A New Method for Ordering Binary States Probabilities in Reliability and Risk Analysis*

Luis González

Department of Mathematics,
University of Las Palmas de Gran Canaria,
Edificio de Informática y Matemáticas,
35017 Las Palmas de Gran Canaria, Spain
luisglez@dma.ulpgc.es

Abstract. A new method is proposed to select the binary n-tuples of 0s and 1s by decreasing order of there occurrence probabilities in stochastic Boolean models. This method can be applied to evaluate fault trees in Reliability Engineering and Risk Analysis, as well as to many other problems described by a stochastic Boolean structure. The selecting criterion is exclusively based on the positions of 0s and 1s in the binary n-tuples. In this way, the computational cost in sorting algorithms is drastically reduced, because the proposed criterion is independent of the probabilities of the Boolean variables. Every step, the algorithm extends the set of selected binary states, obtaining the binary n-tuples just by adding 0 (1) at the end of all the (some) previously selected binary $n-1$-tuples.

1 Introduction

In many applications of Probability Calculus and Operations Research, the basic variables are Boolean variables with a prefixed occurrence probability. This happens, for example, in Reliability Theory and Risk Analysis, where the technical system is described by a Boolean function Φ depending on the basic components of the system. This Boolean function is often represented by a fault tree, a picture describing the superposition of component faults to create system faults [2], [9]. Then, assuming that: $\Phi = 1$ if the system fails, the system unavailability, usually called the top event probability, is evaluated by computing the probability $\Pr\{\Phi = 1\}$.

Many methods have been developed to evaluate the top event probability in Fault Tree Analysis. These methods are usually classified into two great groups: deterministic methods and probabilistic methods [3]. The first ones provide exact lower and upper bounds on top event probability [1], [7], [8]. The second ones provide an estimation of the system unavailability by using sampling and variance reduction techniques [6]. Both, deterministic and probabilistic methods, can also be subdivided into two classes: direct methods and indirect methods. The direct methods do not require any qualitative analysis of the Boolean function Φ [6], [7], [8]. The indirect methods are based on a previous analysis of the

* Partially supported by MCYT, Spain. Grant contract: REN2001-0925-C03-02/CLI

P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS 2329, pp. 137–146, 2002.
© Springer-Verlag Berlin Heidelberg 2002

fault tree logic: minimal cut sets, minimal paths, common cause failure analysis, etc. [1], [5].

It is very simple to describe an elemental direct method to evaluate $\Pr\{\Phi = 1\}$ prefixing the maximum admissible error (difference between upper and lower bounds). This method is based on the canonical normal forms of the Boolean function Φ and it can be summarised as follows [4], [9].

The probability of a Boolean function Φ taking value 1 can be estimated using the canonical normal forms of Φ. More precisely, we can obtain lower and upper bounds L, U on system unavailability by taking any subsets S_1, S_0 of the sets of binary n-tuples for which $\Phi = 1$, $\Phi = 0$, respectively:

$$S_1 \subseteq \{u \in \{0,1\}^n \ / \ \Phi(u) = 1\} \quad ; \quad S_0 \subseteq \{u \in \{0,1\}^n \ / \ \Phi(u) = 0\}.$$

$$L = \sum_{u \in S_1} \Pr\{u\} \leq \Pr\{\Phi = 1\} \leq 1 - \sum_{u \in S_0} \Pr\{u\} = U. \tag{1}$$

where, assuming that the Boolean variables are mutually independent, we have:

$$\forall\ u = (u_1, u_2, \ldots, u_n) \in \{0,1\}^n : \quad \Pr\{u\} = \prod_{i=1}^{n} p_i^{u_i} (1 - p_i)^{1-u_i}. \tag{2}$$

that is, $\Pr\{u\}$ is the product of factors $\begin{cases} p_i, & \text{if} \quad u_i = 1 \\ 1 - p_i, & \text{if} \quad u_i = 0 \end{cases}$.

