

Switching-Algebraic Analysis of System Reliability

Ali Muhammad Rushdi and Mahmoud Ali Rushdi

Abstract This chapter deals with the paradigm of handling system reliability analysis in the Boolean domain as a supplement to (rather than a replacement to) analysis in the probability domain. This paradigm is well-established within the academic circles of reliability theory and engineering, albeit virtually unknown outside these circles. The chapter lists and explains arguments in favor of this paradigm for systems described by verbal statements, fault trees, block diagrams, and network graphs. This is followed by a detailed exposition of the pertinent concept of the Real or Probability Transform of a switching (two-valued Boolean) function, and that of a Probability-Ready Expression (*PRE*). Some of the important rules used in generating a *PRE* are presented, occasionally along with succinct proofs. These include rules to *achieve* disjointness (orthogonality) of ORed formulas, and to *preserve* statistical independence, as much as possible, among ANDed formulas. Recursive relations in the Boolean domain are also discussed, with an application to the four versions of the AR algorithm for evaluating the reliability and unreliability of the k-out-of-n:G and the k-out-of-n:F systems. These four versions of the algorithm are explained in terms of signal flow graphs that are compact, regular, and acyclic, in addition to being isomorphic to the Reduced Ordered Binary Decision Diagram (ROBDD). An appendix explains some important properties of the concept of Boolean quotient, whose expectation in Boolean-based probability is the counterpart of conditional probability in event-based probability.

A.M. Rushdi (✉)
Department of Electrical and Computer Engineering,
Faculty of Engineering, King Abdulaziz University,
P.O. Box 80204, Jeddah 21589, Saudi Arabia
e-mail: arushdi@kau.edu.sa

M.A. Rushdi
Department of Electronics and Communications Engineering,
Faculty of Engineering, Cairo University, Giza 12613, Egypt
e-mail: mahmoud.rushdi@gmail.com

Keywords System reliability · Boolean domain · Probability transform · Probability-ready expression · Recursive relation · Signal flow graph · k-out-of-n system

1 Introduction

This chapter deals with the paradigm of handling system reliability analysis in the Boolean domain as a supplement to (rather than a replacement to) analysis in the probability domain. This paradigm is called *switching-algebraic analysis of reliability* or (two-valued) Boolean analysis of reliability. It is also termed *Logical Probabilistic Analysis (LPA)* by Ryabinin [1], who traced it back to great pioneers such as Boole, Poretskii, and Bernstein. Ryabinin also rebutted dubious criticism cast against *LPA*, and lamented that *LPA* is missing in prominent publications on probability theory, including those by celebrated scholars such as Markov and Kolmogorov. A modern revival of *LPA* was started in 1963 by Merekin [2] and Premo [3] and later in 1973 by Fratta and Montanari [4]. After that, applications of *LPA* in system reliability analysis were covered by literally hundreds of publications in prestigious journals (see, e.g., [5–62]). It is therefore astonishing that *LPA* has so far not found its way to popular texts on probability (see, e.g., [63–72]), and is still virtually unknown outside the circles of reliability theorists and engineers. This state of affairs is possibly an obstacle hindering fruitful interplay between system reliability and many of its potential fields of application. This chapter is, therefore, an attempt to review reliability applications of *LPA* and enhance awareness of the scientific community about it. It is also an invitation to explore its utility in other parts of probability theory that deal with generalized Bernoulli trials.

System reliability analysis deals with expressing the reliability of a system in terms of the reliabilities of its constituent components. Hence, this analysis entails advanced applications of probability theory, and is consequently “based on the algebra of events (a version of set algebra), which is isomorphic to the bivalent or 2-valued Boolean algebra (switching algebra)” [56]. Besides using the algebra of events in what is termed probabilistic or arithmetic analysis [73, 74], modern system reliability analysis also utilizes Boolean algebra by employing the indicator variables for probabilistic events instead of the events themselves. Subsequently, expectations of these indicator variables are used to represent probabilities of the corresponding events. In particular, expectations of system success and failure replace system reliability and unreliability, respectively.

An approach bearing certain similarities to the paradigm exposed herein (without going to the Boolean domain) studies probability via expectations [65]. It is based on an axiomatization of probability theory using expectations instead of probabilities (i.e., using linear operators instead of measures). A related approach is suggested by Feller [63, p. 219] who states that “*The reduction of probability theory to random variables is a short-cut to the use of analysis and simplifies the theory in many ways. However, it also has the drawback of obscuring the probability*

background.” The paradigm exposed herein handles probability in terms of indicator random Bernoulli variables, but it does not obscure the probability background, since it stresses that the random variables are indicators for events, and it keeps two parallel schemes of event-based probability and Boolean-based probability. These schemes are compared in Table 1 which cites well-known results from elementary probability theory. The first scheme utilizes the concepts of probabilities of events, union and intersection operators, and conditional probability, while the second scheme replaces these by expectations of event indicators, logical “OR” and “AND” operators, and Boolean quotient (see Appendix A). Table 1 shows how probability formulas involving the union and intersection operators become simpler when the pertinent events are mutually exclusive or statistically independent, respectively. As indicated in Table 2, the indicator variable I_A is a Bernoulli variable of two values 0 and 1, such that

$$I_A = 0 \quad \text{iff the event } A \text{ does not occur,} \quad (1a)$$

$$I_A = 1 \quad \text{iff the event } A \text{ occurs.} \quad (1b)$$

Table 2 also indicates definitions of the Bernoulli variables \bar{I}_A , $I_A \vee I_B$, $I_A \wedge I_B$, so that the following expectations are obtained

$$E\{I_A\} = Pr\{A\}, \quad (2a)$$

$$E\{\bar{I}_A\} = Pr\{\bar{A}\}, \quad (2b)$$

$$E\{I_A \vee I_B\} = Pr\{\bar{A} \cap B\} + Pr\{A \cap \bar{B}\} + Pr\{A \cap B\} = Pr\{A \cup B\}, \quad (2c)$$

$$E\{I_A \wedge I_B\} = Pr\{A \cap B\}. \quad (2d)$$

In the sequel, we will use the symbols S and \bar{S} to denote system success and failure, respectively, with their expectations $R = E\{S\}$ and $U = E\{\bar{S}\}$ representing system reliability and unreliability, respectively. Subscripts of obvious meanings will be added to distinguish systems of particular types. The corresponding symbols X_i , \bar{X}_i , $p_i = E\{X_i\}$, and $q_i = E\{\bar{X}_i\}$ will denote the success, failure, reliability, and unreliability, respectively, for component of number i , $1 \leq i \leq n$. Unless otherwise stated, component successes are assumed statistically independent all throughout this chapter. A Boolean expression for the indicator of system success will be presented as a function of the indicators of component successes. Transition from the Boolean domain to the probability domain is achieved via the *Real Transform* [75–82] which can be viewed as an expression of system reliability (unreliability) as a function of component reliabilities (unreliabilities).

The organization of the rest of this chapter is as follows. Section 2 lists arguments for handling system reliability in the Boolean domain before going to the probability domain. Section 3 reviews the basic concepts of the Real or Probability Transform of a switching function. Section 4 presents the definition of a

Table 1 Comparison of event-based and boolean-based probability relations

	Event-based probability	Boolean-based probability
Union (disjunction/ORing)	$Pr(A \cup B) = Pr(A) + Pr(B) - Pr(A \cap B)$	$E(I_A \vee I_B) = E(I_A) + E(I_B) - E(I_A \wedge I_B)$
	$Pr(A \cup B) = Pr(A) + Pr(B)$ when A and B are mutually-exclusive events	$E(I_A \vee I_B) = E(I_A) + E(I_B)$ when I_A and I_B are disjoint/orthogonal indicators
Intersection (conjunction/ANDing)	$Pr(A \cap B) = Pr(A)Pr(B A)$ where $Pr(B A)$ denotes the conditional probability of B given A	$E(I_A \wedge I_B) = E(I_A)E(I_B/I_A)$ where (I_B/I_A) denotes the Boolean quotient of I_B given I_A
	$Pr(A \cap B) = Pr(A)Pr(B)$ when A and B are statistically – independent events	$E(I_A \wedge I_B) = E(I_A)E(I_B)$ when I_A and I_B are statistically – independent indicators
Total probability	$Pr\{A\} = \sum_{i=1}^m Pr\{A B_i\} Pr\{B_i\}$ where the events B_i are mutually exclusive and exhaustive	$E\{I_A\} = \sum_{i=1}^m E\{I_A/I_{B_i}\}E\{I_{B_i}\}$ where the indicators I_{B_i} constitute an orthonormal set
Bayes' theorem	$Pr(B_i A) = Pr(A \cap B_i) / Pr(A)$, $Pr(A) \neq 0$.	$E\{I_{B_i}/I_A\} = E(I_A \wedge I_{B_i})/E(I_A)$, $E(I_A) \neq 0$.

Table 2 Indicator variable for set complementation

A	\bar{A}	I_A	\bar{I}_A
Does not occur	Occurs	0	1
Occurs	Does not occur	1	0

Table 3 Indicator variables for set union and intersection

A	B	$A \cup B$	$A \cap B$	I_A	I_B	$I_A \vee I_B$	$I_A \wedge I_B$
Does not occur	Does not occur	Does not occur	Does not occur	0	0	0	0
Does not occur	Occurs	Occurs	Does not occur	0	1	1	0
Occurs	Does not occur	Occurs	Does not occur	1	0	1	0
Occurs	Occurs	Occurs	Occurs	1	1	1	1

Probability-Ready Expression (*PRE*) and lists some of the rules used in generating a *PRE*. Section 5 discusses recursive relations in the Boolean domain. Section 6 concludes the chapter (Table 3).

