On some combinatorial problems concerning the harmonic structure of musical chord sequences

DOMENICO CANTONE

SALVATORE CRISTOFARO

SIMONE FARO

University of Catania, Italy

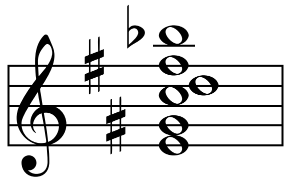
PSC'06: The Prague Stringology Conference '06
Prague, Czech Republic

August 28-30, 2006

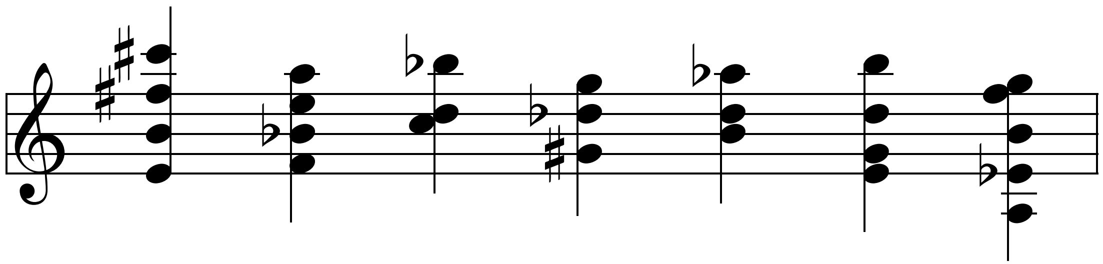
Contents

A bit of music (theory)

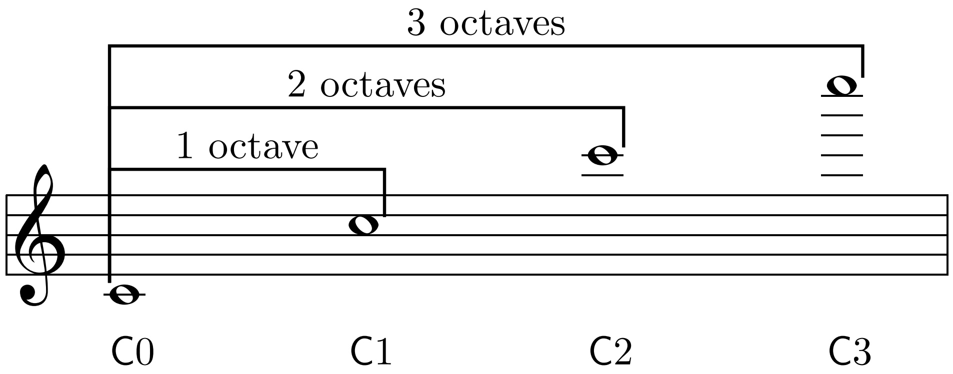
- CHORD: two or more notes sounded simultaneously



- CHORD PROGRESSION: two or more chords played in succession



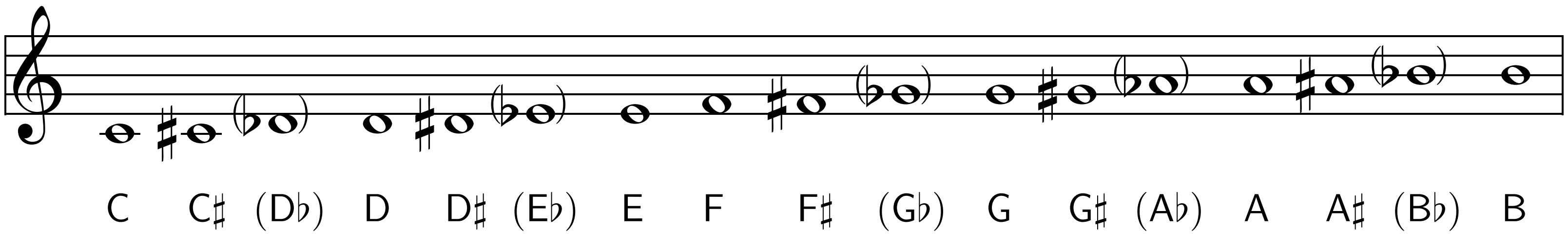
- OCTAVE EQUIVALENCE: no distinction between notes which are one (or more) octave(s) apart



$$\frac{f_{C1}}{f_{C0}} = 2, \ \frac{f_{C2}}{f_{C0}} = 2^2, \ \frac{f_{C3}}{f_{C0}} = 2^3$$

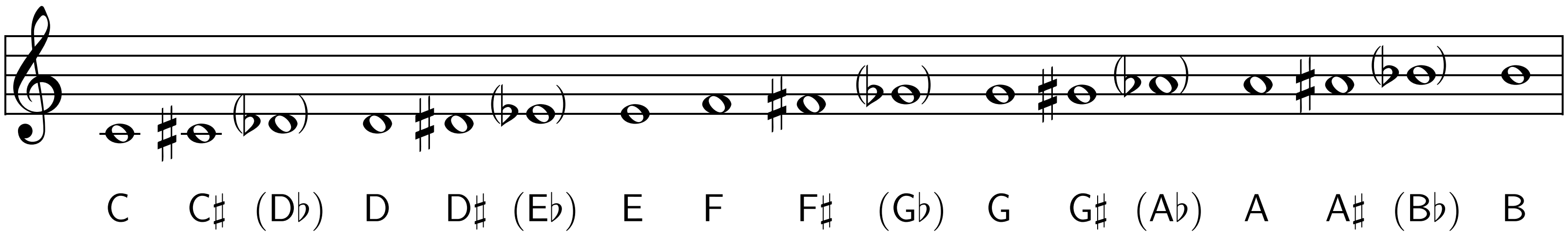
f = fundamental frequency

- Octave equivalence partitions the notes into twelve equivalence classes –*pitch classes*–
  - In each pitch class we choose a note –the *representative*– and we identify the pitch class with its representative;
  - The representatives all belong to the same (musical) octave;

