# Verification of Immediate Observation Population Protocols

**Javier Esparza**[1]
Technische Universität München, Munich, Germany
esparza@in.tum.de
https://orcid.org/0000-0001-9862-4919

**Pierre Ganty**[2]
IMDEA Software Institute, Madrid, Spain
pierre.ganty@imdea.org
https://orcid.org/0000-0002-3625-6003

**Rupak Majumdar**[3]
MPI-SWS, Kaiserslautern, Germany
rupak@mpi-sws.org

**Chana Weil-Kennedy**[4]
Technische Universität München, Munich, Germany
chana.wk@gmail.com

## Abstract

Population protocols (Angluin et al., *PODC*, 2004) are a formal model of sensor networks consisting of identical mobile devices. Two devices can interact and thereby change their states. Computations are infinite sequences of interactions satisfying a strong fairness constraint.

A population protocol is well-specified if for every initial configuration $C$ of devices, and every computation starting at $C$, all devices eventually agree on a consensus value depending only on $C$. If a protocol is well-specified, then it is said to compute the predicate that assigns to each initial configuration its consensus value.

In a previous paper we have shown that the problem whether a given protocol is well-specified and the problem whether it computes a given predicate are decidable. However, in the same paper we prove that both problems are at least as hard as the reachability problem for Petri nets. Since all known algorithms for Petri net reachability have non-primitive recursive complexity, in this paper we restrict attention to immediate observation (IO) population protocols, a class introduced and studied in (Angluin et al., *PODC*, 2006). We show that both problems are solvable in exponential space for IO protocols. This is the first syntactically defined, interesting class of protocols for which an algorithm not requiring Petri net reachability is found.

**2012 ACM Subject Classification** Theory of computation → Distributed computing models

**Keywords and phrases** Population protocols, Immediate Observation, Parametrized verification

**Digital Object Identifier** 10.4230/LIPIcs.CONCUR.2018.31

**Related Version** A full version of the paper is available at `https://arxiv.org/abs/1807.06071`.

[1] Supported by ERC Advanced Grant (787367: PaVeS).
[2] Supported by Madrid Regional Government project S2013/ICE-2731, N-Greens Software – Next-GeneRation Energy-EfficieNt Secure Software, the Spanish Ministry of Economy and Competitiveness project No. TIN2015-71819-P, RISCO – RIgorous analysis of Sophisticated COncurrent and distributed systems, and by a Ramón y Cajal fellowship RYC-2016-20281.
[3] supported by the ERC Synergy award (IMPACT).
[4] Part of this work was done during a visit at the IMDEA Software Institute.

# 1 Introduction

Population protocols [2, 3] are a model of distributed, concurrent computation by anonymous, identical finite-state agents. They capture the essence of distributed computation in different areas. In particular, even though they were introduced to model networks of passively mobile sensors, they are also being studied in the context of natural computing [12, 7]. They also exhibit many common features with Petri nets, another fundamental model of concurrency.

A protocol has a finite set of states $Q$ and a set of transitions of the form $(q, q') \mapsto (r, r')$, where $q, q', r, r' \in Q$. If two agents are in states, say, $q_1$ and $q_2$, and the protocol has a transition of the form $(q_1, q_2) \mapsto (q_3, q_4)$, then the agents can interact and simultaneously move to states $q_3$ and $q_4$. Since agents are anonymous and identical, the global state of a protocol is completely determined by the number of agents at each local state, called a *configuration*. A protocol *computes* a boolean value for a given initial configuration if in all fair executions starting at it, all agents eventually agree to this value[5] – so, intuitively, population protocols compute by reaching a stable consensus. Observe that a protocol may compute no value for some initial configuration, in which case it is deemed not *well-specified* [2].

Population protocols are parameterized systems. Every initial configuration yields a different finite-state instance of the protocol, and the specification is a global property of the infinite family of protocol instances so generated. More precisely, the specification is a predicate $P(x)$ stipulating the boolean value $P(C)$ that the protocol must compute from the initial configuration $C$.

Initial verification efforts for verifying population protocols studied the problem of checking if $P(x)$ is correctly computed for a *finite* set of initial configurations, a task within the reach of finite-state model checkers. In 2015 we obtained the first positive result on parameterized verification [9]. We showed that the problem of deciding if a given protocol is well-specified for all initial configurations is decidable. The same result holds for the correctness problem: given a protocol and a predicate, deciding if the protocol is well-specified and computes the predicate. Unfortunately, we also showed [9, 10] that both problems are as hard as the reachability problem for Petri nets. Since all known algorithms for Petri net reachability run in non-primitive recursive time in the worst case, the applicability of this result is limited.

In this paper we initiate the investigation of subclasses of protocols with a more tractable well specification and correctness problems. We focus on the subclass of *immediate observation* protocols (IO protocols), introduced and studied by Angluin et al. [4]. These are protocols whose transitions have the form $(q_1, q_2) \mapsto (q_1, q_3)$. Intuitively, in an IO protocol an agent can change its state from $q_2$ to $q_3$ by *observing* that another agent is in state $q_1$. This yields an elegant model of protocols in which agents interact through *sensing*: If an agent in state $q_2$ senses the presence of another agent in state $q_1$, then it can change its state to $q_3$. The other agent typically does not even know that it has been sensed, and so it keeps its current state. They also capture the notion of catalysts in chemical reaction networks.

Angluin et al. focused on the expressive power of IO protocols. Our main result is that for IO protocols, both the well specification and correctness problems can be solved in EXPSPACE (we also show the problem is PSPACE-hard). This is the first time that the verification problems of a substantial class of protocols are proved to be solvable in elementary time. To ensure elementary time, our proof uses techniques significantly different from previous results

[5] An execution is fair if it is finite and cannot be extended, or it is infinite and satisfies the following condition: if $C$ appears infinitely often in the execution, then every step enabled at $C$ is taken infinitely often in the execution.

[9]. The key to our result is the use of *counting constraints* to symbolically represent possibly infinite (but not necessarily upward-closed) sets of configurations. A counting constraint is a boolean combination of atomic threshold constraints of the form $x_i \geq k$. We prove that, contrary to the case of arbitrary protocols, the set of configurations reachable from a counting set (the set of solutions of a counting constraint) is again a counting set and we characterize the complexity of representing this set. We believe that this result can be of independent interest for other parameterized systems.

Angluin et al. [4] proved that IO protocols compute exactly the predicates represented by counting constraints. Our main theorem yields a new proof of this result as a corollary. But it also goes further. Using our complexity results, we can provide a lower bound on the state complexity of IO protocols, i.e., on the number of states necessary to compute a given predicate. These results complement recent bounds obtained for arbitrary protocols [5].

## 2 Immediate Observation Population Protocols

### 2.1 Preliminaries

A *multiset* on a finite set $E$ is a mapping $C\colon E \to \mathbb{N}$, thus, for any $e \in E$, $C(e)$ denotes the number of occurrences of element $e$ in $C$. Operations on $\mathbb{N}$ like addition, subtraction, or comparison, are extended to multisets by defining them component wise on each element of $E$. Given $e \in E$, we denote by $\boldsymbol{e}$ the multiset consisting of one occurrence of element $e$, that is, the multiset satisfying $\boldsymbol{e}(e) = 1$ and $\boldsymbol{e}(e') = 0$ for every $e' \neq e$. Given $E' \subseteq E$ define $C(E') \stackrel{\text{def}}{=} \sum_{e \in E'} C(e)$. Given a total order $e_1 \prec e_2 \prec \cdots \prec e_n$ on $E$, a multiset $C$ can be equivalently represented by the vector $(C(e_1), \ldots, C(e_n)) \in \mathbb{N}^n$.

### 2.2 Protocol Schemes

A *protocol scheme* $\mathcal{A} = (Q, \Delta)$ consists of a finite non-empty set $Q$ of states and a set $\Delta \subseteq Q^4$. If $(q_1, q_2, q_1', q_2') \in \Delta$, we write $(q_1, q_2) \mapsto (q_1', q_2')$ and call it a *transition*.

*Confugurations* of a protocol scheme $\mathcal{A}$ are given by *populations*. A population $P$ is a multiset on $Q$ with at least two elements, i.e., $P(Q) \geq 2$. The set of all populations is denoted $\mathrm{Pop}(Q)$. Intuitively, a configuration $C \in \mathrm{Pop}(Q)$ describes a collection of identical finite-state *agents* with $Q$ as set of states, containing $C(q)$ agents in state $q$.

Pairs of agents *interact* using transitions from $\Delta$. Formally, given two configurations $C$ and $C'$ and a transition $\delta = (q_1, q_2) \mapsto (q_1', q_2')$, we write $C \xrightarrow{\delta} C'$ if

$$C \geq (\boldsymbol{q}_1 + \boldsymbol{q}_2) \text{ holds, and } C' = C - (\boldsymbol{q}_1 + \boldsymbol{q}_2) + (\boldsymbol{q}_1' + \boldsymbol{q}_2') \ .$$

(Recall that $\boldsymbol{q}$ is the multiset consisting only of one occurrence of $q$.) From the definition of interaction, it is easily seen that, inside the tuple $(q_1, q_2, q_1', q_2') \in \Delta$, the ordering between $q_1$ and $q_2$ and between $q_1'$ and $q_2'$ is irrelevant. We write $C \xrightarrow{w} C'$ for a sequence $w = \delta_1 \ldots \delta_k$ of transitions if there exists a sequence $C_0, \ldots, C_k$ of configurations satisfying $C = C_0 \xrightarrow{\delta_1} C_1 \cdots \xrightarrow{\delta_k} C_k = C'$. We also write $C \to C'$ if $C \xrightarrow{\delta} C'$ for some transition $\delta \in \Delta$, and call $C \to C'$ an *interaction*. We say that $C'$ is *reachable from* $C$ if $C \xrightarrow{w} C'$ for some (possibly empty) sequence $w$ of transitions.

Note that transitions are enabled only when there are at least two agents. This is why we assume that populations have at least two elements.

