The ddphonism package*

Celia Rubio Madrigal†

September 1, 2019

Abstract

This is a music-related package focused on notation from the Twelve-Tone System, also called Dodecaphonism. It provides LaTeX algorithms that produce typical dodecaphonic diagrams based off a musical series, or row sequence, of variable length.

Keywords

twelve tone system, dodecaphonism, music, mathematics, matrix, row, series, permutation, diagram, clock diagram, notation, algorithm, schoenberg, contemporary music, 20th century

Contents

1 Introduction

There are hundreds of music tools and software online which are able to produce different music notations. However, I have never seen a LaTeX tool that can do the same. This package is not only about notation, but it also calculates mathematically how this notation should work.

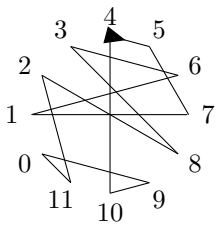
It is said that a twelve-tone matrix is the only thing a twelve-tone composer should need, because it provides the whole serial spectrum with which they may work. I wanted LaTeX users to be able to generate them automatically.

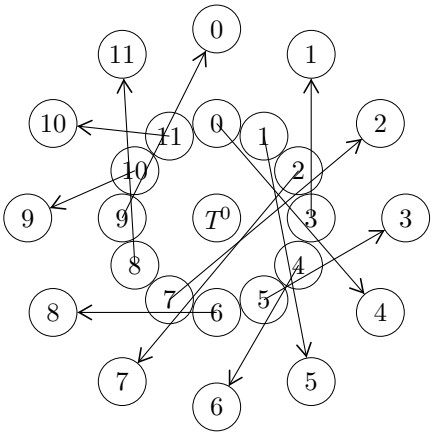
But I also think that a twelve-tone matrix is not enough, that there exist several other notations with which they may understand their series and their potential. These are the diagrams that can be obtained with this package:

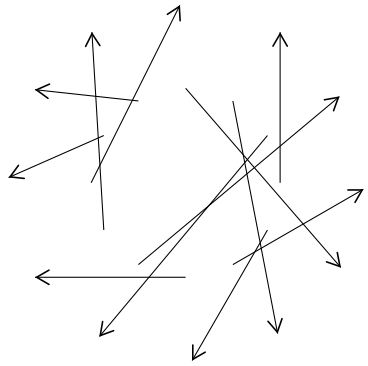
*This document corresponds to ddphonism v0.2, dated 2019/09/01.

†Email: celrubio@ucm.es

4	3	2	1	0
0	4	3	2	1
1	0	4	3	2
2	1	0	4	3
3	2	1	0	4







$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 & 0 \end{pmatrix}$$

2 Using the **ddphonism** package

These are the commands provided by ddphonism. The main parameter in every command is the row sequence.

`\dmatrix` produces a twelve-tone matrix of arbitrary length, as shown in this website. For example, `\dmatrix{0,2,1,4,3,6,5}` produces the matrix

0	2	1	4	3	6	5
5	0	6	2	1	4	3
6	1	0	3	2	5	4
3	5	4	0	6	2	1
4	6	5	1	0	3	2
1	3	2	5	4	0	6
2	4	3	6	5	1	0

sep	scales the matrix.
vsep	scales the matrix vertically.
hsep	scales the matrix horizontally.
lines	draws lines between rows and columns.
outside lines	only draws the outside lines.

inside lines — only draws the inside lines.

vlines — only draws the vertical lines.

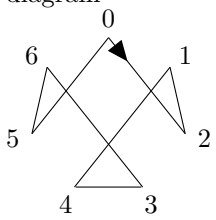
hlines — only draws the horizontal lines.

`\dmatrix[lines,sep=0.75]{0,2,1,4,3,6,5}` produces the matrix

0	2	1	4	3	6	5
5	0	6	2	1	4	3
6	1	0	3	2	5	4
3	5	4	0	6	2	1
4	6	5	1	0	3	2
1	3	2	5	4	0	6
2	4	3	6	5	1	0

no tikz — deletes the tikz environment and lets the user write it instead.

