How to solve noncoherent fault trees by zero-suppressed ternary decision diagram algorithm for probabilistic safety assessment of nuclear power plants

Woo Sik Jung[a*], Seong Kyu Park[b]

[a]Sejong University, Seoul, Republic of Korea
[b]Nuclear Engineering Services & Solutions, Gyeonggi-Do, Republic of Korea

Abstract: *In a probabilistic safety assessment (PSA), the core damage frequency (CDF) is estimated by calculating the minimal cut sets (MCSs) of the noncoherent core-damage fault tree. Because of the computational difficulty in obtaining an accurate solution of noncoherent accident sequence logics, delete-term approximation (DTA) is employed to approximately solve MCSs representing accident sequence logics from noncoherent core-damage fault trees. However, DTA leads to an overestimation of CDF for fault trees containing many nonrare events and complemented gates. To minimize this CDF overestimation, this study introduces a hybrid method of the accurate and approximate solutions and it was implemented into a new tool ZEBRA (ZTDD Equation Based Risk Analyzer) that is based on zero-suppressed ternary decision diagram (ZTDD) algorithm.*

Keywords: *Probabilistic safety assessment, Delete-term approximation, Zero-suppressed ternary decision diagram algorithm.*

1. INTRODUCTION

In a probabilistic safety assessment (PSA), the core damage frequency (CDF) is estimated by calculating the minimal cut sets (MCSs) of the core-damage fault trees. Delete-term approximation (DTA) [1,2] is commonly employed to approximately solve MCSs representing accident sequence logics that are inherently from noncoherent fault trees. However, DTA results in an overestimation of CDF, particularly when fault trees contain many nonrare events and complemented gates. To minimize this CDF overestimation caused by DTA, this paper presents a hybrid method of the accurate and approximate solutions, and its implementation to a new tool ZEBRA (ZTDD Equation Based Risk Analyzer) that is based on zero-suppressed ternary decision diagram (ZTDD) algorithm [1,2] (see Section 4). Furthermore, the efficiency of the proposed method was demonstrated by performing benchmark tests (see Section 5).

The PSA of nuclear power plants is performed using event and fault trees. Each accident sequence in event trees comprises a logical combination of usual and complemented fault trees representing safety system failures and successes, respectively. Because the general Boolean expression for an accident scenario $G_iG_{i+1}\ldots/G_j/G_{j+1}\ldots$ is identical to $G_iG_{i+1}\ldots/(G_j+G_{j+1}+\cdots)$, it can be expressed as G_1/G_2, where $G_1 = G_iG_{i+1}\ldots$ and $G_2 = G_j + G_{j+1} + \cdots$. Eq. (1) represents a typical accident sequence, and accurate or approximate solutions will be explained using Eq. (1). Here, $/G_2$ is a Boolean complement (BC) of G_2, G_1/G_2 denotes AND Boolean combination of G_1 and $/G_2$, ab denotes AND Boolean combination of basic events a and b, and $ab + bc + be$ denotes OR Boolean combination of ab, bc, and be.

$$\begin{gathered} G_1/G_2 \\ G_1 = ab + bc + be \\ G_2 = bc + bd \end{gathered} \tag{1}$$

2. VARIOUS METHODS TO SOLVE G_1/G_2

2.1 Accurate solution by Boolean complement

By using the traditional Boolean algebra in Eqs. (2) and (3), the solution of $/G_2$ and an accurate solution of G_1/G_2 can be calculated. However, it is well known that the Boolean complement of G_2 frequently fails because it generates many solutions and rapidly consumes a limited computational memory. Thus, this Boolean complement is practically impossible for an actual PSA.

$$G_1/G_2 = G_1 \cdot BC(G_2) \tag{2}$$

$$\begin{aligned} BC(G_2) &= /(bc + bd) = (/b + /c)(/b + /d) = /b + /c/d \\ G_1 \cdot BC(G_2) &= (ab + bc + be) \cdot BC(bc + bd) = (ab + bc + be)(/b + /c/d) \\ &= ab/c/d + b/c/de \end{aligned} \tag{3}$$

As an alternative of the traditional Boolean algebra, a solution of a binary decision diagram (BDD) [3-13] for G_1/G_2 can be directly calculated from the fault tree in Eq. (1) using the BDD algorithm [5-7], and the accurate probability $p(G1/G2)$ is calculated with this BDD. However, the BDD calculation from the fault tree also frequently fails for large fault trees in PSA.

