Technical Perspective: Efficient Logspace Classes for Enumeration, Counting, and Uniform Generation

Reinhard Pichler
TU Wien, Austria

Traditionally, by *query answering* we mean the problem of finding *all* answers to a given query over a given database. But what happens if the number of answers is prohibitively big – which may easily occur in a Big Data context? In such situations, it seems preferable to have a mechanism that produces one answer after the other with certain guarantees on the time between any two outputs and to let the user decide when to stop. This leads us to the *enumeration problem*, which has received a lot of interest recently [1]. However, in order for the user to get a "realistic" picture of the entirety of answers, two crucial questions arise: first, how big is the portion of output answers compared with the total number of answers? And second, do the output answers reflect the variety of the complete set of answers? The first question refers to the *counting* problem, where we are interested in the total number of answers. The second question leads us to the problem of *uniform generation*, where we request that the answers be uniformly generated and thus form an unbiased sample of the complete set of answers.

Decades of research have been devoted to the analysis of the *complexity* of query answering of all kinds of query languages and, in particular, to the identification of scenarios (by imposing restrictions on the queries and/or databases) which allow for *efficient* query answering. Such complexity analyses usually consider a suitable decision problem such as checking if a given candidate is indeed an answer to a given query over the given database. As we shift our focus to the triad of enumeration, counting, and uniform generation, the notions of *complexity* and of *efficiency* have to be reconsidered. Moreover, in order to get an understanding of the complexity of a concrete query language for a concrete type of databases, it seems unavoidable to analyse the three problems separately ...

... unless we have a general (and, ideally, simple) framework that allows us to detect at once if all three problems are efficiently solvable. Establishing such a framework is the very goal of the paper *Efficient Logspace Classes for Enumeration, Counting, and Uniform Generation* by Marcelo Arenas, Luis Alberto Croquevielle, Rajesh Jayaram, and Cristian Riveros. The resulting framework is very general and by no means restricted to query answering. It follows the classical formalism of [2], which represents a *problem* as a binary relation R consisting of pairs (x, y), where x is (an encoding of) the *input* and y is (an encoding of) a *solution*. In our case of query answering, the input consists of a database and a query and each answer corresponds to a solution. The authors define classes of such binary relations by means of Turing machines. As a good compromise between expressive power (i.e., capturing an interesting class of relations) and efficiency (i.e., making sure that enumeration, counting, and uniform generation all have acceptable complexity), only non-deterministic log-space Turing machines M are considered, i.e., when run on some input x, machine M produces as output in its accepting computations precisely the solutions y with $(x, y) \in R$. The resulting class of relations is referred to as RelationNL. This compromise between expressive power and efficiency can be shifted a bit more towards efficiency (at the expense of less expressive power) by requiring the Turing machine to be *unambiguous*, i.e., any two different runs of M on input x ending in an accepting state produce distinct outputs. The resulting class of relations is referred to as RelationUL.

The main technical results of the paper are upper bounds on the complexity of enumeration, counting, and uniform generation if a relation is in RelationNL or in RelationUL, respectively. For instance, if R is in RelationUL, then the enumeration problem can be solved with *constant delay* (i.e., after a polynomial-time preprocessing phase, the time between any two successive outputs is independent of the size of the input), the counting problem is polynomial-time solvable and, finally, uniform generation is feasible by a polynomial-time randomized algorithm. Slightly less favorable upper bounds hold for relations in RelationNL. The paper also shows that several interesting problems from various domains (including information extraction and graph databases) indeed fall in one of these two relation-classes.

The paper represents an important first step that opens up many research opportunities such as proving the membership of various problems in RelationNL or RelationUL, searching for further optimizations of the enumeration, counting and uniform generation algorithms for problems in these classes, and also searching for interesting extensions and restrictions of these classes.

1. REFERENCES

[1] E. Boros, B. Kimelfeld, R. Pichler, and N. Schweikardt. Enumeration in data management (dagstuhl seminar 19211). *Dagstuhl Reports*, 9(5):89–109, 2019.

[2] M. Jerrum, L. G. Valiant, and V. V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comput. Sci.*, 43:169–188, 1986.