JKAU: Comp. Sci., **Vol. 5**, pp: 55 - 74 (2016 A.D./ 1437 A.H.)
DOI: 10.4197 / Comp. 5 - 4

Exposition and Comparison of Two Kinds of *a Posteriori* Analysis of Fault Trees

Ali Muhammad Ali Rushdi*, Muhammad Ahmad Al-Qwasmi

Department of Electrical and Computer Engineering,
King Abdulaziz University, Jeddah, Saudi Arabia

*arushdi@kau.edu.sa

Abstract. Fault trees are top-down formal deductive analytic tools with diverse applications in many fields such as reliability, safety and security. Forward fault tree analysis (FTA) can be termed *a priori* analysis since it predicts the top-event probability in terms of basic-event probabilities. This paper offers a tutorial exposition and a detailed comparison of two kinds of backward or *a posteriori* FTA that are implemented in the probability domain and in the Boolean domain, respectively. For the probability-domain *a posteriori* FTA, it is assumed that the top event probability is known. For example, when the top event is presumed to have occurred, then it has a probability of one. The analysis proceeds recursively in the probability domain to assess the probabilities of lower events under certain realistic assumptions such as mutual exclusiveness or statistical independence of the input events for a specific gate, and with the utilization of educated guesses on certain ratios of probabilities of such events. This paper offers a detailed mathematical procedure for implementing this *a posteriori* FTA that makes the most of the concept of duality. The procedure is demonstrated via a detailed illustrative example. The paper also considers the *a posteriori* FTA in the Boolean domain. Such an analysis is available in the literature in terms of the very powerful tool of *Bayesian Networks* (BNs). We demonstrate here that in many cases this analysis is still possible via elementary fault-tree manipulations that use the concept of a *Boolean quotient* to effectively implement Bayes' Theorem in the Boolean domain. Again, a demonstrative example is given to illustrate the Boolean *a posteriori* FTA, explain its details, and show that the power of BNs is not really warranted in simple cases. A detailed comparison between the two kinds of *a posteriori* FTA is also given to identify their similarities and differences.

Keywords: Fault tree; a priori analysis; *a posteriori* analysis; probability domain; Boolean domain.

1. Introduction

Fault trees are top-down formal deductive analytic tools that have applications in many fields such as reliability, safety, and security, albeit sometimes they are used under a variety of unwarranted disguised names such as success trees, elicitation trees, attack trees, defense trees, *etc*. Conventional fault tree analysis (FTA) might be termed a forward, *a priori*, or predictive analysis since it obtains the top-event probability in terms of basic-event probabilities, or more generally it decides the probability of any higher-level event in the tree in terms of the probabilities of its lower-level events[1-20]. The reverse type of FTA might be termed a backward, *a posteriori*, or diagnostic analysis. There are (at least) two kinds of this analysis, which are to be reviewed, analyzed, demonstrated, compared and interrelated in this paper.

The first kind of *a posteriori* FTA is due to Shooman[21, 22]. In this kind of analysis, it is assumed that the top event has a known probability, *e.g.*, the top event could be considered to have actually occurred and hence possess a probability of one. The analysis proceeds recursively in the probability domain to assess the probabilities of lower events under certain realistic assumptions such as mutual exclusiveness or statistical independence of the input events for a specific gate, and utilization of educated guesses on certain ratios of probabilities of such events. By contrast, the second kind of *a posteriori* analysis, as reported by Bobbio, *et al.*,[23] and Langseth and Portinale[24] is in essence a classical Bayesian analysis involving an equivalent of the Total Probability Theorem, and Bayes' Theorem[25]. A detailed comparison between the two kinds of *a posteriori* analysis is given in Table 1. Various aspects of the comparison in Table 1 will become clarified further as we proceed throughout this paper.

While Shooman[22] restricted the a posteriori FTA to that of OR gates with Mutually Exclusive (ME) inputs, we extend the analysis to include both AND gates with Statistically Independent (SI) inputs and OR gates with either ME or SI inputs. We derive general solutions for all types of gates and conditions with arbitrary numbers of inputs. We also outline the solution of the general case based on the use of the Inclusion-Exclusion Principle with simplifying assumptions other than the ME or SI assumptions.

The second kind of *a posteriori* analysis is typically conducted by mapping fault trees into the more powerful tool of Bayesian networks (BNs), which are known also (occasionally with minor differences) as belief nets, causal networks, probabilistic- dependence graphs, or influence diagrams. Bayesian networks have better capabilities than standard fault trees, such as their capabilities to handle uncertainty, statistical dependence or multi-state behavior [23, 24, 26-39]. However, the use of BNs in *a posteriori* FTA might not be warranted in many important problems that can still be handled via (the somewhat modest) capabilities of fault trees.

The organization of the remainder of this paper is as follows. Section 2 lists our notation, abbreviations and certain useful nomenclature. Section 3 presents the *a posteriori* analysis of fault trees in the probability domain. The main thesis of this section is that such an analysis necessitates only the *a posteriori* analysis of single gates. Therefore, section 3 discusses the general *a posteriori* analysis of single AND or OR gates, and then derives (under a variety of appropriate assumptions) *a posteriori* solution for an AND gate with SI inputs, an OR gate with ME inputs, and an OR gate with SI inputs. The results obtained are applied to a detailed fault-tree example. Section 4 treats the *a posteriori* analysis of fault trees in the Boolean domain. We demonstrate here that in many cases this analysis is possible via elementary fault-tree manipulations that use the concept of a Boolean quotient (known also as a Boolean ratio, subfunction or restriction)[40-52] to implement Bayes' Theorem effectively in the Boolean domain. Again, a demonstrative example is given to illustrate the Boolean a posteriori FTA and explain its details, and show that the power of BNs is not really warranted in simple cases. A detailed comparison between the two kinds of *a posteriori* FTA is also given with the hope of setting the stage on how they can be further interrelated and even combined. Section 5 concludes the paper and points out new directions for further research.

Table 1. Comparison of the two kinds of *a posteriori* analysis of fault trees.

	First kind	**Second kind**
Basic assumption	Expert guessing of certain ratios among probabilities of inputs of various gates.	Knowledge of basic-event *a priori* probabilities
Nature of relation considered	*Local* gate relations between the probabilities of the output and input of *single* gates	An *overall* tree relation between the top-event probability and basic-event probabilities
Forward analysis incorporated ?	No	Yes
Mathematics needed	Solution of algebraic equations (essentially quadratic equations)	Classical Bayesian analysis involving an equivalent of the Total Probability Theorem, and Bayes' Theorem
Implementation via Bayesian Networks	No	Possible (Necessary only to avoid limitations of FTAs)
Utility as an aid to *a priori* analysis	Yes	No
Utility as an aid to guessing input probability ratios	No	Yes
Typical applications	Forensic analysis of terrorist attacks	Diagnosis of safety-critical systems
Seminal work	Shooman[22]	Bobbio, *et al*.,[23]; Langseth and Portinale[24]

2. Notation, Abbreviations and Nomenclature

A. Notation

$P(A)$ = Probability of the event A.

r_i = Ratio of $P(A_i)$ to $P(A_n)$ for $i = 1,2,\dots,n,\ r_n = 1$.

$E\{\dots\}$ = Expectation or expected value of a random variable $\{\dots\}$.

e_i = A probabilistic event; input i of an AND or an OR gate.

a_n = A probabilistic event; output of an AND gate of n inputs.

o_n = A probabilistic event; output of an OR gate of n inputs.

R_{n-1} = $Ratio\ of\ P\left(\bigcup_{i=1}^{n-1} A_i\right)\ to\ P(A_n).$

e_X = A fault-tree event labelled by indicator variable X.

t_i = Ratio of $P(\bar{A}_i)$ to $P(\bar{A}_n)$ for $i = 1, 2, \dots,\ n,\ t_n = 1$.

T = Particular name for the indicator variable of the top event e_T of the fault tree.

X = Generic name for the indicator variable of a certain FT event e_X. This is a random Boolean (switching) variable such that:
$X = 1$ $(\bar{X} = 0)$ if the event e_X occurs, and
$X = 0$ $(\bar{X} = 1)$ if the event e_X does not occur.

$x = E\{X\}$ = Expectation of the indicator variable X given by

$$x = E\{X\} = (1)P(X) + (0)P(\bar{X}) = P(X),$$

i.e., it is equal to the probability of occurrence of event e_X.

B. Abbreviations

FTA	Fault-Tree Analysis,
ME	Mutually Exclusive(ness),
SI	Statistically Independent/Statistical Independence,
BN	Bayesian Network.

