Context-Free Path Queries and Sparse Matrix Multiplication

Ekaterina Shemetova
The Thørväld Group
Hekla, Iceland
larst@affiliation.org

Arseniy Terekhov
Inria Paris-Rocquencourt
Rocquencourt, France

Julia Susanina
Inria Paris-Rocquencourt
Rocquencourt, France

Semyon Grigorev
Rajiv Gandhi University
Doimukh, Arunachal Pradesh, India

ABSTRACT

Truly subcubic algorithm for CFPQ is proposed.

Smthng else?

Abstract is very abstract.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

datasets, neural networks, gaze detection, text tagging

ACM Reference Format:

Ekaterina Shemetova, Arseniy Terekhov, Julia Susanina, and Semyon Grigorev. 2018. Context-Free Path Queries and Sparse Matrix Multiplication. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Context-Free Path Querying (CFPQ) is a sublcass of Language-constrained path problem, where language is set to be Context-Free.

Importance of CFPQ. Application areas. RDF, Graph database querying, Graph Segmentation in Data provenance, Biological data analysis, static code analysis.

1.1 An Example

Example of graph and query. Should be used in explataion below.

1.2 Existing CFPQ Algorithms

Number of problem-specific solutions in static code analysis.

Hellings, Ciro et al, Kujpers, Sevon, Verbitskaya, Azimov, Ragozina

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

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

1.3 Existing Theoretical Results

Existing theoretical results

Linear input. Valiant [7], Lee [6].

Yannacacis [8]? Reps?

Bradford [2]

RSM [5].

C alias analysis [9]

Chatterjee [4]

For trees

Truly-subcubic algorithm is stil an open problem.

Truly-subcubic for Language Editing Distance [3].

1.4 Our Contribution

Matrices Can we improve that

(1) Cubic
(2) Reduction to Dyck
(3) Subcubic for !!!
(4) Interconnection with !!!

This paper is organized as follows.

2 PRELIMINARIES

We introduce !!!!

2.1 Context-Free Path Querying

Graph, grammar, etc.

Let $i\pi j$ denote a unique path between nodes i and j of the graph and $l(\pi)$ denotes a unique string which is obtained from the concatenation of edge labels along the path π. For a context-free grammar $G = (\Sigma, N, P, S)$ and directed labelled graph $D = (Q, \Sigma, \delta)$, a triple (A, i, j) is *realizable* iff there is a path $i\pi j$ such that nonterminal $A \in N$ derives $l(\pi)$.

2.2 Matrix-Based algortihm for CFPQ

Custom semiring definition.

Short description of the Rustam algorithm + pseudocode **??**.

Algorithm 1 Context-free recognizer for graphs

```
function contextFreePathQuerying(D, G)
    n ← the number of nodes in D
    E ← the directed edge-relation from D
    P ← the set of production rules in G
    T ← the matrix n × n in which each element is ∅
    for all (i, x, j) ∈ E do                    ▷ Matrix initialization
        T_{i,j} ← T_{i,j} ∪ {A | (A → x) ∈ P}
    while matrix T is changing do
        T ← T ∪ (T × T)    ▷ Transitive closure T^{cf} calculation
    return T
```

2.3 Sparse Matrices

The function $nnz(A)$ denotes the number of non-zero elements in matrix A.

3 CUBIC UPPER BOUND USING SPARSE MATRIX MULTIPLICATION

In this section we show that the Algorithm 1 has complexity $O(n^3)$ if sparse matrix multiplication is used. The reason is that one needs to compute at most $|N|n^2$ realizable triples in total during all iterations of the algorithm, whereas the number of new triples found on each i-th iteration can be relatively small. The number of operations of the i-th iteration can be reduced by multiplying the matrix $B_{(i-2)}$ containing all the previously found triples, on the matrix $A_{(i-1)}$ which has only those triples firstly obtained in the $(i-1)$-th iteraition. In other words, we have:

$$B_i = B_{i-1} + A_i, \tag{1}$$

where

$$A_i = (B_{i-2}A_{i-1} + A_{i-1}B_{i-2} + A_{i-1}A_{i-1}) - B_{i-2}. \tag{2}$$

So the following two conditions hold:

(1) $\forall i, B_{(i-2)} \cap A_{(i-1)} = \emptyset$;
(2) $\forall i, j, A_i \cap A_j = \emptyset$.

Notice that the first condition implies that one of the two multiplied matrices should be sparse, because $nnz(B_{(i-2)})+nnz(A_{(i-1)}) \leq |N|n^2$. Also, by the second condition matrices A are pairwise disjoint, therefore $nnz(\sum_{i=1}^{n^2} A_i) \leq |N|n^2$.

