\shiftOttavaBracket

NOTE: This whitepaper is not affiliated with nor validated by the GNU LilyPond project. It is not a substitute for official documentation. Information within is based on observed behavior of the program and might not align with the intended design. For reference, the version of LilyPond used was 2.19.83.

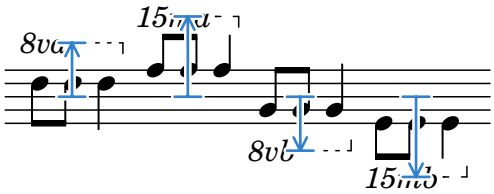
Overview

We would like a method for shifting an OttavaBracket away from its associated staff by a specified distance while preserving as much of the automatic collision avoidance behavior.

There are several obstacles to overcome. An OttavaBracket's final vertical position relative to the staff is not determined by any single grob property. Shifting the bracket will require modifying multiple properties, some of which will have side-effects that must be compensated.

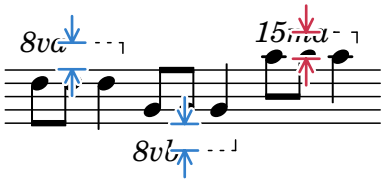
Y-offset

The Y-offset property is the distance, in staff spaces, between the reference points of an object and its parent. The reference point of an OttavaBracket aligns with the horizontal segment of its bracket symbol. The parent of an OttavaBracket is the VerticalAxisGroup, whose reference point coincides with the middle of the StaffSymbol.

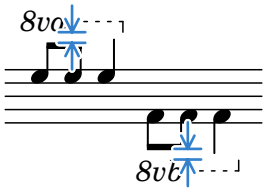


By default, the OttavaBracket does not specify a fixed value for Y-offset but rather uses a procedure to compute it. This procedure considers the position of some grobs, including NoteHeads and Stems, to determine where the bracket should be placed. Several associated properties are consulted.

The staff-padding property specifies the *minimum* distance between an OttavaBracket's reference point and the nearest extent of the associated StaffSymbol. Note the presence of nearby grobs may result in the OttavaBracket being further away than this amount, as shown in the last scenario below.

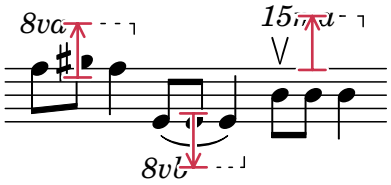


The padding property specifies the *minimum* distance between the OttavaBracket and neighboring grobs.



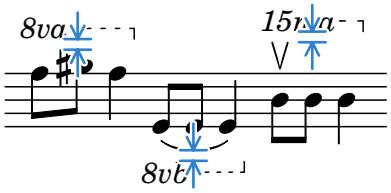
The default procedure for computing an OttavaBracket's Y-offset does not factor in all grobs. Instead, Accidentals, Slurs, and Scripts are among those that are considered in a later round of processing. Should the value of Y-offset be too small, the bracket will be offset accordingly to avoid collisions.

In the examples below, the arrow being overlaid visualizes only the computed value of Y-offset. Its base *should* align with the middle of the StaffSymbol; however, these OttavaBrackets were moved.

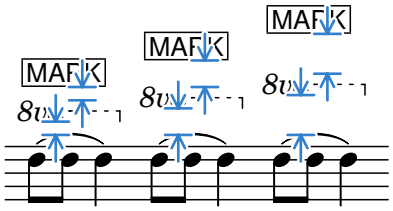


outside-staff-padding

The outside-staff-padding property, similar to the padding property, is consulted for shifting an OttavaBracket to avoid being too close to other grobs that are outside the staff.

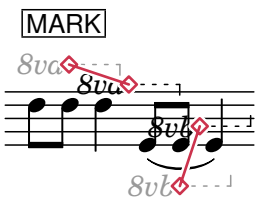


It is important to note that outside-staff-padding applies to both sides of an OttavaBracket, affecting the spacing with grobs on the side opposite to the StaffSymbol, such as RehearsalMarks. For each example below, outside-staff-padding is being increased by one staff space.



extra-offset

Unlike the properties previously discussed, extra-offset is only applied after all layout logic has been performed. It represents an arbitrary shift in position that does not account for potential collisions. As shown below, extra-offset is a two-dimensional value; though we will only be interested in vertical adjustments.

