Proceedings of the World Congress on Engineering and Computer Science 2009 Vol I
WCECS 2009, October 20-22, 2009, San Francisco, USA

Complex Stochastic Boolean Systems: Generating and Counting the Binary n-Tuples Intrinsically Less or Greater than u *

Luis González †

Abstract—**A complex stochastic Boolean system (CSBS) is a system depending on an arbitrary number n of random Boolean variables. The behavior of a CSBS is determined by the ordering between the occurrence probabilities $\Pr\{u\}$ of the 2^n associated binary n-tuples $u = (u_1, \ldots, u_n) \in \{0,1\}^n$. In this context, for every fixed binary n-tuple u, this paper presents two simple algorithms –exclusively based on the vector of positions of the 1-bits (0-bits, respectively) in u– for rapidly generating (and counting) all the binary n-tuples v whose occurrence probabilities $\Pr\{v\}$ are always less than or equal to (greater than or equal to, respectively) $\Pr\{u\}$. Results are illustrated with the intrinsic order graph and they obey to a nice duality property (interchange 0s by 1s and "more probable" by "less probable").**

Keywords: complex stochastic Boolean systems, Hamming weight, intrinsic order, intrinsic order graph, (in)comparable binary n-tuples

1 Introduction

This paper deals with the analysis of complex systems depending on an arbitrary number n of random Boolean variables, the so-called complex stochastic Boolean systems (CSBSs). That is, the n basic variables of the system are assumed to be stochastic and they only take two possible values: either 0 or 1. Each one of the 2^n possible elementary states associated to a CSBS is given by a binary n-tuple $u = (u_1, \ldots, u_n) \in \{0,1\}^n$ of 0s and 1s, and it has its own occurrence probability $\Pr\{(u_1, \ldots, u_n)\}$.

Hence, a CSBS on n variables $x_1, \ldots, x_n$ can be modeled by the n-dimensional Bernoulli distribution with parameters $p_1, \ldots, p_n$ defined by

$$\Pr\{x_i = 1\} = p_i, \ \Pr\{x_i = 0\} = 1 - p_i,$$

Throughout this paper we assume that the n Bernoulli variables x_i are mutually statistically independent, so that the occurrence probability of a given binary string $u = (u_1, \ldots, u_n) \in \{0,1\}^n$ of length n can be easily computed as

$$\Pr\{\, u\} = \prod_{i=1}^{n} p_i^{u_i} \left(1 - p_i\right)^{1-u_i}, \tag{1.1}$$

that is, $\Pr\{\, u\}$ is the product of factors p_i if $u_i = 1$, $1 - p_i$ if $u_i = 0$.

Example 1.1 Let $n = 4$ and $u = (0,1,0,1)$. Then for $p_1 = 0.1,\ p_2 = 0.2,\ p_3 = 0.3,\ p_4 = 0.4$, using (1.1), we have

$$\Pr\{(0,1,0,1)\} = (1 - p_1)\, p_2\, (1 - p_3)\, p_4 = 0.0504.$$

The behavior of a CSBS is determined by the ordering between the current values of the 2^n associated binary n-tuple probabilities $\Pr\{u\}$. Computing all these 2^n probabilities –by (1.1)– and ordering them in decreasing or increasing order of their values is only possible in practice for small values of the number n of basic variables. However, for large values of n, it is necessary to use alternative procedures for comparing the binary string probabilities. For this purpose, in [1] we have established a simple positional criterion that allows one to compare two given binary n-tuple probabilities, $\Pr\{u\}, \Pr\{v\}$, without computing them, simply looking at the positions of the 0s and 1s in the n-tuples u, v. We have called it the *intrinsic order criterion*, because it is independent of the basic probabilities p_i and it *intrinsically* depends on the positions of the 0s and 1s in the binary strings.

Hence, for those pairs (u, v) of binary n-tuples comparable by intrinsic order, the ordering between their occurrence probabilities is always the same for all sets of basic probabilities $\{p_i\}_{i=1}^n$. On the contrary, for those pairs (u, v) of binary n-tuples incomparable by intrinsic order, the ordering between their occurrence probabilities depends on the current values of the probabilities $\{p_i\}_{i=1}^n$.

