Parameterized Analysis of Reconfigurable Broadcast Networks*

A. R. Balasubramanian[1], Lucie Guillou[2], and Chana Weil-Kennedy[3](✉)

[1] Technical University of Munich bala.ayikudi@tum.de
[2] ENS Rennes lucie.guillou@ens-rennes.fr
[3] Technical University of Munich chana.weilkennedy@in.tum.de

Abstract. Reconfigurable broadcast networks (RBN) are a model of distributed computation in which agents can broadcast messages to other agents using some underlying communication topology which can change arbitrarily over the course of executions. In this paper, we conduct *parameterized analysis* of RBN. We consider cubes, (infinite) sets of configurations in the form of lower and upper bounds on the number of agents in each state, and we show that we can evaluate boolean combinations over cubes and reachability sets of cubes in PSPACE. In particular, reachability from a cube to another cube is a PSPACE-complete problem.
To prove the upper bound for this parameterized analysis, we prove some structural properties about the reachability sets and the *symbolic graph* abstraction of RBN, which might be of independent interest. We justify this claim by providing two applications of these results. First, we show that the almost-sure coverability problem is PSPACE-complete for RBN, thereby closing a complexity gap from a previous paper [3]. Second, we define a computation model using RBN, à la population protocols, called RBN protocols. We characterize precisely the set of predicates that can be computed by such protocols.

Keywords: Broadcast networks · Parameterized reachability · Almost-sure coverability · Asynchronous shared-memory systems

1 Introduction

Reconfigurable broadcast networks (RBN) [8,10] are a formalism for modelling distributed systems in which a set of anonymous, finite-state agents execute the same underlying protocol and broadcast messages to their neighbors according to an underlying communication topology. The communication topology is reconfigurable, meaning that the set of neighbors of an agent can change arbitrarily over the course of an execution. Parameterized verification of these networks concerns itself with proving that a given property is correct, irrespective of the number of participating agents. Dually, it can be viewed as the problem of finding an

* This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 787367 (PaVeS).

P. Bouyer and L. Schröder (Eds.): FoSSaCS 2022, LNCS 13242, pp. 61–80, 2022.
https://doi.org/10.1007/978-3-030-99253-8_4

execution of some number of agents which violates a given property. Ever since their introduction within this context [10], RBN have been studied extensively, with various results on (parameterized) reachability and coverability [8,10,3,7], along with various extensions using probabilities and clocks [5,4].

In this paper, we first consider the *cube-reachability* problem for RBN, in which we are given two (possibly infinite) sets of configurations $\mathcal{C}$ and $\mathcal{C}'$ (called *cubes*), each of them defined by lower and upper bounds on the number of agents in each state, and we must decide if there is a configuration in $\mathcal{C}$ which can reach some configuration in $\mathcal{C}'$. The cube-reachability question covers parameterized reachability and coverability problems, and as explained in [3], also covers the parameterized reachability problem for a generalized model of RBN called *RBN with leaders*. Moreover, a sub-problem of cube-reachability has already been studied for RBN in [8]. The authors show that this sub-problem is PSPACE-complete. One of the results in our paper is that the entire cube-reachability problem is PSPACE-complete, hence extending the sub-problem considered in [8], while still retaining the same complexity upper bound.

In fact, our main result, which we call the PSPACE Theorem, is a more general result. It subsumes the above result for cube-reachability and allows for more complex parameterized analysis of RBN. The PSPACE Theorem roughly states that any boolean combination of atoms can be evaluated in PSPACE, where an atom is a finite union of cubes or the reachability set of a finite union of cubes (i.e. $post^*$ or pre^*). To prove the PSPACE Theorem, we first consider the so called *symbolic graph* of a RBN ([8], Section 5). We prove some structural properties about these graphs, using results from [8]. Next, using these structural properties, we show that the set of reachable configurations of a cube $\mathcal{C}$ can be expressed as a finite union of cubes, each having a norm exponentially bounded in the size of the given RBN and $\mathcal{C}$. This result then allows us to give an on-the-fly exploration algorithm for proving the PSPACE Theorem.

We believe that the PSPACE Theorem and the results leading to it that we have proven in this paper have further applications to problems concerning RBN. To justify this claim, we provide two applications. First, we show that the almost-sure coverability problem for RBN is PSPACE-complete, thereby closing a complexity gap from a previous paper ([3], Section 5.3). Second, we define a computation model using RBN, called RBN protocols, which is similar in spirit to the population protocols model [1,2]. We characterize precisely the set of predicates that can be computed using RBN protocols. This result generalizes the corresponding result for IO protocols, which are a sub-class of population protocols that can be simulated by RBN protocols, as shown in ([3], Section 6.2).

Finally, by the reduction given in ([3], Section 4.2), our results on cube-reachability and almost-sure coverability can be transferred to another model of distributed computation called asynchronous shared memory systems (ASMS), giving a PSPACE-completeness result for both of these problems. This solves an open problem from ([6], Section 6).

To summarize, we have shown that many important parameterized problems of RBN can be solved in PSPACE, that the sub-problem of the cube-reachability problem defined in [8] can be generalized while retaining the same upper bounds, and that the almost-sure coverability problems for RBN and ASMS are PSPACE-complete, thereby solving open problems from [3,6]. We believe that our other results might be of independent interest, and we provide an application by introducing RBN protocols and characterizing the set of predicates that they can compute.

The paper is organized as follows. Section 2 contains preliminaries, including the definition of RBN. Section 3 defines the symbolic graph of a RBN, and proves the properties of this graph needed to derive our main result. Section 4 contains the main result that a host of parameterized problems over cubes, including cube-reachability, is PSPACE-complete for RBN. Finally, Sections 5 and 6 give applications of our main results: Section 5 solves the complexity gap for the almost-sure coverability problem, and Section 6 introduces RBN protocols and characterizes their expressive power. Due to lack of space, full proofs of some of the results can be found in the long version.

2 Preliminaries

The definitions and notations in this section are taken from [3].

2.1 Multisets

A *multiset* on a finite set E is a mapping $C\colon E \to \mathbb{N}$, i.e. for any $e \in E$, $C(e)$ denotes the number of occurrences of element e in C. We let $\mathbb{M}(E)$ denote the set of all multisets on E. Let $\wr e_1, \ldots, e_n \int$ denote the multiset C such that $C(e) = |\{j \mid e_j = e\}|$. We sometimes write multisets using set-like notation. For example, $\wr 2 \cdot a, b \int$ and $\wr a, a, b \int$ denote the same multiset. Given $e \in E$, we denote by $\boldsymbol{e}$ the multiset consisting of one occurrence of element e, that is $\wr e \int$. Operations on $\mathbb{N}$ like addition or comparison are extended to multisets by defining them component wise on each element of E. Subtraction is allowed as long as each component stays non-negative. We call $|C| \overset{\text{def}}{=} \sum_{e \in E} C(e)$ the *size* of C.

2.2 Reconfigurable Broadcast Networks

Reconfigurable broadcast networks (RBN) are networks consisting of finite-state, anonymous agents and a communication topology which specifies for every pair of processes, whether or not there is a communication link between them. During a single step, a single agent can broadcast a message which is received by all of its neighbors, after which both the agent and its neighbors change their state according to some transition relation. Further, in between two steps, the communication topology can change in an arbitrary manner. For the problems that we consider in this paper, it is easier to forget the communication topology and define the semantics of an RBN directly in terms of collections of agents.

Definition 1. *A* reconfigurable broadcast network *is a tuple* $\mathcal{R} = (Q, \Sigma, \delta)$ *where* Q *is a finite set of states,* Σ *is a finite alphabet and* $\delta \subseteq Q \times \{!a, ?a \mid a \in \Sigma\} \times Q$ *is the transition relation.*

If $(p, !a, q)$ (resp. $(p, ?a, q)$) is a transition in δ, we will denote it by $p \xrightarrow{!a} q$ (resp. $p \xrightarrow{?a} q$). A *configuration* C of a RBN $\mathcal{R}$ is a multiset over Q, which intuitively counts the number of processes in each state. Given a letter $a \in \Sigma$ and two configurations C and C' we say that there is a *step* $C \xrightarrow{a} C'$ if there exists a multiset $\wr t, t_1, \ldots, t_k \wr$ of δ for some $k \geq 0$ satisfying the following: $t = p \xrightarrow{!a} q$, each $t_i = p_i \xrightarrow{?a} q_i$, $C \geq \boldsymbol{p} + \sum_i \boldsymbol{p_i}$, and $C' = C - \boldsymbol{p} - \sum_i \boldsymbol{p_i} + \boldsymbol{q} + \sum_i \boldsymbol{q_i}$. We sometimes write this as $C \xrightarrow{t+t_1,\ldots,t_n} C'$ or $C \xrightarrow{a} C'$. Intuitively it means that a process at the state p broadcasts the message a and moves to q, and for each $1 \leq i \leq k$, there is a process at the state p_i which receives this message and moves to q_i. We denote by $\xrightarrow{*}$ the reflexive and transitive closure of the step relation. A *run* is then a sequence of steps.

