Playing With Population Protocols*

Olivier Bournez
Olivier.Bournez@lix.polytechnique.fr
Ecole Polytechnique & Laboratoire d'Informatique (LIX),
91128 Palaiseau Cedex, France

Jérémie Chalopin
Jeremie.Chalopin@lif.univ-mrs.fr
CNRS & Laboratoire d'Informatique Fondamentale de Marseille, CNRS & Aix-Marseille Université,
39 rue Joliot Curie, 13453 Marseille Cedex 13, France

Johanne Cohen
Johanne.Cohen@prism.uvsq.fr
CNRS & PRiSM,
45 Avenue des Etats Unis, 78000 Versailles, France

Xavier Koegler
Xavier.Koegler@liafa.jussieu.fr
École Normale Supérieure & Université Paris Diderot - Paris 7,
Case 7014, 75205 Paris Cedex 13, France

Population protocols have been introduced as a model of sensor networks consisting of very limited mobile agents with no control over their own movement: A collection of anonymous agents, modeled by finite automata, interact in pairs according to some rules.

Predicates on the initial configurations that can be computed by such protocols have been characterized under several hypotheses.

We discuss here whether and when the rules of interactions between agents can be seen as a game from game theory. We do so by discussing several basic protocols.

1 Introduction

The computational power of networks of anonymous resource-limited mobile agents has been investigated in several recent papers.

In particular, Angluin et al. proposed in [1] a new model of distributed computations. In this model, called *population protocols*, finitely many finite-state agents interact in pairs chosen by an adversary. Each interaction has the effect of updating the state of the two agents according to a joint transition function.

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 model was originally proposed to model computations realized by sensor networks in which passive agents are carried along by other entities. The canonical example of [1] corresponds to sensors attached to a flock of birds and that must be programmed to check some global

*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

T. Neary, D. Woods, A.K. Seda and N. Murphy (Eds.):
The Complexity of Simple Programs 2008.
EPTCS 1, 2009, pp. 3–15, doi:10.4204/EPTCS.1.1

properties, like determining whether more than 5% of the population has elevated temperature. Motivating scenarios also include models of the propagation of trust [8].

Much of the work so far on population protocols has concentrated on characterizing which predicates on the initial states can be computed in different variants of the model and under various assumptions. In particular, the predicates computable by the unrestricted population protocols from [1] have been characterized as being precisely the semi-linear predicates, that is to say those predicates on counts of input agents definable in first-order Presburger arithmetic [18]. Semilinearity was shown to be sufficient in [1] and necessary in [2].

Variants considered so far include restriction to one-way communications, restriction to particular interaction graphs, to random interactions, with possibly various kind of failures of agents. Solutions to classical problems of distributed algorithmics have also been considered in this model. Refer to survey [3] for a complete discussion.

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 [12], or rumors [7]. In chemistry the chemical master equation has been justified using (stochastic) pairwise interactions between the finitely many molecules present [16, 11]. In that sense, the model of population protocols may be considered as fundamental in several fields of study.

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, paper [9] considers the dynamics of the so-called *PAVLOV* behaviour in the iterated prisoner lemma. Several results about the time of convergence of this particular dynamics towards the stable state can be found in [9], and [10], for rings, and complete graphs.

The purpose of the following discussion is to better understand whether and when pairwise interactions, and hence population protocols, can be considered as the result of a game. 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 [9] is a limitation. We do so by giving solutions to several basic problems using rules of interactions associated to a symmetric game. As such protocols must also be symmetric, we are also discussing whether restricting to symmetric rules in population protocols is a limitation.

In Section 2, we briefly recall population protocols. In Section 3, we recall some basics from game theory. In Section 4, we discuss how a game can be turned into a dynamics, and introduce the notion of *Pavlovian* population protocol. In Section 5 we prove that any symmetric deterministic 2-states population protocol is Pavlovian, and that the problem of computing the OR, AND, as well as the leader election and majority problem admit Pavlovian solutions. We then discuss our results in Section 6.

2 Population Protocols

A protocol 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*. $\iota:\Sigma\rightarrow Q$ is the initial state mapping, and $\omega:Q\rightarrow\{0,1\}$ 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 $(q_1,q_2)\rightarrow(q_1',q_2')$, or even the notation $q_1q_2\rightarrow q_1'q_2'$, for $(q_1,q_2,q_1',q_2')\in\delta$ (with the convention that $(q_1,q_2)\rightarrow(q_1,q_2)$ when no rule is specified with (q_1,q_2) in the left-hand side).

The protocol is termed *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 $(q_2,q_1) \to (q'_2,q'_1)$.

