

FFTA: A FAST FAULT TREE ANALYSIS PROGRAM

Takeshi MATSUOKA

Nuclear Ship Division, Ship Research Institute, 6-38-1, Shinkawa, Mitaka, Tokyo, Japan

Received 19 December 1984

This paper presents a computer program for fault tree analysis. The algorithm is a bottom-up algorithm and the detailed procedure of the program is explained. The analyses are performed for six sample fault trees and the results are compared with those obtained by the BAM and CUT codes. This program has a ten to thousand times faster calculation speed compared with the BAM and CUT codes.

1. Introduction

The Reactor Safety Study [1] was the first detailed application of probabilistic risk assessment (PRA) to nuclear power plants. Since the Reactor Safety Study, the use of PRA techniques has become more widespread in the nuclear community [2,3].

The heart of PRA is contained in logic models known as an event tree and a fault tree, which are widely accepted as valid analytical tools. Fault tree analyses of a complex system require the use of a computer because of the enormous amount of information that must be processed. Many computer programs [4] are presently being used for a fault tree analysis. Each program has some particular objective, such as qualitative evaluation, quantitative evaluation, uncertainty analysis, and the identification or analysis of dependent failure.

The author has developed a computer program [5] which performs a reliability analysis under extreme environmental conditions. The program requires the calculation of the point probabilities of top events of various fault trees for a large number of times. This is the motivation for developing a computer program "FFTA" (a Fast Fault Tree Analysis program), which can obtain the point probability of the top event and the minimal cut sets for a fault tree with little expenditure of computer time.

The algorithm of the program FFTA will be explained and the computer time required by this program will be compared with those required by the FTA-J-BAM and FTA-J-CUT codes [6].

2. Fault tree and its elements

Fault tree analysis is one method of reliability analyses and is generally applicable to complex, dynamic systems which include nuclear reactor systems.

The fault tree analysis process begins with the definition of an undesired event or failed state of a system. This undesired event becomes the top event on the fault tree. A development of the logic diagram depicts basic events and undeveloped events that can, through AND, OR and other gates, produce the top event. The resulting arrangement is a tree-like structure with information flows from the branch tips to the top events, so it is called a "fault tree" (fig. 1).

The basic event is an actual component failure for which failure rate data are expected to be available. An undeveloped event is a specific fault event that is not further developed, either because the event is of insuffi-

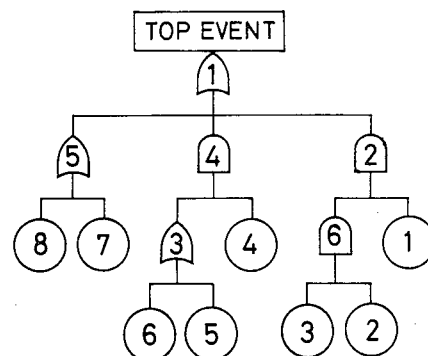


Fig. 1. Sample fault tree.

cient consequence or because relevant information is not available.

A primary event is a basic event or an undeveloped event which is located at the branch tip of a fault tree. An intermediate event is a fault event that occurs because of one or more antecedent events acting through logic gates.

The AND gate is equivalent to the Boolean symbol “ \cdot ” representing the intersection of the events attached to the gate. All of the events input to the AND gate must occur in order for the event above the gate to occur. The OR gate is equivalent to the symbol “ $+$ ” representing the union of events. Any one or more of the events input to the gate must occur in order for the event above the gate to occur. The NOT gate is equivalent to the symbol “ $-$ ” representing the complement. The output of the gate is a complement of the event input to the gate. Other gates, for example NOR, ANOT, can be defined in terms of the AND, OR, and NOT gates.

A replicated event is a primary event which occurs as an input of more than one gate in a fault tree. That is, a replicated event contributes more than one intermediate events. A non-replicated event is one which occurs only once in a fault tree.

