Coverability in 1-VASS with Disequality Tests

Shaull Almagor

Technion, Israel

Nathann Cohen

CNRS & LRI, France

Guillermo A. Pérez

University of Antwerp, Belgium

Mahsa Shirmohammadi

CNRS & IRIF, Université de Paris, France

James Worrell

University of Oxford, UK

Abstract

We study a class of reachability problems in weighted graphs with constraints on the accumulated weight of paths. The problems we study can equivalently be formulated in the model of vector addition systems with states (VASS). We consider a version of the vertex-to-vertex reachability problem in which the accumulated weight of a path is required always to be non-negative. This is equivalent to the so-called control-state reachability problem (also called the coverability problem) for 1-dimensional VASS. We show that this problem lies in NC: the class of problems solvable in polylogarithmic parallel time. In our main result we generalise the problem to allow disequality constraints on edges (i.e., we allow edges to be disabled if the accumulated weight is equal to a specific value). We show that in this case the vertex-to-vertex reachability problem is solvable in polynomial time even though a shortest path may have exponential length. In the language of VASS this means that control-state reachability is in polynomial time for 1-dimensional VASS with disequality tests.

2012 ACM Subject Classification Theory of computation → Models of computaion

Keywords and phrases Reachability, Vector addition systems with states, Weighted graphs

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

In this paper we study reachability problems in weighted graphs with constraints on the accumulated weight along a path. We show that the vertex-to-vertex reachability problem is in NC if the constraint is that the accumulated weight must always be non-negative, and the problem is in polynomial time if we additonally allow disequality constraints on edges (i.e., constraints that prevent an edge from being taken in a path if the accumulated weight prior to taking the edge is equal to a specific value). In both cases a shortest path satisfying the constraints may have length exponential in the problem description. Several related problems have been studied in the literature, including the problem of finding a path from a source vertex to target vertex that has a specific total weight [12].

The problems we study can naturally be formalised as reachability problems for types of one-counter machines, and the majority of the related work has been presented in this context. Under this correspondence, the value of the counter represents the accumulated weight along a path, and tests on the counter encode constraints on allowable paths. Algorithmic properties of one-counter machines have been studied by many authors over several decades [2, 3, 5, 6, 7, 8, 9, 11]. The above references are a small subset of the extensive literature on one-counter machines, but they well illustrate that there are many variations on the basic model and that these variations can lead to the model having substantially different algorithmic properties. Particular features mentioned in the references above, driven by applications to automated verification and program analysis, include equality tests, disequality tests, inequality tests, parametric tests, binary updates, polynomial updates, and parametric updates.

Analysing the complexity of reachability in the presence of the features listed above leads to a rich complexity landscape. It is shown in [11] that control-state reachability is decidable in NL for a "plain vanilla" model of one-counters machine—namely with a counter taking values in the nonnegative integers with operations increment, decrement, and zero testing. Thinking of one-counter machines as one-dimensional vector addition systems with states (1-VASS), it is natural to allow the counter to be updated by adding integer constants in binary. In this case, still with equality tests, control-state reachability becomes NP-complete [9]. The NP upper bound here is non-trivial since, due to the binary encoding of integers, a computation that reaches the goal state may have length exponential in the size of the machine. If one enriches the model further by introducing inequality tests (comparing the counter with an integer constant) then control-state reachability becomes PSPACE-complete [6]. A model of intermediate complexity is one with equality and disequality tests (introduced in [5], with applications to temporal-logic model checking). In this case the complexity of control-state reachability is open (between NP and PSPACE).

In this paper we consider 1-VASS with disequality tests, but no equality tests. In terms of 1-VASS, our main result states that the control-state reachability problem is solvable in polynomial time for 1-VASS with disequality tests. This result confirms the intuition that disequality tests are weaker than equality tests. The main technical challenge to obtaining a polynomial-time bound is that a run witnessing that a given control state is reachable may have length exponential in the description of the counter machine. A standard way to overcome this obstacle in related settings is to show that one may restrict attention to computations that fit a regular pattern (usually in terms of iterating a "small" number of cycles). Here the presence of disequality tests proves to be surprisingly disruptive: it destroys the monotonicity of the transition relation and prevents from freely iterating positive-weight cycles. (For example, the lack of monotonicity means that it is coNP hard to determine whether, given a control state s_0, for all counter values $u \in \mathbb{N}$ the configuration (s_0, u) is unbounded, i.e., can reach infinitely many configurations—see Figure 1—whereas the same problem for 1-VASS without tests is easily seen to be decidable in polynomial time.) Resolving the complexity of reachability for 1-VASS with both equality and disequality tests remains open. We hope that the techniques developed here can help solve this challenging problem.

To complement our main result, we show that for 1-VASS without tests control-state reachability (and hence also boundedness) is decidable in NC, i.e., the subclass of P consisting of problems solvable in polylogarithmic parallel time. Problems in NC are in particular solvable in polylogarithmic space. Related to this, Rosier and Yen [15] have shown that boundedness for VASS is NL-complete in case there are absolute bounds on the dimension and bit-size of integer vectors.

Due to constraints on space, most proofs appear in the appendix.

2 Definitions

We write $\mathbb{N}$ to denote the set of all nonnegative integers $0, 1, 2, \ldots$ In presenting our results we assume familiarity of the reader with basic graph theory and computational complexity.

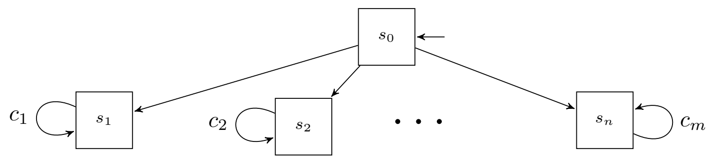


Figure 1 A 1-VASS with disequality tests, derived from a 3-CNF formula φ having propositional variables $X_1, \ldots, X_m$ and clauses $C_1, \ldots, C_n$. We have states $s_1, \ldots, s_n$—one state for each clause—and an initial state s_0. The reduction is such that (s_0, u) is unbounded for all $u \in \mathbb{N}$ iff φ is unsatisfiable. Let $p_1, \ldots, p_m$ be the first m primes and write $P := p_1 \cdots p_m$ for their product. For all $u \in \mathbb{N}$, define the propositional assignment $\mathrm{val}_u : \{X_1, \ldots, X_m\} \to \{0, 1\}$ by $\mathrm{val}_u(X_i) = 1$ if and only if $p_i \mid u$. Suppose that state s corresponds to a clause C that mentions variables $X_{i_1}, X_{i_2}, X_{i_3}$. Then we place a self-loop on s with increment $c_i := p_{i_1} p_{i_2} p_{i_3}$ and add disequlity tests on s (or equivalently on the self-loop on s) for all those values $u \in \{P, P+1, \ldots, P + p_{i_1} p_{i_2} p_{i_3} - 1\}$ where the assignment val_u satisfies the clause C. Given $u \in \{0, 1, \ldots, P-1\}$, observe that the configuration (s_0, u) is bounded iff val_u satisfies φ (see Appendix A for a complete proof).

One-Dimensional Vector Addition Systems with States and Tests. A *1-VASS with disequality tests* is a tuple $\mathcal{V} = (Q, D, \Delta, w)$, where Q is a set of *states*, $D = \{D_q\}_{q \in Q}$ is a collection of cofinite subsets $D_q \subseteq \mathbb{N}$, $\Delta \subseteq Q \times Q$ is a set of *transitions*, and $w : \Delta \to \mathbb{Z}$ is a function that assigns an integer weight to each transition. In the special case that each D_q equals $\mathbb{N}$, we simply call $\mathcal{V}$ a 1-VASS (and we omit the collection D).

A *configuration* of $\mathcal{V}$ is a pair (q, z) comprising a state $q \in Q$ and a nonnegative integer $z \in \mathbb{N}$ referred to as the *counter value*. We write *Conf* for the set $Q \times \mathbb{N}$ of all configurations. We define a partial order on *Conf* by $(q, z) \le (q', z')$ if and only if $q = q'$ and $z \le z'$. A configuration (q, z) is *valid* if $z \in D_q$.

A *path* in $\mathcal{V}$ is a sequence of states $\pi = q_1, \ldots, q_n$ such that $(q_i, q_{i+1}) \in \Delta$ for all $i \in \{1, \ldots, n-1\}$. We sometimes refer to such a path as a q_1-q_n path. Let $\pi' = p_1, p_2, \ldots, p_m$ be another path such that $q_n = p_1$, we define $\pi_1 \cdot \pi_2 := q_1, \ldots, q_n, p_2, \ldots, p_m$. Given states p, q, r, a set P of p-q paths, and a set R of q-r paths, we define $P \cdot R := \{\pi \cdot \pi' \mid \pi \in P, \pi' \in R\}$. The weight of π is defined to be $\mathrm{weight}(\pi) := \sum_{i=1}^{n-1} w(q_i, q_{i+1})$. A (possibly empty) prefix of π is said to be *minimal* if it has minimal weight among all prefixes of π. Define $\mathrm{pmin}(\pi)$ to be the weight of a minimal prefix of π.

A *run* is a sequence $(q_1, z_1), \ldots, (q_n, z_n)$ of configurations of $\mathcal{V}$ such that there is a path $\pi = q_1, \ldots, q_n$ with $z_{i+1} = z_i + w(q_i, q_{i+1})$ for $i = 1, \ldots, n-1$. We write $(q_1, z_1) \xrightarrow{\pi} (q_n, z_n)$ to denote such a run. Observe that runs are not allowed to reach negative counter values. A *valid run* is a run whose configurations are all valid. Intuitively, a valid run through q can proceed if and only if the current counter value is in D_q.

In computational problems all numbers in the description of $\mathcal{V}$ are given in binary. Given a state q we represent the cofinite set D_q as the complement of an explicitly given subset of $\mathbb{N}$. Given this convention, we can assume without loss of generality that for all states q the set D_q is either $\mathbb{N}$ or $\mathbb{N} \setminus \{g\}$ for some $g \in \mathbb{N}$; see Appendix B. For states q with $D_q = \mathbb{N} \setminus \{g\}$, we refer to the single missing value g in the domain as the *disequality guard* on q.

The Coverability and Unboundedness Problems. Let $\mathcal{V} = (Q, \Delta, D, w)$ be a 1-VASS with disequality tests, and let s and t be two distinguished states of $\mathcal{V}$. The *Coverability Problem* asks whether there exists a valid run in $\mathcal{V}$ from $(s, 0)$ to (t, z) for some $z \in \mathbb{N}$ (in which case we say that $(s, 0)$ can *cover* t). The *Unboundedness Problem* asks whether the set of configurations reachable from $(s, 0)$ is infinite (in which case we say that $(s, 0)$ is *unbounded*).

