Ecco! - uma linguagem computacional para descrição e estudo de temperamentos musicais

Lucas Bracher
lucasbracher@gmail.com

Resumo: A linguagem ecco! foi desenvolvida a partir da observação da dificuldade que músicos especializados em instrumentos históricos têm com o ferramental matemático necessário para descrever corretamente um temperamento musical. A partir da interação com a linguagem, um músico poderá descrever um temperamento usando apenas linguagem natural, sem a necessidade de usar operações de exponenciação e logaritmo, e poderá escutar o resultado do temperamento, bem como obterá a tabela de desvios em cents necessária para a programação de afinadores eletrônicos. O método utilizado foi o desenvolvimento da linguagem a partir da necessidade de facilitar o aprendizado dos aspectos matemáticos dos sistemas de temperamentos musicais e a observação direta dos estudantes destes sistemas ao utilizarem a mesma linguagem em comparação com os métodos existentes até hoje. A conclusão é que a linguagem ecco! facilitou o aprendizado e compreensão de aspectos dificilmente tangíveis de teoria de temperamentos, como as comas sintônicas e pitagórica, diferenciação de enarmônicos e iterações longas sobre o ciclo de quintas.

Palavras-chave: Temperamento Musical. Linguagem de programação. Interpretação historicamente orientada.

Title of the Paper in English: Ecco! - a Computational Language for Musical Temperaments Description and Study.

Abstract: ecco! language was developed from the observation of the difficulty that specialized musicians in historical instruments have with the necessary mathematical tools to correctly describe a musical temperament. From the interaction with language, a musician can describe a temperament using only natural language, without the need of use exponentiation and logarithm operations, and can hear the result of temperament, as well as obtain the table of deviations in cents necessary for the programming of electronic tuners. The method used was the development of language from the need to facilitate the learning of the mathematical aspects of musical temperament systems and the direct observation of the students of these systems when using the same language in comparison to the existing methods until today. The conclusion is that the language ecco! facilitated the learning and understanding of difficult tangible aspects of temperament theory, such as the syntonic and pythagorean commas, differentiation of enarmonics, and repeated long iterations over the cycle of fifths.

Keywords: Musical Temperament. Programming Languages. Historically Oriented Interpretation.

1. Introdução

Os sistemas de temperamentos musicais são sistemas que ditam como as notas musicais devem ser afinadas tendo em vista o privilégio de determinados intervalos musicais em detrimento de outros e considerando também que a boa afinação de um determinado intervalo causa imediatamente uma pequena deterioração da qualidade de outros intervalos. Tais deteriorações são historicamente conhecidas pelos teóricos que criaram sistematizações de afinação de instrumentos. Um exemplo disso é o sistema de afinação pitagórico, que consiste em quintas afinadas

puras (relação de frequência 3:2). Tal método de afinação gera automaticamente duas distorções bastante perceptíveis: a primeira é que fica perfeitamente audível a diferença entre um intervalo de 4 quintas justas consecutivas e um intervalo de 2 oitavas mais uma terça maior pura. A segunda é que também se torna audível a diferença entre 12 quintas justas consecutivas e 7 oitavas. Em ambos os casos, os intervalos feitos apenas com quintas justas acabam sendo ligeiramente mais agudos que aqueles construídos apenas com oitavas ou com oitavas e terças maiores. À diferença entre o intervalo de quatro quintas justas consecutivas e o intervalo de duas oitavas mais uma terça pura dá-se o nome de coma sintônica, e ao intervalo de 12 quintas justas consecutivas e o intervalo de 7 oitavas dá-se o nome de coma pitagórica.