C. Nomenclature

Forward (*a priori* or a predictive) fault tree analysis:

A fault-tree analysis in which the basic-event probabilities are known. The analysis chains forward to obtain higher-level event probabilities and terminates with a prediction of the top-event probability. This is the conventional fault-tree analysis, and it is what is meant when simply fault-tree analysis is mentioned.

Backward (*a posteriori* or diagnostic) fault tree analysis:

A fault-tree analysis in which the top-event probability is known. This analysis is mainly used when the top event is assumed to have occurred and hence has a probability of one.

***A posteriori* FTA of the first kind**:

A fault-tree analysis that chains backward to obtain lower-level event probabilities (under certain realistic assumptions), and terminates with a knowledge of all basic-event probabilities. The analysis relies on the solution of algebraic equations expressing probabilities of the inputs of a certain gate in terms of the probability of its output. Such a solution proceeds recursively from the top gate (whose output has a known probability, typically one) to lower-level gates terminating at the leaf gates. Typically, the analysis relies on the expert guessing of certain ratios among probabilities of various gates.

***A posteriori* FTA of the second kind**:

A fault-tree analysis that starts with a priori knowledge of basic-event probabilities, utilizes this knowledge in forward analysis to compute the top-event probability, and then (under the assumption that the top event has occurred) uses Bayes' theorem to deduce the *a posteriori* basic-event probabilities.

Bayesian Network (BN):

A directed acyclic graph in which discrete random variables are assigned to each node, together with the conditional dependence on the parent nodes. Root nodes are nodes with no parents, and marginal prior probabilities are assigned to them. The main feature of a BN is that it is possible to include local conditional dependencies into the model, by directly specifying the causes that influence a given effect. Bayesian Networks[23, 24] are usually defined on discrete random variables, though some extensions have been proposed for extending the formalism to some form of continuous random variables. BN are more suitable to represent complex dependencies among components and to include uncertainty and multi-state behavior in modeling[23, 24].

Mapping BNs into FTs:

It is quite straightforward to map a given FT into an equivalent BN with binary nodes, where the FT's gates (with input and output events) are mapped into small BN fragments, whose combination produces the whole BN corresponding to the given FT. In other words, the modular construction of an FT can be mapped into a modular construction of an equivalent BN. The modeling flexibility of the BN formalism can accommodate various kinds of statistical dependencies Uncertainties, and multi-state behavior that are difficult to include in the FT formalism[23, 24].

Reliability-Ready Expression (RRE): An expression in the switching (Boolean) domain, in which logically multiplied (ANDed) entities are statistically independent and logically added (ORed) entities are disjoint. Such an expression can be directly transformed, on a one-to-one basis, to the algebraic or probability domain by replacing switching (Boolean) indicators by their statistical expectations, and also replacing logical multiplication and addition (ANDing and ORing) by their arithmetic counterparts Rules for the conversion of a general switching (Boolean) expression into a PRE are provided in[8, 9, 52-56].

Duality:
The dual of a switching function is obtained by complementing the function and all its switching arguments (inverting both output and inputs)[56-58].

3. The *a Posteriori* Analysis in the Probability domain

Since the *a posteriori* analysis of a fault tree can be accomplished in terms of that of single gates, this section is devoted to the *a posteriori* analysis of single AND or OR gates, first generally, and then subject to the Mutual Exclusiveness (ME) or Statistical Independence (SI) assumptions. The analysis technique is then demonstrated via a detailed numerical example.

3.1. General Analysis of AND and OR gates

The aim of this subsection is to discuss the general analysis of AND and OR gates, stress the utility of the concept of *duality* in such analysis, and point out the considerable reduction in complexity when the inputs are either Mutually Exclusive (ME) or Statistically Independent (SI).

The output a_n of an AND gate of n inputs $e_1, e_2, \dots, e_n$ has a probability given in terms of conditional probabilities as[25]

$$\mathrm{P}(\mathrm{a_n}) = \mathrm{P}\left(\bigcap_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{e_i}\right) = \mathrm{P}(\mathrm{e_1})\mathrm{P}(\mathrm{e_2}|\mathrm{e_1})\mathrm{P}(\mathrm{e_3}|\mathrm{e_1}\mathrm{e_2}) \dots \mathrm{P}(\mathrm{e_n}|\mathrm{e_1}\mathrm{e_2} \dots \mathrm{e_{n-1}}), \tag{1}$$

while the output o_n of an OR gate of n inputs $e_1, e_2, \dots, e_n$ has a probability given by the Inclusion-Exclusion Principle[25, 59, 60]:

$$P(o_n) = P\left(\bigcup_{i=1}^{n} e_i\right) = \sum_{i=1}^{n} P(e_i) - \sum\sum_{1\le i<j\le n} P(e_i \cap e_j) + \sum\sum\sum_{1\le i<j<k\le n} P(e_i \cap e_j \cap e_k) - \dots + (-1)^{n-1} P\left(\bigcap_{i=1}^{n} e_i\right). \tag{2}$$

Note that (2) expresses the output of an OR gate in terms of the outputs of many binary or multi-input AND gates, which need to be expressed via (1) or extensions thereof. The AND and OR gates are dual gates. Complementation of both inputs and output of one gate produces the other gate. This is the essence of the two De Morgan's laws, visually represented by Fig. 1, and mathematically given by

$$\left\{\mathrm{o_n} = \bigcup_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{e_i}\right\} \Leftrightarrow \left\{\bar{\mathrm{o}}_\mathrm{n} = \bigcap_{\mathrm{i}=1}^{\mathrm{n}} \bar{\mathrm{e}}_\mathrm{i}\right\}, \tag{3}$$

$$\left\{a_n = \bigcap_{i=1}^{n} e_i\right\} \Leftrightarrow \left\{\bar{a}_n = \bigcup_{i=1}^{n} \bar{e}_i\right\}. \qquad (4)$$

According to (3) and (4), the analysis of an AND (OR) gate can be converted to the dual analysis of an OR (AND) gate. Therefore, the analyst has a choice to analyze any given gate directly as is or indirectly in terms of its dual gate.

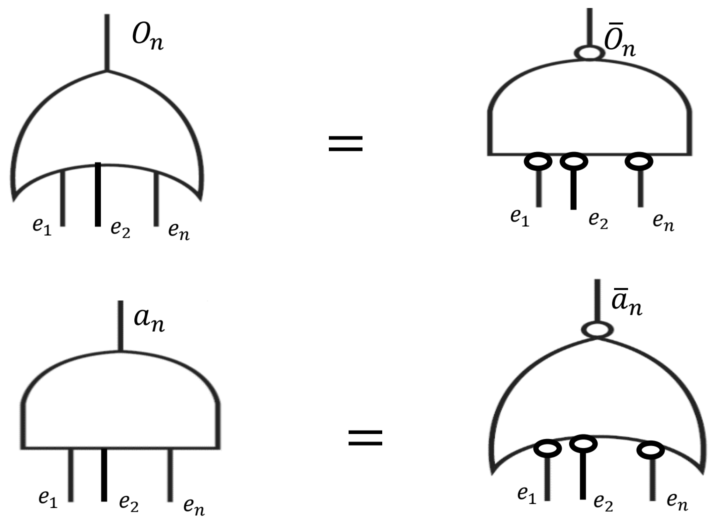

Fig. 1. Visual Interpretation of De Morgan's Laws.

The analysis of an AND gate via (1) requires the use of conditional probabilities, while the analysis of an OR gate via (2) involves an exponential number $(2^n - 1)$ of terms, many of which necessitate the use of conditional probabilities in expressions similar to (1). There is a considerable reduction in the complexity of the analysis when the events e_i are either Mutually-Exclusive (ME) or Statistically Independent (SI).