The Coverability problem reduces to the Unboundedness problem by, intuitively, forc-

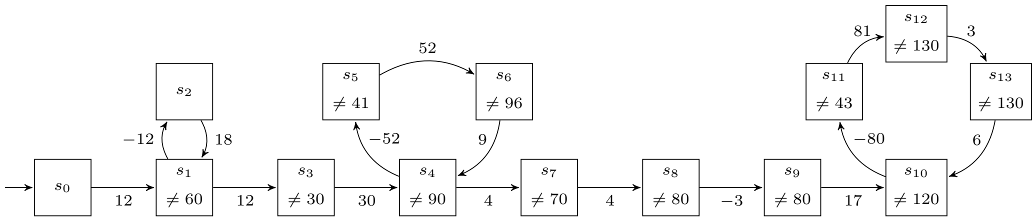


Figure 2 A 1-VASS with disequality tests. Disequality guards are denoted by $\neq$. For example, in state s_1 the set D_{s_1} is $\mathbb{N} \setminus \{60\}$, and no run goes through s_1 if its current counter value is 60.

ing $(t, 0)$ to be unbounded using a positive cycle, and removing all states that cannot reach t in the underlying graph of $\mathcal{V}$. In fact, the following holds.

▶ **Lemma 1.** *There is an* NC^2*-computable many-one reduction from the Coverability Problem to the Unboundedness Problem.*

Henceforth, we focus on the complexity of deciding the Unboundedness Problem. In Section 3 we prove that the Unboundedness Problem for 1-VASS with disequality tests is decidable in polynomial time. Since $\mathsf{NC}^2 \subseteq \mathsf{P}$, by Lemma 1 we also have that the Coverability Problem in this setting is decidable in polynomial time. In Section 4 we prove that the Unboundedness Problem for 1-VASS (without disequality tests) is in NC^2, and we deduce that the Coverability Problem for 1-VASS is decidable in NC^2.

3 Unboundedness for 1-VASS with Disequality Tests

Fix a 1-VASS $\mathcal{V} = (Q, D, \Delta, w)$ with disequality tests and a distinguished state $s \in Q$. We are interested in determining whether the configuration $(s, 0)$ is unbounded.

For a (possibly infinite) path $\pi = q_1, q_2, \ldots$, denote by $\mathrm{blocked}(\pi)$ the set of $z \in \mathbb{N}$ such that the unique induced run either contains a negative counter value or violates a disequality guard. That is, π does not lift to a valid run from the configuration (q_1, z).

▶ **Example 2.** In Figure 2, since 41 is the guard on s_5 the run $(s_4, 93), (s_5, 41), (s_6, 93)$ is not valid and $93 \in \mathrm{blocked}(s_4, s_5, s_6)$. Observe that $\mathrm{blocked}(s_4, s_5, s_6) = [0, 52) \cup \{90, 93, 96\}$ and $\mathrm{blocked}((s_4, s_5, s_6)^\omega) = [0, 52) \cup \{52 \le z \le 96 \mid z \equiv 0, 3, 6 \pmod 9\}$.

Recall that for a path π, $\mathrm{pmin}(\pi)$ is the weight of a minimum-weight prefix of π. Let $Q_+ \subseteq Q$ be the set of states $q \in Q$ such that there is a positive-weight simple cycle on q in the underlying graph of $\mathcal{V}$. For $q \in Q_+$ we pick a simple cycle γ_q such that $\mathrm{pmin}(\gamma_q) \ge \mathrm{pmin}(\gamma)$ for any other positive-weight simple cycle γ on q; write W_q for $\mathrm{weight}(\gamma_q)$.[1] Define $\mathit{Conf}_+ := \{(q, z) \in \mathit{Conf} \mid q \in Q_+,\ z + \mathrm{pmin}(\gamma_q) \ge 0\}$.

Define a path to be *primitive* if no proper infix is a positive cycle (note though that a primitive path may itself be a positive cycle). We say that a run is primitive if the underlying path is primitive. Observe that if ρ is a valid run, none of whose internal configurations (i.e. excluding the first and last configurations) lies in Conf_+, then ρ is primitive.

▶ **Example 3.** In Figure 2, for $s_1 \in Q_+$ we pick the simple cycle $\gamma_{s_1} = s_1, s_2, s_1$ with $W_{s_1} = 6$. Since $\mathrm{pmin}(\gamma_{s_1}) = -12$, we have that $\{z \mid (s_1, z) \in \mathit{Conf}_+\} = [12, \infty)$. Moreover, the path s_4, s_5, s_6, s_4 is primitive, but s_1, s_2, s_1, s_3 is not primitive.

[1] Note that γ_q does not necessarily maximize W_q.

▶ **Proposition 4.** *A configuration $(s, 0)$ is unbounded if, and only if, $(s, 0)$ can reach an unbounded configuration in $Conf_+$.*

In order to decide whether $(s, 0)$ is unbounded, by Proposition 4, it suffices to compute the set of unbounded configurations in $Conf_+$ and determine whether $(s, 0)$ can reach this set. Define $Conf_\infty \subseteq Conf_+$ to be the set of all unbounded configurations in $Conf_+$. Observe that every configuration $(q, z) \in Conf_+$ with $z \notin \text{blocked}(\gamma_q^\omega)$ can take the cycle γ_q arbitrarily many times and is thus included in $Conf_\infty$. However, even if $z \in \text{blocked}(\gamma_q^\omega)$, it may still be the case that (q, z) is unbounded, by traversing more complicated paths.

▶ **Example 5.** In Figure 2, all configurations (s_4, z) with z in $\mathbb{N} \setminus \text{blocked}((s_4, s_5, s_6)^\omega) = \{52 \leq z \leq 96 \mid z \not\equiv 0, 3, 6 \pmod 9\} \cup (96, \infty)$ are trivially unbounded and thus included in $Conf_\infty$. It will transpire that $\{s_4\} \times \{54, 60, 63, 69\} \subseteq Conf_\infty$ even though $\{54, 60, 63, 69\} \in \text{blocked}((s_4, s_5, s_6)^\omega)$.

In order to reason about the aforementioned complicated paths, we proceed as follows. In Section 3.1 we introduce residue classes and chains, which form a partition of $Conf_+$, and are the building blocks of our analysis. In Section 3.2 we characterize $Conf_\infty$ as the limit of an inductive construction. This enables us to reason about the structure of $Conf_\infty$ in Section 3.3. Finally, in Section 3.4 we show how to compute $Conf_\infty$ and decide unboundedness.

3.1 Residue Classes and Chains

Given $q \in Q_+$ and $0 \leq r < W_q$, we call the set of configurations $\{(q, z) \in Conf_+ \mid z \equiv r \pmod{W_q}\}$ a *q-residue class*. We simply speak of a *residue class* if we do not want to specify the state q. Given a q-residue class R, a set $C \subseteq R$ is called a *q-chain* if it is a maximal subset of R with the property that every pair of configurations $(q, z), (q, z') \in C$ with $z < z'$ is connected by a valid run obtained by iterating the cycle γ_q. Again, we speak of a *chain* if we do not want to specify the state q.

We draw a distinction between *bounded chains* and *unbounded chains*, where a chain is bounded if and only if the associated set of counter values is bounded. An unbounded q-chain C is contained in $Conf_\infty$ since the cycle γ_q can be taken arbitrarily many times from any configuration in C to yield a valid run.

▶ Remark 6. For each q-residue class R, each guard value z induces at most two bounded chains. Namely, the set of configurations below (q, z) form a chain; and the singleton $\{(q, z)\}$ is also (vacuously) a chain. Note that every residue class also has one unbounded chain. That is, the set of configurations above (q, z) with z the highest guard on q. Since there are at most $|Q|$ guards, each residue class decomposes as a disjoint union of at most $2|Q|$ bounded chains and a single unbounded chain.

Intuitively, within each bounded chain we can iterate the cycle γ_q until hitting a guard. We call a residue class R *trivial* if it consists solely of a single unbounded chain. Note that the union of all bounded q-chains is equal to $Conf_+ \cap \{q\} \times \text{blocked}(\gamma_q^\omega)$.

▶ **Example 7.** As indicated in Figure 3 for the running example, the residue classes $\{s_4\} \times (52 + i + 9\mathbb{N})$ with $i \in \{0, 1, 3, 4, 6, 7\}$ are indeed trivial, while each residue class $\{s_4\} \times (52 + i + 9\mathbb{N})$ with $i \in \{2, 5, 8\}$ consists of two bounded chains $\{s_4\} \times \{52 \leq z < 88 + i \mid z \equiv i \pmod 9\}$ and $\{s_4\} \times \{88 + i\}$, and a single unbounded chain $\{s_4\} \times (88 + i + 9\mathbb{N})$.

One of the main ideas in this section is to show that a configuration is unbounded if and only if it can reach an unbounded chain via a valid run whose underlying path π has the form

$$\pi = \pi_0 \cdot \gamma_{q_1}^{n_1} \cdot \pi_1 \cdots \pi_{k-1} \cdot \gamma_{q_k}^{n_k} \cdot \pi_k ,$$

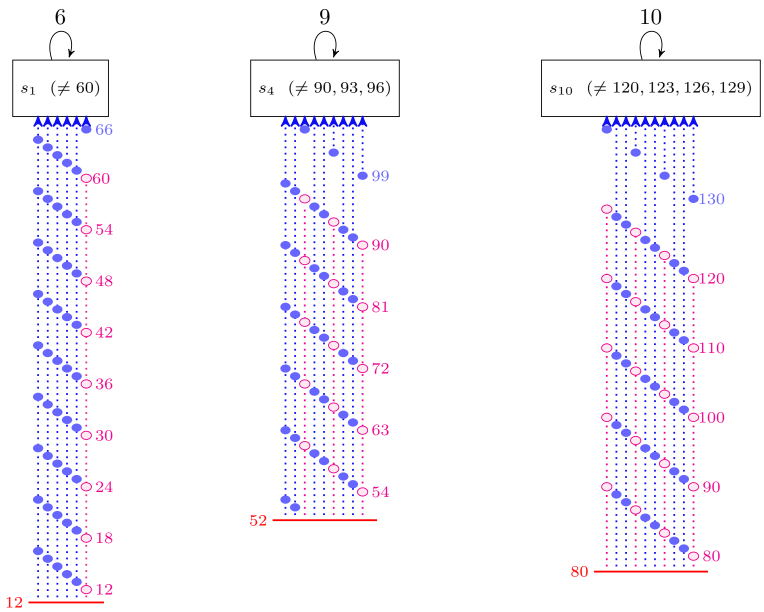


Figure 3 We focus on states s_1, s_4, and s_{10} in the 1-VASS in Figure 2, each of which lies on a simple positive cycle. We also indicate which counter values prevent taking the associated positive cycle. For example, state s_4 has the simple cycle γ_{s_4} with $W_{s_4} = 9$ and taking γ_{s_4} from $\{s_4\} \times \{90, 93, 96\}$ is not allowed due to disequality guards along γ_{s_4}. The columns underneath each state represent residue classes of that state in Conf_+. We colour all unbounded chains in blue and all bounded chains in pink; thus all blue configurations form the set U_0.

