Context-Free Path Querying via Matrix Equations

Yuliya Susanina
Saint Petersburg State University
7/9 Universitetskaya nab.
St. Petersburg, Russia 199034
JetBrains Research
Primorskiy prospekt 68-70, Building 1
St. Petersburg, Russia 197374
st049970@student.spbu.ru

ABSTRACT

Context-free path querying (CFPQ) underlies many problems related to graph-structured data in such areas as bioinformatics, static code analysis, graph databases. The importance of developing highly efficient algorithms for CFPQ connected with a large amount of data to process. Computational mathematics may be very useful due to the meaningful theoretical basis and multiple constantly improving implementations. We show how to reduce GFPQ evaluation to solving the systems of matrix equations over $\mathbb{R}$. Thus we propose a way to reduce CFPQ to problem with avalilable high-performance solutions. Also we demonstrate the applicability of our approach to the real-world data analysis.

CCS CONCEPTS

• **Information systems → Query languages**; • **Theory of computation → Formal languages and automata theory**;

KEYWORDS

context-free path querying, graph databases, context-free grammar, nonlinear matrix equations, newton method

ACM Reference format:
Yuliya Susanina. 2018. Context-Free Path Querying via Matrix Equations. In *Proceedings of Woodstock '18: ACM Symposium on Neural Gaze Detection, Woodstock, NY, June 03–05, 2018 (Woodstock '18),* 3 pages.
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Context-free path querying (CFPQ)—a way to specify path constraoints in terms of context-free grammars—is becoming more popular in many areas, e. g. bioinformatics [11], graph databases [5, 6, 13] or static code analysis [8, 14]. This type of query increases the expressive power of commonly used regular languages constrained path quiry (RPQ) and therefore forms a promising research area. Dispite the fact taht many different algortihms for CFPQ are proposed [2, 6, 9, 12, 15], resent research shows that real-world data handling is stil a problem [5].

One of te resent experiments [7] shows that one of the promising way to get high-performance CFPQ is to reduthe the problem to well-esteblised problem with available high-performance solutions. One of such reduction was proposed by Rustam Azimov [2] who shows that CFPQ can be reduced to Boolean matrix multiplication. By this way we can offload the most computationally intensive part to high-performance libraries for matrix processing, which utilize GPGPUS and other modern harware.

On the other hand, there are several results of applying linear algebra methods to logic programming [1, 10]. In [10] it is showh that ecvaluation of a subset of datalog queryes can be reduced to matrix equation solving. This way one can use numerical linear algebra and computational mathematics to improve performance of query evaluation. The question is may we use this way for CFPQ? By this reduction we can utilize another set of high-performance solutions which are used for numerical analysis and equation solving to improve performance of CFPQ. Also, approximate computational methods can accelerate CFPQs processing. And most importantly, the popularity of artificial intelligence techniques pushed the development and improvement of many efficient libraries for numerical computing.

In this work we propose a new approach for CFPQs processing, based on solving the systems of equations in the set of real numbers $\mathbb{R}$. We also assess the feasibility of using both accurate and approximate methods of computational mathematics, and utilize respective metods, such as a Newton metod with high-performancce implementations available. So, we propose an alternative way to reduce CFPQ to problem with avalilable high-performance solutions. The evaluation of our approach on a set of conventional benchmarks shows that it is camparable with matrix-based approach abd applicabile for real-world data processing.

2 PRELIMINARIES

Context-free grammar (CFG) is a triple $G = (N, \Sigma, R)$, where N is a set of nonterminal symbols, Σ is a set of terminal symbols and R is a set of productions of the followings form: $A \Rightarrow \alpha$, $\alpha \in (N \cup \Sigma)^*$. $\mathcal{L}(G_S)$ denotes a language specified by CFG G with respect to $S \in N$: $\mathcal{L}(G_S) = \{\omega \mid S \Rightarrow^*_G \omega\}$.

Directed graph is a triple $D = (V, E, \sigma)$, where V is a set of vertices, $\sigma \subseteq \Sigma$ is a set of labels, and a set of edges $E \subseteq V \times \sigma \times V$. Denote a path from the node m to the node n in graph D as $m\lambda n$,

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

where λ is the unique word, obtained by concatenating the labels of the edges along this path. P is a set of all paths in D.

Relational semantics for CFPQ (according Hellings [3]) is for a graph D and a CFG G, to find *context-free relations* $R_A \subseteq V \times V$ such that for each $A \in N$: $R_A = \{(m, n) \mid m\lambda n \in P, \lambda \in \mathcal{L}(G_A)\}$. All such relations are finite, so each of them can be represented as a Boolean matrix: $(T_A)_{i,j} = 1 \iff (i, j) \in R_A$.

