Querying graph databases with RPQs

Abstract

Graph database management systems have increased in popularity over the last decades. In database theory, we abstract such databases as labelled graphs. Most real query languages are based on the well-known formalism of regular path queries (RPQs). Such a query is defined from a regular expression R. Any walk in the graph labelled with a word conforming to R is called a match, and in general there are infinitely many matches. The main challenge is to efficiently compute a finite and meaningful output from the matches.

Several approaches are used in practice and theory to reach this goal. Homomorphism semantics is the most studied and enjoy nice theoretical properties, but is not suitable for some practical applications (too little information is kept in the output). On the other side of the spectrum, the most widespread semantics in practice is called trail semantics and seems unreasonable from a theoretical standpoint (high complexity, arbitrary restrictions).

In a recent work, we suggested a new approach, run-based semantics, which seems a reasonable compromise. It restricts the infinitely many matches to a finite number by stopping when a cycle occurs in the computation of the query and in the graph simultaneously. The internship is about further investigating run-based semantics, and more generally about exploring the properties and connections between the semantics of RPQs.

1 Practical details

- Advisor: Victor Marsault
- Co-advisors: Claire David and Nadime Francis
- Laboratory: LIGM (Laboratoire d'Information Gaspard Monge)
- Team: BAAM (Bases de données, Automates, Analyse d'algorithmes et Modèles)
- Location: Université Gustave-Eiffel (30 minutes from Paris, RER A *Noisy-Champs*)

2 Example of graph databases, RPQs and matches

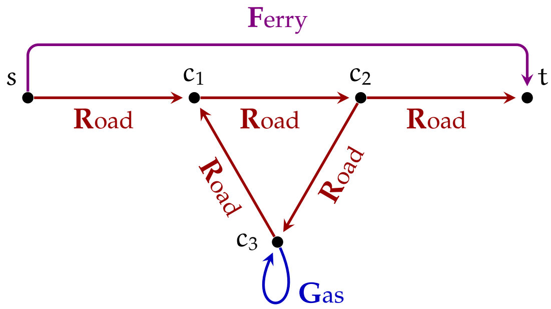

Figure 1: A graph database D

$$Q_1 = (\mathbf{R} + \mathbf{F})^*$$

$$Q_2 = (\mathbf{R} + \mathbf{F})^* \, \mathbf{G} \, (\mathbf{R} + \mathbf{F})^*$$

Figure 2: Q_1, a simple reachability RPQ, and Q_2, reachability with a mandatory stop

Here are two walks and whether they are matches to Q_1, Q_2 or both.

$s \to c_1 \to c_2 \to t$	match to Q_1 but not Q_2
$s \to c_1 \to c_2 \to c_3 \to c_3 \to c_1 \to c_2 \to t$	match to Q_1 and Q_2

3 Description

Graph DBMS (database management systems) have increased in popularity over the last decades. In database theory, we abstract such databases as labelled graphs, like in figure 1. Most real query languages are based on the formalism of RPQs (regular path queries): an RPQ is defined by a regular expression R and is traditionally evaluated under *homomorphism semantics* [Ang+17]. It returns all pairs of vertices that are linked by a walk whose label conforms to R. Figure 2 gives an example of two queries Q_1 and Q_2 that both return the pair (s, t) (among others) : the walk $s \to t$ is labelled by **F** which conforms to Q_1 and the walk $s \to c_1 \to c_2 \to c_3 \to c_3 \to c_1 \to c_2 \to t$ is labelled by **RRRGRRR**, which conforms to Q_2.

Homomorphism semantics enjoy nice theoretical properties but do not meet the needs of real-life graph DBMS. Indeed, one might need the number of matching walks (Tuple Multiplicity[1]), or even the walks themselves (Walk Enumeration). These problems are meaningless under homorphisms semantics since there might be infinitely many matching walks (when the graph contains cycles). Other semantics exist to address this issue. *Trail semantics* restrict the output to walks with no repeated edge. This restriction has a dramatic impact on complexity, as it makes most problems at least NP-hard. Moreover, interesting walks are sometimes discarded: Q_2 returns no walk from s to t under trail semantics. *Shortest-walk semantics* keeps only walks with a minimal number of edges. Most computational problems are PTIME (or equivalent) but some problems are meaningless (Tuple Multiplicity[1], arguably Walk Enumeration) issues arise from the fact that the metrics is arbitrary: Q_1 returns the ferry route $s \to t$ over $s \to c_1 \to c_2 \to t$ but it is arguably less relevant.

