



## **Lewin: desenvolvimento de uma calculadora de classes de altura com funções analíticas e composicionais**

MODALIDADE: COMUNICAÇÃO

*Hildegard Paulino Barbosa*  
UFPB – hildegardpaulino@gmail.com

*Raphael Sousa Santos*  
UFCG – mail@raphaelss.com

*Liduíno José Pitombeira de Oliveira*  
UFRJ – pitombeira@musica.ufrj.br

**Resumo:** Descrevemos, neste artigo, as funcionalidades e características da calculadora Lewin, uma ferramenta útil para a operação com conjuntos de classes de alturas, tanto no âmbito analítico quanto composicional. Uma avaliação do estado da arte nesse campo de pesquisas é apresentada juntamente com um quadro comparativo entre as calculadoras disponíveis.

**Palavras-chave:** Calculadora Analítica e Composicional. Classes de conjuntos de classes de alturas. Derivação Serial. Multiplicação.

### **Lewin: the Development of a Pitch Class Calculator with Analytical and Compositional Functions**

**Abstract:** We describe in this paper, the functionality and features of the Lewin calculator, a useful tool for operation with pitch class sets, both in the analytical and compositional realms. An assessment of the state of the art in this field of research is presented along with a comparison chart between the available calculators.

**Keywords:** Analytical and Compositional Calculator. Set Classes. Serial Derivation. Multiplication.

As operações com conjuntos de classes de alturas, tanto no âmbito analítico quanto composicional, implicam na execução de diversas tarefas algorítmicas repetitivas, como, por exemplo, a obtenção da forma prima ou do vetor intervalar de um determinado conjunto. A execução dessas tarefas manualmente é propensa a erros, além de demandar um tempo considerável. Diante disso, a automatização computacional se apresenta como uma excelente alternativa, já que a máquina é mais adequada para a execução de tarefas dessa natureza do que o ser humano, de modo a acelerar os procedimentos e diminuir os possíveis erros, deixando a mente do compositor e do analista livre para pensar em aspectos estruturais mais importantes.

Visando alcançar esse objetivo, diversos pesquisadores criaram aquilo que se pode chamar de calculadora de classes de alturas, que consiste simplesmente no agrupamento de um conjunto de funcionalidades relacionadas aos procedimentos de operação com conjuntos de classes de alturas, em um único aplicativo computacional. Porém, a maior parte

desse aplicativos não dispõe de algumas funcionalidades importantes (tais como a geração de paletas de classes de conjuntos ou tabelas de adição, por exemplo) ou apresentam pouca usabilidade no manuseio da interface gráfica de usuário (GUI), fazendo com que o compositor ou analista ainda tenham que executar diversos cálculos manualmente. Diante desse cenário, propomos uma nova calculadora composicional, diferente das que já foram criadas, com um conjunto suficiente de funcionalidades (algumas não disponíveis em nenhuma das anteriores) e com boa usabilidade, visando minimizar ao máximo a execução manual de tarefas repetitivas e com uma interface amigável o suficiente para garantir uma boa curva de aprendizagem. Neste artigo, examinaremos o estado da arte das pesquisas nessa área e, em seguida, descreveremos detalhadamente a calculadora que batizamos de Lewin, em homenagem ao teórico e compositor americano David Lewin (1933-2003), formulador da teoria transformacional.

Examinaremos brevemente alguns projetos de calculadoras que operam com conjuntos de classes de altura como ferramenta auxiliar para a análise e para a composição musical. O primeiro é chamado *PC Set Calculator (PCSC)*, desenvolvido por David Walters. Trata-se de um projeto *Web*<sup>1</sup>, cuja interface gráfica pode ser vista na Fig. 1.