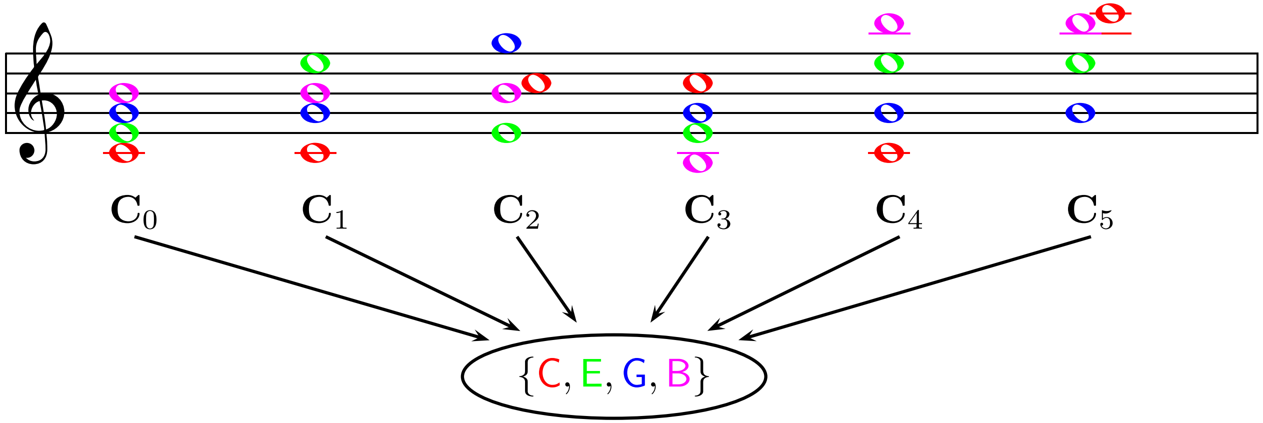


- Representing chords using pitch classes
  - A (non-musical) *chord* is a set of pitch classes;
  - A *voicing* is an ordered tuple (i.e., a string) of distinct pitch classes;

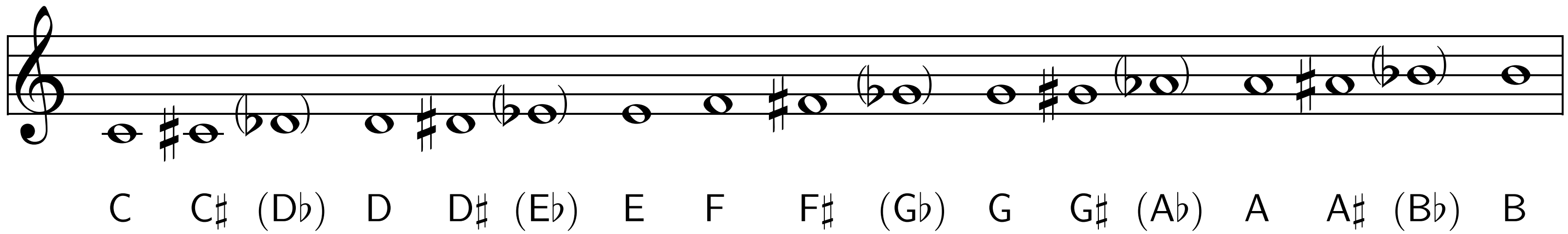
- Octave equivalence partitions the notes into twelve equivalence classes –*pitch classes*–
  - In each pitch class we choose a note –the *representative*– and we identify the pitch class with its representative;
  - The representatives all belong to the same (musical) octave;



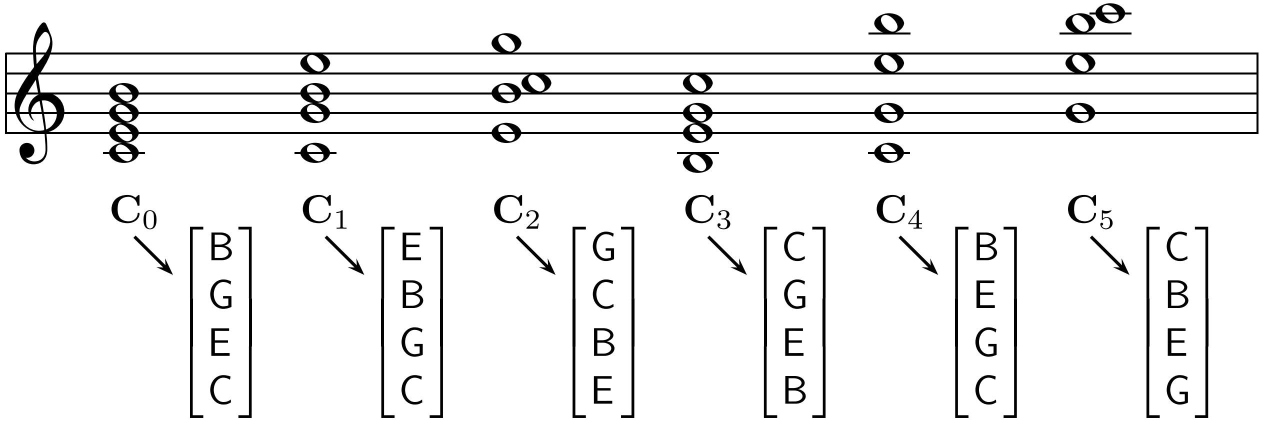
- Representing chords using pitch classes
  - A (non-musical) *chord* is a set of pitch classes;
  - A *voicing* is an ordered tuple (i.e., a string) of distinct pitch classes;



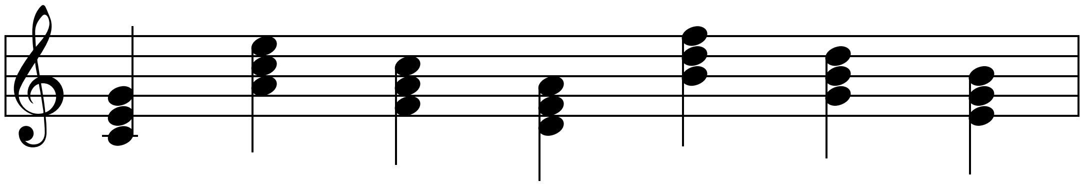
- Octave equivalence partitions the notes into twelve equivalence classes –*pitch classes*–
  - In each pitch class we choose a note –the *representative*– and we identify the pitch class with its representative;
  - The representatives all belong to the same (musical) octave;



