Reconfigurable Broadcast Networks and Asynchronous Shared-Memory Systems are Equivalent

A. R. Balasubramanian   Chana Weil-Kennedy

Technical University of Munich
Munich, Germany *

bala.ayikudi@tum.de   chana.weilkennedy@in.tum.de

We show the equivalence of two distributed computing models, namely reconfigurable broadcast networks (RBN) and asynchronous shared-memory systems (ASMS), that were introduced independently. Both RBN and ASMS are systems in which a collection of anonymous, finite-state processes run the same protocol. In RBN, the processes communicate by selective broadcast: a process can broadcast a message which is received by all of its neighbors, and the set of neighbors of a process can change arbitrarily over time. In ASMS, the processes communicate by shared memory: a process can either write to or read from a shared register. Our main result is that RBN and ASMS can *simulate* each other, i.e. they are equivalent with respect to parameterized reachability, where we are given two (possibly infinite) sets of configurations $\mathscr{C}$ and $\mathscr{C}'$ defined by upper and lower bounds on the number of processes in each state and we would like to decide if some configuration in $\mathscr{C}$ can reach some configuration in $\mathscr{C}'$. Using this simulation equivalence, we transfer results of RBN to ASMS and vice versa. Finally, we show that RBN and ASMS can simulate a third distributed model called immediate observation (IO) nets. Moreover, for a slightly stronger notion of simulation (which is satisfied by all the simulations given in this paper), we show that IO nets cannot simulate RBN.

1 Introduction

In this paper, we consider three models of distributed computation, one in which communication happens by (selective) broadcasts, another in which communication happens by means of a shared memory, and finally one in which communication happens by observation. We first expand a bit more on these models, then describe our main results and finally derive some consequences from these results.

The first model that we consider is *reconfigurable broadcast networks* (RBN)[12, 11]. In this model, we have a collection of anonymous, finite-state processes executing the same protocol. Further, every process has a set of neighbors. At each step, a process can broadcast a message which is then received by all of the processes in its neighborhood. The neighborhood topology is reconfigurable, meaning that the set of neighbors of a process can change arbitrarily between two steps. Parameterized verification of RBN aims to prove that a property is correct, irrespective of the number of participating processes. Dually, it attempts to find an execution of some population of processes for which a property is violated. Within this context, the complexity of different variants of (parameterized) reachability and repeated coverability have been studied for RBN [12, 11, 8]. Moreover, many extensions of RBN with clocks, registers and probabilities have been proposed and studied, mainly within the perspective of parameterized verification [10, 5, 4].

The second model that we consider is a formal model of *asynchronous shared-memory systems* (ASMS)[13, 7, 6]. In this model, we have a collection of anonymous, finite-state processes executing the

same protocol, and a single register which all processes can access to perform a read/write operation. The set of values that can be stored in this register is finite. No locks onto the register are allowed and so no process can perform a sequence of atomic operations whilst preventing other processes from accessing the register. Similar to RBN, major questions of interest in ASMS are those pertaining to parameterized verification, i.e. finding bad executions over some population of processes. The complexity of some (parameterized) reachability and model-checking questions for ASMS have been explored in a series of papers [13, 14, 7]. Further extensions of this model with leaders, stacks, etc. have also been studied [14, 9, 20, 18]. Finally, [6] considers ASMS in the setting in which a stochastic scheduler picks a process (uniformly at random) at each step to be executed, and under this setting studies the question of whether a given state can be reached by some process almost-surely, i.e., with probability 1.

The third model that we consider is *immediate observation Petri nets* (IO nets) [15, 17], which were introduced with motivations from the field of population protocols [2, 1]. Intuitively, in this model, we have a collection of anonymous, finite-state processes executing the same protocol. The only communication allowed between processes is *observation*, i.e., a process can only observe that another process is at some other state, and based on this observation can execute a step. The process being observed cannot detect if some process is observing it. Motivated by application to population protocols, the authors of [15, 17] study parameterized reachability questions for IO nets.

In this paper, we show that RBN and ASMS can *simulate* each other, with respect to (parameterized) reachability. Roughly speaking, we show that any instance of a parameterized reachability question for RBN can be efficiently translated to an instance of parameterized reachability for ASMS and vice versa. More specifically, we consider the question of *cube-reachability*. In the cube-reachability question, we are given an instance of a model (which can be either an RBN, an ASMS or an IO net) and two sets of configurations $\mathscr{C}, \mathscr{C}'$, each of them defined by lower and upper bounds on the number of processes in each state. (The upper bounds on some states might be ∞, which means that we allow arbitrary number of processes in that state). We would then like to decide if there is a configuration in $\mathscr{C}$ which can reach a configuration in $\mathscr{C}'$. As we shall explain in the next section, the cube-reachability question covers parameterized reachability and coverability problems, parameterized reachability problems with leaders, and allows for a uniform transfer of results between the models that we study in this paper.

Our main result is that the cube-reachability questions for RBN and ASMS are polynomial-time equivalent to each other. This result, along with the constructions achieving this result, enable us to translate various parameterized reachability results from RBN to ASMS and vice versa. First, we show that a special case of cube-reachability, called unbounded initial cube reachability, is PSPACE-complete for ASMS, by using our reduction and already existing similar results on RBN. Then, we introduce the model of RBN-leader protocols and use already existing results on ASMS-leader protocols to prove that the RBN-leader reachability problem is NP-complete. Finally, we show that the almost-sure coverability problem for RBN is decidable in EXPSPACE by translating the analogous result for ASMS [6].

Additionally, we show that the cube-reachability problem for IO nets reduces to the cube-reachability problem for RBN, leading to a transfer of results from RBN to IO nets. For the other direction, we actually show an impossibility result. We define a stronger form of reduction for the cube-reachability problem and we notice that the reductions given in this paper all satisfy this stronger property. Then, using results from IO net theory, we show that there can be no reduction from the cube-reachability problem for RBN to the cube-reachability problem for IO nets which satisfies this stronger property. We leave open the problem of whether there can exist other reductions from RBN to IO nets.

The rest of the paper is organized as follows: In Section 2, we present some preliminary definitions and notations, then in Section 3, we describe RBN and ASMS. Section 4 proves our main result that RBN and ASMS can simulate each other. Section 5 presents some transfer of results between RBN and

ASMS. In Section 6, we introduce IO nets, show that they can be simulated by RBN, and prove that the other direction is not true for a stronger form of simulation. For space reasons, all missing proofs are relegated to the full version of this paper [3].

2   Preliminaries

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 $\langle e_1,\ldots,e_n \rangle$ denote the multiset C such that $C(e) = |\{ j \mid e_j = e \}|$. We sometimes write multisets using set-like notation. For example, $\langle 2 \cdot a, b \rangle$ and $\langle a,a,b \rangle$ denote the same multiset. Given $e \in E$, we denote by $\boldsymbol{e}$ the multiset consisting of one occurrence of element e, that is $\langle e \rangle$. 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. Given a multiset C on E and a multiset C' on E' such that $E \cap E' = \emptyset$, we denote by $C \cdot C'$ the multiset on $E \cup E'$ equal to C on E and to C' on E'. We call $|C| \stackrel{\text{def}}{=} \sum_{e \in E} C(e)$ the *size* of C, and $\|C\| \stackrel{\text{def}}{=} \{e \mid C(e) > 0\}$ the *support* of C. Given $E' \subseteq E$ define $C(E') \stackrel{\text{def}}{=} \sum_{e \in E'} C(e)$.

Cubes.   Given a finite set Q, a *cube* $\mathscr{C}$ 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 $\mathscr{C} = \{C : L \le C \le U\}$. Abusing notation, we identify the set $\mathscr{C}$ with the pair (L,U). All the results in this paper are true irrespective of whether the constants are encoded in unary or binary.

Reachability.   Let $\mathscr{T} = (S, \to)$ be a transition system where S is a set of *configurations* and $\to$ is a binary relation on S called the transition (or) step relation. Given configurations C and C', we say C' is *reachable* from C if $C \xrightarrow{*} C'$, where $\xrightarrow{*}$ denotes the reflexive-transitive closure of the step relation. Let $\mathscr{S}$ be a set of configurations. The *predecessor set* of $\mathscr{S}$ is $pre^*_{\mathscr{T}}(\mathscr{S}) \stackrel{\text{def}}{=} \{C' | \exists C \in \mathscr{S} . C' \xrightarrow{*} C\}$, and the *successor set* of $\mathscr{S}$ is $post^*_{\mathscr{T}}(\mathscr{S}) \stackrel{\text{def}}{=} \{C | \exists C' \in \mathscr{S} . C' \xrightarrow{*} C\}$. The *immediate predecessor set* of $\mathscr{S}$ is $pre_{\mathscr{T}}(\mathscr{S}) \stackrel{\text{def}}{=} \{C' | \exists C \in \mathscr{S} . C' \to C\}$, and the *immediate successor set* of $\mathscr{S}$ is $post_{\mathscr{T}}(\mathscr{S}) \stackrel{\text{def}}{=} \{C | \exists C' \in \mathscr{S} . C' \to C\}$. When it is clear from the context, we will drop the $\mathscr{T}$ subscript. The *reachability* problem consists of deciding, given a system $\mathscr{T}$ and configurations C, C', whether C' is reachable from C in $\mathscr{T}$.

