International Journal of Foundations of Computer Science
© World Scientific Publishing Company

The Parallel Complexity of CFL-reachability Problem: Tractable Cases[*]

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)

Whereas it has been shown that context-free language (CFL) reachability problem is P-complete, there are some subclasses of context-free languages, for which the CFL-reachability lies in NC complexity class. We present two common classes which generalize known examples of such tractable subclasses: bounded-oscillation languages and context-free languages with a poly-slender storage languages. Polynomiality of the rational indices of languages in these classes is proved. Also closure properties of tractable subclasses in terms of polynomial rational index are investigated.

Keywords: the CFL-reachability; parallel complexity; digraphs; regular languages; context-free languages; context-free path queries.

1. Introduction

The context-free language (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, 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 [35], alias analysis [7, 43], points-to analysis [28] and other [6, 20, 33], and graph database query evaluation [2, 15, 18, 44].

Unlike context-free language recognition, which is in NC (when context-free grammar is fixed), CFL-reachability is P-complete [34, 42]. Practically, it means

[*]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.

that there is no efficient parallel algorithm for solving this problem (unless P $\neq$ NC).

While the problem 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 [22, 23, 30]. 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) [19, 26, 36].

The CFL-reachability problem has long been known to be P-complete [14]. The parallel complexity of this problem is studied by both static code analysis [34, 35] and database communities [1, 39, 42]. First investigations of this kind were made in terms of Datalog queries, because some classes of Datalog queries (chain queries) can be represented via context-free grammars with the database considered as a graph.

Example 1 (Datalog chain query as a context-free grammar) *Consider a database D with relation "child". It can also 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 the solving the CFL-reachability problem for a context-free grammar representation of this query and the edge-labeled graph G.

An important undecidability result was obtained by Gaifman et al. [9]: it is undecidable for given a context-free grammar G whether the CFL-reachability problem for G is in NC or P-complete. However, Ullman and Van Gelder [39] introduce a notion of a *polynomial fringe property* and show that the CFL-reachability problem for context-free grammars having this property is in NC. A context-free grammar G has the *polynomial fringe property* if and only if there is a polynomial p such that, for each regular language R recognized by an automaton with n states, $L(G) \cap R$ is either empty or contains a word shorter than $p(n)$. It is undecidable whether a context-free grammar has the polynomial fringe property. The results of Ullman and Van Gelder [39] can be reinterpreted in terms of the CFL-reachability as follows:

(1) CFL-reachability for linear languages is in NC, because the corresponding grammars have the polynomial fringe property

(2) The same holds for D_1 (the Dyck language on one pair of parentheses) and its images under finite-state transductions (one-counter languages)
(3) CFL-reachability for D_2 (the Dyck language on two pairs of parentheses) is P-complete.

The third result is important because every context-free language can be represented via a regular language and D_2 by Chomsky-Schützenberger theorem, so it is a direct consequence of P-completeness of CFL-reachability problem in general. Also, using the fact that D_2 is included in many interesting subclasses of context-free languages, such as visibly pushdown languages [30], simple deterministic languages (defined by LL(1) grammars in Greibach normal form), we can state that CFL-reachability for these languages is P-complete. Afrati et al. [1] investigate parallel complexity of Datalog simple chain queries and presents the Polynomial Stack Lemma which will be discussed in detail in Section 5.

The polynomial fringe property is equivalent to having polynomial *rational index*: for a context-free language $L(G)$ having the polynomial rational index is the same as for G to have the polynomial fringe property. More precisely, rational index $\rho_L(n)$ is a function, which denotes the maximum length of the shortest word in $L(G) \cap R$, for arbitrary R recognized by an n-state automaton. The notion of a rational index was introduced by L. Boasson et al. [4] as a complexity measure for context-free languages and was investigated independently of the polynomial fringe property. In particular, it has been proved that the rational index of D_1 is in $O(n^2)$ [8]. Another important result concerns the rational index of languages, which generate all context-free languages (an example of such language is D_2). It states that the rational index of such languages is of the order $exp(\Theta(n^2/\ln n))$ [31] and, hence, this is the upper bound on the value of the rational index for every context-free language. An example of a non-generating language with exponential rational index is given in [36]. Also it has been shown that for every algebraic number γ, a language with the rational index in $\Theta(n^\gamma)$ exists [32].

