Instruction Manual for SPRA Integration in Mastodon

Guidelines and User Manual

Centre for Nuclear Energy

Facilities & Structures

North Carolina State University

Raleigh, NC 27695-7908

Table of Contents

1.0 Background *3*

2.0 Introduction *4*

2.1 Fault Tree analysis **5**

2.2 Event tree analysis **7**

2.3 Common cause failure analysis **8**

3.0 Instructions for Input *10*

3.1 Code inputs **10**

3.2 Input format **12**

4.0 Output *20*

5.0 Example problems for validation *20*

5.1 Example 1 .... **21**

5.2 Example 2 .... **22**

5.3 Example 3 .... **26**

1.0 Background

Seismic probabilistic risk assessment is widely used to examine the seismic risk at nuclear facilities and to identify the key systems, structures, and components (SSC), and accident sequences that have the most impact on safety. Seismic probabilistic risk assessment of nuclear facilities involves five steps: (i) a plant system analysis that involves the development of accident sequences in the form of event trees and fault trees that lead to an accident or an unacceptable performance (e.g., core damage)(ii) a probabilistic seismic hazard analysis that involves the calculation of seismic hazard curves, which describe the mean annual frequency of exceedance of a particular ground motion parameter, e.g., peak ground acceleration or the spectral acceleration at 0.1 sec, (iii) evaluation of probabilistic demands at the locations of safety-related SSCs through probabilistic structural analyses, (iv) evaluation of SSC fragility curves, which describe the probability of failure of the SSCs as function of input seismic demand according to some pre-defined performance criteria, and (v) combining the fragilities of the various SSCs to calculate the system fragilites through fault tree analysis and event tree analysis. The final seismic risk of a system or a facility is then calculated by convolving the seismic hazard curve with the system fragility curve using eq. (1).

$$P_f = \int P_{f|\lambda} \left| \frac{dH(\lambda)}{d\lambda} \right| d\lambda \tag{1}$$

where, λ is the design parameter or the hazard intensity parameter. $P_{f|\lambda}$ is the seismic fragility curve, and $H(\lambda)$ represents the seismic hazard curve.

In current practice, each step of SPRA is conducted separately by using different software and even different teams of engineers. For instance, probabilistic seismic hazard analysis is performed by engineering seismologists, who then provide the seismic hazard curve and hazard and site-consistent earthquake ground motions to structural engineers who perform probabilistic simulations. Results from these simulations are post-processed separately and transferred to another software that performs fault tree analysis and event tree analysis to calculate system level or plant level risk. This requires transfer of large quantities of data between different software and results in a complex, inefficient, and error-prone process. Integration of Seismic probabilistic risk assessment, including fragility calculation, and fault tree analysis and event tree analysis into a finite-element software will avoid such errors and results in a much more seamless process. This report describes and demonstrate seismic probabilistic risk assessment methodology embedded in a risk assessment software called MASTODON. A description of the fault tree analysis and event tree analysis algorithms implemented in MASTODON and corresponding demonstrative examples are presented in the next sections. The report provides a guidance and instructions to the user for incorporating the SPRA analysis in the software.

2.0 Introduction

System risk assessment is an analytical process that quantifies the risk of core damage for a given frequency of a hazard. The hazard is usually referred to as an initiating event which may lead to unsafe conditions at nuclear stations. Generally, an initiating event triggers various system failures with certain probability. These system failures usually form a number of harmful accident sequences. The risk assessment process quantifies the probability of these harmful sequences to evaluate an estimate of total risk. The formation of accident sequences from an initiating event is analyzed in two steps, as described in Saphire Technical Reference. In the first step, event trees are used to model the accident sequences by considering success and failure of safety systems, referred to as "top events". In the second step, the probability of the top event failure, which is propagated due to failure of various basic events, is determined. This step employs fault trees to evaluate failure probabilities of the top events.

An important issue in the risk assessment is failure of multiple components due to a common cause. The frequency of simultaneous multiple failure due to a common cause has relatively low expectancy compared to other random failures. However, such failures lead to direct loss of safety system in many cases. For this reason, the representation of common cause failures in fault tree and event tree analysis is quite important. The common cause event is usually modelled as a separate basic event with its own failure probability. This event is regarded as statistically independent of all other basic events.

The SPRA code in Mastodon has three modules: Fault tree analysis, event tree analysis and common cause failure analysis. A brief discussion of the three analyses and risk quantification measures is provided in the following sections.