In this context, the main goal of this paper is to present two new algorithms for rapidly generating and counting all the binary n-tuples v intrinsically less (greater, respectively) than or equal to an arbitrary, fixed binary

*Manuscript received July 26, 2009. This work was supported in part by the Ministerio de Ciencia e Innovación (Spain) and FEDER through Grant contract: CGL2008-06003-C03-01/CLI.

†University of Las Palmas de Gran Canaria, Research Institute IUSIANI & Department of Mathematics, 35017 Las Palmas de Gran Canaria, Spain. Email: luisglez@dma.ulpgc.es

n-tuple u. In other words, our algorithms –exclusively based on the positions of the 1-bits (0-bits, respectively) of u– determine the sets of binary n-tuples v whose occurrence probabilities $\Pr\{v\}$ are always less than or equal to (greater than or equal to, respectively) $\Pr\{u\}$.

For this purpose, this paper has been organized as follows. In Section 2, we present some previous results on the intrinsic order relation, the intrinsic order graph and other related aspects, enabling non-specialists to follow the paper without difficulty and making the presentation self-contained. Section 3 is devoted to present our new, unpublished results: Two new algorithms and formulas for generating and counting all the binary n-tuples intrinsically less or greater than u. Finally, in Section 4, we present our conclusions.

2 Background on Intrinsic Order

2.1 Intrinsic Order Relation on $\{0,1\}^n$

Fist, we must set the following notations. 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. In the following, we use indistinctly the binary representation or the decimal representation to denote the elements of $\{0,1\}^n$, and we represent the conversion between these two numbers systems by the symbol "$\equiv$". Also, the Hamming weight of a binary n-tuple u (i.e., the number of 1-bits in u) will be denoted, as usual by $w_H(u)$, i.e.,

$$(u_1,\ldots,u_n) \equiv u_{(10} = \sum_{i=1}^{n} 2^{n-i} u_i, \quad w_H(u) = \sum_{i=1}^{n} u_i,$$

e.g., for $n=5$ we have

$$(1,0,1,1,1) \equiv 2^0+2^1+2^2+2^4 = 23, \; w_H(1,0,1,1,1) = 4.$$

According to (1.1), the ordering between two given binary string probabilities $\Pr(u)$ and $\Pr(v)$ depends, in general, on the parameters p_i of the Bernoulli distribution, as the following simple example shows.

Example 2.1 Let $n=3,\; u=(0,1,1)$ and $v=(1,0,0)$. Using (1.1) we have

For $p_1 = 0.1,\; p_2 = 0.2,\; p_3 = 0.3$:

$$\Pr\{(0,1,1)\} = 0.054 < \Pr\{(1,0,0)\} = 0.056,$$

for $p_1 = 0.2,\; p_2 = 0.3,\; p_3 = 0.4$:

$$\Pr\{(0,1,1)\} = 0.096 > \Pr\{(1,0,0)\} = 0.084.$$

As mentioned in Section 1, to overcome the exponential complexity inherent to the task of computing and sorting the 2^n binary string probabilities (associated to a CSBS with n Boolean variables), we have introduced the following intrinsic order criterion [1], denoted from now on by the acronym IOC (see [5] for the proof).

Theorem 2.1 (The intrinsic order theorem) *Let $n \geq 1$. Let $x_1,\ldots,x_n$ be n mutually independent Bernoulli variables whose parameters $p_i = \Pr\{x_i = 1\}$ satisfy*

$$0 < p_1 \leq p_2 \leq \cdots \leq p_n \leq 0.5. \tag{2.1}$$

Then the probability of the n-tuple $v = (v_1,\ldots,v_n) \in \{0,1\}^n$ is intrinsically less than or equal to the probability of the n-tuple $u = (u_1,\ldots,u_n) \in \{0,1\}^n$ (that is, for all set $\{p_i\}_{i=1}^n$ satisfying (2.1)) if and only if the matrix

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

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 2.1 In the following, we assume that the parameters p_i always satisfy condition (2.1). Note that this hypothesis is not restrictive for practical applications because, if for some $i : p_i > \frac{1}{2}$, then we only need to consider the variable $\overline{x_i} = 1 - x_i$, instead of x_i. Next, we order the n Bernoulli variables by increasing order of their probabilities.

