



# Tight bounds for reachability problems on one-counter and pushdown systems

Jakob Cetti Hansen, Adam Husted Kjelstrøm, Andreas Pavlogiannis\*

Department of Computer Science, Aarhus University, Denmark



## ARTICLE INFO

### Article history:

Received 8 January 2021  
Received in revised form 19 April 2021  
Accepted 19 April 2021  
Available online xxxx  
Communicated by Leah Epstein

### Keywords:

Pushdown systems  
One-counter systems  
Dyck reachability  
Formal languages

## ABSTRACT

We consider the problem of reachability on One-Counter Systems (OCSs) and Pushdown Systems (PDSs). The problem has a well-known  $O(n^3)$  bound on both models, while the bound is believed to be tight for PDSs. Here we establish new upper and lower bounds for reachability on OCSs and restricted PDSs. We show that the problem can be solved (i) in  $O(n^\omega \cdot \log^2 n)$  time on OCSs, and (ii) in  $O(n^2)$  time on sparse PDSs when restricted to witness executions of stack height bounded by  $\log n$ . Moreover, we prove similar lower bounds.

© 2021 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Pushdown Systems (PDSs) and One-Counter Systems (OCSs) are fundamental models of computation and among the standard formalisms in program verification. The use of pushdown storage allows static program analyses to model various notions of sensitivity (e.g., context-, field-sensitivity), which increases analysis precision [1]. OCSs can capture the effect of pointer references and dereferences, and thus provide a useful setting for pointer analyses [2].

The most fundamental question in these models is reachability: given two states, is there an execution that starts in one and ends in the other? Pushdown reachability, also known as CFL/Dyck reachability [3], has truly numerous applications in program verification [4,5]. Moreover, when system behavior is guided by inputs, reachability is the standard algorithmic formulation of the language emptiness problem, i.e., whether there exists a sequence of inputs that makes the system reach an accepting state.

Due to its importance, the reachability problem for PDSs has been studied extensively [6–8], and has a well-known  $O(n^3)$  bound. Although sub-cubic algorithms exist [9], they only offer poly-logarithmic improvements, and it is believed that there is no polynomial improvement over the cubic bound [10–12]. OCSs are a special case of PDSs, where the stack alphabet is unary, and thus inherits the cubic bound for reachability.

Given the large interest on the problem, two questions emerge naturally. First, is there an algorithm for OCSs that breaks below the cubic bound? Second, are there natural sub-classes of the problem for PDSs that can be solved more efficiently? We give tight answers to these questions in this work.

**Contributions.** Our contributions are matching upper- and lower-bounds for reachability problems on OCSs and PDSs. In particular, we show the following.

*One-counter systems.* Given an OCS  $\mathcal{C}$  of  $n$  states, we show that the all-pairs reachability problem is solvable in  $O(n^\omega \cdot \log^2 n)$  time, where  $\omega$  is the matrix-multiplication exponent. This generalizes a recent result of [2] and breaks the cubic bound for OCSs. Moreover, we show that even single-pair reachability is hard for finding triangles in undirected graphs [13] and thus inherits the (conditional) (i)  $n^\omega$  and

\* Corresponding author.

E-mail address: [pavlogiannis@cs.au.dk](mailto:pavlogiannis@cs.au.dk) (A. Pavlogiannis).

(ii) combinatorial  $n^3$  lower bounds of that problem. Note that (i) matches our upper bound (up to poly-logarithmic factors), hence our algorithm is *optimal* wrt polynomial improvements.

**Pushdown systems.** Given a sparse PDS  $\mathcal{P}$ , we consider the stack-bounded reachability question with maximum stack height  $b$ . The problem is motivated by static analysis applications, where the setting often gives rise to a sparse PDS (e.g., [3]), and witness bounding is used for fast approximate solutions (e.g., [7]). We show that the problem admits a straightforward algorithm with complexity  $O(n^2)$  for  $b = \log n$ , as opposed to the general  $O(n^3)$ . Moreover, we show that the problem is hard for Orthogonal-Vectors when just  $b = \omega(\log n)$ , in which case it has a (conditional) quadratic lower bound.

