

TikZ Latex Figures for Mathematical Music Theory

Alexandre Popoff

December 31, 2018

1 Musical chords diagrams

Often in mathematical music theory, one has to draw diagram of chords and their transformations. Figure 1 is an example of such a diagram.

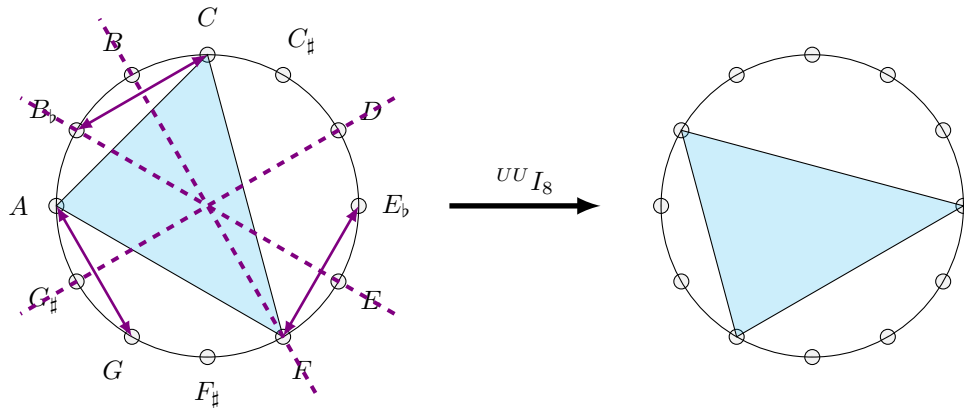


Figure 1: A complex chord transformation.

Defining this diagram with TikZ element by element is feasible, but one clearly sees the benefit in defining a general macro for drawing a twelve-tone circle, the names of the pitch classes, and any number of chords inside, along with the possibility of marking the transformations (usually inversions or transpositions).

The following LaTeX/TikZ code is an attempt at defining such a macro in its most general form.

```

\usepackage{tikz}
\usetikzlibrary{positioning,calc}
\usetikzlibrary{arrows}
\usetikzlibrary{matrix}

\newcommand{\chordcircle}[6] {
  \draw (#1,#2) circle[radius=#3];
  \foreach \i in {0,...,11} {
    \node[circle,draw=black,fill=black,
      fill opacity = 0.05, inner sep=1pt,
      minimum size=3pt]
      (#4_POINTS_\i) at ({#1+#3*sin(\i*30)},{#2+#3*cos(\i*30)}) {.};
  }
  \foreach \x in #5 {
    \foreach \i / \name in {0/$C$,1/$C_\sharp$,2/$D$,3/$E_\flat$,4/$E$,5/$F$,
      6/$F_\sharp$,7/$G$,8/$G_\sharp$,9/$A$,10/$B_\flat$,11/$B$} {
      \node (#4_NAMES_\i) at ({#1+#3*1.2*sin(\i*30)},{#2+#3*1.2*cos(\i*30)}) {\name};
    }
  }
  % Inversion nodes
  \foreach \i in {0,...,23} {
    \node (#4_INV_\i) at ({#1+#3*1.5*sin(\i*15)},{#2+#3*1.5*cos(\i*15)}) {};
  }

  \foreach \chord/\thecolor in #6 {
    \foreach \x [count=\xi from 0] in \chord {
      \node (p_\xi) at (#4_POINTS_\x) {};
    }
    \draw[ draw=black, fill=\thecolor, fill opacity=0.2] (p_0.center)
    \foreach \x [count=\xi from 0] in \chord {
      \ifnum \xi>0
        -- (p_\xi.center)
      \fi
    } --cycle;
  }
}

