Rational index of the languages of bounded dimension*

Ekaterina Shemetova[†‡§] Alexander Okhotin[†] Semyon Grigorev[†§]

January 8, 2022

Abstract

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 chain query evaluation for context-free languages (queries) with the polynomial rational index is in NL, while these problems is P-complete in the general case. We investigate the rational index of the languages of bounded dimension and show that it is of polynomial order. We obtain upper bounds on the values of the rational index for general languages of bounded dimension and for some of its previously studied subclasses.

Keywords. Dimension of a parse tree; rational index; CFL-reachability; parallel complexity; context-free languages; Datalog programs.

1 Introduction

The notion of a rational index was introduced by Boasson et al. [3] 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, determine 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 analyses [13] and graph database query evaluation [16]. The *Datalog chain query* evaluation on a database graph is equivalent to the CFL-reachability problem [15].

Unlike context-free language recognition, which is in NC for a fixed context-free grammar, the CFL-reachability problem is P-complete even for a fixed context-free grammar [7]. 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 extensively investigated by deductive database community [1, 15]. Ullman and Van Gelder [15] introduce the notion of a *polynomial fringe property* and show that chain queries having this property is in NC. The

*This research was supported by the Russian Science Foundation, project 18-11-00100.

[†]Department of Mathematics and Computer Science, 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.

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 chain 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 [12]. 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 $exp(\Theta(n^2/\ln n))$ [11], and, hence, this is the upper bound on the value of the rational index for every context-free language.

While the problem is not parallelizable in general, 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) [9, 10, 14]. These languages have the polynomial rational index.

The family of linear languages (linear Datalog chain programs, respectively) is the well-known subclass of context-free languages having the polynomial rational index [3, 15]. The value of its rational index is in $O(n^2)$ [3]. It is known that problems solvable by a linear Datalog Program are solvable in non-deterministic logarithmic space and, hence, highly parallelizable.

In this work we investigate the rational index of the languages of bounded dimension, which are the natural generalization of the linear languages. The dimension of a parse tree considered as a measure of its branching.

Our contributions. Our results can be summarized as follows:

- We show that the rational index of the languages of bounded dimension is polynomial and give an upper bound on its value in dependence of the value of dimension.
- We give a lower bound on the rational index of the languages of bounded dimension, particularly we show that for any dimension d there is a language of dimension d that has the rational index at least $n^{2d}/2^c$.

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 $A \xRightarrow{*} w$ 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.

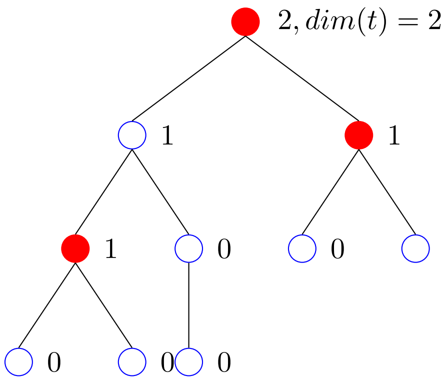


Figure 1: A tree t with $dim(t) = 2$. Nodes having children without unique maximum are filled.

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

$$\rho_L(n) = \max_{\mathcal{A}:\text{NFA with } n \text{ states}, L \cap L(\mathcal{A}) \neq \emptyset} \; \min_{w \in L \cap L(\mathcal{A})} |w|.$$

Languages of bounded 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

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

The dimension of a parse tree t $dim(t)$ is the dimension of its root. It is observable from the definition that the 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 of dimension $dim(t) = 2$ is illustrated in Figure 1.

Definition 1 (Grammars of bounded dimension)**.** *Context-free grammar G is of bounded dimension if the every parse tree t of G has $dim(t) \leq d$, where d is some constant. Then d is called a dimension $dim(G)$ of G.*

Definition 2 (Languages of bounded dimension)**.** *Languages of bounded dimension are languages generated by grammars of bounded dimension.*

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 $\delta \subseteq Q \times \Sigma \times Q$ 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 3 (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 [13]. In this paper we consider single-source/single-target problem.

3 Rational index of languages of bounded dimension

3.1 Upper bounds on the rational index of languages of bounded dimension

Assume w.l.o.g. that considered context free-grammars is a CFGs in Chomsky normal form as this simplifies the notation.

Before we estimate the value of the rational index for languages of bounded dimension, we need to prove the following.