Remark 2.2 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 2.3 The term *corresponding,* used in Theorem 2.1, 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: For each $\binom{1}{0}$ column in matrix M_v^u, the number of preceding $\binom{0}{1}$ columns must be strictly greater than the number of preceding $\binom{1}{0}$ columns.

The matrix condition IOC, stated by Theorem 2.1, is called the *intrinsic order criterion*, because it is independent of the basic probabilities p_i and it *intrinsically* depends on the relative positions of the 0s and 1s in the binary n-tuples u, v. Theorem 2.1 naturally leads to the following partial order relation on the set $\{0,1\}^n$ [1]. The so-called intrinsic order will be denoted by "$\preceq$", and we shall write $v \preceq u$ ($u \succeq v$) to indicate that v (u) is intrinsically less (greater) than or equal to u (v).

Definition 2.1 *For all $u, v \in \{0,1\}^n$*

$$v \preceq u \text{ iff } \Pr\{v\} \leq \Pr\{u\} \text{ for all set } \{p_i\}_{i=1}^n \text{ s.t. (2.1)}$$
$$\text{iff } M_v^u \text{ satisfies IOC.}$$

From now on, the partially ordered set (poset, for short) $(\{0,1\}^n, \preceq)$ will be denoted by I_n.

Example 2.2 For $n = 3$: $3 \equiv (0,1,1) \npreceq 4 \equiv (1,0,0)$, $4 \equiv (1,0,0) \npreceq 3 \equiv (0,1,1)$ because the matrices

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

do not satisfy IOC (Remark 2.3). Thus, $(0,1,1)$ and $(1,0,0)$ are incomparable by intrinsic order, i.e., the ordering between $\Pr\{(0,1,1)\}$ and $\Pr\{(1,0,0)\}$ depends on the parameters $\{p_i\}_{i=1}^3$, as Example 2.1 has shown.

Example 2.3 For $n = 5$: $24 \equiv (1,1,0,0,0) \preceq 5 \equiv (0,0,1,0,1)$ because matrix

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

satisfies IOC (Remark 2.2).
Thus, for all $\{p_i\}_{i=1}^5$ s.t. (2.1)

$$\Pr\{(1,1,0,0,0)\} \leq \Pr\{(0,0,1,0,1)\}.$$

Example 2.4 For all $n \geq 1$, the binary n-tuples

$$\left(0, \overset{n}{\ldots}, 0\right) \equiv 0 \quad \text{and} \quad \left(1, \overset{n}{\ldots}, 1\right) \equiv 2^n - 1$$

are the maximum and minimum elements, respectively, in the poset I_n. Indeed, both matrices

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

satisfy the intrinsic order criterion, since they have no $\binom{1}{0}$ columns!.
Thus, for all $u \in \{0,1\}^n$ and for all $\{p_i\}_{i=1}^n$ s.t. (2.1)

$$\Pr\left\{\left(1, \overset{n}{\ldots}, 1\right)\right\} \leq \Pr\{(u_1, \ldots, u_n)\} \leq \Pr\left\{\left(0, \overset{n}{\ldots}, 0\right)\right\}.$$

2.2 The Intrinsic Order Graph

To finish this Section, we present the graphical representation of the poset I_n. The usual representation of a poset is its Hasse diagram (see, e.g., [6] for more details about posets and Hasse diagrams). This is a directed graph (digraph, for short) whose vertices are the binary n-tuples of 0s and 1s, and whose edges go downward from u to v whenever u covers v (denoted by $u \rhd v$), that is, whenever u is intrinsically greater than v with no other elements between them, i.e.

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

The Hasse diagram of the poset I_n will be also called the intrinsic order graph for n variables. For all $n \geq 2$, in [3] we have developed an algorithm for iteratively building up the digraph of I_n from the digraph of I_1. Basically, I_n is obtained by adding to I_{n-1} its isomorphic copy $2^{n-1} + I_{n-1}$: A nice fractal property of I_n!