- Representing chords using pitch classes
  - A (non-musical) *chord* is a set of pitch classes;
  - A *voicing* is an ordered tuple (i.e., a string) of distinct pitch classes;



Regular Harmonic Structures and Chord Connections



Chord Progression $\mathcal{C}$

- Chord progression $\mathcal{C}$ corresponds to the following list of sets of pitch classes

$$\{\mathsf{C},\mathsf{E},\mathsf{G}\},\{\mathsf{C},\mathsf{E},\mathsf{A}\},\{\mathsf{C},\mathsf{F},\mathsf{A}\},\{\mathsf{D},\mathsf{F},\mathsf{A}\},\{\mathsf{D},\mathsf{F},\mathsf{B}\},\{\mathsf{D},\mathsf{G},\mathsf{B}\},\{\mathsf{E},\mathsf{G},\mathsf{B}\}.$$

- If we look at the voicings of the chords of $\mathcal{C}$ we get the following “regular” matrix $\mathcal{M}$:

$$\mathcal{M} = \begin{bmatrix} \mathsf{G} & \mathsf{E} & \mathsf{C} & \mathsf{A} & \mathsf{F} & \mathsf{D} & \mathsf{B} \\ \mathsf{E} & \mathsf{C} & \mathsf{A} & \mathsf{F} & \mathsf{D} & \mathsf{B} & \mathsf{G} \\ \mathsf{C} & \mathsf{A} & \mathsf{F} & \mathsf{D} & \mathsf{B} & \mathsf{G} & \mathsf{E} \end{bmatrix}$$

- If we “glue” at the left (or right) end of matrix $\mathcal{M}$ a copy of itself, we get:

$$\mathcal{M}^{\star} = \left[\begin{array}{ccccccc|ccccccc} \mathsf{G} & \mathsf{E} & \mathsf{C} & \mathsf{A} & \mathsf{F} & \mathsf{D} & \mathsf{B} & \mathsf{G} & \mathsf{E} & \mathsf{C} & \mathsf{A} & \mathsf{F} & \mathsf{D} & \mathsf{B} \\ \mathsf{E} & \mathsf{C} & \mathsf{A} & \mathsf{F} & \mathsf{D} & \mathsf{B} & \mathsf{G} & \mathsf{E} & \mathsf{C} & \mathsf{A} & \mathsf{F} & \mathsf{D} & \mathsf{B} & \mathsf{G} \\ \mathsf{C} & \mathsf{A} & \mathsf{F} & \mathsf{D} & \mathsf{B} & \mathsf{G} & \mathsf{E} & \mathsf{C} & \mathsf{A} & \mathsf{F} & \mathsf{D} & \mathsf{B} & \mathsf{G} & \mathsf{E} \end{array}\right]$$

Problem Can we voice a chord progression in such a way that it assumes a regular harmonic structure like that of the chord progression $\mathcal{C}$?

Formal definitions

Let Σ be a finite alphabet ($\Sigma = \{a, b, c, x, y\}$ in the examples).

- A CHORD over Σ is a nonempty set C of two or more symbols of Σ.
- The SIZE of a chord C, denoted by $size(C)$, is the number of symbols in C.
- A CHORD PROGRESSION is a sequence $\mathcal{C} = \langle C_0, C_1, \ldots, C_n \rangle$ of chords such that

$$size(C_0) = size(C_1) = \cdots = size(C_n).$$

- A string X of length $m \geq 0$ is represented as a finite array $X[0\,..\,m-1]$. The length of X is denoted by $|X|$. By $X[i]$ we denote the $(i+1)$-th symbol of X, for $0 \leq i < |X|$.
- A VOICING over Σ is a string V of symbols of Σ such that $|V| \geq 2$ and $V[i] \neq V[j]$ for all distinct $i, j \in \{0, 1, \ldots, |V|-1\}$.

  Example

$$V = axby \qquad \left(V = \begin{bmatrix} y \\ b \\ x \\ a \end{bmatrix} \right)$$

- The BASE CHORD $Set(V)$ of a voicing V is the set of the symbols occurring in V.
- A voicing V is said to be a VOICING OF A CHORD C if $Set(V) = C$.

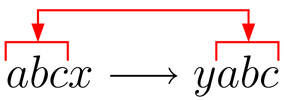
- A VOICE LEADING over Σ is a sequence $\mathcal{V} = \langle V_0, V_1, \ldots, V_n \rangle$ of voicings over Σ such that

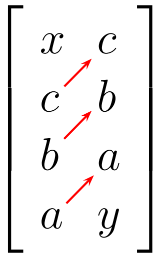
$$|V_0| = |V_1| = \cdots = |V_n|.$$

- A voice leading $\mathcal{V} = \langle V_0, V_1, \ldots, V_n \rangle$ is a VOICE LEADING OF A CHORD PROGRESSION $\mathcal{C} = \langle C_0, C_1, \ldots, C_m \rangle$, if $n = m$ and V_i is a voicing of the chord C_i, for $i = 0, 1, \ldots, n$.

- A voicing V is (IMMEDIATELY) CONNECTED to a voicing W, in symbols $V \longrightarrow W$, if $W = s_{\bullet} V[0 \mathinner{..} |V| - 2]$, for some symbol $s \in \Sigma$.

  Example

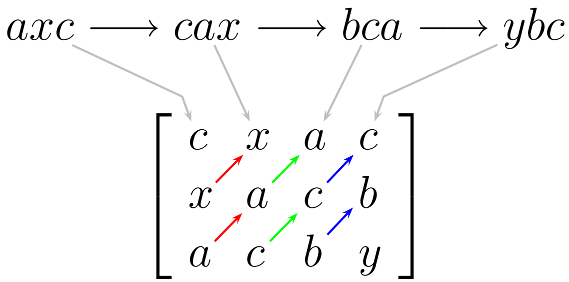




- A voice leading $\mathcal{V} = \langle V_0, V_1, \ldots, V_n \rangle$ is CONNECTED if $V_i \longrightarrow V_{i+1}$, for $i = 0, 1, \ldots, n-1$; $\mathcal{V}$ is CIRCULARLY CONNECTED if it is connected and in addition $V_n \longrightarrow V_0$.