Then, for any set of selected binary n-tuples, the maximum error in the estimation will be:

$$U - L = 1 - \sum_{u \in S_0 \cup S_1} \Pr\{u\}. \tag{3}$$

Therefore, the accuracy in the estimation of the top event probability improves at the same time as the total sum $\sum_{u \in S_0 \cup S_1} \Pr\{u\}$ of the probabilities of all the selected elementary states increases, and this sum is completely independent of the fault tree logic.

Of course, the main question is how to select the minor number of elementary states, with probabilities as large as possible, in order to minimise the computational cost. The first answer to this question: ordering n-tuples attending to their probabilities is not valid, because of the exponential nature of the problem and the high computational costs in sorting algorithms. Then, it is essential to propose an efficient criterion that allows the identification of elementary states with probabilities as large as possible, in order to obtain a good approximation of the top event probability using as few elementary states as possible. For this purpose, we have established a characterisation theorem [4], which provides us the above mentioned criterion.

In Sect. 2, we state this theorem and a few related explanations for a better understanding of the new results we propose in this paper. Beginning with our

new approaches, in Sect. 3 we establish a classification of the n-tuples of 0s and 1s into three groups: Top, Bottom and Jumping n-tuples. The Top n-tuples correspond to the binary states with largest occurrence probabilities. In Sect. 4, we present a simple and elegant algorithm to generate all the Top n-tuples. Next, we prove that this algorithm selects the binary states by decreasing order of their occurrence probabilities. Finally, in Sect. 5 we present our conclusions.

2 Previous Results

In [4] we have established the following theorem that *a priori* assures us that the occurrence probability of a given elementary state is always greater (lower) than another one, for any values of basic probabilities:

Theorem 1. *An intrinsic order criterion for binary n-tuples probabilities. Let $x_1, \ldots, x_n$ be n Boolean variables (corresponding to n basic components of a system), with basic probabilities $p_i = \Pr\{x_i = 1\} \quad (1 \leq i \leq n)$ verifying:*

$$0 < p_1 \leq \ldots \leq p_n \leq \frac{1}{2}. \tag{4}$$

Then, the probability of the n-tuple $(u_1, \ldots, u_n) \in \{0,1\}^n$ is intrinsically greater than or equal to the probability of the n-tuple $(v_1, \ldots, v_n) \in \{0,1\}^n$ (that is, with independence on probabilities p_i) if and only if the matrix

$$\begin{pmatrix} u_1 & \ldots & u_n \\ v_1 & \ldots & v_n \end{pmatrix}. \tag{5}$$

has not any $\binom{1}{0}$ column, or for each $\binom{1}{0}$ column there exists (at least) one corresponding preceding $\binom{0}{1}$ column (Condition IOC).

Remark 1. In the following we assume that basic probabilities p_i always satisfy condition (4). Note that this hypothesis is not restrictive because, if for some i: $p_i > \frac{1}{2}$, we only need to consider the Boolean variable $\overline{x_i}$, instead of x_i.

Remark 2. The $\binom{0}{1}$ column preceding to each $\binom{1}{0}$ column is not necessarily allocated at the immediately previous position.

Remark 3. The term *corresponding,* used in Theorem 1, has the following meaning: for each two $\binom{1}{0}$ columns in matrix (5), there must exist (at least) two different $\binom{0}{1}$ columns preceding to each other.

The partial order relation established by Theorem 1 between the n-tuples of $\{0,1\}^n$ is called intrinsic order, because it only depends on the positions of 0s and 1s in the n-tuples, but not on the basic probabilities p_i. In the following we shall note with the symbol $\preceq$ this order relation:

Definition 1. *For all* $u = (u_1, \dots, u_n), v = (v_1, \dots, v_n) \in \{0,1\}^n$:

$$(u_1, \dots, u_n) \succeq (v_1, \dots, v_n) \Leftrightarrow \Pr\{u\} \geq \Pr\{v\}\text{; } \textit{for all } p_i \textit{ s.t.}: \ (4) \Leftrightarrow \begin{pmatrix} u_1 & \dots & u_n \\ v_1 & \dots & v_n \end{pmatrix} \textit{ satisfies IOC.}$$