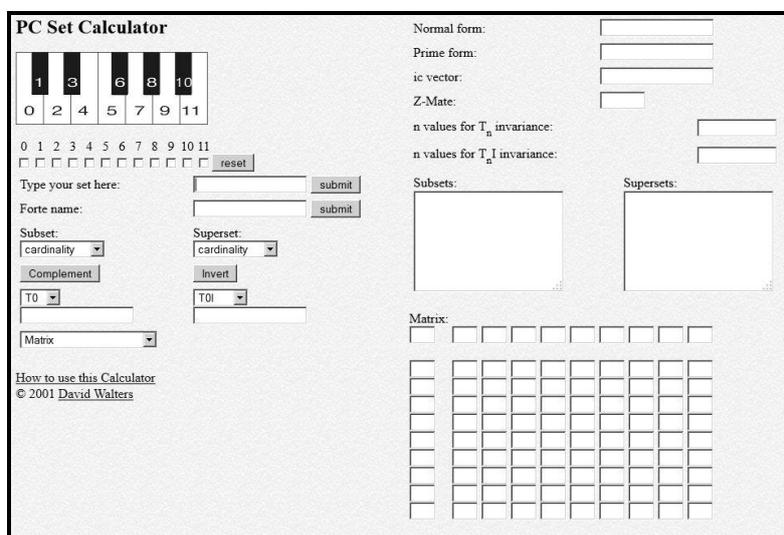


Figura 1: PC Set Calculator (Fonte: David Walters, 2001)

Como se pode observar, essa calculadora oferece não somente muitas funcionalidades (resultados, saída), mas também muitas formas de entrada: digitando o conjunto no formato original na caixa de texto, digitando o conjunto na nomenclatura de Forte, clicando nos componentes *CheckBox* ou no teclado musical, no canto superior esquerdo da tela. O usuário tem a opção de usar aquilo que lhe for mais conveniente. Porém, como é possível constatar, a disposição dos componentes, de entrada e de saída, não proporciona uma

leitura fluente dos resultados, possivelmente por apresentar todas as funcionalidades em uma única tela. Além disso, a obtenção de alguns resultados nos parece muito confusa, como os subconjuntos, superconjuntos e a matriz, nos quais o usuário deve selecionar um dos itens de um componente *ComboBox*, ao invés de clicar em um botão. Por fim, embora tenham sido disponibilizadas várias saídas provenientes de um conjunto de entrada, a quantidade de resultados é ainda tímida em relação à quantidade de informações que a teoria de conjuntos de classes de alturas pode oferecer.

O segundo projeto a ser analisado é denominado *Java Set Theory Machine (JSTM)*, desenvolvido por Jay Tomlim, e é também disponível *online*<sup>2</sup>. Esse aplicativo foi desenvolvido como um *applet* (da plataforma Java), que pode ser baixado para uso *offline*. Sua interface gráfica inicial pode ser vista na Fig. 2.

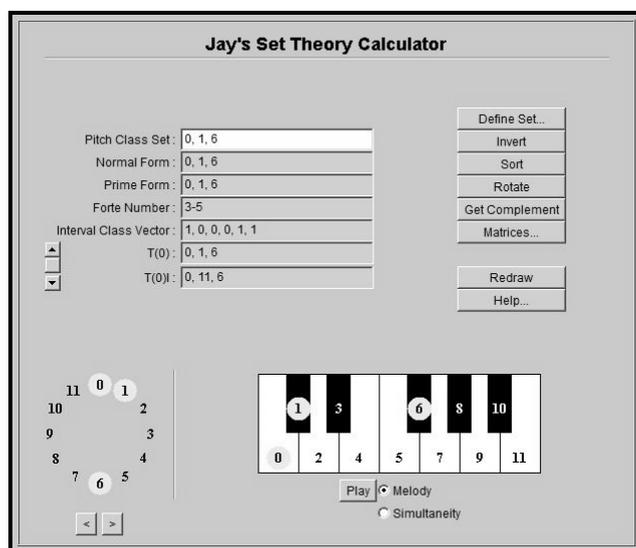


Figura 2: Java Set Theory Machine (Fonte: Jay Tomlim, 2009)

