Advanced Features of the Fault Tree Solver FTREX

Woo Sik Jung, Sang Hoon Han, Jaejoo Ha
Korea Atomic Energy Research Institute, P.O.Box 105, Yusong, Daejon, Korea, woosjung@kaeri.re.kr

1. Introduction

This paper presents advanced features of a fault tree solver FTREX (Fault Tree Reliability Evaluation eXpert). Fault tree analysis is one of the most commonly used methods for the safety analysis of industrial systems especially for the probabilistic safety analysis (PSA) of nuclear power plants.

Fault trees are solved by the classical Boolean algebra[1,2], conventional Binary Decision Diagram (BDD) algorithm[3], coherent BDD algorithm[4,5], and Bayesian networks[6,7]. FTREX could optionally solve fault trees by the conventional BDD algorithm or the coherent BDD algorithm and could convert the fault trees into the form of the Bayesian networks.

The algorithm based on the classical Boolean algebra solves a fault tree and generates MCSs. The conventional BDD algorithm generates a BDD structure of the top event and calculates the exact top event probability. The BDD structure is a factorized form of the prime implicants. The MCSs of the top event could be extracted by reducing the prime implicants in the BDD structure. The coherent BDD algorithm is developed to overcome the shortcomings of the conventional BDD algorithm such as the huge memory requirements and a long run time.

2. Features of FTEX

2.1 Coherent BDD algorithm

A set of new formulae[4] was developed for the operation between the two ITE connectives of a coherent fault tree using the simplified Shannon decomposition. If x and y are two variables with a variable ordering $x<y$, then the following equalities hold for coherent systems

$$\begin{aligned} &ite(x,G_1,G_2)\cdot ite(x,H_1,H_2) \\ &\quad = ite(x,(G_1H_1+G_1H_2+G_2H_1),G_2H_2) \\ &ite(x,G_1,G_2)+ite(x,H_1,H_2) \\ &\quad = ite(x,(G_1+H_1),(G_2+H_2)) \\ &ite(x,G_1,G_2)\cdot ite(y,H_1,H_2)=ite(x,G_1h,G_2h) \\ &ite(x,G_1,G_2)+ite(y,H_1,H_2)=ite(x,G_1,(G_2+h)) \end{aligned} \quad (1)$$

where $h=ite(y,H_1,H_2)$. Here, please note that the first and last equations in Eq. (1) differ from the operations in the conventional BDD algorithm[3].

In order to get minimal solutions of a BDD structure, the subsuming is recursively performed from the root ITE to the child ITE connectives by comparing the left and right ITE connectives. Let us consider recursive ITE connectives $F=ite(t,G,H)$, $G=ite(x,G_1,G_2)$, and $H=ite(y,H_1,H_2)$. In order to get MCSs of F, a cut set in G is deleted if H has its super sets (subsuming operation $G\backslash H$). Rauzy[3] proposed an efficient subsuming operation:

$$\begin{aligned} &subsume(G,H)=G\backslash H \\ &=\begin{cases} ite(x,G_1\backslash H,G_2\backslash H) & ,x<y \\ G\backslash H_2 & ,x>y \\ ite(x,G_1\backslash(H_1 \text{ or } H_2),G_2\backslash H_2) & ,x=y \end{cases} \end{aligned} \quad (2)$$

The term $G_1\backslash(H_1$ or $H_2)$ in the last case denotes that each cut set in G_1 is tested and deleted if it is a subset of a cut set in H_1 or H_2.

The truncation and subsuming in the progress of the construction of the BDD structure is the key to a fast quantification of large coherent fault trees using less memory. Benchmark tests[5] were performed for the large coherent fault trees that could not be solved by the conventional BDD algorithm in a reasonable time and memory usage. FTREX showed a desirable performance.

2.2 Truncated probability estimation

When the truncation limit is 1×10^{-k}, the approximate truncated probability (ATP) and the lower bound of truncated probability (LBTP) could be defined as follows

$$ATP_k=\overline{P}_k+\Delta P_k \quad (3)$$

$$LBTP_k=\overline{P}_k \quad (4)$$

where $\Delta P_k=P_k-P_{k-1}$. Here, $\overline{P}_k$ is a sum of probabilities of MCSs $\overline{C}_i^k$ that are truncated when expanding the modules in the final cut sets, and P_k is a sum of the resultant MCS probabilities. The probability $\overline{P}_k$ could be easily calculated since the truncated MCSs $\overline{C}_i^k$ could be obtained by modifying the existing fault tree solvers.

The measures LBTP and ATP are desirable estimators of the truncated probability and they can be used to estimate the top event probability[8]. A single quantification of the fault tree with an assigned truncation limit is sufficient enough to calculate the measures.