Computations of a protocol proceed in the following way. The computation takes place among n *agents*, where $n \geq 2$. A *configuration* of the system can be described by a vector of all the agents' states. The state of each agent is an element of Q. Because agents with the same states are indistinguishable, each configuration can be summarized as an unordered multiset of states, and hence of elements of Q.

Each agent is given initially some input value from Σ: Each agent's initial state is determined by applying ι to its input value. This determines the initial configuration of the population.

An execution of a protocol proceeds from the initial configuration by interactions between pairs of agents. Suppose that two agents in state q_1 and q_2 meet and have an interaction. They can change into state q'_1 and q'_2 if (q_1,q_2,q'_1,q'_2) is in the transition relation δ. If C and C' are two configurations, we write $C \to C'$ if C' can be obtained from C by a single interaction of two agents: this means that C contains two states q_1 and q_2 and C' is obtained by replacing q_1 and q_2 by q'_1 and q'_2 in C, where $(q_1,q_2,q'_1,q'_2) \in \delta$. An *execution* of the protocol is an infinite sequence of configurations $C_0, C_1, C_2, \cdots$, where C_0 is an initial configuration and $C_i \to C_{i+1}$ for all $i \geq 0$. An execution is *fair* if for all configurations C that appear infinitely often in the execution, if $C \to C'$ for some configuration C', then C' appears infinitely often in the execution.

At any point during an execution, each agent's state determines its output at that time. If the agent is in state q, its output value is $\omega(q)$. The configuration output 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.

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 $p(I)$.

The following was proved in [1, 2]

Theorem 1 ([1, 2])**.** *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 [18].

3 Game Theory

We now recall the simplest concepts from Game Theory. We focus on non-cooperative games, with complete information, in extensive form.

The simplest game is made up of two players, called I and II, with a finite set of options, called *pure strategies*, $Strat(I)$ and $Strat(II)$. Denote by $A_{i,j}$ (respectively: $B_{i,j}$) the score for player I (resp. II) when I uses strategy $i \in Strat(I)$ and II uses strategy $j \in Strat(II)$.

The scores are given by $n \times m$ matrices A and B, where n and m are the cardinality of $Strat(I)$ and $Strat(II)$. The game is termed *symmetric* if A is the transpose of B: this implies that $n = m$, and we can assume without loss of generality that $Strat(I) = Strat(II)$.

Example 1 (Prisoner's dilemma)**.** *The case where A and B are the following matrices*

$$A = \begin{pmatrix} R & S \\ T & P \end{pmatrix}, B = \begin{pmatrix} R & T \\ S & P \end{pmatrix}$$

with $T > R > P > S$ and $2R > T + S$, is called the prisoner's dilemma. *We denote by C (for cooperation) the first pure strategy, and by D (for defection) the second pure strategy of each player.*

As the game is symmetric, matrix A and B can also be denoted by:

		Opponent C	Opponent D
Player	C	R	S
	D	T	P

A strategy $x \in Strat(I)$ is said to be a best response to strategy $y \in Strat(II)$, denoted by $x \in BR(y)$ if

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

for all strategies $z \in Strat(I)$.

A pair (x,y) is a *(pure) Nash equilibrium* if $x \in BR(y)$ and $y \in BR(x)$. A pure Nash equilibrium does not always exist.

In other words, two strategies (x,y) form a Nash equilibrium if in that state neither of the players has a unilateral interest to deviate from it.

Example 2. *On the example of the prisoner's dilemma, $BR(y) = D$ for all y, and $BR(x) = D$ for all x. So (D,D) is the unique Nash equilibrium, and it is pure. In it, each player has score P. The paradox is that if they had played (C,C) (cooperation) they would have had score R, that is more. 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 C.*

We will also introduce the following definition: Given some strategy $x' \in Strat(I)$, a strategy $x \in Strat(I)$ is said to be a best response to strategy $y \in Strat(II)$ among those different from x', denoted by $x \in BR_{\neq x'}(y)$ if

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

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

Of course, the role of II and I can be inverted in the previous definition.

There are two main approaches to discussing dynamics of games. The first consists in repeating games. The second in using models from evolutionary game theory. Refer to [13, 19] for a presentation of this latter approach.

Repeating Games. Repeating k times a game, is equivalent to extending the space of choices into $Strat(I)^k$ and $Strat(II)^k$: player I (respectively II) chooses his or her action $x(t) \in Strat(I)$, (resp. $y(t) \in Strat(II)$) at time t for $t = 1, 2, \cdots, k$. Hence, this is equivalent to a two-player game with respectively n^k and m^k choices for players.