In Fig. 1, the intrinsic order graph for $n = 1,2,3,4$ is shown, using the decimal numbering instead of the binary representation of their 2^n nodes, for a more comfortable and simpler notation. Each pair (u,v) of vertices connected in the digraph of I_n either by one edge or by a longer descending path (consisting of more than one edge) from u to v, means that u is intrinsically greater than v, i.e., $u \succ v$. On the contrary, each pair (u,v) of non-connected vertices in the digraph of I_n either by one edge or by a longer descending path, means that u and v are incomparable by intrinsic order, i.e., $u \nsucc v$ and $v \nsucc u$.

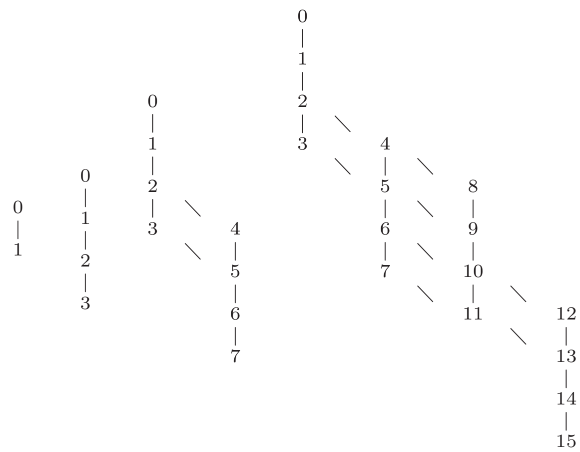

Fig. 1. The intrinsic order graph for $n = 1,2,3,4$.

2.3 The Sets C^u and C_u

Many different properties of the intrinsic order relation can be derived from its simple matrix description IOC (see, e.g., [1, 2, 3]). For the purpose of this paper, we must recall here the following necessary (but not sufficient) condition for intrinsic order; see [2] for the proof.

Corollary 2.1 *For all* $u, v \in \{0,1\}^n$

$$u \succeq v \quad \Rightarrow \quad w_H(u) \leq w_H(v).$$

Definition 2.2 *For every binary n-tuple $u \in \{0,1\}^n$, C^u (C_u, respectively) is the set of all binary n-tuples v whose occurrence probabilities* $\Pr\{v\}$ *are always less (greater, respectively) than or equal to* $\Pr\{u\}$*, i.e., those n-tuples v intrinsically less (greater, respectively) than or equal to u, i.e.,*

$$C^u = \{v \in \{0,1\}^n \mid \Pr\{u\} \geq \Pr\{v\}, \forall \{p_i\}_{i=1}^n \text{ s.t. } (2.1)\},$$

$$C_u = \{v \in \{0,1\}^n \mid \Pr\{u\} \leq \Pr\{v\}, \forall \{p_i\}_{i=1}^n \text{ s.t. } (2.1)\}.$$

ISBN:978-988-17012-6-8

In other words, according to Definition 2.1,

$$C^u = \{v \in \{0,1\}^n \mid u \succeq v\}, \tag{2.2}$$

$$C_u = \{v \in \{0,1\}^n \mid u \preceq v\}. \tag{2.3}$$

For instance, due to Example 2.4, we have, for all $n \geq 1$,

$$C^{\left(0,\overset{n}{\ldots},0\right)} = \{0,1\}^n = C_{\left(1,\overset{n}{\ldots},1\right)},$$

as Fig. 1 shows, for $n = 1, 2, 3, 4$.

The sets C^u and C_u are closely related via complementary n-tuples, as described with precision by the following definition and theorem [4].

Definition 2.3 *The complementary n-tuple of a binary n-tuple $u \in \{0,1\}^n$ is obtained by changing its 0s by 1s and its 1s by 0s*

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

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

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

Theorem 2.2 *For all $n \geq 1$ and for all $u \in \{0,1\}^n$*

$$C_u = \left[C^{u^c}\right]^c, \qquad C^u = \left[C_{u^c}\right]^c. \tag{2.4}$$

3 The Algorithms

This Section is devoted to present our new results. First, we need to set the following nomenclature and notation.