## 2. Reachability on one-counter systems

In this section we present our results for reachability on One-Counter Systems.

**One-counter systems** A *One-Counter System (OCS)* is a tuple  $C = (Q, \delta, \delta_{=0})$  where  $Q$  is a finite set of control states,  $\delta \subseteq Q \times Q \times \{-1, 0, 1\}$  is the transition relation and  $\delta_{=0} \subseteq \delta$  is the zero-test transition relation. When  $\delta_{=0} = \emptyset$  we call  $C$  a *one-counter net (OCN)*. Note that larger counter values in the transition function can be easily transformed to the  $\{-1, 0, 1\}$  case by the addition of extra states.

A *configuration* of an OCS is an element  $(q, c) \in Q \times \mathbb{N}$ , where  $q$  is the control state and  $c$  is a non-negative counter value. The semantics of  $C$  are defined wrt the configuration space  $\mathcal{G}_C$  with edge relation such that  $(q_1, c_1) \rightarrow (q_2, c_2)$  iff there exists some  $z \in \{-1, 0, 1\}$  such that (i)  $(q_1, q_2, z) \in \delta$ , (ii)  $c_2 = c_1 + z$ , and (iii)  $c_2 \geq 0$ . In words, the transition  $(q_1, q_2, z)$  can be fired in  $(q_1, c_1)$  iff  $c_1 + z \geq 0$ , in which case the system transitions to state  $q_2$  and adds  $z$  to the counter. Moreover, we require that if  $(q_1, q_2, z) \in \delta_{=0}$  then  $c_1 = 0$ , i.e., the transition is fired by first testing if the counter is 0.

**The reachability problem for OCS.** The (state) reachability problem for an OCS  $C$  is the following problem.

### One-counter reachability.

Input: An OCS  $C = (Q, \delta, \delta_{=0})$  and two states  $q_s, q_t \in Q$ .

Output: YES if there is a path  $P: (q_1, 0) \rightsquigarrow (q_2, 0)$  in  $\mathcal{G}_C$ , and NO otherwise.

We will also concern ourselves with the *all-pairs* variant, which asks to solve the reachability problem for all pairs of states in  $Q \times Q$ .

### 2.1. Upper bound

We first focus on the upper bound for the all-pairs problem. It was recently shown in [2] that the problem for

OCNs can be solved in  $O(n^\omega \cdot \log^2 n)$  time.<sup>1</sup> Here we generalize that result to OCSs, and obtain the following theorem.

**Theorem 1.** *The all-pairs reachability problem for OCSs can be solved in  $O(n^\omega \cdot \log^2 n)$  time on an OCS  $C$  with  $n$  states, where  $\omega$  is the matrix multiplication exponent.*

**Proof.** Let  $C = (Q, \delta, \delta_{=0})$ . First, we construct an OCS  $C^1 = (Q^1, \delta^1, \delta_{=0}^1)$  such that every zero-test transition is of the form  $(q_1, q_2, 0)$ . We achieve this by first letting  $C^1$  be identical to  $C$ , except that  $\delta_{=0}^1 = \emptyset$  and  $\delta^1 = \delta \setminus \delta_{=0}$ . For every pair  $(p, z) \in Q \times \{-1, 0, 1\}$ , we introduce a fresh state  $p' \in Q^1$ . Then, for every zero-test transition  $(p, q, z) \in \delta_{=0}$ , we introduce the transitions  $(p, p', 0), (p', q, z) \in \delta^1$ . Moreover, we introduce a fresh zero-test transition  $(p, p', 0) \in \delta_{=0}^1$ . It is easy to see that two states are reachable in  $C$  iff they are reachable in  $C^1$ , while  $C^1$  has  $O(n)$  states.

