

# Distributed Consensus on Boolean Information

Adriano Fagiolini,<sup>\*</sup> Simone Martini,<sup>\*</sup> Nevio Dubbini,<sup>\*</sup>  
and Antonio Bicchi<sup>\*</sup>

<sup>\*</sup> *Interdepartmental Research Center “E. Piaggio”,  
Faculty of Engineering, University of Pisa, Italy,  
{a.fagiolini, simone.martini, bicchi}@ing.unipi.it  
dubbini@mail.dm.unipi.it.  
<http://www.piaggio.cci.unipi.it>.*

---

**Abstract:** In this paper we study the convergence towards consensus on information in a distributed system of agents communicating over a network. The particularity of this study is that the information on which the consensus is sought is not represented by real numbers, rather by logical values or sets. Whereas the problems of allowing a network of agents to reach a consensus on logical functions of input events, and that of agreeing on set-valued information, have been separately addressed in previous work, in this paper we show that these problems can indeed be attacked in a unified way in the framework of Boolean distributed information systems. Based on a notion of contractivity for Boolean dynamical systems, a necessary and sufficient condition ensuring the global convergence toward a unique equilibrium point is presented. This result can be seen as a first step toward the definition of a unified framework to uniformly address all consensus problems on Boolean algebras.

Keywords: Boolean consensus systems, global convergence, fixed points or equilibria.

---

## 1. INTRODUCTION

Recent years have witnessed a gradual yet constant migration of interests from applications involving a unique process and controller, to scenarios where many distributed agents harmoniously interact so as to achieve a common control goal. Most of the problems, and of the solutions that have been proposed so far, can be formulated as consensus problems over *continuous domains*, where local agents exchange data that consists of real vectors or scalars. The only difference is in the type of rule each agent uses to combine its own information with the one received from its neighbors of the communication graph. In the simplest case, the evolution of the network of agents can be described by a linear iterative rule

$$x(t+1) = Ax(t) + Bu(t),$$

where  $t$  is a discrete-time,  $x \in \mathbb{R}^n$  is the system's state,  $A$  is a weight matrix, and  $u$  is an input vector. Matrix  $A$  has to be compliant with the available communication graph and is designed to allow the network convergence to a unique decision  $x(\infty) \rightarrow \alpha \mathbb{1}$ , that may or may not depend on the initial system's state. Falling into this linear framework are most of the key papers on consensus (Olfati-Saber et al. [2007], Jadbabaie et al. [2003], Fax and Murray [2004]). Moreover, the nonlinear setting encompasses other important schemes for achieving consensus on continuous, finite variables. Within this setting, the solution proposed by Cortés et al. [2004], based on the centroidal Voronoi tessellation, allows deployment of a collection of mobile agents so as to maximize the network's ability to perform a sensing task within a given environment. These problems,

and indeed many others, can be efficiently solved by means of these agreement mechanisms.

However, new emerging issues in the field of distributed control entail defining different forms of consensus algorithms. Very recently, Greco et al. [2008] have addressed the sensing coverage problem with agents that are allowed to move in a discrete, network-like environment. The problem of averaging a set of initial measures, taken by a collection of distributed sensors, in the presence of communication constraints has been recently addressed by Frasca et al. [2008]. A consensus strategy, where exchanged data consists of symbols obtained through a logarithmic quantizer, is proposed. On the furthest part of this track are problems over a *discrete domain*, where the system's state is a logical vector  $x \in \mathbb{B}^n$ . This includes the problem of building a map of visitors/intruders' presence in the rooms and corridors of an art gallery, that has been attacked by Fagiolini et al. [2008b], through introduction of so-called *logical consensus*.

Inadequacy of available solutions for distributed network agreement arises also in control applications where sensors' measures are affected by uncertainty. Consider an application where mobile robots must simultaneously localize themselves and build a map of the environment, by using their local vision systems. Traditional approaches to model sensors' noise as an additive or multiplicative signal is possible but not natural. Di Marco et al. [2003] proposed a centralized solution, where robots exchange data representing confidence sets of the positions of items detected in the environment. Moreover, Fagiolini et al.

[2008a] considered the problem of detecting misbehaving agents within a collection of robots that are supposed to plan their motions according to a share set of rules. The objective is attained by definition of a *set-valued consensus algorithm*, where local agents exchange data representing free and occupied regions of the environment. The algorithm overcomes limitations of available solutions in the fact that it can operate over *infinite domains*. Finally, Marzullo [1985] and later Mills [1991] considered the problem of synchronizing the clocks of a set of distributed agents, and proposed a centralized solution to the problem. Fagiolini et al. [2009] have very recently shown that this problem can be solved by means of set-valued consensus.

So far, design of logical consensus as well as consensus on sets have been individually addressed, and only ad-hoc solutions have been proposed. As a matter of fact, a network of agents running either types of consensus are instances of a Boolean iterative system, i.e. a system where the state is a vector of elements in a Boolean domain and is updated through operations in a Boolean algebra. The aim of this paper is to present initial results toward the definition of a unified framework for dealing with such consensus problems. With this respect, a notion of a Boolean vector space is known since the seminal work of Subrahmanyam [1964]. However, the behavior of a Boolean iterative system is far from been completely understood. This fact and the existence of problems of practical interests, such as the ones mentioned above, are the main motivations for the current work. The work is based on results available from the literature on cellular automata and logical iterative systems (Robert [1978, 1980]), that are extended to general subsets of  $\mathbb{R}^n$ . Based on a notion of contractivity for Boolean dynamical systems, a necessary and sufficient condition ensuring the global convergence toward a unique equilibrium point is presented. Application of this result is finally shown through some examples.

