Research internship topics about

Querying graphs with regular expressions

2025

1 Practical details

- Advisor: Victor Marsault, victor.marsault@univ-eiffel.fr
- Possible co-advisors:Claire David, Nadime Francis, Antoine Meyer
- Laboratory: LIGM (Laboratoire d'Information Gaspard Monge)
- Team: BAAM (Bases de données, Automates, Analyse d'algorithmes et Modèles)
- Location: Université Gustave-Eiffel in Copernic building (RER A *Noisy-Champs*).
- Level: M1 or M2, with possibility to continue in PhD.

2 Context

Graph databases have emerged in the industry in the mid 2010's with the vendor Neo4j, who introduced the property graph data model and the query language Cypher. ISO published in 2024 a new query standard, GQL, that incorporates a formalization of the property graph data model. Like most real query language for graph databases, Cypher and GQL are based on pattern matching for walks (called Regular Path Queries in database theory, RPQs).

This internship is in the area of database theory, and we abstract graph databases as **labelled graphs** (see Fig. 1) and queries as **regular expressions** (see Fig. 2). A *match* to a regular expression Q is any walk (that is, any sequence of edges) labelled by a word that conforms to Q. For instance, $w_1 = s \to c_1 \to c_2 \to t$ is labelled by **RRR**, hence is a match to Q_1 but not to Q_2, and $w_2 = s \to c_1 \to c_2 \to c_3 \to c_3 \to c_1 \to c_2 \to t$ is a match to both Q_1 and Q_2.

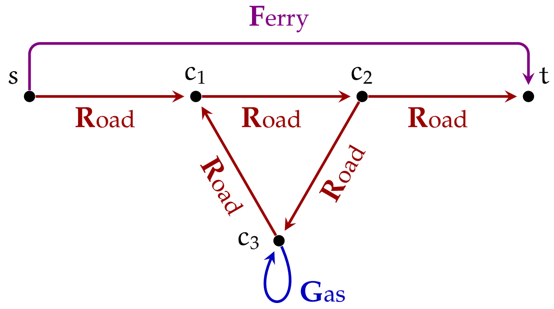

Figure 1: A graph database D

$$Q_1 = (\mathbf{Road} + \mathbf{Ferry})^*$$

$$Q_2 = (\mathbf{Road} + \mathbf{Ferry})^* \, \mathbf{Gas} \, (\mathbf{Road} + \mathbf{Ferry})^*$$

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

Notice that there are infinitely many matches to Q_1 and to Q_2 due to the triangle cycle $c_1 \to c_2 \to c_3 \to c_1$. In order to return a finite set of matches, GQL uses one of several possible *RPQ semantics*, that is several possible ways to filter matches. *Homomorphism semantics* returns only the endpoints of matches: the pair (s, t) is returned by both Q_1 and Q_2 because they both match w_2. *Trail semantics* returns only the matches in which no edge repeats: Q_1 returns $\{s \to t, s \to c_1 \to c_2 \to t\}$, while Q_2 returns nothing. *Shortest semantics* returns only the shortest match (in number of edges) between given endpoints; for walks from s to t, Q_1 returns $\{s \to t\}$ and Q_2 returns $\{s \to c_1 \to c_2 \to c_3 \to c_3 \to c_1 \to c_2 \to t\}$.

3 Possible internship topic

Study of new RPQ semantics In the past few years, we introduced new alternative RPQ semantics to supplement the ones from GQL (trail, acyclic, shortest). The internship could be about continuing the study of simple-run semantics or binding-trail semantics, that we started in [DFM23; Sai24], or the study of subwalk-minimal semantics or shortest-coverage started in [Khi24]. Another internship topic is the development of the framework we started in [MM25] to compare existing RPQ semantics on the basis of criteria other than complexity.

Distinct shortest walk enumeration The data model of GQL allows a single edge to bear multiple labels. This means that a given walk in the graph can be a matched in multiple different ways, while it must be output only once. We proposed in [DFM24] an efficient algorithm to avoid outputting duplicates. The goal of the internship would be to improve this algorithm. Indeed, it requires a fixed initial cost to build the data structure that handles nondeterminism, while it is not needed if no walk is actually matched multiple times. We want the maintenance cost of the data structure to fit the level of nondeterminism actually encountered.

Complexity of GQL features GQL augments RPQs with many features, and new constructs are under consideration to be added in version 2. The impact of existing features on evaluation complexity is not always known, and the one of new constructs. The goal of the internship would be to study these impacts and, if possible to suggests changes in their design to ensure they are well-behaved.

4 Prior knowledge

The student is expected to have a taste for formal methods (complexity, automata theory) and to have some basic knowledge of them. The intern will learn more specialized knowledge (such as enumeration complexity, parametrized complexity, database theory) during the internship.

5 Bibliography

[DFM23] Claire David, Nadime Francis, and Victor Marsault. "Run-Based Semantics for RPQs". In: *KR'23*. 2023. URL: `https://arxiv.org/abs/2211.13313`.

[DFM24] Claire David, Nadime Francis, and Victor Marsault. "Distinct Shortest Walk Enumeration for RPQs". In: *PODS*. Vol. 2. 2. ACM, 2024. URL: `https://doi.org/10.1145/3651601`.

[Khi24] Sara Khichane. "Study of New Semantics for Regular Path Queries". Supervised by Amélie Gheerbrant, Victor Marsault and Antoine Meyer. M1 thesis. Université Paris-Cité, 2024.

[MM25] Victor Marsault and Antoine Meyer. *Designing and Comapring RPQ semantics*. To appear; preprint available at: `https://victor.marsault.xyz/resources/articles/RPQSemanticsFramework_v3.pdf`. 2025.

[Sai24] Steven Sailly. "Compatibilité de la sémantique binding-trail avec GQL". French. Accessible from the website of the author `https://steven.saill.yt`. Supervised by Nadime Francis and Victor Marsault. M2 thesis. Université Paris-Cité, 2024.