2.2 Approximate solution by DTA

In the current PSA for the risk analysis of a nuclear power plant, the MCSs of G_1 and G_2 are separately generated using traditional Boolean algebra [14] or the zero-suppressed binary decision diagram (ZBDD) algorithm [15-17]. Then, approximate MCSs of G_1/G_2 are generated using DTA, which compares the MCSs of G_1 and G_2 and deletes some nonlogical MCSs of G_1as shown in Eq. (5). However, CDF overestimation is inevitable. It can be confirmed by comparing the solutions of Eqs. (3) and (5). If the event probabilities are not small, the probability of $ab + be$ is much higher that of $ab/c/d + b/c/de$.

$$G_1/G_2 \cong DTA(G_1, G_2) \tag{4}$$

$$TOP \cong DTA(ab + bc + be, bc + bd) = ab + be \tag{5}$$

3. PROPOSED METHOD TO SOLVE G_1/G_2

3.1 Accurate solution by combining DTA and BC

As shown in Eqs. (6) and (7), the accurate solution that is identical to that of Eq. (3) can be obtained more quickly by combining Eqs. (2) and (4). However, it is still a difficult calculation because the Boolean complement of G_2 frequently fails.

$$G_1/G_2 = DTA(G_1, G_2) \cdot BC(G_2) \tag{6}$$

$$\begin{aligned} DTA(G_1, G_2) &= ab + be \\ BC(bc + bd) &= /b + /c/d \\ DTA(G_1, G_2) \cdot BC(G_2) &= (ab + be)(/b + /c/d) = ab/c/d + b/c/de \end{aligned} \tag{7}$$

3.2 Approximate solution by combining DTA and BC with truncation

To minimize the CDF overestimation by DTA and overcome the difficulty in accurate solution calculation, the use of the Boolean complement of truncated G_2 is proposed in this study. As shown in Eqs. (8) and (9), $BC(Truncated\ G_2)$ is employed instead of $BC(G_2)$.

$$G_1/G_2 \cong DTA(G_1, G_2) \cdot BC(Truncated\ G_2) \tag{8}$$

$$\begin{aligned} DTA(G_1, G_2) &= ab + be \\ Truncated\ G_2 &= bc \text{ when } p(bc) > p(bd) \\ BC(Truncated\ G_2) &= /(bc) = /b + /c \\ DTA(G_1, G_2) \cdot BC(Truncated\ G_2) &= (ab + be)(/b + /c) = ab/c + b/ce \end{aligned} \tag{9}$$

The inequalities of the probabilities of Eqs. (2)–(9) can be expressed as Eq. (10). The probability of $DTA(G_1, G_2) \cdot BC(Truncated\ G_2)$ is very close to that of G_1/G_2.

$$p(G_1/G_2) \le p\big(DTA(G_1, G_2) \cdot BC(Truncated\ G_2)\big) \le p\big(DTA(G_1, G_2)\big) \tag{10}$$

4. ZTDD ALGORITHM

The new ZTDD algorithm, which was developed recently [1,2], can optionally generate the accurate or approximate solution from G_1/G_2 by DTA. The proposed method described in Section 3.2 was implemented into a new tool ZEBRA that is based on the ZTDD algorithm as described below. Note that the MCSs of G_2 are truncated with an elevated truncation limit right before the Boolean complement of G_2 to avoid the calculation difficulty of the Boolean complement of G_2.