If the events e_i are ME, *i.e.*, if

$$e_i \cap e_j = \emptyset \ \forall \text{ i and j}, \qquad (5)$$

then (1) and (2) reduce respectively to

$$P(a_n) = 0, \text{ ME } e_i, \qquad (6)$$

$$P(o_n) = P\left(\bigcup_{i=1}^{n} e_i\right) = \sum_{i=1}^{n} P(e_i), ME\ e_i. \qquad (7)$$

If, instead the events e_i are SI, *i.e.*, if

$$P(e_i|e_j) = P(e_i), \quad \forall\, i \text{ and j}, \tag{8}$$

or equivalently, if

$$P(e_i \cap e_j) = P(e_i)\, P(e_j), \tag{9}$$

then (1) and (2) reduce respectively to

$$\mathrm{P}(\mathrm{a_n}) = \prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{P}(\mathrm{e_i}), \;\; SI\ e_i. \tag{10}$$

$$P(o_n) = \sum_{i=1}^{n} P(e_i) - \sum\sum_{1 \le i < j \le n} P(e_i)P(e_j) + \sum\sum\sum_{1 \le i < j < k \le n} P(e_i)P(e_j)P(e_k) - \cdots$$
$$+ (-1)^{n-1} \prod_{i=1}^{n} P(e_i) = 1 - \prod_{i=1}^{n} (1 - P(e_i)), SI\ e_i. \tag{11}$$

Note that (11) can also be obtained from (3) and (10) in the equivalent complementary form

$$P(\bar{o}_n) = \prod_{i=1}^{n} P(\bar{e}_i), SI\ e_i. \tag{12}$$

3.2. Analysis of an AND gate with SI inputs

We assume that the probability of the output a_n of the AND gate is known, say S_n. This probability is exactly 1 if the event a_n is known to have occurred. Otherwise, it would be available through expert estimation or through *a posteriori* analysis of higher-level gates. Since the inputs of the AND gate are SI, equation (10) is applicable and reduces to

$$\prod_{i=1}^{n} P(e_i) = S_n, \qquad SI\ e_i. \tag{13}$$

Following Shooman[9], we assume that we can express each of the probabilities in (13) as a ratio r_i of the last probability among them $P(e_n)$, namely

$$P(e_i) = r_i\, P(e_n), 1 \le i \le n, \tag{14}$$

where $r_n = 1$. Substituting (14) in (13), we solve (13) for each of the probabilities $P(e_i)$ as

$$P(e_i) = r_i \left[\left(\prod_{j=1}^{n} r_j \right)^{-1} S_n \right]^{1/n}, 1 \le i \le n, \quad SI\ e_i. \tag{15}$$

3.3. Analysis of an OR gate with ME Inputs

The case studied in this subsection is the only case studied by Shooman[22]. Here, equation (7) is applicable, and the probability of the output T_n of the OR gate is known, say S_n. Hence, equation (7) can be rewritten as:

$$\sum_{i=1}^{n} P(e_i) = S_n, ME \text{ e}_\text{i}. \qquad (16)$$

Now, making the assumption (14) and substituting (14) in (16), we can solve (16) for each of the probabilities $P(e_i)$ as:

$$P(e_i) = S_n\, r_i \left[\sum_{j=1}^{n} r_j\right]^{-1}.\ 1 \leq i \leq n,\ \ ME\ e_i. \qquad (17)$$

Note that (17) for the ME inputs of OR has some resemblance with (15) for the SI inputs of AND.

3.4. Analysis of an OR gate with SI inputs

The OR gate with SI inputs is analyzed in a direct fashion in subsection 3.4.1 and is analyzed via its dual representation in subsection 3.4.2.

3.4.1. Direct Analysis

The output of an OR gate with n inputs can be written as:

$$o_n = \bigcup_{i=1}^{n} e_i = o_{n-1} \cup e_n\ , \qquad (18)$$

where

$$o_{n-1} = \bigcup_{i=1}^{n-1} e_i\ . \qquad (19)$$

Since the event e_n is statistically independent of each of the events e_i $(1 \leq i \leq n-1)$, then it is also independent of their union o_{n-1}. The expression (18) allows the Inclusion-Exclusion Principle (2) to be rewritten as:

$$S_n = P(o_n) = P(o_{n-1}) + P(e_n) - P(o_{n-1})P(e_n). \quad (20)$$

Now, we assume that we can express $P(o_{n-1})$ as a ratio R_{n-1} of $P(e_n)$, *i.e.*

$$P(o_{n-1}) = R_{n-1}\, P(e_n), \qquad (21)$$

and hence obtain the following quadratic equation in $P(e_n)$

$$R_{n-1}[P(e_n)]^2 - (1 + R_{n-1})P(e_n) + S_n = 0. \quad (22)$$

Equation (22) has two solutions:

$$P(e_n) = \frac{1}{2R_{n-1}} \left[(1 + R_{n-1}) \mp \sqrt{D}\right], \qquad (23)$$

where the discriminant D is

$$\begin{aligned} D &= (1 + R_{n-1})^2 - 4R_{n-1}S_n \\ &= 1 + R^2{}_{n-1} + 2R_{n-1} - 4R_{n-1}S_n \\ &\geq (2R_{n-1}) + 2R_{n-1} - 4R_{n-1}S_n \\ &= 4R_{n-1}(1 - S_n) \geq 0. \qquad (24) \end{aligned}$$

In (24), we made use of the fact that S_n is a probability and hence must be less than or equal to 1. Equation (24) indicates that the discriminant D is non-negative, and hence both roots in (22) are *real*. Equation (24) also indicates that

$$\sqrt{D} \leq (1 + R_{n-1}), \qquad (25)$$

and hence both roots in (23) are *positive*. However, we now reject the positive sign in (23) since it corresponds to the solution

$$P(e_n) = \frac{1}{2R_{n-1}} \left[(1 + R_{n-1}) + \sqrt{D}\right], \qquad (26a)$$

$$P(o_{n-1}) = \frac{1}{2}\left[(1 + R_{n-1}) + \sqrt{D}\right], \tag{26b}$$

which corresponds to a probability $P(e_n) > 1$ if $R_{n-1} < 1$, and to a probability $P(o_{n-1}) > 1$ if $R_{n-1} > 1$. The only possibility of accepting the positive sign in (23) is the trivial case $R_{n-1} = 1$, $S_n = 1$ for which D is 0 and the two roots in (22) are equal. Hence, our final solution of (22) is

$$P(e_n) = \frac{1}{2R_{n-1}}\left[(1 + R_{n-1}) - ((1 + R_{n-1})^2 - 4R_{n-1}S_n)^{\frac{1}{2}}\right], \tag{27a}$$

$$P(o_{n-1}) = \frac{1}{2}\left[(1 + R_{n-1}) - ((1 + R_{n-1})^2 - 4R_{n-1}S_n)^{\frac{1}{2}}\right]. \tag{27b}$$

Now we use S_{n-1} to denote $P(o_{n-1})$ and continue our work recursively to obtain the probabilities $P(e_{n-1}), P(e_{n-2}), \ldots, P(e_1)$. Figure 2 summarizes the previous computations in flow-chart form.

3.4.2. Dual Analysis

An alternative analysis of an OR gate with SI inputs is possible via equation (12). Now, we assume that each of the probabilities of the complementary events in (12) is expressed as a ratio t_i of the last probability among them $P(\bar{e}_n)$, *i.e.*,

