天津大学
Tianjin University

WebDB

20th International Workshop on the Web and Databases (WebDB 2017)
Chicago, IL, USA – Sunday, May 14th, 2017

A Path Querying Language for Federation of RDF and Relational Database

Jiahui Zhang, Xiaowang Zhang, and Zhiyong Feng

Tianjin University

zhangjiahui@tju.edu.cn
xiaowangzhang@tju.edu.cn
zyfeng@tju.edu.cn

Outline

- **Introduction**
- **Federated Path Querying Language(FPQ)**
- **Expressiveness of FPQ**
- **Experiment and Evaluation**

Introduction

■ SPARQL

The standard language for querying RDF data since 2008.

■ Navigational Capability

Versa: using XPath over the XML of RDF graphs

SPARQLeR: adding path variables

CPSPARQL: allowing constraints over regular expressions

nSPARQL: applying nested regular expressions

SPARQL 1.1: appending property paths

Introduction

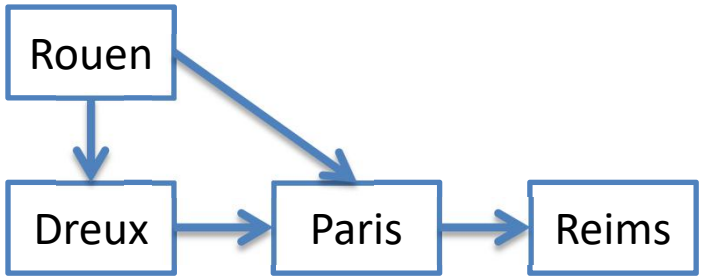


Figure: Geographical map.

Q : Go from Rouen to Reims.

✗ Rouen→Dreux→Paris→Reims

✓ Rouen→Paris→Reims ***Driver***

Introduction

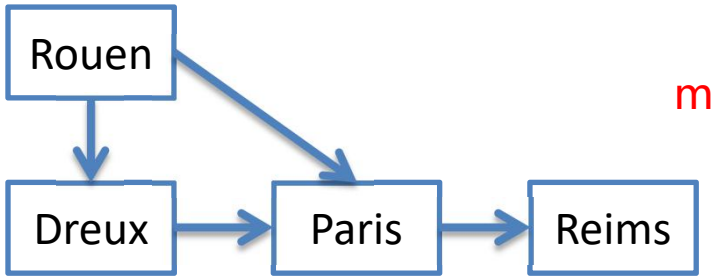


Figure: Geographical map.

Q' : Go from Dreux to Paris.

more orders

Rouen→Dreux→Paris→Reims

Rouen→Paris→Reims → shorter

ID	Time	Passenger	Driver	Start Point	End Point
1	6:40 a.m.	B	E	Rouen	Reims
2	6:50 a.m.	C	?	Dreux	Paris

Table: Taxi-hailing orders.

Introduction

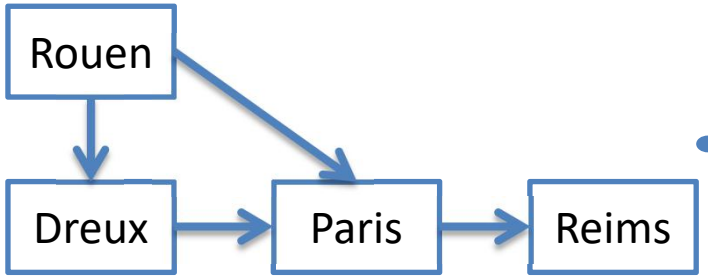


Figure: Geographical map.

- **RDF**

Rouen→Paris→Reims

- **RDF+Relational Database**

Rouen→Dreux→Paris→Reims

ID	Time	Passenger	Driver	Start Point	End Point
1	6:40 a.m.	B	E	Rouen	Reims
2	6:50 a.m.	C	E	Dreux	Paris

Table: Taxi-hailing orders.

Outline

- Introduction
- A Path Querying Language for **Federation** of RDF and Relational Database
- Expressiveness of FPQ
- Experiment and Evaluation

FPQ

■ Federated Path Querying Language(FPQ)

$$q(u,v) := \varphi \wedge \bigwedge_{i=1}^{n} (u_i, e_i, v_i);$$

Conjuction

where

- q is the name of FPQ;
- φ is a conjunctive combination of relations;
- each (u_i, e_i, v_i) for $i \in \{1,\ldots,n\}$ is a NRE triple pattern.