2.1 Fault Tree analysis

A Fault tree is a graphical decomposition of a top event failure into intermediate events and basic events through the use of Boolean logic gates, AND and OR. An AND gate indicates that all the connected events must occur for the failure to occur. An OR gate indicates that any of one the connected events must occur for the failure to occur. Failure of a top event in a fault tree is evaluated by identifying the minimal cutsets. The minimal cutsets are a list of sets such that the failure of even single set in the list would lead to the failure of the top event. A unique quality of minimal cutsets is that none of the cutsets in the list is a subset of another cutset. In MASTODON, MOCUS algorithm is used to evaluate the minimal cutsets (Saphire technical reference). The MOCUS algorithm in the mastodon code has been written in FaultTreeUtils.C. The quantification of risk using the minimal cutsets has been implemented in QuantificationUtils.C.

The quantification of top event failure probability using the minimal cutsets is performed in two steps:

1. calculation of individual cutset probabilities – an individual cutset is connected by an AND gate to all the basic events in it. Hence, an individual cutset probability is determined by multiplying the probabilities of all the basic events in it:

$$P\left(CS_i\right)=\prod_{j=1}^{N} P\left(BE_j\right) \tag{2}$$

where, $P(CS_i)$ is the probability of the cutset i, $P(BE_j)$ is the failure probability of the j-th basic event in the i-th cutset, N is the number of basic events in the minimal cutset i.

2. combine the cutset probability to calculate the top event failure probability – QuantificationUtils.C employs three techniques to find the probability for the union of the minimal cutsets, namely rare event approximation, minimal cutset upper bound, and min-max approach. The rare event approximation and minimal cutset upper bound are based on some approximations and provides conservative estimates. These approximation methods are often used to reduce the high computational cost that arises due to the exact calculation of probability. For most of the SSC failures in

nuclear power plants, the approximate calculation of failure probabilities remains valid. The min-max approach based on inclusion-exclusion rule leads to the most accurate quantification. However, it is employed only when the number of minimal cutsets are small, i.e., less than 50 in order to make the analysis computationally efficient. The formulations for the three techniques are shown in Table 1.

Table 1. Formulations for quantification of probabilities with a list of minimal cutsets

Rare event approximation	$\sum_{i=1}^{m} P(CS_i)$
Minimal cutset upper bound	$1-\prod_{i=1}^{m}\left(1-P(CS_i)\right)$
Min-max approach	$\sum_{i=1}^{m} P(CS_i) - \sum_{i<j}^{m} P(CS_i) \times P(CS_j) + \cdots + (-1)^{m-1} P(CS_1) \times P(CS_2) \ldots P(CS_m)$

where, m is total number of minimal cutsets and $P(CS_i)$ is the probability of the i-th minimal cutset.

In a nuclear power plant, it is also necessary to identify the critical events that leads to a top event failure. For identifying critical events in a fault tree, the concept of importance measures is used. The importance measure of a basic event provides information about its impact on the top event. The following importance measures are implemented in QuantificationUtils.C:

Fussell-Vesely (FV) Importance: It evaluates the contribution of basic event of interest to the top event failure probability. It is calculated as the ratio of upper bound estimate of those cut sets that contains the basic event of interest to the overall upper bound estimate of the top event.

$$FV = \frac{F_i}{F_{TE}} \tag{3}$$

Risk Reduction Ratio (RRR): It represents the maximum reduction in the top event failure probability when the basic event has not occurred i.e. the basic event failure probability is equal to 0. The risk reduction interval is calculated by taking the difference between F_{TE} and F_0.

$$RRR = \frac{F_{TE}}{F_0} \qquad RRI = F_{TE} - F_0 \tag{4}$$

Risk Increase Ratio (RIR): It represents the maximum amplification in the top event failure probability when the basic event has occurred i.e. the basic event failure probability is equal to 1. The risk increase interval is calculated by taking the difference between F_1 and F_{TE}.

$$RIR = \frac{F_1}{F_{TE}} \qquad RII = F_1 - F_{TE} \tag{5}$$

Birnbaum Importance: It gives the sensitivity of top event failure probability with respect to a change in basic event failure probability.

$$BI = F_1 - F_0 \tag{6}$$

where, F_{TE} is the upper bound estimate of top event failure, F_i s the upper bound estimate evaluated for all the cut sets containing the i-th basic event, F_1 represents the upper bound estimate of top event when probability of the basic event of interest is set to 1, and F_0 represents the upper bound estimate of top event when probability of the basic event of interest is set to 0.

