Super Duper Dataset for Experimental Analysis of CFPQ Algorithms

Author 1*
Author 2*
author_1@samplemail.com
author_2@samplemail.com
institution
city, state, country

Author 3
author_3@samplemail.com
institution
city, state, country

АННОТАЦИЯ

Recently, there has been an increasing interest in solving problems related to context-free path queries (CFPQ) on graphs. However, the development of meaningful benchmark datasets and standardized evaluation procedures is lagging, consequently hindering advancements in this area. To solve this, we introduce the CFPQ_Data dataset, which contains the most popular graphs for experimental analysis of CFPQ algorithms. The collection consists of over 40 graphs of varying sizes. We provide Python-based data loaders and implementations of the well-known CFPQ algorithms. Here, we give an overview of the CFPQ_Data dataset, standardized evaluation procedures, and provide baseline experiments. All datasets are available at https://github.com/JetBrains-Research/CFPQ_Data. The experiments are fully reproducible from the code available at https://github.com/JetBrains-Research/CFPQ_PyAlgo.

ACM Reference Format:
Author 1, Author 2, and Author 3. 2018. Super Duper Dataset for Experimental Analysis of CFPQ Algorithms. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Фокус статьи. Во введении важно ввести читателя в курс дела, объяснить что такое CFPQ, рассказать про известные работы. Рассказать, что в каждой статье был свой кустарный подход к экспериментальному анализу алгоритмов и зачастую из-за этого не получается сопоставить алгоритмы по результатам.

Параграф первый - что такое CFPQ. Нужно переписать другими словами.

Formal language constrained path querying, or formal language constrained path problem [3], is a graph analysis problem in which formal languages are used as constraints for navigational path queries. In this approach, a path is viewed as a word constructed by concatenation of edge labels. Paths of interest are constrained with some formal language: a query should find only paths labeled by words from the language. The class of language constraints which is most widely spread is regular: it is used in various graph query languages and engines. Context-free path querying (CFPQ) [17], while being more expressive, is still at the early stage of development. Context-free constraints allow one to express such important class of queries as *same-generation queries* [1] which cannot be expressed in terms of regular constraints.

Параграф второй - работы в области CFPQ.

Several algorithms for CFPQ based on such parsing techniques as (G)LL, (G)LR, and CYK were proposed recently [4–6, 8, 10, 12, 13, 16, 18]. Yet recent research by Jochem Kuijpers et al. [9] shows that existing solutions are not applicable for real-world graph analysis because of significant running time and memory consumption. At the same time, Nikita Mishin et al. show in [11] that the matrix-based CFPQ algorithm demonstrates good performance on real-world data. A matrix-based algorithm proposed by Rustam Azimov [2] offloads the most critical computations onto Boolean matrices multiplication.

All discussed matrix-based algorithms correspond to the CFPQ with relational query semantics and solve the reachability problem (according to Hellings [7]). However, in some areas, it is important to have a proof of existence of certain paths. This problem can be solved using CFPQ algorithms with single-path query semantics (according to Hellings [8]), which provide some path for each node pair if one exists. There are many results on the CFPQ with

*Both authors contributed equally to this research.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . . $15.00
https://doi.org/10.1145/1122445.1122456

single-path query semantics which use the shortest paths to return [3, 4, 8, 15].

Параграф третий - почему нужно стандартизировать экспериментальное исследование.

Due to the wide applicability of context-free path queries, the need to measure the performance of algorithms that implement these queries becomes critical. In order to show the applicability of the CFPQ algorithm in practice, it becomes necessary to conduct an experimental analysis on data simulating real scenarios. It also allows researchers to compare the performance of their proposed solution with existing ones. However, finding and preparing the necessary data for conducting an experimental study can cause many different problems. One solution to these problems in many areas of research is the use of a standardized dataset. Unfortunately, in the field of CFPQ algorithms, at the moment in most works, even recent ones, experiments are carried out on a fixed set of small-scale, non-diverse graphs, using non-standardized experimental protocols and baselines, which makes it difficult to compare results from different publications.

1.1 Present work

Поверхностный обзор того, что есть в CFPQ_Data. The collection consists of over 40 graphs of varying sizes. All graphs are presented in standard RDF format on the collection site https://github.com/JetBrains-Research/CFPQ_Data.

RDF. Где-то нужно написать почему именно RDF используется.

The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications [14] originally designed as a metadata data model. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed. RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a "triple"). Using this simple model, it allows structured and semi-structured data to be mixed, exposed, and shared across different applications. This linking structure forms a directed, labeled graph, where the edges represent the named link between two resources, represented by the graph nodes. This graph view is the easiest possible mental model for RDF and is often used in easy-to-understand visual explanations. We provide Python-based data loaders and implementations of the well-known CFPQ algorithms. We also give standardized evaluation procedures, and provide baseline experiments. Moreover, we report results on an experimental study comparing well-known algorithms in CFPQ area.

1.2 Related work

Обзор существующих работ. И того, как в этих работах проводились эксперименты.

Как проводились эксперименты by Xiaowang Zhang in "Context-Free Path Queries on RDF Graphs-[18]. Как проводились эксперименты by Jochem Kuijpers et al. [9]. Как проводились эксперименты by Nikita Mishin et al. show in [11]. Как проводились эксперименты by Rustam Azimov [2]. The paper measures the performance of the algorithm in isolation while J. Kuijpers provides the evaluation of the algorithms which are integrated with Neo4j[1] graph database.

Указать на то, что они все проводились на разных данных и физически не сравнить алгоритмы между собой (если это так). Указать на то, что все данные для экспериментов в каждой из статей были сфокусированы только на одной области (если это так).

2 THE CFPQ_DATA COLLECTION

Краткое описание того, что собрано. Коллекция состоит из более чем 40 графов разного размера. Также мы предоставляем загрузчики данных и реализации наиболее популярных алгоритмов на основе Python, а также стандарт проведения экспериментов и базовых показателей работы алгоритма. **Возможно сказать про листинг с примером кода.** Например выложить на сайт питоновский ноутбук с примером применения. Подробное описание каждого графа и другую документацию вы сможете найти на сайте коллекции.

2.1 Graphs and grammars

Описываем коллекцию. Наша коллекция наборов данных охватывает графы из разных областей, предоставленные разными авторами. Здесь мы даем обзор некоторых репрезентативных областей и моделей графов.

RDF. Рассказать откуда взялись RDF. Small graphs is a set of popular semantic web ontologies. This set is introduced by Xiaowang Zhang in "Context-Free Path Queries on RDF Graphs".

MemoryAliases. Рассказать откуда взялись MemoryAliases. Потом еще про MemoryAliases что-то вроде MemoryAliases — real-world data for points-to analysis of C code. First part is a dataset form Graspan tool. The original data is placed here. This part is placed in Graspan folder. Second part is a part of dataset form "Demand-driven alias analysis for C". This part is placed in small folder.

LUBM. Рассказать почему отдельно выделили LUBM. LUBM - the Lehigh University Benchmark graphs.

[1]Neo4j graph database web page: https://neo4j.com/. Access date: 15-March-2021.

Synthetic. Рассказать про важность синтетических графов. WorstCase — graphs with two cylces; the query Brackets is a grammar for the language of correct bracket sequences. SparseGraph — graphs generated with NetworkX to emulate sparse data. The grammar provided is a variant of the same-generation query. ScaleFree — graphs generated by using the Barab'asi-Albert model of scale-free networks. Use with grammar ** an_bm_cm_dn**, which is a query for AnBmCmDn language. FullGraph — cycle graphs, all edges are labeled with the same token. Use with A_star queries, which produce full graph on that dataset.

2.2 Baselines methods (GraphBuilders, GraphLoaders GraphSavers)

Здесь описываем как реализовали загрузку и работу с графами и почему так. Любой граф их датасета можно либо построить (если он синтетический) либо загрузить с сайта коллекции (или по пути к имеющемуся в системе графу). А с помощью GraphSaver'а можно любой граф сохранить в нужном исследователю виде. **Возможно стоит упомянуть про грамматики.**

2.3 Evaluation methods (Evaluators)

Тут надо описать каким образом происходит запуск алгоритма на графе и грамматике из коллекции. Возможно описать как это выглядит технически. А также как можно написать свой алгоритм в имеющейся инфраструктуре.

3 EXPERIMENTAL EVALUATION

Наша цель здесь это представить некоторый стандарт того, как можно проводить эксперименты. Стоит описать, стандартный путь проведения эксперимента и классические базовые показатели извлекаемые из работы алгоритма.

3.1 Datasets

Здесь нужно описать какие графы мы взяли для проведения СВОЕГО эксперимента. Мы взяли RDF потому что это классика. Взяли MemoryAliases потому что они большие. Взяли WorstCase потому что это важный случай в теории.

3.2 Algorithms

Здесь нужно описать какие мы алгоритмы использовали в СВОЕМ эксперименте. Думаю достаточно взять пару матричных алгоритмов и тензорный.

