Hybrid method to detect modules in a fault tree for a PSA

Woo Sik Jung[a]
[a] *Sejong University, 209 Neungdong-Ro, Gwangjin-Gu, Seoul 143-747, South Korea*

1. Introduction

In 1996, an efficient Dutuit and Rauzy (DR) algorithm to detect independent subtrees (modules) in a fault tree was introduced (see Appendix A) [1]. More efficient algorithm was recently proposed (see Appendix B) [2]. This paper introduces a hybrid method of the two previous methods.

Fault tree quantification generates fault tree solutions such as cut sets, path sets, or binary decision diagrams (BDDs), and then, calculates top event probability and importance measures. A module or independent subtree is a part of a fault tree whose child gates or basic events are not repeated in the remaining part of the fault tree. Modules are necessarily employed in order to reduce the computational costs of fault tree quantification. A way to reduce this computational complexity is to detect modules in a fault tree and replace them with artificial super-components.

Fault tree analysis has been extensively and successfully applied to the risk assessment of safety-critical systems such as nuclear, chemical, and aerospace systems [3,4]. The fault tree analysis has been used together with event tree analysis in probabilistic safety assessment (PSA) of nuclear power plants since WASH-1400 report [5].

Fault tree analysis for the safety-critical systems requires mostly very expensive calculation. In order to reduce the computational difficulty in performing fault tree analysis, cutset truncation during the cutset generation is usually employed. However, although the cutset truncation is performed, it is frequently difficult to generate a sufficient number of cut sets for the calculation of the accurate top event probability and importance measures. The sufficient number of cut sets is sometimes in the range from 1,000 to 1,000,000 in PSA of nuclear power plants.

3. Hybrid method to find modules

All nodes are visited along a depth-first leftmost traversal. The traversal starts and finally ends at the root node with zero module measures. For clear explanation of the new algorithm, the numbers in node names are increased along the traversal.

For efficient module detection, module measure is newly introduced in this study. When leaving a repeated node v for the first time and going to next node v, module measure is increased one time as

$$P_{in/out}(w) = C_{out}(v) + (Count(v) - 1)\Delta_v \quad (1)$$

Please note that two module measures were introduced in the previous study [2]. The measure in Eq. (1) is one of them.

Whenever leaving this repeated node v from the second time and going to next node w, module measure is decreased stepwise as

$$P_{in/out}(w) = P_{out}(v) - \Delta_v \, . \quad (2)$$

Here, Δ_v is identical for all nodes as

$$\Delta_1 = \Delta_2 = \Delta_3 = \cdots = 1 \, . \quad (3)$$

In Eqs. (1) and (2), $P_{in/out}(w)$ is one of $P_{in}(w)$ and $P_{out}(w)$. Module measures $P_{in}(v)$ and $P_{out}(v)$ along the traversal of a fult tree are illustrated in Fig. 1. Their variations along the traversal in a whole fault tree are depicted in Fig. 1 and listed in Table 1. Module measures are increased by Eq. (1) one time when leaving repeated nodes such as {g3, e12, e13} for the first time, and decreased stepwise by Eq. (2) whenever leaving these repeated nodes the other time. However, there is no change in module measure when leaving non-repeated nodes.

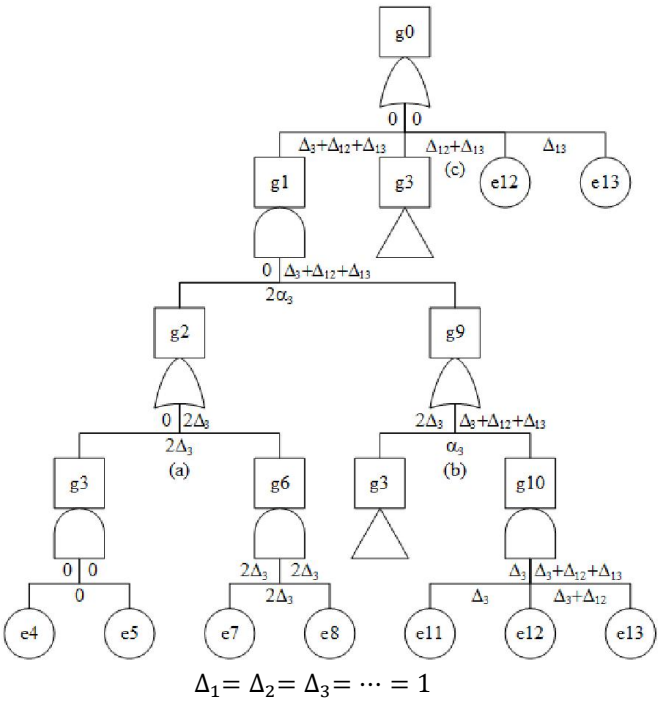

Fig. 1. Hybrid method to find modules

The changes of module measures when entering and leaving repeated e13 under node g0 in Fig. 1 are

$$P_{in}(e13) = P_{out}(e12) - \Delta_{12} = (\Delta_{12} + \Delta_{13}) - \Delta_{12} = \Delta_{13}$$
$$P_{out}(g0) = P_{out}(e13) - \Delta_{13} = \Delta_{13} - \Delta_{13} = 0 \quad (4)$$

