International Journal of Foundations of Computer Science

Rational index of bounded-oscillation languages*

Ekaterina Shemetova[†‡§]
katyacyfra@gmail.com

Alexander Okhotin[†]
alexander.okhotin@spbu.ru

Semyon Grigorev[†§]
s.v.grigoriev@spbu.ru

Received (Day Month Year)
Accepted (Day Month Year)

The rational index of a context-free language L is a function $f(n)$, such that for each regular language R recognized by an automaton with n states, the intersection of L and R is either empty or contains a word shorter than $f(n)$. It is known that the context-free language (CFL-)reachability problem and Datalog query evaluation for context-free languages (queries) with the polynomial rational index is in NC, while these problems is P-complete in the general case. We investigate the rational index of bounded-oscillation languages and show that it is of polynomial order. We obtain upper bounds on the values of the rational index for general bounded-oscillation languages and for some of its previously studied subclasses.

Keywords: bounded-oscillation languages; rational index; CFL-reachability; parallel complexity; context-free languages; Datalog programs; context-free path queries.

1. Introduction

The notion of a rational index was introduced by Boasson et al. [5] as a complexity measure for context-free languages. The rational index $\rho_L(n)$ is a function, which denotes the maximum length of the shortest word in $L \cap R$, for arbitrary R recognized by an n-state automaton. The rational index plays an important role in determining the parallel complexity of such practical problems as the context-free language (CFL-)reachability problem and Datalog chain query evaluation.

The CFL-reachability problem for a fixed context-free grammar G is stated as follows: given a directed edge-labeled graph D and a pair of nodes u and v, de-

*This research was supported by the Russian Science Foundation, grant №18-11-00100.
[†]St. Petersburg State University, 7/9 Universitetskaya nab., Saint Petersburg 199034, Russia.
[‡]St. Petersburg Academic University, ul. Khlopina, 8, Saint Petersburg 194021, Russia.
[§]JetBrains Research, Primorskiy prospekt 68-70, Building 1, St. Petersburg, 197374, Russia.

termine whether there is a path from u to v labeled with a string in $L(G)$. That is, CFL-reachability is a kind of graph reachability problem with path constraints given by context-free languages. It is an important problem underlying some fundamental static code analysis like data flow analysis and program slicing [29], alias analysis [8, 36], points-to analysis [22] and other [7, 18, 27], and graph database query evaluation [3, 14, 16, 37].

The Datalog chain query evaluation on a database graph is equivalent to the CFL-reachability problem.

Example 1 (Datalog query as a context-free grammar) *Consider a database D with relation "child". It also can be represented as a digraph G, where each node of the graph corresponds to a person, and edges are labeled with a word "child".*

The following Datalog query determines all pairs of people x and y such that x is a descendant of y:

$Desc(x, y)$:- $Child(x, y)$

$Desc(x, y)$:- $Child(x, z), Desc(z, y)$

This query can be represented as a context-free grammar with the following rules:

$Desc \to Child \mid Child\ Desc$

$Child \to child$

Thus, evaluating the above mentioned Datalog query over database D is equivalent to solving the CFL-reachability problem for a context-free grammar representation of this query and the edge-labeled graph G.

Unlike context-free language recognition, which is in NC (when context-free grammar is fixed), the CFL-reachability problem is P-complete [13, 28, 35]. Practically, it means that there is no efficient parallel algorithm for solving this problem (unless P $\neq$ NC).

The question on the parallel complexity of Datalog chain queries was investigated independently [1, 10, 31]. Ullman and Van Gelder [31] introduce the notion of a *polynomial fringe property* and show that chain queries having this property is in NC. The polynomial fringe property is equivalent to having the polynomial rational index: for a context-free language $L(G)$ having the polynomial rational index $\rho_L(n) = poly(n)$, where $poly(n)$ is some polynomial, is the same as for corresponding query to have the polynomial fringe property. It has been shown that for every algebraic number γ, a language with the rational index in $\Theta(n^\gamma)$ exists [26]. In contrast, the rational index of languages, which generate all context-free languages (an example of such language is the Dyck language on two pairs of parentheses D_2) is in order $exp(\Theta(n^2/\ln n))$ [25], and, hence, this is the upper bound on the value of the rational index for every context-free language.

