Português (Portugal) → Inglês

Informatics for Musicology (IPM) 2024/25

Jupyter Notebooks

Teacher: JN Oliveira

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

Oct 15 Class : Experience with Haskell libraries for *Computer-Aided Musicology*

Manipulation of sequences that represent music. Interaction with Abc notation. Measures and polyphony.

Formal representation of polyphony. The combiner P . Example: Canon nr.1 of the *Canones diversi super thema regium* of BWV 1079 by JS Bach (1685-1750). Reverse movement ('reverse' function).

Imitation and repetition.

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

In [ ]:

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

In [ ]:

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

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

🔴Bars

Let's go back to:

In [ ]:
```
m = ["A", "^G", "A", "B", "c", "B"]
r = [3%4, 3%4, 3%4, 3%4, 3%4, 3%4, 3%4]
mel = zip m r
---
abcplease mel
```

4.1 - The following visualization adds triple barlines, cf. `3/4` :

In [ ]:
```
abcPlayM "C" "3/4" honey
```

In [ ]:
```
abcPlayM "A" "3/4" honey
```

👉About `abcplease` and `abcPlayM`

Designation	Meaning	Detailed description
`abcplease`	show sheet music	`abcplease m` shows the score of `m` without key signature or time signature, nor barlines
`abcPlayM`	show sheet music	`abcPlay K C m` - shows the score `m` with key signature `K`, `C` time signature and barlines (regular) deducted from `C`

4.2 - Do the same thing for `carnaval_serrano` , using binary (`2/4`) and quaternary (`4/4`):

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

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

🔴Polyphony

➡️Given two melodic lines `m` and `n` , `P [m,n]` represents their simultaneous execution, i.e. *polyphonic* .

The notation extends to more than two voices, for example, `P [m,n,p]` etc.

Sometimes we use the abbreviation `m # n` when there are only two voices, cf:

In [ ]:
```
m # n = P [ m , n ]
```

4.3 - Perform the following example of two-voice polyphony:

In [ ]:
```
soprano = mel
bass    = [( "A," , 3 % 4 ),( "E," , 3 % 4 ),( "A," , 3 % 4 ),( "G," , 3 % 4 ), ( "C" , 3 % 4 ),
---
abcPlayM "C" "3/4" ( soprano # bass )
```

Add a tenor part, also in dotted half notes (edit):

In [ ]:
```
tenor = zip [ "C" , "E" , "C" , "D" , "E" , " D " ] [ 3 % 4 , 3 % 4 , 3 % 4 , 3 % 4 , 3 % 4 , 3 % 4
```