Table 1. Module identification

Node v	$P_{in}(v)$	$P_{out}(v)$	$Visit_{first}(v)$	$Visit_{min}(v)$	Module
g0	0	0	1	2	Yes
g1	0	$\Delta_3 + \Delta_{12} + \Delta_{13}$	2	3	No
g2	0	$2\Delta_3$	3	4	No
g3	0	0	4(a)	5	Yes
e4	0	0	5		
e5	0	0	6		
g6	$2\Delta_3$	$2\Delta_3$	8	9	Yes
e7	$2\Delta_3$	$2\Delta_3$	9		
e8	$2\Delta_3$	$2\Delta_3$	10		
g9	$2\Delta_3$	$\Delta_3 + \Delta_{12} + \Delta_{13}$	13	4(a)	No
g10	Δ_3	$\Delta_3 + \Delta_{12} + \Delta_{13}$	15	16	No
e11	Δ_3	Δ_3	16		
e12	Δ_3	Δ_3	17		
e13	$\Delta_3 + \Delta_{12}$	$\Delta_3 + \Delta_{12}$	18		

(a) $Visit_{first}(g3)$

If there is no change in the module measure between entering and leaving a node v

$$P_{out}(v) - P_{in}(v) = 0 \quad (5)$$

and a node v satisfies the inequality,

$$Visit_{first}(v) < V_{isitmin}(v) . \quad (6)$$

The node v is a module of the fault tree.

As shown in Fig. 1 and Table 1, module measures and visiting steps of nodes g0, g3, and g6 satisfy Eqs. (5) and (6). So, these three nodes are modules of the fault tree as

$$Module(g0) = \{g0, g3, g6\} . \quad (7)$$

4. Conclusions

This paper proposes a hybrid method of (1) DR algorithm using visiting steps of nodes [1], and (2) more efficient algorithm using two module measures [2]. Benchmark tests show that hybrid method in this paper and two module measure-based algorithm [2] detect modules two times faster than the DR method [1].

The algorithm in this paper minimizes computational memory and quickly detects modules. Furthermore, it can be easily implemented into industry fault tree solvers. It is recommended that this method be implemented into fault tree solvers for efficient probabilistic safety assessment of nuclear power plants.

REFERENCES

[1] Y. Dutuit and A. Rauzy, "A Linear Time Algorithm to Find Modules of Fault Trees," IEEE Transactions on Reliability, Vol. 45, no. 3, pp. 422–425, 1996.
[2] W.S. Jung, "New algorithm to detect modules in a fault tree for a PSA," Transactions of the Korean Nuclear Society Spring Meeting, Jeju, Korea, May 6-8, 2015.
[3] W.S. Lee, D. L. Grosh, F. A. Tillman, and C. H. Lie, "Fault Tree Analysis, Methods, and Applications - A Review," IEEE Transactions on Reliability, Vol. R-34, no. 3, pp. 194–203, Aug. 1985.
[4] C. A. Ericson, "Fault tree analysis - A history," in Proceedings of the 17th International System Safety Conference, pp. 1–9, 1999.
[5] U.S. NRC, Reactor safety study - an assessment of accident risk in U.S. commercial nuclear power plants, WASH-1400, NUREG-75/014, 1975.

Appendix A. DR method [1]

Dutuit and Rauzy [1] proposed an efficient module detection algorithm (DR method) that is based on the depth-first leftmost traversal of a fault tree. Visiting steps $Visit_{first}(v)$, $Visit_{second}(v)$, and $Visit_{last}(v)$ are introduced for DR method. Additionally, the steps such as $Visit_{min}(v)$ and $Visit_{max}(v)$ are also defined [1,2] as

$$Visit_{min}(v) = \min_{w \in Event_{in}(v)} Visit_{first}(w) \quad (A.1)$$
$$Visit_{max}(v) = \max_{w \in Event_{in}(v)} Visit_{last}(w). \quad (A.2)$$

In the DR method, a node v is a module if its visiting steps satisfy the inequalities

$$Visit_{first}(v) < Visit_{min}(v) < Visit_{max}(v) < Visit_{second}(v). \quad (A.3)$$

Appendix B. Scalar variable-based algorithm [2]

Similarly to the DR method[1], all nodes are visited along a depth-first leftmost traversal. For efficient module detection, two module measures are newly introduced [2]. Repeated number of a node v in a fault tree is defined as $Count(v)$. When leaving a repeated node v for the first time and going to next node w, module measures are increased one time as

$$P_{in/out}(w) = P_{out}(v) + (Count(v) - 1) \quad (B.1)$$
$$Q_{in/out}(w) = Q_{out}(v) + (Count(v) - 1)\alpha_v . \quad (B.2)$$

Whenever leaving this repeated node v from the second time and going to next node w, module measures are decreased stepwise as

$$P_{in/out}(w) = P_{out}(v) - 1 \quad (B.3)$$
$$Q_{in/out}(w) = Q_{out}(v) - \alpha_v . \quad (B.4)$$

Here, α_v is an integer value as $\alpha_v = v$.

In Eqs. (B.1) to (B.4), $P_{in/out}(w)$ is one of $P_{in}(w)$ and $P_{out}(w)$, and $Q_{in/out}(w)$ denotes one of $Q_{in}(w)$ and $Q_{out}(w)$.

If there are no changes in the module measures between entering and leaving a gate v

$$P_{in}(v) = P_{out}(v) \quad (B.5)$$
$$Q_{in}(v) = Q_{out}(v) , \quad (B.6)$$

the gate v is a module of the fault tree.