3 EQUATION-BASED APPROACH

To reduce CFPQ to equation solving we use the similar way which use Taisuke Sato to reduce Datalog prpgram evaluation to lineear algebra operations [10]. For the given graph and the fiven grammar each terminal and nonterminal specifies a finite relation which can be representad as a Boolena matrix as presented above. After that for each production $N_i \to \beta_0^0 \dots \beta_k^0 \mid \dots \mid \beta_0^l \dots \beta_m^l, \beta_j^i \in \Sigma \cup N$ we can create an equation $T_{N_i} = T_{\beta_0^0} \cdot \dots \cdot T_{\beta_k^0} + \dots + T_{\beta_0^l} \cdot \dots \cdot T_{\beta_m^l}$

For example, consider a simple CFG $G_1 : S \to aSb \mid ab$. Respective Booleman matrix equation:

$$T_S = T_A T_S T_B + T_A T_B$$

We can solve this equation by naïve iterative process:

$$T_S^0 = \mathbf{0}$$
$$T_S^{k+1} = T_a T_S^k T_b + T_a T_b$$

Unfortunately, this modification does not entail any meaningful improvements. But we can consider another equation over $\mathbb{R}$:

$$\mathcal{T}_S = \epsilon(T_A \mathcal{T}_S T_B + T_A T_B)$$

And the corresponding matrix series $\{\mathcal{T}_S^k\}$, which converges to $\mathcal{T}_S^*$ when $\mathcal{T}_S^k \leq \mathbf{1}$ as a a monotonically increasing series of matrices with an upper bound:

$$\mathcal{T}_S^0 = \mathbf{0}$$
$$\mathcal{T}_S^{k+1} = \epsilon(T_A \mathcal{T}_S^k T_B + T_A T_B).$$

It can be proved that $((\mathcal{T}_S^{k+1})_{ij} > 0 \iff (T_S^{k+1})_{ij} = 1)$ and $ceil(\mathcal{T}_S^*) = T_S^*$, where $ceil(x)$ returns the smallest integer not less than x [10].

So, each rule of the form $N_i \to \beta_0^0 \dots \beta_k^0 \mid \dots \mid \beta_0^l \dots \beta_m^l, \beta_j^i \in \Sigma \cup N$ can be replaced with an equation $T_{N_i} = \epsilon_{N_i}(T_{\beta_0^0} \cdot \dots \cdot T_{\beta_k^0} + \dots + T_{\beta_0^l} \cdot \dots \cdot T_{\beta_m^l})$, where ϵ_X is chosen such that $\mathcal{T}_X^k \leq \mathbf{1}$ for each k. In real-world cases we deal with the systems of matrix equations because grammars contain more than one nonterminal. We can consider the following possible cases of these systems.

Linear Equations. If the input CFG is linear, each equation of the system is of the form of generalized Sylvester equation [**?**]: $\sum_{i=1}^k A_i X B_i = C$. For $k = 1, 2$ we can solve it in $O(|V|^3)$ [**?**]. Otherwise, for $k > 2$ it can be reduced using Kronecker product to solving a linear system $Ax = b$, where A is a matrix of size $(|V|^2 \times |V|^2)$ and time required to compute its solution is $O(|V|^6)$ or $O(|V|^{4\omega+2})$ with more efficient matrix multiplication algorithms. The use of sparse matrix representation can be very efficient for solving the equations of this type.

For the system of equations, we construct the dependency graph D_G for nonterminals of the given grammar G and split the set of the equations into the disjoint subsets accordingly to the set of strongly connected components in D_G, which can be found in $O(|V| + |E|)$. Then we solve our system in stages, for each subset.

Nonlinear Equations. For the nonlinear case we can rewrite each equation of the form $X = \Psi(X)$ to the equivalent $F(X) = X - \Psi(X) = 0$ and use Newton's method for nonlinear functions root finding:

$$F(X) = \mathbf{0}, X_0$$
$$X_{i+1} = X_i - (F'(X_i))^{-1} F(X_i) \iff \begin{cases} F'(X_i) H_i = -F(X_i) \\ X_{i+1} = X_i + H_i \end{cases}$$

Here X_0 is an initial guess, in our case $X_0 = \mathbf{0}$, as our solution is a matrix consists of small positive numbers. The convergence of this method can be quadratic which allows finding the solution significantly faster. Even as it is necessary to solve an equation for H_i (also generalized Sylvester equation) on each iteration step, the majority of high-performance implementations do not compute the Jacobian inverse and use its approximate value.

The main difficulty in using Newton's method is choosing an appropriate ϵ, to ensure the least positive solution for nonlinear equations ϵ must be smaller than $\frac{1}{|V|}$.

4 EVALUATION

The equation-based approach for CFPQ was implemented and evaluated on **Query 2** from [2]. The equation constructed for this query were solved on the CPU by using two functions form Python package *scipy* [4]: solving as a sparse linear system using *spsolve* (**sSLV**) and funding roots of a function using *optimize.newton_krylov* (**dNWT**).