Cube reachability.   If $\mathscr{T}$ is a transition system whose set of configurations is the set of all multisets on a finite set Q, then the reachability problem can be generalized to the *cube-reachability* problem which consists of deciding, given $\mathscr{T}$ and two cubes $\mathscr{C}, \mathscr{C}'$ over Q, whether there exists configurations $C \in \mathscr{C}$ and $C' \in \mathscr{C}'$ such that C' is reachable from C in $\mathscr{T}$. If this is the case, we say $\mathscr{C}'$ is reachable from $\mathscr{C}$.

As mentioned before, the cube-reachability problem generalizes the reachability problem. It also generalizes the coverability problem : Given a configuration C and a state $q \in Q$, decide if there exists C' such that $C \xrightarrow{*} C'$ and $C'(q) \ge 1$. It can also talk about *parameterized reachability* problems, for e.g., given two finite sets of states I and F, do there exist configurations C and C' such that $\|C\| \subseteq I, \|C'\| \subseteq F$ and $C \xrightarrow{*} C'$. Further, the cube-reachability problem is important in the model of immediate observation Petri nets (IO nets). This model was introduced to study immediate observation population protocols [15, 17], and the correctness problem for these protocols is solved using cube-reachability in IO nets. Additionally, as we will see in Section 5.2, the cube-reachability problem is a generalization of the so-called *leader*
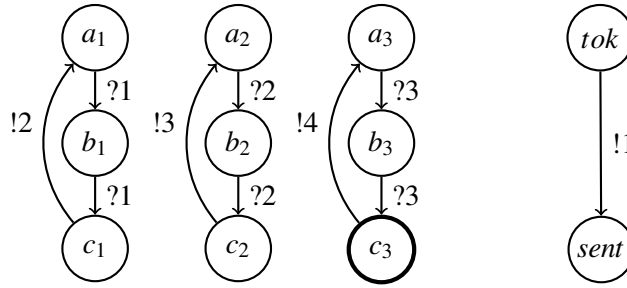
Figure 1: An RBN simulating a counter to 2^3.

reachability problem and allows for an elegant way to transfer results between the models that we study in this paper.

3 Two Models

3.1 Reconfigurable Broadcast Networks

Reconfigurable broadcast networks (RBN) [12, 11] are networks comprising an arbitrary number of finite-state, anonymous processes and a communication topology specifying the presence or absence of communication links between different processes. During a step, a process can broadcast a message which is immediately received by all of its neighbours. The process and its neighbours then update their states according to a transition relation. Before each such broadcast step, the communication topology can reconfigure in an arbitrary manner. Since our main focus in this paper is regarding reachability in this model, we can forget the communication topology and simply define the semantics of an RBN directly in terms of collections of processes.

Definition 1. *A reconfigurable broadcast network is a tuple $\mathscr{R} = (Q, \Sigma, \delta)$ 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 an RBN $\mathscr{R}$ 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 $\{t, t_1, \ldots, t_k\}$ 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 $C \xrightarrow{t+t_1, \ldots, t_n} C'$, and 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.

Example 1. *Consider the RBN of Figure 1, with set of states $\{tok, sent\} \cup \{a_i, b_i, c_i \mid 1 \leq i \leq 3\}$. It is inspired by a similar example described in Section 5.1 of [6]. Let $\mathscr{C}_0$ be the cube which puts exactly one process in each a_i, an arbitrary number of processes in tok and 0 processes elsewhere. That is, $\mathscr{C}_0 = (L, U)$ such that $L(a_i) = U(a_i) = 1$ for all i, $L(tok) = 0$ and $U(tok) = \infty$, and $L(q) = U(q) = 0$ for all other states q. Let $\mathscr{C}_f$ be the cube which puts at least one process in c_3 and an arbitrary number elsewhere. Suppose some configuration in $\mathscr{C}_0$ reaches some configuration in $\mathscr{C}_f$. By construction, for a process to reach c_3 it must start in a_3 and receive 3 twice. For a process to broadcast 3 it must start in a_2 and receive 2 twice, and for a process to broadcast 2 it must start in a_1 and receive 1 twice. So a run*
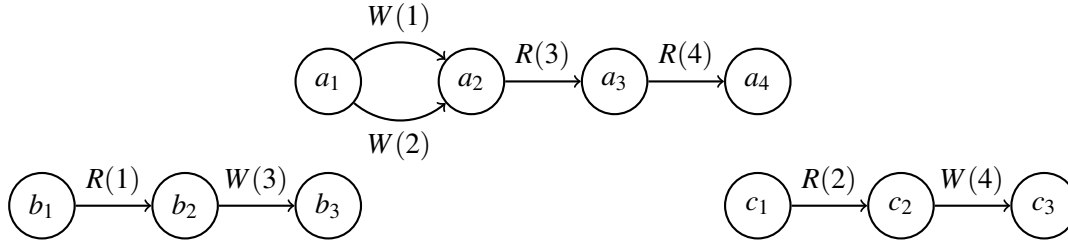
Figure 2: An example of an ASMS

from a configuration of $\mathscr{C}_0$ to a configuration of $\mathscr{C}_f$ must contain at least 2^3 broadcasts of 1. Since the only way to broadcast 1 is for a process to go from tok to sent, there must be at least 2^3 processes in tok in the initial configuration of $\mathscr{C}_0$.

3.2 Asynchronous Shared-Memory Systems

Asynchronous shared-memory systems (ASMS) [14, 13] consist of an arbitrary number of finite-state, anonymous processes. These processes can communicate with each other by means of a single shared register, to which they can either write a value or from which they can read a value.

Definition 2. *An asynchronous shared-memory system (ASMS) is a tuple $\mathscr{P} = (Q, \Sigma, \delta)$ where Q is a finite set of states, Σ is a finite alphabet, and $\delta \subseteq Q \times \{R, W\} \times \Sigma \times Q$ is the set of transitions. Here R stands for* read, *and W stands for* write.

We use $p \xrightarrow{R(d)} q$ (resp. $p \xrightarrow{W(d)} q$) to denote that $(p, R, d, q) \in \delta$ (resp. $(p, W, d, q) \in \delta$). The semantics of an ASMS is given by means of *configurations*. A configuration C of an ASMS is a multiset over $Q \cup \Sigma$ such that $\sum_{d \in \Sigma} C(d) = 1$, i.e., C contains exactly one element from the set Σ. Hence, we sometimes denote a configuration C as (M, d) where M is a multiset over Q (which counts the number of processes in each state) and $d \in \Sigma$ (which denotes the content of the shared register). The value d will be denoted by $data(C)$.

A *step* between configurations $C = (M, d)$ and $C' = (M', d')$ exists if there is $t = (p, \text{op}, d'', q) \in \delta$ such that $M(p) > 0$, $M' = M - \boldsymbol{p} + \boldsymbol{q}$ and either $\text{op} = R$ and $d = d' = d''$ or $\text{op} = W$ and $d' = d''$. If such a step exists, we denote it by $C \xrightarrow{t} C'$ and we let $\xrightarrow{*}$ denote the reflexive transitive closure of the step relation. A *run* is then a sequence of steps. Given a sequence of transitions $\sigma = t_1, \ldots, t_n$, we sometimes use $C \xrightarrow{\sigma} C'$ to denote that there is a run of the form $C \xrightarrow{t_1} C_1 \xrightarrow{t_2} \ldots C_{n-1} \xrightarrow{t_n} C'$.

A cube $\mathscr{C} = (L, U)$ of an ASMS $\mathscr{P} = (Q, \Sigma, \delta)$ is defined to be a cube over $Q \cup \Sigma$ satisfying the following property : There exists $d \in \Sigma$ such that $L(d) = U(d) = 1$ and $L(d') = U(d') = 0$ for every other d'. Hence, we sometimes denote a cube $\mathscr{C}$ as (L, U, d) where (L, U) is a cube over Q and $d \in \Sigma$. Membership of a configuration C in a cube $\mathscr{C}$ is then defined in a straightforward manner. The cube-reachability problem for ASMS is then to decide, given $\mathscr{P}$ and two cubes $\mathscr{C}, \mathscr{C}'$ whether $\mathscr{C}$ can reach $\mathscr{C}'$, i.e., whether there are configurations $C \in \mathscr{C}, C' \in \mathscr{C}'$ such that $C \xrightarrow{*} C'$.

Example 2. *Consider the ASMS of Figure 2 where the alphabet is $\{\#, 1, 2, 3, 4\}$. Let $\mathscr{C}$ be the cube which puts exactly one process in a_1, arbitrary number of processes in b_1 and c_1 and exactly 0 processes elsewhere. Let $\mathscr{C}'$ be the cube where $\mathscr{C}'$ puts at least one process in a_4 and arbitrary number of processes elsewhere. It can be verified that the cube $\mathscr{C}$ cannot reach $\mathscr{C}'$ for the following reason: Since there is only one process in a_1 in $\mathscr{C}$, it follows that this process can either write 1 or 2, but not both. Hence,*

either processes from b_1 can move into b_2 to write 3 or processes from c_1 can move into c_2 to write 4, but both cannot happen. It then follows that it is impossible to read both 3 and 4, and so the state a_4 cannot be reached.

4   RBN and ASMS are Cube-Reachability Equivalent

Throughout this paper, whenever we talk about one model simulating another model, we mean that the cube-reachability problem for the second model can be reduced in polynomial time to the cube-reachability problem for the first model. In this section, we prove our main result that RBN and ASMS can *simulate* each other. As we will see in the next section, this simulation will allow us to transfer results from RBN to ASMS and vice versa.