Lemma 1. *Let $G = (\Sigma, N, P, S)$ be a context-free grammar and $\mathcal{A}$ be an NFA with n states. Let w be the shortest string in $L(G) \cap L(\mathcal{A})$. 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 construction by Bar-Hillel et al. [2]: $N' \subseteq N \times V \times V$ 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. But w is the shortest, then we have a contradiction. $\square$

From Lemma 1 one can deduce an alternative proof of the fact that the rational index of linear languages is in $O(n^2)$ [3]: the number of leaves in a parse tree in linear grammar is proportional to its height, and thus it is in $O(n^2)$.

Lemma 2. *Let $G = (\Sigma, N, P, S)$ be a context-free grammar and let $\mathcal{A}$ be an NFA with n states. Let w be the shortest string with parse tree of dimension d labeled by $A \in N$, such that $\mathcal{A}$ can read w starting from state p and ending in state q in $\mathcal{A}$. Then $|w| \le |N^d|n^{2d}$.*

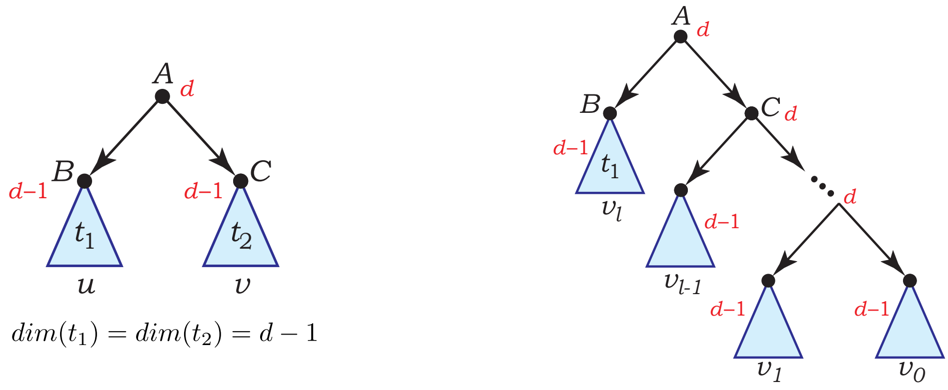
Proof. Proof by induction on $dim(G)$.

Basis. $dim(G) = 0$.

Consider the grammar G_0 with $dim(G_0) = 0$. $dim(G_0) = 0$ if G_0 has single rule of the form $S \to a$, where $a \in \Sigma$. Clearly, $|w| = 1 \le |N|n^0 = |N|$.

Inductive step. $dim(G) = d$.

Let $d \ge 1$, then $A \to BC \in P$. Let $w = uv$, where $B \stackrel{*}{\Rightarrow} u$ and $C \stackrel{*}{\Rightarrow} v$. Let t_1 and t_2 are parse trees for u and v respectively. By the definition of the dimension there are two cases:



(a) $dim(t_1) = dim(t_2) = d - 1$

(b) $dim(t_1) = d - 1$ and $dim(t_2) = d$

Figure 2: Two cases for the dimensions of children of the root

1. $dim(t_1) = dim(t_2) = d - 1$ (Figure 2a). Since w is the shortest, then u and v are the shortest, and the computation for w can be factorized as $p \overset{w}{\rightsquigarrow} q = p \overset{u}{\rightsquigarrow} r \overset{v}{\rightsquigarrow} q$. By the induction hypothesis, $|u|, |v| \leq |N|^{d-1}n^{2(d-1)}$. Then $|w| \leq 2|N|^{d-1}n^{2(d-1)} \leq |N|^d n^{2d}$.

2. $dim(t_1) < d$ and $dim(t_2) = d$ (Figure 2b). Then w can be factorized as $w = v_l v_{l-1} ... v_1 v_0$, where parse tree for v_i has dimension $d - 1$ in the worst case. By the induction hypothesis, $|v_i| \leq |N|^{d-1}n^{2(d-1)}$ for all i and by Lemma 1 $w = \sum_i |v_i| \leq l|N|^{d-1}n^{2(d-1)} \leq h|N|^{d-1}n^{2(d-1)} \leq |N|n^2|N|^{d-1}n^{2(d-1)} = |N|^d n^{2d}$.

   Case $dim(t_2) < d$ and $dim(t_1) = d$ can be proved symmetrically.