A entrada pode ser digitada na caixa de texto rotulada *Pitch Class Set* ou escolhida a partir de vários conjuntos padronizados, utilizando o botão *Define Set*. Abaixo da caixa de texto de entrada, que pode ser editada a qualquer momento, várias outras contêm informações de saída. Para atualizar as informações de saída, é necessário apertar o botão *Enter* do teclado do computador. Os botões do lado direito possibilitam ao usuário manipular o conjunto de entrada (o que interfere automaticamente nas informações de saída). Os botões (setas) do lado esquerdo dos rótulos de  $T(0)$  e  $T(0)I$  (transposição e inversão) permitem a exibição de uma transposição ou inversão acima ou abaixo da atual. O botão *Matrices* exibe uma matriz quadrada transpositiva ou inversiva com a mesma cardinalidade do conjunto de entrada. Outras características disponíveis são o círculo de classes de alturas e a representação

do conjunto de entrada em um teclado de piano, que ainda permite a reprodução desse conjunto. Esse projeto se mostra mais organizado que o anteriormente descrito. Porém, o número de funcionalidades disponíveis é ainda muito pequeno. Além disso, boa parte dessas funcionalidades consiste em nada mais do que processar a entrada de alguma forma, isto é, ordenar, rotacionar ou inverter. O número de informações novas que se pode obter a partir de um conjunto de entrada específico é ainda menor, de modo que esse projeto se mostra pouco arrojado para a composição.

O terceiro projeto a ser analisado, Processador de Classes de Notas (PCN), desenvolvido por Jmary Oliveira, é o que disponibiliza o maior número de informações a ser obtido a partir de um conjunto de entrada. Diferente dos anteriores, ele não funciona em um *website*, mas sim em um arquivo executável do Microsoft Windows, gratuitamente distribuído<sup>3</sup>. A interface gráfica da versão 98 desse aplicativo é exibida na Fig. 3. Como pode ser visto nessa figura, sua interface, além de organizada, é a que mais se parece com uma calculadora (mais até do que a proposta neste artigo) e o número de funcionalidades é consideravelmente grande. O visor contém, além da entrada, a identificação da funcionalidade executada (pelas iniciais) e a saída.

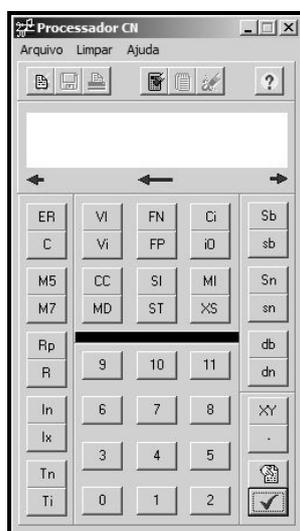


Figura 3: Processador de Classes de Notas

A grande limitação desse projeto é não estar ainda disponível em várias plataformas. É certo que o sistema Windows ainda é o mais popular do mundo, mas outras plataformas, a exemplo do Linux e OS da Apple, estão em número sempre crescente de usuários (hoje é comum um mesmo usuário ter máquinas com dois ou três dos sistemas citados, ou ainda uma mesma máquina ter Windows e Linux), mas mesmo essa limitação tende a diminuir com as chamadas máquinas virtuais – programas que permitem que outros

programas projetados para funcionar em uma plataforma, funcionem em outra. Outra limitação é a não disponibilização de algumas funcionalidades que o programa proposto neste documento disponibiliza, as quais serão descritas juntamente com as funcionalidades em comum com o PCN.

O projeto da calculadora Lewin (Fig. 4), desenvolvido em Java, surgiu inicialmente da necessidade de automatizar procedimentos que gerassem séries derivadas a partir de um conjunto de classes de alturas específico, construíssem matrizes dodecafônicas, tendo como base estas séries, e identificassem combinatoriedade e invariância nas matrizes.