The use of fault tree analysis module in MASTODON has been explained in Section 5.1 using an illustrative example.

2.2 Event tree analysis

An event tree is a graphical model of response of various systems due to an initiating event. The various order and combination of system failures lead to many different accident sequences. An event tree represents all those possible accident scenarios out of which some sequences lead to unsafe conditions at plant. The event tree analysis is used to calculate the total risk imposed on the plant due to the initiating event leading to unsafe accident consequences. The event tree analysis followed in the Mastodon is same as the analysis method in Saphire. The analysis procedure is briefly described below. A detailed description can be found in Saphire technical reference.

Event tree analysis procedure Each event tree accident sequence is converted into two fault trees where success and failure of top events are combined separately. The top events that have failed in the accident sequence are combined by creating a dummy AND gate with the fault tree of the failed events as inputs. Similarly, the success top events in the accident sequence are combined by creating a dummy OR gate with the fault tree of the corresponding failed events as inputs. The fault tree analysis method is used to determine the minimal cut sets for both the fault trees. The common minimal cut sets between the success and failure fault trees are eliminated from the minimal cut sets of fault tree for the failed systems. These modified cut sets for the failed system fault tree are used to quantify the risk associated with the corresponding accident sequence.

The quantification procedure for fault tree analysis can also be applied to an accident sequence in event tree analysis. In general, an accident sequence would contain a total of *n* top events which can be divided into m top events that exhibit failure and (*n*-*m*) top events that are safe or do not exhibit failure. Therefore, the accident sequence can be represented as a Fault tree of (*n*-*m*) success events being connected using OR gate and a Fault tree of *m* failed events being connected using AND gate. Hence, the analysis and quantification for accident sequences in an event tree are similar to the quantification techniques associated with fault tree analysis. The event tree analysis in MASTODON has been implemented through a C++ code named ETAUtils.C. The use of event tree analysis module in MASTODON has been explained in Section 5.2 using an illustrative example.

2.3 Common cause failure analysis

In fault tree analysis, it is a generic assumption that the basic events or components are statistically independent. However, if a set of components fail due to a common cause event, the components no longer remain independent. In such cases, MASTODON employs a common cause failure (CCF) technique to modify the quantification procedure of accident sequence failure by accounting for common cause event. The components impacted by a common cause event form a common cause component group (CCCG). Then, a basic parameter model (BPM) is used to model the dependent failures due to the common cause event. BPM states that if there are n components in CCCG, then the m components can fail at a time where m ranges from 1 to n. Hence, the failure probability of a component involves independent failure of the component and combinations of CCFs with other components in the CCCG. Thereby a component failure in CCGG in the fault tree model is replaced by an OR gate which is connected to all the possible failures (basic events) due to common cause event linked to component failure of interest and an independent failure. This has been illustrated for a fault tree where two component A and B shares a common cause failure event in in Fig. 2 below.

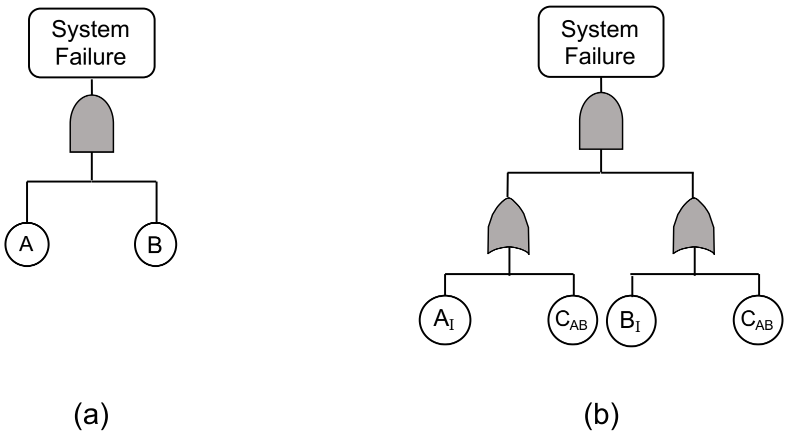


Figure 1*:* Illustration for the inclusion of common cause failures: (a) Fault tree for system failure without CCF, (b) Fault tree for system failure with CCF

where, C_{AB} is the failure of event A and B due to common causes.

