Quantitative analysis of a non-coherent fault tree structure using binary decision diagrams

PUBLISHER

REPOSITORY RECORD

Beeson, Sally C., and J.D. Andrews. 2008. “Quantitative Analysis of a Non-coherent Fault Tree Structure Using Binary Decision Diagrams”. figshare. https://hdl.handle.net/2134/3620.

QUANTITATIVE ANALYSIS OF A NON-COHERENT FAULT TREE STRUCTURE USING BINARY DECISION DIAGRAMS

S. Beeson and J.D. Andrews; Department of Mathematical Sciences, Loughborough University, Loughborough, Leicestershire, UK

Keywords: Non-coherent, Fault Trees, Binary Decision Diagrams, Expected Number of system failures

ABSTRACT

The BDD technique first introduced for the purposes of Fault Tree Analysis (FTA) by Rauzy in 1993 enables efficient qualitative analysis of all fault trees and accurate quantitative analysis of coherent fault trees. Quantitative analysis of non-coherent fault trees using the BDD technique is limited to calculating the top event probability. However, the extension of Birnbaum's measure of component reliability importance to non-coherent application developed by Beeson and Andrews in 2001 enables the calculation of expected number of system failures, $W_{SYS}(0,t)$ directly from the SFBDD.

INTRODUCTION

Fault Tree Analysis (FTA) is a well-known deductive technique introduced by Watson in the early 1960's to enable system reliability assessment. A fault tree diagram contains two basic elements, gates and events. The three fundamental gate types are: the AND gate, the OR gate and the NOT gate. Fault tree structures can be categorised as either coherent or non-coherent. A fault tree can be non-coherent if the NOT gate is used or directly implied, and results in component working states contributing to system failure.

FTA is split into two stages, qualitative analysis, which involves identifying all the possible causes of system failure, known as the minimal cut sets (prime implicant sets for non-coherent fault trees) and quantitative analysis, which involves quantifying system parameters relating to system availability and reliability. Although this is a useful technique it does have some disadvantages.

One major disadvantage of FTA is that it can be inefficient, even the most powerful computers may not be able to perform exact qualitative analysis especially for large fault trees with many repeated events. Although culling techniques can be employed they in turn have the disadvantage of producing only a partial list of minimal cut sets, which means approximate methods must be employed during quantification. These approximations usually rely on the basic events having a small likelihood of occurrence. If this condition is not met the results obtained can be inaccurate.

Rauzy introduced the Binary Decision Diagram (BDD) technique to overcome these shortfalls [1]. This technique requires the fault tree to be converted into a BDD, known as the SFBDD because it encodes the structure function of the fault tree. The SFBDD can be used to perform exact quantitative analysis of coherent fault trees. However quantitative analysis of non-coherent fault trees is limited. It would be desirable to be able to calculate the expected number of system failures in a given interval $W_{SYS}(0,t)$. This paper will show how the SFBDD can be used to calculate $W_{SYS}(0,t)$ for a non-coherent fault tree.

DEFINITION OF COHERENCY

A fault tree can be classified as either coherent or non-coherent according to its underlying logic. If during fault tree construction the failure logic is restricted to the use of the AND gate and the OR gate the fault tree is coherent. If however, the NOT gate is used or directly implied (by XOR) the fault tree can be non-coherent. A more precise definition of coherency is given below:

A fault tree is coherent if its structure function $\phi(\underline{x})$ complies with the definition of coherency given by the properties of relevance and monotonicity [2].

- Every component i is relevant

$$\phi(1_i,\underline{x}) \neq \phi(0_i,\underline{x}) \qquad \text{For some } \underline{x}$$

- Its structure function is monotonically increasing

$$\phi(1_i,\underline{x}) \geq \phi(0_i,\underline{x}) \qquad \forall i$$

Where

$$\phi(1_i,\underline{x}) = \phi(x_1,\ldots,x_{i-1},1,x_{i+1},\ldots,x_n)$$
$$\phi(0_i,\underline{x}) = \phi(x_1,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n)$$

And x_i are the Boolean indicator variables defining the state of each component

The first condition ensures that each component contributes to the system state. The second, an increasing[1] structure function ensures that the system state deteriorates (at least does not improve) with increasing numbers if component failures.

The use of NOT logic is generally discouraged during fault tree construction. This is because in a non-coherent system, component working states can contribute to system failure, which can be considered to be a bad design in that it has components working correctly contributing to system failure. Analysis of such structures also tends to be more complex and rarely provides additional information about the system. However, Andrews demonstrated that NOT logic is essential for successful analysis of multitasking systems [3].