(Step 1) Generate the MCSs of G_1 and G_2 with a given truncation limit such as 1.0×10^{-11}.
(Step 2) Perform the DTA as $DTA(G_1, G_2)$ with the MCSs generated in Step 1.
(Step 3) Truncate the MCSs of G_2 with an elevated truncation limit such as 1.0×10^{-6}.
(Step 4) Perform the Boolean complement as $BC(Truncated\ G_2)$.
(Step 5) Combine two Boolean solutions as $DTA(G_1, G_2) \cdot BC(Truncated\ G_2)$.

4.1 ZTDD structure

ZTDD is newly defined for encoding the factorized MCSs or prime implicants (PIs) that have complemented basic events. ZTDD has a Boolean structure that comprises recursively connected if-then-else connectives (ITEs) that have three terms of L, R, and N as given by Eq. (11). ZTDD encodes the Boolean equation $xL + /xR + N$ into three Boolean equations as xL, $/xR$, and N, where L, R, and N are child ZTDDs. The ZTDD can be interpreted as a factorized form of MCSs or PIs.

$$xL + /xR + N = < x, L, R, N > \tag{11}$$

The ZTDD in Eq. (11) can be encoded into BDD using Eq. (12) or converted into two connected ZBDDs by employing Eq. (13). Clearly, the ZTDD in Eq. (11) is much more intuitive and simpler than the BDD and ZBDD in Eqs. (12) and (13), respectively.

$$xL + /xR + N = xL + /xR + (x + /x)N = < x, L + N, R + N >_{BDD} \tag{12}$$
$$xL + /xR + N = < x, L, < /x, R, N >>_{ZBDD} \tag{13}$$

4.2 ZTDD algorithm

To solve a fault tree in a bottom-up way, two ZTDDs need to be combined in a logical manner. In this study, a set of ZTDD formulae was developed for combining two ZTDDs, as given by Eq. (14). If x and y are two variables with a given variable ordering $x < y$, x is located at a higher position in ZTDD than y. Thereafter, the ZTDD combining operation with $G(x) = < x, L_1, R_1, N_1 >$ and $H(y) = < y, L_2, R_2, N_2 >$ is recursively performed from top to bottom ITEs following Eq. (14). Thus, a coherent or noncoherent fault tree can be solved in a bottom-up way using Eq. (14).

$$\begin{gathered} G(x) \cdot H(x) = < x, (L_1L_2 + L_1N_2 + N_1L_2), (R_1R_2 + R_1N_2 + N_1R_2), N_1N_2 > \\ G(x) + H(x) = < x, (L_1 + L_2), (R_1 + R_2), (N_1 + N_2) > \\ G(x) \cdot H(y) = < x, L_1H, R_1H, N_1H > \\ G(x) + H(y) = < x, L_1, R_1, (N_1 + H) > \end{gathered} \tag{14}$$

4.3 ZTDD for DTA

When a fault tree is solved in a bottom-up way using Eq. (14), nonminimal solutions (subsets) are introduced in ZTDD, and they need to be deleted. These nonminimal solutions exist in the L and R of $< \alpha, L, R, N >$ because ZTDD is $\alpha L + /\alpha R + N$. The subsets in L and R are deleted by the $Subsume(L, N)$ and $Subsume(R, N)$ operations in Eq. (15) if their minimal solutions (supersets) exist in N. The term $L_1 \backslash (L_2$ or $N_2)$ signifies that each solution in L_1 is tested and deleted if L_2 or N_2 has a superset.

$$Subsume(G,H) = G\backslash H = \begin{cases} G\backslash N_2 & ,x > y \\ < x, L_1\backslash H, R_1\backslash H, N_1\backslash H > & ,x < y \\ < x, L_1\backslash (L_2 \text{ or } N_2), R_1\backslash (R_2 \text{ or } N_2), N_1\backslash N_2 > & ,x = y \end{cases} \quad (15)$$
$$G(x) =< x, L_1, R_1, N_1 > = xL_1 +/xR_1 + N_1$$
$$H(y) =< y, L_2, R_2, N_2 >= yL_2 +/yR_2 + N_2$$

To calculate the approximate solutions of G_1/G_2, DTA is employed. It is accomplished via the subsuming operation given by Eq. (16).