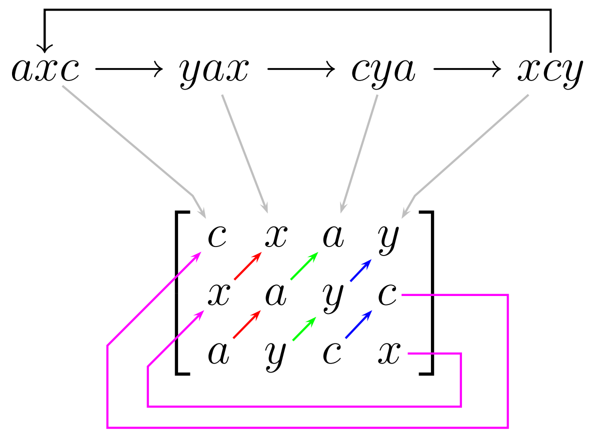
  Example

  A connected voice leading



Example

A circularly connected voice leading



- A voicing V is CONNECTABLE to a voicing W with respect to the alphabet Σ, in symbols $V \Longrightarrow W$, if there is a connected voice leading $\mathcal{V} = \langle V_0, V_1, \ldots, V_n \rangle$ over Σ, with $n \geq 1$, such that $V_0 = V$ and $V_n = W$.

The *connectivity relation* "$\Longrightarrow$" (between voicings) is an equivalence relation

- $V \Longrightarrow V$ (**Reflexivity**)
- $V \Longrightarrow W$ implies $W \Longrightarrow V$ (**Symmetry**)
- $V \Longrightarrow W$ and $W \Longrightarrow Z$ imply $V \Longrightarrow Z$ (**Transitivity**)

for all voicings V, W and Z.

- A chord C is CONNECTED to a chord D, written $C \longrightarrow D$, if $V \longrightarrow W$, for some voicings V of C and W of D.

- A chord progression $\mathcal{C}$ is CONNECTED (resp., CIRCULARLY CONNECTED) if it has a connected (resp., circularly connected) voice leading. A chord progression $\mathcal{C} = \langle C_0, C_1, \ldots, C_n \rangle$ is REGULAR if it is circularly connected and, in addition, $C_i \neq C_{(i+1) \mod (n+1)}$, for $i = 0, 1, \ldots, n$.

Example

The chord progression

$$\mathcal{C} = \langle C_0, C_1, C_2, C_3 \rangle,$$

where

$$C_0 = \{a, c, x\},\ C_1 = \{a, x, y\},\ C_2 = \{a, c, y\},\ C_3 = \{c, x, y\},$$

is regular:

– the following is a circularly connected voice leading of $\mathcal{C}$

$$\underbrace{axc \longrightarrow yax \longrightarrow cya \longrightarrow xcy}_{\longleftarrow}$$

– and, in addition,

$$C_0 \neq C_1 \neq C_2 \neq C_3 \neq C_0$$

Discovering regular structures: some algorithms

Problem 1 Given a chord progression $\mathcal{C} = \langle C_0, C_1, \ldots, C_n \rangle$ over an alphabet Σ, a voicing V of C_0, and a voicing W of C_n, construct, if it exists, a connected voice leading $\mathcal{V} = \langle V_0, V_1, \ldots, V_n \rangle$ of $\mathcal{C}$ such that $V_0 = V$ and $V_n = W$.

- We start by setting $V_0 = V$.
- Suppose we have constructed the partial connected voice leading $\langle V_0, V_1, \ldots, V_i \rangle$ of $\langle C_0, C_1, \ldots, C_i \rangle$.

  - Let $S_i =_{\text{Def}} Set(V_i[0 \mathinner{..} m-2])$
    if $S_i \subseteq C_{i+1}$ **then**
      - let $C_{i+1} \setminus S_i = \{c\}$
      $V_{i+1} =_{\text{Def}} c_\bullet V_i[0 \mathinner{..} m-2]$
    else STOP

ALGO1($\mathcal{C}$, V, W)
$m := |V|$
$X := V$
for $i := 1$ **to** n **do**
  if $Set(X[0 \mathinner{..} m-2]) \subseteq C_i$ **then**
    - let z be such that $C_i = Set(X[0 \mathinner{..} m-2]) \cup \{z\}$
    $X := z_\bullet X[0 \mathinner{..} m-2]$
    OUTPUT(X)
  else
    return *false*
if $X \neq W$ **then**
  return *false*
return *true*

OUTPUT: A sequence $V_1, V_2, \ldots, V_k$ of voicings such that $\langle V, V_1, V_2, \ldots, V_k \rangle$ is the longest connected voice leading, starting at V, of an initial segment of $\mathcal{C}$.

Let

$$\mathcal{C} = \langle \underset{C_0}{\{a,b,c\}}, \underset{C_1}{\{a,b,x\}}, \underset{C_2}{\{a,b,x\}}, \underset{C_3}{\{b,x,y\}} \rangle, \quad V = abc, \; W = ybx$$

0) $X = V = abc$

Let

$$\overset{\textstyle C_0}{\{a,b,c\}},\ \overset{\textstyle C_1}{\{a,b,x\}},\ \overset{\textstyle C_2}{\{a,b,x\}},\ \overset{\textstyle C_3}{\{b,x,y\}}$$

$$\mathcal{C} = \langle\{a,b,c\},\{a,b,x\},\{a,b,x\},\{b,x,y\}\rangle, \quad V = abc, \ W = ybx$$

0) $X = V = abc$

1) $Set(X[0\,..\,m-2]) = \{a,b\} \subseteq C_1 \ \rightarrow \ C_1 \setminus \{a,b\} = \{x\} \ \rightarrow \ X = xab$

OUTPUT: xab

Let

$$\overset{C_0}{} \quad \overset{C_1}{} \quad \overset{C_2}{} \quad \overset{C_3}{}$$

$$\mathcal{C} = \langle \{a,b,c\}, \{a,b,x\}, \{a,b,x\}, \{b,x,y\} \rangle, \quad V = abc, \;\; W = ybx$$

0) $X = V = abc$

1) $Set(X[0\,..\,m-2]) = \{a,b\} \subseteq C_1 \;\rightarrow\; C_1 \setminus \{a,b\} = \{x\} \;\rightarrow\; X = xab$

