Context-Free Path Querying with All-Path Semantics by Matrix Multiplication

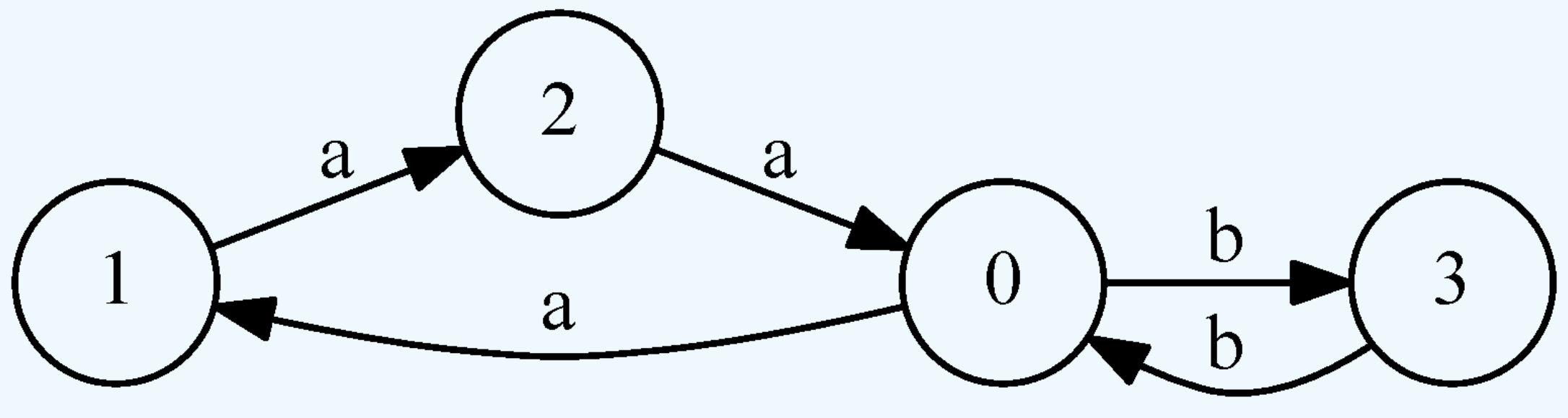
Rustam Azimov

JetBrains Research, Saint Petersburg University, Russia

Rustam.Azimov@jetbrains.com

Relational CFPQ

Find paths which satisfy constraints in form of a formal language $L = \{a^n b^n \mid n > 0\}$



Query = grammar for L: $S \to a\ b \mid a\ S\ b$

Result: $\{(u, v) \mid \exists p \text{ from } u \text{ to } v : \text{word}(p) \in L\}$

Results

- We provide the matrix-based algorithm for CFPQ with all-path query semantics
- We implement the provided algorithm using the GraphBLAS API
- We compare our implementation with other implementations of the linear algebra-based CFPQ algorithms

Future Research

- We plan to obtain GPU-based and distributed implementations
- Try to update the query results dynamically when data changes
- We plan to provide the multiple-source modifications for all linear algebra-based CFPQ algorithms
- Find new applications that required CFPQ

Matrix-Based Algorithm [1]

T is an adjacency matrix of the input graph
The grammar is in the normal form

$$T_{ij} = \{N \mid N \overset{*}{\Rightarrow} \omega, \omega - \text{path bw } i \text{ and } j\}$$

$$T_{ik} \times T_{kj} = \{A \mid B \in T_{ik}, C \in T_{kj}, A \to BC\}$$
$$T^{(i)} = T^{(i-1)} \cup (T^{(i-1)} \times T^{(i-1)})$$

- Can be formulated in terms of boolean matrices multiplication
- Easy to run in parallel environments: GPUs, multithreaded CPUs

CFPQ with All-Path Semantics

- We also need to provide all such paths for all vertex pairs (u, v)
- Use $AllPathIndex = (left, right, middles)$ as matrix elements
  - *left, right* — the starting and the ending vertex of the path
  - *middles* — the set of intermediate vertices of last path concatenation
- Update the matrix operations to keep AllPathIndexes correct
- It is assumed that the sets of paths are computed lazily, to ensure the termination in case of an infinite number of paths