## 2. BOOLEAN ALGEBRAS AND BOOLEAN ITERATIONS

A *Boolean algebra* is defined as the 6-tuple  $\tilde{\mathcal{B}} = (\tilde{\mathbb{B}}, \wedge, \vee, \neg, \tilde{0}, \tilde{1})$ , where  $\tilde{\mathbb{B}}$  is a set called domain,  $\wedge$  and  $\vee$  are binary operations called meet and join, respectively,  $\neg$  is a unary operation called complement, and  $\tilde{0}$  and  $\tilde{1}$  are called the null and unity elements of the algebra and belong to the domain  $\tilde{\mathbb{B}}$ . The elements of a Boolean algebra must satisfy a set of axioms, omitted here for the sake of brevity, that allow us to introduce a *partial order relation*  $\leq$ . In particular, for any two elements  $a, b \in \tilde{\mathbb{B}}$ , we say that  $a \leq b$  if, and only if,

$$a = a \wedge b \text{ or, equivalently } a \vee b = b.$$

$\tilde{0}$  and  $\tilde{1}$  are the least and greatest elements of the algebra. Furthermore, an element  $a \in \tilde{\mathbb{B}}$  is referred to as a *Boolean scalar*. Consider the set  $\tilde{\mathbb{B}}^n$  composed of *Boolean vectors*  $x$ , provided with the meet  $\wedge$  and join  $\vee$  with another vector  $y \in \tilde{\mathbb{B}}^n$ , and the meet  $\wedge$  with a Boolean scalar  $a \in \tilde{\mathbb{B}}$ . Finally, consider the set composed of all square *Boolean matrices*  $A \in \tilde{\mathbb{B}}^{n \times n}$ , provided with the meet and join operations between two Boolean matrices, and the meet of a Boolean matrix  $A$  with a Boolean scalar  $a$ .

*Definition 1.* If  $A = \{a_{ij}\}, B = \{b_{ij}\} \in \tilde{\mathbb{B}}^{n \times n}$  and  $v = (v_1, \dots, v_n)^T, w = (w_1, \dots, w_n)^T \in \tilde{\mathbb{B}}^n$ , we define the scalar product  $w \cdot v$  to be

$$\bigvee_{i=1}^n v_i \wedge w_i,$$

$(Av)_i$  to be the scalar product between the  $i$ -th row of  $A$  and the vector  $v$ , and  $AB_{ij}$  to be the scalar product between the  $i$ -th row of  $A$  and the  $j$ -th column of  $B$ .

In other words products between matrices and vectors, and between two matrices, are computed in the usual way, substituting  $+$  with  $\cup$  and  $\cdot$  with  $\cap$ . We will denote with  $0$  the null scalar, vector, or matrix, according to the context. The above described partial order relation  $\leq$  between any two elements of  $\tilde{\mathbb{B}}$  can be extended to Boolean vectors and matrices by assuming component-wise evaluation.

*Remark 1.* Examples of Boolean algebras are binary values with logical **and**, and **or** operations, and set-valued Boolean algebras with union  $\cup$  and intersection  $\cap$  operations. See examples considered in the application section.

A *Boolean vector function* is an application  $f : U \rightarrow Y$ , where  $U \subseteq \tilde{\mathbb{B}}^m$ , and  $Y \subseteq \tilde{\mathbb{B}}^p$ . A *Boolean iteration map* is an application  $F : X \times U \rightarrow X$ , where  $X \subseteq \tilde{\mathbb{B}}^n$ ,  $U \subseteq \tilde{\mathbb{B}}^m$ , that links any pair of points  $(X, u)$  of the set  $X \times U$  to a point in  $X$ . Given an initial vector or point  $x(0) \in \tilde{\mathbb{B}}^n$ , and an input sequence  $u(0), u(1), \dots$ , one can consider the sequence of vectors generated by the iterative rule

$$x(t+1) = F(x(t), u(t)), \quad (1)$$

where  $t = 0, 1, \dots$  is a discrete time.

*Definition 2.* (Equilibrium point). Given a constant input sequence  $u(t) = \bar{u}$ , a vector  $x^* \in \tilde{\mathbb{B}}^n$  is said to be an *equilibrium* of application  $F$  if  $F(x^*, \bar{u}) = x^*$ .

*Definition 3.* (Basin of attraction). Given an equilibrium point  $x^*$  of  $F$ , i.e.  $F(x^*, \bar{u}) = x^*$ , the *basin of attraction* of  $x^*$  is defined as the set of all initial Boolean vectors  $x(0)$  for which iterations of  $F$  eventually converge to  $x^*$ , i.e. there exists a time  $\bar{k} \geq 0$  such that  $x(\bar{k}) = x^*$ .