2) $Set(X[0\,..\,m-2]) = \{a,x\} \subseteq C_2 \;\rightarrow\; C_2 \setminus \{a,x\} = \{b\} \;\rightarrow\; X = bxa$

OUTPUT: xab, bxa

Let

$$\mathcal{C} = \langle \underset{C_0}{\{a,b,c\}}, \underset{C_1}{\{a,b,x\}}, \underset{C_2}{\{a,b,x\}}, \underset{C_3}{\{b,x,y\}} \rangle, \quad V = abc, \ W = ybx$$

0) $X = V = abc$

1) $Set(X[0\,..\,m-2]) = \{a,b\} \subseteq C_1 \;\rightarrow\; C_1 \setminus \{a,b\} = \{x\} \;\rightarrow\; X = xab$

2) $Set(X[0\,..\,m-2]) = \{a,x\} \subseteq C_2 \;\rightarrow\; C_2 \setminus \{a,x\} = \{b\} \;\rightarrow\; X = bxa$

3) $Set(X[0\,..\,m-2]) = \{b,x\} \subseteq C_3 \;\rightarrow\; C_3 \setminus \{b,x\} = \{y\} \;\rightarrow\; X = ybx = W \;\rightarrow$ TRUE

OUTPUT: xab, bxa, ybx

$$abc \longrightarrow xab \longrightarrow bxa \longrightarrow ybx$$

Time Complexity

```
ALGO1(C, V, W)
m := |V|
X := V
for i := 1 to n do                                        ┐
    if Set(X[0..m-2]) ⊆ C_i then              ┐            │
        - let z be such that C_i = Set(X[0..m-2]) ∪ {z}   │
        X := z • X[0..m-2]                    │            │
        OUTPUT(X)                             │ T_1(m)     │ O(n × T_1(m))
    else                                      │            │
        return false                          │            │
    end if                                    ┘            │
end for                                                   ┘
if X ≠ W then                                             ┐
    return false                                          │ T_2(m)
end if                                                    ┘
return true
```

Representing chords and voicings as linear arrays:

$$\begin{cases} \mathrm{T}_1(m) = \mathcal{O}(m^2) \\ \mathrm{T}_2(m) = \mathcal{O}(m) \end{cases} \longrightarrow \text{Overall Running Time} = \mathcal{O}(n \times m^2)$$

However, by using bit-parallelism we can reduce time complexity to $\mathcal{O}(n+m)\ldots$

Representation of Chords and Voicings

Let

$$\Sigma = \{s_0, s_1, \ldots, s_{\sigma-1}\}$$

be a fixed alphabet. We use the following representations:

- a singleton $\{s_i\} \subseteq \Sigma$ is represented as the bit mask $\mathsf{B}(s_i) = b_0 b_1 \cdots b_{\sigma-1}$ (of length σ), where

$$b_j = \begin{cases} \mathsf{1} & \text{if } j = \sigma - 1 - i \\ \mathsf{0} & \text{otherwise}\,, \end{cases}$$

  for $j = 0, 1, \ldots, \sigma - 1\,$;

- a nonempty subset $A = \{s_{i_0}, s_{i_1}, \ldots, s_{i_k}\}$ of Σ is represented as the bit mask

$$\mathsf{B}(A) =_{\text{Def}} \mathsf{B}(s_{i_0}) \vee \mathsf{B}(s_{i_1}) \vee \cdots \vee \mathsf{B}(s_{i_k})\,;$$

- the empty subset of Σ is represented by the bit mask $\mathsf{0}^\sigma$, i.e., the string consisting of σ copies of the bit $\mathsf{0}$;

- a chord progression $\mathcal{C} = \langle C_0, C_1, \ldots, C_n \rangle$ is represented as an array $\boldsymbol{\mathcal{C}}[0\,..\,n]$ of $n+1$ bit masks, where $\boldsymbol{\mathcal{C}}[i] = \mathsf{B}(C_i)$ for $i = 0, 1, \ldots, n$;

- a voicing V of length m is represented as an array $\mathbf{V}[0\,..\,m-1]$ of m bit masks, where $\mathbf{V}[i] = \mathsf{B}(V[i])$, for $i = 0, 1, \ldots, m-1$ (this amounts to represent a voicing $V = v_0 v_1 \cdots v_{m-1}$ as the ordered tuple of the bit masks corresponding to the singletons $\{v_0\}, \{v_1\}, \ldots, \{v_{m-1}\}$).

Examples

Let

$$\Sigma = \{a, b, c, x, y\}$$

- Singletons

$$\{a\}, \ \{b\}, \ \{c\}, \ \{x\}, \ \{y\}$$

are represented by the bit masks

$$\mathsf{B}(a) = \mathsf{00001}, \ \mathsf{B}(b) = \mathsf{00010}, \ \mathsf{B}(c) = \mathsf{00100}, \ \mathsf{B}(x) = \mathsf{01000}, \ \mathsf{B}(y) = \mathsf{10000}$$

- Chords

$$A = \{a, b, c\}, \ B = \{a, b, c, y\}, \ C = \{y, b, x\}$$

are represented by the bit masks

$$\mathsf{B}(A) = \mathsf{00111}, \ \mathsf{B}(B) = \mathsf{10111}, \ \mathsf{B}(C) = \mathsf{11010}$$

- Voicings

$$V = abc, \ W = xbya$$

are represented by the arrays (of bit masks)

$$\mathbf{V} = [\mathsf{00001}, \mathsf{00010}, \mathsf{00100}], \quad \mathbf{W} = [\mathsf{10000}, \mathsf{00010}, \mathsf{01000}, \mathsf{00001}]$$

Algorithm ALGO2: a bit-parallel version of ALGO1

ALGO2($\mathcal{C}$, V, W)