2 Advantages of Working in the Switching (Boolean) Domain

There is a large potpourri of ways for describing a reliability system, including verbal statements, fault trees, block diagrams, and network graphs. Generally, reliability analysis of a system conducted in the probabilistic (arithmetic) domain is lengthy and error-prone [73, 74]. We will now discuss other arguments for working in the switching (Boolean) domain for each specific type of system description:

- For systems described by *verbal statements*, work in the Boolean domain allows a separation and distinction between the mathematical formalization of a reliability problem, and the algorithmic derivation of its formal solution. Table 4 shows the verbal description versus the Boolean specification of some common reliability systems, mostly of the k-out-of-n or related types. For lack of space, we omitted more sophisticated systems including some dual, multidimensional, multi-valued extensions of the reported systems. It is clear that the word statement and the mathematical description of each reported system are immediately and obviously equivalent. There is no mental strain or conceptual difficulty in going from the somewhat vague realm of language to the exactly precise arena of mathematics. The primary Boolean description is in terms of

Table 4 Boolean specification of some common reliability systems

System name	Verbal description	Primary boolean specification	Secondary (complementary) boolean specification
Series (nonredundant) system	“A system that is good iff all its n components are good” [56]	$S_S = \bigwedge_{i=1}^n X_i$	$\bar{S}_S = \bigvee_{i=1}^n \bar{X}_i$
Parallel (totally-redundant) system	“A system that is failed iff all its n components are failed” [56]	$\bar{S}_P = \bigwedge_{i=1}^n \bar{X}_i$	$S_P = \bigvee_{i=1}^n X_i$
k -out-of- n :G (partially-redundant) system ($1 \leq k \leq n$)	“A system that is good iff at least k out of its n components are good” [27, 39, 61]	$S_G(k, n) = \bigvee_{K \subseteq \mathbf{K}} (\bigwedge_{i \in K} X_i)$	$\bar{S}_G(k, n) = \bigwedge_{K \subseteq \mathbf{K}} (\bigvee_{i \in K} \bar{X}_i)$
k -out-of- n :F (Partially-redundant) system ($1 \leq k \leq n$)	“A system that is failed iff at least k out of its n components are failed” [27, 39, 61]	$\bar{S}_F(k, n) = \bigvee_{K \subseteq \mathbf{K}} (\bigwedge_{i \in K} \bar{X}_i)$	$S_F(k, n) = \bigwedge_{K \subseteq \mathbf{K}} (\bigvee_{i \in K} X_i)$
Linear consecutive- k -out-of- n :F system	“A sequence of n components arranged in a straight line such that the system fails iff at least k consecutive components fail” [32, 52, 83–85]	$\bar{S}_L(k, n) = \bigvee_{i=1}^{n-k+1} (\bigwedge_{j=1}^k \bar{X}_{i+j-1})$	$S_L(k, n) = \bigwedge_{i=1}^{n-k+1} (\bigvee_{j=1}^k X_{i+j-1})$
Circular consecutive- k -out-of- n :F system	“A sequence of n components arranged on a circle such that the system fails iff at least k consecutive components fail” [33, 52, 83–85]	$\bar{S}_C(k, n) = \bigvee_{i=1}^n (\bigwedge_{j=1}^k \bar{X}_{(i+j-1)})$	$S_C(k, n) = \bigwedge_{i=1}^n (\bigvee_{j=1}^k X_{(i+j-1)})$
(n, f, k) system	“A sequence of n components arranged in a straight line or a circle such that the system fails iff at least f components fail OR at least k consecutive components fail” [52, 88–91]	$\bar{S}_{OL}(n, f, k) = \bar{S}_F(f, n) \vee \bar{S}_L(k, n)$ $\bar{S}_{OC}(n, f, k) = \bar{S}_F(f, n) \vee \bar{S}_C(k, n)$	$S_{OL}(n, f, k) = S_F(f, n) \wedge S_L(k, n)$ $S_{OC}(n, f, k) = S_F(f, n) \wedge S_C(k, n)$
$<n, f, k >$ system	“A sequence of n components arranged in a straight line or a circle such that the system fails iff at least f components fail AND at least k consecutive components fail” [89, 91]	$\bar{S}_{AL}(n, f, k) = \bar{S}_F(f, n) \wedge \bar{S}_L(k, n)$ $\bar{S}_{AC}(n, f, k) = \bar{S}_F(f, n) \wedge \bar{S}_C(k, n)$	$S_{AL}(n, f, k) = S_F(f, n) \vee S_L(k, n)$ $S_{AC}(n, f, k) = S_F(f, n) \vee S_C(k, n)$
Threshold system (weighted k -out-of- n :G system)	“A system that is good iff a certain weighted sum of its component successes equals or exceeds a certain threshold” [36, 52, 53, 55, 60, 61, 86, 87]	Both S_T and \bar{S}_T are threshold switching functions $S_T(\mathbf{X}) = 1$ iff $\sum_{i=1}^n W_i X_i \geq T$ where T is the system threshold and W_i is the weight of component i	

system success or failure, depending on the nature of the system, with the complementary description delegated to a secondary role. Once in the Boolean domain, one can solve the problem of reliability evaluation by utilizing an extensive set of algorithmic tools and visual aids. For most of the systems in Table 4, both probabilistic and Boolean solutions exist. For some of the systems, such as the k -out-of- n systems, Boolean solutions [27] were earlier to develop. For other systems, such as the consecutive- k -out-of- n ones, Boolean solutions [32, 33] arrived later, as an afterthought, to provide a more insightful and easy-to-comprehend alternative.

- *Formulation* of a system reliability problem in the Boolean domain is the natural strategy to use for systems described by *fault trees* [9, 10, 44, 49, 50, 58], which are composed of logical AND–OR gates, and can similarly be utilized with systems described by *block diagrams* [15]. Fault trees are used to relate probabilistic events, with its inputs being basic events and its output being the top event. However, it could be more convenient to apply Boolean-based probability to fault trees by using indicators for events rather the events themselves [57, 62, 78, 92].
- *Formulation* in the Boolean domain is the method of choice with systems described by network graphs when these systems are handled through the enumeration of minimal pathsets and minimal cutsets [12, 23, 26, 48], rather than other graph techniques.

Work in the Boolean domain might allow the treatment of certain types of statistical dependencies among component successes. For example, in the case of *total* positive dependency ($X_i = X_j$) idempotency leads to $(X_i \vee X_j = X_i)$ and $(X_i \wedge X_j = X_i)$, while in the case of *total* negative dependency ($X_i = \bar{X}_j$) orthogonality leads to $(X_i \vee X_j = 1)$ and $(X_i \wedge X_j = 0)$. Cases of *partial* dependencies might be handled via expectations of Boolean quotients.

Work in the Boolean domain also provides the insight necessary to avoid traps and pitfalls, as well as to detect anomalies, inconsistencies, and errors of the third kind (errors of solving the wrong problem). A case in point is that of the so-called *strict consecutive- k -out-of- n : $G(F)$ system* [93–96], which lacks a definition that avoids ambiguity, inconsistency, or self-contradiction. Unfortunately, many capable and great mathematicians wasted their precious time producing elegant solutions for what they mistakenly assumed to be this system. Insight of the Boolean domain allowed a critical review of literature that partially prevented the publication of more such *irrelevant* work.

Without resort to the Boolean domain, one might be obliged to use the notorious *Inclusion–Exclusion (IE) Principle* to compute the probability of the union of n events [63, 67], where n might be large. Use of the IE Principle in system reliability analysis is undesirable, since “(a) it produces an exponential number of terms that have to be reduced subsequently via addition and cancellation, and (b) it involves too many subtractions, making it highly sensitive to round-off errors, and possibly leading to catastrophic cancellations” [52, 54, 56].

3 The Real Transform of a Switching Function

Various forms of the Real Transform (also called the Probability or Arithmetic transform) are discussed in [75–82]. The definition given in [79, 82] states that the real transform $R(\mathbf{p}) = R(p_1, p_2, \dots, p_n)$ of a switching function $f(\mathbf{X})$ possesses the following two properties:

- (a) $R(\mathbf{p})$ is a multiaffine continuous real function of continuous real variables $\mathbf{p} = [p_1 p_2 \dots p_n]^T$, i.e., $R(\mathbf{p})$ is a first-degree polynomial in each of its arguments p_i .
- (b) $R(\mathbf{p})$ has the same “truth table” as $f(\mathbf{X})$, i.e.

$$R(\mathbf{p} = \mathbf{t}_j) = f(\mathbf{X} = \mathbf{t}_j), \quad \text{for } j = 0, 1, \dots, (2^n - 1), \quad (3)$$

where \mathbf{t}_j is the j th input line of the truth table; \mathbf{t}_j is an n -vector of binary components such that

$$\sum_{i=1}^n 2^{n-i} t_{ji} = j, \quad \text{for } j = 0, 1, \dots, (2^n - 1). \quad (4)$$

We stress that property (b) above does not suffice to produce a unique $R(\mathbf{p})$ and it must be supplemented by the requirement that $R(\mathbf{p})$ be multiaffine to define $R(\mathbf{p})$ uniquely [77, 79]. We also note that if the Real Transform R and its arguments \mathbf{p} are restricted to discrete binary values (i.e., if $R : \{0, 1\}^n \rightarrow \{0, 1\}$) then R becomes the *multilinear form* of a switching function [11, 97]. This form is typically referred to as the *structure function* [98, 99] in system reliability, and is a way to mimic Boolean algebra in terms of arithmetic operators rather than Boolean ones.