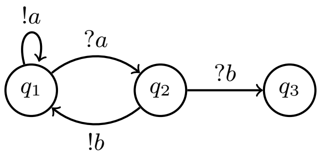

Fig. 1. An RBN $\mathcal{R}$ with three states.

Let $\mathcal{R} = (Q, \Sigma, \delta)$ be an RBN. Given configurations C and C', we say C' is *reachable* from C if $C \xrightarrow{*} C'$. We say C' is *coverable* from C if there exists C'' such that $C \xrightarrow{*} C''$ and $C'' \geq C'$. The *reachability* problem consists of deciding, given a RBN $\mathcal{R}$ and configurations C, C', whether C' is reachable from C in $\mathcal{R}$. The *coverability* problem consists of deciding, given a RBN $\mathcal{R}$ and configurations C, C', whether C' is coverable from C in $\mathcal{R}$. Let $\mathcal{S}$ be a set of configurations. The *predecessor set* of $\mathcal{S}$ is *pre*$^*(\mathcal{S}) \stackrel{\text{def}}{=} \{C' | \exists C \in \mathcal{S} \,.\, C' \xrightarrow{*} C\}$, and the *successor set* of $\mathcal{S}$ is *post*$^*(\mathcal{S}) \stackrel{\text{def}}{=} \{C | \exists C' \in \mathcal{S} \,.\, C' \xrightarrow{*} C\}$.

Example 1. Figure 1 illustrates a RBN $\mathcal{R} = (Q, \Sigma, \delta)$ with $Q = \{q_1, q_2, q_3\}$. Configuration $\wr 3 \cdot q_1 \wr$ can reach $\wr 2 \cdot q_1, q_3 \wr$ in two steps. First, a process broadcasts a, the two other processes receive it and move to q_2. Then, one of the processes in q_2 broadcasts b and moves to q_1, while the other one receives b and moves to q_3. Notice that $\wr q_3 \wr$ is only coverable from a configuration $\wr k \cdot q_1 \wr$ if $k \geq 3$.

2.3 Cubes and Counting Sets

Given a finite set Q, a *cube* $\mathcal{C}$ is a subset of $\mathbb{M}(Q)$ described by a lower bound $L \colon Q \to \mathbb{N}$ and an upper bound $U \colon Q \to \mathbb{N} \cup \{\infty\}$ such that $\mathcal{C} = \{C : L \leq C \leq$

$U\}$. Abusing notation, we identify the set $\mathcal{C}$ with the pair (L, U). Notice that since $U(q)$ can be ∞ for some state q, a cube can contain an infinite number of configurations. All the results in this paper are true irrespective of whether the constants in a given input cube are encoded in unary or binary.

A finite union of cubes $\bigcup_{i=1}^{m}(L_i, U_i)$ is called a *counting constraint* and the set of configurations $\bigcup_{i=1}^{m}\mathcal{C}_i$ it describes is called a *counting set*. Notice that two different counting constraints may describe the same counting set. For example, let $Q = \{q\}$ and let $(L, U) = (1, 3)$, $(L', U') = (2, 4)$, $(L'', U'') = (1, 4)$. The counting constraints $(L, U) \cup (L', U')$ and (L'', U'') define the same counting set. It is easy to show (see also Proposition 2 of [11]) that counting constraints and counting sets are closed under Boolean operations.

Norms. Let $\mathcal{C} = (L, U)$ be a cube. Let $\|\mathcal{C}\|_l$ be the the sum of the components of L. Let $\|\mathcal{C}\|_u$ be the sum of the finite components of U if there are any, and 0 otherwise. The *norm* of $\mathcal{C}$ is the maximum of $\|\mathcal{C}\|_l$ and $\|\mathcal{C}\|_u$, denoted by $\|\mathcal{C}\|$. We define the norm of a counting constraint $\Gamma = \bigcup_{i=1}^{m}\mathcal{C}_i$ as $\|\Gamma\| \stackrel{\text{def}}{=} \max_{i\in[1,m]}\{\|\mathcal{C}_i\|\}$. The norm of a counting set $\mathcal{S}$ is the smallest norm of a counting constraint representing $\mathcal{S}$, that is, $\|\mathcal{S}\| \stackrel{\text{def}}{=} \min_{\mathcal{S}=[\![\Gamma]\!]}\{\|\Gamma\|\}$. Proposition 5 of [11] entails the following results for the norms of the union, intersection and complement.

Proposition 1. *Let $\mathcal{S}_1, \mathcal{S}_2$ be counting sets. The norms of the union, intersection and complement satisfy: $\|\mathcal{S}_1 \cup \mathcal{S}_2\| \leq \max\{\|\mathcal{S}_1\|, \|\mathcal{S}_2\|\}$, $\|\mathcal{S}_1 \cap \mathcal{S}_2\| \leq \|\mathcal{S}_1\| + \|\mathcal{S}_2\|$, and $\|\overline{\mathcal{S}_1}\| \leq |Q| \cdot \|\mathcal{S}_1\| + |Q|$.*

Reachability. The reachability problem can be generalized to the *cube-reachability* problem which consists of deciding, given an RBN $\mathcal{R}$ and two cubes $\mathcal{C}, \mathcal{C}'$, whether there exists configurations $C \in \mathcal{C}$ and $C' \in \mathcal{C}'$ such that C' is reachable from C in $\mathcal{R}$. If this is the case, we say $\mathcal{C}'$ is reachable from $\mathcal{C}$. The *counting set-reachability* problem asks, given an RBN $\mathcal{R}$ and two counting sets $\mathcal{S}, \mathcal{S}'$, whether there exists cubes $\mathcal{C} \in \mathcal{S}$ and $\mathcal{C}' \in \mathcal{S}'$ such that $\mathcal{C}'$ is reachable from $\mathcal{C}$ in $\mathcal{R}$. We define *cube-coverability* and *counting set-coverability* in an analoguous way.

Remark 1. In the paper [8], the authors define a sub-class of the cube-reachability problem, which is called the *unbounded initial cube-reachability* problem in [3]. More precisely, the sub-class considered in [8] is the following: We are given an RBN and two cubes $\mathcal{C} = (L, U)$ and $\mathcal{C}' = (L', U')$ with the special property that $L(q) = 0$ and $U(q) \in \{0, \infty\}$ for every state q. We then have to decide if $\mathcal{C}$ can reach $\mathcal{C}'$. This problem was shown to be PSPACE-complete ([8], Theorem 5.5), whenever the numbers in the input are given in unary. As we shall show later in this paper, the cube-reachability problem itself is in PSPACE, even when the input numbers are encoded in binary, thereby generalizing the upper bound results given in that paper.

3 Reachability sets of counting sets

In this section, we set the stage for proving the main result of this paper. This main result is given in two stages: First, we show that given a RBN with state set Q and a counting set $\mathcal{S}$, the set $post^*(\mathcal{S})$ is also a counting set and $\|post^*(\mathcal{S})\| \leq 2^{p(\|\mathcal{S}\| \cdot |Q|)}$ where p is some fixed polynomial. Using this, we then prove that a host of cube-parameterized problems for RBN can be solved in PSPACE.

The rest of this section is organized as follows: To prove the first result, we recall the notion of a *symbolic graph* of a RBN from [8]. In the symbolic graph, each node is a *symbolic configuration* of the RBN, which intuitively represents an infinite set of configurations in which the number of agents is fixed in some states, and arbitrarily big in the others. Next, by exploiting the special structure of the symbolic graph, we prove some properties which allow us to show that whenever two nodes in this graph are reachable, they are reachable by a path having a special structure. Finally, using these properties and the connection between symbolic configurations and configurations of the RBN, we prove the desired first result. Once we have shown the first result, we then show how the PSPACE Theorem can be obtained from it.

