\experimentalGraceSyncer

»**egs.ly**« **by ArnoldTheresius**

Experimenteller Grace-Syncer - ich wollte eine Funktion schreiben, welche die übergebene ly:music-Struktur auswertet, Grace-Skips hinzufügt, gegebenenfalls einzelne Events verschiebt (umsortiert), und die überarbeitete ly:music-Struktur für die Grafik-Erstellung zurückgibt, also so einfach wie das **\unfoldRepeats** für die MIDI-Ausgabe ähnlich einer Music-Map-Prozedur angewandt wird.

Als erster Schritt wird nur eine Teilmenge der zu synchronisierenden Events (bzw. Schlüsselwörter) betrachtet: **\set, \override, \applyContext, \unset** und **\revert,** welche eine logische Untergruppe bilden - eng verwandt mit **\break, \noBreak, \pageBreak, \pageTurn** u.s.w. Zusätzlich werden auch noch **\clef** und **\key** betrachtet; diese sind aus der Gruppe der Events, welche sowohl bezüglich der ganzen Partitur (Score) als auch bezüglich der Notzenzeile (Staff) zu synchronisieren sind - zusammen mit der Stimme (Voice) also auf drei Ebenen, diese Gruppe nenne ich daher in Englisch »three-tier-group«.

1. Bearbeitungsschritt: In jedem **\new Staff** wird eine zusätzliche parallel laufende **\new Voice** eingefügt, in welche die zu synchronisierenden Events hineinkommen, zusammen mit den (neu zu erstellenden) Skips und Grace-Skips zur partiturweiten Ausrichtung.

Analyse: Um die Analyse zu vereinfachen, verpacke ich die zu synchronisierenden Events in »Briefcouverts« (Envelops). Ich implementiere diese Envelops als *SequentialMusic,* bei denen ein zusätzlicher Parameter gesetzt ist. Logischerweise muß die zeitliche Länge dieser Envelops gleich Null sein. Im gleichen Bearbeitungsschritt wird zu jedem Hauptton-Zeitschritt registiert, welcher maximale Vorschlag-Zeitschritt über alle Stimmen auftritt. Zukünftig werden eventuell statt nur *SequentialMusic* auch andere Events direkt unterstützt - das wären dann quasi »Postkarten« statt »Briefcouverts«.

2. Bearbeitungsschritt: Die entsprechenden **Envelops** werden entweder in diese Grace-Syncing-Voice kopiert oder verschoben. Folglich macht das Kopieren aus dem **\once** quasi ein **\twice** ☺. Zudem implementiere ich das Verschieben im Kern genauso wie das Kopieren, nur daß anschließend das Original-Envelop durch ein No-Operation-Event in seiner *SequentialMusic* ersetzt wird.

Keine Automatik: Im ersten Schritt wird auch von der Analyse keine automatische Entscheidung getroffen, ob der Envelop zu kopieren oder zu verschieben sei - der Anwender gibt es vor. Damit aber die Envelops je nach Partiturlayout unterschiedlich behandelt werden können (z. Bsp. zwei bezüglich Position des Schlüsselwechsels konkurrierende Stimmen werden entweder in einer gemeinsamen Notenzeile oder in zwei getrennten Notenzeilen notiert), kann man dieses Grace-Sync-Envelop-Property auch noch »nachträglich« an über **\tag** identifizierbaren *SequentialMusic* Elementen abändern.

3. Bearbeitungsschritt: Nachdem die »Envelops« je Synchroniseriungsstimme gesammelt wurden, werden diese darin aufgereiht und die Zeit-Zwischenräume mit Skips und (davor) mit Grace-Skips in der Größe des Maximums der ganzen Partitur zu diesem Haupt-Zeitstempel aufgefüllt.

Weiterentwicklungen: • Einige Elemente werden jetzt auch direkt, also ohne umhüllende SequentialMusic, gekennzeichnet. • Die meisten (mit viel Glück alle) Elemente, welche für die Synchronisation zu markieren wären, werden nun automatisch entdeckt und mit 'move markiert — also müssen nur noch die Elemente manuell markiert werden, welche den Wert *'copy* oder *'keep* benötigen.

Experimental Grace Sycner - the idea is a music function which analyzes the ly:music structure, adds some grace skips, moves some events if required, and returns the altered (grace synced) ly:music structure for rendering, just as easy to use as **\unfoldRepeats** (which is used for MIDI output).

In a first step only a sub group of events (or say: key words) which require grace snychonisation will be checked: **\set, \override, \applyContext, \unset** and **\revert.** They form a sub group in the grace sync issue. Additionally **\clef** and **\key** from the tree-tier-group will be checked, too. I call »three-tier-group« the group of events, which need to be grace synchronized in respective to both the whole score and the staff.

1st phase: I add a parallel **\new Voice** into each **\new Staff** . Into this voice I collect the events addressed to grace synchonisation, and the space between is filled with skips and grace skips according to the grace length of the complete score.

Analysis: To simplify the analysis, each event which is addressed for grace synchronisation must be put into an »envolop«. I implemented these envelops as *SequentialMusic* with an additional parameter, and off course the time-length of such an envelop has to be zero. During the analysis phase for each main time step the leftmost grace time is recorded.

2nd phase: As defined in the envelops, these will be moved or copied into the Grace-Syncing-Voice. Well, move is executed as a copy first, then the original envelop will be replaced with a No-Operation-Event.

no automatism: The selection, how to process an envelop (move or copy), is set manually. Thus a *\once \set* will finally be executed twice if the copy option is choosen. To manually solve the 3-tier-situations you can also modify the handling flag in all envelops you can identify by a tag.

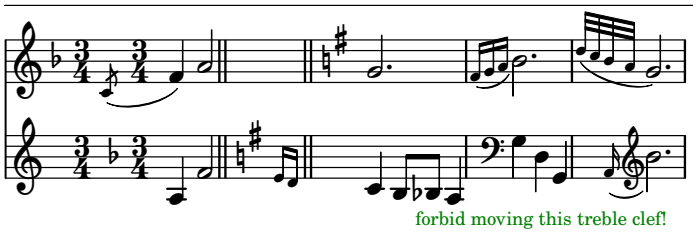
3rd phase: Move the envelops from the inbox of each Grace-Syncing-Voice into its sequential music list, and fill the space between with skips and grace skips. The size of the grace skips corresponds to the maximum seen in the complete score.

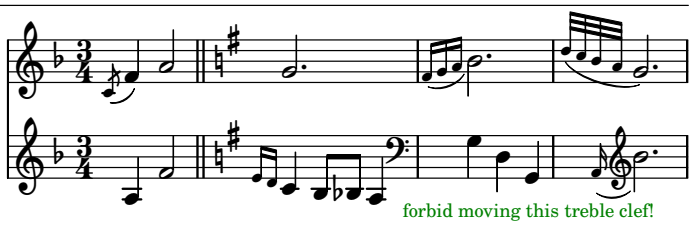
Development update: • Some events will now be marked for grace-syncing directly, i.e. without surrounding SequentialMusic. • Most (hopefully all) of the events requiring a grace-syncing will now be selected automatically with the 'move option applied on them — so you only need to apply the other values ('copy or 'keep) to the few events which need a different handling.

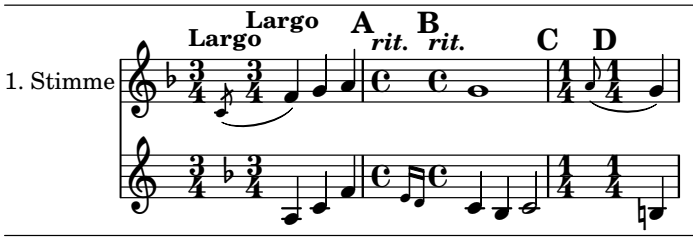
Testsystem: Linux [#911 SMP Thu Sep 15 14:22:38 BST 2016] armv7l
(ohne/without Repeatcommands_graceecho_purger Patch)

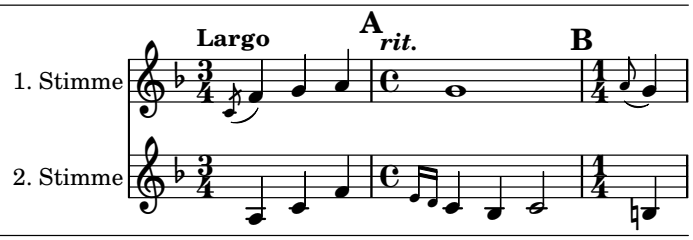
```
\score {
  \myMusic
}
```

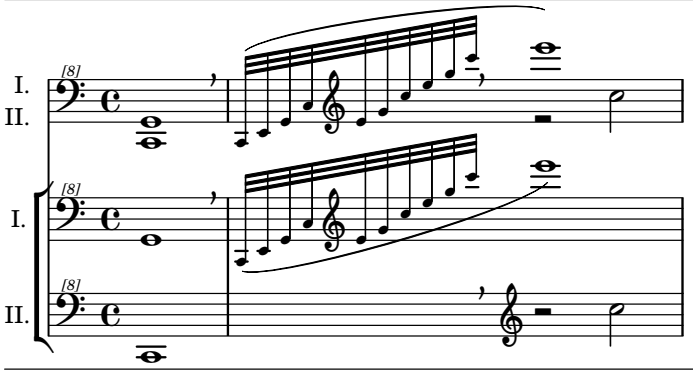
```
\score {
  \experimentalGraceSyncer \myMusic
}
```