$$G_1/G_2 \approx Delterm(G_1, G_2) = Subsume(G_1, G_2) \quad (16)$$

4.4 ZTDD for Boolean complement

Various Boolean complements of ZTDD are listed in Table 1. Here, the Boolean complements $/(xL) =/x + x/L$ and $/(/xR) = x +/x/R$ are applied instead of $/(xL) =/x +/L$ and $/(/xR) = x +/R$ to maintain disjoint solutions as much as possible. To calculate the accurate solutions of G_1/G_2, the ZTDD of G_2 is complemented into $/G_2$ using the Boolean complements in Table 1 and the two ZTDDs of G_1 and $/G_2$ are combined (see Section 3.2).

Table 1. Boolean complements of ZTDDs

Boolean complement	Derivation
$/< x, L, R, N >=< x,/L/N,/R/N, 0 >$	$/(xL +/xR + N) = (/x + x/L)(x +/x/R)/N = x/L/N +/x/R/N$
$/< x, L, 0, N >=< x, 0,/N,/L/N >$	$x/L/N +/x/N = x/L/N +/x(/N +/L/N) =/x/N +/L/N$
$/< x, 0, R, N >=< x,/N, 0,/L/R >$	$x/N +/x/R/N = x(/N +/R/N) +/x/R/N = x/N +/L/R$
$/< x, L, 0, 0 >=< x, 0,1,/L >$	$x/L +/x = x/L +/x(1 +/L) =/x +/L$
$/< x, 0, R, 0 >=< x, 1,0,/R >$	$x +/x/R = x(1 +/R) +/x/R = x +/R$

4.5 ZTDD probability calculation

First, the sum of PI probabilities is calculated by recursively calculating the probability given by Eq. (17) from the bottom to the top of the ZTDD.

$$p(f) = p_x \times p(L) + (1 - p_x) \times p(R) + p(N) \quad (17)$$

Second, the min cut upper bound (MCUB) probability of PIs can be optionally calculated by navigating all minimal solutions in the ZTDD. Finally, the exact probability can be calculated from BDD by converting the ZTDD into BDD if necessary.

5. BENCHMARK TESTS

The CDF calculation practice in PSA is depicted in Fig. 1, and the advanced CDF calculation method [18] for seismic PSA is explained in Fig. 2. This advanced CDF calculation method in Fig. 2 was developed in order to avoid CDF overestimation. However, although MCSs are converted into a BDD and more accurate probability is calculated by using the BDD, the calculated CDF is still overestimated when MCSs for G_1/G_2 are calculated by DTA.

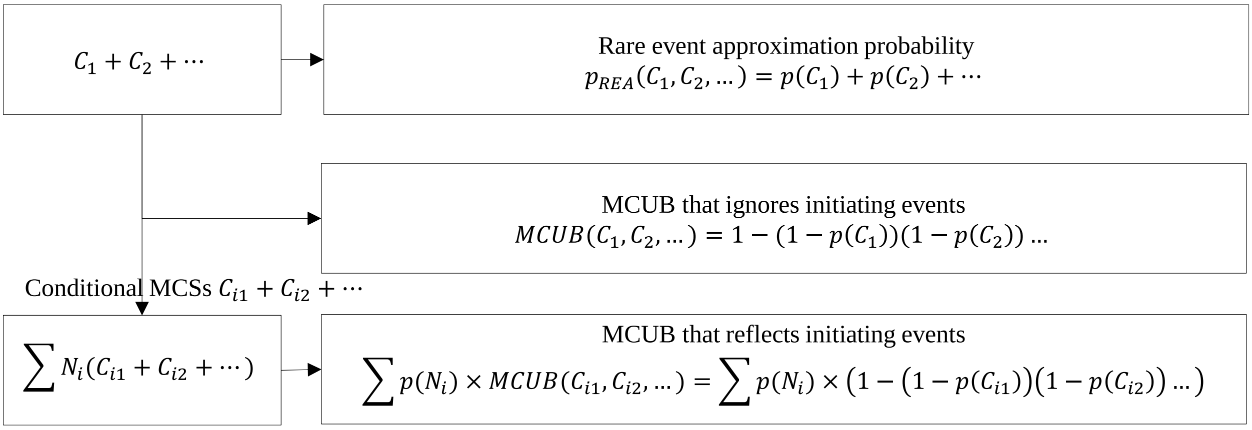

Fig 1. Traditional CDF calculation in PSA[18]

