We found with use of BigDecimal that we were often needing to compare BigDecimal with zero or initialize BigDecimal as 0 for a default value. This introduces a bit move overhead than is typically needed, since a simple BigDecimal.new("0") will suffice without any additional arguments. This BigDecimal instance does not change, so it makes sense to make it a constant.

In Java, they found that constants for zero, one, and ten were useful (see http://docs.oracle.com/javase/7/docs/api/java/math/BigDecimal.html#ZERO), so I thought it would be good to submit a patch for Ruby 2.3.0, just to see whether others would find this useful. I understand that we don't want to have constants for every possible value, and that it might be helpful if these had the same object id across ruby implementations similar to lower integer values having the same object id, however I don't think it is necessary or useful to compare object_id's for BigDecimal value comparison- the point is only not to have to retain and create new instances of BigDecimal for ZERO, ONE, or TEN.