A notation-based query language for searching in symbolic music

We propose a query language that is based on music notation and is designed to facilitate searching for patterns in symbolic music. Queries are encoded in an augmented version of the Music Encoding Initiative (MEI) format. The goal of our notation-based approach is to make pattern matching accessible to a wider audience of musicologists.

Previous Work

Existing approaches to symbolic music searching (Typke, Wiering, and Veltkamp 2005) can be classified into two categories. Similarity-based searching systems aim to find a certain excerpt within a large collection of works and allow "fuzzy" searching, so that even approximate matches are returned. A second class of systems aims at precise pattern-based searching and its audience is mainly musicologists. An example task for such a system is searching for parallel fifths in a corpus of polyphonic music.

Our approach is geared towards the second category: pattern-based searching. Two existing tools used for this purpose are music21 (Cuthbert and Ariza 2010) and Humdrum (Huron 2002). music21 is a programming library for manipulating symbolic music. It includes searching utilities and provides a basis on which users write their custom searching logic. Since users must know how to code in order to use music21, its accessibility is naturally limited. Humdrum does not require coding skills to use. It is comprised of a text-based symbolic music representation and a set of tools to manipulate it. Humdrum enables powerful searches but its learning curve is still steep; crafting non-trivial queries is complicated and error-prone. More importantly, there is an inherent representation mismatch: the data is music but the manipulation must be done in text.

Proposed Method

Our approach aims to bridge this representation gap by representing the query itself as musical notation. We choose to model our query language after another system that has gained widespread popularity for text: regular expressions (Thompson 1968). Regular expressions are flexible and fast to execute, and are built from a small set of primitives. We use similar primitives to form our query language, but instead of using a text-based syntax we use musical notation, which is more familiar to musicologists.

We demonstrate the merit of our approach by constructing a limited prototype system that works on monophonic music only. The system is implemented using web technologies. It displays the query and the score side by side using Verovio (Pugin, Zitellini, and Roland 2014). Figure 1 shows an example of an "exact match" query on the left-hand side and its results on the right-hand side. The query is encoded in MEI (Roland 2002), which is flexible enough to encode our notation extensions and can be easily rendered by Verovio (with minimal modification).

The query language consists of three primitives: *note*, *or* operator, and *grouping* operator. Figure 2 shows several example queries along with their encoding. Figure 2a shows a rhythm-only query where note pitches are ignored; this is specified using *tenuto* (`@artic="ten"`). In Figure 2b, note durations are ignored and only pitches are taken into account; this is encoded using stemless notes (`@stem.visible="false"`). A note can also match any accidental by specifying the *1/4-tone sharp* accidental (`@accid="1qs"`). Figure 2c shows the *or* operator, which allows matching either of two patterns; it is represented by a vertical line and encoded as `<note type="or">`. Figure 2d shows the *grouping* operator (`<tuplet type="group">`) that groups together any sequence of notes or patterns. A group's `@quantifier` attribute specifies how many occurrences of that group are to be matched, using standard regular expression quantifiers: '+' (one or more), '*' (zero or more), or '?' (zero or one).

Instead of implementing a complex query execution engine manually, we use the web browser's built-in regular expression engine. To facilitate that, the system requires that the score is encoded in the Humdrum **kern representation. When a query is executed, the system translates the query from MEI to an equivalent regular expression that operates on the score's **kern representation. Any `<note>` element in the query is translated to an

