Examiners' commentaries 2016–17

CO3346 Sound and Music – Zone B

General remarks

Overall performance on this paper was reasonably good: however there were a number of weak papers with around 20% of candidates not achieving a pass grade. There was, in contrast, one candidate who obtained nearly full marks for the paper.

What follows is a brief discussion of the individual questions on this paper, with hints towards the answers expected by the examiners.

Comments on specific questions

Question 1 : Algorithmic composition

This question was a popular one, with candidates able to give reasonably good answers.

Part (a) simply required an understanding that the term 'meta' generally involves a level of abstraction of the key concept itself. So, algorithmic composition involves 'composing' a process that can compose music.

Part (b) required description, and not simply listing. So, one example could be swarm music. It was also essential to include a brief description of the approach, such as that the properties of particles in a swarm are used to generate a musical sequence. Another example might be the dice game, where segments of a musical piece can be assigned to numbers on the dice and a new composition can be generated by rolling the dice.

It is generally useful to read ahead in each question. Part (c) is a good example of why this is the case: candidates were required to discuss the likelihood of good music being produced by one of the approaches identified in their previous answer, and so choosing one which facilitates your discussion would be a good idea.

For part (d), a large proportion of candidates listed properties rather than components of a swarm music system. In addition, some did not describe the components. For example, one component might be the particles: these generate a data pattern. Another component is particle-music mapping, the function of which is to convert a data pattern into music.

For part (e), some candidates discussed advantages and disadvantages of using Java for algorithmic composition, which is not what the question asked. For this question, candidates should have argued directly for either whether it is, or isn't useful for algorythmic composition, giving reasons in support of the view chosen.

Question 2 : Musical interaction

A question chosen by about half of the candidates, and not always answered well.

While part (a) required the description of two aspects of music relating to pitch and to rhythm, some candidates described pitch and rhythm themselves. Rhythm concentrates on the time dimension of music, and is seen most clearly in drum and percussion music in general. It underlies nearly all Western tonal music, from Beethoven to the blues. An individual note, say one note played on a piano, is associated with a given pitch, associated with the physical frequency of sound waves in the air as each note on the piano or guitar has a different pitch. The way that pitches are organised together is fundamental to many musics.

Part (b) required both a description of what a pitch interval is, and also a justification regarding why it might be more useful than absolute pitch for comparing melodies. Justification could include that as pitch intervals describe the relationships between notes rather than simply the notes themselves, they could be more useful. Also, with intervals, it is possible to compare melody sequences with different starting notes.

For part (c), most candidates knew that beat tracking is used to detect the pulse in a piece of music.

For part (d), some candidates were able to identify that the descriptors needed are the onsets, so that the algorithm would know where the notes in the piece fell.

Answers for both part (e) and (f) were generally weak. For part (e), candidates understood that the beat might fall at any of the levels in the metric hierarchy and this means the challenge is to find out at which level the beat falls. However, not many supplied a diagram to show what is meant by a metric hierarchy.

Question 3 : Music Information Retrieval

This was a popular question, chosen by almost all candidates and answered generally well.

For part (a), examiners awarded marks for any three relevant tasks: these could include multiple version handling; music identification; performer identification; music-speech segmentation; and others. To obtain full marks, the explanation of specificity level was required (an understanding of the importance of true negatives would help with this), as well as a correct ordering of specificity.

Most candidates were able to describe what recommender systems are used for, in response to part (b).

For part (c), a reasonable approach could be to detect all people listening to the company’s music; select other people with similar play lists to these people; and suggest that those people listen to the company’s music. Full marks required a description of how each of the identified subtasks might be achieved.

Part (d) required that candidates understood that the purpose is to get documents that are similar to a sent document in terms of the actual content, rather than making use of the meta-data, *e.g.* title, author. A complete answer needed to include that the inputs would be a document and a database of other documents; while the output would be a list of documents from the database sorted by similarity to the input document.

Part (e) was a straightforward question, based directly on material in the subject guide.

Question 4 : Pure Data

This question was chosen by many candidates, but not always answered very well.

Part (a) required an understanding of the properties of each programming environment, and the kinds of things musicians might prefer. Some examples could be: Java is not a great idea as the syntax is too verbose; musicians prefer to work visually (debatable); Pure Data is more like a score; nobody anywhere is writing good music with Java; Java is quite good if you code in Processing as you can make visuals that go along with the sound.

For part (b), it is certainly the case that Pure Data is supportive of livecoding. Reasons include that you can edit a running program in Pure Data, and that you can edit a program without recompiling.

Examples that demonstrated familiarity with Pure Data, and showed understanding of how the patch might have been made were acceptable for part (c).

Answers for part (d) should show a noisy waveform fading in and out with a sine-wave amplitude envelope shape.

For part (e), not many candidates knew that for additive synthesis, multiple oscillators are added together to produce a complex tone. One reason that it is expensive is that it requires many oscillators to generate a decent sound.

Finally, for part (f), the patch should show multiple oscillators added together with different frequencies. To obtain full marks, answers needed to include labels to describe this.