To avoid confusion, we will call *actions* the choices $x(t), y(t)$ of each player at a given time, 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.

If the game is repeated an infinite number of times, a strategy becomes a function from integers to the set of actions, and the game is still equivalent to a two-player game[1].

Behaviours. 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 I play at time t? In other words, how to choose $x(t) \in Strat(I)$? (resp. $y(t) \in Strat(II)$?)

Is is natural to suppose that this is given by some behaviour rules:

$$x(t) = f(X_{t-1}, Y_{t-1}),$$

$$y(t) = g(X_{t-1}, Y_{t-1})$$

for some particular functions f and g.

The Specific Case of the Prisoner's Lemma. The question of the best behaviour rule to use for the prisoner lemma gave birth to an important literature. In particular, after the book [4], that describes the results of tournaments of behaviour rules for the iterated prisoner lemma, and that argues that there exists a best behaviour 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.

A lot of other behaviours, most of them with very picturesque names have been proposed and studied: see for example [4], [5], [15].

Among possible behaviours is $PAVLOV$: in the iterated prisoner lemma, a player cooperates if and only if both players opted for the same alternative in the previous move. This name [14, 17, 4] 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 R or T points, but switches behaviour if it was punished by receiving only P or S points. Refer to [17] for some study of this strategy in the spirit of Axelrod's tournaments.

The $PAVLOV$ behaviour can also be termed *WIN-STAY, LOSE-SHIFT* as if the play on the previous round resulted 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 [17, 4].

Going From 2 Players to N Players. $PAVLOV$ behaviour is Markovian: a behaviour f is *Markovian*, if $f(X_{t-1}, Y_{t-1})$ depends only on $x(t-1)$ and $y(t-1)$.

From such a behaviour, it is easy to obtain a distributed dynamic. For example, let's follow [9], for the prisoner's dilemma.

Suppose that we have a connected graph $G = (V, E)$, with N vertices. The vertices correspond to players. An instantaneous configuration of the system is given by an element of $\{C, D\}^N$, that is to say by the state C or D of each vertex. Hence, there are 2^N configurations.

[1] but whose matrices are infinite.

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 dilemma with the $PAVLOV$ behaviour. It is easy to see that this corresponds to executing the following rules:

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

What is the final state reached by the system? The underlying model is a very large Markov chain with 2^N states. The state $E^* = \{C\}^N$ is absorbing. If the graph G does not have any isolated vertex, this is the unique absorbing state, and there exists a sequence of transformations that transforms any state E into this state E^*. As a consequence, from well-known classical results in Markov chain theory, whatever the initial configuration is, with probability 1, the system will eventually be in state E^* [6]. The system is *self-stabilizing*.

Several results about the time of convergence towards this stable state can be found in [9], and [10], for rings, and complete graphs.

What is interesting in this example is that it shows how to go from a game, and a behaviour to a distributed dynamics 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 1 (Associating a Protocol to a Game)**.** *Assume a symmetric two-player game is given. Let Δ be some threshold.*

The protocol associated to the game is a population protocol whose set of states is Q, where $Q = Strat(I) = Strat(II)$ 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* $M_{q_1,q_2} \geq \Delta$
- $q_1' \in BR_{\neq q_1}(q_2)$ *when* $M_{q_1,q_2} < \Delta$

and

- $q_2' = q_2$ *when* $M_{q_2,q_1} \geq \Delta$
- $q_2' \in BR_{\neq q_2}(q_1)$ *when* $M_{q_2,q_1} < \Delta$,

where M is the matrix of the game.

Definition 2 (Pavlovian Population Protocol)**.** *A population protocol is* Pavlovian *if it can be obtained from a game as above.*

Remark 1. *Clearly a Pavlovian population protocol must be* symmetric*: indeed, whenever $(q_1, q_2, q_1', q_2') \in \delta$, one has $(q_2, q_1, q_2', q_1') \in \delta$.*

5 Some Specific Pavlovian Protocols

We now discuss whether assuming protocols Pavlovian is a restriction.

We start by an easy consideration.

Theorem 2. *Any symmetric deterministic* 2*-states population protocol is Pavlovian.*

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

$$\begin{cases} ++ & \rightarrow & \alpha_{++}\alpha_{++} \\ +- & \rightarrow & \alpha_{+-}\alpha_{-+} \\ -+ & \rightarrow & \alpha_{-+}\alpha_{+-} \\ -- & \rightarrow & \alpha_{--}\alpha_{--} \end{cases} \tag{4}$$

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

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

		Opponent +	Opponent -
Player	+	β_{++}	β_{+-}
	-	β_{-+}	β_{--}

taking threshold $\Delta = 1$, where for all $q_1, q_2 \in \{+,-\}$,

- $\beta_{q_1q_2} = 2$ if $\alpha_{q_1q_2} = q_1$,
- $\beta_{q_1q_2} = 0$ otherwise.