The definition above for $R(\mathbf{p})$ implies that it is a function from the n -dimensional real space to the real line ($R(\mathbf{p}) : \mathbf{R}^n \rightarrow \mathbf{R}$). Though both R and \mathbf{p} could be free real values, they have very interesting interpretations as probabilities, i.e., when restricted to the $[0.0, 1.0]$ and $[0.0, 1.0]^n$ real intervals. An important property of the Real Transform $R(\mathbf{p})$ is that if its vector argument or input \mathbf{p} is restricted to the domain within the n -dimensional interval $[0.0, 1.0]^n$, i.e., if $0.0 \leq p_i \leq 1.0$ for $1 \leq i \leq n$, then the image of $R(\mathbf{p})$ will be restricted to the unit real interval $[0.0, 1.0]$.

The Real Transform is a bijective (one-to-one and onto) mapping from the set of switching functions to the subset of multiaffine functions such that if the function’s domain is the power binary set $\{0, 1\}^n$ then its image belongs to the binary set $\{0, 1\}$. Evidently, an $R(\mathbf{p})$ restricted to binary values whenever its arguments are restricted to binary values can produce the “truth table” that completely specifies its inverse image $f(\mathbf{X})$ via (3). On the other hand, a multiaffine function of n variables is completely specified by 2^n independent conditions [77, 79], e.g., the ones in (3).

The transform $R(\mathbf{p})$ is related to the original function $f(\mathbf{X})$ via the truth table in (3), which consists of 2^n lines. Therefore, the complexity of implementing the Real

Transform, i.e., of going from $f(X)$ to $R(p)$ is exponential, and the problem of implementing the Real Transform is generally intractable. However, if $f(X)$ is cast in *PRE* form, then it can be converted trivially (at no cost) to $R(p)$. In this case, the burden of achieving the Real Transform is shifted to the Boolean domain where the initial formula for $f(X)$ is to be converted (at a potentially reduced cost) to a *PRE*.

4 Probability-Ready Expressions

A Reliability-Ready Expression (*RRE*) is an expression in the switching (Boolean) domain that can be directly transformed, on a one-to-one basis, to its Real or Probability Transform by replacing switching (Boolean) indicators by their statistical expectations, and also replacing logical multiplication and addition (ANDing and ORing) by their arithmetic counterparts [25, 42, 49, 50, 54, 56]. We now present some useful rules that might be required in the conversion of an arbitrary switching (Boolean) expression to an equivalent compact *PRE*:

1. A switching expression is a *PRE* expression if
 - (a) all *Ored* terms are *orthogonal (disjoint)*, and
 - (b) all *ANDed* sums are *statistically independent*.

The conversion is achieved by replacing Boolean variables by their expectations, AND operations by arithmetic multiplications, and OR operations by arithmetic additions. While there are literally hundreds of methods to introduce characteristic (a) of orthogonality (disjointness) into a Boolean expression (see, e.g., Bennetts [9, 15], Abrahams [13], Dotson and Gobien [14], and Rushdi [17, 20]) there is no way to induce characteristic (b) of statistical independence. The best that one can do is to observe statistical independence when it exists, and then take care to preserve it and take advantage of it [25, 42, 62].

2. As an analog to series-parallel reduction, any set of ANDed or Ored variables that do not appear elsewhere in a formula should be combined into a single variable. This might considerably reduce the number of variables involved, and helps in producing a more compact expression.
3. Two terms are disjoint if they have at least a single *opposition*, i.e., if there is at least one variable that appears complemented in one term and appears un-complemented in the other, e.g., the two terms $A\bar{B}C$ and BC are disjoint since the complemented literal \bar{B} appears in $A\bar{B}C$ while the un-complemented literal B appears in BC .
4. If neither of the two terms A and B in the sum $(A \vee B)$ subsumes the other ($A \vee B \neq A$ and $A \vee B \neq B$) and the two terms are not disjoint ($A \wedge B \neq 0$), then B can be disjointed with A by the relation

$$\begin{aligned} A \vee B &= A \vee B(\overline{y_1 y_2 \dots y_e}) \\ &= A \vee B(\overline{y_1} \vee y_1 \overline{y_2} \vee \dots \vee y_1 y_2 \dots y_{e-1} \overline{y_e}), \end{aligned} \quad (5)$$

where $\{y_1, y_2, \dots, y_e\}$ is the set of literals that appear in the term A and do not appear in the term B . Note that the term B is replaced by e (≥ 1) terms that are disjoint with one another besides being disjoint with the term A [9, 15, 54, 56]. Formula (5) might be seen as an immediate extension of the familiar *Reflection Law* [100–105].

$$a \vee b = a \vee (b \wedge \bar{a}). \quad (5a)$$

A formal proof of (5) is possible via *perfect induction* over a *much reduced truth table* that consists of the $(e + 1)$ orthonormal cases $\{y_1 = 0\}$, $\{y_1 = 1, y_2 = 0\}$, $\{y_1 = y_2 = 1, y_3 = 0\}$, \dots , $\{y_1 = y_2 = \dots = y_{e-1} = 1, y_e = 0\}$, and $\{y_1 = y_2 = \dots = y_{e-1} = y_e = 1\}$. For the first e cases, the L.H.S. = the R.H.S = B . For the last case, the L.H.S. = $A/y_1 y_2 \dots y_e \vee B$, while the R.H.S. = $A/y_1 y_2 \dots y_e$. The two sides are equal since B subsumes $A/y_1 y_2 \dots y_e$, and hence is absorbed in it. The term $(A/y_1 y_2 \dots y_e)$ is the *Boolean quotient* obtained by restricting A through the assignment $\{y_1 = y_2 = \dots = y_e = 1\}$ (See Appendix A). In passing, we note that the roles of A and B can be reversed in (5), and that (5) does not guarantee minimality of the resulting disjointed expression [51]. Disjointness typically results in an (often dramatic) increase in the number of terms in sum-of-products (sop) switching expressions. However, there are more sophisticated disjointness techniques that achieve “shellability”, i.e., they obtain a disjoint sop expression that retains the same number of terms as the original sop expression [34, 46, 55, 105]. Notable examples of shellable expressions include those of the success or failure of a k -out-of- n :G or a k -out-of- n :F system [39], and that of the success of a coherent threshold system [55].

5. Given a term P and two sums of products A and B where A has no variables in common with P , while B shares some variables with P , then the logical product $(A \wedge B)$ is disjointed with P by disjointing B and P and keeping A unchanged [42].
6. The complements of a sum and a product are given by a product and a disjoint sum, namely, *PRE* versions of De Morgan’s Laws [102]

$$\overline{\bigvee_{i=1}^n A_i} = \bigwedge_{i=1}^n \bar{A}_i, \quad (6)$$

$$\overline{\bigwedge_{i=1}^n A_i} = \bar{A}_1 \vee A_1 \bar{A}_2 \vee \dots \vee A_1 A_2 \dots A_{n-1} \bar{A}_n. \quad (7)$$

A formal proof of (6) is achieved by considering just two exhaustive cases, namely: (a) the case of all A_i ’s being 0, for which the L.H.S. = the R.H.S = 1, and (b) the case when at least one A_i is 1, for which the L.H.S. = the R.H.S. = 0.

Likewise, a formal proof of (7) is obtained via *perfect induction* over the $(n + 1)$ orthonormal cases $\{A_1 = 0\}$, $\{A_1 = 1, A_2 = 0\}$, $\{A_1 = A_2 = 1, A_3 = 0\}$, ..., $\{A_1 = A_2 = \dots = A_{n-1} = 1, A_n = 0\}$, and $\{A_1 = A_2 = \dots = A_{n-1} = A_n = 1\}$. For the first n cases, the L.H.S. = the R.H.S. = 1. For the last case, the L.H.S. = the R.H.S. = 0.

7. The most effective way for converting a Boolean formula into a *PRE* form is the Boole–Shannon Expansion

$$f(\mathbf{X}) = (\bar{X}_i \wedge f(\mathbf{X}|0_i)) \vee (X_i \wedge f(\mathbf{X}|1_i)), \quad (8)$$

which expresses a (two-valued) Boolean function $f(\mathbf{X})$ in terms of its two subfunctions $f(\mathbf{X}|0_i)$ and $f(\mathbf{X}|1_i)$. These subfunctions are equal to the Boolean quotients $f(\mathbf{X})/\bar{X}_i$ and $f(\mathbf{X})/X_i$, and hence are obtained by restricting X_i in the expression $f(\mathbf{X})$ to 0 and 1, respectively. If $f(\mathbf{X})$ is a sop expression of n -variables, the two subfunctions $f(\mathbf{X}|0_i)$ and $f(\mathbf{X}|1_i)$ are functions of at most $(n - 1)$ variables. A formal proof of (8) is achieved by considering just two exhaustive cases, namely: (a) the case $\{X_i = 0\}$, for which the L.H.S. = the R.H.S. = $f(\mathbf{X}|0_i)$, and (b) the case $\{X_i = 1\}$, for which the L.H.S. = the R.H.S. = $f(\mathbf{X}|1_i)$. The expansion (8) serves our purpose very well. Once the subfunctions in (8) are expressed by *PRE* expressions, $f(\mathbf{X})$ will be also in *PRE* form, thanks to the facts that (a) The R.H.S. of (8) has two disjoint parts, with the first part containing the complemented literal \bar{X}_i and the second part containing the un-complemented literal X_i , and (b) Each of these two parts is a product of two statistically-independent entities. The Boole–Shannon Expansion in the Boolean domain is equivalent to the *Total Probability Theorem* [67] in the probability domain and to the *Factoring Theorem* [56] in the “Graph Domain”. A visual aid or tool to implement this expansion is provided by the Variable-Entered Karnaugh Map (VEKM) [17].

