Int. Journ. of Unconventional Computing, Vol. 0, pp. 1–32
Reprints available directly from the publisher
Photocopying permitted by license only

Population Protocols that Correspond to Symmetric Games*

Olivier Bournez[1], Jérémie Chalopin[2], Johanne Cohen[3], Xavier Koegler[4] and Mikaël Rabie[5]

[1]*Ecole Polytechnique & Laboratoire d'Informatique (LIX), 91128 Palaiseau Cedex, France*
E-mail: Olivier.Bournez@lix.polytechnique.fr
[2]*CNRS & Laboratoire d'Informatique Fondamentale de Marseille, CNRS & Aix-Marseille Université, 39 rue Joliot Curie, 13453 Marseille Cedex 13, France*
E-mail: Jeremie.Chalopin@lif.univ-mrs.fr
[3]*CNRS & PRiSM, 45 Avenue des Etats Unis, 78000 Versailles, France*
E-mail: Johanne.Cohen@prism.uvsq.fr
[4]*École Normale Supérieure & Université Paris Diderot, Paris 7, Case 7014, 75205 Paris Cedex 13, France*
Xavier.Koegler@liafa.jussieu.fr
[5]*École Normale Supérieure de Lyon & 46 Allée d'Italie, 69007 Lyon, France*
E-mail: Mikael.Rabie@ens-lyon.fr

Received: July 18, 2012. Accepted: November 11, 2012.

Population protocols have been introduced by Angluin et al. as a model of networks consisting of very limited mobile anonymous agents that interact in pairs but with no control over their own movement. The model has been considered as a computational model.

In an orthogonal way, several distributed systems have been termed in literature as being realizations of games in the sense of game theory. In this paper, we investigate under which conditions population protocols, or more generally pairwise interaction rules, can be considered as the result of a symmetric game.

We prove that not all symmetric rules can be considered as symmetric games. We prove that some basic protocols can be realized using symmetric games.

Keywords: Population protocols, computation theory, distributed computing, algorithmic game theory

* This work and all authors were partly supported by ANR Project SOGEA and by ANR Project SHAMAN, Xavier Koegler was partly supported by COST Action 295 DYNAMO and ANR Project ALADDIN

1 INTRODUCTION

The population protocol model of Angluin *et al.* [3] describes a population of anonymous finite-state agents that interact in pairs according to a transition function. The agents are identically programmed finite state machines. Interactions between pairs of agents cause the two agents to update their states. These interactions are under the control of an adversary scheduler subject to a fairness constraint. Input values are initially distributed to the agents, and the agents must eventually converge to a common output value that represents the result of the computation. A protocol is said to *(stably) compute* a predicate on the initial states of the agents if, in any fair execution, after finitely many interactions, all agents reach a common output that corresponds to the value of the predicate. The population protocols do not halt but they stabilize.

The population protocol model [4] is motivated to represent sensor networks consisting of very limited mobile agents with no control over their own movement. In the seminal paper [3], the canonical example corresponds to sensors attached to a flock of birds and that must be programmed to check some global properties, like determining whether more than 5% of the population has elevated temperature. Motivating scenarios also include models of the propagation of trust [16]. This model can represent sensor networks, ad-hoc networks, or models from chemistry. All these systems have highly mobile objects.

This goal of our work is to understand when the pairwise interactions of the population protocol can be viewed as the result of a symmetric game. This is inspired by [17, 18, 24] that consider the dynamics of a particular set of rules termed the *PAVLOV* behavior in the iterated prisoner's dilemma. The *PAVLOV* behavior is sometimes also termed *WIN-STAY, LOSE-SHIFT* [9,28]. Our original motivation was to consider rules corresponding to two-player games, and population protocols arose quite incidentally. The main advantage of the setting introduced in [3] is that it provides a clear understanding of what is called a computation by the model. Many distributed systems have been described as the result of games.

In this spririt, we recently discussed the general case of population protocols corresponding to Pavlovian strategies obtained from games and we showed that all predicates computable by protocols can actually be computed by protocols corresponding to games [12]. However, in this paper, we consider the impact of restricting ourselves to symmetric games in which both players have the same set of possible strategies and the result of one instance of the game does not depend on the order in which the players interact. Such a symmetric game yields symmetric transition rules for the corresponding population protocols. As far as we know, the constraint of restricting transitions in a population protocol to symmetric rules has not been explicitly considered, nor has restricting to rules that correspond to symmetric games.

Related work The seminal work [3] proposes a simple version of the population protocol model. In this model, any pair of agents may interact and these interactions are scheduled by an adversary, subject to a fairness constraint. The predicates computable by the unrestricted population protocols have been characterized as being precisely the semi-linear predicates, that is those predicates on counts of input agents definable in first-order Presburger arithmetic [30]. Semi-linearity was shown to be sufficient in [3] and necessary in [5].

Variants of the original model considered so far include restriction to one-way communications [1], restriction to particular interaction graphs [2], random interactions [3], with “speed” [10]. Various kinds of fault tolerance have been considered for population protocols [15], including the search for self-stabilizing solutions [7]. Solutions to classical problems of distributed algorithms have also been considered in this model (see [26]).

Certain works extend this model. On the one hand, the edges of the interaction graph may have states that belong to a constant-size set. This model called the *mediated population protocol* is presented in [25]. The class of stably computable predicates in this model is understood. On the other hand, probabilistic population protocols were proposed in [4], in which the scheduler selects randomly and uniformly the next interaction pair. Some works have concentrated on performance (see e.g. [6]) and focus on a generic definition of probabilistic schedulers [13].

The population protocol model shares many features with other models already considered in the literature. In particular, models of pairwise interactions have been used to study the propagation of diseases [21], or rumors [14]. In chemistry, the chemical master equation has been justified using (stochastic) pairwise interactions between the finitely many molecules [20,27]. In that sense, the model of population protocols may be considered as fundamental in several fields of study, since it appears as soon as anonymous agents interact pairwise.

In this paper, we turn two player games into dynamics over agents, by considering *PAVLOV* behavior. The pairwise interactions between finite-state agents are sometimes motivated by the study of the dynamics of particular two-player games from game theory. For example, Dyer *et al.* [17] considers the dynamics of the so-called *PAVLOV* behavior in the iterated prisoner’s dilemma. Several results about the time of convergence of this particular dynamics towards the stable state can be found in [17], and [18], for rings, and complete graphs.

This is clearly not the only way to associate a dynamic to a game. There are several famous classical approaches: The first consists in repeating games: see for example [11,29]. The second corresponds to models from evolutionary game theory: refer to [22, 31] for a presentation of this latter approach. The approach here considers dynamics obtained by selecting at each step some players and let them play a fixed game. Alternatives to *PAVLOV* behavior could include *MYOPIC* dynamics (at each step each player chooses the best response to previously played strategy by its adversary), or the well-known and studied *FICTIOUS-PLAYER* dynamics (at each step each player chooses the best response to the statistics of the past history of strategies played by its adversary). We refer to [11, 19] for a presentation of results known about the properties of the obtained dynamics according to the properties of the underlying game. This is clearly non-exhaustive, and we refer to [9] for an incredible zoology of possible behaviors for the particular iterated prisoner's dilemma game, with discussions of their comparative merits in experimental tournaments.

Recently Jaggard *et al.* [23] studied a distributed model similar to protocol populations where the interactions between pairs of agents correspond to a game. Unlike in our model, there each agent has its own pay-off matrix and has some knowledge of the history. This work gives several non-convergence results.

Results We want to understand if restricting to rules that come from a symmetric game is a limitation, and in particular whether restricting to rules that can be termed *PAVLOV* in the spirit of [17] is a limitation. We show that directly restricting the definitions introduced in [12] indeed yields a strong limitation of the computational power of such protocols as they become unable to detect if three or more occurrences of a single input symbol are present in the population. However, a slight modification of the *PAVLOV* behaviour, forcing dissatisfied agents to change even if their current strategy is already the *Best Response* to their opponent's allows us to circumvent this limitation. We call this new behavior the *Exclusive PAVLOV Behavior*. We then show the solutions to several basic problems using rules of interactions associated to a symmetric game by the exclusive *PAVLOV* behavior, and discuss the power of such rules. We prove that they can count up to 2, they can compute *MAJORITY* and more generally that they can count to any threshold 2^k for any given k.

In Section 2, we briefly recall population protocols and we discuss the power of computation in population protocols using only the symmetric rules. In Section 3, we recall some basics from game theory. In Section 4, we discuss how a game can be turned into a dynamic, and introduce the notion of symmetric *Pavlovian* population protocols and give an impossibility result for symmetric Pavlovian population protocols. In Section 5 we introduce *exclusive Pavlovian* population protocols. In Section 6, we prove that any symmetric deterministic 2-state population protocol is exclusive Pavlovian, and that the problem of computing the OR of inputs, the AND of inputs and the majority problem admit exclusive Pavlovian solutions. In Section 7, we describe some exclusive Pavlovian population protocols that count up to any threshold of form 2^k for any given k.

2 POPULATION PROTOCOLS

2.1 Formal Model

A protocol [3, 8] is given by $(Q, \Sigma, \iota, \omega, \delta)$ with the following components. Q is a finite set of *states*. Σ is a finite set of *input symbols*. $\iota : \Sigma \to Q$ is the initial state mapping, and $\omega : Q \to \{0, 1\}$ is the individual output function. $\delta \subseteq Q^4$ is a joint transition relation that describes how pairs of agents can interact. Relation δ is sometimes described by listing all possible interactions using the notation $(q_1, q_2) \to (q_1', q_2')$, or even the notation $q_1 q_2 \to q_1' q_2'$, for $(q_1, q_2, q_1', q_2') \in \delta$ (with the convention that $(q_1, q_2) \to (q_1, q_2)$ when no rule is specified with (q_1, q_2) in the left-hand side). The protocol is said *deterministic* if for all pairs (q_1, q_2) there is only one pair (q_1', q_2') with $(q_1, q_2) \to (q_1', q_2')$. In that case, we write $\delta_1(q_1, q_2)$ for the unique q_1' and $\delta_2(q_1, q_2)$ for the unique q_2'.

