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

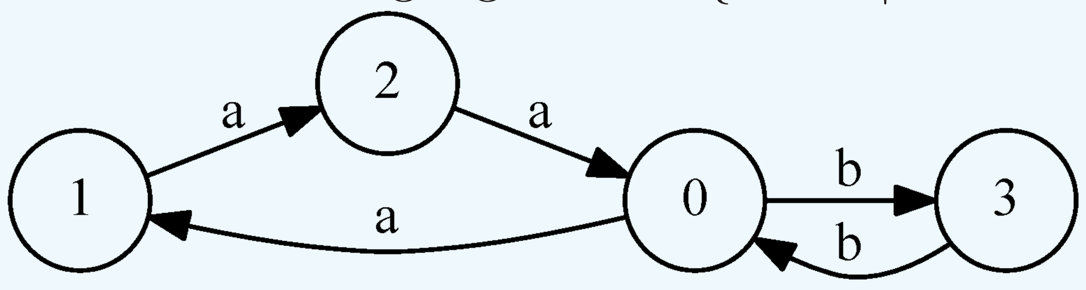
Rustam Azimov

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 single-path query semantics
- We provide several implementations of the CFPQ algorithms for both query semantics which use RedisGraph as graph storage
- We extend the dataset presented in [3] with new real-world and synthetic cases of CFPQ

Future Research

- Extend the matrix-based CFPQ algorithm to all-path query semantics
- Update the query results dynamically when data changes
- Include real-world cases from the area of static code analysis to the dataset
- 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, clusters

CFPQ with Single-Path Semantics

- We also need to provide one such path for all node pairs (u, v)
- Use $PathIndex = (left, right, middle, height, length)$ as matrix elements
  - $left, right$ — the starting and the ending node of the path
  - $middle$ — the intermediate node of last path concatenation
  - $height, length$ — the height and the length of path
- Update the matrix operations to keep PathIndexes correct
- After the CFPQ algorithm we can extract the path stored
- The path extraction time is small and linear in the length of the path

New Implementations of CFPQ

- We use RedisGraph graph database as storage
- CPU-based implementations:
  - **RG_CPU**$_{rel}$ — for the relation query semantics
  - **RG_CPU**$_{path}$ — for the single-path query semantics
- GPGPU-based implementations:
  - **RG_CUSP**$_{rel}$ — relational semantics, utilizes a **CUSP** library
  - **RG_SPARSE**$_{rel}$ — relational semantics, uses low-latency on-chip shared memory for the hash table of each row of the result matrix
  - **RG_SPARSE**$_{path}$ — single-path semantics, operating over PathIndex

CFPQ Evaluation with Relational and Single-Path Query Semantics

Name	#V	#E	Relational semantics index						Single path semantics index			
			RG_CPU$_{rel}$		RG_CUSP$_{rel}$		RG_SPARSE$_{rel}$		RG_CPU$_{path}$		RG_SPARSE$_{path}$	
			Time	Mem	Time	Mem	Time	Mem	Time	Mem	Time	Mem
pathways	6,238	37,196	0.011	0.1	0.019	0.1	0.007	0.1	0.021	0.5	0.021	2.0
go-hierarchy	45,007	1,960,436	0.091	16.3	0.433	650.0	0.108	121.2	0.976	92.0	0.336	125.0
enzyme	48,815	219,390	0.018	5.9	0.021	0.1	0.018	4.0	0.029	8.1	0.043	6.0
eclass_514en	239,111	1,047,454	0.067	13.8	0.075	14.0	0.166	16.0	0.195	31.2	0.496	26.0
go	272,770	1,068,622	0.604	28.8	0.590	70.0	0.365	30.2	1.286	75.7	0.739	45.4
geospecies	450,609	4,622,922	7.146	16934.2	—	—	0.856	5274	15.134	35803.6	1.935	5282

- Graph: classical ontologies (RDFs)
- Query: same-generation query
- Example of a grammar: $S \to scor\ S\ sco \mid tr\ S\ t \mid scor\ sco \mid tr\ t$
- GPGPUs utilization significantly increases the performance
- Implementations with sparse matrices significantly faster than others
- The cost of computing matrices with PathIndexes is not high

Contact Us

Our team:

- Semyon Grigorev: s.v.grigoriev@spbu.ru
- Rustam Azimov:Rustam.Azimov@jetbrains.com
- Arseniy Terekhov: simpletondl@yandex.ru
- Artyom Khoroshev: arthoroshev@gmail.com

Both dataset and implementations are available on GitHub

- Dataset link: `https://github.com/JetBrains-Research/CFPQ_Data`
- Implementations link: `https://github.com/YaccConstructor/RedisGraph`

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] Arseniy Terekhov, Artyom Khoroshev, Rustam Azimov, and Semyon Grigorev. Context-free path querying with single-path semantics by matrix multiplication. In *Proceedings of the 3rd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, GRADES-NDA '20, 2020.

[3] Nikita Mishin, Iaroslav Sokolov, Egor Spirin, Vladimir Kutuev, Egor Nemchinov, Sergey Gorbatyuk, and Semyon Grigorev. Evaluation of the context-free path querying algorithm based on matrix multiplication. In *Proceedings of the 2Nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, GRADES-NDA'19, pages 12:1–12:5, New York, NY, USA, 2019. ACM.

Acknowledgments

The reported study was funded by grant from JetBrains Research and by RFBR, project number 19-37-90101