

Advances in Complex Systems  
© World Scientific Publishing Company

## ALGORITHM COMPARING BINARY STRING PROBABILITIES IN COMPLEX STOCHASTIC BOOLEAN SYSTEMS USING INTRINSIC ORDER GRAPH

LUIS GONZÁLEZ

*Department of Mathematics, University of Las Palmas de Gran Canaria,  
Campus Universitario de Tafira, 35017 Las Palmas de Gran Canaria, Spain  
luisglez@dma.ulpgc.es*

Received (received date)

Revised (revised date)

This paper deals with a special kind of complex systems which depend on an arbitrary (and usually large) number  $n$  of random Boolean variables. The so-called complex stochastic Boolean systems often appear in many different scientific, technical or social areas. Clearly, there are  $2^n$  binary states associated to such a complex system. Each one of them is given by a binary string  $u = (u_1, \dots, u_n) \in \{0, 1\}^n$  of  $n$  bits, which has a certain occurrence probability  $\Pr\{u\}$ . The behavior of a complex stochastic Boolean system is determined by the current values of its  $2^n$  binary  $n$ -tuple probabilities  $\Pr\{u\}$  and by the ordering between pairs of them. Hence, the intrinsic order graph provides an useful representation of these systems by displaying (scaling) the  $2^n$  binary  $n$ -tuples which are ordered with decreasing probabilities of occurrence. The intrinsic order reduces the complexity of the problem from the exponential ( $2^n$  binary  $n$ -tuples) to the linear ( $n$  Boolean variables). For any fixed binary  $n$ -tuple  $u$ , this paper presents a new, simple algorithm enabling rapid, elegant determination of all the binary  $n$ -tuples  $v$  with occurrence probabilities less than or equal to (greater than or equal to)  $\Pr\{u\}$ . This algorithm is closely related to the lexicographic (truth-table) order in  $\{0, 1\}^n$ , and it is illustrated through the connections (paths) in the intrinsic order graph.

*Keywords:* Complex stochastic Boolean system; binary string probabilities; lexicographic order; intrinsic order; algorithm.

### 1. Introduction

Many different phenomena arising from diverse scientific fields could be considered as a complex stochastic Boolean system (CSBS). By a CSBS we mean a complex system depending on an arbitrary (and, in practice, usually large) number  $n$  of random Boolean variables. That is, the  $n$  basic variables of the system are assumed to be stochastic (i.e., non-deterministic) and to take only two possible values: 0, 1. Using the statistical terminology, the mathematical modeling of such systems begins with the simplest multi-dimensional discrete distribution used in Statistics, i.e. the  $n$ -dimensional Bernoulli distribution [11]. It is the well known fact that this distribution is consisting on  $n$  random variables,  $x_1, \dots, x_n$ , which only take two

2 *Luis González*

possible values, zero or one, with probabilities

$$\Pr \{x_i = 1\} = p_i, \quad \Pr \{x_i = 0\} = 1 - p_i \quad (1 \leq i \leq n).$$

Of course, the sample space, i.e., the set of elementary events of this distribution is the set of  $n$ -tuples of 0s and 1s

$$\{0, 1\}^n = \{(u_1, \dots, u_n) \mid u_i \in \{0, 1\}, 1 \leq i \leq n\}.$$

In the following, the marginal Bernoulli variables  $x_1, \dots, x_n$  are mutually independent, so that the probability of occurrence of each one of the  $2^n$  binary  $n$ -tuples,  $u = (u_1, \dots, u_n) \in \{0, 1\}^n$ , can be computed as the product

$$\Pr \{(u_1, \dots, u_n)\} = \prod_{i=1}^n \Pr \{x_i = u_i\} = \prod_{i=1}^n p_i^{u_i} (1 - p_i)^{1-u_i}, \quad (1)$$

that is,  $\Pr \{(u_1, \dots, u_n)\}$  is the product, taken over all components  $u_i$  ( $1 \leq i \leq n$ ), of factors  $p_i$  or  $1 - p_i$  if  $u_i = 1$  or  $u_i = 0$ , respectively. Throughout this paper, the binary  $n$ -tuples  $(u_1, \dots, u_n)$  of 0s and 1s will be also called binary strings or system binary states.

As an example of stochastic Boolean system, we can consider an application depending on a certain number  $n$  of basic components. These applications, as well as other CSBSs taken from diverse scientific areas (Climatology, Cybernetics, Economy, etc.), have been widely studied in Reliability Theory and Risk Analysis in order to estimate the system unavailability using different probabilistic and/or algebraic techniques. See, e.g., [1, 7, 8, 9] for more details about these techniques and for many different real world cases of CSBSs.

As one specific application, we can consider an accumulator system of a pressured water reactor in a nuclear power plant, taken from [12] and analyzed in [6]. This technical system depends on  $n = 83$  independent basic components  $x_1, \dots, x_{83}$ . Assuming that  $x_i = 1$  if component  $i$  fails,  $x_i = 0$  otherwise, then the failure and working probabilities of the component  $i$  will be  $\Pr \{x_i = 1\} = p_i$  and  $\Pr \{x_i = 0\} = 1 - p_i$ , respectively. Thus, this accumulator system can be considered as a CSBS where each one of its  $2^{83}$  system binary states (i.e., binary 83-tuples  $(u_1, \dots, u_{83}) \in \{0, 1\}^{83}$ ) describes the current situation of its 83 basic components (failing or working). For instance, the binary 83-tuple

$$u = (\underbrace{1, \dots, 1}_{13}, \underbrace{0, \dots, 0}_{70}) \quad (2)$$

represents the system state for which the 13 first components fail, while the 70 last components work. Moreover, the occurrence probability of  $u$  can be computed using Eq. (1) as follows

$$\Pr \{u\} = \prod_{i=1}^{13} p_i \prod_{i=14}^{70} (1 - p_i).$$

The behavior of each given stochastic Boolean system depends on the ordering between the current values of the  $2^n$  binary  $n$ -tuple probabilities  $\Pr\{u\}$ . At the same time, the ordering between the occurrence probabilities,  $\Pr\{u\}$ ,  $\Pr\{v\}$ , of two given binary  $n$ -tuples,  $u, v$ , depends, through Eq. (1), on the  $n$  current parameters  $\{p_i\}_{i=1}^n$  of the associated Bernoulli distribution, as the following simple example shows.

**Example 1.** For  $n = 3, u = (0, 1, 1), v = (1, 0, 0)$ , using Eq. (1) we have

$$\Pr\{u\} = (1 - p_1)p_2p_3, \quad \Pr\{v\} = p_1(1 - p_2)(1 - p_3)$$

and then we can consider the following two cases

$$(a) \ p_1 = 0.1, \ p_2 = 0.2, \ p_3 = 0.3 : \quad \Pr\{u\} = 0.054 < \Pr\{v\} = 0.056,$$

$$(b) \ p_1 = 0.2, \ p_2 = 0.3, \ p_3 = 0.4 : \quad \Pr\{u\} = 0.096 > \Pr\{v\} = 0.084.$$

This example seems to suggest us that to order the  $2^n$  binary  $n$ -tuple probabilities, we need to compute first all these probabilities using Eq. (1). Of course, this procedure is not feasible due to its exponential nature. That is, since the number of binary  $n$ -tuples  $u \in \{0, 1\}^n$  is  $2^n$  then the problem of computing the  $2^n$  corresponding probabilities  $\Pr\{u\}$  has exponential complexity with respect to  $n$ . To overcome this obstacle, in [2, 6] we have established a simple, positional criterion that allows us to compare (to order) *a priori* two given binary string probabilities,  $\Pr\{u\}$ ,  $\Pr\{v\}$ , without computing them (i.e., without using Eq. (1)), simply looking at the relative positions of their 0s and 1s. This positional criterion (which will be described in the next section) is called *intrinsic* order criterion (IOC), because it is completely independent of the parameters  $p_i$  ( $1 \leq i \leq n$ ), and it only (i.e., *intrinsically*) depends on the positions of 0s and 1s in the binary  $n$ -tuples.

The only hypothesis that we require to apply the IOC to a CSBS is that its  $n$  parameters  $\{p_i\}_{i=1}^n$  must be less than or equal to one half and they must be arranged in non-decreasing order, i.e.,

$$0 < p_1 \leq \dots \leq p_n \leq 0.5. \quad (3)$$

Fortunately, this assumption, although essential for theoretical results, is not restrictive for practical applications. Moreover, in this way, to compare binary string probabilities, we drastically reduce the computational cost by avoiding computation, via Eq. (1), of the  $2^n$  binary  $n$ -tuple probabilities  $\Pr\{u\}$ . More precisely, instead of computing and ordering the  $2^n$  binary  $n$ -tuple probabilities  $\Pr\{u\}$  we only need to order the  $n$  parameters  $p_i$ , as shown in Eq. (3), and to apply IOC for rapidly comparing pairs of binary string probabilities. In other words, we would be able to understand the behavior of the whole CSBS from the behavior of its  $n$  basic components,  $x_1, \dots, x_n$ , reducing the complexity of the problem from the exponential to the linear!

It is also important to mention that not all pairs  $(u, v)$  of binary strings satisfy IOC. Hence, there are two possibilities:

- (i) When  $(u, v)$  ( $(v, u)$ ) satisfies IOC, then we can assure that  $\Pr\{u\} \geq \Pr\{v\}$  ( $\Pr\{v\} \geq \Pr\{u\}$ ) without computing none of these two probabilities.
- (ii) When none of the two pairs  $(u, v)$ ,  $(v, u)$  satisfies IOC, then we can assure that sometimes  $\Pr\{u\} \geq \Pr\{v\}$  and sometimes  $\Pr\{u\} < \Pr\{v\}$  depending on the current values of the  $n$  basic probabilities  $p_1, \dots, p_n$ . So, in this case we must use Eq. (1) to compute and compare  $\Pr\{u\}$  and  $\Pr\{v\}$ . The example 1 corresponds to this second possibility.

In this context, for any CSBS and for any given fixed binary  $n$ -tuple  $u$ , the main goal of this paper is to provide a new algorithm for rapidly determining the set  $C^u$  ( $C_u$ ) of all binary  $n$ -tuples  $v$  whose occurrence probabilities are always less than or equal to (greater than or equal to) the occurrence probability of  $u$ , i.e.,

$$C^u = \{v \in \{0, 1\}^n \mid \Pr\{u\} \geq \Pr\{v\}\}, \quad (4)$$

$$C_u = \{v \in \{0, 1\}^n \mid \Pr\{u\} \leq \Pr\{v\}\}. \quad (5)$$

Of course, the interest of this fast determination algorithm for both the theoretical and practical analysis of CSBSs is clear. Let us stress out that, by its own nature, our algorithm will exclusively use pairs  $(u, v)$  or  $(v, u)$  of binary strings satisfying IOC. The reason is that the algorithm determines the sets  $C^u$  and  $C_u$ , i.e. those binary strings  $v$  such that the respective inequalities  $\Pr\{u\} \geq \Pr\{v\}$  and  $\Pr\{u\} \leq \Pr\{v\}$ , always (intrinsically) hold. Hence, we are always in the above defined case (i) and we do not need to use Eq. (1). Neither  $\Pr\{u\}$  needs to be computed to obtain the sets  $C^u$  and  $C_u$ ! Hence, as explained above, the exponential complexity ( $2^n$  probabilities  $\Pr\{v\}$ ) is reduced to the linear complexity ( $n$  parameters  $p_i$ ). For instance, for the above mentioned accumulator system, the direct determination of the sets  $C^u$  and  $C_u$  (without our algorithm) would be computationally extremely expensive: there are  $2^{83}$  binary 83-tuples. Furthermore, for a CSBS with  $n = 203$  basic components (a reasonable quantity in practice; see, e.g., [1, 7, 8]) this direct determination would be just physically impossible: there are  $2^{203}$  binary 203-tuples. Think that the age of the Universe from the Big-Bang to this instant is approximately  $2^{203}$  Planck times. Let us recall that the Planck time is the smallest possible measurement of time that has any physical meaning (1 Planck time  $\approx 5.391 \times 10^{-44}$  seconds).

Now, let us recall that the lexicographic order defined on the set  $\{0, 1\}^n$  of binary  $n$ -tuples, i.e. the usual truth-table order, coincides with the natural ordering between the decimal representations of the binary  $n$ -tuples  $u \in \{0, 1\}^n$ . This well-known fact is illustrated for the set  $\{0, 1\}^3$  by Table 1, where the left column gives the decimal representation  $u_{(10)}$  of each binary 3-tuple  $u$ . Throughout this paper, the decimal numbering of a binary string  $u$  is denoted by the symbol  $u_{(10)}$ . We use this symbol, instead of the more usual notation  $u_{10}$ , to avoid confusions with the 10-th component  $u_{10}$  of the binary string  $u$ .

The algorithm that is proposed in this paper is closely related to the lexicographic (truth-table) order. This tight, elegant relationship between our algorithm

and the lexicographic order will be illustrated by the connections (paths) in the intrinsic order graph.

Table 1. Lexicographic ordering in  $\{0, 1\}^3$  corresponds to decimal numbering of binary 3-tuples.