`\ddiagram` produces a twelve tone clock diagram of arbitrary length, as shown in this website. For example, `\ddiagram{0,2,1,4,3,6,5}` produces the diagram

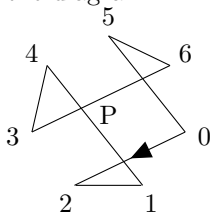


name — writes a name at the center of the diagram.

up — lets the user choose which number is up north. The default value is the first number in the row.

arrow shift — lets the user choose where the arrow should fall on the line. The values range from 0 to 10. The default value is 2.5.

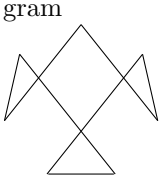
`\ddiagram[name=P, up=5, arrow shift=5]{0,2,1,4,3,6,5}` produces the diagram



no numbers — deletes the numbers around the diagram.

no arrow — deletes the arrow inside the diagram.

`\ddiagram[no numbers, no arrow]{0,2,1,4,3,6,5}` produces the diagram



xshift shifts the figure horizontally.

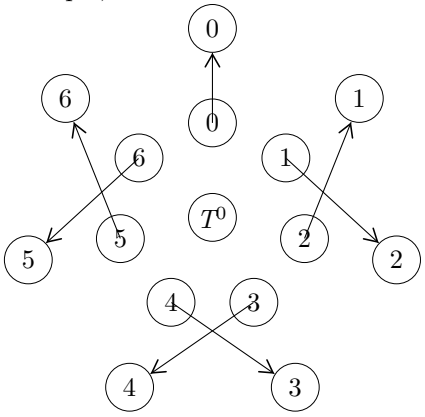
yshift shifts the figure vertically.

no tikz deletes the tikz environment and lets the user write it instead. The option up does not work anymore and the up position becomes 0. It is recommended that the user passes the option ddiagram to the environment:

```
\begin{tikzpicture}[ddiagram]
\ddiagram[no tikz]{0,2,1,4,3,6,5}
\end{tikzpicture}
```

produces the same diagram as `\ddiagram{0,2,1,4,3,6,5}`.

`\ddihedral` produces a dihedral representation of a series of arbitrary length. For example, `\ddihedral{0,2,1,4,3,6,5}` produces the diagram



t applies the transformation *transposition* to the diagram.

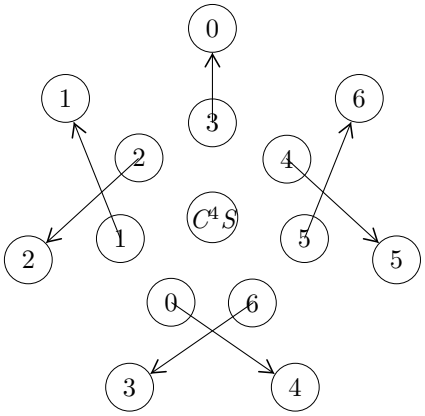
s applies the transformation *inversion* to the diagram.

c applies the transformation *cyclic shift* to the diagram.

v applies the transformation *retrograde* to the diagram.

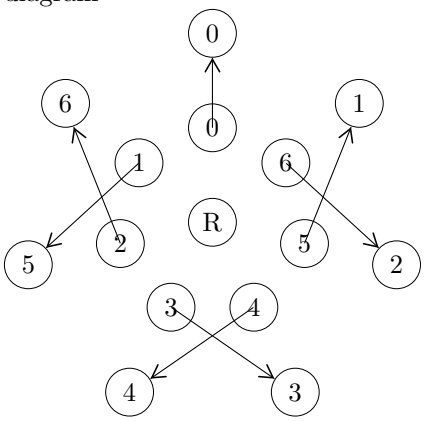
The transformations are applied in that exact order.

`\ddihedral[s=1, c=4]{0,2,1,4,3,6,5}` produces the diagram



no italics	removes the italics from the diagram name.
new t	renames the transformation *transposition*.
new s	renames the transformation *inversion*.
new c	renames the transformation *cyclic shift*.
new v	renames the transformation *retrograde*.

