MUSIC CHORDS CHARTS

IN ELM

Christophe Benz – christophe.benz@jailbreak.paris

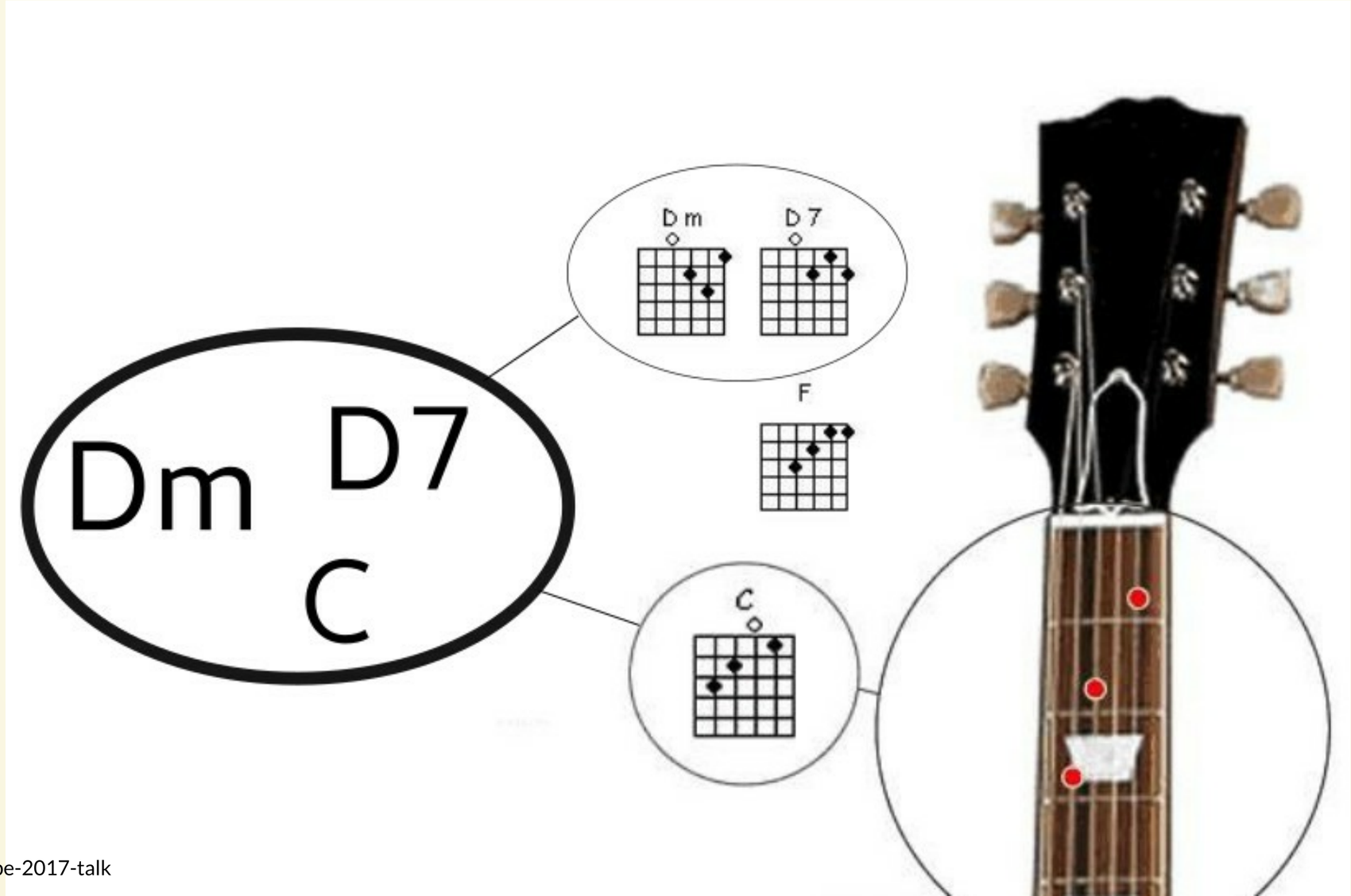
open-chords-charts / elm-europe-2017-talk