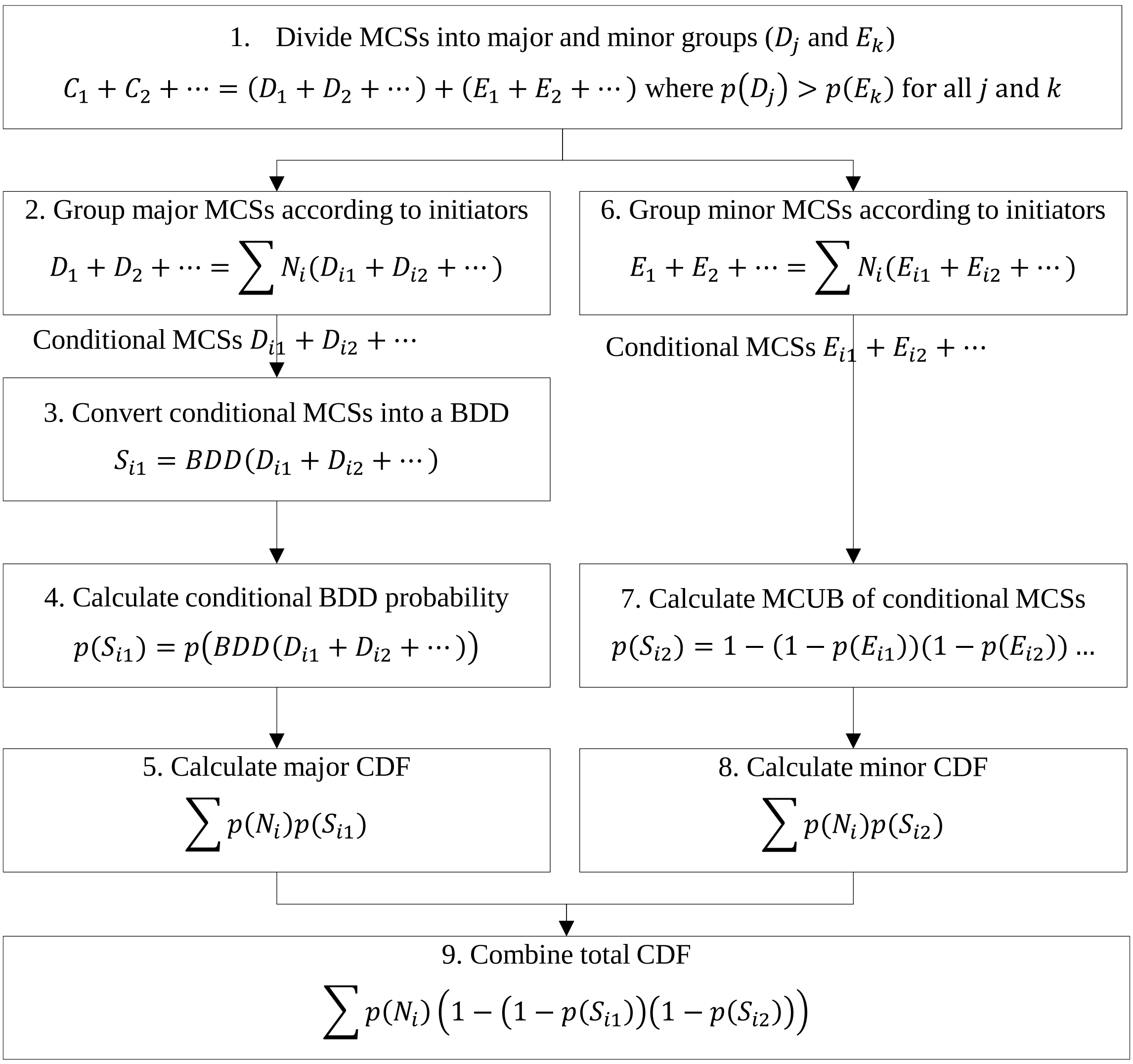

Fig 2. Advanced CDF calculation in seismic PSA[18]

To demonstrate the strength of the proposed method, benchmark tests were performed using a seismic multi-unit PSA model in Table 2, and the calculation results are summarized in Tables 3 and 4.