Second, we construct an OCN  $C^2 = (Q^1, \delta^1 \setminus \delta_{=0}^1, \emptyset)$ , i.e.,  $C^2$  is identical to  $C^1$  without the zero-test transitions. We use the algorithm of [2] to compute all-pairs reachability on  $C^2$ . Finally, we construct a graph  $G = (Q^1, E)$ , where

$$E = \delta_{=0}^1 \cup \{(q_1, q_2, 0) : q_2 \text{ is reachable from } q_1 \text{ in } C^2\}$$

and compute the transitive closure on  $G$ . The correctness of the above procedure follows straightforwardly.

Regarding the complexity, the construction of  $C^1, C^2$  and  $G$  requires time linear in the size of the corresponding OCS/OCN, which is bounded by  $O(n^2)$ . Moreover, the algorithm of [2] runs on an OCN with  $O(n)$  nodes, and thus takes  $O(n^\omega \cdot \log^2 n)$  time, while the transitive closure on  $G$  takes  $O(n^\omega)$  time.  $\square$

### 2.2. Lower bound

Finally, we establish two lower bounds for the reachability problem of OCNs. Our source of hardness is the problem of detecting triangles in undirected graphs.

#### Triangle detection.

Input: An undirected graph  $G = (V, E)$

Output: YES if there exist three nodes  $i, j, k \in V$  with  $(i, j), (j, k), (k, i) \in E$ , and NO otherwise.

The triangle detection problem can be solved in  $O(n^3)$  time by combinatorial algorithms, and in  $O(n^\omega)$  time in general. The corresponding hypothesis states that these bounds are tight wrt polynomial improvements, and it has been recently connected to other popular hypotheses in fine-grained complexity [13]. Our lower bounds follow a fine-grained reduction [14] from triangle detection.

**Reduction.** Consider an instance  $G = (V, E)$  of the triangle detection problem, and we construct a OCN  $C$  as follows. We assume wlog that  $V = [m]$ , i.e., it is the set of integers  $\{1, \dots, m\}$ . The set of states is

<sup>1</sup> The bound in [2] is phrased in the context of Dyck reachability with one parenthesis, which is a model isomorphic to OCNs.

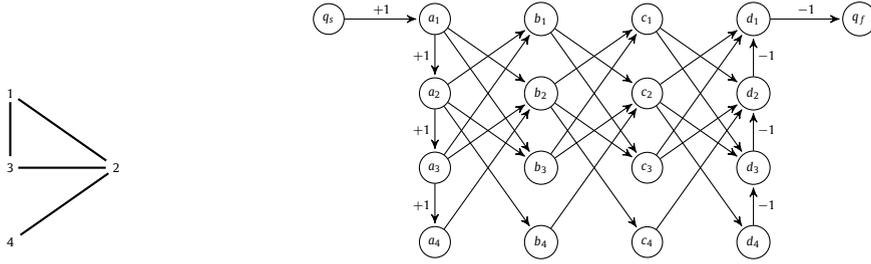


Fig. 1. An input graph  $G$  (left) and the OCA  $C$  constructed in our reduction (right). The path  $s, a_1, b_2, c_3, d_1, e$  in  $C$  is a witness of the triangle  $(1, 2, 3)$  in  $G$ .

$$Q = \{q_s, q_f\} \cup \bigcup_{i \in [m]} \{a_i, b_i, c_i, d_i\}$$

i.e., we have four states per node  $i \in V$ , plus two auxiliary states. The transition relation is  $\delta = A \cup B \cup C$ , where  $A = \{(q_s, a_1, 1), (d_1, q_f, -1)\}$ , and

$$B = \bigcup_{(i,j) \in E} \{(a_i, b_j, 0), (b_i, c_j, 0), (c_i, d_j, 0)\}$$

$$C = \bigcup_{i \in [m-1]} \{(a_i, a_{i+1}, +1), (d_{i+1}, d_i, -1)\}$$

Fig. 1 provides an illustration. We now prove the correctness of the above reduction, i.e., that  $q_f$  is reachable from  $q_s$  iff  $G$  has a triangle.

