Informatics for Musicology (IPM) 2024/25

Jupyter Notebooks

Teacher: JN Oliveira

Department of IT at U. Minho, in collaboration with ENSICO

Class from 5-Nov:

> Continued exploration in the Jupyter environment of the IPM (Haskell) libraries for 'Computer-Aided Musicology'. Octave transpositions (`octave`). Usefulness of the function `sample` in analyzing the musical text and constructing its variations. Removal of passing notes. Relationship between `nrep` and `sample` . Examples and case studies.

⚠️ **Important** : run without moving the next cells.

In [ ]:
```
: opt  no - lint
: m  Data . Char
: m  Date . List
: m  Date . List . Split
: m  Data . Ratio
```

Modules developed for the discipline:

In [ ]:
```
: l  ../ src / Cp . hs
: l  ../ src / Reducer . hs
: l  ../ src / Ipm . hs
: l  ../ src / Abc . hs
```

Data ("case studies"):

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

🔴Sampling (continued)

Remember:

> | Designation | Meaning | Detailed description |
> |---|---|---|
> | `sample` | sampling | `sample s m` sample the melody `m` according to the sequence of durations `s` |

Today we will see how to use this important operation to prepare **tonal analysis** of a piece of music.

Analysis means **abstracting** *details* , with a view to preserving the **essence** of what is being studied.

Let's start with something simpler: *removing repetitions* .

8.1 - Scroll to the next cell to view a fragment of string quartet No. 10 in Eb ('Harp', op74) by Ludwig van Beethoven (1770-1827) - 3rd movement (1st violin):

In [ ]:
```
c  =  3 % 8 : tern
d  =  3 % 8 : d
abcPlay  "Eb"  "3/4"  c  op74iii
```

NB: to hear the violin tone (in MIDI) add

> `%%MIDI program 1 40`

to the Abc generated by the next cell and tap on the web editor .

In [ ]:
```
abcShow
```

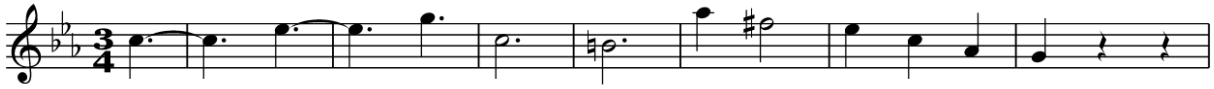
8.2 - Now anticipate what will be produced in the next cell, without executing:

In [ ]:
```
abcPlay  "Eb"  "3/4"  c  ( nrep  op74iii )
```

8.3 - Then, define it `s` so that we have one note per beat (but pay attention to the anacrusa):

In [ ]:
```
s  =  undefined
d  =  undefined
---
abcPlay  "Eb"  "3/4"  c  ( sample  d  op74iii )
```

8.4 - Define the sampling sequence `s` that allows, in the next cell, to make the following abstraction of the given topic:



In [ ]:
```
c  =  undefined
s  =  undefined
x  =  sample  s  ( nrep  op74iii )
---
abcPlay  "Eb"  "3/4"  c  x
```

8.5 - Returning to

In [ ]:
```
mel  =  [( "d" , 3  %  4 ),( "^c" , 3  %  4 ), ( " d" , 3  %  4 ),( "e" , 3  %  4 ),( "f" , 3  %  4 ),( "e"
bas  =  [( "D," , 3  %  4 ),( "A,," , 3  %  4 ),( "D," , 3  %  4 ), ( " C ," , 3  %  4 ), ( " F," , 3 % 4 )
```

extract the top line from this famous theme

In [ ]:
```
abcPlayM  "F"  "3/4"  revelry
```

by Arcangelo Corelli (1653-1713), in the following cell:

In [ ]:
```
abcPlayM  undefined
```

8.6 - Comparing

Designation	Meaning	Detailed description
`sample`	sampling	`sample s m` sample the melody `m` according to the sequence of durations `s`
`nrep`	"ligatures"	consecutive notes with the same pitch are linked into a single note with the corresponding total duration

It can be seen that the two functions do the opposite of each other:

> while one divides notes (`sample`) the other joins them (`nrep`).