To study the behavior of the iterations of Eq. 1, given an input sequence  $u(0), u(1), \dots, u(t)$ , one can in principle directly compute its evolution, which is inefficient and not systematic. To avoid this, suitable analytical tools would be required, but a full understanding of the behavior of such systems has been reached only in the case of the binary Boolean algebra  $\tilde{\mathbb{B}} = \{0, 1\}$  (Robert [1978, 1980]). In fact, the evolution of a cellular automaton, that can be described by a logical iterative system, can either converge to a fixed point or enter into a cycle. Conversely, the fact that a generic Boolean iteration can be defined over *infinite* domains  $\tilde{\mathbb{B}}$  does not allow us to *a priori* exclude more complex evolutions, such as the presence of accumulation points. For this reason, we restrict to consider a class of Boolean iterations showing “nice” behaviors. More precisely, we focus on *contractive* iterations that can build upon the Boolean algebra of the sets. Due to Stone’s Representation Theorem, the presented results remain valid for all Boolean algebras that are isomorphic to the algebra of the sets. In this vein, recall that, given a (possibly infinite) nonempty set  $\mathcal{X}$ , the power set  $\mathcal{P}(\mathcal{X}) = 2^{\mathcal{X}}$ , the set of all subsets of  $\mathcal{X}$ , forms a

Boolean algebra with the two binary operations  $\wedge := \cap$  (set intersection), and  $\vee := \cup$  (set union), the unary operation  $\neg := \mathcal{C}(\cdot)$  (set complementation), the smallest element is  $\tilde{0} := \emptyset$ , and the largest element is  $1 := \mathcal{X}$  (the set itself). The partial order relation  $\leq$  specializes as follows. Given two sets  $x, y \in \mathcal{X}$ , we say that  $x$  is contained in  $y$ , and we write  $x \subseteq y$ , if, and only if, one of the two following equivalent relations holds: 1)  $x = x \cap y$ , 2)  $x \cup y = y$ .

For the remainder of this paper, we will assume a constant input  $u(t) = \bar{u}$ , which will only translate the system's equilibrium and thus will be omitted for brevity.

### 3. CONVERGENCE OF CONTRACTIVE BOOLEAN MAPS

We first need to introduce a metric over the Boolean domain  $\mathcal{P}(\mathcal{X})$ . To this aim, consider the *Boolean vector distance* defined through the application  $\mathcal{D} : \mathcal{P}(\mathcal{X})^n \times \mathcal{P}(\mathcal{X})^n \rightarrow \mathcal{P}(\mathcal{X})^n$  given by

$$\mathcal{D}(x, y) \stackrel{\text{def}}{=} (\mathcal{D}_1(x, y), \dots, \mathcal{D}_n(x, y))^T, \quad (2)$$

where  $x, y \in \mathcal{P}(\mathcal{X})^n$ , and

$$\mathcal{D}_i(x, y) = (\mathcal{C}(x_i) \cap y_i) \cup (x_i \cap \mathcal{C}(y_i)),$$

is the *symmetric difference* between any pair of two elements in  $\mathcal{P}(\mathcal{X})$ . It is worth noting that, in case of the binary Boolean algebra, with  $\mathcal{P}(\mathcal{X}) = \{0, 1\}$ , this definition specializes to the binary vector distance used by Robert [1980]. Moreover, it can be shown that the vector distance  $\mathcal{D}$  satisfies the following axioms

$$\begin{aligned} \mathcal{D}(x, y) &= \mathcal{D}(y, x), \quad \forall x, y \in \mathcal{X}^n, \\ \mathcal{D}(x, y) &= \emptyset \quad \text{iff } x = y, \\ \mathcal{D}(x, y) &\subseteq \mathcal{D}(x, z) \cup \mathcal{D}(z, y). \end{aligned}$$

We are also interested in the following notion:

*Definition 4.* (Incidence matrix). The *incidence matrix*  $B(F) = \{b_{i,j}\}$  of a Boolean iteration  $F = (F_1, \dots, F_n)^T$  is a binary Boolean matrix with

$$b_{i,j} = \begin{cases} \mathcal{X} & \text{if } F_i \text{ depends on } x_j, \\ \emptyset & \text{otherwise.} \end{cases}$$

*Example 1.* Consider the following application

$$F(x) = \begin{pmatrix} (x_1 \cap x_2) \cup x_3 \\ \mathcal{C}(x_3) \\ \mathcal{X} \end{pmatrix}.$$

Its incidence matrix is

$$B(F) = \begin{pmatrix} \mathcal{X} & \mathcal{X} & \mathcal{X} \\ \emptyset & \emptyset & \mathcal{X} \\ \emptyset & \emptyset & \emptyset \end{pmatrix}.$$

We can prove the following results:

*Proposition 1.* Given any two generic vectors  $x, y \in \mathcal{P}(\mathcal{X})^n$ , the following Boolean inequality holds

$$\mathcal{D}(F(x), F(y)) \subseteq B(F) \mathcal{D}(x, y). \quad (3)$$

*Proof 1.* Let us consider the  $i$ -th component of the inequality

$$\begin{aligned} &\mathcal{D}_i(F_i(x_1, \dots, x_n), F_i(y_1, \dots, y_n)) \subseteq \\ &\subseteq \mathcal{D}_i(F_i(x_1, \dots, x_n), F_i(y_1, x_2, \dots, x_n)) \cup \\ &\cup \mathcal{D}_i(F_i(y_1, x_2, \dots, x_n), F_i(y_1, y_2, x_3, \dots, x_n)) \cup \\ &\dots \\ &\cup \mathcal{D}_i(F_i(y_1, \dots, y_{n-1}, x_n), F_i(y_1, \dots, y_{n-1}, y_n)) \subseteq \\ &\subseteq b_{i,1} \mathcal{D}_1(x_1, y_1) \cup b_{i,2} \mathcal{D}_2(x_2, y_2) \cup \dots \\ &\dots \cup b_{i,n} \mathcal{D}_n(x_n, y_n). \end{aligned}$$

The thesis follows by repeating this results for all rows of the inequality.