□

From Lemma 2 we immediately deduce the following

Theorem 1. *Let G be a grammar of bounded dimension d and let $\mathcal{A}$ be an NFA with n states. Then the length of the shortest string in $L(G) \cap L(\mathcal{A})$ is at most $|N^d|n^{2d}$.*

3.2 Lower bounds on the rational index of languages of bounded dimension

Theorem 2. *For every d there is a context-free grammar G of bounded dimension d such that for every $n \geq 2$ there is an n-state NFA $\mathcal{A}$, such that the shortest string w in $L(G) \cap L(\mathcal{A})$ is of length at least $n^{2d}/2^c$, where c is a constant dependent on the value of d.*

Proof. Automaton and grammar can be constructed inductively on $dim(G)$.

Basis. $dim(G) = 1$.

The family of the languages having dimension $d = 1$ coincides with the family of linear languages. Consider a linear grammar $G_1 = (\{a, b\}, \{S\}, \{S \to aSb \mid ab\}, S)$ which generates a language $L(G_1) = \{a^k b^k | k > 0\}$. Consider a NFA $\mathcal{A}_1$ consisting of two cycles connected via a shared node q_0 (Figure 3). Suppose the first cycle consists of m edges labeled with a, and the second cycle consists of m' edges labeled with b. Let m and m' be coprime integers, and let q_0 be start and final state of $\mathcal{A}_1$. Then the length of the shortest word $w \in L(G_1) \cap L(\mathcal{A}_1)$ equals $2mm'$. Suppose $\mathcal{A}_1$ has n states, and let $m = n/2 + 1$, $m' = n/2$. It is easy to see that m and m' are coprime for all n. Then $|w| = 2mm' = 2n/2(n/2 + 1) = O(n^2) = O(n^{2d})$. This example is well-known to the community [8, 16].

Inductive step. $dim(G) = d$.

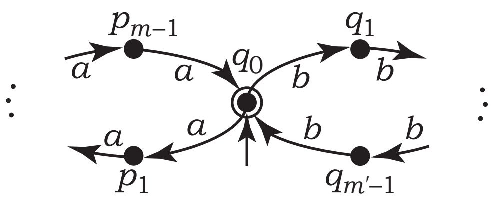
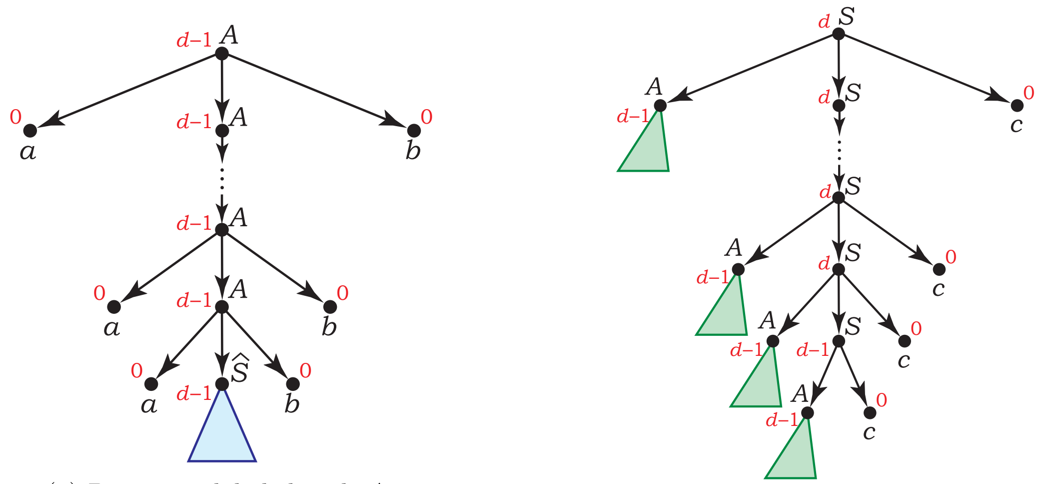


Figure 3: Worst-case NFA $\mathcal{A}_1$ for $L(G_1)$

Let $\hat{G} = (\hat{\Sigma}, \hat{N}, \hat{P}, \hat{S})$ be a context-free grammar with $dim(\hat{G}) = d - 1$ and $\hat{\mathcal{A}} = (\hat{Q}, \hat{\Sigma}, \hat{\delta}, \hat{q}_0, \{\hat{q}_0\})$ be a NFA. Let $\hat{w}$ be the shortest string in $L(\hat{G}) \cap L(\hat{\mathcal{A}})$.