$u_{(10)}$	$u = (u_1, u_2, u_3)$
0	(0, 0, 0)
1	(0, 0, 1)
2	(0, 1, 0)
3	(0, 1, 1)
4	(1, 0, 0)
5	(1, 0, 1)
6	(1, 1, 0)
7	(1, 1, 1)

This paper has been organized as follows. In Section 2, we present overview of intrinsic ordering enabling non-specialists to follow the paper without difficulties. Sections 3 and 4 are respectively devoted to obtain the sets  $C^u$  and  $C_u$ , for each given binary  $n$ -tuple  $u$ . Finally, conclusions are presented in Section 5.

## 2. Background on Intrinsic Ordering

### 2.1. Basic notation

First, the basic notation is presented in the following definition.

**Definition 1.** For all binary  $n$ -tuple  $u = (u_1, \dots, u_n) \in \{0, 1\}^n$

- (i) The Hamming weight (or, simply, weight) of  $u$  is the number of 1 bits in  $u$

$$w_H(u) = \sum_{i=1}^n u_i.$$

- (ii) The decimal numbering of  $u$  is its representation in the decimal number system

$$u_{(10)} = (u_1, \dots, u_i, \dots, u_n)_{(10)} = \sum_{i=1}^n 2^{n-i} u_i.$$

- (iii) The lexicographic order in  $\{0, 1\}^n$  is the usual truth-table order between binary  $n$ -tuples. It coincides with the natural ordering between the decimal representations of the binary  $n$ -tuples, i.e.,  $u_{(10)} \leq v_{(10)}$  (see Table 1).
- (iv) The vector of positions of 1s of  $u$  is the vector of positions of its 1 bits, numbered from the right-most position 0 to the left-most position  $n - 1$ . That is, the  $n$  positions in the binary  $n$ -tuple  $u$  are labeled with the corresponding exponents,

6 *Luis González*

in the powers of 2, used when converting  $u$  from binary to decimal. For all  $n$ -tuple  $u$  with weight  $w_H(u) = m$  ( $1 \leq m \leq n$ ), we denote

$$u = [i_1, \dots, i_m]_n \Leftrightarrow u_{(10)} = 2^{i_1} + \dots + 2^{i_m}, \quad 0 \leq i_1 < \dots < i_m \leq n-1.$$

- (v) The complementary  $n$ -tuple of a binary  $n$ -tuple  $u$  is obtained by changing its 0s by 1s and its 1s by 0s

$$u^c = (u_1, \dots, u_n)^c = (1 - u_1, \dots, 1 - u_n).$$

- (vi) The complementary set of a set  $S$  of binary  $n$ -tuples is the set of the complementary  $n$ -tuples of all the  $n$ -tuples of  $S$ . For all  $S \subseteq \{0, 1\}^n$

$$S^c = \{u^c \mid u \in S\}.$$

**Example 2.** For  $n = 6$  and  $u = (1, 0, 1, 0, 1, 1) \in \{0, 1\}^6$ , we have

$$w_H(u) = 4, \quad u_{(10)} = 2^0 + 2^1 + 2^3 + 2^5 = 43, \quad u = [0, 1, 3, 5]_6, \quad u^c = (0, 1, 0, 1, 0, 0).$$

**Remark 1.** Throughout this paper, we shall denote any binary  $n$ -tuple  $u$ , indistinctly by its binary representation, by its decimal representation or by the vector of positions of its 1s, since each one of them clearly identifies the binary  $n$ -tuple. We use the symbol “ $\equiv$ ” to denote the equivalence between these different representations of the same binary  $n$ -tuple, i.e.,

$$(u_1, \dots, u_n) \equiv u_{(10)} \equiv [i_1, \dots, i_m]_n, \quad \text{e.g., } (1, 0, 1, 0, 1, 1) \equiv 43 \equiv [0, 1, 3, 5]_6.$$

Note that the sum of arbitrary two complementary binary  $n$ -tuples is always same

$$\begin{aligned} u + u^c &= (u_1, \dots, u_n) + (1 - u_1, \dots, 1 - u_n) = \underbrace{(1, \dots, 1)}_n \\ &\equiv 2^0 + \dots + 2^{n-1} = 2^n - 1 = u_{(10)} + u_{(10)}^c, \end{aligned} \quad (6)$$

e.g., Example 2 gives

$$\begin{aligned} u + u^c &= (1, 0, 1, 0, 1, 1) + (0, 1, 0, 1, 0, 0) = (1, 1, 1, 1, 1, 1) \\ &\equiv 2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 = 63 = 2^6 - 1. \end{aligned}$$

**Remark 2.** As is well-known, the lexicographic order in the set  $\{0, 1\}^n$  (Definition 1-(iii)) can be easily characterized as follows. Let  $u = (u_1, \dots, u_n), v = (v_1, \dots, v_n)$  be two any binary  $n$ -tuples. Then  $u$  precedes  $v$  in the lexicographic (truth-table) order, i.e.,  $u_{(10)} < v_{(10)}$ , if and only if the left-most column of matrix

$$M_v^u = \begin{pmatrix} u_1 & \dots & u_n \\ v_1 & \dots & v_n \end{pmatrix}$$

different from  $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$  and  $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ , is  $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ , or equivalently, if and only if the left-most column of matrix

$$M_u^v = \begin{pmatrix} v_1 & \dots & v_n \\ u_1 & \dots & u_n \end{pmatrix}$$

different from  $\binom{0}{0}$  and  $\binom{1}{1}$ , is  $\binom{1}{0}$  (see Table 1).

**Example 3.** Let  $n = 7$  and  $u = (1, 0, 1, 0, 1, 1, 1)$ ,  $v = (1, 0, 1, 1, 0, 1, 0) \in \{0, 1\}^7$ . Then, according to Remark 2,  $u$  precedes  $v$  in the lexicographic (truth-table) order, i.e.,  $u_{(10)} = 87 < v_{(10)} = 90$ , because the left-most column of matrix

$$M_v^u = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

different from  $\binom{0}{0}$  and  $\binom{1}{1}$ , is (its fourth column)  $\binom{0}{1}$ , or equivalently, because the left-most column of matrix

$$M_u^v = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

different from  $\binom{0}{0}$  and  $\binom{1}{1}$ , is (its fourth column)  $\binom{1}{0}$ .

**Remark 3.** In particular, as an immediate consequence of the matrix description of the lexicographic ordering in  $\{0, 1\}^n$  (Remark 2), we derive that, for all  $n \geq 1$ , the first and last binary  $n$ -tuples, respectively, in the lexicographic (truth-table) order are

$$\underbrace{(0, \dots, 0)}_n \text{ and } \underbrace{(1, \dots, 1)}_n,$$

respectively (see Table 1).

## 2.2. Intrinsic order

As we mentioned in Section 1, the evaluation of the  $2^n$  binary  $n$ -tuple probabilities  $\Pr\{u\}$  of a CSBS using Eq. (1) to order them is not in general possible due to the exponential nature of the problem. To overcome this obstacle, the next theorem (see [2, 6]) provides us with a simple criterion that allows to order two given binary string probabilities,  $\Pr\{u\}$ ,  $\Pr\{v\}$ , without computing them, simply looking at the relative positions of their 0s and 1s. Recall that in Section 1 we mentioned that this positional criterion is called the intrinsic order criterion (IOC).

**Theorem 1. (The intrinsic order theorem)** *Let  $x_1, \dots, x_n$  be  $n$  independent Bernoulli variables, with parameters  $p_i = \Pr\{x_i = 1\}$  ( $1 \leq i \leq n$ ) satisfying:*

$$0 < p_1 \leq \dots \leq p_n \leq \frac{1}{2}. \quad (7)$$

*Then, the probability of the  $n$ -tuple  $u = (u_1, \dots, u_n) \in \{0, 1\}^n$  is intrinsically greater than or equal to the probability of the  $n$ -tuple  $v = (v_1, \dots, v_n) \in \{0, 1\}^n$  (that is, for all set of parameters  $\{p_i\}_{i=1}^n$  such that (7)) if, and only if, the matrix*

$$M_v^u = \begin{pmatrix} u_1 & \dots & u_n \\ v_1 & \dots & v_n \end{pmatrix}$$

either has no  $\binom{1}{0}$  columns, or for each  $\binom{1}{0}$  column in  $M_v^u$  there exists (at least) one corresponding preceding  $\binom{0}{1}$  column (IOC).

**Remark 4.** In the following, we assume that the parameters  $p_i$  always satisfy condition (7). Note that, fortunately for the analysis of CSBSs, this hypothesis is not restrictive in practice. Indeed, if  $p_i > 0.5$  for some  $i$ , then we only need to consider the variable  $\bar{x}_i = 1 - x_i$  instead of  $x_i$ . Next, we order the  $n$  Bernoulli variables by increasing order of their probabilities.

**Remark 5.** The  $\binom{0}{1}$  column preceding to each  $\binom{1}{0}$  column is not required to be necessarily placed at the immediately previous position, but just at previous position.

**Remark 6.** The term *corresponding* used in Theorem 1, has the following meaning. For each two  $\binom{1}{0}$  columns in matrix  $M_v^u$ , there must exist (at least) two *different*  $\binom{0}{1}$  columns preceding to each other. In other words, IOC can be reformulated as follows. Either matrix  $M_v^u$  has no  $\binom{1}{0}$  columns, or for each given  $\binom{1}{0}$  column  $C_0^1$  in  $M_v^u$  the number of  $\binom{0}{1}$  columns preceding  $C_0^1$  is strictly greater than the number of  $\binom{1}{0}$  columns preceding  $C_0^1$ .

The matrix condition stated by Theorem 1 naturally leads to define the following order relation between the binary  $n$ -tuples of 0s and 1s. We use the standard abbreviations “iff” and “s.t.” to denote the mathematical expressions “if and only if” and “such that”, respectively.

**Definition 2.** For all  $u, v \in \{0, 1\}^n$

$u \succeq v$  iff  $\Pr\{u\} \geq \Pr\{v\}$  for all set  $\{p_i\}_{i=1}^n$  s.t. (7) iff  $M_v^u$  satisfies IOC.

The order relation “ $\preceq$ ” defined on  $\{0, 1\}^n$  is called *intrinsic* order, because it *intrinsically* depends on the positions of 0s and 1s, and it is independent of the values of the parameters  $\{p_i\}_{i=1}^n$  satisfying hypothesis (7). Now, we can rewrite the two possibilities (i) & (ii), described in Section 1, in a more precise way as follows.

For each given pair  $(u, v)$  of binary  $n$ -tuples two cases are possible:

- (i) When  $u \succeq v$  (resp.  $v \succeq u$ ), i.e., when  $M_v^u$  (resp.  $M_u^v$ ) satisfies IOC, then we can assure that  $\Pr\{u\} \geq \Pr\{v\}$  (resp.  $\Pr\{v\} \geq \Pr\{u\}$ ) for all set of parameters  $\{p_i\}_{i=1}^n$  satisfying (7), without computing the probabilities  $\Pr\{u\}$ ,  $\Pr\{v\}$ . In this case we say that  $u$  and  $v$  are *comparable* by intrinsic order.
- (ii) When neither  $u \succeq v$ , nor  $v \succeq u$ , i.e., when neither matrix  $M_v^u$ , nor matrix  $M_u^v$  satisfies IOC, then sometimes  $\Pr\{u\} \geq \Pr\{v\}$  and sometimes  $\Pr\{u\} < \Pr\{v\}$  depending on the current values of the parameters  $\{p_i\}_{i=1}^n$  satisfying (7). So, in this case we must use Eq. (1) to compute and compare  $\Pr\{u\}$  and  $\Pr\{v\}$ . In this case we say that  $u$  and  $v$  are *incomparable* by intrinsic order.

The binary relation “ $\preceq$ ” is a partial order relation on the set  $\{0, 1\}^n$ . In the following, we shall denote the corresponding partially ordered set (poset, for short)



by  $I_n = (\{0, 1\}^n, \preceq)$ . We refer the reader to [10] for more details about posets. The following examples illustrate Theorem 1 and Definition 2.

**Example 4.** For  $n = 3$ ,  $(0, 1, 1) \not\preceq (1, 0, 0)$  and  $(1, 0, 0) \not\preceq (0, 1, 1)$  because

$$\text{neither } \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \text{ nor } \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

satisfies IOC (Remark 6). Hence, the ordering between the occurrence probabilities  $\Pr\{(0, 1, 1)\}$  and  $\Pr\{(1, 0, 0)\}$  depends on the current values of the basic probabilities  $0 < p_1 \leq p_2 \leq p_3 \leq 0.5$ , as Example 1 has shown.