```

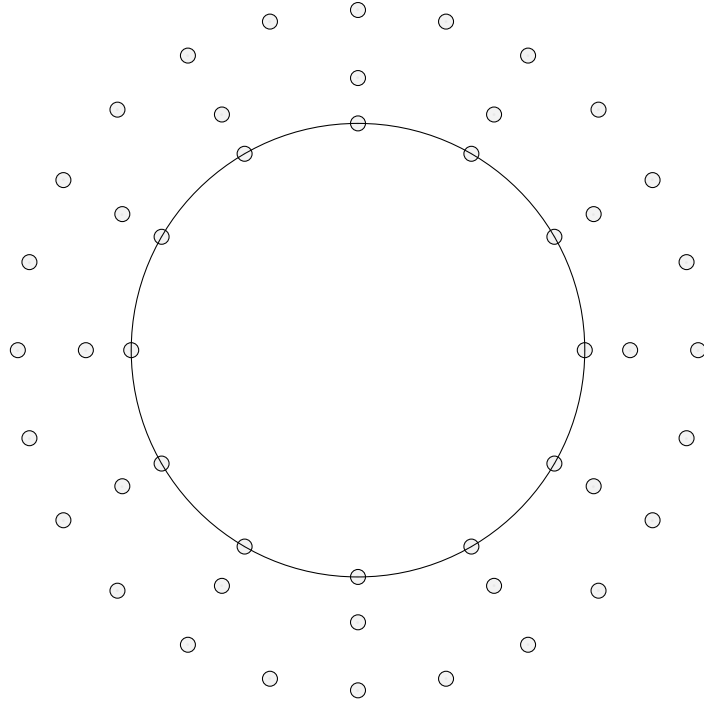


Figure 2: Structure of the diagram of nodes created by the macro.

This code defines

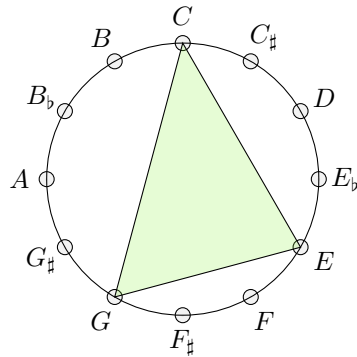
1. a twelve-tone circle with nodes for each pitch class, whose center and radius can be defined (first three arguments). These nodes are named `#4_POINTS_`*i* where `#4` is the fourth argument of the macro, and *i* is their index (from 0 to 11).
2. a series of twelve nodes with pitch classes names, at $1.2 \times \text{radius}$; their visibility can be switched on or off with the fifth argument. These nodes are named `#4_NAMES_`*i* where `#4` is the fourth argument of the macro, and *i* is their index (from 0 to 11).
3. a series of invisible nodes, which can be used for drawing inversions and transpositions. These nodes are named `#4_INV_`*i* where `#4` is the fourth argument of the macro, and *i* is their index (from 0 to 23).

This structure is shown in Figure 2.

The syntax of this macro is as follows.

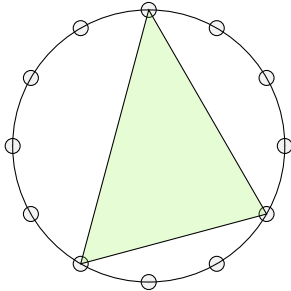
```
\definecolor{color1}{RGB}{r1,g1,b1}
\definecolor{color2}{RGB}{r2,g2,b2}
\chordcircle{center_X}{center_Y}{radius}{name}{pitch_class_names_flag}{
{
{list of pitch-class indices for chord 1}/color1,
{list of pitch-class indices for chord 2}/color2,
...
}
}
```

The code below is a simple example of a single major chord, whose corresponding Figure is shown on the right.



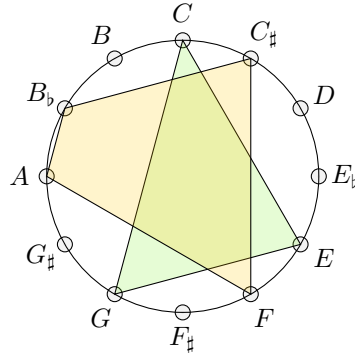
```
\definecolor{my_green}{RGB}{130,240,45}
\chordcircle{0}{0}{4.0}{A}{1}{
{
{0,4,7}/my_green,
}
}
```

The pitch classes names can be removed by passing nothing in the fifth argument.