The BPM for CCF relies on an underlying assumption of symmetry: the probabilities events in Common Cause Component Group (CCCG) involving similar components are the same. For example, probabilities of events A_i and C_i are equal. This is called symmetry assumption. The probability of failure for all the added basic event due to common cause event are evaluated using alpha factor parameterization. The failure probability involving m simultaneous component failure in a CCCG of size n is given by

$$Q_m = \frac{1}{\binom{n-1}{m-1}} \alpha_k Q_t \tag{7}$$

where, Q_t is the total probability of each component failing due to all independent and common cause events, α_k is the probability that when a common cause basic event occurs in a common cause group of size m, it involves the failure of k components. The common cause failure analysis for an event tree diagram has been written in CCFUtils.C. The CCFUtils.C codes cannot be used by users directly. The CCF files can only be passed through ETAUtils.C analysis for any evaluation.

3.0 Instructions for Input

The main purpose of the code is to provide risk estimates for accidental sequences of the event tree diagrams or top event of fault tree diagrams. For this purpose, users need to provide details of accidental sequences, fault trees, and probability/fragility information of all the components, etc.

3.1 Code inputs

MASTODON SPRA code enables the user to perform either event tree analysis or fault tree analysis. The common cause failure module is currently only available for event tree analysis. In order to perform these analyses, various inputs are required in the SPRA code. Usually an input is in a form of CSV text files, string, integer or Boolean, etc. All the inputs that are required in these two types of analyses are described in the following sections.

<u>Fault tree analysis (FTA)</u>

FTA code requires the following inputs from the users.

Input info	**Allowed values**	**Format**	**Priority**
File for fault tree structure		CSV text file	Required
File for probability values	Point estimate,	CSV text file	Required

	Normal distribution, or Lognormal distribution		
Analysis type	'Risk' or 'fragility'	String	Default- 'Risk'
Hazard curve		CSV text file	Default- Referred in section above
Intensity measure		Integer range	Default- [0.1, 4]
Number of bins	Value ≥ 1	Integer	Default- 15
Uncertainty	True or False	Boolean	Default- False
Number of samples	Value ≥ 1	Integer	Default- 1
Seed	Between 0 - (2^{32}- 1)	Integer	Default- None/set seed from clock

Event tree analysis

ETA code requires the following inputs from users.

Input info	**Allowed values**	**Format**	**Priority**
File for accident sequences		CSV text file	Required
File for fault tree names		CSV text file	Required
File for fault tree structure	All the files referred in fault tree names	CSV text files	Required
File for probability values	Point estimate, Normal distribution, or Lognormal distribution	CSV text file	Required
Analysis type	'Risk' or 'fragility'	String	Default- 'Risk'
Files for common cause failure	[File1*, File2**, File3***]	List of strings for	Default- None

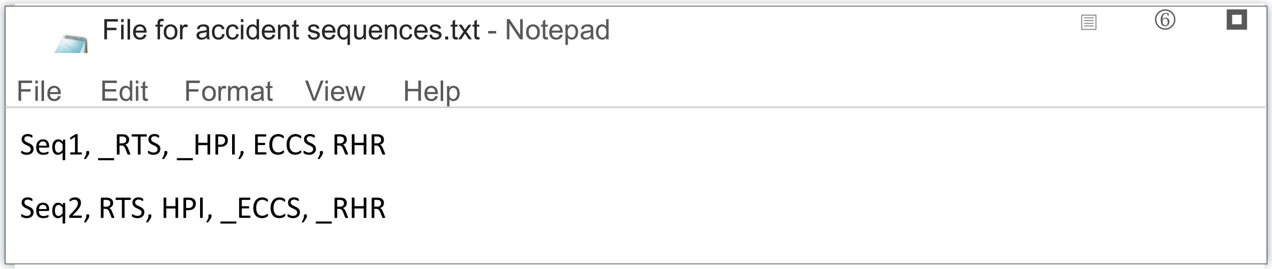
input		file names	
Hazard curve		CSV text file	Default- Referred in section above
Intensity measure		Integer range	Default- [0.1, 4]
Number of bins	Value ≥ 1	Integer	Default- 15
Uncertainty	True or False	Boolean	Default- False
Number of samples	Value ≥ 1	Integer	Default- 1
Seed	Between 0 - (2^{32}- 1)	Integer	Default- None/set seed from clock
File1* - File for common cause failure groups		CSV text file	Required
File2** - File for common cause failure parameters		CSV text file	Required
File3*** - File for CCF sequence information		CSV text file	Required

These analysis inputs referred in section above can be used to evaluate failure probability, risk estimates or importance measures. The necessary inputs are indicated as required or otherwise a default values are listed in the column at right. The detail for each of the input required in the above tables are described in the following section 3.2.