Example 1. $(0,0,1,1) \succeq (1,1,0,0)$ because $\begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$ satisfies *IOC* (Remark 2):

$$\Pr\{(0,0,1,1)\} \geq \Pr\{(1,1,0,0)\}, \text{ for all } p_1 \leq p_2 \leq p_3 \leq p_4 \leq \frac{1}{2}. \quad (6)$$

Example 2. $(0,1,1,1) \not\succeq, \not\preceq (1,1,0,0)$ because neither $\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$ nor $\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}$ satisfy *IOC* (Remark 3). Therefore, depending on p_i:

$$\Pr\{(0,1,1,1)\} \leq, \geq \Pr\{(1,1,0,0)\}. \quad (7)$$

For instance, if for all $i = 1,2,3,4$:

$$p_i = 10^{-1} i \Rightarrow \Pr\{(0,1,1,1)\} = 2.1 \cdot 10^{-2} > \Pr\{(1,1,0,0)\} = 8.4 \cdot 10^{-3}. \quad (8)$$

$$p_i = 10^{-2} i \Rightarrow \Pr\{(0,1,1,1)\} = 2 \cdot 10^{-5} < \Pr\{(1,1,0,0)\} = 1.8 \cdot 10^{-4}. \quad (9)$$

Now, we present the graph of the intrinsic order for $n = 4$.

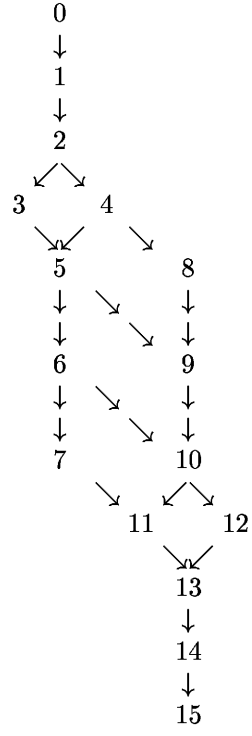

Fig. 1. Graph of the partially ordered set $\left(\{0,1\}^4, \preceq\right)$

The 4-tuples are given using decimal numbering and the relation $a \succeq b$ is noted by $a \longrightarrow b$. To finish Sect. 2, let us mention that intrinsic order implies lexicographic (truth table) order, i.e.: If $(u_1, \ldots, u_n) \succeq (v_1, \ldots, v_n)$ then, $(u_1, \ldots, u_n)$ is placed at previous position to $(v_1, \ldots, v_n)$ in the truth table [4]. The converse is false, and this is the most simple counter-example ($n = 3$): $(0,1,1) \not\succeq (1,0,0)$ (Remark 3).

3 Top, Bottom and Jumping n-Tuples

The subgraph of $\left(\{0,1\}^4, \preceq\right)$ corresponding to $n = 3$ (Fig. 1), shows that there exist only two binary 3-tuples non-comparable by intrinsic order: $3 \not\succeq 4$ and $3 \not\preceq 4$. Therefore, when we write the eight binary 3-tuples by decreasing order of their occurrence probabilities, only two cases are possible:

0	0
1	1
2	2
3	4
...	...
4	3
5	5
6	6
7	7

Fig. 2. Binary 3-tuples by decreasing order of their occurrence probabilities

The left column corresponds to the case $\Pr\{(0,1,1)\} \geq \Pr\{(1,0,0)\}$, and the right column corresponds to the case $\Pr\{(0,1,1)\} \leq \Pr\{(1,0,0)\}$. Anyway, Fig. 2 shows that for the two possible cases: the 3-tuples 0, 1 and 2 are among the four first ones; the 3-tuples 5, 6 and 7 are among the four last ones and the 3-tuples 3 and 4 can be allocated at both positions (depending on basic probabilities p_i). This fact suggests us the following definition.

Definition 2. *Let the 2^n binary n-tuples be ordered by decreasing order of their occurrence probabilities. Then:*

(i) The binary n-tuple $(u_1, \ldots, u_n)$ is called Top if it is always among the 2^{n-1} first n-tuples (for any basic probabilities p_i).

(ii) The binary n-tuple $(u_1, \ldots, u_n)$ is called Bottom if it is always among the 2^{n-1} last n-tuples (for any basic probabilities p_i).