**Theorem 2.** *Under the triangle hypothesis, the following hold for the reachability problem on OCNs with  $n$  states, for any constant  $\epsilon > 0$ .*

1. *The problem has no algorithm with complexity  $O(n^{\omega-\epsilon})$ .*
2. *The problem has no combinatorial algorithm with complexity  $O(n^{3-\epsilon})$ .*

**Proof.** Consider the configuration space  $\mathcal{G}_C$ . Note that  $\mathcal{G}_C$  is acyclic, so every path visits each configuration at most once. It is easy to see that if  $(q_s, 0) \rightsquigarrow (a_i, k)$ , for some  $k \in \mathbb{N}$ , then  $k = i$ . Symmetrically, if  $(d_i, k) \rightsquigarrow (q_f, 0)$ , for some  $k \in \mathbb{N}$ , then  $k = i$ . Thus, any path  $P: (q_s, 0) \rightsquigarrow (q_f, 0)$  must be of the form

$$P: (q_s, 0) \rightsquigarrow (a_i, i) \rightarrow (b_j, i) \rightarrow (c_k, i) \rightarrow (d_\ell, i) \rightsquigarrow (q_f, 0)$$

which is only possible if  $i = \ell$ . By construction, the intermediate transitions exist in  $C$  iff  $(i, j), (j, k), (i, k) \in E$ , meaning that  $\{i, j, k\} \subseteq V$  form a triangle.

We now turn our attention to complexity. Let  $n = |Q|$ , and note that  $n = 4 \cdot m + 2 = O(m)$ , while clearly  $|\delta| = O(n^2)$ . Since  $O(n^2)$  is below the bounds stated in Item 1 and Item 2, we have a fine-grained reduction from the problem of triangle detection to the reachability problem of OCNs, as desired.  $\square$

Finally, observe that, under Item 1 of Theorem 2, our upper bound of Theorem 1 is optimal wrt polynomial improvements, as OCNs are a special case of OCSs.

### 3. Pushdown reachability with bounded witnesses

In this section we present our results for reachability on pushdown systems.

**Pushdown systems.** A pushdown system (PDS) is a tuple  $\mathcal{P} = (Q, \Gamma, \delta)$  where  $Q$  is a finite set of control states,  $\Gamma$  is the finite stack alphabet, and  $\delta \subseteq Q \times \Gamma \times Q \times \Gamma^*$  is a finite transition relation. We assume wlog that  $\Gamma = \{0, 1\}$ , as any PDS with stack alphabet of constant size can be easily converted to an equivalent one where the alphabet is binary.

A configuration of  $\mathcal{P}$  is an element  $(q, w) \in Q \times \Gamma^*$ , where  $q$  is a control state and  $w$  is the stack word. The semantics of  $\mathcal{P}$  are defined wrt the configuration space  $\mathcal{G}_{\mathcal{P}}$  with edge relation such that  $(q_1, w_1) \rightarrow (q_2, w_2)$  iff  $(q_1, \gamma, q_2, w) \in \delta$ , where  $w \in \Gamma^*$  and  $\gamma \in \Gamma \cup \{\varepsilon\}$  are such that either (i)  $w_1 = \gamma = \varepsilon$  and  $w_2 = w$ , or (ii)  $\gamma \neq \varepsilon$  and there exists  $w' \in \Gamma^*$  such that  $w_1 = \gamma w'$  and  $w_2 = w w'$ . We write  $(q_1, w_1) \rightsquigarrow (q_2, w_2)$  to denote a path  $P: (q^1, w^1) \rightarrow (q^2, w^2) \rightarrow \dots \rightarrow (q^k, w^k)$  with  $(q^1, w^1) = (q_1, w_1)$  and  $(q^k, w^k) = (q_2, w_2)$ . The *maximum stack height* of  $P$  is defined as  $\text{MSH}(P) = \max_i |w^i|$ , i.e., it is the length of the largest stack word in a configuration of  $P$ .