**Example 5.** For  $n = 4$ ,  $(0, 0, 1, 1) \succeq (1, 1, 0, 0)$  because matrix

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

satisfies IOC (Remark 5). Hence,

$$\Pr\{(0, 0, 1, 1)\} \geq \Pr\{(1, 1, 0, 0)\}, \text{ i.e.,}$$

$$(1 - p_1)(1 - p_2)p_3p_4 \geq p_1p_2(1 - p_3)(1 - p_4) \text{ for all } 0 < p_1 \leq p_2 \leq p_3 \leq p_4 \leq 0.5.$$

**Example 6.** For  $n = 83$  and for the accumulator system mentioned in Section 1

$$u = \underbrace{(0, \dots, 0)}_{38}, \underbrace{(1, \dots, 1)}_{45} \succeq \underbrace{(0, \dots, 0)}_{30}, \underbrace{(1, \dots, 1)}_{48}, \underbrace{(0, \dots, 0)}_5 = v$$

because matrix

$$\begin{pmatrix} 0 \dots 0 & 0 \dots 0 & 1 \dots 1 & 1 \dots 1 \\ \underbrace{0 \dots 0}_{30} & \underbrace{1 \dots 1}_8 & \underbrace{1 \dots 1}_{40} & \underbrace{0 \dots 0}_5 \end{pmatrix}$$

satisfies IOC (Remark 5). Hence,  $\Pr\{u\} \geq \Pr\{v\}$  for all  $0 < p_1 \leq \dots \leq p_{83} \leq 0.5$ .

**Example 7.** For all  $n \geq 1$ , and for all  $u = (u_1, \dots, u_n) \in \{0, 1\}^n$

$$0 \equiv (0, \dots, 0) \succeq (u_1, \dots, u_n) \succeq (1, \dots, 1) \equiv 2^n - 1,$$

because both matrices

$$\begin{pmatrix} 0 & \dots & 0 \\ u_1 & \dots & u_n \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} u_1 & \dots & u_n \\ 1 & \dots & 1 \end{pmatrix}$$

satisfy IOC, since they have no  $\binom{1}{0}$  columns. Hence, for all  $u = (u_1, \dots, u_n) \in \{0, 1\}^n$

$$\Pr\{(0, \dots, 0)\} \geq \Pr\{(u_1, \dots, u_n)\} \geq \Pr\{(1, \dots, 1)\},$$

for all  $0 < p_1 \leq \dots \leq p_n \leq 0.5$ . So, 0 and  $2^n - 1$  are the maximum and minimum elements, respectively, in the poset  $I_n$ .

To finish this subsection, we describe the tight relationship between the lexicographic and intrinsic orderings in the set  $\{0, 1\}^n$  of binary  $n$ -tuples. Suppose that  $u$  is intrinsically greater than  $v$ , i.e.,  $u \succ v$ . Thus, according to Definition 2, matrix

$M_v^u$  satisfies IOC. Consequently, the left-most column of  $M_v^u$ , different from  $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$  and  $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ , must be  $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$  because, otherwise, if this column is  $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$  then  $M_v^u$  does not satisfy IOC. But to affirm that the left-most column of  $M_v^u$ , different from  $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$  and  $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ , is  $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$  is equivalent to say that  $u$  precedes  $v$  in the truth-table (lexicographic) order, i.e.,  $u_{(10)} < v_{(10)}$  (Remark 2, Example 3). In this way, we have proved that the lexicographic order is a necessary condition for intrinsic order. More precisely

**Corollary 1.** *For all  $n \geq 1$  and for all  $u, v \in \{0, 1\}^n$*

$$u \succ v \Rightarrow u_{(10)} < v_{(10)}, \quad \text{i.e.,} \quad u \succeq v \Rightarrow u_{(10)} \leq v_{(10)}. \quad (8)$$

**Remark 7.** The converse of Corollary 1 is false, i.e., lexicographic order is not a sufficient condition for intrinsic order. Otherwise, intrinsic order and lexicographic order would be the same thing and this paper would have not any sense! The simplest counter-example that one can find is: For  $n = 3$ ,  $u = (0, 1, 1)$ ,  $v = (1, 0, 0)$

$$u_{(10)} = 3 < v_{(10)} = 4, \quad \text{but} \quad u \not\succeq v$$

as shown by (the left matrix in) Example 4 and as confirmed by Example 1.

We refer the reader to [2, 3] for more theoretical properties of the intrinsic order. For applications of the intrinsic order to the reliability analysis of technical systems and, in general, of any CSBS, see [5, 6].

### 2.3. *Intrinsic order graph*

Now, we present the graphical representation of the poset  $I_n = (\{0, 1\}^n, \succeq)$ . The usual representation of a poset is its Hasse diagram (see [10] for more details about these diagrams). Specifically, for the intrinsic partial order relation, the Hasse diagram is a directed graph (digraph, for short) whose vertices are the binary  $n$ -tuples of 0s and 1s, and whose edges connect each pair  $(u, v)$  of binary  $n$ -tuples whenever  $u$  is intrinsically greater than  $v$  and there are no other elements between them, i.e.,

$$u \succ v \quad \text{and there is no} \quad w \in \{0, 1\}^n \quad \text{s.t.} \quad u \succ w \succ v.$$

Consequently, there are two possibilities:

- (i) Each pair  $(u, v)$  of vertices connected in the Hasse diagram of  $I_n$  either by one edge, or by a path consisting on more than one edge means that  $u$  and  $v$  are comparable by intrinsic order, i.e.,  $u \succ v$  or  $v \succ u$ . This situation corresponds to case (i) in the previous subsection.
- (ii) On the contrary, each pair  $(u, v)$  of vertices non connected in the Hasse diagram of  $I_n$  means that  $u$  and  $v$  are incomparable by intrinsic order, i.e.,  $u \not\succeq v$  and  $v \not\succeq u$ . This situation corresponds to case (ii) in the previous subsection.

Moreover, according to the usual convention for Hasse diagrams, if  $u \succ v$  then  $u$  is drawn above  $v$ . The Hasse diagram of the poset  $I_n$  will be also called the intrinsic order graph for  $n$  variables. From now on, looking for a more comfortable and simple

notation, we shall denote the vertices (binary  $n$ -tuples) in the intrinsic order graph by their decimal numbering (Remark 1).

For small values of  $n$ , the Hasse diagram of  $I_n$  can be constructed by direct application of IOC. For instance, the Hasse diagram of  $I_1 = (\{0, 1\}, \preceq)$  is

$$\begin{array}{c} 0 \\ | \\ 1 \end{array} \quad (9)$$

because  $0 \succ 1$ , since matrix  $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$  satisfies IOC (it has no  $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$  columns!). However, for large values of  $n$  we need a more efficient method. For this purpose, in [4] we have developed an algorithm for iteratively building up the digraph of  $I_n$  from the digraph (9) of  $I_1$ , for all  $n \geq 2$ . The next theorem states this algorithm, which uses the decimal representation of the binary strings. See [4] for the proof and for additional properties of the intrinsic order graph.

**Theorem 2. (Iterative construction of  $I_n$  from  $I_1$ )** *For all  $n > 1$ , the digraph of  $I_n = \{0, \dots, 2^n - 1\}$  can be drawn simply by adding to the digraph of  $I_{n-1} = \{0, \dots, 2^{n-1} - 1\}$  its isomorphic copy  $2^{n-1} + I_{n-1} = \{2^{n-1}, \dots, 2^n - 1\}$ . This addition must be performed placing the powers of 2 at consecutive levels of the Hasse diagram of  $I_n$ . Finally, the edges connecting one vertex  $u$  of  $I_{n-1}$  with the other vertex  $v$  of  $2^{n-1} + I_{n-1}$  are given by the set of vertex pairs*

$$\{(u, v) \equiv (u_{(10)}, 2^{n-2} + u_{(10)}) \mid 2^{n-2} \leq u_{(10)} \leq 2^{n-1} - 1\}.$$

In Figure 1, we illustrate the algorithm described by Theorem 2 with the intrinsic order graph for  $n = 1, 2, 3, 4$ . Of course, the digraphs of  $I_1, I_2, I_3, I_4$  can be also drawn substituting the decimal representations of their 2, 4, 8, 16 nodes or binary 1, 2, 3, 4-tuples, respectively, by their corresponding binary representations.

The following two examples respectively correspond to the above cases (i) & (ii).

- (i) Looking at the Hasse diagram or digraph of  $I_4$ , the most right one in Figure 1, we observe that the binary 4-tuples 3 and 12 are connected in the digraph (it does not mind if the connection is by one edge or by a path with length  $> 1$ ), and 3 is drawn above 12. Hence, they are comparable by intrinsic order. Using the binary representation  $3 \equiv (0, 0, 1, 1)$  and  $12 \equiv (1, 1, 0, 0)$ , this is in accordance with Example 5 where we have shown that  $(0, 0, 1, 1) \succ (1, 1, 0, 0)$ .
- (ii) Looking at the Hasse diagram or digraph of  $I_3$ , the third one from left to right in Figure 1, we observe that the binary 3-tuples 3 and 4 are non connected in the digraph. Hence, they are incomparable by intrinsic order. Using the binary representation  $3 \equiv (0, 1, 1)$  and  $4 \equiv (1, 0, 0)$ , this is in accordance with Example 4 where we have shown that  $(0, 1, 1) \not\prec (1, 0, 0)$  and  $(1, 0, 0) \not\prec (0, 1, 1)$ .

Also, we can confirm that for the four digraphs  $n = 1, 2, 3, 4$ , the maximum and minimum elements are 0 and  $2^n - 1 = 1, 3, 7, 15$ , respectively, as we have proved in Example 7. Finally, looking at any of the four Hasse diagrams, we can confirm

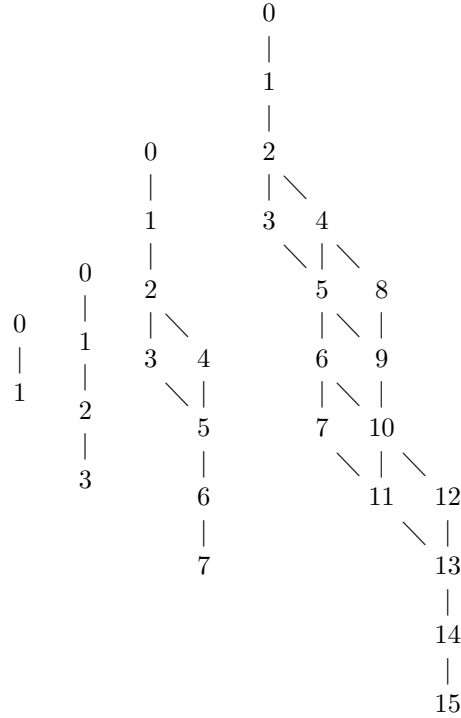


Fig. 1. The intrinsic order graph for  $n = 1, 2, 3, 4$  shown from the left side to the right one. Nodes connected either by one edge, or by a longer path are comparable, i.e.,  $u \succ v$  if  $u$  is connected with  $v$  and drawn above  $v$ . Non-connected nodes are incomparable, i.e.,  $u \not\succeq v$  and  $v \not\succeq u$ .

Corollary 1. Whenever  $u$  is written above  $v$  and they are connected (either by one edge, or by a longer path) then the decimal numbering of  $u$  is less than the decimal numbering of  $v$ . In other words, whenever  $u \succ v$  then  $u_{(10)} < v_{(10)}$ , as Corollary 1 has stated.

### 3. The Set $C^u$

Using the ideas and definitions presented in Section 2, we can reformulate our main goal stated in Section 1 and the corresponding Eq. (4) and Eq. (5), in a more precise and rigorous way, as follows. For any given binary  $n$ -tuple  $u$ , our purpose is to characterize, in an efficient way, the sets

$$C^u = \{v \in \{0, 1\}^n \mid \Pr\{u\} \geq \Pr\{v\}, \forall \{p_i\}_{i=1}^n \text{ s.t. (7)}\},$$

$$C_u = \{v \in \{0, 1\}^n \mid \Pr\{u\} \leq \Pr\{v\}, \forall \{p_i\}_{i=1}^n \text{ s.t. (7)}\}.$$

Let us rewrite it in a more compact form, according to Definition 2

$$C^u = \{v \in \{0, 1\}^n \mid u \succeq v\}, \tag{10}$$

$$C_u = \{v \in \{0, 1\}^n \mid u \preceq v\}. \quad (11)$$

Now, let us denote by  $L_u$  (resp.  $L^u$ ) the set of all binary  $n$ -tuples  $v$  whose decimal numbering are greater than or equal to (resp. less than or equal to) the decimal numbering of  $u$ , i.e.,

$$L_u = \{v \in \{0, 1\}^n \mid u_{(10)} \leq v_{(10)}\} = \{v \in \{0, 1\}^n \mid u_{(10)} \leq v_{(10)} \leq 2^n - 1\}, \quad (12)$$

$$L^u = \{v \in \{0, 1\}^n \mid u_{(10)} \geq v_{(10)}\} = \{v \in \{0, 1\}^n \mid u_{(10)} \geq v_{(10)} \geq 0\}. \quad (13)$$

Note that the second expression for the set  $L_u$  in Eq. (12) is an obvious consequence of the fact that the last binary  $n$ -tuple in the lexicographic (truth-table) order is (see Remark 3)

$$\underbrace{(1, \dots, 1)}_n \equiv \underbrace{(1, \dots, 1)}_n_{(10)} = 2^0 + \dots + 2^{n-1} = 2^n - 1.$$

Analogously, the second expression for the set  $L^u$  in Eq. (13) is an obvious consequence of the fact that the first binary  $n$ -tuple in the lexicographic (truth-table) order is (see Remark 3)

$$\underbrace{(0, \dots, 0)}_n \equiv \underbrace{(0, \dots, 0)}_n_{(10)} = 0.$$

In this section we address only the case of the set  $C^u$ , while the next section is devoted to the set  $C_u$ . First, taking into account Corollary 1, we have  $u \succeq v$  implies  $u_{(10)} \leq v_{(10)}$ , that is, we have

$$C^u = \{v \in \{0, 1\}^n \mid u \succeq v\} \subseteq \{v \in \{0, 1\}^n \mid u_{(10)} \leq v_{(10)}\} = L_u. \quad (14)$$

### 3.1. A relevant special case

From Eq. 14, we have the inclusion  $C^u \subseteq L_u$ . Moreover, sometimes, for some binary strings  $u$  these two sets coincide. Thus, before answering to the general question of identifying the set  $C^u$  for all binary strings  $u$ , we consider the following relevant special case/question: for which binary  $n$ -tuples  $u$ , the set inclusion  $C^u \subseteq L_u$  is in fact the set equality  $C^u = L_u$ ? In other words: can we identify the binary  $n$ -tuples  $u$  for which the set of binary  $n$ -tuples with smaller occurrence probabilities, is simply, exactly the set of binary  $n$ -tuples with larger decimal numbering? The following theorem answers to this question characterizing, by a surprisingly easy criterion, those binary strings  $u$  for which  $C^u = L_u$ .