FPQ

■ **Query:**

At a certain time, whether a passenger take a ride?

$$q(?x,?y) = [R(v) \wedge R(e)] \wedge [(?x, exp_1, ?y) \wedge (?x, exp_2, ?y) \wedge (?x, exp_3, ?y)]$$

Conjuction

where

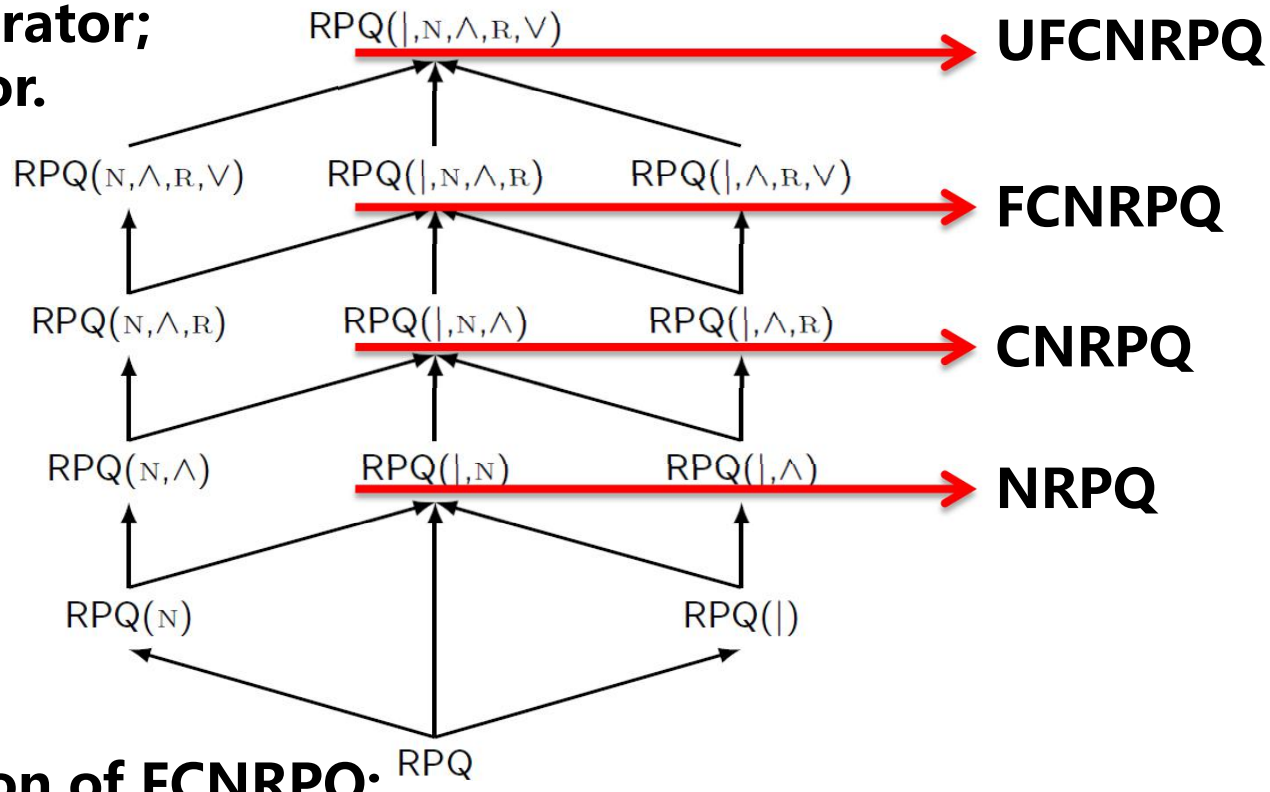
- R(v) := Position(Time, ?x, ?y, ?driverId);
- R(e) := Orders(Time, lonUp, latUp, ?x, ?y, ?driverId);
- exp1 : $next^{-1}$ :: lon;
- exp2 : next :: lat;
- exp3 : next :: nd / next :: ref / $next^{-1}$ :: id.

Outline

- Introduction
- Federated Path Querying Language(FPQ)
- Expressiveness of FPQ
- Experiment and Evaluation

Expressiveness of FPQ

| : Disjunctive operator;
N : Nesting operator;
∧ : Conjunctive operator;
R : Federated operator;
∨ : Union Operator.



The query evaluation of FCNRPQ:

Data complexity → Polynomial time

Combined complexity → NP-complete time

Outline

- Introduction
- Federation Path Querying Language
- Expressiveness of FPQ
- **Experiment and Evaluation**
  - Query and Result
  - Extention

Query

- **Query 1:**

At a certain time, where did the passengers get on vehicles or off?

- **Query 2:**

At a certain time, did the passengers visit tourist attractions on the map?

- **Query 3:**

At a certain time, which roads did the passengers go down from the vehicles?

Query

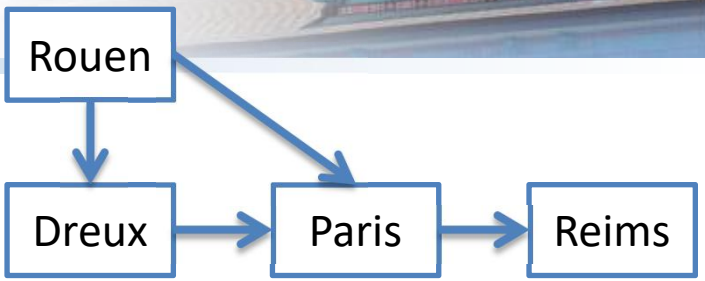


Figure: Geographical map.