**The reachability problem for PDS.** The (state) reachability problem for a PDS  $\mathcal{P}$  asks, given two states  $q_s, q_t \in Q$ , to decide whether there exists a path  $P: (q_1, \varepsilon) \rightsquigarrow (q_2, \varepsilon)$  in  $\mathcal{G}_{\mathcal{P}}$ . The problem is known to be solvable in  $O(n^3)$  time, where  $n = |Q|$ , while no truly sub-cubic algorithm is expected to exist [10,11]. PDSs are one of the standard formalisms in static program analysis, where the analysis is phrased as a PDS reachability question [4,5,11]. The practical performance of these analyses further depends on two aspects of this setting. First, PDSs are typically sparse, i.e., we have  $|\delta| = O(n)$  [3,15,16]. Second, for efficiency reasons, such analyses typically restrict the search space to paths where the stack height is bounded [7,8,17–19]. Motivated by these two aspects, we study *stack-bounded* and *sparse* pushdown reachability.

#### Stack-bounded, sparse pushdown reachability.

Input: A PDS  $\mathcal{P} = (Q, \Gamma, \delta)$  with  $|\delta| = O(|Q|)$ , a bound  $b$ , and two states  $q_1, q_2 \in Q$ .

Output: YES if there is a path  $P: (q_1, \varepsilon) \rightsquigarrow (q_2, \varepsilon)$  in  $\mathcal{G}_{\mathcal{P}}$  with  $\text{MSH}(P) \leq b$ , NO if there is no path  $(q_1, \varepsilon) \rightsquigarrow (q_2, \varepsilon)$  in  $\mathcal{G}_{\mathcal{P}}$ , and YES/NO in any other case.

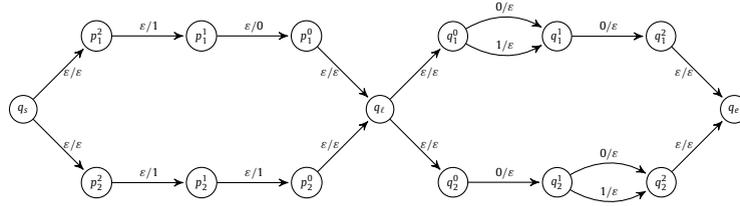


Fig. 2. The PDS  $\mathcal{P}$  for the OV instance  $A = \{01, 11\}$  and  $B = \{01, 10\}$ . An edge from  $p$  to  $q$  labeled  $x/y$  corresponds to a transition  $(p, x, q, y) \in \delta$ .

### 3.1. Upper bound

In this section we establish the following theorem.

**Theorem 3.** *The stack-bounded, sparse reachability problem with stack bound  $b = \log n$  can be solved in  $O(n^2)$  time on a PDS  $\mathcal{P}$  with  $n$  states.*

**Proof.** Let  $\mathcal{G}_{\mathcal{P}}^{\log n}$  be the configuration space of  $\mathcal{P}$  projected on configurations that are elements of  $(Q, \Gamma^{\leq \log n})$ . We construct  $\mathcal{G}_{\mathcal{P}}^{\log n}$  explicitly and use plain reachability to check if there exists a path  $q'_1 \rightsquigarrow q'_2$  in  $\mathcal{G}_{\mathcal{P}}^{\log n}$  where  $q'_1 = (q_1, \epsilon)$  and  $q'_2 = (q_2, \epsilon)$ .

