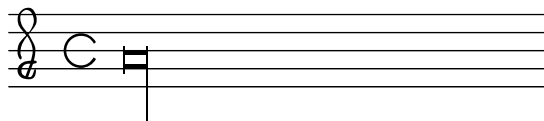


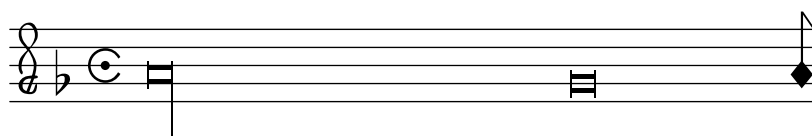
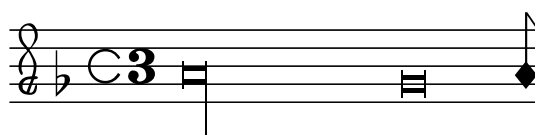
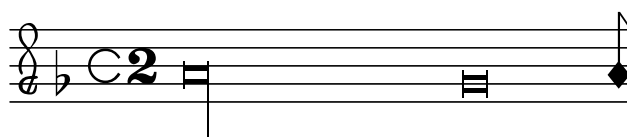
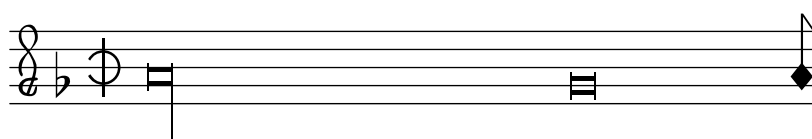
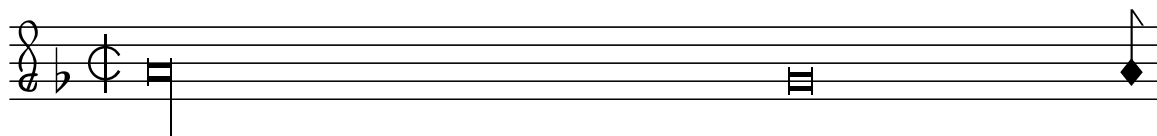
2.2.3. Ligature

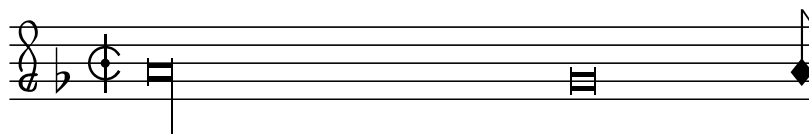


2.2.4. Meter

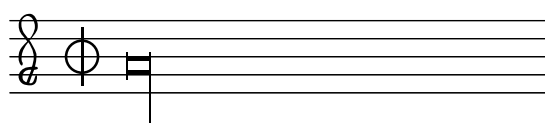
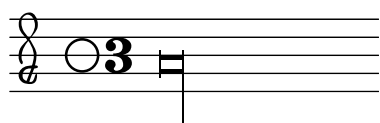
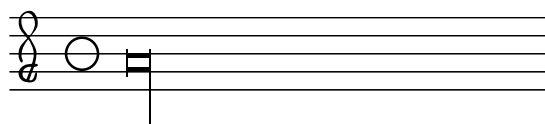
Common Meter

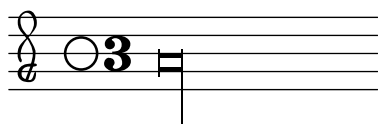
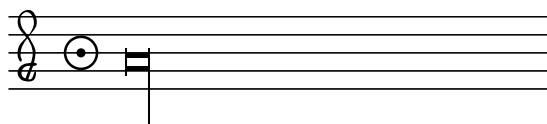
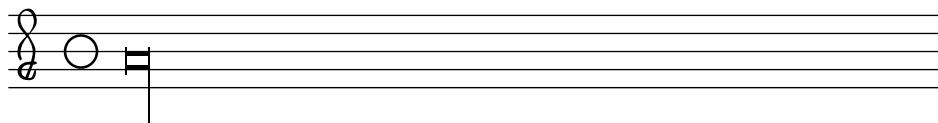
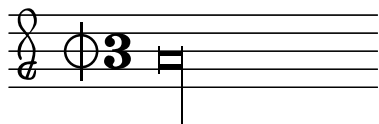


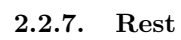
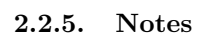




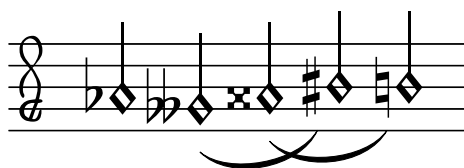
Perfect Meter







2.2.9. slurs



Referencias

- [1] Digital Alternate Representation of Musical Scores creado en 1966 por Stefan Bauer-Mengelberg.
- [2] Sistema de representación musical creado en 1971 por Leland Smith.
- [3] Sistema de representación desarrollado para crear música monofónica a finales de los 80s por Helmut Schafrasth.
- [4] Notación musical creada por David Huron en la década de 1980.
<https://www.humdrum.org/guide/ch02/>
- [5] <https://www.humdrum.org>.
- [6] Notación mensural blanca derivada de `**kern`
<https://doc.verovio.humdrum.org/humdrum/mens/>

Cuidado, las ligaduras / slurs se usan poco en mensural, mejor un ejemplo de notación moderna

Los títulos de las secciones en español