□

Unfortunately, not all rules correspond to a game.

Proposition 1. *Some symmetric population protocols are not Pavlovian.*

Proof. Consider for example a deterministic 3-states 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-states population protocol. Consider its payoff matrix M. Let $M(q_0, q_0) = \beta_0$, $M(q_1, q_0) = \beta_1$, $M(q_2, q_0) = \beta_2$. We must have $\beta_0 \geq \Delta, \beta_1 \geq \Delta$ 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.

5.1 Basic Protocols

Proposition 2. *There is a Pavlovian protocol that computes the logical* OR *(resp.* AND*) of input bits.*

Proof. Consider the following protocol to compute OR,

$$\begin{cases} 01 & \rightarrow & 11 \\ 10 & \rightarrow & 11 \\ 00 & \rightarrow & 00 \\ 11 & \rightarrow & 11 \end{cases} \tag{5}$$

and the following protocol to compute AND,

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

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

□

Remark 2. *Notice that* OR *(respectively* AND*) protocol corresponds to the predicates on counts of input agents* $n_0 \geq 1$ *(resp.* $n_1 = 0$*) where* n_0*,* n_1 *are the number of input agents in state* 0 *and* 1 *respectively.*

Remark 3. *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* $n_1 \equiv 1$ $(mod\ 2)$*.*

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

Indeed, the answer is not eventually known by all the agents. 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*.*

5.2 Leader Election

The classical solution [1] to the leader election problem (starting from a configuration with ≥ 1 leaders, eventually exactly one leader survives) is the following:

$$\begin{cases} LL & \rightarrow & LN \\ LN & \rightarrow & LN \\ NL & \rightarrow & NL \\ NN & \rightarrow & NN \end{cases} \tag{8}$$

Unfortunately, this protocol is non-symmetric, and hence non-Pavlovian.

Remark 4. *Actually, the problem is with the first rule, since one wants two leaders to become only one. If the two leaders are identical, this is clearly problematic with symmetric rules.*

However, the leader election problem can actually be solved by a Pavlovian protocol, at the price of a less trivial protocol.

Proposition 3. *The following Pavlovian protocol solves the leader election problem, as soon as the population is of size* ≥ 3.

$$\begin{cases} L_1L_2 & \to & L_1N \\ L_1N & \to & NL_2 \\ L_2N & \to & NL_1 \\ NN & \to & NN \\ L_2L_1 & \to & NL_1 \\ NL_1 & \to & L_2N \\ NL_2 & \to & L_1N \\ L_1L_1 & \to & L_2L_2 \\ L_2L_2 & \to & L_1L_1 \end{cases} \tag{9}$$

Proof. Indeed, starting from a configuration containing not only Ns, eventually after some time configurations will have exactly one leader, that is one agent in state L_1 or L_2.

Indeed, the first rule and the fifth rule decrease strictly the number of leaders whenever there are more than two leaders. Now the other rules, preserve the number of leaders, and are made such that an L_1 can always be transformed into an L_2 and vice-versa, and hence are made such that a configuration where first or fifth rule applies can always be reached whenever there are more than two leaders. The fact that it solves the leader election problem then follows from the hypothesis of fairness in the definition of computations.

This is a Pavlovian protocol, since it corresponds to the following payoff matrix, with threshold $\Delta = 4$

		Opponent		
		L_1	L_2	N
Player	L_1	1	4	1
	L_2	3	1	1
	N	2	1	4

□

5.3 Majority

Proposition 4. *The majority problem (given some population of* 0*s and* 1*s, determine whether there are more* 0*s than* 1*s) can be solved by a Pavlovian population protocol.*

If one prefers, the predicate $n_0 \geq n_1$ on counts of input agents can be computed by a Pavlovian population protocol.

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

$$\begin{cases} NY & \to & YY \\ YN & \to & YY \\ N0 & \to & Y0 \\ 0N & \to & 0Y \\ Y1 & \to & N1 \\ 1Y & \to & 1N \\ 01 & \to & NY \\ 10 & \to & YN \end{cases} \tag{10}$$

taking

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

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 number of 0s and 1s is preserved except when a 0 meets a 1. In that latter case, the two agents are deleted and transformed into a Y and a N.