Our  $\text{MSH}(\mathcal{P})$  is bounded by  $\log n$  which means the amount of different configurations of the stack is bounded by  $2^{\log n} = n$ . The number of nodes in  $\mathcal{G}_{\mathcal{P}}^{\log n}$  is then bounded by  $n \cdot 2^{\log n} = n^2$ . Because  $\mathcal{P}$  is sparse, then  $\mathcal{G}_{\mathcal{P}}^{\log n}$  is also sparse, so we can compute the problem in  $O(n^2)$  using plain reachability as  $\mathcal{G}_{\mathcal{P}}^{\log n}$  is just a plain, directed graph with  $O(n^2)$  nodes and  $O(n^2)$  edges.  $\square$

### 3.2. Lower bound

In the previous section we saw that a straightforward algorithm reduces the complexity of stack-bounded, sparse pushdown reachability from the general cubic bound down to quadratic when the stack bound is  $b = \log n$ . The simplicity of that algorithm motivates the following question: can a better bound (i.e., below quadratic) be established by a (possibly) more involved algorithm? Here we show that this is unlikely when  $b = \omega(\log n)$ . Our reduction is from the problem of Orthogonal Vectors.

#### Orthogonal Vectors (OV).

Input: Two set of vectors  $A, B \subseteq \{0, 1\}^D$  of size  $m$  each

Output: YES if there exists two vectors  $a \in A$  and  $b \in B$  that are orthogonal, i.e.,  $\forall i \in [D]: a[i] \cdot b[i] = 0$ , and NO otherwise.

The naive algorithm to solve OV is by matching every pair of vectors in  $A$  and  $B$  and checking if they are orthogonal, which yields a straightforward bound  $O(m^2 \cdot D)$ . The corresponding hypothesis states that the problem is not solvable in  $O(m^{2-\epsilon})$  time, for any  $\epsilon > 0$ , as long as  $D = \omega(\log n)$  [14]. We show that stack-bounded, sparse pushdown reachability is OV-hard for stack bound  $b = D$ .

**Reduction.** Consider an instance  $A, B$  of OV, where  $A = (a_i)_{1 \leq i \leq m}$  and  $B = (b_i)_{1 \leq i \leq m}$ , and we construct a PDS  $\mathcal{P}$  as follows. Fig. 2 provides an illustration. The set of states is  $Q = \{q_s, q_e, q_\ell\} \cup \bigcup_{j \in [D+1]-1} \{p_i^j, q_i^j\}$ .

We now define the transition function of  $\mathcal{P}$ . We start with the vectors of  $A$ .

For every  $i \in [m]$ , we have  $(q_s, \epsilon, p_i^D, \epsilon), (p_i^0, \epsilon, q_\ell, \epsilon) \in \delta$ . Moreover, for every  $j \in [D]$ , we have  $(p_i^{j+1}, \epsilon, p_i^j, a_i[j], \epsilon) \in \delta$ . We now proceed to the vectors of  $B$ .

For every  $i \in [m]$ , we have  $(q_\ell, \epsilon, q_i^0, \epsilon), (q_i^D, \epsilon, q_e, \epsilon) \in \delta$ . Moreover, for every  $j \in [D]$ , we have  $(q_i^{j-1}, 0, q_i^j, \epsilon) \in \delta$  and if  $a_i[j] = 0$ , we also have  $(p_i^{j-1}, 1, q_i^j, \epsilon) \in \delta$ .

Let  $\mathcal{P} = (Q, \{0, 1\}, \delta)$  and solve pushdown reachability from  $q_s$  to  $q_e$  on  $\mathcal{P}$ .

We now prove the correctness of the above construction, which establishes the following lower bound.

**Theorem 4.** *The stack-bounded, sparse pushdown reachability problem with stack bound  $b = \omega(\log n)$  has no  $O(n^{2-\epsilon})$ -time algorithm, for any  $\epsilon > 0$ , under the OV hypothesis.*