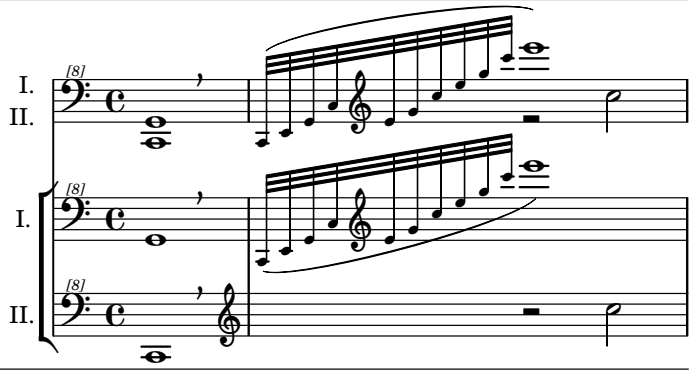


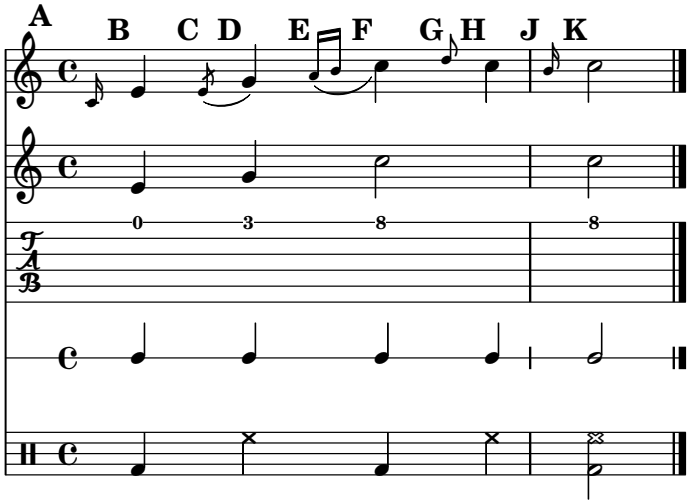


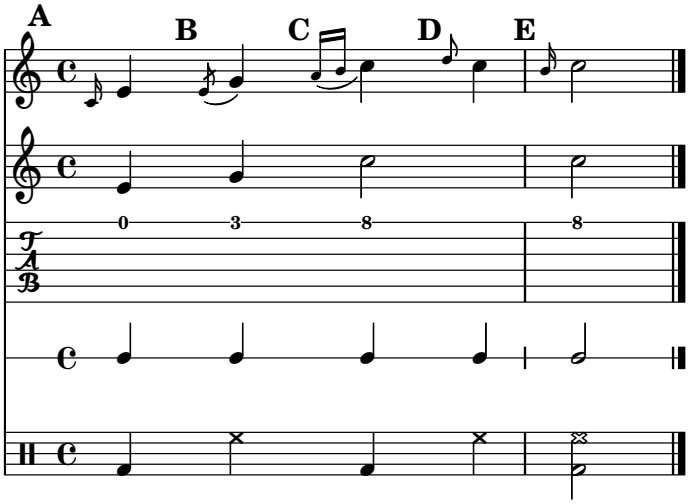




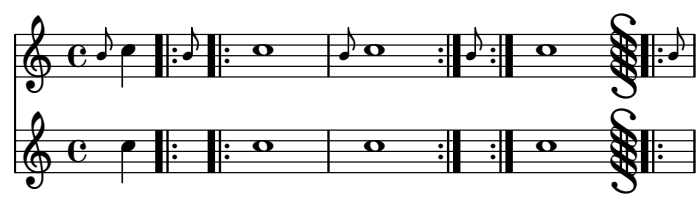


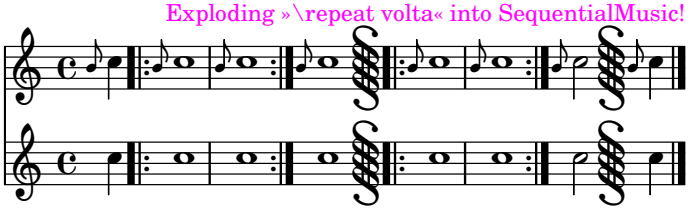


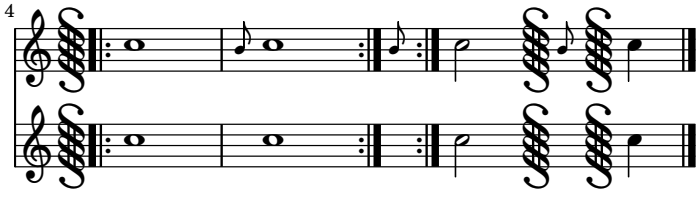




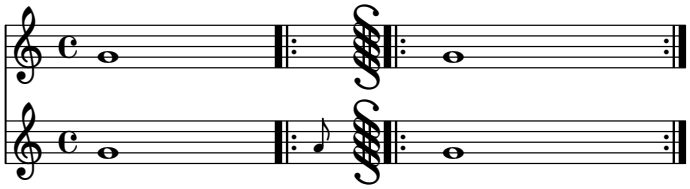
Exploding »\repeat volta« into SequentialMusic!



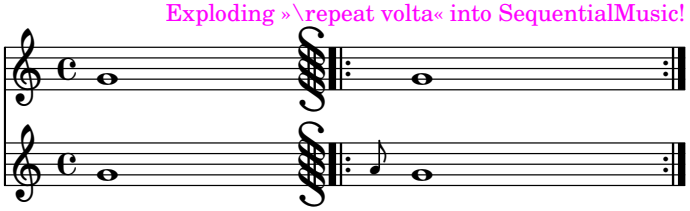




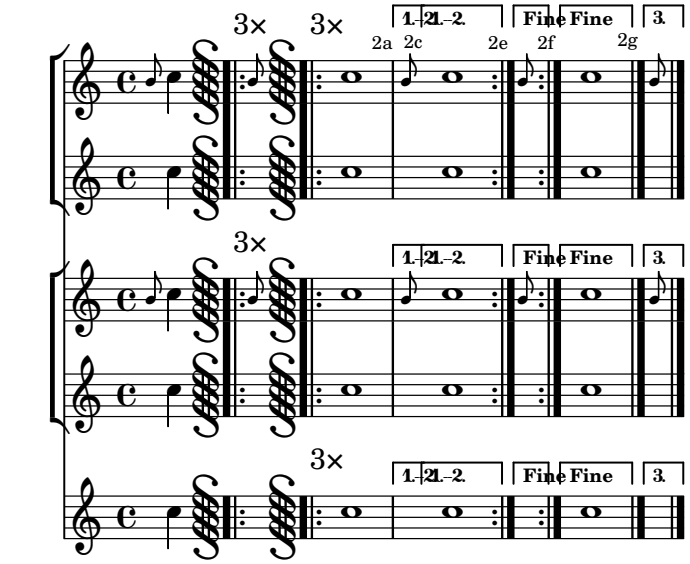
\inStaffSegno only in the voice without grace



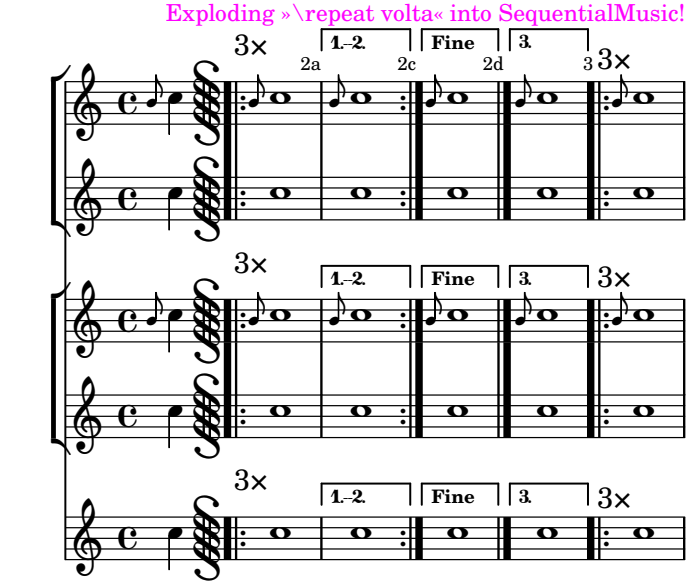
Exploding »\repeat volta« into SequentialMusic!

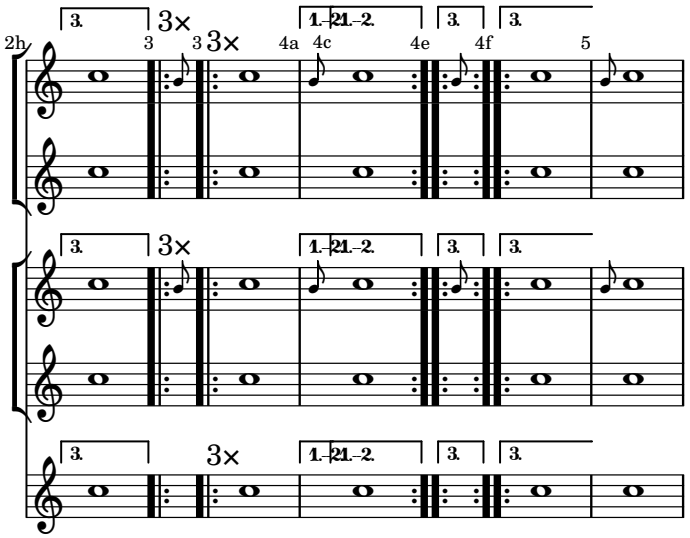


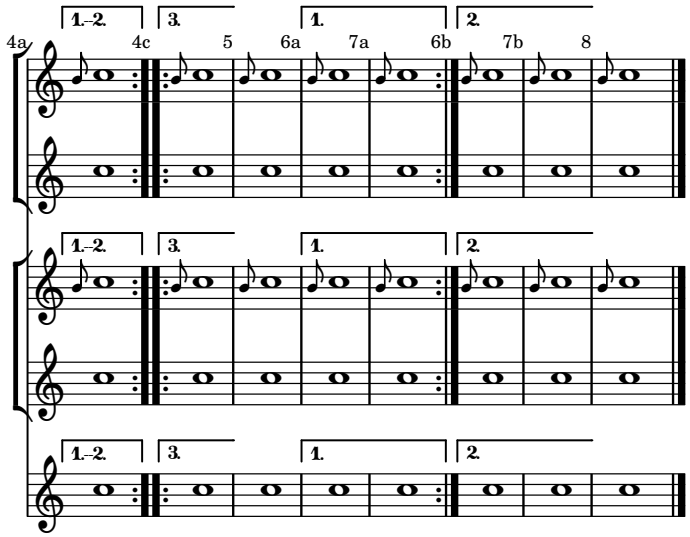
\set Score.alternativeNumberingStyle = #'numbers-with-letters

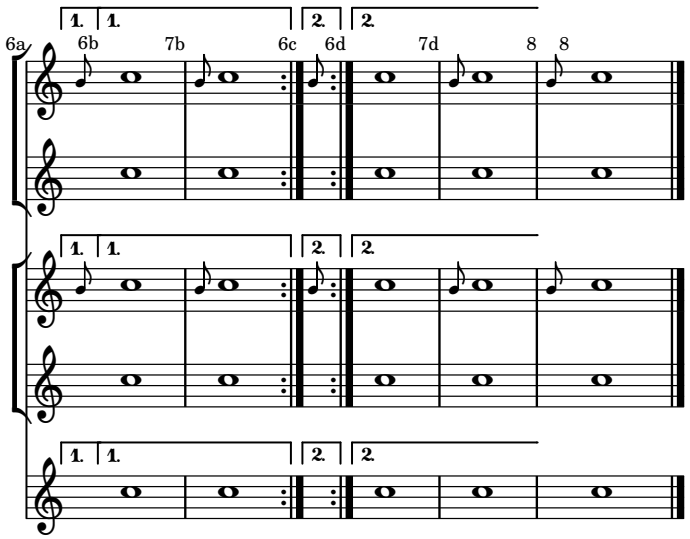


Exploding »\repeat volta« into SequentialMusic!