*Proposition 2.* A Boolean matrix  $M$  satisfies the Boolean inequality

$$\mathcal{D}(F(x), F(y)) \subseteq M \mathcal{D}(x, y), \quad (4)$$

for every pair of vectors  $x, y \in \mathcal{P}(\mathcal{X})^n$ , if, and only if,

$$B(F) \subseteq M.$$

*Proof 2.* The proof of sufficiency is trivial. Let us then focus on the necessity, and suppose by absurd that there exists a Boolean matrix  $M = (m_{i,j})$  satisfying the inequality in Eq. 2, but also admitting an element  $m_{i,j} \subset b_{i,j}$ , where  $b_{i,j}$  is the corresponding element in the incidence matrix  $B(F)$ . This necessarily means that  $b_{i,j} = \mathcal{X}$ , and  $m_{i,j} \subset \mathcal{X}$ . Then, as  $b_{i,j} = \mathcal{X}$ ,  $F_i$  depends on  $x_j$ , and there must exist two vectors  $x = (x_1, \dots, x_j, \dots, x_n)^T$ , and  $x' = (x_1, \dots, y_j, \dots, x_n)^T$ , with  $x_j \neq y_j$ , such that

$$\mathcal{D}_i(F_i(x), F_i(x')) = \mathcal{X}.$$

Computation of  $x'$  starting from  $x$  is always possible, but is omitted here. Basically, given  $x$ , and  $F_i(x)$ , we look for a vector  $y$  such that  $F_i(y)$  is complementary to  $F_i(x)$ . This last quantity is upper bounded by

$$\begin{aligned} &\left( \bigcup_{s=1}^{j-1} m_{i,s} \underbrace{\mathcal{D}_s(x_s, x_s)}_{=\emptyset} \right) \cup \underbrace{m_{i,j} \mathcal{D}_j(x_j, y_j)}_{\subset \mathcal{X}} \cup \\ &\cup \left( \bigcup_{s=j+1}^n m_{i,s} \underbrace{\mathcal{D}_s(x_s, x_s)}_{=\emptyset} \right) \subset \mathcal{X}, \end{aligned}$$

that is a contradiction. Therefore, it must hold  $b_{i,j} \subseteq m_{i,j}$ , for all  $i$ , and  $j$ .

*Proposition 3.* Given two Boolean iteration maps  $F(x)$  and  $G(x)$ , the incidence matrix of the application obtained as the composition of the two,  $F(G(x))$ , satisfies the logical inequality

$$B(F(G)) \subseteq B(F) B(G). \quad (5)$$

*Proof 3.* The proof trivially follows from above. Indeed, if  $(F \circ G)_i$  depends on  $x_j$ , then there exists  $k$  s.t.  $F_i$  depends on  $x_k$  and  $G_k$  depends on  $x_j$ . Hence,  $B(F)_{i,k} \cap B(G)_{k,j} = \mathcal{X}$  which in turn implies that  $(B(F) B(G))_{i,j} = \mathcal{X}$ .

#### 3.1 Boolean eigenvalues

Let us first consider linear Boolean iterative maps  $F$  that are represented by constant matrices.

*Definition 5.* (basis vectors). The set composed of the  $n$  Boolean vectors  $e_1, e_2, \dots, e_n \in \mathcal{X}$ , with

$$e_j = (\underbrace{\emptyset, \dots, \emptyset}_{j-1 \text{ times}}, \mathcal{X}, \underbrace{\emptyset, \dots, \emptyset}_{n-j \text{ times}})^T,$$

forms a base of the infinite set  $\mathcal{P}(\mathcal{X})$ .

The definition implies that every vector  $x \in \mathcal{P}(\mathcal{X})^n$  can be represented as a linear combination of the basis vectors.

*Definition 6.* (Boolean eigenvalues and eigenvectors) Given a Boolean matrix  $A \in \mathcal{P}(\mathcal{X})^{n \times n}$ , a Boolean scalar  $\lambda \in \mathcal{P}(\mathcal{X})$  is an *eigenvalue* of  $A$  if, and only if, there exists a Boolean *eigenvector*  $x \in \mathcal{P}(\mathcal{X})^n$ ,  $x \neq \emptyset$ , s.t.

$$Ax = \lambda x.$$

*Definition 7.* (Boolean spectrum)

The set of the eigenvalues of a Boolean matrix  $A$  represents its *Boolean spectrum* and it is denoted as

$$\sigma(A) = \{\lambda \mid \exists x \neq \emptyset : Ax = \lambda x\}.$$

Then, we can prove the following

*Proposition 4.* A Boolean matrix  $A \in \mathcal{P}(\mathcal{X})^{n \times n}$ ,  $A = \{a_{i,j}\}$  has a Boolean eigenvalue  $\lambda = \emptyset$  if, and only if,  $A$  has at least one column for which the union (join) of all its elements is less than  $\mathcal{X}$ , i.e. there exists  $j \in \{1, \dots, n\}$  s.t.

$$\bigcup_{i=1}^n a_{i,j} \subset \mathcal{X}. \quad (6)$$

*Proof 4.* (Sufficiency)