**Theorem 3.** *Let  $n \geq 1$  and  $u = (u_1, \dots, u_n) \in \{0, 1\}^n$ . Then*

$$C^u = L_u \quad (15)$$

*if and only if  $u$  does not contain any 0 bit followed by two (or more) 1 bits, placed at consecutive or non consecutive positions, i.e.,  $u$  has the general pattern*

$$u = (\underbrace{1, \dots, 1}_p, \underbrace{0, \dots, 0}_q, \underbrace{1}_1, \underbrace{0, \dots, 0}_r), \quad p + q + r + 1 = n, \quad (16)$$

14 *Luis González*

where any (but not all) of the above four subsets of bits grouped together can be omitted.

**Proof.** *Necessary condition.* Suppose that  $u$  contains, at least, one 0 bit followed by two (or more) 1 bits. In other words,  $(u_1, \dots, u_n)$  contains, at least, one subsequence of three bits  $0 \dots 1 \dots 1$  (placed at consecutive or non consecutive positions), that is,

$$\exists 1 \leq i < j < k \leq n \text{ s.t. } u_i = 0, u_j = 1, u_k = 1.$$

Then defining  $v = (v_1, \dots, v_n)$  by

$$v_l = \begin{cases} 1 - u_l & \text{if } l = i, j, k, \\ u_l & \text{if } l \neq i, j, k, \end{cases}$$

we have

$$M_v^u = \begin{pmatrix} u_1 & \dots & u_{i-1} & 0 & \dots & 1 & \dots & 1 & u_{k+1} & \dots & u_n \\ u_1 & \dots & u_{i-1} & 1 & \dots & 0 & \dots & 0 & u_{k+1} & \dots & u_n \end{pmatrix}, \quad (17)$$

and then, obviously,  $u_{(10)} \leq v_{(10)}$ . However,  $u \not\leq v$  because matrix (17) contains exactly two  $\binom{1}{0}$  columns preceded by exactly one  $\binom{0}{1}$  column, and then it does not satisfies IOC (Remark 6). Hence,  $v \in L_u$  but  $v \notin C^u$ , so that the inclusion (14) is strict and then the equality (15) does not hold.

*Sufficient condition.* Conversely, suppose that  $u$  does not contain any 0 bit followed by two (or more) 1 bits. Then the general pattern of  $u$  is (16), where any (but not all) of the underlined groups of bits can be omitted. Let  $v \in L_u$ , i.e.,  $u_{(10)} \leq v_{(10)}$  and consider the matrix

$$M_v^u = \begin{pmatrix} 1 & \dots & 1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ v_1 & \dots & v_p & v_{p+1} & \dots & v_{p+q} & v_{p+q+1} & v_{p+q+2} & \dots & v_n \end{pmatrix}. \quad (18)$$

Let us prove that matrix (18) satisfies IOC. First, note that from the assumption  $u_{(10)} \leq v_{(10)}$  we can assure that  $v_1 = \dots = v_p = 1$ . Now, there are two possible cases:

- If  $v_{p+q+1} = 1$ , then matrix (18) has no  $\binom{1}{0}$  columns. Thus, (18) satisfies IOC.
- If  $v_{p+q+1} = 0$ , then taking again into account that  $u_{(10)} \leq v_{(10)}$ , we can assure that at least one of the components  $v_{p+1}, \dots, v_{p+q}$  of  $v$  is 1. Otherwise, the left-most column of matrix (18), different from  $\binom{0}{0}$  and  $\binom{1}{1}$ , would be its  $(p+q+1)$ -th column  $\binom{1}{0}$ , so that  $u_{(10)} > v_{(10)}$  which contradicts the assumption  $u_{(10)} \leq v_{(10)}$ . Consequently, (18) has only one  $\binom{1}{0}$  column (the  $(p+q+1)$ -th one), which is preceded by, at least, one  $\binom{0}{1}$  column (placed among the positions,  $p+1, \dots, p+q$ ). Thus, (18) satisfies IOC.

So, we have shown that for all  $v \in L_u$ , matrix (18) satisfies IOC, i.e.,  $u \succeq v$ , i.e.,  $v \in C^u$ . This proves the inclusion  $L_u \subseteq C^u$  which, together with the reciprocal inclusion (14), leads to the equality (15).  $\square$

**Example 8.** For  $n = 4$ , the 11 following binary 4-tuples

$$\begin{aligned} 0 \equiv (0, 0, 0, 0), \quad 1 \equiv (0, 0, 0, 1), \quad 2 \equiv (0, 0, 1, 0), \quad 4 \equiv (0, 1, 0, 0), \quad 8 \equiv (1, 0, 0, 0), \\ 9 \equiv (1, 0, 0, 1), \quad 10 \equiv (1, 0, 1, 0), \quad 12 \equiv (1, 1, 0, 0), \quad 13 \equiv (1, 1, 0, 1), \\ 14 \equiv (1, 1, 1, 0), \quad 15 \equiv (1, 1, 1, 1), \end{aligned} \quad (19)$$

and only them, have the pattern (16). In other words, none of them contains a 0 bit followed by two or three 1 bits. Therefore, Theorem 3 assures as that for each one of these “exclusive” 4-tuples  $u$ , the set  $C^u$  of 4-tuples  $v$  with occurrence probability less than or equal to the occurrence probability of  $u$  is exactly (and simply) the set  $L_u$  of 4-tuples  $v$  with decimal numbering greater than or equal to the decimal numbering of  $u$ . This fact can be illustrated by Figure 1 for each one of the binary 4-tuples given in (19). For instance, for  $u = (1, 0, 0, 1) \equiv 9$ , using Eq. (12), we have

$$C^u = L_u = \left\{ v \in \{0, 1\}^4 \mid u_{(10)} = 9 \leq v_{(10)} \leq 2^4 - 1 \right\} = \{9, 10, 11, 12, 13, 14, 15\}$$

and looking at the right-most digraph ( $n = 4$ ) in Figure 1, we observe that 10, 11, 12, 13, 14, 15 are exactly the vertices connected (comparable by intrinsic order) with 9 and drawn below 9.

**Example 9.** For the accumulator system with  $n = 83$  basic components described in Section 1, the binary 83-tuple  $u$  defined by Eq. (2) has the pattern (16), i.e.,  $u$  does not contain any 0 bit followed by two (or more) 1 bits. Therefore, Theorem 3 assures that  $C^u = L_u$ . Hence, using Eq. (12), we get that  $C^u$  is given by the closed interval

$$C^u = L_u = \left\{ v \in \{0, 1\}^{83} \mid u_{(10)} \leq v_{(10)} \leq 2^{83} - 1 \right\} = [2^{70} + \dots + 2^{82}, 2^{83} - 1],$$

since

$$u = \underbrace{(1, \dots, 1)}_{13} \underbrace{(0, \dots, 0)}_{70} \equiv 2^{70} + \dots + 2^{82} = u_{(10)}.$$

### 3.2. The general case

In the previous subsection, Theorem 3 has answered to the proposed question, namely the characterization of the set  $C^u$ , only for a special case: when  $u$  satisfies the positional condition (16). In this case, and only in this case,  $C^u$  simply coincides with  $L_u$ . Unfortunately, not all binary strings have the pattern (16). For these “unfortunate” cases, the characterization Theorem 3 assures us that the set equality  $C^u = L_u$  does not hold. In other words, the set inclusion  $C^u \subseteq L_u$  is strict, i.e.,  $C^u \subsetneq L_u$ . In this case we proceed as follows. To determine the set  $C^u$  we develop an algorithm for obtaining the set difference or complementary set

$$\begin{aligned} \overline{C^u} &= L_u - C^u = \{v \in \{0, 1\}^n \mid v \in L_u, v \notin C^u\} \\ &= \{v \in \{0, 1\}^n \mid u_{(10)} < v_{(10)}, u \not\preceq v\}, \end{aligned} \quad (20)$$

and then  $C^u$  will be immediately obtained as the set difference

$$C^u = L_u - \overline{C^u}. \quad (21)$$

For instance, the binary 3-tuple  $u = (0, 1, 1) \equiv 3$  contains a 0 bit followed by two 1 bits (it has not the pattern (16)). Hence,  $C^u \subsetneq L_u$ , and to obtain the set  $C^u$  we shall apply our algorithm to obtain the set (this particular application of the algorithm will be shown in Example 13)

$$\overline{C^u} = \left\{ v \in \{0, 1\}^3 \mid u_{(10)} = 3 < v_{(10)}, u \not\leq v \right\} = \{4\}$$

and taking into account that (see Eq. (12))

$$L_u = \left\{ v \in \{0, 1\}^3 \mid u_{(10)} = 3 \leq v_{(10)} \leq 2^3 - 1 \right\} = \{3, 4, 5, 6, 7\}$$

then, using Eq. (21), we immediately obtain

$$C^u = L_u - \overline{C^u} = \{3, 4, 5, 6, 7\} - \{4\} = \{3, 5, 6, 7\}.$$

This example can be confirmed by (the third digraph of) Figure 1. On one hand, we observe that the only node with decimal numbering greater than or equal to 3 and non connected with 3 is 4: the only element of  $\overline{C^u}$ . On the other hand, we observe that the nodes with decimal numbering greater than or equal to 3, connected with 3 and drawn below 3 are 3, 5, 6, 7: the elements of  $C^u$ . So, the problem is reduced to present an algorithm for obtaining the set  $\overline{C^u}$ . In the next theorem we present this algorithm, but first we need two auxiliary lemmas.

**Lemma 1.** *Let  $n \geq 1$  and  $u = (u_1, \dots, u_n), v = (v_1, \dots, v_n) \in \{0, 1\}^n$ . Then  $v \in \overline{C^u}$  if and only if the matrix*

$$M_v^u = \begin{pmatrix} u_1 & \dots & u_n \\ v_1 & \dots & v_n \end{pmatrix}$$

*can be split into the two following submatrices  $M_{v'}^{u'}$  and  $M_{v''}^{u''}$*

$$M_v^u = \left( M_{v'}^{u'} \mid M_{v''}^{u''} \right), \quad M_{v'}^{u'} = \begin{pmatrix} u_1 & \dots & u_r \\ v_1 & \dots & v_r \end{pmatrix}, \quad M_{v''}^{u''} = \begin{pmatrix} u_{r+1} & \dots & u_l & \dots & u_n \\ v_{r+1} & \dots & v_l & \dots & v_n \end{pmatrix}, \quad (22)$$

*where*

(i)  $M_{v'}^{u'}$  *satisfies IOC, it has the same number of  $\binom{1}{0}$  columns as  $\binom{0}{1}$  columns, its left-most column different from  $\binom{0}{0}$  and  $\binom{1}{1}$  is  $\binom{0}{1}$ , and its right-most (or its last) column is  $\binom{u_r}{v_r} = \binom{1}{0}$ .*

(ii)  $M_{v''}^{u''}$  *satisfies that its left-most column different from  $\binom{0}{0}$  and  $\binom{1}{1}$  is  $\binom{u_l}{v_l} = \binom{1}{0}$ .*

**Proof.** Using Eq. (20), we have that  $v \in \overline{C^u}$  if and only if  $u_{(10)} < v_{(10)}$  and  $u \not\leq v$ . On one hand, according to Remark 2,  $u_{(10)} < v_{(10)}$  if and only if the first or the left-most column of  $M_v^u$ , different from  $\binom{0}{0}$  and  $\binom{1}{1}$ , is  $\binom{0}{1}$ . Call this column  $\binom{u_f}{v_f}$ . On the other hand,  $u \not\leq v$  if and only if matrix  $M_v^u$  contains, at least, one  $\binom{1}{0}$  column without its corresponding preceding  $\binom{0}{1}$  column, i.e., avoiding IOC (Definition 2, Theorem