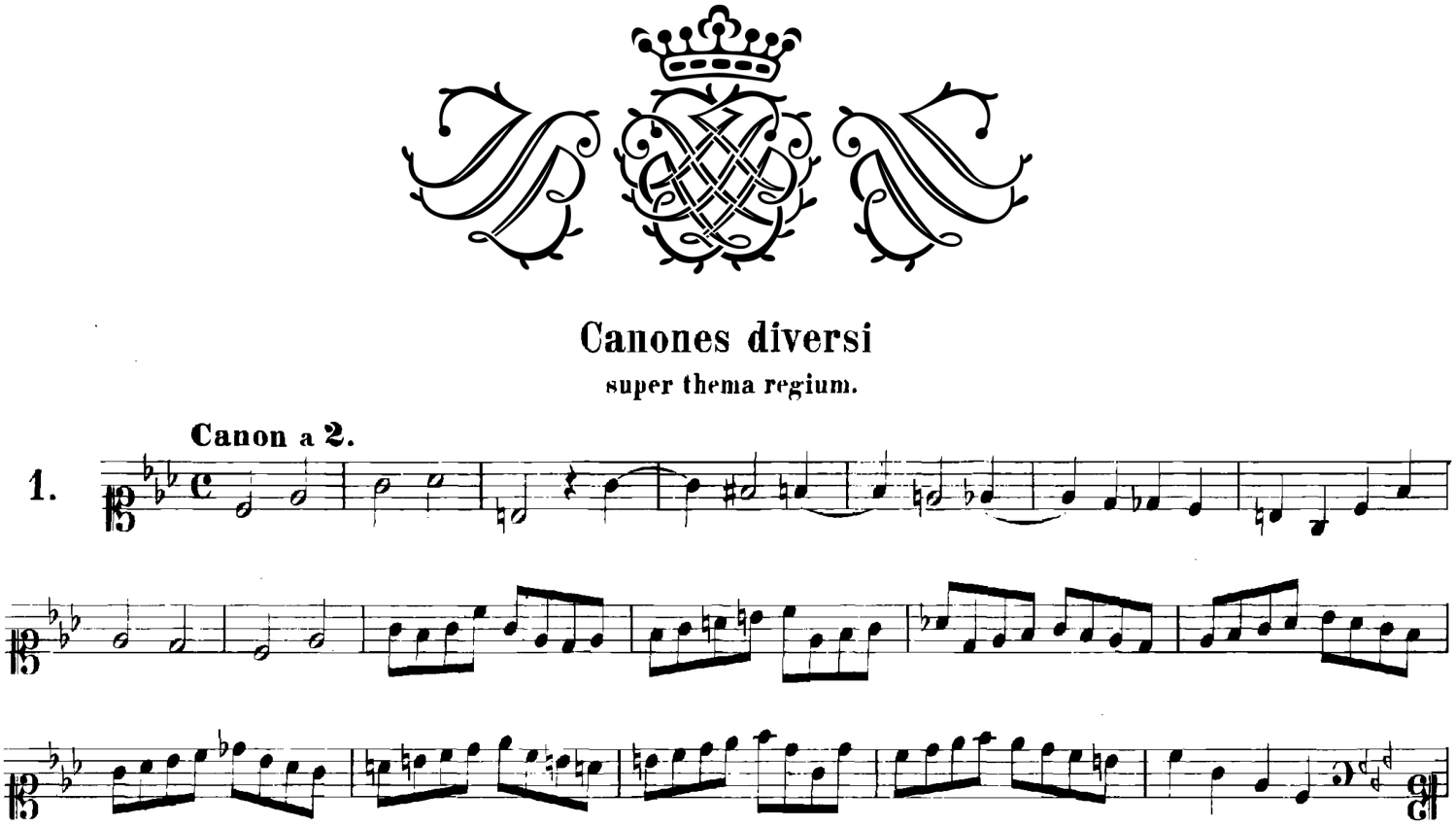
4.4 - As before, to hear the generated ABC run `abcShow` (next cell) and copy to the *clipboard* ; Then go to the online ABC editor , paste and play.

In [ ]:
```
abcShow
```

🔴Case study

Canon nr.1 of the *Canones diversi super thema regium* of **BWV 1079** by JS Bach (1685-1750):

Analogy:



First part of the canon - the *real theme* (a):

In [ ]:
```
a = royalTheme
---
abcPlayM "Eb" "C" a
```

In [ ]:
```
abcShow
```

Second part of the canon - the countertheme (b):

In [ ]:
```
b = counterTheme
---
abcPlayM "Eb" "C" b
```

In [ ]:
```
abcPlayM "Eb" "C" ( reverse ( a ++ b ))
```

4.5 - Build the entire canon, in two parts, knowing that the code `(a) ++ (b)` is in counterpoint with itself, but in retrograde movement. Show it in sheet music.

In [ ]:
```

```

NB : as above, to hear the generated ABC run `abcShow` , copy code `abc` that is shown to the *clip-board* and enter the online ABC editor to play it.

4.6 - Back to `mel` above, build in the next cell `contra` such that

> `(mel ++ contra) # reverse (mel ++ contra)`

make harmonious sense.

Preview the result and listen using `abcShow` (etc).

In [ ]:
```

```

In [ ]:
```
abcShow
```

4.7 - The following cell contains the 5 parts of the *Earle of Essex Galiard* by John Dowland (1563-1626) as previously worked on in ABC. Put it on sheet music, listen to it and try adding more notes to the most incomplete parts, using this PDF .

In [ ]:
```
cantus  =  [( "G" , 3  %  8 ),( "A" , 1  %  8 ),( "B" , 1  %  4 ),( "G" , 1  %  4 ),( "g" , 1  %  2 ),( "f"
bassus    =  [( "G," , 1  %  2 ),( "G," , 3  %  4 ),( " A," , 1  %  4 ),( "B," ,3  %  2 ),( "B,," , 1  %  2
                    ),( "B," , 1  %  4 ),( "G," , 1  %  4 ),( "A," , 3  %  2 ),( "A," , 1  %  4 )]
tenor    =  [( "d," , 1  %  2 ),( "g," , 1  %  2 ),( "d," , 3  %  4 ),( "c," , 1  %  4 ),( "B," , 1  %  2 )]
altus    =  [( "G" , 1  %  2 ),( "G" , 3  %  8 ),( "A" , 1  %  8 ),( "B" , 1  %  4 ),( "c " , 1  %  4 ),( "c
```

In [ ]:
```

```

🔴Repetition and imitation

4.8 - Can you identify which melody will be next, if in F major?