Throughout this section, we fix an RBN $\mathcal{R} = (Q, \Sigma, \delta)$.

3.1 Symbolic graph

In this subsection, we recall the notion of a *symbolic graph* of an RBN from [8]. Here, for the sake of convenience, we define it in a slightly different way, but the underlying notion is the same as [8]. Throughout this subsection and the next, we fix a number $k \in \mathbb{N}$.

The *symbolic graph of index k* associated with the RBN $\mathcal{R}$ is an edge-labelled graph $\mathcal{G}_k = (N, E, L)$ where $N = \mathbb{M}_k(Q) \times 2^Q$ is the set of nodes. Here $\mathbb{M}_k(Q)$ denotes the set of multisets on Q of size at most k. E is the set of edges and $L : E \to \Sigma$ is the labelling function. Each node of $\mathcal{G}_k$ is also called a *symbolic configuration*. Intuitively, in each symbolic configuration (v, S), the multiset v (called the *concrete part*) is used to keep track of a fixed set of at most k agents, and the subset S (called the *abstract part*) is used to keep track of the *support* of the remaining agents.

Let $\theta = (v, S)$ and $\theta' = (v', S')$ be two symbolic configurations. There is an edge labelled by a between θ and θ' if and only if the following is satisfied: There exists a transition $(q, !a, q') \in \delta$ such that at least one of the following two conditions holds

- *(Broadcast from v)* There exists a multiset of transitions $\wr(p_1, ?a, p'_1), \ldots, (p_l, ?a, p'_l)\wr$ such that $v' = v - \sum_i \boldsymbol{p_i} + \sum_i \boldsymbol{p'_i} - \boldsymbol{q} + \boldsymbol{q'}$, and for each $q_s \in Q$:
  - If $q_s \in S' \setminus S$ then there exists $q'_s \in S$ and $(q'_s, ?a, q_s) \in R$,
  - If $q_s \in S \setminus S'$ then there exists $q'_s \in S'$ and $(q_s, ?a, q'_s) \in R$.
- *(Broadcast from S)* There exists a multiset of transitions $\wr(p_1, ?a, p'_1), \ldots, (p_l, ?a, p'_l)\wr$ such that $v' = v - \sum_i \boldsymbol{p_i} + \sum_i \boldsymbol{p'_i}$, $q \in S, q' \in S'$, and for each $q_s \in Q \setminus \{q, q'\}$:

- if $q_s \in S' \setminus S$ then there exists $q'_s \in S$ and $(q'_s, ?a, q_s) \in R$,
- if $q_s \in S \setminus S'$ then there exists $q'_s \in S'$ and $(q_s, ?a, q'_s) \in R$.

An edge labelled by a between θ and θ' is denoted by $\theta \rightsquigarrow^{a}_{\mathcal{G}_k} \theta'$. The relation $\rightsquigarrow^{*}_{\mathcal{G}_k}$ is the reflexive and transitive closure of $\rightsquigarrow_{\mathcal{G}_k} := \cup_{a \in \Sigma} \rightsquigarrow^{a}_{\mathcal{G}_k}$. Whenever the index k is clear, we will drop the subscript $\mathcal{G}_k$ from these notations.

Remark 2. Let $\theta = (v, S), \theta' = (v', S')$ be two symbolic configurations. By construction, θ can only reach θ' if $|v| = |v'|$.

To give an intuition behind the edges in $\mathcal{G}_k$, recall the intuition that in a symbolic configuration, the concrete part is used to keep track of a fixed set of at most k processes and the abstract part is used to keep track of the support of the remaining processes. The first condition for the existence of an edge asserts the following: 1) In the concrete part, some process broadcasts the message a and some subset of processes receive a, 2) In the abstract part, any new state added or any old state deleted comes because of receiving a. The second condition asserts exactly the same, except we now require the process broadcasting the message a to be from the abstract part.

The symbolic graph of index k can be thought of as an abstraction of the set of configurations of $\mathcal{R}$, where only a fixed number of processes are explicitly represented and the rest are abstracted by means of their support alone. To formalize this, given a symbolic configuration $\theta = (v, S)$, we let $[\![\theta]\!]$ denote the following (infinite) set of configurations: $C \in [\![\theta]\!]$ if and only if $C(q) = v(q)$ for $q \notin S$ and $C(q) \geq v(q)$ for $q \in S$.

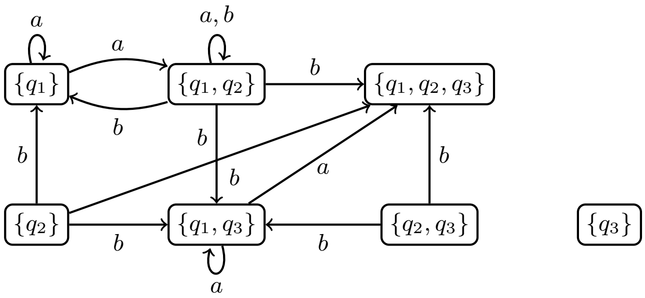

Fig. 2. Symbolic graph $\mathcal{G}_0$ of index 0 of the RBN of Example 1.

Example 2. The symbolic graph $\mathcal{G}_0$ of index 0 of the RBN of Example 1 is illustrated in Figure 2. At this index, the graph only keeps track of a subset $S \subseteq Q$, and the edges correspond to broadcasts from S. Consider the edges from $\{q_1\}$. The self-loop corresponds to a broadcast of a that is not received. The edge to $\{q_1, q_2\}$ corresponds to a broadcast of a received by at least one process in q_1. There is no edge from $\{q_3\}$ because there is no broadcast transition from q_3.

We then have the following lemma, which asserts that runs between two configurations in an RBN induce corresponding runs in the symbolic graph. The proof of the lemma is easily obtained from the definition of the symbolic graph.

Lemma 1. *Let C, C' be two configurations of $\mathcal{R}$ such that $C \xrightarrow{a} C'$. Then, for every θ such that $C \in [\![\theta]\!]$, there exists θ' such that $C' \in [\![\theta']\!]$ and $\theta \rightsquigarrow^a \theta'$.*

3.2 Properties of the symbolic graph

In this subsection, we prove some properties of the symbolic graph (of any index k). The first two properties that we prove exhibit some structural properties on the paths of the symbolic graph. The next two properties relate paths over the symbolic graph to runs over the configurations of the given RBN. These four properties will ultimately lead us to prove our two main contributions in the next section.

First property: Monotonicity. Let $k \in \mathbb{N}$ and let $\mathcal{G}_k$ be the symbolic graph of index k associated with $\mathcal{R}$. The first key property of $\mathcal{G}_k$ is the following property, which we call *monotonicity.*

Proposition 2. *Let $\theta = (v, S)$ and $\theta' = (v', S')$ be symbolic configurations of $\mathcal{G}_k$. Then the following are true:*

- *If $Z \subseteq S$ and $\theta \rightsquigarrow^a \theta'$, then $(v, S) \rightsquigarrow^a (v', Z \cup S')$.*
- *If $Z \subseteq Q$ and $\theta \rightsquigarrow^a \theta'$, then $(v, Z \cup S) \rightsquigarrow^a (v', Z \cup S')$.*

Proof. The two points follow immediately from the definition of $\rightsquigarrow^a$. □

Second property: Normal Form. To state the second property, we first need a small definition.

Definition 2. *Let $(v_0, S_0) \rightsquigarrow \cdots \rightsquigarrow (v_m, S_m)$ a path in $\mathcal{G}_k$. A pair of indices $0 \leq i < j \leq m$ is called a* bad pair *if $(S_i \setminus S_{i+1}) \cap S_j \neq \emptyset$. A path is said to be in* normal form *if it contains no bad pairs, i.e., for all $0 \leq i < m$ and any $j > i$, $(S_i \setminus S_{i+1}) \cap S_j = \emptyset$.*

Intuitively, a path is in normal form if during each step, the states that disappear from the abstract part never reappear again. The following lemma asserts that whenever there is a path between two symbolic configurations, then there is a path between them that is in normal form.