An *execution* of $\mathcal{A}$ is a finite or infinite sequence of configurations $C_0, C_1, \ldots$ such that $C_i \to C_{i+1}$ for each $i \geq 0$. An execution $C_0, C_1, \ldots$ is *fair* if it is finite and cannot be extended, or it is infinite and for every step $C \to C'$, if $C_i = C$ for infinitely many indices

$i \geq 0$, then $C_j = C$ and $C_{j+1} = C'$ for infinitely many indices $j \geq 0$ [2, 3]. Informally, if $C$ appears infinitely often in a fair execution, then every step enabled at $C$ is taken infinitely often in the execution.

Given a set $S$ of configurations and a transition $t$ of a protocol scheme $(Q, \Delta)$, we define:

- $post[t](S) \stackrel{\text{def}}{=} \{C' \mid C \xrightarrow{t} C' \text{ for some } C \in S\}$ and $post(S) \stackrel{\text{def}}{=} \bigcup_{t \in \Delta} post[t](S)$.
- $post^0(S) \stackrel{\text{def}}{=} S$; $post^{i+1}(S) \stackrel{\text{def}}{=} post(post^i(S))$ for every $i \geq 0$; and $post^*(S) \stackrel{\text{def}}{=} \bigcup_{i \geq 0} post^i(S)$.

We also define $pre[t](S) \stackrel{\text{def}}{=} \{C' \mid C' \xrightarrow{t} C \text{ for some } C \in S\}$. The sets $pre(S)$ and $pre^*(S)$ are defined as above for *post*.

### 2.2.1 Immediate Observation Protocol Schemes

A protocol scheme is *immediate observation* (IO) if all its transitions are immediate observation. A transition $(q_1, q_2) \mapsto (q_1', q_2')$ is immediate observation iff $\{q_1, q_2\} \cap \{q_1', q_2'\} \neq \emptyset$. Consider, for instance, a transition $(q_s, q_o, q_d, q_o)$ where $q_s, q_o$ and $q_d$ are all distinct. Observe that the transition is immediate observation since $\{q_s, q_o\} \cap \{q_d, q_o\} = \{q_o\} \neq \emptyset$. Intuitively, in an interaction specified by an immediate observation transition, one agent observes the state of another and updates it own state, but the observed agent remains as it was (and its state, unmodified by the interaction, is given by $\{q_1, q_2\} \cap \{q_1', q_2'\}$). Other typical examples of immediate observation transitions are $(q_o, q_o, q_d, q_o)$, $(q_s, q_o, q_o, q_o)$ $(q_s, q_o, q_s, q_o)$ and $(q_o, q_o, q_o, q_o)$ where $q_s, q_o$ and $q_d$ are all distinct. Note that in the last two cases, the state of two agents are the same before and after interacting.

## 2.3 Population Protocols

As Angluin et al. [2], we consider population protocols as a computational model, computing predicates $\Pi\colon \mathrm{Pop}(\Sigma) \to \{0, 1\}$, where $\Sigma$ is a non-empty, finite set of *input variables.*

An *input mapping* for a protocol scheme $\mathcal{A}$ is a function $I\colon \mathrm{Pop}(\Sigma) \to \mathrm{Pop}(Q)$ that maps each input population $X \in \mathrm{Pop}(\Sigma)$ to a configuration of $\mathcal{A}$. The set of *initial configurations* is $\mathcal{I} = \{I(X) \mid X \in \mathrm{Pop}(\Sigma)\}$. An input mapping $I$ is *Presburger* if the set of pairs $(X, C) \in \mathrm{Pop}(\Sigma) \times \mathrm{Pop}(Q)$ such that $C = I(X)$ is definable in Presburger arithmetic. An input mapping $I$ is *simple* if there is an injective map $\nu\colon \Sigma \to Q$ such that $I(X) = \sum_{\sigma \in \Sigma} X(\sigma)\boldsymbol{\nu(\sigma)}$. That is, each input variable is assigned a (distinct) state, and a population $X$ over $\Sigma$ is assigned the initial configuration consisting of $X(\sigma)$ agents in the state $\nu(\sigma)$ and no other agents. Unless otherwise specified, we restrict our attention to the class of *simple* input mappings.

An *output mapping* for a protocol scheme is a function $O\colon Q \to \{0, 1\}$ that associates to each state $q$ of $\mathcal{A}$ an output value in $\{0, 1\}$. The output mapping induces the following properties on configurations: a configuration $C$ is a

- *b-consensus* for $b \in \{0, 1\}$ if $\sum_{p \in O^{-1}(1-b)} C(p) = 0$ and a *consensus* if it is a $b$-consensus for some $b$;
- *dissensus* if it is a $b$-consensus for no $b$ (that is $C$ is a dissensus if $\sum_{p \in O^{-1}(b)} C(p) > 0$ and $\sum_{p \in O^{-1}(1-b)} C(p) > 0$).

A *population protocol* is a triple $(\mathcal{A}, I, O)$, where $\mathcal{A}$ is a protocol scheme, $I$ is a simple input mapping, and $O$ is an output mapping. The population protocol is *immediate observation* (IO) if $\mathcal{A}$ is immediate observation.

An execution $C_0, C_1, \ldots$ *stabilizes to* $b$ for a given $b \in \{0, 1\}$ if there exists $n \in \mathbb{N}$ such that $C_m$ is a $b$-consensus for every $m \geq n$ (if the execution is finite, then this means for every $m$ between $n$ and the length of the execution). Notice that there may be many different executions from a given configuration $C_0$, each of which may stabilize to 0 or to 1 or not stabilize at all (by visiting infinitely many dissensus or infinitely many 0 and 1 consensus).

A population protocol $(\mathcal{A}, I, O)$ is *well-specified* if for every input configuration $C_0 \in \mathcal{I}$, every fair execution of $\mathcal{A}$ starting at $C_0$ stabilizes to the same value $b \in \{0, 1\}$. Otherwise, it is *ill-specified*. The *well specification problem* asks if a given population protocol is well-specified?

Finally, a population protocol $(\mathcal{A}, I, O)$ *computes* a predicate $\Pi\colon \mathrm{Pop}(\Sigma) \to \{0, 1\}$ if for every $X \in \mathrm{Pop}(\Sigma)$, every fair execution of $\mathcal{A}$ starting at $I(X)$ stabilizes to $\Pi(X)$. It follows easily from the definitions that a protocol computes a predicate iff it is well-specified. The *correctness* problem asks, given a population protocol and a predicate whether the protocol computes the predicate.

## 3 Counting Constraints and Counting Sets

▶ **Definition 1.** Let $X = \{x_1, \ldots, x_n\}$ be a set of variables, and let $x \in X$. A constraint of the form $l \le x$, where $l \in \mathbb{N}$, is a *lower bound*, and a constraint of the form $x \le u$, where $u \in \mathbb{N} \cup \{\infty\}$, is an *upper bound*. A *literal* is a lower bound or an upper bound.

A *counting constraint* is a boolean combination of literals. A counting constraint is in *counting normal form* (CoNF) if it is a disjunction of conjunctions of literals, where each conjunction, called a *counting minterm*, contains exactly two literals for each variable, one of them an upper bound and the other a lower bound. We often write a counting constraint in CoNF as the set of its counting minterms.

The semantics of a counting constraint is a *counting set*, a set of vectors in $\mathbb{N}^n$ or, equivalently, a set of valuations to the variables in $X$. The semantics is defined inductively on the structure of a counting constraint, as expected. Define $[\![l \le x]\!] = \{x \mapsto m \in \mathbb{N} \mid m \ge l\}$ ($[\![\infty \le x]\!] = \emptyset$) and $[\![x \le u]\!] = \{x \mapsto m \in \mathbb{N} \mid m \le u\}$. Disjunction, conjunction, and negation of counting constraints translates into union, intersection, and complement of counting sets.

The following proposition follows easily from the definition of counting sets and the disjunctive normal form for propositional logic.

▶ **Proposition 2.**

1. *Counting sets are closed under Boolean operations.*
2. *Every counting constraint is equivalent to a counting constraint in CoNF.*

**Proof Sketch.** **1.** Proof is easy. **2.** Put the constraint in disjunctive normal form. Remove negations in front of literals using $[\![\neg(x_i \le c)]\!] = [\![x_i \ge c + 1]\!]$ if $c \in \mathbb{N}$ and remove the enclosing minterm otherwise; and $[\![\neg(x_i \ge c)]\!] = [\![x_i \le c - 1]\!]$ if $c \in \mathbb{N} \setminus \{0\}$ and remove the enclosing minterm otherwise. Remove minterms containing unsatisfiable literals $l \le x_i \wedge x_i \le u$ with $l > u$. Remove redundant bounds, e.g., replace $(l_1 \le x \wedge l_2 \le x)$ by $\max\{l_1, l_2\} \le x$. If a minterm does not contain a lower bound (upper bound) for $x_i$, add $0 \le x_i$ ($x_i \le \infty$). ◀

Next, we introduce a representation of CoNF-constraints used in the rest of the paper.

▶ **Definition 3** (Representation of CoNF-constraints)**.** We represent a counting minterm by a pair $M \overset{\text{def}}{=} (L, U)$ where $L\colon X \to \mathbb{N}$ and $U\colon X \to \mathbb{N} \cup \{\infty\}$ assign to each variable its lower and upper bound, respectively. We represent a CoNF-constraint $\Gamma$ as the set of representations of its minterms: $\Gamma = \{M_1, \ldots, M_m\}$.

▶ **Definition 4** (Measures of counting constraints)**.** The *L-norm* of a counting minterm $M = (L, U)$ is $\|M\|_l \overset{\text{def}}{=} \sum_{x \in X} L(x)$, and its *U-norm* is $\|M\|_u \overset{\text{def}}{=} \sum_{\substack{x \in X \\ U(x) < \infty}} U(x)$ (and 0 if

$U(x) < \infty$ for no $x$). The $L$- and $U$-norms of a CoNF-constraint $\Gamma = \{M_1, \ldots, M_m\}$ are $\|\Gamma\|_l \stackrel{\text{def}}{=} \max_{i\in[1,m]}\{\|M_i\|_l\}$ and $\|\Gamma\|_u \stackrel{\text{def}}{=} \max_{i\in[1,m]}\{\|M_i\|_u\}$.