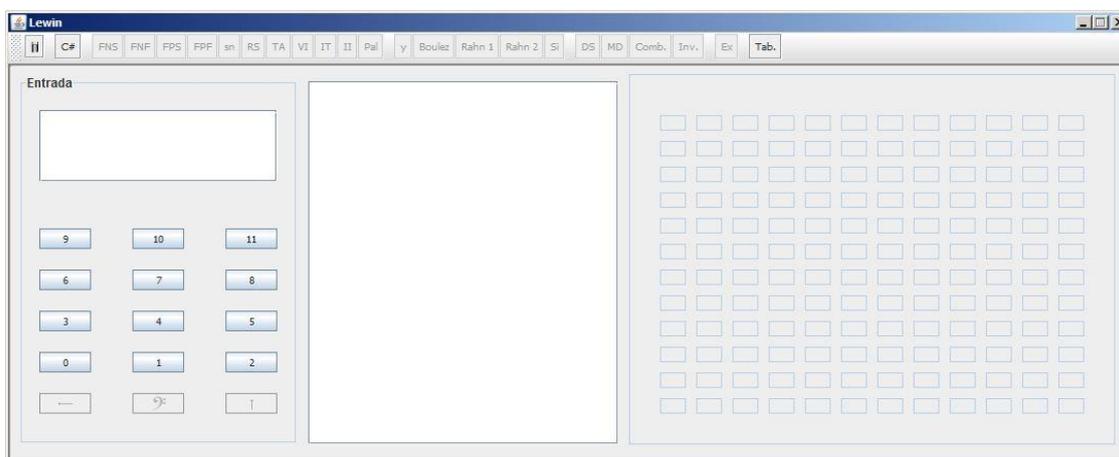


Figura 4: Interface Gráfica de Usuário da calculadora Lewin

Como sabemos, o conceito de derivação serial é intimamente associado à produção composicional de Anton Webern, cuja música tem “uma alta concentração motivica” (STRAUS, 2000:179). A derivação serial consiste na geração de uma série dodecafônica, cujos tricordes ou tetracordes discretos são membros da mesma classe de conjuntos de classes de alturas. Um exemplo clássico de derivação serial pode ser encontrado no *Concerto para Nove Instrumentos*, Op. 24, de Webern, cuja série é inteiramente formada pela justaposição de quatro tricordes [014]<sup>4</sup>. Outro compositor que se utilizou desse princípio de construção serial no planejamento de suas obras foi Milton Babbitt. A função derivação serial (DS), na calculadora Lewin, gera, quando possível<sup>5</sup>, séries dodecafônicas, cujos tricordes ou tetracordes pertencem à mesma classe de conjuntos. Uma vez geradas as séries, o usuário pode escolher uma delas para produzir uma matriz dodecafônica, através da função MD. Gerada essa matriz, é possível determinar o grau de combinatoriedade e invariância através das funções Comb e Inv, respectivamente. Na Tab. 1 são relacionadas as funções operacionais até o presente momento em comparação com funcionalidades oferecidas pelas outras calculadoras citadas neste artigo.

Função	Descrição	PCSC	JSTM	PCN	Lewin
<b>C#</b>	Alterna entre notas musicais e números			✓	✓
<b>FNS</b>	Forma normal segundo Straus	✓	✓	✓	✓
<b>FNF</b>	Forma normal segundo Forte				✓
<b>FPS</b>	Forma prima segundo Straus	✓	✓	✓	✓
<b>FPF</b>	Forma prima segundo Forte				✓
<b>sn</b>	Subconjuntos de $n$ elementos	✓		✓	✓
<b>RS</b>	Rotação Stravinskyana				✓
<b>TA</b>	Tabela de adição				✓
<b>VI</b>	Vetor intervalar	✓	✓	✓	✓
<b>IT</b>	Invariância transpositiva				✓
<b>II</b>	Invariância inversiva				✓
<b>Pal</b>	Paleta de classes de conjuntos				✓
<b>y</b>	Entrada de um segundo conjunto				✓
<b>Boulez</b>	Multipliação segundo Boulez				✓
<b>Rahn 1</b>	Multipliação segundo Rahn (tradicional)			✓	✓
<b>Rahn 2</b>	Multipliação segundo Rahn (expandida)				✓
<b>Si</b>	Similaridade entre conjuntos				✓
<b>DS</b>	Derivação serial				✓
<b>MD</b>	Matriz dodecafônica	✓		✓	✓
<b>Comb</b>	Combinatoriedade				✓
<b>Inv.</b>	Invariância				✓
<b>Ex</b>	Exportar para HTML				✓
<b>Tab</b>	Tabelas de classes de conjuntos				✓

Tabela 1: Quadro comparativo entre as calculadoras a partir das funcionalidades da calculadora Lewin