While both problems is not parallelizable in general, it is useful to develop more efficient parallel solutions for specific subclasses of the context-free languages. For example, there are context-free languages which admit more efficient parallel algo-

rithms in comparison with the general case of context-free recognition [19, 20, 23]. The same holds for the CFL-reachability problem: there are some examples of context-free languages, for which the CFL-reachability problem lies in NL complexity class (for example, linear and one-counter languages) [17, 21, 30, 32]. These languages have the polynomial rational index.

The family of linear languages (linear Datalog programs, respectively) is the well-known subclass of context-free languages having the polynomial rational index [5, 31]. The value of its rational index is in $O(n^2)$ [5]. Linear Datalog programs have been widely studied by deductive database community, because efficient evaluation methods and optimization techniques exist for such programs [2, 24, 31]. For instance, the Datalog program in Example 1 is linear. Many efforts have been devoted to find larger subclasses of context-free languages (Datalog programs) having the polynomial rational indices [1, 2, 5, 9, 30, 31, 32]. Two equivalent classes generalizing linear languages were proposed: piecewise linear programs [9, 31] and the family of quasi-rational languages (the substitution closure of the family of linear languages) [5]; both were independently shown to have the polynomial rational index. Quasi-rational languages are generated by *nonexpansive* grammars. A variable S in a context-free grammar G is expansive if there exists a derivation $S \stackrel{*}{\Rightarrow} uSvSw$ for some words u, v, w. A grammar which contains no expansive variable is said to be nonexpansive. However, it was shown by Afrati et al. that piecewise linear programs is equivalent to linear programs [2]. Therefore, the generalization of linear languages preserving polynomiality of the rational index is remain to be found.

In this work we investigate the rational index of bounded-oscillation languages. Bounded-oscillation languages were introduced by Ganty and Valput [11]. Just like linear languages, it is defined by restriction on the pushdown automata. This restriction is based on the notion of *oscillation*, a special measure of how the stack height varies over time. This class generalizes a family of linear languages.

Our contributions. Our results can be summarized as follows:

- We show that the rational index of bounded-oscillation languages of Ganty and Valput [11] is polynomial and give an upper bound on its value in dependence of the value of oscillation.
- We give upper bounds on the value of rational indices of previously studied subclasses of bounded-oscillation languages: superlinear and ultralinear languages.

2. Preliminaries

Formal languages. A *context-free grammar* is a 4-tuple $G = (\Sigma, N, P, S)$, where Σ is a finite set of alphabet symbols, N is a set of nonterminal symbols, P is a set of production rules and S is a start nonterinal. $L(G)$ is a context-free language generated by context-free grammar G. We use the notation $A \stackrel{*}{\Rightarrow} w$ to denote that the string $w \in \Sigma^*$ can be derived from a nonterminal A by sequence of applying the production rules from P. A *parse tree* is an entity which represents the structure of

the derivation of a terminal string from some nonterminal.

A grammar G is said to be is in the *Chomsky normal form*, if all production rules of P are of the form: $A \to BC$, $A \to a$ or $S \to \varepsilon$, where $A, B, C \in N$ and $a \in \Sigma$.

The set of all context-free languages is identical to the set of languages accepted by pushdown automata (PDA). *Pushdown automaton* is a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$, where Q is a finite set of states, Σ is a input alphabet, Γ is a finite set which is called the stack alphabet, δ is a finite subset of $Q \times (\Sigma \cap \{\varepsilon\}) \times \Gamma \times Q \times \Gamma^*$, $q_0 \in Q$ is the start state, $Z \in \Gamma$ is the initial stack symbol and $F \subseteq Q$ is the set of accepting states.

A *regular language* is a language that can be expressed with a regular expression or a deterministic or non-deterministic finite automata. A *nondeterministic finite automaton* (NFA) is represented by a 5-tuple, $(Q, \Sigma, \delta, Q_0, F)$, where Q is a finite set of states, Σ is a finite set of input symbols, $\delta : Q \times \Sigma \to 2^{|Q|}$ is a transition function, $Q_0 \subseteq Q$ is a set of initial states, $F \subseteq Q$ is a set of accepting (final) states. *Deterministic finite automaton* is a NFA with the following restrictions: each of its transitions is uniquely determined by its source state and input symbol, and reading an input symbol is required for each state transition.