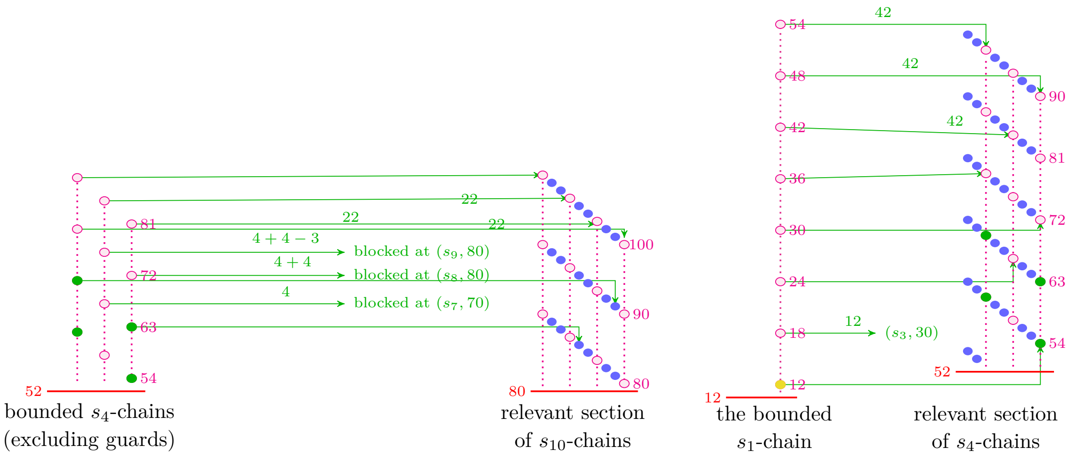
where $\pi_0, \ldots, \pi_k$ are primitive paths and $n_1, \ldots, n_k$ are non-negative integers. Moreover, we give a polynomial bound on the length of the π_i and the magnitude of k in terms of the size of the underlying 1-VASS (in general, the exponents n_i may be exponential in the size of the 1-VASS). We also show how to detect the existence of such a path in polynomial time.

Recall the structure of Conf as a partially ordered set. We will use standard order-theoretic terminology and notation to refer to sets of configurations: in particular given sets of configurations $S, S' \subseteq \mathit{Conf}$, we say that S is *downward closed in* S' if for all $(q, z) \in S \cap S'$ and $(q, z') \in S'$ with $z' \leq z$, we have $(q, z') \in S$.

3.2 Inductive Characterization of Conf_∞

We now give an inductive backward-reachability construction of the set of all configurations in Conf_+ that can reach an unbounded chain. Since unbounded configurations can, in particular, reach unbounded chains, this set is exactly Conf_∞.

In order for our inductive construction to converge in a polynomial number of steps, we essentially consider meta-transitions of the form $\gamma_q^k \cdot \pi$ for γ_q a simple cycle, $k \in \mathbb{N}$, and π a primitive path. Formally, we define an increasing sequence $U_0 \subseteq U_1 \subseteq U_2 \subseteq \cdots$ of subsets of Conf_+ such that $\bigcup_{n \in \mathbb{N}} U_n = \mathit{Conf}_\infty$. Define U_0 to be the union of the collection of unbounded chains. Given $n \in \mathbb{N}$ we inductively construct U_{n+1} as follows. First, define $U'_n \subseteq \mathit{Conf}_+$ as the set of configurations $(q, z) \notin U_n$ whose distance to U_n is minimal among all configurations in $\mathit{Conf}_+ \setminus U_n$ (here the distance of a configuration (q, z) to U_n is the length of the shortest valid run from (q, z) to U_n). Now define $U_{n+1} \subseteq \mathit{Conf}_+$ to be the smallest set such that $U_n, U'_n \subseteq U_{n+1}$ and $U_{n+1} \cap C$ is downward closed in every chain C. Then $\bigcup_{n \in \mathbb{N}} U_n$ is the set of configurations in Conf_+ that can reach an unbounded chain



(a) The set U_1 is obtained from U_0 in Figure 3. **(b)** The set U_2.

Figure 4 The sets U_1 and U_2 of the running example. The blue configurations are in U_0; green ones are in $U_1 \setminus U_0$; yellow one is in $U_2 \setminus U_1$. The pink configurations are in $\mathit{Conf}_+ \setminus U_1$ and $\mathit{Conf}_+ \setminus U_2$, respectively. While computing U_1, the green configurations $(s_4, 63)$ and $(s_4, 69)$ take the primitive path $\pi = s_4, s_7, s_8, s_9, s_{10}$ to U_0. In all other pink configurations in s_4-chains, although enabled, the path π either hits a guard or ends in $(s_{10}, z) \in \mathit{Conf}_+ \setminus U_1$.

which, as noted above, is equal to Conf_∞.

▶ Remark 8. By definition, a shortest run from a configuration $(q, z) \in U'_{n+1} \setminus U_n$ to U_n has no internal configurations in Conf_+, and is therefore primitive.

▶ **Example 9.** Figure 3 indicates the set U_0 for the running example. Note that U_0 contains all trivial residue classes. Observe that $U'_0 = \{(s_4, 63), (s_4, 69)\}$; see Figure 4a. These two configurations belong to two distinct chains. The downward closure of $\{(s_4, 63)\}$ in its chain is $\{s_4\} \times \{54, 63\}$, and the downward closure of $\{(s_4, 69)\}$ in its chain is $\{s_4\} \times \{60, 69\}$. We have that $U_1 = U_0 \cup (\{s_4\} \times \{54, 60, 63, 69\})$. The second iteration to compute U_2 only adds the configuration $(s_1, 12)$ to U_1; see Figure 4b. The sequence stabilizes in this iteration.

3.3 The Structure of Conf_∞

In this section we analyze the structure of Conf_∞, based on its inductive characterization. This analysis will be key in obtaining a polynomial-time algorithm to compute Conf_∞.

The guiding intuition is that for all n the set U_n is *almost upward closed* in each residue class R. By this we mean that if (q, z) is the least configuration in $R \cap U_n$, then all but polynomially many configurations of R above (q, z) are also in U_n. More specifically, we show that for any bounded chain C in R that lies above (q, z), although the number of configurations in C may be exponential in $|Q|$, the size of $C \setminus U_n$ is bounded by a polynomial in $|Q|$. (Note here that the unique unbounded chain in R is contained in U_0 and hence is contained in U_n for all $n \in \mathbb{N}$.) Using this observation, we provide a polynomial bound on the number of iterations until the inductive construction converges. Indeed, in every iteration, unless a fixed point has been reached, there must exist some bounded chain C such that the size of $C \setminus U_n$ strictly decreases. After showing that $C \setminus U_n$ is of polynomial size, we obtain a polynomial bound on the number of iterations until U_n converges by Remark 6.

We start by characterizing the paths between chains.

▶ **Proposition 10.** *Let $(q,z),(q',z') \in Conf_+$ and let $(q,z) \xrightarrow{\pi} (q',z')$ be a (not necessarily valid) run such that π is a primitive path. Then there exists a run $(q,z) \xrightarrow{\pi'} (q',z'')$ of length at most $|Q|^2+2$ such that*

1. $\mathrm{pmin}(\pi') \geq \mathrm{pmin}(\pi)$,
2. *$z'' \geq z'$, and*
3. *the q'-residue class of (q',z'') is either trivial or identical to that of (q',z').*

Given a q-residue class R, in general U_n is not an upward closed subset of R. The following definitions are intended to measure the defect of U_n in this regard.

We say that a bounded chain C that is contained in a residue class R is *n-active* if there exists a configuration in $U_n \cap R$ that lies below some configuration in C. Let C be an n-active chain. Recall that U_n is downward closed in C and hence $C \setminus U_n$ is upward closed in C. Suppose that $C \setminus U_n$ is non-empty, write $m_1 := \min\{x : (q,x) \in C \setminus U_n\}$ and $m_2 := \max\{x : (q,x) \in C \setminus U_n\}$, and define

$$\delta_n(C) := \{(q,x) \in Conf_+ : m_1 \leq x \leq m_2 \text{ and } (q,x) \notin U_n\}.$$

Thus $\delta_n(C)$ contains all configurations in $C \setminus U_n$, as well as all configurations "between" elements of $C \setminus U_n$, apart from those that are themselves in U_n. If $C \setminus U_n = \emptyset$ then we define $\delta_n(C) := \emptyset$. Finally for a residue class R we write

$$\delta_n(R) := \bigcup \{\delta_n(C) : C \subseteq R \text{ an } n\text{-active chain}\} . \tag{1}$$

For $(q, x_{\min})$ the least element in $R \cap U_n$ we have that $|\{(q,x) \in R \setminus U_n : x_{\min} \leq x\}| \leq |\delta_n(R)|$.

▶ **Example 11.** In Figure 4a consider the chain $C := \{s_4\} \times \{54, 63, 72, 81\}$, which is 1-active as $(s_4, 54) \in U_1$. Since $C \setminus U_1 = \{s_4\} \times \{72, 81\}$ we have that $\delta_1(C) = \{s_4\} \times \{72, 75, 78, 81\}$.

▶ **Lemma 12.** *For all $n \in \mathbb{N}$ and every chain C we have that $|\delta_n(C)| \leq |Q| \cdot |C \setminus U_n|$.*

We now come to the central technical part of the paper, controlling the growth of $\delta_n(R)$ as a function of n:

▶ **Lemma 13.** *There exists a polynomial poly_2 such that for each residue class R and all $n \in \mathbb{N}$ we have $|\delta_{n+1}(R)| \leq \max\{|\delta_n(R')| : R' \text{ a residue class}\} + \mathrm{poly}_2(|Q|)$ if R contains a chain that is $(n+1)$-active but not n-active.*

Before proceeding to prove Lemma 13, we demonstrate the underlying intuition. Consider a configuration $(q,z) \in R \cap U'_{n+1}$ that has a primitive path π to a configuration $(q',z') \in U_n$. To prove Lemma 13, we argue that π lifts to a valid run from a "dense" subset of configurations in $\{(q,z'') \in R : z'' \geq z\}$. There are two main cases in this argument based on whether one of the larger configurations in the chain induces a valid run ending in a trivial residue class.