`\ddihedral[no italics, new v=R, v=1]{0,2,1,4,3,6,5}` produces the diagram

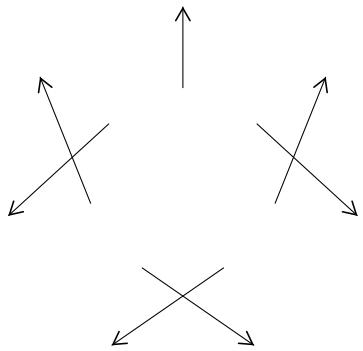


no tikz	deletes the tikz environment and lets the user write it instead. It is recommended that the user passes the option ddihedral to the environment:

```
\begin{tikzpicture}[ddihedral]
\ddihedral[no tikz]{0,2,1,4,3,6,5}
\end{tikzpicture}
```

produces the same diagram as `\ddihedral{0,2,1,4,3,6,5}`.

`\darrows` produces the arrows from the `\ddihedral` diagram. For example, `\darrows{0,2,1,4,3,6,5}` produces the arrows



no tikz deletes the tikz environment and lets the user write it instead.

`\drow` produces a twelve-tone row sequence as a permutation in its matrix form. For example, `\drow{0,2,1,4,3,6,5}` produces the row

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 0 & 2 & 1 & 4 & 3 & 6 & 5 \end{pmatrix}$$

sep lets the user choose the column separation.

`\drow[sep=10pt]{0,2,1,4,3,6,5}` produces the row

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 0 & 2 & 1 & 4 & 3 & 6 & 5 \end{pmatrix}$$