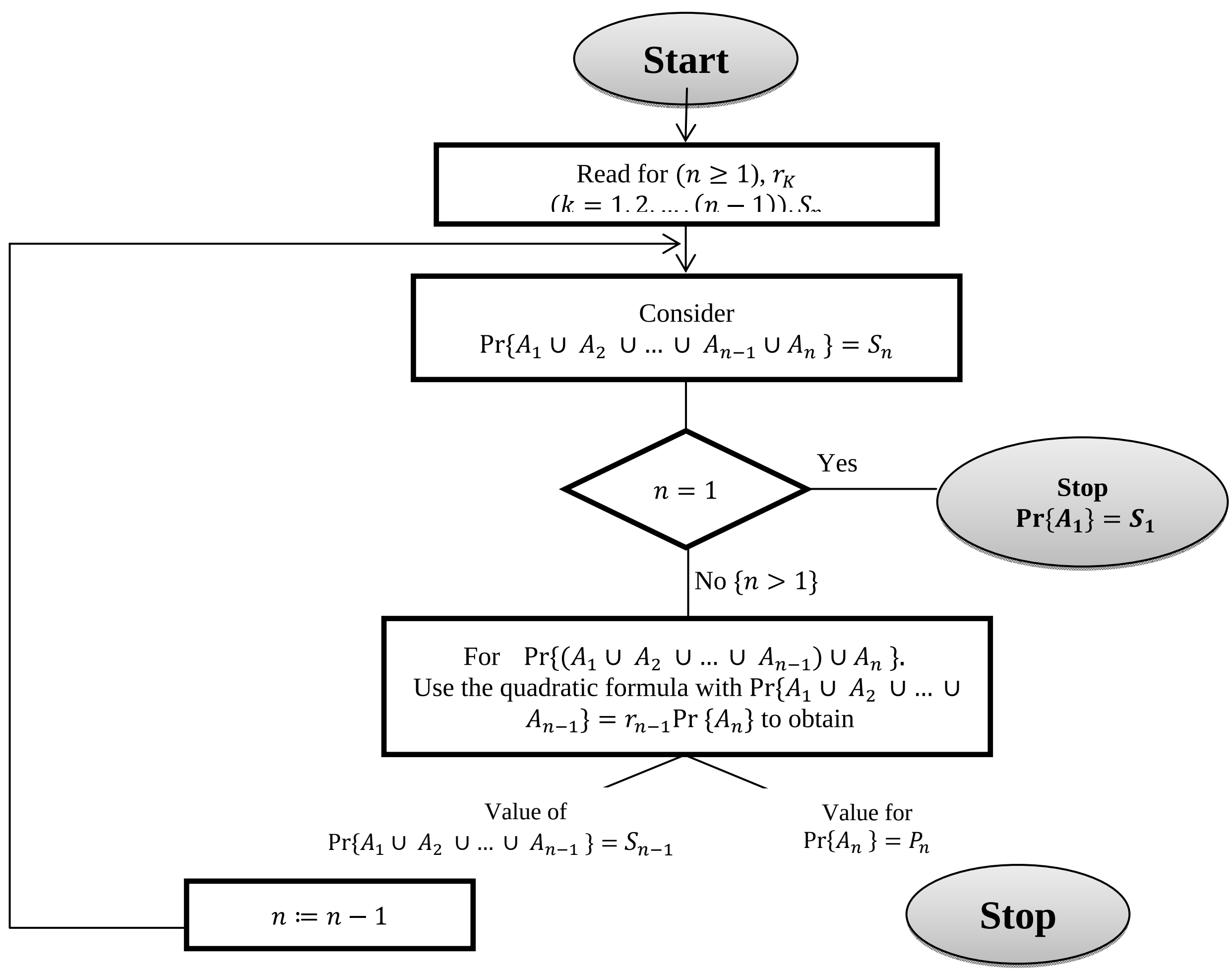

Fig. 2. Assigning probabilities for n statistically-independent inputs of an OR gate given the probability of its output.

$$P(\bar{e}_i) = t_i\, P(\bar{e}_n), \quad 1 \leq i \leq n, \tag{28}$$

where $t_n = 1$. Equating the RHS of (12) to S_n and substituting (28) into the resulting equation, we can solve (12) for each of the complementary probabilities $P(\bar{e}_i)$ as

$$P(\bar{e}_i) = t_i \left[\left(\prod_{j=1}^{n} t_j\right)^{-1} (1 - S_n)\right]^{1/n}, 1 \leq i \leq n, \quad \text{SI } \bar{e}_i. \tag{29}$$

In passing, we note that we used the assumption (28) to obtain a simple solution. Had we insisted on using the assumption (14), we would have obtained an nth – degree equation in each $P(e_i)$. The alternative (equally good) assumption in (28) saved us the trouble of solving an nth degree polynomial equation and the associated difficulty of selecting the appropriate root from a set of n roots.

Example 1:

Figure 3 displays a fault tree that combines all the special cases considered. It has an OR gate with three ME inputs, an AND gate with three SI inputs, and an OR gate with three SI inputs. Let us assume that the top event probability $P(o_3)$ is known to be $S_3 = 0.9$. We need to find all the basic-event probabilities. We start by estimating the probabilities of the events $e_1, e_2,$ and e_3 which are the ME inputs of the top OR gate. We now assume we know the following probability ratios.

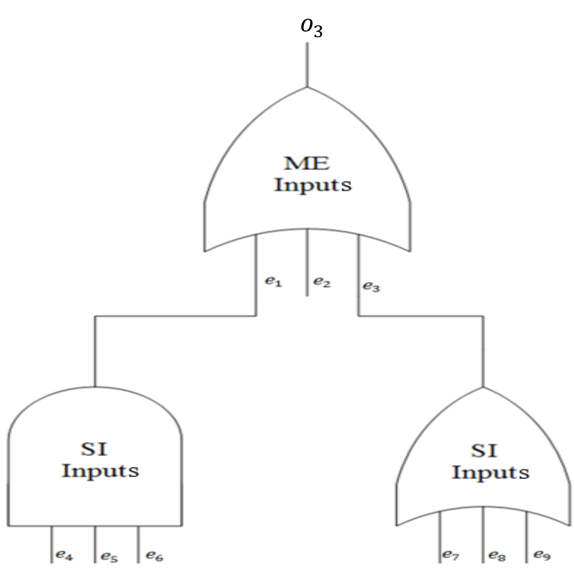

Fig. 3. A simple example of a fault tree that has an OR gate with MI inputs, an AND gate with SI input and an OR gate with SI inputs.

$$r_1 = P(e_1)/P(e_3) = 0.2, \tag{30a}$$
$$r_2 = P(e_2)/P(e_3) = 0.3, \tag{30b}$$
$$r_3 = P(e_3)/P(e_3) = 1.0. \tag{30c}$$

Hence, according to (17), we obtain

$$P(e_1) = \frac{S_3 r_1}{r_1 + r_2 + r_3} = \frac{(0.9)(0.2)}{0.2 + 0.3 + 1.0} = 0.12, \qquad (31a)$$

$$P(e_2) = \frac{S_3 r_2}{r_1 + r_2 + r_3} = 0.18, \qquad (31b)$$

$$P(e_3) = \frac{S_3 r_3}{r_1 + r_2 + r_3} = 0.60, \qquad (31c)$$

As expected $P(e_1), P(e_2)$ and $P(e_3)$ are in the ratio of $0.2{:}\,0.3{:}\,1.0$ and add up to $S_3 = 0.9$. We now know the probability of the output e_1 of the AND gate, and need to assess the probabilities of its inputs $e_4, e_5,$ and e_6. Again we assume we know the following probability ratios.

$$r_4 = P(e_4)/P(e_6) = 0.4, \qquad (32a)$$
$$r_5 = P(e_5)/P(e_6) = 0.5, \qquad (32b)$$
$$r_6 = P(e_6)/P(e_6) = 1.0. \qquad (33c)$$

Hence , according to (15), we obtain

$$P(e_4) = r_4[P(e_1)/(r_4 r_5 r_6)]^{\frac{1}{3}} = 0.33737, \qquad (33a)$$

$$P(e_5) = r_5[P(e_1)/(r_4 r_5 r_6)]^{\frac{1}{3}} = 0.42172, (33b)$$

$$P(e_6) = r_6[P(e_1)/(r_4 r_5 r_6)]^{\frac{1}{3}} = 0.84343. (33c)$$

As expected $P(e_4), P(e_5)$ and $P(e_6)$ are (to within roundoff-errors) in the ratio $0.4{:}\,0.5{:}\,1.0$ and their product is 0.12. Likewise, we use our knowlodge of the probability of the output e_3 of the OR gate with SI inputs e_7, e_8, e_9 to estimate the probabilities of these inputs. We use the dual analysis in Sec. 3.4.2, and starting with $P(\bar{e}_3) = 0.4$, we obtain $P(\bar{e}_7), P(\bar{e}_8)$ and $P(\bar{e}_9)$. We assume we know the probability ratios

$$t_7 = P(\bar{e}_7)/P(\bar{e}_9) = 0.6, \qquad (34a)$$
$$t_8 = P(\bar{e}_8)/P(\bar{e}_9) = 0.7, \qquad (34b)$$
$$t_9 = P(\bar{e}_9)/P(\bar{e}_9) = 1.0, \qquad (34c)$$

Hence, according to (29), we obtain

$$P(\bar{e}_7) = t_7[P(\bar{e}_3)/(t_7 t_8 t_9)]^{\frac{1}{3}} = 0.59032, \ (35a)$$
$$P(\bar{e}_8) = t_8[P(\bar{e}_3)/(t_7 t_8 t_9)]^{\frac{1}{3}} = 0.68871, (35b)$$
$$P(\bar{e}_9) = t_9\,[P(\bar{e}_3)/(t_7 t_8 t_9)]^{\frac{1}{3}} = 0.98387, (35c)$$

As expected, $P(\bar{e}_7), P(\bar{e}_8)$, and $P(\bar{e}_9)$ are (to within roundoff-error) in the ratio $0.6{:}\,0.7{:}\,1.0$ and their product is 0.4. The original probabilities are $P(e_7) = 0.40968$, $P(e_8) = 0.31129$, and $P(e_9) = 0.01613$.

