Transactions of the Korean Nuclear Society Autumn Meeting
PyeongChang, Korea, October 25-26, 2007

Verification of Fault Tree Analysis Results Particularly in the Logical Loop Problems

Jong Soo Choi
Korea Institute of Nuclear Safety, 19 Guseong-dong Yuseong-gu Daejeon 305-338, k209cjs@kins.re.kr

1. Introduction

A logical loop or a circular logic is defined as the infinite circulation of supporting relations due to their mutual dependencies among the systems in the fault tree analysis. This happens typically when a mainline system is fed by support systems in the PSAs of nuclear power plants. The logical loop problem has been solved by manually or automatically breaking the circular logic at the point where the dependency among the systems is relatively weak. While many methods [1-3] to break the circular logic have been developed and used in the fault tree quantification codes, the general solution for a logical loop is not generally known as yet. The breaking of the logical loops could be one of uncertainty sources in the fault tree analysis.

This paper presents a direct simulation method verifying whether a fault tree analyst yields correct minimal cut sets particularly in the logical loop problems.

2. Methods and Results

2.1 Characteristics of Minimal Cut Sets (MCSs)

A fault tree is mathematically represented by a set of Boolean equations that integrate the basic events to the top event. Here, we consider a fault tree consisting of k basic events $\mathbf{B} = (b_1, \ldots, b_k)$. For $i = 1, \ldots, k$, x_i is the binary variable for the basic event b_i, and hence it is equal to 1 if and only if the basic event is occurring. The structure function $\phi(\mathbf{X})$ of a fault tree is a deterministic binary function of the vector $\mathbf{X}$ of the basic event states, and hence it is equal to 1 when the top event is occurring.

For $\mathbf{X}$ indicating the states of the basic events, we define $\mathbf{B_1}(\mathbf{X}) = \{b_i | x_i = 1\}$ and $\mathbf{B_0}(\mathbf{X}) = \{b_i | x_i = 0\}$. A cut vector is any $\mathbf{X}$ such that $\phi(\mathbf{X}) = 1$ and $\mathbf{B_1}(\mathbf{X})$ is the corresponding cut set. $\mathbf{B_1}(\mathbf{X})$ is a MCS if it is a cut set with no proper subset as a cut set. In other words, each MCS, $\mathbf{K}$, has the following properties:

P1: If $\mathbf{B_1}(\mathbf{X}) = \mathbf{K}$, then $\phi(\mathbf{X}) = 1$. (1)

P2: If $\mathbf{B_1}(\mathbf{X}) = \mathbf{K}$ and $i \in \mathbf{K}$, then $\phi(0_i, \mathbf{X}) = 0$. (2)

In fault tree analyses, only MCSs with probability above the pre-established cut-off value V_c are developed and the other cut sets are discarded. Thus, each developed MCS has probability greater than V_c:

$$\Pr\{\mathbf{K}_j\} > V_c \quad \text{for } j = 1, \mathrm{K}, m. \tag{3}$$

In evaluating a looped fault tree, analysts should manually or automatically break the circular logics. The accuracy of the quantification results is consequently dependent on the capability of the analyst and the analysis code.

2.2 Proposed Solver of Boolean Equations

Given a random state vector $\mathbf{X}$, the binary values $\phi(\mathbf{X})$ and $\phi(0_i, \mathbf{X})$ of a fault tree can be determined by the set of Boolean equations that integrate the basic events to the top event. The computing time to solve the states of the top event and all intermediate gates given $\mathbf{X}$ depends mainly on the number of the relevant Boolean equations and the algorithm to solve Boolean equations.

The procedure of the proposed Boolean solver is as follows.

(1) Initiate the states of all the intermediate gates as 0 and allocate all the basic event states as a new $\mathbf{X}$.
(2) Determine the states of the parent gates of the occurring basic events, $\mathbf{B_1}(\mathbf{X})$, using their Boolean equations.
(3) Identify the gates whose states are changed from 0 to 1 or from 1 to 0.
(4) Identify the parent gates of the newly-changed gates as shown in Figure 1.
(5) Determine the states of the newly-identified parent gates using their Boolean equations.
(6) Go to Step (3) until no gates change their states.

The proposed Boolean solver was implemented into CUTREE [4,5] and is applicable to coherent fault trees. It is efficient and fast, because only a small number of gates will be calculated in case of a small $\mathbf{B_1}(\mathbf{X})$. Even in the logical loop problems, it is also a correct and efficient solver to determine the states of the top event and all intermediate gates given $\mathbf{X}$.