4.1   ASMS Simulate RBN

Construction Let $\mathscr{R} = (Q_{\mathscr{R}}, \Sigma_{\mathscr{R}}, \delta_{\mathscr{R}})$ be an RBN. We construct an ASMS that simulates $\mathscr{R}$. The register value is used to store which message can be received, additional states are used to represent that a broadcast is in progress, and a fresh register value is written when the simulation of a broadcast is over. For every $a \in \Sigma_{\mathscr{R}}$, we let $\delta_{\mathscr{R}}^{!a}$ (resp. $\delta_{\mathscr{R}}^{?a}$) be the subset of the transitions in $\delta_{\mathscr{R}}$ that broadcast (resp. receive) the letter a. Let $\mathscr{P} = (Q_{\mathscr{P}}, \Sigma_{\mathscr{P}}, \delta_{\mathscr{P}})$ be the following ASMS: The set of states $Q_{\mathscr{P}}$ is $Q_{\mathscr{R}} \cup I$) with $I = \{[p, a, p'] : (p, ?a, p') \in \delta \text{ or } (p, !a, p') \in \delta\}$, where I stands for intermediary. The alphabet $\Sigma_{\mathscr{P}}$ is $\Sigma_{\mathscr{R}} \cup \{\#\}$ where $\#$ is a letter which is not in Σ. The transition relation $\delta_{\mathscr{P}}$ is such that for every $t = (q, !a, q') \in \delta_{\mathscr{R}}$ there are transitions $\hat{t} := q \xrightarrow{W(a)} [q, a, q']$ and $t^{\#} := [q, a, q'] \xrightarrow{W(\#)} q'$ in $\delta_{\mathscr{P}}$, and for every $t = (q, ?a, q') \in \delta_{\mathscr{R}}$ there are transitions $\hat{t} := q \xrightarrow{R(a)} [q, a, q']$ and $t^{\#} := [q, a, q'] \xrightarrow{W(\#)} q'$ in $\delta_{\mathscr{P}}$, as represented in Figure 3.

Figure 3: Simulation in $\mathscr{P}$ of transitions $q \xrightarrow{!a} q'$ and $p \xrightarrow{?a} p'$ of $\mathscr{R}$.

A configuration (C, d) of $\mathscr{P}$ is called *good* if $C(I) = 0$ and $d = \#$. There is a natural bijection between configurations of $\mathscr{R}$ and good configurations of $\mathscr{P}$. If C is a configuration of $\mathscr{R}$, we will use $(\widehat{C}, \#)$ to denote the corresponding good configuration of $\mathscr{P}$.

Correctness of construction We now show that $C' \in post^*_{\mathscr{R}}(C)$ iff $(\widehat{C'}, \#) \in post^*_{\mathscr{P}}(\widehat{C}, \#)$ for any configurations C and C' of $\mathscr{R}$. Suppose $C \xrightarrow{t + t_1, \dots, t_n} C'$ is a step in $\mathscr{R}$. It is easy to see that we have a run in $\mathscr{P}$ of the form $(\widehat{C}, \#) \xrightarrow{\hat{t}, \hat{t}_1, \dots, \hat{t}_n, t^{\#}, t_1^{\#}, \dots, t_n^{\#}} (\widehat{C'}, \#)$. Hence, if $C \xrightarrow{*} C'$ for some configurations C, C' in $\mathscr{R}$, then $(\widehat{C}, \#) \xrightarrow{*} (\widehat{C'}, \#)$ in $\mathscr{P}$.

For the other direction, we first define the notion of a *pseudo-step* between two good configurations of $\mathscr{P}$. A run $(\widehat{C}, \#) \xrightarrow{\sigma} (\widehat{C'}, \#)$ of $\mathscr{P}$ is called a *pseudo-step* if there exists $a \in \Sigma$ and transitions $t \in \delta_{\mathscr{R}}^{!a}$ and $t_1, \dots, t_n \in \delta_{\mathscr{R}}^{?a}$ such that $\sigma = \hat{t}, \hat{t}_1, \dots, \hat{t}_n, t^{\#}, t_1^{\#}, \dots, t_n^{\#}$. The intuition behind this notion is that if $(\widehat{C}, \#) \xrightarrow{\sigma} (\widehat{C'}, \#)$ where σ is a pseudo-step with $\sigma = \hat{t}, \hat{t}_1, \dots, \hat{t}_n, t^{\#}, t_1^{\#}, \dots, t_n^{\#}$ then $C \xrightarrow{t + t_1 \dots t_n} C'$ is a step in $\mathscr{R}$. Hence, pseudo-steps of $\mathscr{P}$ "behave" similarly to a single step in $\mathscr{R}$.

Now a run $(\widehat{C}, \#) \xrightarrow{\sigma} (\widehat{C'}, \#)$ of $\mathscr{P}$ is said to be in *normal form* if it is either the empty run or if it can be decomposed into a sequence of *pseudo-steps*. Hence, it follows that if $(\widehat{C}, \#) \xrightarrow{\sigma} (\widehat{C'}, \#)$ is a run in normal form then $C \xrightarrow{*} C'$ in $\mathscr{R}$.

The following lemma asserts that whenever there is a run between two good configurations of $\mathscr{P}$, then there is also a run between those configurations in normal form. Hence, using this lemma and the discussion in the previous paragraph, it follows that if $(\widehat{C}, \#) \xrightarrow{*} (\widehat{C'}, \#)$ in $\mathscr{P}$ then $C \xrightarrow{*} C'$ in $\mathscr{R}$.

Lemma 3 (Normal form lemma). *Suppose $(\widehat{C}, \#) \xrightarrow{\rho} (\widehat{C'}, \#)$ is a run in $\mathscr{P}$. Then there exists σ such that $(\widehat{C}, \#) \xrightarrow{\sigma} (\widehat{C'}, \#)$ is a run in normal form.*

Proof sketch of normal form lemma. Let n be the length of ρ. We proceed by induction on n. If $n = 0$, we are done. Let $n > 0$ and $\rho = \rho_1, \ldots, \rho_n$. Assume now that any run of length strictly less than n can be put in normal form. By analysing the structure of the transitions in $\mathscr{P}$ and noticing that ρ begins at a good configuration, we can first show that ρ_1 must be of the form $q \xrightarrow{W(a)} [q, a, q']$ for some $a \in \Sigma$. Then we consider two cases:

Case 1: Suppose there is no $i > 1$ such that ρ_i is a transition which writes a value $b \neq \#$. Hence, every transition in $\rho_2, \ldots, \rho_n$ either reads the value a or writes $\#$ and so there must be an index $2 \leq j \leq n$ such that every transition in $\rho_2, \ldots, \rho_{j-1}$ reads a and every transition in $\rho_j, \ldots, \rho_n$ writes $\#$. Now, by analysing the transitions going in and out of the subset I and noticing that the run begins and ends at good configurations, we can show that $(\widehat{C}, \#) \xrightarrow{\rho} (\widehat{C'}, \#)$ must be a pseudo-step.

Case 2: Suppose there is $i > 1$ such that ρ_i is a transition which writes a value $b \neq \#$. By the same argument as before, it is easy to see that there must exist $2 \leq j \leq i - 1$ such that every transition in $\rho_2, \ldots, \rho_{j-1}$ reads a and every transition in $\rho_j, \ldots, \rho_{i-1}$ writes $\#$. Let Z be the configuration reached after ρ_{i-1}. Let $M = Z(I)$, i.e., M is the multiset of processes at the configuration Z which are in some intermediary state. Since the only way out of the set I is to write $\#$ onto the register, if $M = \{[p_1, a, p'_1], \ldots, [p_k, a, p'_k]\}$ then there must exist $i_1, \ldots, i_k > i$ such that each ρ_{i_l} is $[p_l, a, p'_l] \xrightarrow{W(\#)} p'_l$. We can then rearrange the run by first following ρ up till ρ_{i-1}, then "preponing" the transitions $\rho_{i_1}, \ldots, \rho_{i_k}$ and then firing the rest of ρ to reach $(\widehat{C'}, \#)$. With this rearrangement, the run up till ρ_{i_k} becomes a pseudo-step and so we can apply induction hypothesis on the rest of the run. $\qquad\square$

The reduction With this construction, we can now simulate RBN by ASMS as follows: Let $\mathscr{R}$ be an RBN with states $Q_{\mathscr{R}}$ and let $\mathscr{C}_1 = (L_1, U_1), \mathscr{C}'_1 = (L'_1, U'_1)$ be two cubes of $\mathscr{R}$. Construct the ASMS $\mathscr{P}$ as described above. Then construct the following two cubes $\mathscr{C}_2 = (L_2, U_2, \#), \mathscr{C}'_2 = (L'_2, U'_2, \#)$ of $\mathscr{P}$: $L_2(q), U_2(q), L'_2(q)$ and $U'_2(q)$ are respectively equal to $L_1(q), U_1(q), L'_1(q)$ and $U'_1(q)$ if q is a state of $\mathscr{R}$. If q is in I, then $L_2(q) = U_2(q) = L'_2(q) = U'_2(q) = 0$. It is then easy to see that $C \in \mathscr{C}_1$ (resp. $\mathscr{C}'_1$) iff $\widehat{C} \in \mathscr{C}_2$ (resp. $\mathscr{C}'_2$). Hence, by correctness of our construction, it follows that $\mathscr{C}_1$ can reach $\mathscr{C}'_1$ iff $\mathscr{C}_2$ can reach $\mathscr{C}'_2$.