1. $m :=$ length($\mathbf{V}$)
2. $n :=$ length($\mathcal{C}$) $- 1$
3. **for** $h = m - 2$ **down to** 0 **do**
4. $\mathbf{Q}[h] := \mathbf{V}[m - 2 - h]$
5. $\mathsf{S} := 0^\sigma$
6. **for** $i := 0$ **to** $m - 2$ **do**
7. $\mathsf{S} := \mathsf{S} \vee \mathbf{V}[i]$
8. $h := 0$
9. **for** $i := 1$ **to** n **do**
10.   **if** $(\mathcal{C}[i] \wedge \mathsf{S}) = \mathsf{S}$ **then**
11. $\mathsf{Z} := (\mathcal{C}[i] \wedge \sim \mathsf{S})$
12. $\mathsf{D} := \mathbf{Q}[h]$
13. $\mathbf{Q}[h] := \mathsf{Z}$
14. $h := (h + 1) \mod (m - 1)$
15. $\mathsf{S} := (\mathsf{S} \wedge \sim \mathsf{D}) \vee \mathsf{Z}$
16.   **else**
17.   **return** *false*
18. **for** $j := 0$ **to** $m - 2$ **do**
19.   **if** $\mathbf{Q}[(h + j) \mod (m - 1)] \neq \mathbf{W}[m - 2 - j]$ **then**
20.   **return** *false*
21. **return** *true*

Time complexity: $\mathcal{O}(n + m)$

The algorithm ALGO2 returns *true* if there is a connected voice leading of the chord progression $\mathcal{C}$ from the voicing V to voicing W, and *false*, otherwise.

The algorithm "constructs" the longest connected voice leading $\langle V_0, V_1, \ldots, V_k \rangle$ (starting at V) of an initial segment of $\mathcal{C}$.

For $i = 1, 2, \ldots, k$, immediately after iteration i of the **for-loop** of line 9:

- the partial voicing $V_i[0\,..\,m-2]$ is stored circularly into the array **Q**:

$V_i[h-1]$	$\cdots$	$V_i[0]$	$V_i[m-2]$	$\cdots$	$V_i[h]$
0	$\cdots$	$h-1$	h	$\cdots$	$m-2$

- the bit mask S stores the partial chord $Set(V_i[0\,..\,m-2])$;

ALGO2($\mathcal{C}$, V, W)

1. $m := \text{length}(\mathbf{V})$
2. $n := \text{length}(\mathcal{C}) - 1$
3. **for** $h = m - 2$ **down to** 0 **do**
4. $\mathbf{Q}[h] := \mathbf{V}[m - 2 - h]$
5. $\mathsf{S} := 0^{\sigma}$
6. **for** $i := 0$ **to** $m - 2$ **do**
7. $\mathsf{S} := \mathsf{S} \vee \mathbf{V}[i]$
8. $h := 0$
9. **for** $i := 1$ **to** n **do**
10.   **if** $(\mathcal{C}[i] \wedge \mathsf{S}) = \mathsf{S}$ **then**
11. $\mathsf{Z} := (\mathcal{C}[i] \wedge \sim \mathsf{S})$

        $X := \mathit{decode}(\mathsf{Z})$
        for $j := 0$ **to** $m - 2$ **do**
            $X := X_{\bullet}\, \mathit{decode}(\mathbf{Q}[(h + m - 2 - j) \mod (m - 1)])$
        OUTPUT(X)

12. $\mathsf{D} := \mathbf{Q}[h]$
13. $\mathbf{Q}[h] := \mathsf{Z}$
14. $h := (h + 1) \mod (m - 1)$
15. $\mathsf{S} := (\mathsf{S} \wedge \sim \mathsf{D}) \vee \mathsf{Z}$
16.   **else**
17.   **return** *false*
18. **for** $j := 0$ **to** $m - 2$ **do**
19.   **if** $\mathbf{Q}[(h + j) \mod (m - 1)] \neq \mathbf{W}[m - 2 - j]$ **then**
20.   **return** *false*
21. **return** *true*

If we use an auxiliary string-variable X and add the following lines of code between lines 11 and 12:

$X := \mathit{decode}(\mathsf{Z})$
for $j := 0$ **to** $m - 2$ **do**
    $X := X_{\bullet}\, \mathit{decode}(\mathbf{Q}[(h + m - 2 - j) \mod (m - 1)])$
OUTPUT(X)

we get as output the longest connected voice leading of an initial segment of $\mathcal{C}$.

The one-argument function *decode* yields the symbol s, when applied to the bit mask $\mathsf{B}(s)$ which represents the singleton $\{s\}$, for $s \in \Sigma$.

If we assume that Σ is the set of the first σ nonnegative integers, $\Sigma = \{0, 1, \ldots, \sigma - 1\}$, then

- $\mathsf{B}(s) = (1 << s) = 2^s$, for each $s \in \Sigma$;
- $\mathit{decode}(x) =_{\text{Def}} \log_2 x$;
  $(s = \log_2 2^s = \log_2 \mathsf{B}(s) = \mathit{decode}(\mathsf{B}(s)))$

Problem 2 Given a chord progression $\mathcal{C} = \langle C_0, C_1, \ldots, C_n \rangle$, check whether $\mathcal{C}$ is regular.

- **A natural (but inefficient) solution**
  Let m be the size of the chords $C_0, C_1, \ldots, C_n$.
  - We start by checking that $C_i \neq C_{i+1}$, for $i = 0, 1, \ldots, n-1$;
  - Then we form the set $Voic(C_0)$ of all possible voicings of the first chord C_0;
  - For each voicing $V \in Voic(C_0)$ we run the algorithm ALGO1 to search for a connected voice leading of $\mathcal{C}$ from V to the voicing $W = V[1\,..\,m-1]\,.\,w$, where w is the only symbol of C_n not contained in C_0 (if, indeed, $size(C_n \setminus C_0) \neq 1$, then, certainly, $\mathcal{C}$ would not be regular).

  However, since there are $m!$ possible voicings of C_0, such an approach is very time-consuming.

