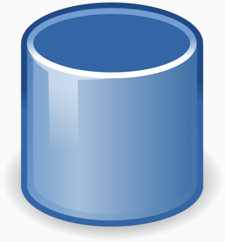
DIG Seminar December 2020

Short overview of my research

Louis Jachiet

Context: Query evaluation

The evaluation problem for databases



Database

$\mathcal{Q}$

Query

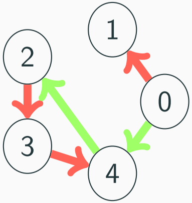
x	y
x_1	y_1
x_2	y_2
x_3	y_3
x_4	y_4
x_5	y_5

Solutions

More expressive queries and more diverse models

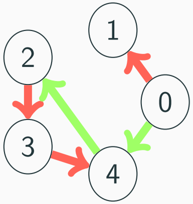
Graph databases and/or recursive queries

How to efficiently evaluate queries on graph?



Graph databases and/or recursive queries

How to efficiently evaluate queries on graph?



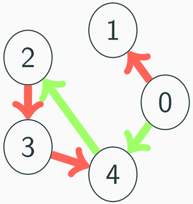
$$\begin{tikzcd}
 & & 1 & \\
2 \arrow[d] & & & \\
 & & & 0 \arrow[uul] \arrow[dl] \\
3 \arrow[r] & & 4 \arrow[uull] &
\end{tikzcd}$$

?x Red/Green$^+$?y

How to efficiently evaluate Regular Path Queries?

Graph databases and/or recursive queries

How to efficiently evaluate queries on graph?



?x Red/Green$^+$?y

How to efficiently evaluate Regular Path Queries?

How to efficiently evaluate relational queries with recursion?

Enumeration algorithms for query evaluation

Idea of enumeration algorithms

Problem

Outputting all solutions to a query might take a very long time.

Idea of enumeration algorithms

Problem

Outputting all solutions to a query might take a very long time.

Solution

Consider solutions that read the input and then output solutions one by one.

A very simple model

Input

A string or a sequence of events or a stream of events:

$$E_1 E_2 \ldots E_k$$

Query

A “string pattern” (typically a regular expression).

Our results

Recent

We can enumerate with linear preprocessing and constant delay when the pattern is regular.

Our results

Recent

We can enumerate with linear preprocessing and constant delay when the pattern is regular.

New

With a regular expression that "scores" each solution, we can enumerate in score order with linear preprocessing and logarithmic delay.

Our results

Recent

We can enumerate with linear preprocessing and constant delay when the pattern is regular.

New

With a regular expression that “scores” each solution, we can enumerate in score order with linear preprocessing and logarithmic delay.

Future 1

We can improve the above bound to constant delay.

Our results

New

With a regular expression that “scores” each solution, we can enumerate in score order with linear preprocessing and logarithmic delay.

Future 1

We can improve the above bound to constant delay.

Future 2

We can enumerate with linear preprocessing and constant delay when the pattern is deterministic context-free.

Miscellaneous

Side project on the transdichotomous RAM model

What really takes constant time? Is division allowed? etc.

Miscellaneous

Side project on the transdichotomous RAM model

What really takes constant time? Is division allowed? etc.

Side project on ontologies and reasoning

Can we publish and make sure that it does not reveal secrets?

Miscellaneous

Side project on the transdichotomous RAM model

What really takes constant time? Is division allowed? etc.

Side project on ontologies and reasoning

Can we publish and make sure that it does not reveal secrets?

Side project on updating strings

We have a string S and updates "change letter to c at position k" what properties we can check efficiently on S after updates.

Future

Develop the reasoning and ontology part.