Construction of the grammar $G = (\Sigma, N, P, S)$. Grammar G can be defined as follows:

- $\Sigma = \hat{\Sigma} \cup \{a, b, c\}$, where $a, b, c \notin \hat{\Sigma}$
- $N = \hat{N} \cup \{S, A\}$, where $A, S \notin \hat{N}$
- $P = \hat{P} \cup P'$, where $P' = \{$
  $S \to ASc \mid Ac$
  $A \to aAb \mid a\hat{S}b$
  $\}$.



(a) Parse tree labeled with A_d

(b) Parse tree labeled with S_d

Figure 4: Parse trees of G_d and dimensions of their vertices

It is left to show that dimension $dim(G) = dim(\hat{G}) + 1 = d$. Consider the dimension of the parse tree t labeled by the nonterminal A (Figure 4a). By the induction $dim(\hat{S}) = d - 1$. It is easy to see that dimension of $dim(t) = dim(\hat{G}) = d - 1$. Multiple applications of the rule $A \to aAb$ do not increase the dimension of the parse tree because the dimensions of nodes labeled with a, b are equal to 0.

Now consider the dimension of the parse tree t labeled by the nonterminal S (Figure 4b). As it was mentioned above, nodes labeled with A have dimension $d - 1$, nodes labeled with S have dimension $d - 1$ and nodes labeled with c have dimension 0. As there is no unique maximum, $dim(t) = \max_i(v_i) + 1 = d - 1 + 1 = d$. Notice that only one application of the rule $S \to ASc$ increases the dimension of t, whereas further applications do not make any effect on the dimension of parse tree.

Construction of NFA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ with n vertices. Suppose w.l.o.g. that n is divisible by 4. Assume that the number of states in NFA $\hat{\mathcal{A}} = (\hat{Q}, \hat{\Sigma}, \hat{\delta}, \hat{q}_0, \{\hat{q}_0\})$ equals to $n/2$ (by the induction hypothesis such an automaton exists). Fix two coprime integers $m = n/4 + 1$ and $m' = n/4$.

Then NFA $\mathcal{A}$ can be defined as follows:

- $\Sigma = \hat{\Sigma} \cup \{a, b, c\}$, where $a, b, c \notin \hat{\Sigma}$
- $\delta = \hat{\delta} \cup \{$
  $(q_0, a) \to \hat{q}_0,$
  $(\hat{q}_0, b) \to q_1,$
  $(q_i, b) \to q_{i+1}, 1 \leq i \leq m-1,$
  $(q_i, a) \to q_{i-1}, 1 \leq i \leq m-1,$
  $(q_{m-1}, b) \to q_0,$
  $(q_0, c) \to p_1,$
  $(p_i, c) \to p_{i+1}, 1 \leq i \leq m'-1,$
  $(p_{m'-1}, c) \to q_0\}$
- $F = \{q_0\}$
- $Q = \hat{Q} \cup \{q_0, ..., q_{m-1}, p_0, ..., p_{m'-1}\}$

The general form of $\mathcal{A}$ is shown in Figure 5.