3 The package code

```
% ddphonism
%
% (c) Celia Rubio Madrigal
%
%% This program can be redistributed and/or modified under the terms
%% of the LaTeX Project Public License Distributed from CTAN archives
%% in directory macros/latex/base/lppl.txt.

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{ddphonism}
[2019/09/01 v0.2 Dodecaphonic diagrams: twelve-tone matrices, clock diagrams, etc.]

\RequirePackage{etoolbox}
\RequirePackage{xparse}
\RequirePackage{tikz}
\RequirePackage{xstring}
\RequirePackage{pgfkeys}

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Matrices

\ usetikzlibrary {matrix}

\ExplSyntaxOn
\DeclareExpandableDocumentCommand{\Evaluation}{m}{\int_eval:n {#1}}
\ExplSyntaxOff

\newcounter{Dsize}
\newcommand{\DsizeMake}[1]{%
    \setcounter{Dsize}{0}%
    \foreach \n in {#1}{%
        \stepcounter{Dsize}%
    }%
}

% Only with numbers.
\newcounter{Dfirst}
\newcommand{\DheadMake}[1]{%
    \setcounter{ Dfirst }{−1}%
    \foreach \n in {#1}{%
        \ifnum\theDfirst=−1%
        \setcounter{ Dfirst }{\n}%
        \ fi %
    }%
}

% Only when DsizeMake is already done.
\newcounter{Dmod}
\newcommand{\Modulo}[1]{%
    \setcounter{Dmod}{#1}%
    \loop%
    \ifnum\theDmod>\Evaluation{\theDsize−1}%
    \setcounter{Dmod}{\Evaluation{\theDmod−\theDsize}}%
    \repeat%
    \ifnum\theDmod<0%
    \setcounter{Dmod}{\Evaluation{\theDmod+\theDsize}}%
    \repeat%
    \theDmod%
}

\newif\ ifdmatrixLines
\newif\ ifdmatrixOutside
\newif\ ifdmatrixInside
\newif\ifdmatrixV
\newif\ifdmatrixH
\newif\ ifdmatrixTikz
\pgfkeys{
    /dmatrix/. is family
    , /dmatrix
    , default /. style =
    { lines = false
        , outside lines = false
        , inside lines = false
        , sep = 1
        , vsep = 1
```

```
            , hsep = 1
            , no tikz = false
        }
        , no tikz /.is if=dmatrixTikz
        , lines /.is if=dmatrixLines
        , outside lines /.is if=dmatrixOutside
        , inside lines /.is if=dmatrixInside
        , vlines /.is if=dmatrixV
        , hlines /.is if=dmatrixH
        , sep/.estore in=\dmatrixSep
        , vsep/.estore in=\dmatrixVsep
        , hsep/.estore in=\dmatrixHsep
    }

    \newcommand{\DLOH}{%
        \draw (0.05*\dmatrixSep*\dmatrixHsep,0) --%
        (\theDsize*\dmatrixSep*\dmatrixHsep+0.05*\dmatrixSep*\dmatrixHsep,0);%
        \draw (0.05*\dmatrixSep*\dmatrixHsep,-\theDsize*0.5*\dmatrixSep*\dmatrixVsep) -- %
        (\theDsize*\dmatrixSep*\dmatrixHsep+0.05*\dmatrixSep*\dmatrixHsep,-\theDsize*0.5*\dmatrixSep*\dmatrixVsep);%
    }

    \newcommand{\DLOV}{%
        \draw (0.05*\dmatrixSep*\dmatrixHsep,0) -- %
        (0.05*\dmatrixSep*\dmatrixHsep,-\theDsize*0.5*\dmatrixSep*\dmatrixVsep);%
        \draw (\theDsize*\dmatrixSep*\dmatrixHsep+0.05*\dmatrixSep*\dmatrixHsep,0) -- %
        (\theDsize*\dmatrixSep*\dmatrixHsep+0.05*\dmatrixSep*\dmatrixHsep,-\theDsize*0.5*\dmatrixSep*\dmatrixVsep);
    }

    \newcommand{\DLIH}{%
        \draw (0.05*\dmatrixSep*\dmatrixHsep,-\xD*0.5*\dmatrixSep*\dmatrixVsep) -- %
        (\theDsize*\dmatrixSep*\dmatrixHsep+0.05*\dmatrixSep*\dmatrixHsep,-\xD*0.5*\dmatrixSep*\dmatrixVsep);%
    }

    \newcommand{\DLIV}{%
        \draw (\xD*\dmatrixSep*\dmatrixHsep+0.05*\dmatrixSep*\dmatrixHsep,0) -- %
        (\xD*\dmatrixSep*\dmatrixHsep+0.05*\dmatrixSep*\dmatrixHsep,-\theDsize*0.5*\dmatrixSep*\dmatrixVsep);%
    }

    \newcommand{\dmatrix}[2][]{%
        \DsizeMake{#2}%
        \DheadMake{#2}%
        %
        \pgfkeys{/dmatrix, default , #1}%
        %
        \ifdmatrixTikz\else%
        \begin{ tikzpicture }%
        \ fi %
        \foreach [count=\nj] \j in {#2} {%
            \foreach [count=\ni] \i in {#2} {%