Suppose that  $j$  satisfies (6). We want to prove that  $\lambda = \emptyset$  is a Boolean eigenvalue of  $A$ , i.e. there exists  $x \neq \emptyset$  s.t.  $Ax = \emptyset x = \emptyset$ . Consider a vector composed of emptysets for all components except for the  $j$ -th. Then, we have  $Ax = A_j x_j$ , where  $A_i$  is the  $i$ -th column of  $A$ , that we want to be the vector of emptysets. This last equation can be explicitly written as

$$\begin{aligned} a_{i,1} \cap x_j &= \emptyset, \\ a_{i,2} \cap x_j &= \emptyset, \\ &\vdots \\ a_{i,n} \cap x_j &= \emptyset. \end{aligned}$$

This holds if, and only if, it also holds

$$(a_{1,j} \cap x_j) \cup (a_{2,j} \cap x_j) \cup \dots \cup (a_{n,j} \cap x_j) = \emptyset,$$

and by the distributivity property it yields

$$\begin{aligned} (a_{1,j} \cup a_{2,j} \cup \dots \cup a_{n,j}) \cap x_j &= \emptyset, \\ \bigcup_{i=1}^n a_{i,j} \cap x_j &= \emptyset, \end{aligned}$$

that requires that the two sets are disjoint. Moreover, the value  $\bar{x}_j = \mathcal{X} \setminus (\bigcup_{i=1}^n a_{i,j}) \neq \emptyset$  satisfies this condition and, due to the hypothesis in Eq. 6, is different from  $\emptyset$ , which implies that  $x = (\emptyset, \dots, \emptyset, \bar{x}_j, \emptyset, \dots, \emptyset)^T$  is an eigenvector of  $A$ .

(Necessity)

Suppose that  $\lambda = \emptyset$  is an eigenvalue of  $A$ . This implies that there exists  $x \neq \emptyset$  s.t.  $Ax = \emptyset$ . This means  $\bigcup_{i=1}^n a_{i,j} \cap x_j = \emptyset$ , for all  $j$ . This condition is trivially satisfied for every null component of  $x$ . For every other component of  $x$  that is different than  $\emptyset$ , the component itself must be disjoint to the union of the sets composing the corresponding column of  $A$ . This implies that their union can not cover the entire set  $\mathcal{X}$ , which finally gives the thesis.

*Remark 2.* It is worth noting that, if  $A$  has a Boolean eigenvalue  $\lambda$ , with assigned eigenvector  $x$ , then, for every permutation  $P$ , the matrix  $A' \stackrel{\text{def}}{=} P^T A P$  has the same eigenvalue, assigned with eigenvector  $v = P^T x$ . Note that  $P$  is a permutation in the classical sense, but where  $\emptyset$ , and

1 are replaced with  $\emptyset$ , and  $\mathcal{X}$ , respectively. To prove this, observe that, for hypothesis, we have  $Ax = \lambda x$ . Left-multiplying by  $P^T$  this gives  $P^T A x = \lambda P^T x$ , and, from the identity  $I = P^T P$ , where  $I$  is the matrix with  $\mathcal{X}$  on the diagonal elements and  $\emptyset$  elsewhere, we have  $(P^T A P)(P^T x) = \lambda(P^T x)$ , which proves the statement.

Let us give some examples of eigenvalues  $\lambda$  with relative eigenvectors  $v_\lambda$ . It is worth noting that eigenvalues (and eigenvectors) show behaviors which are not possible in  $\mathbb{R}^n$ . For instance, we can have several eigenvalues associated with the same eigenvector, or  $\sigma(A) = \mathcal{P}(\mathcal{X})$  (see also following Prop. 6).

*Example 2.* Suppose  $\mathcal{X} = (\infty, \infty)$ . Consider the matrix

$$\begin{bmatrix} \emptyset & \emptyset \\ (17, 28] & 13 \end{bmatrix} \rightarrow \lambda = \emptyset, v_\lambda = (x_1, \emptyset)^T, \forall x_1 \subset \mathcal{X}.$$

By Prop. 4,  $\lambda = \emptyset$  is an eigenvalue as the union of the elements of the first column is not the complete set  $\mathcal{X}$ .

*Example 3.* As a second example, consider the matrix  $\begin{bmatrix} [3, 5] & \mathcal{X} \\ \mathcal{X} & 4 \end{bmatrix}$ . In this case, direct computation shows that, for every  $x \subset \mathcal{X}$ ,  $\lambda \subset x$  is an eigenvalue with (an) associated eigenvector  $v_\lambda = (x, x)^T$ .

In the remainder of the section, we focus on Boolean matrices whose elements can only be in the set  $\{\emptyset, \mathcal{X}\}$ .

*Proposition 5.* Let  $A$  be a  $n \times n$  matrix s.t.  $[A]_{ij} \in \{\emptyset, \mathcal{X}\}$ .  $\lambda = \mathcal{X}$  is not an eigenvalue of  $A$  if, and only if, there exists a permutation that brings  $A$  in strictly lower form.