Durante um período considerável da história da música ocidental registrada, tais comas ou suas frações foram largamente usadas para os mais diversos sistemas de afinação aplicados em diversos instrumentos. A tais sistemas se dá o nome genérico de "temperamento". O sistema temperado mais utilizado é o sistema mesotônico, que basicamente consiste em dividir uma determinada coma em várias frações iguais e distribuí-las através dos intervalos de quinta justa. O mais conhecido desses é o mesotônico regular, que pega a coma sintônica e a distribui em quatro partes iguais através de quatro quintas justas consecutivas. Desta forma, o intervalo de terça maior surge como consonância, e por outro lado, surge também a quinta do lobo (HORA, 2007), intervalo este que é inutilizável utilizado apenas como intervalo de efeito em passagens musicais cuja intenção é expressar dor ou desconforto. Outro temperamento que surge e que hoje é praticamente hegemônico é o temperamento igual, que divide a coma pitagórica em 12 partes iguais e os distribui através das 12 quintas consecutivas, fazendo assim com que as quintas se estreitem e as terças maiores se alarguem. A partir das dificuldades que surgem em se explicar para um aluno todo este arcabouço teórico, acaba se tornando premente a necessidade de uma ferramenta para o auxílio do ensino de temperamentos musicais.

A ideia da linguagem ecco! surgiu a partir da observação de que o estudo de temperamentos musicais necessariamente precisaria ser feito a partir de um instrumento musical que permite a execução de diversas notas musicais ou a partir de um gerador de frequências, sendo que o operador de tal gerador de frequências precisaria ter conhecimento de aritmética modular e das operações de logaritmo e exponenciação para operá-lo devidamente. A linguagem ecco! extingue tais necessidades, permitindo que uma pessoa calcule e ouça tais notas apenas usando

linguagem natural, sem a utilização de outros instrumentos de apoio ou de operações matemáticas.

Assim surge a linguagem ecco! como ferramenta de apoio pedagógico a partir da observação direta em sala de aula. E tal ideia surgiu quando se observou que os alunos tinham dificuldade de entender como soaria um intervalo de coma sintônica, um intervalo de coma pitagórica, um intervalo entre três terças maiores puras e uma oitava ou um intevalo entre 53 quintas justas puras e 31 oitavas. A partir desta observação e a partir do fato de que temperamentos musicais podem ser mapeáveis em um sistema esparso de equações de primeiro grau, surgiu a ideia de criar uma linguagem computacional que utilizasse um subconjunto da linguagem natural que fosse mapeável nestas equações de primeiro grau, cujas variáveis armazenariam os valores das notas em hertz e em cents e que permitisse a alteração destas notas em fração de comas arbitrárias, cents e batimentos por segundo. Após alguns esboços surgiu o primeiro protótipo funcional, cuja tela pode ser vista abaixo e cujo video, armazenado em canal anônimo, poderá ser visto na bibliografia (BRACHER, 2017[a]):

```
ecco!> B is 1 perfect fifth above E minus 1/4 of syntonic comma;
  A: 440.000000 hertz, 0.000000 cents
  B: 491.934955 hertz, 193.156857 cents
  E: 657.953464 hertz, 696.578428 cents
ecco!> F# is 1 perfect fifth above B minus 1/4 of syntonic comma;
  A: 440.000000 hertz, 0.000000 cents
  B: 491.934955 hertz, 193.156857 cents
  E: 657.953464 hertz, 696.578428 cents
  F#: 735.614335 hertz, 889.735285 cents
ecco!> C# is 1 perfect fifth above F# minus 1/4 of syntonic comma;
  A: 440.000000 hertz, 0.000000 cents
  B: 491.934955 hertz, 193.156857 cents
  C#: 550.000000 hertz, 386.313714 cents
  E: 657.953464 hertz, 696.578428 cents
  F#: 735.614335 hertz, 889.735285 cents
ecco!> play;
ecco!> play A C#;
ecco!> transpose 100 cents down;
  A: 415.304690 hertz, 0.000000 cents
  B: 464.324768 hertz, 193.156857 cents
  C#: 519.130872 hertz, 386.313714 cents
  E: 621.025373 hertz, 696.578428 cents
  F#: 694.327475 hertz, 889.735285 cents
ecco!> play A C#;
ecco!> drop;
ecco!> A is 440 hertz;
  A: 440.000000 hertz, 0.000000 cents
ecco!> C# is 1 major third above A;
  A: 440.000000 hertz, 0.000000 cents
  C#: 550.000000 hertz, 386.313714 cents
ecco!> X is 4 perfect fifths above A;
  A: 440.000000 hertz, 0.000000 cents
  C#: 550.000000 hertz, 386.313714 cents
  X: 556.875000 hertz, 407.820003 cents
ecco!> compare C# X;
21.5062895967
ecco!>
```

