Parsing techniques for graph analysis

Semyon Grigorev
Saint Petersburg State University
7/9 Universitetskaya nab.
St. Petersburg, 199034 Russia
Semen.Grigorev@jetbrains.com

Ekaterina Verbitskaia
Saint Petersburg State University
7/9 Universitetskaya nab.
St. Petersburg, 199034 Russia
kajigor@gmail.com

Nowadays input data for parsing algorithms are not limited to be linear strings, and context-free grammars are used not only for programming languages specification. One classical example is context-free path querying for graph data bases where an input is a graph and path constraints are specified by a grammar. Graph parsing may find an application in different areas: in software engineering for dynamically generated strings analysis, in graph data bases for paths querying, etc. The idea of multiple input GLL parsing, presented at Parsing@SLE-2016 by Elizabeth Scott and Adrian Johnstone, is also a particular case of graph parsing: a set of token-with-extent can be treated as a directed graph where extents are vertices and tokens label the edges. Thus, graph parsing can be considered as a great connection (!!!!!!) of multiple computer science areas: formal languages theory, parsing algorithms, data bases, graph theory.

Our group is working on several questions posed in this area [4, 11] which still do not have satisfying solutions. Our efforts are mostly aimed at improving performance, lifting up limitations on an input and finding new fields of application for graph parsing.

We already developed several graph parsing algorithms (??? and applied them to different problems ???). First of all, we created a RNGLR-based algorithm and applied it to the analysis of dynamically generated SQL queries [6]. GLL-based context-free path querying algorithm [3], implemented by the authors, runs faster than the solution presented at ISWC-2016 [7]. Our algorithm based on matrix multiplication [1] allows one to utilize GPGPU for graph processing, and it is faster than the GLL-based, but it does not construct a parsing forest.

Currently, we are working on an extension for Meerkat [12] library which allows one to use parser-combinators for graph parsing and integrates context-free querying into the programming language with no need to use designated DSLs. Another direction of work is extending matrix-based algorithm with the support for conjunctive grammars [8]. This will make it possible to execute more complex queries which can be utilized for pseudoknots finding. By mechanization of the GLL-based algorithm in Coq and proving its correctness, we hope to build a foundation for formal reasoning about the extensions under development.

We are also working on some ideas of graph parsing applications. One of the most interesting areas is bioinformatics and problem of context-free pattern search in metagenomical assemblies: assembly may be presented as a graph, and secondary structure of some sequences can be specified in terms of grammar. Moreover, some structures in biological sequences, for example pseudoknots, require conjunctive grammars for structure description, which make bioinformatics interesting area for application.

All existing applications seem to be special cases of the Bar-Hillel [2] theorem for context-free and regular language intersection, and can be generalized, but today many of them are developed as stand alone solutions. Thus, the one goal of our work is to create an abstract framework for parsing based on generalization of GLL parsing algorithm [5] proposed by Elizabeth Scott and Adrian Johnstone. On the other hand we want to adopt advanced matrix multiplication techniques, such as approximated matrix multiplication, sparse matrix multiplication, for graph parsing. We hope to get more effective algorithms for huge graphs processing. Also we want to apply matrix-based algorithm for boolean grammars [8]. It is possible for linear input, but problem is undecidable for graphs: even for conjunctive grammars we get approximation of result. Additional problem with boolean grammar is that parsing with it is not monotonic, and it prevent naive using of solution for conjunctive grammars. Another research direction is an effective algorithms intersection of other types, and finding of other types of grammars. One of possible start point is non-recursive context-free grammars intersection [9, 10] which can be used in speech recognition or for compressed strings processing. We also want to investigate practical areas of application and to create solutions based on our framework to demonstrate its practical value.

1. REFERENCES

[1] Azimov, Rustam, and Semyon Grigorev. “Graph Parsing by Matrix Multiplication.” *arXiv preprint arXiv:1707.01007* (2017).

[2] Bar-Hillel, Yehoshua, Micha Perles, and Eliahu Shamir. “On formal properties of simple phrase structure grammars.” *Sprachtypologie und Universalienforschung* 14 (1961): 143-172.

[3] Grigorev, Semyon, and Anastasiya Ragozina. “Context-Free Path Querying with Structural Representation of Result.” *arXiv preprint arXiv:1612.08872* (2016).

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

[5] Scott, Elizabeth, and Adrian Johnstone. “GLL parsing.”, *Electronic Notes in Theoretical Computer Science*, 253.7 (2010): 177–189.

[6] Verbitskaia, Ekaterina, Semyon Grigorev, and Dmitry Avdyukhin. "Relaxed Parsing of Regular Approximations of String-Embedded Languages." *International Andrei Ershov Memorial Conference on Perspectives of System Informatics.* Springer International Publishing, 2015.

[7] Zhang, Xiaowang, et al. "Context-free path queries on RDF graphs." *International Semantic Web Conference.* Springer International Publishing, 2016. 632–648.

[8] Okhotin, Alexander. "Conjunctive and Boolean grammars: the true general case of the context-free grammars." *Computer Science Review* 9 (2013): 27-59.

[9] Nederhof, Mark-Jan, and Giorgio Satta. "Parsing non-recursive context-free grammars." *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics.* Association for Computational Linguistics, 2002.

[10] Nederhof, Mark-Jan, and Giorgio Satta. "The language intersection problem for non-recursive context-free grammars." *Information and Computation* 192.2 (2004): 172-184.

[11] Yannakakis, Mihalis. "Graph-theoretic methods in database theory." *Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems.* ACM, 1990.

[12] Izmaylova, Anastasia, Ali Afroozeh, and Tijs van der Storm. "Practical, general parser combinators." *Proceedings of the 2016 ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation.* ACM, 2016.