Notice that, in general, rules can be non-symmetric: if $(q_1, q_2) \to (q_1', q_2')$, it does not necessarily follow that $(q_2, q_1) \to (q_2', q_1')$.

2.2 Computation

Computations of a protocol proceed in the following way. $n \geq 2$ *agents* take part in this computation. A *configuration* of the system is a multiset of elements of Q. Since the agents are anonymous, the agents with the same state are indistinguishable and thus we can see a configuration as a vector of integers of length $|Q|$ corresponding to counts of agents for each state.

To start the computation, we give to each agent an input, which is an element of Σ. To get the initial configuration, we just need to apply ι to each agent's input.

Two agents q_1 and q_2 *interact* together if they both change their state into q_1' and q_2', where these two new states are such as (q_1, q_2, q_1', q_2') is in δ. A *step* in the computation is a transition between two configurations C and C' (noted $C \to C'$), where two agents have interacted together. An *execution* for a protocol from an initial configuration C_0 is a sequence of configurations $(C_i)_{i \in \mathbb{N}}$, where, for each i in $\mathbb{N}$, $C_i \to C_{i+1}$.

An execution is fair if, for each configuration C appearing infinitely often, if one can reach C' from C in one step, then C' also appears an infinite number of times in the execution.

The output of a state of Q is its image through ω, hence either 0 or 1. The output of a configuration is 0 (respectively 1) if all the individual outputs are 0 (respectively 1). If the individual outputs are mixed 0s and 1s then the output of the configuration is undefined.

2.3 Computable Predicates

Let p be a predicate over multisets of elements of Σ. Predicate p can be considered as a function whose range is $\{0, 1\}$ and whose domain is the collection of these multisets. The predicate is said to be computed by the protocol if, for every multiset I, and every fair execution that starts from the initial configuration corresponding to I, the output value of every agent eventually stabilizes to $p(I)$.

Multisets of elements of Σ are in clear one-to-one correspondence with elements of $\mathbb{N}^{|\Sigma|}$: a multiset over Σ can be identified by a vector of $|\Sigma|$ components, where each component represents the multiplicity of the corresponding element of Σ in this multiset. It follows that predicates can also be considered as functions whose range is $\{0, 1\}$ and whose domain is $\mathbb{N}^{|\Sigma|}$.

The following was then proved in [3, 5].

Theorem 1 [3, 5]. *A predicate is computable in the population protocol model if and only if it is semilinear.*

Recall that semilinear sets are known to correspond to predicates on counts of input agents definable in first-order Presburger arithmetic [30].

We will use the following notation as in [1]: the set of all functions from a set X to a set Y is denoted by Y^X. Let Σ be a finite non-empty set. For all $f, g \in \mathbb{R}^\Sigma$, we define the usual vector space operations

$$
\begin{array}{llll}
(f+g)(\sigma) & = & f(\sigma)+g(\sigma) & \text{for all } \sigma \in \Sigma \\
(f-g)(\sigma) & = & f(\sigma)-g(\sigma) & \text{for all } \sigma \in \Sigma \\
(cf)(\sigma) & = & cf(\sigma) & \text{for all } \sigma \in \Sigma, c \in \mathbb{R} \\
(f.g)(\sigma) & = & \sum_\sigma f(\sigma)g(\sigma) & \text{for all } \sigma \in \Sigma.
\end{array}
$$

Abusing notation as in [1], we will write σ for the function $\sigma(\sigma') = [\sigma = \sigma']$, for all $\sigma' \in \Sigma$, where $[condition]$ is 1 if condition is true, 0 otherwise.

Corollary 1 [3, 5]. *All semilinear predicates can be computed by a deterministic population protocol.*

2.4 Symmetric Rules

A Population protocol is *symmetric* if, for all (q_1, q_1, q_1', q_2') in δ, $q_1' = q_2'$ and for all (q_1, q_2, q_1', q_2') in δ, (q_2, q_1, q_2', q_1') is in δ. The remainder of this section is devoted to proving that the predicates computable by the Symmetric population protocols are the same as those computable by the unrestricted population protocols.

First, we can notice that

Proposition 1. *Any population protocol can be simulated by a symmetric population protocol, as soon as the population is of size* ≥ 3*.*

Proof. The idea of the proof is to construct a symmetric Population Protocol simulating a given (asymmetric) population protocol by using two copies of the set of states from the original protocol and using them to determine whether an agent should be the first or the second agent in an asymmetric interaction. The key to the proof is to have agents alternate between both roles (that is between the two copies of the set of states) when they encounter another agent trying to play the same role in the interaction. Thus when two agents trying to act as first members of a transition interact, they both move to try to act as second members of a transition the next time they are picked to interact (without changing their state with relation to the first protocol), and vice-versa. Note that we assume all protocols to be deterministic here.

Let $\mathcal{P} = (Q, \Sigma, \iota, \omega, \delta)$ be some (asymmetric) Population Protocol. We will now construct a symmetric Population Protocol $\mathcal{P}' = (Q', \Sigma, \iota', \omega', \delta')$ such that any computation in $\mathcal{P}'$ can be mapped to a computation in $\mathcal{P}$ and, conversely, any computation in $\mathcal{P}$ for a population of size ≥ 3 can be simulated in $\mathcal{P}'$. We will then say that $\mathcal{P}'$ simulates P and it is clear that $\mathcal{P}'$ computes the same predicate as $\mathcal{P}$. This, of course implies that we use the same input alphabet Σ for both protocols.

We define $Q' = Q \times \{1, 2\}$ to have two copies of Q. For any state $q \in Q$ we call an agent in state $(q, 1)$ (resp. $(q, 2)$) a *first-minded* (resp. *second-minded*) agent in state q. We use natural extensions for ι and ω: for any symbol $\sigma \in \Sigma$, $\iota'(\sigma) = (\iota(\sigma), 1)$ and for any state $q \in Q$, we set $\omega'(q, 1) = \omega'(q, 2) = \omega(q)$. Finally, we define δ' from δ as follows. For any pair $(u, v) \in Q$, let $(u', v') = \delta(u, v)$. Then set:

- $\delta'((u, 1), (v, 1)) = ((u, 2), (v, 2))$ and $\delta'((u, 2), (v, 2)) = ((u, 1), (v, 1))$
- $\delta'((u, 1), (v, 2)) = ((u', 1), (v', 2))$ and $\delta'((v, 2), (u, 1)) = ((v', 2), (u', 1))$.

This defines $\mathcal{P}'$ as a symmetric protocol. To see that any valid computation in $\mathcal{P}'$ corresponds to a valid computation in $\mathcal{P}$, simply project the states of all agents according to their first coordinate (that is, disregard whether they are first- or second-minded). Any computation $C_1 \to \ldots \to C_k \to \ldots$ in $\mathcal{P}'$ becomes a valid computation in $\mathcal{P}$ with a few additional stagnations in the same configuration (when two identically-minded agents change from first- to second-minded or the other way around).

Conversely, consider a possible transition $C_1 \to C_2$ in $\mathcal{P}$ where C_1 and C_2 are two configurations of a population of size $n \geq 3$. Then for any configuration C_1' of the same population of agents but over states in Q', such that we have $C'_1(w) \in C(w) \times \{1, 2\}$ for any agent w in the population. (Recall that, for a configuration C and an agent w, $C(w)$ denotes the state of w in C.) That is, any configuration in which we only added a first- or second-minded state to each agent satisfies that there is a configuration C'_2 reachable from C'_1 such that, for any agent w, $C'_2(w) \in C_2(w) \times \{1, 2\}$. Indeed, let (u, v) be two agents such that $C_1 \overset{u,v}{\to} C_2$. We have $(C_2(u), C_2(v)) = \delta(C_1(u), C_1(v))$ and, by definition of δ',

$$\delta'((C_1(u), 1), (C_1(v), 2)) = ((C_2(u), 1), (C_2(v), 2)).$$

If $C'_1(u) = (C_1(u), 1)$ and $C'_1(v) = (C_1(v), 2)$, then by having agents u, v interact we directly get a satisfactory configuration C'_2. Otherwise, we will show that we can, using a third agent w (hence the condition that $n \geq 3$) change the mind of agents u and v to get to this case.

1. If u, v, w are all identically-minded, then having w interact with whoever is of the wrong mind (v if they are all first-minded, u otherwise) results in the desired configuration for u and v.
2. If u, v are identically-minded but w is differently-minded, then having u, v interact brings us back to case 1.
3. If u is second-minded and v is first-minded, then having w interact with whoever is like-minded brings us back to case 1.

This proves that any valid computation in $\mathcal{P}$ can be simulated by a valid configuration in $\mathcal{P}'$ (with the addition of at most two “mind changes” at each computation step). □

Corollary 2. *A predicate is computable by a symmetric population protocol if and only if it is semilinear.*

3 GAME THEORY

We now recall the simplest concepts from Game Theory. In this paper, we will restrict to symmetric games. In the presentation, we focus on symmetric non-cooperative games, with complete information, in normal form.

Definition 1 Symmetric two-player game. *A symmetric two-player game is a couple (S, A) in which both players are endowed with the same finite set of pure actions S. For each player, the payoff function is denoted by $A : S \times S \to \mathbb{R}$ where $A(i, j)$ denotes the payoff to the player choosing the first argument when his opponent chooses the second argument.*

In the remainder of this paper, the payoff function is viewed as the payoff matrix. With some abuse of notation, $A_{i,j}$ denotes the payoff to the player choosing the first argument when his opponent chooses the second argument.