4. The *a Posteriori* Analysis in the Boolean Domain

In this section, we demonstrate how to apply Bayes' theorem to achieve *a posteriori* FTA via manipulations in the Boolean domain. Let the top event be denoted by e_T and a basic event be denoted by e_X, then Bayes' Theorem[25] states that

$$P\{\, e_X | e_T\} = P\frac{\{e_T \cap e_X\}}{}P\{e_T\}, \qquad (36)$$

provided $P\{e_T\} \neq 0$. This theorem can be restated In terms of the indicator variables T and X of the events e_T and e_X when noting that the various probabilities in (36) can be rewritten as expectations, *i.e.*,

$$P\{e_x|e_T\} = E\{X|T\}, \quad (37a)$$

$$P(e_T \cap e_X\} = E\{T \wedge X\}, \quad (37b)$$

$$P\{e_T|e_X\} = E\{T|X\}, \quad (37c)$$

So that (36) can be rewritten as

$$P\{e_X|e_T\} \equiv E\{X|T\} = E\{T \wedge X\}/ E\{T\}. \quad (38)$$

Equation (38) is valid provided $E\{T\} \neq 0$. Now we can obtain the *a posteriori* probability $P\{e_X|e_T\}$ by pursuing the following steps:

1. Express T as a PRE (preferably the simplest possible) as a Boolean function of indicator variables (including X). Note that the job of forming a PRE is needed only once (at this initial step).
2. The indicator variable $(T|X)$ is the Boolean quotient of T with respect to X, *i.e.*

   $$T|X = T/X = T]_{X=1}, \quad (39)$$

   Further information on the Boolean quotient is given in Appendix A. Since T is in PRE form, each $(T|X)$, for any choice of X, is also in PRE form.

3. The indicator variable $(T \wedge X)$ is obtained via $(A4)$ as

   $$(T \wedge X) = X \wedge (T|X). \quad (40)$$

   Since $(T|X)$ is independent of X and is in PRE form, then $X \wedge (T|X)$ is also in PRE form.

4. The expectations $E\{T \wedge X\}$ and $E\{T\}$ in the RHS of (38) are now obtained immediately as one –to- one transformations of the PREs for $(T \wedge X)$ and (T).

The details of this method is now illustrated by applying it to a fault-tree example studied via Bayesian Networks by Bobbio, *et al.*,[23].

Example 2:

This example, originally taken from Malhotra and Trivedi[61], deals with the fault tree shown in Fig. 4. Bobbio, *et al.*,[23] solve this example by mapping the fault tree into a Bayesian network. We will demonstrate that such a mapping is not really warranted since fault-tree techniques suffice in this case. The fault tree represents a redundant multiprocessor system, with a single bus N connecting two processors P_1 and P_2 having access to a local memory bank each (M_1 and M_2), and through the bus to a shared memory bank M_3, so that if the local memory bank fails, the processor can use the shared one. Each processor is connected to a mirrored disk unit. If one of the disks fails, the processor switches on the mirror. The whole system is functional if the bus N is functional and one of the processing subsystems is functional. With a little abuse of notation, we are using the same upper-case un-complemented literal to denote a component, and also to denote the indicator variable for its failure. We can write the indicator T for the top event as a disjunction of cutset failures as in (3) of Bobbio, *et al.*,[23], but if we do so, we lose the ability to utilize statistical independence among basic events and end up with a complicated expression for the top-event probability. Instead, we write T as

$$T = N \vee (S_1 S_2), \quad (41)$$

where S_1 and S_2 are given by

$$S_1 = P_1 \vee D_{11}D_{12} \vee M_1M_3, \quad (42)$$

$$S_2 = P_2 \vee D_{21}D_{22} \vee M_2M_3, \quad (43)$$

We note that S_1 and S_2 would have been statistically independent had there been no common element M_3 between them. To circumvent this problem, we use a Boolean-Shannon expansion about M_3 to obtain

$$\begin{aligned} S_1S_2 &= \bar{M}_3\left(S_1S_2\middle|0_{M_3}\right) \vee M_3\left(S_1S_2\middle|1_{M_3}\right) \\ &= \bar{M}_3\,(P_1 \vee D_{11}D_{12})\,(P_2 \vee D_{21}D_{22}) \\ &\qquad \vee\ M_3(P_1 \vee M_1 \vee D_{11}D_{12})(\,P_2 \\ &\qquad \vee M_2 \vee D_{21}D_{22}), \qquad (44) \end{aligned}$$

Note that (44) contains two disjoint parts, thanks to the appearance of $\bar{M}_3$ in the first part and M_3 in the second part. The subfuctions of S_1S_2 in the two parts now consist each of factored statistically independent entities. We substitute (44) into (41), and use disjointing techniques[2, 8, 9, 53-59, 62-71] to convert the resulting expression into the Probability-Ready Expression

$$\begin{aligned} T = N\ \vee \bar{N}\ \Big(&\bar{M}_3(P_1 \vee \bar{P}_1\ D_{11}D_{12})(P_2 \\ &\vee \bar{P}_2\ D_{21}D_{22}) \\ &\vee M_3\ \big(P_1 \\ &\vee \bar{P}_1(M_1 \vee \bar{M}_1\ D_{11}D_{12})\big)\big(P_2 \\ &\vee \bar{P}_2(M_2 \\ &\vee \bar{M}_2\ D_{21}D_{22})\big)\Big). \qquad (45) \end{aligned}$$

The PRE (45) is converted, on a one-to-one basis, into the probability expression

$$"t = n + (1-n)((1-m\_3\,)(p\_1 + (1 - p\_1\,)\ d\_11\,d\_12\,)(p\_2 + (1 - p\_2\,)\,d\_21\,d\_22\,) + m\_3\,(p\_! + (1 - p\_1\,)(m\_1 + (1-m\_1)d\_11\,d\_12\,)\ (p\_2 + (1 - p\_2\,)(m\_2 + (1 - m\_2\,)\,d\_21\,d\_22\,)). \qquad (46)"$$

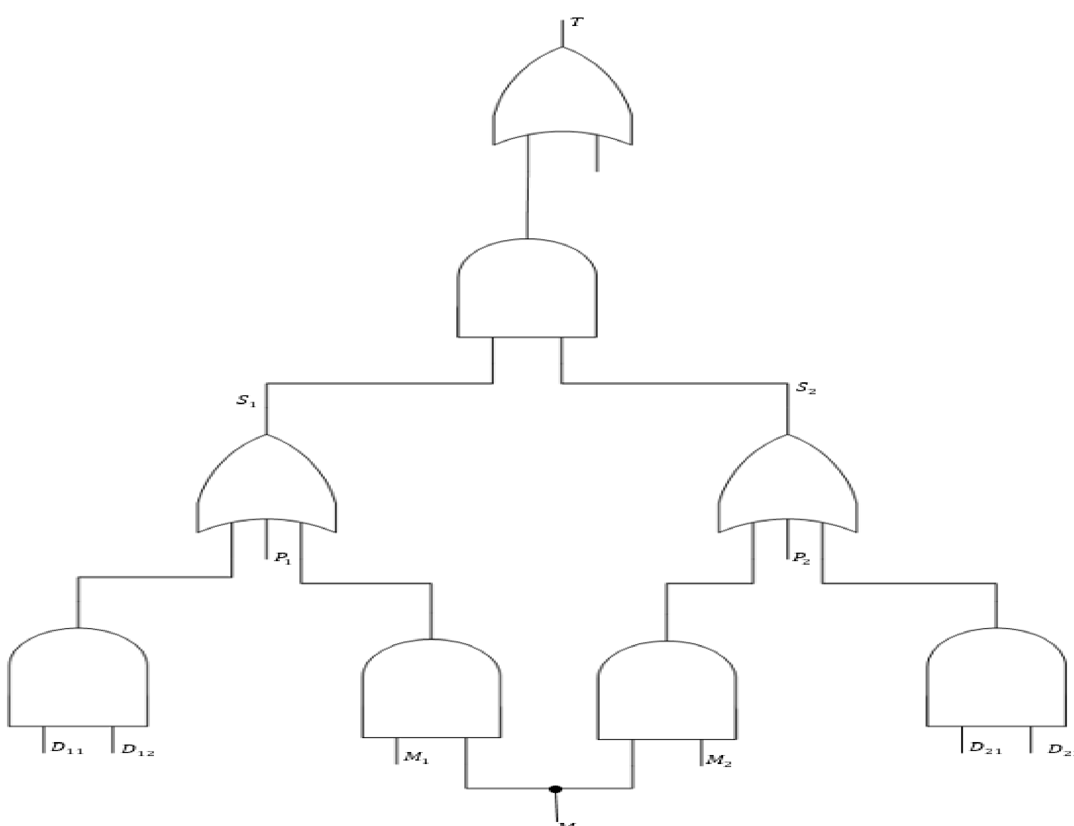

Fig. 4. Fault Tree for a multiprocessor system (taken from[23] and[61]).

Table 2. Values of the indicator of the top event conditioned by the indicator X of a basic event (equivalent to the Boolean quotient T/X).