THE BDD TECHNIQUE

A BDD is a directed acyclic graph. Thus all paths through the BDD are directed in one straight route from the top node known as the *Root Vertex* through *Non-Terminal Vertices* until a *Terminal Vertex* is reached. Paths terminate in one of two states, 1, corresponding to system failure or, 0, corresponding to the system functioning. Non-terminal vertices represent components and are connected to other vertices by branches. Each non-terminal vertex has a one branch, and a zero branch corresponding to the component failing and functioning respectively. Figure 1 highlights the features of a SFBDD.

---

[1] Non-decreasing

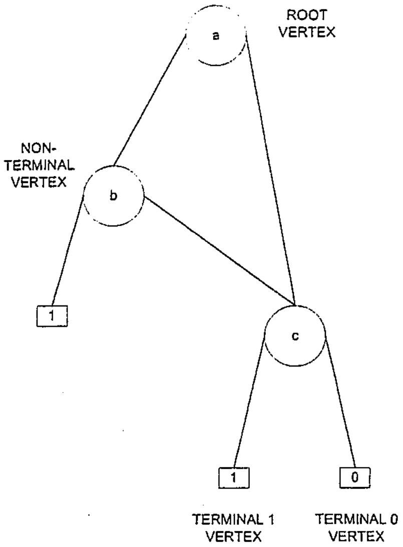

Figure 1: A general SFBDD

Rauzy developed an if-then-else (*ite*) method for computing the SFBDD from the fault tree. An example of this *ite* structure is given below in equation (1):

$$ite(x_i, f_1, f_2) \tag{1}$$

Where x_i represents a variable and f_1 and f_2 represent logic functions. This *ite* structure is interpreted as follows:

If i fails i.e. $x_i = 1$ *then* consider the logic function f_1
else consider the logic function f_2.

Thus in the BDD, f_1 forms the logic function for the one branch of x_i and f_2 forms the logic function for the zero branch of x_i. Figure 2 shows the diagram that represents this *ite* structure.

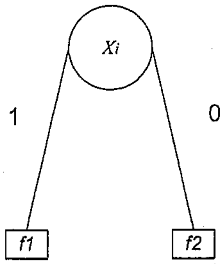

Figure 2: *ite* structure for $ite(x_i, f1, f2)$

In order to compute a SFBDD it is necessary to order the basic events in the fault tree, for example $a < b < c$. Numerous variable ordering schemes have been developed, one such scheme is the Top-Down approach, which will be used to order the basic events in the example below. Once a variable ordering scheme has been selected the *ite* procedure outlined below is applied to compute a SFBDD.

1. Assign each basic event x_i in the fault tree an *ite* structure.

   $x_i = ite(x_i, 1, 0)$ *ite* structure for normal literal

   $\overline{x_i} = ite(x_i, 0, 1)$ *ite* structure for a negated literal

2. Modify the fault tree structure so that each gate has only two inputs.

3. Consider each gate in a bottom-up fashion.
4. If the two gate inputs are J and H such that:

$$J = ite(x, F1, F2) \qquad H = ite(y, G1, G2)$$

Then the following rules are applied:

- *If* $x < y$, $J < op > H = ite(x, F1 < op > H, F2 < op > H)$
- *If* $x = y$, $J < op > H = ite(x, F1 < op > G1, F2 < op > G2)$

These rules are applied in conjunction with the identities given below:

$$1 < op > H = H,\ 0 < op > H = 0 \quad \text{if } <op> \text{ is an AND gate}$$
$$1 < op > H = 1,\ 0 < op > H = H \quad \text{if } <op> \text{ is an OR gate}$$

Where $< op >$ describes the Boolean operation of the logic gates of the fault tree. For an AND gate $< op >$ is the dot product (.) and for an OR gate $< op >$ is the sum symbol (+).

To illustrate how an SFBDD is computed consider the non-coherent fault tree in figure 3.

