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 of classical examples is a context-free path querying for graph data bases where input is a graph and path constraints are specified by a grammar. Graph parsing may be applied in different areas, in software engineering for dynamically generated strings analysis, graph data bases for paths querying, etc. For example, the idea of multiple input GLL parsing which was presented at Parsing@SLE-2016 by Elizabeth Scott and Adrian Johnstone, is an partial case of graph parsing: set of token-with-extent can be treated as directed graph where extents are vertices and tokens are labels of edges. So, we think that graph parsing is a great connection of different areas: formal languages, parsing algorithms, data bases, graph theory, etc.

There are some open questions in this area [4, 11]. Of course, there are a number of solutions, but many of questions are still open, and existing solutions have different problems with performance, restrictions on input, and other. We are working on these problems. Also we want to find new fields for graph parsing application.

Our current results are some graph parsing algorithms, based on different parsing algorithms. We create RNGLR-based algorithm and apply it for dynamically generated SQL queries [6]. GLL-based context-free path querying algorithm [3] implemented by the authors is faster than solution which was presented at ISWC-2016 [7]. Our algorithm which is based on matrix multiplication [1] allow one to utilize GPGPU for graph processing, and it is faster than GLL-based, but can not build forest.

Currently we are working on extension for Meerkat [12] which allow one to use parser-combinators for graph parsing, and to integrate context-free querying in your favorite language without special DSLs. Also we are working on matrix-based algorithm extension for conjunctive grammars [8] support. It should allow one to perform more complex queries (for examples for psewdoknots finding). Another part of work is mechanization of GLL-based algorithm in Coq. GLL-based algorithm looks pretty simple for implementation and for different extensions, but complex for formal reasoning about correctness and other properties. We want to get formal proof of correctness of current algorithm and create base for formal reasoning about different complex extensions which we plan to do.

Also we are working on some ideas of graph parsing applications. One of the most interesting area 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.