Definition 3.1 *Let $n \geq 1$ and let $u \in \{0,1\}^n$ with Hamming weight $w_H(u) = m$. Then*

(i) The vector of positions of 1s of u is the vector of positions of its m 1-bits, displayed in increasing order from the left-most position to the right-most position, and it will be denoted by

$$u = \left[i_1^1, \ldots, i_m^1\right]_n^1, \; 1 \leq i_1^1 < \cdots < i_m^1 \leq n, \; 0 < m \leq n.$$

(ii) The vector of positions of 0s of u is the vector of positions of its $(n-m)$ 0-bits, displayed in increasing order from the left-most position to the right-most position, and it will be denoted by

$$u = \left[i_1^0, \ldots, i_{n-m}^0\right]_n^0, \; 1 \leq i_1^0 < \cdots < i_{n-m}^0 \leq n, \; 0 \leq m < n.$$

Example 3.1 For $n = 6$ and $u = 27 \equiv (0,1,1,0,1,1)$, we have $m = w_H(u) = 4$, $n - m = 2$, and

$$u = \left[i_1^1, i_2^1, i_3^1, i_4^1\right]_6^1 = [2,3,5,6]_6^1, \quad u = \left[i_1^0, i_2^0\right]_6^0 = [1,4]_6^0.$$

In [4], the authors present an algorithm for obtaining the set C^u, for each given binary n-tuple u. Basically, this algorithm determines C^u by expressing the set difference $\{0,1\}^n - C^u$ as a set union of certain half-closed intervals of natural numbers (decimal representation of binary strings). The next two theorems present new, more efficient algorithms for rapidly determining C^u and C_u, using the binary representation of their elements. Moreover, our new algorithms allow us not only to generate, but also to count the number of elements of C^u and C_u.

Theorem 3.1 **(The algorithm for C^u)** *Let $u \in \{0,1\}^n$, $u \neq 0$, with weight $w_H(u) = m$ $(0 < m \leq n)$. Let $u = \left[i_1^1, \ldots, i_m^1\right]_n^1$ be the vector of positions of 1s of u. Then C^u is the set of binary n-tuples v generated by the following algorithm:*

(i) Step 1 (Generation of the sequences $\{j_1^1, \ldots, j_m^1\}$) :

For $j_1^1 = 1$ to i_1^1 Do:

For $j_2^1 = j_1^1 + 1$ to i_2^1 Do:

. . . .

For $j_m^1 = j_{m-1}^1 + 1$ to i_m^1 Do:

Write $\{j_1^1, j_2^1, \ldots, j_m^1\}$

EndDo

. . . .

EndDo

EndDo

(ii) Step 2 (Generation of the n-tuples $v \in C^u$) :

For every sequence $\{j_1^1, \ldots, j_m^1\}$ generated by Step 1, define $v = (v_1, \ldots, v_n)$ as follows

$$v_i = \begin{cases} 1 & \text{if } \; i \in \{j_1^1, \ldots, j_m^1\}, \\ 0 & \text{if } \; i \notin \{j_1^1, \ldots, j_m^1\}, \; i < j_m^1, \\ 0,1 & \text{if } \; i \notin \{j_1^1, \ldots, j_m^1\}, \; i > j_m^1. \end{cases} \tag{3.1}$$

Moreover, the cardinality of the set C^u is given by the multiple sum

$$|C^u| = \sum_{j_1^1=1}^{i_1^1} \sum_{j_2^1=j_1^1+1}^{i_2^1} \cdots \sum_{j_m^1=j_{m-1}^1+1}^{i_m^1} 2^{n-j_m^1}. \tag{3.2}$$

Proof. Note that, according to (2.2) and Corollary 2.1, we have

$$v \in C^u \Leftrightarrow u \succeq v \Rightarrow m = w_H(u) \leq w_H(v),$$

that is, the weight of every n-tuple $v \in C^u$ is necessarily greater than or equal to the weight m of u.