Example 1 Prisoner's dilemma. *The* prisoner's dilemma *game is a couple* (S, A) *where* $S = \{C, D\}$ *and the matrix* A *is as follows :*

$$\begin{array}{ccc} & & \textit{Opponent} \\ & & \begin{array}{cc} C & \quad D \end{array} \\ \textit{Player} & \begin{array}{c} C \\ D \end{array} & \begin{array}{|cc|} \hline 1 & -3 \\ 3 & -1 \\ \hline \end{array} \end{array}$$

We will also introduce some game theory concepts: best response and Nash equilibrium.

Definition 2 Best reponse. *Let G be a symmetric two-player game (S, A).*

- *A strategy $x \in S$ is said to be a* best response *to strategy $y \in S$, denoted by $x \in BR(y)$ if for all strategies $z \in S$, we have*

$$A_{z,y} \leq A_{x,y} \tag{1}$$

- *A strategy $x \in S$ is said to be a best response to strategy y among those different from x', denoted by $x \in BR_{\neq x'}(y)$ if for all strategies $z \in S \backslash \{x'\}$, we have*

$$A_{z,y} \leq A_{x,y} \tag{2}$$

Given some strategy $x' \in Strat(I)$, a strategy $x \in Strat(I)$ for all strategies $z \in Strat(I)$, $z \neq x'$.

Example 2 Prisoner's dilemma. $BR(C) = BR(D) = D$, $BR_{\neq D}(D) = C$.

Definition 3 Pure Nash equilibrium. *Let G be a symmetric two-player game (S, A). A pair (x, y) is a* (pure) Nash equilibrium *if $x \in BR(y)$ and $y \in BR(x)$.*

In other words, two strategies (x, y) form a Nash equilibrium if in that state, no player has a unilateral interest to deviate from it. Note that a pure Nash equilibrium does not always exist.

Example 3 Prisoner's dilemma. *Since $BR(C) = D$ and $BR(D) = D$, (D, D) is the unique pure Nash equilibrium. In it, each player has score -1. The well-known paradox is that if they had played (C, C) (cooperation) they would have had score 1. The social optimum (C, C), is different from the equilibrium that is reached by rational players (D, D), since in any other state, each player fears that the adversary plays D.*

Repeating Games In this paper, we consider that the players play the same game again and again. After each game, they can decide to change or not their strategy. They do not have any memory about their previous games nor their past strategies.

Repeating k times the same game, is equivalent to extending the space of choices into S^k: player I (respectively II) chooses his or her action $x(t) \in S$, (resp. $y(t) \in S$)) at time t for $t = 1, 2, \cdots, k$. Hence, this is equivalent to a two-player game with n^k choices for players, where n is the cardinality of S. To avoid confusion, we will call *actions* the choices $x(t)$, $y(t)$ of each player at time t, and *strategies* the sequences $X = x(1), \cdots, x(k)$ and $Y = y(1), \cdots, y(k)$, that is to say the strategies for the global game.

Behaviors In practice, player I (respectively II) has to solve the following problem at each time t: given the history of the game up to now, that is to say $X_{t-1} = x(1), \cdots, x(t-1)$ and $Y_{t-1} = y(1), \cdots, y(t-1)$ what should player I (resp. II) play at time t? In other words, how to choose $x(t) \in S$? (resp. $y(t) \in S$?). It is natural to assume that this is given by some behavior rules:

$$x(t) = f(X_{t-1}, Y_{t-1}) \text{and}\, y(t) = g(X_{t-1}, Y_{t-1})$$

for some particular functions f and g.

The Specific Case of the Prisoner's Dilemma The question of the best behavior rule to use for the prisoner's dilemma gave birth to an important literature. In particular, after the book [9], that describes the results of tournaments of behavior rules for the iterated prisoner's dilemma, and that argues that there exists a best behavior rule called *TIT-FOR-TAT*. This consists in cooperating at the first step, and then do the same thing as the adversary at subsequent times.

Among possible behaviors there is *PAVLOV* behavior: in the iterated prisoner's dilemma, a player cooperates if and only if both players opted for the same alternative in the previous move. This name [9, 24, 28] stems from the fact that this strategy embodies an almost reflex-like response to the payoff:

it repeats its former move if it was rewarded by 1 or 3 points, but switches behavior if it was punished by receiving only -1 or -3 points. We refer to [28] for a study of this strategy in the spirit of Axelrod's tournaments.

The *PAVLOV* behavior can also be termed *WIN-STAY, LOSE-SHIFT* since if the play on the previous round results in a success, then the agent plays the same strategy on the next round. Alternatively, if the play resulted in a failure the agent switches to another action [9, 28].

Example 4 Prisoner's dilemma. *If players i and j play the prisoner's dilemma having a PAVLOV behavior, then it is easy to see that this corresponds to executing the following rules:*

$$\begin{cases} CC & \rightarrow & CC \\ CD & \rightarrow & DD \\ DC & \rightarrow & DD \\ DD & \rightarrow & CC. \end{cases} \tag{3}$$

PAVLOV behavior is Markovian: a behavior f is *Markovian*, if $f(X_{t-1}, Y_{t-1})$ depends only on $x(t-1)$ and $y(t-1)$. From such a behavior, it is easy to obtain a distributed dynamic. For example, let's follow [17], for the prisoner's dilemma.

There are several works on the prisoner's dilemma dynamic. In these studies, the interactions between players correspond to a connected graph $G = (V, E)$, with N vertices corresponding to players and with edges representing interaction. An instantaneous configuration of the system is given by an action of $\{C, D\}^N$, that is to say by the state C or D of each vertex. At each time t, one chooses randomly and uniformly one edge (i, j) of the graph. At this moment, players i and j play the prisoner's dilemma with the *PAVLOV* behavior. The goal of these work is to answer the following question: what is the final state reached by the system? Several results about the time of convergence towards this stable state can be found in [17], and [18], for rings, and complete graphs.

What is interesting in this example is that it shows how to go from a game, and a behavior to a distributed dynamic on a graph, and in particular to a population protocol when the graph is the complete graph.

4 FROM GAMES TO POPULATION PROTOCOLS

In the spirit of the previous discussion, to any symmetric game, we can associate a population protocol as follows.

Definition 4 Associating a Protocol to a Game. *Let (S, A) be a symmetric two-player game where S (resp. A) is a set of actions (resp. the payoff matrix). Let Δ be some threshold.*

The protocol associated to the game is a population protocol whose set of states is Q, where $Q = S$ is the set of strategies of the game, and whose transition rules δ are given as follows:

$$(q_1, q_2, q_1', q_2') \in \delta$$

where

- $q_1' = q_1$ *when* $A_{q_1,q_2} \geq \Delta$
- $q_1' \in BR(q_2)$ *when* $A_{q_1,q_2} < \Delta$

and

- $q_2' = q_2$ *when* $A_{q_2,q_1} \geq \Delta$
- $q_2' \in BR(q_1)$ *when* $A_{q_2,q_1} < \Delta$,

Remark 1. *By subtracting Δ from each entry of the payoff matrix A, we get a new game, to which the same population protocol is associated when the satisfaction threshold is considered to be* 0*. Therefore we can assume without loss of generality that $\Delta = 0$. We will do so from now on.*

This definition corresponds to the direct adaptation of Definition 1 from [12] to symmetric games. A population protocol obtained from a game as above must be *symmetric*. Indeed, whenever $(q_1, q_2, q_1', q_2') \in \delta$, one has $(q_2, q_1, q_2', q_1') \in \delta$.

Definition 5 Deterministic Pavlovian population protocol.. *A population protocol is said to be* deterministically *obtained from a game as per Definition 4 if the best responses of the game are assumed to be unique, that is, if for any strategy q_1, $BR(q_1)$ is reduced to a singleton. Indeed, the rules obtained from such a game are deterministic: for all q_1, q_2, there is a unique q_1' and a unique q_2' such that $(q_1, q_2, q_1', q_2') \in \delta$.*

Definition 6 symmetric Pavlovian Population Protocol. *A symmetric* Pavlovian *population protocol is a population protocol that can be obtained deterministically from a game as per Definition 4.*

Theorem 2. *There is no symmetric Pavlovian protocol that computes the threshold predicate $[x.\sigma \geq 3]$, which is true when there are at least* 3 *occurrences of input symbol σ in the input x.*

Proof. The proof is by contradiction. Assume that there exists such a symmetric Pavlovian protocol $\mathcal{P}$. Without loss of generality, we may assume that $\Sigma = \{0, \sigma\}$ is a subset of the set of states Q. Let A be the payoff matrix from a symmetric game associated with this protocol. In keeping with a previous remark, we may assume without loss of generality that $\Delta = 0$ is the gain threshold for the *PAVLOV* behaviour corresponding to $\mathcal{P}$. We will derive a contradiction by showing that $\mathcal{P}$ cannot possibly distinguish between the inputs $x_3 = \{\sigma, \sigma\}$ and $x_4 = \{\sigma, \sigma, \sigma, \sigma\}$.

Since the protocol is symmetric, for any $q \in Q$, the rule $qq \to q'q''$, is such that $q' = q''$, that is to say of the form $qq \to q'q'$. Let us consider the sequence of rules such that $\sigma\sigma \to q_1q_1 \to q_2q_2 \to \cdots \to q_kq_k \to \ldots$ where $\sigma, q_1q_2, q_3, \ldots, q_k \in Q$. Since Q is finite, there exist two distinct integers k and ℓ such that $q_k = q_\ell$ and $k < \ell$.