The CFL-reachability problem is the same as the intersection non-emptiness problem for a context-free language (pushdown automaton) and a regular language (finite automaton), because a labeled graph is a special kind of a nondeterministic finite automaton. The complexity of this problem was studied by Ganardi et al. [10] , Swernofsky et al. [37] and Vyalyi [40].

Computational complexity of the language reachability for different classes of languages (regular, context-free, context-sensitive) and graphs (acyclic graphs, trees, grid graphs) is discussed in detail by Barret et al. [3], Holzer et al.[19] and Komarath et al. [26].

Our focus is on investigating the parallel complexity of the CFL-reachability. Especially we are interested in generalizing of "easy" subclasses and discovering new examples of context-free languages, for which CFL-reachability is in NC. Effective subclasses can be useful in practice, because the general problem is not tractable [27]. For example, in the case of graph databases, it is important to know the

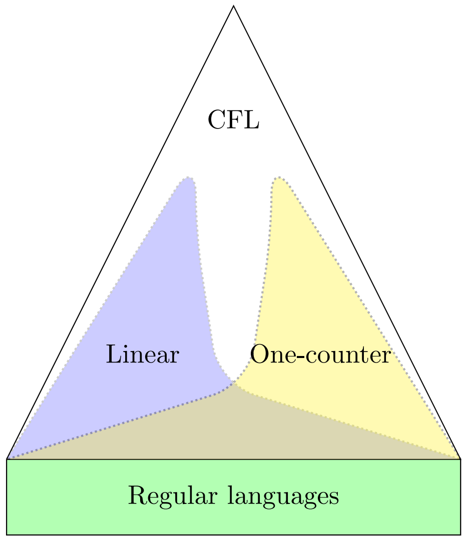

Fig. 1. The hierarchy of languages, for which the CFL-reachability problem is in NC.

complexity of a given context-free path query. Also it is natural to ask, which properties of subclasses imply that the problem is in NC. Why some languages have polynomial rational indices? What is the difference between them and other subclasses of context-free languages?

The hierarchy of subfamilies of context-free languages, for which the CFL-reachability problem is in NC, is presented in Figure 1. Linear languages and one-counter languages are incomparable families of context-free languages, but both have polynomial rational indices (the polynomial fringe property). These subfamilies have one thing in common: both are defined by strong restrictions on the stack in a pushdown automaton. Our main idea is to generalize the known tractable classes by investigating the restrictions on the PDA store.

Our contributions. Our results can be summarized as follows:

- We show that the CFL-reachability problem for bounded-oscillation languages of Ganty and Valput [11], is in NC (see Section 3). This class generalizes the case of linear languages.
- Closure properties of the languages with polynomial rational indices are investigated in Section 4, particularly it is shown that the family of languages with polynomial rational index is closed under the Kleene star and insertion of a regular language.
- In Section 5 we introduce a new subclass of context-free languages: context-free languages with a poly-slender pushdown store languages. These languages are the natural generalization of one-counter languages, and the CFL-reachability problem for them is in NC. Also we show that deciding

poly-slenderness of a pushdown store language is in P.

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 $A \overset{*}{\Rightarrow} w$ 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.

Some operations on languages will be mentioned during this paper.

A *homomorphism* is a function $h : \Sigma^* \to \Delta^*$ defined as follows:

- $h(\varepsilon) = \varepsilon$ and for $a \in \Sigma$, $h(a)$ is any string in Δ^*,
- for $a = a_1 a_2 ... a_k \in \Sigma^*$ $(k \geq 2)$, $h(a) = h(a_1)h(a_2)...h(a_k)$.

Given a homomorphism $h : \Sigma^* \to \Delta^*$ and a language L define

$$h(L) = \{h(w) | w \in L\} \subseteq \Delta^*.$$

Insertion of a language K into a language L is a language

$$L' = \{uxv | x \in K, u, v \in L\}.$$

A *full trio* is a family of languages is closed under arbitrary homomorphism and intersection with regular language. A *full AFL (abstract family of languages)* is a full trio closed under union, concatenation and the Kleene plus.

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, $q_0 \in Q$ is a start state, $F \subseteq Q$ 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.