First, we generate all binary n-tuples $v \in C^u$ with weight m. Let $v \in \{0,1\}^n$ with $w_H(v) = m$ and let $v = \left[j_1^1, \ldots, j_m^1\right]_n^1$ be the vector of positions of 1-bits of v. Using (2.2) and IOC (Theorem 2.1), we have that $v \in C^u$ if and only if

$$u = \left[i_1^1, \ldots, i_m^1\right]_n^1 \succeq \left[j_1^1, \ldots, j_m^1\right]_n^1 = v,$$

if and only if v contains at least one 1-bit among the positions 1 and i_1^1, at least two 1-bits among the positions 1 and i_2^1 , ..., at least $(m-1)$ 1-bits among the positions 1 and i_{m-1}^1, exactly m 1-bits among the positions 1 and i_m^1, and (if $i_m^1 < n$) v has no 1-bits among the positions $i_m^1 + 1$ and n. That is, $v \in C^u$, $w_H(v) = m$ if and only if

$$1 \leq j_1^1 \leq i_1^1,\ j_1^1 + 1 \leq j_2^1 \leq i_2^1, \ldots, j_{m-1}^1 + 1 \leq j_m^1 \leq i_m^1, \tag{3.3}$$

and these are exactly the sequences $\left\{j_1^1, \ldots, j_m^1\right\}$ generated by Step 1. Hence, the binary n-tuples $v \in C^u$ with the same weight m as u are exactly the n-tuples $v = \left[j_1^1, \ldots, j_m^1\right]_n^1$, that is, those binary n-tuples v defined by Step 2, when we choose $v_i = 0$ for all $i > j_m^1$. In addition, by the way we derive that the number of those n-tuples is exactly

$$\left|\{v \in C^u \mid w_H(v) = m\}\right| = \sum_{j_1^1=1}^{i_1^1} \sum_{j_2^1=j_1^1+1}^{i_2^1} \cdots \sum_{j_m^1=j_{m-1}^1+1}^{i_m^1} 1.$$

Second, we generate all binary n-tuples $v \in C^u$ (with all possible weights $w_H(v) \geq m$). Once we have characterized all n-tuples $v \in C^u$ with $w_H(v) = m = w_H(u)$, the question is quite simple. Indeed, let $u \succeq v$ with $w_H(v) = t \geq m$. Let $\left\{j_1^1, \ldots, j_m^1\right\}$ be the sequence of positions of the m left-most 1-bits of v. Then, on one hand, according to IOC, this sequence must necessarily satisfy (3.3) (i.e., it must be one of the sequences generated by Step 1). On the other hand, note that the substitution of 0s by 1s in any n-tuple v such that $u \succeq v$ does not avoid the IOC condition, because this substitution changes the $\binom{0}{0}$ and $\binom{1}{0}$ columns of matrix M_v^u into $\binom{0}{1}$ and $\binom{1}{1}$ columns, respectively. Hence, to obtain all the binary strings of the set C^u, it is enough to assign, in all possible ways, both values, either 0 or 1, to any of the $n - j_m^1$ (if $j_m^1 < n$) null right-most components $v_{j_m^1+1}, \cdots, v_n$ of all the binary strings $v = \left[j_1^1, \ldots, j_m^1\right]_n^1$ with eight m, generated by (3.3). In other words, we define v as described by (3.1).

Finally, since there are exactly $2^{n-j_m^1}$ different ways of assigning the values $v_i = 0, 1$ for all $j_m^1 < i \leq n$, and since by this procedure we generate all elements of C^u without repetitions, then the cardinality of C^u is given by (3.2). $\square$

Theorem 3.2 (The algorithm for C_u) *Let $u \in \{0,1\}^n$, $u \neq 2^n - 1$, with weight $w_H(u) = m$ $(0 \leq m < n)$. Let $u = \left[i_1^0, \ldots, i_{n-m}^0\right]_n^0$ be the vector of positions of 0s of u. Then C_u is the set of binary n-tuples v generated by the following algorithm:*

(i) Step 1 (Generation of the sequences $\left\{j_1^0, \ldots, j_{n-m}^0\right\}$) :

For $j_1^0 = 1$ to i_1^0 Do:

For $j_2^0 = j_1^0 + 1$ to i_2^0 Do:

....

For $j_{n-m}^0 = j_{n-m-1}^0 + 1$ to i_{n-m}^0 Do:

Write $\left\{j_1^0, j_2^0, \ldots, j_{n-m}^0\right\}$

EndDo

....