4.2 RBN Simulate ASMS

Construction Let $\mathscr{P} = (Q_{\mathscr{P}}, \Sigma, \delta_{\mathscr{P}})$ be an ASMS. We construct an RBN $\mathscr{R}$ where one agent acts like the register of $\mathscr{P}$ and all the other agents behave like agents of $\mathscr{P}$. Let $\mathscr{R} = (Q_{\mathscr{R}}, \Sigma_{\mathscr{R}}, \delta_{\mathscr{R}})$ be an RBN defined as follows: The set of states $Q_{\mathscr{R}}$ is comprised of two parts. The first part consists of the set $Q_{\mathscr{P}} \cup \{[p, a, q] : p \xrightarrow{W(a)} q \in \delta_{\mathscr{P}}\}$, which will intuitively be used to simulate the processes of $\mathscr{P}$. The second part consists of the set $\Sigma \cup \{\overline{a} : a \in \Sigma\}$ which will intuitively be used to simulate the register of $\mathscr{P}$. The set $\{\overline{a} : a \in \Sigma\}$ is denoted by $\overline{\Sigma}$. The alphabet $\Sigma_{\mathscr{R}}$ is $\{Read_a, Ch_a, Ack_a : a \in \Sigma\}$.

$$!Read_a$$

$$q \xrightarrow{?Ch_a} [q,a,q'] \xrightarrow{!Ack_a} q' \qquad d \xrightarrow{!Ch_a} \overline{a} \xrightarrow{?Ack_a} a \qquad p \xrightarrow{?Read_a} p'$$
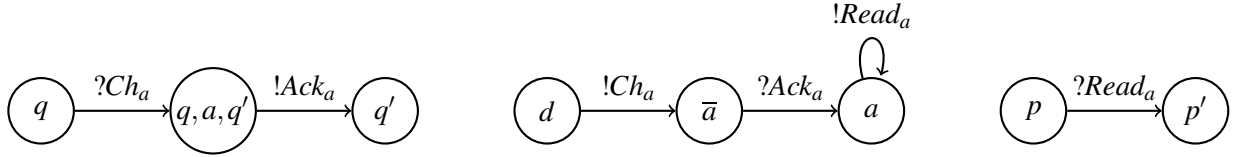
Figure 4: Simulation in $\mathscr{R}$ of transitions $q \xrightarrow{W(a)} q'$ and $p \xrightarrow{R(a)} p'$ of $\mathscr{P}$.

Before describing the transition relation $\delta_{\mathscr{R}}$ we set up some notation: A *good* configuration of $\mathscr{R}$ is a configuration C such that $\sum_{a \in \Sigma} C(a) = 1$ and $C(p) = 0$ if $p \notin Q_{\mathscr{P}} \cup \Sigma$. Intuitively, in a good configuration, there is one process which stores the value of the register of $\mathscr{P}$ and all the other processes are in some state of $Q_{\mathscr{P}}$. Notice that there is a natural bijection between configurations of $\mathscr{P}$ and good configurations of $\mathscr{R}$. If C is a configuration of $\mathscr{P}$, we will use $\widehat{C}$ to denote the corresponding good configuration of $\mathscr{R}$.

Now, the transition relation $\delta_{\mathscr{R}}$ is constructed so that the following invariant is satisfied: For any configurations C and C' of $\mathscr{P}$, $C' \in post^*_{\mathscr{P}}(C)$ iff $\widehat{C'} \in post^*_{\mathscr{R}}(\widehat{C})$.

- Suppose $t = p \xrightarrow{R(a)} q$ is a transition in $\mathscr{P}$. Correspondingly, we have two transitions $a \xrightarrow{!Read_a} a$ and $p \xrightarrow{?Read_a} q$ in $\mathscr{R}$. Hence, if $C \xrightarrow{t} C'$ in $\mathscr{P}$, then $\widehat{C} \xrightarrow{(a,!Read_a,a)+(p,?Read_a,q)} \widehat{C'}$ in $\mathscr{R}$.

- Suppose $t = p \xrightarrow{W(a)} q$ is a transition in $\mathscr{P}$. We first have two transitions $p \xrightarrow{?Ch_a} [p,a,q]$ and $[p,a,q] \xrightarrow{!Ack_a} q$. Further, for **every** $d \in \Sigma$, we have the transitions, $d \xrightarrow{!Ch_a} \overline{a}$ and $\overline{a} \xrightarrow{?Ack_a} a$. Intuitively, the process responsible for the register requests to *change* the value of the register from d to a by broadcasting the message Ch_a and moving to $\overline{a}$. The process at state p is capable of receiving this message and moves to the state $[p,a,q]$ and from there it is capable of sending the message Ack_a *acknowledging* the change sent by the register. The process at $\overline{a}$ can receive Ack_a and move to a. Hence, if $C \xrightarrow{t} C'$ then $\widehat{C} \xrightarrow{(d,!Ch_a,\overline{a})+(p,?Ch_a,[p,a,q])} C_{int} \xrightarrow{([p,a,q],!Ack_a,q)+(\overline{a},?Ack_a,a)} \widehat{C'}$. Figure 4 represents the transitions needed for this simulation.

Correctness of construction Hence, if $C \xrightarrow{*} C'$ in $\mathscr{P}$ then we have shown that $\widehat{C} \xrightarrow{*} \widehat{C'}$ in $\mathscr{R}$. Notice that we have also shown that it is possible to go from $\widehat{C}$ to $\widehat{C'}$ where every broadcasted message is received by *exactly* one other process. Our next lemma shows that this is not an accident, and indeed any run between $\widehat{C}$ and $\widehat{C'}$ can be transformed into this form.

A run between good configurations of $\mathscr{R}$ is said to be in *normal form* if whenever $Z \xrightarrow{t+t_1,\ldots,t_n} Z'$ is a step in that run, then $n = 1$. We have the following lemma.

Lemma 4 (Normal form lemma). *Suppose there is a run from Z to Z' in $\mathscr{R}$ where Z and Z' are good configurations. Then there is a run from Z to Z' which is in normal form.*

First we will see how our simulation is correct, using the normal form lemma. Suppose $\widehat{C} \xrightarrow{*} \widehat{C'}$ in $\mathscr{R}$ for some configurations C and C' of $\mathscr{P}$. By the normal form lemma, we can assume that this run is in normal form and so let $\widehat{C} \xrightarrow{b^1+r^1} C_1 \xrightarrow{b^2+r^2} C_2 \ldots C_{m-1} \xrightarrow{b^m+r^m} \widehat{C'}$. We proceed by induction on m. The base case of $m = 0$ is trivial. Suppose $m > 0$ and assume the claim holds for all numbers less than m. Since $\widehat{C}$ is a good configuration, there are only two possible cases for b^1:

Case 1: Suppose $b^1 = (a, !Read_a, a)$ for some $a \in \Sigma$. Hence r^1 must be $(p, ?Read_a, q)$ for some $p, q \in Q_{\mathscr{P}}$. It follows that $C_1 = \widehat{Z}$ for some configuration Z of $\mathscr{P}$. Since $C \xrightarrow{(p,R,a,q)} Z$ in $\mathscr{P}$, by applying the induction hypothesis on the run from $\widehat{Z}$ to $\widehat{C'}$, we are done.

Case 2: Suppose $b^1 = (d, !Ch_a, \overline{a})$ for some $d, a \in \Sigma$. Hence r^1 must be $(p, ?Ch_a, [p, a, q])$ for some $p, q \in Q_{\mathscr{P}}$. The only process which can broadcast from C_1 is the process at $[p, a, q]$ and moreover it can only broadcast Ack_a. The only process which can receive Ack_a from C_1 is the process at the state $\overline{a}$. Hence $b^2 = ([p, a, q], !Ack_a, q)$ and $r^2 = (\overline{a}, ?Ack_a, a)$. Therefore, $C_2 = \widehat{Z}$ for some configuration Z of $\mathscr{P}$. Since $C \xrightarrow{(p,W,a,q)} Z$ in $\mathscr{P}$, by applying the induction hypothesis on the run from $\widehat{Z}$ to $\widehat{C'}$, we are done.

Proof sketch of normal form lemma. Suppose $Z_0 := Z \xrightarrow{b^1 + r_1^1, \ldots, r_{n_1}^1} Z_1 \xrightarrow{b^2 + r_1^2, \ldots, r_{n_2}^2} Z_2 \ldots Z_{m-1} \xrightarrow{b^m + r_1^m, \ldots, r_{n_m}^m} Z_m := Z'$. We proceed by induction on m. The case of $m = 0$ is trivial.

Suppose $m > 0$ and assume that the claim is true for all numbers less than m. Since Z_0 is a good configuration, there are only two possible choices for b^1.

Case 1: Suppose $b^1 = a \xrightarrow{!Read_a} a$ for some $a \in \Sigma$. By firing b^1 repeatedly, we can fire $r_1^1, r_2^1, \ldots, r_{n_1}^1$ "one at a time" and reach Z_1 from Z_0 using a run in normal form. We can then apply the induction hypothesis on the run between Z_1 and Z'.

Case 2: Suppose $b^1 = d \xrightarrow{!Ch_a} \overline{a}$ for some $d, a \in \Sigma$. Hence, Z_1 is a bad configuration and so $Z_1 \neq Z'$. If $n_1 = 0$, then no process in Z_1 can broadcast any message, which leads to a contradiction. So, $n_1 > 0$.