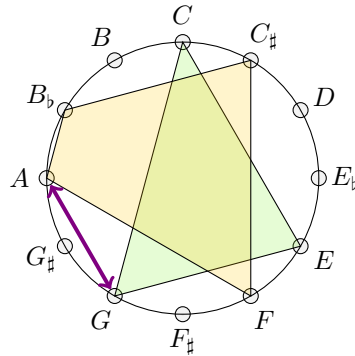
```
\definecolor{my_green}{RGB}{130,240,45}
\chordcircle{0}{0}{4.0}{A}{}{
{
{0,4,7}/my_green,
}
}
```

Multiple chords can be drawn, by adding them to the list of the sixth argument.



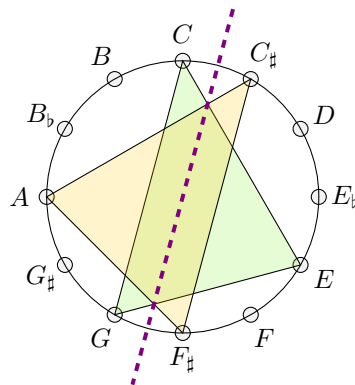
```
\definecolor{my_green}{RGB}{130,240,45}
\definecolor{my_yellow}{RGB}{255,200,0}
\chordcircle{0}{0}{4.0}{A}{1}{
{
{0,4,7}/my_green,
{1,5,9,10}/my_yellow,
}
}
```

The pitch class nodes can be reused, for additional drawing.



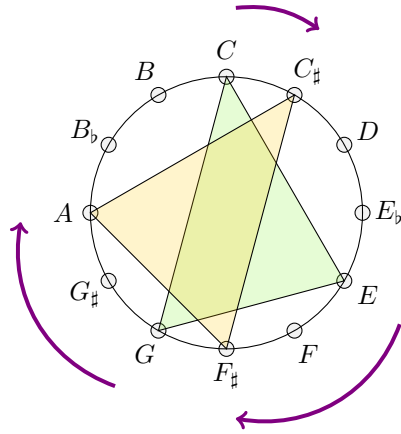
```
\definecolor{my_green}{RGB}{130,240,45}
\definecolor{my_yellow}{RGB}{255,200,0}
\chordcircle{0}{0}{4.0}{A}{1}{
{
{0,4,7}/my_green,
{1,5,9,10}/my_yellow,
}
}
\draw[ <->, line width=1.5, color=violet]
(A_POINTS_7) to (A_POINTS_9);
```

The transformation nodes can be used, for additional drawing. Below is an example showing an inversion.



```
\definecolor{my_green}{RGB}{130,240,45}
\definecolor{my_yellow}{RGB}{255,200,0}
\chordcircle{0}{0}{4.0}{A}{1}{
{
{0,4,7}/my_green,
{1,6,9}/my_yellow,
}
}
\draw[ -, dashed, line width=1.5, color=violet]
(A_INV_1) to (A_INV_13);
```

The same nodes can be used to show transpositions.



```
\definecolor{my_green}{RGB}{130,240,45}
\definecolor{my_yellow}{RGB}{255,200,0}
\chordcircle{0}{0}{4.0}{A}{1}{
{
{0,4,7}/my_green,
{1,5,9,10}/my_yellow,
}
}
\draw[ ->, line width=1.5, color=violet]
(A_INV_0) to[bend left=20] (A_INV_2);
\draw[ ->, line width=1.5, color=violet]
(A_INV_8) to[bend left=40] (A_INV_12);
\draw[ ->, line width=1.5, color=violet]
(A_INV_14) to[bend left=40] (A_INV_18);
```