The case $k + 1 = \ell$ is not possible. Indeed, we would have the rule $q_kq_k \to q_kq_k$. Then, consider the inputs $x_3 = \{\sigma, \sigma\}$ and $x_4 = \{\sigma, \sigma, \sigma, \sigma\}$. x_4 must be accepted. From x_4 there is a derivation $x_4 \to \{q_1, q_1, \sigma, \sigma\} \to \{q_1, q_1, q_1q_1\} \to^* \{q_k, q_k, q_k, q_k\}$. This latter configuration is terminal from the above rule. Since x_4 must be accepted, we must have $\omega(q_k) = 1$. However, from x_3 there is a derivation $x_3 \to \{q_1, q_1\} \to^* \{q_k, q_k\}$, where the last configuration is also terminal. We reach a contradiction, since the output of this last configuration would be $\omega(q_k) = 1$, whereas x_3 must be rejected. Hence, $k + 1 < \ell$, and $q_kq_k \to q_{k+1}q_{k+1} \to \cdots \to q_\ell q_\ell = q_kq_k$. Let T be the set of states $T = \{q_i : k \le i \le \ell\}$. Since $q_iq_i \to q_{i+1}q_{i+1}$ is among the rules, by definition of Pavlovian behaviour, we have $q_{i+1} = BR(q_i)$.

Let us discuss the rules

$$q_iq_j \to q_i'q_j' \tag{4}$$

for $(q_i, q_j) \in T^2$. There are two possibilities for the value of q_i':

$$q_i' = \begin{cases} q_i' = q_i \text{ if } A_{q_iq_j} \ge 0 \\ q_i' = BR(q_j) = q_{j+1} \text{ otherwise.} \end{cases}$$

In any case, the value of q_i' is in T. Symmetrically, we have two possibilities for q_j', all of them in T. Hence, all rules of the form (4) preserve T: we have $q_i', q_j' \in T$, as soon as $q_i, q_j \in T$.

Similarly to what we did in the case $k + 1 = \ell$, there is a derivation

$$x_4 \to \{q_1, q_1, \sigma, \sigma\} \to \{q_1, q_1, q_1q_1\} \to^* \{q_k, q_k, q_k, q_k\}.$$

From the last configuration, by the previous remark, the state of all agents will be in T. Since x_4 must be accepted, ultimately all agents will be in states that belong to T and such that their image by ω is 1. Consider now x_3. There is a derivation

$$x_3 \to \{q_1, q_1\} \to^* \{q_k, q_k\}$$

that will go through all configurations $\{q_i q_i\}$, for all $q_i \in T$. This cannot eventually stabilize to elements whose image by ω is 0, because some of the elements of T have image 1 by ω, and hence x_3 is not accepted. This yields the desired contradiction. □

Remark 2. *We can also prove, using a similar proof, that there is no symmetric Pavlovian protocol that computes the threshold predicate* $[x.\sigma \geq 2k+1]$ *for* $k > 2$ *if we use two populations of respective size* $2k$ *and* $2k+1$.

Remark 3. *No similar proof seems possible for exclusive pavlovian population protocols (to be defined in next section).*

5 EXCLUSIVE PAVLOVIAN POPULATION PROTOCOLS

We denote by $BR_{\neq x'}(y)$ the set of best responses to strategy y, different from strategy x. Similarly to what was done in Definition 4, we can then define an *Exclusive Pavlovian Protocol* to be a Protocol obtained from a game by following the *Exclusive PAVLOV* behaviour instead of the traditional *PAVLOV* behaviour.

Definition 7 Exclusive Pavlovian Protocol. *Let* (S, A) *be a symmetric game and let* Δ *be some threshold. A protocol exclusively associated to the game is a population protocol whose set of states is* $Q = S$ *the set of strategies of the game, and whose transition rules* δ *are given by:* $(q_1, q_2, q_1', q_2') \in \delta$ *if and only if*

$$q_1' = \begin{cases} q_1 \text{ if } A_{q_1,q_2} \geq \Delta \\ x \in BR_{\neq q_1}(q_2) \text{ otherwise,} \end{cases}$$

$$q_2' = \begin{cases} q_2 \text{ if } B_{q_2,q_1} \geq \Delta \\ x \in BR_{\neq q_2}(q_1) \text{ otherwise.} \end{cases}$$

An exclusive Pavlovian protocol is a population protocol exclusively and deterministically associated to a symmetric game.

Note that Definition 7 only differs from Definition 4 by the use of exclusive Best Responses $BR_{\neq q}$ instead of BR. Again, we may assume without loss of generality that the satisfaction threshold Δ is 0 and will do so in the rest of this paper.

6 SOME SIMPLE EXCLUSIVE PAVLOVIAN PROTOCOLS

We now discuss the computational power of exclusive Pavlovian population protocols. We start with an easy consideration.

Theorem 3. *Any symmetric deterministic* 2*-state population protocol is exclusive Pavlovian.*

Proof. Consider a deterministic symmetric 2-state population protocol. Note $Q = \{+, -\}$ its set of states. Its transition function can be written as follows:

$$\left\{\begin{array}{lll} ++ & \to & \alpha_{++}\alpha_{++} \\ +- & \to & \alpha_{+-}\alpha_{-+} \\ -+ & \to & \alpha_{-+}\alpha_{+-} \\ -- & \to & \alpha_{--}\alpha_{--} \end{array}\right. \tag{5}$$

for some $\alpha_{++}, \alpha_{+-}, \alpha_{-+}, \alpha_{--}$.

This corresponds to the symmetric game given by the following pay-off matrix A

$$\begin{array}{ccc} & & \text{Opponent} \\ & & \begin{array}{cc} + & - \end{array} \\ \text{Player} & \begin{array}{c} + \\ - \end{array} & \boxed{\begin{array}{cc} \beta_{++} & \beta_{+-} \\ \beta_{-+} & \beta_{--} \end{array}} \end{array}$$

where for all $q_1, q_2 \in \{+, -\}$,

- $\beta_{q_1q_2} = 1$ if $\alpha_{q_1q_2} = q_1$,
- $\beta_{q_1q_2} = -1$ otherwise.

To prove this, we just need to notice that :

- if $\alpha_{q_1q_2} = q_1$, the first agent keeps its state. With $\beta_{q_1q_2} = 1$, $A_{q_1,q_2} \geq 0$, so the first agent will keep its state.

- if $\alpha_{q_1 q_2} = q_2$, the first agent switches its state. With $\beta_{q_1 q_2} = -1$, $A_{q_1,q_2} < 0$, so the first agent will chose $BR_{\neq q_1}(q_2)$. Because there are only 2 states, the agent's new state with our exclusive Pavlovian protocol will be the same that the one with the original symmetric protocol.

The proof is enough for the second agent because the protocols are symmetric.

□

Unfortunately, not all rules correspond to a game.

Proposition 2. *Some symmetric population protocols are not Pavlovian.*

Proof. Consider for example a deterministic 3-state population protocol with set of states $Q = \{q_0, q_1, q_2\}$ and a joint transition function δ such that $\delta_1(q_0, q_0) = q_1$, $\delta_1(q_1, q_0) = q_2$, $\delta_1(q_2, q_0) = q_0$.

Assume by contradiction that there exists a 2-player game corresponding to this 3-state population protocol. Consider its payoff matrix A. Let $A_{q_0,q_0} = \beta_0$, $A_{q_1,q_0} = \beta_1$, $A_{q_2,q_0} = \beta_2$. We must have $\beta_0, \beta_1, \beta_2 < 0$ since all agents that interact with an agent in state q_0 must change their state. Now, since q_0 changes to q_1, q_1 must be a strictly better response to q_0 than q_2: hence, we must have $\beta_1 > \beta_2$. In a similar way, since q_1 changes to q_2, we must have $\beta_2 > \beta_0$, and since q_2 changes to q_0, we must have $\beta_0 > \beta_1$. From $\beta_1 > \beta_2 > \beta_0$ we reach a contradiction. □

This indeed motivates the following study, where we discuss which problems admit a Pavlovian solution.

6.1 Basic Protocols

Proposition 3. *There is an exclusive Pavlovian protocol that computes the logical OR (resp. AND) of input bits.*

Proof. Consider the following protocol to compute OR,

$$\begin{cases} 01 & \rightarrow & 11 \\ 10 & \rightarrow & 11 \\ 00 & \rightarrow & 00 \\ 11 & \rightarrow & 11 \end{cases} \tag{6}$$

and the following protocol to compute AND,

$$\begin{cases} 01 & \rightarrow & 00 \\ 10 & \rightarrow & 00 \\ 00 & \rightarrow & 00 \\ 11 & \rightarrow & 11 \end{cases} \tag{7}$$

Since they are both deterministic 2-state population protocols, they are Pavlovian. □

Remark 4. *Notice that OR (respectively AND) protocol corresponds to the predicate $[x.1 \geq 1]$ (resp. $[x.0 = 0]$), where x is the input. A simple change of notation yields a protocol to compute $[x.\sigma \geq 1]$ and $[x.\sigma = 0]$ for any input symbol σ.*

Remark 5. *All previous protocols are "naturally broadcasting" i.e., eventually all agents agree on some (the correct) value. With previous definitions (which are the classical ones for population protocols), the following protocol does not compute the XOR or input bits, or equivalently does not compute predicate $[x.1 \equiv 1 \ (mod\ 2)]$.*

$$\begin{cases} 01 & \rightarrow & 01 \\ 10 & \rightarrow & 10 \\ 00 & \rightarrow & 00 \\ 11 & \rightarrow & 00 \end{cases} \tag{8}$$

With this protocol, the answer is not eventually known by all the agents. Even if this protocol might look good, you can notice that if $[x.1 \equiv 0 \ (mod\ 2)]$, all agents finish in the state 0, so $\omega(0) = 0$. If $[x.1 \equiv 1 \ (mod\ 2)]$, at some point, then all agents but one will be in state 0 (the last one being in state 1). Even if $\omega(1) = 1$, this configuration does not have an output.

It computes the XOR in a weaker form i.e., eventually, all agents will be in state 0, if the XOR of input bits is 0, or eventually only one agent will be in state 1, if the XOR of input bits is 1.