For each $1 \leq i \leq n_1$, let $r_i^1 = (p_i, ?Ch_a, [p_i, a, q_i])$. Let $S := \sum_{i=2}^{n_1} p_i - \sum_{i=2}^{n_1} [p_i, a, q_i]$ and let $M := \sum_{i=2}^{n_1} [p_i, a, q_i]$. Notice that the only processes which can broadcast a message at the configuration Z_1 are the processes in the multiset $[p_1, a, q_1] + M$. Hence $b^2 = (p_i[a]q_i, !Ack_a, q_i)$ for some i. Without loss of generality, we can assume that $i = 1$.

Notice that the only process which can receive the message Ack_a at the configuration Z_1 is the process at the state $\overline{a}$. It then follows that either $n_2 = 0$ or $n_2 = 1$. Hence, we get two subcases:

Case 2a): Suppose $n_2 = 0$. Then reorder the run between Z_0 and Z_2 as follows: $Z_0 \xrightarrow{b^1 + r_1^1} Z_1 + S \xrightarrow{b^2 + (\overline{a}, ?Ack_a, a)} Z_2 + S - \overline{a} + a \xrightarrow{(a, !Ch_a, \overline{a}) + r_2^1, \ldots, r_{n_1}^1} Z_2$. Notice that the configuration $Z_2 + S - \overline{a} + a$ is a good configuration and has a run of length $m - 1$ to Z'. Applying induction hypothesis, we are then done.

Case 2b): Suppose $n_2 = 1$. Hence, $r_1^2 = (\overline{a}, ?Ack_a, a)$ and so $Z_2(\overline{a}) = 0$ and $Z_2(a) = 1$. We consider two further subcases:

- Suppose there exists $\alpha > 2$ such that $Z_\alpha(\overline{a}) = 1$. Let α be the minimum such index. Hence, there must exist some $d' \in \Sigma$ such that b^α is $(d', !Ch_a, \overline{a})$.

  Suppose no $b \in \{b^3, \ldots, b^{\alpha-1}\}$ is labelled by $!Ack_a$. Intuitively, we can then show that none of the processes in any of the states in the multiset M ever make a step between Z_2 and Z_α. Hence, we can "postpone" firing the transitions $r_2^1, \ldots, r_{n_1}^1$ and get $Z_0 \xrightarrow{b^1 + r_1^1} Z_1 + S \xrightarrow{b^2 + r_1^2} Z_2 + S \xrightarrow{*} Z_{\alpha-1} + S \xrightarrow{b^\alpha + r_2^1, \ldots, r_{n_1}^1, r_1^\alpha, \ldots, r_{n_\alpha}^\alpha} Z_\alpha$. The configuration $Z_2 + S$ is a good configuration and has a run to Z' of length $m - 2$ and so we can apply the induction hypothesis.

  Suppose some $b \in \{b^3, \ldots, b^{\alpha-1}\}$ is labelled by $!Ack_a$. Let $b = b^i$ be the first such transition. By definition of α and by construction of the protocol, we can show that b^i must be $([p_j, a, q_j], !Ack_a, q_j)$ for some $2 \leq j \leq n_1$ (without loss of generality we can assume $j = 2$) and we can also show that no process at the ith step receives this message, i.e. $n_i = 0$. Hence, we can "prepone" firing the transition b^i and get $Z_0 \xrightarrow{b^1 + r_1^1} Z_1 + S \xrightarrow{b^2 + r_1^2} Z_2 + S \xrightarrow{(a, !Ch_a, \overline{a}) + r_2^1, \ldots, r_{n_1}^1} Z_2 - a + \overline{a} \xrightarrow{(b^i + (\overline{a}, ?Ack_a, a))} Z_2 - [p_2, a, q_2] + p_2 \xrightarrow{*} Z_{i-2} - [p_2, a, q_2] + p_2 \xrightarrow{b^{i-1} + r_1^{i-1} \ldots r_{n_{i-1}}^{i-1}} Z_i$. Notice that $Z_2 + S$ is a good configuration and has a run to Z' of length $< m$ and so we can apply the induction hypothesis.

- Suppose there does not exist $\alpha > 2$ such that $Z_\alpha(\overline{a}) = 1$.

Suppose no $b \in \{b^3, \ldots, b^m\}$ is labelled by $!Ack_a$. We can once again show that none of the processes in the multiset M ever make a step between Z_2 and Z_m. Since Z_m is a good configuration, it must then be the case that $n_1 = 1$, which means that Z_2 is a good configuration and the run between Z_0 and Z_2 is already in normal form. Because of the induction hypothesis, we are done.

Suppose some $b \in \{b^3, \ldots, b^m\}$ is labelled by $!Ack_a$. Let $b = b^i$ be the first such transition. In this case, we can do a similar rearrangement like the corresponding previous case by "preponing" b^i and then conclude by applying the induction hypothesis.

$\square$

The reduction Now, suppose we are given an ASMS $\mathscr{P}$ and two cubes $\mathscr{C}_1 = (L_1, U_1, d)$ and $\mathscr{C}_1' = (L_1', U_1', d')$. We construct the protocol $\mathscr{R}$ as we have described in this section. Then we construct two cubes $\mathscr{C}_2 = (L_2, U_2)$ and $\mathscr{C}_2' = (L_2', U_2')$ of $\mathscr{R}$ as follows: $L_2(q), U_2(q), L_2'(q)$ and $U_2'(q)$ are all respectively equal to $L_1(q), U_1(q), L_1'(q)$ and $U_1'(q)$ if $q \in Q_\mathscr{P}$, $L_2(d) = U_2(d) = L_2'(d') = U_2'(d') = 1$ and otherwise $L_2(q) = U_2(q) = L_2'(q) = U_2'(q) = 0$. It is easy to see that a configuration $C \in \mathscr{C}_1$ (resp. $\mathscr{C}_1'$) iff its corresponding configuration $\hat{C} \in \mathscr{C}_2$ (resp. $\mathscr{C}_2'$). Hence, by our simulation it follows that $\mathscr{C}_1$ can reach $\mathscr{C}_1'$ in $\mathscr{P}$ iff $\mathscr{C}_2$ can reach $\mathscr{C}_2'$ in $\mathscr{R}$.

Another reduction While this construction proves the desired result, we need a slightly different construction for the purposes of the next section which we now describe. Given an ASMS $\mathscr{P}$ and two cubes $\mathscr{C}_1 = (L_1, U_1, d)$ and $\mathscr{C}_1' = (L_1', U_1', d')$, once again construct the RBN $\mathscr{R}$ described in this section and construct two cubes $\mathscr{C}_3 = (L_3, U_3)$ and $\mathscr{C}_3' = (L_3', U_3')$ of $\mathscr{R}$ as follows: The cube $\mathscr{C}_3'$ is the same as $\mathscr{C}_2'$ described before. The cube $\mathscr{C}_3$ is also exactly the same as $\mathscr{C}_2$, except for the constraints $L_2(d) = U_2(d) = 1$ which are replaced by $L_3(d) = 0, U_3(d) = \infty$.

Since $\mathscr{C}_2 \subseteq \mathscr{C}_3$, it follows from the previous reduction that if $\mathscr{C}_1$ can reach $\mathscr{C}_1'$, then $\mathscr{C}_3$ can reach $\mathscr{C}_3'$. For the other direction, notice that, by construction of the protocol $\mathscr{R}$,

$$\text{If } C \to C' \text{ is a step in } \mathscr{R}, \text{ then } \sum_{q \in \Sigma \cup \overline{\Sigma}} C(q) = \sum_{q \in \Sigma \cup \overline{\Sigma}} C'(q) \tag{1}$$

Using this equation and the fact that any configuration in $\mathscr{C}_3'$ is a good configuration, it is then clear that if $C_3 \in \mathscr{C}_3$ such that $C_3 \xrightarrow{*} C_3'$ with $C_3' \in \mathscr{C}_3'$, then C_3 must also be a good configuration. Hence, we can then conclude that $\mathscr{C}_3$ can reach $\mathscr{C}_3'$ iff $\mathscr{C}_1$ can reach $\mathscr{C}_1'$.

5 Transferring Existing Results

In the previous section, we have shown that RBN and ASMS are polynomial-time equivalent with respect to the cube-reachability problem. Though the precise complexity of this problem has not been established for either one of these models, our result shows that it is sufficient to characterize the complexity of cube-reachability for one of these models. Moreover, there exist results for subclasses of the cube-reachability problem for both RBN and ASMS. In this section, we use the reductions constructed in the previous section to transfer these results from RBN to ASMS and vice versa.

5.1 Unbounded initial cube reachability

We consider the following problem for RBN, which we call the *unbounded initial cube reachability* problem: We are given an RBN $\mathscr{R}$ and two cubes $\mathscr{C} = (L, U), \mathscr{C}' = (L', U')$ with the special property

that $L(q) = 0$ and $U(q) \in \{0, \infty\}$ for every state q and we would like to check if $\mathscr{C}$ can reach $\mathscr{C}'$. Notice that there is no restriction on the cube $\mathscr{C}'$. We will call such a pair $(\mathscr{C}, \mathscr{C}')$ as an *unbounded initial cube pair*. This problem was proved to be PSPACE-complete for RBN in ([11], Theorem 5.5). (In [11], this result is only stated for cubes with constants encoded in unary, but the proof can be modified easily to also give the same upper bound when the constants are encoded in binary).