2 Tonnetz diagrams

Tonnetz diagrams are often used in neo-Riemannian music theory to show the P , L , and R transformations and their successive application. A typical Tonnetz has the topology of a torus, and can therefore be unfolded in 2D. The following code defines a macro for creating a Tonnetz diagram in TikZ, where the number of repeated cells can be defined in both directions.

```
\usepackage{tikz}
\usetikzlibrary{positioning,calc}
\usetikzlibrary{arrows}
\usetikzlibrary{matrix}

\newcommand{\tonnetzdiagram}[2] {
\foreach \i in {0,...,#1} {
  \foreach \j in {0,...,#2} {
    \node[] (Fs\i\j) at (0-1.5*\j+3.5*\i,0+\j*2.598+\i*0.866) {$F_\sharp$};
    \node[] (B\i\j) at (1-1.5*\j+3.5*\i,0+\j*2.598+\i*0.866) {$B$};
    \node[] (E\i\j) at (2-1.5*\j+3.5*\i,0+\j*2.598+\i*0.866) {$E$};
    \node[] (A\i\j) at (3-1.5*\j+3.5*\i,0+\j*2.598+\i*0.866) {$A$};
    \draw[black, draw opacity=0.1, line width=0.5]
      (Fs\i\j.center) -- (B\i\j.center) -- (E\i\j.center) -- (A\i\j.center);
    \node[] (D\i\j) at (0.5-1.5*\j+3.5*\i,-0.866+\j*2.598+\i*0.866) {$D$};
    \node[] (G\i\j) at (1.5-1.5*\j+3.5*\i,-0.866+\j*2.598+\i*0.866) {$G$};
    \node[] (C\i\j) at (2.5-1.5*\j+3.5*\i,-0.866+\j*2.598+\i*0.866) {$C$};
    \node[] (F\i\j) at (3.5-1.5*\j+3.5*\i,-0.866+\j*2.598+\i*0.866) {$F$};
    \draw[black, draw opacity=0.1, line width=0.5]
      (D\i\j.center) -- (G\i\j.center) -- (C\i\j.center) -- (F\i\j.center);
    \node[] (Bb\i\j) at (1-1.5*\j+3.5*\i,-2*0.866+\j*2.598+\i*0.866) {$B_\flat$};
    \node[] (Eb\i\j) at (2-1.5*\j+3.5*\i,-2*0.866+\j*2.598+\i*0.866) {$E_\flat$};
    \node[] (Ab\i\j) at (3-1.5*\j+3.5*\i,-2*0.866+\j*2.598+\i*0.866) {$A_\flat$};
    \node[] (Cs\i\j) at (4-1.5*\j+3.5*\i,-2*0.866+\j*2.598+\i*0.866) {$C_\sharp$};
    \draw[black, draw opacity=0.1, line width=0.5]
      (Bb\i\j.center) -- (Eb\i\j.center) -- (Ab\i\j.center) -- (Cs\i\j.center);

    \draw[black, draw opacity=0.1, line width=0.5]
      (Fs\i\j.center) -- (D\i\j.center) -- (Bb\i\j.center);
    \draw[black, draw opacity=0.1, line width=0.5]
      (B\i\j.center) -- (G\i\j.center) -- (Eb\i\j.center);
    \draw[black, draw opacity=0.1, line width=0.5]
      (E\i\j.center) -- (C\i\j.center) -- (Ab\i\j.center);
    \draw[black, draw opacity=0.1, line width=0.5]