X	$T\|X$
D_{11}	$N \vee \bar{N}\,(\bar{M}_3(P_1 \vee \bar{P}_1\,D_{12})(P_2 \vee \bar{P}_2\,D_{21}D_{22}) \vee M_3\big(P_1 \vee \bar{P}_1\,(M_1 \vee \bar{M}_1 D_{12})\big)\big(P_2 \vee \bar{P}_2\,(M_2 \vee \bar{M}_2\,D_{21}D_{22})\big))$
P_1	$N \vee \bar{N}\,(\bar{M}_3(P_2 \vee \bar{P}_2\,D_{21}D_{22}) \vee M_3\,\,(P_2 \vee \bar{P}_2\,(M_2 \vee \bar{M}_2\,D_{21}D_{22})))$
M_1	$N \vee \bar{N}\,(\bar{M}_3(P_1 \vee \bar{P}_1\,D_{11}D_{12})(P_2 \vee \bar{P}_2\,D_{21}D_{22}) \vee M_3(P_2 \vee \bar{P}_2\,(M_2 \vee \bar{M}_2\,D_{21}D_{22})\,))$
M_3	$N \vee \bar{N}\,(P_1 \vee \bar{P}_1(M_1 \vee \bar{M}_1\,D_{11}D_{12}) \vee \big(P_2 \vee \bar{P}_2\,(M_2 \vee \bar{M}_2\,D_{21}D_{22})\big))$
N	1

Table 3. The *a priori* and *a posteriori* probabilities of component failures in Example 2.

Component *X*	**The *a priori* failure probabilities**	**The *a posteriori* failure probabilities of Bobbio, *et al.*,[10]**	**The *a posteriori* failure probabilities of (38)**
D_{11}	$\boldsymbol{d = 0.32968}$	$\mathbf{0.98436}$	$\mathbf{0.9978947}$
P_1	$\boldsymbol{P = 0.00025}$	$\mathbf{0.02252}$	$\mathbf{0.0022937}$
M_1	$\boldsymbol{m = 0.000015}$	$\mathbf{0.000015}$	$\mathbf{0.0000150018}$
M_3	$\boldsymbol{m = 0.000015}$	$\mathbf{0.000015}$	$\mathbf{0.0000150034}$
N	$\boldsymbol{n = 0.00001}$	$\mathbf{0.000081}$	$\mathbf{0.0008425}$

Table 2 lists the conditional indicators or Boolean quotients (T/X), where X stands for D_{11}, P_1, M_1, M_3, and N. Table 3 shows the *a priori* failure probabilities assumed by Bobbio, *et al.*,[23], and the *a posteriori* failure probabilities computed by them via Bayesian-Network modelling. Table 3 also reports a posteriori probabilities computed via (38), under the assumption of equal reliabilities for similar components, *i.e.*, $d = d_{11} = d_{12} = d_{21} = d_{22}$, $P = P_1 = P_2$, and $m = m_1 = m_2 = m_3$. Thanks to (38), (40) and (46), one obtains

$$t = E\{T\} = n + (1-n)((1-m)(P + (1-P)d^2)^2 + m(P + (1-P)(m + (1-m)d^2)^2)), \quad (47)$$

$$E\{D|T\} = \left(\frac{d}{t}\right)\left(n + (1-n)\left((1-m)(P + (1-P)d)(P + (1-P)d^2) + m\left(P + (1-P)(m + (1-m)d)\big(P + (1-P)(m + (1-m)d^2)\big)\right)\right)\right), \quad (48)$$

$$E\{P|T\} = \left(\frac{P}{t}\right)\left(n + (1-n)\left((1-m)(P + (1-P)d^2) + m\big(P + (1-P)(m + (1-m)d^2)\big)\right)\right), \quad (49)$$

$$E\{M_1|T\} = \left(\frac{m}{t}\right)\left(n + (1-n)\left((1-m)(P + (1-P)d^2)^2 + m\big(P + (1-P)(m + (1-m)d^2)\big)\right)\right), \quad (50)$$

$$E\{M_3|T\} = \left(\frac{m}{t}\right)\left(n + (1-n)\left(P + (1-P)(m + (1-m)d^2)\right)^2\right), \tag{51}$$

$$E\{N|T\} = n/t. \tag{52}$$

In passing, we note that $E\{D_{11}|T\} = E\{D_{12}|T\} = E\{D_{21}|T\} = E\{D_{22}|T\} = E\{D|T\},\ E\{P_1|T\} = E\{P_2|T\} = E\{P|T\}$.
However, $E\{M_1|T\} = E\{M_2|T\} \neq E\{M_3|T\}$.

The results obtained in Table 3 are at best intriguing. We were expecting to obtain identical or at least approximately equal results in the second column of Table 3 (the *a posteriori* failure probabilities of Bobbio, *et al.*,[23]), and the third column of Table 3 (the *a posteriori* failure probabilities computed herein via (38)). However, while the values for D_{11}, M_1, and M_3 are somewhat reasonably similar, the values for each of P_1 and N differ by one order of magnitude. We argue that our computations are based on a simple fault-tree model that exactly fits our needs, and hence it is preferable according to Ockham's (Occam's) razor, which requires a model to retain the minimum of assumptions and details needed to capture all the essential features of what the model represents while excluding any extraneous or distracting features[72]. The details of our model are visible enough to allow an interested reader to check it by verifying the derivation of our equations and reproducing our numbers with a small calculator. In particular, our *a posteriori* failure probabilities can be easily seen to pass a simple check of satisfying the following conditional-probability equation derivable from (45).

$$\begin{aligned}1 = E\{T|T\} = E\{N|T\} + (1 - E\{N|T\})((1 - E\{M\_3|T\})(E\{P\_1|T\} + (1 \\ - E\{P\_1|T\})E\{D\_11|T\}\ E\{D\_12|T\})(E\{P\_2|T\} + (1 \\ - E\{P\_2|T\})E\{D\_21|T\}E\{D\_22|T\}) + E\{M\_3|T\}\ (E\{P\_2|T\} + (1 - E\{P\_2|T\})(E\{M\_1|T\} + (1 \\ - E\{M\_1|T\})E\{D\_11|T\}\ E\{D\_12|T\}))(E\{D\_11|T\}\ E\{D\_12|T\}(E\{M\_2|T\} + (1 \\ - E\{M\_2|T\})\ E\{D\_21|T\}\ E\{D\_22|T\}))).\end{aligned} \tag{53}$$

Conclusions

We presented and compared two kinds of *a posteriori* analysis of fault trees, namely an analysis in the probability domain, and another in the Boolean domain. The main thesis of the probability-domain FTA is that it necessitates only the *a posteriori* analysis of single gates. Therefore, we discussed the general *a posteriori* analysis of single AND or OR gates, and then derived (under a variety of appropriate assumptions) *a posteriori* solution for an AND gate with SI inputs, an OR gate with ME inputs, and an OR gate with SI inputs. The results obtained are applied to a detailed fault-tree example. In addition, we treated the *a posteriori* analysis of fault trees in the Boolean domain. We demonstrated that in many cases this analysis is possible via *elementary* fault-tree manipulations that use the concept of a *Boolean quotient* (known also as a Boolean ratio, subfunction or restriction) to effectively implement Bayes' Theorem in the Boolean domain. Again, a demonstrative example was given to illustrate the Boolean *a posteriori* FTA and explain its details, and show that the power of Bayesian networks (BNs) is not really warranted in many simple (albeit significant) cases. A detailed comparison between the two kinds of *a posteriori* FTA was also given to set the stage for explaining how these two kinds can be interrelated and even combined. The essential difference between the two kinds is that the first kind takes place in the probability domain and relies on educated guessing and solution of algebraic equations, while the second kind is a novel implementation of Bayes' Theorem in the Boolean domain, and acts occasionally as a suitable alternative to using the too-powerful technique of Bayesian networks. We stress herein that results obtained via the second kind of *a posteriori* FTA are much easier to verify and replicate than those obtained via Bayesian networks.

Further research is needed to utilize the two aforementioned kinds of *a posteriori* FTA in more practical situations, and to explore the possibility of existence of other kinds of *a posteriori* FTA. The comparison between the given two kinds of *a posteriori* FTA should be extended to further interrelate and even combine them. The implementation of Bayes' Theorem in the Boolean domain warrants further investigation, and opens new avenues for pedagogical and computational applications in probability theory and reliability engineering. An urgent issue to pursue is to solve many simple as well as complicated examples via both the second kind of *a posteriori* FTA and the Bayesian-network analysis to see if they do really agree or to identify reasons of disagreement between them and to locate where discrepancy between them emerges.