Таблица 1: Таблица с результатами эксперимента

Graph	Grammar	Algorithm	Time
RDF	G_1	Tensor	0.001
RDF	G_1	Tensor	0.001
RDF	G_1	Tensor	0.001
RDF	G_1	Tensor	0.001

3.3 Results and discussion

Здесь мы суммируем полученные результаты. Както описываем полученные результаты. Возможно сравниваем их с полученными в других статьях. Можно написать насколько это было просто провести эксперимент в имеющейся инфраструктуре. Если код эксперимента маленький, то можно его вставить.

4 CONCLUSION

Описываем то, к чему пришла статья. Собрали графы, грамматики, сделали удобную инфраструктуру для проведения экспериментов. Улучшили сопоставляемость результатов экспериментов. Мы сделали обзор коллекции CFPQ_Data и сообщили о результатах экспериментального исследования, сравнивающего наиболее популярные алгоритмы на подмножестве данных. Мы считаем, что наша коллекция наборов данных будет способствовать дальнейшему прогрессу в обучении представлению графов, и что наши унифицированные процедуры оценки улучшат сопоставимость результатов. Мы с нетерпением ждем добавления новых наборов данных и рады вкладу сообщества, исследователей и практиков из других областей. Дальнейшая работа включает более обширное сравнение существующих алгоритмов и добавление новых графов.

5 ACKNOWLEDGMENTS

Мы очень благодарны всем, кто принимал участие в этой нелегкой работе.

СПИСОК ЛИТЕРАТУРЫ

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. 1995. *Foundations of Databases: The Logical Level* (1st ed.). Addison-Wesley Longman Publishing Co., Inc., USA.

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

[3] Chris Barrett, Riko Jacob, and Madhav Marathe. 2000. Formal-Language-Constrained Path Problems. *SIAM J. Comput.* 30, 3 (May 2000), 809–837. https://doi.org/10.1137/S0097539798337716

[4] Phillip G. Bradford. 2007. Quickest Path Distances on Context-Free Labeled Graphs. In *Proceedings of the 6th WSEAS International Conference on Information Security and Privacy* (Tenerife, Spain) *(ISP'07)*. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 22–29.

[5] Phillip G Bradford and Venkatesh Choppella. 2016. Fast point-to-point Dyck constrained shortest paths on a DAG. In *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 1–7.

[6] Semyon Grigorev and Anastasiya Ragozina. 2017. Context-free Path Querying with Structural Representation of Result. In *Proceedings of the 13th Central & Eastern European Software Engineering Conference in Russia* (St. Petersburg, Russia) *(CEE-SECR '17)*. ACM, New York, NY, USA, Article 10, 7 pages. https://doi.org/10.1145/3166094.3166104

[7] Jelle Hellings. 2014. Conjunctive context-free path queries. In *Proceedings of ICDT'14*. 119–130.

[8] Jelle Hellings. 2015. Querying for Paths in Graphs using Context-Free Path Queries. *arXiv preprint arXiv:1502.02242* (2015).

[9] 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* (Santa Cruz, CA, USA) *(SSDBM '19)*. ACM, New York, NY, USA, 121–132. https://doi.org/10.1145/3335783.3335791

[10] 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* (Pau, France) *(SAC '18)*. ACM, New York, NY, USA, 1230–1237. https://doi.org/10.1145/3167132.3167265

[11] 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)* (Amsterdam, Netherlands) *(GRADES-NDA'19)*. ACM, New York, NY, USA, Article 12, 5 pages. https://doi.org/10.1145/3327964.3328503

[12] 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.

[13] 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* (St. Louis, MO, USA) *(Scala 2018)*. ACM, New York, NY, USA, 13–23. https://doi.org/10.1145/3241653.3241655

[14] The World Wide Web Consortium (W3C). [n.d.]. SEMANTIC WEB. https://www.w3.org/standards/semanticweb/. [Online; accessed 11-December-2008].

[15] Charles B. Ward and Nathan M. Wiegand. 2010. Complexity Results on Labeled Shortest Path Problems from Wireless Routing Metrics. *Comput. Netw.* 54, 2 (Feb. 2010), 208–217. https://doi.org/10.1016/j.comnet.2009.04.012

[16] Charles B. Ward, Nathan M. Wiegand, and Phillip G. Bradford. 2008. A Distributed Context-Free Language Constrained Shortest Path Algorithm. In *Proceedings of the 2008 37th International Conference on Parallel Processing (ICPP '08)*. IEEE Computer Society, Washington, DC, USA, 373–380. https://doi.org/10.1109/ICPP.2008.67

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

[18] 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.