      (A\i\j.center) -- (F\i\j.center) -- (Cs\i\j.center);
    \draw[black, draw opacity=0.1, line width=0.5]
      (D\i\j.center) -- (B\i\j.center);
    \draw[black, draw opacity=0.1, line width=0.5]
      (Bb\i\j.center) -- (G\i\j.center) -- (E\i\j.center);
    \draw[black, draw opacity=0.1, line width=0.5]
      (Eb\i\j.center) -- (C\i\j.center) -- (A\i\j.center);
  }
}
```

```

\draw[black, draw opacity=0.1, line width=0.5]
    (Ab\i\j.center) -- (F\i\j.center);
}
}

\foreach \i in {0,...,\the\numexpr#1-1\relax} {
  \foreach \j in {0,...,#2} {
    \draw[black, draw opacity=0.1, line width=0.5]
        (Cs\i\j.center) -- (Bb\the\numexpr\i+1\relax\j.center);
    \draw[black, draw opacity=0.1, line width=0.5]
        (F\i\j.center) -- (D\the\numexpr\i+1\relax\j.center);
    \draw[black, draw opacity=0.1, line width=0.5]
        (A\i\j.center) -- (Fs\the\numexpr\i+1\relax\j.center);
    \draw[black, draw opacity=0.1, line width=0.5]
        (A\i\j.center) -- (D\the\numexpr\i+1\relax\j.center);
    \draw[black, draw opacity=0.1, line width=0.5]
        (F\i\j.center) -- (Bb\the\numexpr\i+1\relax\j.center);
  }
}

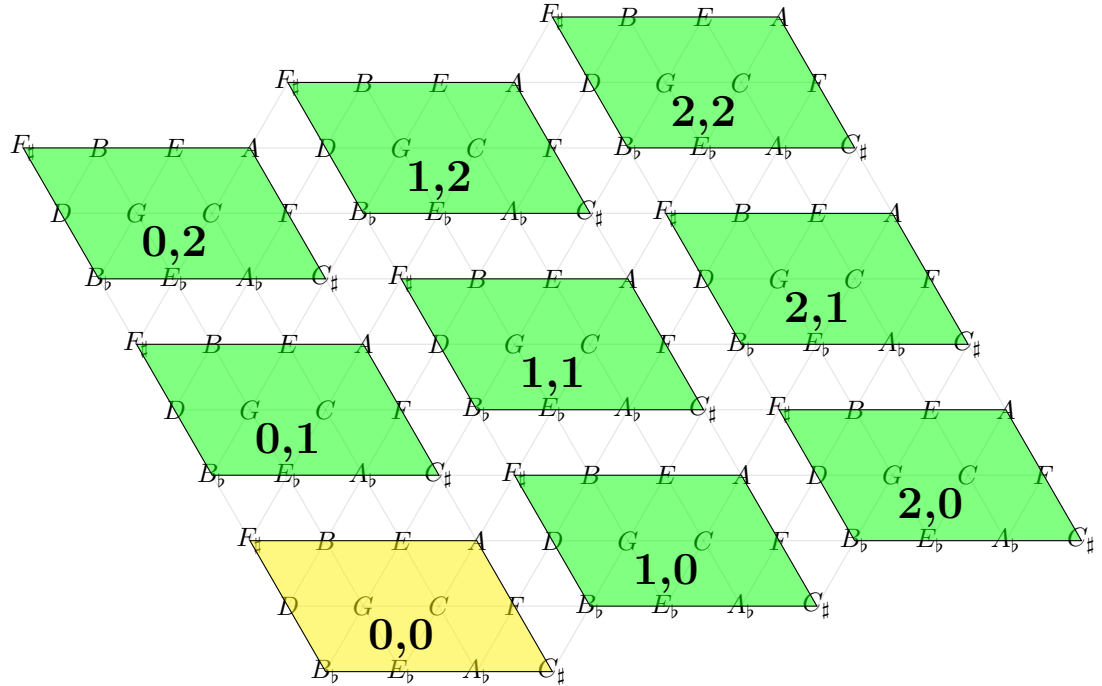
\foreach \i in {0,...,#1} {
  \foreach \j in {0,...,\the\numexpr#2-1\relax} {
    \draw[black, draw opacity=0.1, line width=0.5]
        (Fs\i\j.center) -- (Bb\i\the\numexpr\j+1\relax.center);
    \draw[black, draw opacity=0.1, line width=0.5]
        (B\i\j.center) -- (Eb\i\the\numexpr\j+1\relax.center);
    \draw[black, draw opacity=0.1, line width=0.5]
        (E\i\j.center) -- (Ab\i\the\numexpr\j+1\relax.center);
    \draw[black, draw opacity=0.1, line width=0.5]
        (A\i\j.center) -- (Cs\i\the\numexpr\j+1\relax.center);
    \draw[black, draw opacity=0.1, line width=0.5]
        (Fs\i\j.center) -- (Eb\i\the\numexpr\j+1\relax.center);
    \draw[black, draw opacity=0.1, line width=0.5]
        (B\i\j.center) -- (Ab\i\the\numexpr\j+1\relax.center);
    \draw[black, draw opacity=0.1, line width=0.5]
        (E\i\j.center) -- (Cs\i\the\numexpr\j+1\relax.center);
  }
}

\foreach \i in {0,...,\the\numexpr#1-1\relax} {
  \foreach \j in {1,...,#2} {
    \draw[black, draw opacity=0.1, line width=0.5] (Cs\i\j.center)
        -- (Fs\the\numexpr\i+1\relax\the\numexpr\j-1\relax.center);
  }
}
}