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 a NFA obtained from a directed labeled graph D.

Let $i\pi j$ denote a unique path between nodes i and j of the input graph and $l(\pi)$ denote a unique string which is obtained from concatenation of edge labels along the path π. Then the general formulation of CFL-reachability can be stated 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 [35]. 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.

Definition 3 (Realizable triple) *For a context-free grammar $G = (\Sigma, N, P, S)$ and directed labeled graph $D = (Q, \Sigma, \delta)$, a triple (A, i, j) is realizable iff there is a path $i\pi j$ such that $A \overset{*}{\Rightarrow} l(\pi)$ for some nontermimal $A \in N$.*

Notice that if there exists a rule $A \to A_1 A_2 ... A_k \in P$ and a triple (A, i, j) is realizable then there exist realizable triples $(A_1, i, v_1), (A_2, v_2, v_3), ..., (A_k, v_k, j)$ for some vertices $v_1, ..., v_k$. Also if there exists an edge (i, j) labeled by a and the rule $A \to a \in P$ then triple (A, i, j) is realizable.

3. Bounded-oscillation languages

Bounded-oscillation languages were introduced by Ganty and Valput [11]. Just like one-counter and 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.

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 well-nested sequence $\alpha(r)$ of $\bar{a}$-s and a-s. Two positions $i < j$ form a *matching pair* if the corresponding

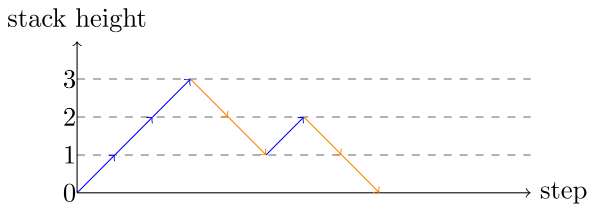

Fig. 2. Stack heights during the run of PDA.

$\bar{a}$ 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 is contained in r after removing 0 or more matching pairs. *Bounded-oscillation languages* are languages accepted by pushdown automata restricted to k-oscillating runs. It is important that the problem whether a given CFL is a bounded-oscillation language is undecidable [11].

Example 4. *Consider Figure 2. 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 $(2,5)$): $\bar{a}\bar{a}\mathbf{\bar{a}a}a\mathbf{\bar{a}a}a$, therefore oscillation of the run r is 1.*

Oscillation of the parse tree of a context-free grammar can be defined similiarly to 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 $dim(v) = 0$

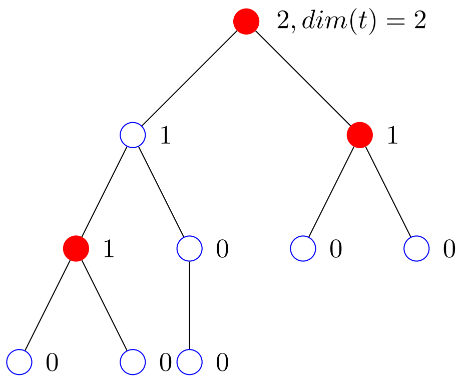

Fig. 3. A tree t with $dim(t) = 2$. Nodes having children without unique maximum are filled.

- 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 3.

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

Lemma 5 ([11]) *Let a grammar $G = (\Sigma, N, P, S)$ be in Chomsky normal form and let t be a parse tree of G. Then $osc(t) - 1 \leq dim(t) \leq 2osc(t)$.*

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

Lemma 6. *Let $G = (\Sigma, N, P)$ be a context-free grammar, $D = (V, E, \Sigma)$ 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 does not exceed $|N|n^2$.*

Proof. Consider a parse tree for w which is constructed in the following way: every node of the parse tree is marked with realizable triple (see Definition 3); for every rule $A \to A_1 A_2 ... A_k \in P$ a node labeled with (A, i, j) has children marked with $(A_1, i, v_1), (A_2, v_1, v_2), ..., (A_k, v_k, j)$, and every leaf is marked with a triple (a, i, j), where (i, j) is an edge of the input graph labeled with a symbol a. Assume that such parse tree 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 the length of more 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. □

From Lemma 6 we have that rational index of linear languages is in $O(n^2)$.