A cut set is a collection of events such that if all these events occur, then top event on a fault tree occurs. A minimal cut set is the smallest set of events which must all occur in order for the top event to occur.

3. The algorithm

The FFTA algorithm is a bottom-up algorithm, starting from inputs to the lowest-level gate and working up to the top event.

For a fault tree which contains no replicated events, a numerical calculation is performed at each gate and the result is transferred to the input to the upper gate. Finally the point probability of the top event is obtained.

If there is a replicated event in a fault tree, the summation between the cut sets which contain one or more of replicated events is performed at the final stage where the point probability of the top event is calculated.

If minimal cut sets are required, all the primary events are labeled and the summation between all the cut sets are performed at the final stage.

The following two sections describe the detailed procedures of obtaining the top event probability of a fault tree which has replicated events.

4. Procedure at each gate

At each gate, the Boolean identities and laws are applied to replicated events.

Here, we represent the replicated events by A , B , and non-replicated events by X , Y . At the OR gate the following identifications are performed.

$$A + A = A, \quad (1)$$

$$A + A \cdot X = A. \quad (2)$$

At AND gate, the followings are performed.

$$A \cdot A = A, \quad (3)$$

$$A \cdot \bar{A} = 0, \quad (4)$$

$$(A \cdot X) \cdot (A \cdot Y) = A \cdot X \cdot Y = A \cdot Z, \quad (5)$$

$$(A + B) \cdot X = A \cdot X + B \cdot X = A \cdot C + B \cdot C, \quad (6)$$

where, \bar{A} is the complement of A ; Z is an intermediate event which is an intersection between non-replicated events; X is renamed as C and hereafter it is treated like a replicated event.

Calculation between non-replicated events is performed at each gate, and the resultant intermediate event and its probability are transferred to input to the upper gate.

An output of a gate is composed of an intermediate event obtained from non-replicated events, and of the terms which contain one or more of the replicated events.

In addition to the above procedures, the elimination is carried out of the term whose order (number of events in the term) is larger than a preselected number, or whose probability is less than a cut off value (COV) specified by the user.

By these procedures, the term number of output of one gate is reduced to as few as possible.

5. Probability of top event

By iterating the procedures shown in the previous section, we obtain the top event in the following Boolean form.

$$\text{TOP} = C_0 + \sum_{i=1}^I C_i, \quad (7)$$

$$C_i = X_i \cdot \prod_j A_{ij} (i > 0), \quad (8)$$

where, A_{ij} is a replicated primary event, C_0 and X_i are terms which consist of only non-replicated events.

The point probability of the top event: $P(\text{TOP})$, is expressed by the following equation.

$$P(\text{TOP}) = P(C_0) + \sum_{i=1}^I P(C_i) - \sum_{\substack{i,j=0 \\ i < j}}^I P(C_i)P(C_j)/f_{ij} + \sum_{\substack{i,j,k=0 \\ i < j < k}}^I P(C_i)P(C_j)P(C_k)/f_{ijk} + 0(h), \quad (9)$$

where, $P(C_i)$ is occurrence probability of C_i , $0(h)$ expresses higher order terms. The f_{ij} and f_{ijk} are factors defined by the following equation.

$$f_{ij}, f_{ijk} = \prod_l P(A_l)^{m-1}, \quad (10)$$

where, A_l is a replicated primary event contained in C_i , C_j and C_k and m is the number of times that the A_l appears in C_i , C_j and C_k .

In order to obtain the top event probability, it is necessary to perform the calculation shown by eq. (9). As the number of cut sets increases, the possible combination between the cut sets becomes enormously large. Therefore, in the FFTA, the calculation is performed in the following approximate method.