                \draw node at
                ( \ni*\dmatrixSep*\dmatrixHsep-0.5*\dmatrixSep*\dmatrixHsep
                , -\nj*\dmatrixSep*\dmatrixVsep/2+0.25*\dmatrixSep*\dmatrixVsep) {%
                    \Modulo{\Evaluation{\i-\j+\theDfirst}}%
                };%
            }%
        }%
        \foreach \xD in {1,...,\ Evaluation{\theDsize-1}} {%
```

```
            \ifdmatrixLines
135         \DLOH\DLOV\DLIH\DLIV
            \fi
137         \ifdmatrixOutside
            \DLOH\DLOV
139         \fi
            \ ifdmatrixInside
141         \DLIH\DLIV
            \fi
143         \ifdmatrixH
            \DLOH\DLIH
145         \fi
            \ifdmatrixV
147         \DLOV\DLIV
            \fi
149     }%
        %
151     \ifdmatrixTikz\else%
        \end{tikzpicture}%
153     \fi%
    }
155

157 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Diagrams
159
    \ usetikzlibrary {shapes,arrows, decorations.markings,shapes.misc}
161
    \ tikzstyle {ddiagram}=[minimum height=0pt,inner sep=0pt,outer sep=0pt,scale=0.65]
163
    \newif\ifddiagramTikz
165 \newif\ifddiagramNoNum
    \newif\ifddiagramNoArr
167 \pgfkeys{
        /ddiagram/.is family
169     , /ddiagram
        , default /. style =
171     { name =\empty%
            , up =\empty%
173         , no tikz = false
            , no numbers = false
175         , no arrow = false
            , xshift = 0
177         , yshift = 0
            , arrow shift = 2.5
179     }
        , no tikz /. is if =ddiagramTikz
181     , no numbers/.is if =ddiagramNoNum
        , no arrow/. is if =ddiagramNoArr
183     , name/.estore in=\ddiagramName
        , up/. estore in=\ddiagramUp
185     , xshift /. estore in=\ddiagramX
        , yshift /. estore in=\ddiagramY
187     , arrow shift /. estore in=\ddiagramArrS
    }
189
    \newcounter{Dprev}
```

```
\newcommand{\Dvar}{}
\newcommand{\ddiagram}[2][]{%
    \DsizeMake{#2}%
    \DheadMake{#2}%
    %
    \pgfkeys{/ddiagram, default , #1}%
    %
    \ ifdefequal {\ddiagramUp}{\empty}%
    {\renewcommand{\Dvar}{\theDfirst}}% if empty
    {\renewcommand{\Dvar}{\ddiagramUp}}% if not empty
    %
    \ifddiagramTikz\ else %
    \begin{ tikzpicture }[ddiagram,rotate=360∗\Dvar/\theDsize]%
    \ fi %
    \foreach \x in  {0,...,\ Evaluation{\theDsize−1}} {%
        \ifddiagramNoNum\else
        \node [ xshift =\ddiagramX,yshift=\ddiagramY] at (90−360∗\x/\theDsize:2) {\x};%
        \ fi
        \node [ xshift =\ddiagramX,yshift=\ddiagramY] (\x) at (90−360∗\x/\theDsize:1.6) {};%
    };%
    %
    \setcounter{Dprev}{−1}%
    \foreach \x in {#2}{%
        \ifnum \theDprev=\theDfirst%
        \ifddiagramNoArr
        \draw [ xshift =\ddiagramX,yshift=\ddiagramY] (\theDprev) −− (\x);%
        \ else
        \draw [ xshift =\ddiagramX,yshift=\ddiagramY,
        decoration=
        {markings,mark=at position 0.099∗\ddiagramArrS with
            {\arrow[scale=1.25,>=triangle 45]{>}}},
        postaction={decorate}
        ] (\theDprev) −− (\x);%
        \ fi
        \ else \ifnum \theDprev=−1 \else%
        \draw [ xshift =\ddiagramX,yshift=\ddiagramY] (\theDprev) −− (\x);%
        \ fi \ fi %
        \setcounter{Dprev}{\x}%
    };%
    \draw [ xshift =\ddiagramX,yshift=\ddiagramY] (\theDprev) −− (\theDfirst);%
    %
    \ ifdefequal {\ddiagramName}{\empty}%
    {}% if empty
    {\node [ xshift =\ddiagramX,yshift=\ddiagramY] at (0,0) [ circle , fill =white] {\ddiagramName};}% if not empty
    \ifddiagramTikz\ else %
    \end{ tikzpicture }%
    \ fi %
}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dihedral diagrams

\ tikzstyle ddihedralArrow=[decoration=
{markings,mark=at position 1 with {\arrow[scale=1.5,>=angle 60]{>}}},
postaction={decorate}]

