Evaluation of the Context-Free Path Querying Algorithm Based on Matrix Multiplication

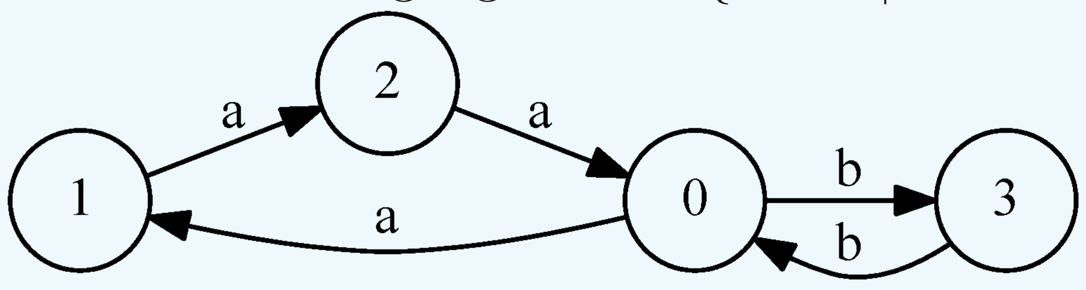
Semyon Grigorev

JetBrains Research, Saint Petersburg University, Russia

s.v.grigoriev@spbu.ru, Semyon.Grigorev@jetbrains.com

Context-Free Path Querying

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

- Dataset for CFPQ evaluation is collected and published
  - Contains both graphs and queries
  - Contains both real-world and synthetic graphs
- Several CFPQ algorithms implementations are created, evaluated and published

Future Research

- Create an open extensible platform for CFPQ algorithms evaluation
- Extend dataset with new data
- Implement and evaluate distributed matrix-based CFPQ algorithms
- Implement and evaluate sparse boolean matrix-based CFPQ algorithms

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

Implementations

Our implementations [2]:

[Scipy] Matrix-based algorithm which uses sparse matrices from **Scipy** library (**Python**)

[M4RI] Matrix-based algorithm which uses dense matrices multiplication from **m4ri** library (Method of Four Russians, **C**)

[GPU] Matrix-based algorithm which uses our own implementation of the naïve boolean matrix multiplication in **CUDA C**

Reference implementations:

[CuSprs] Matrix-based algorithm [1] uses NVIDIA cuSPARSE library (**CUDA C, GPGPU**)

[CYK] CYK-based algorithm [3] implemented in **Java** (CPU)

We Need More Real-World Data

Graph: classical ontologies (RDFs)
Query: same-generation query over `type` and `SubClassOf` relations
Grammar: $S \to scor\ S\ sco \mid tr\ S\ t \mid scor\ sco \mid tr\ t$

RDF			Algorithms				
Name	#V	#E	Scipy	M4RI	GPU	CuSprs	CYK
atm-prim	291	685	3 ms	2 ms	1 ms	269 ms	8.5 min
biomed	341	711	3 ms	5 ms	1 ms	283 ms	7.1 min
pizza	671	2604	6 ms	8 ms	1 ms	292 ms	54 min
wine	733	2450	7 ms	6 ms	1 ms	294 ms	68 min

- 2019 (GPU) is 10^6 times faster than 2016 (CYK) on real-world data
  - Reasonable time even for CPU based implementations
- We should find bigger RDFs
- We should find other real-world cases for CFPQ
  - Both graphs and queries

We Should Do More Research on the Algorithms Scaling

	Graph	Scipy	M4RI	GPU	CuSprs
Sparse graphs are generated by GTgraph Query: $S \to a\ S\ b \mid a\ b$	G10k-0.001	37 s	2 s	0.2 s	35 s
	G10k-0.1	601 s	1 s	0.1 s	395 s
	G40k-0.001	-	97 s	8.1 s	-
	G80k-0.001	-	1142 s	65 s	-
Graph is a cycle Query: $S \to S\ S \mid a$	G25k	-	33 s	5 s	-
	G50k	-	360 s	44 s	-
	G80k	-	1292 s	190 s	-

- We can handle graphs with 80k vertices in a reasonable time by using GPGPU
  - Technical bound: GPGPU RAM does not fit bigger graphs
- We should evaluate multi-GPU systems
- We should evaluate distributed solutions
- We should implement a sparse boolean matrices library for GPGPU

Contact Us

Our team:

- Semyon Grigorev: s.v.grigoriev@spbu.ru
- Nikita Mishin: mishinnikitam@gmail.com
- Iaroslav Sokolov: sokolov.yas@gmail.com
- Egor Spirin: egor@spirin.tech
- Egor Nemchinov: nemchegor@gmail.com
- Vladimir Kutuev: vladimir.kutuev@gmail.com
- Sergey Gorbatyuk: sergeygorbatyuk171@gmail.com

Both dataset and implementations are available on GitHub:
`https://github.com/SokolovYaroslav/CFPQ-on-GPGPU`

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] 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.

[3] Xiaowang Zhang, Zhiyong Feng, Xin Wang, Guozheng Rao, and Wenrui Wu. Context-free path queries on rdf graphs. In Paul Groth, Elena Simperl, Alasdair Gray, Marta Sabou, Markus Krötzsch, Freddy Lecue, Fabian Flöck, and Yolanda Gil, editors, *The Semantic Web – ISWC 2016*, pages 632–648, Cham, 2016. Springer International Publishing.

Acknowledgments

The research is supported by the JetBrains Research grant and the Russian Science Foundation grant 18-11-00100