de la HUCHETTE
Pearl
MUSIC GALLERY
OPGEVAL • DREU • EVREUX

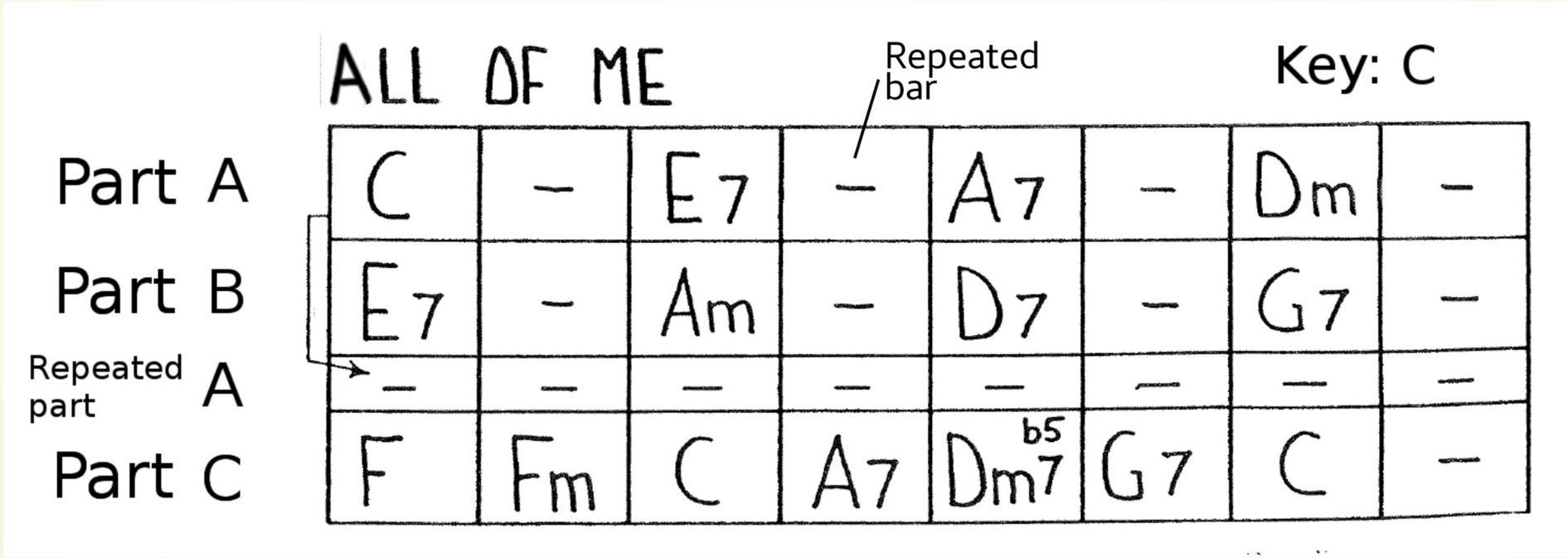
WHAT'S A SONG?

