M2 Internship Proposal: Enumerating simple paths satisfying a regular path query

Antoine Amarilli, Mikaël Monet

An important recent topic in database theory research is how to efficiently evaluate recursive queries over *graph databases*. At a theoretical level, this question is posed in terms of *regular path queries*. Letting Σ be an alphabet, a *graph database* over Σ is simply a directed graph with edges labeled by Σ, formally it is an ordered pair $G = (V, E)$ with $E \subseteq V \times \Sigma \times V$. A *regular path query* is a regular language over Σ, defined for instance by a finite deterministic automaton A. We wish to efficiently find directed paths in G that form words over Σ^* accepted by the automaton A. The problem can be studied in several different semantics, but a common one is that of *simple paths*, i.e., paths where no vertices are repeated: we wish to produce all *simple paths* in G that form a word accepted by A.

The question has already been studied in several settings. For instance, there is a known trichotomy [BBG20] on the complexity of identifying, given a graph database G and two endpoints s and t, whether there is a simple path from s to t forming a word accepted by A: the problem is sometimes in PTIME and sometimes NP-hard depending on the fixed automaton A. The problem has also been studied in terms of *enumerating* the successive simple paths satisfying A [MT18]. However, all these works are about enumerating results *with specified endpoints* s and t.

To our knowledge, there is no known complexity classification of the problem in the setting where we want to enumerate all simple paths, regardless of the endpoints. Formally, the task is the following: we fix an alphabet Σ and automaton A over Σ, we are given a graph database G, and we want to produce the sequence of all simple paths of G forming a word accepted by A, in an arbitrary order. We expect that the task is still NP-hard for some languages, e.g., $a(bb)^*c$; and obviously it is in PTIME for others, e.g., finite languages, or the language a^* via Yen's algorithm (see [MT18]). More interestingly, we expect the task to be in PTIME for some languages, e.g., $(aa)^*$, for which deciding the existence of a path with prescribed endpoints is NP-hard [BBG20].

The goal of the internship is to study this problem, towards the goal of obtaining a complexity classification: for which regular languages is the problem tractable, and for which regular languages is it NP-hard? The question can also be explored around the following variants:

- How does the tractability boundary change when we impose constraints on the enumeration order, e.g., enumerating path by increasing order of size, or by lexicographic order on the endpoints?
- How about other semantics, such as enumerating *trails* (paths where vertices may be repeated but where no edge is repeated) or *walks* (paths where vertices and edges may be repeated)?
- How small of an *enumeration delay* can we guarantee between any two consecutive solutions? A weak guarantee to aim for is that of *polynomial delay*, i.e., each successive path must be produced

in polynomial time. A more ambitious delay guarantee is *output-linear delay* (i.e., delay linear in each path), or *constant delay*. Given that the paths may have linear size, the notion of constant delay must be explored in a formalism of *enumeration via edits* where each successive result is produced by applying a small number of changes to the previous result [AM23].

- Exploring connections to practice: recursive queries on graph databases with simple paths semantics are highly relevant for practical applications, and questions related to the problems discussed here have been very recently explored in practical work: [LOZ$^+$24].

Supervision and environment. This proposal is for a master-level internship (M2 level, for a duration of 4–6 months), expected to start in Spring 2025. The internship will take place in the LINKS team of the Inria center at University of Lille, in the North of France. The LINKS team focuses on logics, algorithms, formal language theory, and database theory, and offers a dynamic environment for research on these topics. The internship will be co-supervised by Antoine Amarilli[1] (Advanced Research Position at Inria) and Mikaël Monet[2] (Chargé de recherche Inria).

Applications should be sent by email to: `a3nm@a3nm.net` and `mikael.monet@inria.fr`.

References

[AM23] Antoine Amarilli and Mikaël Monet. Enumerating regular languages with bounded delay. In *STACS*, 2023.

[BBG20] Guillaume Bagan, Angela Bonifati, and Benoît Groz. A trichotomy for regular simple path queries on graphs. *Journal of Computer and System Sciences*, 108:29–48, 2020.

[LOZ$^+$24] Qi Liang, Dian Ouyang, Fan Zhang, Jianye Yang, Xuemin Lin, and Zhihong Tian. Efficient regular simple path queries under transitive restricted expressions. *Proceedings of the VLDB Endowment*, 17(7):1710–1722, 2024.

[MT18] Wim Martens and Tina Trautner. Evaluation and enumeration problems for regular path queries. In *ICDT*, 2018.

[1] `https://a3nm.net/`

[2] `https://mikael-monet.net/`