Technical Perspective for: MATLANG: Matrix Operations and Their Expressive Power

Dan Suciu
University of Washington

The main processing paradigm in data management is bulk processing. As introduced by Codd in the early 70's, under this paradigm relations are processed in bulk, one operator at a time. When applied to relations, this paradigm leads to relational algebra, and its variants, relational calculus, and SQL. Over the years, data management was faced with the challenge of extending bulk processing operators to new kinds of data, and/or new kinds of queries: nested relations, semistructured data, recursive queries. Each such extension requires significant systems development, which should be accompanied, in fact preceded, by a careful study of the expressive power of the new language. Is it as expressive, more expressive, or less expressive than relational algebra? The answer to this question has profound implications on the ability of data processing engines to optimize, compute, distribute, reuse queries in that language. For example, extending relational algebra with nested relations does not increase its expressive power, while extending it with fixpoint does, explaining why modern query engines have an easier time supporting JSON than recursion.

Today, data management is faced with a new task: incorporate into the relational engine linear algebra operations required by machine learning algorithms. A superficial inspection suggests that this is easy, because operations such as matrix multiplication, transpose, addition, are already expressible in SQL. In fact, in SciDB users indeed express these operations in SQL, and the engine can already apply the same optimizations as for general SQL queries. But this answer is unsatisfactory. Most popular ML or linear algebra systems today, such as NumPy, ScaLAPAK, SystemML, support *only* linear algebra operations, and, because of that, they outperform relational engines for the tasks they are designed to do (see reference [39] in the paper). This raises a natural and important question: what is the expressive power of linear algebra, and how does it relate to that of the relational algebra?

The paper by Brijder, Geerts, Van den Bussche and Weerwag, first published in ICDT'2018, initiates the study into precisely this question. This is undoubtedly only the first paper in this line, meant mostly to raise the question; more are likely to follow, for example see references [15] and [8], and the important results there.

The first hurdle the paper faces is that there is no standard linear algebra language. The authors propose one such language, MATLANG, which includes the usual operations on matrices, and not much else. The second hurdle is that expressions in linear algebra and in relational algebra have different types, hence are not directly comparable. To circumvent that, the authors focus on the graph properties that these languages can express, since graphs can be represented both as relations and as matrices. With these assumptions, the paper establishes a few results comparing the two languages. For example, properties in linear algebra can be expressed in FO^3 (which can be thought of as the restriction of relational algebra to intermediate results of arity at most 3), however, if one adds a matrix inverse operation to linear algebra then one can express graph connectivity, suggesting that this language is closer to the extension of relational algebra with fixpoints. The reader will enjoy the arguments used to prove these statements, and will definitely be enticed to follow future progress on the important study of the expressive power of linear algebra.