1). Let  $\binom{u_l}{v_l}$  be the left-most  $\binom{1}{0}$  column of matrix  $M_v^u$  avoiding IOC ( $3 \leq l \leq n$ ). Thus, according to Remark 6,  $\binom{u_l}{v_l}$  is exactly the left-most  $\binom{1}{0}$  column of matrix  $M_v^u$  for which the number of  $\binom{1}{0}$  columns and the number of  $\binom{0}{1}$  columns preceding it coincide. Let  $\binom{u_r}{v_r}$  be the right-most  $\binom{1}{0}$  column preceding  $\binom{u_l}{v_l}$  ( $2 \leq r < l$ ). That is, more briefly:  $\binom{u_r}{v_r}$  is defined as the right-most  $\binom{1}{0}$  column preceding the left-most  $\binom{1}{0}$  column  $\binom{u_l}{v_l}$  of matrix  $M_v^u$  that avoids IOC. From these definitions of the columns  $\binom{u_l}{v_l}$  and  $\binom{u_r}{v_r}$ , we conclude that  $v \in \overline{C^u}$  if and only in the split (22) of  $M_v^u$  the following conditions hold

- $M_{v'}^{u'}$  satisfies IOC, because  $\binom{u_l}{v_l}$  is the left-most  $\binom{1}{0}$  column of matrix  $M_v^u$  avoiding IOC and  $r < l$ .
- $M_{v'}^{u'}$  has the same number of  $\binom{1}{0}$  columns as  $\binom{0}{1}$  columns, because  $\binom{u_l}{v_l}$  is the left-most  $\binom{1}{0}$  column of matrix  $M_v^u$  for which the number of  $\binom{1}{0}$  columns and the number of  $\binom{0}{1}$  columns preceding it coincide, and all these columns belong to the submatrix  $M_{v'}^{u'}$ , because of the definition of  $\binom{u_r}{v_r}$ .
- The left-most column of  $M_{v'}^{u'}$  different from  $\binom{0}{0}$  and  $\binom{1}{1}$  is  $\binom{0}{1}$ , because the left-most column of  $M_v^u$ , different from  $\binom{0}{0}$  and  $\binom{1}{1}$ , is  $\binom{u_f}{v_f} = \binom{0}{1}$  and  $\binom{u_f}{v_f}$  necessarily belongs to  $M_{v'}^{u'}$ , since  $M_{v'}^{u'}$  satisfies IOC and then it must contain (at least) one  $\binom{0}{1}$  preceding  $\binom{u_r}{v_r} = \binom{1}{0}$  (Theorem 1).
- The right-most column of  $M_{v'}^{u'}$  is  $\binom{u_r}{v_r} = \binom{1}{0}$ , by construction.
- The left-most column of  $M_{v''}^{u''}$ , different from  $\binom{0}{0}$  and  $\binom{1}{1}$ , is  $\binom{u_l}{v_l} = \binom{1}{0}$ , by construction.  $\square$

The following example illustrates Lemma 1.

**Example 10.** Let  $n = 11$  and let

$$u = (0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1),$$

$$v = (0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1).$$

Then  $u$  and  $v$  satisfy the hypothesis  $v \in \overline{C^u}$  of Lemma 1. That is, on one hand,  $u_{(10)} < v_{(10)}$  because  $u_{(10)} = 343 < v_{(10)} = 417$  (Definition 1-(ii)) or, equivalently, because the left-most column of the matrix

$$M_v^u = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (23)$$

different from  $\binom{0}{0}$  and  $\binom{1}{1}$ , is (its fourth column)  $\binom{0}{1}$  (Remark 2).

On the other hand,  $u \not\leq v$  because the matrix (23) does not satisfy IOC. In fact, the left-most  $\binom{1}{0}$  column of  $M_v^u$  without its corresponding preceding  $\binom{0}{1}$  column, i.e. the left-most  $\binom{1}{0}$  column of  $M_v^u$  for which the number of  $\binom{1}{0}$  columns and the number of  $\binom{0}{1}$  columns preceding it coincide (in this example, this number is 2) is its ninth column  $\binom{u_l}{v_l} = \binom{u_9}{v_9} = \binom{1}{0}$ , i.e.,  $l = 9$ .

Hence, we can apply Lemma 1, where  $\binom{u_r}{v_r}$  was defined as the right-most  $\binom{1}{0}$  column of  $M_v^u$  preceding  $\binom{u_l}{v_l}$ , so that  $\binom{u_r}{v_r} = \binom{u_7}{v_7} = \binom{1}{0}$ , i.e.,  $r = 7$ .

18 *Luis González*

Thus, Lemma 1 assures that we can obtain the following split (22) of matrix (23), with  $r = 7$ ,  $l = 9$ ,  $n = 11$

$$M_v^u = \left( M_{v'}^{u'} \vdots M_{v''}^{u''} \right),$$

$$M_{v'}^{u'} = \begin{pmatrix} u_1 & \dots & u_7 \\ v_1 & \dots & v_7 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}, \quad M_{v''}^{u''} = \begin{pmatrix} u_8 & \dots & u_{11} \\ v_8 & \dots & v_{11} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (24)$$

where one can immediately confirm that the submatrices  $M_{v'}^{u'}$  and  $M_{v''}^{u''}$ , respectively, satisfy the conditions described in Lemma 1-(i) and Lemma 1-(ii), respectively.

**Lemma 2.** *Let  $n \geq 1$  and  $u = (u_1, \dots, u_n), v = (v_1, \dots, v_n) \in \{0, 1\}^n$ . Then  $v \in \overline{C^u}$  if and only if the matrix*

$$M_v^u = \begin{pmatrix} u_1 & \dots & u_n \\ v_1 & \dots & v_n \end{pmatrix}$$

can be split into the two following submatrices  $M_{v'}^{u'}$  and  $M_{v''}^{u''}$

$$M_v^u = \left( M_{v'}^{u'} \vdots M_{v''}^{u''} \right), \quad M_{v'}^{u'} = \begin{pmatrix} u_1 & \dots & u_r \\ v_1 & \dots & v_r \end{pmatrix}, \quad M_{v''}^{u''} = \begin{pmatrix} u_{r+1} & \dots & u_l & \dots & u_n \\ v_{r+1} & \dots & v_l & \dots & v_n \end{pmatrix},$$

where denoting by  $m$  the Hamming weight of  $u$  ( $0 < w_H(u) = m < n$ ), we have

$$w_H(u_1, \dots, u_r) = w_H(v_1, \dots, v_r) = m - (s - 1), \quad \text{with } 1 \leq s - 1 \leq n - r, \quad (25)$$

and denoting as follows the vectors of positions of the  $[m - (s - 1)]$  1 bits of the two following binary  $n$ -tuples

$$(u_1, \dots, u_r, \underbrace{0, \dots, 0}_{n-r}) \equiv [i_s, \dots, i_m]_n \equiv 2^{i_s} + \dots + 2^{i_m}, \quad i_s < \dots < i_m \leq n - 1, \quad (26)$$

$$(v_1, \dots, v_r, \underbrace{0, \dots, 0}_{n-r}) \equiv [j_s, \dots, j_m]_n \equiv 2^{j_s} + \dots + 2^{j_m}, \quad j_s < \dots < j_m \leq n - 1, \quad (27)$$

we have

(i')  $j_s \geq i_s, j_{s+1} \geq i_{s+1}, \dots, j_m \geq i_m$ .

(ii')  $M_{v'}^{u'}$  satisfies that its left-most column different from  $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ , and  $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$  is  $\begin{pmatrix} u_l \\ v_l \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ .

**Proof.** First, we note that the assumption  $0 < w_H(u) = m < n$  excludes the two extreme cases

$$w_H(u) = 0 \Leftrightarrow u = \underbrace{(0, \dots, 0)}_n \quad \text{and} \quad w_H(u) = n \Leftrightarrow u = \underbrace{(1, \dots, 1)}_n.$$

The reason is that these two  $n$ -tuples have the pattern (16) and then, they correspond to the special case studied in the previous subsection. In fact, using Theorem 3 and Eq. (12) we get (see Example 7 and Figure 1)

$$u = \underbrace{(0, \dots, 0)}_n : C^u = L_u = \{0, 1\}^n, \quad u = \underbrace{(1, \dots, 1)}_n : C^u = L_u = \{2^n - 1\}.$$

From Lemma 1, we know that  $v \in \overline{C^u}$  if and only if the matrix

$$M_v^u = \begin{pmatrix} u_1 & \dots & u_n \\ v_1 & \dots & v_n \end{pmatrix}$$

can be split as

$$M_v^u = \left( M_{v'}^{u'} \vdots M_{v''}^{u''} \right), \quad M_{v'}^{u'} = \begin{pmatrix} u_1 & \dots & u_r \\ v_1 & \dots & v_r \end{pmatrix}, \quad M_{v''}^{u''} = \begin{pmatrix} u_{r+1} & \dots & u_l & \dots & u_n \\ v_{r+1} & \dots & v_l & \dots & v_n \end{pmatrix},$$

where the submatrices  $M_{v'}^{u'}$  and  $M_{v''}^{u''}$  satisfy the conditions described in Lemma 1-(i) and Lemma 1-(ii), respectively. In particular, we know that in  $M_{v'}^{u'}$  the number of  $\binom{1}{0}$  columns is equal to the number of  $\binom{0}{1}$  columns. Then, denoting by  $n_0^1, n_1^0$  and  $n_1^1$  the number of  $\binom{1}{0}, \binom{0}{1}$  and  $\binom{1}{1}$  columns, respectively, of the submatrix  $M_{v'}^{u'}$ , we have

$$n_0^1 = n_1^0 \Leftrightarrow n_0^1 + n_1^1 = n_1^0 + n_1^1 \Leftrightarrow w_H(u_1, \dots, u_r) = w_H(v_1, \dots, v_r). \quad (28)$$

Moreover, let us denote by  $s - 1$  the number of 1 bits in the binary  $(n - r)$ -tuple  $(u_{r+1}, \dots, u_n)$ . We can assure that

$$1 \leq w_H(u_{r+1}, \dots, u_n) = s - 1 \leq n - r, \quad (29)$$

because this binary  $(n - r)$ -tuple has at least one 1 bit ( $u_l=1$ , since  $\binom{u_l}{v_l} = \binom{1}{0}$ ; see Lemma 1-(ii)) and at most  $(n - r)$  1 bits (because it has  $n - r$  components!). Hence, from the identity

$$w_H(u_1, \dots, u_r) + w_H(u_{r+1}, \dots, u_n) = w_H(u_1, \dots, u_n) = m,$$

and from (29) we get

$$w_H(u_1, \dots, u_r) = m - w_H(u_{r+1}, \dots, u_n) = m - (s - 1), \quad 1 \leq s - 1 \leq n - r,$$

and then, from this last expression and from Eq. (28) we derive Eq.(25).

Now, we prove that

$$j_s \geq i_s, \quad j_{s+1} \geq i_{s+1}, \quad \dots, \quad j_m \geq i_m, \quad (30)$$

where  $\{i_k\}_{k=s}^m$  and  $\{j_k\}_{k=s}^m$  are the sets of indices defined by Eq. (26) and Eq. (27), respectively. From Lemma 1-(i), we know that the submatrix  $M_{v'}^{u'}$  satisfies IOC. This is equivalent to say that for each  $\binom{1}{0}$  column in  $M_{v'}^{u'}$  there exists a corresponding preceding  $\binom{0}{1}$  column (Theorem 1). On one hand, the  $(m - s + 1)$  1 bits of the binary  $r$ -tuple  $(u_1, \dots, u_r)$  (with positions  $i_s < i_{s+1} < \dots < i_m$ ) correspond to the  $\binom{1}{0}$  and  $\binom{1}{1}$  columns of matrix  $M_{v'}^{u'}$ . On the other hand, the  $(m - s + 1)$  1 bits of the binary  $r$ -tuple  $(v_1, \dots, v_r)$  (with positions  $j_s < j_{s+1} < \dots < j_m$ ) correspond to the  $\binom{0}{1}$  and  $\binom{1}{1}$  columns of matrix  $M_{v'}^{u'}$ . Hence, since the positions of the 1 bits are, by convention, numbered in increasing order from right to left (see Definition 1-(iv)), then the matrix description IOC of  $M_{v'}^{u'}$  is equivalent to Eq. (30).

To finish the proof of (i'), we must prove that the first inequality in Eq. (30) is strict, i.e.,  $j_s \geq i_s$ . Using again Lemma 1-(i), we have that the right-most column

20 *Luis González*

of  $M_v^{u'}$  is  $\binom{u_r}{v_r} = \binom{1}{0}$ , i.e.,  $u_r = 1, v_r = 0$ . But, this is equivalent to say that the right-most 1-bit ( $v_i = 1$  for some  $i < r$ ) of the  $r$ -tuple  $(v_1, \dots, v_r)$  (i.e., the one placed at the position  $j_s$ ) precedes in  $M_v^{u'}$  the right-most 1-bit ( $u_r = 1$ ) of the  $r$ -tuple  $(u_1, \dots, u_r)$  (i.e., the one placed at the position  $i_s$ ). Since the positions of the 1 bits are numbered in increasing order from right to left, this is equivalent to say that  $j_s \gneq i_s$ .