Table 2. Seismic multi-unit PSA model

Number of events	18,094
Number of complemented events	0
Number of gates	67,131
Number of complemented gates	195
Number of initiators	1

Table 3. Calculation 1 (Truncation limit = 1.0×10^{-11})

See Figs. 1 and 2	$DTA(G_1, G_2)$	$DTA(G_1, G_2) \cdot BC(Truncated\ G_2$ (a))
MCSs	18,783	15,658
$\sum p(N_i)\left(1-(1-p(S_{i1}))(1-p(S_{i2}))\right)$	4.518E-06	2.494E-06
$\sum p(N_i) \times p(S_{i1})$	4.097E-06 (b)	2.494E-06 (c)
$\sum p(N_i) \times p(S_{i2})$	5.353E-07	0.000E+00
$\sum p(N_i) \times MCUB(C_{i1}, C_{i2}, \ldots)$	8.759E-06	5.695E-06
$MCUB(C_1, C_2, \ldots)$	1.167E-05	6.749E-06

(a) Truncation limit = 1.0×10^{-5} is applied to G_2, and $BC(Truncated\ G_2)$ was calculated.
(b) Top 3,000 MCSs are converted into a BDD.
(c) Top 20,000 MCSs are converted into a BDD.

Table 4. Calculation 2 (Truncation limit = 1.0×10^{-12} for G1 and G2)

See Figs. 1 and 2	$DTA(G_1, G_2)$	$DTA(G_1, G_2) \cdot BC(Truncated\ G_2$ (a))
MCSs	85,133	72,754
$\sum p(N_i)\left(1-(1-p(S_{i1}))(1-p(S_{i2}))\right)$	4.667E-06	2.611E-06
$\sum p(N_i)p(S_{i1})$	4.097E-06 (b)	2.497E-06 (c)
$\sum p(N_i)p(S_{i2})$	7.244E-07	1.316E-07
$\sum p(N_i) \times MCUB(C_{i1}, C_{i2}, \ldots)$	8.865E-06	5.813E-06
$MCUB(C_1, C_2, \ldots)$	1.187E-05	6.917E-06

(a) Truncation limit = 1.0×10^{-5} is applied to G_2, and $BC(Truncated\ G_2)$ was calculated.
(b) Top 3,000 MCSs are converted into a BDD.
(c) Top 20,000 MCSs are converted into a BDD.

6. CONCLUSIONS

This study introduces a hybrid method of the accurate and approximate solutions for obtaining more accurate solution for noncoherent accident sequence logics, which was implemented into a new tool ZEBRA that is based on the ZTDD algorithm. To validate the strength of the proposed method, benchmark tests were performed with a seismic multi-unit PSA model. The conclusions of this study can be summarized as follows:

1. The Boolean complement of G_2 should be performed for calculating the accurate solution of G_1/G_2 by $G_1 \cdot BC(G_2)$. However, as the size of the Boolean complement of G_2 is always very large, the accurate solution of G_1/G_2 cannot be produced owing to the computational calculation difficulty.
2. In actual PSAs, instead of the accurate solution of G_1/G_2, the approximate solution of G_1/G_2 is calculated by $DTA(G_1, G_2)$ to avoid the calculation difficulty of the accurate solution. This $DTA(G_1, G_2)$ calculation is employed in all PSAs. However, $DTA(G_1, G_2)$ leads to CDF overestimation, and this overestimation is proportional to the number of nonrare events and complemented gates.
3. In actual PSA, to minimize this CDF overestimation, $DTA(G_1, G_2)$ is converted into a BDD, and then CDF is calculated using this BDD. However, even if $DTA(G_1, G_2)$ is converted into a BDD, there is a limitation in minimizing CDF overestimation because the complemented terms from $BC(G_2)$ do not exist in $DTA(G_1, G_2)$.
4. In this study, a hybrid method of the accurate and approximate solutions of G_1/G_2 is proposed to minimize CDF overestimation and calculate more accurate CDF.
5. The efficiency and strength of the proposed method are validated by benchmark tests. The implementation and use of $DTA(G_1, G_2) \cdot BC(Truncated\ G_2)$ are strongly recommended.