- song = melody + chords
- applies to rock, blues, jazz, folk

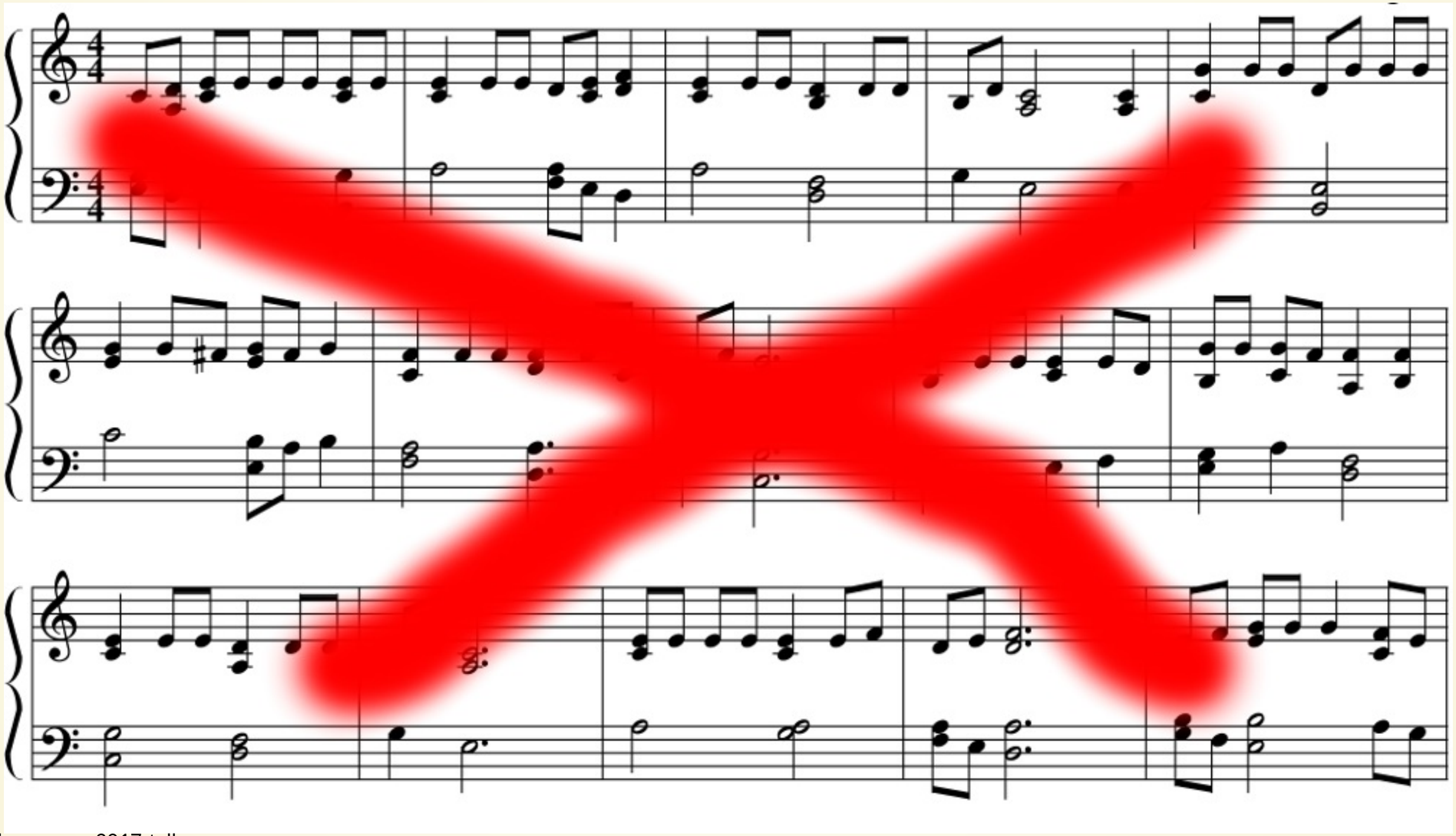
ALREADY SEEN THAT?



