

# Rational index of bounded-oscillation languages\*

Ekaterina Shemetova<sup>†‡§</sup>

Alexander Okhotin<sup>†</sup>

Semyon Grigorev<sup>†§</sup>

July 19, 2021

## 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 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  $v$ , 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 analysis like data flow analysis and program slicing [29], alias analysis [8, 37], points-to analysis [22] and other [7, 18, 27], and graph database query evaluation [3, 14, 16, 38]. The *Datalog chain query* evaluation on a database graph is equivalent to the CFL-reachability problem [31, 32].

Unlike context-free language recognition, which is in NC (when context-free grammar is fixed), the CFL-reachability problem is P-complete [13, 28, 36]. 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, 32]. Ullman and Van Gelder [32] introduce the notion of a *polynomial fringe property* and show that chain queries having this property is in NC. The polynomial fringe

---

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

<sup>†</sup>Department of Mathematics and Computer Science, St. Petersburg State University, 7/9 Universitetskaya nab., Saint Petersburg 199034, Russia.

<sup>‡</sup>St. Petersburg Academic University, ul. Khlopina, 8, Saint Petersburg 194021, Russia.

<sup>§</sup>JetBrains Research, Primorskiy prospekt 68-70, Building 1, St. Petersburg, 197374, Russia.

property is equivalent to having the polynomial rational index: for a context-free language  $L(G)$  having the polynomial rational index  $\rho_L(n) = \text{poly}(n)$ , where  $\text{poly}(n)$  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  $\gamma$ , 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 algorithms 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, 33]. 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 [5, 32]. The value of its rational index is in  $O(n^2)$  [5]. It is known that problems solvable by a linear Datalog Program are solvable in non-deterministic logarithmic space and, hence, highly parallelizable. This class has received a lot of interest in complexity of constraint satisfaction, deductive databases and logic [2, 9, 24, 32].

In this work we investigate the rational index of bounded-oscillation languages. Bounded-oscillation languages were introduced by Ganty and Valput [11] as the generalization of the class of linear languages. 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.

**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  $\Sigma$  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 nonterminal.  $L(G)$  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 in the *Chomsky normal form*, if all production rules of  $P$  are of the form:  $A \rightarrow BC$ ,  $A \rightarrow a$  or  $S \rightarrow \varepsilon$ , 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,  $\Sigma$  is a input alphabet,  $\Gamma$  is a finite set which is called the stack alphabet,  $\delta$  is a finite subset of  $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times Q \times \Gamma^*$ ,  $q_0 \in Q$  is the start state,  $Z \in \Gamma$  is the initial stack symbol and  $F \subseteq Q$  is the set of accepting states.

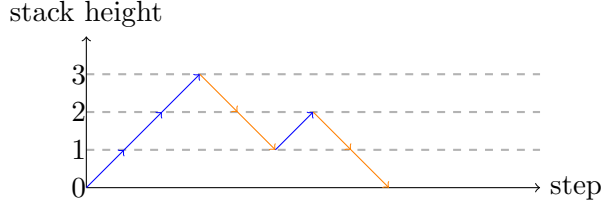


Figure 1: Stack heights during the run of PDA.

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,  $\Sigma$  is a finite set of input symbols,  $\delta : Q \times \Sigma \rightarrow 2^{|Q|}$  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  $\Sigma$ , its rational index  $\rho_L$  is a function defined as follows:

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

**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  $\varepsilon$
- $h_{(i+1)}$  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.

**Definition 1** (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 [11].

**Example 1.** Consider Figure 1. It shows how the stack height changes during the 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)):  $\bar{a}\bar{a}\bar{a}\bar{a}aa\bar{a}aa$ , 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  $\alpha(n) = a$

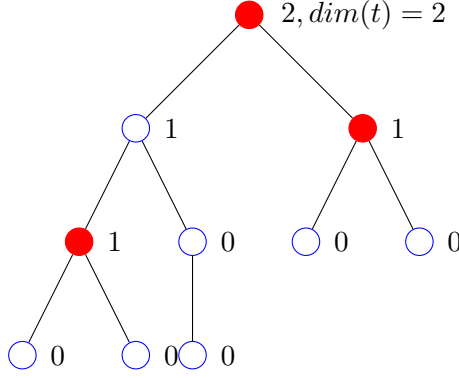


Figure 2: A tree  $t$  with  $\dim(t) = 2$ . Nodes having children without unique maximum are filled.