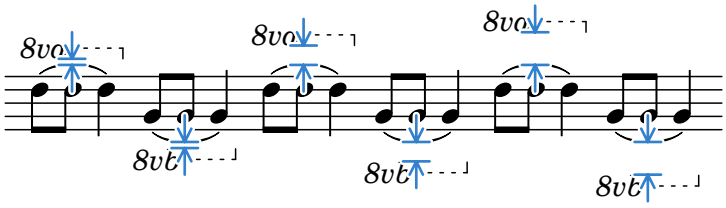


First Attempt

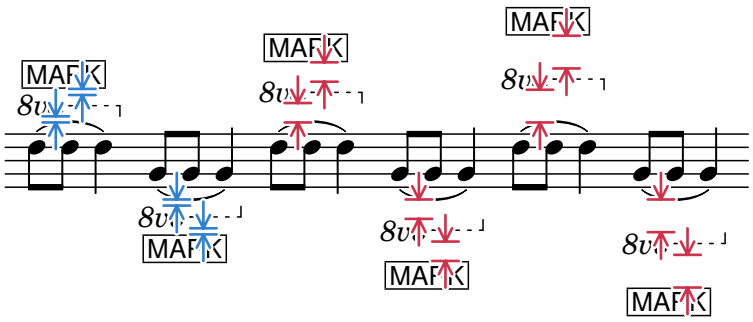
The simplest way to shift our OttavaBracket would be to \offset the outside-staff-padding property. Unfortunately, OttavaBracket does not have a base value that \offset can build upon. We can work around this issue by noting that, internally, the value 0.46 is used as the default outside-staff-padding. Instead of \offsetting the property, we will \override it after manually computing the value.

```
shiftOttavaBracket =
#(define-music-function
  (amount) (number?)
  #{
    \override Staff.OttavaBracket
      .outside-staff-padding = #(+ 0.46 amount)
  #}
  )
```

NOTE: For the remainder of the whitepaper, the examples will be testing with no shift, a shift of one staff space, and a shift of two staff spaces.



Initial testing seems promising, however we need to remember that the outside-staff-padding property affects spacing on both sides of the OttavaBracket. Let us add in some RehearsalMarks to see the impact of \shiftOttavaBracket .



The RehearsalMarks are moving by *twice* the amount we requested the OttavaBracket to move. This is because the extra amount added to outside-staff-padding exists on both sides of the bracket. Since we cannot selectively apply our extra padding to just one side, we will need a clever trick.

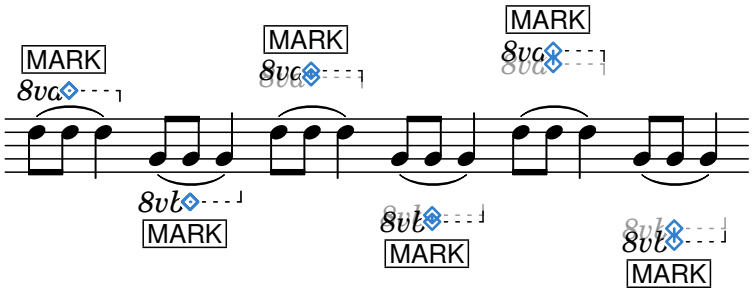
The first thing to recognize is that we should only be adding *half* of the requested amount to outside-staff-padding. This ensures that a grob like RehearsalMark will be moved the correct distance. However, doing this means the OttavaBracket only moves by half of what we need. This is where extra-offset can be used, as it will take care of moving our bracket the rest of the way.

On its own, extra-offset would not avoid collisions; but we have adjusted outside-staff-padding to ensure other grobs will already be moved sufficiently far to accommodate the additional nudging.

Second Attempt

In review, the method now involves applying half of the requested shift amount to outside-staff-padding and half to extra-offset. Note that care must be taken to ensure extra-offset is offsetting the proper direction.

