Graph Parsing Application for Bioinformatics Problems

Grammar-based approach for graph structured data analysis

Semyon Grigorev[1], Artem Gorokhov[1], Rustam Azimov[1]

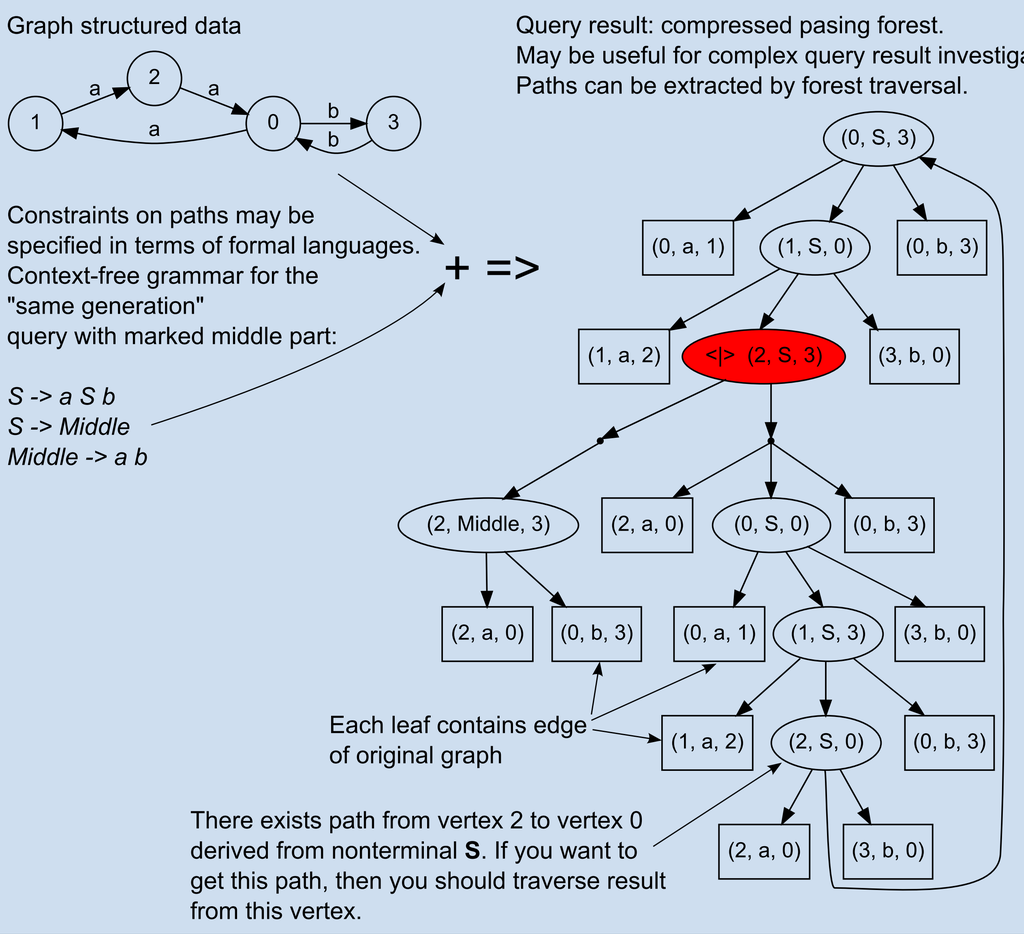
[1] *Saint Petersburg State University, JetBrains, St. Petersburg, Russia*

E-mail: *semen.grigorev@jetbrains.com*

Motivation

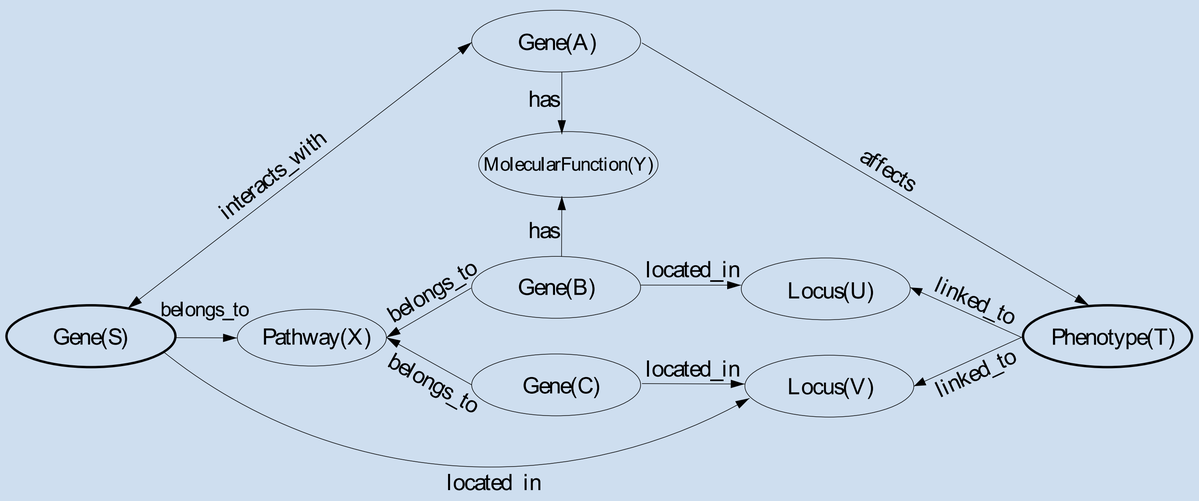
Biomedical databases contain huge amounts of rich data which can be represented as a labelled graph. In order to investigate such data, it may be useful to extract connections with specific constraints. One natural way to provide constraints is to specify the language of paths labels which can be done by using of grammars. For example, one can use context-free grammars with productions $\{S \to aSb; S \to \varepsilon\}$ to query paths which labels should take form of $ab; aabb; aaabbb; ...$, or, generally, should belong to the language $L = \{a^n b^n, n \geq 0\}$. This approach is named *context-free path querying* (or graph parsing) and can be applied to some problems in bioinformatics: metagenomic assemblies analysis, graph data base querying.

Context-free path querying



Data base querying

One of the examples is an analysis of graphs where vertices correspond to entities and concepts such as gene or phenotype while edges represent known relationships such as “codes for”, “interacts with”, etc.
Eample of graph structured data (from [4]):



Querying paths with special constraints may shed light upon unknown before links between vertices, forming the basis for new hypotheses.

References

[1] Semyon Grigorev and Anastasiya Ragozina. Context-free path querying with structural representation of result. *arXiv preprint arXiv:1612.08872*, 2016.

[2] Ekaterina Verbitskaia, Semyon Grigorev, and Dmitry Avdyukhin. Relaxed parsing of regular approximations of string-embedded languages. In *International Andrei Ershov Memorial Conference on Perspectives of System Informatics*, pages 291–302. Springer, 2015.

[3] Rustam Azimov and Semyon Grigorev. Graph parsing by matrix multiplication. *arXiv preprint arXiv:1707.01007*, 2017.

[4] Petteri Sevon and Lauri Eronen. Subgraph queries by context-free grammars. *Journal of Integrative Bioinformatics (JIB)*, 5(2):157–172, 2008.

Results

- We propose some graph parsing algorithms [1, 2, 3].
- We solve some problems of existing approaches (such as cycles processing problem in [4]).
- Our solution provide an ability to use GPGPU and multi-core systems for graph parsing which can be useful for huge biological data analysis.

Context-free querying algorithms performance comparison

Graph	#edges	#results	GLL(ms)	GPGPU(ms)
g_1	8688	141072	1926	82
g_2	14712	532576	6246	185
g_3	15840	449560	7014	127

Future Research

- Currently we are working on long subsequences of 16s rRNA reconstruction from metagenomic assembly and on entities connections detection.
- We want to find applications for context-free graph querying techniques and implement required tools.

Metagenomic assemblies analysis

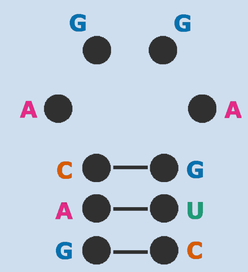
Metagenomic assemblies can be presented as graph structured data. Some sequences have specific secondary structure, which can be described in terms of context-free grammar, and this grammar can be used for searching and classification.

Context-free grammar for stem:

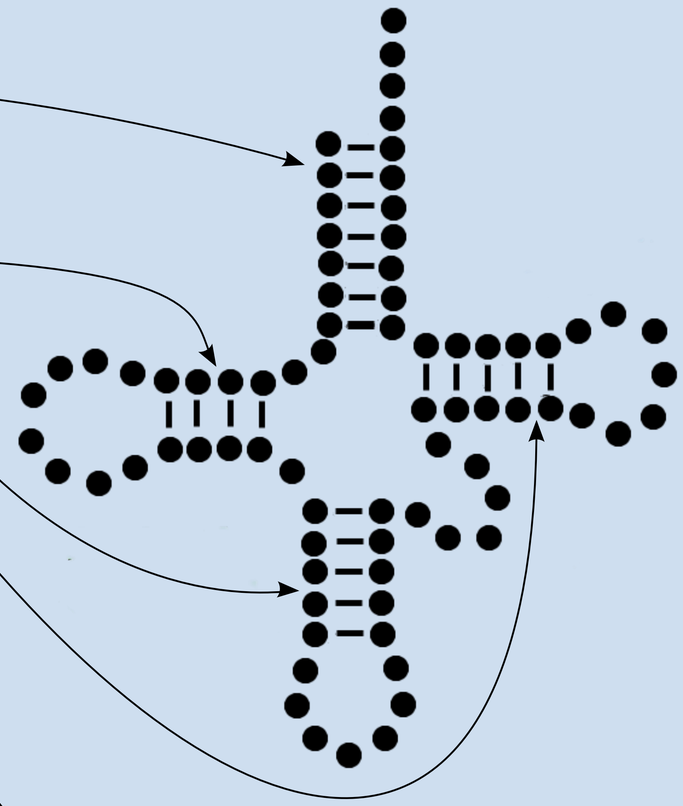
stem<s> ->
A stem<s> U
| U stem<s> A
| G stem<s> C
| C stem<s> G
| s

Arbitrary string with limited length: ***any*[i..j]***, where ***i*** is a lower bound of length and ***j*** is a upper bound.

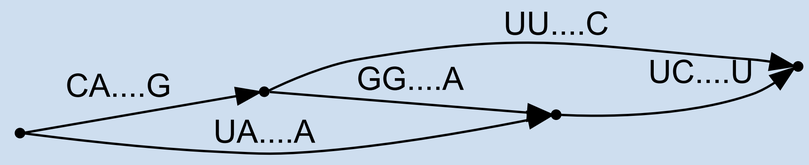
stem<A G G A>



stem<(
any[i1..j1]*
stem<any[i2..j2]>*
any[i3..j3]*
stem<any[i4..j4]>*
any[i5..j5]*
stem<any[i6..j6]>*
any[i7..j7]*
)>
any[i8..j8]*



Metagenomic assembly is a directed graph:



Acknowledgments

This work is supported by grant from JetBrains Research.

Information

All materials is available on GitHub: `https://github.com/YaccConstructor`