EndDo

EndDo

(ii) Step 2 (Generation of the n-tuples $v \in C_u$) :

For every sequence $\left\{j_1^0, \ldots, j_{n-m}^0\right\}$ generated by Step 1, define $v = (v_1, \ldots, v_n)$ as follows

$$v_i = \begin{cases} 0 & \text{if } \ i \in \left\{j_1^0, \ldots, j_{n-m}^0\right\}, \\ 1 & \text{if } \ i \notin \left\{j_1^0, \ldots, j_{n-m}^0\right\},\ i < j_{n-m}^0, \\ 0,1 & \text{if } \ i \notin \left\{j_1^0, \ldots, j_{n-m}^0\right\},\ i > j_{n-m}^0. \end{cases} \tag{3.4}$$

Moreover, the cardinality of the set C_u is given by the multiple sum

$$|C_u| = \sum_{j_1^0=1}^{i_1^0} \sum_{j_2^0=j_1^0+1}^{i_2^0} \cdots \sum_{j_{n-m}^0=j_{n-m-1}^0+1}^{i_m^0} 2^{n-j_{n-m}^0}. \tag{3.5}$$

Proof. For proving this theorem, it is enough to use Theorems 2.2 and 3.1. Indeed, due to the left-hand set equality of Theorem 2.2, obtaining the set C_u is equivalent to obtaining the set $\left[C^{u^c}\right]^c$.

First, we obtain the set C^{u^c} using Theorem 3.1. Since the $(n-m)$ 1-bits of u^c are placed at the same positions as the $(n-m)$ 0-bits of u (i.e., $i_1^0, \ldots, i_{n-m}^0$), then the sequences for u^c, generated by Step 1 in Theorem 3.1, are exactly the sequences $\left\{j_1^0, \ldots, j_{n-m}^0\right\}$ for u, generated by Step 1 in Theorem 3.2. Thus, all binary n-tuples $v \in C^{u^c}$ can be obtained by applying Step 2 in Theorem 3.1 to each one of the sequences $\left\{j_1^0, \ldots, j_{n-m}^0\right\}$, generated by Step 1 in Theorem 3.2.

Second, once we have obtained the set C^{u^c}, for obtaining the set $\left[C^{u^c}\right]^c$, we only need to take the complementary n-tuples of all n-tuples $v \in C^{u^c}$. That is, we only need to change the 0s by 1s and the 1s by 0s in all n-tuples $v \in C^{u^c}$, obtained from (3.1) for the sequences $\left\{j_1^0, \ldots, j_{n-m}^0\right\}$. But this is precisely to use (3.4).

Finally, (3.5) follows immediately using the fact that $|C_u| = \left|\left[C^{u^c}\right]^c\right| = \left|C^{u^c}\right|$ and (3.2). $\square$

Remark 3.1 Note the strong duality relation between Theorems 3.1 and 3.2. The statement of each theorem is exactly the statement of the other one after interchanging 1s and 0s, C^u and C_u. Indeed, due to Theorem 2.2, one can determine the set C^u (C_u, respectively) by determining the set $[C_{u^c}]^c$ ($\left[C^{u^c}\right]^c$, respectively) using Theorem 3.2 (Theorem 3.1, respectively).

Remark 3.2 Let $u \in \{0,1\}^n$ be a fixed binary n-tuple. On one hand, $v \in \{0,1\}^n$ is comparable by intrinsic order with u if either $u \succeq v$ or $u \preceq v$, i.e., if either $v \in C^u$ or $v \in C_u$, respectively (see Definition 2.2). On the other hand, due to the reflexivity and antisymmetry properties of the intrinsic order relation, we have $C^u \cap C_u = \{u\}$. Hence, the number of binary n-tuples v incomparable by intrinsic order with u is given by

$$|\{v \in \{0,1\}^n \mid u \nsucceq v,\ u \npreceq v\}| = 2^n + 1 - |C^u| - |C_u| \,. \tag{3.6}$$

The following example illustrates Theorems 3.1 and 3.2.