▶ **Proposition 5.** *Let $\Gamma_1, \Gamma_2$ be CoNF-constraints over $n$ variables.*

- *There exists a CoNF-constraint $\Gamma$ with $[\![\Gamma]\!] = [\![\Gamma_1]\!] \cup [\![\Gamma_2]\!]$ such that $\|\Gamma\|_u \leq \max\{\|\Gamma_1\|_u, \|\Gamma_2\|_u\}$ and $\|\Gamma\|_l \leq \max\{\|\Gamma_1\|_l, \|\Gamma_2\|_l\}$.*
- *There exists a CoNF-constraint $\Gamma$ with $[\![\Gamma]\!] = [\![\Gamma_1]\!] \cap [\![\Gamma_2]\!]$ such that $\|\Gamma\|_u \leq \|\Gamma_1\|_u + \|\Gamma_2\|_u$ and $\|\Gamma\|_l \leq \|\Gamma_1\|_l + \|\Gamma_2\|_l$.*
- *There exists a CoNF-constraint $\Gamma$ with $[\![\Gamma]\!] = \mathbb{N}^n \setminus [\![\Gamma_1]\!]$ such that $\|\Gamma\|_u \leq n\|\Gamma_1\|_l$ and $\|\Gamma\|_l \leq n\|\Gamma_1\|_u + n$.*

**Proof.** Remember that a CoNF constraint for $m$ minterms in dimension $n$ is a $m$-disjunction of $n$-conjunctions, and that the $L$-norm (respectively $U$-norm) is the maximum sum of lower (resp. upper) bounds in one conjunction. The union of two counting sets $\Gamma_1, \Gamma_2$ with CoNF constraints is represented by the disjunction of the two constraints, and it is still CoNF so the result follows. The intersection is represented by a conjunction of the two constraints and so is not CoNF and needs to be rearranged as in Proposition 2. The new $n$-conjunctions of literals (i.e. the new minterms) mix unmodified bounds from $\Gamma_1$ and $\Gamma_2$, so the result follows. The complement is represented by the negation of the original constraint, which we rearrange into CoNF using $\neg(l \leq x \leq u) \equiv (0 \leq x \leq l-1) \vee (u+1 \leq x \leq \infty)$. We obtain $n$-conjunctions with lower bounds of the form $u+1$, with $u \leq \|\Gamma_1\|_u$ an upper bound in a minterm of the original constraint. This yields $\|\Gamma\|_l \leq n\|\Gamma_1\|_u + n$ and the reasoning is similar for the $U$-norm. ◀

▶ Remark 6. The counting sets contain the finite, upward-closed and downward-closed sets:

- Every finite subset of $\mathbb{N}^n$ is a counting set. Indeed, $\{(k_1, \ldots, k_n)\} = [\![(L, U)]\!]$ with $L(x_i) = k_i = U(x_i)$ for every $x_i \in X$, and so finite sets are counting sets too.
- A set $S \subseteq \mathbb{N}^n$ is upward-closed if whenever $v \in S$ and $v \leq_\times v'$, we have $v' \in S$, where we write $v \leq_\times v'$ if the ordering holds pointwise (meaning $v(x) \leq v'(x)$ for every $x \in X$). Upward-closed sets are counting sets. Indeed, by Dickson's lemma, every upward-closed set has a finite set $\{v_1, \ldots, v_k\}$ of minimal elements with respect to $\leq_\times$, and so the set is $[\![\{(L_1, U), \ldots, (L_k, U)\}]\!]$ where $L_i(x_j) = v_i(j)$ and $U(x_j) = \infty$ for every $1 \leq j \leq n$.
- A set $S \subseteq \mathbb{N}^n$ is downward-closed if whenever $v \in S$ and $v' \leq_\times v$, we have $v' \in S$. Since a set is downward-closed iff its complement is upward-closed, every downward-closed set is a counting set. Further, it is easy to see that downward-closed sets are represented by counting constraints $\{(L, U_1), \ldots, (L, U_k)\}$ where $L(x_j) = 0$ for every $1 \leq j \leq n$.

Next, we define a well-quasi-ordering on counting sets. For two counting minterms $M_1$ and $M_2$, we write $M_1 \preceq M_2$ if $[\![M_1]\!] \supseteq [\![M_2]\!]$. For CoNF-constraints $\Gamma_1$ and $\Gamma_2$, define the ordering $\Gamma_1 \sqsubseteq \Gamma_2$ if for each counting minterm $M_2 \in \Gamma_2$ there is a counting minterm $M_1 \in \Gamma_1$ such that $M_1 \preceq M_2$. Note that $\Gamma_1 \sqsubseteq \Gamma_2$ implies $[\![\Gamma_1]\!] \supseteq [\![\Gamma_2]\!]$.

▶ **Theorem 7.** *For every $u \geq 0$, the ordering $\sqsubseteq$ on counting sets represented by CoNF-constraints of $U$-norm at most $u$ is a well-quasi-order.*

**Proof.** We first prove that counting minterms with $\preceq$ form a better quasi order. For two counting minterms $M_1$ and $M_2$, we write $M_1 \preceq M_2$ if $[\![M_1]\!] \supseteq [\![M_2]\!]$. Let $\mathcal{M} = M_1, M_2, \ldots$ be an infinite sequence of counting minterms of $U$-norm at most $u$, where $M_i = (L_i, U_i)$. Since there are only finitely many mappings $U\colon X \to \mathbb{N} \cup \{\infty\}$ of norm at most $u$, the sequence $\mathcal{M}$ contains an infinite subsequence $\mathcal{M}'$ such that every minterm $M_i$ of $\mathcal{M}'$ satisfies $U_i = U$ for some mapping $U$. So $\mathcal{M}'$ is of the form $(L_1, U), (L_2, U) \ldots$ By Dickson's lemma, there are $i < j$ such that $L_i \leq_\times L_j$ , and so $[\![(L_i, U)]\!] \supseteq [\![(L_j, U)]\!]$. Hence, defining $C$ be the set of all counting minterms of $U$-norm at most $u$ we find that $(C, \preceq)$ is a well-quasi-order. In fact, standard arguments show that this is a better-quasi-order [1]. Hence, the ordering $\sqsubseteq$ is a better quasi order on counting constraints [1], implying it is also a well-quasi-order. ◀

# 4 Reachability Sets of IO Population Protocols

We show that if $S$ is a counting set, then $post^*(S)$ and $pre^*(S)$ are also counting sets. First we show that we can restrict ourselves to IO protocols in a certain normal form.

## 4.1 A Normal Form for Immediate Observation Protocols

An IO protocol is in *normal form* if $q_s \neq q_o$ for every transition $(q_s, q_o) \mapsto (q_o, q_d)$, i.e., the state of the observed agent is different from the source state of the observer.

Given an IO population protocol $\mathcal{P} = (\mathcal{A}, I, O)$ we define an IO protocol in normal form $\mathcal{P}' = (\mathcal{A}', I', O')$ which is well-specified iff $\mathcal{P}$ is well-specified. Further, the number of states and transitions of $\mathcal{P}'$ is linear in the number of states and transitions of $\mathcal{P}$. The mapping $I'$ is a Presburger mapping even if $I$ is simple, but this does not affect our results.

$\mathcal{P}'$ is defined adding transition and states to $\mathcal{P}$. First we add a state $r$. Then, we replace each transition $t = (q, q) \mapsto (q, q_d)$ of $\mathcal{P}$ by a transition $t' = (q', q) \mapsto (q', q_d)$, where $q'$ is a primed copy of $q$, and add two further transitions $(q, r) \mapsto (r, q')$ and $(q', r) \mapsto (r, q)$.

It remains to define the output function of the new states as well as the input mapping $I'$ of $\mathcal{P}'$. We define $I'$ to be a Presburger initial mapping which coincides with $I$ on the state of $\mathcal{P}$ and such that $I(X)(r) = 1$ for all $X$ and $I(X)(q') = 0$ for all $X$ and primed state $q'$. The output of primed copies is the same as their unprimed version, that is $O(q') = O(q)$. The only technical difficulty is the definition of the output of state $r$. Because of the way in which we have defined the transitions involving $r$, the agent initially in state $r$ cannot leave $r$. Therefore, whatever the output $O(r)$ we assign to $r$, the protocol $\mathcal{P}'$ can never reach consensus $1 - O(r)$, and so $\mathcal{P}'$ may not be well-specified even if $\mathcal{P}$ is. To solve this problem, we add a primed copy $r'$ of $r$ such that $r$ and $r'$ have distinct outputs. Every transition with $r$ as observer is duplicated but this time with $r'$ as observed state. Finally, for every state $q$ of $\mathcal{P}$, if $O(q) = O(r')$ we add the transition $(q, r) \mapsto (q, r')$, and otherwise we add the transition $(q, r') \mapsto (q, r)$. After adding these states, the agent initially in $r$ switches between $r$ and $r'$, and finally stabilizes to the same value the other agents stabilize to.

## 4.2 The Functions $pre^*$ and $post^*$ Preserve Counting Sets

We show that if $S$ is a counting set, then $post^*(S)$ and $pre^*(S)$ are also counting sets. Further, given a CoNF-constraint $\Gamma$ representing $S$, we show how to construct a CoNF-constraint representing $post^*(S)$ and $pre^*(S)$. In the following, we abbreviate $post([\![\Gamma]\!])$ to $post(\Gamma)$, and similarly for other notations involving $post$ and $pre$, like $post[t](\Gamma)$, $post^*(\Gamma)$, etc.

We start with some simple examples. First, we observe that the result does not hold for arbitrary population protocols. Consider the protocol with four distinct states $\{q_1, q_2, q_3, q_4\}$ and one single transition $(q_1, q_2) \mapsto (q_3, q_4)$. Let $M = [\![0 \leq x_3 \leq 0 \wedge 0 \leq x_4 \leq 0]\!]$. Then $post^*(M) = [\![x_3 = x_4]\!]$, which is not a counting set. Intuitively, the reason is that the transitions links the number of agents in states $x_3$ and $x_4$. However, this is only possible because the transition is not IO. Indeed, consider now the protocol $\mathcal{P}_1$ with states $\{q_1, q_2, q_3\}$ and one single IO transition $(q_1, q_2) \mapsto (q_1, q_3)$. Table 1 lists some typical constraints for $M$, and gives constraints for $post^*(M)$.