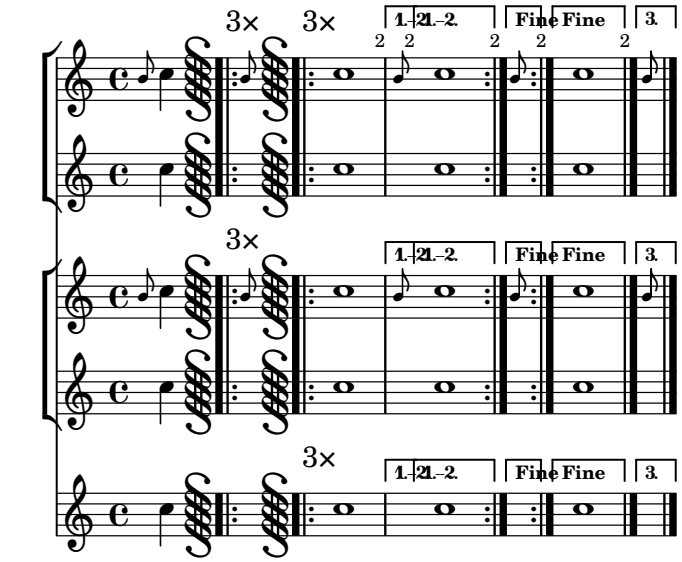


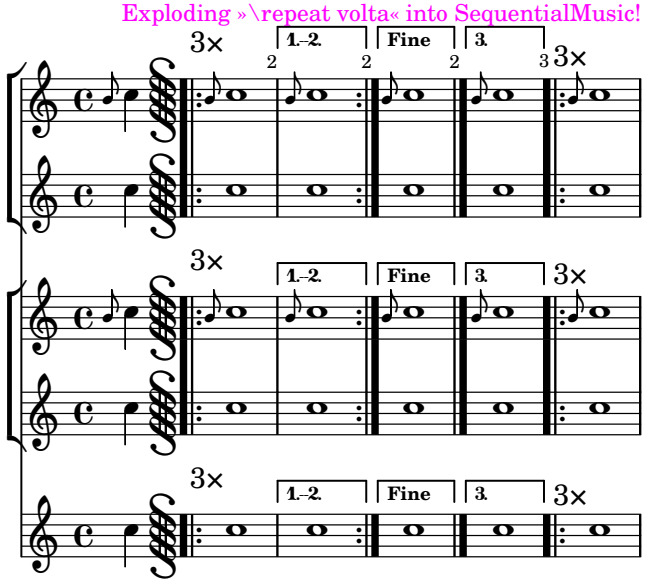


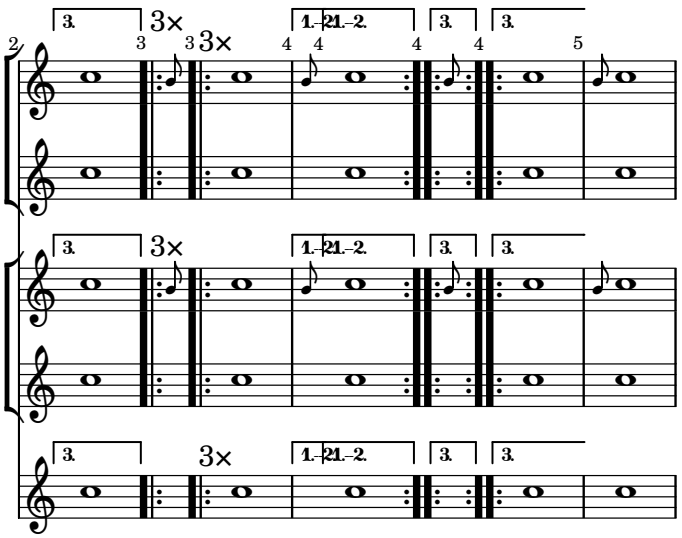


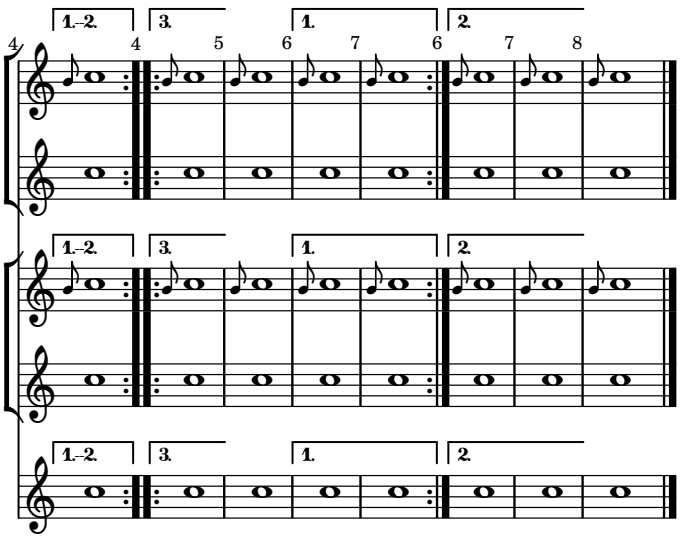
\set Score.alternativeNumberingStyle = #'numbers

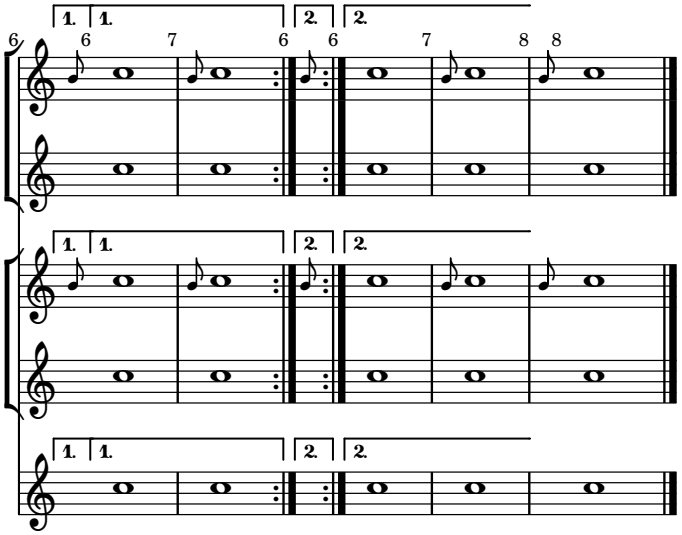
Exploding »\repeat volta« into SequentialMusic!





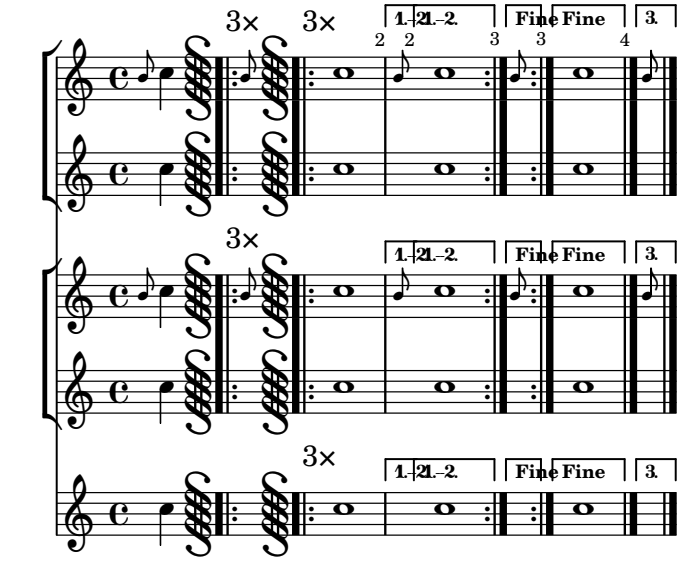


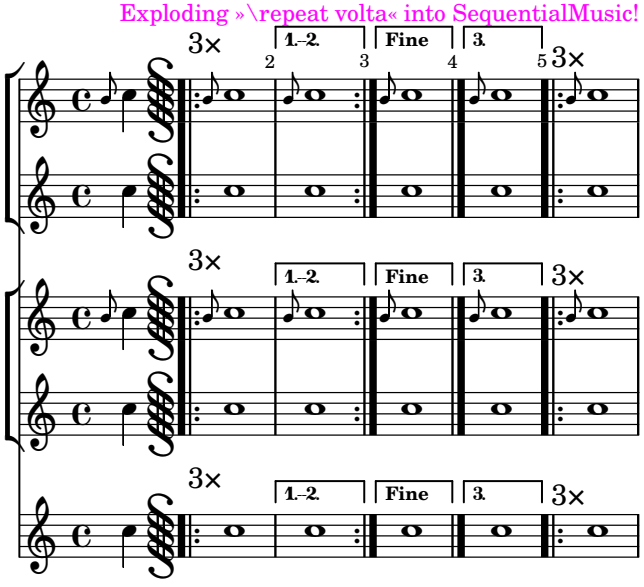


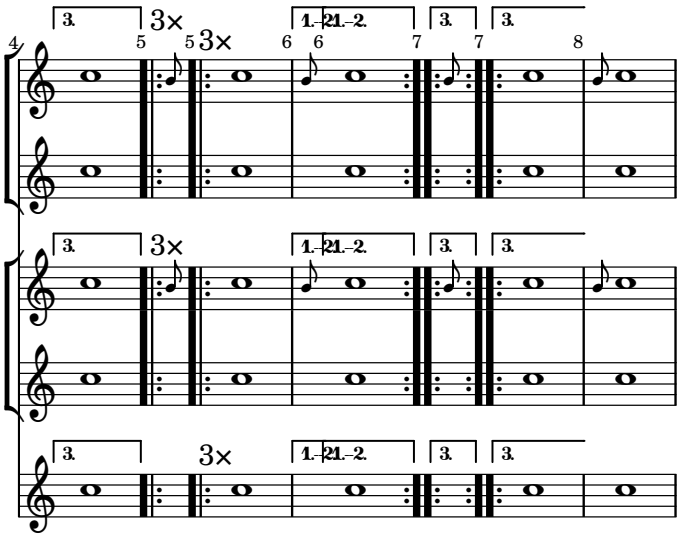


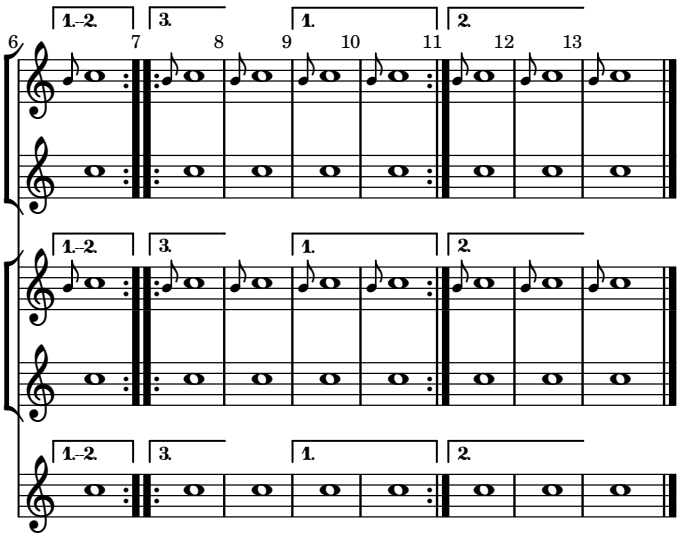
\unset Score.alternativeNumberingStyle

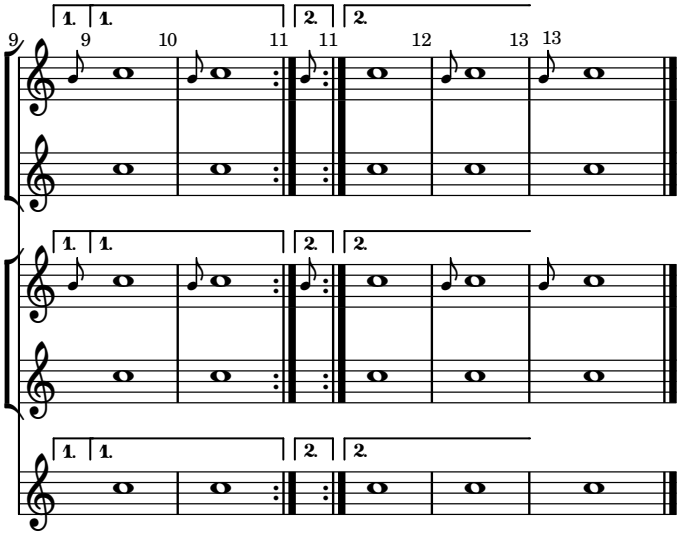
Exploding »\repeat volta« into SequentialMusic!





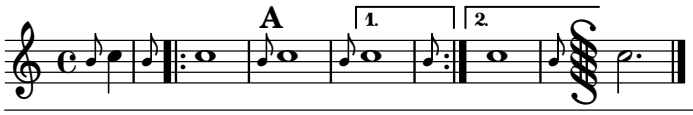


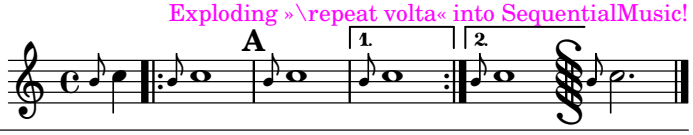




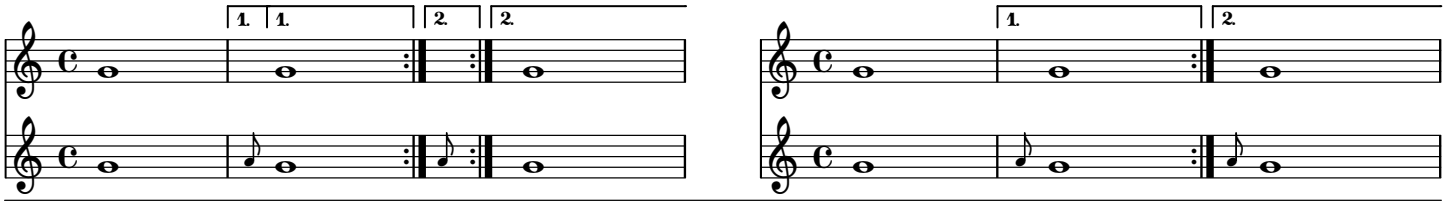
with parallel control voice inside \new Staff

Exploding »\repeat volta« into SequentialMusic!

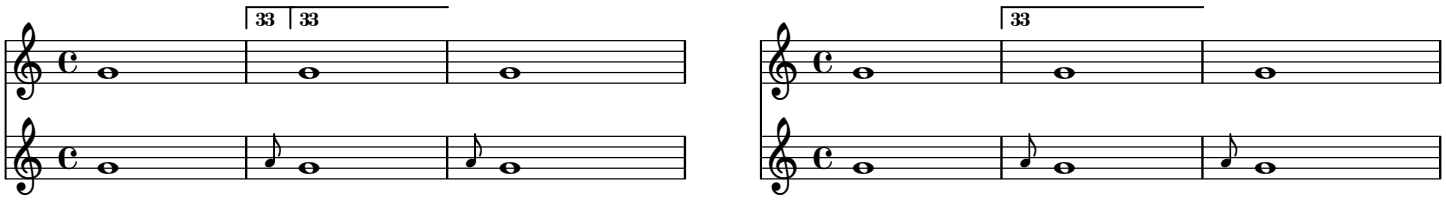




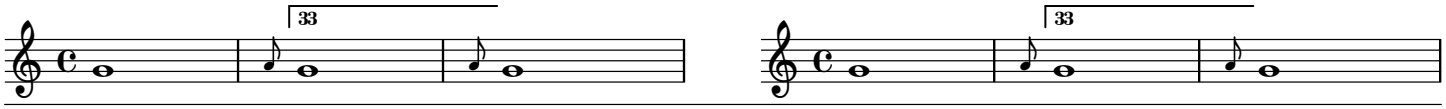
\repeat with simple \alternative



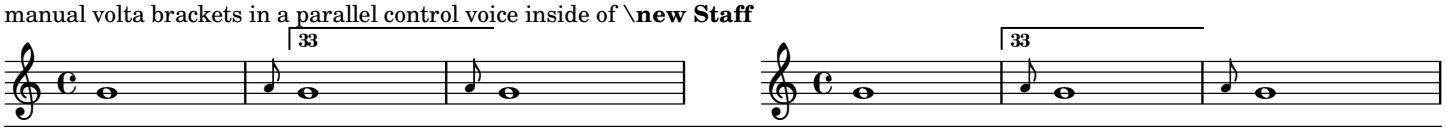
manual volta brackets in both staves



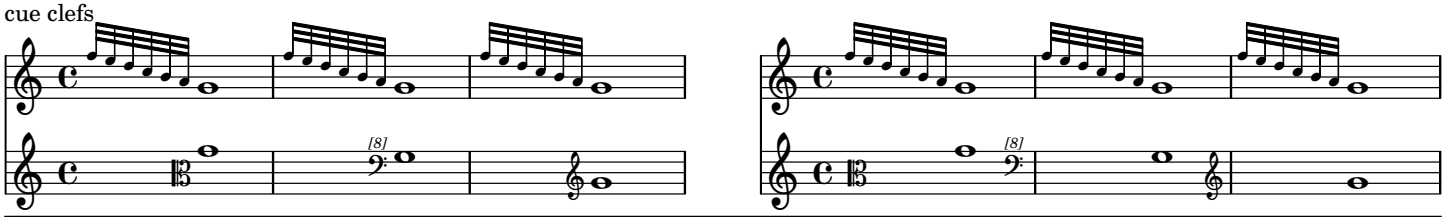
manual volta brackets in a (blanked) parallel control voice, but outside of **\new Staff**
— not a target of the \experimentalGraceSyncer



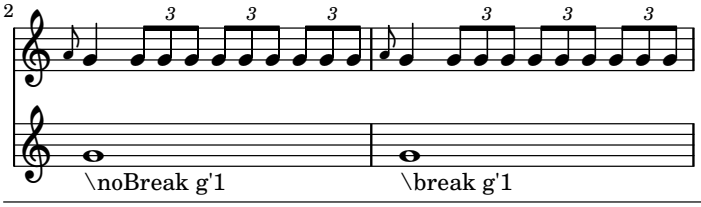
manual volta brackets in a parallel control voice inside of **\new Staff**



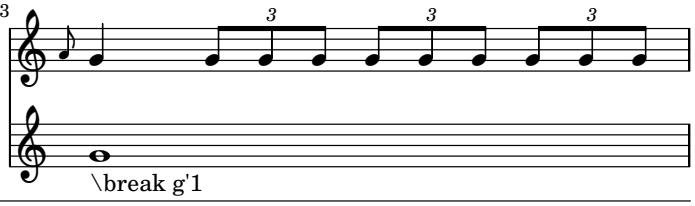
cue clefs

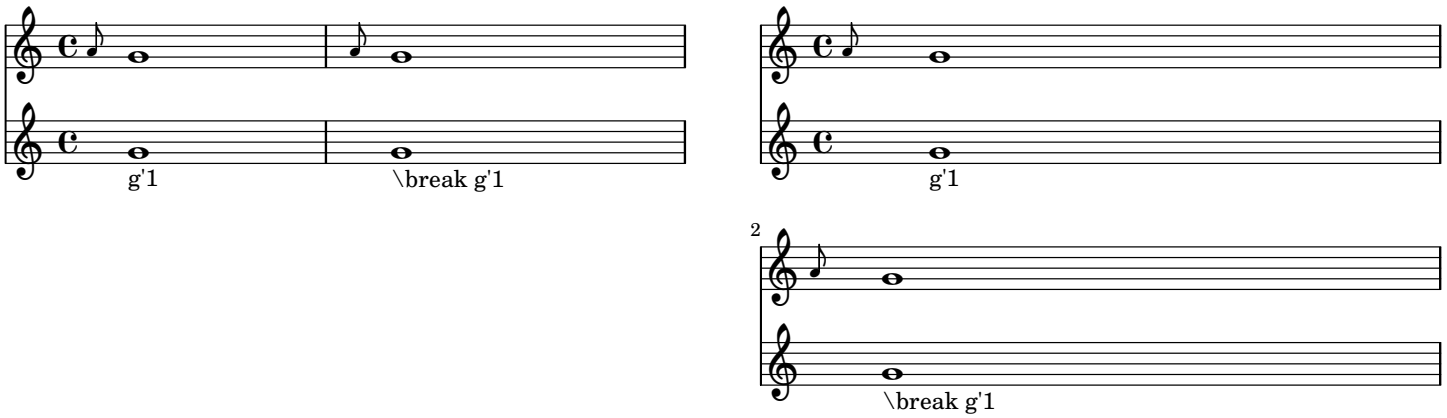


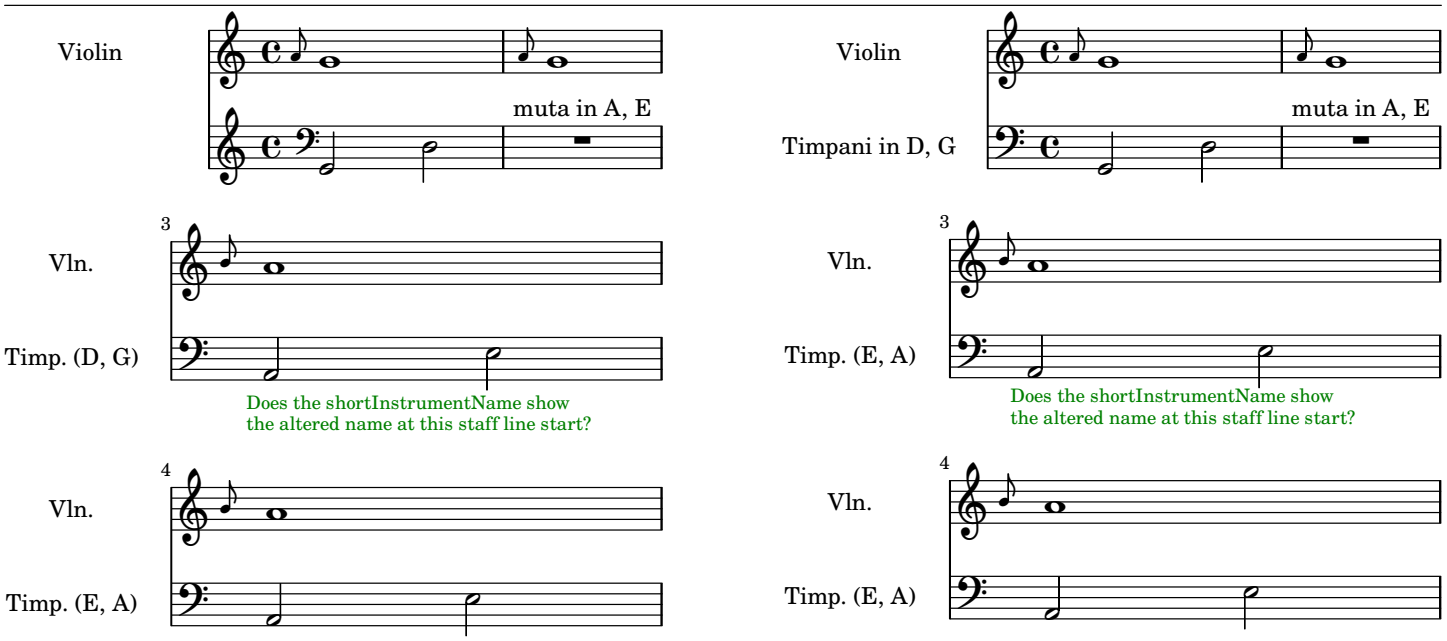
2



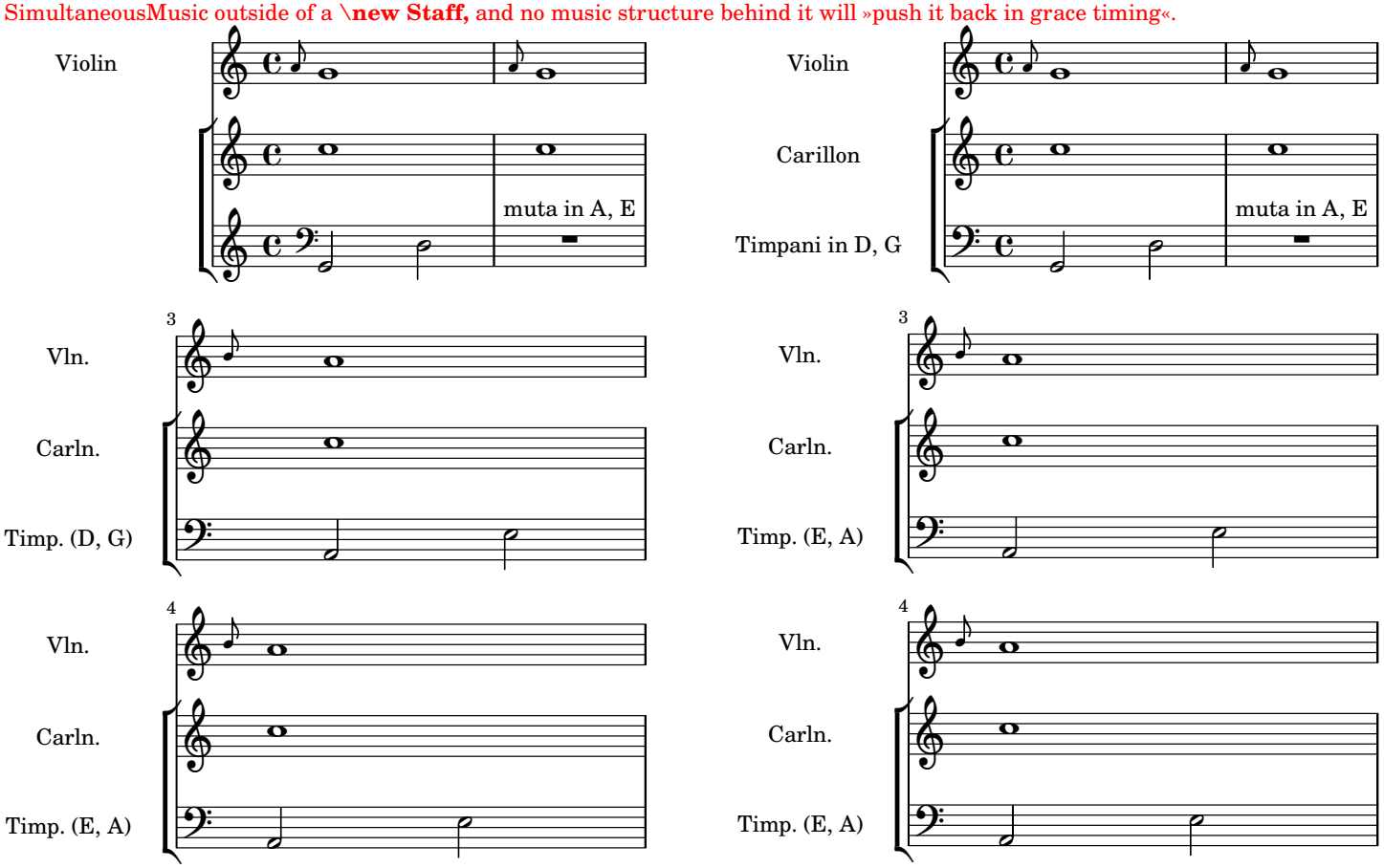
3



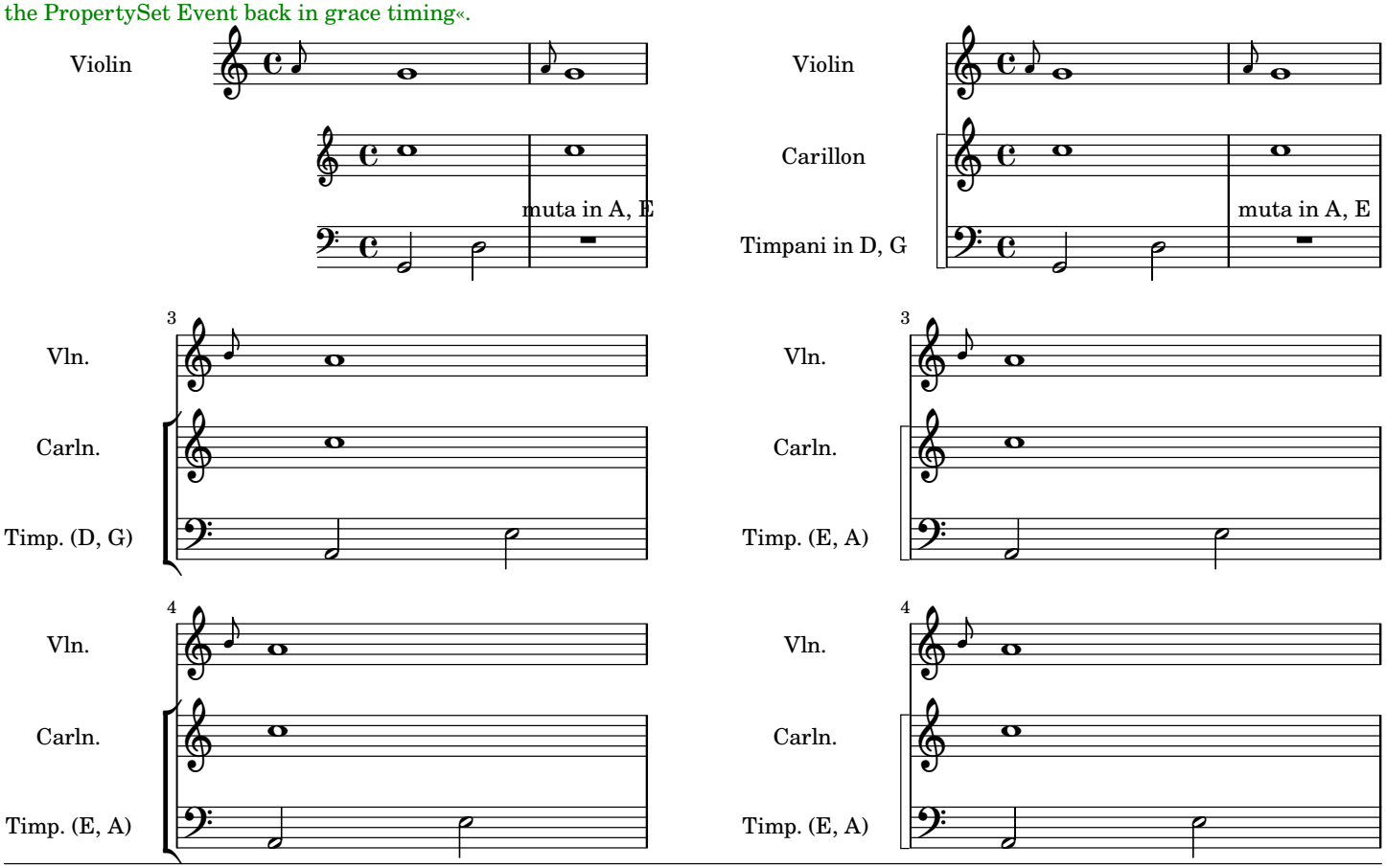




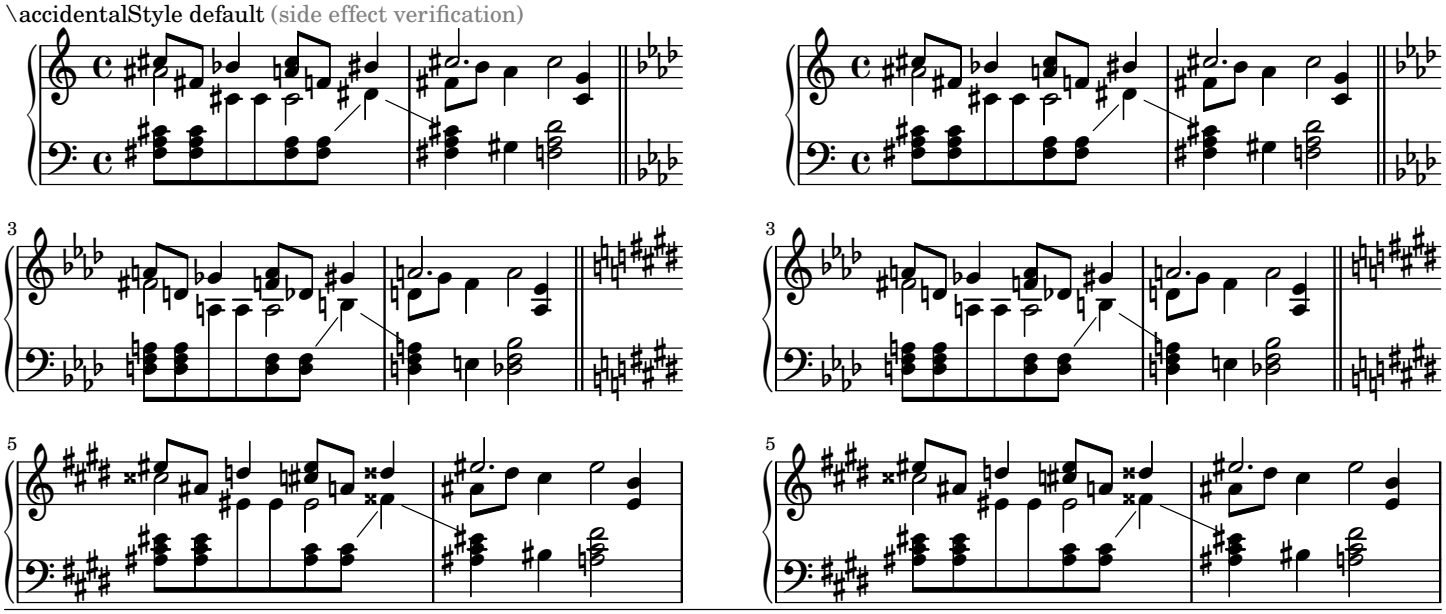
The experimentalGraceSyncer cannot sync this modification of the StaffGroup.systemStartDelimiter, because it's »running« in a SimultaneousMusic outside of a **\new Staff,** and no music structure behind it will »push it back in grace timing«.



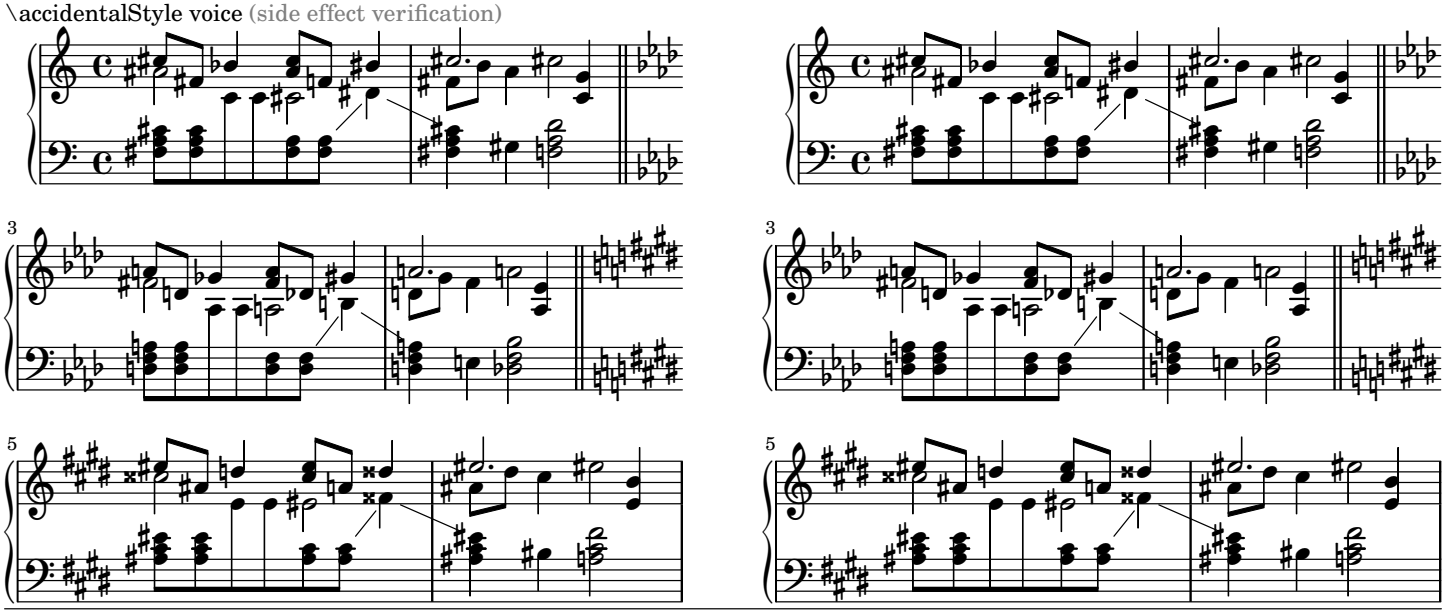
The unsynced output looks much more desolate than the previous example, but here the experimentalGraceSyncer can sync this modification of the StaffGroup.systemStartDelimiter, because it's »running« in a SequentialMusic before **\new Staff,** which will »push the PropertySet Event back in grace timing«.



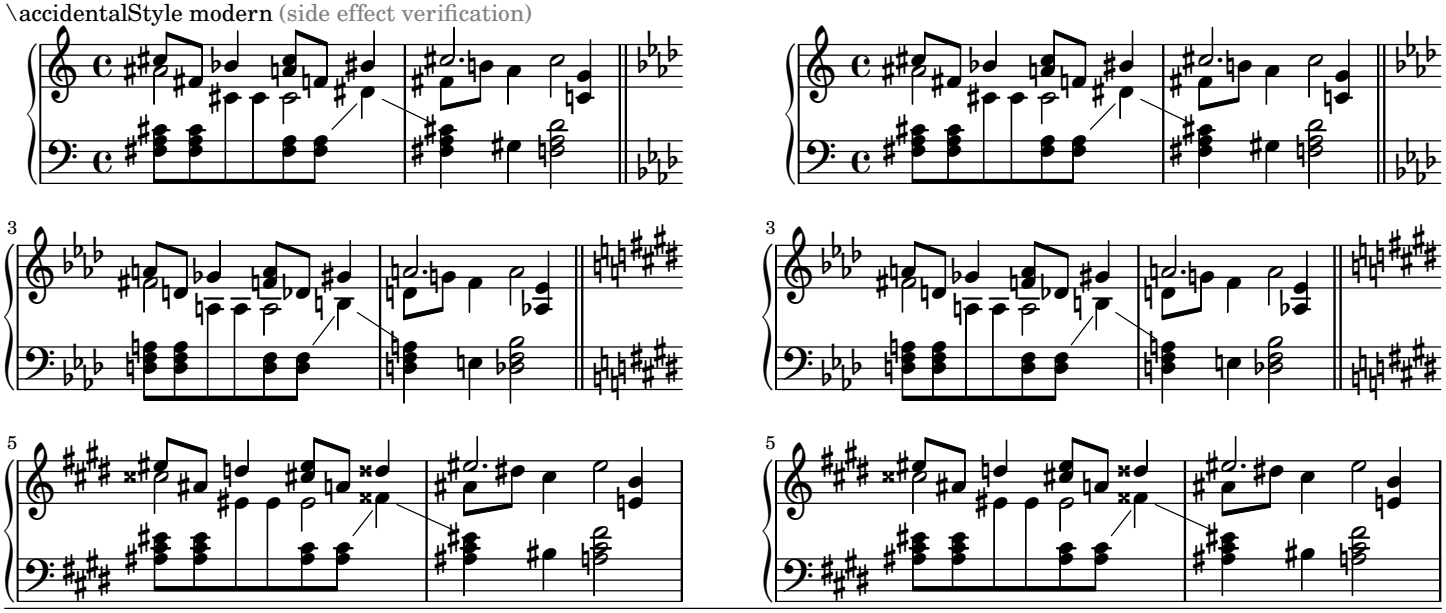
\accidentalStyle default (side effect verification)



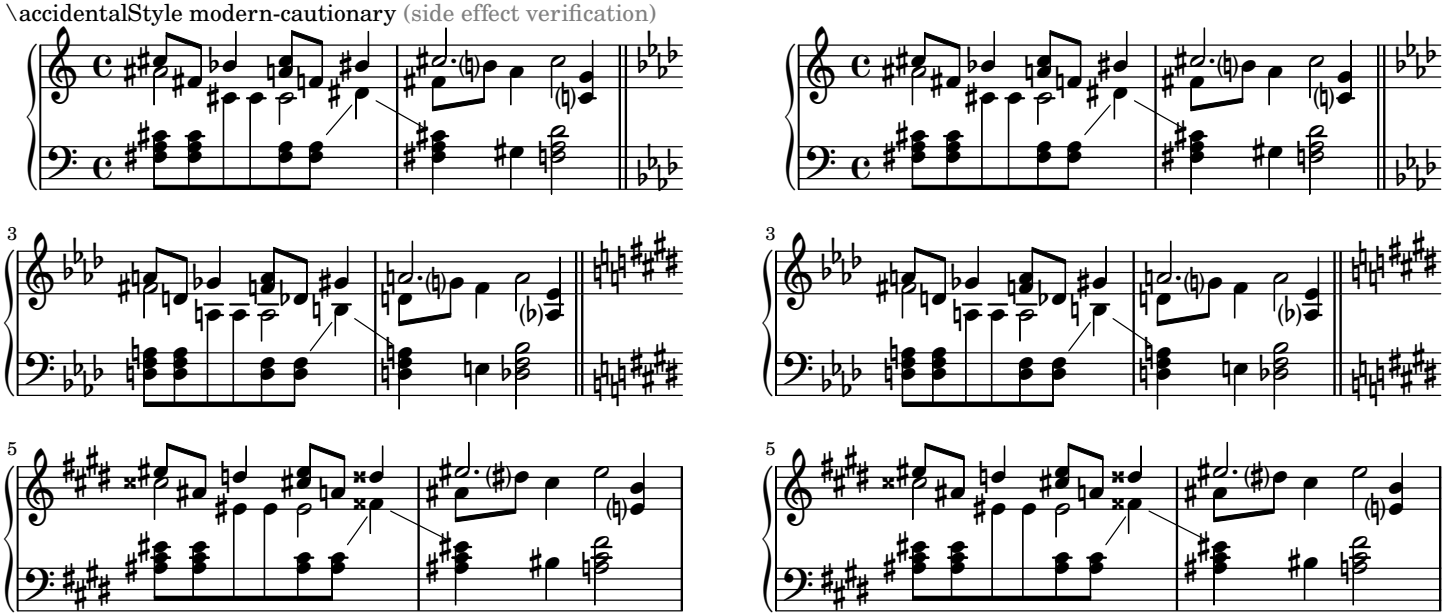
\accidentalStyle voice (side effect verification)



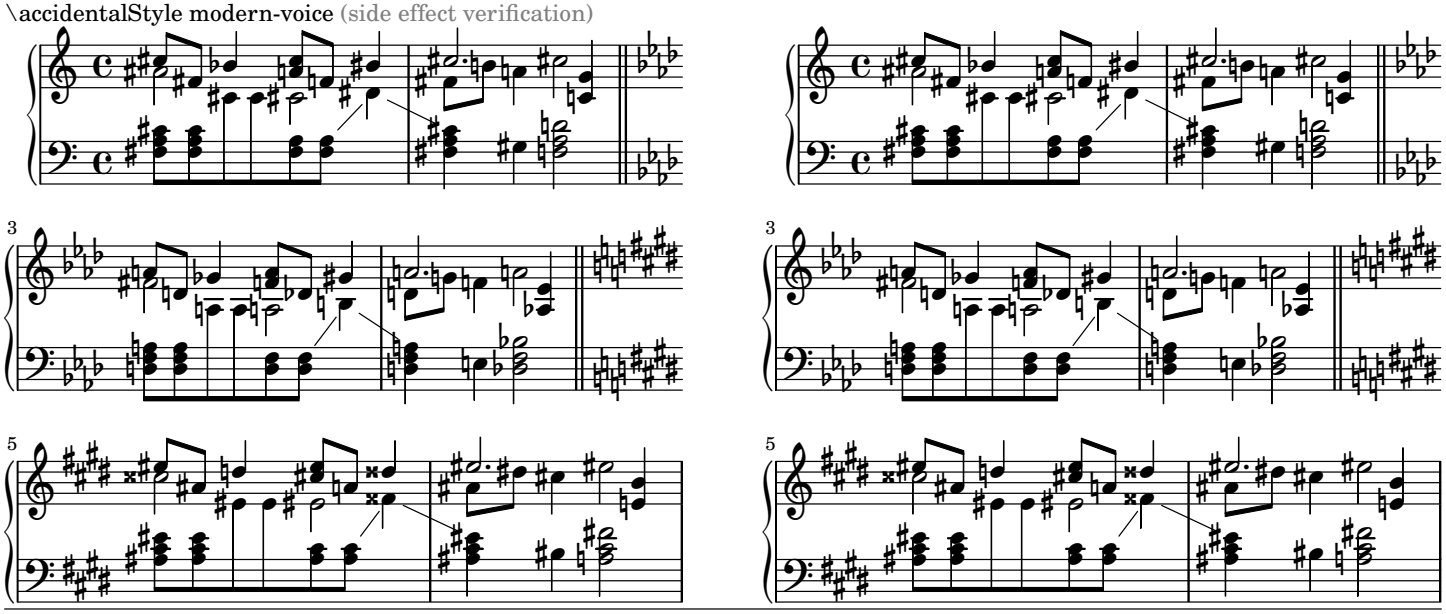
\accidentalStyle modern (side effect verification)



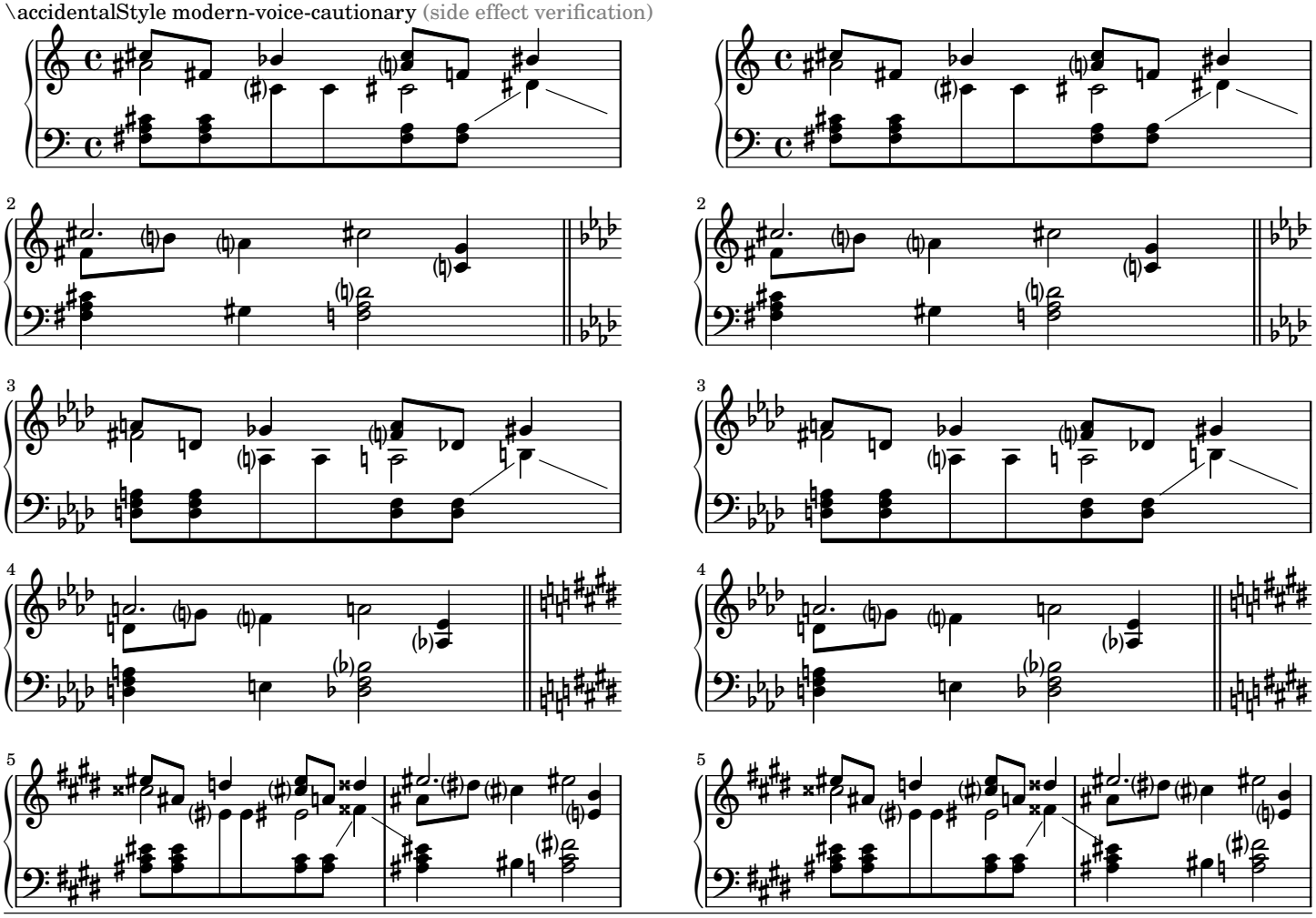
\accidentalStyle modern-cautionary (side effect verification)