```

The cell (fundamental domain) being repeated by this macro is shown in yellow in the Figure below, wherein the Tonnetz diagram is generated by the command `\tonnetzdiagram{2}{2};`.

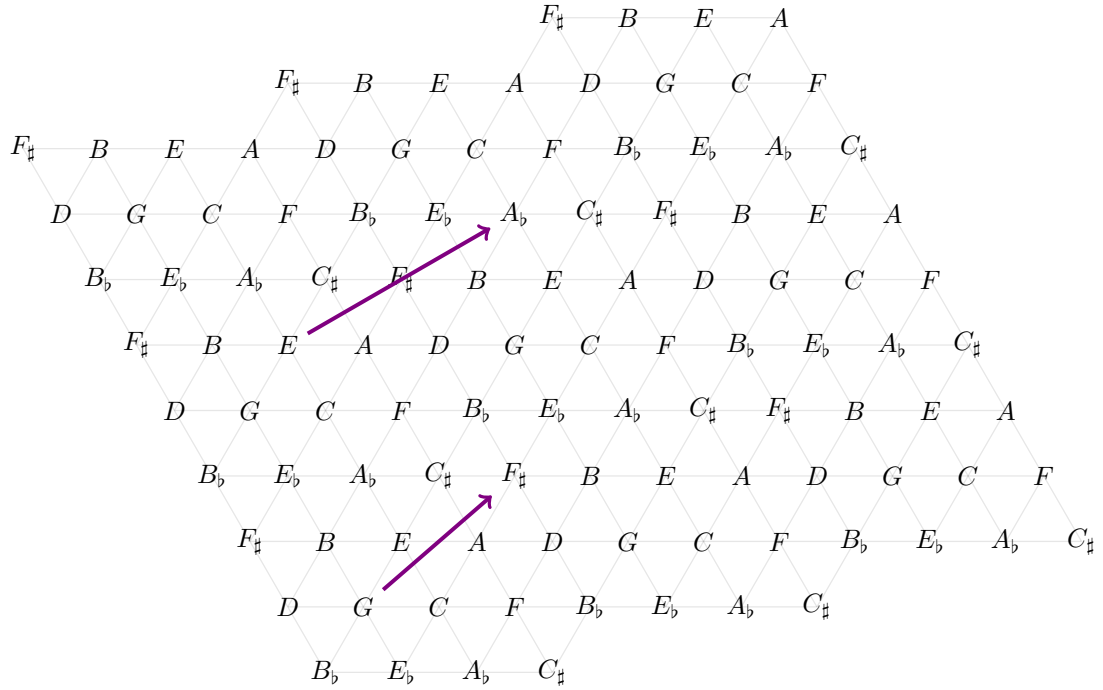
This cell corresponds to the coordinates (0,0), while the coordinates of the repeated cells follow the pattern shown.