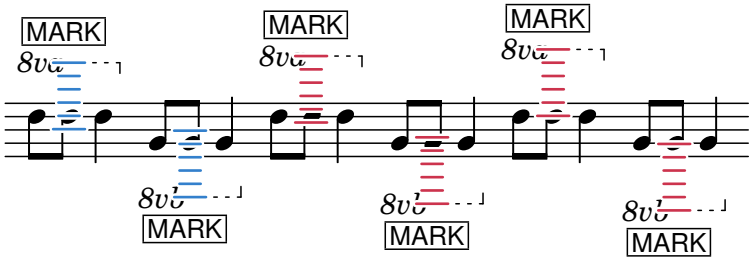
```
shiftOttavaBracket =
#(define-music-function
   (amount) (number?)
   (let ((half (/ amount 2)))
     #{
       \override Staff.OttavaBracket
         .outside-staff-padding = #(+ 0.46 half)
       \override Staff.OttavaBracket
         .extra-offset = #(lambda (grob)
           (let ((dir (ly:grob-property grob 'direction)))
             (cons 0 (* dir half))))
     #}
   ))
```



Testing shows improved results, where both the OttavaBracket and RehearsalMark appear to move as a single unit, maintaining their normal separation.

Thus far, we have only considered the situation when Y-offset is not large enough to avoid collision. In order for \shiftOttavaBracket to be useful, it must work also when the bracket is being influenced by Y-offset alone.

To see what happens with a larger Y-offset, let us increase the staff-padding property, which will in turn increase Y-offset. We will also remove the Slurs, so there is plenty of room between the bracket and the staff.



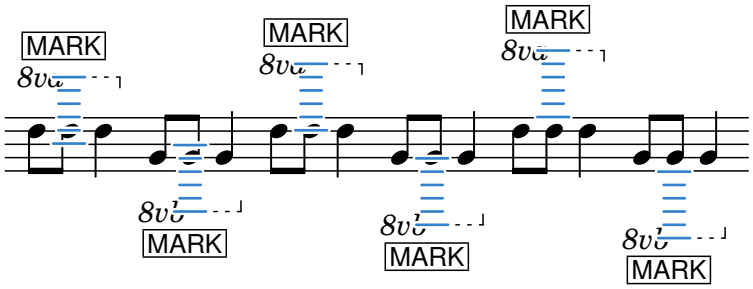
Since the OttavaBrackets are far enough away from any grobs, the increase of outside-staff-padding has no impact on the bracket itself. There is nothing to push against. With only extra-offset applying, the brackets have moved half of what we need.

However, we have a good idea for our next improvement. We should only need to \offset Y-offset by the remaining distance to get the OttavaBrackets in the correct spot.

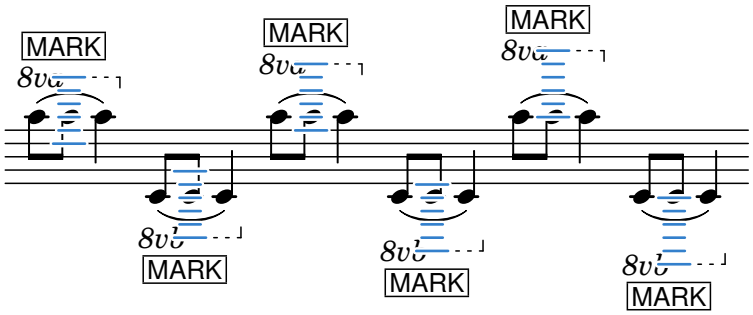
Third Attempt

Since Y-offset is a signed value, we cannot simply \offset it by a fixed amount. This would only work for positive values. LilyPond provides grob-transformer, a handy construction that makes it easy to modify a base value with a custom procedure, even when that base value might be the result of another procedure.

```
shiftOttavaBracket =
#(define-music-function
   (amount) (number?)
   (let ((half (/ amount 2)))
     #{
       \override Staff.OttavaBracket
         .Y-offset =
           #(grob-transformer 'Y-offset
              (lambda (grob orig)
                (let ((dir (ly:grob-property grob 'direction)))
                  (+ orig (* dir half)))))
       \override Staff.OttavaBracket
         .outside-staff-padding = #(+ 0.46 half)
       \override Staff.OttavaBracket
         .extra-offset = #(lambda (grob)
           (let ((dir (ly:grob-property grob 'direction)))
             (cons 0 (* dir half))))
     #}
   ))
```



We can also verify that this new function still behaves properly when nearby grobs have pushed the OttavaBracket further than Y-offset.



Closing

In summary, we have produced a utility function that properly adjusts grob properties to shift an OttavaBracket away from the staff by a specified amount.

What are the next steps? First would be testing the function given real-world projects. These artificial scenarios could be hiding bad behavior. Second would be ensuring the function can cope with other adjustments. For instance, the function assumes that OttavaBracket will not have outside-staff-padding set, thus it can safely base its computation on the known default value.