

Informática para a Musicologia (IPM) 2024/25

Jupyter Notebooks

Docente: [J.N. Oliveira](#)

Departamento de Informática da U. Minho, em colaboração com a  ENSICO

Aula de 26-Nov: aula TP sobre bibliotecas IPM em Haskell

Sumário Continuação da exploração das bibliotecas IPM (Haskell) para 'Computer-Aided Musicology'. Continuação do estudo das operações de verticalização. Síntese e manipulação de acordes e arpejos. Exemplos e caso de estudo: *Alegretto* da 7.^a sinfonia de Beethoven (1770-1827).

⚠ Importante: correr sem mexer as células seguintes.

In []:

```
:opt no-lint
:m Data.Char
:m Data.List
:m Data.List.Split
:m Data.Ratio
```

Módulos desenvolvidos para a disciplina:

In []:

```
:l ../src/Cp.hs
:l ../src/Reducer.hs
:l ../src/Ipm.hs
:l ../src/Abc.hs
```

Dados ("case studies"):

```
In [ ]: :l ../src/CS.hs
```

Verticalização

A seguinte redução do início do segundo andamento (*Alegretto*) da 7.ª sinfonia de Beethoven

foi obtida a partir da partitura original - partes de viola, violoncelo e baixo -

através funções cujo estudo será o principal objectivo deste *notebook*.

11.1 O fragmento musical em causa está disponível sobre a forma de uma lista `op93ii` de quatro partes, a saber:

```
In [ ]: (abcPlayM "C" "2/4" . P) op93ii
```

Antes de correr as células seguintes indique qual o significado do que espera obter. Que informação é que essas células lhe poderão dar sobre `op93ii` ?

```
In [ ]: length op93ii
map length op93ii
```

```
In [ ]: ritmo = map snd
-----
(length . nub . map ritmo) op93ii
```

11.2 Pretende-se agora que cada uma das partes de `op93ii` não repita notas iguais sucessivas. Adapte

```
| (abcPlayM "C" "2/4" . P) op93ii
```

por forma a isso acontecer, e ouça o resultado.

```
In [ ]: (abcPlayM "C" "2/4" ..... ) op93ii
-- abcShow
```

11.3 - Podemos decompor a lista `op93ii` nas suas quatro partes. Complete a célula seguinte para obter as que faltam:

```
In [ ]: v1 = head op93ii
v2 = .....
vc = .....
ba = last op93ii
```

11.4 - Começando por definir

```
In [ ]: beet = abcPlayM "C" "2/4"
        r = ritmo ba
```

corram as células seguintes e observem os resultados. No fim, digam por palavras vossas o que faz a função `dchords` .

```
In [ ]: (beet . dchords r) op93ii
```

```
In [ ]: (beet . dchords bin) op93ii
```

```
In [ ]: s = 3%8 : 1%8 : s
        ---
        (beet . dchords s) op93ii
```

Qual é então a diferença entre `dchords` e `chordify` ? Respondam na célula seguinte.

11.5 - Programe agora a célula seguinte para obter nela a redução de `op93ii` que foi mostrada no início, isto é:

The image shows a musical score for a piece in 2/4 time, bass clef. It consists of two systems of music. The first system has 8 measures. The second system starts with a measure rest (indicated by a 'z' in the bass line) and a '10' marking above the first measure, followed by 7 more measures. The score is written in bass clef with a 2/4 time signature.

In []:

```
.....
----
(beet . P) .....
```

11.6 - Finalmente, comparar o que se obteve acima em

```
| beet (dchords s op93ii)
```

com o resultado da célula seguinte - qual é a relação entre `dvert` e `dchords` ?

In []:

```
dvert bin op93ii
```

11.7 - Em cima, `dvert` produziu sequências de várias notas para a mesma duração, por exemplo,

```
| (["A","E","C","A,,"],1 % 2)
```

que podemos considerar uma boa representação da noção de **acorde**. De facto,



é o que se obtém correndo a célula seguinte, convertendo o acorde para notação Abc.

In []:

```
abch (["A","E","C","A,,"],1 % 2)
----
abcplease it
```

11.8 - No sentido inverso, podemos pensar agora em arpejar um dado acorde, por exemplo, em converter o acorde

```
| (["A","E","C","A,,"],1 % 2)
```

em



É o que obterá correndo a célula seguinte:

In []:

```
harp (["A","E","C","A,,"],1 % 2)
```

11.9 - Analisando os detalhes da definição da função harp

```
harp(ns,d) = zip ns s where s = (d*(1%length ns)) : s
```

usada na célula anterior, explique o seu funcionamento por palavras suas. Qual é o papel de `s` na definição?

Em suma:

Designação	Significado	Descrição detalhada
chordify	acordes (eixo horizontal)	chordify s m agrupa as notas de m em acordes segundo as durações prescritas em s
dvert	verticalização	dvert d [a,b,..] agrupa verticalmente as notas de várias linhas melódicas [a,b,..] perviamente amostradas por d
dchords	acordes (eixo vertical)	dchords d [a,b,..] faz uma verticalização e converte-a em acordes segundo a notação Abc
dchunks0f	dividir em blocos	dchunks0f d m divide melodia m numa sequência de melodias segundo as durações especificadas em d
harp	arpejar	harp (n,d) arpeja o acorde (n,d) dividindo a duração d pelo número de notas de n

11.10 - (Síntese - para realizar após a aula) Conceba na célula seguinte as etapas necessárias para, usando todos os operadores estudados até ao momento, gerar a partir de `op93ii` a "variação" que é mostrada no pentagrama superior de:

System 1: Two staves in 2/4 time. The upper staff features a continuous eighth-note pattern. The lower staff contains a sparse accompaniment with a few notes and rests.

System 2: Two staves in 2/4 time. The upper staff continues the eighth-note pattern. The lower staff has a few notes and rests.

System 3: Two staves in 2/4 time. The upper staff has a more complex eighth-note pattern. The lower staff has a few notes and rests. A measure rest is present in the first measure of the upper staff. A measure number '10' is written above the third measure of the upper staff.

System 4: Two staves in 2/4 time. The upper staff has a complex eighth-note pattern. The lower staff has a few notes and rests.

System 5: Two staves in 2/4 time. The upper staff has a complex eighth-note pattern. The lower staff has a few notes and rests.

In []: