

Generalized LL parsing for context-free constrained path search problem

Semyon Grigorev
rsdpisuy@gmail.com
Saint Petersburg State University, Russia

Path querying is an actual problem in bioinformatic, graph databases, ... One of specific problem is formal languages path problem [1] which means that paths constraints formulated as Query may be specified as context-free grammar: path $P = e_0, \dots, e_n$, $\omega = e_0.tag \dots e_n.tag$, $\omega \in L(G)$

Let we want to find all path with form $A^n B^n$ where $n \geq 1$: $\{AB; AAB; AAAB; \dots\}$. This constraint can not be specified with regular language as far as $L = \{a^n b^n; n \geq 1\}$ is not regular but context free. Required language can be specified by grammar G presented in picture 1. Here s and $middle$ are nonterminals and A and B are terminals and we means that edges of input graph tagged by them.

$s: A s B \mid middle$
 $middle: A B$

Figure 1: Grammar G for language $L = \{a^n b^n; n \geq 1\}$

We propose a context-free language constrained path problem solution which allow to find all paths and construct implicit representation of result.

Our solution is based on generalized LL (GLL) [2] parsing algorithm which allow to process arbitrary context-free grammars. Complexity is $O(n^3)$ in worst case and linear for unambigues grammars, that better then complexity of CYK and Erly which used as base in other solutions.

All-path semantic — SPPF constructed by algorithm contains all paths matched with specified constraints. Let we . And we want to find all paths in graph M (presented in picture 2) matched this query. SPPF for grammar $G = \{N, \Sigma, P\}$ and graph $M = (V, E); V \in; E \in N \Sigma N$ is presented in picture 3. Extensions allow to check whether path from u to v exists and extract it. For example

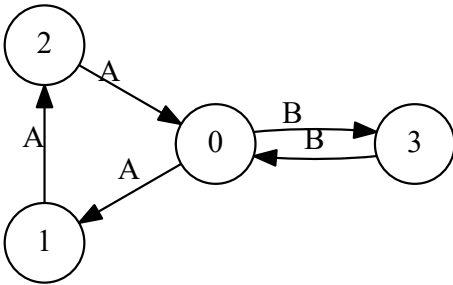


Figure 2: Input graph M

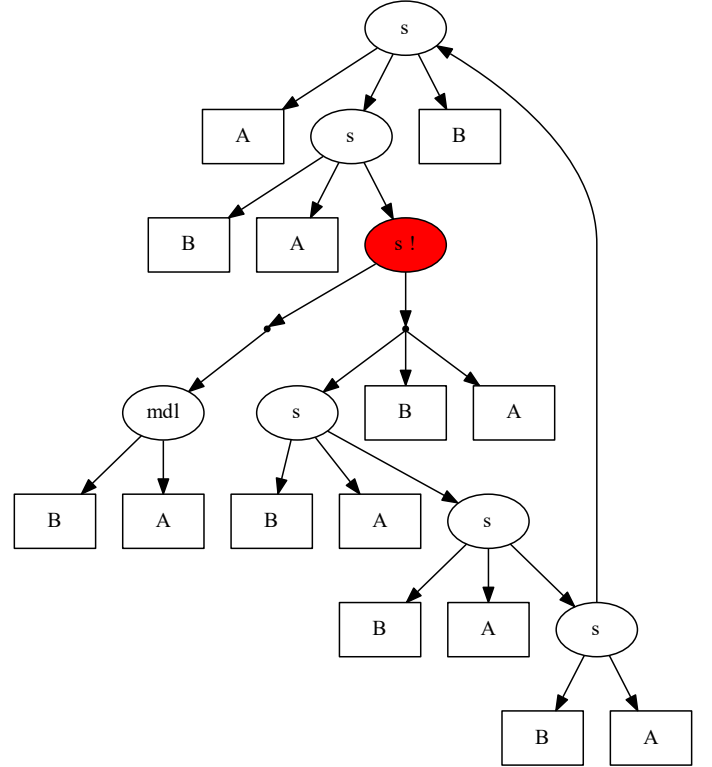


Figure 3: Result SPPF for input graph M (pic. 2) and query G (pic. 1)

SPPF contains infinite set of paths (cycles in SPPF). Also its represent a structure of pats: 'middle' of any path can be found simply. It may be useful not only for results understanding but also for query debugging.

1. REFERENCES

- [1] Miller, J. A., Ramaswamy, L., Kochut, K. J., & Fard, A. (2015, June). Research Directions for Big Data Graph Analytics. In 2015 IEEE International Congress on Big Data (pp. 785–794). IEEE.
- [2] Scott, E., & Johnstone, A. (2010). GLL parsing. Electronic Notes in Theoretical Computer Science, 253(7), 177–189.
- [3] Hellings, J. (2014). Conjunctive context-free path queries.

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