- **The main observation**
  Suppose $\mathcal{C} = \langle C_0, C_1, \ldots, C_n \rangle$ is regular, and let $\mathcal{V} = \langle V_0, V_1, \ldots, V_n \rangle$ be a circularly connected voice leading of $\mathcal{C}$, where $V_0 = v_0 v_1 v_2 \cdots v_{m-1}$. Moreover, let

$$X_k = \bigcap_{i=0}^{m-1-k} C_i \,, \quad \text{for } k = 0, 1, \ldots, m-1.$$

  Then

  1. $X_0 = \{v_0\},\ X_1 \setminus X_0 = \{v_1\},\ X_2 \setminus X_1 = \{v_2\},\ \ldots\ , X_{m-1} \setminus X_{m-2} = \{v_{m-1}\}$;
  2. $V_n = v_1 v_2 \cdots v_{m-1} w$ where $C_n \setminus C_0 = \{w\}$;

Example

The chord progression $\mathcal{C} = \langle C_0, C_1, C_2, C_3, C_4 \rangle$, where

$$C_0 = \{a, b, c, x\},\ C_1 = \{a, c, x, y\},\ C_2 = \{a, b, x, y\},\ C_3 = \{a, b, c, y\},\ C_4 = \{b, c, x, y\},$$

is regular.

$$\begin{array}{c} \begin{matrix} C_0 & C_1 & C_2 & C_3 & C_4 \end{matrix} \\ \begin{bmatrix} b & c & x & a & y \\ c & x & a & y & b \\ x & a & y & b & c \\ a & y & b & c & x \end{bmatrix} \end{array}$$

$X_3 = C_0 = \{a, b, c, x\}$

$$\begin{array}{c} \begin{matrix} C_0 & C_1 & C_2 & C_3 & C_4 \end{matrix} \\ \begin{bmatrix} b & c & x & a & y \\ c & x & a & y & b \\ x & a & y & b & c \\ a & y & b & c & x \end{bmatrix} \end{array}$$

$X_2 = C_0 \cap C_1 = \{a, c, x\}$

$X_3 \setminus X_2 = \{b\}$

$$\begin{array}{c} \begin{matrix} C_0 & C_1 & C_2 & C_3 & C_4 \end{matrix} \\ \begin{bmatrix} b & c & x & a & y \\ c & x & a & y & b \\ x & a & y & b & c \\ a & y & b & c & x \end{bmatrix} \end{array}$$

$X_1 = C_0 \cap C_1 \cap C_2 = \{a, x\}$

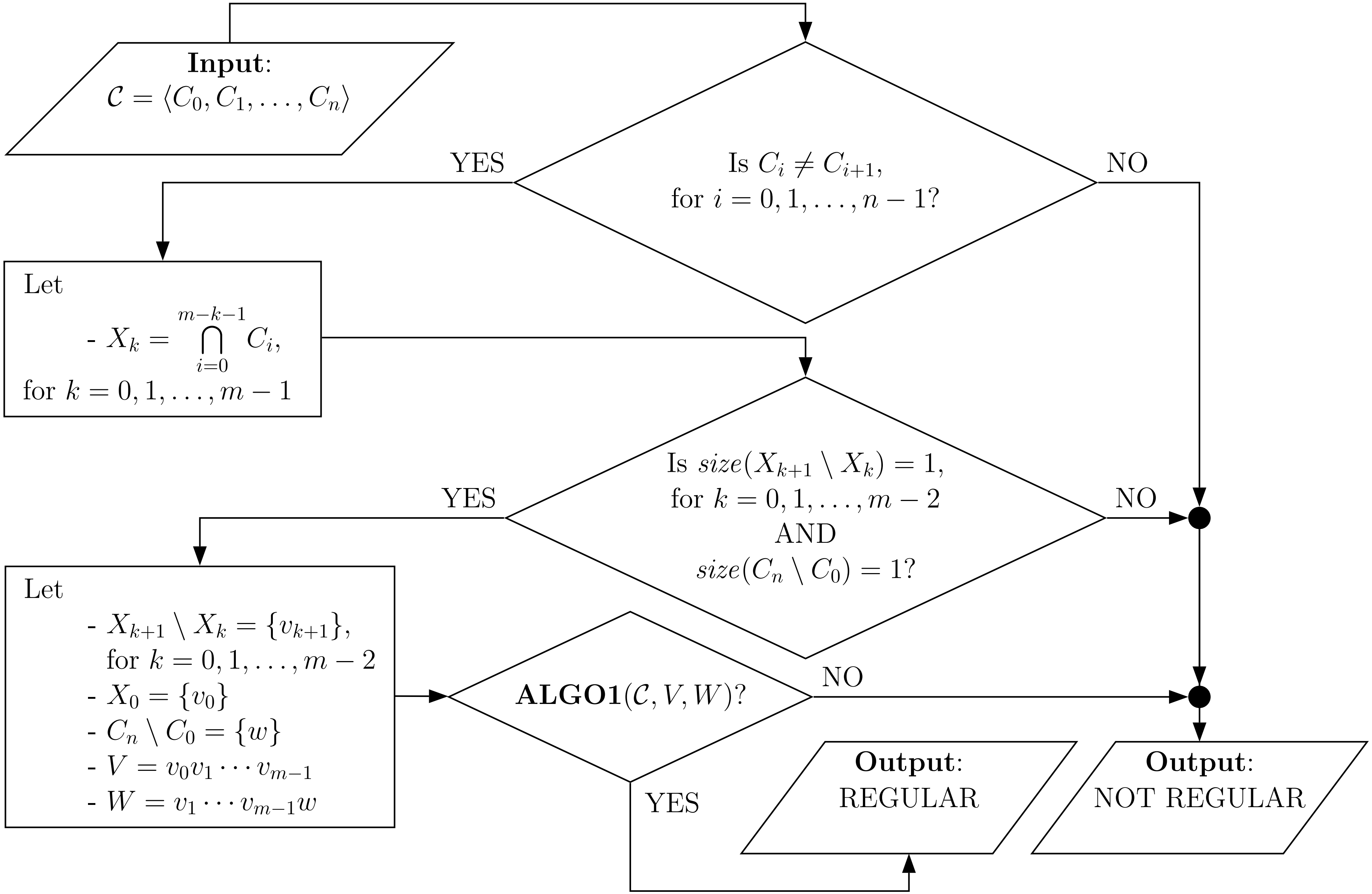
$X_2 \setminus X_1 = \{c\}$

$$\begin{array}{c} \begin{matrix} C_0 & C_1 & C_2 & C_3 & C_4 \end{matrix} \\ \begin{bmatrix} b & c & x & a & y \\ c & x & a & y & b \\ x & a & y & b & c \\ a & y & b & c & x \end{bmatrix} \end{array}$$

$X_0 = C_0 \cap C_1 \cap C_2 \cap C_3 = \{a\}$

$X_1 \setminus X_0 = \{x\}$

The Algorithm ALGO3 to check whether a chord progression $\mathcal{C} = \langle C_0, C_1, \ldots, C_n \rangle$ is regular.