8. Let a success formula be written in the form

$$S = N_1 M_1 \vee N_2 M_2, \quad (9)$$

where the set of formulas $\{N_1, N_2\}$ is statistically independent of the set of formulas $\{M_1, M_2\}$. Expression (9) might result due to path enumeration via network decomposition [22, 106–109]. A *PRE* formula of S (that preserves the original statistical independence as much as possible) is [22]

$$S_{PRE} = N_{1(PRE)} M_{1(PRE)} \vee (\bar{N}_1 N_2)_{(PRE)} M_{2(PRE)} \vee (N_1 N_2)_{(PRE)} (\bar{M}_1 M_2)_{(PRE)}. \quad (10)$$

A formal proof of (10) is possible via *perfect induction* over a *much reduced truth table* that consists of only the three orthonormal cases $\{N_1 = 0\}$, $\{N_1 = 1, M_1 = 0\}$, and $\{N_1 = M_1 = 1\}$. For the first two cases, the L.H.S. = the

R.H.S. = $N_{2(PRE)}M_{2(PRE)}$. For the third case, the L.H.S. = the R.H.S = 1. Note that this proof entails a partitioning of the higher-level space of N_1 and M_1 rather than a partitioning of the space of the underlying arguments X .

9. Products might be kept compact by utilizing the following rules of *disjoint intelligent multiplication*

$$(a \vee x)(a \vee y) = a \vee \bar{a}xy, \quad (11)$$

$$(a \vee x)(\bar{a} \vee y) = \bar{a}x \vee ay. \quad (12)$$

Formula (11) results due to the absorption of the terms ax and ay in their subsumed term a , and then using the Reflection Law (5a). The term xy that would have appeared in formula (12) is deleted since it is the consensus of $\bar{a}x$ and ay , and hence it is covered by their disjunction $\bar{a}x \vee ay$. A corollary of (11) is

$$\overline{ab} \wedge \overline{ac} \wedge \overline{ad} = \bar{a} \vee \bar{a}b\bar{c}\bar{d}. \quad (13)$$

The complemented form of (13) is a ANDed with the complement of $\bar{b}\bar{c}\bar{d}$. This complemented form allows separating the common factor a in the formula without introducing the OR (\vee) operator in it, and hence leads to full utilization of statistical independence [25].

10. When one has a *partially-factored* expression of system success or failure such as the following one [56]

$$S = X_3 \vee X_7 \vee (X_5 \vee X_8)(X_4 \vee X_2X_9) \vee (X_1 \vee X_6)(X_2 \vee X_4X_9), \quad (14)$$

then it should be converted into *PRE* form via (10) without spoiling or expanding the factored form, so as to *preserve statistical independence* [22, 25, 42, 56]

$$S_{PRE} = X_3 \vee \bar{X}_3(X_7 \vee \bar{X}_7((X_5 \vee \bar{X}_5X_8)(X_4 \vee \bar{X}_4X_2X_9) \vee (X_1 \vee \bar{X}_1X_6)(\bar{X}_5\bar{X}_8(X_2 \vee \bar{X}_2X_4X_9) \vee (X_5 \vee \bar{X}_5X_8)X_2\bar{X}_4\bar{X}_9))). \quad (15)$$

This transforms on a one-to-one basis to the reliability expression [56]

$$R = p_3 + q_3(p_7 + q_7((p_5 + q_5p_8)(p_4 + q_4p_2p_9) + (p_1 + q_1p_6)(q_5q_8(p_2 + q_2p_4p_9) + (p_5 + q_5p_8)p_2q_4q_9))). \quad (16)$$

5 Recursive Relations in the Boolean Domain

This Section provides a novel discussion about the utility of the Boolean domain in deriving recursive relations for the two versions of the k-out-of-n system. The reader is referred to [32, 33] for Boolean-domain derivations of recursive relations for consecutive systems. The success $S_G(k, j, \mathbf{X}_j)$ of a k-out-of-j:G system is a symmetric monotonically nondecreasing switching function, and hence satisfies recursive relations obeyed by such a function [27]. Equivalently, it is given by its Boole–Shannon expansion about its last argument X_j (namely, Eq. (5.40a) of Rushdi [39]):

$$S_G(k, j, \mathbf{X}_j) = \overline{X_j}S_G(k, j-1, \mathbf{X}_{j-1}) \vee X_jS_G(k-1, j-1, \mathbf{X}_{j-1}), 1 \leq k \leq j \leq n, \quad (17a)$$

where $\mathbf{X}_j = [X_1, X_2, \dots, X_j]^T$ is the vector of the first j component successes. The R.H.S. of Eq. (17a) involves two subfunctions $S_G(k, j-1, \mathbf{X}_{j-1})$ and $S_G(k-1, j-1, \mathbf{X}_{j-1})$ of the original success function. The region of validity of (17a) in the kj -plane is bordered by two straight lines on which the following two boundary conditions hold

$$S_G(k, j, \mathbf{X}_j) = 1, \quad k = 0, j \geq 1, \quad (17b)$$

$$S_G(k, j, \mathbf{X}_j) = 0, \quad k = j+1, j \geq 1, \quad (17c)$$

Equations (17) are in *PRE* form, with Probability Transforms

$$R_G(k, j, \mathbf{p}_j) = q_jR_G(k, j-1, \mathbf{p}_{j-1}) + p_jR_G(k-1, j-1, \mathbf{p}_{j-1}), \quad 1 \leq k \leq j \leq n, \quad (18a)$$

$$R_G(k, j, \mathbf{p}_j) = 1, \quad k = 0, j \geq 1, \quad (18b)$$

$$R_G(k, j, \mathbf{p}_j) = 0, \quad k = j+1, j \geq 1, \quad (18c)$$

which govern the reliability of the k-out-of-j:G system. Based on the recursive relation (18a) together with the boundary conditions (18b and 18c), a quadratic-time *iterative* algorithm was developed by Rushdi [27], and later named the AR algorithm [37, 39]. The AR algorithm has the beautiful characteristic of having the same form (and hence same complexity) for computing both the reliability and unreliability of either the k-out-of-n:G system or its dual the k-out-of-n:F system. Table 5 illustrates this point by listing various recursive relations together with boundary conditions for the two dual types of k-out-of-n systems in the Boolean and probabilistic domains. Entries in the lower two rows of Table 5 are Probability Transforms of the corresponding entries of the upper two rows, which happen to be in *PRE* form. Entries in the left column of Table 5 are inversions of the

Table 5 Various recursive relations together with boundary conditions for the two dual types of k-out-of-n systems in the Boolean and probabilistic domains