EXAMPLE: JAZZ SONG



NOT A MUSIC SCORE



CHORDS

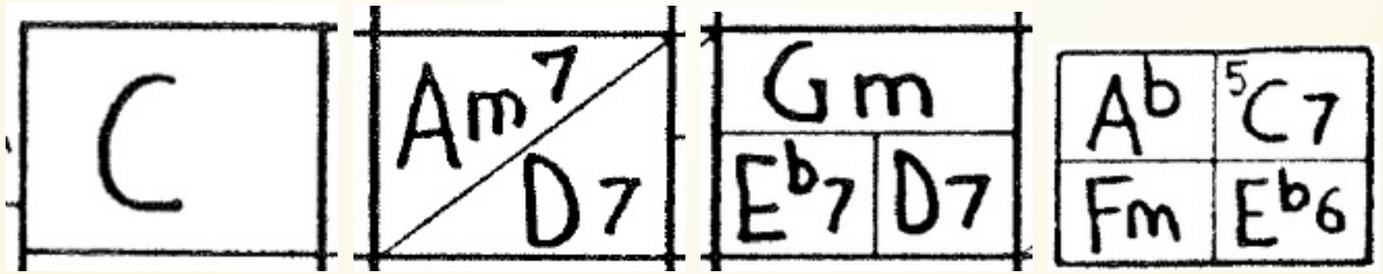
- a note and a quality
- C

  means "C Major"
- Fm

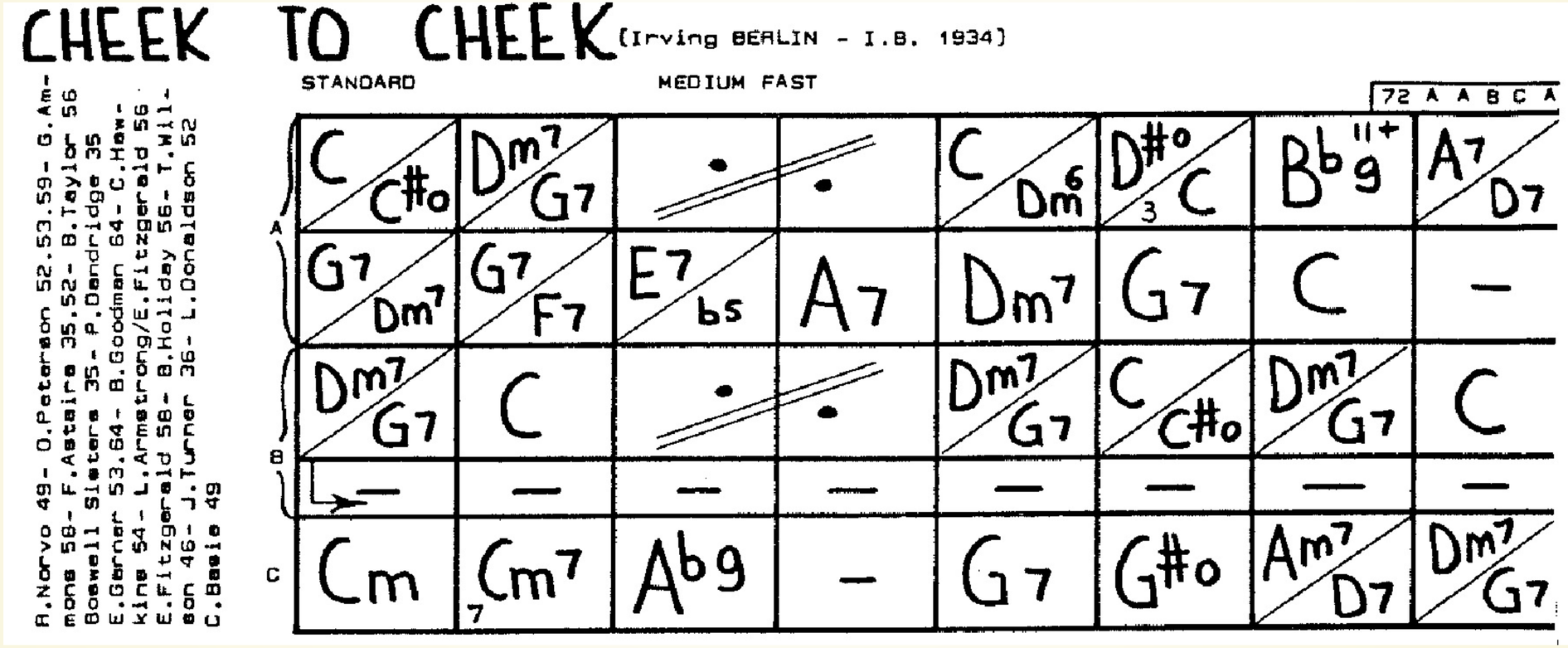
  means "F minor"

