TITLE

A1

Saint Petersburg State University
St. Petersburg, Russia

A2

ITMO University
St. Petersburg, Russia

Rustam Azimov
rustam.azimov19021995@gmail.com
Saint Petersburg State University
St. Petersburg, Russia
JetBrains Research
St. Petersburg, Russia

Semyon Grigorev
s.v.grigoriev@spbu.ru
semyon.grigorev@jetbrains.com
Saint Petersburg State University
St. Petersburg, Russia
JetBrains Research
St. Petersburg, Russia

ABSTRACT

CCS CONCEPTS

• **Information systems** → **Query languages for non-relational engines**; • **Theory of computation** → **Grammars and context-free languages**; *Parallel computing models*; • **Computing methodologies** → **Massively parallel algorithms**; • **Computer systems organization** → *Single instruction, multiple data*;

ACM Reference Format:
A1, A2, Rustam Azimov, and Semyon Grigorev. 2021. TITLE. In . ACM, New York, NY, USA, 2 pages.

1 INTRODUCTION

CFPQ as a separated algorithms.

Integration with graph DB.

Integration with query languages. The problem. We cannon separate regular and context-free queryes.

Contribution

(1) New algorithm. Correctness and time complexity.
(2) The way to optimize queries.
(3) Evaluation.

SIGMOD'21, ,

ACM ISBN XXX-X-XXXXX-XXX-X…$15.00

2 CONTEXT-FREE PATH QUERYING BY KRONECKER PRODUCT

2.1 The algorithm

Lemma 2.1. *Let $\mathcal{G} = (V, E, L)$ be a graph and $G = (\Sigma, N, P)$ be a grammar. Let $\mathcal{G}_k = (V, E_k, L \cup N)$ be graph and M_k its adjacency matrix of the execution some iteration $k \geq 0$ of the algorithm* **??**. *Then for each edge $e = (m, S, n) \in E_k$, where $S \in N$, the following statement holds: $\exists m \pi n : S \rightarrow_G l(\pi)$.*

Proof. (Proof by induction)

Basis: For $k = 0$ and the statement of the lemma holds, since $M_0 = M$, M where is adjacency matrix of the graph G , $L \cap N$ is empty. Non-terminals, which allow to derive ε-word, are also added, since each vertex of the graph is reachable by itself through an ε-transition.

Inductive step: Assume that the statement of the lemma hold for any $k \geq (p - 1)$ and show that it also holds for $k = p$, where $p \geq 1$.

For the algorithm iteration p the Kronecker product K_p and transitive closure C_p are evaluated as described in the algorithm. By the properties of this operations, some edge $e = ((s, x), (f, y))$ exists in the oriented graph, represented by adjacency matrix C_p, if and only if $\exists s \pi_1 f$ in the RSM graph, represented by matrix M_r, and $\exists x \pi y$ in graph, represented by M_{p-1}. Concatenated symbols along the path π_1 form some derivation string, composed from terminals and non-terminals, included in the graph by inductive assumption.

Therefore, if s an f are initial and final states of some box B of the RSM, new edge between vertices x and y with the respective non-terminal S_B will be added to the matrix M_p and this completes the proof of the lemma.

□

Lemma 2.2. *Let $\mathcal{G} = (V, E, L)$ be a graph and $G = (\Sigma, N, P)$ be a grammar. Let $\mathcal{G}_k = (V, E_k, L \cup N)$ be graph and M_k its* *adjacency matrix of the execution some iteration $k \geq 1$ of the algorithm* **??**. *For any path $m\pi n$ in graph $\mathcal{G}$ with word $l = l(\pi) : S \rightarrow_G l$, if $h \leq k$, where h is a derivation tree height, then $\exists e = (m, S, n) : e \in E_k$.*

Proof. (Proof by induction)

Basis: Show that statement of the lemma holds for the $k = 1$. Matrix M and edges of the graph $\mathcal{G}$ contains only labels from L. Since the derivation tree of height $h = 1$ contains only one non-terminal S as a top node and only symbols from Σ as leafs, for all paths, which form a word with derivation tree of the height $h = 1$, the corresponding top nonterminals will be added to the M_1 via algorithm first iteration. Non-terminals, which allow to derive ε-word, are also added via algorithm preprocessing step. Thus, the lemma statement holds for the $k = 1$.

Inductive step: Assume that the statement of the lemma hold for any $k \geq (p-1)$ and show that it also holds for $k = p$, where $p \geq 2$.

For the algorithm iteration p the Kronecker product K_p and transitive closure C_p are evaluated as described in the algorithm. By the properties of this operations, some edge $e = ((s, x), (f, y))$ exists in the oriented graph, represented by adjacency matrix C_p, if and only if $\exists s\pi_1 f$ in the RSM graph, represented by matrix M_r, and $\exists x\pi y$ in graph, represented by M_{p-1}.

Suppose, that exists derivation tree T of height $h = p$ with the top non-terminal S for the path $x\pi y$. The tree T is formed as $S \rightarrow a_1..a_d, d \geq 1$ where $\forall i \in [1..d]$ a_i is sub-tree of height $h_i \leq p-1$ for the sub-path $x_i\pi_i y_i$. By inductive hypothesis, there exists path π_i for each derivation sub-tree, that $x = x_1\pi_1 x_2..x_d\pi_d x_{d+1} = y$ and concatenation of these paths forms $x\pi y$, and the top non-terminals of this sub-trees are included in the matrix M_{p-1}.

Therefore, vertices x_i $\forall i \in [1..d]$ form path in the graph, represented by matrix M_{p-1}, with complete set of labels. Thus, new edge between vertices x and y with the respective non-terminal S will be added to the matrix M_p and this completes the proof of the lemma.

□

Theorem 2.3. *Let $\mathcal{G} = (V, E, L)$ be a graph and $G = (\Sigma, N, P)$ be a grammar. Let $\mathcal{G}_R = (V, E_R, L)$ be a result graph for the execution of the algorithm* **??**. *Then $e = (m, S, n) \in E_R$, where $S \in N$ if and only if $\exists m\pi n : S \rightarrow_G l(\pi)$.*

Proof. Todo. □

Theorem 2.4. *Let $\mathcal{G} = (V, E, L)$ be a graph and $G = (\Sigma, N, P)$ be a grammar. The algorithm* **??** *terminates in finite number of steps.*

Proof. Todo. □

3 EVALUATION

Questions.

(1) Compare classical RPQ algorithms and our agorithm
(2) Compare other CFPQ algorithms and our algorithms
(3) Iveatigate effect of grammar optimization

3.1 RPQ

3.2 CFPQ

Comparison with matrix-based.

On query optimization.

4 CONCLUSION

REFERENCES