Figura 1: Exemplo de tela da linguagem ecco! mostrando as quatro primeiras notas de um sistema temperado em mesotônico de quarto de coma sintônica, com operações de execução de intervalo através de placa de som, transposição e exibição do valor da coma sintônica em cents através de operação direta.

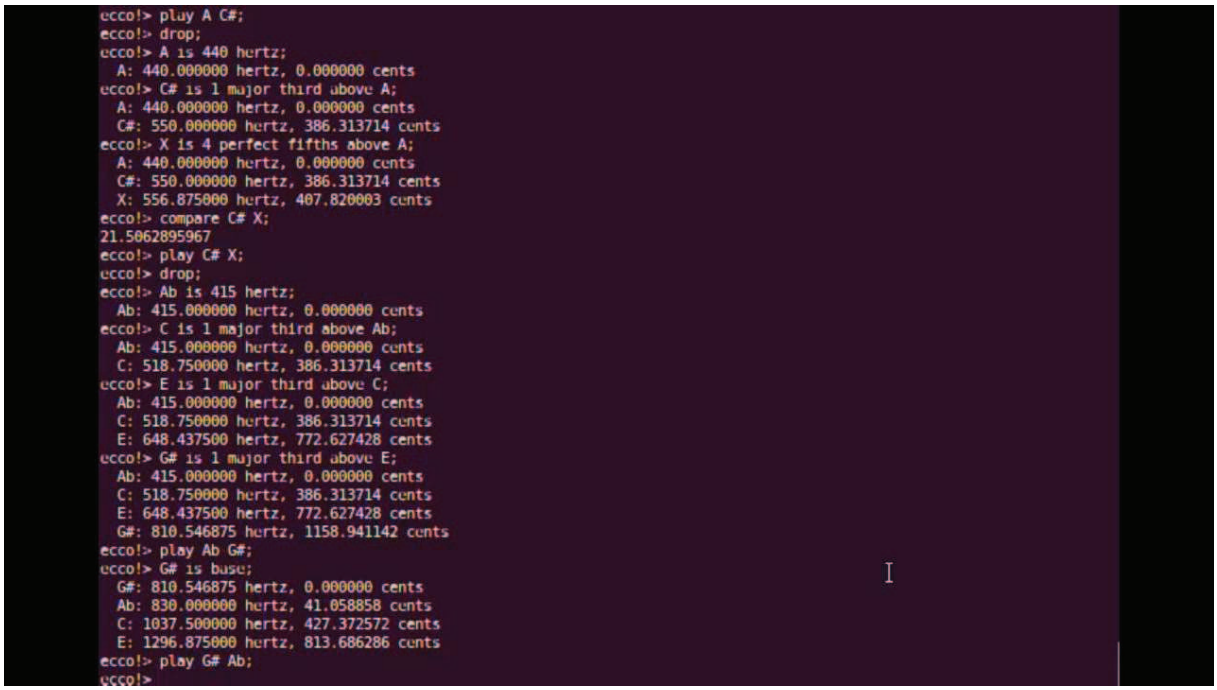


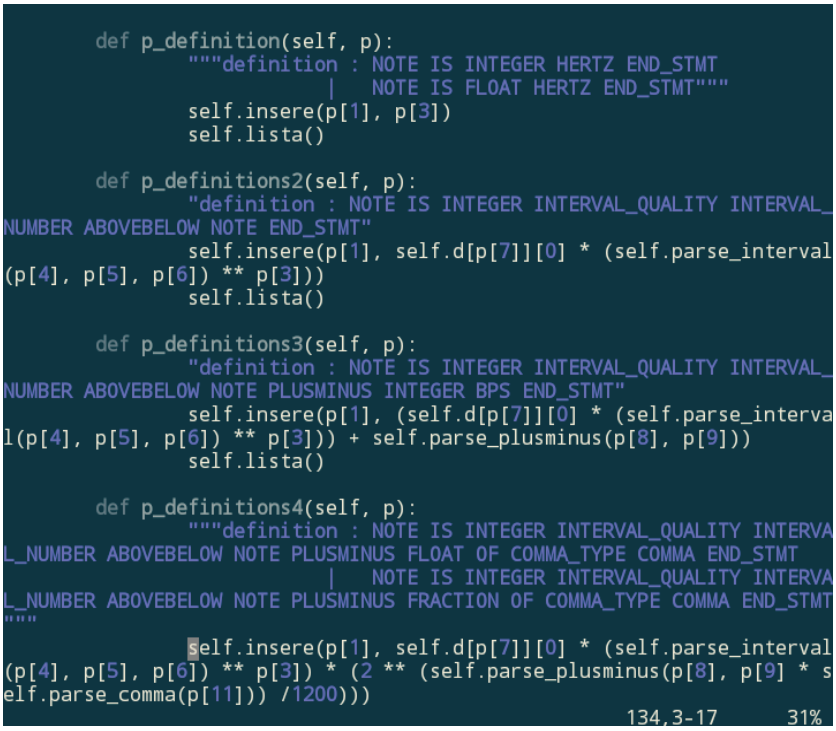
Figura 2: Exemplo de tela da linguagem ecco! mostrando a diferença entre três terços maiores consecutivas e uma oitava (a oitava é ligeiramente maior que 3 terças consecutivas)

2. Implementação

A linguagem ecco! foi implementada utilizando-se uma versão em Python dos geradores de lexer e parser Lex e Yacc (BRACHER, 2017b). O resultado foi uma linguagem que utiliza um parser LALR (BRACHER, 2017a) para processar cada declaração.

```
            return t

        t_HERTZ            = r'\bhertz\b'
        t_INTERVAL_NUMBER  = r'\b(third|fourth|fifth|sixth|octave|unison
)s?\b'
        t_INTERVAL_QUALITY = r'(major|minor|perfect)'
        t_ABOVEBELOW       = r'\b(above|below)\b'
        t_PLUSMINUS        = r'\b(plus|minus)\b'
        t_COMMA_TYPE       = r'\b(syntonic|pythagorean)\b'
        t_COMMA            = r'\bcommas?\b'
        t_BPS              = r'\bbps\b'
        t_CENT             = r'\bcents?\b'
        t_DROP             = r'\bdrop\b'
        t_COMPARE          = r'\bcompare\b'
        t_GENERATE         = r'\bgenerate\b'
        t_BASE             = r'\bbase\b'
        t_TRANSPOSE        = r'\btranspose\b'
        t_UPDOWN           = r'\b(up|down)\b'
        t_PLAY             = r'\bplay\b'
        t_TO               = r'\bto\b'
        t_RPAREN           = r'\)'
        t_LPAREN           = r'\('
        t_LOAD             = r'\bload\b'
        t_FILENAME         = r'[0-9A-Za-z_]+\.ecco'
        t_PRINT            = r'\bprint\b'
        t_ON               = r'\bon\b'
        t_OFF              = r'\boff\b'
        t_END_STMT         = r';'

        def t_error(self, t):
                                                    61,2-9        9%
```