However, this is not always `nrep (sample s x) = x` the case. What could be the difference?

Remembering

In [ ]:
```
help  =  abcPlay  "C"  "none"  ( 1 % 4 : quatern )  .  P
---
help  [ susana ]
```

(**NB** : the function `help` is intended to save code...) anticipate the differences that will be noticed in the transformations `susana` shown in the next cell and comment on them:

In [ ]:
```
help  [
        susana ,                          --- original
        nrep  ( sample  half  susana ),     --- eighth note sampling
        nrep  ( sample  una  susana )       --- quarter note sampling
      ]
```

8.7 - Listen to the following fragment of *Dido's famous Lament* from the opera *Dido and Aeneas* by Henry Purcell (1659-1695):

In [ ]:
```
dido  =  P  [ dido0 ,  dido4 ]
---
abcPlayM  "Bb"  "6/4"  dido
```

(a) Indicate why `rep(sample s dido0)` it is always different from `dido0` , whatever `s` .

(b) Make two samples `dido4` such that `rep(sample s dido4) = dido4` it is verified in one but not in the other.

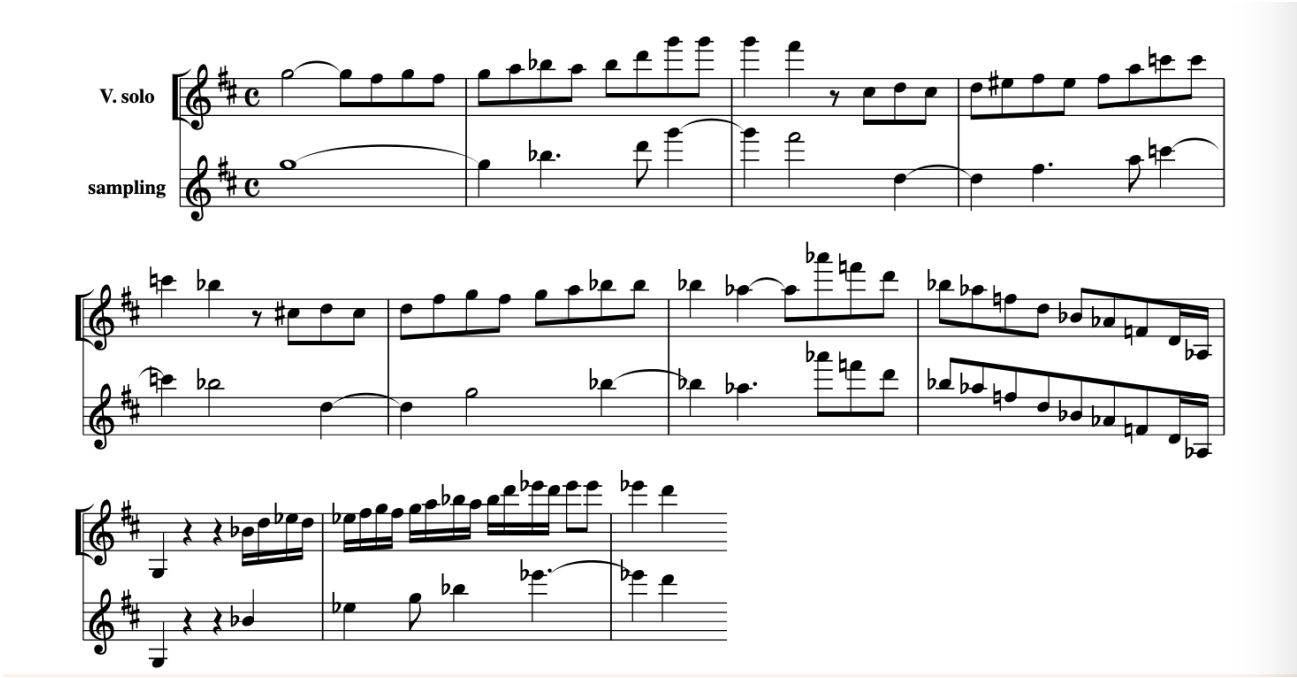
In [ ]:

In [ ]:

8.8 - Returning to Beethoven, let us now see and listen to the following fragment of the solo violin part of the 1st movement of his concerto opus 61 (`op61i`):

In [ ]:
```
abcPlayM  "D"  "C"  op61i
---
abcShow
```

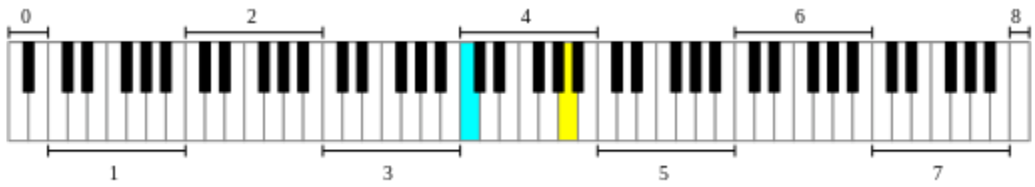
The aim is now to remove, by sampling , *passing notes* , thus highlighting those that have tonal expression - see `sampling` below



Build, in the next cell, the sampling sequence `s` that produces the effect shown in the figure.

In [ ]:

8.9 - The fourth octave of the piano, which starts at C_4, is the reference octave of ABC notation, cf. :W_4, is the reference octave of ABC notation, cf. `C D E F G A B`



Looking at the following table,

Designation	Meaning	Detailed description

Designation	Meaning	Detailed description
`octave`	octave below or above	`octave i m` is music `m` `i` octaves up or down, if ABC notation is being used
`collapse`	force the octave 4	`collapse m` is the song `m` forced not to leave the octave that starts in C_4W_4

Comment on the results of the following cells:

In [ ]:
```
abcplease ( octave 1 susana # susana )
```

In [ ]:
```
abcplease ( abegg # abegg collapse )
```

In [ ]:
```
abcplease ( susana # collapse susana )
```

8.10 (consolidation) - Start by viewing and playing the following ABC:

```
%%scale 0.7
%%pagewidth 20cm
%%barnumbers 10
X:1
%-- Abc file generated by Haskell library Abc.hs (IPM 2023/24)
M:C
L:1/1
K:F
%%staves [1 2 3]
V:1
V:2
V:3
%-- the parts now
[V:1] F1/4G1/4A1/4F1/4|F1/4G1/4A1/4F1/4|A1/4B1/4c1/2|A1/4B1/4c1/2|
c3/16d1/16c1/8B1/8A1/4F1/4|c3/16d1/16c1/8B1/8A1/4F1/4|F1/4C1/4F1/2|F1/4C1/4F1/2|
F1/4G1/4A1/4F1/4|F1/4G1/4A1/4F1/4|A1/4B1/4c1/2|A1/4B1/4c1/2|
[V:2] z1/1-|z1/1|F3/8G1/8A3/8F1/8|F3/8G1/8A3/8F1/8|A3/8B1/8c3/8c1/8|A3/8B1/8c3/8c1/8|
c3/8B1/8A3/8F1/8|c3/8B1/8A3/8F1/8|F3/8C1/8F3/8F1/8|F3/8C1/8F3/8F1/8|F3/8G1/8A3/8F1/8|
F3/8G1/8A3/8F1/8|
```

```
[V:3] z1/1-|z1/1-|z1/1-|z3/4F1/4-|F1/4A1/2F1/4-|F1/4A1/2A1/4-|A1/4c1/2A1/4-|A1/4c1/2c1/4-|
c1/4A1/2c1/4-|c1/4A1/2F1/4-|F1/4F1/2F1/4-|F1/4F1/2
%-- end of generated abc, key is -1
%
```

Then being given

In [ ]:

```
n  =  frerej ++  n   -- Frere Jacques 'ad eternum'
```

fill in the `undefined` in the next cell to obtain the effect of the same ABC, using the functions `delay` , `sample` , `dtake` and `octave` on what is already given.

In [ ]:

```
p  =  undefined
b  =  undefined
x  =  undefined

---
abcPlayM  "F"  "C"  ( P  [ x , p , b ])
abcShow
```