- if  $n$  has  $k$  children then  $\alpha(n) = a \underbrace{\bar{a} \dots \bar{a}}_{k \text{ times}} \alpha(n_1) \dots \alpha(n_k)$

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]. 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*. 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  $v_1, v_2, \dots, v_k$  for  $k \geq 1$ , then

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

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 2.

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

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

Thus if language  $L$  is bounded-oscillation language then all parse trees in a corresponding context-free grammar have 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,  $\Sigma$  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 initial 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  $\pi$ . Then the CFL-reachability can be defined as follows.

**Definition 2** (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 2.** *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 construction by Bar-Hillel et al. [4]:  $N' \subseteq N \times V \times V$  contains all tipes  $(A, i, j)$  such that  $A \in N, i, j \in V$ ;  $P'$  contains production rules in one of the following forms:

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

A triple  $(A, i, j)$  is *realizable* if and only if there is a path  $i\pi j$  such that  $A \xRightarrow{*} l(\pi)$  for some nonterminal  $A \in N$ . 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 2 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 3.** *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 and let  $w$  be the shortest string in  $L(G) \cap L(D)$ . Let  $h$  be the height of parse tree for  $w$  in  $G$ . Then  $|w|$  is in  $O(h^d)$  in the worst case.*

*Proof.* Proof by induction on  $d$ .

**Basis.**  $d = 1$ .

Consider a tree  $t$  with the dimension  $\dim(t) = d = 1$ . 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,  $|w|$ ) in such a tree is bounded by its height and is in  $O(h)$ .

**Inductive step.**  $d = d + 1$ .

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  $(h-1)^d + (h-2)^d + (h-3)^d + \dots + 1 = O(h^{d+1})$  leaves, so the claim holds for  $d = d + 1$ .  $\square$

**Lemma 4.** *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. Then  $\rho_{L(G)}$  is in  $O((|N|n^2)^d)$*

*Proof.* By Lemma 3 and the definition of the rational index  $\rho_{L(G)} = O(h^d)$ . By Lemma 2 for any  $n$ -node graph  $D$  ( $n$ -state NFA) and any context-free language,  $h$  is bounded by  $|N|n^2$ .  $\square$

**Theorem 1.** *Let  $L$  be a  $k$ -bounded-oscillation language with grammar  $G = (\Sigma, N, P, S)$  in Chomsky normal form. Then  $\rho_{L(G)}$  is in  $O(|N|^{2k}n^{4k})$  in the worst case.*

*Proof.* By Lemma 1, 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 Lemma 4,  $\rho_{L(G)}$  is in  $O((|N|n^2)^d)$  and, thus,  $\rho_{L(G)}$  does not exceed  $O((|N|n^2)^{2k}) = O(|N|^{2k}n^{4k})$ .  $\square$

As we can see from the proof of Lemma 3, 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, 34]. 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  $A \rightarrow aB$  or  $A \rightarrow Ba$ , 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  $A \rightarrow BC$  where  $B \in N_L$  and  $C \in N$ , or linear productions of the form  $A \rightarrow \alpha B \mid B\alpha \mid \alpha$  for  $B \in N_L$ ,  $\alpha \in \Sigma^*$ .

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

**Theorem 2.** *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 4, if dimensions of all parse trees are bounded by some  $k$  then the rational index  $\rho_{L(G)}$  of such language is in  $O(n^4)$ .  $\square$

**Ultralinear languages.** A context-free grammar  $G = (\Sigma, N, P, S)$  is *ultralinear* if there exists a partition  $\{N_0, N_1, \dots, N_k\}$  of  $N$  such that  $S \in N_k$  and if  $A \in N_i$ , where  $0 \leq i \leq k$ , then  $(A \rightarrow \alpha) \in P$  implies  $\alpha \in \Sigma^* N_i \Sigma^*$  or  $\alpha \in (\Sigma \cup N_0 \cup \dots \cup N_{i-1})^*$ . 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* [35].