(iii) The binary n-tuple $(u_1, \ldots, u_n)$ is called Jumping if it is neither Top nor Bottom.

Denoting:

$$\begin{aligned} \mathcal{T}(n) &= \{(u_1, \ldots, u_n) \in \{0,1\}^n / (u_1, \ldots, u_n) \text{ is Top}\} \\ \mathcal{B}(n) &= \{(u_1, \ldots, u_n) \in \{0,1\}^n / (u_1, \ldots, u_n) \text{ is Bottom}\} \\ \mathcal{J}(n) &= \{(u_1, \ldots, u_n) \in \{0,1\}^n / (u_1, \ldots, u_n) \text{ is Jumping}\}. \end{aligned} \tag{10}$$

Definition 2 leads to the following partition of the set of n-tuples of 0s and 1s:

$$\{0,1\}^n = \mathcal{T}(n) \cup \mathcal{J}(n) \cup \mathcal{B}(n). \tag{11}$$

Now, we characterise the Top, Bottom and Jumping n-tuples, by a simple and elegant criterion. For this purpose, we need to state the following definition.

Definition 3. *For all* $(u_1, \ldots, u_n) \in \{0,1\}^n$, *the complementary of* $(u_1, \ldots, u_n)$ *is the* n*-tuple:*

$$(u_1, \ldots, u_n)^c = (1 - u_1, \ldots, 1 - u_n). \tag{12}$$

Obviously, the sum of two complementary n-tuples is always $(1, \overset{n}{\ldots}, 1) \equiv 2^n - 1$. Then, we can observe in Fig. 2 that the complementary 3-tuples are always allocated at symmetric positions with respect to the medium line. Indeed, this fact is an immediate consequence of the following result.

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

$$\Pr\{u\} \geq \Pr\{v\} \Leftrightarrow \Pr\{u^c\} \leq \Pr\{v^c\}. \tag{13}$$

Proof. For all $u = (u_1, \ldots, u_n), v = (v_1, \ldots, v_n) \in \{0,1\}^n$, from (2) we get:

$$\Pr\{u\} \geq \Pr\{v\} \Leftrightarrow \prod_{i=1}^{n} p_i^{u_i} (1 - p_i)^{1-u_i} \geq \prod_{i=1}^{n} p_i^{v_i} (1 - p_i)^{1-v_i} \Leftrightarrow$$

$$\prod_{i=1}^{n} p_i^{(1-v_i)-(1-u_i)} (1 - p_i)^{v_i - u_i} \geq 1 \Leftrightarrow$$

$$\prod_{i=1}^{n} p_i^{1-u_i} (1 - p_i)^{1-(1-u_i)} \leq \prod_{i=1}^{n} p_i^{1-v_i} (1 - p_i)^{1-(1-v_i)} \Leftrightarrow \Pr\{u^c\} \leq \Pr\{v^c\} \quad \square$$

From Lemma 1 we obtain the following nice characterisation of Top, Bottom and Jumping n-tuples.

Theorem 2. *For all* $(u_1, \ldots, u_n) \in \{0,1\}^n$:

(i) $(u_1, \ldots, u_n) \in \mathcal{T}(n)$ *iff* $(u_1, \ldots, u_n)$ *has not any* 1, *or for each* 1 *there exists (at least) one corresponding preceding* 0 *(Condition T).*

(ii) $(u_1, \ldots, u_n) \in \mathcal{B}(n)$ *iff* $(u_1, \ldots, u_n)$ *has not any* 0, *or for each* 0 *there exists (at least) one corresponding preceding* 1 *(Condition B).*

(iii) $(u_1, \ldots, u_n) \in \mathcal{J}(n)$ *iff* $(u_1, \ldots, u_n)$ *has at least one* 1 *without its corresponding preceding* 0, *and at least one* 0 *without its corresponding preceding* 1 *(Condition J).*

Remark 4. The term *corresponding,* used in Theorem 2, has the same meaning that the one explained in Remark 3 for Theorem 1: Condition T (B) requires that for each two 1s (0s) in $(u_1, \ldots, u_n)$ there must exist (at least) two different 0s (1s) preceding to each other.

