Examiners' commentary 2018–2019

CO3346 Creative computing: sound and music – Zone A

General remarks

Overall performance on this paper was very good, with a number of candidates achieving a high first class mark, and only a very small proportion of fails.

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: Computational models of music cognition

Performance on this question was mostly good, particularly on the earlier sub-questions.

a. Part (a) required knowledge of the material in the subject guide. First, candidates had to be able to say that a computational model is a piece of software that implements some sort of system that attempts to emulate or model something. This is usually a model of something that exists in the real world. The second part of the question was not always that well addressed. One reason why computational models are important is that a cognitive scientist cannot observe the operation of these algorithms and representations directly in the human mind, but can constrain the design of the model on the basis of what is known about the mind. Examiners were not expecting answers specific to music cognition in this question, but required a connection to cognitive science more generally, though some candidates did not understand this.

b. For part (b) examiners expected two reasons for the creation of these kinds of models in the context of music cognition. Some possible reasons could be: so they can make computer systems that can process music in similar ways to humans to better understand humans; so they can make computer systems that can understand music; so they can generate new music or compare music, and so on. As an example of an explanation, models are the basis of scientific enquiry and they can be used to generate predictions about how a real system will respond, and then check if the model is correct.

c. Part (c) required a description of each possible representation chosen, as well as an appropriate data structure for it. As an example, music can be represented as a sequence of digital samples that are taken from a sensor that measures the sound pressure wave generated when the music is played. Samples can be represented as an array of floating point values. Another example might be a symbolic one, where music is represented as a score with notes and durations, as well as which instrument plays the notes. Here there would need to be a more complex representation, for example, note class with volume, duration and instrument type fields.

d. Responses for part (d) were often good. We want to group musical information in order to identify different sections in a piece of music or to identify different instrumental parts.

e. Part (e) was a straightforward question requiring knowledge of the Gestalt principles, and understanding which are about grouping. For full marks candidates were required to explain how they relate to musical information, which was not always done.

f. Appropriate answers for part (f) described what a statistical model actually is, as well as giving an example of one that is used in musical cognition. Most candidates could say that a two-factor model of melodic expectation is an appropriate example, though not all were able to describe what a statistical model itself is.

g. Most candidates were able to give correct answers for part (g), which is that an n-gram is a sequence of n symbols and an n-gram model is simply a collection of such sequences, each of which is associated with a frequency count.

h. Part (h) was the most weakly answered part of this question, the main omission being that some candidates did not include responses to all three questions that were asked.

Question 2: Computer music and pure data

This was a less popular question, answered reasonably well by those candidates who chose it. The main weakness was that candidates failed to provide adequate responses to all parts of the questions, thereby not obtaining the full number of marks available.

a. For part (a), candidates were asked to describe three features of the programming environment. Examiners expected a description, plus an advantage and disadvantage of its use for each feature chosen. Not all candidates included a complete coverage.

b. Similarly, for part (b), examiners expected at least five steps in the instructions for how to build a synthesizer, as well as diagrams. Steps should have included the creation of two controls with appropriate wiring to oscillator frequencies or filter, for example, and marks were given for the clarity of how these were explained.

c. To obtain the two marks available for part (c), candidates were expected to explain that sampled sound is a digital representation of a real sound wave and that it is derived (via the process of sampling) from a recording, whereas synthesised sound is generated algorithmically.

d. Finally, for part (d) many candidates were able to draw a correct Pure Data patch. However, not all candidates supplied the corresponding annotation, and therefore failed to achieve the full four marks available.

Question 3: Algorithmic composition

This was a very popular question, chosen by almost all candidates and answered reasonably well by most.

a. A common issue with part (a) was that not all candidates understood that meta-composition means working at a higher level of abstraction, and that meta-composers write software that writes music, as opposed to writing the music itself.

b. Part (b) was a straightforward question requiring candidates to name a composer who has used algorithmic composition. The most obvious response would be Xenakis. However, any other correct answers were accepted, though a couple of names offered were not actually algorithmic composers. For example, Devin Townsend, who is a rock musician but not an algorithmic composer. Some candidates correctly identified Mozart. A couple of candidates cited themselves as algorithmic composers, which amused the examiners but unfortunately they were not awarded marks for this.

c. Part (c) required some thought. Good answers showed understanding that algorithmic composition is a creative activity, with different or fewer constraints on how it might be evaluated compared with computational music in general, which has more clarity on how it might be evaluated from an engineering perspective.

d. Part (d) was answered correctly by almost all candidates. Live coding is a computer music performance technique where a coder writes and edits a program that generates music in front of an audience, normally whilst the program is running.

e. Only a couple of candidates incorrectly believed that it is possible to live code in Java. You cannot modify a running program in Java as the sound would have to stop then start again.

f. Part (f) was straightforward material from the subject guide requiring an explanation of the four concepts: (i–iv), together with application to the scenario of improvising. As long as a reasonable justification of what was claimed was given, candidates obtained the full marks available.

g. For part (g), candidates were required to apply the four concepts covered in part (f) to the concept of a swarm music system. An example answer might be that it exhibits reflection (as it is able to capture musical input from the player and play this back in a new form); it is innovative (as it is able to generate new patterns); it is autonomous (as it is able to operate under its own goals), and it is transparent (the algorithm is well described and is also visualised so its inner workings are exposed).

h. As above, for part (h), candidates were required to apply the four concepts covered in part (f) to the concept of DJing, again explaining or justifying the choices made.

Question 4: Music information retrieval

This question was chosen by many candidates, and generally answered reasonably well, though some candidates failed to answer the final part.

a. Part (a) required an understanding that specificity includes both relevance and retrieval aspects, as well as a demonstration that the candidate understood how these are evaluated.

b. Part (b) was generally answered well, though not all candidates included all aspects asked for. A genre classification system has a set of labelled items, e.g. music tracks with genre labels. It extracts features such as MFCC from the tracks, then it attempts to cluster them in feature space. New tracks can be assigned genres based on which cluster they belong to. Some candidates included appropriate diagrams, which also gained them marks.

c. For part (c), it was possible to either agree or disagree with the appropriateness of timbre for genre differentiation. Whichever candidates chose, it was essential for them to justify their answer. For example, using timbre when you have two genres with the same instruments, e.g. pop and hip hop, might get mixed up, or different types of electronic music. There are other examples where the distinction would be more obvious. So, it might work well for distinguishing between some genres but not others.

d. Part (d) was a straightforward question, though it did require candidates to relate the differences with reference to musical document retrieval, rather than general information retrieval.

e. For part (e), most candidates were able to identify and justify which approach (content or metadata) would be most appropriate for retrieving all different versions of the same track. However, not all candidates included a discussion of how a lack of complete data might impact the problem.

f. For the final part, candidates were expected to consider how a search would work, and the kinds of information it would require. Based on this, for each of the three example inputs (i–iii), candidates were expected to describe how the system would find matching documents. Many, but not all, candidates provided a coherent answer for this. Some answered for each of the three in different ways, implying that three different systems were being proposed, which was not what was expected.