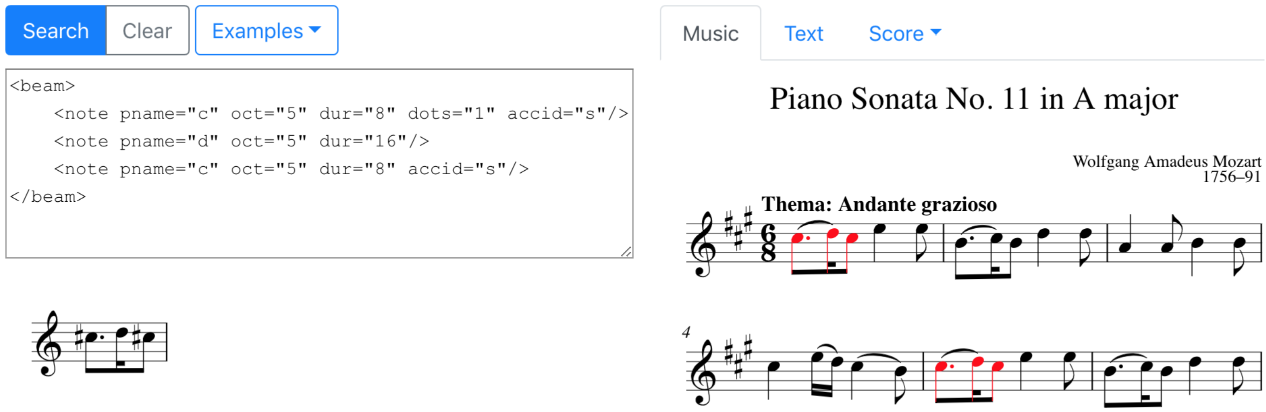

Figure 1: The prototype query system showing an example query. The query is displayed on the left half of the screen: the user inputs the MEI encoding (top) and it is rendered to musical notation (bottom). The score is displayed on the right, with query results highlighted in red.

expression that matches a single Humdrum record with that note's pitch and duration, allowing for the wildcards described above (any pitch, any duration, or any accidental). An *or* operator in the query is translated to the '|' regular expression operator, and a *grouping* operator is translated to a regular expression group and quantifier. The translation is done recursively to support MEI elements that contain other elements such as `<beam>` and `<tuplet>`.

Finally, a complete regular expression is obtained that is equivalent to the MEI query, and it is executed on the Humdrum score using the browser's regular expression engine. When the regular expression matches a portion of the score text, the system highlights the corresponding notes in the score by annotating the matched Humdrum records with a marker ('`@`') and adding a '`@ = marked note`' directive to the file. This causes Verovio to color the corresponding notes in red when rendering the score.

Future Improvements

The query encoding described above is tightly coupled to its graphical representation. For example, a group is encoded with `<tuplet>` even though `<tuplet>` already has another meaning in MEI. We would wish to create an encoding that is more semantically correct, e.g., using `<group>` rather than `<tuplet>` for grouping. Furthermore, users should be able to input queries using a graphical interface rather than typing the textual encoding.

The musicological applications of the prototype system are currently limited because it only supports monophonic music. Extending it to support polyphonic music is challenging; one interesting approach would be to use Structured Polyphonic Patterns (Bergeron and Conklin 2008). Other extensions the system could support include transposition-invariance, rests and chords, and additional data layers such as harmony, lyrics, contour, and phrase boundaries.

Conclusion

Our prototype system demonstrates the merit of a notation-based query language for symbolic music. To our knowledge, no similar work has been done in the past. MEI allows the required flexibility to represent our augmented query notation, and it can be further extended for a more semantic encoding. We hope that this is a first step towards making pattern-based queries accessible to a wider audience of musicologists.