Suppose the summation from the first to $(m-1)$ th term is obtained as $P_{m-1}(\text{TOP})$. In this case, $P_m(\text{TOP})$ is calculated by eq. (11) to the first approximation.

$$P_m(\text{TOP}) = P_{m-1}(\text{TOP}) + P(C_m) - P_{m-1}(\text{TOP}) \cdot P(C_m). \quad (11)$$

Substitute eq. (9) into eq. (11), then,

$$P_m(\text{TOP}) = P(C_0) + \sum_{i=1}^m P(C_i) - \sum_{\substack{i,j=0 \\ i < j}}^{m-1} P(C_i)P(C_j)/f_{ij} - P(C_m) \cdot \sum_{i=0}^{m-1} P(C_i) + \sum_{\substack{i,j,k=0 \\ i < j < k}}^{m-1} P(C_i)P(C_j)P(C_k)/f_{ijk} + P(C_m) \cdot \sum_{\substack{i,j=0 \\ i < j}}^{m-1} P(C_i)P(C_j)/f_{ij} + 0(h). \quad (12)$$

From eq. (12), it becomes apparent that only the first

order terms of $P(C_i)$ have the exact form. Then, to the 2nd approximation, add the following terms to the right side of eq. (11).

$$P(C_m) \cdot \sum_{i=0}^{m-1} P(C_i) - P(C_m) \cdot \sum_{i=0}^{m-1} P(C_i)/f_{im}.$$

$P_m(\text{TOP})$ becomes,

$$P_m(\text{TOP}) = P(C_0) + \sum_{i=1}^m P(C_i) - \sum_{\substack{i,j=0 \\ i < j}}^m P(C_i)P(C_j)/f_{ij} + \sum_{\substack{i,j,k=0 \\ i < j < k}}^{m-1} P(C_i)P(C_j)P(C_k)/f_{ijk} + P(C_m) \cdot \sum_{\substack{i,j=0 \\ i < j}}^{m-1} P(C_i)P(C_j)/f_{ij} + 0(h). \quad (13)$$

In eq. (13), the first and second order terms of $P(C_i)$ have exact form. In most cases, eq. (13) is a good approximation for obtaining the accurate value of the top event probability. In the FFTA program, the third approximation is given as an option; this is to add the following terms to the right side of the eq. (13).

$$-P(C_m) \cdot \sum_{\substack{i,j=0 \\ i < j}}^{m-1} P(C_i)P(C_j)\{1/f_{ij} - 1/f_{ijm}\},$$

where i, j are restricted only to those which satisfy the next conditions, in order to avoid performing so many calculations.

$$P(C_m)P(C_i)/f_{mi} > \text{COV},$$

$$P(C_m)P(C_j)/f_{mj} > \text{COV}.$$

If there are no replicated events in a fault tree, eq. (7) becomes,

$$\text{TOP} = C_0, \quad (14)$$

and the top event probability is obtained as follows,

$$P(\text{TOP}) = P(C_0). \quad (15)$$

In obtaining the minimal cut sets, all the primary events are treated like a replicated event. The calculation procedures are basically the same as those shown in the above. The terms C_0, X_i vanish from the above equations, and C_i becomes a minimal cut set.

6. Program description

6.1. Program overview

This program is written in Fortran 77 for the FACOM 180 II-AD computer (compatible to IBM 360/370). Current program dimensions allow the code to load with under 2048K bites of memory.

The point probability of the top event is calculated for a given fault tree. If it is requested, the minimal cut sets can be obtained. In this case, the cut sets are obtained for all the intermediate gates.

This program is capable of evaluating fault trees with up to 500 primary events and 500 gates, including AND, OR, NOT, ANOT, NAND, and ONOT gates. Up to 2800 minimal cut sets can be found, and the maximum order of minimal cut sets is specified by the user.

6.2. Practical process in the FFTA program

The required input consists of control information and a description of the fault tree: a fault tree structure, occurrence probabilities of primary events. Optional input includes ten-character alphanumeric names for the primary events in a fault tree.