In a similar way, it is possible to define the corresponding problem for ASMS. Notice that if $(\mathscr{C}, \mathscr{C}')$ is an unbounded initial cube pair for an ASMS $\mathscr{P}$, then the second reduction in Section 4.2 produces an RBN $\mathscr{R}$ along with an unbounded initial cube pair as well. This shows that the corresponding problem for ASMS can be solved in PSPACE.

Further, notice that given an RBN $\mathscr{R}$ and an unbounded initial cube pair for $\mathscr{R}$, our reduction in Section 4.1 produces an ASMS $\mathscr{P}$ with an unbounded initial cube pair as well. This shows that the unbounded initial cube reachability problem for ASMS is PSPACE-hard.

Theorem 5. *The unbounded initial cube reachability problem for ASMS is PSPACE-complete.*

5.2   Leader protocols

The notion of an ASMS equipped with a *leader* has been studied in [14, 9]. Formally, an ASMS-leader protocol is a pair of ASMS protocols $\mathscr{P}_C = (Q_C, \Sigma, \delta_C), \mathscr{P}_D = (Q_D, \Sigma, \delta_D)$, where $\mathscr{P}_C$ is called the *contributor protocol* and $\mathscr{P}_D$ is called the *leader protocol*. Intuitively, there is exactly one process which executes $\mathscr{P}_D$ (the leader) and all the other processes execute $\mathscr{P}_C$ (contributors). This is formalized as follows: A configuration of such a system is defined to be a triple (q, M, a) where $q \in Q_D$, M is a multiset on Q_C and $a \in \Sigma$. A step between $C = (q, M, a)$ and $C' = (q', M', a')$ exists if one of the following is true:

- There exists $(q, \mathrm{op}, a', q') \in \delta_D$ such that $M' = M$ and either $\mathrm{op} = R$ and $a = a'$, or $\mathrm{op} = W$.

- There exists $(p, \mathrm{op}, a', p') \in \delta_C$ such that $q = q'$, $M(p) \geq 1$, $M' = M - \boldsymbol{p} + \boldsymbol{p}'$, and either $\mathrm{op} = R$ and $a = a'$, or $\mathrm{op} = W$.

We can then define the notion of a run for an ASMS-leader protocol in the usual way. The *ASMS-leader reachability* problem is to decide, given an ASMS-leader protocol $(\mathscr{P}_C, \mathscr{P}_D)$, two leader states q_D^I, q_D^f, a contributor state q_C^I and two data values $a, a' \in \Sigma$ whether there exists a $k \geq 1$ such that the configuration $(q_D^I, \langle k \cdot q_C^I \rangle, a)$ can reach a configuration $C' = (q_D^f, M', a')$ for some M'.

We now define a special case of cube-reachability in ASMS and notice that this special case is exactly equivalent to ASMS-leader reachability. An *ASMS-leader cube* is a pair $(\mathscr{P}, \mathscr{C}, \mathscr{C}')$ of the following form: The protocol $\mathscr{P} = (Q, \Sigma, \delta)$ is such that there exists a partition of the states and transition relation as $Q = Q_C \cup Q_D, \delta = \delta_C \cup \delta_D$ and $\mathscr{C} = (L, U, a), \mathscr{C}' = (L', U', a')$ satisfy: There exists exactly two states $q_D^I, q_D^f \in Q_D$ such that $L(q_D^I) = U(q_D^I) = 1, L'(q_D^f) = U'(q_D^f) = 1$ and for every other state $q \in Q_D$, $L(q) = L'(q) = U(q) = U'(q) = 0$ and there exists exactly one state $q_C^I \in Q_C$ such that $L(q_C^I) = L'(q_C^I) = 0, U(q_C^I) = U'(q_C^I) = \infty$ and for every other state $q \in Q_C$, $L(q) = U(q) = L'(q) = 0, U'(q) = \infty$. Notice that Example 2 is an example of an ASMS-leader cube.

It is easy to see that the ASMS-leader reachability problem is equivalent to the cube-reachability problem for ASMS-leader cubes. The following result has been shown for ASMS.

Theorem 6 ([14]). *The ASMS-leader reachability problem is in NP.*

Now, we can define RBN-leader protocols and RBN-leader cubes in exactly the same way as was done for ASMS. Further, notice that the reduction given in Section 4.1 has the following special property: If we are given an RBN-leader cube $(\mathscr{R}, \mathscr{C}_1, \mathscr{C}_1')$, then the reduction produces an ASMS-leader cube

$(\mathscr{P}, \mathscr{C}_2, \mathscr{C}_2')$. This proves that the RBN-leader cube reachability problem (and hence the RBN-leader reachability problem) is in NP.

Notice that the reduction given in Section 4.2 does not output a RBN-leader cube when it is given an ASMS-leader cube as input. Hence, we do not immediately get NP-hardness of the RBN-leader reachability problem. Nevertheless, by a reduction from 3-SAT similar to that of the one given in Theorem 10 of [9], we can prove NP-hardness of RBN-leader reachability. We then get

Theorem 7. *The RBN-leader reachability problem is NP-complete.*

5.3 Almost-sure coverability

We now consider the notion of *almost-sure coverability* for ASMS. Let $\mathscr{P} = (Q, \Sigma, \delta)$ be an ASMS with two distinguished states q_I, q_f and a distinguished letter $d \in \Sigma$. Let $\uparrow q_f$ denote the set of all configurations C such that $C(q_f) \geq 1$. For any $k \geq 1$, we say that the configuration $(\wr k \cdot q_I \wr, d)$ *almost-surely covers* q_f iff $post^*((\wr k \cdot q_I \wr, d)) \subseteq pre^*(\uparrow q_f)$. The reason behind calling this the almost-sure coverability relation is that the definition given here is equivalent to covering the state q_f from $(\wr k \cdot q_I \wr, d)$ with probability 1 under a probabilistic scheduler which picks processes uniformly at random at each step.

The number k is called a *cut-off* if one of the following is true: 1) Either for all $h \geq k$, the configuration $(\wr h \cdot q_I \wr, d)$ almost-surely covers q_f. In this case, k is a positive cut-off. Or, 2) for all $h \geq k$, the configuration $(\wr h \cdot q_I \wr, d)$ does not almost-surely cover q_f. In this case, k is a negative cut-off. Note that from the definition alone, it is not clear that a cut-off must exist for every ASMS. The following result is known.

Theorem 8 (Theorem 3 of [6]). *Given an ASMS with two states q_I, q_f and a letter d, a cut-off always exists. Whether the cut-off is positive or negative can be decided in EXPSPACE.*

We can now translate this result to RBNs. Given an RBN $\mathscr{R}$ and two states q_I, q_f, we first set $\uparrow q_f := \{C : C(q_f) \geq 1\}$. Then for any $k \geq 1$, we say that $\wr k \cdot q_I \wr$ almost-surely covers q_f iff $post^*(\wr k \cdot q_I \wr) \subseteq pre^*(\uparrow q_f)$. We can then define positive and negative cut-offs in a similar manner. Now for the RBN $\mathscr{R}$, let $\mathscr{P}$ be the ASMS protocol that we construct in our reduction given in Section 4.1. Using the construction of $\mathscr{P}$, we can then easily show that

for any $k \geq 1$, $post^*_{\mathscr{R}}(\wr k \cdot q_I \wr) \subseteq pre^*_{\mathscr{R}}(\uparrow q_f)$ iff $post^*_{\mathscr{P}}(\wr k \cdot q_I \wr, \#) \subseteq pre^*_{\mathscr{P}}(\uparrow q_f)$.

This then directly implies that

Theorem 9. *Given an RBN with two states q_I, q_f, a cut-off always exists. Whether the cut-off is positive or negative can be decided in EXPSPACE.*

6 A Third Model

We have shown that RBN and ASMS are cube-reachability equivalent and using this we have transferred some results between these two models. In this section, we will introduce a third model called *Immediate Observation* (IO) nets and show that cube-reachability for IO nets can be reduced to cube-reachability for RBN. Further, we show that a stronger notion of reduction – which is satisfied by all the reductions given in this paper – cannot exist from RBN to IO nets.

6.1 Immediate Observation Nets

Immediate observation nets, or *IO nets*, were introduced in [16]. They are a subclass of Petri nets with applications in population protocols and chemical reaction networks. An IO net is a Petri net with transitions of a certain shape: Informally, a process (or token) in a state (or place) *p observes* the presence of a process in q and moves to state p', for some states p, q, p' not necessarily distinct. Because of this, IO nets can be described in a simpler manner that does not use the full Petri net formalism. We will present them this way here, to highlight the similarity to the other models and to simplify notation.

Definition 10. *An* immediate observation net *is a tuple $\mathcal{N} = (Q, \delta)$ where Q is a finite set of states and $\delta \subseteq Q \times Q \times Q$ is the transition relation.*

If $(p, q, p') \in \delta$, then we sometimes denote it by $p \xrightarrow{q} p'$. A *configuration* C of an IO net $\mathcal{N}$ is a multiset over Q. It intuitively counts the number of processes in each state. There is a *step* from a configuration C to a configuration C' if there exists $t = p \xrightarrow{q} p' \in \delta$, such that $C \geq \{p, q\}$ and $C' = C - \boldsymbol{p} + \boldsymbol{p'}$. We denote by $C \xrightarrow{t} C'$ such a step, and by $\xrightarrow{*}$ the reflexive transitive closure of the step relation. We can then define runs of an IO net in the usual way.

