**Readme document to accompany reduction data, 8 March 2013**

This data arises from the paper ‘Schenkerian analysis by computer: a proof of concept’ published in Journal of New Music Research, v.39 (2010), pp. 269-289 (preprint at <http://www.lancs.ac.uk/staff/marsdena/publications/ProofOfConcept.pdf>).

The data show the ‘segments’ (i.e., collections of simultaneous notes) of reduction trees of short extracts of music. The trees are represented in breadth-first manner, with shorter branches extended by copying the last segment so that all branches are the same length. The rows of data therefore correspond in a way to the levels of a Schenkerian analysis, but they might not correspond well to the levels which an analysis would actually show. No attempt has been made to determine the most appropriate level of reduction to use on each branch in order to present meaningful and coherent levels in the data; the transformation from trees to levels has been entirely mechanistic. The copying of segments to make all branches the same length means that the bottom row of each reduction shows the segments of the actual extract (the ‘surface’).

Each segment is represented by the following elements in sequence, followed by a space:

1. a number representing the duration of the segment, as a multiple of the greatest common divisor of all the durations in the extract;
2. an opening square bracket;
3. representations of the notes in the segment, from lowest to highest, separated by spaces;
4. a closing square bracket.

Notes are represented by the following elements in sequence:

1. possibly an underscore character to indicate that the note is tied over from the preceding segment;
2. an upper-case letter A-G;
3. possibly ‘b’ to indicate flat, or ‘#’ to indicate sharp; no character here represents natural;
4. a number to indicate the octave: 4 indicates the octave beginning at middle C.

All reductions are binary trees, and the duration of a parent segment is the sum of the durations of its children. It is thus easy to reconstruct the trees from the breadth-first representation by keeping track of the durations.

Each of the five folders contains data concerning the reduction of one short extract from a Mozart piece for solo piano. In some cases the extracts have been slightly altered, as explained in the paper. ‘Mozart5’ refers to an extract from K.333, ‘Mozart3’ from K.570, and ‘Mozart2’ from K.494. The other two are identified by their Köchel numbers.

Each folder contains two sets of data. The folder ‘fullTree’ shows the reduction to a single root segment (i.e., a single binary tree). The folder ‘fromUrsatz’ shows the reduction to an Ursatz rather than to a single root (i.e., a sequence with 3-2-1, 5-4-3-2-1 or 8-7-6-5-4-3-2-1 (no cases in this dataset) in the top notes of segments and I-V-I in the bass). The trees from which these representations are derived are the same as in the corresponding ‘fullTree’ case, but the levels represented (the rows) might not correspond. In some cases more than one Ursatz can be found in a tree (e.g., 5-4-3-2-1 might reduce to 3-2-1) and in such cases, both reductions are represented in the data, separated by a blank line. The dataset also contains a log file which includes a report of each of the cases where there are multiple Ursätze and so multiple representations.

In each set of reductions, the data contains the following:

1. A ‘preferred’ reduction which is the one I considered to be the best reduction of the extract, based on my knowledge of Schenkerian analysis.
2. Two to four sets of ‘conforming’ reductions. Each set contains 1000 reductions which conform to a previous analysis of the extract, in the sense explained in the paper. The previous analyses did not give complete binary trees (Schenkerian analyses generally do not), so there is freedom about the portions of the tree not represented in the previous analysis, and previous analyses often ignore inner voices, so there is freedom also about the inner notes of segments. It is for this reason that there can be many reduction trees which conform to a previous analysis. There is no guarantee that the 1000 reductions do not contain duplicates.
3. A set of 1000 ‘free’ reductions selected at random from all those reductions which contain an Ursatz and are possible under the definition of Schenkerian analysis embodied in the reduction software and explained in the paper. Again, there is no guarantee that these sets do not contain duplicates.

Broadly speaking, the ‘preferred’ reduction can be considered to be the best reduction, and the ‘conforming’ reductions to be good reductions, while the ‘free’ ones are likely to be poor reductions. However, it has to be acknowledged that the definition of ‘best’ here depends on my personal judgement while the definition of ‘good’ depends on the judgements of other acknowledged experts.

Alan Marsden