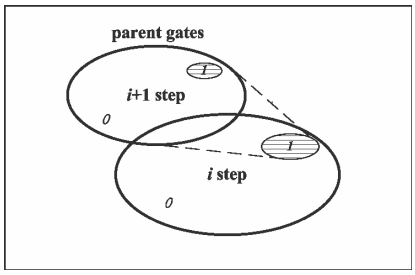

Figure 1.Procedure of the proposed Boolean solver

2.3 Verification of Identified Minimal Cut Sets (MCSs)

In the logical loop problems, we generally identify the MCSs only from a tree without logical loops nearly equivalent to the original looped tree. So, we need to verify the identified MCSs.

As described in Sec.2.2, the proposed Boolean solver is applicable to looped trees and provides exact binary values $\phi(\mathbf{X})$ and $\phi(0_i, \mathbf{X})$ given $\mathbf{X}$. So, we can test the correctness of the identified MCSs using the two MCS properties formulated in Eq. (1) and (2). If any MCS

does not meet **P1** and **P2**, it is certain that there are defects in breaking the logical loops and the unsatisfying set is not a real MCS. This algorithm to verify minimal cut sets was implemented into CUTREE [4,5]

2.4 Searching Missing MCSs

Ref. [5] proposed "Delta-X Monte Carlo method" to quantify the truncation errors. The binary function $\delta(\mathbf{X})$ is associated only with unidentified cut sets and it is defined as:

$$\delta(\mathbf{X}) \equiv \phi(\mathbf{X}) - \phi^m(\mathbf{X}), \quad (4)$$

where $\phi^m(\mathbf{X})$ is associated with all the identified MCSs.

After a great number of simulations, cut sets which are not subset of the identified MCSs can be newly identified. Using newly-identified cut sets, we can get new MCSs by recursive calculations of $\phi(0_i,\mathbf{X})$. If any newly-identified MCS meets Eq. (3), it is certain that there are defects in breaking the logical loops and there are some missing MCSs with occurring probability greater than V_c.

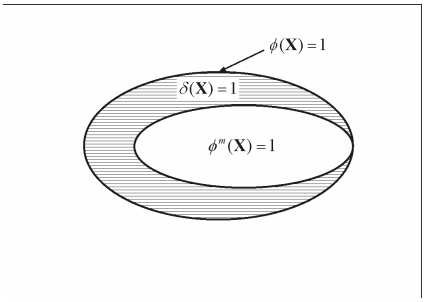

Figure 2. Newly-identified cut-set region, $\{\mathbf{X}|\delta(\mathbf{X})=1\}$.

2.5 Application to Logical Loop Problems

An example fault tree with logical loops is made by modifying the tree 'European 1' (given in Ref. [6]) as:

from: G068 := (C001 & C008)
to: G068 := (C001 & C008 & ROOT)

Example tree is summarized as follows:

Number of basic events (BE): 61
Number of gates: 84
Number of gates located on logical loops: 26
Number of MCSs: 32876 (using FTREX [2]) (without cut-off)
Prob. of top event: 7.1306E-7

Through a proposed direct simulation on Example tree, no defects in the FTREX results were identified. The simulation results are summarized as follows:

(1) Verification of the identified MCSs
- CPU time: 66 sec. (on a 2 GHz Pentium IV)
- All MCSs meet **P1** and **P2**.

(2) Search missing MCSs
- # of simulations: 1E9
- CPU time: 173 sec. (on a 2 GHz Pentium IV)
- No missing MCSs are identified.

3. Conclusion

This paper presents a verification method of fault tree analysis results which is based on direct simulation of Boolean equations of a fault tree. This paper also shows that we can verify the fault tree analysis results particular in the logical loop problems using the proposed method.

REFERENCES

[1] J. E. Yang, S. H. Han, J. H. Park, Y. H. Jin, Analytic Method to Break Logical Loops Automatically in PSA, Reliability Engineering and System Safety, Vol.56, p.101, 1997.
[2] W. S. Jung, S. H. Han, Development of an Analytical Method to Break Logical Loops at the System Level, Reliability Engineering and System Safety, Vol.90, p.37, 2005.
[3] H. G. Lim, S. C. Jang, An Analytic Solution for a Fault Tree with Circular Logics in which the Systems are Linearly Interrelated, Reliability Engineering and System Safety, Vol.92, p.804, 2007.
[4] J. S. Choi, N. Z. Cho, NurapT-2006-03, A User's Manual of the CUTREE Code Package for Minimal Cutset-Based Fault Tree Analysis, Ver. 1.0, KAIST, Department of Nuclear and Quantum Engineering, June 2006.
[5] J. S. Choi, N. Z. Cho, A Practical Method for Accurate Quantification of Large Fault Trees, Reliability Engineering and System Safety, Vol.92, p.971, 2007.
[6] A. Rauzy, New Algorithms for fault trees analysis, Reliability Engineering and System Safety, Vol.40, p.203, 1993.