**Definition 3** (The reduced form of ultralinear grammar). *An ultralinear grammar  $G = (\Sigma, N, P, S)$  is in reduced form if its ultralinear decomposition  $\{N_0, N_1, \dots, N_k\}$  is in the following form:*

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

**Theorem 3.** *Let  $G = (\Sigma, N, P, S)$  be an ultralinear grammar with the ultralinear decomposition  $\{N_0, N_1, \dots, N_k\}$ . 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  $A \rightarrow BC$ , 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  $\{S \rightarrow AB, A \rightarrow A_1 A_2, B \rightarrow B_1 B_2, A_1 \rightarrow A_3 A_4, \dots, A_i \rightarrow A_{i+2}, A_{i+3}, \dots\}$ , where  $A, B \in N_{k-1}, A_1, A_2, B_1, B_2 \in N_{k-2}, A_3, A_4, \dots \in N_{k-3}, \dots, A_{i+2}, A_{i+3}, \dots \in N_0$ , 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 4  $\rho_{L(G)}$  is in  $O(n^{2k})$ .  $\square$

## 4 Conclusions and open problems

We have proved that bounded-oscillation languages 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 linear languages: *the one-counter languages*. Moreover, it is not comparable with the bounded-oscillation languages: for example, the Dyck language  $D_1$  is a one-counter language, but not a bounded-oscillation language for any  $k$ . Could 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 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 [5].

## 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] Foto Afrati, Manolis Gergatsoulis, and Francesca Toni. Linearisability on datalog programs. *Theoretical Computer Science*, 308(1):199 – 226, 2003.
- [3] Rustam Azimov and Semyon Grigorev. Context-free path querying by matrix multiplication. In *Proceedings of the 1st ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, GRADES-NDA '18, pages 5:1–5:10, New York, NY, USA, 2018. ACM.
- [4] Y. Bar-Hillel, M. Perles, and E. Shamir. On formal properties of simple phrase structure grammars. *STUF - Language Typology and Universals*, 14(1-4):143 – 172, 01 Apr. 1961.
- [5] L. Boasson, B. Courcelle, and M. Nivat. The rational index: a complexity measure for languages. *SIAM Journal on Computing*, 10(2):284–296, 1981.
- [6] 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.
- [7] Cheng Cai, Qirun Zhang, Zhiqiang Zuo, Khanh Nguyen, Guoqing Xu, and Zhendong Su. Calling-to-reference context translation via constraint-guided cfl-reachability. pages 196–210, 06 2018.
- [8] Krishnendu Chatterjee, Bhavya Choudhary, and Andreas Pavlogiannis. Optimal dyck reachability for data-dependence and alias analysis. *Proc. ACM Program. Lang.*, 2(POPL):30:1–30:30, December 2017.
- [9] V. Dalmau. Linear datalog and bounded path duality of relational structures. *Log. Methods Comput. Sci.*, 1, 2005.
- [10] Haim Gaifman, Harry Mairson, Yehoshua Sagiv, and Moshe Vardi. Undecidable optimization problems for database logic programs. volume 40, pages 106–115, 01 1987.
- [11] Pierre Ganty and Damir Valput. Bounded-oscillation pushdown automata. *Electronic Proceedings in Theoretical Computer Science*, 226:178–197, Sep 2016.
- [12] S. Ginsburg and E. H. Spanier. Finite-turn pushdown automata. *Siam Journal on Control*, 4:429–453, 1966.
- [13] 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.
- [14] Semyon Grigorev and Anastasiya Ragozina. Context-free path querying with structural representation of result. In *Proceedings of the 13th Central & Eastern European Software Engineering Conference in Russia, CEE-SECR '17*, pages 10:1–10:7, New York, NY, USA, 2017. ACM.
- [15] Thomas Gundermann. A lower bound on the oscillation complexity of context-free languages. In *Fundamentals of Computation Theory, FCT '85, Cottbus, GDR, September 9-13, 1985*, pages 159–166, 1985.
- [16] Jelle Hellings. Path results for context-free grammar queries on graphs. *CoRR*, abs/1502.02242, 2015.
- [17] 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.