3.2 Input format

File for accident sequences

The accident sequences should be inputted as text files, where each row denotes an unsafe accident sequence. The sequences that lead to safe outcomes should not be mentioned. The first element of each row denotes the name of the sequence. This name of the sequence must be consistent for all the sequence related inputs in the code. The sequence name is followed by name of top event failures along the sequence. If a top event does not fail along the sequence progression, it must be written with a prefix '_'. All the top event names or sequence name must be inputted as a single word and should be separated by comma. The order of rows does not matter. Below is an example of CSV text file's content.

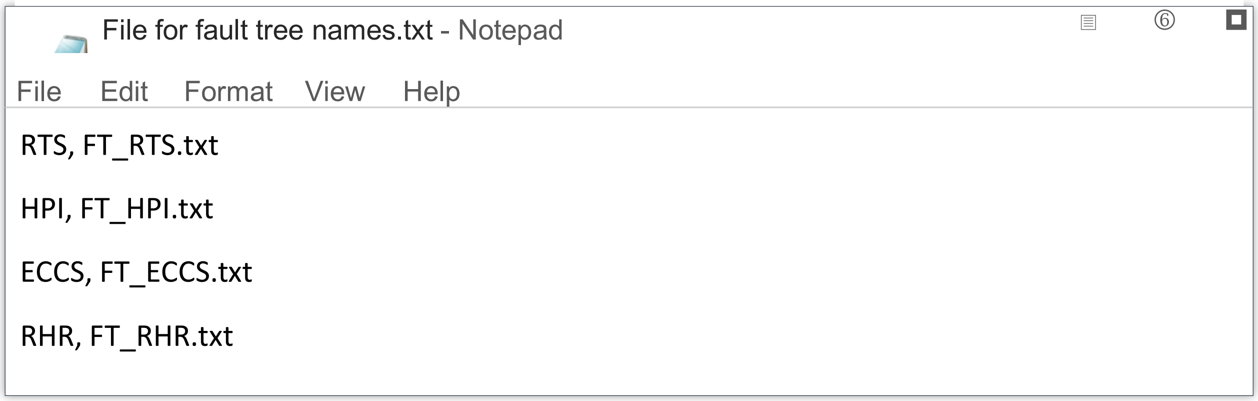

File for accident sequences.txt - Notepad
File Edit Format View Help

```
Seq1, _RTS, _HPI, ECCS, RHR

Seq2, RTS, HPI, _ECCS, _RHR
```

The above example has two accident sequences. The first accident sequence is denoted by Seq1. Along this sequence, RTS and HPI do not fail while ECCS and RHR fail.

<u>File for fault tree names</u>

All the top events in an event tree must be related to a fault tree. This information should be provided to model in a form of CSV test files. In the CSV file, each row contains the information of a unique fault tree. The first element of the row is the name of top event and second element of a row is the name of CSV text file which contains the structure of the fault tree. A fault tree text file must only refer to the failure of top event. The user does not provide fault trees for success events. Below is an example of such CSV file for Seq1 and Seq2 shown previously,

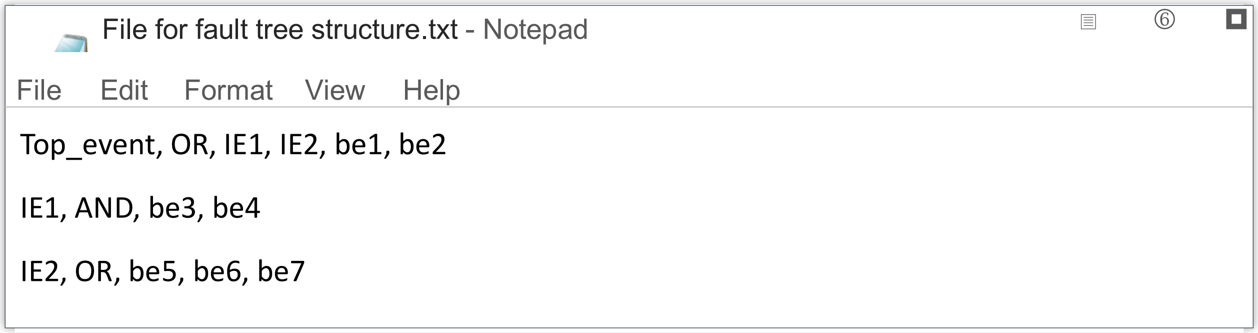

File for fault tree names.txt - Notepad
File Edit Format View Help

```
RTS, FT_RTS.txt

HPI, FT_HPI.txt

ECCS, FT_ECCS.txt

RHR, FT_RHR.txt
```