If there are initially strictly more 0s than 1s, from the fairness condition, each 1 will be paired with a 0 and at some point no 1 will left. By fairness and since there is still at least a 0, a configuration containing only 0 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 1s than 0s.

Suppose now that initially, there are exactly the same number of 0s and 1s. By fairness, there exists a step when no more agents in the state 0 or 1 left. Note that at the moment where the last 0 is matched with the last 1, a Y is created. Since this Y can be "broadcast" 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 with threshold 2.

		Opponent			
		N	Y	0	1
Player	N	3	1	1	3
	Y	2	3	3	1
	0	2	2	2	1
	1	2	2	1	2

□

6 Discussions

We proved that predicates on counts of input agents $n \geq 0$, $n = 0$, $n \geq m$, where n, m are some counts of input agents, can be computed by some Pavlovian population protocols.

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

However, some work remains to be done to fully characterize which predicates can be computed by a Pavlovian population protocol. The first steps would be to understand the following questions.

Question 1. *Is* $mod\ 2$*, or equivalently the predicate* $n \equiv 1\ (mod\ 2)$*, computable by a Pavlovian population protocol?*

Question 2. *Is* $\geq k$*, or equivalently the predicate* $n \geq k$*, for fixed* k*, computable by a Pavlovian population protocol?*

Notice that, unlike what happens for general population protocols, composing Pavlovian population protocols into a Pavlovian population protocol is not easy. It is not clear whether Pavlovian computable predicates are closed by conjunctions: classical constructions for general population protocols can not be used directly.

As we said, Pavlovian Population protocols are symmetric. We however know that assuming population protocols symmetric is not a restriction.

Proposition 5. *Any population protocol can be simulated by a symmetric population protocol, as soon as the population is of size* ≥ 3.

Before proving this proposition, we state the (immediate) main consequence.

Corollary 1. *A predicate is computable by a symmetric population protocol if and only if it is semilinear.*

Proof (of proposition): To a population protocol $(Q, \Sigma, \iota, \omega, \delta)$, with $Q = \{q_1, \cdots, q_n\}$ associate population protocol $(Q \cup Q', \Sigma, \iota, \omega, \delta')$ with $Q' = \{q'_1, \cdots, q'_n\}$, $\omega(q') = \omega(q)$ for all $q \in Q$, and for all rules

$$qq \to \alpha\beta$$

in δ, the following rules in δ':

$$\begin{cases} qq' & \to & \alpha\beta \\ q'q & \to & \beta\alpha \\ qq & \to & q'q' \\ q'q' & \to & qq \\ q\gamma & \to & q'\gamma \\ q'\gamma & \to & q\gamma \\ \gamma q & \to & \gamma q' \\ \gamma q' & \to & \gamma q \end{cases}$$

for all $\gamma \in Q \cup Q', \gamma \neq q, \gamma \neq q'$, and for all pairs of rules

$$\begin{cases} qr & \to & \alpha\beta \\ rq & \to & \delta\varepsilon \end{cases}$$

with $q,r \in Q$, the following rules in δ':

$$\begin{cases} qr' & \rightarrow & \alpha\beta \\ r'q & \rightarrow & \beta\alpha \\ rq' & \rightarrow & \delta\varepsilon \\ q'r & \rightarrow & \varepsilon\delta. \end{cases}$$

The obtained population protocol is clearly symmetric. Now the first set of rules guarantees that a state in Q can always be converted to its primed version in Q' and vice-versa. By fairness, whenever a rule $qq \rightarrow \alpha\beta$ (respectively $qr \rightarrow \alpha\beta$) can be applied, then the corresponding two first rules of the first set of rules (resp. of the second set of rules) can eventually be fired after possibly some conversions of states into their primed version or vice-versa. □

References

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

[2] Dana Angluin, James Aspnes, and David Eisenstat. 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, 2006. ACM Press.

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

[4] Robert M. Axelrod. *The Evolution of Cooperation*. Basic Books, 1984.

[5] Bruno Beaufils. *Modèles et simulations informatiques des problèmes de coopération entre agents*. PhD thesis, Université de Lille I, 2000.

[6] Pierre Brémaud. *Markov Chains, Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer-Verlag, New York, 2001.

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

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

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

[10] Laurent Fribourg, Stéphane Messika, and Claudine Picaronny. 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, 2004.

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

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

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

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

[15] Ouassila Labbani. Comparaison des théories des jeux pour l'étude du comportement d'agents. Master's thesis, Université de Lille I, 2003.

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

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

[18] M. Presburger. 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, 1929.

[19] Jörgen W. Weibull. *Evolutionary Game Theory*. The MIT Press, 1995.