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[44], *i.e.*

$$f/t = [f]_{t=1}, \qquad (A1)$$

The Boolean quotient is also known as a ratio[40], a subfunction[41, 43, 45-50], or a restriction[42]. Brown[44] lists and proves several useful properties of Boolean quotients, of which we reproduce the following ones:

$$f/1 = f, \qquad (A2)$$

$$f/st = (f/s)/t = (f/t)/s\,, \ \ for\ st \neq 0, (A3)$$

$f \leq g \ \Rightarrow \ f/t \leq g/t$
{for n-variable functions f and g and an m-variable term t with $m \leq n$}, (A4)

$$t \wedge f = t \wedge (f/t) \qquad (A5)$$
$$\bar{t} \vee f = \bar{t} \vee (f/t) \qquad (A6)$$

$$t \wedge f \leq f/t \leq \bar{t} \vee f \qquad (A7)$$

In this Appendix, we followed Brown[44] in denoting a Boolean quotient by an inclined slash (f/t). However, in the main text we denote it by a vertical bar $(f|t)$ to stress the equivalent meaning of f conditioned by t or f given t.

References

[1] **Shooman, M. L.** The equivalence of reliability diagrams and fault-tree analysis. *IEEE Transactions on Reliability*, **19,** (2): 74-75, (1970).

[2] **Bennetts, R. G.**, On the analysis of fault trees, *IEEE Transactions on Reliability*, **R-24,** (3): 175-185, (1975).

[3] **Henley, E. J.** and **Kumamoto, H.**, *Reliability Engineering and Risk Assessment*, Englewood Cliffs, NJ: Prentice Hall, (1981).

[4] **Rushdi, A. M.**, Uncertainty analysis of fault-tree outputs, *IEEE Transactions on Reliability*, **R-34**, (5): 458-462, (1985).

[5] **Rushdi, A. M.** and **Kafrawy, K. F.**, Uncertainty propagation in fault-tree analysis using an exact method of moments, *Microelectronics and Reliability*, **28**: 945–965 (1988).

[6] **Kafrawy, K. F.** and **Rushdi, A. M.**, Uncertainty analysis of fault trees with statistically correlated failure data, *Microelectronics and Reliability*, **30**: 157–175, (1990).

[7] **Rausand, M.** and **Hoyland**, **A.**, *System Reliability Theory: Models, Statistical Methods, and Applications, 2nd Ed.*, Wiley, Hoboken, NJ, USA, (2004).

[8] **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, 1-17, (2005).

[9] **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).

[10] **Xing, L.** and **Amari, S. V.**, *Fault tree analysis*. In **K. B. Misra** (Editor), *Handbook of Performability Engineering*, Springer London, pp: 595-620, (2008).

[11] **Contini, S., Fabbri, L.** and **Matuzas, V. A.**, novel method to apply importance and sensitivity analysis to multiple fault-trees, *Journal of Loss Prevention in the Process Industries*, **23**(5): 574-584, (2010).

[12] **Contini, S.** and **Matuzas, V.**, Analysis of large fault trees based on functional decomposition, *Reliability Engineering & System Safety*, **96,** (3): 383-390, (2011).

[13] **Cha, S.** and **Yoo, J.**, A safety-focused verification using software fault trees, *Future Generation Computer Systems*, **28,** (8): 1272-1282, (2012).

[14] **Ruijters, E.** and **Stoelinga, M.,** Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools, *Computer Science Review*, **15**: 29-62, (2015).

[15] **Krcˇál, J.** and **Krcˇál, P.,** Scalable Analysis of Fault Trees with Dynamic Features, In *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp: 89-100, (2015).

[16] **Deng, Y., Wang, H.** and **Guo, B.,** BDD algorithms based on modularization for fault tree analysis, *Progress in Nuclear Energy*, **85**: 192-199, (2015).

[17] **Amstutz, J.,** Parallel evaluation of fault tree expressions, pp: 117-128 in **Jeffers, J.** and **Reinders, J.** (Editors), *High Performance Parallelism Pearls, Volume Two: Multicore and Many-core Programming Approaches*, Morgan Kaufmann, Burlington, CA, USA, (2015).

[18] **Makajic-Nikolic, D., Petrovic, N., Belic, A., Rokvic, M., Radakovic, J. A.** and **Tubic, V.,** The fault tree analysis of infectious medical waste management, *Journal of Cleaner Production*, **113**: 365-373, (2016).

[19] **Liu, P., Yang, L., Gao, Z., Li, S.** and **Gao, Y.,** Fault tree analysis combined with quantitative analysis for high-speed railway accidents, *Safety Science*, **79**: 344-357, (2015).

[20] **Hu, Y. N.,** Research on the application of fault tree analysis for building fire safety of hotels, *Procedia Engineering*, **135**: 523-529, (2016).

[21] **Shooman, M. L.,** Use of *a posteriori* fault trees for accident and terriorist investigation, *Proceedings of the 22nd International System Safety Conference*, Aug. 2-6, Providence RI, USA (2004).

[22] **Shooman, M. L.,** Terrorist risk evaluation using *a posteriori* fault trees, *IEEE 2006 Annual Reliability and Maintainability Symposium (RAMS'06)*, pp: 450-455, (2006).

[23] **Bobbio, A., Portinale, L., Minichino, M.** and **Ciancamerla, E.,** Improving the analysis of dependable systems by mapping fault trees into Bayesian networks, *Reliability Engineering and System Safety*, **71,** (3): 249-260, (2001).

[24] **Langseth, H.** and **Portinale**, **L.,** Bayesian networks in reliability, *Reliability Engineering and System Safety*, **92**: 92-108, (2007).

[25] **Trivedi, K. S.,** *Probability and Statistics with Reliability, Queuing, and Computer Science Applications, 2nd Ed.*, Prentice-Hall, Englewood Cliffs, NJ, USA, (2002).

[26] **Heckermann D., Wellman, M.** and **Mamdani, A.,** Real-world applications of Bayesian networks, *Communications of the ACM*, **38,** (3): 24-26, (1995).

[27] **Poole N.** and **Zhang**, **L.,** Exploiting causal independence in Bayesian network inference, *Journal of Artificial Intelligence Research*, **5**: 301–328, (1996).

[28] **Torres-Toledano J. G.** and **Sucar**, **L. E.,** Bayesian networks for reliability analysis of complex systems, In: *Proceedings of the 6th IberoAmerican conference on AI (IBERAMIA 98), Lecture notes in artificial intelligence*, Berlin, Germany: Springer, **1484**: 195-206, (1998).

[29] **Portinale, L.** and **Bobbio, A.,** Bayesian networks for dependability analysis: An application to digital control reliability, *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pp: 551–8, (1999).

[30] **Bobbio, A., Portinale, L., Minichino, M.** and **Ciancamerla, E.,** Comparing fault trees and Bayesian networks for dependability analysis, pp: 310-322 in *Computer Safety, Reliability and Security*, Springer Berlin Heidelberg, (1999).

[31] **Marsh, W.** and **Bearfield, G.,** Representing parameterised fault trees using Bayesian networks, pp: 120-133 in *Computer Safety, Reliability, and Security*, Springer Berlin Heidelberg, (2007).

[32] **Hosseini, S. H.** and **Takahashi, M.,** Combining static/dynamic fault trees and event trees using Bayesian networks, pp: 93-99 in *Computer Safety, Reliability, and Security*, Springer Berlin Heidelberg, (2007).

[33] **Marquez, D., Neil**, **M.** and **Fenton**, **N.,** Solving dynamic fault trees using a new hybrid Bayesian network inference algorithm. *IEEE 2008 16th Mediterranean Conference on Control and Automation*, pp: 609-614, (2008)

[34] **Mengshoel, O. J., Darwiche**, **A.** and **Uckun, S.,** Sensor validation using Bayesian networks. *In Proc. 9th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (iSAIRAS-08),* (2008).

[35] **Khakzad, N., Khan, F.** and **Amyotte, P.,** Safety analysis in process facilities: Comparison of fault tree and Bayesian network approaches, *Reliability Engineering & System Safety*, **96,** (8): 925-932, (2011).

[36] **Duan, R. X.** and **Zhou, H. L**., A New fault diagnosis method based on fault tree and Bayesian networks, *Energy Procedia*, **17**: 1376-1382, (2012).

[37] **Kabir, S., Walker, M.** and **Papadopoulos, Y**., Reliability Analysis of Dynamic Systems by Translating Temporal Fault Trees into Bayesian Networks, pp. 96-109 in *Model-Based Safety and Assessment,* pp: 96-109, Springer International Publishing, (2014).

[38] **Wang, Y.** and **Sun, Q.,** Bayesian network technology to analyze fault trees, pp: 87-94 in Proceedings of the First Symposium on Aviation Maintenance and Management-Volume II, Springer Berlin Heidelberg, (2014).

[39] **Gribaudo, M., Iacono, M.** and **Marrone, S.,** Exploiting Bayesian Networks for the Analysis of Combined Attack Trees, *Electronic Notes in Theoretical Computer Science*, **310**: 91-111, (2015).

[40] **Ghazala, M. J.,** Irredundant disjunctive and conjunctive forms of a Boolean function, *I.B.M. Journal of Research and Development*, **1**: 171-176, (1957).