For a language L over an alphabet Σ, its rational index ρ_L is a function defined as follows:

$$\rho_L(n) = \max\{\min\{|w| : w \in L \cap K\}, K \in Rat_n, L \cap K \neq \emptyset\},$$

where $|w|$ is the length of a word w and Rat_n denotes the set of regular languages on an alphabet Σ, recognized by a finite nondeterministic automation with at most n states.

Bounded-oscillation languages. Oscillation is defined using a hierarchy of *harmonics*. Let $\bar{a}$ be a *push*-move and a be a *pop*-move. Then a PDA run r can be described by a well-nested sequence $\alpha(r)$ of $\bar{a}$-s and a-s. Two positions $i < j$ form a *matching pair* if the corresponding $\bar{a}$ at i-th position of the sequence matches with a at j-th position. For example, word $\bar{a}\bar{a}\bar{a}aa\bar{a}aa$ has the following set of matching pairs: $\{(1,8),(2,5),(3,4),(6,7)\}$ $(\bar{a}(\bar{a}(\bar{a}a)a)(\bar{a}a)a)$.

Harmonics are inductively defined as follows:

- order 0 harmonic h_0 is ε
- $h_{(i+1)}$ harmonic is $\bar{a}h_i a\ \bar{a}h_i a$.

PDA run r is *k-oscillating* if the harmonic of order k is the greatest harmonic that occurs in r after removing 0 or more matching pairs. *Bounded-oscillation languages* are languages accepted by pushdown automata with all runs k-oscillating. It is important that the problem whether a given CFL is a bounded-oscillation language is undecidable [11].

Example 2. *Consider Figure 1. It shows how the stack height changes during the*

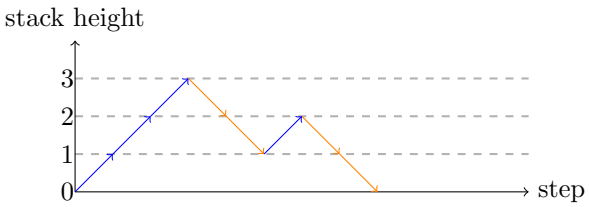

Fig. 1. Stack heights during the run of PDA.

run of a PDA. Corresponding well-nested word $\alpha(r)$ *is* $\bar{a}\bar{a}\bar{a}aa\bar{a}aa$*. The greatest harmonic in this word is order 1 harmonic (moves forming harmonic are marked in bold, removed matching pairs are* $(1, 8)$ *and* $(2, 5)$*):* $\bar{a}\bar{a}\mathbf{\bar{a}a}a\mathbf{\bar{a}a}a$*, therefore oscillation of the run* r *is 1.*

The oscillation of a parse tree of a context-free grammar can be defined similarly to the oscillation of a PDA run. Given a parse tree t, we define corresponding well-nested word $\alpha(t)$ inductively as follows:

- if n is the root of t then $\alpha(t) = \bar{a}\alpha(n)$
- if n is a leaf then $\alpha(n) = a$
- if n has k children then $\alpha(n) = a \underbrace{\bar{a}...\bar{a}}_{k \text{ times}} \alpha(n_1)...\alpha(n_k)$

Moreover, given a PDA run r, there exists a corresponding parse tree t with the same well-nested word $\alpha(t) = \alpha(r)$ and vice versa [11].

The oscillation of a parse tree is closely related with its *dimension*. For each node v in a tree t, its dimension $dim(v)$ is inductively defined as follows:

- if v is a leaf, then $dim(v) = 0$
- if v is an internal node with k children $v_1, v_2, ..., v_k$ for $k \geq 1$, then

$$dim(v) = \begin{cases} \max_{i \in \{1...k\}} dim(v_i) & \text{if there is a unique maximum} \\ \max_{i \in \{1...k\}} dim(v_i) + 1 & \text{otherwise} \end{cases}$$

Dimension of a parse tree t $dim(t)$ is a dimension of its root. It is observable from the definition that dimension of a tree t is the height of the largest perfect binary tree, which can be obtained from t by contracting edges and accordingly identifying vertices. A tree with dimension $dim(t) = 2$ is illustrated in Figure 2.