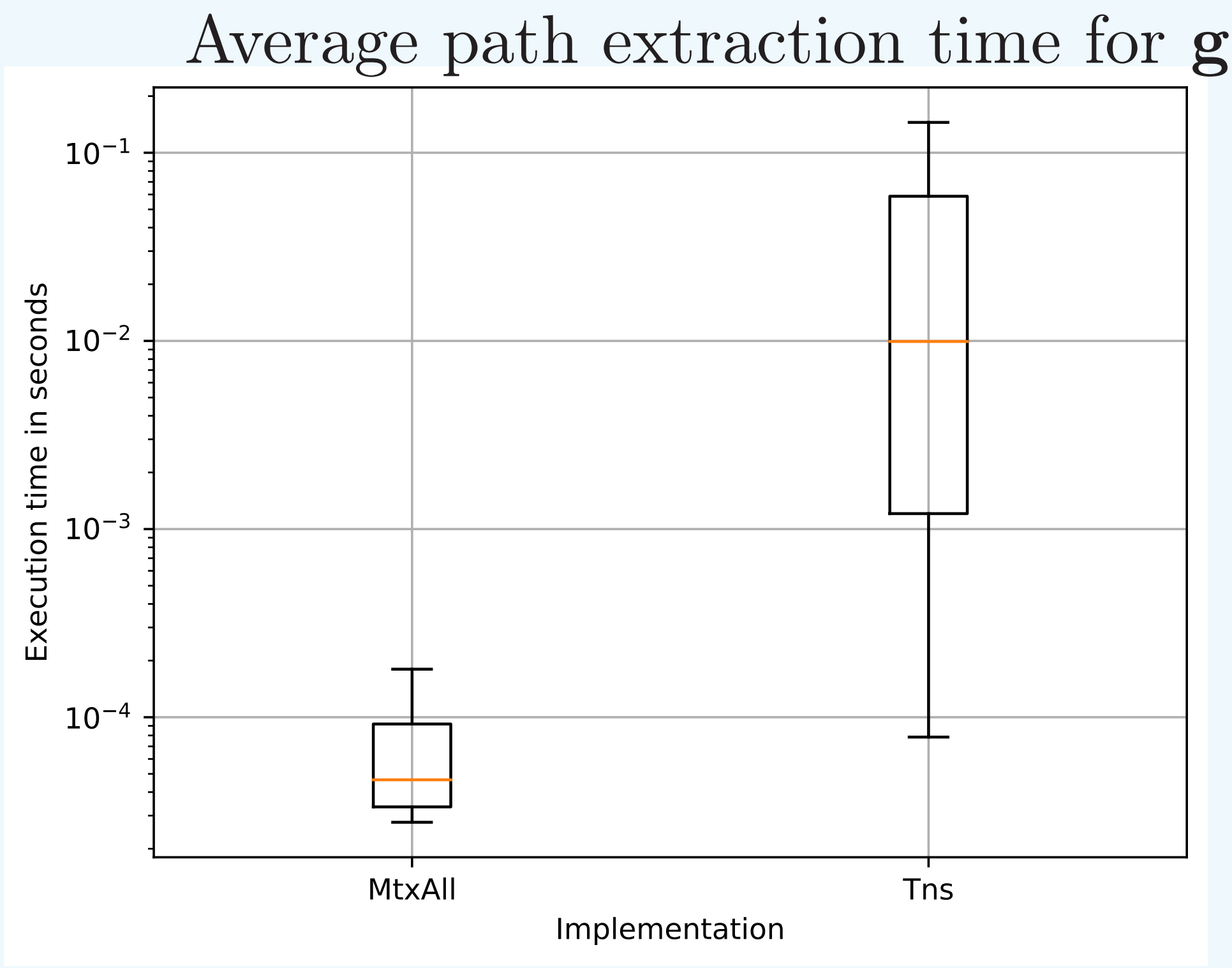
Linear Algebra-Based CFPQ Implementations

- We use the GraphBLAS API and sparse matrix representation
- Our implemenationts are CPU-based
- Implementation of the Kronecker product-based algorithm:
  - **Tns** — our Python implementation for all-path query semantics
- Our matrix-based CFPQ implementations:
  - **MtxRel** — relational query semantics, uses **pygraphblas** — a Python wrapper around the GraphBLAS API
  - **MtxSingle** — single-path query semantics, utilizes **pygraphblas**
  - **MtxAll** — all-path query semantics, operating over AllPathIndex

CFPQ Evaluation with Relational, Single-Path and All-Path Query Semantics

Graph	#V	#E	MtxRel		MtxSingle		MtxAll		Tns	
			Time	Mem	Time	Mem	Time	Mem	Time	Mem
pathways	6 238	18 598	0.01	140	0.01	671	0.01	49	0.01	122
go-hierarchy	45 007	980 218	0.09	255	0.84	671	0.35	195	0.24	252
enzyme	48 815	109 695	0.01	181	0.01	217	0.02	61	0.02	132
eclass_514en	239 111	523 727	0.06	181	0.16	216	0.22	126	0.27	193
go	272 770	534 311	0.94	246	0.93	217	1.13	990	1.27	243
geospecies	450 609	2 311 461	7.48	7645	15.54	22941	32.06	44235	26.32	19537
taxonomy	5 728 398	14 922 125	0.72	1175	1.15	2250	3.84	1507	3.56	1776

Average path extraction time for **go**



- Time in seconds and memory is measured in megabytes
- Graph: real-world ontologies (RDFs), query: same-generation query
- Example of a grammar: $S \to scor\ S\ sco \mid tr\ S\ t \mid scor\ sco \mid tr\ t$

- **MtxAll** constructs index up to 2-3 times slower and consumes more memory than **MtxSingle**
- If we must extract paths many times for a once constructed index then **MtxAll** is preferable than **Tns**

Contact Us

- Semyon Grigorev: s.v.grigoriev@spbu.ru
- Rustam Azimov:Rustam.Azimov@jetbrains.com
- Ilya Epelbaum: iliyepelbaun@gmail.com

- Dataset: `https://github.com/JetBrains-Research/CFPQ_Data`
- Implementations: `https://github.com/JetBrains-Research/CFPQ_PyAlgo`

X'an, Shaanxi, China
SIGMOD/PODS 2021

GRADES NDA

June 20th, 2021

References

[1] Rustam Azimov and Semyon Grigorev. Context-free path querying by matrix multiplication. In *Proceedings of the 1st ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, GRADES-NDA '18, pages 5:1–5:10, 2018.

[2] Rustam Azimov, Ilya Epelbaum, and Semyon Grigorev. Context-free path querying with all-path semantics by matrix multiplication. In *Proceedings of the 4th Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, GRADES-NDA '21, 2021.

[3] Egor Orachev, Ilya Epelbaum, Rustam Azimov, and Semyon Grigorev. Context-free path querying by kronecker product. In *European Conference on Advances in Databases and Information Systems*, pages 49–59. Springer, 2020.

Acknowledgments

The research was supported by the Russian Science Foundation, grant number 18-11-00100