BARS

- 1 bar = 4 beats
- 1 to 4 chords per bar



COMPLEX EXAMPLE

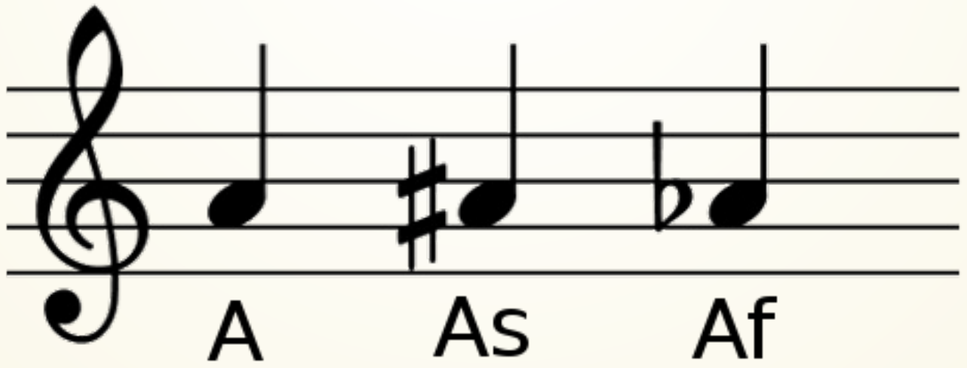


Music

PACKAGE IN ELM

NOTE TYPE

```
type Note
    = A | Af | As -- f = "flat"
    | B | Bf | Bs -- s = "sharp"
    | C | Cf | Cs
    ...
    | F | Ff | Fs
    | G | Gf | Gs
```



CHORD TYPE

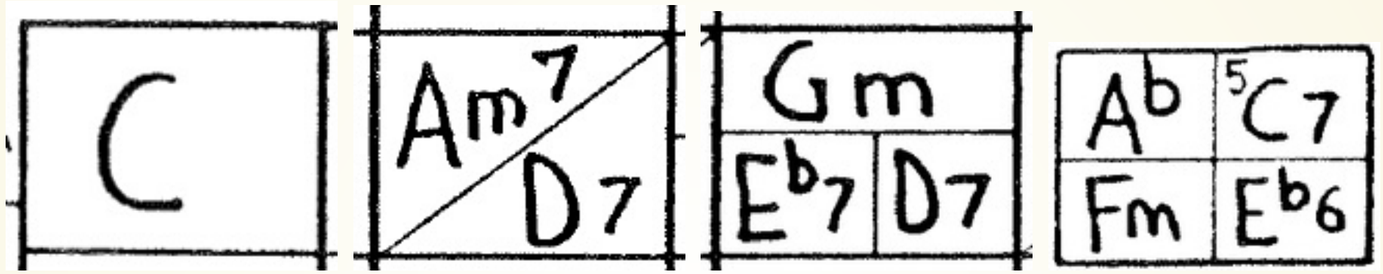
```
type alias Chord =
    ( Note, Quality )

type Quality
    = Major
    | Minor
    | Augmented
    | MajorSixth
    | MinorSixth
    | Seventh
    ...
```

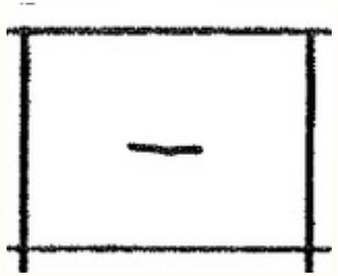
CHORD VALUE

Fm

```
fMinor : Chord
fMinor = ( F, Minor )
```

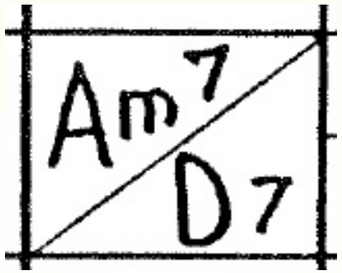
BAR TYPE





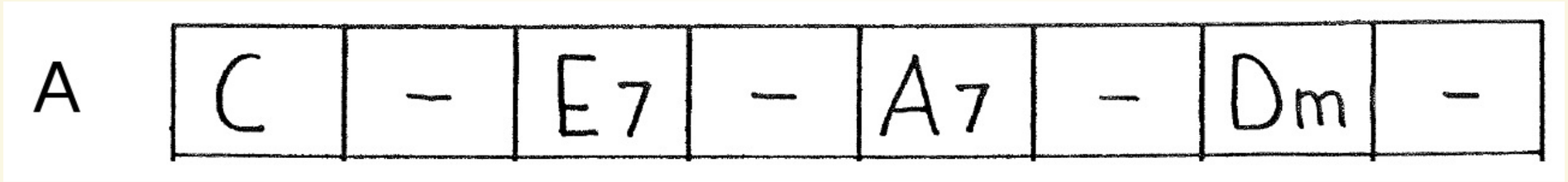
```
type Bar
    = Bar (List Chord)
    | BarRepeat
```