It is known that the dimension of parse trees and its oscillation are in linear relationship.

Lemma 3 ([11]) *Let a grammar* $G = (\Sigma, N, P, S)$ *be in Chomsky normal form and let* t *be a parse tree of* G*. Then* $osc(t) - 1 \leq dim(t) \leq 2osc(t)$*.*

Context-free language reachability. A *directed labeled graph* is a triple $D = (Q, \Sigma, \delta)$, where Q is a finite set of nodes, Σ is a finite set of alphabet symbols, and

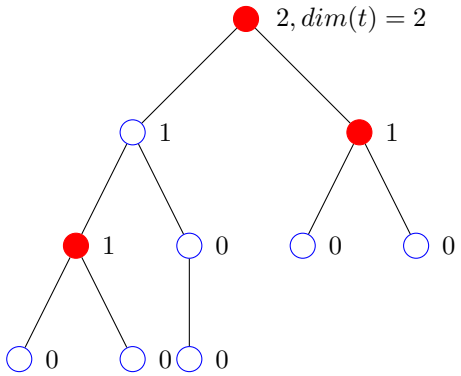

Fig. 2. A tree t with $dim(t) = 2$. Nodes having children without unique maximum are filled.

$\delta \subseteq Q \times \Sigma \times Q$ is a finite set of labeled edges. Let $L(D)$ denote a graph language—a regular language, which is recognized by the NFA $(Q, \Sigma, \delta, Q, Q)$ obtained from D by setting every state as inial and accepting.

Let $i\pi j$ denote a unique path between nodes i and j of the input graph and $l(\pi)$ denote a unique string obtained by concatenating edge labels along the path π. Then the CFL-reachability can be defined as follows.

Definition 4 (Context-free language reachability) *Let $L \subseteq \Sigma^*$ be a context-free language and $D = (Q, \Sigma, \delta)$ be a directed labeled graph. Given two nodes i and j we say that j is reachable from i if there exists a path $i\pi j$, such that $l(\pi) \in L$.*

There are four varieties of CFL-reachability problems: all-pairs problem, single-source problem, single-target problem and single-source/single-target problem [29]. In this paper we consider all-pairs problem. The *all-pairs problem* is to determine all pairs of nodes i and j such that j is reachable from i.

3. Rational index of bounded-oscillation languages

3.1. *Upper bounds on the rational index of bounded oscillation languages*

Before we consider the value of the rational index for k-bounded-oscillation languages, we need to prove the following.

Lemma 5. *Let $G = (\Sigma, N, P, S)$ be a context-free grammar in Chomsky normal form, $D = (V, E, \Sigma)$ be a directed labeled graph with n nodes. Let w be the shortest string in $L(G) \cap L(D)$. Then the height of every parse tree for w in G does not exceed $|N|n^2$.*

Proof. Consider grammar G' for $L(G) \cap L(D)$. The grammar $G = (\Sigma, N', P', S')$ can be constructed from G using the classical Bar-Hillel et al. [4] construction:

$N' \subseteq N \times V \times V$ contains all tiples (A, i, j) such that $A \in N, i, j \in V$; P' contains production rules in one of the following forms:

(1) $(A, i, j) \to (B, i, k), (C, k, j)$ for all (i, k, j) in V if $A \to BC \in P$
(2) $(A, i, j) \to a$ for all (i, j) in V if $A \to a$.

A triple (A, i, j) is *realizable* if and only if there is a path $i\pi j$ such that $A \stackrel{*}{\Rightarrow} l(\pi)$ for some nontermimal $A \in N$. Then the parse tree t_G for w in G can be converted into parse tree $t_{G'}$ in G'. Notice that every node of $t_{G'}$ is realizable triple. Also it is easy to see that the height of t_G is equal to the height of $t_{G'}$. Assume that $t_{G'}$ for w has a height of more than $|N|n^2$. Consider a path from the root of the parse tree to a leaf, which has length greater than $|N|n^2$. There are $|N|n^2$ unique labels (A, i, j) for nodes of the parse tree, so according to the pigeonhole principle, this path has at least two nodes with the same label. This means that the parse tree for w contains at least one subtree t with label (A, i, j) at the root, which has a subtree t' with the same label. Then we can change t with t' and get a new string w' which is shorter than w, because the grammar is in Chomsky normal form. But w is the shortest, then we have a contradiction. □

From Lemma 5 one can deduce an alternative proof of the fact that the rational index of linear languages is in $O(n^2)$ [5]: the number of leaves in a parse tree in linear grammar in Chomsky normal form is proportional to its height, and thus it is in $O(n^2)$.