Acknowledgements

This work was supported by the Korea Foundation Of Nuclear Safety (KOFONS) grant funded by the Nuclear Safety and Security Commission (NSSC), Republic of Korea (No. RS-2022-KN067010 and RS-2021-KN050610).

References

[1] W.S. Jung, A New Zero-suppressed Ternary Decision Diagram Algorithm, Probabilistic Safety Assessment and Management (PSAM) Topical, October 23-25, 2023, Virtual meeting.
[2] W.S. Jung, Zero-suppressed ternary decision diagram algorithm for solving noncoherent fault trees in probabilistic safety assessment of nuclear power plants, Nuclear Engineering and Technology, 2024. **56**: p.2092-2098.
[3] C.Y. Lee, Representation of switching circuits by binary-decision programs, Bell System Technical Journal, 1959. **38**: p.985-999.
[4] B. Akers, Binary decision diagrams. IEEE Transactions on Computers, 1978. **C-27 (6**): p.509-516.
[5] R. Bryant, Graph-based algorithms for Boolean function manipulation, IEEE Transactions on Computers, 1986. **C-35 (8)**: p.677-691.
[6] R. Bryant, Symbolic Boolean manipulation with ordered binary decision diagrams, ACM Computing Surveys, 1992. **24**: p.293-318.
[7] A. Rauzy, New algorithms for fault trees analysis, Reliability Engineering and System Safety, 1993. **5(59)**: p.203-211.
[8] Y. Duituit, A. Rauzy, Efficient algorithms to assess component and gate importance in fault tree analysis, Reliability Engineering and System Safety, 2001. **72**: p.213-22.
[9] A. Rauzy, BDD for Reliability Studies. In Handbook of Performability Engineering, Ed. K.B. Misra, Elsevier, Amsterdam, The Netherlands, 2008. p.381-396.
[10] R. Remenyte-Prescott, J. Andrews, An enhanced component connection method BDD, Reliability Engineering and System Safety, 2008. **93**: p.1543-1550.
[11] O. Nusbaumer, A. Rauzy, Fault tree linking versus event tree linking approaches - a reasoned comparison, Journal of Risk and Reliability, 2013. **227**: p.315-326.

[12] A. Rauzy, L. Yang, Decision diagram algorithms to extract minimal cutsets of finite degradation models, Information, 2019. **10**: p.368.

[13] W.S. Jung, S.H. Han, J.E. Yang, Fast BDD truncation method for efficient top event probability calculation, Nuclear Engineering and Technology, 2008. **40(7)**: p.571-580.

[14] W.E. Vesely, F.F. Goldberg, N.H. Roberts, D.F. Haasl, Fault Tree Handbook, U.S. Nuclear Regulatory Commission, NUREG-0492, 1981.

[15] S. Minato, Zero-suppressed BDDs for set manipulation in combinatorial problems, Proceedings of the 30th International Conference on Design Automation, 1993. p.272-277.

[16] W.S. Jung, S.H. Han, J.J. Ha, A fast BDD algorithm for the reliability analysis of large coherent systems, Reliability Engineering and System Safety, 2004. **83**: p.369-374.

[17] W.S. Jung, ZBDD algorithm features for an efficient probabilistic safety assessment, Nuclear Engineering and Design, 2009. **239**: p.2085-2092.

[18] W.S. Jung, A method to improve cutset probability calculation in probabilistic safety assessment of nuclear power plants, Reliability Engineering and System Safety, 2015, **134**: p.134-142.