**Table 1** The set $post^*[t](M)$ for two IO transitions and counting minterm $M$. For conciseness and clarity we use equality constraints instead of two inequalities.

| $M$ | $\|M\|_l$ | $\|M\|_u$ | $\Gamma \stackrel{\text{def}}{=} post^*[t](M)$ where $t \stackrel{\text{def}}{=} (q_1, q_2) \mapsto (q_1, q_3)$ | $\|\Gamma\|_l$ | $\|\Gamma\|_u$ |
|---|---|---|---|---|---|
| $x_1 = 0 \wedge x_2 \geq 2 \wedge x_3 = 1$ | 3 | 1 | $x_1 = 0 \wedge x_2 \geq 2 \wedge x_3 = 1$ | 3 | 1 |
| $x_1 = 1 \wedge x_2 = 2 \wedge x_3 \geq 1$ | 4 | 3 | $(x_1 = 1 \wedge x_2 = 2 \wedge x_3 \geq 1)$ $\vee(x_1 = 1 \wedge x_2 = 1 \wedge x_3 \geq 2)$ $\vee(x_1 = 1 \wedge x_2 = 0 \wedge x_3 \geq 3)$ | 4 | 3 |
| $x_1 = 1 \wedge x_2 \geq 1 \wedge x_3 = 2$ | 4 | 3 | $(x_1 = 1 \wedge x_2 \geq 1 \wedge x_3 = 2)$ $\vee(x_1 = 1 \wedge x_2 \geq 0 \wedge x_3 \geq 3)$ | 4 | 3 |
| $x_1 \geq 0 \wedge x_2 \geq 1 \wedge x_3 \geq 2$ | 3 | 0 | $(x_1 \geq 0 \wedge x_2 \geq 1 \wedge x_3 \geq 2)$ $\vee(x_1 \geq 1 \wedge x_2 \geq 0 \wedge x_3 \geq 3)$ | 4 | 0 |
| $M$ | $\|M\|_l$ | $\|M\|_u$ | $\Gamma \stackrel{\text{def}}{=} post^*[t](M)$ where $t \stackrel{\text{def}}{=} (q_1, q_2) \mapsto (q_2, q_2)$ | $\|\Gamma\|_l$ | $\|\Gamma\|_u$ |
| $x_1 \geq 1 \wedge x_2 = 0$ | 1 | 0 | $x_1 \geq 1 \wedge x_2 = 0$ | 1 | 0 |
| $x_1 = 1 \wedge x_2 \geq 2$ | 3 | 1 | $(x_1 = 1 \wedge x_2 \geq 2) \vee (x_1 = 0 \wedge x_2 \geq 3)$ | 3 | 1 |
| $x_1 \geq 2 \wedge x_2 = 1$ | 3 | 1 | $(x_1 \geq 2 \wedge x_2 \geq 1) \vee (x_1 \geq 1 \wedge x_2 \geq 2)$ $\vee(x_1 \geq 0 \wedge x_2 \geq 3)$ | 3 | 0 |

Given a minterm $(L, U)$, we syntactically define a CoNF-constraint $(L, U)_{t^*}$ for the set:

$$post^*[t](L, U) \stackrel{\text{def}}{=} \{C' \mid \exists k \geq 0 \exists C \in [\![(L, U)]\!] \text{ such that } C \xrightarrow{t^k} C'\} \ .$$

That is, $(L, U)_{t^*}$ captures the set of all configurations that can be obtained from $(L, U)$ by firing transition $t$ an arbitrary number of times.

▶ **Definition 8.** Let $(L, U)$ be a minterm and let $t = (q_s, q_o) \mapsto (q_d, q_o)$ be an IO transition. Define $(L, U)_{t^*}$ to be the set given by $(L, U)$ and all the minterms $(L', U')$ such that all the following conditions hold:

1. $[\![(L'', U)]\!] \neq \emptyset$ where $[\![L'']\!] = [\![L]\!] \cap [\![x_s \geq 1 \wedge x_o \geq 1]\!]$.
2. $U'(x) = U(x)$ and $L'(x) = L''(x)$ for every $x \in X \setminus \{x_s, x_d\}$.
3. If $U(x_s) < \infty$, then there exists $1 \leq k \leq U(x_s)$ such that $U'(x_s) = U(x_s) - k$, $L'(x_s) = \max\{0, L''(x_s) - k\}$, $U'(x_d) = U(x_d) + k$ and $L'(x_d) = L''(x_d) + k$.
4. If $U(x_s) = \infty$, then $U'(x_s) = U'(x_d) = \infty$ and there exists $1 \leq k \leq L''(x_s)$ such that $L'(x_s) = L''(x_s) - k$ and $L'(x_d) = L''(x_d) + k$.

Given a CoNF-constraint $\Gamma = \{M_1, \ldots, M_m\}$, we define $\Gamma_{t^*} = \bigcup_{i=1}^{m} M_{it^*}$.

▶ **Lemma 9.** *Let $\mathcal{P}$ be an IO protocol and let $\Gamma$ be a CoNF-constraint. Then $\Gamma_{t^*} = post^*[t](\Gamma)$. Further, $\|\Gamma_{t^*}\|_u \leq \|\Gamma\|_u$.*

**Proof.** It suffices to prove that for every minterm $(L, U)$ and for every transition $t$ we have $post^*[t](L, U) = (L, U)_{t^*}$ and $\|(L, U)_{t^*}\|_u \leq \|(L, U)\|_u$. The rest follows easily from the definitions of $post^*$ and of a counting constraint.

Condition (1) holds iff some vector in $[\![(L, U)]\!]$ enables $t$, hence $[\![(L'', U)]\!]$ is the set $[\![(L, U)]\!]$ of vectors minus those disabling $t$. If no vector enables $t$ then $(L, U)_{t^*}$ is the singleton $\{(L, U)\}$. Condition (2) states that the number of agents in states other than $q_s$ and $q_d$ does not change. Condition (3–4) defines the result of firing $t$ one or more times.

The inequality $\|(L, U)_{t^*}\|_u \leq \|(L, U)\|_u$ follows immediately from (1–4). Observe that $\|(L, U)_{t^*}\|_u < \|(L, U)\|_u$ may hold if $U(x_s) = \infty$ and $U(x_d) < \infty$. ◀

To prove the main theorem of the section, we introduce the following definition.

▶ **Definition 10.** Given a protocol $\mathcal{P}$, let $S$ be a set of configurations and let $\Gamma$ be a CoNF-constraint.

- Define: $post_a(S) \stackrel{\text{def}}{=} \bigcup_{t\in\Delta} post^*[t](S)$; $post_a^0(S) \stackrel{\text{def}}{=} S$ and $post_a^{i+1}(S) \stackrel{\text{def}}{=} post_a(post_a^i(S))$ for every $i \geq 0$; $post_a^*(S) \stackrel{\text{def}}{=} \bigcup_{i\geq 0} post_a^i(S)$.
- Similarly, define in the constraint domain: $post_a(\Gamma) \stackrel{\text{def}}{=} \bigcup_{t\in\Delta} \Gamma_{t^*}$; $post_a^0(\Gamma) \stackrel{\text{def}}{=} \Gamma$ and $post_a^{i+1}(\Gamma) \stackrel{\text{def}}{=} post_a(post_a^i(\Gamma))$ for every $i \geq 0$.

The $a$-subscript stands for "accelerated." Observe that we cannot define $post_a^*(\Gamma)$ directly as the infinite union $\bigcup_{i\geq 0} post_a^i(\Gamma)$ because constraints are only closed under finite unions.

▶ **Theorem 11.** *Let $\mathcal{P}$ be an IO protocol and let $S$ be a counting set. Then both $post^*(S)$ and $pre^*(S)$ are counting sets.*

**Proof.** We first prove that $post^*(S)$ is a counting set. It follows from Definition 10 that $post^i(S) \subseteq post_a^i(S)$ but $post_a^i(S) \subseteq post^*(S)$ for every $i \geq 0$, hence $post_a^*(S) = post^*(S)$, and so it suffices to prove that $post_a^*(S)$ is a counting set.

Let $\Gamma$ be a CoNF-constraint such that $[\![\Gamma]\!] = S$. By Lemma 9, $post_a^i(\Gamma)$ is a counting set and $\|post_a^i(\Gamma)\|_u \leq \|\Gamma\|_u$ for every $i \geq 0$. By Theorem 7, there exist indices $i < j$ such that $post_a^j(\Gamma) \subseteq post_a^i(\Gamma)$, hence $post_a^j(\Gamma) = post_a^i(\Gamma)$ since $\Gamma' \subseteq post_a(\Gamma')$ for all $\Gamma'$, and finally $post_a^*(\Gamma) = \bigcup_{k=1}^{j} post_a^k(\Gamma)$. Since counting sets are closed under finite union, $post_a^*(S)$ is a counting set.

Finally we show that $pre^*(S)$ is also a counting set. Consider the protocol $\mathcal{P}_r$ obtained by "reversing" the transitions of $\mathcal{P}$, i.e., $\mathcal{P}_r$ has a transition $(q_1, q_2) \mapsto (q_3, q_4)$ iff $\mathcal{P}$ has a transition $(q_3, q_4) \mapsto (q_1, q_2)$. Then $pre^*(S)$ in $\mathcal{P}$ is equal to $post^*(S)$ in $\mathcal{P}_r$. ◀

## 4.3 Bounding the Size of $post^*(\Gamma)$

Given a CoNF-constraint $\Gamma$, we obtain an upper bound on the size of a CoNF-constraint denoting $post^*(\Gamma)$ and $pre^*(\Gamma)$. More precisely, we obtain bounds on the $L$-norm and $U$-norm of a constraint for $post^*(\Gamma)$ as a function of the same parameters for $\Gamma$.

We first recall a theorem of Rackoff [14] recast in the terminology of population protocols.

▶ **Theorem 12** ([14, 6]). *Let $\mathcal{P}$ be a population protocol with set of states $Q$ and let $C$ be a configuration of $\mathcal{P}$. For every configuration $C'$, if there exists $C''$ such that $C' \xrightarrow{*} C'' \geq_\times C$, then there exists $\sigma$ and $C'''$ such that $C' \xrightarrow{\sigma} C''' \geq_\times C$ and $|\sigma| \leq (3 + C(Q))^{(3|Q|)!+1} \in C(Q)^{2^{\mathcal{O}(|Q|\log|Q|)}}$. (Recall that $C(Q) \stackrel{\text{def}}{=} \sum_{q\in Q} C(q)$ and $C(Q) \geq 2$.)*

Observe that the bound on the length of $\sigma$ depends only on $C$ and $\mathcal{P}$, but not on $C'$. Using this theorem we can already obtain an upper bounds for $pre^*(\Gamma)$ when $[\![\Gamma]\!]$ is upward-closed. The bound is valid for arbitrary population protocols.