Descreveremos as características da calculadora Lewin, incluindo uma explanação sucinta de cada função. Conforme exibida na Fig. 4, a interface gráfica de usuário consiste basicamente de uma barra de ferramentas, na qual se encontram as funções, de dois visores (no menor são exibidos os dados de entrada e no maior, saídas de operações com conjuntos e resultados de derivação serial), de uma tabela para exibição de uma matriz dodecafônica e de um teclado de classes de alturas, indicadas com números ou notas (essa alternância é feita através da função C#). Usando as funções FNS e FNF é possível obter, respectivamente, a forma normal segundo Straus e segundo Forte de um conjunto de classes de altura. Usando as funções FPS e FPF obtém-se a forma prima segundo Straus/Rahn e segundo Forte, respectivamente. Como se sabe, Allen Forte (1973:4) propõe que se calcule a forma normal detectando a estrutura mais compacta partindo da esquerda, enquanto Rahn (1980:37-38) e Straus (2000:30-51) partem da direita. Essa diferença de procedimento causa diferença na forma prima em algumas classes de conjuntos. Por exemplo, o conjunto  $\{2,3,7,8,A\}$ <sup>6</sup> tem forma prima 01568, segundo Straus/Rahn e forma prima 01378, segundo Forte. A Tab. 2 ilustra todas as divergências.

A função sn calcula os subconjuntos de  $n$  elementos de um conjunto de classes de alturas. A função RS produz a rotação Stravinskyana de uma série dodecafônica, que consiste em manter fixo o primeiro hexacorde ( $\alpha$ ) e rotacionar o segundo hexacorde ( $\beta$ ), transpondo as

rotações resultantes ( $\beta_1$  a  $\beta_6$ ) de tal forma que iniciem todas com a altura do segundo hexacorde<sup>7</sup>. Uma das séries que Stravinsky utiliza no *Requiem Canticles* é: Fá, Dó, Si, Lá, Lá#, Ré, Dó#, Ré#, Sol#, Fá#, Mi, Sol. As seis primeiras alturas correspondem ao hexacorde  $\alpha$  e o segundo ao hexacorde  $\beta$ . Ao entrarmos com esta série na calculadora Lewin e escolhermos a função RS, aparecem no segundo visor o hexacorde  $\alpha$  e as seis rotações de  $\beta$  já na transposição correta (Fig. 5).

Straus/Rahn	Forte
[01568]	[01378]
[023679]	[013689]
[014579]	[013589]
[0145679]	[0234589]
[0125679]	[0124789]
[0134578A]	[0124579A]