**Proof.** Consider any  $i, j \in [m]$ . We show by induction on  $k$  that  $(p_i^k, \epsilon) \rightsquigarrow (q_j^k, \epsilon)$  iff  $\forall k' \leq k$  we have  $a_i[k'] \cdot b_j[k'] = 0$ . The statement holds trivially for  $k = 0$ . Now, assume by the induction hypothesis that the statement holds for some  $k$ . Indeed, if  $a_i[k+1] \cdot b_j[k+1] = 0$ , then by construction,  $(p_i^{k+1}, \epsilon) \rightarrow (p_i^k, \gamma) \rightsquigarrow (q_j^k, \gamma) \rightarrow (p_j^{k+1}, \epsilon)$ , as we have  $(q_j^k, 0, q_j^{k+1}, \epsilon) \in \delta$ , and moreover  $(p_i^{k+1}, \epsilon, p_i^k, 0) \in \delta$  (if  $a_i[k+1] = 0$ ), otherwise  $(p_i^{k+1}, \epsilon, p_i^k, 1) \in \delta$  and  $(q_j^k, 1, q_j^{k+1}, \epsilon) \in \delta$  (if  $a_i[k+1] = 1$  and  $b_j[k+1] = 0$ ). On the other hand, if  $a_i[k+1] \cdot b_j[k+1] \neq 0$ , then by construction,  $(p_i^{k+1}, \epsilon, p_i^k, 0) \notin \delta$  and  $(q_j^k, 1, q_j^{k+1}, \epsilon) \notin \delta$ , and thus no path  $(p_i^{k+1}, \epsilon) \rightsquigarrow (q_j^{k+1}, \epsilon)$  exists. So given  $(p_i^k, \epsilon) \rightsquigarrow (p_j^k, \epsilon)$ , then  $(p_i^{k+1}, \epsilon) \rightsquigarrow (p_j^{k+1}, \epsilon)$  iff  $a_i[k+1] \cdot b_j[k+1] = 0$ .

In the end, we have  $(p_i^D, \epsilon) \rightsquigarrow (q_j^D, \epsilon)$  iff  $\forall k \in [D]$ , we have  $a_i[k] \cdot b_j[k] = 0$ , meaning  $a_i$  and  $b_j$  are orthogonal. By construction, we have  $(q_s, \epsilon) \rightsquigarrow (q_e, \epsilon)$  iff  $(p_i^D, \epsilon) \rightsquigarrow (q_j^D, \epsilon)$  for some  $i, j \in [m]$ . Finally, note that  $\mathcal{P}$  contains no cycles. Thus any path  $P: (q_s, \epsilon) \rightsquigarrow (q_e, \epsilon)$  must be of the form  $(p_s, \epsilon) \rightsquigarrow (q_\ell, w) \rightsquigarrow (p_e, \epsilon)$ . Finally,  $b = \text{MSH}(P) = |w| = D$ .

We now look at the complexity. The PDS  $\mathcal{P}$  has  $|\delta| \leq (2 \cdot D + 2) \cdot 2 \cdot m$  edges so  $|\delta| = O(D \cdot m)$  and  $|Q| \leq (D + 1) \cdot 2 \cdot m + 3$  nodes, thus  $|Q| = O(D \cdot m)$ . Hence the construction is an instance of stack-bounded, sparse pushdown reachability. Since the reduction takes  $O(D \cdot m)$  time,

an  $O(n^{2-\varepsilon})$  time algorithm for reachability on  $\mathcal{P}$  would yield an  $O(m^{2-\varepsilon'})$  bound for OV, for  $\varepsilon, \varepsilon' > 0$ .  $\square$

**Remark 1.** Program analyses are often parametric on the treewidth of the underlying structure [12,18,20], as control-flow graphs are known to have low treewidth [21]. In our reduction, the PDS  $\mathcal{P}$  is a series-parallel graph, hence our lower-bound in Theorem 4 also holds for graphs of treewidth 2. Thus treewidth alone is not a sufficient restriction to break the quadratic bound.