Lemma 2. *Let θ, θ' be symbolic configurations of $\mathcal{G}_k$ such that there is a path between θ and θ' of length m. Then, there is a path in normal form between θ and θ' of length m.*

Proof Sketch. Let $\theta = \theta_0 \rightsquigarrow \theta_1 \rightsquigarrow \theta_2 \rightsquigarrow \dots \theta_{m-1} \rightsquigarrow \theta_m = \theta'$ be the path between θ and θ'. We proceed by induction on m. The claim is clearly true for $m = 0$. Suppose $m > 0$ and the claim is true for $m-1$. By induction hypothesis, we can assume that the path $\theta_0 \rightsquigarrow \theta_1 \rightsquigarrow \dots \rightsquigarrow \theta_{m-1}$ is already in normal form.

Let each $\theta_i = (v_i, S_i)$. Let l be the number of bad pairs in the path between θ_0 and θ_m. If $l = 0$, then the path is already in normal form and we are done. Suppose $l > 0$ and let (w, w') be a bad pair. Since the path between θ_0 and θ_{m-1} is already in normal form, it has to be the case that $w' = m$. Hence, we have $Z := (S_w \setminus S_{w+1}) \cap S_m \neq \emptyset$.

By Proposition 2, the following is a valid path: $(v_w, S_w) \rightsquigarrow (v_{w+1}, S_{w+1} \cup Z) \rightsquigarrow (v_{w+2}, S_{w+2} \cup Z) \dots (v_{m-1}, S_{m-1} \cup Z) \rightsquigarrow (v_m, S_m \cup Z) = (v_m, S_m)$. Let $\theta'_j := \theta_j$ if $j \leq w$ and $(v_j, S_j \cup Z)$ otherwise. Hence, we get a path $\theta'_0 \rightsquigarrow \theta'_1 \rightsquigarrow \dots \theta'_{m-1} \rightsquigarrow \theta'_m$.

Let each $\theta'_e = (v'_e, S'_e)$ and let $0 \leq i < j \leq m-1$. By a case analysis on where i and j are relative to the index w, we can prove that $(S'_i \setminus S'_{i+1}) \cap S'_j = \emptyset$. Having proved this, it is then clear by construction, that this new path from $\theta'_0 := \theta_0$ to $\theta'_m := \theta_m$ has at most $l-1$ bad pairs only. Hence, we now have a path from θ_0 to θ_m such that the prefix of length $m-1$ is in normal form and the number of bad pairs has been strictly reduced to $l-1$. Repeatedly applying this procedure leads to a path in normal form between θ_0 and θ_m. □

Third property: Refinement. Before we state the third property, we need a small definition. Recall that, given a symbolic configuration $\theta = (v, S)$, the set $[\![\theta]\!]$ denotes the set of configurations C such that $C(q) = v(q)$ if $q \notin S$ and $C(q) \geq v(q)$ otherwise. The following definition refines the set $[\![\theta]\!]$.

Definition 3. *Given a symbolic configuration $\theta = (v, S)$ and a number $N \in \mathbb{N}$, let $[\![\theta]\!]_N$ denote the set of configurations C such that $C(q) = v(q)$ if $q \notin S$ and $C(q) \geq v(q) + N$ otherwise. Note that $[\![\theta]\!] = [\![\theta]\!]_0$.*

This definition along with the above two properties now enable us to prove the third property. It roughly states that if a symbolic configuration θ' can be reached from another symbolic configuration θ, then there is a "small" N such that any configuration in $[\![\theta']\!]_N$ can be reached from some configuration in $[\![\theta]\!]$.

Theorem 1. *Let θ, θ' be symbolic configurations of $\mathcal{G}_k$ such that $\theta \rightsquigarrow^* \theta'$. Then there exists $N \leq k \times (2k)^{|Q|} \times (|Q|+1)^{|Q|+1} + 1$ such that for all $C' \in [\![\theta']\!]_N$, there exists $C \in [\![\theta]\!]$ such that $C \xrightarrow{*} C'$.*

Proof Sketch. Suppose $\theta \rightsquigarrow^* \theta'$. If the length of the path is 0, then there is nothing to prove. Hence, we restrict ourselves to the case when the length of the path is bigger than 0. By Lemma 2, there is a path in normal from from θ to θ' (say) $\theta = \theta_0 \rightsquigarrow \theta_1 \rightsquigarrow \theta_2 \dots \theta_{m-1} \rightsquigarrow \theta_m = \theta'$ with each $\theta_i := (v_i, S_i)$.

Let $N_0 = 0$ and let $N_i = (N_{i-1} + 1) \cdot (|S_{i-1} \setminus S_i| + 1)$ for every $1 \leq i \leq m$. In Lemma 5.3 of [8] (more precisely in its proof, in Lemma 6 of the long version [9]), the following fact has been proved:

> For every $1 \leq i \leq m$ and for every $C' \in [\![\theta_i]\!]_{N_i+1}$, there exists $C \in [\![\theta_{i-1}]\!]_{N_{i-1}+1}$ such that $C \xrightarrow{*} C'$.

This immediately proves that for all $C' \in [\![\theta']\!]_{N_m+1}$, there exists $C \in [\![\theta]\!]$ such that $C \xrightarrow{*} C'$. If we prove $N_m \leq k \times (2k)^{|Q|} \times (|Q|+1)^{|Q|+1}$, then the proof of the theorem will be complete.

Notice that if $(v, \emptyset) \rightsquigarrow (v', S')$ is an edge in $\mathcal{G}_k$ then $S' = \emptyset$. This fact, along with the definition of a path in normal form, allows us to easily conclude that the number of indices i such that $|S_{i-1} \setminus S_i| > 0$ is at most $|Q|$. It then follows that except for at most $|Q|$ indices, each index N_i is obtained from N_{i-1} by simply adding 1 and in the remaining indices, N_i is obtained from N_{i-1} by adding 1 and then multiplying by a number which is at most $|Q|+1$. Using this, we can deduce that the maximum value for N_m is at most $(m - |Q| + 1)|Q|(|Q|+1)^{|Q|}$. Since m is itself the length of the path between θ_0 and θ_m, m is upper bounded by the number of symbolic configurations in $\mathcal{G}_k$ which is at most $k \times k^{|Q|} \times 2^{|Q|}$. Overall we get that $N_m \leq k \times (2k)^{|Q|} \times (|Q|+1)^{|Q|+1}$. $\square$

Remark 3. A similar result was proved in Lemma 5.3 of [8], but there it was just stated that there exists an N satisfying this property. Moreover from the proof of that lemma, only a doubly exponential bound on N could be inferred.

Fourth property: Compatibility. To describe the fourth property, we need the following notion of order on configurations, *relative* to a given symbolic configuration.

Definition 4. *Let $\theta = (v, S)$ be a symbolic configuration, and let C, C' be two configurations of $\mathcal{R}$. We define an order $\preceq_\theta$ such that $C \preceq_\theta C'$ if and only if $C, C' \in [\![\theta]\!]$, and $\forall q \in S$, $C(q) \leq C'(q)$.*

This definition enables us to state our next property, which we dub *compatibility*. It intuitively says that the order that we have defined is, in some sense, compatible with the edges of the symbolic configurations.

Lemma 3. *Let θ be a symbolic configuration of $\mathcal{G}_k$, and let C, C' be two configurations of $\mathcal{R}$. If $C \in [\![\theta]\!]$ and $C \xrightarrow{*} C'$, then there exists a symbolic configuration θ' such that 1) $C' \in [\![\theta']\!]$, 2) $\theta \rightsquigarrow^* \theta'$ and 3) for all C'_1 such that $C'_1 \succeq_{\theta'} C'$, there exists $C_1 \in [\![\theta]\!]$ such that $C_1 \xrightarrow{*} C'_1$.*

Proof. Let θ be a symbolic configuration and C, C' be configurations such that $C \in [\![\theta]\!]$ and $C \xrightarrow{*} C'$. Let $C = C_0 \to \cdots \to C_{m-1} \to C_m = C'$ denote the run between C and C'. We prove the property by induction on m. For $m = 0$, we have $C = C'$. The property is easily seen to hold with $\theta' = \theta$.