Proof (of Theorem 2). (i) Using Lemma 1, we have:

$$(u_1,\dots,u_n)\in\mathcal{T}(n)\Leftrightarrow(u_1,\dots,u_n)^c\in\mathcal{B}(n)\Leftrightarrow$$

$$(u_1,\dots,u_n)\succeq(u_1,\dots,u_n)^c\Leftrightarrow$$

$$\begin{pmatrix} u_1 & \dots & u_n \\ 1-u_1 & \dots & 1-u_n \end{pmatrix} \text{ satisfies } IOC \Leftrightarrow (u_1,\dots,u_n) \text{ satisfies } T.$$

(ii) Using (i), we have:

$$(u_1,\dots,u_n)\in\mathcal{B}(n)\Leftrightarrow(u_1,\dots,u_n)^c\in\mathcal{T}(n)\Leftrightarrow$$

$$(u_1,\dots,u_n)^c \text{ satisfies } T\Leftrightarrow(u_1,\dots,u_n) \text{ satisfies } B.$$

(iii) Using (i) and (ii), we have:

$$(u_1,\dots,u_n)\in\mathcal{J}(n)\Leftrightarrow(u_1,\dots,u_n)\notin\mathcal{T}(n)\cup\mathcal{B}(n)\Leftrightarrow$$

$$(u_1,\dots,u_n) \text{ does not satisfy } T \text{ and does not satisfy } B \ \square$$

Example 3. For $n=1,2,3$ we obtain:

$$\mathcal{T}(1)=\{(0)\}\ ;\ \mathcal{J}(1)=\emptyset\ ;\ \mathcal{B}(1)=\{(1)\}. \tag{14}$$

$$\mathcal{T}(2)=\{(0,0),(0,1)\}\ ;\ \mathcal{J}(2)=\emptyset\ ;\ \mathcal{B}(2)=\{(1,0),(1,1)\}. \tag{15}$$

$$\mathcal{T}(3)=\{(0,0,0),(0,0,1),(0,1,0)\}\ ;\ \mathcal{B}(3)=\{(1,0,1),(1,1,0),(1,1,1)\}$$
$$\mathcal{J}(3)=\{(0,1,1),(1,0,0)\}. \tag{16}$$

In the following, we restrict to the Top n-tuples, since they correspond to the binary states with largest probabilities. The following theorem states three elementary necessary conditions for Top n-tuples.

Theorem 3. *For all* $u=(u_1,\dots,u_n)\in\mathcal{T}(n)$:

(i) $u_1=0$

(ii) The number of 0s of u *is greater than or equal to its number of 1s.*

(iii) $(u_1,\dots,u_k)\in\mathcal{T}(k)\ ;\ \forall k=1,2,\dots,n$

Proof. Using Theorem 2-(i) the proof is straightforward $\square$

Identifying all the binary states with the same decimal (lexicographic) numbering, for any number of components, i.e.:

$$(u_1,\dots,u_n)\equiv(0,u_1,\dots,u_n)\equiv(0,0,u_1,\dots,u_n)\equiv\dots \tag{17}$$

the following theorem states that the sets of Top n-tuples can be disposed in an increasing chain for inclusion.

Theorem 4. $\forall n\in N:\ \mathcal{T}(1)\subset\mathcal{T}(2)\subset\dots\subset\mathcal{T}(n)\subset\mathcal{T}(n+1)\subset\dots$

Proof. Using Theorem 2-(i), we have:

$$\forall n\in N:\ (u_1,\dots,u_n)\in\mathcal{T}(n)\Rightarrow(0,u_1,\dots,u_n)\in\mathcal{T}(n+1)\ \square$$

Now, we can present in Sect. 4 an efficient algorithm to generate all the Top n-tuples.

4 Generation of Top n-Tuples

The simple description T of Top n-tuples (Theorem 2-(i)) allows us to present a simple algorithm for their generation (illustrated in Fig. 3). Starting from the only Top 1-tuple (see subgraph $n = 1$ in Fig.1, or Example 3):

$$\mathcal{T}(1) = \{(0)\}. \tag{18}$$

we only need adding, in an adequate way, 0 (1) at the end of all (some) Top $n-1$-tuples, to obtain all the Top n-tuples. More precisely:

Theorem 5. *For all $n \geq 2$, we have:*

(i) If $n-1$ is odd, then, adding 0 at the end of all the Top $n-1$-tuples and adding 1 at the end of all the Top $n-1$-tuples, we obtain all the Top n-tuples.