Finally the assertion Lemma 2-(ii') is identical to the assertion Lemma 1-(ii).  $\square$

The following example, the same used to illustrate Lemma 1, also illustrates Lemma 2.

**Example 11.** Let  $n = 11$  and let

$$u = (0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1),$$

$$v = (0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1).$$

Then  $u$  and  $v$  satisfy the hypothesis  $v \in \overline{C^u}$  of Lemma 2 (the same hypothesis of Lemma 1), as we have shown in Example 10. Hence, Lemma 2 assures us that we can obtain the following split (22) of matrix (23), i.e., exactly the same split (24), with  $r = 7, l = 9, n = 11$ , of Example 10

$$M_v^u = \left( M_v^{u'} \vdots M_v^{u''} \right),$$

$$M_v^{u'} = \begin{pmatrix} u_1 & \dots & u_7 \\ v_1 & \dots & v_7 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}, \quad M_v^{u''} = \begin{pmatrix} u_8 & \dots & u_{11} \\ v_8 & \dots & v_{11} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Since  $m = w_H(u) = 6$ , using Eq. (29), we have

$$w_H(u_{r+1=8}, \dots, u_{n=11}) = w_H(0, 1, 1, 1) = s - 1 = 3 \Rightarrow s = 4$$

and thus, from Eq. (25), we get

$$w_H(u_1, \dots, u_{r=7}) = w_H(v_1, \dots, v_{r=7}) = m - (s - 1) = 6 - (4 - 1) = 3.$$

The binary  $n$ -tuples defined by (26) and (27) are

$$(u_1, \dots, u_7, \underbrace{0, \dots, 0}_{n-r=4}) = (0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0) \equiv [i_s, \dots, i_m]_{11}, \quad i_s < \dots < i_m,$$

$$(v_1, \dots, v_7, \underbrace{0, \dots, 0}_{n-r=4}) = (0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0) \equiv [j_s, \dots, j_m]_{11}, \quad j_s < \dots < j_m,$$

respectively, so that, the sets of indices

$$\begin{aligned} i_s = i_4 = 4, & \quad i_{s+1} = i_5 = 6, & \quad i_m = i_6 = 8 \\ j_s = j_4 = 5, & \quad j_{s+1} = j_5 = 7, & \quad j_m = j_6 = 8 \end{aligned}$$

satisfy Lemma 2-(i'), i.e.,  $j_s \gneq i_s, j_{s+1} \geq i_{s+1}, j_m \geq i_m$ .

Finally,  $M_v^{u''}$  obviously satisfies Lemma 2-(ii'), as shown in example 10.

Now, we can prove, using Lemma 2, the above mentioned algorithm for obtaining the set  $\overline{C^u}$ , which represents the binary strings by the vectors of positions of their 1 bits (Definition 1-(iv)). We call this algorithm “the intrinsically incomparable binary  $n$ -tuples algorithm”, because it provides us with the set  $\overline{C^u}$  of all the binary  $n$ -tuples  $v$  (with  $u_{(10)} < v_{(10)}$ ) such that  $u$  and  $v$  are incomparable by intrinsic order (see Eq. (20)).

Before establishing this theorem, let us briefly explain the intuitive idea underlying our algorithm. Our purpose is to express the set  $\overline{C^u}$ , using the decimal numbering, as a set union of half-closed intervals of consecutive natural numbers. For this purpose, we use the split (22) of matrix  $M_v^u$

$$M_v^u = \left( M_{v'}^{u'} : M_{v''}^{u''} \right), \quad M_{v'}^{u'} = \begin{pmatrix} u_1 & \dots & u_r \\ v_1 & \dots & v_r \end{pmatrix}, \quad M_{v''}^{u''} = \begin{pmatrix} u_{r+1} & \dots & u_l & \dots & u_n \\ v_{r+1} & \dots & v_l & \dots & v_n \end{pmatrix},$$

and its properties described in Lemmas 1 and 2. The basic idea to obtain the binary  $n$ -tuples  $v \in \overline{C^u}$  is that in this split the left sub-string  $(v_1, \dots, v_r)$  of  $v = (v_1, \dots, v_n) \in \overline{C^u}$ , i.e., the one placed at the left submatrix  $M_{v'}^{u'}$ , is always the same for all the  $n$ -tuples of the same half-closed interval, and it is defined by the conditions of Lemma 1-(i) or, equivalently, of Lemma 2-(i'). On the contrary, the right sub-string  $(v_{r+1}, \dots, v_n)$  of  $v = (v_1, \dots, v_n) \in \overline{C^u}$ , i.e., the one placed at the right submatrix  $M_{v''}^{u''}$ , takes all possible consecutive values in the truth-table (lexicographic) order, from  $\underbrace{(0, \dots, 0)}_{n-r}$  to the  $(n-r)$ -tuple immediately previous to  $(u_{r+1}, \dots, u_n)$ , according to the condition of Lemma 1-(ii) or Lemma 2-(ii').

**Example 12.** Let  $n = 5$  and let  $u = (0, 1, 1, 0, 0) \equiv 12$ . Then the intuitive idea is to use the split (22) of matrix  $M_v^u$ , according to Lemmas 1 and 2, with, e.g.  $r = 2, l = 3$ , as follows

$$M_v^u = \begin{pmatrix} u_1 & u_2 & u_3 & u_4 & u_5 \\ v_1 & v_2 & v_3 & v_4 & v_5 \end{pmatrix} = \left( M_{v'}^{u'} : M_{v''}^{u''} \right),$$

$$M_{v'}^{u'} = \begin{pmatrix} u_1 & u_2 \\ v_1 & v_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad M_{v''}^{u''} = \begin{pmatrix} u_3 & u_4 & u_5 \\ v_3 & v_4 & v_5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ v_3 & v_4 & v_5 \end{pmatrix},$$

where

$$(v_3, v_4, v_5) = (0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1)$$

and then

$$v = (v_1, v_2, v_3, v_4, v_5) = (1, 0, 0, 0, 0), (1, 0, 0, 0, 1), (1, 0, 0, 1, 0), (1, 0, 0, 1, 1)$$

that is, using the decimal representation, we get

$$\overline{C^u} = \{16, 17, 18, 19\} = [16, 20),$$

as we shall explain, more precisely, in Example 15.

22 *Luis González*

**Theorem 4. (The intrinsically incomparable binary  $n$ -tuples algorithm)**

Let  $n \geq 3$ . Let  $u \in \{0, 1\}^n$  with Hamming weight  $0 < w_H(u) = m < n$  and with decimal numbering and vector of positions of its 1s

$$u_{(10)} = 2^{i_1} + 2^{i_2} + \cdots + 2^{i_m} \equiv [i_1, i_2, \dots, i_m]_n, \quad 0 \leq i_1 < i_2 < \cdots < i_m \leq n-1. \quad (31)$$

Then the set  $\overline{C^u}$  can be obtained by the following algorithm:

*Step 1. Generate all the binary  $n$ -tuples with Hamming weight  $m$*

$$2^{j_1} + 2^{j_2} + \cdots + 2^{j_m} \equiv [j_1, j_2, \dots, j_m]_n, \quad (32)$$

where each one of the sequences  $\{j_1, j_2, \dots, j_m\}$  must satisfy the following conditions

$$\begin{aligned} j_1 = i_1, \quad \forall k = 2, \dots, m : j_k \geq i_k \quad \text{with} \quad j_k \not\equiv i_k \quad \text{for some} \quad k = 2, \dots, m, \\ 0 \leq j_1 < j_2 < \cdots < j_m \leq n-1. \end{aligned} \quad (33)$$

*Step 2. For each one of the  $n$ -tuples  $[j_1, j_2, \dots, j_m]_n$  generated in the step 1, call  $s$  ( $2 \leq s \leq m$ ) the smallest index  $k$  for which the inequality  $j_k \geq i_k$  is strict, i.e.,*

$$j_1 = i_1, \dots, j_{s-1} = i_{s-1}, \quad j_s \not\equiv i_s, \quad j_{s+1} \geq i_{s+1}, \dots, j_m \geq i_m. \quad (34)$$

Then consider the half-closed interval of consecutive natural numbers

$$[2^{j_s} + \cdots + 2^{j_m}, 2^{j_1} + \cdots + 2^{j_s} + \cdots + 2^{j_m}). \quad (35)$$

*Step 3. Finally, the set  $\overline{C^u}$  is given, using the decimal representation, by*

$$\overline{C^u} = \bigcup_{[j_1, \dots, j_m]_n} [2^{j_s} + \cdots + 2^{j_m}, 2^{j_1} + \cdots + 2^{j_s} + \cdots + 2^{j_m}), \quad (36)$$

where the above set union (of all half-closed intervals (35) constructed in the step 2) is extended over all binary  $n$ -tuples  $[j_1, \dots, j_m]_n$  generated in the step 1.

**Proof.** To prove this theorem is equivalent to prove the set equality (36). That is, we must prove that

$$v \in \overline{C^u} \Leftrightarrow v_{(10)} \in \bigcup_{[j_1, \dots, j_m]_n} [2^{j_s} + \cdots + 2^{j_m}, 2^{j_1} + \cdots + 2^{j_s} + \cdots + 2^{j_m}), \quad (37)$$

where the above set union is extended over all binary  $n$ -tuples  $[j_1, \dots, j_m]_n$  generated in the step 1. In other words, we shall prove that all the  $n$ -tuples of  $\overline{C^u}$  are generated by our algorithm (implication “ $\Rightarrow$ ” in (37)) and, conversely, that all the  $n$ -tuples generated by our algorithm belong to  $\overline{C^u}$  (implication “ $\Leftarrow$ ” in (37)).

Before proving the equivalence (37), we must stress out the following fact: all the intervals (35) generated in the step 2 are, by construction, pair-wise disjoint. Indeed, let

$$\begin{aligned} & [2^{j_s} + \cdots + 2^{j_m}, 2^{j_1} + \cdots + 2^{j_{s-1}} + 2^{j_s} + \cdots + 2^{j_m}), \\ & [2^{j'_s} + \cdots + 2^{j'_m}, 2^{j'_1} + \cdots + 2^{j'_{s-1}} + 2^{j'_s} + \cdots + 2^{j'_m}) \end{aligned} \quad (38)$$

be two different half-closed intervals generated in the step 2, that is, corresponding to the two different binary  $n$ -tuples

$$[j_1, \dots, j_{s-1}, j_s, \dots, j_m]_n \neq [j'_1, \dots, j'_{s-1}, j'_s, \dots, j'_m]_n$$

generated in the step 1. Since, from Eq. (34) we derive that the first  $(s-1)$  indices of both  $n$ -tuples must coincide, i.e.,

$$j_1 = i_1 = j'_1, \dots, j_{s-1} = i_{s-1} = j'_{s-1} \Rightarrow j_1 = j'_1, \dots, j_{s-1} = j'_{s-1},$$

then, we conclude that the sets of the last  $(m - (s-1))$  indices of both  $n$ -tuples must be different, i.e.,

$$\{j_s, \dots, j_m\} \neq \{j'_s, \dots, j'_m\},$$

and then the lower endpoints  $2^{j_s} + \dots + 2^{j_m}$  and  $2^{j'_s} + \dots + 2^{j'_m}$  of the two intervals (38) are different, while their upper endpoints are obtained by adding to the corresponding lower endpoints the same quantity  $2^{i_1} + \dots + 2^{i_{s-1}}$ . Hence the two intervals (38) are disjoint.