Lemma 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.*

Proof. Proof by induction on dimension $dim(t)$.
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. By Lemma 6, the maximum height h of such tree is at most $|N|n^2$, so the number of leaves in it is $O(h) = O(|N|n^2)$.
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 heght 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 $dim = d + 1$. Finally, $\rho_{L(G)}$ is almost $O(h^d) = O((|N|n^2)^d)$ for every d, because $h \leq |N|n^2$ by Lemma 6. □

Combining Lemma 5 and Lemma 7, we can deduce the following.

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 5, every parse tree of bounded-oscillation language has also bounded dimension. Then the maximum value of dimension of every parse tree of

k-bounded-oscillation language is $2k$. By Lemma 7, $\rho_{L(G)}$ is in $O((|N|n^2)^d)$ and, thus, $\rho_{L(G)}$ does not exceed $O((|N|n^2)^{2k}) = O(|N|^{2k}n^{4k})$. $\square$

As we can see from the proof of Lemma 7, 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 osciillation is not constant-bounded: it depends on the length of input and does not exeed $O(\log n)$ for the input of length n [16, 41]. However, for some previously studied subclasses of context-free languages, oscillation is bounded by a constant.

Example 9 (Superlinear languages [5]) *A context-free grammar $G = (\Sigma, N, P, S)$ is superlinear 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 $A \in N \backslash N_L$, 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 $B \in N_L$, $\alpha \in \Sigma^*$.

From the grammar G it is observable that its parse trees have dimension at most 2. From Lemma 7, if dimensions of all parse trees are bounded by some k then the rational index of such language is polynomial, so CFL-reachability problem for superlinear languages and an arbitrary graph is in NC.

4. Closure properties of languages with polynomial rational indices

Given a context-free language L with a polynomial rational index, it is interesting to find which language operations preserve this property. Boasson et al. [4] give following useful relations for polynomial indices of two languages L and L'.

Theorem 10 ([4]) *Context-free languages with polynomial rational indices are closed under intersection with a regular language, union, concatenation, homomorphism and inverse homomorphism. More precisely,*

- $\rho_{L\cup L'}(n) \leq \max(\rho_L(n), \rho_{L'}(n))$
- $\rho_{LL'}(n) \leq \rho_L(n) + \rho_{L'}(n)$
- $\rho_{L\cap R}(n) \leq \rho_L(nm)$, *where R is a regular language recognised by an m-state automaton*
- $\rho_{h(L)}(n) \leq \rho_L(n)$ *and* $\rho_{h^{-1}(L)}(n) < n(\rho_L(n) + 1)$, *where $h : \Sigma^* \to \Delta^*$ is a homomorphism.*

From the relations above it is easy to see that the family of context-free languages with polynomial rational indices is a full trio. Every full trio is closed under prefix

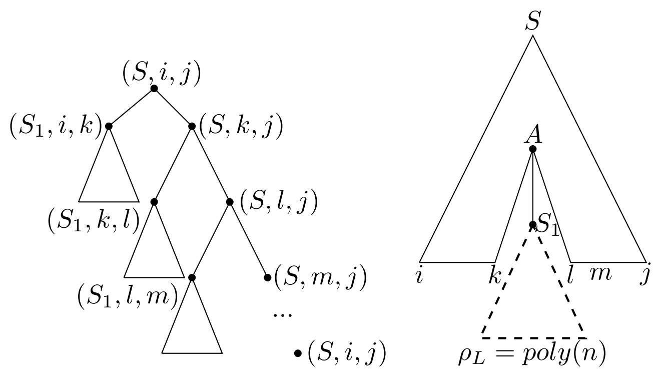

Fig. 4. Parse trees after applying the Kleene star operation (left) and insersion of the language with a polynomial rational index (right).

and quotient with regular languages. Obviously, CFLs with polynomial rational indices languages are closed under reversal. Next we show that context-free languages with polynomial rational indices are closed under the Kleene star and insertion of a regular language (or context-free language with a polynomial rational index).