The NOT gate is treated by replacing the input event by its complement and removing the NOT gate. In this case, the complement is labeled so as to perform relation (4). The gates NOR, ANOT, NAND, ONOT are treated by expressing in terms of AND, OR, and NOT gates.

Integers are assigned to gates and primary events and they are identified by these numbers. A structure of a fault tree is expressed by assigning a series of integers which represent the gates linking the top event to a primary event. For example, the structure of a fault tree shown in fig. 1 is expressed as shown in fig. 2. The numbers in the first to third columns are gate numbers and numbers in the fourth column are primary event numbers. The gate number 0 indicates that there are no gates at this position. AND gates are expressed by even numbers and OR gates are expressed by odd numbers.

1	2	0	1
1	2	6	2
1	2	6	3
1	4	0	4
1	4	3	5
1	4	3	6
1	5	0	7
1	5	0	8

Fig. 2. Expression of the structure of sample fault tree.

The order of the row corresponds to the configuration of a fault tree.

Any structure of fault tree can be expressed in this form and from this form the structure of the fault tree is easily figured out. Further, this expression is well adapted to the calculation procedure in this program.

To illustrate the steps performed in the FFTA program, examples are given using the sample fault tree shown in fig. 1.

Fig. 3 shows the steps in order to obtain the point probability of the top event. The sample fault tree has no replicated events, then the steps become rather simple.

In the first step, read the first line of input data, that is, the gate numbers, primary event number, and its probability (1, 2, 0, 1, 1.0×10^{-2}). From the gate numbers, it is found out that the event "1" is an input to gate "2" which is placed on the second level from the top event. Then, in the second step, the probability of event "1" is stored in the variable $P(2, 1)$ which stores the input value to the second level gate. In the third step, read the next line of input data. The primary event "2" is an input to gate "6" which is placed on the third level. Then, in the 4th step, the probability of event "2" is stored in $P(3, 1)$. In the 5th step, read the next input data. The event "3" is also an input to gate "6", and the gate "6" is an AND gate. In the 6th step, calculate the product of $P(3, 1)$ and the probability of event "3". The result is restored in $P(3, 1)$. In the 7th step, read the next input data. The primary event "4" is an input to gate "4". The gate "4" is located on a second level, and it is found out that the event "4" does not belong to any branch of gate "2", that is, all the input information to gate "2" has been already given, and the output of gate "2" can be obtained. Then in the 8th step, calculate the product of $P(2, 1)$ and $P(3, 1)$, and restore the results in $P(2, 1)$. In the 9th step, the value of $P(2, 1)$ is transferred to $P(1, 1)$ which stores the input value to the first level gate. In the 10th step, the probability of event "4" is stored in $P(2, 1)$. Iterate the process until the 21st step where the point probability of the top event is obtained as 4.569×10^{-4} .

Fig. 4 shows the steps in order to obtain minimal cut sets. The steps will parallel those shown in fig. 3 with memorizing the primary event numbers along with the probabilities of cut sets.

The first step is to read the first line of input data. In the 2nd step, the probability of event "1" is stored in $P(2, 1)$, and event number "1" is stored in $N(2, 1, 1)$. Variables $N(2, 2, 1)$ and $N(2, 3, 1)$ have an initial value "0" which does not mean any primary event. In the 4th step, the probability of event "2" and event number "2"