Proposition 4. *There is an exclusive Pavlovian protocol that computes the threshold predicate $[x.\sigma \geq 2]$, which is true when there are at least 2 occurrences of input symbol σ in the input x.*

Proof. The following protocol is a solution taking

- $\Sigma = \{0, \sigma\}$, $Q = \{0, \sigma, 2\}$,
- $\omega(0) = \omega(\sigma) = 0$,
- $\omega(2) = 1$.

$$\begin{cases} 00 \to & 00 \\ 0\sigma \to & 0\sigma \\ \sigma 0 \to & \sigma 0 \\ 02 \to & 22 \\ 20 \to & 22 \\ \sigma\sigma \to & 22 \\ \sigma 2 \to & 22 \\ 2\sigma \to & 22 \\ 22 \to & 22 \end{cases} \tag{9}$$

Indeed, if there are at least two σ, then by fairness and by the rule $\sigma\sigma \to 22$, then they will ultimately be changed into two 2s. Then 2s will turn all other agents into 2s. Now, this is the only way to create a 2.

This is a Pavlovian protocol since it corresponds to the following payoff matrix.

$$\begin{array}{cc} & \text{Opponent} \\ & \begin{array}{ccc} 0 & \sigma & 2 \end{array} \\ \text{Player} \begin{array}{c} 0 \\ \sigma \\ 2 \end{array} & \boxed{\begin{array}{ccc} 0 & 0 & -1 \\ 0 & -1 & -1 \\ 1 & 1 & 1 \end{array}} \end{array}$$

□

7 SOME NOT SO SIMPLE PAVLOVIAN PROTOCOLS

7.1 Some Structural Properties on Pavlovian Rules

We are now going to describe some not so simple Pavlovian protocols. Before doing so, and in order to help to prove that a given set of rules is Pavlovian, without building explicitly possibly intricate matrices, we start with some structural properties on Pavlovian protocols.

Proposition 5. *Consider a set of rules. For all rules $ax \to a'x'$, we denote $\delta_a(x) = x'$ and $\delta_x(a) = a'$.*

Let $Stable(a) = \{x \in Q | \delta_a(x) = x\}$.

Then the set of rules is deterministic Pavlovian iff there exists a function $max : Q \to Q$ such that $\forall a \in Q$:

- *$Stable(a) \neq \emptyset$ implies that $max(a) \in Stable(a)$*
- *$\forall b \notin Stable(a), b \neq max(a)$ implies $\delta_a(b) = max(a)$.*

Proof. First, we consider a Pavlovian population protocol P obtained from corresponding matrix A. For each state a in Q, let q be the best response to strategy a for matrix A (i.e $q = BR(a)$). We set $max(a) = q$, i.e. max corresponds to BR.

If $Stable(a)$ is not empty, then there exists some b such that $ab \to a''b$. By Definition 4, we have $M_{b,a} \geq 0$. $M_{BR(a),a} \geq M_{b,a} \geq 0$, so $BR(a) = max(a) \in Stable(a)$.

Let $b \notin Stable(a), b \neq max(a)$. We have $A_{b,a} < 0$ (otherwise, $b \in Stable(a)$). We focus on the rule $ab \to a''b'$. $A_{b,a} < 0$ implies $b' \neq b$. We have $b' = BR_{\neq b}(a)$. Because $b \neq max(a)$, $b \neq BR(a)$, so $BR_{\neq b}(a) = BR(a)$. We deduce that $\delta_a(b) = max(a)$.

The function max with the restrictions exists.

Conversely, consider a population protocol P satisfying the properties of the proposition. All rules $ab \to a'b'$ are such that $\delta_a(b) = b'$ and $\delta_b(a) = a'$. We focus on the construction of a two-player game having the corresponding matrix A.

- If $Stable(a) = Q$, then $\forall x \in Q$, $A_{x,a} = 0$.
- If $Stable(a) = \emptyset$, then we have $A_{max(a),a} = -1$ and $A_{\delta_a(max(a)),a} = -2$. Moreover, $A_{x,a} = -3$ for each state x except $max(a)$,$\delta_a(max(a))$.
- If $\emptyset \subset Stable(a) \subset Q$ then the value $A_{x,a}$ of each element x depends on the set $Stable(a)$ and $max(a)$.
  - $A_{max(a),a} = 1$.
  - If $x \in Stable(a)$ and if $x \neq max(a)$, then $A_{x,a} = 0$.
  - If $x \notin Stable(a)$, then $A_{x,a} = -1$.

  It is easy to see that this game describes all rules of P. So, P is a Pavlovian population Protocol. □

Remark 6. *As a consequence, a protocol can be given by describing for any state a:*

1. *the set $Stable(a)$,*
2. *the value of $max(a)$,*
3. *and whenever $Stable(a) = \emptyset$, the value of $\delta_a(max(a))$.*

Note that if $Stable(a)$ is not $\emptyset$, then $\delta_a(max(a)) = max(a)$ otherwise $\delta_a(max(a)) \neq max(a)$.

7.2 Counting up to 3

We provide a logical characterization of rules that correspond to Pavlovian protocols.

Proposition 6. *There is an exclusive Pavlovian protocol that computes the threshold predicate* $[x.\sigma \geq 3]$*, which is true when there are at least* 3 *occurrences of input symbol* σ *in the input* x*.*

Proof. We give a population protocol that computes the threshold predicate $[x.\sigma \geq 3]$:

- $Q = \{0, 1, 2_+, 2_-, X, Y, \top\}$.
- $\Sigma = \{x, y\}$.
- $\iota(x) = 1$, $\iota(y) = 0$.
- $\omega = 1_{\{\top\}}$.
- with interaction rules defined as follow :

$$\begin{array}{cccc}
00 \to 00 & 11 \to 2_+2_+ & 2_+2_+ \to 2_-2_- & 2_-2_- \to 2_+2_+ \\
01 \to 01 & 12_+ \to 2_-2_+ & 2_+2_- \to YX & 2_-X \to \top 2_+ \\
02_+ \to 2_-0 & 12_- \to 2_+2_- & 2_+X \to \top 2_- & 2_-Y \to \top 2_+ \\
02_- \to 2_+0 & 1X \to \top X & 2_+Y \to \top 2_- & 2_-\top \to 2_-2_+ \\
0X \to 0X & 1Y \to \top Y & 2_+\top \to 2_+2_- & \\
0Y \to 0Y & 1\top \to \top\top & & \\
0\top \to \top\top & & &
\end{array}$$

$$\begin{array}{ccc}
XX \to \top\top & YY \to \top\top & \top\top \to \top\top \\
XY \to XY & Y\top \to \top\top & \\
X\top \to \top\top & &
\end{array}$$

We now show that the protocol is Pavlovian. Therefore, we will prove that it computes the threshold predicate $[x.\sigma \geq 3]$.

Now, this protocol is Pavlovian as it corresponds to the following payoff matrix :

		Opponent						
		0	1	2_+	2_-	X	Y	$\top$
Player	0	1	0	-3	-3	0	0	-1
	1	0	-1	-3	-3	-1	-1	-1
	2_+	-1	1	-3	-1	-1	-1	0
	2_-	-1	0	-1	-3	-1	-1	0
	X	0	0	-2	-3	-1	0	-1
	Y	0	0	-3	-2	0	-1	-1
	$\top$	0	0	-3	-3	1	1	1

As said in the previous remark, it can be described by the parameters $Stable(a)$, $max(a)$, $\delta_a(max(a))$ for each state a:

a	$Stable(a)$	$max(a)$	$\delta_a(max(a))$
0	$\{0, 1, X, Y, \top\}$	0	0
1	$\{0, 2_+, 2-, X, Y, \top\}$	2_+	2_+
2_+	$\emptyset$	2_-	X
2_-	$\emptyset$	2_+	Y
X	$\{0, Y, \top\}$	$\top$	$\top$
Y	$\{0, X, \top\}$	$\top$	$\top$
$\top$	$\{2_+, 2_-, \top\}$	$\top$	$\top$

Now, we prove that this Pavlovian protocol computes the threshold predicate $[x.\sigma \geq 3]$ using the number of occurrences of input symbol σ.

If there is no occurrence of input symbol σ in the input x, then the starting configuration is $(0\ldots0)$. In this case, no interaction allows agents to change their state. This argument can be also applied for the case where there is one occurrence of input symbol σ in the input x: the starting configuration is $(1)(0\ldots0)$.

Now we consider the case where there are two occurrences of input symbol σ in the input x. If the number of agents is 2, then, after the first interaction, the system switches between two configurations (2_+2_+) and (2_-2_-). Otherwise, i.e. if there are three or more agents in the population, then Figure 7.2 shows the graph of possible configurations (where $\overline{0}$ designates an arbitrary number of agents in state 0) and the possibility to switch from one configuration to another. All possible interactions in any of these configurations not shown in Figure 7.2 leave the configuration unchanged. Thus, in any reachable configuration from the initial $11\overline{0}$ configuration, all agents agree on output 0.

Finally, we consider the case where there are at least three occurrences of input symbol σ in the input x. First of all, the number of the agents being in state 0 never increases, so there are always, in any reachable configuration, at least three elements in $\{1, 2_+, 2_-, X, Y, \top\}$. Additionally, no agent can go to state 1 from any other given state. The principle of the proof of correctness is as follows: we will prove that from any configuration with at least three agents with states in $\{1, 2_+, 2_-, X, Y, \top\}$, there is a sequence of interactions increasing the number of agents in state $\top$ by at least one. This in turn means that, by iterating on such sequences, we can construct a sequence which leads us to a configuration where all agents are in state $\top$, which is a stable configuration in which everyone agrees on the correct output. The fairness hypothesis then ensures that any valid computation eventually leads to this configuration which will conclude the proof.