(ii) If $n-1$ is even, then, adding 0 at the end of all the Top $n-1$-tuples and adding 1 at the end of the Top $n-1$-tuples for which the number of 0s is greater than the number of 1s, we obtain all the Top n-tuples.

Proof. First, we prove that the addition of 0s and 1s, in the described way, always generates different Top n-tuples. Taking into account the characterisation (T) of the Top binary states given in Theorem 2-(i), we deduce:
- On one hand, for both cases (i) and (ii), the addition of one 0 at the end of any Top $n-1$-tuple obviously leads to a Top n-tuple.
- On the other hand, if $n-1$ is odd (case (i)) then, all the Top $n-1$-tuples have at least one 0 which is not preceding to any corresponding 1. The same happens, when $n-1$ is even, for those Top $n-1$-tuples for which the number of 0s is greater than the number of 1s (case (ii)). Then, for both cases, the addition of one 1 at the end of the described Top $n-1$-tuples also leads to a Top n-tuple.
- Obviously, all the obtained Top n-tuples are different because, if they are generated by the same $n-1$-tuple then, they differ in the n^{th} component.

Second, we must prove that all the Top n-tuples can be generated by the described addition algorithm. For both cases (i) and (ii), we have: If $(u_1, \ldots, u_n) \in \mathcal{T}(n)$, then suppressing the last component u_n $(u_n = 0, 1)$ we get $(u_1, \ldots, u_{n-1})$. Theorem 3-(iii) guarantees that $(u_1, \ldots, u_{n-1}) \in \mathcal{T}(n-1)$ and, obviously, this is the Top $n-1$-tuple which generates $(u_1, \ldots, u_n)$ □

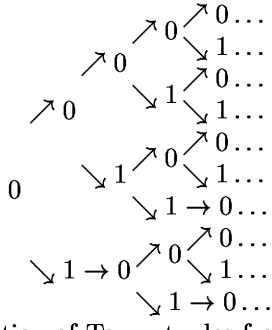

Fig. 3. Generation of Top n-tuples from $\mathcal{T}(1) = \{(0)\}$

The number of rows in the n^{th} column ($n = 1, 2, 3, 4, 5, \ldots$) is the number of Top n-tuples, which can be written following the arrows from the first column (0) to each row of the n^{th} column. Obviously, all Top n-tuples (for each natural number n) begin with 0 (Theorem 3-(i)) and they are generated following lexicographic order.

Example 4. For $n = 5$ we have:

$$\mathcal{T}(5) = \left\{ \begin{array}{l} (0,0,0,0,0), (0,0,0,0,1), (0,0,0,1,0), (0,0,0,1,1), (0,0,1,0,0), \\ (0,0,1,0,1), (0,0,1,1,0), (0,1,0,0,0), (0,1,0,0,1), (0,1,0,1,0) \end{array} \right\}.$$

According to the identification (17), next theorem is to justify that the above algorithm selects the binary states by decreasing order of their occurrence probabilities. More precisely:

Theorem 6. *The occurrence probability of any Top n-tuple generated from the Top $n-1$-tuple $(u_1, \ldots, u_{n-1})$, by the algorithm described in Theorem 5, is less than or equal to* $\Pr\{(0, u_1, \ldots, u_{n-1})\}$ *(for any basic probabilities verifying* (4)*).*

Proof. Indeed, the assertion stated by this theorem is more general. We shall prove that, according to the identification (17), the addition of 0 or 1 at the end of any $n-1$-tuple, always produces n-tuples with minor occurrence probabilities (for any basic probabilities p_i such that (4)), i.e. $\forall (u_1, \ldots, u_{n-1}) \in \{0,1\}^{n-1}$:

$$(0, u_1, \ldots, u_{n-1}) \succeq (u_1, \ldots, u_{n-1}, 0) \succeq (u_1, \ldots, u_{n-1}, 1). \tag{19}$$