Suppose now that $m \geq 1$, and that the property holds for all $n \leq m$. By induction hypothesis, for the configuration C_{m-1}, there exists a symbolic configuration θ_{m-1} satisfying the property, in particular $\theta \rightsquigarrow^* \theta_{m-1}$. Since $C_{m-1} \xrightarrow{a} C_m$ for some $a \in \Sigma$, by Lemma 1, there exists a symbolic configuration θ_m such that $C_m \in [\![\theta_m]\!]$, and $\theta_{m-1} \rightsquigarrow^a \theta_m$. Using $\theta \rightsquigarrow^* \theta_{m-1}$, we obtain that $\theta \rightsquigarrow^* \theta_m$.

Let $\theta_{m-1} = (v_{m-1}, S_{m-1})$ and $\theta_m = (v_m, S_m)$. Let $C'_m \in [\![\theta_m]\!]$ be such that $C'_m \succeq_{\theta_m} C_m$. We will construct a configuration $C'_{m-1} \in [\![\theta_{m-1}]\!]$ such that $C'_{m-1} \succeq_{\theta_{m-1}} C_{m-1}$ and $C'_{m-1} \xrightarrow{*} C'_m$. If we construct such a configuration, then by induction hypothesis, there is a $C_1 \in [\![\theta]\!]$ such that $C_1 \xrightarrow{*} C'_{m-1} \xrightarrow{*} C'_m$, which will conclude the proof.

Let $C'_{m-1}(q) = C_{m-1}(q)$ for all $q \notin S_{m-1}$. To define C'_{m-1} on S_{m-1}, we first define a mapping *pred* from states in S_m to states of $S_{m-1} \cup \overline{S_{m-1}} = Q$ as follows. Given $q' \in S_m$:

- If $q' \in S_{m-1}$, $\mathit{pred}(q') = q'$;
- If $q' \notin S_{m-1}$, by definition of edges in the symbolic graph, there exists $q \in S_{m-1}$ such that $(q, ?a, q')$ is a transition. Then $\mathit{pred}(q') = q$ for one (arbitrary but fixed) such q.

By definition, $C'_m(q) = C_m(q)$ for all $q \notin S_m$. For all $q \in S_m$, let $n_q = C'_m(q) - C_m(q)$. Intuitively, we want to place these n_q processes in the right places of C'_{m-1} so that $C'_{m-1} \rightarrow C'_m$. For all $q \in S_{m-1}$, let $C'_{m-1}(q) = C_{m-1}(q) + \sum_{q' \in S_m, \mathit{pred}(q')=q} n_{q'}$. By definition, $C'_{m-1} \succeq_{\theta_{m-1}} C_{m-1}$. So all that remains is to prove that $C'_{m-1} \xrightarrow{*} C'_m$.

Let $C_{m-1} \xrightarrow{t+t_1,\ldots,t_n} C_m$ where $t = (p, !a, p')$ and each $t_i = (p_i, ?a, p'_i)$. If we let $S_m \setminus S_{m-1} = \{q'_1, \ldots, q'_w\}$, then by definition there is a transition $t'_i := (\mathit{pred}(q'_i), ?a, q'_i)$ for each i. Additionally, $C'_{m-1}(\mathit{pred}(q'_i)) \geq C_{m-1}(\mathit{pred}(q'_i)) + n_{q'_i}$. This allows us to do $C'_{m-1} \xrightarrow{t+t_1,\ldots,t_n, n_{q'_1}\cdot t'_1, n_{q'_2}\cdot t'_2, \ldots, n_{q'_w}\cdot t'_w} C'_m$, which concludes the proof.

□

4 The PSPACE Theorem

In this section, we prove our two main contributions. First, we show that given a cube $\mathcal{C}$, $\mathit{post}^*(\mathcal{C})$ is a counting set of bounded size. Using this, we show our main result: any boolean combination of atoms can be evaluated in PSPACE, where an atom is a counting set or the reachability set of a counting set. We call this the PSPACE *Theorem.* The intuition behind the PSPACE Theorem is that the norms of the counting sets obtained by such combinations are "small", and so we only need to examine small configurations to verify them, thus yielding a PSPACE algorithm for checking correctness. In particular, the PSPACE Theorem will show that the cube-reachability problem is in PSPACE. We fix an arbitrary RBN $\mathcal{R} = (Q, \Sigma, \delta)$ for the rest of the section.

We start by drawing links between cubes and symbolic configurations.

- Given a symbolic configuration $\theta = (v, S)$, we let $\mathcal{C}_\theta$ be the cube (L, U) where $L = v$, and $U(q) = v(q)$ if $q \notin S$ and $U(q) = \infty$ otherwise. Then $\mathcal{C}_\theta = [\![\theta]\!]$.
- Given a cube $\mathcal{C} = (L, U)$, we define $\Delta_\mathcal{C}$ to be the set of symbolic configurations $\theta = (v, S)$ with $S = \{q \mid U(q) = \infty\}$ and $L(q) \leq v(q) \leq U(q)$ if $q \notin S$ and $v(q) = L(q)$ otherwise. Then $[\![\Delta_\mathcal{C}]\!] = \mathcal{C}$.

Notice that the set $\Delta_{\mathcal{C}}$ is included in the symbolic graph of index $2\|\mathcal{C}\|$. Indeed, if $\mathcal{C} = (L, U)$ and $(v, S) \in \Delta_{\mathcal{C}}$, then $|v| \leq |L| + |U_f|$ where $U_f(q) = 0$ if $U(q) = \infty$ and $U_f(q) = U(q)$ otherwise. Since $\|\mathcal{C}\| = \max(|L|, |U_f|)$, we have the desired result. By Remark 2, we know that symbolic configurations in the graph of index $2\|\mathcal{C}\|$ can only reach symbolic configurations which are also in the graph of index $2\|\mathcal{C}\|$.

Lemma 4. *Given a cube $\mathcal{C}$, the sets $\Delta_{\mathcal{C}}$ and $post^*(\Delta_{\mathcal{C}})$ are included in the symbolic graph of index $2\|\mathcal{C}\|$.*

There are only a finite number of symbolic configurations in the graph of a given index. Therefore $post^*(\Delta_{\mathcal{C}})$ is a finite set of symbolic configurations θ. It follows that $[\![post^*(\Delta_{\mathcal{C}})]\!]$ is the finite union of the cubes $\mathcal{C}_\theta$, and thus a counting set.

Unfortunately, it is in general not the case that $post^*(\mathcal{C}) = [\![post^*(\Delta_{\mathcal{C}})]\!]$, which would close our argument. However, we will show that for each symbolic configuration θ in $post^*(\Delta_{\mathcal{C}})$, there is a counting set $\mathcal{S}_\theta \subseteq [\![\theta]\!]$ such that the finite union of these counting sets is equal to $post^*(\mathcal{C})$. This will then show our first important result, namely that the reachability set of a counting set is also a counting set with "small" norm.

Theorem 2. *Let $\mathcal{C}$ be a cube. Then $post^*(\mathcal{C})$ is a counting set and*

$$\|post^*(\mathcal{C})\| \in O((\|\mathcal{C}\| \cdot |Q|)^{|Q|+2})$$

The same holds for pre^* by using the given RBN with reversed transitions.

Proof. We start by defining a counting set $\mathcal{M}$ of configurations, which we will then prove to be equal to $post^*(\mathcal{C})$.Given a symbolic configuration θ of $post^*(\Delta_{\mathcal{C}})$, we define the set $\min(\theta, \mathcal{C})$ to be the set of configurations $C \in [\![\theta]\!]$ such that C is minimal for the order $\preceq_\theta$ over the configurations of $post^*(\mathcal{C})$, i.e.

$$\min(\theta, \mathcal{C}) = \min_{\preceq_\theta} \{C \in [\![\theta]\!] \mid C \in post^*(\mathcal{C})\}$$

We can now define $\mathcal{M}$ to be the following set

$$\mathcal{M} = \bigcup_{\theta \in post^*(\Delta_{\mathcal{C}})} \; \bigcup_{C \in \min(\theta, \mathcal{C})} \mathcal{C}^\theta_C,$$

where $\mathcal{C}^\theta_C$ is the cube $\mathcal{C}_{(C,S)}$ for S such that $\theta = (v, S)$. Since $\mathcal{M}$ is a finite union of cubes, it is a counting set.