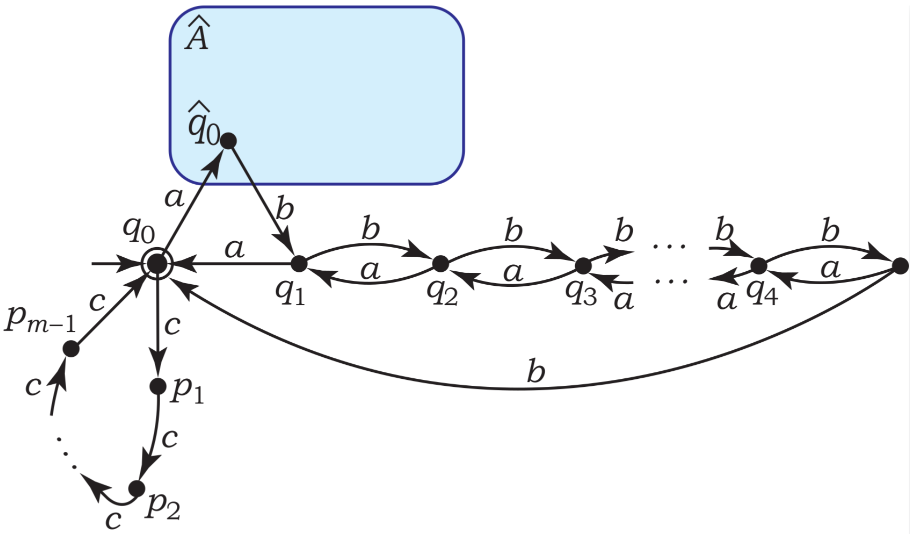


Figure 5: NFA $\mathcal{A}$

Let w be the shortest sting in $L(G) \cap L(\mathcal{A})$. Consider how w is formed. Start state is q_0. According to the grammar rule $S \to ASc \mid Ac$, w should start from a substring u such that $A \stackrel{*}{\Rightarrow} u$. There is the only one outgoing edge labeled with a, so the next state is $\hat{q}_0$. The next part of w should be a symbol a or a word v such that $\hat{S} \stackrel{*}{\Rightarrow} u$. As there is no outgoing edge labeled with a, u is the shortest string in $L(\hat{G}) \cap L(\hat{\mathcal{A}})$, and, hence, $u = \hat{w}$. Now the first part of w is $a\hat{w}$. To complete a substring derived by the nonterminal A, there is only one possible transition, which is an edge from $\hat{q}_0$ to q_1 labeled with b. The next substring should be symbol c (the rule $S \to Ac$) or a word derived by A. The only suitable transition here is an edge from q_1 to q_0 labeled by a, so the substring derived by the nonterminal A is started. Again, to complete the word generated by A, one goes to the state q_2, and w now starts with $a\hat{w}baa\hat{w}bb$. By the construction of NFA $\mathcal{A}$, this process continues until one comes to the state q_0 without starting a substring derived by the nonterminal A (notice that such substrings are the shortest possible). Clearly, it happens after m iterations. Then it is left to read m symbols c by going from q_0 to q_0.

But m and m' are coprime, so to balance the number of substrings derived by the nonterminal A and the number of symbols c, one needs to repeat the first cycle m' times and the second cycle m times.

Now estimate the length of w. Let w_i be the shortest string such that there exists computation $q_{i-1} \overset{w_i}{\rightsquigarrow} q_i$ ($q_{m-1} \overset{w_m}{\rightsquigarrow} q_0$ for w_m) for $1 \leq i \leq m$ in $\mathcal{A}$ and $A \overset{*}{\Rightarrow} w_i$. Notice that $w_i = aw_{i-1}b$ and $w_0 = \hat{w}$, and there exists computation $q_0 \overset{w_1}{\rightsquigarrow} q_1 \overset{w_2}{\rightsquigarrow} q_2 \overset{w_3}{\rightsquigarrow} \ldots \overset{w_{m-1}}{\rightsquigarrow} q_m \overset{w_m}{\rightsquigarrow} q_0$ in $\mathcal{A}$.

Considering the above and the rules of the grammar G $S \to ASc|Ac$, w is of the following form:

$$w = (\prod_{i=1}^{m} w_i)^{m'} c^{mm'}$$

Then the length of w can be calculated as follows:

$$|w| = (\sum_{i=1}^{m} |w_i|)m' + mm' = (\sum_{i=1}^{m}(|\hat{w}| + 2i))m' + mm' = (\sum_{i=1}^{m} |\hat{w}| + \sum_{i=1}^{m} 2i)m' + mm' =$$

$$= |\hat{w}|mm' + 2mm' + (m-1)mm' + mm' = |\hat{w}|mm' + 3mm' + (m-1)mm'.$$

As the NFA $\hat{\mathcal{A}}$ has $n/2$ states, by the induction assumption $|\hat{w}| = O((n/2)^{2d-2})$. Recall that $m = n/4 + 1$ and $m' = n/4$. Then the length of w is equal to:

$$|w| = O((n/2)^{2d-2})(n/4+1)n/4 + 3(n/4+1)n/4 + (n/4+1)n^2/16 =$$

$$= O((n/2)^{2d-2})n^2/16 = O(n^{2d}).$$

Let $n = 2^k$, where $k > 0$ is some constant.

For $d = 1$ let $m = 2^{k-1} + 1$ and $m' = 2^{k-1}$. Then $|w| = 2^k(2^{k-1} + 1) \geq 2^{2k-1}$.

For $d = 2$ let $m = 2^{k-2} + 1$ and $m' = 2^{k-2}$. Then $|w| \geq 2^{2k-3}(2^{k-2} + 1)2^{k-2} \geq 2^{4k-7}$.