Recall that if $[\![\Gamma]\!]$ is upward-closed we can assume $\|\Gamma\|_u = 0$ (see Remark 6).

▶ **Proposition 13.** *Let $\mathcal{P}$ be population protocol with $n$ states. Let $S$ be an upward-closed set of configurations and let $\Gamma$ be a CoNF-constraint with $\|\Gamma\|_u = 0$ such that $[\![\Gamma]\!] = S$. There exists a CoNF constraint $\Gamma'$ such that $[\![\Gamma']\!] = pre^*(\Gamma)$ and $\|\Gamma'\|_u = 0$, $\|\Gamma'\|_l \in (\|\Gamma\|_l)^{2^{\mathcal{O}(n\log n)}}$.*

**Proof.** It is well known that if $S$ is upward-closed, then so is $pre^*(S)$. (This follows from Lemma 9, but is also an easy consequence of the fact that $C \xrightarrow{*} C'$ implies $C + C'' \xrightarrow{*} C' + C''$ for every $C''$). Let $K \stackrel{\text{def}}{=} (3 + \|\Gamma\|_l)^{(3n)!+1}$. By Theorem 12, for every configuration $C$, if $C \in pre^*(S)$ then $C \in \bigcup_{i=0}^{K} pre^i(S)$, and so $pre^*(S) = \bigcup_{i=0}^{K} pre^i(S) = pre_a^K(S)$. Let $\Gamma' = pre_a^K(\Gamma)$. Then $[\![\Gamma']\!] = pre^*(S)$. Further, we have $\|\Gamma'\|_u = 0$ by Lemma 9 (the Lemma proves the result for $post^*$, but exactly the same proof works for $pre^*$ by reversal of transitions). To prove the bound for the $L$-norm, observe that by the definition of $(L,U)_{t^*}$ we have $\|(L,U)_{t^*}\|_l \leq \|(L,U)\|_l + 1$, as we are always in case 4. of Definition 8 (because $S$ is upward-closed). Since $pre_a(\Gamma) = \bigcup_{t\in\Delta_r} \Gamma_{t^*}$ and the $L$-norm of a union is the maximum of the $L$-norms, we get $\|pre_a(\Gamma)\|_l \leq \|\Gamma\|_l + 1$. By induction, $\|pre_a^K(\Gamma)\|_l \leq \|\Gamma\|_l + K$, and the result follows. ◀

In the rest of the section we obtain a bound valid not only for upward-closed sets, but for arbitrary counting sets. The price to pay is a restriction to IO protocols. We start with some miscellaneous notations that will be useful.

- Given a mapping $f\colon X \to \mathbb{N}$ and $Y \subseteq X$ we write $f(Y)$ for $\sum_{x\in Y} f(x)$, and $f|_Y$ for the projection of $f$ onto $Y$.
- Given a transition sequence $\sigma$, we denote by $c(\sigma)$ the "compression" of $\sigma$ as the shortest regular expression $r = t_1^* \dots t_m^*$ such that $\sigma \in L(r)$, and denote $|c(\sigma)| = m$. While $\sigma$ induces a sequence of $pre[t]$ or $post[t]$, $c(\sigma)$ induces a sequence of $pre^*[t]$ or $post^*[t]$.

For the rest of the section we fix an IO protocol $\mathcal{P}$ with a set of states $Q$ and $|Q| = n$. We say that $C$ *covers* $C'$ if $C \geq_\times C'$. We introduce a relativization.

▶ **Definition 14.** Let $E \subseteq Q$. A configuration $C$ *$E$-covers* $C'$, denoted $C \geq_E C'$, if $C(q) = C'(q)$ for every $q \in E$ and $C(q) \geq C'(q)$ for every $q \in Q \setminus E$. $\mathcal{P}$ is *$E$-increasing* if for every transition $(q_s, q_o) \mapsto (q_d, q_o)$ either $q_s \notin E$ or $q_d \in E$.

Observe that $\mathcal{P}$ is vacuously $\emptyset$-increasing and $Q$-increasing. Intuitively, if $\mathcal{P}$ is $E$-increasing then the total number of agents in the states of $E$ cannot decrease. Indeed, for that we would need a transition that removes agents from $E$ without replacing them, i.e., a transition such that $q_s \in E$ and $q_d \notin E$. So, by induction, we have:

▶ **Lemma 15.** *If $\mathcal{P}$ is $E$-increasing and $C' \xrightarrow{*} C$ then $C'(E) \leq C(E)$.*

Now we give a result bounding the length of $E$-covering sequences for $E$-increasing protocols.

▶ **Lemma 16.** *Let $\mathcal{P} = (Q,\Delta)$ be an IO protocol scheme, let $C$ be a configuration of $\mathcal{P}$, and let $E \subseteq Q$ such that $\mathcal{P}$ is $E$-increasing. For every configuration $C'$, if there exists $C''$ such that $C' \xrightarrow{*} C'' \geq_E C$, then there exists $\sigma$ and $C'''$ such that $C' \xrightarrow{\sigma} C''' \geq_E C$ and $|\sigma| \in C(Q)^{2^{\mathcal{O}(n\log n)}}$, where the constant in the Landau symbol is independent of $\mathcal{P}$ and $C$.*

**Proof.** We use a theorem of Bozzelli and Ganty [6] that generalizes Rackoff's theorem to Vector Addition Systems with States (VASS). Recall that a $d$-VASS is a pair $(P,\Delta)$ where $P$ is a set of control points and $\Delta \subseteq P \times \mathbb{Z}^d \times P$ is a finite set of transitions. The number $d$ is called the dimension. A configuration of a $d$-VASS is a pair $(p,v)$, where $p \in P$ and $v \in \mathbb{N}^d$. Intuitively, the VASS acts on $d$ counters that can only take non-negative values. Formally, we have $(p,v) \to (p',v')$ if there is a transition $(p,v'',p')$ such that $v + v'' = v'$, i.e., the machine moves from $p$ to $p'$ by updating the counters with $v''$. Given two configurations $(p,v)$ and $(p',v')$, we write $(p,v) \geq_\times (p',v')$ if $p = p'$ and $v \geq_\times v'$. It is shown [6] in Theorem 1 that given a $d$-VASS $(P,\Delta)$ and a configuration $C$, for each configuration $C'$, if there exists $C''$ such that $C' \xrightarrow{*} C'' \geq_\times C$, then there exists $\sigma$ and $C'''$ such that $C' \xrightarrow{\sigma} C''' \geq_\times C$ and $|\sigma| \leq |P| \cdot (\|\Delta\|_1 + \|C\|_1 + 2)^{(3d)!+1}$, where $\|\Delta\|_1$ and $\|C\|_1$ denote the maximal components of $\Delta$ and $C$, respectively.

Let $n = |Q|$. We construct a VASS $V_{\mathcal{P},E}$ that simulates the protocol $\mathcal{P}$, and then apply Bozzelli and Ganty's theorem. We do not give all the formal details of the construction.

Intuitively, given a configuration $C$ of $\mathcal{P}$, we split it into $(C|_E, C|_{Q\setminus E})$. Since $\mathcal{P}$ is $E$-increasing, every configuration $(C'|_E, C'|_{Q\setminus E})$ from which we can reach $(C|_E, C|_{Q\setminus E})$ satisfies $C'|_E(E) \le C|_E(E)$ (Lemma 15), and so there are only finitely many (at most $(C(E)+1)^n$) possibilities for $C'|_E$. The control points of the VASS $V_{\mathcal{P},E}$ correspond to these finitely many possibilities. Formally, the set of control points of $V_{\mathcal{P},E}$ is the set of all mappings $M\colon E \to \mathbb{N}$ such that $M(E) \le C(E)$, plus some auxiliary control points (see below). The dimension, or number of counters, is $|Q \setminus E|$. The transitions of $V_{\mathcal{P},E}$ simulate the transitions of $\mathcal{P}$. For example, assume $t = (q_o, q_s) \mapsto (q_o, q_d)$ is a transition of $\mathcal{P}$ such that $q_s, q_o \notin E$ and $q_d \in E$. Then for every control point $M$ of $V_{\mathcal{P},E}$ the VASS has a transition $t_1$ leading from $M$ to an auxiliary control point $\langle M, t\rangle$, and a transition $t_2$ leading from $\langle M, t\rangle$ to the control point $M'$ given by $M'(q_d) = M(q_d) + 1$ and $M'(q) = M(q)$ for every other $q \in E$. Transition $t_1$ decrements the counter of $q_s$ and $q_o$ by 1, leaving all other counters untouched, and transition $t_2$ increments the counters $q_o$, leaving all other counters untouched.

It follows that there is an execution $C' \xrightarrow{*} C'' \ge_E C$ in $\mathcal{P}$ iff there is an execution $(C'|_E, C'|_{Q\setminus E}) \xrightarrow{*} (C''|_E, C''|_{Q\setminus E}) \ge_\times (C|_E, C|_{Q\setminus E})$ in $V_{\mathcal{P},E}$ of at most twice the length.

Applying Bozzelli and Ganty's theorem, we obtain that the length of $\sigma$ is bounded by $|P| \cdot (\|\hat{\Delta}\|_1 + \|C\|_1 + 2)^{(3d)!+1}$, where $|P|$, $\hat{\Delta}$, and $d$ are now the set of control points, transitions, and dimension of $V_{\mathcal{P},E}$. We have $|P| \le (C(E)+1)^n + |\Delta|(C(E)+1)^n$, $d = |Q \setminus E| \le n$, $\|\hat{\Delta}\|_1 = 2$. Further, we have $\|C\|_1 \le C(Q \setminus E)$, which leads to a bound of $(1 + |\Delta|)(C(E)+1)^n \cdot (C(Q\setminus E) + 4)^{(3n)!+1} \in C(Q)^{2^{\mathcal{O}(n \log n)}}$. ◀

Next we prove a double exponential bound on the length of $E$-covering sequences. The result is similar to Lemma 16 with two important changes: the restriction to $E$-increasing protocols is dropped, and we consider the bound on the length of $c(\sigma)$ instead of $\sigma$.

