

# Context-Free Path Querying via Matrix Equations



#### Yuliya Susanina

 $JetBrains\ Research,\ Saint\ Petersburg\ University,\ Russia\\ jsusanina@gmail.com$ 

# Context-Free Path Querying

CFPQ—a way to specify path constraints in terms of context-free grammars—is gaining popularity in many areas, such as bioinformatics, graph databases or static code analysis. It is crucial to develop highly efficient algorithms for CFPQ since the size of the input data is typically large. We propose a new way to reduce CFPQ to a problem with available high-performance solutions: CFPQ can be reduced to solving the systems of equations over real numbers  $\mathbb{R}$ . So, we can use numerical linear algebra and computational mathematics to improve the performance of query evaluation.

# Equation-Based Approach

We reduce CFPQ to solving the equations similarly to the way Sato reduces Datalog program evaluation to linear algebra [1].

Reduction from solving Boolean matrix equations

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$T_S^* = \lim_{k \to \infty} T_S^k = \lim_{k \to \infty} A T_S^{k-1} B + A B = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

 $T_S^*$ — the least solution of  $T_S = AT_SB + AB$ 

Reduction from solving matrix equations over  $\mathbb R$ 

$$0 < \epsilon \le \frac{1}{1 + \|T_B^T \otimes T_A\|} \Rightarrow \mathcal{T}_S = \epsilon (A\mathcal{T}_S B + AB)$$
  
has a unique solution  $\mathcal{T}_S^*$ 

$$(\mathcal{T}_S^*)_{ij} > 0 \Leftrightarrow (\mathcal{T}_S^*)_{ij} = 1$$

In real-world cases, we deal with the systems of matrix equations because grammars contain more than one nonterminal. There are different methods to solve these systems:

- Naive iterative process
- Methods for linear systems of form Ax = b
- Efficient algorithms for special cases (for example, Sylvester equations)
- Approximate methods (Newton's method and its modifications)

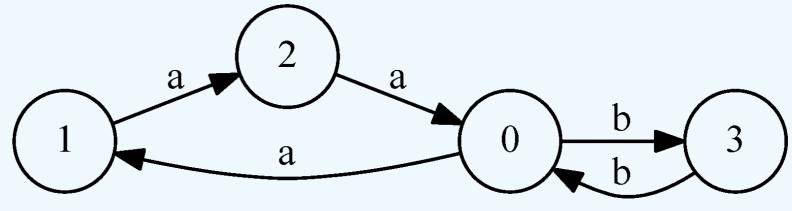
# Acknowledgments

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



## Example

Labeled directed graph D:



CF-grammar G for the language  $\mathcal{L}(G_S) = \{a^n b^n\}$ :  $S \to A S B \mid A B; A \to a; B \to b$ 

The result of CFPQ is Boolean matrix  $T_S$ :  $(T_S)_{ij} = 1 \Leftrightarrow \exists$  path from i to j in Dand its labeling is in  $\mathcal{L}(G_S)$ 

#### Evaluation

Query:  $S \to A S B \mid B; A \to a; B \to b$ Graphs: classical ontologies (RDFs) CPU-based implementations using *scipy* library: [sSLV] solve equation as sparse linear systems [dNWT] find roots of  $F(T_S) = \mathcal{T}_S - \epsilon(A\mathcal{T}_S B + B) = 0$ 

Comparative analysis with matrix-based approach [2]

Ontology		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

#### Results

- Equation-based approach for CFPQ was proposed
- The feasibility of using both accurate and approximate methods of computational mathematics was assessed
- The evaluation on a set of conventional benchmarks showed that it is comparable with the matrix-based approach and applicable for real-world data processing

#### Future Research

- Employ high-performance solvers which utilize sparse matrices, GPGPU and distributed computations
- Determine the subclasses of polynomial equations the solution of which can be reduced to CFPQ and try to construct a bidirectional reduction between them, thereby finding efficient solutions for both problems

### References

- [1] TAISUKE SATO. A linear algebraic approach to datalog evaluation. Theory and Practice of Logic Programming, 17(3):244–265, May 2017.
- [2] Rustam Azimov and Semyon Grigorev. Context-free path querying by matrix multiplication. In *Proceedings of the 1st ACM SIG-MOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, GRADES-NDA '18, pages 5:1–5:10, New York, NY, USA, 2018. ACM.