Process Steps	Inputs					Variables		
	NG(1)	NG(2)	NG(3)	NE	PP	P(3,1)	P(2,1)	P(1,1)
1	1	2	0	1	1.0×10^{-2}			
2							1.0×10^{-2}	
3	1	2	6	2	3.0×10^{-2}		1.0×10^{-2}	
4						3.0×10^{-2}	1.0×10^{-2}	
5	1	2	6	3	6.0×10^{-2}		1.0×10^{-2}	
6						AND 1.8×10^{-3}	1.0×10^{-2}	
7	1	4	0	4	3.0×10^{-3}		AND 1.8×10^{-5}	
8							1.8×10^{-5}	
9								1.8×10^{-5}
10								1.8×10^{-5}
11	1	4	3	5	1.0×10^{-2}		3.0×10^{-3}	
12						1.0×10^{-2}	3.0×10^{-3}	
13	1	4	3	6	3.0×10^{-3}		3.0×10^{-3}	
14						OR 1.297×10^{-2}	3.0×10^{-3}	
15	1	5	0	7	1.0×10^{-4}		AND 3.891×10^{-5}	
16							3.891×10^{-5}	
17								OR 1.8×10^{-5}
18								5.691×10^{-5}
19	1	5	0	8	3.0×10^{-4}		OR 1.0×10^{-4}	
20							4.0×10^{-4}	
21								OR 4.569×10^{-4}

Fig. 3. Process steps of obtaining a top event probability.

are stored in $P(3, 1)$ and $N(3, 1, 1)$, respectively. In the 6th step, the product of $P(3, 1)$ and the probability of event "3" is restored in $P(3, 1)$. Event number "3" is stored in $N(3, 2, 1)$. The values of $N(3, J, 1)$ express that the $P(3, 1)$ stores a product of probabilities of events "2" and "3". In the 8th step, a product of the probabilities of events "1", "2" and "3", is stored in $P(2, 1)$. In the 14th step, the probability of event "6" is stored in $P(3, 2)$ and the event number "6" is stored in $N(3, 1, 2)$. In this case, the values of $N(3, J, K)$ and $P(3, K)$ express that the inputs to third level gate are events "5" and "6" and their probabilities are 1.0×10^{-2} and 3.0×10^{-3} , respectively. In the 16th step, the product of the probabilities of events "4" and "5" is stored in $P(2, 1)$ and the product of the probabilities of events "4" and "6" is stored in $P(2, 2)$. They are the output of the second level gate. In the 17th step, the above values are transferred to variables $P(1, 2)$ and $P(1, 3)$. The primary event numbers "4", "5" and "4", "6" are also stored in $N(1, 1, 2)$, $N(1, 2, 2)$ and in $N(1, 1, 3)$, $N(1, 2, 3)$, respectively.

By continuing the process, we can obtain the inputs

to top event in the 21st step. From the values of $N(I, J, K)$ in the 21st step, minimal cut sets (M.C.S.) are determined as follows,

$$\text{M.C.S.} = \{ X_1, X_2, X_3 \}, \{ X_4, X_5 \}, \{ X_4, X_6 \}, \{ X_7 \}, \{ X_8 \}. \tag{16}$$

The probabilities of these minimal cut sets are stored in the variables from $P(1, 1)$ to $P(1, 5)$. The point probability of the top event is calculated by eq. (13) as 4.569×10^{-4} .

Information on C_i and $P(C_i)$ was stored in program variables $N(I, J, K)$, $P(I, K)$ as shown in the 21st step in fig. 4. The primary event numbers which comprise the term C_i were stored in the variable $N(I, J, K)$.

The output includes the structure of a fault tree, alphanumeric names and assigned numbers for primary events, occurrence probabilities of primary events, the point probability of the top event, and if requested, a list of minimal cut sets and their occurrence probabilities.