▶ **Theorem 17.** *Let $\mathcal{P}$ be an IO protocol with a set $Q$ of $n$ states, and let $C$ be a configuration of $\mathcal{P}$. For every $E \subseteq Q$ and for every configuration $C_0$, if there exists $\tau$ and $C'$ such that $C_0 \xrightarrow{\tau} C' \ge_E C$, then there exists $\sigma$ and $C''$ such that $C_0 \xrightarrow{\sigma} C'' \ge_E C$ and $|c(\sigma)| \in C(Q)^{2^{\mathcal{O}(n^2 \log n)}}$, where the constant in the Landau symbol is independent of $\mathcal{P}$, $C$, and $C_0$.*

**Proof.** We prove by induction on $|E|$ that the result holds with $|c(\sigma)| \in C(Q)^{2^{e\mathcal{O}(n \log n)}}$, where $e \stackrel{\text{def}}{=} \max\{1, |E|\}$, and then apply $e \le n$.

**Base:** $|E| = 0$. Then $\mathcal{P}$ is vacuously $E$-increasing, and the result follows from Lemma 16.

**Step:** $|E| > 0$. We use the following notation: Given a transition sequence $\rho$, we denote $\mathcal{P}_\rho$ the restriction of $\mathcal{P}$ to the transitions that occur in $\rho$.

If $\mathcal{P}_\tau$ is $E$-increasing, then we can apply Lemma 16, and we are done. Else, the definition of $E$-increasing shows there exist $C_1$ and $C_2$ and a decomposition $\tau = \tau_1\, t\, \tau_2$ such that

$$C_0 \xrightarrow{\tau_1} C_1 \xrightarrow{t} C_2 \xrightarrow{\tau_2} C' \ge_E C \ .$$

The protocol $\mathcal{P}_{\tau_2}$ is $E$-increasing, but $\mathcal{P}_{t\tau_2}$ is not $E$-increasing (observe that possibly $\tau_2 = \epsilon$). By Lemma 16 applied to $\mathcal{P}_{\tau_2}$, there exists $\sigma_2$ and $\tilde{C}''$ such that

$$C_0 \xrightarrow{\tau_1} C_1 \xrightarrow{t} C_2 \xrightarrow{\sigma_2} \tilde{C}'' \ge_E C \quad \text{and} \quad |\sigma_2| \in C(Q)^{2^{\mathcal{O}(n \log n)}} \ .$$

Since $\sigma_2$ can remove at most $|\sigma_2|$ agents from a state, there exist $C_1', C_2', C''$ such that

$$C_0 \xrightarrow{\tau_1} C_1 \ge_E C_1' \xrightarrow{t} C_2' \xrightarrow{\sigma_2} C'' \ge_E C \quad \text{and } C_1'(Q) \in C(Q)^{2^{\mathcal{O}(n \log n)}} \ .$$

Indeed, it suffices to define

- $C_1'(q) = \min\{C_1(q), |\sigma_2| + C(q)\}$ for every $q \in Q \setminus E$ and $C_1'(q) = C_1(q)$ for every $q \in E$,

- $C'_2(q) = \min\{C_2(q), |\sigma_2| + C(q)\}$ for every $q \in Q \setminus (E \cup \{q_d\})$, $C'_2(q) = C_2(q)$ for every $q \in E$ and $C'_2(q_d) = \min\{C_2(q_d), 1 + |\sigma_2| + C(q)\}$ where $t = (q_o, q_s) \mapsto (q_o, q_d)$.

Recall that $\mathcal{P}_{t\tau_2}$ is not $E$-increasing, and so $t = (q_o, q_s) \mapsto (q_o, q_d)$ for states $q_s, q_d$ such that $q_s \in E$ and $q_d \notin E$. (Intuitively, the occurrence of $t$ "removes agents" from $E$.) Let $E' \stackrel{\text{def}}{=} E \setminus \{q_s\}$. Since $C_0 \xrightarrow{\tau_1} C_1 \geq_E C'_1$, we also have $C_0 \xrightarrow{\tau_1} C_1 \geq_{E'} C'_1$. By induction hypothesis, there exists $\sigma_1$ and $C''_1$ such that $C_0 \xrightarrow{\sigma_1} C''_1 \geq_{E'} C'_1$ and

$$\begin{aligned} |c(\sigma_1)| &\in C'_1(Q)^{2^{e'\mathcal{O}(n \log n)}} \in \left(C(Q)^{2^{\mathcal{O}(n \log n)}}\right)^{2^{e'\mathcal{O}(n \log n)}} \in C(Q)^{2^{\mathcal{O}(n \log n)} \cdot 2^{e'\mathcal{O}(n \log n)}} \\ &\in C(Q)^{2^{\mathcal{O}(n \log n) + e'\mathcal{O}(n \log n)}} \in C(Q)^{2^{e\mathcal{O}(n \log n)}} \ . \end{aligned}$$

(Observe that $C''_1 \geq_{E'} C'_1$ holds, but $C''_1 \geq_E C'_1$ may not hold, we may have $C''_1(q_s) > C'_1(q_s)$.) To sum up, we have configurations $C'_1, C''_1, C'_2, C''$ and transition sequences $\sigma_1, \sigma_2$ such that

$$C_0 \xrightarrow{\sigma_1} C''_1 \geq_{E'} C'_1 \xrightarrow{t} C'_2 \xrightarrow{\sigma_2} C'' \geq_E C \quad \text{ and } \quad |c(\sigma_1\, t\, \sigma_2)| \in C(Q)^{2^{e\mathcal{O}(n \log n)}} \ .$$

**Claim.** There exist $C''_2$ and $C'''$ such that

$$C_0 \xrightarrow{\sigma_1} C''_1 \xrightarrow{t^{C''_1(q_s) - C'_1(q_s) + 1}} C''_2 \xrightarrow{\sigma_2} C''' \geq_E C \ .$$

**Proof of the claim.** Since $C''_1 \geq_{E'} C'_1$ and $C'_1$ enables $t$, so does $C''_1$. Since $\mathcal{P}$ is an IO protocol (a hypothesis we had not used so far), $C''_1$ enables not only $t$, but also the sequence $t^{C''_1(q_s) - C'_1(q_s) + 1}$. So there indeed exists a configuration $C''_2$ such that

$$C_0 \xrightarrow{\sigma_1} C''_1 \xrightarrow{t^{C''_1(q_s) - C'_1(q_s) + 1}} C''_2 \ .$$

It remains to prove that $C''_2 \xrightarrow{\sigma_2} C''' \geq_E C$ holds for some configuration $C'''$. First we show $C''_2 \geq_E C'_2$, which amounts to proving $C''_2 \geq_{E'} C'_2$ and $C''_2(q_s) = C'_2(q_s)$.

The first part, i.e., $C''_2 \geq_{E'} C'_2$, follows from: $C''_1 \xrightarrow{t^{C''_1(q_s) - C'_1(q_s) + 1}} C''_2$, $C''_1 \geq_{E'} C'_1$, $C'_1 \xrightarrow{t} C'_2$, $q_d \notin E$, which implies $q_d \notin E'$, and the fact that $t$ move agents from $q_s$ to $q_d$ (thus increasing their number in $q_d$). The second part, $C''_2(q_s) = C'_2(q_s)$, is proved by

$$C''_2(q_s) = C''_1(q_s) - (C''_1(q_s) - C'_1(q_s) + 1) = C'_1(q_s) - 1 = C'_2(q_s) \ .$$

So indeed we have $C''_2 \geq_E C'_2$. Now, since $C'_2$ enables $\sigma_2$ and $C''_2 \geq_E C'_2$, the configuration $C''_2$ enables $\sigma_2$ too. So there exists a configuration $C'''$ such that $C''_2 \xrightarrow{\sigma_2} C'''$. Further, since

$$\begin{array}{ccccc} C''_1 & \xrightarrow{t^{C''_1(q_s) - C'_1(q_s) + 1}} & C''_2 & \xrightarrow{\sigma_2} & C''' \\ \geq_{E'} & & \geq_E & & \\ C'_1 & \xrightarrow{\qquad t \qquad} & C'_2 & \xrightarrow{\sigma_2} & C'' \geq_E C \end{array}$$

holds, we have

$$\begin{array}{ccccc} C''_1 & \xrightarrow{t^{C''_1(q_s) - C'_1(q_s) + 1}} & C''_2 & \xrightarrow{\sigma_2} & C''' \\ \geq_{E'} & & \geq_E & & \geq_E \\ C'_1 & \xrightarrow{\qquad t \qquad} & C'_2 & \xrightarrow{\sigma_2} & C'' \geq_E C \end{array}$$

So $C''' \geq_E C'' \geq_E C$, and the claim is proved. ◀

By the claim we have $C_0 \xrightarrow{\sigma_1\, t^{C''_1(q_s) - C'_1(q_s) + 1}\, \sigma_2} C''' \geq_E C$ . Let $\sigma = \sigma_1 t^{C''_1(q_s) - C'_1(q_s) + 1} \sigma_2$. While $C''_1(q_s) - C'_1(q_s)$ can be arbitrarily large, we have $c(\sigma) = c(\sigma_1\, t\, \sigma_2)$, and so we conclude $C_0 \xrightarrow{\sigma} C''' \geq_E C$ and $|c(\sigma)| \in C(Q)^{2^{e\mathcal{O}(n \log n)}}$. ◀

Theorem 17 allows to derive the promised bounds on a constraint for $pre^*(\Gamma)$ and $post^*(\Gamma)$.