Algorithm ALGO4: a bit-parallel version of ALGO3

ALGO4($\mathcal{C}$, m)

$n := \text{length}(\mathcal{C}) - 1$
for $i := 0$ **to** $n - 1$ **do**
    if $\mathcal{C}[i] = \mathcal{C}[i+1]$ **then**
        return *false*
end for

(Is $C_0 \neq C_1 \neq \cdots \neq C_n$?)

$X[m-1] := \mathcal{C}[0]$
for $k := m-2$ **down to** 0 **do**
    $X[k] := X[k+1] \wedge \mathcal{C}[m-k-1]$
    if $X[k] \neq 0^\sigma$ **and** $X[k] \neq X[k+1]$ **then**
        $\mathbf{V}[k+1] := \mathbf{W}[k] := X[k+1] \wedge \sim X[k]$
    else
        return *false*
    end if
end for

(Construct sets $X_{m-1}, X_{m-2}, \ldots, X_0$ and check whether $X_{m-1} \supsetneq X_{m-2} \supsetneq \cdots \supsetneq X_0 \neq \varnothing$)

$\mathbf{V}[0] := X[0]$
if $X[0] \wedge \mathcal{C}[n] = 0^\sigma$ **and** $(\mathcal{C}[n] \wedge \mathcal{C}[0]) \vee X[0] = \mathcal{C}[0]$ **then** (Is $\mathit{size}(C_n \setminus C_0) = 1$?)
    $\mathbf{W}[m-1] := \mathcal{C}[n] \wedge \sim \mathcal{C}[0]$
    return ALGO2$(\mathcal{C}, \mathbf{V}, \mathbf{W})$
else
    return *false*
end if

Time complexity: $\mathcal{O}(n+m)$

Further questions on the connectivity of chords

Property 3 Any two chords of the same size can always be connected by a voice leading.

Let C and D be two chords of size m, and let V_0 be any voicing of C. We define a connected voice leading $\mathcal{V} = \langle V_0, V_1, \ldots, V_m \rangle$ such that $Set(V_m) = D$:

- $V_{i+1} = s_i \centerdot V_i[0\,..\,m-2]$, where s_i is any symbol in $D \setminus Set(V_i[0\,..\,m-2])$,

for $i = 0, 1, \ldots, m-2$.

We notice that the voicing V_0 of C has been selected arbitrarily ... therefore, we can conclude that the following property holds too:

Property 4 Any given chord progression $\mathcal{C} = \langle C_0, C_1, \ldots, C_n \rangle$ can always be extended to a connected chord progression $\mathcal{C}' = \langle C'_0, C'_1, \ldots, C'_p \rangle$, in the sense that $C_i = C'_{k_i}$, for some strictly increasing sequence of indices $0 \leq k_i \leq p$, for $i = 0, 1, \ldots, n$.

An interesting problem is then the following:

Open Problem 5 Given a chord progression $\mathcal{C}$, find a connected chord progression of minimal length which extends $\mathcal{C}$.

The connectivity relation between voicings depends on the richness of the alphabet.

Let $V = abcd$ and $W = abdc$ be two voicings of the same chord $C = \{a, b, c, d\}$. If we try to connect V to W by using only symbols of the alphabet $\Sigma = \{a, b, c, d\}$, then we end up with the *periodic voice leading*

$$abcd \longrightarrow dabc \longrightarrow cdab \longrightarrow bcda \longrightarrow abcd \longrightarrow dabc \longrightarrow cdab \longrightarrow bcda \longrightarrow abcd \longrightarrow \ \ldots$$

However, if we are allowed to use a new symbol, say x, then it is immediate to see that

$$\langle abcd, xabc, cxab, dcxa, bdcx, abdc \rangle$$

is a voice leading which connects V to W (with respect to the extended alphabet $\Sigma \cup \{x\}$).

A connectability test for voicings:

> **Property 6** Given any two voicings V and W of the same length over an alphabet Σ, if $Set(V) \neq \Sigma$ or $Set(W) \neq \Sigma$, then V can be connected to W with respect to Σ, otherwise V can be connected to W if and only if W is a substring of $V \centerdot V$.

... and the related optimization problem:

> **Open Problem 7** Given two voicings V and W of the same length over an alphabet Σ, determine a shortest voice leading connecting V to W.

... however, Property 6 does not say anything on the fact that a voice leading $\mathcal{V}$ which connects V to W have to satisfy the additional property that any two or more consecutive voicings of $\mathcal{V}$ must have distinct base chords.

Example

Although voicing $V = abcd$ is connectable to voicing $W = abdc$ with respect to the alphabet $\Sigma = \{a, b, c, d\} \cup \{x\}$, there is no way to connect V to W by a voice leading $\langle V_0, V_1, \dots, V_n \rangle$ over Σ such that $Set(V_0) \neq Set(V_1) \neq \dots Set(V_n)$.

Indeed, if we try to connect V to W by a such voice leading:

$$abcd \longrightarrow xabc \longrightarrow dxab \longrightarrow cdxa \longrightarrow bcdx \longrightarrow abcd \longrightarrow xabc \longrightarrow dxab \longrightarrow cdxa \longrightarrow \dots$$

... fortunately, we have:

Property 8 Let V and W be voicings of length m over an alphabet Σ of size at least $m + 2$. Then there is a connected voice leading $\langle V_0, V_1, \dots, V_n \rangle$, which connects V to W with respect to Σ, such that $Set(V_i) \neq Set(V_{i+1})$, for $i = 0, 1, \dots, n - 1$.

... therefore, any given chord progression can always be extended to a regular chord progression by adding at most two new symbols. An interesting question is then the following:

Open Problem 9 Given a chord progression $\mathcal{C} = \langle C_0, C_1, \dots C_n \rangle$ over an alphabet Σ and a fixed bound $k > n$, determine the minimum number of new symbols we need to add to Σ in order that $\mathcal{C}$ can be extended to a regular chord progression of length at most k.

Conclusions and future works

- A bit of music (theory)
- Regular Harmonic Structures and Chord Connections
- Formal definitions
- Discovering regular structures: some algorithms
- Further questions on the connectivity of chords
- Solutions to the open problems
- Other notions of connectivity of chords