Trading determinism for speed: An algebra for large graph analytics using nondeterminism

Ashkey Elizabeth Bince and Robert Brijder

Department of Mathematics and Computer Science,
Eindhoven University of Technology, Netherlands
{a.ashkey.elizabeth.bince, r.brijder}@tue.nl

Data scientists often rely on operations from linear algebra to analyze large graphs. However, current database systems are not designed to support such operations natively. There is significant recent work on providing data management systems with linear algebra support [4, 5]. In order to provide scalable and principled data management systems that natively support linear algebra, a theory is needed that does the heavy lifting under the hood, akin to what relational algebra does for SQL in relational databases. The formal language for-MATLANG [3] (which is an extension of a basic formal language called MATLANG [2]) has been proposed as a candidate for such a theory. The expressive power of for-MATLANG is significant, mainly because the order of the rows and columns of matrices is exposed. For graph analytics, however, this order is irrelevant since queries should not depend on the ordering of the vertices used in constructing the adjacency matrix.

In this ongoing work, we propose a less expressive counterpart of for-MATLANG in which ordering is not available, i.e., the language is by design permutation-invariant with respect to the matrices. For this we carry over the non-deterministic witness operator in first-order logic [1] to the domain of linear algebra. We show that the obtained language, called wfor-MATLANG ("w" for witness, or weak), is able to express common graph analytics algorithms like the Floyd-Warshall algorithm and testing for connectedness, bipartiteness, and four-cliques.

As noted in [1], expressing deterministic queries using nondeterministic language constructs potentially provides opportunities for query optimization by leveraging the freedom in the computation. We show that, indeed, wfor-MATLANG, due to its nondeterministic witness operator, has query optimization opportunities that are not available in for-MATLANG. In this way, we trade a loss in expressivity for a gain in query execution speed.

References

[1] S. Abiteboul and V. Vianu. "Non-determinism in logic-based languages". In: *Annals of Mathematics and Artificial Intelligence* 3.2 (1991), pp. 151–186. DOI: 10.1007/BF01530924.

[2] R. Brijder et al. "MATLANG: Matrix operations and their expressive power". In: *ACM SIGMOD Record* 48.1 (2019), pp. 60–67. DOI: 10.1145/3371316.3371331.

[3] F. Geerts et al. "Expressive power of linear algebra query languages". In: *Proceedings of the 40th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 2021, pp. 342–354. DOI: 10.1145/3452021.3458314.

[4] D. Hutchison, B. Howe, and D. Suciu. "LaraDB: A minimalist kernel for linear and relational algebra computation". In: *Proceedings 4th ACM SIGMOD Workshop on Algorithms and Systems for MapReduce and Beyond*. Ed. by F. Afrati and J. Sroka. 2017, 2:1–2:10. DOI: 10.1145/3070607.3070608.

[5] S. Luo et al. "Scalable linear algebra on a relational database system". In: *Communications of the ACM* 63.8 (2020), pp. 93–101. DOI: 10.1145/3405470.