Table 1: Evaluation results for Query 2 (in ms)

Ontology	|V|	dNWT	sSLV	dGPU	sCPU	sGPU
bio-meas	341	284	35	276	91	24
people-pets	337	73	49	144	38	6
funding	778	502	184	1246	344	27
wine	733	791	171	722	179	6
pizza	671	334	161	943	256	23

We compare (Table 1) our solution with the first matrix-based algorithm implementations described in [2] and conclude that equation based approach can be applied on real-world data as well as the matrix-based algorithm. Moreover, we can improve the performance by the utilization of parallel techniques for matrix operations.

5 CONCLUSION AND FUTURE WORK

We proposed a new approach for CFPQ, based on solving the systems of equations over $\mathbb{R}$. The evaluation of our approach on a set of conventional benchmarks showed its applicability for real-world data analysis.

Direction for future research is utitlization of high-performance solvers, for example which utilize GPGPU and destributed computations, and comparison with ither algorithmns for CFPQ. A lso, it is interesting to determine the subclasses of (system of) polynomial equations whose solution can be reduced to CFPQ and try to construct bidirectional reduction between CFPQ and these subclasses, thereby finding efficient solutions for both these problems at a time.

ACKNOWLEDGMENTS

The research was supported by the Russian Science Foundation grant 18-11-00100 and a grant from JetBrains Research.

REFERENCES

[1] Yaniv Aspis. 2018. *A Linear Algebraic Approach to Logic Programming*. Ph.D. Dissertation. Imperial College London.

[2] Rustam Azimov and Semyon Grigorev. 2018. 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)*. ACM, New York, NY, USA, Article 5, 10 pages. https://doi.org/10.1145/3210259.3210264

[3] Jelle Hellings. 2014. Conjunctive Context-Free Path Queries. OpenProceedings.org. https://doi.org/10.5441/002/ICDT.2014.15

[4] Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001–2019. SciPy: Open source scientific tools for Python. http://www.scipy.org/ [Online; accessed 5.3.2019].

[5] Jochem Kuijpers, George Fletcher, Nikolay Yakovets, and Tobias Lindaaker. 2019. An Experimental Study of Context-Free Path Query Evaluation Methods. In *Proceedings of the 31st International Conference on Scientific and Statistical Database Management (SSDBM '19)*. ACM, New York, NY, USA, 121–132. https://doi.org/10.1145/3335783.3335791

[6] Ciro M. Medeiros, Martin A. Musicante, and Umberto S. Costa. 2018. Efficient Evaluation of Context-free Path Queries for Graph Databases. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing (SAC '18)*. ACM, New York, NY, USA, 1230–1237. https://doi.org/10.1145/3167132.3167265

[7] Nikita Mishin, Iaroslav Sokolov, Egor Spirin, Vladimir Kutuev, Egor Nemchinov, Sergey Gorbatyuk, and Semyon Grigorev. 2019. 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)*. ACM, New York, NY, USA, Article 12, 5 pages. https://doi.org/10.1145/3327964.3328503

[8] Thomas Reps. 1997. Program Analysis via Graph Reachability. In *Proceedings of the 1997 International Symposium on Logic Programming (ILPS '97)*. MIT Press, Cambridge, MA, USA, 5–19. http://dl.acm.org/citation.cfm?id=271338.271343

[9] Fred C. Santos, Umberto S. Costa, and Martin A. Musicante. 2018. A Bottom-Up Algorithm for Answering Context-Free Path Queries in Graph Databases. In *Web Engineering*, Tommi Mikkonen, Ralf Klamma, and Juan Hernández (Eds.). Springer International Publishing, Cham, 225–233.

[10] TAISUKE SATO. 2017. A linear algebraic approach to datalog evaluation. *Theory and Practice of Logic Programming* 17, 3 (May 2017), 244–265. https://doi.org/10.1017/s1471068417000023

[11] Petteri Sevon and Lauri Eronen. 2008. Subgraph Queries by Context-free Grammars. *Journal of Integrative Bioinformatics* 5, 2 (June 2008). https://doi.org/10.1515/jib-2008-100

[12] Ekaterina Verbitskaia, Ilya Kirillov, Ilya Nozkin, and Semyon Grigorev. 2018. Parser Combinators for Context-free Path Querying. In *Proceedings of the 9th ACM SIGPLAN International Symposium on Scala (Scala 2018)*. ACM, New York, NY, USA, 13–23. https://doi.org/10.1145/3241653.3241655

[13] Mihalis Yannakakis. 1990. Graph-theoretic Methods in Database Theory. In *Proceedings of the Ninth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS '90)*. ACM, New York, NY, USA, 230–242. https://doi.org/10.1145/298514.298576

[14] Qirun Zhang, Michael R. Lyu, Hao Yuan, and Zhendong Su. 2013. Fast Algorithms for Dyck-CFL-reachability with Applications to Alias Analysis. In *Proceedings of the 34th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '13)*. ACM, New York, NY, USA, 435–446. https://doi.org/10.1145/2491956.2462159

[15] X. Zhang, Z. Feng, X. Wang, G. Rao, and W. Wu. 2016. Context-free path queries on RDF graphs. In *International Semantic Web Conference*. Springer, 632–648.