*Proof 5.* (Sufficiency)

Suppose the existence of a permutation matrix  $P$  s.t.  $A' \stackrel{\text{def}}{=} P^T A P$  is strictly lower triangular. Then, we need to prove that there exists no vector  $x \neq \emptyset$  s.t.  $A'x = \mathcal{X}x = x$ . This trivially holds due to the form of matrix  $A'$ . Direct computation of the previous equation gives

$$\begin{aligned} \emptyset &= x_1, \\ a'_{2,1} \cap x_1 &= x_2, \\ a'_{3,1} \cap x_1 \cup a'_{3,2} \cap x_2 &= x_3, \\ &\vdots \\ a'_{n,1} \cap x_1 \cup \dots \cup a'_{n,n-1} \cap x_{n-1} &= x_n. \end{aligned}$$

The only vector that solves the system of equations is  $x = \emptyset$ , which means that  $\lambda = \mathcal{X}$  can not be a Boolean eigenvector  $A$ .

(Necessity)

We need to prove that, if  $\lambda = \mathcal{X}$  is not an eigenvalue of  $A$ , then there exists a permutation that brings  $A$  in strictly lower triangular form.

Note that  $\mathcal{X}$  is an eigenvalue of  $A$  if, and only if,  $A$  has a fixed point. So, let us start imposing that the vector  $w = (\mathcal{X}, \dots, \mathcal{X})^T$  is not a fixed point. Then, if  $A$  has not an empty row, the scalar product between every row of  $A$  and  $w$  is  $\mathcal{X}$ , and therefore  $w$  would be a fixed point. Then suppose that the  $i$ -th row of  $A$  is made of emptysets. We can now apply to  $A$  a permutation that exchanges the  $i$ -th row with the first one, and then exchanges the  $i$ -th with the first column. In this way we obtain a matrix where the first row is empty.

By induction, suppose that there exists a permutation matrix  $P$  such that  $P^T A P$  has the form

$$\begin{pmatrix} \emptyset & \dots & \emptyset \\ a'_{2,1} & \emptyset & \dots & \emptyset \\ \vdots & \ddots & \vdots & \vdots \\ a'_{i,1} & \dots & a'_{i,i-1} & \emptyset & \dots & \emptyset \\ a'_{i+1,1} & \dots & \dots & a'_{i+1,n} \\ \vdots & \ddots & \vdots & \vdots \\ a'_{n,1} & \dots & \dots & a'_{n,n} \end{pmatrix},$$

and consider the vector  $v = (\emptyset, \dots, \emptyset, \mathcal{X}, \dots, \mathcal{X})^T$ , where the first  $i$  rows are null.  $v$  is not a fixed point of  $P^T A P$  only if there exists  $j > i$  such that the  $j$ -th row of  $P^T A P$  has the form  $(a'_{j,1}, \dots, a'_{j,i}, \emptyset, \dots, \emptyset)$ . We can now apply to  $P^T A P$  a permutation that exchanges the  $j$ -th row with the  $i$ -th one, and then exchanges the  $j$ -th with the  $i$ -th column. The inductive step is complete because we obtain a matrix of the form

$$\begin{pmatrix} \emptyset & \dots & \emptyset \\ a'_{2,1} & \emptyset & \dots & \emptyset \\ \vdots & \ddots & \vdots & \vdots \\ a'_{i+1,1} & \dots & a'_{i+1,i} & \emptyset & \dots & \emptyset \\ a'_{i+2,1} & \dots & \dots & a'_{i+2,n} \\ \vdots & \ddots & \vdots & \vdots \\ a'_{n,1} & \dots & \dots & a'_{n,n} \end{pmatrix},$$

which concludes the proof.

*Proposition 6.* Let  $A = \{a_{i,j}\}$  be a  $n \times n$  matrix s.t.  $a_{i,j} \in \{\emptyset, \mathcal{X}\}$ . If  $\mathcal{X} \notin \sigma(A)$ , then

$$\sigma(A) = \mathcal{P}(\mathcal{X}) \setminus \mathcal{X}.$$

*Proof 6.* By Remark 2 we can suppose that  $A$  is strictly triangular. The eigenvector  $v = (\emptyset, \dots, \emptyset, \mathcal{C}(\lambda))^T$  is associated with eigenvalue  $\lambda$ . In fact  $\emptyset = A v = \lambda v = \emptyset$  with  $\lambda \neq \emptyset, v \neq \emptyset$ .

### 3.2 Contractive Iterative Maps

A notion of contraction for a generic Boolean iterative map  $F$  w.r.t. the vector distance  $\mathcal{D}$  can be defined as

*Definition 8.* (Contractive Iterative Map). An application  $F : \mathcal{P}(\mathcal{X})^n \rightarrow \mathcal{P}(\mathcal{X})^n$  is said to be *contractive* w.r.t. the vector distance  $\mathcal{D} : \mathcal{P}(\mathcal{X})^n \times \mathcal{P}(\mathcal{X})^n \rightarrow \mathcal{P}(\mathcal{X})^n$  if

- $\mathcal{X} \notin \sigma(B(F))$ , and,
  - $\exists M \in \mathbb{B}^{n \times n} : \forall x, y \in \mathcal{P}(\mathcal{X})^n,$
- $$\mathcal{D}(F(x), F(y)) \subseteq M \mathcal{D}(x, y). \quad (7)$$

where Eq. (7) is called *contraction inequality*.

*Remark 3.* From Prop. 5, it is worth noting that an application  $F$  is said to be contractive if there exists a permutation matrix  $P \in \{\emptyset, \mathcal{X}\}^{n \times n}$  s.t.  $P^T B(F) P$  is strictly triangular.

The main result can be stated in the following

*Theorem 1.*  $F$  is contractive w.r.t. the vector distance  $\mathcal{D}$  if, and only if, there exists a positive integer  $q$  such that  $F^q$  is a constant application.