▶ **Example 14.** The first case occurs in obtaining U_1 from U_0 in the running example; see Figure 4a. Consider the chain $C := \{s_4\} \times \{54, 63, 72, 81\}$. The primitive path $s_4, s_7, s_8, s_9, s_{10}$ from the largest configuration $(s_4, 81)$ in C leads to a non-trivial s_{10}-residue class (out of U_0). However, one among the n-next largest configurations in C, for $n = |\mathrm{blocked}(s_4, s_7, s_8, s_9, s_{10})| \cdot |Q|$, lifts to a valid run to a trivial s_{10}-residue class. In the example, this is the case for $(s_4, 63)$. The second case occurs in obtaining U_2 from U_1 in the running example; see Figure 4b. Consider the chain $C' := \{s_1\} \times \{12, 18, 24, \cdots, 54\}$. The primitive path s_1, s_3, s_4, from none of the configurations in this chain, ends in a trivial s_4-residue class. However, we provide a subtle argument to bound $|C' \setminus U_2|$ with $|\delta_1(C)| + \mathrm{poly}_2(|Q|)$.

Proof of Lemma 13. Pick the minimal element $(q, z_0) \in R \cap U'_{n+1}$. Moreover, let $(q', z') \in U_n$ and $(q, z_0) \xrightarrow{\pi} (q', z')$ be such that π is a shortest run from (q, z_0) to U_n. By Remark 8, π is a primitive path. By Proposition 10 there is a run $(q, z_0) \xrightarrow{\pi'} (q', z'')$, for some $z'' \geq z'$, such that π' has length at most $|Q|^2 + 2$, and the residue class R' of (q', z'') is either *trivial* or the same as the residue class of (q', z').

Note that we do not claim that $(q', z'') \in U_n$, nor that π' lifts to a valid run. In what follows we will argue that if there are more than some polynomial number of configurations above (q, z_0) in $C \setminus U'_{n+1}$, where C is an $(n+1)$-active chain of R, then π' does lift to a valid run from one of them. Moreover, the run leads to some configuration in the same residue class as (q', z') or to a trivial residue class. Observe that, intuitively, this means we "pump" γ_q before taking π' so if we wanted to reach the same residue class as (q', z') we would need some nonnegative integer c such that

$$z_0 + W_q \cdot c + \text{weight}(\pi') \equiv z_0 + \text{weight}(\pi') \pmod{W_{q'}}.$$

Based on this intuition, we now identify two cases according to the order of W_q in the group $\mathbb{Z}/\mathbb{Z}W_{q'}$ of integers modulo $W_{q'}$, which is $\frac{W_{q'}}{\gcd(W_q, W_{q'})}$. Recall that this quantity is the smallest integer $c \geq 1$ such that $W_q \cdot c \equiv 0 \pmod{W_{q'}}$.

Case (i): $\frac{W_{q'}}{\gcd(W_q, W_{q'})} > |Q|$. We first show that $|C \setminus U_{n+1}| \leq (|Q|^2 + 2)(|Q| + 1)$ for every $(n+1)$-active chain C in R.

Let C be an $(n+1)$-active chain of R and suppose for a contradiction that $|C \setminus U_{n+1}| > (|Q|^2 + 2)(|Q| + 1)$. Since C is $(n+1)$-active, for every configuration $(q, z) \in C \setminus U_{n+1}$ we have $z \geq z_0$. Further, since $\text{pmin}(\pi') + z_0 \geq 0$, π' can only be blocked on a configuration due to a violation of a disequality guard. Since the length of π' is at most $|Q|^2 + 2$, it follows that at most $|Q|^2 + 2$ elements of $C \setminus U_{n+1}$ lie in $\{q\} \times \text{blocked}(\pi')$.

Recall that $C \setminus U_{n+1}$ is upward closed in C, so by the assumption that $|C \setminus U_{n+1}| > (|Q|^2+2)(|Q|+1)$, there exists a set $S := \{(q, z_1 + iW_q) : 0 \leq i \leq |Q|\}$ of $|Q|+1$ "consecutive" elements of $C \setminus U_{n+1}$, for some z_1, such that no element of S lies in $\{q\} \times \text{blocked}(\pi')$. Then π' lifts to a valid run from each element of S. Moreover, since the order of W_q in $\mathbb{Z}/\mathbb{Z}W_{q'}$ is assumed to be greater than $|Q|$, the images of the elements of S, after following π', lie in pairwise distinct q'-residue classes. But the number of non-trivial q'-residue classes is at most $|Q|$ and hence some configuration in S has a run over π' to a trivial q'-residue class and hence to U_n. But then such a configuration lies in U_{n+1}, which is a contradiction.

We conclude that $|C \setminus U_{n+1}| \leq (|Q|^2 + 2)(|Q| + 1)$ for every $(n+1)$-active chain C in R. But then $|\delta_{n+1}(C)| \leq |Q|(|Q|^2 + 2)(|Q| + 1)$ by Lemma 12. Finally, since R comprises at most $2|Q|$ bounded chains by Remark 6, we have that $|\delta_{n+1}(R)| \leq 2|Q|^2(|Q|^2 + 2)(|Q| + 1)$.

Case (ii): $\frac{W_{q'}}{\gcd(W_q, W_{q'})} \leq |Q|$. For the residue classes R and R' as above, define an injective partial mapping $\Phi : \delta_{n+1}(R) \to \delta_n(R')$ by $\Phi(q, x) = (q', x')$ if and only if $x' = x + \text{weight}(\pi')$ and $(q', x') \in \delta_n(R')$. We will prove that Φ is defined on all but $\text{poly}_3(|Q|)$ many configurations in $\delta_{n+1}(R)$, for some polynomial poly_3, thereby showing that $|\delta_{n+1}(R)| \leq |\delta_n(R')| + \text{poly}_3(|Q|)$. To this end, it suffices to show that Φ is defined on all but $\text{poly}_4(|Q|)$ many configurations in $\delta_{n+1}(C)$ for every $(n+1)$-active chain C in R, for some polynomial poly_4.

Let C be an $(n+1)$-active chain in R and let $C_1, \ldots, C_s$ be a list, given in increasing order, of the chains in R' that are mapped into by Φ from some configuration in $\delta_{n+1}(C)$. Then $C_1, \ldots, C_s$ are all n-active (as they are above $(q', z') \in U_n$). For $i \in \{1, \ldots, s\}$, write $(q, x^{(i)}_{\min})$

for the minimum configuration in $\delta_{n+1}(C)$ that is mapped by Φ to C_i and write $(q, x^{(i)}_{\max})$ for the maximum configuration in $\delta_{n+1}(C)$ that is mapped to C_i. Then for each $i = 1, \ldots, s$, every configuration $(q, x) \in \delta_{n+1}(C)$ such that $x^{(i)}_{\min} \le x \le x^{(i)}_{\max}$ and $x \notin \times\text{blocked}(\pi')$ is mapped by Φ to $\delta_n(R')$. Thus, writing $(q, x_{\max})$ and $(q, x_{\min})$ respectively for maximum and minimum configurations in $\delta_{n+1}(C)$, we have that Φ is defined on all non-blocked elements of $\delta_{n+1}(C)$ lying outside the set below.

$$\left\{ (q,x) \in \delta_{n+1}(C) \;\middle|\; x \in \left(x^{(s)}_{\max}, x_{\max}\right] \cup \left[x_{\min}, x^{(1)}_{\min}\right) \cup \bigcup_{i=1}^{s-1} \left(x^{(i)}_{\max}, x^{(i+1)}_{\min}\right) \right\} \tag{2}$$

Since blocked(π') contains at most $|Q|^2 + 2$ elements, it remains to prove that the set (2) has polynomial cardinality. We claim its size is at most $(2|Q| + 1) \cdot \text{poly}_5(|Q|)$, for some polynomial poly$_5$. For this it will suffice to show that any sub-interval I of $\delta_{n+1}(C)$ of the form $\{(q, x) \in \delta_{n+1}(C) : a \le x \le b\}$, where $a, b \ge x_{\min}$, and such that it does not meet the domain of Φ, has cardinality at most $\text{poly}_5(|Q|)$. (Indeed, note that (2) is a union of at most $2|Q| + 1$ such intervals since there are at most $2|Q|$ chains in R by Remark 6.)

Let $\text{poly}_6(x) := (x^2 + 2)(x + 1) + 1$. Since blocked$(\pi')$ has cardinality at most $|Q|^2 + 2$, if we take $\text{poly}_6(|Q|)$ consecutive elements of $C \setminus U_{n+1}$ then there are at least $|Q|+1$ consecutive elements that lie outside $\{q\} \times \text{blocked}(\pi')$ and at least one of these elements—say (q, x)—has a valid run over π' to the residue class R' by the assumption that $\frac{W_{q'}}{\gcd(W_q, W_{q'})} \le |Q|$. Since $(q, x) \notin U_{n+1}$ we have that $(q', x + \text{weight}(\pi')) \notin U_n$ and hence (q, x) is in the domain of Φ. We conclude that any sequence of at least $\text{poly}_6(|Q|)$ consecutive elements of $C \setminus U_{n+1}$ meets the domain of Φ. Hence any sub-interval I, as defined above, contains at most $\text{poly}_6(|Q|)$ elements of $C \setminus U_{n+1}$ and, by Lemma 12, contains at most $|Q| \cdot \text{poly}_6(|Q|)$ elements in total. ◀

Proposition 15 follows from Lemma 13 by induction, as follows.

▶ **Proposition 15.** *There exists a polynomial* poly$_1$ *such that for each residue class R and all $n \in \mathbb{N}$ we have $|\delta_n(R)| \le \text{poly}_1(|Q|)$.*

Proof. Let α_n be the number of chains in Conf_+ that are n-active. Since n-active chains are by definition bounded, we have that $\alpha_n \le 2|Q|^2$ for all $n \in \mathbb{N}$ (see Remark 6). We argue by induction on n that $|\delta_n(R)| \le \alpha_n \cdot \text{poly}_2(|Q|)$ for all $n \in \mathbb{N}$ and all residue classes R. We conclude that $|\delta_n(R)| \le 2|Q|^2 \cdot \text{poly}_2(|Q|)$.