Here, the rows refer to fault tree file for the failure of RTS, HPI, ECCS and RHR, respectively. If a user does not wish to input a fault tree for the top events of event tree, s/he can input a fault tree with only one basic event connected with OR logic- the top event failure probability can be assigned to this dummy basic event. All the failure events referred in File for accident sequences must have a fault tree file name in this CSV file.

File for fault tree structure

Fault tree for all the top events contain their own fault tree structure. This fault tree structure is inputted as a CSV text file. In the CSV text file, component failures are logically combined through several gates to result in top event failure. These gates could be AND gate or OR gate. The intermediate events can have a dummy name (only one word allowed). The name for component/ basic event failure cannot be used as gates. Again, for the failure of component, one-word description is allowed. The component failure names must be consistent across all the fault trees to ensure the common component failures among multiple fault trees are treated correctly. The user needs to ensure that the fault tree input to the PRA code is correct. In the fault tree CSV file, each row refers to a sub-tree in which two levels (top-intermediate or intermediate-bottom) of failures are connected to each other through a logical gate (AND/OR). The first row of the file must refer to the top event failure and its dependence. The first element of a row denotes the top event of the sub-tree, the second element of a row denotes the logical gate (AND/OR) and the subsequent elements of a row denotes the second level events of a sub-tree. For example,


File for fault tree structure.txt - Notepad
File Edit Format View Help

```
Top_event, OR, IE1, IE2, be1, be2

IE1, AND, be3, be4

IE2, OR, be5, be6, be7
```

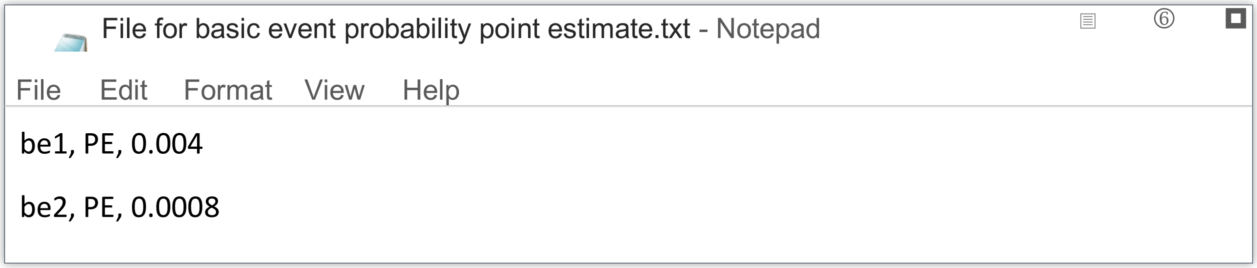
In the first row, the top_event is connected to two intermediate events IE1 and IE2 and two basic events be1 and be2. The intermediate events for a fault tree are identified when they are again referred in the rows as a top event of another sub-tree, like IE1 and IE2 are referred again in second and third row.

The rows other than the first row can be in any order as long as the intermediate event is referred in the rows mentioned above it. There should not be duplicate rows in a fault tree text file and there should not be any duplicate first element in any row. The component failures/basic event failures must not the first element of any of the rows in this file.