The Algorithm 1 can be modified using Equations 1 and 2 instead of the naive calculation of transitive closure on every iteration. It is important that the modified agorithm has the same number of iterations in the worst case as the original one — $|N|n^2 = O(n^2)$. This is because the height of the parse tree does not exceed this value. As in the Algorithm 1, modified version derives new triples, going from leaves to the root of the parse tree.

Theorem 3.1. *The matrix B_{n^2} containing all possible realizable triples can be calculated in $O(n^3)$ time.*

Proof. The correctness of the algorithm can be easily deduced from the correctness of the Algorithm 1 [1]. Now we show the cubic time complexity of the modified algorithm. Consider the equation for calculating B_{n^2}:

$$B_{n^2} = B_{n^2-1} + B_{i-2}A_{i-1} + A_{i-1}B_{i-2} + A_{i-1}A_{i-1} =$$

$$= B_{n^2-2} + B_{i-3}A_{i-2} + A_{i-2}B_{i-3} + A_{i-2}A_{i-2}+$$
$$+B_{i-2}A_{i-1} + A_{i-1}B_{i-2} + A_{i-1}A_{i-1} = \ldots =$$
$$= B_1 + B_1B_1 + \sum_{i=2}^{n^2-1} B_{i-2}A_{i-1} + \sum_{i=2}^{n^2-1} A_{i-1}B_{i-2} + \sum_{i=2}^{n^2-1} A_{i-1}A_{i-1}.$$

Suppose, without loss of generality, that the matrix B_{i-2} is dense, than the matrix A_{i-1} is sparse. Using naive sparse matrix multiplication algorithm, we have that $O(nnz(A_{i-1})n)$ operations are needed for multiplication of the matrix B_{i-2} and the matrix A_{i-1}. Let $T(AB)$ be the number of operations that need to be performed to multiply matrices A and B. Thus, the total number of operation for obtaining B_{n^2} is in

$$T(B_1B_1)+ \sum_{i=2}^{n^2-1} T(B_{i-2}A_{i-1})+ \sum_{i=2}^{n^2-1} T(A_{i-1}B_{i-2})+ \sum_{i=2}^{n^2-1} T(A_{i-1}A_{i-1}) =$$
$$= O(n^{\omega}) + O(\sum_{i=2}^{n^2-1} nnz(A_{i-1})n) =$$
$$= O(n^{\omega}) + O(nnz(\sum_{i=2}^{n^2-1} A_{i-1})n) = O(|N|n^2 n) = O(n^3).$$

□

4 CONCLUSION

Conclusion and future work.

Efficient implementation?

!!!

REFERENCES

[1] 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)* (Houston, Texas) *(GRADES-NDA '18)*. ACM, New York, NY, USA, Article 5, 10 pages. https://doi.org/10.1145/3210259.3210264

[2] P. G. Bradford. 2017. Efficient exact paths for dyck and semi-dyck labeled path reachability (extended abstract). In *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*. 247–253. https://doi.org/10.1109/UEMCON.2017.8249039

[3] Karl Bringmann, Fabrizio Grandoni, Barna Saha, and Virginia Vassilevska Williams. 2019. Truly Subcubic Algorithms for Language Edit Distance and RNA Folding via Fast Bounded-Difference Min-Plus Product. *SIAM J. Comput.* 48, 2 (2019), 481–512. https://doi.org/10.1137/17M112720X arXiv:https://doi.org/10.1137/17M112720X

[4] Krishnendu Chatterjee, Bhavya Choudhary, and Andreas Pavlogiannis. 2017. Optimal Dyck Reachability for Data-Dependence and Alias Analysis. *Proc. ACM Program. Lang.* 2, POPL, Article 30 (Dec. 2017), 30 pages. https://doi.org/10.1145/3158118

[5] Swarat Chaudhuri. 2008. Subcubic Algorithms for Recursive State Machines. In *Proceedings of the 35th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (San Francisco, California, USA) *(POPL '08)*. Association for Computing Machinery, New York, NY, USA, 159–169. https://doi.org/10.1145/1328438.1328460

[6] Lillian Lee. 2002. Fast Context-free Grammar Parsing Requires Fast Boolean Matrix Multiplication. *J. ACM* 49, 1 (Jan. 2002), 1–15. https://doi.org/10.1145/505241.505242

[7] Leslie G. Valiant. 1975. General Context-free Recognition in Less Than Cubic Time. *J. Comput. Syst. Sci.* 10, 2 (April 1975), 308–315. https://doi.org/10.1016/S0022-0000(75)80046-8

[8] Mihalis Yannakakis. 1990. Graph-theoretic methods in database theory. In *Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*. ACM, 230–242.

[9] Qirun Zhang, Xiao Xiao, Charles Zhang, Hao Yuan, and Zhendong Su. 2014. Efficient Subcubic Alias Analysis for C. *SIGPLAN Not.* 49, 10 (Oct. 2014), 829–845. https://doi.org/10.1145/2714064.2660213