▶ **Theorem 18.** *Let $\mathcal{P}$ be an IO population protocol with $n$ states, and let $\Gamma$ be a CoNF-constraint. There exists a CoNF-constraint $\Gamma'$ satisfying $[\![\Gamma']\!] = pre^*(\Gamma)$, $\|\Gamma'\|_u \leq \|\Gamma\|_u$ and $\|\Gamma'\|_l \in \|\Gamma\|_u \left(\|\Gamma\|_l + \|\Gamma\|_u\right)^{2^{\mathcal{O}(n^2 \log n)}}$. Further, $\Gamma'$ can be constructed in $(2 + \|\Gamma\|_u)^n \cdot \|\Gamma\|_u \left(\|\Gamma\|_l + \|\Gamma\|_u\right)^{2^{\mathcal{O}(n^2 \log n)}}$ time and space. Further, the same holds for $post^*(\Gamma)$.*

**Proof.** The bound on $\|\Gamma'\|_u$ follows from Lemma 9. The bound on $\|\Gamma'\|_l$ is proved in a similar way to Proposition 13, but using Theorem 17 instead of Theorem 12. Let $(L, U)$ be a counting minterm in $\Gamma$. We define the set of states $E_{(L,U)} = \{q_i \mid U(x_i) < \infty\}$ and $\mathcal{C}^{\min}_{(L,U)} = \{C \mid \forall q_i \in Q \backslash E_{(L,U)}, L(x_i) \leq C(q_i) \leq U(x_i) \text{ and } \forall q_i \in E_{(L,U)}, C(q_i) = L(x_i)\}$ the configurations of $(L, U)$ minimal over $Q \backslash E_{(L,U)}$. Notice that a configuration is in $(L, U)$ if and only if it covers a configuration in $\mathcal{C}^{\min}_{(L,U)}$. By applying Theorem 17 to every $C \in \mathcal{C}^{\min}_{(L,U)}$ and to $E_{(L,U)}$, we get $pre^*(L, U) = \bigcup_{i=0}^{K} pre^i_a(L, U)$ for $K$ the bound in Theorem 17 but with $\left(\sum_{q_i \in Q\backslash E} L(x_i) + \sum_{q_i \in E} U(x_i)\right)$ instead of $C(Q)$. Now since $\Gamma$ is the union of such minterms $(L, U)$, and by definition of the $L$ and $U$-norms, $pre^*(\Gamma) = \bigcup_{i=0}^{K} pre^i_a(\Gamma)$ for $K \in (\|\Gamma\|_l + \|\Gamma\|_u)^{2^{\mathcal{O}(n^2 \log n)}}$. By Definition 8, we have $\|(L, U)_{t^*}\|_l \leq \|(L, U)\|_l + (\|(L, U)\|_u - 1)$. Using $\|\Gamma_{t^*}\|_u \leq \|\Gamma\|_u$, we reason by induction and get $\|pre^i_a(\Gamma)\|_l \leq \|\Gamma\|_l + i(\|\Gamma\|_u - 1)$ for all $i$, and the result on the $L$-norm follows.

The algorithm needs linear time and space in the number of minterms of $\Gamma'$. An upper bound on the number of minterms $(L, U)$ is computed as follows. Since $\|\Gamma'\|_l \in \|\Gamma\|_u (\|\Gamma\|_l + \|\Gamma\|_u)^{2^{\mathcal{O}(n^2 \log n)}}$, there are at most $(1 + \|\Gamma'\|_l)^n \in \|\Gamma\|_u (\|\Gamma\|_l + \|\Gamma\|_u)^{2^{\mathcal{O}(n^2 \log n)}}$ possibilities for $L$, and since $\|\Gamma'\|_u \leq \|\Gamma\|_u$ at most $(2 + \|\Gamma\|_u)^n$ possibilities for $U$. ◀

The following result characterizes the size of counting constraints.

▶ **Corollary 19.** *Let $\mathcal{P}$ be an IO protocol with $n$ states. Given $c \geq 2, d \geq 1$, let $\mathcal{G}(c, d)$ be the class of CoNF-constraints $\Gamma$ such that $\|\Gamma\|_l, \|\Gamma\|_u \leq c^{2^{d\cdot(n^2 \log n)}}$. There exists a constant $k$ that does not depend on $n$ or $\mathcal{P}$ such that :*

1. *for every $\Gamma_1, \Gamma_2 \in \mathcal{G}(c, d)$, there exists $\Gamma \in \mathcal{G}(c, d)$ such that $[\![\Gamma]\!] = [\![\Gamma_1]\!] \cup [\![\Gamma_2]\!]$.*
2. *for every $\Gamma_1, \Gamma_2 \in \mathcal{G}(c, d)$, there exists $\Gamma \in \mathcal{G}(c, d + 1)$ such that $[\![\Gamma]\!] = [\![\Gamma_1]\!] \cap [\![\Gamma_2]\!]$.*
3. *for every $\Gamma_1 \in \mathcal{G}(c, d)$, there exists $\Gamma \in \mathcal{G}(c, d + 1)$ such that $[\![\Gamma]\!] = \mathbb{N}^n \setminus [\![\Gamma_1]\!]$.*
4. *for every $\Gamma_1 \in \mathcal{G}(c, d)$, there exists $\Gamma \in \mathcal{G}(c, d + k + 2)$ such that $[\![\Gamma]\!] = pre^* ([\![\Gamma_1]\!])$.*
5. *for every $\Gamma_1 \in \mathcal{G}(c, d)$, there exists $\Gamma \in \mathcal{G}(c, d + k + 2)$ such that $[\![\Gamma]\!] = post^* ([\![\Gamma_1]\!])$.*

The first three bounds follow from Prop 5. For the last two, the constant $k$ is the one from the Landau symbol in Theorem 18.

## 5 An Algorithm for Deciding Well Specification

We show that the well-specification and correctness problems can be solved in exponential space for IO protocols, improving on the result for general protocols stating that they are at least as hard as the reachability problem for Petri nets [9]. We first introduce some notions.

▶ **Definition 20.** Given a population protocol $\mathcal{P}$, a configuration $C$ is a *stable b-consensus* if $C$ is a $b$-consensus and so is $C'$ for every $C'$ reachable from $C$. Let $\mathcal{C}_b$ and $\mathcal{ST}_b$ denote the sets of $b$-consensus and stable $b$-consensus configurations of $\mathcal{P}$. Observe that $\mathcal{ST}_b = \overline{pre^*(\overline{\mathcal{C}_b})}$.

Next, we characterize the well-specified protocols starting with the following lemma.

▶ **Lemma 21.** *Let $\mathcal{P}$ be a population protocol, let $C_0, C_1, C_2, \ldots$ be a fair execution of $\mathcal{P}$, and let $S$ be a set of configurations. If $S$ is reachable from $C_i$ for infinitely many indices $i \geq 0$, then $C_j \in S$ for infinitely many indices $j \geq 0$.*

**Proof.** Let $n$ be the number of states of $\mathcal{P}$ and let $m$ be the number of agents of $C_0$. Then there are at most $K \stackrel{\text{def}}{=} (m + 1)^n$ configurations reachable from $C_0$. So for infinitely many indices $i \geq 0$ we have $C_i \in \cup_{i \leq K} pre^i(S)$. We proceed by induction on $K$. If $K = 0$, then

$C_i \in S$ and we are done. If $K > 0$, then by fairness there exist infinitely many indices $j \geq 0$ such that $C_j \in \cup_{i \leq K-1} pre^i(S)$, and we conclude by induction hypothesis. ◀

▶ **Proposition 22.** *A population protocol $\mathcal{P}$ is well-specified iff the following hold:*
1. $post^*(\mathcal{I}) \subseteq pre^*(\mathcal{ST}_0 \cup \mathcal{ST}_1)$ *(or, equivalently,* $post^*(\mathcal{I}) \cap \overline{pre^*(\mathcal{ST}_0)} \cap \overline{pre^*(\mathcal{ST}_1)} = \emptyset$ *);*
2. $pre^*(\mathcal{ST}_0) \cap pre^*(\mathcal{ST}_1) \cap \mathcal{I} = \emptyset$.

**Proof.** We start with $\mathcal{ST}_b$ which is defined (Definition 20) as the set of configurations $C$ such that $C$ is a $b$-consensus and so is $C'$ for every $C'$ reachable from $C$.

By definition, $\mathcal{P}$ is *well-specified* if for every input configuration $C_0 \in \mathcal{I}$, every fair execution of $\mathcal{P}$ starting at $C_0$ stabilizes to the same value $b \in \{0, 1\}$. Equivalently, $\mathcal{P}$ is *well-specified* if every input configuration $C_0 \in \mathcal{I}$ satisfies the following two conditions:

**(a)** every fair execution starting at $C_0$ stabilizes to some value; and

**(b)** no two fair executions starting at $C_0$ stabilize to different values (i.e., to 0 and to 1 ).

We claim that **(a)** is equivalent to:

$$\text{for every } C \in post^*(\mathcal{I}) \text{ there exists } C' \text{ such that } C \xrightarrow{*} C' \text{ and } C' \in \mathcal{ST}_0 \cup \mathcal{ST}_1. \quad \textbf{(A)}$$

Assume **(a)** holds, and let $C \in post^*(\mathcal{I})$. Then $C_0 \xrightarrow{*} C$ for some $C_0 \in \mathcal{I}$. Extend $C_0 \xrightarrow{*} C$ to a fair execution. By **(a)**, the execution stabilizes to some value $b$. So $\mathcal{ST}_b$ is reachable from every configuration of the execution. By Lemma 21, the execution reaches a configuration $C' \in \mathcal{ST}_b$. For the other direction, assume **(A)** holds, and consider a fair execution starting at $C_0 \in \mathcal{I}$. By Lemma 21, the execution reaches a configuration of $\mathcal{ST}_b$ for $b \in \{0, 1\}$. By the definition of $\mathcal{ST}_b$, all successor configurations also belong to $\mathcal{ST}_b$, and so the execution stabilizes to $b$. Now we claim that **(b)** is equivalent to:

$$\text{no configuration } C \in post^*(\mathcal{I}) \text{ can reach both } \mathcal{ST}_0 \text{ and } \mathcal{ST}_1. \quad \textbf{(B)}$$

Assume **(B)** does not hold, i.e., there is $C \in post^*(\mathcal{I})$ and configurations $C_0 \in \mathcal{ST}_0$ and $C_1 \in \mathcal{ST}_1$ such that $C \xrightarrow{*} C_0$ and $C \xrightarrow{*} C_1$. These two executions can be extended to fair executions, and by the definition of $\mathcal{ST}_0$ and $\mathcal{ST}_1$ these executions stabilize to 0 and 1, respectively. So **(b)** does not hold.

Assume now that **(b)** does not hold. Then two fair executions starting at $C_0$ stabilize to different values. So $C_0$ can reach both $\mathcal{ST}_0$ and $\mathcal{ST}_1$, and **(B)** does not hold.