Theorem 11. *Context-free languages with polynomial rational indices are closed under the Kleene star and insertion of a regular language (or context-free language with a polynomial rational index). Particularly,*

- $\rho_{L^*}(n) \leq n(\rho_L(n))$
- $\rho_{L_{INSERT(K)}}(n) \leq \rho_L(n) + \rho_K(n)$

Proof. *(The Kleene star)* Let $G = (\Sigma, N, P, S)$ be a grammar in Chomsky normal form and $L(G)$ be a language with polynomial rational index. Consider the language L^*. A grammar G' for L^* can be constructed from G as usual by renaming start nonterminal S to S_1 and adding a new start nonterminal S with production rules $S \to S_1 S$ and $S \to \varepsilon$. Consider a labeled parse tree for G' and some NFA (see Figure 4 (left)), constructed in the same way as a tree from the proof of Lemma 6. Suppose that this tree is a parse tree for the shortest word w in the intersection of $L(G')$ and an arbitrary n-state NFA with the start and final states i and j respectively. Such tree has no more than n subtrees rooted by S_1, because there is no more than n triples in form of (S, x, j) for a fixed j (otherwise w is not the shortest). Every tree rooted by S_1 has at most polynomial number of leaves, because $L(G)$ has a polynomial rational index. Thus $\rho_{L^*}(n) \leq n(\rho_L(n))$.
(Insertion of a regular language (or context-free language with a polynomial rational index)) Proof by contradiction. Consider a parse tree T for the shortest word w from the intersection of $L_{INSERT(K)}$, where K is a language with a polynomial rational index, and an arbitrary n-state NFA $\mathcal{A}$ with the start and final states i and j

respectively (see Figure 4 (right)). Let $w = l(i\pi j)$. Suppose that T has exponential on n number of leaves and, hence, a path $i\pi j$ has an exponential length. A subtree rooted by S_1 belongs to the grammar, which generates words from K, therefore this subtree has at most polynomial number of leaves by the definition of rational index. So, the whole tree, except the subtree rooted by S_1, has an exponential number of leaves. Suppose m is the next state in the shortest path from l to j in $i\pi j$. Then one can construct from $\mathcal{A}$ a new n-state NFA $\mathcal{A}'$, where state k has an edge to m labeled the same way as the transition from l to m in $\mathcal{A}$. Notice that the new edge does not affect the path from i to k, because this path is the shortest and, hence, does not contain any subpath from k to m. The same holds for the path from m to j. The the shortest word in the intersection of L and $\mathcal{A}'$ should have an exponential length, but L has a polynomial rational index, a contradiction. $\square$

The family of languages with polynomial rational indices is a full trio closed under union, concatenation and the Kleene star, therefore it is a full AFL. Full AFLs is known to be closed under substitution.

Using closure properties, it is easier to find new subclasses of context-free languages for which CFL-reachability problem is in NC.

Example 12 (Metalinear languages [13].) *Let $G = (\Sigma, N, P, S)$ be a context-free grammar. G is metalinear if all productions of P are of the following forms:*

(1) $S \to A_1A_2...A_k$, where $A_i \in N - \{S\}$
(2) $A \to u$, where $A \in N \setminus \{S\}$ and $u \in (\Sigma^*((N \setminus \{S\}) \cup \varepsilon)\Sigma^*)$

The width of a metalinear grammar is $max\{k|S \to A_1A_2...A_k\}$. Metalinear languages of width 1 are obviously linear languages. It is easy to see that every metalinear language is a union of concatenations of k linear languages. Linear languages have polynomial rational index, CFLs with polynomial rational index are closed under concatenation and union, so metalinear languages have polynomial rational index.

5. Languages with poly-slender store languages

In the Section 3 restriction on PDA in terms of variability of stack height was described. This restriction does not hold for PDA for the language D_1; however this language has polynomial rational index. In this section, another kind of stack restriction is considered— poly-slenderness of a pushdown store language as a measure of how stack contents vary along accepting computations of PDA.

For a PDA M, its *pushdown store language* $P(M)$ consists of all words occurring on the stack in accepting computations of M. It is well-known that the store language of every PDA is regular. The language D_1 is a one-counter language, so its pushdown store language is Z^*Z_0, where Z is a single pushdown symbol and Z_0 is the bottom symbol.

Afrati et al. [1] define the notion of *polynomial stack property* and show that if a PDA has the polynomial stack property, then the corresponding query has the polynomial fringe property (and hence, lies in NC). A PDA M has the polynomial stack property if and only if the number of different strings of length k occuring in the stack in any accepting computation of M is bounded by $O(k^d)$, for $d \geq 0$. For example, the usual PDA for D_1 has the polynomial stack property, because there is only one possible variant of contents for every stack height.