Tabela 2: Divergências nas formas primas, segundo Straus/Rahn e Forte

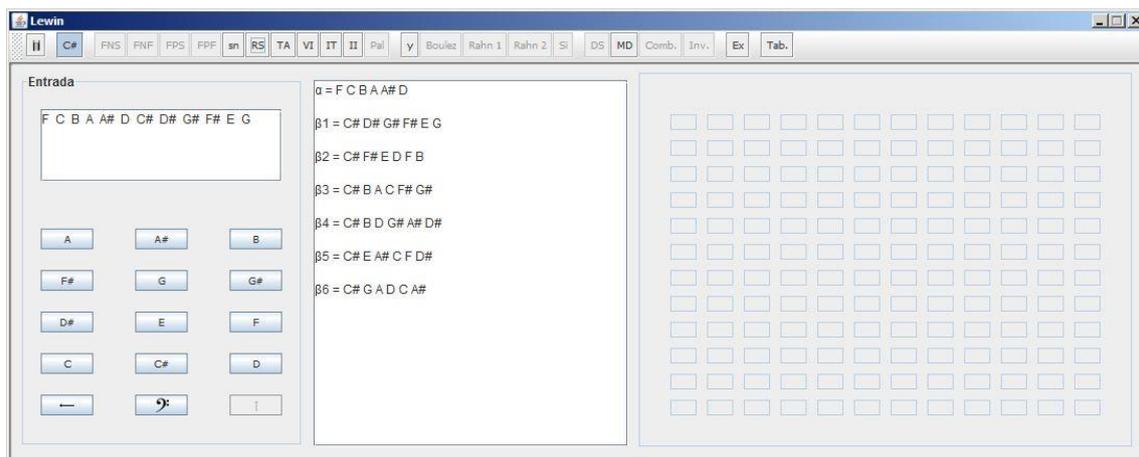


Figura 5: A calculadora Lewin gerando rotações Stravinskyanas

A função TA gera a tabela de adição, uma ferramenta útil para determinar notas comuns em TnI (inversão). A função VI gera o vetor intervalar, enquanto as funções IT e II informam sobre invariância sob transposição e inversão respectivamente. A função Pal é muito útil na geração de um repositório de conjuntos pertencentes à mesma classe. O botão y permite inserir outro conjunto e realizar comparações, tais como o cálculo de similaridade entre conjuntos (função Si). Três funções operam com a multiplicação entre dois conjuntos: Boulez, Rahn1 e Rahn2. A multiplicação Boulez consiste na transposição das notas de um conjunto para níveis de altura definidos por outro conjunto. Assim, por exemplo, o conjunto  $X = \{\text{Ré, Mi}\}$  multiplicado pelo conjunto  $Y = \{\text{Dó, Fá, Sol}\}$  aplica o intervalo de segunda maior (conjunto X) a cada nota de Y, produzindo o conjunto  $Z = \{\text{Dó, Ré, Fá, Sol, Lá}\}$ ,



excluídas as repetições. A multiplicação segundo Rahn1, se baseia em Rahn (1980:53-54) e consiste basicamente em multiplicar (aritmeticamente) um conjunto de classes de alturas por uma constante qualquer. Assim, por exemplo, o conjunto  $A = \{\text{Dó}, \text{Ré}, \text{Mi}\}$  multiplicado por 2 produz o conjunto  $B = \{\text{Dó}, \text{Mi}, \text{Sol}\# \}$ . A multiplicação Rahn2 é uma expansão desta operação que utiliza outro conjunto como segundo membro da multiplicação, ao invés de apenas uma constante. Assim, a multiplicação do conjunto A pelo conjunto  $C = \{\text{Dó}\#, \text{Ré}\# \}$  produz o conjunto  $D = \{\text{Dó}, \text{Ré}, \text{Sib}\}$ . Os resultados das operações podem ser exportados para o formato HTML e daí serem copiados e reutilizados em um editor de textos, por exemplo. A calculadora Lewin ainda oferece o conjunto de tabelas de tricordes, tetracordes, pentacordes e hexacordes.

Descrevemos neste artigo as características da calculadora Lewin, bem como realizamos uma sucinta comparação com outras calculadoras disponíveis. Este é um projeto aberto à implementação de novas características e funções sugeridas pelos usuários da composição e da análise. Assim, a calculadora está disponível *online* gratuitamente<sup>8</sup> e esperamos que os usuários possam identificar problemas (inclusive na interface gráfica), sugerir novas funções e utilizar a calculadora em seus trabalhos analíticos e composicionais.

## Referências:

- FORTE, Allen. *The Structure of Atonal Music*. New Haven: Yale University Press, 1973.  
RAHN, John. *Basic Atonal Theory*. New York: Schirmer Books, 1980.  
STRAUS, Joseph. *Introduction to Post-Tonal Theory*. 2.Ed. Upper Saddle River, NJ: Prentice Hall, 2000.

## Notas

---

<sup>1</sup> [http://www.mta.ca/faculty/arts-letters/music/pc-set\\_project/calculator/pc\\_calculate.html#](http://www.mta.ca/faculty/arts-letters/music/pc-set_project/calculator/pc_calculate.html#)

<sup>2</sup> <http://www.jaytomlin.com/music/settheory/>

<sup>3</sup> <http://www.angelfire.com/music2/bahia/pcn/pcn2001ptb.htm>

<sup>4</sup> Neste artigo utilizamos a forma prima entre colchetes para indicar a classes de conjuntos de classes de alturas, em vez da nomenclatura de Forte.

<sup>5</sup> É impossível produzir uma série derivada a partir do tricorde [036] ou de qualquer tetracorde que contenha a classe intervalar 4.

<sup>6</sup> Representamos o 10 por A e o 11 por B. Essa nomenclatura é oriunda do sistema numérico de base hexadecimal. É também a nomenclatura utilizada por Jamary Oliveira em seu PCN.

<sup>7</sup> Um maior aprofundamento sobre este tópico pode ser encontrado em STRAUS (2000:193-196).

<sup>8</sup> <http://www.pitombeira.com/Lewin>