Example 3.2 Let $n = 4$ and $u = 6 \equiv (0,1,1,0)$. Then, we have $m = w_H(u) = 2$, $n - m = 2$, and

$$u = \left[i_1^1, i_2^1\right]_4^1 = [2,3]_4^1 \,, \quad u = \left[i_1^0, i_2^0\right]_4^0 = [1,4]_4^0 \,.$$

Using Theorems 3.1 and 3.2 we can generate all the 4-tuples $v \in C^u$ and all the 4-tuples $v \in C_u$, respectively. Results are depicted in Tables I and II, respectively. The first column of Tables I and II, show the auxiliary sequences generated by Step 1 in Theorems 3.1 and 3.2, respectively. The second column of Tables I and II, show all the binary 4-tuples $v \in C^u$ and $v \in C_u$ generated by Step 2 in Theorems 3.1 and 3.2, respectively. The symbol "$*$" means that we must choose both binary digits: 0 and 1, as described by the last line of (3.1) and (3.4). According to (3.2) and (3.5), the sums of all quantities in the third column of Tables I and II, give the number of elements of C^u and C_u, respectively, using just the first columns and then with no need to obtain these elements by the second columns.

Table I. The set $C^{(0,1,1,0)}$ and its cardinality.

$\{j_1^1, j_2^1\}$	$v \in C^u$	$2^{n-j_m^1} = 2^{4-j_2^1}$
$\{1,2\}$	$(1,1,*,*) \equiv 12,13,14,15$	$2^{4-2} = 4$
$\{1,3\}$	$(1,0,1,*) \equiv 10,11$	$2^{4-3} = 2$
$\{2,3\}$	$(0,1,1,*) \equiv 6,7$	$2^{4-3} = 2$

So, $C^u = \{6,7,10,11,12,13,14,15\}$ and $|C^u| = 8$ (see the digraph of I_4, the right-most one in Fig. 1).

Table II. The set $C_{(0,1,1,0)}$ and its cardinality.

$\{j_1^0, j_2^0\}$	$v \in C_u$	$2^{n-j_{n-m}^0} = 2^{4-j_2^0}$
$\{1,2\}$	$(0,0,*,*) \equiv 0,1,2,3$	$2^{4-2} = 4$
$\{1,3\}$	$(0,1,0,*) \equiv 4,5$	$2^{4-3} = 2$
$\{1,4\}$	$(0,1,1,0) \equiv 6$	$2^{4-4} = 1$

So, $C_u = \{0,1,2,3,4,5,6\}$ and $|C_u| = 7$ (see the digraph of I_4, the right-most one in Fig. 1).

Finally, the binary n-tuples v incomparable by intrinsic order with $u = (0,1,1,0)$ are $\{0,1\}^n - (C^u \cup C_u) = \{8,9\}$ (see the digraph of I_4, the right-most one in Fig. 1). So, they are exactly $2^n + 1 - |C^u| - |C_u| = 16 + 1 - 8 - 7 = 2$ binary strings, in accordance with (3.6).

4 Conclusions

The two dual algorithms presented in this paper for rapidly generating and counting the sets C^u and C_u are exclusively based on the positions of the 1-bits or 0-bits, respectively, of u, and they are easily implementable. Hence, results can be applied to any CSBS with an arbitrarily large number n of independent Boolean variables.

References

[1] González, L., "A New Method for Ordering Binary States Probabilities in Reliability and Risk Analysis," *Lect Notes Comp Sc*, V2329, N1, pp. 137-146, 2002.

[2] González, L., "N-tuples of 0s and 1s: Necessary and Sufficient Conditions for Intrinsic Order," *Lect Notes Comp Sc*, V2667, N1, pp. 937-946, 2003.

[3] González, L., "A Picture for Complex Stochastic Boolean Systems: The Intrinsic Order Graph," *Lect Notes Comput Sc*, V3993, N3, pp. 305-312, 2006.

[4] González, L., "Algorithm comparing binary string probabilities in complex stochastic Boolean systems using intrinsic order graph," *Adv Complex Syst*, V10, N1, pp. 111-143, 2007.

[5] González, L., García, D., Galván, B., "An Intrinsic Order Criterion to Evaluate Large, Complex Fault Trees," *IEEE Trans on Reliability*, V53, N3, pp. 297-305, 2004.

[6] Stanley, R.P., *Enumerative Combinatorics,* Vol 1, Cambridge University Press, 1997.