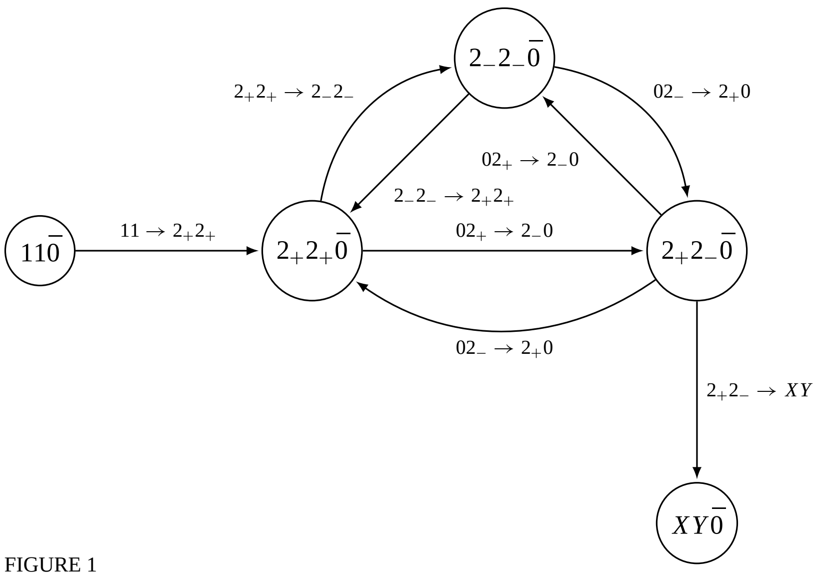

FIGURE 1
Graph of configurations for two occurrences of symbol σ in the input.

So, let us prove that from any configuration with at least three agents with states in $\{1, 2_+, 2_-, X, Y, \top\}$, there exists a series of interaction between those three agents which will increase the number of agents in state $\top$ by one. First let us consider the case where three agents are in states in $\{1, 2_+, 2_-, X, Y\}$. We will show that we can turn one of these agents to state $\top$. If two of these agents are in state 1, the rule $11 \to 2_+2_+$ guarantees that we can have at least two agents with states in $\{2_+, 2_-, X, Y\}$ which we will assume from now on.

If three agents are in states X or Y, then at least two are in the same state. Thus, by transition rule $XX \to \top\top$ or $YY \to \top\top$, a $\top$ can appear. If this is not the case, then at least one agent has a state s in $\{1, 2_+, 2_-\}$. If there is another agent in state $s' \in \{X, Y\}$, the interaction between s and s' creates a $\top$.

The final case is when all agents have states in $\{1, 2_+, 2_-\}$ (recall that at least two are in states 2_+ or 2_-). Then the remaining possible configurations, and their derivations are as follow where $i \in \{+, -\}$ (the underlined pairs interact):

- $\underline{12_i}2_i \to \underline{2_{\bar{i}}2_i}2_i \to X\underline{Y2_i} \to X\top 2_{\bar{i}}$
- $1\underline{2_+2_-} \to \underline{1X}Y \to \top XY$
- $\underline{2_i2_i}2_i \to 2_{\bar{i}}\underline{2_{\bar{i}}2_i} \to \underline{2_{\bar{i}}X}Y \to \top 2_i Y$
- $2_i\underline{2_i2_{\bar{i}}} \to \underline{2_iX}Y \to \top 2_{\bar{i}}Y$

In any case, we can bring one of the three agents to state $\top$. Thus if there are at least three agents in states $\{1, 2_+, 2_-, X, Y\}$, then we can increase the number of agents in state $\top$ by at least one.

If there are at most two agents in states $\{1, 2_+, 2_-, X, Y\}$, then there is necessarily at least one agent in state $\top$, because at least three agents are in non-0 states. If one of the agents in states $\{1, 2_+, 2_-, X, Y\}$ is actually in state 1 (resp. X and Y) then, by pairing it with the agent in state $\top$, it will be converted to $\top$. This again increases the number of agents in state $\top$ by one.

Otherwise, if exactly two agents are in states $\{2_+, 2_-\}$ and at least one agent is in state $\top$, then one of the following sequences of interactions will increase the number of agents in state $\top$ by at least one.

- $2_+\underline{2_+\top} \rightarrow 2_+\underline{2_+2_-} \rightarrow \underline{2_+Y}X \rightarrow \top\underline{2_-X} \rightarrow \top\top 2_+$
- $\underline{2_+2_-}\top \rightarrow X\underline{Y\top} \rightarrow \underline{X\top}\top \rightarrow \top\top\top$
- $2_-\underline{2_-\top} \rightarrow 2_-\underline{2_-2_+} \rightarrow \underline{2_-X}Y \rightarrow \top\underline{2_+Y} \rightarrow \top\top 2_-$.

If only one agent is in state 2_+ (resp. 2_-) and every other agent is either in state $\top$ or in state 0, then there are at least two agents in state $\top$. In this case, the following sequence of interactions will increase the number of agents in state $\top$ by one (the other case is symmetric).

- $\underline{2_+\top}\top \rightarrow \underline{2_+2_-}\top \rightarrow\rightarrow Y\underline{X\top} \rightarrow \underline{Y\top}\top \rightarrow \top\top\top$

Finally, if no agents are in states $\{1, 2_+, 2_-, X, Y\}$, then every agent is in state 0 or $\top$ with at least three of them being in state $\top$. The rule $0\top \rightarrow \top\top$ then allows us to convert the 0s to $\top$.

Thus, from any configuration with at least three non-0 agents, the number of agents in state $\top$ can be strictly increased which, according to what we said earlier concludes our proof. □

7.3 Majority

Proposition 7. *The majority problem (given some population of input symbols σ and σ', determine whether there are more σ than σ', i.e. $[x.\sigma \geq x.\sigma']$) can be solved by an exclusive Pavlovian population protocol.*

Proof. We claim that the following protocol outputs 1 if there are more σ than σ' in the initial configuration and 0 otherwise,

$$\begin{cases} NY & \rightarrow & YY \\ YN & \rightarrow & YY \\ N\sigma & \rightarrow & Y\sigma \\ \sigma N & \rightarrow & \sigma Y \\ Y\sigma' & \rightarrow & N\sigma' \\ \sigma' Y & \rightarrow & \sigma' N \\ \sigma\sigma' & \rightarrow & NY \\ \sigma'\sigma & \rightarrow & YN \end{cases} \tag{10}$$

taking

- $\Sigma = \{\sigma, \sigma'\}$, $Q = \{\sigma, \sigma', Y, N\}$,
- $\omega(\sigma) = \omega(Y) = 1$,
- $\omega(\sigma') = \omega(N) = 0$.

In this protocol, the states Y and N are "neutral" elements for our predicate but they should be understood as *Yes* and *No*. They are the "answers" to the question: are there more 0s than 1s.

This protocol is made such that the numbers of σ and σ' are preserved except when a σ meets a σ'. In that latter case, the two agents are deleted and transformed into a Y and a N.

If there are initially strictly more σ than σ', from the fairness condition, each σ' will be paired with a σ and at some point no σ' will left. By fairness and since there is still at least a σ, a configuration containing only σ and Ys will be reached. Since in such a configuration, no rule can modify the state of any agent, and since the output is defined and equals to 1 in such a configuration, the protocol is correct in this case

By symmetry, one can show that the protocol outputs 0 if there are initially strictly more σ' than σ.

Assume now that initially, there are exactly the same number of σ and σ'. By fairness, there exists a step when no more agents in the state σ or σ' left. Note that at the moment where the last σ is matched with the last σ', a Y is created. Since this Y can be "broadcasted" over the Ns, in the final configuration all agents are in the state Y and thus the output is correct.

This protocol is Pavlovian, since it corresponds to the following payoff matrix.

		Opponent N	Opponent Y	Opponent σ	Opponent σ'
Player	N	1	−1	−1	1
Player	Y	0	1	1	−1
Player	σ	0	0	0	−1
Player	σ'	0	0	−1	0

□

7.4 Counting up to 2^k

This section is devoted to proving the following theorem.

Theorem 4. *There is an exclusive Pavlovian protocol that computes the threshold predicate* $[x.\sigma \geq 2^k]$, *with the integer* $k \in \mathbb{N}$ *which is true when there are at least* 2^k *occurrences of input symbol* σ *in the input* x.

The case where $k = 0$ corresponds to the OR protocol in Proposition 3, and the case where $k = 1$ has been treated in Proposition 4. We now prove that the following protocol $\mathcal{P}$ is a solution for $k \geq 2$.

- $Q = \{0, \top\} \bigcup_{i=1}^{2^k-1} \{i_+, i_-\}$.
- $\Sigma = \{0, \sigma\}$.
- $\iota(\sigma) = 1_+$, $\iota(0) = 0$.
- $\omega = 1_{\{\top\}}$.
- Its transition function can be written as follows:

$$\begin{array}{ccc}
00 \rightarrow 00 & n_-m_+ \rightarrow m_-n_+ & \\
0n_+ \rightarrow n_-0 & n_+m_- \rightarrow m_+n_- & \text{whenever } m \neq n \\
0n_- \rightarrow n_+0 & n_-m_- \rightarrow m_+n_+ & \\
0\top \rightarrow \top\top & n_+m_+ \rightarrow m_-n_- &
\end{array}$$

$$\begin{array}{cc}
n_+n_+ \rightarrow n_-n_- & n_-n_+ \rightarrow (2n)_+(2n+1)_+ \text{ whenever } n < 2^{k-1} \\
n_-n_- \rightarrow n_+n_+ & n_-n_+ \rightarrow \top\top \text{ whenever } 2^{k-1} \leq n < 2^k \\
n_-\top \rightarrow n_-n_+ & \top\top \rightarrow \top\top \\
n_+\top \rightarrow n_+n_- &
\end{array}$$

First, we will prove that this protocol is is Pavlovian because we can define $Stable(q)$, $m(q)$ and $\delta_q(m(q))$ for any state q as follows:

$$\begin{array}{ccc}
Stable(0) = \{0, \top\} & Stable(\top) = Q \setminus \{0\} & Stable(n_+) = \emptyset \\
max(0) = 0 & max(\top) = \top & max(n_+) = n_- \\
 & & \delta_{n_+}(n_-) = (2n+1)_+ \\
Stable(n_-) = \emptyset & & \\
max(n_-) = n_+ & & \\
\delta_{n_-}(n_+) = (2n)_+ & &
\end{array}$$