BAR VALUE



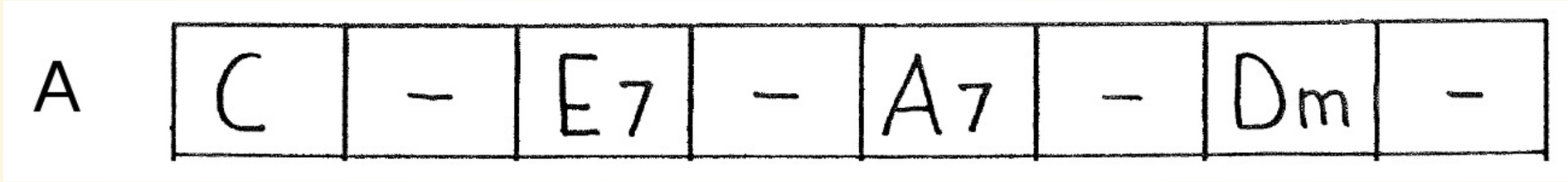

```
bar1 : Bar
bar1 =
    Bar
        [ ( A, MinorSeventh )
        , ( D, Seventh )
        ]
```

PART TYPE



```
type Part
    = Part String (List Bar)
    | PartRepeat String
```

PART VALUE




```
partA : Part
partA =
    Part "A"
        [ Bar [ ( C, Major ) ]
        , BarRepeat
        , Bar [ ( E, Seventh ) ]
        , BarRepeat
        , Bar [ ( A, Seventh ) ]
        , BarRepeat
        , Bar [ ( D, Minor ) ]
        , BarRepeat
        ]
```

CHART TYPE

TOP-LEVEL TYPE

```
type alias Chart =
    { title : String
    , key : Note
    , parts : List Part
    }
```

