A Circuit-Based Approach to Efficient Enumeration

Antoine Amarilli

April 21, 2017

Many problems in data management require us to evaluate a query on some data, and compute a set of answers. For instance, we may write a conjunctive query in SQL on a relational data base, or an XPath query on data trees. On large data sets, however, the complete set of answers can be huge, and impossible to compute completely. This makes it necessary to compute quickly an initial set of solutions, and to compute more solutions as they are requested by the user, analogously to how search engine results are presented in a paginated fashion.

The theoretical computer science community has been studiying *enumeration algorithms* to give a formal foundation to such tasks. An enumeration algorithm for a query on a dataset consists of two phases: a *preprocessing phase*, where the input data is read and indexed; and a *computation phase*, where the results are computed and output one after the other. The efficiency of the algorithm is measured by giving the complexity of the preprocessing phase, and the *delay*, i.e., the time spent in the computation phase to compute each successive answer. The most stringent requirements on enumeration algorithms impose *linear-time preprocessing*, i.e., we index the data set in linear time, and *constant-delay*, i.e., we produce each solution after a constant amount of time. When the size of the solutions can be large, *linear-delay* requires that the delay spent to produce each solution is linear in the size of that solution (but independent from the size of the data set).

This talk will present my recent work with Pierre Bourhis, Louis Jachiet, and Stefan Mengel, which is currently under review [1]. In this work, we study constant-delay enumeration algorithms to evaluate queries on data trees, and relational instances of a restricted kind, i.e., bounded-treewidth instances. We show this result by leveraging the tools of knowledge compilation, and factorized representations, to compute efficiently a circuit representation of the answers of the query to be enumerated, using a variant of our earlier work [2, 3]. We then show how the valuation of such circuits, corresponding to the answers, can be enumerated efficiently, thanks to structural restrictions on the circuits [5]. This allows us to re-prove the well-known result [4, 6] that the answers to monadic second-order queries with free first-order variables can be enumerated with linear-preprocessing and constant-delay over structures of bounded treewidth. We also derive new enumeration results for circuit enumeration and factorized representations.

The talk may also sketch some of our ongoing work using the same techniques to show efficient enumeration results for queries when the underlying data structures can be updated, in the spirit of works such as [7].

References

[1] Antoine Amarilli, Pierre Bourhis, Louis Jachiet, and Stefan Mengel. A Circuit-Based Approach to Efficient Enumeration. Under review, 2017.

[2] Antoine Amarilli, Pierre Bourhis, and Pierre Senellart. Provenance circuits for trees and treelike instances. In *ICALP*, 2015.

[3] Antoine Amarilli, Pierre Bourhis, and Pierre Senellart. Tractable lineages on treelike instances: Limits and extensions. In *PODS*, 2016.

[4] Guillaume Bagan. MSO queries on tree decomposable structures are computable with linear delay. In *CSL*, 2006.

[5] Adnan Darwiche. On the tractable counting of theory models and its application to truth maintenance and belief revision. *J. Applied Non-Classical Logics*, 11(1-2), 2001.

[6] Wojciech Kazana and Luc Segoufin. Enumeration of monadic second-order queries on trees. *TOCL*, 14(4), 2013.

[7] Katja Losemann and Wim Martens. MSO queries on trees: enumerating answers under updates. In *CSL-LICS*, 2014.