Lemma 6. *Let G be a grammar $G = (\Sigma, N, P, S)$ in Chomsky normal form, such that every parse tree t has $dim(t) \leq d$, where d is some constant. Let $D = (V, E, \Sigma)$ be a directed labeled graph with n nodes. Then $\rho_{L(G)}$ is in $O(h^d)$ in the worst case.*

Proof. Proof by induction on dimension $dim(t)$.

(1) **Basis.** $dim(t) = 1$. Consider a tree t with the dimension $dim(t) = 1$. The root of the tree has the same dimension and has two children (because the grammar is in Chomsky normal form). There are two cases: first, when both of child nodes have dimension equal to 0, then the tree has only two leaves, and second, when one of the children has dimension 1, and the second child has dimension 0. For the second case we can recursively construct a tree with the maximum number of leaves in the following way. Every internal node of such a tree has two children, one of which has dimension equal to 0 and therefore has only one leaf. This means that the number of leaves (and, hence, $\rho_{L(G)}$) in such a tree is bounded by its height and is in $O(h)$.
(2) **Inductive step.** $dim(t) = d + 1$. Assume that $\rho_{L(G)}$ is at most $O(h^d)$ for every d in the worst case, where h is the height of the tree. We have two cases for the root node with dimension equal to $d + 1$: 1) both of children have a dimension equal to d, then by proposition the tree of height h has no more than $O(h^d)$ leaves; 2) one of the children has a dimension

$d+1$, and the second child v has a dimension $dim(v) \leq d$. Again, a tree with the maximum number of leaves can be constructed recursively: each node of such tree has two children u and v with dimensions $d+1$ and d respectively (the greater the dimension of the node, the more leaves are in the corresponding tree in the worst case). By the induction assumption there are no more than $(h-1)^d + (h-2)^d + (h-3)^d + ... + 1 = O(h^{d+1})$ leaves, so the claim holds for $dim = d+1$. □

Combining Lemma 5 and Lemma 6, we can deduce the following.

Corollary 7. *Let G be a grammar $G = (\Sigma, N, P, S)$ in Chomsky normal form, such that every parse tree t has $dim(t) \leq d$, where d is some constant. Let $D = (V, E, \Sigma)$ be a directed labeled graph with n nodes. Then $\rho_{L(G)}$ is in $O((|N|n^2)^d)$ in the worst case.*

Theorem 8. *Let L be a k-bounded-oscillation language with grammar $G = (\Sigma, N, P, S)$ in Chomsky normal form and $D = (V, E, \Sigma)$ be a directed labeled graph with n nodes. Then $\rho_{L(G)}$ is in $O(|N|^{2k} n^{4k})$ in the worst case.*

Proof. By Lemma 3, every parse tree of bounded-oscillation language has also bounded dimension. Then the maximum value of the dimension of every parse tree of k-bounded-oscillation language is $2k$. By Corollary 7, $\rho_{L(G)}$ is in $O((|N|n^2)^d)$ and, thus, $\rho_{L(G)}$ does not exceed $O((|N|n^2)^{2k}) = O(|N|^{2k} n^{4k})$. □

As we can see from the proof of Lemma 6, the family of linear languages is included in the family of bounded-oscillation languages. The reason is that the family of bounded-oscillation languages generalizes the family of languages accepted by finite-turn pushdown automata [11]. It is interesting that for general PDA, particularly for D_2, the value of oscillation is not constant-bounded: it depends on the length of input and does not exceed $O(\log n)$ for the input of length n [15, 33]. However, for some previously studied subclasses of context-free languages, oscillation is bounded by a constant.