The named nodes can be accessed by the pitch class name followed by the coordinates. For example, this code

```
\begin{tikzpicture}[scale=1.0]
\tonnetzdiagram{2}{2};
\draw[ ->, line width=1.5, color=violet] (G00) to (Fs10);
\draw[ ->, line width=1.5, color=violet] (E01) to (Ab11);
\end{tikzpicture}
```

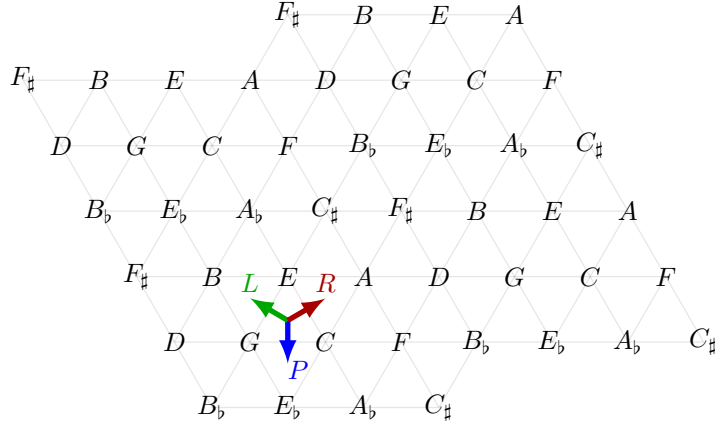
generates the following Figure.



There are no nodes for the center of the triangles (major and minor chords) but they easily can be created from the pitch class name nodes. For example, the code below

```
\begin{tikzpicture}[scale=1.0]
\tonnetzdiagram{1}{1};
\node[] (start) at ($(C00)!0.5!(G00)!0.33!(E00)$) {};
\node[] (endP) at ($(C00)!0.5!(G00)!0.33!(Eb00)$) {};
\node[] (endL) at ($(B00)!0.5!(E00)!0.33!(G00)$) {};
\node[] (endR) at ($(C00)!0.5!(E00)!0.33!(A00)$) {};
\draw[ -latex, line width=2, color=blue ] (start.center) to (endP.center);
\draw[ -latex, line width=2, color=black!35!green] (start.center) to (endL.center);
\draw[ -latex, line width=2, color=black!35!red] (start.center) to (endR.center);
\node[color=blue] (Plabel) at ($(C00)!0.5!(Eb00)!0.15!(G00)$) {$P$};
\node[color=black!35!green] (Llabel) at ($(B00)!0.5!(E00)!0.1!(G00)$) {$L$};
\node[color=black!35!red] (Rlabel) at ($(A00)!0.5!(E00)!0.1!(C00)$) {$R$};
\end{tikzpicture}
```

generates the following Figure, which shows the three basic neo-Riemannian operations starting from the C major chord.



3 Rhythmic canons

A rhythmic canon consists in a motive and entries, both subsets of \mathbb{Z} . The following macro creates a TikZ diagram showing a grid of regularly spaced beats. The abscissa corresponds to time (\mathbb{Z}), whereas the different entries of the canon are shown vertically. A box is marked in black if a beat is played. The first and second argument determines the size of the grid. The third argument is a list of integers corresponding to the motive, whereas the fourth argument is a list of integers corresponding to the entries.

```
\newcommand{\rhythmiccanon}[4] {
  \foreach \i in {0,...,#1} {
    \foreach \j in {0,...,#2} {
      \filldraw[draw=lightgray,fill=none] (\i,\j) rectangle (\i+1,\j+1);
    }
  }
  \foreach \i in #3 {
    \foreach \j [count=\cj] in #4 {
      \filldraw[draw=lightgray,fill=black] (\i+\j,\cj-1) rectangle (\i+\j+1,\cj);
    }
  }
}
```

As an example, the following code

```
\begin{tikzpicture}[scale=0.4]
\rhythmiccanon{20}{10}{{0,1,5}}{{0,2,4,5,8,9,10,12,13,14,15}}
\end{tikzpicture}
```

generates the following diagram.