k-out-of-n:G system	$\begin{aligned} \bar{S}_G(k, j, \mathbf{X}_j) = & \bar{X}_j \bar{S}_G(k, j-1, \mathbf{X}_{j-1}) \vee \\ & X_j \bar{S}_G(k-1, j-1, \mathbf{X}_{j-1}), 1 \leq k \leq \\ & j \leq n, \\ \bar{S}_G(k, j, \mathbf{X}_j) = 0, & k = 0, j \geq 1, \\ \bar{S}_G(k, j, \mathbf{X}_j) = 1, & k = j+1, j \geq 1, \end{aligned}$	$\begin{aligned} S_G(k, j, \mathbf{X}_j) = & \bar{X}_j S_G(k, j-1, \mathbf{X}_{j-1}) \vee \\ & X_j S_G(k-1, j-1, \mathbf{X}_{j-1}), 1 \leq k \leq \\ & j \leq n, \\ S_G(k, j, \mathbf{X}_j) = 1, & k = 0, j \geq 1, \\ S_G(k, j, \mathbf{X}_j) = 0, & k = j+1, j \geq 1, \end{aligned}$
k-out-of-n:F system	$\begin{aligned} S_F(k, j, \mathbf{X}_j) = & X_j S_F(k, j-1, \mathbf{X}_{j-1}) \vee \\ & \bar{X}_j S_F(k-1, j-1, \mathbf{X}_{j-1}), 1 \leq k \leq \\ & j \leq n, \\ S_F(k, j, \mathbf{X}_j) = 0, & k = 0, j \geq 1, \\ S_F(k, j, \mathbf{X}_j) = 1, & k = j+1, j \geq 1, \end{aligned}$	$\begin{aligned} \bar{S}_F(k, j, \mathbf{X}_j) = & \bar{X}_j \bar{S}_F(k, j-1, \mathbf{X}_{j-1}) \vee \\ & X_j \bar{S}_F(k-1, j-1, \mathbf{X}_{j-1}), 1 \leq k \leq \\ & j \leq n, \\ \bar{S}_F(k, j, \mathbf{X}_j) = 1, & k = 0, j \geq 1, \\ \bar{S}_F(k, j, \mathbf{X}_j) = 0, & k = j+1, j \geq 1, \end{aligned}$
k-out-of-n:G system	$\begin{aligned} U_G(k, j, \mathbf{p}_j) = & q_j U_G(k, j-1, \mathbf{p}_{j-1}) + \\ & p_j U_G(k-1, j-1, \mathbf{p}_{j-1}), 1 \leq k \leq \\ & j \leq n, \\ U_G(k, j, \mathbf{p}_j) = 0, & k = 0, j \geq 1, \\ U_G(k, j, \mathbf{p}_j) = 1, & k = j+1, j \geq 1, \end{aligned}$	$\begin{aligned} R_G(k, j, \mathbf{p}_j) = & q_j R_G(k, j-1, \mathbf{p}_{j-1}) + \\ & p_j R_G(k-1, j-1, \mathbf{p}_{j-1}), 1 \leq k \leq \\ & j \leq n, \\ R_G(k, j, \mathbf{p}_j) = 1, & k = 0, j \geq 1, \\ R_G(k, j, \mathbf{p}_j) = 0, & k = j+1, j \geq 1, \end{aligned}$
k-out-of-n:F system	$\begin{aligned} R_F(k, j, \mathbf{p}_j) = & p_j R_F(k, j-1, \mathbf{p}_{j-1}) + \\ & q_j R_F(k-1, j-1, \mathbf{p}_{j-1}), 1 \leq k \leq j \\ & \leq n, \\ R_F(k, j, \mathbf{p}_j) = 0, & k = 0, j \geq 1, \\ R_F(k, j, \mathbf{p}_j) = 1, & k = j+1, j \geq 1, \end{aligned}$	$\begin{aligned} U_F(k, j, \mathbf{p}_j) = & p_j U_F(k, j-1, \mathbf{p}_{j-1}) + \\ & q_j U_F(k-1, j-1, \mathbf{p}_{j-1}), 1 \leq k \leq \\ & j \leq n, \\ U_F(k, j, \mathbf{p}_j) = 1, & k = 0, j \geq 1, \\ U_F(k, j, \mathbf{p}_j) = 0, & k = j+1, j \geq 1, \end{aligned}$

corresponding entries of the right column. Power of working in the Boolean domain is manifested in the *complementation* and *dualization* operations that transform $S_G(k, j, \mathbf{X}_j)$ in (17) to $\bar{S}_G(k, j, \mathbf{X}_j)$ and $S_F(k, j, \mathbf{X}_j)$, respectively. First, we note that both complementation and dualization change a boundary condition of 1 to 0 and a boundary condition of 0 to 1. We next consider the recursive domain $1 \leq k \leq j \leq n$. Inverting $S_G(k, j, \mathbf{X}_j)$ to produce $\bar{S}_G(k, j, \mathbf{X}_j)$ is straightforward since $S_G(k, j, \mathbf{X}_j)$ When one has a *partially-factored* expression of system success or failure such as the following one [56] basis $\{\bar{X}_j, X_j\}$ with coefficients $S_G(k, j-1, \mathbf{X}_{j-1})$ and $S_G(k-1, j-1, \mathbf{X}_{j-1})$, respectively, and hence $\bar{S}_G(k, j, \mathbf{X}_j)$ is expressed in terms of the *same* orthonormal basis, but with inverted coefficients $\bar{S}_G(k, j-1, \mathbf{X}_{j-1})$ and

$\bar{S}_G(k-1, j-1, \bar{X}_{j-1})$. Since $S_F(k, j, X_j)$ is the *dual* function of $S_G(k, j, X_j)$, it is obtained from it by inverting both its output S_G and inputs (arguments) X_j , namely $S_F(k, j, X_j) = \bar{S}_G(k, j, \bar{X}_j)$, and hence it is given by the complement of (17a) with inputs inverted, i.e., it is given by the complement of

$$X_j S_G(k, j-1, \bar{X}_{j-1}) \vee \bar{X}_j S_G(k-1, j-1, \bar{X}_{j-1}).$$

This complement is given by

$$\begin{aligned} X_j \bar{S}_G(k, j-1, \bar{X}_{j-1}) \vee \bar{X}_j \bar{S}_G(k-1, j-1, \bar{X}_{j-1}) \\ = X_j S_F(k, j-1, X_{j-1}) \vee \bar{X}_j S_F(k-1, j-1, X_{j-1}). \end{aligned}$$

Rushdi [52] noted that the AR algorithm is, in fact, an implementation of the Reduced Ordered Binary Decision Diagram (ROBDD) strategy when this strategy is adapted for computing the k-out-n-reliability. The ROBDD strategy was proposed by Bryant [110] as an extension of the BDD methodology of Akers [111]. The ROBDD deals with *general* switching (two-valued Boolean) functions, and is now considered the state-of-the-art data structure for handling such functions, with extensive applications in reliability [112–122]. As stated earlier, the AR algorithm has a domain of applicability that is narrower than that of the ROBDD algorithm, as it is restricted to switching functions that are both *monotonically nondecreasing* and *totally symmetric*. Apart from this, the AR algorithm has exactly the same features as the ROBDD algorithm, namely:

1. Both the AR and ROBDD algorithms are based on the Boole–Shannon expansion in the Boolean domain.
2. Both algorithms visit the variables in a certain order, typically monotonically ascending or monotonically descending.
3. Both algorithms reduce the resulting expansion tree (which is exponential in size) to a rooted acyclic graph that is both canonical and hopefully compact or sub-exponential. The reduction rules [122] require 3(a) merging isomorphic subtrees, and 3(b) deletion of useless nodes whose outgoing edges point to the same child node.

Figure 1 translates the lower two rows of Table 5 into a quad of Mason Signal Flow Graph (SFGs) for computing the reliability and unreliability of the k_1 -out-of- $(k_1 + k_2)$:G system and the k_1 -out-of- $(k_1 + k_2)$:F system for $k_1 \leq 0$ and $k_2 \leq -1$. Figure 1 is drawn over a rectangular grid of coordinates $k_1 = k$, $k_2 = n - k_1$. The graphs in Fig. 1 are the essence of the four versions of the iterative AR algorithm. Striking similarities (and trivial, albeit subtle differences) of these graphs with ROBDDs are discussed in Rushdi and Alturki [61].

Though the recursive relation (17a) was discovered initially in the Boolean domain [27], it can be derived, as an afterthought, in the probability domain. For this purpose, we employ the Total Probability Theorem with the two mutually

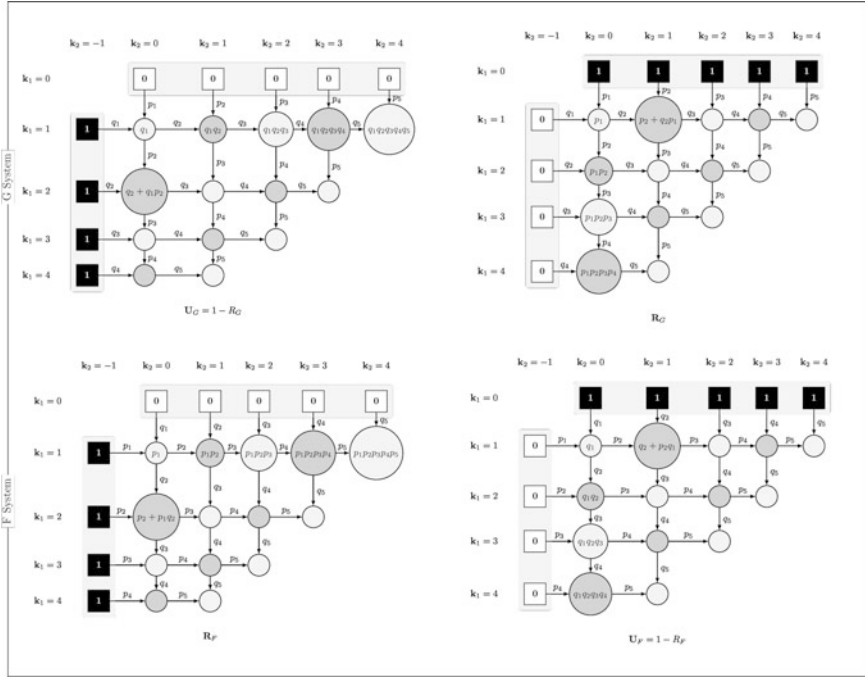


Fig. 1 Translation of the lower two rows of Table 5 into a quad of Mason signal flow graph (SFGs)

exclusive and exhaustive events {component j is failed} and {component j is good}, for $1 \leq k \leq j \leq n$, as follows

$$\begin{aligned}
 R_G(k, j, \mathbf{p}_j) &= Pr\{\text{at least } k \text{ components out of } j \text{ are good}\} \\
 &= Pr\{\text{at least } k \text{ components out of } j \text{ are good} | \text{component } j \text{ is failed}\} \\
 &\quad Pr\{\text{component } j \text{ is failed}\} \\
 &\quad + Pr\{\text{at least } k \text{ components out of } j \text{ are good} | \text{component } j \text{ is good}\} \\
 &\quad Pr\{\text{component } j \text{ is good}\} \\
 &= Pr\{\text{at least } k \text{ components out of } (j - 1) \text{ are good}\} q_j \\
 &\quad + Pr\{\text{at least } (k - 1) \text{ components out of } (j - 1) \text{ are good}\} p_j \\
 &= q_j R_G(k, j - 1, \mathbf{p}_{j-1}) + p_j R_G(k - 1, j - 1, \mathbf{p}_{j-1}).
 \end{aligned}$$

The derivation above should not be viewed as a defeat of purpose of the paradigm presented herein. Simple as it may be, this derivation was arrived at only after (18a) became known, as a transform of the Boolean result (17a).