*Proof 7.* (Sufficiency) Being  $F$  contractive,  $\mathcal{X} \notin \sigma(B(F))$  and  $B(F)$  up to transformation  $P^T B(F) P$ , where  $P$  is a permutation matrix, is strictly triangular. Therefore, it is ensured the existence of a positive integer  $q \leq n$  s.t.  $(B(F))^q = \emptyset$ . It also holds that

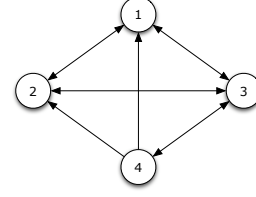


Fig. 1. Example of communication Graph.

$$\begin{aligned} \emptyset &\subseteq B(F^q) = B(F \cdots F) \subseteq \\ &\subseteq B(F) \cdots B(F) = (B(F))^q. \end{aligned}$$

Then, it must hold  $B(F^q) = \emptyset$ . This implies that the application  $F^q$  is independent of  $x$  and it guarantees the existence of a point  $\xi \in \mathcal{P}(\mathcal{X})^n$  s. t.  $F^q(x) = \xi$ , for every  $x \in \mathcal{P}(\mathcal{X})$ . Moreover, the rest iteration of  $F$  gives  $F^{q+1}(\xi) = F^q(F(\xi)) = \xi$ , as  $F^q$  is constant, but also  $F^{q+1}(\xi) = F(F^q(\xi)) = F(\xi)$ , which implies that  $F(\xi) = \xi$ . Then,  $\xi$  is a fixed point of  $F$ , and it is also unique. Indeed, suppose by absurd the existence of a second fixed point  $\eta \in \mathcal{P}(\mathcal{X})^n$  of  $F$ , with  $\eta \neq \xi$ . Then, we have

$$\begin{aligned} \emptyset &\subseteq \mathcal{D}(\xi, \eta) = \mathcal{D}(F(\xi), F(\eta)) \subseteq B(F) \mathcal{D}(\xi, \eta) \subseteq \\ &\subseteq \dots \subseteq (B(F))^q \mathcal{D}(\xi, \eta) = \emptyset, \end{aligned}$$

as  $(B(F))^q = \emptyset$ . Then,  $\mathcal{D}(\xi, \eta) = \emptyset$ , and  $\xi = \eta$ , which is a contradiction.

(Necessity)

Suppose that  $F^q$  is a constant application. Then  $B(F^q) = B(F)^q = \emptyset$ . It follows that  $B(F)$  does not admit a fixed point. Then,  $\mathcal{X} \notin \sigma(B(F))$  and, by Proposition 5,  $B(F)$  must be strictly triangular up to a transformation  $P^T B(F) P$ , where  $P$  is a permutation matrix. We have that  $q \leq n$  and  $F$  is contractive.

*Remark 4.* If  $\xi$  is the unique fixed point, iterations of  $F$  starting from any initial point  $x(0) \in \mathcal{P}(\mathcal{X})^n$  converge to  $\xi$  in at most  $q$  steps.

## 4. APPLICATION

Consider a network of 4 agents, that are able to communicate according to the graph depicted in Fig. 1. To begin with, suppose that the agents exchange binary values, i.e.  $\mathbb{B} = \{0, 1\}$  and thus  $\tilde{0} = 0$  and  $\tilde{1} = 1$ , and that update their states according to the binary iterative rule

$$\begin{cases} x_1(t+1) = u(t), \\ x_2(t+1) = u(t), \\ x_3(t+1) = x_1(t) x_2(t), \\ x_4(t+1) = x_2(t) x_3(t) + x_1(t) x_2(t) x_3(t), \end{cases} \quad (8)$$

whose incidence matrix is

$$B(F) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

Being  $B(F)$  strictly lower triangular and by Prop. 5, it is straightforward to verify that  $\lambda = \tilde{1} = 1 \notin \sigma(B(F))$ . Then, Theorem 1 allows us to conclude that the iterative rule in Eq. 8 is contractive. Since  $\bar{x} = \mathbb{1}u$  is a (unique) fixed point of the map, iterations will eventually converge to  $\bar{x}$  irrespectively of the initial point  $x(0)$ .

Let us now consider the binary iterative map

$$\begin{cases} x_1(t+1) = u(t) + x_2(t), \\ x_2(t+1) = x_1(t)x_3(t) + x_1(t)x_2(t), \\ x_3(t+1) = x_1(t)x_2(t) + x_4(t), \\ x_4(t+1) = x_1(t) + x_2(t)x_3(t)x_4(t), \end{cases} \quad (9)$$

whose incidence matrix is

$$B(F) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

By using Prop. 5, it is easy to show that  $\lambda = \tilde{1} = 1$  is an eigenvalue of  $B(F)$ . Indeed, there exists no permutation matrix  $P$  s.t. the matrix  $P^T B P$  is strictly lower triangular. Starting from the initial value  $x_a(0) = (0, 1, 1, 0)^T$ , the system will converge to  $\mathbb{1}u$ , whereas starting from  $x_b(0) = (0, 1, 1, 1)^T$ , the system will not reach a steady state, and enter into a cycle composed of the states  $(u, 1, 1, 1)^T$  and  $(1, u, 1, 1)^T$ .