File for probability values

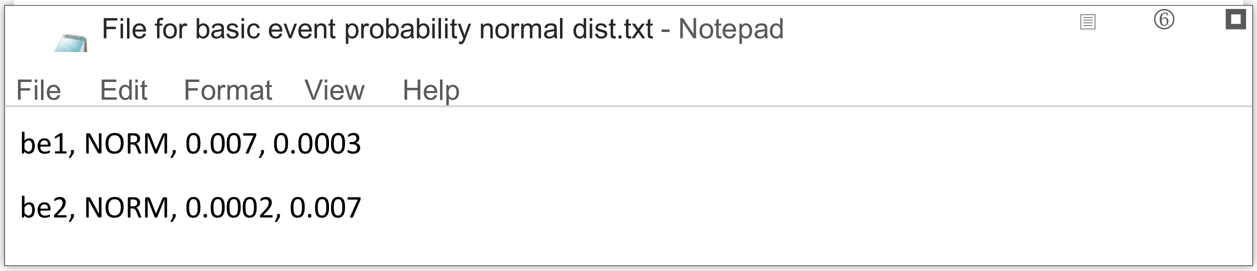
All the component failure probabilities should be listed in this file. The information related to each component must be listed in a separate row. Each row should contain three information (comma-separated) in the following order: component name, probability type and probability value. Component name should be same as the one used in File for fault tree structures. The probability type could be point estimate, normal distribution or lognormal distribution. The user should use PE, NORM and LNORM, respectively to indicate the probability type. When PE is indicated, write only one value that indicate the point estimate of the probability of failure. When NORM or LNORM is used, write mean and standard deviation for the probability of failure distribution.

Point estimate:


File for basic event probability point estimate.txt - Notepad
File Edit Format View Help

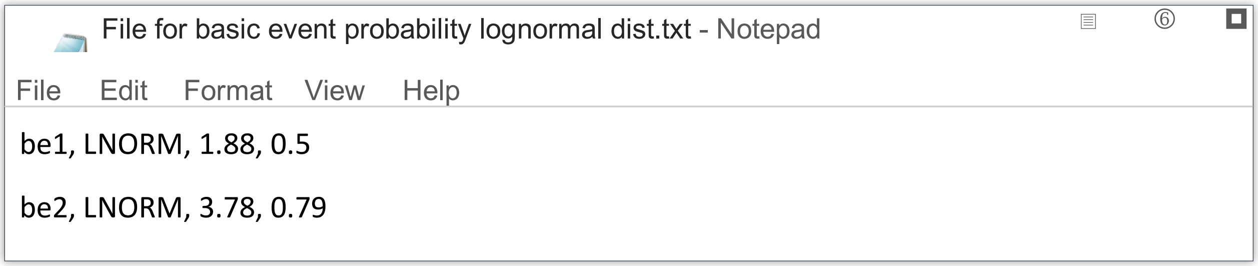
```
be1, PE, 0.004

be2, PE, 0.0008
```

Fragility- Normal distribution:


File for basic event probability normal dist.txt - Notepad
File Edit Format View Help

```
be1, NORM, 0.007, 0.0003

be2, NORM, 0.0002, 0.007
```

Fragility- Lognormal distribution:


File for basic event probability lognormal dist.txt - Notepad
File Edit Format View Help

```
be1, LNORM, 1.88, 0.5

be2, LNORM, 3.78, 0.79
```

In an event tree analysis, all the basic event failures existing in fault trees of all the top events of event tree are referred using only one *basic event probability* file. Initiating event frequency for the event tree should also be inputted using this file. Use 'IE' to indicate the initiating event probability. If this information is not provided, IE is assumed to be 1. When common cause failures are present in the event tree, then the failure probability for CCF should also be provided in this file.