The base case is trivial as there are no 0-active chains and $\delta_0(R)$ is empty for all residue classes. The induction step has two cases. First, suppose that $\alpha_{n+1} = \alpha_n$, i.e., all chains in Conf_+ that are $(n + 1)$-active were already n-active. Since $U_n \subseteq U_{n+1}$, we have that $\delta_{n+1}(C) \subseteq \delta_n(C)$ for all chains C in R. We conclude that $\delta_{n+1}(R) \subseteq \delta_n(R)$ and so $|\delta_{n+1}(R)| \le |\delta_n(R)|$. Since $|\delta_n(R)| \le \alpha_n \cdot \text{poly}_2(|Q|)$ by induction hypothesis, and $\alpha_n = \alpha_{n+1}$ we get that $|\delta_{n+1}(R)| \le \alpha_{n+1} \cdot \text{poly}_2(|Q|)$.

The second case is that $\alpha_{n+1} > \alpha_n$. Then by Lemma 13 we have $|\delta_{n+1}(R)| \le \max\{|\delta_n(R')| : R' \text{ a residue class}\} + \text{poly}_2(|Q|)$. Since the right-hand side of the latter is at most $\le \alpha_n \cdot \text{poly}_2(|Q|) + \text{poly}_2(|Q|)$, by induction hypothesis, and $\alpha_{n+1} > \alpha_n$ we get that $|\delta_{n+1}(R)| \le \alpha_{n+1} \cdot \text{poly}_2(|Q|)$. ◀

As a consequence of Proposition 15 we have:

▶ **Corollary 16.** *The sequence $(U_n)_{n\in\mathbb{N}}$ stabilizes in at most $\text{poly}_1(|Q|)$ steps.*

3.4 Computing $Conf_\infty$ and Deciding Unboundedness

In this section we show how to compute $Conf_\infty$ in polynomial time and how to decide in polynomial time whether the initial configuration $(s, 0)$ can reach $Conf_\infty$.

We start by showing that if a configuration can reach U_n via a primitive run, then it can also reach U_n via a polynomial-length run (see Appendix G for the proof).

▶ **Proposition 17.** *There exists a polynomial* poly_7 *such that the following holds. Let* $(q, z), (q', z') \in Conf_+$ *and let* $(q, z) \xrightarrow{\pi} (q', z')$ *be a valid run such that* $(q', z') \in U_n$ *and* π *is primitive. Then there is a valid run* $(q, z) \xrightarrow{\pi'} (q', z'')$ *such that* $(q', z'') \in U_n$ *and* π' *has length at most* $\mathrm{poly}_7(|Q|)$.

Recall that U'_{n+1} consists of all configurations in $Conf_+$ with minimal distance to U_n. Combining Remark 8 and Proposition 17, we have that the minimal distance from a configuration $(q, z) \in U'_{n+1} \setminus U_n$ to U_n is at most $\mathrm{poly}_7(|Q|)$. It follows that we can restrict the search for configurations that can reach U_n, to those within a polynomially-bounded distance to U_n. By itself this is not sufficient to obtain a polynomial-time algorithm to decide whether U_n is reachable. However, using our analysis of the structure of U_n in Section 3.3, we are able to formulate the bounded reachability problem above in a form that admits a polynomial-time algorithm.

Specifically, we consider the *Bounded Coverability problem with a Disequality Objective*: Given as input a 1-VASS $\mathcal{V} = (Q, D, \Delta, w)$ with a distinguished state q_f, a positive integer L (written in unary), an initial configuration (q_0, x_0), and a coverability objective of the form

$$O = \left\{ (q_f, x) \mid x \geq \ell \wedge \bigwedge_{i=1}^{m} (x \not\equiv a_i \bmod W) \wedge \bigwedge_{i=1}^{n} (x \neq b_i) \right\}, \tag{3}$$

where ℓ, W and the a_i and b_i are non-negative integers given in binary, decide whether O is reachable from (q_0, x_0) via a valid run of length at most L.

▶ **Proposition 18.** *The Bounded Coverability problem with a Disequality Objective is decidable in polynomial time.*

We now show how to compute $Conf_\infty$ in polynomial time. By Corollary 16, the sequence $\{U_n\}_{n\in\mathbb{N}}$ converges in at most $\mathrm{poly}_1(|Q|)$ steps. It remains to show how to compute U_{n+1} from U_n in polynomial time for each n.

Recall that all unbounded chains are contained in U_0 and hence are contained in U_n for all n. Recall also that the total number of bounded chains is at most $2|Q|$ and that U_n is downward closed in each bounded chain. Thus U_n is determined by giving, for every bounded chain C such that $U_n \cap C \neq \emptyset$, the maximum configuration in $U_n \cap C$. In particular, U_n can be described in space polynomial in the description of the given 1-VASS.

Recall that U_{n+1} is obtained from U_n by adding the configurations in $\mathrm{Conf}_+ \setminus U_n$ that have minimum distance to U_n and then closing downward in each bounded chain. By Remark 8 and Proposition 17, a configuration in $\mathrm{Conf}_+ \setminus U_n$ that has minimum distance to U_n has distance at most $\mathrm{poly}_7(|Q|)$. The idea to compute U_{n+1} from U_n is as follows:

For each bounded chain C, and each configuration $(q, x) \in C \setminus U_n$ that is among the top $\mathrm{poly}_1(|Q|)$ configurations in C, we determine the distance of (q, x) to U_n up to a bound of $\mathrm{poly}_7(|Q|)$. To do this we use the procedure described in Proposition 18, having first written U_n as a polynomial-size union of sets of the form (3)—see below for details. The reason that it suffices to look only among the top $\mathrm{poly}_1(|Q|)$ configurations in each bounded chain is because we know from Proposition 15 that $|C \setminus U_{n+1}| \le \mathrm{poly}_1(|Q|)$ for every $(n+1)$-active chain C.

We next show how to decompose U_n into a polynomial union of sets of the form (3) in order to apply Proposition 18. Fixing $q \in Q_+$, let $R_1, \ldots, R_m$ be a list of the non-trivial q-residue classes and for each $i \in \{1, \ldots, m\}$, write a_i for the corresponding residue modulo W_q and define $\ell_i := \min(R_i \cap U_n)$. Moreover, let $b_1, \ldots, b_k$ be a list of the counter values x such that $x \ge \ell_i$ and $(q, x) \in R_i \setminus U_n$ for some i. Note that $m \le |Q|$ and $k \le m\,\mathrm{poly}_1(|Q|)$. We decompose the set of configurations $\{(q, z) \in U_n\}$ into the following two components:

1. $\{(q, z) : z \ge \mathrm{pmin}(\gamma_q) \wedge \bigwedge_{i=1}^m z \not\equiv a_i \pmod{W_q}\}$, i.e., all configurations in trivial q-residue classes,
2. for all $j \in \{1, \ldots, m\}$, the set $\{(q, z) : z \ge \ell_j \wedge \bigwedge_{i:i \ne j} z \not\equiv a_i \pmod{W_q} \wedge \bigwedge_{i=1}^k z \ne b_i\}$, which includes $R_j \cap U_n$ for the non-trivial residue class R_j.

Finally, it remains to decide whether the configuration $(s, 0)$ is unbounded. By Proposition 4, $(s, 0)$ is unbounded if and only if it can reach Conf_∞. Now a shortest run from $(s, 0)$ to Conf_∞ is necessarily primitive: if an internal configuration in such a run lies in Conf_+ then it is also in Conf_∞—a contradiction. By Proposition 17, a shortest run from $(s, 0)$ to Conf_∞ has length at most $\mathrm{poly}_7(|Q|)$. Thus we can decide whether such a run exists in polynomial time using Proposition 18. In conclusion we have

▶ **Theorem 19.** *The Unboundedness Problem and the Coverability Problem for 1-VASS with disequality tests are decidable in polynomial time.*

4 Unboundedness for 1-VASS

In this section we show that the Unboundedness Problem for 1-VASS (i.e., with no disequality tests) is in NC^2. Recall that NC^i is the class of decision problems solvable in time $O(\log^i n)$, with n the size of the input, on a parallel computer with a polynomial number of processors [13, 1].

Let $\mathcal{V} = (Q, \Delta, w)$ be a 1-VASS with a distinguished state $s \in Q$. We want to decide whether the configuration $(s, 0)$ is unbounded. Since $\mathcal{V}$ has no disequality tests, deleting a negative-weight or zero-weight cycle that appears as an infix of a valid run yields another valid run. It follows that $(s, 0)$ is unbounded if and only if there is a valid run from $(s, 0)$ consisting of a simple path (of length at most $|Q|$) followed by a positive-weight simple cycle (again, of length at most $|Q|$). We call such a run a *lasso.*

Let $\mathcal{V} = (Q, \Delta, w)$ be a 1-VASS and let $\pi = q_1, \ldots, q_n$ be a path in $\mathcal{V}$. Recall that a (possibly empty) prefix of π is said to be *minimal* if it has minimal weight among all prefixes of π. Likewise a (possibly empty) suffix of π is said to be *maximal* if it has maximal weight among all suffixes. It is clear that $q_1, \ldots, q_m$ is a minimal prefix of π if and only if $q_m, \ldots, q_n$ is a maximal suffix. In such a case let us call q_m a *nadir* of π (the nadir is the lowest point reached in any run over π). Recall that $\mathrm{pmin}(\pi)$ is the weight of a minimal prefix of π; correspondingly we define $\mathrm{smax}(\pi)$ to be the weight of a maximal suffix.

Given paths π and π', say that π is *dominated* by π' if $\mathrm{pmin}(\pi) \le \mathrm{pmin}(\pi')$ and $\mathrm{smax}(\pi) \le \mathrm{smax}(\pi')$. Observe that if π is dominated by π' then $\mathrm{weight}(\pi) \le \mathrm{weight}(\pi')$.

▶ **Example 20.** In Figure 5, the path s_0, s_1, s_4 dominates s_0, s_2, s_4. However, despite it being the case that $\mathrm{weight}(s_0, s_3, s_4) > \mathrm{weight}(s_0, s_2, s_4)$, s_0, s_3, s_4 does not dominate s_0, s_2, s_4 since the weight of a minimal prefix of the former is smaller than that of the latter.