Process Steps	Inputs				Variables			
	NG(1)	NG(2)	NG(3)	NE, PP	N(3,J,K),P(3,K)	N(2,J,K),P(2,k)	N(1,J,K),P(1,K)	
1	1,	2,	0,	1, 1.0x10 ⁻²				
2						1,0,0, 1.0x10 ⁻²		
3	1,	2,	6,	2, 3.0x10 ⁻²				
4						1,0,0, 1.0x10 ⁻²		
5	1,	2,	6,	3, 6.0x10 ⁻²				
6					AND ↓			
7	1,	4,	0,	4, 3.0x10 ⁻³	2,3,0, 1.8x10 ⁻³	1,0,0, 1.0x10 ⁻²		
8						AND ↓		
9						1,2,3, 1.8x10 ⁻⁵		
10							1,2,3, 1.8x10 ⁻⁵	
11	1,	4,	3,	5, 1.0x10 ⁻²				
12					5,0,0, 1.0x10 ⁻²	4,0,0, 3.0x10 ⁻³	1,2,3, 1.8x10 ⁻⁵	
13	1,	4,	3,	6, 3.0x10 ⁻³				
14					5,0,0, 1.0x10 ⁻² 6,0,0, 3.0x10 ⁻³	4,0,0, 3.0x10 ⁻³	1,2,3, 1.8x10 ⁻⁵	
15	1,	5,	0,	7, 1.0x10 ⁻⁴		AND ↓		
16						4,5,0, 3.0x10 ⁻⁵ 4,6,0, 9.0x10 ⁻⁶	1,2,3, 1.8x10 ⁻⁵	
17							1,2,3, 1.8x10 ⁻⁵ 4,5,0, 3.0x10 ⁻⁵ 4,6,0, 9.0x10 ⁻⁶	
18						7,0,0, 1.0x10 ⁻⁴	1,2,3, 1.8x10 ⁻⁵ 4,5,0, 3.0x10 ⁻⁵ 4,6,0, 9.0x10 ⁻⁶	
19	1,	5,	0,	8, 3.0x10 ⁻⁴				
20						7,0,0, 1.0x10 ⁻⁴ 8,0,0, 3.0x10 ⁻⁴	1,2,3, 1.8x10 ⁻⁵ 4,5,0, 3.0x10 ⁻⁵ 4,6,0, 9.0x10 ⁻⁶	
21							1,2,3, 1.8x10 ⁻⁵ 4,5,0, 3.0x10 ⁻⁵ 4,6,0, 9.0x10 ⁻⁶ 7,0,0, 1.0x10 ⁻⁴ 8,0,0, 3.0x10 ⁻⁴	

Fig. 4. Process steps of obtaining minimal cut sets.

7. Comparison of FFTA and WAM-BAM, WAM-CUT

7.1. Fault trees

Six sample fault trees were chosen for the analysis. Fault trees 1 to 4 are taken from WASH-1400 [7]. They are, respectively, TEST 1, CSIS (Containment Spray Injection System), CLCS (Consequence Limiting Control System), SHAS (Sodium Hydroxide Addition System) trees. Fault tree 5 is the ISCCS (Initiation Signal Circuit for Core Spray) tree analyzed by Watanabe et al. [8]. Fault tree 6 is a complex fault tree given in the report NUREG/CR-0465 [9].

These trees are summarized in table 1. They are small to medium size trees.

7.2. Analysis conditions and results

The fault trees shown in table 1 were analyzed by the FFTA program. The analysis was also performed by the FTA-J-BAM and FTA-J-CUT codes; basically they are same as the WAM-BAM [10] and WAM-CUT [11] codes, which have been used in several PRA studies.

In ref. [9], the probabilities of primary events were not given, so in the present analysis, arbitrary values were assigned to the primary events in fault tree 6.

The point probability of the top event was calculated by eq. (13), i.e., to the 2nd approximation. Minimal cut sets were obtained up to order 5, and the cut off value (COV) at each gate was given as 10⁻⁶ times the top event probability.

Table 1
Fault trees

Tree number	Tree name	Number of primary events	Number of OR gates	Number of AND gates	Number of replicated events
1	TEST1	8	1	3	2
2	CSIS	18	3	5	1
3	CLCS	35	3	1	0
4	SHAS	48	3	5	0
5	ISCCS	83	15	7	22
6	CFT	34	23	18	27

The BAM code obtains only the point probability of the top event, and by the CUT code, minimal cut sets and the point probability can be obtained. In the analysis, cut sets were obtained for 3 intermediate gates for TEST 1 and CLCS trees and for 7 gates for other trees. The WAM preprocessor [6] was performed prior to the execution of the BAM and CUT codes.