6 Conclusions

This chapter gives a detailed overview of the current practice of solving system reliability problems by first formulating and manipulating them in the Boolean domain, and subsequently going to the probability domain. This practice is a supplementary alternative of the more dominant practice of working solely in the probability domain all throughout the solution process. The equivalence between event-based probability (which uses probabilities of events) and Boolean-based probability (which employs expectations of indicator variables of such events) is demonstrated. In particular, the expectation of a Boolean quotient is identified as a conditional probability, and the Boole–Shannon expansion is recognized as a Boolean-domain counterpart of the Total Probability Theorem.

Formulation of a system reliability problem in the Boolean domain is conceptually very simple for systems described by verbal statements. It is the natural strategy to use for systems described by fault trees, which are composed of logical AND–OR gates, and can similarly be utilized with systems described by block diagrams. It is the method of choice with systems described by network graphs, when these systems are handled through the enumeration of minimal pathsets and minimal cutsets, rather than via other graph techniques. *Manipulations* in the Boolean domain are facilitated for small problems by the existence of insightful manual tools such as the Karnaugh map or the variable-entered Karnaugh map, and for large problems due to the availability of many scalable efficient algorithms such as those utilizing the ROBDD. *Transfer* from the Boolean domain to the probability domain is achieved via the Real or Probability Transform of a switching (two-valued Boolean) function, which is generally an intractable process of exponential complexity. However, computational burden can be shifted to the Boolean domain, with a potential decrease in complexity, if the pertinent Boolean function is first cast into the form of a *PRE*. Some of the important rules used in generating a *PRE* are presented, occasionally along with succinct proofs. These include rules to *achieve* disjointness (orthogonality) of ORed formulas, and to *preserve* statistical independence as much as possible among ANDED formulas.

The chapter reports a special case of Boolean-domain success by discussing recursive relations governing the reliability and unreliability of the k-out-of-n:G and the k-out-of-n:F systems. The complementation (inversion) and dualization operations in the Boolean domain are utilized to interrelate these four entities, and to evaluate them via four versions of the same algorithm. These four versions of the algorithm are explained in terms of signal flow graphs that are compact, regular, and acyclic, in addition to being isomorphic to the ROBDD, which is reputed to be the most efficient technique for manipulating Boolean functions.

From a pedagogical point of view, the paradigm and concepts discussed herein might be too involved to be of utility in elementary probability education. However, a somewhat bold suggestion is to employ the present paradigm and concepts in areas constituting advanced applications of probability (beside that of system reliability) such as telecommunications, genetics, and finance.

Appendix A: Boolean Quotient

Let us define a literal to be a letter or its complement, where a letter is a constant or a variable. A Boolean term or product is a conjunction or ANDing of m literals in which no letter appears more than once. For $m = 1$, a term is a single literal and for $m = 0$, a term is the constant 1. Note that, according to this definition the constant 0 is not a term. Given a Boolean function f and a term t , the Boolean quotient of f with respect to t , denoted by f/t , is defined to be the function formed from f by imposing the constraint $\{t = 1\}$ explicitly [103], i.e.,

$$f/t = [f]_{t=1}, \quad (19)$$

The Boolean quotient is also known as a ratio, a subfunction, or a restriction. Brown [103] lists and proves several useful properties of Boolean quotients, of which we reproduce the following ones:

$$f/1 = f, \quad (20)$$

$$f/st = (f/s)/t = (f/t)/s, \quad \text{for } st \neq 0, \quad (21)$$

{for n -variable functions f and g and an m -variable term t with $m \leq n$ },

$$f \leq g \implies f/t \leq g/t \quad (22)$$

$$t \wedge f = t \wedge (f/t) \quad (23)$$

$$\bar{t} \vee f = \bar{t} \vee (f/t) \quad (24)$$

$$t \wedge f \leq f/t \leq \bar{t} \vee f \quad (25)$$

In this Appendix, we followed Brown [103] in denoting a Boolean quotient by an inclined slash f/t . However, it is possible to denote it by a vertical bar $f|t$ to stress the equivalent meaning (borrowed from conditional probability) of f conditioned by t or f given t .

References

1. **Ryabinin, I. A.**, Logical probabilistic analysis and its history. *International Journal of Risk Assessment and Management*, **18**(3-4): 256-265, (2015).
2. **Merekin, Yu. V.**, Solution of problems of probabilistic calculation of single-cycle schemes by an orthogonalization method, *Computing Systems*, Issue 4, (1963) (In Russian).
3. **Premo, A. F.**, The use of Boolean algebra and a truth table in the formulation of a mathematical model of success, *IEEE Transactions on Reliability*, **R-12**(3): 45-49, (1963).
4. **Fratta, L.**, and **Montanari, U.**, A Boolean algebra method for computing the terminal reliability in a communication network, *IEEE Transactions on Circuit Theory*, **20**(3): 203-211, (1973).

5. **Hurley, R. B.**, Probability maps, *IEEE Transactions on Reliability*, **R-12**(3): 39-44, (1963).
6. **Krauss, P. H.**, Representation of conditional probability measures on Boolean algebras, *Acta Mathematica Hungarica*, **19**(3-4), 229-241, (1968).
7. **Parker, K. P.**, and **E. J. McCluskey**, Probabilistic treatment of general combinational networks, *IEEE Transactions on Computers*, **24**(6): 668-670, (1975)
8. **Ogus, R. C.**, The probability of a correct output from a combinational circuit, *IEEE Transactions on Computers*, **24**(5): 534-544, (1975).
9. **Bennetts, R. G.**, On the analysis of fault trees, *IEEE Transactions on Reliability*, **R-24** (3): 175-185, (1975).
10. **Aggarwal, K. K.**, Comments on "On the analysis of fault trees," *IEEE Transactions on Reliability*, **R-25**(2): 126-127, (1976).
11. **Schneeweiss, W. G.**, Calculating the probability of Boolean expression being 1, *IEEE Transactions on Reliability*, **R-26**(1): 16-22, (1977).
12. **Locks, M. O.**, Inverting and minimizing path sets and cut sets, *IEEE Transactions on Reliability*, **R-27**(2): 107-109, (1978).
13. **Abraham, J. A.**, An improved algorithm for network reliability, *IEEE Transactions on Reliability*, **R-28** (1): 58-61, (1979).
14. **Dotson, W.**, and **Gobien, J.**, A new analysis technique for probabilistic graphs, *IEEE Transactions on Circuits and Systems*, **CAS-26**(10): 855-865, (1979).
15. **Bennetts, R. G.**, Analysis of reliability block diagrams by Boolean techniques, *IEEE Transactions on Reliability*, **R-31**(2): 159-166, (1982).
16. **Locks, M. O.**, Recursive disjoint products: a review of three algorithms, *IEEE Transactions on Reliability*, **R-31**(1): 33-35, (1982).
17. **Rushdi, A. M.**, Symbolic reliability analysis with the aid of variable-entered Karnaugh maps, *IEEE Transactions on Reliability*, **R-32**(2): 134-139, (1983).
18. **Rushdi, A. M.**, and **Al-Khateeb, D. L.**, A review of methods for system reliability analysis: A Karnaugh-map perspective, *Proceedings of the First Saudi Engineering Conference, Jeddah, Saudi Arabia*, vol. 1, pp. 57-95, (1983).
19. **Gupta, P. P.**, and **Agarwal, S. C.**, A Boolean algebra method for reliability calculations, *Microelectronics and Reliability*, **23**(5): 863-865, (1983).
20. **Rushdi, A. M.**, Overall reliability analysis for computer-communication networks, *Proceedings of the Seventh National Computer Conference*, Riyadh, Saudi Arabia, pp. 23-38, (1984).
21. **Schneeweiss, W. G.**, Disjoint Boolean products via Shannon's expansion, *IEEE Transactions on Reliability*, **R-34** (4): 329-332, (1984).
22. **Rushdi, A. M.**, On reliability evaluation by network decomposition, *IEEE Transactions on Reliability*, **R-33**(5): 379-384, (1984), Corrections: *ibid*, **R-34**(4): 319 (1985).
23. **Shier, D. R.**, and **Whited, D. E.**, Algorithms for generating minimal cutsets by inversion, *IEEE Transactions on Reliability*, **R-34**(4): 314-319, (1985).
24. **Locks, M. O.**, Recent developments in computing of system reliability, *IEEE Transactions on Reliability*, **R-34**(5): 425-436, (1985).
25. **Rushdi, A. M.**, and **Goda, A. S.**, Symbolic reliability analysis via Shannon's expansion and statistical independence, *Microelectronics and Reliability*, **25**(6): 1041-1053, (1985).
26. **Rushdi, A. M.**, Map derivation of the minimal sum of a switching function from that of its complement, *Microelectronics and Reliability*, **25**: 1055-1065, (1985).
27. **Rushdi, A. M.**, Utilization of symmetric switching functions in the computation of k-out-of-n system reliability, *Microelectronics and Reliability*, **26**(5): 973-987, (1986).
28. **Rushdi, A. M.**, Efficient computation of k-to-l-out-of-n system reliability, *Reliability Engineering*, **17**(3): 157-163, (1987), Erratum: *ibid*, **19**(4): 321, (1987).
29. **Rushdi, A. M.**, and **Dehlawi, F.**, Optimal computation of k-to-l-out-of-n system reliability, *Microelectronics and Reliability*, **27**(5): 875-896, (1987), Erratum : *ibid*, **28**(4): 671, (1988).
30. **Locks, M. O.**, A minimizing algorithm for sum of disjoint products. *IEEE Transactions on Reliability*, **R-36**(4): 445-453, (1987).