2.3 Automatic logical loop break

There are two ways to break logical loops such as the analytical[9,10] and manual breaking methods. Yang[9] presented an analytical method to break the logical loops. A much easier analytical method than that was invented

by Yang had been implemented into KIRAP[1] and FTREX[5]. By recursively navigating the fault tree, every combination that causes the logical loops is found one by one and the last gate of each logical loop is deleted. KIRAP and FTREX recursively search and break the logical loops in a fault tree and then solve the broken fault tree.

The paper by Jung[10] presents an analytical method to break the logical loops at the system level. The analytical solution at the system level is obtained in a mathematical way without an actual manipulation of the fault tree. Then, the actual manipulation of the fault tree in the analytical solution is performed and the resultant broken fault tree is solved by the fault tree quantifier.

2.4 Rule-based post processing

In a PSA, operator actions that could prevent an accident sequence may not be specifically included in the logic models. To model the accident sequences as accurately as practical, the reliability analyst apply recovery events to the appropriate MCSs. The recovery events denote the failure of the recovery action.

In FTREX, the rule-based operations could be performed on the BDD structure by using the subsuming operator in Eq. (2). Let us illustrate the rule-based operation on the BDD structure. A Boolean algebra for a given gate F is

$$F = ab \cdot G_1 + cde \cdot G_2 + G_0 \quad (5)$$

and

$$G_1 = f \cdot G_{11} + gh \cdot G_{12} + G_{10}$$
$$G_2 = f \cdot G_{21} + gh \cdot G_{22} + G_{20}.$$

The given condition with an exception that we want to apply to the gate is

$$condition\ \ C = ab + cde \quad (6)$$
$$exception\ \ E = f + gh. \quad (7)$$

By using the subsuming operation that is defined in Eq. (2), a new Boolean equation that satisfies the given condition could be calculated as follows

$$F_1 = subsume(F, C) = G_0 \quad (8)$$
$$F_2 = subsume(F, F_1) = ab \cdot G_1 + cde \cdot G_2$$
$$F_3 = subsume(F_2, E) = ab \cdot G_{10} + cde \cdot G_{20}$$

where F_3 denotes MCSs satisfying the condition with an exception. As shown in this example, the new Boolean equation that is in the form of a BDD structure could be obtained by some subsuming operations on the BDD structure. Furthermore, the rule based operations on the BDD structure (add new events, delete MCSs, and replace some events with new events) could be done on the new BDD structure.

2.5 Minimal Prevention Sets

FTREX calculates “minimal prevention sets” for the vital area identification of a nuclear power plant[11]. A minimal prevention set of level L contains at least L basic events from each MCS and it guarantees no occurrence of top event.

4. Conclusion

The fault tree solver FTREX showed a desirable performance. FTREX optionally solves fault trees by the conventional BDD algorithm or coherent BDD algorithm. FTREX could convert the fault trees into the input files to the Bayesian network algorithms. Furthermore, FTREX has special features such as a truncated probability estimation, logical loop breaking, and rule-based post processing capabilities.

REFERENCES

[1] S.H. Han, "PC-Workstation Based Level 1 PRA Code Package-KIRAP," Reliability Engineering and System Safety, Vol. 30, pp.313-322, 1990.

[2] W.S. Jung, et al., "FORTE: A Fast New Algorithm for Risk Monitors and PSA," Proceedings of the 4th International Conference on Probabilistic Safety Assessment and Management, September, New York, USA, p.1221, 1998.

[3] A. Rauzy, "New Algorithms for Fault Trees Analysis," Reliability Engineering and System Safety, Vol. 40, pp. 203-211, 1993.

[4] W.S. Jung, et al., "A Fast BDD Algorithm for Large Coherent Fault Trees Analysis," Reliability Engineering and System Safety, Vol. 83, pp. 369-374, 2004.

[5] W.S. Jung, et al., "Development of an Efficient BDD Algorithm to Solve Large Fault Trees," Proceedings of the 7th International Conference on Probabilistic Safety Assessment and Management, June, Berlin, Germany, 2004.

[6] M.C. Kim, et al., "Reliability Graph with General Gates: An Intuitive and Practical Method for System Reliability Analysis, Reliability Engineering and System Safety", Vol. 78, pp.239-246, 2002.

[7] M.C. Kim, et al., "A Comparison between Fault Tree Analysis and Reliability Graph with General Gates," Proceedings of the Korean Nuclear Society Fall Meeting, Yongpyong, Korea, 2004.

[8] W.S. Jung, et al., "Development of measures to estimate truncation error in fault tree analysis," Reliability Engineering & System Safety, in Press.

[9] J.E. Yang, 1997, "Analytic Method to Break Logical Loops Automatically in PSA," Reliability Engineering and Safety, Vol. 56, pp. 101-105, 2005.

[10] W.S. Jung, et al., "Development of an analytical method to break logical loops at the system level," Reliability Engineering & System Safety, in Press, 2005.

[11] C.K. Park, et al., "Development of a PSA-based Vital Area Identification Methodology for the Physical Security of Nuclear Power Plants," Proceedings of the 7th International Conference on Probabilistic Safety Assessment and Management, June, Berlin, Germany, 2004.