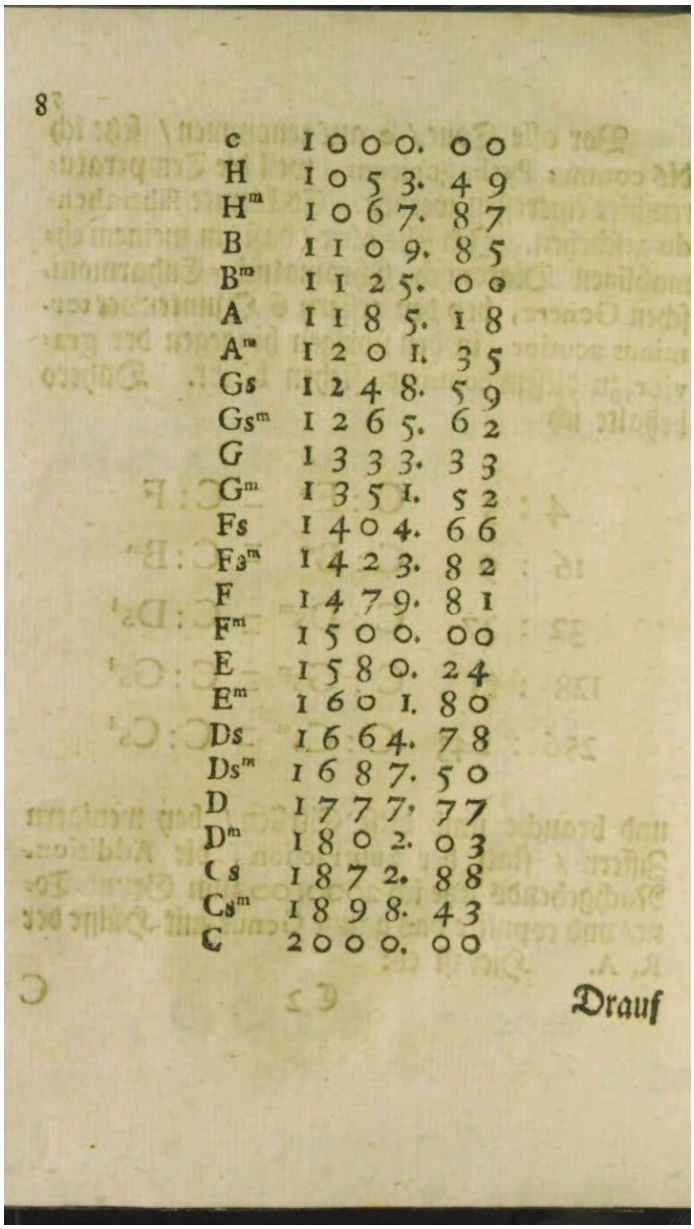
Figura 3: Definições de algumas tokens utilizadas pela linguagem ecco!



```
        def p_definition(self, p):
                """definition : NOTE IS INTEGER HERTZ END_STMT
                              |   NOTE IS FLOAT HERTZ END_STMT"""
                self.insere(p[1], p[3])
                self.lista()

        def p_definitions2(self, p):
                "definition : NOTE IS INTEGER INTERVAL_QUALITY INTERVAL_
NUMBER ABOVEBELOW NOTE END_STMT"
                self.insere(p[1], self.d[p[7]][0] * (self.parse_interval
(p[4], p[5], p[6]) ** p[3]))
                self.lista()

        def p_definitions3(self, p):
                "definition : NOTE IS INTEGER INTERVAL_QUALITY INTERVAL_
NUMBER ABOVEBELOW NOTE PLUSMINUS INTEGER BPS END_STMT"
                self.insere(p[1], (self.d[p[7]][0] * (self.parse_interva
l(p[4], p[5], p[6]) ** p[3])) + self.parse_plusminus(p[8], p[9]))
                self.lista()

        def p_definitions4(self, p):
                """definition : NOTE IS INTEGER INTERVAL_QUALITY INTERVA
L_NUMBER ABOVEBELOW NOTE PLUSMINUS FLOAT OF COMMA_TYPE COMMA END_STMT
                              |   NOTE IS INTEGER INTERVAL_QUALITY INTERVA
L_NUMBER ABOVEBELOW NOTE PLUSMINUS FRACTION OF COMMA_TYPE COMMA END_STMT
"""
                self.insere(p[1], self.d[p[7]][0] * (self.parse_interval
(p[4], p[5], p[6]) ** p[3]) * (2 ** (self.parse_plusminus(p[8], p[9] * s
elf.parse_comma(p[11])) /1200)))
                                                          134,3-17      31%
```

Figura 4: Trecho do parser LALR (BRACHER, 2017a) que define cada nota através de frequência em hertz, através de intervalo puro, através de intervalo temperado por batidas por segundo e intervalo temperado por fração ou por coma inteira.