31. **Rushdi, A. M.**, On computing the syndrome of a switching function, *Microelectronics and Reliability*, **27**(4): 703-716, (1987).
32. **Rushdi, A. M.**, A switching-algebraic analysis of consecutive-k-out-of-n: F systems, *Microelectronics and Reliability*, **27**(1):171-174, (1987).
33. **Rushdi, A. M.**, A switching-algebraic analysis of circular consecutive-k-out-of-n: F systems, *Reliability Engineering & System Safety*, **21**(2): 119-127, (1988).
34. **Ball M.**, and **Provan J.**, Disjoint products and efficient computation of reliability, *Operations Research*, **36**(5): 703-715, (1988).
35. **Heidtmann, K. D.**, Smaller sums of disjoint products by subproduct inversion, *IEEE Transactions on Reliability*, **38**(3): 305-311, (1989).
36. **Rushdi, A. M.**, Threshold systems and their reliability, *Microelectronics and Reliability*, **30**(2): 299-312, (1990).
37. **Rushdi, A. M.**, Comment on: An efficient non-recursive algorithm for computing the reliability of k-out-of-n systems, *IEEE Transactions on Reliability*, **40**(1): 60-61, (1991).
38. **Veeraraghavan, M.**, and **Trivedi, K. S.**, An improved algorithm for symbolic reliability analysis, *IEEE Transactions on Reliability*, **40**(3): 347-358, (1991).
39. **Rushdi, A. M.**, *Reliability of k-out-of-n Systems*, Chapter 5 in Misra, K. B. (Editor), *New Trends in System Reliability Evaluation*, Vol. **16**, Fundamental Studies in Engineering, Elsevier Science Publishers, Amsterdam, The Netherlands, 185-227, (1993).
40. **Soh, S.**, and **Rai, S.**, Experimental results on preprocessing of path/cut terms in sum of disjoint products technique, *IEEE Transactions on Reliability*, **42**(1): 24-33, (1993).
41. **Liu, H. H.**, **Yang, W. T.**, and **Liu, C. C.**, An improved minimizing algorithm for the summation of disjoint products by Shannon's expansion, *Microelectronics and Reliability*, **33**(4), 599-613, (1993).
42. **Rushdi, A. M.**, and **AbdulGhani A. A.**, A comparison between reliability analyses based primarily on disjointness or statistical independence, *Microelectronics and Reliability*, **33**: 965-978, (1993).
43. **Rai, S.**, **Veeraraghavan, M.**, and **Trivedi, K. S.**, A survey of efficient reliability computation using disjoint products approach, *Networks*, **25**(3): 147-163, (1995).
44. **Schneeweiss, W. G.**, Advanced hand calculations for fault tree analysis and synthesis, *Microelectronics and Reliability*, **37**(3): 403-415, (1997).
45. **Tsuchiya, T.**, **Kajikawa, T.**, and **Kikuno, T.**, Parallelizing SDP (Sum of disjoint products) algorithms for fast reliability analysis, *IEICE Transactions on Information and Systems*, **83**(5): 1183-1186, (2000).
46. **Boros, E.**, **Crama, Y.**, **Ekin, O.**, **Hammer, P. L.**, **Ibaraki, T.**, and **Kogan, A.**, Boolean normal forms, shell ability, and reliability computations, *SIAM Journal on Discrete Mathematics*, **13**(2): 212-226, (2000).
47. **Balan, A. O.**, and **Traldi, L.**, Preprocessing minpaths for sum of disjoint products, *IEEE Transactions on Reliability*, **52**(3): 289-295, (2003).
48. **Miltersen, P. B.**, **Radhakrishnan, J.**, and **Wegener, I.**, On converting CNF to DNF. *Theoretical Computer Science*, **347**(1): 325-335, (2005).
49. **Rushdi, A. M.**, and **Ba-Rukab O. M.**, A doubly-stochastic fault-tree assessment of the probabilities of security breaches in computer systems, *Proceedings of the Second Saudi Science Conference, Part Four: Computer, Mathematics, and Statistics, Jeddah, Saudi Arabia*, pp. 1-17, (2005).
50. **Rushdi, A. M.**, and **Ba-Rukab O. M.**, Fault-tree modelling of computer system security, *International Journal of Computer Mathematics*, **82**(7): 805-819, (2005).
51. **Traldi, L.**, Non-minimal sums of disjoint products, *Reliability Engineering & System Safety*, **91**(5): 533-538, (2006).
52. **Rushdi, A. M.**, Partially-redundant systems: Examples, reliability, and life expectancy, *International Magazine on Advances in Computer Science and Telecommunications*, **1**(1): 1-13, (2010).

53. **Higashiyama Y.**, and **Rumchev V.**, New version of SDP method for weighted-k-out-of-n system, *Proceedings of the 15th World Multi-Conference on Systemics, Cybernetics, and Informatics 2012*; **1**: 120-125, (2012).
54. **Rushdi, A. M. A.**, and **Hassan A. K.**, Reliability of migration between habitat patches with heterogeneous ecological corridors, *Ecological Modelling*, **304**: 1-10, (2015).
55. **Rushdi, A. M. A.** and **Alturki, A. M.**, Reliability of coherent threshold systems, *Journal of Applied Sciences*, **15**(3): 431-443, (2015).
56. **Rushdi, A. M. A.**, and **Hassan, A. K.**, An exposition of system reliability analysis with an ecological perspective, *Ecological Indicators*, **63**, 282-295, (2016).
57. **Rushdi, A. M. A.**, and **Al-Qwasm, M. A.**, Exposition and comparison of two kinds of a posteriori analysis of fault trees, *Journal of King Abdulaziz University: Computing and Information Technology*, **5**(1): 55-74, (2016).
58. **Bamasak, S. M.**, and **Rushdi, A. M. A.**, Uncertainty analysis of fault-tree models for power system protection. *Journal of Qassim University: Engineering and Computer Sciences*, **8**(1): 65-80, (2016).
59. **Rushdi, A. M. A.**, and **Hassan, A. K.**, Quantification of uncertainty in the reliability of migration between habitat patches. *Computational Ecology and Software*, **6**(3): 66-82, (2016).
60. **Alturki, A. M.**, and **Rushdi A. M. A.**, Weighted voting systems: A threshold-Boolean perspective, *Journal of Engineering Research*, **4**(1): 125-143, (2016).
61. **Rushdi A. M. A.**, and **Alturki, A. M.**, Unification of mathematical concepts and algorithms of k-out-of-n system reliability: A perspective of improved disjoint products, *Journal of Engineering Research*, **5**(1), (2017).
62. **Rushdi, A. M. A.**, and **Al-Qwasm, M. A.**, Utilization of basic-event statistical independence in simplifying the uncertainty analysis of fault trees, *Journal of King Abdulaziz University: Computing and Information Technology*, **6**(2), (2017).
63. **Feller, W.**, *An Introduction to Probability Theory and Its Applications, Vol. 1*, 3rd Edition, Wiley, (1968).
64. **Mosteller, F.**, *Fifty Challenging Problems in Probability with Solutions*, Revised Edition, Dover Publications, (1987).
65. **Whittle, P.**, *Probability via Expectation*, Fourth Edition, Springer, New York, (2000).
66. **Chung, K. L.**, *A Course in Probability Theory*, Third Edition, Academic Press, (2001).
67. **Trivedi, K. S.**, *Probability & Statistics with Reliability, Queuing, and Computer Science Applications*, 2nd Edition, Prentice-Hall, Englewood Cliffs, NJ, USA, (2002).
68. **Jaynes, E. T.**, *Probability Theory: The Logic of Science*, Cambridge University Press, (2003).
69. **Bertsekas, D. P.**, and **Tsitsiklis, J. N.**, *Introduction to Probability*, 2nd Edition, Athena Scientific, (2008).
70. **Ross, S.**, *A First Course in Probability*, 8th Edition, Pearson Prentice Hall, (2009).
71. **Ross, S. M.**, *Introduction to Probability Models*, 10th Edition, Academic Press, (2014).
72. **Morin, D. J.**, *Probability: For the Enthusiastic Beginner*, Create Space Independent Publishing Platform, (2016).
73. **Falkowski, B. J.**, **Lozano, C. C.**, and **Łuba, T.** Reliability analysis based on arithmetic and Boolean representations, *Proceedings of IET 15th International Conference on Mixed Design of Integrated Circuits and Systems (MIXDES 2008)*, pp. 505-509, (2008).
74. **Falkowski, B. J.**, **Lozano, C. C.**, and **Łuba, T.**, Arithmetic and Boolean techniques for derivation of system reliability expressions, *Electronics and Telecommunications Quarterly*, **54**(2): 167-177, (2008).
75. **Papaioannou, S. G.**, and **Barrett, W. A.**, The real transform of a Boolean function and its applications, *Computers & Electrical Engineering*, **2**(2): 215-224, (1975).
76. **Kumar, S. K.**, and **Breuer, M. A.**, Probabilistic aspects of Boolean switching functions via a new transform, *Journal of the ACM (JACM)*, **28**(3): 502-520, (1981).
77. **Rushdi, A. M.**, How to hand-check a symbolic reliability expression, *IEEE Transactions on Reliability*, **R-32**(5): 402-408, (1983).