Fix two states $p, q \in Q$ and let P be a set of p-q paths. We say that a set P' of p-q paths is a *Pareto set* for P if for every $\pi \in P$ there exists $\pi' \in P'$ such that π is dominated by π'.

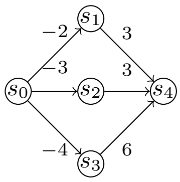


Figure 5 The topmost path dominates the middle one; the bottom path dominates no other path

We observe some simple properties of Pareto sets:

▶ **Lemma 21.** *Let $p, q, r \in Q$. Then all of the following statements hold:*

1. *If P_1, P_2, P_3 are sets of p-q paths such that P_1 is a Pareto set of P_2 and P_2 is a Pareto set of P_3, then P_1 is a Pareto set of P_3.*
2. *If P, R are sets of p-q paths with respective Pareto sets P', R', then $P' \cup R'$ is a Pareto set for $P \cup R$*
3. *If P is a set of p-q paths and R is a set of q-r paths with respective Pareto sets P', R', then $P' \cdot R'$ is a Pareto set of $P \cdot R$.*

▶ **Proposition 22.** *Let $p, q \in Q$. Then every set P of p-q paths of length at most k has a Pareto set P' of cardinality at most $|Q|$ such that each path in P' has length at most $2k$. Moreover such a set P' can be computed from P in NC^1.*

An NC^2 Upper Bound

▶ **Theorem 23.** *The Unboundedness Problem and the Coverability Problem for 1-VASS are decidable in NC^2.*

Proof. By Lemma 1, it will suffice to show that Unboundedness is in NC^2.

Let $\mathcal{V} = (Q, \Delta, w)$ be a 1-VASS. Given $p, q \in Q$ and $m \in \mathbb{N}$, denote by $\mathrm{Paths}_{p,q,m}$ the set of all p-q paths in $\mathcal{V}$ of length at most m.

Given a state $s \in Q$, recall that $(s, 0)$ is unbounded if and only if there exists a lasso run that starts at $(s, 0)$. To determine the existence of such a run we compute a Pareto set P_q for $\mathrm{Paths}_{s,q,|Q|}$ and a Pareto set P'_q for $\mathrm{Paths}_{q,q,|Q|}$ for every state $q \in Q$. Having done this we look for $q \in Q$ and paths $\pi \in P_q$ and $\pi' \in P'_q$ such that $\pi \cdot \pi'$ induces a valid run from $(s, 0)$ and π' has positive weight.

It remains to show how to compute a Pareto set of $\mathrm{Paths}_{p,q,|Q|}$ for all pairs of states $p, q \in Q$ (together with the values $\mathrm{weight}(\pi)$ and $\mathrm{pmin}(\pi)$ for every path π in the Pareto set) in NC^2.

For $k = 1, \ldots, \lceil \log |Q| \rceil$, we show how to compute a family $\mathcal{P}_k = \{P_{p,q,k}\}_{p,q \in Q}$ such that for all $p, q \in Q$:

1. $P_{p,q,k}$ is a Pareto set for $\mathrm{Paths}_{p,q,2^k}$;
2. $P_{p,q,k} \subseteq \mathrm{Paths}_{p,q,4^k}$;
3. $|P_{p,q,k}| \le |Q|$.

By Item 1, if $k = \lceil \log |Q| \rceil$ then $P_{p,q,k}$ is a Pareto set for $\mathrm{Paths}_{p,q,|Q|}$. (Note that for $k = \lceil \log |Q| \rceil$, $\mathcal{P}_k$ consists of paths of length at most $|Q|^2$.)

The construction of $\mathcal{P}_k$ is by induction on k. Suppose we have computed $\mathcal{P}_k$ with Properties 1-3 above. Fix $p, q \in Q$. In order to compute $P_{p,q,k+1}$ we observe that

$$P := \{\pi_1 \cdot \pi_2 : \exists r \in Q(\pi_1 \in P_{p,r,k} \wedge \pi_2 \in P_{r,q,k})\} \tag{4}$$

is a Pareto set for $\mathrm{Paths}_{p,q,2^{k+1}}$ by Items 2 and 3 of Lemma 21. Applying Proposition 22, we obtain a Pareto set P' for P of cardinality at most $|Q|$. By Item 1 of Lemma 21, P' is a

Pareto set for $\text{Paths}_{p,q,2^{k+1}}$. Finally, it is clear from the length bound in Proposition 22 that all paths in P' have length at most 4^{k+1}. Thus we define $P_{p,q,k+1} := P'$.

It remains to establish the NC^2 complexity bound for computing $\mathcal{P}_{\lceil \log |Q| \rceil}$. For this it suffices to show that for all k the computation of $\mathcal{P}_{k+1}$ from $\mathcal{P}_k$ can be carried out in NC^1. But we may compute each set $P_{p,q,k+1}$ in parallel (over $p, q \in Q$), and the computation of each such set can be done in NC^1 by Proposition 22. ◀

5 Conclusion

We have shown that control-state reachability for 1-VASS with disequality tests can be solved in polynomial time. The complexity of reaching a given *configuration* in this model is open (being equivalent to control-state reachability in the presence of both equality and disequality tests), lying between NP and PSPACE. For multi-dimensional VASS with disequality tests, the classical argument of Rackoff [14] easily generalises to show that control-state reachability remains in EXPSPACE. By contrast, decidability of reachability is open to the best of our knowledge. For comparison, recall that without disequality tests reachability is decidable but non-elementary [4].

References

1. Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. URL: `http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264`.
2. Benedikt Bollig, Karin Quaas, and Arnaud Sangnier. The complexity of flat freeze LTL. In *28th International Conference on Concurrency Theory, CONCUR*, volume 85 of *LIPIcs*, pages 33:1–33:16, 2017.
3. Daniel Bundala and Joël Ouaknine. On parametric timed automata and one-counter machines. *Inf. Comput.*, 253:272–303, 2017.
4. Wojciech Czerwinski, Slawomir Lasota, Ranko Lazic, Jérôme Leroux, and Filip Mazowiecki. The reachability problem for petri nets is not elementary. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019*, pages 24–33. ACM, 2019.
5. S. Demri, R. Lazic, and A. Sangnier. Model checking memoryful linear-time logics over one-counter automata. *Theor. Comput. Sci.*, 411(22-24):2298–2316, 2010.
6. John Fearnley and Marcin Jurdzinski. Reachability in two-clock timed automata is pspace-complete. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings*, volume 7966 of *Lecture Notes in Computer Science*, pages 212–223. Springer, 2013.
7. Alain Finkel, Stefan Göller, and Christoph Haase. Reachability in register machines with polynomial updates. In *Mathematical Foundations of Computer Science 2013 - 38th International Symposium, MFCS*, volume 8087 of *Lecture Notes in Computer Science*, pages 409–420. Springer, 2013.
8. Stefan Göller, Christoph Haase, Joël Ouaknine, and James Worrell. Model checking succinct and parametric one-counter automata. In *Automata, Languages and Programming, 37th International Colloquium, ICALP*, volume 6199 of *Lecture Notes in Computer Science*, pages 575–586. Springer, 2010.
9. C. Haase, S. Kreutzer, J. Ouaknine, and J. Worrell. Reachability in succinct and parametric one-counter automata. In *Proceedings of CONCUR*, volume 5710 of *LNCS*, pages 369–383. Springer, 2009.
10. Neil Immerman. *Descriptive complexity*. Graduate texts in computer science. Springer, 1999. URL: `https://doi.org/10.1007/978-1-4612-0539-5`, `doi:10.1007/978-1-4612-0539-5`.
11. P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for *AC*-like equational theories with homomorphisms. In *Research Report LSV-04-16, LSV, ENS de Cachan*, 2004.
12. Matti Nykänen and Esko Ukkonen. The exact path length problem. *J. Algorithms*, 42(1):41–53, 2002.
13. Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.
14. Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theor. Comput. Sci.*, 6:223–231, 1978.
15. Louis E. Rosier and Hsu-Chun Yen. A multiparameter analysis of the boundedness problem for vector addition systems. *J. Comput. Syst. Sci.*, 32(1):105–135, 1986.
16. Heribert Vollmer. *Introduction to Circuit Complexity - A Uniform Approach*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 1999. URL: `https://doi.org/10.1007/978-3-662-03927-4`, `doi:10.1007/978-3-662-03927-4`.

A Proof of the reduction in Figure 1

Let us recall that for every value $u \in \mathbb{N}$, the assignment $\mathrm{val}_u : \{X_1, \dots, X_m\} \to \{0,1\}$ is defined by $\mathrm{val}_u(X_i) = 1$ if and only if $p_i \mid u$. For convenience, define the domain $D_s \subseteq \mathbb{N}$ containing all allowable counter values in state s (exclude all disequality guards on s).

The key observation is the following: let $u \in \{0, \dots, P-1\}$, and consider a clause $C_i = \ell_{i_1} \vee \ell_{i_2} \vee \ell_{i_3}$, where ℓ_{i_j} is a literal of variable X_{i_j}, then val_u satisfies C_i iff there exists some $k \in \mathbb{N}$ such that $u + kp_{i_1}p_{i_2}p_{i_3} \notin D_i$.

Indeed, note that for every $j \in \{1,2,3\}$ and every $k \in \mathbb{N}$ we have that $p_{i_j}|u$ iff $p_{i_j}|u + kp_{i_1}p_{i_2}p_{i_3}$. Recall that $\mathrm{val}_u(X_{i_j}) = 1$ iff $p_{i_j}|u$, and observe that since $u < P$, there exists $k \in \mathbb{N}$ such that $u + kp_{i_1}p_{i_2}p_{i_3} \in \{P, P+1, \dots, P + p_{i_1}p_{i_2}p_{i_3} - 1\}$. We thus have that val_u satisfies C_i iff $\mathrm{val}_{u+kp_{i_1}p_{i_2}p_{i_3}}$ satisfies C_i, iff $u + kp_{i_1}p_{i_2}p_{i_3} \notin D_i$.

Now, assume φ is satisfiable, and let π be a satisfying assignment. We associate with π the number $u = \prod_{j:\pi(X_j)=1} p_j \pmod{P}$ (note that taking modulo P simply means that if the product is exactly P, we take $u = 0$). Clearly $\pi = \mathrm{val}_u$. We claim that (s_0, u) is bounded. Indeed, the only paths possible from (s_0, u) start by choosing a state s_i, and then repeatedly applying the cycle of cost c_i. However, since val_u satisfies all clauses, then by the above, all such paths are blocked by a disequality guard after taking the c_i for k times, for some $k \in \mathbb{N}$ (which depends on i). Thus, (s_0, u) is bounded.