Generalizing an example of the family of one-counter languages, we can define the family of languages whose PDAs have the polynomial stack property—languages with a *poly-slender* pushdown store language (or storage language with polynomial density). The density of a language is a function $f(n)$ that shows the number of words of length n in the language. A language $L \subseteq \Sigma^*$ is called *poly-slender language (or with the polynomial density)* if the function $f(n)$ is bounded by $O(n^k)$ for some $k \geq 0$. For example, the language $Z^* Z_0$ is of polynomial density (even of a constant density), whereas the language $(Z_1 + Z_2)^* Z_0$ is of exponential density.

Whereas the polynomial fringe property of a query is undecidable [39], it is decidable in polynomial time whether a given PDA has a poly-slender storage language. At first, for a given NFA it is decidable whether its language has a polynomial or exponential density [17, 38]. Gawrychowski et al. [12] give an algorithm for testing whether $L(M)$ is of polynomial or exponential density in $O(|Q| + |\delta|)$ time for an NFA $M = (Q, \Sigma, \delta, q_0, F)$. An NFA for pushdown store language of a given PDA $\mathcal{A} = (Q', \Sigma', \Gamma, \delta', q_0', Z_0, F')$ can be constructed directly in $O(|Q'|^5 |\Gamma|^2 |\delta'|)$ time [29]. This construction uses the notion of meaningful triples, which form the states of NFA. A triple $[p, Z, q] \in Q' \times \Gamma \times Q'$ is *meaningful* if there exists a computation of $\mathcal{A}$ starting from state p with the sole symbol Z in the pushdown, and ending in q with the empty pushdown. By definition, there are at most $|\Gamma||Q'|^2$ meaningful triples, and, hence, states of NFA.

6. Conclusions and open problems

We have obtained two classes, which extend the classes in the recent literature [1, 19, 26, 36, 39], for which the CFL-reachability problem is in NC. One of them is the class of bounded-oscillation languages, which generalizes the linear languages. The second class is context-free languages with a poly-slender pushdown store languages, which is generalization of the one-counter languages. Recall that regular languages have polynomial fringe property (and are accepted by PDA with a bounded stack height), also it is known that L-reachability for regular languages is in NL [26, 42]. Thereby it has been demonstrated that some natural restrictions on the pushdown storage implies polynomial rational index for the corresponding context-free languages: low variability of stack height during the PDA run (bounded-oscillation PDA) and limited number of possible stack contents (languages with poly-slender pushdown store languages). The updated hierarchy of tractable sub-

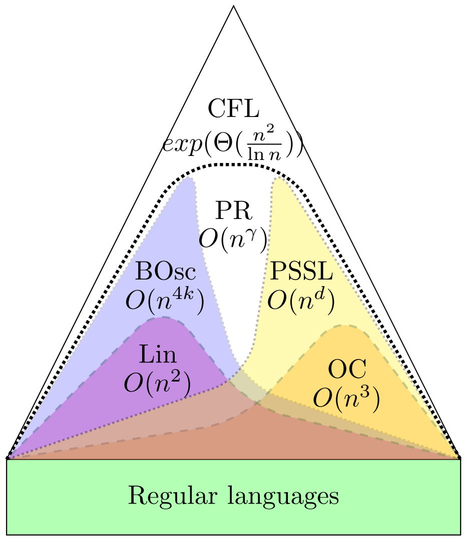

Fig. 5. The hierarchy of languages with polynomial rational indices and corresponding upper bounds on the value of rational index. PR — the family of CFLs with a polynomial rational indices, BOsc — bounded-oscillation languages, PSSL — CFLs with poly-slender storage languages, OC — one-counter languages, Lin — linear languages, n — number of vertices in graph (NFA), $|N|$ — the number of non-terminals of grammar in Chomsky normal form, k — the oscillation value, d — degree of polynomial density of a pushdown storage language, γ — algebraic number.

classes and corresponding upper bounds on the rational indices are illustrated in Figure 5.