#### 4. Conclusion

In this work we have studied reachability on OCSs and PDSs. We have shown that the former model admits an  $O(n^\omega \cdot \log^2 n)$  bound, while the latter admits an  $O(n^2)$  bound when the input PDSs is sparse and we restrict the search space to witness executions of bounded stack height. Moreover, we have proven similar lower bounds based on popular conjectures in fine-grained complexity, showing that our upper-bounds are close to optimal.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### References

- [1] J. Späth, K. Ali, E. Bodden, Context-, flow-, and field-sensitive dataflow analysis using synchronized pushdown systems, in: POPL, 2019.
- [2] A.A. Mathiasen, A. Pavlogiannis, The fine-grained and parallel complexity of Andersen's pointer analysis, in: Proc. ACM Program. Lang., vol. 5 (POPL), Jan. 2021.
- [3] Q. Zhang, M.R. Lyu, H. Yuan, Z. Su, Fast algorithms for Dyck-CFL-reachability with applications to alias analysis, in: PLDI, 2013, pp. 435–446.
- [4] T. Reps, S. Horwitz, M. Sagiv, Precise interprocedural dataflow analysis via graph reachability, in: POPL, 1995, pp. 49–61.
- [5] T. Reps, Program analysis via graph reachability, in: Proceedings of the 1997 International Symposium on Logic Programming, ILPS '97, MIT Press, Cambridge, MA, USA, 1997, pp. 5–19.
- [6] A. Bouajjani, J. Esparza, O. Maler, Reachability analysis of pushdown automata: application to model-checking, in: CONCUR, Springer Berlin Heidelberg, 1997, pp. 135–150.
- [7] R. Alur, M. Benedikt, K. Etessami, P. Godefroid, T.W. Reps, M. Yannakakis, Analysis of recursive state machines, ACM Trans. Program. Lang. Syst. (2005).
- [8] K. Chatterjee, B. Kragl, S. Mishra, A. Pavlogiannis, Faster Algorithms for Weighted Recursive State Machines, Springer, Berlin, Heidelberg, 2017, pp. 287–313.
- [9] S. Chaudhuri, Subcubic algorithms for recursive state machines, in: POPL, 2008, pp. 159–169.
- [10] N. Heintze, D. McAllester, On the cubic bottleneck in subtyping and flow analysis, in: LICS '97, IEEE Computer Society, Washington, DC, USA, 1997, p. 342.
- [11] K. Chatterjee, B. Choudhary, A. Pavlogiannis, Optimal Dyck reachability for data-dependence and alias analysis, in: Proc. ACM Program. Lang., vol. 2 (POPL), Dec. 2018.
- [12] K. Chatterjee, G. Osang, Pushdown reachability with constant treewidth, Inf. Process. Lett. 122 (2017) 25–29.
- [13] V.V. Williams, R.R. Williams, Subcubic equivalences between path, matrix, and triangle problems, J. ACM 65 (5) (Aug. 2018).
- [14] V.V. Williams, On some fine-grained questions in algorithms and complexity, Tech. Rep, 2019.
- [15] M. Madsen, A. Møller, Sparse dataflow analysis with pointers and reachability, in: M. Müller-Olm, H. Seidl (Eds.), Static Analysis, Springer International Publishing, Cham, 2014, pp. 201–218.
- [16] A. Tavares, B. Boissinot, F. Pereira, F. Rastello, Parameterized construction of program representations for sparse dataflow analyses, in: A. Cohen (Ed.), Compiler Construction, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 18–39.
- [17] G. Xu, A. Rountev, Detecting inefficiently-used containers to avoid bloat, in: PLDI '10, ACM, New York, NY, USA, 2010, pp. 160–173.
- [18] K. Chatterjee, R. Ibsen-Jensen, A. Pavlogiannis, P. Goyal, Faster algorithms for algebraic path properties in recursive state machines with constant treewidth, in: POPL, 2015, pp. 97–109.
- [19] J. Späth, L.N.Q. Do, K. Ali, E. Bodden, Boomerang: demand-driven flow- and context-sensitive pointer analysis for Java, in: ECOOP, vol. 56, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016, 22.
- [20] K. Chatterjee, A.K. Goharshady, R. Ibsen-Jensen, A. Pavlogiannis, Optimal and perfectly parallel algorithms for on-demand data-flow analysis, in: ETAPS (ESOP), vol. 12075, Springer, 2020, pp. 112–140.
- [21] M. Thorup, All structured programs have small tree width and good register allocation, Inf. Comput. (1998).