Conversely, assume (s_0, u) is bounded for some value u, we claim that val_u satisfies φ. Indeed, by the same reasoning above, it follows that for every cycle of cost c_i, we have $u + kc_i \notin D_i$ for some $k \in \mathbb{N}$, so val_u satisfies C_i. Since this is true for all clauses, we have that val_u satisfies φ.

We conclude that φ is satisfiable iff some configuration (s_0, u) is bounded, which completes the reduction.

Finally, we note that the reduction indeed takes polynomial time — indeed, the construction clearly has polynomially many states. Also, the first m primes $p_1, \dots, p_m$ can be listed in time polynomial in m, and are representable in polynomially many bits. Therefore, the binary representation of the transition values and the amount of missing elements in the domain of each state are both polynomial.

B Single disequality guards suffice

Given a 1-VASS $\mathcal{V} = (Q, \Delta, D, w)$ with disequality tests, we can assume that for all states q the set D_q is either $\mathbb{N}$ or $\mathbb{N} \setminus \{g\}$ for some $g \in \mathbb{N}$. This assumption is without loss of generality, as a state q with $D_q = \mathbb{N} \setminus \{a_1, \dots, a_n\}$ can be replaced with a sequence of new states $q_1, \cdots, q_n$, connected with 0-weight transitions, such that $D_{q_i} = \mathbb{N} \setminus \{a_i\}$ for $i \in \{1, \dots, n\}$. The transformation yields only a polynomial blow-up in the size of the 1-VASS, and there is a natural correspondence between runs in the original 1-VASS and the modified one.

C Proof of Lemma 1

Consider a 1-VASS $\mathcal{V} = (Q, \Delta, D, w)$ with disequality tests, and let $s, t \in Q$. We reduce the Coverability problem to the Unboundedness problem as follows.

We obtain from $\mathcal{V}$ a new 1-VASS $\mathcal{V}'$ as follows. First, we remove from $\mathcal{V}$ all the states that cannot reach t in the underlying graph. Second, we introduce a new state t' with a self-loop of weight $+1$, that is reachable from t with a transition of weight 0. The output of the reduction is $\mathcal{V}'$ with the distinguished state s.

Recall that reachability in directed graphs can be decided in $\mathsf{NL} \subseteq \mathsf{NC}^2$, and hence this reduction is NC^2-computable.

Henceforth assume that s can reach t in the underlying graph of $\mathcal{V}$ (otherwise s cannot cover t, and the reduction can output a trivial negative instance). We proceed to prove the correctness of the reduction.

First, if $(s, 0)$ can cover t in $\mathcal{V}$, then in particular it can only cover t using states in $\mathcal{V}'$. We now have that $(s, 0)$ is unbounded in $\mathcal{V}'$, by covering t, and then taking the transition to t' and repeating the self loop unboundedly. Note that crucially, there are no disequality guards on t', and therefore once t is reached, we can take the transition to t and repeat the self loop unboundedly.

Conversely, suppose $(s, 0)$ is unbounded in $\mathcal{V}'$, then either there is a valid run in $\mathcal{V}$ from $(s, 0)$ to (t', z) for some z, in which case $(s, 0)$ can cover t in $\mathcal{V}$, or $(s, 0)$ is unbounded already in $\mathcal{V}$ and, moreover, it is unbounded in $\mathcal{V}$ using only states that can reach t in the underlying graph. We claim that in the latter case, $(s, 0)$ can cover t in $\mathcal{V}$. Indeed, from $(s, 0)$ there is a valid run to a configuration (q, z) with z that is large enough, such that a simple path from q to t in the underlying graph lifts to a valid run from (q, z) to (t, z') for some z'. Specifically, taking $z > |Q| \cdot W \cdot G$ where W is the maximal absolute value of the weight of a transition in $\mathcal{V}$, and G is the maximal disequality guard, suffices for such a run.

D Proof of Proposition 4

Clearly if $(s, 0)$ can reach an unbounded configuration in Conf_+ then it is unbounded.

Conversely, if $(s, 0)$ is unbounded, then there is a state q such that for all $z_0 \in \mathbb{N}$, there exist $z, z' \geq z_0$ and a valid run π starting in $(s, 0)$ that visits (q, z) and ends in (q, z'). Thus, there is a positive cycle γ on q. The positive cycle γ on q may not be simple, but it certainly visits a state p with a simple positive cycle γ_p on it. Pick z_0 such that $z_0 > \mathrm{pmin}(\gamma) + x$. for all $x \in \mathrm{blocked}(\gamma_p^\omega)$ (Note that $\mathrm{blocked}(\gamma_p^\omega)$ is finite since γ_p is a positive cycle. The maximum is thus well-defined.) Hence, there is a valid run from $(s, 0)$ to (p, y) where $y > \max(\mathrm{blocked}(\gamma_p^\omega))$. Observe that $(p, y) \in \mathit{Conf}_+$ and it is unbounded.

E Proof of Proposition 10

Suppose that π has length strictly greater than $|Q|^2 + 2$. Then we can find $|Q| + 1$ distinct proper prefixes (i.e. prefixes that are not just the initial state, or the entire path) of π that end in the same state. That is, $|Q|$ proper cycles on the same state. Let $\pi_1, \ldots, \pi_{|Q|+1}$ be a list of these prefixes, given in order of increasing length, and let the corresponding suffixes be $\pi'_1, \ldots, \pi'_{|Q|+1}$. We now consider two cases.

First, suppose that there exist $i < j$ such that $\mathrm{weight}(\pi_i)$ and $\mathrm{weight}(\pi_j)$ have the same residue modulo $W_{q'}$. Then define $\pi' := \pi_i \cdot \pi'_j$. In this case path π' lifts to a run from (q, z) to (q', z'') such that (q', z'') lies in the same q'-residue class as (q', z''). The second case is that the respective residue classes of $\mathrm{weight}(\pi_1), \ldots, \mathrm{weight}(\pi_{|Q|+1})$ modulo $W_{q'}$ are all distinct. Then there exists $i > 1$ such that, defining $\pi' := \pi_1 \cdot \pi'_i$, the path π' lifts to a run from (q, z) to (q', z'') such that (q', z'') lies in a trivial q'-residue class (as there are at most $|Q|$ non-trivial residue classes).

Continuing in this fashion we can recursively remove cycles from the original path π to eventually obtain a path π' that has length at most $|Q|^2 + 2$ and such that Item 3 is satisfied. Consider all maximal infixes that were removed from π to obtain π'. Note that each such infix must necessarily be a cycle as they arise from iteratively removing cycles. Since π was primitive, all of them must have non-positive weight. Hence, Items 1 and 2 also hold[2].

F Proof of Lemma 12

Consider two "consecutive" configurations $(q, z), (q, z + W_q) \in C \setminus U_n$, then all configurations (q, z') for $z \leq z' < z + W_q$ lie in pairwise-distinct q-residue classes. In particular, since there are at most $|Q|$ non-trivial residue classes, and since trivial residue classes are contained in U_0, we have that at most $|Q|$ such elements are in $\delta_n(C)$.

G Proof of Proposition 17

By Proposition 15 we can find a polynomial poly'_7 such that

$$\mathrm{poly}'_7(|Q|) \geq |Q|^2 + |Q| + 3 + \sum_{R \text{ non-trivial}} |\delta_n(R)| \tag{5}$$

for all $n \in \mathbb{N}$.

Set $\mathrm{poly}_7(|Q|) := |Q| \cdot (\mathrm{poly}'_7(|Q|))^2 + |Q|^2 + 4$, and consider a valid, primitive path π such that $\mathrm{length}(\pi) > \mathrm{poly}_7(|Q|)$ and $(q, z) \xrightarrow{\pi} (q', z')$.

Since π has length greater than $|Q| \cdot (\mathrm{poly}'_7(|Q|))^2 + 2$, there exists a state $q'' \in Q$ that occurs at least $(\mathrm{poly}'_7(|Q|))^2$ times in internal configurations within the first $|Q| \cdot (\mathrm{poly}'_7(|Q|))^2 + 2$ configurations of π. Thus, there exists a sequence of proper prefixes $\pi_1 < \ldots < \pi_{\mathrm{poly}'_7(|Q|)}$ of π that all end in q'' and such that one of the following two cases holds.

(i) The numbers $\mathrm{weight}(\pi_i)$ all have the same residue modulo $W_{q'}$.

(ii) The numbers $\mathrm{weight}(\pi_i)$ have pairwise distinct residues modulo $W_{q'}$.

Indeed, since there are $(\mathrm{poly}'_7(|Q|))^2$ prefixes to choose from, either Case (i) holds, or there are strictly less than $\mathrm{poly}'_7(|Q|)$ prefixes per residue class. If the latter holds then there must be least $\mathrm{poly}'_7(|Q|)$ such distinct residue classes, so Case (ii) holds.

In either case, we decompose the computation π as $\pi = \pi_{\mathrm{poly}'_7(|Q|)} \cdot \pi'$. Observe that since π is primitive, then so is π'. Applying Proposition 10 to π' we obtain a path π'' of length at most $|Q|^2 + 1$ such that $\pi_{\mathrm{poly}'_7(|Q|)} \cdot \pi''$ leads from (q, x) to either the same residue class as (q', z') or to a trivial q'-residue class.

It is important to note that we cannot assume π'' is not blocked after the prefix $\pi_{\mathrm{poly}'_7(|Q|)}$. However, since $|\mathrm{blocked}(\pi'')| \leq |Q|^2$, we can remove from the list of prefixes at most $|Q|^2$ prefixes such that the remaining prefixes do not cause π'' to block. (Indeed, we will not modify the path by literally removing prefixes but rather cycles which correspond to the path from a prefix to a longer prefix. For now, we are only speaking about removing elements from the collection of prefixes we can choose from.) W.l.o.g, let $\pi_1, \ldots \pi_d$ be the remaining prefixes.

Consider the family of paths $\theta_i := \pi_i \cdot \pi''$ for $i \in \{1, \ldots, d\}$. Note that every θ_i is of length at most $\mathrm{poly}_7(|Q|)$, and since the θ_i are obtained by removing q''-cycles, and since π is primitive, the configurations reached by θ_i are above (q', z'). We claim that one of the θ_i is a valid run from (q, z) to U_n.

We separate the analysis according to the cases above.

---

[2] Note that we do not claim that the intermediate paths obtained in the procedure are primitive nor that the individual cycles removed in this process are negative. Rather the observation is that π' can equivalently be obtained from π in one step by simultaneously removing a disjoint family of infixes, where each infix is a cycle (necessarily non-positive).