A gramática foi toda baseada em um subconjunto do idioma inglês suficientemente capaz de descrever um temperamento por mais complexo que este seja, uma vez que cada nota de um temperamento pode ser mapeada em uma equação de primeiro grau e cada temperamento em um sistema esparso de equações de primeiro grau. Cada nota pode ser definida diretamente pela frequência e por relação com outra nota através de cents, intervalo puro (quinta justa, quarta justa, terças maiores e menores) e intervalo puro temperado por fração de coma, cents ou batimentos por segundo. A partir dos dados inseridos em linguagem natural, o parser da linguagem retira de cada expressão a informação necessária para para calcular a próxima nota a partir de uma nota anterior ou a partir da frequência em hertz. A partir da informação retirada de cada frase, o próprio sistema já exibe a frequência e os cents de cada nota calculada. Além disso, a linguagem também permite que se possa ouvir cada uma das notas calculadas em intervalos, acordes ou em separado. A linguagem também permite a definição de um temperamento através de comprimento de corda de monocórdio, sendo que este método tem fundamento histórico, e para nosso privilégio, algumas definições para monocórdio coletadas por Neidhardt (1724) e outros seus contemporâneos podem ser usadas diretamente na linguagem, sem auxílio de monocórdio ou outro instrumento.


8

c	1000. 00
H	1053. 49
H^{m}	1067. 87
B	1109. 85
B^{m}	1125. 00
A	1185. 18
A^{m}	1201. 35
Gs	1248. 59
Gs^{m}	1265. 62
G	1333. 33
G^{m}	1351. 52
Fs	1404. 66
Fs^{m}	1423. 82
F	1479. 81
F^{m}	1500. 00
E	1580. 24
E^{m}	1601. 80
Ds	1664. 78
Ds^{m}	1687. 50
D	1777, 77
D^{m}	1802. 03
Cs	1872. 88
Cs^{m}	1898. 43
C	2000. 00

Drauf

Figura 5: temperamento registrado por Neidhardt em seu livro Sectio Canonis Harmonici, dedicado ao estudo de temperamentos para instrumento de teclado. Os números aqui se referem a um comprimento arbitrário marcado na escala do monocórdio.

3. Utilização Prática

A partir da liberação de um primeiro protótipo da linguagem para alguns estudantes de instrumento de teclado históricos, constatou-se que este se sentindo à vontade para reproduzir temperamentos históricos, alterar e criar seus próprios temperamentos. Algumas sugestões foram acolhidas, como por exemplo, a informação mais clara de erros, carregamento de temperamentos a partir de arquivos, utilização de formas de onda senoidal, triangular, quadrada e dente de serra, e a utilização de histórico de comandos para repetição de comandos já dados e alteração para aqueles que são parecidos.

A linguagem ecco! Será disponibilizada livremente sob a licença GPL2 e sua url de disponibilização será informada em breve.

Referências:

BRACHER, Lucas. **ecco! Uma Linguagem para Descrição de Temperamentos**. 2017a. Youtube. Disponível em: <https://youtu.be/Qa-uRBhQ3f8>. Acesso em: 30 out. 2017.

BRACHER, Lucas. **ecco! Uma Linguagem para Descrição de Temperamentos - aspectos computacionais**. 2017b. Youtube. Disponível em: <https://youtu.be/vhuN4ugs1Es>. Acesso em: 30 out. 2017.

NEIDHARDT, J. G. **Sectio canonis harmonici**. 1724. Disponível em <http://imslp.org/wiki/Sectio_canonis_harmonici_(Neidhardt%2C_Johann_Georg)>. Acesso em: 30 out. 2017.

HORA, E. P. A Afinação Mesotônica do século XVII e sua aplicação prática. **ICTUS**, Salvador: UFBA, v. 8 n. 2, p. 49-64, 2007.