Suppose that $|w| \geq 2^{2k(d-1)-c}$ for $d-1$ and some constant $c > 0$. Then for $dim = d$ $|w| \geq 2^{2(k-1)(d-1)-c}(2^{k-2}+1)2^{k-2} \geq 2^{2k-4+2kd-2k-2d+2-c} = 2^{2kd-2d-2-c} = 2^{2kd-c'}$, where $c' = c + 2d + 2$.

As $n = 2^k$, we have that $|w| \geq 2^{2kd-c} = n^{2d}/2^c$. □

3.3 The rational indices of some subclasses of languages of bounded dimension

Superlinear languages. A context-free grammar $G = (\Sigma, N, P, S)$ is *superlinear* [4] 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 3. *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 Theorem 1, if dimensions of all parse trees are bounded by some k then the rational index $\rho_{L(G)}$ of such language is in $O(n^4)$. □

Bounded-oscillation languages Bounded-oscillation languages were introduced by Ganty and Valput [6] as the generalization of the class of linear 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 $\bar{a}h_ia\ \bar{a}h_ia$.

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.

Definition 4 (Bounded-oscillation languages)**.** *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 [6].

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 [6]. Therefore, a language L is of bounded oscillation if all parse trees in a corresponding context-free grammar have bounded oscillation.

The oscillation of a parse tree is closely related with its dimension. It is known that the dimension of parse trees and its oscillation are in linear relationship.

Lemma 3 ([6])**.** *Let a grammar $G = (\Sigma, N, P, S)$ be in Chomsky normal form and let t be a parse tree of G. Then $osc(t) - 1 \leq dim(t) \leq 2osc(t)$.*

Combining Theorem 1 and Lemma 3 we obtain the following.

Corollary 1. *Let L be a k-bounded-oscillation language. Then $\rho_{L(G)}$ does not exceed $O(n^{4k})$.*

4 Conclusion and open problems

We have proved that the languages of bounded dimension have polynomial rational index. This 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 might be the largest class of queries among such generalizations that is known to be in NC.

There is a family of languages which has polynomial rational index, but is incomparable with the languages of bounded dimension: *the one-counter languages*. For example, the Dyck language D_1 is a one-counter language, but not the language of bounded dimension for any d. It is known that the rational index of one-counter languages is $O(n^2)$ [5].

Could this class be generalized in the same manner as linear languages preserving the polynomial order of the rational index? One can consider the Polynomial Stack Lemma by Afrati et al. [1], where some restriction on the PDA stack contents are given, or investigate the properties of the substitution closure of the one-counter languages, which is known to have polynomial rational index [3].

References

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

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

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

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

[5] Dmitry Chistikov, Wojciech Czerwiński, Piotr Hofman, Michał Pilipczuk, and Michael Wehar. Shortest paths in one-counter systems. In Bart Jacobs and Christof Löding, editors, *Foundations of Software Science and Computation Structures*, pages 462–478, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[6] Pierre Ganty and Damir Valput. Bounded-oscillation pushdown automata. *Electronic Proceedings in Theoretical Computer Science*, 226:178–197, Sep 2016.

[7] Raymond Greenlaw, H. James Hoover, and Walter L. Ruzzo. *Limits to Parallel Computation: P-completeness Theory.* Oxford University Press, Inc., New York, NY, USA, 1995.

[8] Jelle Hellings. Path results for context-free grammar queries on graphs. *CoRR*, abs/1502.02242, 2015.

[9] Markus Holzer, Martin Kutrib, and Ursula Leiter. Nodes connected by path languages. In Giancarlo Mauri and Alberto Leporati, editors, *Developments in Language Theory*, pages 276–287, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[10] Balagopal Komarath, Jayalal Sarma, and K. S. Sunil. On the complexity of l-reachability. In Helmut Jürgensen, Juhani Karhumäki, and Alexander Okhotin, editors, *Descriptional Complexity of Formal Systems*, pages 258–269, Cham, 2014. Springer International Publishing.

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

[12] Laurent Pierre and Jean-Marc 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):275–322, 1990.

[13] Thomas W. Reps. Program analysis via graph reachability. *Information & Software Technology*, 40:701–726, 1997.

[14] A. Rubtsov and M. Vyalyi. Regular realizability problems and context-free languages. In Jeffrey Shallit and Alexander Okhotin, editors, *Descriptional Complexity of Formal Systems*, pages 256–267, Cham, 2015. Springer International Publishing.

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

[16] Mihalis Yannakakis. Graph-theoretic methods in database theory. pages 230–242, 01 1990.