6.2 RBN Simulate IO Nets

Construction Let $\mathcal{N} = (Q, \delta)$ be an IO net. We construct an RBN that simulates $\mathcal{N}$ in which processes send messages signaling their current state. Let $\mathcal{R} = (Q', \Sigma', \delta')$ be the following RBN: The set of states Q' and the alphabet Σ' are both equal to Q. The transition relation δ' is such that for every $q \in Q$ there is a transition $q \xrightarrow{!q} q$ in δ', and for every $p \xrightarrow{q} p' \in \delta$ there is a transition $p \xrightarrow{?q} p'$ in δ'.

Correctness of construction There is a natural bijection between configurations of $\mathcal{R}$ and configurations of $\mathcal{P}$. If C is a configuration of $\mathcal{N}$, we will abuse notation and denote the corresponding configuration of $\mathcal{R}$ also as C. We now show that $C' \in post^*_{\mathcal{N}}(C)$ iff $C' \in post^*_{\mathcal{R}}(C)$ for any configurations C and C' of $\mathcal{N}$. Indeed, if C reaches C' by one step $p \xrightarrow{q} p'$ in $\mathcal{N}$, then $C \xrightarrow{t + t_1} C'$ with $t = q \xrightarrow{!q} q$ and $t_1 = p \xrightarrow{?q} p'$ in $\mathcal{R}$. Conversely, let $C \xrightarrow{t + t_1, \dots, t_k} C'$ be a step in $\mathcal{R}$ with $t = q \xrightarrow{!q} q$ and $t_i = p_i \xrightarrow{?q} p'_i$ for some $k \geq 0$. The step must be of this form because the only broadcast transitions of $\mathcal{R}$ are of the form $q \xrightarrow{!q} q$. Then C reaches C' by the sequence of transitions $(p_1 \xrightarrow{q} p'_1), (p_2 \xrightarrow{q} p'_2), \dots, (p_k \xrightarrow{q} p'_k)$ in $\mathcal{N}$.

The reduction With this construction, RBN can simulate IO nets as follows: Let $\mathcal{N}$ be an IO net and let $\mathscr{C}_1 = (L_1, U_1), \mathscr{C}'_1 = (L'_1, U'_1)$ be two cubes of $\mathcal{N}$. Construct the RBN $\mathcal{R}$ as described above, and let $\mathscr{C}_2 = \mathscr{C}_1$ and $\mathscr{C}'_2 = \mathscr{C}'_1$. By our construction, $\mathscr{C}_1$ can reach $\mathscr{C}'_1$ iff $\mathscr{C}_2$ can reach $\mathscr{C}'_2$.

Consequences. In [11], two further restrictions of the unbounded initial cube reachability problem (presented in Section 5.1) are considered. The first restriction, dubbed $CRP[\geq 1]$ (where CRP stands for cardinality reachability problem), considers only unbounded initial cube pairs $\mathscr{C}, \mathscr{C}'$ in which $\mathscr{C}' = (L', U')$ is such that $L'(q) \in \{0, 1\}$ and $U'(q) = \infty$ for all q. The second restriction, dubbed $CRP[\geq 1, = 0]$, considers only unbounded initial cube pairs $\mathscr{C}, \mathscr{C}'$ in which $\mathscr{C}' = (L', U')$ is such that $L'(q) \in \{0, 1\}$ and $U'(q) \in \{0, \infty\}$ for all q. For RBN, the problems $CRP[\geq 1]$ and $CRP[\geq 1, = 0]$ are shown to be in PTIME and in NP (Theorem 3.3 and 4.3 of [11]), respectively. By the construction given above, it is then immediately clear that

Theorem 11. *For IO nets, $CRP[\geq 1]$ and $CRP[\geq 1, = 0]$ are in PTIME and in NP respectively.*

Strong simulation. Consider the following alternative definition of simulation between models *A* and *B* (where a model is to be understood as either an RBN, ASMS or IO net): Given an instance *I* of model *A* with states Q_I, there exists an instance *J* of model *B* with states Q_J such that *J* is polynomial in the size of *I* with $Q_I \subseteq Q_J$, and there exists a multiset *h* over $Q_J \setminus Q_I$ of polynomial size such that $C' \in post^*(C)$ if and only if $C' \cdot h \in post^*(C \cdot h)$ for any configurations C, C' of *I*. Notice that strong simulation is a transitive relation. The simulation constructions of this paper verify this strong definition of simulation.

Theorem 12. *RBN and ASMS strongly simulate each other. Further, IO nets are strongly simulated by RBN (and hence by ASMS as well).*

We show that this is not the case for IO nets: they cannot strongly simulate RBN (nor ASMS).

6.3 IO Does not Strongly Simulate RBN

Assuming that IO nets can strongly simulate RBN, we will derive a contradiction. Under this assumption, we will first transfer results on the closure of cubes from IO nets to RBN, then exhibit a particular RBN which contradicts these results. We start by recalling definitions and properties relating to cubes.

Counting sets and norms. We consider cubes over a finite set *Q*. A finite union of cubes $\bigcup_{i=1}^m (L_i, U_i)$ is called a *counting constraint* and the set of configurations $\bigcup_{i=1}^m \mathscr{C}_i$ it describes is called a *counting set*. We write $[\![\Gamma]\!]$ for the counting set described by the counting constraint Γ. 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 [15]) that counting constraints and counting sets are closed under Boolean operations.

Let $\mathscr{C} = (L, U)$ be a cube. Let $\|\mathscr{C}\|_l$ be the the sum of the components of *L*. Let $\|\mathscr{C}\|_u$ be the sum of the finite components of *U* if there are any, and 0 otherwise. We call *norm* of $\mathscr{C}$ the maximum of $\|\mathscr{C}\|_l$ and $\|\mathscr{C}\|_u$, denoted by $\|\mathscr{C}\|$. We define the norm of a counting constraint $\Gamma = \bigcup_{i=1}^m \mathscr{C}_i$ as $\|\Gamma\| \stackrel{\text{def}}{=} \max_{i \in [1,m]} \{\|\mathscr{C}_i\|\}$. The norm of a counting set $\mathscr{S}$ is the smallest norm of a counting constraint representing $\mathscr{S}$, that is, $\|\mathscr{S}\| \stackrel{\text{def}}{=} \min_{\mathscr{S} = [\![\Gamma]\!]} \{\|\Gamma\|\}$. Proposition 5 of [15] entails the following results for the norms of the union, intersection and complement.

Proposition 13. *Let $\mathscr{S}_1, \mathscr{S}_2$ be counting sets. The norms of the union, intersection and complement satisfy: $\|\mathscr{S}_1 \cup \mathscr{S}_2\| \leq \max\{\|\mathscr{S}_1\|, \|\mathscr{S}_2\|\}$, $\|\mathscr{S}_1 \cap \mathscr{S}_2\| \leq \|\mathscr{S}_1\| + \|\mathscr{S}_2\|$ and $\|\mathbb{N}^n \setminus \mathscr{S}_1\| \leq \|\mathscr{S}_1\| + \|\mathscr{S}_2\|$.*

The following result for IO nets is deduced directly from Theorem 6 in [17]. It states that the forward and backward reachability set of a counting set is still a counting set, and bounds its norms polynomially. This result, transferred to RBN under the assumption of a strong simulation, will amount to a contradiction.

Theorem 14. *Let $\mathscr{N} = (Q, \delta)$ be an IO net, and let $\mathscr{S}$ be a counting set of $\mathscr{N}$. Then $post^*(\mathscr{S})$ is also a counting set and $\|post^*(\mathscr{S})\| \leq \|\mathscr{S}\| + |Q|^3$. The same holds for $pre^*(\mathscr{S})$.*

Assuming that IO nets strongly simulate RBN, we can transfer the result of Theorem 14 to RBN.

Theorem 15. *Assume that IO nets can strongly simulate RBN. There exists a constant k such that for any RBN $\mathscr{R} = (Q, \Sigma, \delta)$, for any counting set $\mathscr{S}$ of $\mathscr{R}$, $post^*(\mathscr{S})$ is also a counting set and $\|post^*(\mathscr{S})\| \in O(\|\mathscr{S}\| + |Q|)^k$. The same holds for $pre^*(\mathscr{S})$.*

Proof Sketch. It suffices to show the result for $\mathscr{S}$ a cube, since for a counting set $\cup_{i=1}^{n}\mathscr{C}_i$, we have $post^*(\cup_i\mathscr{C}_i) = \cup_i post^*(\mathscr{C}_i)$. Fix an RBN $\mathscr{R} = (Q, \Sigma, \delta)$ and a cube $\mathscr{C}$ over Q. Let $\mathscr{N} = (Q_{\mathscr{N}}, \delta_{\mathscr{N}})$ be the IO net of the strong simulation whose existence we assume. The definition of strong simulation entails the existence of a bijection b from configurations of $\mathscr{R}$ to a subset $\mathscr{G}$ of "good" configurations of $\mathscr{N}$. The bijection verifies that a cube of $\mathscr{R}$ is mapped to a cube of $\mathscr{N}$, and that a cube of $\mathscr{N}$ restricted to configurations of $\mathscr{G}$ is mapped to a cube of $\mathscr{R}$.