- **Query 1:**

At a certain time, where did the passengers get on vehicles or off?

- **Query 2:**

At a certain time, did the passengers visit tourist attractions on the map?

- **Query 3:**

At a certain time, which roads did the passengers go down from the vehicles?

RDF

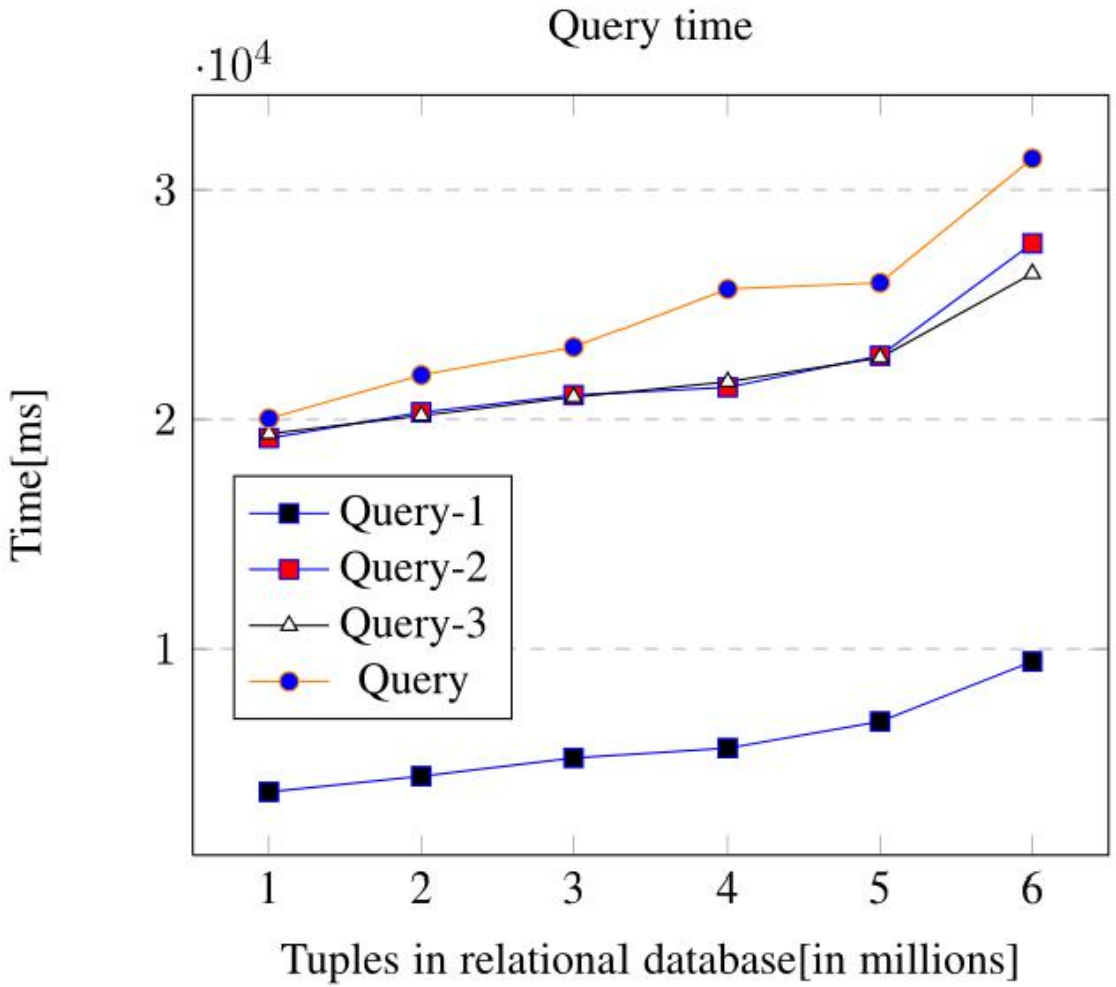
Conjuction

Relational Database

ID	Time	Passenger	Driver	Start Point	End Point
1	6:40 a.m.	B	E	Rouen	Reims
2	6:50 a.m.	C	E	Dreux	Paris

Table: Taxi-hailing orders.

Result


Query time
$\cdot 10^4$
3
2
1
Time[ms]
Query-1
Query-2
Query-3
Query
1
2
3
4
5
6
Tuples in relational database[in millions]

Extention

- Distributed environment

  to solve the memory overflow
- Streaming data

  to process the real time query

Q & A

- Jiahui Zhang

  zhangjiahui@tju.edu.cn

- Xiaowang Zhang

  xiaowangzhang@tju.edu.cn

- Zhiyong Feng

  zyfeng@tju.edu.cn

Thank you!