Formal Languages in Information Extraction and Graph Databases

Wim Martens(✉)

University of Bayreuth, Bayreuth, Germany
wim.martens@uni-bayreuth.de

This abstract covers two areas of data management research in which formal language theory plays a central role, namely in Information Extraction and Graph Databases.

Information Extraction. Automata-based foundations of Information Extraction (e.g., [9,16,34]) have become a popular research topic over the last years. One framework that has been studied in this context is that of *document spanners* [16]. Document spanners model information extraction tasks as functions that map input text documents to a relation of *spans*, i.e., intervals of start and end positions in the text. A particular interesting class of spanners is the class of regular spanners, which is based on regular languages with capture variables. This class satisfies a number of interesting complexity and expressiveness properties and therefore caused a revival of automata- and formal language techniques in database research. Examples of such work are on the enumeration of answers [1,18], expressiveness [20,21,33], complexity issues [22,29,32], integration of weights [15], and distributed evaluation [14]. That said, the document spanners framework is not the only one that is studied in this context, and there are other elegant frameworks that can express information extraction functions beyond the spanner framework, e.g., [9,34].

Graph Databases. Formal languages have played a central role in Graph Databases since the SIGMOD 1987 paper of Cruz et al. [12], which is one of the first and most influential papers on the topic. Indeed, this paper introduced regular expressions for querying paths (later named *regular path queries* or *RPQs*), which are still used in graph query languages today [13,19,24]. This early work on Graph Databases only allowed RPQs to match *simple paths* in graphs, i.e., paths without repeated nodes. However, after discovering that this restriction already makes simple queries difficult to evaluate [30], it was largely abandoned by the research community, and a huge body of research on fundamental problems followed, in which RPQs were allowed to match all paths. This line of work is too extensive to discuss here, but its state until 2013 is nicely surveyed by Barceló [7]. It still produces high-quality and exciting results today (e.g., [8,17]).

W. Martens—Supported by grant MA 4938/4–1 from the Deutsche Forschungsgemeinschaft and grant I-1502-407.6/2019 of the German-Israeli Foundation for Scientific Research and Development.

© Springer Nature Switzerland AG 2020
M. Anselmo et al. (Eds.): CiE 2020, LNCS 12098, pp. 306–309, 2020.
https://doi.org/10.1007/978-3-030-51466-2_28

Perhaps ironically, the *simple paths* and the similar *trail* restriction (which only allows paths without repeated edges) resurfaced on the systems side of graph databases. Indeed, an early incarnation[1] of SPARQL 1.1 [24] used (a variant of) the simple path restriction, whereas the default semantics of Cypher [13] uses the trail restriction. This new development on the practical side of graph query languages motivated several research groups to build a scientific basis that can be used to guide the design of RPQs in graph query languages [5,6,26,27]. Furthermore, it seems that, in order to understand RPQ evaluation in practical graph query languages, it is very useful to combine fundamental research with query log analysis [10,11,28].

To conclude, it seems that the research communities' efforts to connect theory and practice (which go far beyond what I've been able to mention here, see, e.g. [2–4,25,31,35]) are paying off, in the sense that we are now experiencing an increased interaction between researchers and practitioners in the *Graph Query Language (GQL) Standard* initiative [23] and the process that has led to it.[2] The GQL initiative was recently inaugurated as an official ISO project that aims at becoming an international standard for graph database querying. Furthermore, the story does not stop here at all—a large number of initiatives is currently brainstorming on next-generation logical foundations of graph databases and their query languages, schema languages for graphs, etc.

References

1. Amarilli, A., Bourhis, P., Mengel, S., Niewerth, M.: Constant-delay enumeration for nondeterministic document spanners. In: International Conference on Database Theory (ICDT), pp. 22:1–22:19 (2019)
2. Angles, R., et al.: G-CORE: a core for future graph query languages. In: International Conference on Management of Data (SIGMOD), pp. 1421–1432 (2018)
3. Angles, R., Arenas, M., Barceló, P., Hogan, A., Reutter, J.L., Vrgoc, D.: Foundations of modern query languages for graph databases. ACM Comput. Surv. **50**(5), 68:1–68:40 (2017)
4. Angles, R., et al.: The linked data benchmark council: a graph and RDF industry benchmarking effort. SIGMOD Rec. **43**(1), 27–31 (2014)
5. Arenas, M., Conca, S., Pérez, J.: Counting beyond a yottabyte, or how SPARQL 1.1 property paths will prevent adoption of the standard. In: Proceedings of the World Wide Web Conference (WWW), pp. 629–638 (2012)
6. Bagan, G., Bonifati, A., Groz, B.: A trichotomy for regular simple path queries on graphs. In: ACM Symposium on Principles of Database Systems (PODS), pp. 261–272 (2013)
7. Barceló, P.: Querying graph databases. In: ACM Symposium on Principles of Database Systems (PODS), pp. 175–188 (2013)
8. Barceló, P., Figueira, D., Romero, M.: Boundedness of conjunctive regular path queries. In: International Colloquium on Automata, Languages, and Programming (ICALP), pp. 104:1–104:15 (2019)

[1] https://www.w3.org/TR/2012/WD-sparql11-query-20120105, see the definition of *ZeroOrMorePath*.