Since b preserves cubes, $b(\mathscr{C})$ is a cube. By Theorem 14, $post^*(b(\mathscr{C}))$ is a counting set, and thus there exist cubes $\mathscr{C}_1, \ldots, \mathscr{C}_n$ of $\mathscr{N}$ such that $post^*(b(\mathscr{C})) = \cup_{i=1}^{n}\mathscr{C}_i$. Let $\mathscr{M}$ be the set $\cup_{i=1}^{n} b^{-1}(\mathscr{C}_i|_{\mathscr{G}})$ of $\mathscr{R}$. We show that $post^*(\mathscr{C}) = \mathscr{M}$. Since the $b^{-1}(\mathscr{C}_i|_{\mathscr{G}})$ are cubes by strong simulation, $post^*(\mathscr{C})$ is a counting set as a union of cubes. The size of $post^*(b(\mathscr{C}))$ is polynomial in $\mathscr{C}$ and $\mathscr{R}$ by Theorem 14, and thus the size of $post^*(\mathscr{C})$ is too. $\qquad\square$

Deriving the contradiction. We now exhibit a contradiction to the result of Theorem 15, thus proving that IO nets do not strongly simulate RBN. Recall the RBN represented in Figure 1. We can generalize it to a family of RBN $\mathscr{R}_n = (Q, \Sigma, \delta)$, parameterized by $n \geq 1$, with set of states $\{tok, sent\} \cup \{a_i, b_i, c_i | 1 \leq i \leq n\}$. Let $\mathscr{C}_0$ be the cube in which there are arbitrarily many agents in tok, exactly one agent in each a_i and 0 agents in the other states. Let $\mathscr{C}_f$ be the cube in which there is a least one agent in c_n and an arbitrary number elsewhere. We claim that if we start from a configuration of $\mathscr{C}_0$, we can only reach $\mathscr{C}_f$ if we initially have 2^n or more agents in tok. Indeed we can show by induction on $i \in \{1, \ldots, n\}$ that 1 must be broadcasted 2^i times to reach c_i, and thus that 2^i agents are needed in tok initially to reach c_i. By Proposition 13 and Theorem 15, the set $S := post^*(\mathscr{C}_0) \cap \mathscr{C}_f$ is a counting set of size at most polynomial in $|Q|, \|\mathscr{C}_0\|$ and $\|\mathscr{C}_f\|$. The cubes $\|\mathscr{C}_0\|$ and $\|\mathscr{C}_f\|$ have norms n and 1 respectively, so S is of norm polynomial in n. Thus if it is non-empty it must contain a configuration of size at most polynomial in n: simply take the configuration equal to the lower bounds L of one of the cubes whose union is the counting set $post^*(\mathscr{C}_0) \cap \mathscr{C}_f$. This contradicts the fact that 2^n agents are needed to reach $\mathscr{C}_f$.

References

[1] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer & René Peralta (2006): *Computation in networks of passively mobile finite-state sensors*. Distributed Comput. 18(4), pp. 235–253. Available at `https://doi.org/10.1007/s00446-005-0138-3`.

[2] Dana Angluin, James Aspnes, David Eisenstat & Eric Ruppert (2007): *The computational power of population protocols*. Distributed Comput. 20(4), pp. 279–304. Available at `https://doi.org/10.1007/s00446-007-0040-2`.

[3] A. R. Balasubramanian & Chana Weil-Kennedy (2021): *Reconfigurable Broadcast Networks and Asynchronous Shared-Memory Systems are Equivalent (Long Version)*. Available at `https://arxiv.org/abs/2108.07510`.

[4] Nathalie Bertrand & Paulin Fournier (2013): *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, doi:`10.4230/LIPIcs.FSTTCS.2013.501`.

[5] Nathalie Bertrand, Paulin Fournier & Arnaud Sangnier (2014): *Playing with Probabilities in Reconfigurable Broadcast Networks*. In: *Foundations of Software Science and Computation Structures - 17th International Conference, FOSSACS*, pp. 134–148, doi:`10.1007/978-3-642-54830-7_9`.

[6] Patricia Bouyer, Nicolas Markey, Mickael Randour, Arnaud Sangnier & Daniel Stan (2016): *Reachability in Networks of Register Protocols under Stochastic Schedulers*. In Ioannis Chatzigiannakis, Michael Mitzen-macher, Yuval Rabani & Davide Sangiorgi, editors: *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, LIPIcs 55, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 106:1–106:14. Available at `https://doi.org/10.4230/LIPIcs.ICALP.2016.106`.

[7] Peter Chini, Roland Meyer & Prakash Saivasan (2019): *Complexity of Liveness in Parameterized Systems*. In Arkadev Chattopadhyay & Paul Gastin, editors: *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2019, December 11-13, 2019, Bombay, India*, LIPIcs 150, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 37:1–37:15. Available at `https://doi.org/10.4230/LIPIcs.FSTTCS.2019.37`.

[8] Peter Chini, Roland Meyer & Prakash Saivasan (2019): *Liveness in Broadcast Networks*. In Mohamed Faouzi Atig & Alexander A. Schwarzmann, editors: *Networked Systems - 7th International Conference, NETYS 2019, Marrakech, Morocco, June 19-21, 2019, Revised Selected Papers*, Lecture Notes in Computer Science 11704, Springer, pp. 52–66. Available at `https://doi.org/10.1007/978-3-030-31277-0_4`.

[9] Peter Chini, Roland Meyer & Prakash Saivasan (2020): *Fine-Grained Complexity of Safety Verification*. J. Autom. Reason. 64(7), pp. 1419–1444. Available at `https://doi.org/10.1007/s10817-020-09572-x`.

[10] Giorgio Delzanno, Arnaud Sangnier & Riccardo Traverso (2013): *Parameterized Verification of Broadcast Networks of Register Automata*. In: *Reachability Problems - 7th International Workshop, RP*, pp. 109–121, doi:10.1007/978-3-642-41036-9_11.

[11] Giorgio Delzanno, Arnaud Sangnier, Riccardo Traverso & Gianluigi Zavattaro (2012): *On the Complexity of Parameterized Reachability in Reconfigurable Broadcast Networks*. In Deepak D'Souza, Telikepalli Kavitha & Jaikumar Radhakrishnan, editors: *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, LIPIcs 18, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 289–300. Available at `https://doi.org/10.4230/LIPIcs.FSTTCS.2012.289`.

[12] Giorgio Delzanno, Arnaud Sangnier & Gianluigi Zavattaro (2010): *Parameterized Verification of Ad Hoc Networks*. In Paul Gastin & François Laroussinie, editors: *CONCUR 2010 - Concurrency Theory, 21th International Conference, CONCUR 2010, Paris, France, August 31-September 3, 2010. Proceedings*, Lecture Notes in Computer Science 6269, Springer, pp. 313–327. Available at `https://doi.org/10.1007/978-3-642-15375-4_22`.

[13] Antoine Durand-Gasselin, Javier Esparza, Pierre Ganty & Rupak Majumdar (2017): *Model checking parameterized asynchronous shared-memory systems*. Formal Methods Syst. Des. 50(2-3), pp. 140–167. Available at `https://doi.org/10.1007/s10703-016-0258-3`.

[14] Javier Esparza, Pierre Ganty & Rupak Majumdar (2016): *Parameterized Verification of Asynchronous Shared-Memory Systems*. J. ACM 63(1), pp. 10:1–10:48. Available at `https://doi.org/10.1145/2842603`.

[15] Javier Esparza, Pierre Ganty, Rupak Majumdar & Chana Weil-Kennedy (2018): *Verification of Immediate Observation Population Protocols*. In: *CONCUR*, LIPIcs 118, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 31:1–31:16. Available at `https://doi.org/10.4230/LIPIcs.CONCUR.2018.31`.

[16] Javier Esparza, Mikhail Raskin & Chana Weil-Kennedy (2019): *Parameterized Analysis of Immediate Observation Petri Nets*. In: *Lecture Notes in Computer Science*, 11522, pp. 365–385. Available at `https://doi.org/10.1007/978-3-030-21571-2_20`.

[17] Javier Esparza, Mikhail A. Raskin & Chana Weil-Kennedy (2019): *Parameterized Analysis of Immediate Observation Petri Nets*. In Susanna Donatelli & Stefan Haar, editors: *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 11522, Springer, pp. 365–385. Available at `https://doi.org/10.1007/978-3-030-21571-2_20`.

[18] Marie Fortin, Anca Muscholl & Igor Walukiewicz (2017): *Model-Checking Linear-Time Properties of Parametrized Asynchronous Shared-Memory Pushdown Systems*. In Rupak Majumdar & Viktor Kuncak, editors: *Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part II, Lecture Notes in Computer Science* 10427, Springer, pp. 155–175. Available at `https://doi.org/10.1007/978-3-319-63390-9_9`.

[19] Paulin Fournier (2015): *Parameterized verification of networks of many identical processes. (Vérification paramétrée de réseaux composés d'une multitude de processus identiques)*. Ph.D. thesis, University of Rennes 1, France. Available at `https://tel.archives-ouvertes.fr/tel-01355847`.

[20] Salvatore La Torre, Anca Muscholl & Igor Walukiewicz (2015): *Safety of Parametrized Asynchronous Shared-Memory Systems is Almost Always Decidable*. In Luca Aceto & David de Frutos-Escrig, editors: *26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1.4, 2015, LIPIcs* 42, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 72–84. Available at `https://doi.org/10.4230/LIPIcs.CONCUR.2015.72`.