Second, in order to simplify the proof, we transform protocol $\mathcal{P}$ into $\mathcal{P}'$. Protocol $\mathcal{P}'$ differs from $\mathcal{P}$ only in its transition function given by:

$$\begin{array}{cc}
00 \rightarrow 00 & n_-m_+ \rightarrow n_+m_- \\
0n_+ \rightarrow 0n_- & n_+m_- \rightarrow n_-m_+
\end{array}$$

$$
\begin{array}{ccc}
0n_- \to 0n_+ & n_-m_- \to n_+m_+ & \text{whenever } m \neq n \\
0\top \to \top\top & n_+m_+ \to n_-m_- &
\end{array}
$$

$$
\begin{array}{cc}
n_+n_+ \to n_-n_- & n_-n_+ \to (2n)_+(2n+1)_+ \text{ whenever } n < 2^{k-1} \\
n_-n_- \to n_+n_+ & n_-n_+ \to \top\top \text{ whenever } 2^{k-1} \leq n < 2^k \\
n_-\top \to n_-n_+ & \\
n_+\top \to n_+n_- & \top\top \to \top\top.
\end{array}
$$

The transition function of $\mathcal{P}'$ is the transition function of $\mathcal{P}$ in which transition rules of form $ab \to a'b'$ have sometimes been replaced by corresponding (and computationally equivalent) rule $ab \to b'a'$. The anonymity of agents in a population protocol implies that from a population protocol point of view, protocols $\mathcal{P}$ and $\mathcal{P}'$ are equivalent and compute the exact same predicate (if any). The difference is that $\mathcal{P}'$ is not Pavlovian.

Thus, proving that $\mathcal{P}'$ computes the desired predicate will yield that the same holds for $\mathcal{P}$. We will now study $\mathcal{P}'$ instead of $\mathcal{P}$.

Let a be an arbitrary agent of the population. Let q and C be respectively a state in Q and a configuration. We introduce several notations :

- $C(a)$ is the state of agent a in configuration C
- $C^{\#}(q)$ is the number of agents in state q in configuration C.
- $v(q)$ is the level of a state q corresponding to the integer number defined for $n \in \{1...2^k\}$ by $v(0) = 0$, $v(n_+) = v(n_-) = n$, $v(\top) = 2^k$.

Lemma 1. *For any two configurations C and C' such that $C \to C'$ in protocol P', $v(C'(a)) \geq v(C(a))$ or $v(C(a)) = 2^k$.*

Proof. To prove the lemma, it is sufficient to check each rule. □

The previous lemma means that the level of each agent can not decrease during an execution as long as no $\top$ has appeared. Note that this lemma does not hold for $\mathcal{P}$ and will allow us to simplify the following proofs.

Lemma 2. *Let $C_0, ..., C_{i_1}$ be an execution of configurations such that $\forall i, 0 \leq i \leq i_1, C_i \to C_{i+1}$, $C_0^{\#}(\top) = ... = C_{i_1}^{\#}(\top) = 0$ and $\forall q \notin \{0, 1_+\}, C_0^{\#}(q) = 0$. Then $\forall n, 1 \leq n \leq 2^{k-1}, \forall i \leq i_1, C_i^{\#}(n_+) + C_i^{\#}(n_-) \leq C_0^{\#}(1_+)2^{-\lfloor log(n) \rfloor}$.*

Proof. We prove the following statement by induction on n: in the execution of configurations $C_0, \ldots, C_{i_1}$, at most $C_0^{\#}(1_+)2^{-\lfloor log(n) \rfloor}$ agents may ever reach states n_+ or n_-.

Let $S(n)$, be the set of agents that ever reach states n_+ or n_- in this computation. We will prove that Card(S(n)) $\leq C_0^\#(1_+)2^{-\lfloor log(n) \rfloor}$.

First, from the assumptions of this lemma, in this execution, no agent can be in state $\top$. From Lemma 1, an agent can only increase the value of its level in the execution $C_0, ..., C_{i_1}$. In addition, agents initially in state 0 can not change their state. Thus, the set of agents having state different from 0 is the same in every configuration in the execution. It follows naturally that at most $C_0^\#(1_+)$ agents may ever reach states 1_+ or 1_-.

Second, to prove the statement for a given $n \geq 2$, we assume by induction that it is true for all $k < n$. From the transition rules, we can deduce that $S(k) \subseteq S(\lfloor \frac{k}{2} \rfloor)$ by looking at the state an agent was before it ever reached level k. This also holds for level n. In fact, every agent in $S(n)$ first reaches level n through an interaction of type $\frac{n}{2}_+ \frac{n}{2}_- \to n_+(n+1)_+$ if n is even (and $\frac{n-1}{2}_+ \frac{n-1}{2}_- \to (n-1)_+ n_+$ if n is odd). Thus, it appears that at most half the agents in $S(\lfloor \frac{n}{2} \rfloor)$ ever reach level n (the other half either being sent to level $n-1$ or $n+1$ depending on the parity of n or never going beyond level $\lfloor \frac{n}{2} \rfloor$).

Therefore $|S(n)| \leq \frac{|S(\frac{n}{2})|}{2} \leq C_0^\#(1_+)2^{-\lfloor log(n) \rfloor}$. □

Now, we will focus on the state $\top$: we discuss when this state appears in a execution according to the initial configuration.

Lemma 3. *Let $C_0, C_1, \ldots, C_i, \ldots$ be a correct execution of protocol P'.*
If $C_0^\#(1_+) < 2^k$ then $\forall i \geq 0, C_i^\#(\top) = 0$.

Proof. We prove the lemma by contradiction. We assume that $C_0^\#(1_+) < 2^k$ and that there exists at least one configuration C_i with at least one agent in state $\top$. Let us consider C_{i_0}, the earliest such configuration. This means that the transition $C_{i_0-1} \to C_{i_0}$ happens through an encounter of two agents n_+n_- with $n \in [2^{k-1}...2^k - 1]$.

Then $C_0, \ldots, C_{i_0-1}$ is a non-empty execution fitting the conditions of Lemma 1. Thus, $C_{i_0-1}^\#(n_+) + C_{i_0-1}^\#(n_-) \leq 2^{-\lfloor log(n) \rfloor} C_0^\#(1_+)$. Since $C_{i_0-1}^\#(n_+) \geq 1$ and $C_{i_0-1}^\#(n_-) \geq 1$, we have $C_0^\#(1_+) \geq 2^{\lfloor log(n) \rfloor} \geq 2^k$. This implies a contradiction with the assumption $C_0^\#(1_+) < 2^k$. □

We have now proved that if strictly less than 2^k agents are in state 1_+ initially, no agent will ever reach state $\top$ and thus all agents will always agree on output 0 and the computation will be correct. We will now prove that the computation will be correct if at least 2^k agents are initially in state 1_+.

Lemma 4. *For any configuration C in which at least 2^k agents are in non-zero states, there exists a configuration C' such that $C \to^* C'$ and $C'^\#(\top) \geq 1$.*

Proof. If $C^{\#}(\top) \geq 1$, then C' is C. Now, we assume $C^{\#}(\top) = 0$ and the pigeonhole principle insures that there are at least 2 agents at the same level. Let k be the smallest level with at least 2 agents. We will now prove that there is a finite sequence of configurations forming valid computation steps that increases the value of k. We shall now differentiate the cases where $C^{\#}(k_+) + C^{\#}(k_-) > 2$ and $C^{\#}(k_+) + C^{\#}(k_-) = 2$.

If $C^{\#}(k_+) + C^{\#}(k_-) > 2$, there is always a way to have a configuration having at least one agent in state k_+ and one agent in state k_- : Either 2 of them have different states, either all these agents have the same set and we apply $k_+k_+ \to k_-k_-$ to two of them (if there are all in state k_+, otherwise we apply $k_-k_- \to k_+k_+$). Now we can perform the interaction $k_+k_- \to 2k_+(2k+1)_+$ to diminish the number of agents in level k by two (and not create agents in lower levels). By iterating this process, we can bring the number of agents at level k to two or less. If only one remains, then we have achieved our goal, otherwise, we handle the two remaining agents as follows.

If $C^{\#}(k_+) + C^{\#}(k_-) = 2$, we again, differentiate: if $C^{\#}(k_+) = C^{\#}(k_-) = 1$ then selecting the two agents in states k_+ and k_- and perform the interaction $k_+k_- \to 2k_+(2k+1)_-$ will yield the desired result. Otherwise, if both agents at level k are in the same state, then we can break this symmetry by having one interact with any other non-zero agent to come back to the previous case. Such a non-zero and non-level-k agent exists since there are at least 2^k non zero agents.

Thus, from configuration C we have an execution $C \to^* C_1$ where $C_1^{\#}(k_-) + C_1^{\#}(k_+) < 2$. If $C_1^{\#}(\top) > 0$ we have achieved our desired result, otherwise, we can reiterate on C_1, knowing that the minimal level with at least two agents in C_1 is $k_1 > k$ by construction. Iterating this process gives us an execution $C \to^* C_1 \to^* \ldots \to^* C_j$ such that either an agent in state $\top$ appears or we have a corresponding execution $k < k_1 < \ldots < k_j$ of minimal level with at least two agents. Since this strictly growing execution is upper bounded by 2^k it is finite which guarantees that after a certain number of iterations, at least one agent will reach state $\top$. □

Definition 8. *For any configuration C with at least one agent in a non-zero state, we define $m(C)$ the smallest level with non-zero count in C.*

Note that $m(C)$ can never decrease: no interaction rule can create agents in a level lower than those already existing in the system.