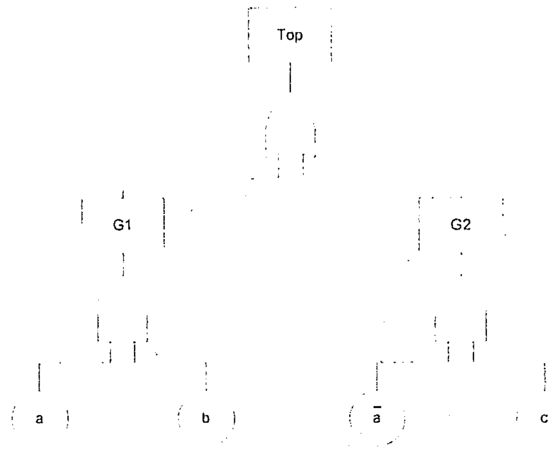

Figure 3: Non-coherent fault tree

Beginning by assuming a top-down variable ordering $a < b < c$:

Assigning each basic event in the fault tree an *ite* structure:

$$a = ite(a,1,0),\ \bar{a} = ite(a,0,1)$$
$$b = ite(b,1,0),\ c = ite(c,1,0)$$

Considering the gates in a bottom-up fashion beginning with gate $G1$:

$G1 = a \cdot b$

$G1 = ite(a,1,0) \cdot ite(b,1,0)$

$G1 = ite(a, [1 \cdot ite(b,1,0)], [0 \cdot ite(b,1,0)])$

$G1 = ite(a, ite(b,1,0), 0)$ Since $1 \cdot H = H$ and $0 \cdot H = 0$

Now dealing with gate $G2$

$G2 = \overline{a} \cdot c$

$G1 = ite(a,0,1) \cdot ite(c,1,0)$

$G1 = ite(a,[0 \cdot ite(c,1,0)],[1 \cdot ite(c,1,0)])$

$G1 = ite(a,0,ite(c,1,0))$ Since $1 \cdot H = H$ and $0 \cdot H = 0$

Finally dealing with the top gate Top:

$Top = G1 + G2$

$Top = ite(a,ite(b,1,0),0) + ite(a,0,ite(c,1,0))$

$Top = ite(a,[0 + ite(b,1,0)],[0 + ite(c,1,0)])$

$Top = ite(a,ite(b,1,0),ite(c,1,0))$ Since $0 + H = H$

The SFBDD obtained from the fault tree in figure 3 is given in figure 4.

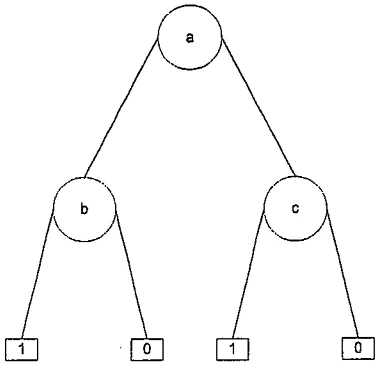

Figure 4: SFBDD for fault tree in figure 3

CALCULATING $W_{SYS}(0,t)$ USING BIRNBAUM'S MEASURE OF IMPORTANCE

The expected number of system failures is a valuable measure for assessing the system reliability. For coherent systems $W_{SYS}(0,t)$ can be expressed in terms of Birnbaum's measure of importance. This measure was developed by Birnbaum in 1969 [4] it is denoted by $G_i(\underline{q}(t))$ and defined as the probability that component i is critical to system failure.

$$G_i(\underline{q}(t)) = \frac{\partial Q_{SYS}(t)}{\partial q_i(t)} \tag{2}$$

The following identity can be used to calculate the expected number of system failures:

$$W_{SYS}(0,t) = \int_0^t \left(\sum_{i=1}^{n} G_i(\underline{q}(u)) w_i(u) \right) du \tag{3}$$

Where $w_i(t)$ is the unconditional failure intensity of component i and $q_i(t)$ is the failure probability of component i and n is the total number of components in the system.

Andrews and Sinnamon developed a procedure for calculating $W_{SYS}(0,t)$ using the SFBDD. This procedure is efficient, eliminating the need to evaluate lengthy series expansions. However Birnbaum developed this measure strictly for the analysis of coherent systems and the identity in equation (3) cannot be used to calculate $W_{SYS}(0,t)$ for a non-coherent system. Hence for non-coherent systems $W_{SYS}(0,t)$ must be calculated using traditional fault tree techniques, consequently approximations are unavoidable even for moderate sized trees.

In 2001 Beeson and Andrews extended Birnbaum's measure of component reliability importance to enable the analysis of non-coherent fault trees [6]. In a non-coherent system, a component i could be critical to the system state in one of two ways; it could be failure critical, or it could be repair critical. Thus the probability that component i is critical to the system is:

The probability that component i is failure critical
or component is repair critical.