```

```
\tikzstyle{ddihedral}=[inner sep=0,minimum height=18pt]

\newif\ifddihedralTikz
\newif\ifddihedralItalics
\pgfkeys{
    /ddihedral/.is family, /ddihedral,
    default/.style =
        { t = 0, c = 0, s = 0, v = 0
        , no tikz=false
        , new t = T, new c = C, new s = S, new v = V
        , no italics = false
        },
    no tikz/.is if=ddihedralTikz,
    t/.estore in = \ddihedralT,
    c/.estore in = \ddihedralC,
    s/.estore in = \ddihedralS,
    v/.estore in = \ddihedralV,
    no italics/.is if=ddihedralItalics,
    new t/.estore in = \ddihedralNewT,
    new c/.estore in = \ddihedralNewC,
    new s/.estore in = \ddihedralNewS,
    new v/.estore in = \ddihedralNewV,
}

\newif\ifdarrowsTikz
\pgfkeys{
    /darrows/.is family, /darrows,
    default/.style = {no tikz=false},
    no tikz/.is if=darrowsTikz,
}
\newcommand{\darrows}[2][]{%
    \DsizeMake{#2}%
    %
    \pgfkeys{/darrows, default, #1}%
    %
    \ifdarrowsTikz\else%
    \begin{tikzpicture}%
    \fi%
    \draw foreach \x in {0,...,\Evaluation{\theDsize-1}} {%
        (90-360*\x/\theDsize:2.5) node[circle] (\x) {}%
    };%
    \foreach \x [count=\y] in {#2} {%
        \draw [style=ddihedralArrow] (90-360*\Evaluation{\y-1}/\theDsize:1.25) -- (\x);%
    };%
    \ifdarrowsTikz\else%
    \end{tikzpicture}%
    \fi%
}

\newcommand\ddihedral[2][]{%
    \DsizeMake{#2}%
    %
    \pgfkeys{/ddihedral, default, #1}%
    %
    \ifddihedralTikz\else%
    \begin{tikzpicture}[ddihedral]%
    \fi%
```

```
        \draw foreach \x in  {0,...,\ Evaluation{\theDsize-1}} {%
            (\Evaluation{(90+\ddihedralT*360/\theDsize)+(2*\ddihedralS-1)*\x*360/\theDsize}:2.5)%
            node[very thin, circle ,draw] (\x) {\x}%
        };%
        %
        \draw foreach \x in  {0,...,\ Evaluation{\theDsize-1}} {%
            (\Evaluation{(90-\ddihedralC*360/\theDsize)+(2*\ddihedralV-1)*\x*360/\theDsize}:1.25)%
            node[very thin, circle ,draw] {\x}%
        };%
        %
        \darrows[no tikz]{#2}%
        %
        \node at (0,0) [very thin ,draw, circle ,  fill =white] {%
        {\ ifddihedralItalics \else\it\fi%
            \ifodd\ddihedralV%
            \ddihedralNewV\else%
            \ifnum\ddihedralC=0%
            \ifodd\ddihedralS\else%
            \ifnum\ddihedralT=0%
            \ddihedralNewT$^0$%
            \fi\fi\fi\fi%
            \ifnum\ddihedralC=0%
            \else \ddihedralNewC$^{\ddihedralC}$\fi%
            \ifodd\ddihedralS%
            \ddihedralNewS\fi%
            \ifnum\ddihedralT=0%
            \else \ddihedralNewT$^{\ddihedralT}$\fi}%
        };%
        \ifddihedralTikz \else%
        \end{tikzpicture}%
        \fi%
    }

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Rows

    \pgfkeys{
        /drow/.is family, /drow,
        default /.style = {sep=\arraycolsep},
        sep/.estore in = \drowSep,
    }

    \long\def\addto#1#2{\expandafter\def\expandafter#1\expandafter{#1#2}}
    \newcounter{myDDcntr}
    \newlength{\Dvarr}

    \newcommand{\drow}[2][]{%
        \DsizeMake{#2}%
        %
        \pgfkeys{/drow, default, #1}%
        \setlength{\Dvarr}{\arraycolsep}
        \setlength{\arraycolsep}{\drowSep}
        %
        \ifnum\theDsize=0%
        \ensuremath{\left(\right)}%
        \else\ifnum\theDsize=1%
```

```
        \ensuremath{%
363         \left(\begin{array}{*{\theDsize}c}%
                0\\%
365             #2\\%
            \end{array}\right)%
367     }%
        \else%
369     \def\TableDDdata{}%
        \setcounter{myDDcntr}{0}%
371     \loop%
        \addto\TableDDdata{\themyDDcntr\stepcounter{myDDcntr} &}%
373     \stepcounter{myDDcntr}%
        \ifnum\themyDDcntr<\Evaluation{\theDsize-1}%
375     \repeat%
        \addto\TableDDdata{\themyDDcntr \\}%
377     \setcounter{myDDcntr}{0}%
        %
379     \ensuremath{%
            \left(\begin{array}{*{\theDsize}c}%
381             \TableDDdata%
                \StrSubstitute{#2}{,}{&}\\%
383         \end{array}\right)%
        }%
385     \fi\fi%
        \setlength{\arraycolsep}{\Dvarr}
387 }

389 \endinput

391

    %% End of file `ddphonism.sty'.
```