- [18] Wei Huang, Yao Dong, Ana Milanova, and Julian Dolby. Scalable and precise taint analysis for android. pages 106–117, 07 2015.
- [19] Oscar H. Ibarra, Tao Jiang, Jik H. Chang, and Bala Ravikumar. Some classes of languages in nc1. *Information and Computation*, 90(1):86 – 106, 1991.
- [20] Oscar H. Ibarra, Tao Jiang, and Bala Ravikumar. Some subclasses of context-free languages in nc1. *Information Processing Letters*, 29(3):111 – 117, 1988.
- [21] Balagopal Komarath, Jayalal Sarma, and K. S. Sunil. On the complexity of l-reachability. In Helmut Jürgensen, Juhani Karhumäki, and Alexander Okhotin, editors, *Descriptive Complexity of Formal Systems*, pages 258–269, Cham, 2014. Springer International Publishing.
- [22] Yi Lu, Lei Shang, Xinwei Xie, and Jingling Xue. An incremental points-to analysis with cfl-reachability. In Ranjit Jhala and Koen De Bosschere, editors, *Compiler Construction*, pages 61–81, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [23] Alexander Okhotin and Kai Salomaa. Complexity of input-driven pushdown automata. *SIGACT News*, 45:47–67, 2014.
- [24] José Paramá, Nieves Brisaboa, Miguel Penabad, and Ángeles Saavedra Places. A semantic query optimization approach to optimize linear datalog programs. volume 2435, pages 277–290, 01 2002.
- [25] Laurent Pierre. Rational indexes of generators of the cone of context-free languages. *Theoretical Computer Science*, 95(2):279 – 305, 1992.
- [26] Laurent Pierre and Jean-Marc Farinone. Context-free languages with rational index in  $\theta(n^\gamma)$  for algebraic numbers  $\gamma$ . *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications*, 24(3):275–322, 1990.
- [27] Jakob Rehof and Manuel Fähndrich. Type-base flow analysis: From polymorphic subtyping to cfl-reachability. In *Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL ’01, pages 54–66, New York, NY, USA, 2001. Association for Computing Machinery.
- [28] Thomas Reps. On the sequential nature of interprocedural program-analysis problems. *Acta Inf.*, 33(5):739–757, August 1996.
- [29] Thomas W. Reps. Program analysis via graph reachability. *Information & Software Technology*, 40:701–726, 1997.
- [30] A. Rubtsov and M. Vyalyi. Regular realizability problems and context-free languages. In Jeffrey Shallit and Alexander Okhotin, editors, *Descriptive Complexity of Formal Systems*, pages 256–267, Cham, 2015. Springer International Publishing.
- [31] O. Shmueli. Decidability and expressiveness aspects of logic queries. In *Proceedings of the Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS ’87, page 237–249, New York, NY, USA, 1987. Association for Computing Machinery.
- [32] 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.

- [33] Mikhail N. Vyalyi. Universality of regular realizability problems. In Andrei A. Bulatov and Arseny M. Shur, editors, *Computer Science – Theory and Applications*, pages 271–282, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [34] Gerd Wechsung. The oscillation complexity and a hierarchy of context-free languages. In *Fundamentals of Computation Theory, FCT 1979, Proceedings of the Conference on Algebraic, Arithmetic, and Categorical Methods in Computation Theory, Berlin/Wendisch-Rietz, Germany, September 17-21, 1979.*, pages 508–515, 1979.
- [35] D.A. Workman. Turn-bounded grammars and their relation to ultralinear languages. *Information and Control*, 32(2):188 – 200, 1976.
- [36] Mihalis Yannakakis. Graph-theoretic methods in database theory. pages 230–242, 01 1990.
- [37] Qirun Zhang, Michael R. Lyu, Hao Yuan, and Zhendong Su. Fast algorithms for dyck-cfl-reachability with applications to alias analysis. *SIGPLAN Not.*, 48(6):435–446, June 2013.
- [38] Xiaowang Zhang, Zhiyong Feng, Xin Wang, Guozheng Rao, and Wenrui Wu. Context-free path queries on rdf graphs. In Paul Groth, Elena Simperl, Alasdair Gray, Marta Sabou, Markus Krötzsch, Freddy Lecue, Fabian Flöck, and Yolanda Gil, editors, *The Semantic Web – ISWC 2016*, pages 632–648, Cham, 2016. Springer International Publishing.