We show that $post^*(\mathcal{C}) \subseteq \mathcal{M}$. Let $C \in post^*(\mathcal{C})$. There exists $C_0 \in \mathcal{C}$ such that $C_0 \xrightarrow{*} C$, and there exists $\theta_0 \in \Delta_{\mathcal{C}}$ such that $C_0 \in [\![\theta_0]\!]$. Applying Lemma 1, we obtain the existence of $\theta \in post^*(\theta_0) \subseteq post^*(\Delta_{\mathcal{C}})$ such that $C \in [\![\theta]\!]$. Now, there exists a configuration $C' \in \min(\theta, \mathcal{C})$ such that $C' \preceq_\theta C$. By definition of $\mathcal{C}^\theta_{C'}$, C is in $\mathcal{C}^\theta_{C'}$ and thus in $\mathcal{M}$.

Now we show that $\mathcal{M} \subseteq post^*(\mathcal{C})$. Let $C \in \mathcal{M}$. By definition, there must be a symbolic configuration $\theta \in post^*(\Delta_{\mathcal{C}})$ and a configuration $C' \in post^*(\mathcal{C})$ such that $C' \preceq_\theta C$. By the Compatibility Lemma (Lemma 3), C is in $post^*(\mathcal{C})$ as well.

All that remains is to bound the norm of $\mathcal{M}$. To do this, let $\theta = (v, S) \in post^*(\Delta_\mathcal{C})$ and let $C \in \min(\theta, \mathcal{C})$. If we bound the norm of $\mathcal{C}_C^\theta$ by the desired quantity, then the proof will be complete. Noticing that $\|\mathcal{C}_C^\theta\| = |C|$, it suffices to bound $|C|$ by the desired quantity, which is what we shall do now.

By Theorem 1 and Lemma 4, there exists an $N \leq 2\|\mathcal{C}\| \times (4\|\mathcal{C}\|)^{|Q|} \times (|Q|+1)^{|Q|+1}$ such that $[\![post^*(\Delta_\mathcal{C})]\!]_N \subseteq post^*([\![\Delta_\mathcal{C}]\!]) = post^*(\mathcal{C})$. By definition of C, there must be a smallest N' such that $C(q) \leq v(q) + N'$ for every state q. If $N' > N$, then let C_N be the configuration given by $C_N(q) = \min(C(q), v(q)+N)$. We get that $C_N \in [\![\theta]\!]_N \subseteq [\![post^*(\Delta_\mathcal{C})]\!]_N \subseteq post^*(\mathcal{C})$, and so $C_N \preceq_\theta C$ and $C_N \in post^*(\mathcal{C})$, which is a contradiction to the minimality of C. Hence $N' \leq N$ and so $|C| \leq |v| + |Q| \cdot N$. Since $\theta = (v, S)$ is in $post^*(\Delta_\mathcal{C})$, by Lemma 4, we have that $|v| \leq 2\|\mathcal{C}\|$. Substituting the upper bounds for $|v|$ and N in the inequality $|C| \leq |v| + |Q| \cdot N$ then gives the required upper bound for $|C|$, thereby finishing the proof.

This result also holds for $pre^*(\mathcal{C})$. If $\mathcal{R} = (Q, \Sigma, R)$ is the given RBN, consider the "reverse" RBN $\mathcal{R}_r$, defined as $\mathcal{R} = (Q, \Sigma, R_r)$ where R_r has a transition $(q, \star a, q')$ for $\star \in \{!, ?\}$ iff R_r has a transition $(q', \star a, q)$. Notice that $\mathcal{R}_r$ is still an RBN and that $post^*(\mathcal{C})$ in $\mathcal{R}$ is equal to $pre^*(\mathcal{C})$ in $\mathcal{R}_r$. $\square$

Recall that counting sets are closed under boolean operations. With the above theorem, plus the fact that counting sets are finite unions of cubes, we obtain the following closure result.

Corollary 1 (Closure). *Counting sets are closed under post*, pre* and boolean operations.*

We are now ready to show our main result, the PSPACE Theorem. We show that there exist PSPACE algorithms to evaluate boolean combinations over counting sets and reachability set of counting sets. This result and its proof are adapted from a similar result for population protocols in [12].

Given a counting constraint Γ, we let $[\Gamma]$ denote the counting set described by Γ. To state our result, we first define some "nice" expressions.

Definition 5. *A* nice *expression is any expression that is constructed by the following syntax:*

$$E := \Gamma \mid post^*(\Gamma) \mid pre^*(\Gamma) \mid E \cap E \mid E \cup E \mid \overline{E}$$

where Γ is any counting constraint.

If E is a nice expression, then the size *of E, denoted by $|E|$, is defined as follows:*

- *If $E = \Gamma$ or $post^*(\Gamma)$ or $pre^*(\Gamma)$, then $|E| = 1$;*
- *If $E = E_1 \cup E_2$ or $E = E_1 \cap E_2$, then $|E| = |E_1| + |E_2|$;*
- *If $E = \overline{E_1}$, then $|E| = |E_1| + 1$.*

The set of configurations that is described by a nice expression E can be defined in a straightforward manner, and is denoted as $[E]$.

Notice that any nice expression E is a counting constraint, and $[E]$ is a counting set, by the Closure Corollary 1.

Theorem 3 (PSPACE Theorem). *Let E be a nice expression and let N be the maximum norm of the counting constraints appearing in E. Then $[E]$ is a counting set of norm at most exponential in $N, |E|$ and $|Q|$. Further, the membership and emptiness problems for $[E]$ are in* PSPACE*.*

Proof. Recall that $[E]$ is a counting set , by the Closure Corollary (Corollary 1). The exponential bounds for the norms follow immediately from Proposition 1 and Theorem 2. The membership complexity for union, intersection and complement is easy to see. Without loss of generality it suffices to prove that membership in $post^*(\Gamma)$ is in PSPACE, where Γ is a counting constraint.

By Savitch's Theorem NPSPACE=PSPACE, so we provide a nondeterministic algorithm. Given (C, Γ), we want to decide whether $C \in post^*(\Gamma)$. The algorithm first guesses a configuration $C_0 \in \Gamma$ of the same size as C, verifies that C_0 belongs to Γ, and then simply guesses an execution starting at C_0, step by step. The algorithm stops if either the configuration reached at some step is C, or if it has guessed more steps than the number of configurations of size $|C|$. This concludes the discussion regarding the membership complexity.

To see that checking emptiness of E is in PSPACE, notice that if E is nonempty, then it has an element of size at most $\|E\|$. We can guess such an element C in polynomial space (by representing each coefficient in binary), and verify that C is indeed in E by means of the PSPACE membership algorithm. $\square$

This result is a powerful tool which can be used to prove that a host of problems are in PSPACE for RBN. For instance, the cube-reachability problem for cubes $\mathcal{C}$ and $\mathcal{C}'$ is just checking if $post^*(\mathcal{C}) \cap \mathcal{C}'$ is empty, which by the PSPACE Theorem can be done in PSPACE. Combining this with Remark 1, we obtain the following result.

Theorem 4. *Cube-reachability is* PSPACE*-complete for RBN.*

By the reduction given in Section 4.2 of [3], this result also proves that cube-reachability is PSPACE-complete for asynchronous shared-memory systems (ASMS), which is another model of distributed computation where agents communicate by a shared register. Due to lack of space, we defer a discussion of this result to the appendix.

We will demonstrate further applications of the PSPACE Theorem in the next section.

5 Application 1: Almost-sure coverability

Having presented our PSPACE Theorem and the closure property for reachability sets of counting sets, we now provide two applications. For the first one, we consider the *almost-sure coverability* problem for RBN. Using our new results, we prove that this problem is PSPACE-complete.

The rest of the section is as follows: We first recall the definition of the almost-sure coverability problem, give a characterization of it in terms of counting sets and then prove PSPACE-completeness. Throughout this section, we fix a RBN $\mathcal{R} = (Q, \Sigma, \delta)$ with two special states $\mathit{init}, \mathit{fin} \in Q$, which will respectively be called the initial and final states.