The results are shown in table 2. FFTA-TP means that only the point probability of the top event was calculated by the FFTA program, and FFTA-MCS means that minimal cut sets and the point probability were obtained.

The values of the top event probabilities in this table are the ones calculated by FTA-J-BAM. Other codes gave exactly the same results, except for, 6.468×10^{-3} for FT-1 by FFTA, 6.447×10^{-4} for FT-3 by FFTA-MCS, and 2.063×10^{-2} for FT-4 by FTA-J-CUT.

The CPU time does not include the WAM process, which performs the preprocess and print out of the fault tree structure and the probabilities of primary events.

The required CPU time is also shown in fig. 5 in order to intuitively compare the results.

8. Discussions and conclusions

Fig. 5 shows the great reduction in CPU time of the FFTA program compared with those of BAM and CUT codes.

For the fault trees 1 and 2, the CPU time of FFTA-MCS is close to that of FFTA-TP. This is because these fault trees are small and the numbers of minimal cut sets are only 8 and 13. The minimal cut sets were found with little CPU time.

For the fault trees 3 and 4, there are large differences between the CPU time of FFTA-MCS and FFTA-TP. The reason is these fault trees are medium size trees and have many minimal cut sets (155 and 300). A great deal of time was required to find the minimal cut sets. On the other hand, the structures of these fault trees are simple, that is, there are no replicated events. In this case, the FFTA algorithm is effective for the calculation of the point probability of the top event. So, the top event probabilities were obtained with little CPU time.

For the fault trees 5 and 6, the CPU time of FFTA-TP increases and is close to that of FFTA-MCS. These fault

Table 2
Results of the analyses by FFTA, BAM, and CUT codes

Fault tree number	Point probability of top event	Number of minimal cut sets determined	CPU time (s)			
			FFTA-TP	FFTA-MCS	FTA-J-BAM	FTA-J-CUT
1	6.469×10^{-3}	8	0.005	0.007	1.56	1.75
2	2.062×10^{-2}	13	0.006	0.011	2.77	5.95
3	6.449×10^{-4}	155	0.009	1.06	4.63	615
4	2.055×10^{-4}	300	0.009	3.90	6.40	4459
5	7.545×10^{-3}	73	0.134	1.06	10.71	3108
6	1.211×10^{-4}	20	0.71	0.81	6.47	240

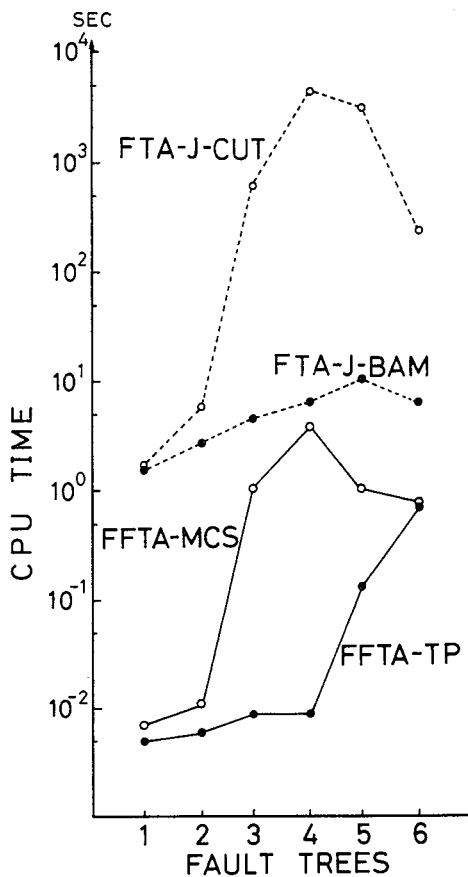


Fig. 5. Required CPU time by FFTA, BAM, and CUT.

trees have complex structure: there are many replicated events. Even in the FFTA-TP process, cut sets containing a replicated event have to be obtained in the intermediate steps, and the calculation procedure becomes almost the same as that of the FFTA-MCS. Therefore, the CPU time of FFTA-TP became close to that of FFTA-MCS.