3.2. *The rational indices of some subclasses of bounded-oscillation languages*

Superlinear languages. A context-free grammar $G = (\Sigma, N, P, S)$ is *superlinear* [6] if all productions of P satisfy these conditions:

(1) there is a subset $N_L \subseteq N$ such that every $A \in N_L$ has only linear productions $A \to aB$ or $A \to Ba$, where $B \in N_L$ and $a \in \Sigma$.
(2) if $A \in N \setminus N_L$, then A can have non-linear productions of the form $A \to BC$ where $B \in N_L$ and $C \in N$, or linear productions of the form $A \to \alpha B \mid B\alpha \mid \alpha$ for $B \in N_L$, $\alpha \in \Sigma^*$.

A language is *superlinear* if it is generated by some superlinear grammar.

Theorem 9. *Let G be a superlinear grammar. Then $\rho_{L(G)}$ is in $O(n^4)$.*

Proof. From the definition of superlinear grammar G it is observable that its parse trees have dimension at most 2. From Corollary 7, if dimensions of all parse trees are bounded by some k then the rational index $\rho_{L(G)}$ of such language is in $O(n^4)$. □

Ultralinear languages. A context-free grammar $G = (\Sigma, N, P, S)$ is *ultrealinear* if there exists a partition $\{N_0, N_1, ..., N_k\}$ of N such that $S \in N_k$ and if $A \in N_i$, where $0 \leq i \leq k$, then $(A \to w) \in P$ implies $w \in \Sigma^* N_i \Sigma^*$ or $w \in (\Sigma \cup N_0 \cup ... \cup N_i - 1)^*$. Such a partition is called an *ultralinear decomposition*. A language is *ultralinear* if it is generated by some ultralinear grammar.

The ultralinear languages were originally defined by Ginsburg and Spanier [12] as languages recognizable by finite-turn pushdown PDAs (a finite-turn PDA is a PDA with a fixed constant bound on the number of switches between push and pop operations in accepting computation paths).

Every ultralinear language is generated by an ultralinear grammar in *reduced form* [34].

Definition 10 (The reduced form of ultralinear grammar.) *An ultralinear grammar $G = (\Sigma, N, P, S)$ is in reduced form if its ultralinear decomposition $\{N_0, N_1, ..., N_k\}$ is in the following form:*

(1) $N_k = \{S\}$ and S does not appear in the right part of any production rule
(2) if $(A \to w) \in P \setminus \{S \to \varepsilon\}$ and $A \in N_i$, $0 \leq i \leq k$, then $w \in (\Sigma \cup N_i \Sigma \cup \Sigma N_i \cup N_j N'_j)$, where $j, j' < i$.

Theorem 11. *Let $G = (\Sigma, N, P, S)$ be an ultralinear grammar with the ultralinear decomposition $\{N_0, N_1, ..., N_k\}$. Then $\rho_{L(G)}$ is in $O(n^{2k})$.*

Proof. Recall that by definition dimension of a parse tree is the height of its largest perfect subtree. Consider the maximum possible size of a perfect subtree which occurs in the parse tree in ultralinear grammar in reduced form. It is easy to see that the rules of the form $A \to BC$, where $A \in N_i, B, C \in N_{i-1}$ should be used as often as possible to construct the largest binary subtree. Therefore, if grammar has the subset of rules of the form $\{S \to AB, A \to A_1A_2, B \to B_1B_2, A_1 \to A_3A_4, ..., A_i \to A_{i+2}, A_{i+3}, ...\}$, where $A, B \in N_{k-1}, A_1, A_2, B_1, B_2 \in N_{k-2}, A_3, A_4, ... \in N_{k-3}, ..., A_{i+2}, A_{i+3}, ... \in N_0$, the perfect binary subtree obtained with these rules will be of height not greater than k, so the maximum dimension of the parse tree in a ultralinear grammar in reduced form is k. By Corollary 7 $\rho_{L(G)}$ is in $O(n^{2k})$. □

4. Conclusions and open problems

We have proved that the bounded-oscillation languages have the polynomial rational index. It means that the CFL-reachability problem and Datalog query evaluation for these languages is in NC. This class is a natural generalization of linear languages, and possibly it comprises the largest previously known class of queries which is NC. It is interesting whether Datalog programs corresponding to these languages are linearizable (can always be transformed into linear Datalog programs).