78. **Rushdi, A. M.**, Uncertainty analysis of fault-tree outputs, *IEEE Transactions on Reliability*, **R-34**(5): 458-462, (1985).
79. **Rushdi, A. M.**, On computing the spectral coefficients of a switching function, *Microelectronics and Reliability*, **27**(6): 965-979, (1987).
80. **Heidtmann, K. D.**, Arithmetic spectrum applied to fault detection for combinational networks, *IEEE Transactions on Computers*, **40**(3): 320-324, (1991).
81. **Jain, J.**, Arithmetic transform of Boolean functions, in *Representations of Discrete Functions*, in **Sasao, T., and M. Fujita** (Editors), Kluwer Academic Publishers, Dordrecht, the Netherlands, pp. 133-161, (1996).
82. **Rushdi, A. M. A. and Ghaleb, F. A. M.**, The Walsh spectrum and the real transform of a switching function: A review with a Karnaugh-map perspective, *Journal of Qassim University: Engineering and Computer Sciences*, **7**(2): 73-112, (2015).
83. **Kossov, A., and Preuß, W.** Zuverlässigkeitsanalyse konsekutiver Systeme—eine Übersicht (Reliability analysis of consecutive systems—an overview), *Operations-Research-Spektrum*, **11**(3): 121-130, (1989).
84. **Preuss, W. W., and Boehme, T. K.**, On reliability analysis of consecutive-k out-of-n: F systems and their generalizations—A survey. In *Approximation, Probability, and Related Fields*, pp. 401-411, Springer US, (1994).
85. **Malinowski, J., and Preuss, W.** On the reliability of generalized consecutive systems—A survey, *International Journal of Reliability, Quality and Safety Engineering*, **2**(02): 187-201, (1995).
86. **Eryilmaz, S.**, Capacity loss and residual capacity in weighted k-out-of-n: G systems, *Reliability Engineering & System Safety*, **136**: 140-144, (2015).
87. **Li, X., You, Y., and Fang, R.**, On weighted k-out-of-n systems with statistically dependent component lifetimes, *Probability in the Engineering and Informational Sciences*, 1-14, published online (2016).
88. **Chang G. J., Cui, L. and Hwang, F. K.**, Reliabilities for (n, f, k) systems, *Statistics & Probability Letters*, **43**: 237-242, (1999).
89. **Guo, Y., Cui, L., Li, J., and Gao, S.**, Reliabilities for $(n, f, k(i, j))$ and $\langle n, f, k(i, j) \rangle$ systems, *Communications in Statistics-Theory and Methods*, **35**(10): 1779-1789, (2006).
90. **Eryilmaz, S., and Aksoy, T.**, Reliability of Linear (n, f, k) Systems with weighted components, *Journal of Systems Science and Systems Engineering*, **19**(3): 277-284, (2010).
91. **Kamalja, K. K., and Shinde, R. L.**, On the reliability of (n, f, k) and $\langle n, f, k \rangle$ systems, *Communications in Statistics-Theory and Methods*, **43**(8): 1649-1665, (2014).
92. **Rushdi, A. M., and Ba-Rukab, O. M.** The modern syllogistic method as a tool for engineering problem solving. *Journal of Engineering and Computer Sciences: Qassim University*, **1**(1): 57-70, (2008).
93. **Rushdi, A. M.**, A conditional probability treatment of strict consecutive-k-out-of-n: F systems, *Microelectronics and Reliability*, **29**(4): 581-586, (1989).
94. **Rushdi, A. M.**, Effect of statistical dependencies in strict consecutive-k-out-of-n: F systems. *Microelectronics Reliability*, **28**(2): 309-318, (1988).
95. **Rushdi, A. M., S. G. Papastavridis., and R. A. Evans**, Some open questions (and replies) on: Strict consecutive-k-out-of-n:F systems, *IEEE Transactions on Reliability*, **39**(3): 380-381, (1990).
96. **Hwang, F. K.**, Comment on strict consecutive-k-out-of-n:F systems, *IEEE Transactions on Reliability*, **40**(3): 264, 270, (1991).
97. **Schneeweiss, W. G.**, *Boolean Functions with Engineering Applications and Computer Programs*, Springer-Verlag, New York, NY, USA, (1989).
98. **Barlow, R. E., and F. Prochan**, *Mathematical Theory of Reliability*, Wiley, New York, NY, USA, (1996).
99. **Aven, T., and U. Jensen**, *Stochastic Models in Reliability*, Springer-Verlag, New York, NY, USA, Vol. 41, (1999).
100. **Hammer, P. L. and S. Rudeanu**, *Boolean Methods in Operations Research and Related Areas*, Springer Verlag, Berlin, Germany, (1968).

101. **Rudeanu, S.**, *Boolean Functions and Equations*, North-Holland Publishing Company & American Elsevier, Amsterdam, the Netherlands (1974).
102. **Muroga, S.**, *Logic Design and Switching Theory*, Wiley, New York, NY, USA, (1979).
103. **Brown, F. M.**, *Boolean Reasoning: The Logic of Boolean Equations*, Kluwer Academic Publishers, Boston, MA, USA (1990).
104. **Brown, F. M.**, *Boolean Reasoning: The Logic of Boolean Equations, 2nd Ed.*, Dover Publications, Mineola, NY, USA, (2003).
105. **Crama, Y.**, and **Hammer, P. L.**, *Boolean functions: Theory, Algorithms, and Applications*, Cambridge, United Kingdom, Cambridge University Press, (2011).
106. **Pecht, M. G.**, **Jieyu, S. H. E.**, and **Barbe, D. F.**, Evaluating terminal pair system reliability, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, **76**(4): 555-564, (1993).
107. **Butler, A. C.**, **Rao, S. S.**, and **LeClair, S. R.**, Reliability analysis of complex systems using symbolic logic, *Reliability Engineering & System Safety*, **40**(1): 49-60, (1993).
108. **Nahman, J. M.**, Exact enumeration of minimal paths of partitioned networks, *Microelectronics and Reliability*, **34**(7): 1167-1176, (1994).
109. **Shen, Y.**, Computerization of the R-ABC algorithm, *Microelectronics and Reliability*, **36**(9): 1219-1221, (1996).
110. **Bryant, R. E.**, Graph-based algorithms for Boolean function manipulation, *IEEE Transactions on Computers*, **100**(8): 677-691, (1986).
111. **Akers S.**, Binary decision diagrams, *IEEE Transaction on Computers*, **C-27**(6): 509-516, (1960).
112. **Singh, H.**, **Vaithilingam, S.**, **Anne, R. K.**, and **Anneberg, L.**, Terminal reliability using binary decision diagrams, *Microelectronics and Reliability*, **36**(3): 363-365, (1996).
113. **Rauzy, A.**, A brief introduction to binary decision diagrams, *Journal Européen des Systèmes Automatisés*, **30**(8): 1033-1050, (1996).
114. **Jinglun, Z.**, and **Quan, S.**, Reliability analysis based on binary decision diagrams. *Journal of Quality in Maintenance Engineering*, **4**(2): 150-161, (1998).
115. **Zang, X.**, **Sun, N.**, and **Trivedi, K. S.**, A BDD-based algorithm for reliability analysis of phased-mission systems, *IEEE Transactions on Reliability*, **48**(1): 50-60, (1999).
116. **Andrews, J. D.**, and **Dunnett, S. J.**, Event-tree analysis using binary decision diagrams, *IEEE Transactions on Reliability*, **49**(2): 230-238, (2000).
117. **Reay, K. A.**, and **Andrews, J. D.**, A fault tree analysis strategy using binary decision diagrams, *Reliability Engineering & System Safety*, **78**(1): 45-56, (2002).
118. **Dutuit, Y.**, and **Rauzy, A.**, Approximate estimation of system reliability via fault trees, *Reliability Engineering & System Safety*, **87**(2): 163-172, (2005).
119. **Xing, L.**, An efficient binary-decision-diagram-based approach for network reliability and sensitivity analysis, *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, **38**(1): 105-115, (2008).
120. **Rauzy A.**, Binary Decision Diagrams for Reliability Studies, Chapter 25 in **Misra, K. B.** (Editor), *Handbook of Performability Engineering*, Springer, London, UK, pp. 381-396, (2008).
121. **Mo, Y.**, A multiple-valued decision-based approach to solve dynamic fault tree, *IEEE Transactions on Reliability*, **63**(1): 81-93, (2014).
122. **Mo, Y.**, **Xing, L.**, **Amari, S. V.**, and **Dugan, J. B.**, Efficient analysis of multi-state k-out-of-n system, *Reliability Engineering & System Safety*, **133**: 95-105, (2015).