5.1 The almost-sure coverability problem

Let $\uparrow \mathit{fin}$ denote the set of all configurations C of $\mathcal{R}$ such that $C(\mathit{fin}) \geq 1$. For any $k \geq 1$, we say that the configuration $\Lbag k \cdot \mathit{init} \Rbag$ *almost-surely covers fin* if and only if $\mathit{post}^*(\Lbag k \cdot \mathit{init} \Rbag) \subseteq \mathit{pre}^*(\uparrow \mathit{fin})$. The reason behind calling this the almost-sure coverability relation is that the definition given here is equivalent to covering the state fin from $\Lbag k \cdot \mathit{init} \Rbag$ with probability 1 under a probabilistic scheduler which picks agents uniformly at random at each step.

The number k is called a *cut-off* if one of the following is true: Either, 1) for all $h \geq k$, the configuration $\Lbag h \cdot \mathit{init} \Rbag$ almost-surely covers fin, in which case k is called a *positive* cut-off; or, 2) for all $h \geq k$, the configuration $\Lbag h \cdot \mathit{init} \Rbag$ does not almost-surely cover fin, in which case k is called a *negative* cut-off. The following was proved in Theorem 9 of [3].

Theorem 5. *Given an RBN with two states init, fin, a cut-off always exists. Whether the cut-off is positive or negative can be decided in* EXPSPACE.

Our main result of this section is that

Theorem 6. *Deciding whether the cut-off of a given RBN is positive or negative is* PSPACE*-complete. Moreover, a given RBN always has a cut-off which is at most exponential in its number of states.*

5.2 A characterization of almost-sure coverability

We now rewrite the definition of almost-sure coverability in terms of counting sets. Let $[\mathit{init}]$ be the cube such that $L(q) = U(q) = 0$ if $q \neq \mathit{init}$ and $L(\mathit{init}) = 0, U(\mathit{init}) = \infty$. Notice that by definition, $\uparrow \mathit{fin}$ is a cube. We now consider the set of configurations defined by $\mathcal{S} := \mathit{post}^*([\mathit{init}]) \cap \overline{\mathit{pre}^*(\uparrow \mathit{fin})}$. By our PSPACE Theorem 3, $\mathcal{S}$ is a counting set such that the norm of $\mathcal{S}$ is at most $2^{p(|Q|)}$ for some fixed polynomial p. We now claim the following.

Theorem 7. *$\mathcal{R}$ has a positive cut-off if and only if $\mathcal{S}$ is finite. Moreover, $|Q| \cdot |\mathcal{S}|$ is an upper bound on the size of the cut-off for $\mathcal{R}$ and so $\mathcal{R}$ has a cut-off which is exponential in its number of states.*

Proof. Let N be the norm of $\mathcal{S}$. Suppose $\mathcal{S}$ is finite. If $C \in \mathcal{S}$, then $\sum_{q \in Q} C(q) \leq |Q| \cdot N$. So, if C is any configuration of size $h > |Q| \cdot N$ such that $C \in \mathit{post}^*(\Lbag h \cdot \mathit{init} \Rbag)$ then $C \in \mathit{pre}^*(\uparrow \mathit{fin})$. Hence, $|Q| \cdot N$ is a positive cut-off for $\mathcal{R}$.

Suppose $\mathcal{S}$ is infinite, and let $\cup_i \mathcal{C}_i$ be a counting constraint for $\mathcal{S}$ whose norm is N. Then there must exist an index i with $\mathcal{C}_i := (L, U)$ and a state p such that $U(p) = \infty$. For each $h \geq N$, consider the configuration C_h given by $C_h(q) = L(q)$ if $q \neq p$ and $C_h(p) = h$. Notice that $C_h \in \mathcal{S}$ and so $C_h \in post^*([init]) \cap \overline{pre^*(\uparrow fin)}$. Hence, for every $h \geq |Q| \cdot N$, we have exhibited a configuration of size h, reachable from $(\!|h \cdot init|\!)$ but from which fin is not coverable. Thus N is a negative cut-off for $\mathcal{R}$. $\square$

Remark 4. Notice that we have shown that if $\mathcal{S}$ is finite, then $\mathcal{R}$ has a positive cut-off and if $\mathcal{S}$ is infinite, then $\mathcal{R}$ has a negative cut-off. This gives an alternative proof of the fact that a cut-off always exists for a given RBN.

5.3 PSPACE-completeness of the almost-sure coverability problem

Because of Theorem 7, we now have the following result.

Lemma 5. *Deciding whether the cut-off of a given RBN is positive or negative can be done in* PSPACE.

Proof Sketch. By Theorem 7, it follows that a given RBN has a negative cut-off iff $\mathcal{S} = post^*([init]) \cap \overline{pre^*(\uparrow fin)}$ is infinite. We have already seen that $\mathcal{S}$ is a counting set such that the norm of $\mathcal{S}$ is at most $N := 2^{p(|Q|)}$ for some fixed polynomial p.

Let $\cup_i \mathcal{C}_i$ be a counting constraint for $\mathcal{S}$ which minimizes its norm and let each $\mathcal{C}_i = (L_i, U_i)$. Hence, $L_i(q) \leq N$ for every state q. Further, $\mathcal{S}$ is infinite iff there is an index i and a state q such that $U_i(q) = \infty$. Using these two facts, we can then show that $\mathcal{S}$ is infinite iff there is a state q and a configuration $C \in \mathcal{S}$ such that $C(q') \leq N$ for every $q' \neq q$ and $C(q) = N + 1$.

Hence, to check if $\mathcal{S}$ is infinite, we just have to guess a state q and a configuration C such that $C(q') \leq N$ for every $q' \neq q$ and $C(q) = N + 1$ and check if $C \in \mathcal{S}$. Since guessing C can be done in polynomial space (by representing every number in binary), by the PSPACE Theorem (Theorem 3), we can check if $C \in \mathcal{S}$ in polynomial space as well, which concludes the proof of the theorem. $\square$

We also have the accompanying hardness result.

Lemma 6. *Deciding whether the cut-off of a given RBN is positive or negative is* PSPACE*-hard.*

Similar to the cube-reachability problem, our result on almost-sure coverability also applies to the related model of ASMS. This solves an open problem from [6]. For lack of space, we once again defer this discussion to the appendix.

6 Application 2: Computation by RBN

In this section we give another application of our results. We introduce a model of computation using RBN called *RBN protocols*. We take inspiration from the extensively-studied model of population protocols [1,2,12]. The reader can consult the above references for more details on population protocols.

In our model, reconfigurable networks of identical, anonymous agents interact to compute a predicate $\varphi : \mathbb{N}^k \to \{0, 1\}$. We show that RBN protocols compute exactly the threshold predicates, which we will define more formally below.

6.1 RBN Protocols

We introduce our computation model. The notation mimics that of [13].

Definition 6. *An* RBN protocol *is a tuple* $\mathcal{P} = (Q, \Sigma, \delta, I, O)$ *where* (Q, Σ, δ) *is an RBN,* $I = \{q_1, \ldots, q_k\}$ *is a set of* input states*, and* $O : Q \to \{0, 1\}$ *is an* output function*.*

Configurations and *runs* of $\mathcal{P}$ are the same as that of the underlying RBN. A configuration C is called a *0-consensus* (respectively a *1-consensus*) if $C(q) > 0$ implies $O(q) = 0$ (respectively $O(q) = 1$). For $b \in \{0, 1\}$, a b-consensus C is *stable* if every configuration reachable from C is also a b-consensus. A run $C_0 \to C_1 \to C_2 \cdots$ of $\mathcal{P}$ is *fair* if it is finite and cannot be extended by any step, or if it is infinite and the following condition holds for all configurations C, C': if $C \to C'$ and $C = C_i$ for infinitely many $i \geq 0$, then the step $C \to C'$ appears infinitely along the run. In other words, if a fair run reaches a configuration infinitely often, then all the configurations reachable in a step from that configuration will be reached infinitely often from it.

A fair run $C_0 \to C_1 \to \ldots$ *converges to* b if there is $i \geq 0$ such that C_j is a b-consensus for every $j \geq i$. For every $\boldsymbol{v} \in \mathbb{N}^k$, let $C_{\boldsymbol{v}}$ be the configuration given by $C_{\boldsymbol{v}}(q_i) = \boldsymbol{v}_i$ for every $q_i \in I$, and $C_{\boldsymbol{v}}(q) = 0$ for every $q \in Q \setminus I$. We call $C_{\boldsymbol{v}}$ the *initial configuration for input* $\boldsymbol{v}$. The protocol $\mathcal{P}$ *computes the predicate* $\varphi \colon \mathbb{N}^k \to \{0, 1\}$, if for every $\boldsymbol{v} \in \mathbb{N}^k$, every fair run starting at $C_{\boldsymbol{v}}$ converges to $\varphi(\boldsymbol{v})$.