Recently, we proposed [DFM22] a new semantics that seems to be a good compromise. Similarly to trail semantics, it discards cyclic results, but only if a cycle in the walk coincides with a cycle in the computation of the query. Some important computational problems are PTime (Tuple Membership[1], Walk Enumeration), others remain NP-hard (Tuple Multiplicity, Walk Membership).

The internship is about further exploring the different semantics of RPQs, in particular the semantics based on run. Here are a few examples of research direction.

The complexity of the problem Deduplicated Walk Membership[1]is still open for run-based semantics. Note that this problem is also open for the more classical shortest-walk semantics [Vrg22]. It seems likely that the two problems are related.

In [DFM22], one of the semantics is based on an automaton A equivalent to the input RPQ R. However, the output depends on the automaton A, and not only on the language accepted by A. Hence, the expression-to-automaton algorithm used to obtain A from R matters, and the impact of this choice on the semantics remains to be explored.

Another goal is to adapt run-based semantics so that it can be used in practice, for instance in GPML [Deu+22], the pattern matching part of GQL and SQL/PGQ, two standards in development by ISO [GQL; PGQ]; GQL is a new language for property graphs and SQL/PGQ is an extension of SQL. A first step could be to consider GPC [Fra+23], a theoretical abstraction of GPML.

Other semantics exist, or could be designed, and require further research. For instance, *cheapest-walk semantics* could act similarly to shortest-walk semantics, but leave the choice of the metrics to the user. This idea has been mentioned during the design process of GPML but was discarded due to the lack of supporting material.

[1] Computational problems are briefly described in Section 4.

4 Short description of the computational problems

Note that all problems below are parameterised by the some semantics S.

Tuple Membership – Given a graph database, an RPQ R and two vertices s, t, is there a match to R in D that starts in s and ends in t?

Tuple Multiplicity – Given a graph database D, an RPQ R and two vertices s, t, how many matches to R in D start in s and end in t?

Walk Membership – Given a graph database D, an RPQ R and a walk w in D. Is w a match to R?

Walk Enumeration – Given a graph database D and an RPQ R, enumerate the **multiset**[2] of walks in D whose label conforms to R.

Deduplicated Walk Enumeration – Given a graph database D and an RPQ R, enumerate the **set** of walks in D whose label conforms to R.

5 Bibliography

[Ang+17] Renzo Angles, Marcelo Arenas, Pablo Barceló, Aidan Hogan, Juan L. Reutter, and Domagoj Vrgoc. "Foundations of Modern Query Languages for Graph Databases". In: *ACM Comput. Surv.* 50.5 (2017).

[DFM22] Claire David, Nadime Francis, and Victor Marsault. *Run-Based Semantics for RPQs*. ArXiv preprint. 2022. URL: https://arxiv.org/abs/2211.13313.

[Deu+22] Alin Deutsch, Nadime Francis, Alastair Green, Keith Hare, Bei Li, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Wim Martens, Jan Michels, Filip Murlak, Stefan Plantikow, Petra Selmer, Hannes Voigt, Oskar van Rest, Domagoj Vrgoč, Mingxi Wu, and Fred Zemke. "Graph Pattern Matching in GQL and SQL/PGQ". In: *SIGMOD'22*. 2022. URL: https://arxiv.org/abs/2112.06217.

[Fra+23] Nadime Francis, Amélie Gheerbrant, Paolo Guagliardo, Leonid Libkin, Victor Marsault, Wim Martens, Filip Murlak, Liat Peterfreund, Alexandra Rogova, and Domagoj Vrgoč. "GPC: A Pattern Calculus for Property Graphs". In: *PODS'23*. To appear. 2023. URL: https://arxiv.org/abs/2210.16580.

[GQL] International Organization for Standardization. *GQL*. Standard under development ISO/IEC CD 39075. 2023 (expected). URL: https://www.iso.org/standard/76120.html.

[PGQ] International Organization for Standardization. *SQL — Part 16: SQL Property Graph Queries (SQL/PGQ)*. Standard under development ISO/IEC CD 9075-16.2. 2023 (expected). URL: https://www.iso.org/standard/79473.html.

[Vrg22] Domagoj Vrgoč. *Evaluating regular path queries under the all-shortest paths semantics*. Arxiv preprint. 2022. URL: https://arxiv.org/abs/2204.11137.

[2] In the data model we use, each edges in the graph can actually bear several labels. Hence the same walk might be a match to R multiple times.