So **(a)** and **(b)** are equivalent to **(A)** and **(B)**. Since **(A)** is equivalent to $post^*(\mathcal{I}) \subseteq pre^*(\mathcal{ST}_0 \cup \mathcal{ST}_1)$, and **(B)** is equivalent to $pre^*(\mathcal{ST}_0) \cap pre^*(\mathcal{ST}_1) \cap \mathcal{I} = \emptyset$, we are done. ◀

▶ **Theorem 23.** *The well specification problem for IO protocols is in* EXPSPACE *and is* PSPACE*-hard.*

**Proof.** Let $\mathcal{P}$ be an IO protocol with $n$ states. Recall that $\mathcal{ST}_b$ is given by $\overline{pre^*(\overline{\mathcal{C}_b})}$ where $\mathcal{C}_b$, for $b \in \{0, 1\}$, can be represented by the CoNF-constraint of single minterm defined by $x_i = 0$ for all $q_i \in O^{-1}(1-b)$ and $0 \leq x_i \leq \infty$ otherwise. By Corollary 19, there exists a constant $d$, independent of $\mathcal{P}$, and a CoNF constraint $\Gamma \in \mathcal{G}(2, d)$ such that $[\![\Gamma]\!]$ is given by $post^*(\mathcal{I}) \cap \overline{pre^*(\mathcal{ST}_0)} \cap \overline{pre^*(\mathcal{ST}_1)}$.

In order to falsify condition **1.** of Proposition 22 it suffices to exhibit, following the previous reasoning, a "small" configuration $C$, such that $C(Q) \leq c^{2^{d \cdot (n^2 \log n)}}$, in the intersection. Note that $C$ can be written in EXPSPACE. The EXPSPACE decision procedure follows the following steps: **1.** Guess a "small" configuration $C$. **2.** Check that $C$ belongs to $post^*(\mathcal{I})$. **3.** Check that $C$ belongs to $\overline{pre^*(\mathcal{ST}_b)}$, for $b = 0, 1$.

Algorithm for **2.**: Guess a at most double exponential sequence of minterms such that the first one is a minterm of $\mathcal{I}$, and every pair of consecutive minterms is related by $post^*[t]$

(given by Definition 8) for some $t$. Observe that we keep track of the last computed element and the number of steps performed so far in exponential space. Then, check that $C$ belongs to the resulting minterm.
Algorithm for **3.**: It follows from EXPSPACE = coEXPSPACE that it is equivalent to check $C \in pre^*(\mathcal{ST}_b)$ is in EXPSPACE. Our algorithm is divided in two steps.
Step 1. Let $c, d$ be such that $\mathcal{ST}_b \in \mathcal{G}(c, d)$. Guess a minterm $M$ in $\mathcal{G}(c, d)$ and proceed similarly to Algorithm for **2.** to compute a minterm of $pre^*(M)$ and then check that $C$ belongs to the resulting minterm.
Step 2. Verify that $M$ does indeed belong to $\mathcal{ST}_b$. Formally, we rely on the following equivalences: $[\![M]\!] \subseteq \mathcal{ST}_b$ iff $[\![M]\!] \subseteq \overline{pre^*(\overline{\mathcal{C}_b})}$ iff $[\![M]\!] \cap pre^*(\overline{\mathcal{C}_b}) = \emptyset$. Using EXPSPACE = coEXPSPACE we now show that $[\![M]\!] \cap pre^*(\overline{\mathcal{C}_b}) \neq \emptyset$ belongs to EXPSPACE. We nondeterministically choose a minterm in $\overline{\mathcal{C}_b}$ and as previously explained guess a minterm in $pre^*(\overline{\mathcal{C}_b})$. Finally, we check whether it intersects with $[\![M]\!]$.

We use a similar reasoning for checking in EXPSPACE condition **2.** of Proposition 22.

The proof for PSPACE-hardness reduces from the acceptance problem for deterministic Turing machines running in linear space [13]. The proof follows the structure of analogous proofs for 1-safe Petri nets [11] (and also [8]) and will be provided in the full version. ◀

## 5.1 Consequences

In this section we list some consequences of Theorem 18 and Theorem 23.

In [4], Angluin *et al.* showed that IO protocols can compute exactly the counting predicates, i.e., the predicates that can be expressed by counting constraints. This is also a consequence of the proof of Theorem 23. Moreover, our results allow us to go further, and provide a bound on the number of states required to compute a predicate.

▶ **Corollary 24.** *IO population protocols compute exactly the counting predicates, i.e., the predicates corresponding to counting constraints.*

**Proof.** Let $\mathcal{P}$ be a well-specified IO protocol. The sets $\mathcal{I} \cap \overline{pre^*(\overline{pre^*(\mathcal{ST}_b)})}$ for $b \in \{0, 1\}$ are the sets of initial configurations from which $\mathcal{P}$ stabilizes to $b = 0, 1$. Theorem 18 shows that they are counting sets. ◀

▶ **Corollary 25.** *Let $\mathcal{P}$ be an IO protocol computing a counting predicate $P(x_1, \ldots, x_k)$ of $U$-norm $u$ and $L$-norm $\ell$. Then there exists a constant $c$, independent of $\mathcal{P}$, such that $\mathcal{P}$ has at least $g \log\log(\max\{u, \ell\})$ states, where $g$ denotes the inverse of the function $n \mapsto c \cdot (n^2 \log n)$.*

**Proof.** The set $\mathcal{I} \cap \overline{pre^*(\overline{pre^*(\mathcal{ST}_1)})}$ describes the initial configurations that stabilize to 1, i.e., the initial configurations for which the predicate computed by the protocol is true. By Corollary 19 (using a reasoning similar to that of Theorem 23), if $\mathcal{P}$ has $n$ states, then the $U$-norm and $L$-norm of $\mathcal{I} \cap \overline{pre^*(\overline{pre^*(\mathcal{ST}_1)})}$ are bounded by the function $f(n) = 2^{2^{\mathcal{O}(n^2 \log n)}}$. Therefore, for a certain constant $c$, $\log\log\max\{u, \ell\} \leq c \cdot (n^2 \log n)$ and the number of states of a protocol computing a predicate of $U$-norm $u$ and $L$-norm $\ell$ is at least $g \log\log(\max\{u, \ell\})$, where $g(x)$ is the inverse function of $x \mapsto c \cdot (x^2 \log x)$. ◀

Finally, we can show that the correctness problem for IO protocols is also in EXPSPACE.

▶ **Corollary 26.** *Let $\mathcal{P}$ be an IO population protocol with $n$ states and $k$ input states, and let $P(x_1, \ldots, x_k)$ be a counting predicate, expressed as a CoNF-constraint. The correctness problem for $\mathcal{P}$ and $P$, i.e., the problem of deciding if $\mathcal{P}$ computes $P$, is in* EXPSPACE.

**Proof Sketch.** We give a nondeterministic, exponential space algorithm for the complement of the correctness problem. The algorithm guesses nondeterministically a minterm of $\mathcal{I} \cap pre^*(\overline{pre^*(\overline{\mathcal{ST}_1})})$, and checks that $\mathcal{I} \cap pre^*(\overline{pre^*(\overline{\mathcal{ST}_1})})$ contains a configuration that does not satisfy $P$. The algorithm does a similar check for $\mathcal{ST}_0$ and a configuration that does satisfy $P$. The minterm can be constructed in exponential space by Theorem 23, and the check whether a minterm implies a CoNF-constraint can be done in polynomial time. ◀

## References


1 Parosh A. Abdulla and Aletta Nylén. Better is better than well: on efficient verification of infinite-state systems. In *LICS '00*. IEEE Comput. Soc, 2000. `doi:10.1109/lics.2000.855762`.

2 Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. In *PODC '04*, pages 290–299. ACM, 2004. `doi:10.1145/1011767.1011810`.

3 Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18(4):235–253, 2006. `doi:10.1007/s00446-005-0138-3`.

4 Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. The computational power of population protocols. *Distributed Computing*, 20(4):279–304, 2007. `doi:10.1007/s00446-007-0040-2`.

5 Michael Blondin, Javier Esparza, and Stefan Jaax. Large flocks of small birds: on the minimal size of population protocols. In *STACS '18*, volume 96, pages 16:1–16:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. `doi:10.4230/LIPIcs.STACS.2018.16`.

6 Laura Bozzelli and Pierre Ganty. Complexity analysis of the backward coverability algorithm for vass. In *RP '11*, volume 6945 of *LNCS*, pages 96–109. Springer, 2011. `doi:10.1007/978-3-642-24288-5_10`.

7 Ioannis Chatzigiannakis, Othon Michail, and Paul G. Spirakis. Algorithmic verification of population protocols. In *SSS '10*, volume 6366 of *LNCS*, pages 221–235. Springer, 2010. `doi:10.1007/978-3-642-16023-3_19`.

8 Allan Cheng, Javier Esparza, and Jens Palsberg. Complexity results for 1-safe nets. *Theoretical Computer Science*, 147(1&2):117–136, 1995. `doi:10.1016/0304-3975(94)00231-7`.

9 Javier Esparza, Pierre Ganty, Jérôme Leroux, and Rupak Majumdar. Verification of population protocols. In *CONCUR '15*, volume 42 of *LIPIcs*, pages 470–482. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015. `doi:10.4230/LIPIcs.CONCUR.2015.470`.

10 Javier Esparza, Pierre Ganty, Jérôme Leroux, and Rupak Majumdar. Model checking population protocols. In *FSTTCS '16*, volume 65. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016. `doi:10.4230/lipics.fsttcs.2016.27`.

11 Neil D. Jones, Lawrence H. Landweber, and Y. Edmund Lien. Complexity of some problems in petri nets. *Theoretical Computer Science*, 4(3):277–299, 1977. `doi:10.1016/0304-3975(77)90014-7`.

12 Saket Navlakha and Ziv Bar-Joseph. Distributed information processing in biological and computational systems. *Commun. ACM*, 58(1):94–102, 2014. `doi:10.1145/2678280`.

13 Christos H. Papadimitriou. *Computational complexity*. Academic Internet Publ., 2007.

14 Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6(2):223–231, 1978. `doi:10.1016/0304-3975(78)90036-1`.