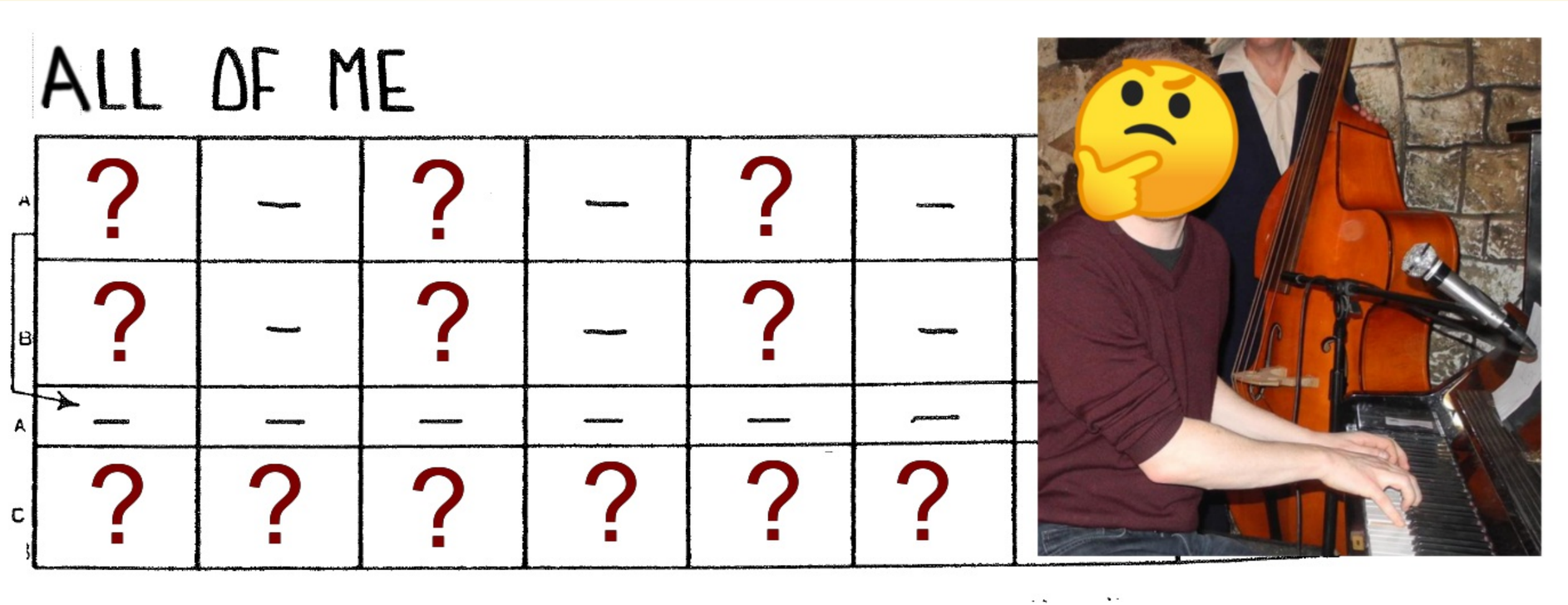
VIEW A CHART

The Elm Architecture

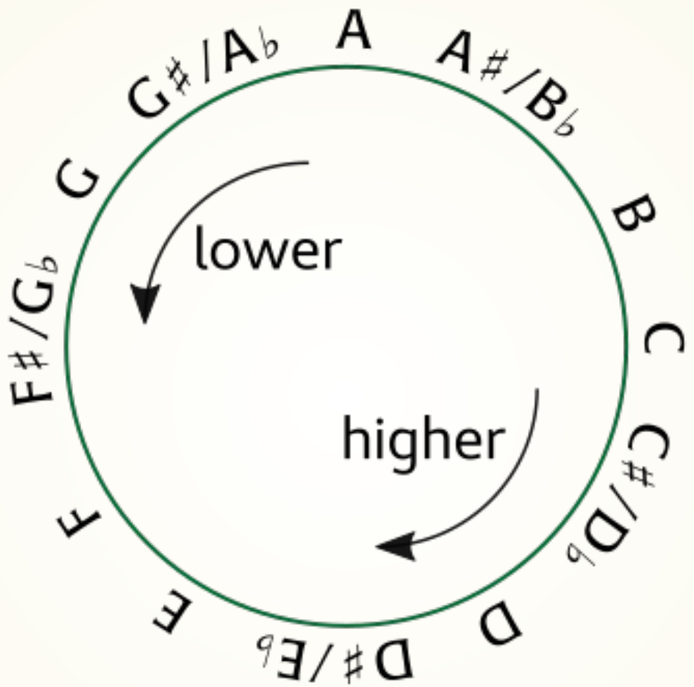
```
view : Chart -> Html msg
```

A	C	–	E7	–	A7	–	Dm	–
B	E7	–	Am	–	D7	–	G7	–
A	–	–	–	–	–	–	–	–
C	F	Fm	C	A7	Dø	G7	C	–

PLAY IT IN G!



TRANSPOSE



$$transpose(C, 1) \rightarrow D$$

$$transpose(G, 1) \rightarrow A$$

NOTE INDEXES

```
toIndex : Note -> Int

toIndex C = 3
toIndex G = 10
```

```
fromIndex : Int -> Note

fromIndex 3 = C
fromIndex 10 = G
```

```
transposeNote : Int -> Note -> Note
transposeNote interval note =
    fromIndex ((toIndex note) + interval)
```

NESTED PIECES

- recompute chords
- chart → part → bar → chord → note

REWIRE BOTTOM-UP

```
transposeChord : Int -> Chord -> Chord
transposeChord interval ( note, quality ) =
    ( transposeNote interval note, quality )

transposeBar : Int -> Bar -> Bar
transposeBar interval bar =
    mapBarChords (transposeChord interval) bar

transposePart : Int -> Part -> Part
transposePart interval part =
    mapPartBars (transposeBar interval) part
```

INTERVALS

$$interval(chartKey, newKey)$$

```
interval : Note -> Note -> Int
interval note1 note2 =
    (toIndex note2 - toIndex note1) % 12
```

```
toIndex C = 3
toIndex G = 10

interval C G = 7
```