[41] **Reusch, B.,** Generation of prime implicants from subfunctions and a unifying approach to the covering problem, *IEEE Transactions on Computers,* **C-24** (9): 924-930 (1975).

[42] **Bryant, R.,** Graph-based algorithms for Boolean function manipulation, *IEEE Transactions on Computers*, **C-35,** (8): 677-691, (1986).

[43] **Rushdi, A. M.,** Improved variable-entered Karnaugh map procedures, *Computers and Electrical Engineering,* **13,** (1): 41-52, (1987).

[44] **Brown, F. M.,** *Boolean Reasoning: The Logic of Boolean Equations*, Kluwer Academic Publishers, Boston, MA, USA (1990).

[45] **Rushdi, A. M.** and **Al-Yahya**, **H. A.,** A Boolean minimization procedure using the variable-entered Karnaugh map and the generalized consensus concept, *International Journal of Electronics*, **87,** (7): 769-794, (2000).

[46] **Rushdi, A. M.,** Prime-implicant extraction with the aid of the variable-entered Karnaugh map, *Umm Al-Qura University Journal : Science, Medicine and Engineering*, **13,** (1): 53-74 (2001).

[47] **Rushdi, A. M.** and **Al-Yahya**, **H. A.,** Further improved variable-entered Karnaugh map procedures for obtaining the irredundant forms of an incompletely-specified switching function, *Journal of King Abdulaziz University: Engineering Sciences*, **13**, (1): 111-152, (2001).

[48] **Rushdi, A. M.,** Using Variable-Entered Karnaugh Maps to Solve Boolean Equations, *International Journal of Computer Mathematics*, **78,** (1): 23-38 (2001).

[49] **Rushdi, A. M.** and **Al-Yahya**, **H. A.,** Derivation of the complete sum of a switching function with the aid of the variable-entered Karnaugh map, *Journal of King Saud University: Engineering Sciences*, **13,** (2): 239-269, (2001).

[50] **Rushdi, A. M.** and **Amashah**, **M. H.,** Using variable–entered Karnaugh maps to produce compact parametric solutions of Boolean equations, *International Journal of Computer Mathematics*, **88**, (15): 3136-3149 (2011).

[51] **Crama, Y.** and **Hammer**, **P. L.,** *Boolean Functions: Theory, Algorithms, and Applications,* Cambridge University Press, Cambridge, United Kingdom (2011).

[52] **Rushdi, A. M. A.** and **Alturki**, **A. M.,** Reliability of coherent threshold systems, *Journal of Applied Sciences*, **15,** (3): 431-443, (2015).

[53] **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).

[54] **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).

[55] **Rushdi, A. M. A.** and **Hassan**, **A. K.,** Reliability of migration between habitat patches with heterogeneous ecological corridors, *Ecological Modelling*, **304**: 1-10, (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.,** *Reliability of k-out-of-n Systems*, Chapter 5 in **K. B. Misra** (Editor), *New Trends in System Reliability Evaluation*, Vol. **16**, Fundamental Studies in Engineering, Elsevier Science Publishers, Amsterdam, The Netherlands, pp: 185-227, (1993).

[58] **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).

[59] **Dohmen, K.,** Inclusion-exclusion and network reliability, *Journal of Combinatorics*, **5**: 537-544, (1998).

[60] **Dohmen, K.,** Inclusion-Exclusion: Which terms cancel, *Archiv der Mathematik*, **74**(4): 314-316. (2000).

[61] **Malhotra, M.** and **Trivedi**, **K.,** Dependability modeling using Petri-nets. *IEEE Transactions on Reliability,* **44,** (3): 428-440, (1995).

[62] **Hurley, R. B.,** Probability maps, *IEEE Transactions on Reliability*, **R-12,** (3): 39-44, (1963).

[63] **Abraham, J. A.,** An improved algorithm for network reliability, *IEEE Transactions on Reliability*, **R-28,** (1): 58-61, (1979).

[64] **Dotson, W.** and **Gobien**, **J.,** A new analysis technique for probabilistic graphs, *IEEE Transactions on Circuits and Systems,* **CAS-26,** (10): 855-865, (1979).

[65] **Bennetts, R. G.,** Analysis of reliability block diagrams by Boolean techniques, *IEEE Transactions on Reliability*, **R-31**, (2): 159-166, (1982).

[66] **Rushdi, A. M.,** Symbolic reliability analysis with the aid of variable-entered Karnaugh maps, *IEEE Transactions on Reliability*, **R-32,** (2): 134-139, (1983).

[67] **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, **1**:57-95, (1983).

[68] **Schneeweiss, W. G.,** Disjoint Boolean products via Shannon's expansion, *IEEE Transactions on Reliability*, **R-34**, (4): 329-332, (1984).

[69] **Heidtmann, K. D.,** Smaller sums of disjoint products by subproduct inversion, *IEEE Transactions on Reliability*, **39**, (3): 305-311, (1989).

[70] **Rushdi, A. M.,** Karnaugh map, Encyclopaedia of Mathematics, Supplement Volume I, **M. Hazewinkel** (Editor), Boston, Kluwer Academic Publishers, pp: 327-328, (1997), Available at http://eom.springer.de/K/k110040.htm.

[71] **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).

[72] **Rushdi, A. M.,** Occam 's razor, *KAU Engineering Magazine,* **5**, (1): 58-61, (2011).

استعراض ومقارنة نوعين من التحليل اللاحق لأشجار الأخطاء

علي محمد علي رشدي ومحمد أحمد القواسمي

قسم الهندسة الكهربائية وهندسة الحاسبات, كلية الحاسبات وتقنية المعلومات،

جامعة الملك عبدالعزيز, جدة, المملكة العربية السعودية

arushdi@kau.edu.sa

المستخلص. إن أشجار الأخطاء هي أدوات تحليلية للاستنباط المنظم من أعلى إلى أسفل، وهي تتمتع بتطبيقات متنوعة في العديد من المجالات مثل المعولية والسلامة والأمن. ويمكن تسمية التحليل الأمامي لأشجار الأخطاء بالتحليل المسبق لأنه يتوقع احتمال الحدث الأوجي لشجرة الأخطاء بدلالة احتمالات أحداثها الأساسية. تقدم ورقة البحث هذه استعراضًا تعليميًا ومقارنة تفصيلية لنوعين من التحليل الخلفي أو اللاحق (البَعدي) لأشجار الأخطاء يتم تنفيذهما في الميدان الاحتمالي والميدان المنطقي (البولاني) على التوالي. نفترض في حالة التحليل اللاحق لأشجار الأخطاء في الميدان الاحتمالي كون احتمال الحدث الأوجي معلومًا، كأن يتأكد لنا وقوع هذا الحدث ومن ثم يصبح احتماله مساويًا للواحد الصحيح. يمضي هذا التحليل قدمًا بصورة معاودة في الميدان الاحتمالي لتقدير احتمالات الأحداث الأدنى في إطار بعض الافتراضات الواقعية، مثل: التنافي أو الاستقلال الإحصائي لأحداث المدخلات لبوابة منطقية محددة، ومع الاستفادة من تخمينات حصيفة لقيم نسب معينة بين احتمالات مثل هذه الأحداث. تقدم هذه الورقة إجراءً رياضيًا مفصلاً لتنفيذ هذا التحليل اللاحق لأشجار الأخطاء يعظم الانتفاع بمفهوم المزاوجة. ويتجلى هذا الإجراء من خلال مثال توضيحي مفصل. تدرس الورقة أيضًا التحليل اللاحق لأشجار الأخطاء في الميدان المنطقي. وهذا التحليل متوفر في أدبيات الموضوع في صورة أداة قوية جدًا تعرف باسم الشبكات البايبزية. نظهر هنا أنه في كثير من الحالات يظل هذا التحليل ممكنًا عن طريق معالجات أولية لأشجار الأخطاء تستخدم مفهوم خارج القسمة البولاني (المنطقي) للتنفيذ الفعال لنظرية بايبز في الميدان المنطقي. ومرة أخرى، يتم إعطاء مثال توضيحي لبيان التحليل اللاحق لأشجار الأخطاء في الميدان المنطقي، وشرح تفاصيله، وإظهار أن اللجوء لقوة الشبكات البايبزية ليس له ما يبرره حقًا في الحالات البسيطة. نورد مقارنة تفصيلية بين نوعي التحليل اللاحق لأشجار الأخطاء لإيضاح أوجه الشبه وأوجه الاختلاف بينهما.

الكلمات الدالة: شجرة الأخطاء، التحليل المسبق، التحليل اللاحق (البَعدي)، الميدان الاحتمالي، الميدان المنطقي (البولاني).