Let us prove the equivalence (37). From Lemma 2, we know that  $v \in \overline{C^u}$  if and only if the matrix

$$M_v^u = \begin{pmatrix} u_1 & \dots & u_n \\ v_1 & \dots & v_n \end{pmatrix}$$

can be split into the two following submatrices  $M_{v'}^{u'}$  and  $M_{v''}^{u''}$

$$M_v^u = \begin{pmatrix} M_{v'}^{u'} \\ M_{v''}^{u''} \end{pmatrix}, \quad M_{v'}^{u'} = \begin{pmatrix} u_1 & \dots & u_r \\ v_1 & \dots & v_r \end{pmatrix}, \quad M_{v''}^{u''} = \begin{pmatrix} u_{r+1} & \dots & u_l & \dots & u_n \\ v_{r+1} & \dots & v_l & \dots & v_n \end{pmatrix},$$

which satisfy all the conditions established in this lemma. In particular, Lemma 2-(ii') assures that the left-most column of the submatrix  $M_{v''}^{u''}$  different from  $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$  and  $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$  is  $\begin{pmatrix} u_l \\ v_l \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ . But, this is equivalent to affirm that (Remark 2)

$$\begin{aligned} (v_{r+1}, \dots, v_n)_{(10)} &\not\leq (u_{r+1}, \dots, u_n)_{(10)}, \text{ i.e.,} \\ 0 = \underbrace{(0, \dots, 0)}_{n-r} &\leq (v_{r+1}, \dots, v_n)_{(10)} \not\leq (u_{r+1}, \dots, u_n)_{(10)}, \text{ i.e.,} \\ \underbrace{(0, \dots, 0, 0, \dots, 0)}_{n-r} &\leq \underbrace{(0, \dots, 0, v_{r+1}, \dots, v_n)}_{(10)} \not\leq \underbrace{(0, \dots, 0, u_{r+1}, \dots, u_n)}_{(10)}, \end{aligned} \quad (39)$$

since the addition of any number  $r$  of 0 bits on the left of a binary string does not modify its decimal numbering. Now, adding

$$(v_1, \dots, v_r, \underbrace{0, \dots, 0}_{n-r})_{(10)}$$

to the three terms of (39), we get

$$(v_1, \dots, v_r, \underbrace{0, \dots, 0}_{n-r})_{(10)} \leq (v_1, \dots, v_r, v_{r+1}, \dots, v_n)_{(10)} \not\leq (v_1, \dots, v_r, u_{r+1}, \dots, u_n)_{(10)}. \quad (40)$$

24 *Luis González*

Now, on one hand, using Eq. (27) we have

$$(v_1, \dots, v_r, \underbrace{0, \dots, 0}_{n-r})_{(10)} = 2^{j_s} + \dots + 2^{j_m}, \quad (41)$$

and, on the other hand, using Eq. (26) and Eq. (31) we have

$$(u_1, \dots, u_r, \underbrace{0, \dots, 0}_{n-r})_{(10)} = 2^{i_s} + \dots + 2^{i_m},$$

$$(u_1, \dots, u_r, u_{r+1}, \dots, u_n)_{(10)} = 2^{i_1} + \dots + 2^{i_{s-1}} + 2^{i_s} + \dots + 2^{i_m}$$

and from the last two equations, we get

$$(u_{r+1}, \dots, u_n)_{(10)} = \underbrace{(0, \dots, 0)}_r, u_{r+1}, \dots, u_n)_{(10)} = 2^{i_1} \dots + 2^{i_{s-1}}. \quad (42)$$

Using (41) and (42), Eq. (40) is equivalent to

$$2^{j_s} + \dots + 2^{j_m} \leq v_{(10)} \leq 2^{i_1} + \dots + 2^{i_{s-1}} + 2^{j_s} + \dots + 2^{j_m}. \quad (43)$$

Now call

$$j_1 = i_1, \dots, j_{s-1} = i_{s-1}. \quad (44)$$

Note that, from Lemma 2-(i') we have  $j_s \geq i_s$ , and then  $j_{s-1} = i_{s-1} < i_s < j_s$ , and consequently we assure that  $j_{s-1} < j_s$ . Hence, Eq. (43) can be rewritten as

$$2^{j_s} + \dots + 2^{j_m} \leq v_{(10)} \leq 2^{j_1} + \dots + 2^{j_{s-1}} + 2^{j_s} + \dots + 2^{j_m}, \text{ i.e.,}$$

$$v_{(10)} \in [2^{j_s} + \dots + 2^{j_m}, 2^{j_1} + \dots + 2^{j_{s-1}} + 2^{j_s} + \dots + 2^{j_m}),$$

for some set of indices  $\{j_1, \dots, j_m\}$  satisfying (34), i.e., such that (see Eq. (44) and Lemma 2-(i'))

$$j_1 = i_1, \dots, j_{s-1} = i_{s-1}, \quad j_s \geq i_s, \quad j_{s+1} \geq i_{s+1}, \dots, j_m \geq i_m,$$

but this is equivalent to say that

$$v_{(10)} \in \bigcup_{[j_1, \dots, j_m]_n} [2^{j_s} + \dots + 2^{j_m}, 2^{j_1} + \dots + 2^{j_s} + \dots + 2^{j_m}),$$

where the above set union (of all half-closed intervals (35) constructed in the step 2) is extended over all binary  $n$ -tuples  $[j_1, \dots, j_m]_n$  generated in the step 1. The proof is concluded.  $\square$

**Remark 8.** Note that the exponential notation used in the intervals (35) does not imply exponential complexity of our algorithm. The union of these half-closed intervals provides us with the exact, clearly defined solution (36) to our problem, without computing the powers of 2.

**Remark 9.** The generation of the sequences  $[j_1, \dots, j_m]_n$  from the sequence  $[i_1, \dots, i_m]_n$  of the positions of the 1 bits in  $u$ , described in step 1 of the algorithm, has the following intuitive meaning. Step 1 generates all the sequences



$[j_1, \dots, j_m]_n$ , by moving from right to left one or more 1 bits of  $u$ , with the only exception of its right-most 1 bit, which must always stay at its original position in  $u$  (i.e.,  $j_1 = i_1$ ). Moreover, the index  $i_s$  defined in the step 2 of the algorithm is exactly the position of the right-most 1 bit in  $u$  that has been moved from right to left, while the index  $j_s$  is the new position of this 1 bit after being moved (i.e.,  $j_1 = i_1, \dots, j_{s-1} = i_{s-1}, j_s \geq i_s$ ).

**Remark 10.** Note that, according to Remark 9, to apply the step 1 of the algorithm, the binary string  $u$  must contain at least one 1 bit that could be moved from right to left. Since the right-most 1 bit of  $u$  never could be moved, then this means that  $u$  must contain, at least, one 0 bit followed by two (or more) 1 bits. In other words  $u$  must contain, at least, one subsequence  $0 \dots 1 \dots 1$ . Otherwise,  $u$  would have the pattern (16) and step 1 does not generate any sequence. So, in this case, the solution provided by our theorem/algorithm is  $\overline{C^u} = \emptyset$  and thus, using Eq. (21), we get

$$C^u = L_u - \overline{C^u} = L_u - \emptyset = L_u,$$

which is in accordance with Theorem 3!

In the following examples, we obtain the sets  $\overline{C^u}$  and  $C^u$  using Theorem 4 and Eq. (21), respectively. Both sets can be illustrated by the connections/paths in the intrinsic order graph, as explained in the paragraph immediately before the statement of Lemma 1 (see Figure 1 for  $n = 1, 2, 3, 4$ ). We begin with the simplest possible example for which the algorithm can be applied.

**Example 13.** For  $n = 3$  and  $u = (0, 1, 1) \equiv 3$ , we have

$$m = w_H(u) = 2, \quad u_{(10)} = 2^0 + 2^1 \equiv [i_1, i_2]_3 = [0, 1]_3.$$

Using the algorithm (Theorem 4), we obtain

- Step 1:  $[j_1, j_2]_3 = [0, 2]_3$ .
- Step 2:  $j_s = j_2 = 2, j_m = j_2 = 2 \rightarrow [2^2, 2^0 + 2^2] = [4, 5]$ .
- Step 3:  $\overline{C^u} = [4, 5] = \{4\}$ .

Now, using Eq. (12) we have

$$L_u = \left\{ v \in \{0, 1\}^3 \mid u_{(10)} = 3 \leq v_{(10)} \leq 2^3 - 1 \right\} = \{3, 4, 5, 6, 7\}$$

and, finally, from Eq. (21) we get

$$C^u = L_u - \overline{C^u} = \{3, 4, 5, 6, 7\} - \{4\} = \{3, 5, 6, 7\}.$$

**Example 14.** For  $n = 4$  and  $u = (0, 1, 1, 1) \equiv 7$ , we have

$$m = w_H(u) = 3, \quad u_{(10)} = 2^0 + 2^1 + 2^2 \equiv [i_1, i_2, i_3]_4 = [0, 1, 2]_4.$$

Using the algorithm (Theorem 4), we obtain

26 *Luis González*

- Step 1:  $[j_1, j_2, j_3]_4 = \begin{cases} \text{(i)} [0, 1, 3]_4, \\ \text{(ii)} [0, 2, 3]_4. \end{cases}$
- Step 2:  $\begin{cases} \text{(i)} j_s = j_3 = 3, j_m = j_3 = 3 \rightarrow [2^3, 2^0 + 2^1 + 2^3] = [8, 11], \\ \text{(ii)} j_s = j_2 = 2, j_m = j_3 = 3 \rightarrow [2^2 + 2^3, 2^0 + 2^2 + 2^3] = [12, 13]. \end{cases}$
- Step 3:  $\overline{C^u} = [8, 11] \cup [12, 13] = \{8, 9, 10, 12\}$ .

Now, using Eq. (12) we have

$$L_u = \left\{ v \in \{0, 1\}^4 \mid u_{(10)} = 7 \leq v_{(10)} \leq 2^4 - 1 \right\} = \{7, 8, 9, 10, 11, 12, 13, 14, 15\}$$

and, finally, from Eq. (21) we get

$$C^u = L_u - \overline{C^u} = \{7, 8, 9, 10, 11, 12, 13, 14, 15\} - \{8, 9, 10, 12\} = \{7, 11, 13, 14, 15\}.$$

**Example 15.** For  $n = 5$  and  $u = (0, 1, 1, 0, 0) \equiv 12$ , we have

$$m = w_H(u) = 2, \quad u_{(10)} = 2^2 + 2^3 \equiv [i_1, i_2]_5 = [2, 3]_5.$$

Using the algorithm (Theorem 4), we obtain

- Step 1:  $[j_1, j_2]_5 = [2, 4]_5$ .
- Step 2:  $j_s = j_2 = 4, j_m = j_2 = 4 \rightarrow [2^4, 2^2 + 2^4] = [16, 20]$ .
- Step 3:  $\overline{C^u} = [16, 20] = \{16, 17, 18, 19\}$ .

Now, using Eq. (12) we have

$$L_u = \left\{ v \in \{0, 1\}^5 \mid u_{(10)} = 12 \leq v_{(10)} \leq 2^5 - 1 \right\} = \{12, 13, \dots, 31\}$$

and, finally, from Eq. (21) we get

$$C^u = L_u - \overline{C^u} = \{12, 13, \dots, 31\} - \{16, 17, 18, 19\} = \{12, \dots, 15, 20, \dots, 31\}.$$

We finish with an example of a CSBS with 203 basic components. For this or for larger numbers of variables (appearing in practice), as we commented in Section 1, the time required for computing all the  $2^{203}$  binary string probabilities (at the theoretical/inaccessible speed of one Planck time for each one of these computations) would be larger than the age of the Universe. This example illustrates the importance of our algorithm.

**Example 16.** For  $n = 203$  and  $u = (\underbrace{1, \dots, 1}_{100}, 0, 1, 0, 1, 1, 0, \dots, 0)_{203}$ , we have

$$m = w_H(u) = 103,$$

$$\begin{aligned} u_{(10)} &= 2^{98} + 2^{99} + 2^{101} + 2^{103} + \dots + 2^{202} \\ &\equiv [i_1, \dots, i_{103}]_{203} = [98, 99, 101, 103, \dots, 202]_{203}. \end{aligned}$$

Using the algorithm (Theorem 4), we obtain

- Step 1:  $[j_1, \dots, j_{103}]_{203} = \begin{cases} \text{(i)} [98, 99, 102, 103, \dots, 202]_{203}, \\ \text{(ii)} [98, 100, 101, 103, \dots, 202]_{203}, \\ \text{(iii)} [98, 100, 102, 103, \dots, 202]_{203}, \\ \text{(iv)} [98, 101, 102, 103, \dots, 202]_{203}. \end{cases}$
- Step 2:  $\begin{cases} \text{(i)} j_s = j_3 = 102, j_m = j_{103} = 202 \rightarrow \\ \text{(ii)} j_s = j_2 = 100, j_m = j_{103} = 202 \rightarrow \\ \text{(iii)} j_s = j_2 = 100, j_m = j_{103} = 202 \rightarrow \\ \text{(iv)} j_s = j_2 = 101, j_m = j_{103} = 202 \rightarrow \end{cases}$
- $\begin{cases} [2^{102} + 2^{103} + \dots + 2^{202}, 2^{98} + 2^{99} + 2^{102} + 2^{103} + \dots + 2^{202}) = S_1, \\ [2^{100} + 2^{101} + 2^{103} + \dots + 2^{202}, 2^{98} + 2^{100} + 2^{101} + 2^{103} + \dots + 2^{202}) = S_2, \\ [2^{100} + 2^{102} + 2^{103} + \dots + 2^{202}, 2^{98} + 2^{100} + 2^{102} + 2^{103} + \dots + 2^{202}) = S_3, \\ [2^{101} + 2^{102} + 2^{103} + \dots + 2^{202}, 2^{98} + 2^{101} + 2^{102} + 2^{103} + \dots + 2^{202}) = S_4. \end{cases}$
- Step 3:  $\overline{C^u} = S_1 \cup S_2 \cup S_3 \cup S_4$ .

