Regular Parameterized Query in Millennium DB

Heyang Li

July 2024

Problem Overview

In this project, a graph database should have labels on both edges and nodes.

Definition 1 (Graph Database)**.** *A graph database G is a tuple $(V, E, \rho, \lambda, Attr, \mu)$, where:*

- *V is a finite set of nodes.*
- *E is a finite set of edges.*
- *$\rho : E \to (V \times V)$ is a total function. Intuitively, $\rho(e) = (v_1, v_2)$ means that e is a directed edge going from v_1 to v_2.*
- *$\lambda : E \to Lab$ is a total function assigning a label to an edge.*
- *$Attr$ is a finite set of attributes*
- *$\mu : V \times Attr \to Val$, where $Val \subseteq \Sigma^* \cup \mathbb{R}$*

Path a *path* in a graph database $G = (V, E, \rho, \lambda, Attr, \mu)$ from s to t is

$$s = v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \dots \xrightarrow{e_k} v_k = t$$

where $\rho(e_i) = (v_{i-1}, v_i)$.

Definition 2 (Regular Query Parameterized Automaton)**.** *A regular query parameterized automaton is a tuple $(L, \chi, Q, \Sigma, q_0, F, \Delta)$, where:*

- *L is a set of labels*
- *$\chi = \mathbb{P} \cup Attr$ is a set of variables, where $\mathbb{P}$ is a set of global real valued variables.*
- *Q is a finite set of states*
- *q_0 is the start state, where q_0 should not have income transitions.*
- *F is a set of final states, where $q_0 \notin F$ and each $q_f \in F$ should not have outcome transitions.*

- *We define the transition by*

$$\Delta \subseteq Q \times (\Sigma \times T(\chi)) \times Q$$

For each $\Delta \in (q, (\sigma, \phi), q')$*, we write* $q \xrightarrow{(\sigma,\phi)} q'$

Parameterized Automaton Query Question: A regular query problem is defined as following: given a graph database $G = (V, E, \rho, \lambda, Attr, \mu)$, a start point $v \in V$, and a regular query parameterized automaton $A = (L, \chi, Q, \Sigma, q_0, F, \Delta)$, if there exists a path:

$$v \xrightarrow{e_1} v_1 \xrightarrow{e_2} \dots \xrightarrow{e_k} v_k$$

And a run of the automaton A:

$$q_0 \xrightarrow{(\sigma_1,\phi_1)} q_1 \xrightarrow{(\sigma_2,\phi_2)} \dots \xrightarrow{(\sigma_k,\phi_k)} q_k \in F$$

such that

$$\forall i \in \{1, \dots, k\} : \lambda(e_i) = \sigma_i$$

and there exists an assignment A for the global parameter, such that

$$\forall i \in \{1, \dots, k\} : A \models \phi_i[x/\mu(v_k, x)]$$

The Algorithm

Millennium DB implement regular queries based on product graph. Intuitively, the product graph is the product between a graph database and a regular query automaton.

Product Graph Given a graph database $G = (V, E, \rho, \lambda, Attr, \mu)$, a regular query parameterized automaton $Aut = (Q, \Sigma, q_0, F, \Delta)$, the product graph $G_\times$, is then defined as the graph database. $G_\times = (V_\times, E_\times, \rho_\times, \lambda_\times, Attr_\times, \mu_\times)$, where:

- $V_\times = V \times Q$
- $E_\times = \{(e, (q_1, (\sigma, \phi), q_2) \in E \times \Delta) \mid \lambda(e) = \sigma\}$
- $\rho_\times(e, d) = ((x, q_1), (y, q_2))$ if:
  $d = (q_1, (\sigma, \phi), q_2)$
  $\lambda(e) = (\sigma, \phi)$
  $\rho(e) = (x, y)$
- $\lambda(e, d) = \lambda(e)$
- $\mu(v, q) = \mu(q)$

We use macro states to store the vertex in the product graph and upper bounds and lower bounds of formulas which only contain global parameters.

Definition 3. *Macro State A macro state S is a tuple $(state, upper, lower)$, where*

- $state \in V_\times$
- $upper : T(\chi) \to \mathbb{R}$ *is a map which contains upper bounds of formulas.*
- $lower : T(\chi) \to \mathbb{R}$ *is a map which contains lower bounds of formulas.*

Given two marco states $S_1 = (state_1, upper_1, lower_1)$ and $S_2 = (state_2, upper_2, lower_2)$. We say $S_1 \models S_2$ if:

- $state_1 = state_2$
- for each formula f, $upper_1(f) \leq upper_2(f)$
- for each formula f, $lower_1(f) \geq lower_2(f)$

Algorithm 1 RPQ in a product graph