[2] https://www.gqlstandards.org/existing-languages has an influence diagram.

9. Beedkar, K., Gemulla, R., Martens, W.: A unified framework for frequent sequence mining with subsequence constraints. ACM Trans. Database Syst. **44**(3), 11:1–11:42 (2019)
10. Bonifati, A., Martens, W., Timm, T.: Navigating the maze of Wikidata query logs. In: The World Wide Web Conference (WWW), pp. 127–138 (2019)
11. Bonifati, A., Martens, W., Timm, T.: An analytical study of large SPARQL query logs. VLDB J. (2020, to appear)
12. Cruz, I.F., Mendelzon, A.O., Wood, P.T.: A graphical query language supporting recursion. In: ACM Special Interest Group on Management of Data (SIGMOD), pp. 323–330 (1987)
13. Cypher Query Language. https://neo4j.com/cypher-graph-query-language/
14. Doleschal, J., Kimelfeld, B., Martens, W., Nahshon, Y., Neven, F.: Split-correctness in information extraction. In: ACM Symposium on Principles of Database Systems (PODS), pp. 149–163 (2019)
15. Doleschal, J., Kimelfeld, B., Martens, W., Peterfreund, L.: Weight annotation in information extraction. In: International Conference on Database Theory (ICDT) (2020, to appear)
16. Fagin, R., Kimelfeld, B., Reiss, F., Vansummeren, S.: Document spanners: a formal approach to information extraction. J. ACM **62**(2), 12:1–12:51 (2015)
17. Figueira, D.: Containment of UC2RPQ: the hard and easy cases. In: International Conference on Database Theory (ICDT) (2020, to appear)
18. Florenzano, F., Riveros, C., Ugarte, M., Vansummeren, S., Vrgoc, D.: Constant delay algorithms for regular document spanners. In: ACM Symposium on Principles of Database Systems (PODS), pp. 165–177 (2018)
19. Francis, N., et al.: Cypher: an evolving query language for property graphs. In: International Conference on Management of Data (SIGMOD), pp. 1433–1445 (2018)
20. Freydenberger, D.D.: A logic for document spanners. In: International Conference on Database Theory (ICDT), pp. 13:1–13:18 (2017)
21. Freydenberger, D.D., Holldack, M.: Document spanners: from expressive power to decision problems. In: International Conference on Database Theory (ICDT), pp. 17:1–17:17 (2016)
22. Freydenberger, D.D., Kimelfeld, B., Peterfreund, L.: Joining extractions of regular expressions. In: ACM Symposium on Principles of Database Systems (PODS), pp. 137–149 (2018)
23. GQL Standard. https://www.gqlstandards.org/
24. Harris, S., Seaborne, A.: SPARQL 1.1 Query Language (2013). https://www.w3.org/TR/sparql11-query
25. Libkin, L., Martens, W., Vrgoc, D.: Querying graphs with data. J. ACM **63**(2), 14:1–14:53 (2016)
26. Losemann, K., Martens, W.: The complexity of regular expressions and property paths in SPARQL. ACM Trans. Database Syst. **38**(4), 24:1–24:39 (2013)
27. Martens, W., Niewerth, M., Trautner, T.: A trichotomy for regular trail queries. In: Annual Symposium on Theoretical Aspects of Computer Science (STACS) (2020, to appear)
28. Martens, W., Trautner, T.: Dichotomies for evaluating simple regular path queries. ACM Trans. Database Syst. **44**(4), 16:1–16:46 (2019). Article 16
29. Maturana, F., Riveros, C., Vrgoc, D.: Document spanners for extracting incomplete information: expressiveness and complexity. In: ACM Symposium on Principles of Database Systems (PODS), pp. 125–136 (2018)

30. Mendelzon, A.O., Wood, P.T.: Finding regular simple paths in graph databases. SIAM J. Comput. **24**(6), 1235–1258 (1995)
31. Pérez, J., Arenas, M., Gutiérrez, C.: Semantics and complexity of SPARQL. ACM Trans. Database Syst. **34**(3), 16:1–16:45 (2009)
32. Peterfreund, L., Freydenberger, D.D., Kimelfeld, B., Kröll, M.: Complexity bounds for relational algebra over document spanners. In: ACM Symposium on Principles of Database Systems (PODS), pp. 320–334 (2019)
33. Peterfreund, L., ten Cate, B., Fagin, R., Kimelfeld, B.: Recursive programs for document spanners. In: International Conference on Database Theory (ICDT), pp. 13:1–13:18 (2019)
34. Renz-Wieland, A., Bertsch, M., Gemulla, R.: Scalable frequent sequence mining with flexible subsequence constraints. In: IEEE International Conference on Data Engineering (ICDE), pp. 1490–1501 (2019)
35. Reutter, J.L., Romero, M., Vardi, M.Y.: Regular queries on graph databases. In: International Conference on Database Theory (ICDT), pp. 177–194 (2015)