In [ ]:
```
fj  =  [( "F" , 1  %  4 ),  ( "G" , 1  %  4 ),  ( "A" , 1  %  4 ), ( "F" , 1  %  4 ), ( "F" , 1  %  4 ),
     ( "G" , 1  %  4 ),  ( "A" , 1  %  4 ),  ( "F" , 1  %  4 ), ( "A" , 1  %  4 ), ( "B" , 1  %  4 ),
     ( "c" , 1  %  2 ),  ( "A" , 1  %  4 ),  ( "B" , 1  %  4 ), ( "c" , 1  %  2 ), ( "c" , 3  %  16 ),
     ( "d" , 1  %  16 ), ( "c" , 1  %  8 ),  ( "B" , 1  %  8 ), ( "A" , 1  %  4 ), ( "F" , 1  %  4 ),
     ( "c" , 3  %  16 ),( "d" , 1  %  16 ),( "c" , 1  %  8 ),( "B" , 1  %  8 ),( "A" , 1  %  4 ),
     ( "F" , 1  %  4 ),  ( "F" , 1  %  4 ),  ( "C" , 1  %  4 ), ( "F" , 1  %  2 ), ( "F" ,1  %  4 ),
     ( "C" , 1  %  4 ),  ( "F", 1  %  2 )]
```

Make it play on the next cell (don't forget: in F major)

In [ ]:
```
abcPlayM  "F"  "C"  fj
abcShow
```

4.9 - The operation `delay t m` delays the melody line `m` `t` -time units. For example:

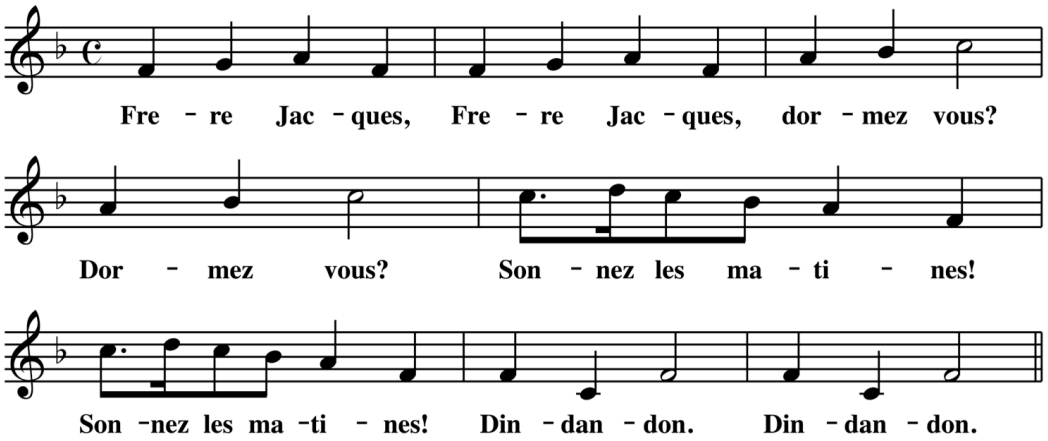
In [ ]:
```
( abcPlayM "F" "3/4" . P ) [ mel , delay ( 2 % 4 ) mel ]
```

Using `delay` , build the 4-voice (unison) canon so `fj` that the 4th voice only performs the melody once and the other voices never stop singing:

In [ ]:
```
v1 = undefined
v2 = undefined
v3 = undefined
v4 = undefined
canone = ( P [ v1 , v2 , v3 , v4 ])
---
abcPlayM "F" "C" canone
```

In [ ]:
```
abcShow
```

4.10 - The operation `a ⊗ p` is repeated `p` `a` several times, for example `2 ⊗ "ab" = "abab"` . Looking to



determine `i` standards `p0` in `p3` order to achieve equality:

In [ ]:

```
i  =  undefined
n  =  i  ⊗  p0  ++  i  ⊗  p1  ++  i  ⊗  p2  ++  i  ⊗  p3
---
n == m
```