It will be interesting to know whether there is any other stack restriction which implies polynomial rational index. Are there any other properties (except polynomial rational index) which make the CFL-reachability problem solvable in NC? For example, there is a Datalog query, which does not have the polynomial fringe property but its evaluation is in NC [25]. One can also approach this question from another direction by looking for simple subfamilies of context-free languages that would have P-complete CFL-reachability problem.

Also it is interesting to consider other machine models with stores having store restrictions examined in this work. Do these restrictions imply that certain properties could become decidable or their complexity could be reduced? Store languages of different machine models and their applications are studied in [21, 24].

We considered the CFL-reachability problem for a fixed context-free languages and arbitrary graphs. What tractable cases can be obtained for a fixed graphs and an arbitrary context-free language? The known and trivial examples are acyclic graphs and trees. Can we have more complicated classes of graphs for which the CFL-reachability problem is in NC? Interesting algebraic properties of such graphs (NFA) are given in [10], but automata-theoretic characterizations of these properties remain to be found.

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] 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.

[3] C. Barrett, R. Jacob and M. Marathe, Formal-language-constrained path problems, *SIAM J. Comput.* **30** (01 2000) 809–837.

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

[5] 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.

[6] 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.

[7] 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.

[8] J.-L. Deleage and L. Pierre, The rational index of the dyck language ${D_1}^*$, *Theoretical Computer Science* **47** (1986) 335 – 343.

[9] 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.

[10] M. Ganardi, D. Hucke, D. König and M. Lohrey, Circuit evaluation for finite semirings (2016).

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

[12] P. Gawrychowski, D. Krieger, N. Rampersad and J. Shallit, Finding the growth rate of a regular of context-free language in polynomial time, *Developments in Language Theory*, eds. M. Ito and M. Toyama (Springer Berlin Heidelberg, Berlin, Heidelberg, 2008), pp. 339–358.

[13] S. Ginsburg, *The Mathematical Theory of Context-Free Languages* (McGraw-Hill, Inc., USA, 1966).

[14] 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).

[15] 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.

[16] 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.

[17] O. H. Ibarra and B. Ravikumar, On sparseness, ambiguity and other decision problems for acceptors and transducers, *STACS 86*, eds. B. Monien and G. Vidal-Naquet (Springer Berlin Heidelberg, Berlin, Heidelberg, 1986), pp. 171–179.

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

[19] 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.

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

[21] O. Ibarra and I. McQuillan, On store languages of language acceptors, *Theor. Comput. Sci.* **745** (2018) 114–132.

[22] 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.

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

[24] O. H. Ibarra and I. McQuillan, On store languages and applications, *Information and Computation* **267** (2019) 28 – 48.

[25] P. C. Kanellakis, Logic programming and parallel complexity, *ICDT '86*, eds. G. Ausiello and P. Atzeni (Springer Berlin Heidelberg, Berlin, Heidelberg, 1986), pp. 1–30.

[26] 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.

[27] J. Kuijpers, G. Fletcher, N. Yakovets and T. Lindaaker, An experimental study of context-free path query evaluation methods, *Proceedings of the 31st International Conference on Scientific and Statistical Database Management*, *SSDBM '19*, (ACM, New York, NY, USA, 2019), pp. 121–132.

[28] 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.

[29] A. Malcher, K. Meckel, C. Mereghetti and B. Palano, Descriptional complexity of pushdown store languages, *J. Autom. Lang. Comb.* **17** (March 2012) 225–244.

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

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

[32] 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.

[33] 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.

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

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

[36] 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.

[37] J. Swernofsky and M. Wehar, On the complexity of intersecting regular, context-free, and tree languages, *Automata, Languages, and Programming*, eds. M. M. Halldórsson, K. Iwama, N. Kobayashi and B. Speckmann (Springer Berlin Heidelberg, Berlin, Heidelberg, 2015), pp. 414–426.

[38] A. Szilard, S. Yu, K. Zhang and J. Shallit, Characterizing regular languages with polynomial densities, *Mathematical Foundations of Computer Science 1992*, eds. I. M. Havel and V. Koubek (Springer Berlin Heidelberg, Berlin, Heidelberg, 1992), pp. 494–503.

[39] 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.

[40] 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.

[41] 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.

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

[43] 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.

[44] 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.