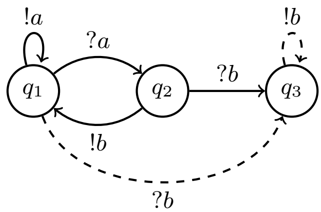

Fig. 3. An RBN protocol $\mathcal{P}$.

Example 3. Adding the dashed line transitions to the RBN of Example 1 yields the RBN protocol $\mathcal{P} = (Q, \Sigma, \delta, I, O)$ illustrated in Figure 3. The initial state is

q_1, i.e. $I = \{q_1\}$, and the output function is defined such that $O(q_1) = O(q_2) = 0$ and $O(q_3) = 1$. If there is a process in q_3, it can "attract" the rest of the processes there using the new dashed transitions. As with the RBN of Example 1, a process can be put in q_3 starting from the initial configuration $\wr k \cdot q_1 \int$ if and only if $k \geq 3$. This RBN protocol computes the predicate $x \geq 3$: if there are less than 3 processes originally in q_1 then they stay in states with output 0, and if there are more, then in a fair run a process eventually enters q_3, and eventually the others follow, thus converging to 1.

6.2 Expressivity

In this section, we show that RBN protocols compute exactly the predicates definable by counting sets. A predicate $\varphi : \mathbb{N}^k \to \{0, 1\}$ is *definable by counting sets* if for every $b \in \{0, 1\}$, the sets $\{\boldsymbol{v} \mid \varphi(\boldsymbol{v}) = b\}$ are counting sets.

For $b \in \{0, 1\}$, define the following sets of configurations:

- Let $\mathcal{C}_b$ be the set of b-consensus configurations.
- Let $\mathcal{ST}_b$ be the set $\overline{pre^* \left(\overline{\mathcal{C}_b}\right)}$ of stable b-consensuses. These are the configurations from which one can reach only b-consensuses.
- Let $\mathcal{I}_b$ be the set of initial configurations $C_{\boldsymbol{v}}$ for inputs $\boldsymbol{v}$ such that $\varphi(\boldsymbol{v}) = b$.

The next lemma states that every predicate computed by a protocol is definable by counting sets.

Lemma 7. *Let $\mathcal{P}$ be a RBN protocol that computes the predicate $\varphi : \mathbb{N}^k \to \{0, 1\}$. Then for every $b \in \{0, 1\}$, the sets $\mathcal{I}_b, \mathcal{C}_b$ and $\mathcal{ST}_b$ are all counting sets. This entails that φ is definable by counting sets.*

Proof Sketch. Fix a $b \in \{0, 1\}$. It is easy to see that $\mathcal{C}_b$ is a cube. Unraveling the definitions of $\mathcal{I}_b$ and $\mathcal{ST}_b$, we can express them in terms of $\mathcal{C}_b$ by using boolean operations and *pre**. By the Closure Corollary (Corollary 1), they are counting sets. Set $\{\boldsymbol{v} \mid \varphi(\boldsymbol{v}) = b\}$ is simply $\mathcal{I}_b$ restricted to I, and so we are done. □

The next lemma states the converse result. It essentially uses the fact that there is a sub-class of population protocols called IO protocols which compute exactly the predicates definable by counting sets (Theorem 7 and Theorem 39 of [2,13]), and that IO protocols are a sub-class of RBN (Section 6.2 of [3]).

Lemma 8. *Let $\varphi : \mathbb{N}^k \to \{0, 1\}$ be a predicate definable by counting sets. Then there exists a RBN protocol computing φ.*

By Lemma 7 and Lemma 8, we get our result.

Theorem 8. *RBN protocols compute exactly the predicates definable by counting sets.*

Acknowledgements

We thank Nathalie Bertrand and Javier Esparza for many helpful discussions.

References

1. Angluin, D., Aspnes, J., Diamadi, Z., Fischer, M.J., Peralta, R.: Computation in networks of passively mobile finite-state sensors. Distributed Comput. **18**(4), 235–253 (2006). https://doi.org/10.1007/s00446-005-0138-3
2. Angluin, D., Aspnes, J., Eisenstat, D., Ruppert, E.: The computational power of population protocols. Distributed Comput. **20**(4), 279–304 (2007). https://doi.org/10.1007/s00446-007-0040-2
3. Balasubramanian, A.R., Weil-Kennedy, C.: Reconfigurable broadcast networks and asynchronous shared-memory systems are equivalent. In: Ganty, P., Bresolin, D. (eds.) Proceedings 12th International Symposium on Games, Automata, Logics, and Formal Verification, GandALF 2021, Padua, Italy, 20-22 September 2021. EPTCS, vol. 346, pp. 18–34 (2021). https://doi.org/10.4204/EPTCS.346.2
4. Bertrand, N., Fournier, P.: Parameterized verification of many identical probabilistic timed processes. In: IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS. pp. 501–513 (2013). https://doi.org/10.4230/LIPIcs.FSTTCS.2013.501
5. Bertrand, N., Fournier, P., Sangnier, A.: Playing with probabilities in reconfigurable broadcast networks. In: Foundations of Software Science and Computation Structures - 17th International Conference, FOSSACS. pp. 134–148 (2014). https://doi.org/10.1007/978-3-642-54830-7_9
6. Bouyer, P., Markey, N., Randour, M., Sangnier, A., Stan, D.: Reachability in networks of register protocols under stochastic schedulers. In: Chatzigiannakis, I., Mitzenmacher, M., Rabani, Y., Sangiorgi, D. (eds.) 43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy. LIPIcs, vol. 55, pp. 106:1–106:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2016). https://doi.org/10.4230/LIPIcs.ICALP.2016.106
7. Chini, P., Meyer, R., Saivasan, P.: Liveness in broadcast networks. In: Atig, M.F., Schwarzmann, A.A. (eds.) Networked Systems - 7th International Conference, NETYS 2019, Marrakech, Morocco, June 19-21, 2019, Revised Selected Papers. Lecture Notes in Computer Science, vol. 11704, pp. 52–66. Springer (2019). https://doi.org/10.1007/978-3-030-31277-0_4
8. Delzanno, G., Sangnier, A., Traverso, R., Zavattaro, G.: On the complexity of parameterized reachability in reconfigurable broadcast networks. In: D'Souza, D., Kavitha, T., Radhakrishnan, J. (eds.) IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India. LIPIcs, vol. 18, pp. 289–300. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2012), https://doi.org/10.4230/LIPIcs.FSTTCS.2012.289
9. Delzanno, G., Sangnier, A., Traverso, R., Zavattaro, G.: On the complexity of parameterized reachability in reconfigurable broadcast networks. *Long version* (2012), https://www.irif.fr/~sangnier/publis/DSTZ-FSTTCS12-long.pdf
10. Delzanno, G., Sangnier, A., Zavattaro, G.: Parameterized verification of ad hoc networks. In: Gastin, P., Laroussinie, F. (eds.) CONCUR 2010 - Concurrency Theory, 21th International Conference, CONCUR 2010, Paris, France, August 31-September 3, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6269, pp. 313–327. Springer (2010). https://doi.org/10.1007/978-3-642-15375-4_22
11. Esparza, J., Ganty, P., Majumdar, R., Weil-Kennedy, C.: Verification of immediate observation population protocols. In: CONCUR. LIPIcs, vol. 118, pp. 31:1–31:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2018). https://doi.org/10.4230/LIPIcs.CONCUR.2018.31

12. Esparza, J., Jaax, S., Raskin, M.A., Weil-Kennedy, C.: The complexity of verifying population protocols. Distributed Comput. **34**(2), 133–177 (2021). https://doi.org/10.1007/s00446-021-00390-x
13. Esparza, J., Raskin, M.A., Weil-Kennedy, C.: Parameterized analysis of immediate observation petri nets. In: Donatelli, S., Haar, S. (eds.) Application and Theory of Petri Nets and Concurrency - 40th International Conference, PETRI NETS 2019, Aachen, Germany, June 23-28, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11522, pp. 365–385. Springer (2019). https://doi.org/10.1007/978-3-030-21571-2_20