Provided that Henley and Inagaki's procedure is used to calculate an expression for the system unavailability [7] the failure importance of component i is denoted by $G_i^F(\underline{q}(t))$ and defined as the probability that the system is in a state such that the failure of component i would cause the system to fail. Similarly the repair importance of component i is denoted by $G_i^R(\underline{q}(t))$ and defined as the probability that the system is in a state such that the repair of component i would cause system failure. The failure and repair importance of i can be expressed as follows:

$$G_i^F(\underline{q}(t)) = \frac{\partial Q_{SYS}(t)}{\partial q_i} \tag{4}$$

$$G_i^R(\underline{q}(t)) = \frac{\partial Q_{SYS}(t)}{\partial p_i} \tag{5}$$

Where p_i is the working probability of component i *and* q_i is the failure probability of component i.

The identity in equation (3) can now be extended to enable the analysis of non-coherent systems:

$$W_{SYS}(0,t) = \int_0^t \left(\sum_{i=1}^n G_i^F(\underline{q}(u)) w_i(u) + \sum_{i=1}^n G_i^R(\underline{q}(u)) v_i(u) \right) du \tag{6}$$

This extension enables $W_{SYS}(0,t)$ to be evaluated using the SFBDD enabling exact and efficient evaluation of this parameter. To evaluate $W_{SYS}(0,t)$ using equation (6) it is first necessary to obtain expressions for the failure and repair importance of each component. The procedure for calculating the failure and repair importance from the SFBDD is outlined below.

Consider a general node in the SFBDD, x_i for a non-coherent fault tree representing component i. The one branch of node x_i corresponds to the failure of component i, therefore component i is either failure critical or irrelevant. Similarly the zero branch of node x_i corresponds to the functioning of component i, therefore i is either repair critical or irrelevant.

Hence the sum of the probabilities of the terminal one paths that pass through the one branch of a node x_i represents the probability that component i is failure critical or irrelevant.

$$P(\text{Component } i \text{ is failure critical or irrelevant}) = \sum_{\substack{\text{All nodes}\\ x_i}} \Pr_{x_i}(\underline{q}(t)) Po^{1}_{x_i}(\underline{q}(t)) \tag{7}$$

Where $\Pr_{x_i}(\underline{q}(t))$ is the probability of the path section from the root vertex to node x_i and $Po^{1}_{x_i}(\underline{q}(t))$ is the probability of the path section from the 1 branch of node x_i to a terminal 1 node (excluding the probability of x_i).

An expression for the probability that component i is repair critical or irrelevant is obtained by summing the probabilities of the terminal one paths passing through the zero branch of a node x_i.

$$P(\text{Component } i \text{ is repair critical or irrelevant}) = \sum_{\substack{\text{All nodes}\\ x_i}} \Pr_{x_i}(\underline{q}(t)) Po^{0}_{x_i}(\underline{q}(t)) \tag{8}$$

Where $Po^{0}_{x_i}(\underline{q}(t))$ is the probability of the path section from the 0 branch of the node x_i to a terminal 1 node (excluding the probability of x_i).

The probability that component i is irrelevant can be obtained by taking the intersection of equations (7) and (8).

$$P(\text{Component } i \text{ is irrelevant}) = \sum_{\substack{\text{All nodes}\\ x_i}} \Pr_{x_i}(\underline{q}(t)) Po^{1}_{x_i}(\underline{q}(t)) \cap \sum_{\substack{\text{All nodes}\\ x_i}} \Pr_{x_i}(\underline{q}(t)) Po^{0}_{x_i}(\underline{q}(t)) = \sum_{\substack{\text{All nodes}\\ x_i}} \Pr_{x_i}(\underline{q}(t)) Po^{1}_{x_i}(\underline{q}(t)) Po^{0}_{x_i} \tag{9}$$

Then given that the failure importance is defined as the probability that component I is failure critical and the repair importance is defined as the probability that component I is repair critical:

$$G_i^F(\underline{q}(t)) = \sum_{\substack{\text{All nodes}\\ x_i}} \Pr_{x_i}(\underline{q})(t) Po^{1}_{x_i}(\underline{q}(t)) - \sum_{\substack{\text{All nodes}\\ x_i}} \Pr_{x_i}(\underline{q}(t)) Po^{1}_{x_i}(\underline{q}(t)) Po^{0}_{x_i}(\underline{q}(t)) \tag{10}$$

$$G_i^R(\underline{q}(t)) = \sum_{\substack{\text{All nodes}\\ x_i}} \Pr_{x_i}(\underline{q}(t)) Po^{0}_{x_i}(\underline{q}(t)) - \sum_{\substack{\text{All nodes}\\ x_i}} \Pr_{x_i}(\underline{q}(t)) Po^{1}_{x_i}(\underline{q}(t)) Po^{0}_{x_i}(\underline{q}(t)) \tag{11}$$

To illustrate how $W_{SYS}(0,t)$ is calculated consider the SFBDD in figure 4 obtained from the non-coherent fault tree in figure 3. Table 1 records $\Pr_{x_i}(\underline{q}(t))$, $Po^{1}_{x_i}(\underline{q}(t))$, $Po^{0}_{x_i}(\underline{q}(t))$ and for each node in the SFBDD. From these result the failure and repair importance of each component can be calculated.