Lemma 5. *For any configuration C such that $C^{\#}(\top) \geq 1$, and $m(C) < 2^k$, there exists a configuration C' such that $C \to {}^*C'$ and $m(C') > m(C)$.*

Proof. Similarly to what was done before, if there are at least two agents at level $m(C)$ then we can reduce the number of agents at level $m(C)$ by two and, iterating the process bring it to at most 1. Note that this process can be done by preserving the existence of agents in state $\top$. If we are left with no agents in state $m(C)$, we have achieved our goal. If not, we are left with a single agent a in state $m(C)$ and all other agents in states greater than $m(C)$, including at least one agent a' in state $\top$. Assume that a is in state k_+ (with, $k = m(C)$), the case k_- being symmetric). Then we can eliminate our final agent a through two interactions with a' :

$$k_+\top \to k_+k_- \to 2k_+(2k+1)_+.$$

This brings us to a configuration C' such that $C'^{\#}(m(C)) = 0$ and thus, $m(C') > m(C)$. □

Lemma 6. *From any configuration C in which at least 2^k agents are in non-zero states, there exists a computation execution leading to a configuration in which all agents are in state $\top$.*

Proof. This is achieved mainly by iteration of the previous two lemmas: from configuration C, following Lemma 4, one can reach a configuration C' where there is at least one agent in state $\top$. If some non-zero agents are not in state $\top$, increase the minimal non-zero level in the system by Lemma 5. Iterate until the minimal non-zero level is 2^k, ie. all agents are either in state $\top$ or in state 0. Then use the transition rule $0\top \to \top\top$ to convert all remaining agents from state 0 to state $\top$. Note that such a configuration is trivially stable. □

Theorem 5. *Protocol $\mathcal{P}$ computes the predicate* $[x.\sigma \geq 2^k]$.

Proof. From Lemma 6, the fairness property ensures that any fair computation of protocol $\mathcal{P}'$ starting in a configuration with at least 2^k agents in state 1_+ stably converges to a configuration in which all agents are in state $\top$ and thus agree on output 1. Contrariwise, if the initial configuration holds strictly fewer than 2^k agents in state 1_+ then Lemma 3 guarantees that all agents will always agree on output 0.

Thus protocol $\mathcal{P}'$ computes the predicate $[x.\sigma \geq 2^k]$ and, since they are equivalent, so does $\mathcal{P}$. □

8 CONCLUSION

We proved that predicates $[x.\sigma = 0]$, $[x.\sigma \geq 1]$, $[x.\sigma \geq 2]$ can be computed by some simple exclusive Pavlovian population protocols, as well as

$[x.\sigma \geq x.\sigma']$. We also prove that $[x.\sigma \geq 3]$, $[x.\sigma \geq 2^k]$ can be computed in this model.

It is clear that the subset of the predicates computable by exclusive Pavlovian population protocols is closed under negation: just switch the value of the individual output function of a protocol computing a predicate to get a protocol computing its negation.

Notice that, unlike what happens for general population protocols, composing exclusive Pavlovian population protocols into a exclusive Pavlovian population protocol is not easy. It is not clear whether Pavlovian computable predicates are closed under conjunctions: classical constructions for general population protocols can not be used directly, because there is no possible merging of the game matrices to create the conjunction game. We removed this difficulty in [12] by introducing the idea of Multigames. But in it, we proved that all basic semilinear predicate (like $[x.\sigma \equiv 1[2]]$) can be computed, and we did not find such a protocol in the case of exclusive Pavlovian population protocols.

We conjecture that some semilinear predicate cannot be computed by the symmetric *PAVLOV* population protocol model. A proof seems to require new techniques to get these impossibility results. Moreover, we can also focus on the *non-deterministic* symmetric *PAVLOV* population protocol model. We believe that the computing power is the same as the deterministic *PAVLOV* population protocol model.

REFERENCES

[1] D. Angluin, J. Aspnes, D. Eisenstat, and E. Ruppert. (2007). The computational power of population protocols. *Distributed Computing*, 20(4):279–304.

[2] Dana Angluin, James Aspnes, Melody Chan, Michael J. Fischer, Hong Jiang, and René Peralta. (June 2005). Stably computable properties of network graphs. In Viktor K. Prasanna, Sitharama Iyengar, Paul Spirakis, and Matt Welsh, editors, *Distributed Computing in Sensor Systems: First IEEE International Conference, DCOSS 2005, Marina del Rey, CA, USE, June/July, 2005, Proceedings*, volume 3560 of *Lecture Notes in Computer Science*, pages 63–74. Springer-Verlag.

[3] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. (July 2004). Computation in networks of passively mobile finite-state sensors. In *Twenty-Third ACM Symposium on Principles of Distributed Computing*, pages 290–299. ACM Press.

[4] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. (2006). Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18(4):235–253.

[5] Dana Angluin, James Aspnes, and David Eisenstat. (2006). Stably computable predicates are semilinear. In *PODC '06: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 292–299, New York, NY, USA. ACM Press.

[6] Dana Angluin, James Aspnes, and David Eisenstat. (2008). Fast computation by population protocols with a leader. *Distributed Computing*, 21(3):183–199.

[7] Dana Angluin, James Aspnes, Michael J. Fischer, and Hong Jiang. (December 2005). Self-stabilizing population protocols. In *Ninth International Conference on Principles of Distributed Systems (OPODIS'2005)*, Lecture Notes in Computer Science, pages 79–90. Springer. To appear.

[8] James Aspnes and Eric Ruppert. (2007). An introduction to population protocols. In *Bulletin of the EATCS*, volume 93, pages 106–125.

[9] Robert M. Axelrod. (1984). *The Evolution of Cooperation*. Basic Books.

[10] Joffroy Beauquier, Janna Burman, Julien Clément, and Shay Kutten. (2010). On utilizing speed in networks of mobile agents. In *Proceedings of the 29th Annual ACM Symposium on Principles of Distributed Computing, PODC*, pages 305–314.

[11] Ken Binmore. (1999). *Jeux et Théorie des jeux*. DeBoeck Université, Paris-Bruxelles. Translated from 'Fun and Games: a text on game theory" by Francis Bismans and Eulalia Damaso.

[12] Olivier Bournez, Jérémie Chalopin, Johanne Cohen, Xavier Koeger, and Rabie Mikaël. (2011). Computing with pavlovian oopulations. In *15th International Conference On Principles Of Distributed Systems (OPODIS).*, LNCS, pages 409–420.

[13] Ioannis Chatzigiannakis, Shlomi Dolev, Sándor P. Fekete, Othon Michail, and Paul G. Spirakis. (2009). Not all fair probabilistic schedulers are equivalent. In *Principles of Distributed Systems, 13th International Conference, OPODIS*, pages 33–47.

[14] DJ Daley and DG Kendall. (1965). Stochastic Rumours. *IMA Journal of Applied Mathematics*, 1(1):42–55.

[15] Carole Delporte-Gallet, Hugues Fauconnier, Rachid Guerraoui, and Eric Ruppert. (2006). When birds die: Making population protocols fault-tolerant. In Phillip B. Gibbons, Tarek F. Abdelzaher, James Aspnes, and Ramesh Rao, editors, *Distributed Computing in Sensor Systems, Second IEEE International Conference, DCOSS 2006, San Francisco, CA, USA, June 18-20, 2006, Proceedings*, volume 4026 of *Lecture Notes in Computer Science*, pages 51–66. Springer.

[16] Z. Diamadi and M.J. Fischer. (2001). A simple game for the study of trust in distributed systems. *Wuhan University Journal of Natural Sciences*, 6(1-2):72–82.

[17] Martin E. Dyer, Leslie Ann Goldberg, Catherine S. Greenhill, Gabriel Istrate, and Mark Jerrum. (2002). Convergence of the iterated prisoner's dilemma game. *Combinatorics, Probability & Computing*, 11(2).

[18] Laurent Fribourg, Stéphane Messika, and Claudine Picaronny. (2004). Coupling and self-stabilization. In Rachid Guerraoui, editor, *Distributed Computing, 18th International Conference, DISC 2004, Amsterdam, The Netherlands, October 4-7, 2004, Proceedings*, volume 3274 of *Lecture Notes in Computer Science*, pages 201–215. Springer.

[19] Drew Fudenberg and David K. Levine. (December 1996). The theory of learning in games. Levine's Working Paper Archive 624, UCLA Department of Economics.

[20] D.T. Gillespie. (1992). A rigorous derivation of the chemical master equation. *Physica A*, 188(1-3):404–425.

[21] Herbert W. Hethcote. (December 2000). The mathematics of infectious diseases. *SIAM Review*, 42(4):599–653.

[22] J. Hofbauer and K. Sigmund. (2003). Evolutionary game dynamics. *Bulletin of the American Mathematical Society*, 4:479–519.

[23] Aaron D. Jaggard, Michael Schapira, and Rebecca N. Wright. (2011). Distributed computing with adaptive heuristics. In *Proceedings of Innovations in Computer Science ICS*.

[24] D. Kraines and V. Kraines. (1988). Pavlov and the prisoner's dilemma. *Theory and Decision*, 26:47–79.

[25] Othon Michail, Ioannis Chatzigiannakis, and Paul G. Spirakis. (2011). Mediated population protocols. *Theor. Comput. Sci.*, 412(22):2434–2450.

[26] Othon Michail, Ioannis Chatzigiannakis, and Paul G. Spirakis. (2011). *New Models for Population Protocols*. Morgan & Claypool Publishers.

[27] James Dickson Murray. (2002). *Mathematical Biology. I: An Introduction*. Springer, third edition.

[28] M. Nowak and K. Sigmund. (1993). A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game. *Nature*, 364(6432):56–58.

[29] Martin J. Osbourne and Ariel Rubinstein. (1994). *A Course in Game Theory*. MIT Press.

[30] M. Presburger. (1929). Uber die Vollstandig-keit eines gewissen systems der Arithmetik ganzer Zahlen, in welchemdie Addition als einzige Operation hervortritt. *Comptes-rendus du I Congres des Mathematicians des Pays Slaves*, pages 92–101.

[31] Jörgen W. Weibull. (1995). *Evolutionary Game Theory*. The MIT Press.