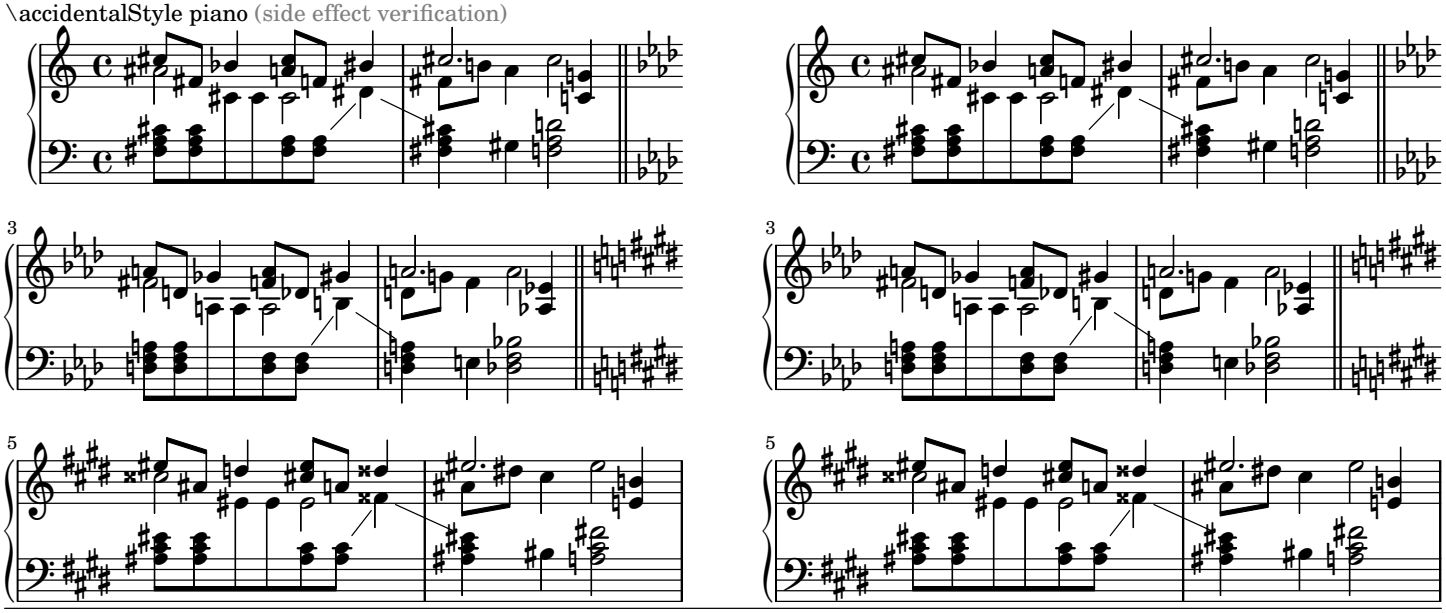
\accidentalStyle modern-voice (side effect verification)



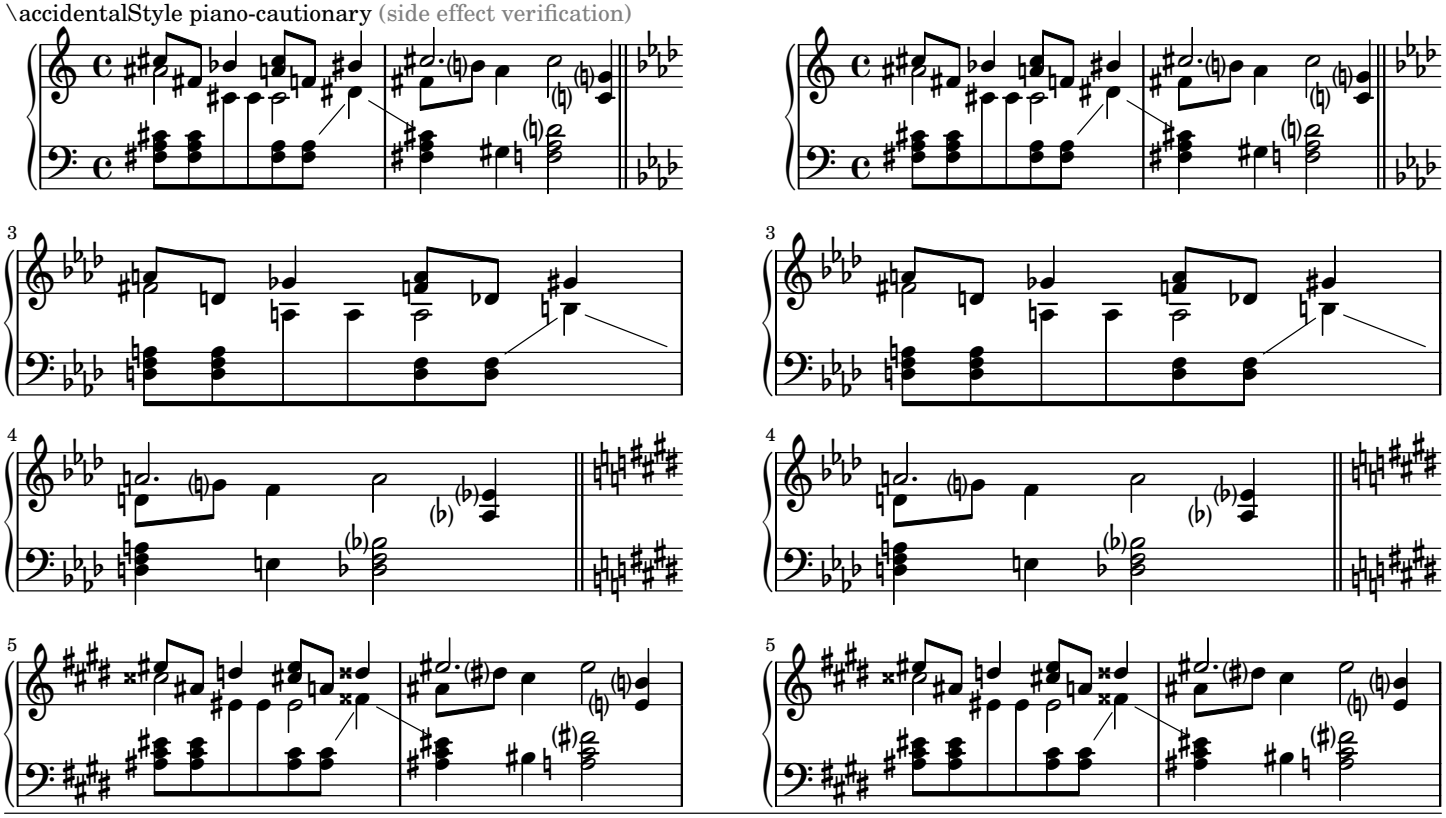
\accidentalStyle modern-voice-cautionary (side effect verification)



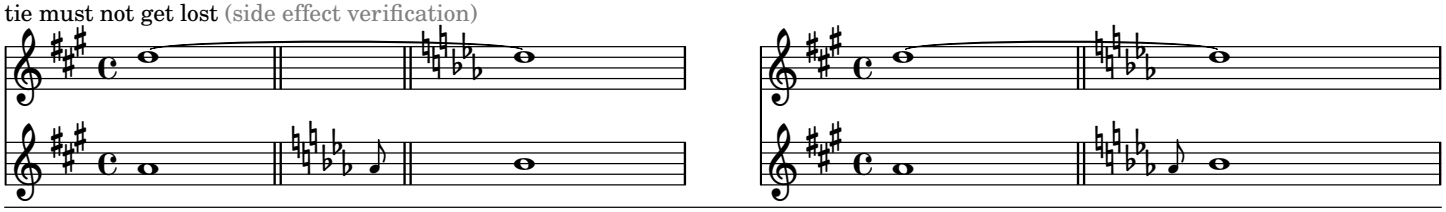
\accidentalStyle piano (side effect verification)



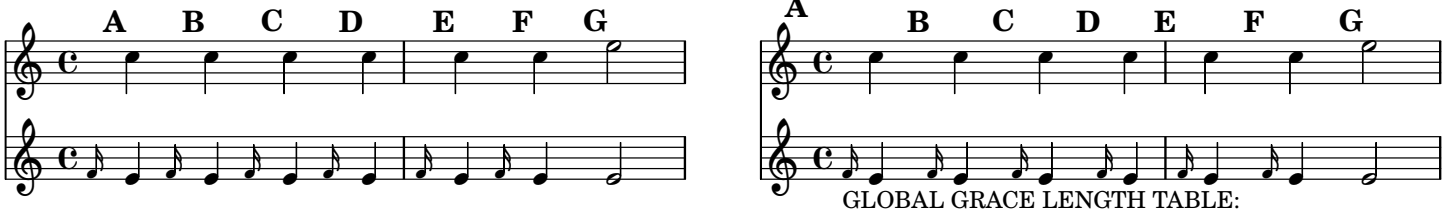
\accidentalStyle piano-cautionary (side effect verification)

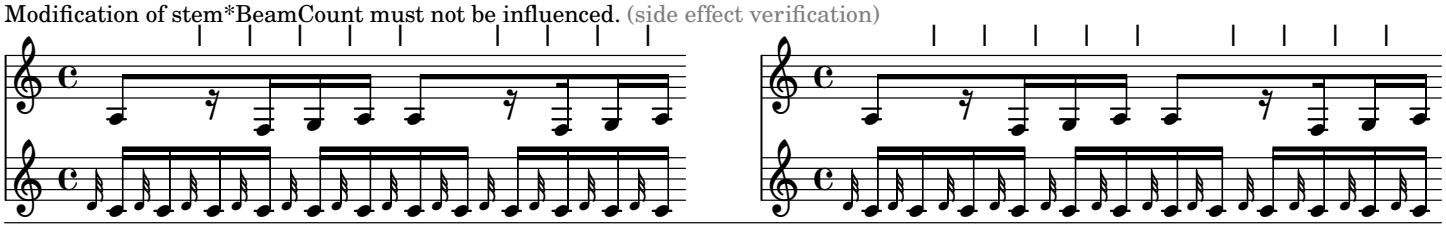


tie must not get lost (side effect verification)



Are repeats unfolded correctly, i.e. all grace offsets are recorded? (side effect verification)



GLOBAL GRACE LENGTH TABLE:
0: #<Mom 1/16>
1/4: #<Mom 1/16>
1/2: #<Mom 1/16>
3/4: #<Mom 1/16>
1: #<Mom 1/16>
5/4: #<Mom 1/16>

Modification of stem*BeamCount must not be influenced. (side effect verification)