Using the matrix description IOC of the intrinsic order (Theorem 1) the second inequality is obvious. To prove the first one, let us suppose that matrix

$$\begin{pmatrix} 0 & u_1 & \ldots & u_{n-2} & u_{n-1} \\ u_1 & u_2 & \ldots & u_{n-1} & 0 \end{pmatrix}. \tag{20}$$

does not satisfy IOC. Then, matrix (20) contains (at least) one $\binom{1}{0}$ column without its corresponding preceding $\binom{0}{1}$ column. For a homogeneous notation we define $u_n = 0$, and let $\binom{u_{m-1}}{u_m}$ be the left-most $\binom{1}{0}$ column in matrix (20) without its corresponding preceding $\binom{0}{1}$ column. Then, attending to the particular pattern of matrix (20), we have:

$$u_{m-1} = 1 \Rightarrow u_{m-2} = 1 \Rightarrow \cdots \Rightarrow u_1 = 1.$$

and thus, $\binom{u_{m-1}}{u_m}$ has its corresponding preceding $\binom{0}{1}$ column: the first column of matrix (20). But this contradicts the hypothesis on $\binom{u_{m-1}}{u_m}$ □

5 Conclusions

We have justified the convenience of selecting binary n-tuples with occurrence probabilities as large as possible in Reliability Theory, Risk Analysis and in many

other application areas with stochastic Boolean structures. Using an intrinsic order criterion (IOC) for binary n-tuples probabilities [4], we have classified the binary n-tuples of 0s and 1s into three groups: Top, Bottom and Jumping n-tuples. We have characterised these three types of n-tuples, attending to the relative positions of their 0s and 1s. The Top n-tuples correspond to the binary states with largest occurrence probabilities and we have constructed a simple algorithm for their generation. Starting from the only Top 1-tuple (0), this algorithm obtains all the Top n-tuples just by adding 0 (1) at the end of all the (some) previous Top $n-1$-tuples. The algorithm selects the binary states respecting lexicographic order, and by decreasing order of their occurrence probabilities. The main advantage is that the proposed method is exclusively based on the positions of 0s and 1s in the binary n-tuples, but it selects the binary states with the intrinsically largest occurrence probabilities, that is, for any (basic) probabilities of the Boolean variables.

References

1. Coudert, O., Madre, J.C.: Metaprime: An Interactive Fault-Tree Analyzer. IEEE Trans. Reliability **43**(1) (1994) 121-127
2. Fussell, J.B.: A Formal Methodology for Fault-Tree Construction. Nuclear Ebg. And Design **52** (1973) 337-360
3. Galván, B.J., García, D., González, L.: Un Método Determinista para la Evaluación de Funciones Booleanas Estocásticas: Análisis Comparativo. In: Font, J., Jorba, A. (eds.): Proc. XIV CEDYA -IV CMA. Barcelona (1995) 257-258
4. González, L., García, D., Galván, B.J.: An Intrinsic Order Criterion to Evaluate Large, Complex Fault Trees. IEEE Trans. Reliability, to appear.
5. Kumamoto, H., Henley, E.: Top-Down Algorithm for Obtaining Prime Implicants Sets on Non Coherent Fault Trees. IEEE Trans. Reliability **27**(4) (1978) 242-249
6. Kumamoto, H., Tanaka, T., Inoue, K., Henley, E.: Dagger Sampling Montecarlo for System Unavailability Evaluation. IEEE Trans. Reliability **29**(2) (1980) 122-125
7. Page, L.B., Perry, J.E.: Direct Evaluation Algorithms for Fault-Tree Probabilities. Computers in Chemical Engineering **15**(3) (1991) 157-169
8. Rai, S.: A Direct Approach to Obtain Tighter Bounds for Large Fault Trees with Repeated Events. In: Evans Associates (eds.): Proc. Annual. Rel.& Maint. Symposium, Vol. 23. North Carolina (1994) 475-480
9. Schneeweiss, W.G.: Boolean Functions with Engineering Applications and Computer Programs. Springer-Verlag, Berlin Heidelberg New York (1989)