- In Case (i), if π'' leads to a trivial residue class, then all the θ_i reach U_n, and we are done. Otherwise, π'' leads to the same residue class as (q', z'). By our choice of $\text{poly}'_7(|Q|)$ in (5), we have that $d > \sum_{R \text{ non-trivial}} |\delta_n(R)|$. That is, there are more prefixes that do not cause π'' to block than there are missing elements above (q', z') in U_n. We conclude that some θ_i reaches U_n.
- In Case (ii), the paths θ_i all reach distinct residue classes. In particular, since there are more than $|Q|$ such prefixes — i.e. $d > |Q|$ by our choice of $\text{poly}'_7(|Q|)$ — then some θ_i reach trivial residue classes, and thus reach U_n.

H Proof of Proposition 18

We carry out a forward reachability analysis starting from the initial configuration (q_0, x_0). The algorithm runs for $L + 1$ rounds. In the k-th round, we maintain for each state q a set $S_{q,k}$ of configurations (q, x) that are reachable from (q_0, x_0) by valid runs of length k. Let $R_{q,k}$ denote the set of all configurations (q, x) that are reachable from (q_0, x_0) by valid runs of length k. We maintain the invariant that if some configuration $(q, x) \in R_{q,k}$ can reach the objective O in $L - k$ steps via a path π then some configuration $(q, x') \in S_{q,k}$ can also reach O via the same path π. We output that the objective is reachable if and only if one of the sets $S_{q_f,k}$ for some $k \in \{0, \ldots, L\}$ intersects O. This last step is clearly sound, given the invariant.

The key to obtaining a polynomial-time runtime bound is to suitably prune the sets $S_{q,k}$ to keep them of polynomial size. In order to compute $\{S_{q,k+1}\}_{q\in Q}$ from $\{S_{q,k}\}_{q\in Q}$ we proceed as follows. First define $\{S'_{q,k}\}_{q\in Q}$ to be the indexed set of all valid configurations reachable in one step from $\{S_{q,k}\}_{q\in Q}$. Now we obtain $S_{q,k+1}$ from $S'_{q,k}$ by the following two steps:

- First, we delete from $S'_{q,k}$ all configurations (q, x) such that there are $(n + L)$ configurations (q, x') in $S'_{q,k}$ with $x' > x$ and $x' \equiv x \pmod{W}$.
- Secondly, we delete from $S'_{q,k}$ all configurations (q, x) such that there are $(n + L)(m + 1)$ configurations (q, x') in $S'_{q,k}$ with $x' > x$.

Clearly each set $S_{q,k}$ has cardinality at most $(n + L)(m + 1)$, and moreover, it can be computed from the collection of sets $\{S_{q',k-1} \mid q' \in Q\}$ in polynomial time.

It remains to argue that the invariant is maintained between rounds. To this end, suppose some state $(q, x) \in R_{q,k+1}$ can reach the objective in $L - k - 1$ steps via a path π. Then there exists a state $(q', x') \in R_{q',k}$ that can reach the objective in $L - k$ steps via the path $q'\pi$. By the loop invariant there exists a state $(q', x'') \in S_{q',k}$ that can also reach the objective via the path $q'\pi$. Hence there is a state $(q, y) \in S'_{q',k}$ that can reach the objective via the path π. Now if (q, y) is deleted in the first stage of pruning then there is some configuration (q, y') such that $y' > y$, $y' \equiv y \pmod{W}$, and π yields a valid computation from (q, y') to the objective O. After the first stage of pruning, each residue class in $S'_{q,k}$ contains at most $n + L$ elements. Hence if (q, y') is deleted in the second stage of pruning, there are at least $n + L$ configurations (q, y'') in $S_{q,k+1}$ that are above (q, y') and are such that the run over π from (q, y') leads to a configuration (q_f, z) with $\bigwedge_{i=1}^m z \not\equiv a_i \bmod W$. Now from one of these configurations π yields a valid run that reaches O since one of $n + L$ choices of (q, y'') will avoid $\text{blocked}(\pi)$ and lead to a configuration (q_f, z) such that $\bigwedge_{i=1}^n z \neq b_i$.

I Proof of Lemma 21

Items 1 and 2 are obvious. Item 3 follows from the fact that if $\pi_1 \in P$ is dominated by $\pi_1' \in P'$ and $\pi_2 \in R$ is dominated by $\pi_2' \in R'$ then $\pi_1 \cdot \pi_2$ is dominated by $\pi_1' \cdot \pi_2'$. Indeed,

$$\begin{aligned}\mathrm{pmin}(\pi_1 \cdot \pi_2) &= \min(\mathrm{pmin}(\pi_1), \mathrm{weight}(\pi_1) + \mathrm{pmin}(\pi_2))\\ &\leq \min(\mathrm{pmin}(\pi_1'), \mathrm{weight}(\pi_1') + \mathrm{pmin}(\pi_2'))\\ &= \mathrm{pmin}(\pi_1' \cdot \pi_2')\,.\end{aligned}$$

We can similarly argue that $\mathrm{smax}(\pi_1 \cdot \pi_2) \leq \mathrm{smax}(\pi_1' \cdot \pi_2')$.

J Proof of Proposition 22

Fix a state $r \in Q$. Consider all p-r paths that appear as a minimal prefix of some path in P. Pick a single such prefix π_1 of maximum weight. Likewise consider all r-q paths that appear as a maximal suffix of some path in P and pick a single such suffix π_2 of maximum weight. Now form the path $\pi := \pi_1 \cdot \pi_2$. This path dominates any path in P with nadir r. We define P' to be the set of paths π formed in this way as r runs through Q. Without loss of generality, we will henceforth suppose the absolute weight of all paths in P' is at most 2^k. That is, it can be encoded in binary using $k + 1$ bits.

The NC^1 bound on computing P' relies on the well-known fact that the sum of a list of binary integers can be computed in NC^1 [16, Chapter 1]. To obtain P' we compute the weight of each prefix and suffix of every path in P in parallel. According to [16], this can be done in time $O(\log k)$ on a parallel computer with $|P|k$ processors: one for each element of P and each midpoint $0 \leq m \leq k$. Finally, for each state $r \in Q$ in parallel, we find a maximum-weight prefix of a path in P that connects p and r and a maximum-weight suffix of a path in P that connects r and q. It is straightforward to prove the latter is also in NC^1 (see Appendix K) thus completing the proof.

K Computing the maximum of a list of numbers is in NC^1

We will actually prove that the problem is in AC^0. Since AC^0 is known to be strictly contained in NC^1 [1], this is slightly stronger than what we require.

Let us formalize the problem we focus on and define the complexity class AC^0.

Given n numbers $x_0, \ldots, x_{n-1}$ encoded in binary as m-bit strings, the ITMAX problem asks to compute the maximum of the given list of numbers in binary.

The complexity class AC^0 consists of all decision problems decidable by a logspace-uniform family of Boolean circuits with unbounded fan-in $\{\wedge, \vee\}$-gates, polynomial size, and constant depth [1]. We will make use of the following equivalent descriptive-complexity definition: The class AC^0 consists of the set of all languages describable in first-order logic with the addition of the BIT predicate [10]. The latter predicate is defined as follows

$$\mathrm{BIT}(x, i) \overset{\text{def}}{\iff} \text{the } (i+1)\text{-th bit of } x \text{ is set to } 1.$$

In the sequel, let 0 be the index of the most significant bit in the binary representation of the given integers. That is, the first bit of the binary string is the most significant one.

▶ **Proposition 24.** *ITMAX can be computed in* AC^0.

Proof. We proceed by first defining AC^0 circuits $\mathrm{GEQ}_{i,j}$ (or rather first-order logic predicates using BIT) that output 1 for the input if and only if $x_i \geq x_j$ holds. That is, for all $i \neq j$ we define the following.

$$\begin{aligned} \mathrm{GT}_{i,j} &\overset{\text{def}}{\iff} \exists \ell \in [0,m] : \mathrm{BIT}(x_i,\ell) \wedge \neg\mathrm{BIT}(x_j,\ell) \wedge \\ &\qquad (\forall k \in [0,\ell) : \mathrm{BIT}(x_j,k) \leftrightarrow \mathrm{BIT}(x_i,k)) \\ \mathrm{GEQ}_{i,j} &\overset{\text{def}}{\iff} (\forall k \in [0,m] : \mathrm{BIT}(x_i,k) \leftrightarrow \mathrm{BIT}(x_j,k)) \vee \mathrm{GT}_{i,j} \end{aligned}$$

Then, we make use of those circuits to define new AC^0 circuits M_i which recognize whether the i-th element x_i from the input list of integers is a maximal one

$$M_i \overset{\text{def}}{\iff} \forall j \in [0,m] : \mathrm{GEQ}_{i,j} .$$

Finally, in order to output a single maximal element, we build an AC^0 circuit M'_i, per input integer, which outputs 1 if i is the first index such that M_i outputs 1. Concretely, for all $0 \leq i \leq n$ and all $0 \leq k \leq m$, we let

$$\begin{aligned} M'_i &\overset{\text{def}}{\iff} M_i \wedge (\forall j \in [0,i) : \neg M_j) \\ b_k &\overset{\text{def}}{\iff} \exists i \in [0,m] : (M'_i \wedge \mathrm{BIT}(x_i,k)) \end{aligned}$$

where b_k is the k-th output bit of the circuit. Since M'_i clearly holds only if M'_j does not hold, for all $i \neq j$, the circuit correctly outputs the binary representation of the first maximal element of the given list. ◀

Contents

1 Introduction **1**

2 Definitions **2**

3 Unboundedness for 1-VASS with Disequality Tests **4**
3.1 Residue Classes and Chains 5
3.2 Inductive Characterization of $Conf_\infty$ 6
3.3 The Structure of $Conf_\infty$ 7
Case (i): 9
Case (ii): 9
3.4 Computing $Conf_\infty$ and Deciding Unboundedness 11

4 Unboundedness for 1-VASS **12**

5 Conclusion **14**

A Proof of the reduction in Figure 1 **16**

B Single disequality guards suffice **16**

C Proof of Lemma 1 **16**

D Proof of Proposition 4 **17**

E Proof of Proposition 10 **17**

F Proof of Lemma 12 **18**

G Proof of Proposition 17 **18**

H Proof of Proposition 18 **19**

I Proof of Lemma 21 **20**

J Proof of Proposition 22 **20**

K Computing the maximum of a list of numbers is in NC^1 **20**