A marked increase is seen in the CPU time of FFTA-TP for fault tree 6, but the CPU time of FFTA-TP never exceeds that of FFTA-MCS, and it is still nearly ten times faster than that of the FFTA-J-BAM code.

The reduction in CPU time is achieved by the following:

1. Primary events and gates are identified by the assigned numbers.
2. Fault tree structure is expressed by the rows of gate and event numbers, and this expression is well adapted to the process steps in the program.
3. Process steps in the program does use computer core memory effectively and does not use external storage devices.
4. Numerical calculations are performed as early as possible.

5. Unnecessary information is eliminated at every process step, for example, cut sets with high order or low occurrence probability are discarded at every intermediate gate.

The FFTA program calculation speed is ten to a thousand times faster compared with the FFTA-J-BAM and FFTA-J-CUT codes. This small expenditure of computer time enables us to easily perform a fault tree analysis with usual computers without using a high-speed computer system.

Acknowledgement

The author wishes to express his thanks to section chief T. Tobioka, Mr N. Watanabe and Mr M. Hikawa of the department of Nuclear Safety Evaluation, Japan Atomic Energy Research Institute, for providing the computer programs FFTA-J-BAM and FFTA-J-CUT, and also for their useful discussions and suggestions.

I am also grateful to the Editor and referees for their helpful comments and suggestions.

Nomenclature

A, B	= replicated primary events,
\bar{A}	= complement of A ,
$C_i (i > 0)$	= term defined by eq. (8),
COV	= cut off value at each gate,
f_{ij}, f_{ijk}	= factors defined by eq. (11),
$O(h)$	= higher order terms,
$P(\)$	= probability of occurrence,
$P_m(\)$	= numerically summated value from the first to m th term,
TOP	= top event of a fault tree,
X, Y	= non-replicated events,
X_i, C_0	= terms consisting of only non-replicated events,
Z	= intermediate event consisting of non-replicated event.

References

- [1] USNRC (US Nuclear Regulatory Commission), Reactor Safety Study: An Assessment of Accident Risks in US Commercial Nuclear Power Plants, WASH-1400 (NUREG-75/014) (1975).
- [2] Commonwealth Edison Company, Zion Probabilistic Safety Study (1981).

- [3] A. Bayer and F.W. Heuser, Basic aspects and results of German risk study, Nucl. Safety 22 (1981) 695.
- [4] J.B. Fussell and J.S. Arendt, System reliability engineering methodology: a discussion of the state of the art, Nucl. Safety 20 (1979) 541.
- [5] T. Matsuoka, A computer program for automated reliability analysis under extreme environmental condition, ANS Trans. 44 (1983) 363–364.
- [6] T. Ishigami et al., Users' Manual for the FTA-J (Fault Tree Analysis – JAERI) Code System, JAERI-M83-169 (1983).
- [7] USNRC, Reactor Safety Study: Appendix II Fault Trees, WASH-1400 (NUREG-75/014) (1975).
- [8] N. Watanabe et al., A Comparative Study: Top Event Unavailability by Point Estimates and Uncertainty Propagation, JAERI-M82-215 (1983).
- [9] R.B. Worrell and D.W. Stack, A SETS User's Manual for Fault Tree Analyst, NUREG/CR-0465 SAND77-2051 (1978).
- [10] F.L. Leverenz and H.R. Kirch, User's Guide for the WAM-BAM Computer Code, EPRI-217-2-5 (1976).
- [11] F.L. Leverenz and H.R. Kirch, WAMCUT – A Computer Code for Fault Tree Evaluation, EPRI-NP-803 (1978).