Consider now two examples where agents exchange set-valued data ( $\mathbb{B} = \mathcal{X}$ ,  $\tilde{0} = \emptyset$ , and  $\tilde{1} = \mathcal{X}$ ). Assume the continuous domain to be  $\mathcal{X} = [0, \infty)$ . For convenience, suppose that  $u \in [10, 20]$  and that agents update their states according to the Boolean iterative map

$$\begin{cases} x_1(t+1) = u(t) \cup [10, 20], \\ x_2(t+1) = x_1(t), \\ x_3(t+1) = x_1(t) \cup x_2(t), \\ x_4(t+1) = (x_1(t) \cup x_2(t)) \cap x_3(t). \end{cases} \quad (10)$$

As its incidence matrix  $B(F)$  is strictly lower triangular,  $\lambda = \tilde{1} = \mathcal{X} \notin \sigma(B(F))$ . According to Theorem 1, we can conclude that the Boolean iterative map in Eq. 10 is contractive. This implies that the point  $\bar{x} = \mathbb{1}(u \cup [10, 20])$  is the unique fixed point of the system, and every iterations of it will converge to that point.

A variation of the above map is represented by the system

$$\begin{cases} x_1(t+1) = (x_1(t) \cap x_3(t)) \cup [10, 20], \\ x_2(t+1) = x_1(t), \\ x_3(t+1) = x_1(t) \cup x_2(t), \\ x_4(t+1) = (x_1(t) \cup x_2(t)) \cap x_3(t), \end{cases} \quad (11)$$

whose incidence matrix is

$$B(F) = \begin{pmatrix} \mathcal{X} & \emptyset & \mathcal{X} & \emptyset \\ \mathcal{X} & \emptyset & \emptyset & \emptyset \\ \mathcal{X} & \mathcal{X} & \mathcal{X} & \emptyset \\ \mathcal{X} & \mathcal{X} & \mathcal{X} & \emptyset \end{pmatrix}.$$

By Prop. 5 is it easy to show that  $\lambda = \tilde{1} = \mathcal{X}$  is an eigenvalue of  $B(F)$ , and that the Boolean iterative map in Eq. 11 is not contractive. Indeed, starting from the initial point  $x_a(0) = (\emptyset, [2, 5], [7, 10], [8, 12])^T$ , the system will converge to  $\mathbb{1}[10, 20]$ , whereas, starting from  $x_b(0) = ([30, 40], [40, 50], [20, 60], [30, 90])^T$ , the system will converge to  $\mathbb{1}([10, 20] \cup [30, 40])$ .

## 5. CONCLUSION

The paper focused on the convergence of information in distributed systems, where agents share data that represents logical values or sets. Based on a notion of contractivity for Boolean dynamical systems, a necessary and sufficient condition ensuring the global convergence toward a unique equilibrium point is presented. Such contractive systems and the corresponding iterative maps

are good candidates for achieving consensus on Boolean information, and their design will be developed in future work.

## 6. ACKNOWLEDGMENT

This work has been partially supported by the European Commission with contract FP7-IST-2008-224428 ‘‘CHAT - Control of Heterogeneous Automation Systems: Technologies for scalability, reconfigurability and security’’, and with contract number FP7-2007-2-224053 CONET, the ‘‘Cooperating Objects Network of Excellence’’.

## REFERENCES

- J. Cortés, S. Martínez, T. Karataş, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.
- M. Di Marco, A. Garulli, A. Giannitrapani, and A. Vicino. Simultaneous localization and map building for a team of cooperating robots: a set membership approach. *IEEE Trans. on Robotics and Automation*, 19(2):238–249, 2003.
- A. Fagiolini, M. Pellinacci, G. Valenti, G. Dini, and A. Bicchi. Consensus based Distributed Intrusion Detection for Multi Robot Systems. In *IEEE Intl. Conf. on Robotics and Automation*, pages 120–127, 2008a.
- A. Fagiolini, E.M. Visibelli, and A. Bicchi. Logical Consensus for Distributed Network Agreement. In *IEEE Conf. on Decision and Control*, pages 5250–5255, 2008b.
- A. Fagiolini, S. Martini, and A. Bicchi. Set-valued consensus for distributed clock synchronization. *IEEE Conf. on Automation Science and Engineering*, 2009.
- J.A. Fax and R.M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, Sept. 2004.
- P. Frasca, R. Carli, F. Fagnani, and S. Zampieri. Average consensus on networks with quantized communication. *Intl. Journal of Robust and Nonlinear Control*, 2008.
- L. Greco, M. Gaeta, and B. Piccoli. Deployment of sensors in a network-like environment. In *IEEE Conf. on Decision and Control, 2008*, pages 4257–4262, 2008.
- A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.
- K. Marzullo. Maintaining the time in a distributed system: An example of a loosely-coupled distributed service. *Dissert. Abstracts Intl. Part B: Science and Engineering*, 46(1), 1985.
- D. Mills. Internet time synchronization: the network time protocol. *IEEE Trans. on Comm.*, 39(10), Oct 1991.
- R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215, 2007.
- F. Robert. Théorèmes de Perron–Frobenius et Stein–Rosenberg Booleens. *Linear Algebra and its Applications*, 19:237–250, 1978.
- F. Robert. Itérations sur des ensembles finis convergence d’automates cellulaires contractants. *Linear Algebra and its Applications*, 29:393–412, 1980.
- NV Subrahmanyam. Boolean vector spaces. I. *Mathematische Zeitschrift*, 83(5):422–433, 1964.