Now, using Eq. (12) we have

$$\begin{aligned} L_u &= \left\{ v \in \{0, 1\}^{203} \mid u_{(10)} = 2^{98} + 2^{99} + 2^{101} + 2^{103} + \dots + 2^{202} \leq v_{(10)} \leq 2^{203} - 1 \right\} \\ &= [2^{98} + 2^{99} + 2^{101} + 2^{103} + \dots + 2^{202}, 2^{203} - 1] \end{aligned}$$

and, finally, from Eq. (21) we get

$$C^u = L_u - \overline{C^u} = [2^{98} + 2^{99} + 2^{101} + 2^{103} + \dots + 2^{202}, 2^{203} - 1] - \bigcup_{1 \leq k \leq 4} S_k.$$

#### 4. The Set $C_u$

In this section, we characterize the set  $C_u$  of all binary  $n$ -tuples  $v$  whose occurrence probabilities are always (i.e., for all set of parameters  $\{p_i\}_{i=1}^n$  satisfying the non-restrictive hypothesis (7)) greater than or equal to the occurrence probability of  $u$ . First, from Corollary 1, we know that if  $u \preceq v$  then  $u_{(10)} \geq v_{(10)}$ . That is, using the notation introduced in (11) and (13), we have the following set inclusion, dual of the inclusion (14)

$$C_u = \{v \in \{0, 1\}^n \mid u \preceq v\} \subseteq \{v \in \{0, 1\}^n \mid u_{(10)} \geq v_{(10)}\} = L^u. \quad (45)$$

The results for the set  $C_u$  are analogous to the ones obtained in Section 3 for the set  $C^u$ . Moreover, the corresponding dual propositions for  $C_u$  can be proved in a similar way to the ones used for proving Theorems 3 and 4 for  $C^u$ . However, we shall proceed in an alternative, shorter way obtaining the results for  $C_u$  from the corresponding results already proved for  $C^u$ . For this purpose, we need the following technical lemma.

**Lemma 3. (Duality or Symmetry property)** *Let  $n \geq 1$  and  $u, v \in \{0, 1\}^n$ . Then*

$$(i) \quad v_{(10)} \leq u_{(10)} \Leftrightarrow v_{(10)}^c \geq u_{(10)}^c,$$

28 *Luis González*

(ii)  $v \preceq u \Leftrightarrow v^c \succeq u^c$ ,

where  $u^c$  and  $v^c$  are the complementary  $n$ -tuples of  $u$  and  $v$ , respectively (Definition 1 – (v)).

**Proof.**

(i) Using Eq. (6), we have

$$v_{(10)} \leq u_{(10)} \Leftrightarrow (2^n - 1) - v_{(10)} \geq (2^n - 1) - u_{(10)} \Leftrightarrow v_{(10)}^c \geq u_{(10)}^c.$$

(ii) The  $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ ,  $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ ,  $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$  and  $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$  columns in matrix  $M_v^u$  become  $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ ,  $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ ,  $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$  and  $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$  columns in matrix  $M_{u^c}^{v^c}$ , respectively (note that this second matrix is  $M_{u^c}^{v^c}$  and not  $M_v^{u^c}$ ). Hence, we have

$$v \preceq u \Leftrightarrow M_v^u \text{ satisfies IOC} \Leftrightarrow M_{u^c}^{v^c} \text{ satisfies IOC} \Leftrightarrow v^c \succeq u^c,$$

and this concludes the proof.  $\square$

#### 4.1. A relevant special case

Sometimes, for some binary  $n$ -tuples  $u$ , the inclusion (45) becomes the set identity  $C_u = L^u$ . These binary strings  $u$  satisfying this nice property are characterized by the following theorem, which is the dual of Theorem 3 because the 0s are changed by 1s and the 1s are changed by 0s in the corresponding positional criteria.

**Theorem 5.** *Let  $n \geq 1$  and  $u = (u_1, \dots, u_n) \in \{0, 1\}^n$ . Then*

$$C_u = L^u \tag{46}$$

*if and only if  $u$  does not contain any 1 bit followed by two (or more) 0 bits, placed at consecutive or non consecutive positions, i.e.,  $u$  has the general pattern*

$$u = (\underbrace{0, \dots, 0}_p, \underbrace{1, \dots, 1}_q, \underbrace{0}_1, \underbrace{1, \dots, 1}_r), \quad p + q + r + 1 = n, \tag{47}$$

*where any (but not all) of the above four subsets of bits grouped together can be omitted.*

**Proof.** Using Lemma 3, we have

$$\begin{aligned} C_u = L^u & \text{ iff } \{v \in \{0, 1\}^n \mid u \preceq v\} = \{v \in \{0, 1\}^n \mid u_{(10)} \geq v_{(10)}\} \text{ iff} \\ & \{v^c \in \{0, 1\}^n \mid u^c \succeq v^c\} = \left\{v^c \in \{0, 1\}^n \mid u_{(10)}^c \leq v_{(10)}^c\right\} \text{ iff} \\ & \{v \in \{0, 1\}^n \mid u^c \succeq v\} = \left\{v \in \{0, 1\}^n \mid u_{(10)}^c \leq v_{(10)}\right\} \text{ iff } C^{u^c} = L_{u^c} \end{aligned}$$

iff (apply Theorem 3 to  $u^c$ ) the  $n$ -tuple  $u^c$  does not contain any 0 bit followed by two (or more) 1 bits. Finally, this positional criterion for  $u^c$  is transformed into the corresponding positional criterion for  $u$ , changing the 0s by 1s and the 1s by 0s. In this way, the last assertion about  $u^c$  is equivalent to say that  $u$  does not contain

any 1 bit followed by two (or more) 0 bits, i.e.,  $u$  does not contain any subsequence of three bits  $1 \dots 0 \dots 0$  (placed at consecutive or non consecutive positions), i.e.,  $u$  has the general pattern given by Eq. (47).  $\square$

**Remark 11.** The symbol  $C^{u^c}$  used in the above proof (as well as in the next theorem) represents the set  $C^{(u^c)}$  and not the set  $(C^u)^c$ . This is in accordance with the usual convention for the exponential notation  $x^{y^z} = x^{(y^z)}$ . In other words,  $C^{u^c}$  is the set of binary  $n$ -tuples which are intrinsically less than or equal to the complementary  $n$ -tuple  $u^c$  of  $u$ .

**Example 17.** For  $n = 4$  the binary 4-tuple  $u = (1, 0, 1, 1) \equiv 11$  does not contain any 1 bit followed by two or three 0 bits, that is,  $u$  has the pattern given by the condition (47). Therefore, Theorem 5 assures that the set  $C_u$  of binary 4-tuples  $v$  with occurrence probability greater than or equal to the occurrence probability of  $u$  is exactly (and simply) the set  $L^u$  of 4-tuples  $v$  with decimal numbering less than or equal to the decimal numbering of  $u$ , i.e.,

$$C_u = L^u = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}.$$

This fact can be illustrated by looking at the right-most digraph ( $n = 4$ ) in Figure 1, where we observe that 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 are exactly the vertices connected (comparable by intrinsic order) with 11 and drawn above 11.

#### 4.2. The general case

Of course, not all the binary strings have the pattern given by the condition (47). For these cases, the characterization Theorem 5 assures us that the set equality  $C_u = L^u$  does not hold. In other words, the set inclusion  $C_u \subseteq L^u$  is strict, i.e.,  $C_u \subsetneq L^u$ . In such cases, using the symmetry (duality) property stated by Lemma 3, we can reduce (via complementary  $n$ -tuples) the problem of determining the sets  $C_u$  to the determination of the sets  $C^{u^c}$ , which has been completely solved in Section 3. This procedure is described precisely in the following theorem.

**Theorem 6.** For all  $n \geq 1$  and for all  $u \in \{0, 1\}^n$

$$C_u = [C^{u^c}]^c. \quad (48)$$

**Proof.** Using Definition 1-(vi) and Lemma 3-(ii), we get

$$v \in C_u \Leftrightarrow u \preceq v \Leftrightarrow u^c \succeq v^c \Leftrightarrow v^c \in C^{u^c} \Leftrightarrow v \in [C^{u^c}]^c, \quad (49)$$

as was to be shown.  $\square$

**Example 18.** For  $n = 4$  the binary 4-tuple  $u = (1, 0, 0, 0) \equiv 8$  contains one 1 bit followed by two or more 0 bits (in this case, exactly by three 0 bits). That is,  $u$  has not the pattern (47). Hence, we must use Theorem 6, instead of Theorem 5.

$$C_8 = [C^{8^c}]^c = [C^7]^c = \{7, 11, 13, 14, 15\}^c = \{0, 1, 2, 4, 8\}, \quad (50)$$

since  $8^c \equiv (1, 0, 0, 0)^c = (0, 1, 1, 1) \equiv 7$ , and the set  $C^7 = \{7, 11, 13, 14, 15\}$  has been obtained in Example 14.

This example can be illustrated looking at the right-most digraph ( $n = 4$ ) in Figure 1, where we observe that 0, 1, 2, 4, 8 are exactly the vertices connected (comparable by intrinsic order) with 8 and drawn above 8.

## 5. Conclusions

A complex stochastic Boolean system (CSBS) is a system depending on an arbitrary (and, in practice, usually large) number  $n$  of random Boolean variables. The behavior of a CSBS is determined by the current values of the  $2^n$  corresponding binary  $n$ -tuple probabilities. In this context, this paper has proposed the following question: For any fixed binary  $n$ -tuple  $u$ , how can we determine the set  $C^u$  of all binary  $n$ -tuples  $v$  with occurrence probabilities less than or equal to the probability of  $u$ ? To answer this question, the evaluation of all  $2^n$  binary string probabilities is not feasible due to its exponential complexity.

To overcome this obstacle, we presented the intrinsic order criterion (IOC): a simple positional criterion that allows to compare most of pairs of binary  $n$ -tuple probabilities,  $\Pr\{u\}, \Pr\{v\}$ , without computing them, simply looking at the positions of their 0s and 1s. The intrinsic ordering, as well as our examples and new results, have been illustrated through a directed graph called the intrinsic order graph. For applying IOC, as well as the other theoretical results that we presented in this paper, it is enough to order the  $n$  basic component probabilities of the CSBS, instead of ordering the  $2^n$  binary string probabilities. Hence, the proposed method drastically reduces the complexity of the problem from the exponential to the linear.

Using IOC, we have answered the proposed question in two different cases. First, we have characterized, by a surprisingly easy condition, those binary  $n$ -tuples  $u$  for which the set  $C^u$  of binary strings with occurrence probabilities less than or equal to the one of  $u$  is simply the set of binary strings with decimal numberings greater than or equal to the one of  $u$ . The required condition for this first, “nice” case is that  $u$  can not contain any 0 bit followed by two (or more) 1 bits. Second, for the binary  $n$ -tuples  $u$  that do not satisfy this positional condition, we have developed an algorithm which quickly determines the set  $C^u$ . Analogously, we have solved the dual problem of determining, in an efficient way, the set  $C_u$  of all binary  $n$ -tuples  $v$  with occurrence probabilities greater than or equal to the probability of  $u$ . Since the only assumption on the parameters of the associated Bernoulli distribution, is non restrictive in practice, our results provide a unified approach for the analysis of CSBSs. In particular, for future research, our model can be applied to the analysis and modeling of cellular automata (CAs). As parameters or basic probabilities  $p_i$ , we consider the transition probabilities defined by the (probabilistic) CA rules. In general, the theoretical results can be applied to many scientific, technical or social areas, it means, wherever CSBSs appear.

### Acknowledgments

The author would like to thank Jiri Kroc, from the University of West Bohemia, for his many helpful suggestions and detailed comments which substantially improved this work and to Peter Sloot, from the University of Amsterdam, for strongly encouraging me to work on these topics related to the intrinsic order graph. This work was partially supported by MEC (Spain) and FEDER through grant contract CGL2004-06171-C03-02/CLI.

### References

- [1] Andrews, J. D. and Moss, B., *Reliability and Risk Assessment*, 2nd ed. (Professional Engineering Publishing, 2002).
- [2] González, L., A new method for ordering binary states probabilities in Reliability and Risk Analysis, *Lect. Notes Comput. Sci.* **2329** (1), 137–146 (2002).
- [3] González, L., *N*-tuples of 0s and 1s: Necessary and sufficient conditions for intrinsic order, *Lect. Notes Comput. Sci.* **2667** (1), 937–946 (2003).
- [4] González, L., A picture for complex stochastic Boolean systems: The intrinsic order graph, *Lect. Notes Comput. Sci.* **3993** (3), 305–312 (2006).
- [5] González, L., A mathematical model to evaluate the unavailability of a technical system, in *Fifth Int. Conference on Enginnering Computational Technology (ECT '06)*, eds. Topping, B. H. V. *et al.* (Civil-Comp Press, 2006), pp. 493–494.
- [6] González, L., García, D. and Galván, B., An intrinsic order criterion to evaluate large, complex fault trees, *IEEE Trans. Reliability.* **53** (3), 297–305 (2004).
- [7] National Aeronautics and Space Administration, *Fault Tree Analysis: A Bibliography*, Technical Report NASA/SP-2000-6111 (2000).
- [8] O'Connor, P. D. T., *Practical Reliability Engineering*, 4th ed. (John Wiley & Sons, 2002).
- [9] Singpurwalla, N. D., Foundational issues in Reliability and Risk Analysis, *SIAM Rev.* **30** (2), 264–282 (1988).
- [10] Stanley, R. P., *Enumerative Combinatorics*, vol. 1 (Cambridge University Press, 1997), pp. 96–201.
- [11] Stuart, A. and Ord, J. K., *Kendall's Advanced Theory of Statistics*, vol. 1 (Oxford University Press, 1998).
- [12] U. S. Nuclear Regulatory Commission, *Reactor Safety Study: An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants*, Technical Report NUREG-75/014: WASH-1400 (1975).