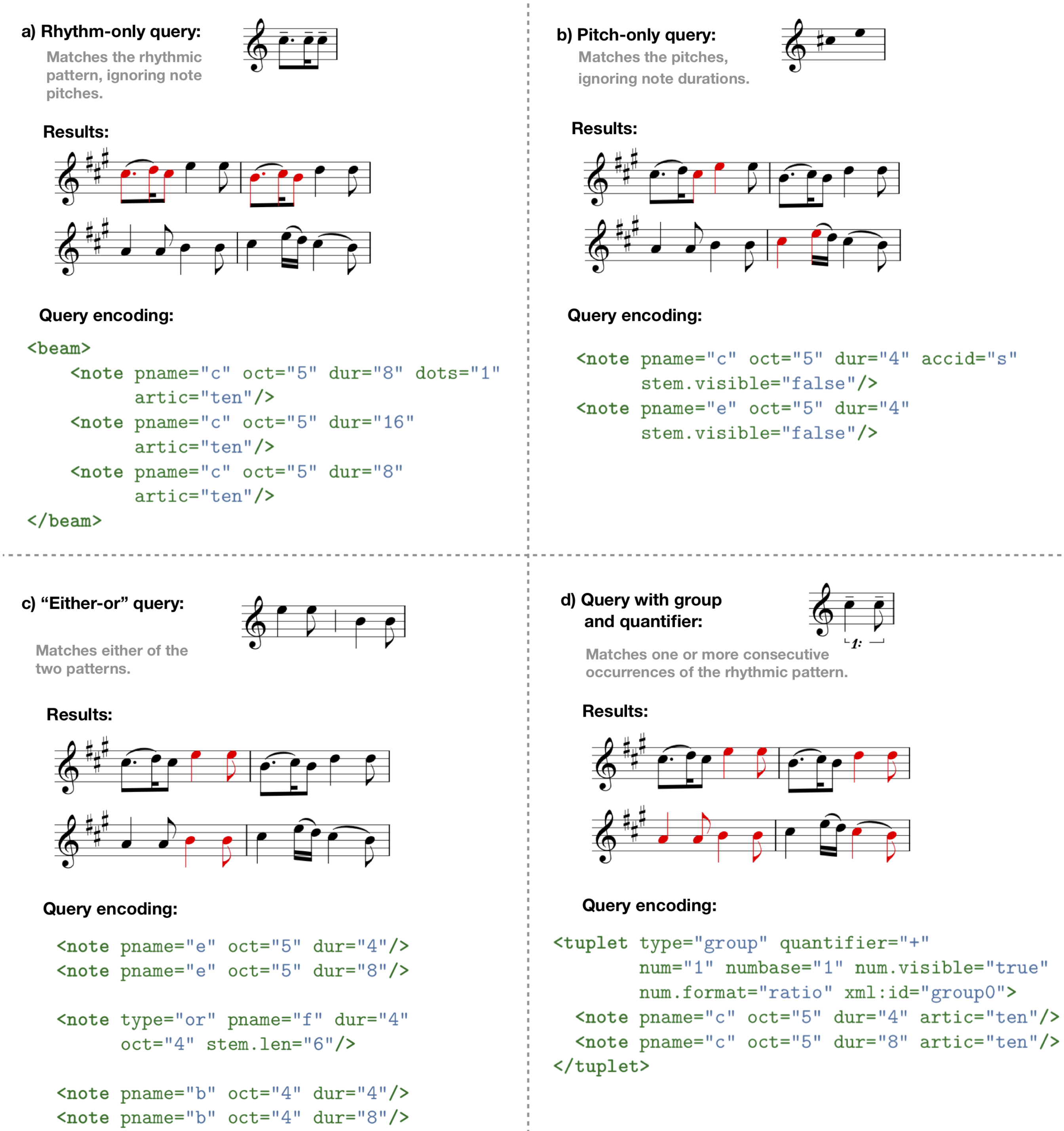

Figure 2: Several example queries and their MEI encoding. The results of each query are highlighted in red in the score excerpt below it. See text for a full explanation of the query notation and encoding.

References

Bergeron, Mathieu, and Darrell Conklin. 2008. "Structured Polyphonic Patterns." In *Proceedings of the International Society for Music Information Retrieval Conference*, 69–74.

Cuthbert, Michael S., and Christopher Ariza. 2010. "Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data." In *Proceedings of the International Society for Music Information Retrieval Conference*, 637–42.

Huron, David. 2002. "Music Information Processing Using the Humdrum Toolkit: Concepts, Examples, and Lessons." *Computer Music Journal* 26 (2): 11–26.

Pugin, Laurent, Rodolfo Zitellini, and Perry Roland. 2014. "Verovio: A Library for Engraving MEI Music Notation into SVG." In *Proceedings of the International Society for Music Information Retrieval Conference*, 107–12.

Roland, Perry. 2002. "The Music Encoding Initiative (MEI)." In *Proceedings of the First International Conference on Musical Applications Using XML*, 55–59.

Thompson, Ken. 1968. "Programming Techniques: Regular Expression Search Algorithm." *Communications of the ACM* 11 (6): 419–22.

Typke, Rainer, Frans Wiering, and Remco C. Veltkamp. 2005. "A Survey of Music Information Retrieval Systems." In *Proceedings of the International Society for Music Information Retrieval Conference*, 153–60.