There is a family of languages which has the polynomial rational index, but is not comparable with the linear languages: the family of one-counter languages. Moreover, it is not comparable with bounded-oscillation languages: for example, the Dyck language D_1 which is one-counter language, is not k-bounded-oscillation language for any k. Can this class be generalized in the same manner as linear languages with respect to the polynomiality of the rational index? One can consider the Polynomial Stack Lemma defined by Afrati et al. [1], where some restriction on the PDA stack contents are given, or investigate the properties of the family of the substitution closure of one-counter languages, which contains one-counter languages and has the polynomial rational index [5].

Acknowledgments

This research was supported by the Russian Science Foundation, grant №18-11-00100.

References

[1] F. Afrati and C. Papadimitriou, The parallel complexity of simple chain queries, *Proceedings of the Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, *PODS '87*, (ACM, New York, NY, USA, 1987), pp. 210–213.

[2] F. Afrati, M. Gergatsoulis and F. Toni, Linearisability on datalog programs, *Theoretical Computer Science* **308**(1) (2003) 199 – 226.

[3] R. Azimov and S. Grigorev, Context-free path querying by matrix multiplication, *Proceedings of the 1st ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, *GRADES-NDA '18*, (ACM, New York, NY, USA, 2018), pp. 5:1–5:10.

[4] Y. Bar-Hillel, M. Perles and E. Shamir, On formal properties of simple phreise structure grammars, *STUF - Language Typology and Universals* **14**(1-4) (01 Apr. 1961) 143 – 172.

[5] L. Boasson, B. Courcelle and M. Nivat, The rational index: a complexity measure for languages, *SIAM Journal on Computing* **10(2)** (1981) 284–296.

[6] J. A. Brzozowski, Regular-like expressions for some irregular languages, *9th Annual Symposium on Switching and Automata Theory (swat 1968)*, (Oct 1968), pp. 278–286.

[7] C. Cai, Q. Zhang, Z. Zuo, K. Nguyen, G. Xu and Z. Su, Calling-to-reference context translation via constraint-guided cfl-reachability (06 2018), pp. 196–210.

[8] K. Chatterjee, B. Choudhary and A. Pavlogiannis, Optimal dyck reachability for data-dependence and alias analysis, *Proc. ACM Program. Lang.* **2** (December 2017) 30:1–30:30.

[9] S. Cosmadakis and P. Kanellakis, Parallel evaluation of recursive rule queries, *Proceedings of the Fifth ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, *PODS '86*, (Association for Computing Machinery, New York, NY, USA, 1985), p. 280–293.

[10] H. Gaifman, H. Mairson, Y. Sagiv and M. Vardi, Undecidable optimization problems for database logic programs *Journal of the ACM* **40** (01 1987), pp. 106–115.

[11] P. Ganty and D. Valput, Bounded-oscillation pushdown automata, *Electronic Proceedings in Theoretical Computer Science* **226** (Sep 2016) 178–197.

[12] S. Ginsburg and E. H. Spanier, Finite-turn pushdown automata, *Siam Journal on Control* **4** (1966) 429–453.

[13] R. Greenlaw, H. J. Hoover and W. L. Ruzzo, *Limits to Parallel Computation: P-completeness Theory* (Oxford University Press, Inc., New York, NY, USA, 1995).

[14] S. Grigorev and A. Ragozina, Context-free path querying with structural representation of result, *Proceedings of the 13th Central & Eastern European Software Engineering Conference in Russia*, *CEE-SECR '17*, (ACM, New York, NY, USA, 2017), pp. 10:1–10:7.

[15] T. Gundermann, A lower bound on the oscillation complexity of context-free languages, *Fundamentals of Computation Theory, FCT '85, Cottbus, GDR, September 9-13, 1985*, (1985), pp. 159–166.

[16] J. Hellings, Path results for context-free grammar queries on graphs, *CoRR* **abs/1502.02242** (2015).

[17] M. Holzer, M. Kutrib and U. Leiter, Nodes connected by path languages, *Developments in Language Theory*, eds. G. Mauri and A. Leporati (Springer Berlin Heidelberg, Berlin, Heidelberg, 2011), pp. 276–287.