NODE	$\Pr_{x_i}(\underline{q}(t))$	$Po^{1}_{x_i}(\underline{q}(t))$	$Po^{0}_{x_i}(\underline{q}(t))$

F1	1	$q_b(t)$	$q_c(t)$
F2	$q_a(t)$	1	0
F3	$1-q_a(t)$	1	0

Table 1: Results obtained for $\Pr_{x_i}(\underline{q}(t))$, $Po^1_{x_i}(\underline{q}(t))$, and $Po^0_{x_i}(\underline{q}(t))$ for each node in the SFBDD

From Table 1:

$$Po^1_a(\underline{q}(t))Po^0_a(\underline{q}(t)) = q_b(t)q_c(t)$$

$$Po^1_b(\underline{q}(t))Po^0_b(\underline{q}(t)) = 0$$

$$Po^1_c(\underline{q}(t))Po^0_c(\underline{q}(t)) = 0$$

From equations 10 and 11:

$$G^F_a(\underline{q}(t)) = (1 \cdot q_b(t)) - q_b(t)q_c(t) = q_b(t)(1 - q_c(t))$$

$$G^R_a(\underline{q}(t)) = (1 \cdot q_c(t)) - q_b(t)q_c(t) = q_c(t)(1 - q_b(t))$$

$$G^F_b(\underline{q}(t)) = (q_a(t) \cdot 1) - q_a(t) \cdot 0 = q_a(t)$$

$$G^R_b(\underline{q}(t)) = (q_a(t) \cdot 0) - q_a(t) \cdot 0 = 0$$

$$G^F_c(\underline{q}(t)) = ((1 - q_a(t)) \cdot 1) - (1 - q_a(t)) \cdot 0 = 1 - q_a(t)$$

$$G^R_c(\underline{q}(t)) = ((1 - q_a(t)) \cdot 0) - (1 - q_a(t)) \cdot 0 = 0$$

Hence from equation 6:

$$W_{SYS}(0,t) = \int_0^t ((q_b(u)(1 - q_c(u)))w_a(u) + q_a(u)w_b(u) + (1 - q_a(u))w_c(u) + (q_c(u)(1 - q_b(u)))v_a(u))du$$

CONCLUSION

The BDD technique for FTA improves the efficiency of qualitative analysis of both coherent and non-coherent fault trees and the accuracy of quantitative analysis of coherent fault trees.

This paper has demonstrated that it is possible to use the BDD technique to calculate the expected number of system failures for a non-coherent fault tree. Calculating this measure using the BDD technique has the advantage of producing an exact result, which is rarely possible using conventional FTA. Furthermore the calculation procedure is straightforward and efficient eliminating the need to evaluate lengthy series expansions.

REFERENCES

[1] A. Rauzy. "New Algorithms for Fault Tree Analysis". *Reliability Engineering and System Safety*, vol. 40, 1993, p203-211.

[2] A. Bendall and J. Ansell, "*The incoherency of Multistate Coherent Systems*", Reliability Engineering, vol. 81984, pp165-178.

[3] J. D. Andrews. "To Not or Not to Not". *Proceedings of the International System Safety Conference*, Forte Worth, Sept 2000, pp267-275.

[4] Z. W. Birnbaum. "*On the importance of Different Components in a Multi-component System*" Multivariate Analysis II, PR Krishnaiah, ed., Academic Press, 1969.

[5] R. M. Sinnamon and J. D. Andrews. "Quantitative Fault Tree Analysis Using Binary Decision Diagrams". European Journal of Automation, Vol. 30, No. 8, 1996.

[6] S. Beeson and J. D. Andrews. "*Birnbaum's measure of component importance for non coherent systems*"

[7] Inagaki and E. J. Henley. "Probabilistic Evaluation of Prime Implicants and Top Events for Non-Coherent Systems". *IEEE Transactions on Reliability*, vol. R-29 No. 5, Dec 1980.