TOP-LEVEL

```
transpose : Note -> Chart -> Chart
transpose newKey chart =
    let
        interval =
            Note.interval chart.key newKey

        newParts =
            chart.parts
                |> List.map (transposePart interval)
    in
        { chart | key = newKey, parts = newParts }
```

CHART VIEWER / EDITOR

DEMO

CHART VIEWER / EDITOR

All of me **C**

A	G	–	B7	–	E7	–	Am	–
B	B7	–	Em	–	A7	–	D7	–
A	–	–	–	–	–	–	–	–
C	C	Cm	G	E7	Aø	D7	G	–

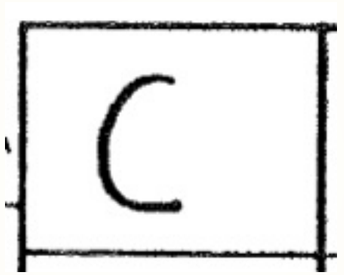
Transpose to: G ▾

A TEXT FORMAT

In Elm: verbose!

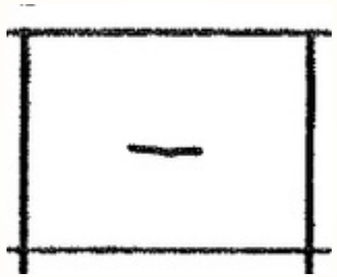
```
allOfMe : Chart
allOfMe =
    let
        partA = Part "A"
            [ Bar [ ( C, Major ) ]
            , BarRepeat
            ...
            ]
        partB = ...
        partC = ...
    in
        { title = "All of me"
        , key = C
        , parts = [ partA, partB, PartRepeat "A", partC ]
```

A CHORD

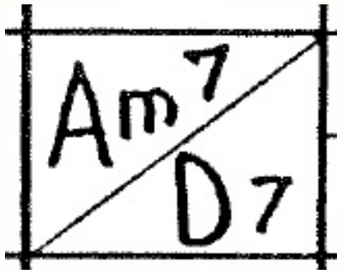


C

A REPEATED BAR

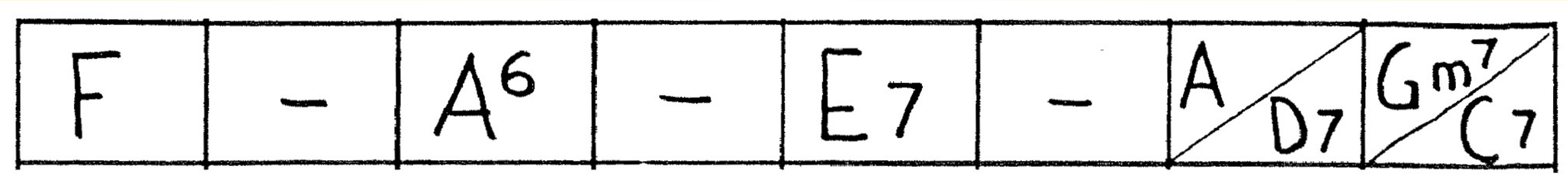


2 CHORDS IN A BAR



Am7/D7

A PART



= A

F - A6 - E7 - A/D7 GM7/C7

THE WHOLE CHART

```
title: All of me
key: C

= A
C - E7 - A7 - Dm -

= B
E7 - Am - D7 - G7 -

= A

= C
F Fm C A7 Dø G7 C -
```

CHART TO STRING

Using pattern matching

```
barToString : Bar -> String
barToString bar =
    case bar of
        Bar chords ->
            chords
                |> List.map Chord.toString
                |> String.join "/"

        BarRepeat ->
            "-"
```

STRING TO CHART

Using elm-tools/parser

```
chart : Parser Chart
chart =
    succeed Chart
        |. spacesAndNewlines
        |. symbol "title:"
        |. spaces
        |= keepUntilEndOfLine
        |. newLine
        |. symbol "key:"
        |. spaces
        |= note
        |. spacesAndNewlines
        |= repeat oneOrMore (part |. spacesAndNewlines)
        |. end
```

THANK YOU!

- Elm is awesome to model a domain like music
- refactoring is a real pleasure
- https://open-chords-charts.github.io/chart-editor/