[18] W. Huang, Y. Dong, A. Milanova and J. Dolby, Scalable and precise taint analysis for android (07 2015), pp. 106–117.

[19] O. H. Ibarra, T. Jiang, J. H. Chang and B. Ravikumar, Some classes of languages in nc1, *Information and Computation* **90**(1) (1991) 86 – 106.

[20] O. H. Ibarra, T. Jiang and B. Ravikumar, Some subclasses of context-free languages in nc1, *Information Processing Letters* **29**(3) (1988) 111 – 117.

[21] B. Komarath, J. Sarma and K. S. Sunil, On the complexity of l-reachability, *Descriptional Complexity of Formal Systems*, eds. H. Jürgensen, J. Karhumäki and A. Okhotin (Springer International Publishing, Cham, 2014), pp. 258–269.

[22] Y. Lu, L. Shang, X. Xie and J. Xue, An incremental points-to analysis with

cfl-reachability, *Compiler Construction*, eds. R. Jhala and K. De Bosschere (Springer Berlin Heidelberg, Berlin, Heidelberg, 2013), pp. 61–81.

[23] A. Okhotin and K. Salomaa, Complexity of input-driven pushdown automata, *SIGACT News* **45** (2014) 47–67.

[24] J. Paramá, N. Brisaboa, M. Penabad and Á. Saavedra Places, A semantic query optimization approach to optimize linear datalog programs **2435** (01 2002), pp. 277–290.

[25] L. Pierre, Rational indexes of generators of the cone of context-free languages, *Theoretical Computer Science* **95**(2) (1992) 279 – 305.

[26] L. Pierre and J.-M. Farinone, Context-free languages with rational index in $\theta(n^\gamma)$ for algebraic numbers γ, *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications* **24**(3) (1990) 275–322.

[27] J. Rehof and M. Fähndrich, Type-base flow analysis: From polymorphic subtyping to cfl-reachability, *Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, *POPL '01*, (Association for Computing Machinery, New York, NY, USA, 2001), pp. 54–66.

[28] T. Reps, On the sequential nature of interprocedural program-analysis problems, *Acta Inf.* **33** (August 1996) 739–757.

[29] T. W. Reps, Program analysis via graph reachability, *Information & Software Technology* **40** (1997) 701–726.

[30] A. Rubtsov and M. Vyalyi, Regular realizability problems and context-free languages, *Descriptional Complexity of Formal Systems*, eds. J. Shallit and A. Okhotin (Springer International Publishing, Cham, 2015), pp. 256–267.

[31] J. D. Ullman and A. Van Gelder, Parallel complexity of logical query programs, *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, (Oct 1986), pp. 438–454.

[32] M. N. Vyalyi, Universality of regular realizability problems, *Computer Science – Theory and Applications*, eds. A. A. Bulatov and A. M. Shur (Springer Berlin Heidelberg, Berlin, Heidelberg, 2013), pp. 271–282.

[33] G. Wechsung, The oscillation complexity and a hierarchy of context-free languages, *Fundamentals of Computation Theory, FCT 1979, Proceedings of the Conference on Algebraic, Arthmetic, and Categorial Methods in Computation Theory, Berlin/Wendisch-Rietz, Germany, September 17-21, 1979.*, (1979), pp. 508–515.

[34] D. Workman, Turn-bounded grammars and their relation to ultralinear languages, *Information and Control* **32**(2) (1976) 188 – 200.

[35] M. Yannakakis, Graph-theoretic methods in database theory. (01 1990), pp. 230–242.

[36] Q. Zhang, M. R. Lyu, H. Yuan and Z. Su, Fast algorithms for dyck-cfl-reachability with applications to alias analysis, *SIGPLAN Not.* **48** (June 2013) 435–446.

[37] X. Zhang, Z. Feng, X. Wang, G. Rao and W. Wu, Context-free path queries on rdf graphs, *The Semantic Web – ISWC 2016*, eds. P. Groth, E. Simperl,

A. Gray, M. Sabou, M. Krötzsch, F. Lecue, F. Flöck and Y. Gil (Springer International Publishing, Cham, 2016), pp. 632–648.