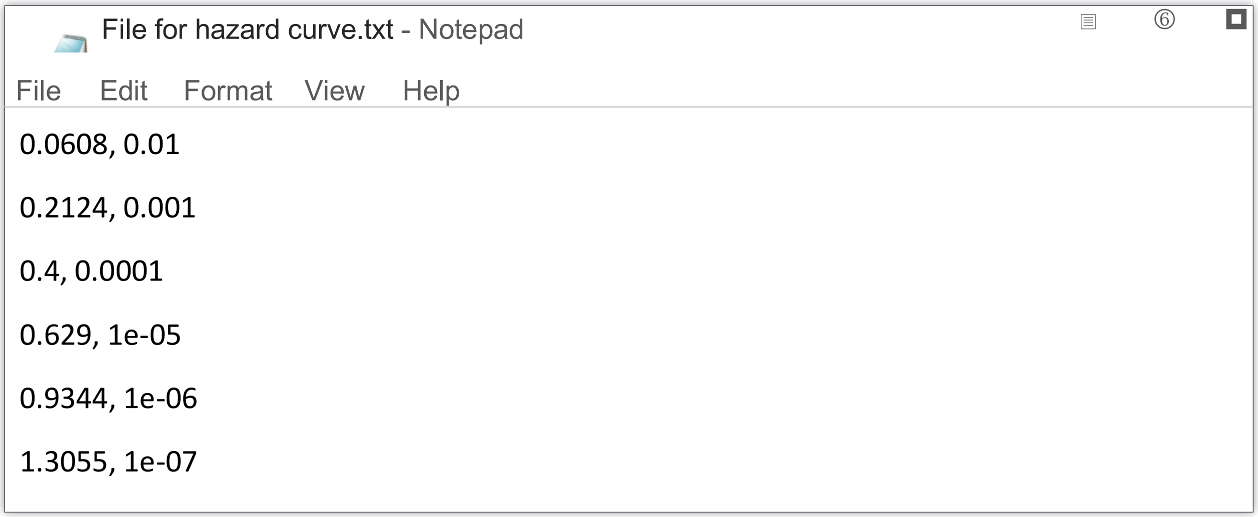
Analysis type

Based on the probability of failure provided (point estimate or distribution), the analysis type could be 'Risk' or 'Fragility'. This information is inputted as a string. The default value for analysis is 'Risk'.

Hazard curve

When the analysis type is fragility, a hazard curve should be provided by user to estimate the total risk. The hazard curve can be inputted using a CSV text file. Each row in the text file gives the hazard intensity and its probability of exceedance. The first element in the row should be hazard intensity and the second element should be probability of exceedance. The two quantities in each row must be separated by a comma.

When user does not provide a hazard curve, a default seismic hazard curve is used (as shown below).


File for hazard curve.txt - Notepad
File Edit Format View Help

```
0.0608, 0.01
0.2124, 0.001
0.4, 0.0001
0.629, 1e-05
0.9344, 1e-06
1.3055, 1e-07
```

<u>Intensity Measure</u>

It is provided as a range in a form of a list with two digits. It is the upper bound and lower bound for estimation of risk from the fragility curve. This is only applicable when a distribution is provided through <u>File for probability values</u>. The component failure probabilities from the distribution will be considered between this range. A default value is [0.1, 4]. The system level risk estimate is calculated at regular interval between the intensity measure. The interval is defined using the following parameter, nbins.

<u>Number of bins (nbins)</u>

This is inputted as an integer. It is total number of bins to be considered for risk estimation between the given IM range. A default value is 15. Any other value greater than 1 can be inputted for the calculation.

<u>Uncertainty</u>

It is inputted as Boolean (True/False). Default value for this parameter is False. When it is False, the risk is evaluated at point estimate values. If it is set as True, a monte carlo simulation is performed.

<u>Number of samples (nsamp)</u>

It is inputted as an integer. The default value is 1. This indicates the number of samples for uncertainty analysis. Any integer greater than 1 can be inputted for this parameter.

Seed

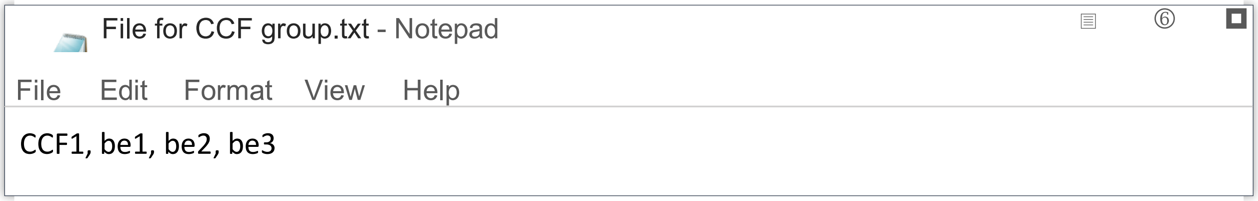
It is inputted as an integer. The default value is None (set seed from clock). Any integer between 0 and (2^{32}- 1) can be inputted for this parameter.

Common cause failure module:

The common cause failure module can be used for the event tree analysis. The event tree analysis performs CCF analysis when File for common cause failure group is provided. The user must provide additional two files (File for common cause failure parameter, File for sequence information) for CCF when the first file is inputted. The common cause failure information should be inputted for the different sequences present in the event tree. Each common cause failure present in the system should have a unique name. This name should be consistent in all the following input files.

File for common cause failure group

It contains the name of common cause failure group and the component failures associated with this failure group. The relevant information about the common cause failure groups in the event tree should be inputted using the input CSV text file. Each row has the information of a different common cause failure group in the event tree. The first element in each row is the name of the CCF group and the rest of the elements in each row are the names of component failures in the CCF group. Any row in this file will have the following format,



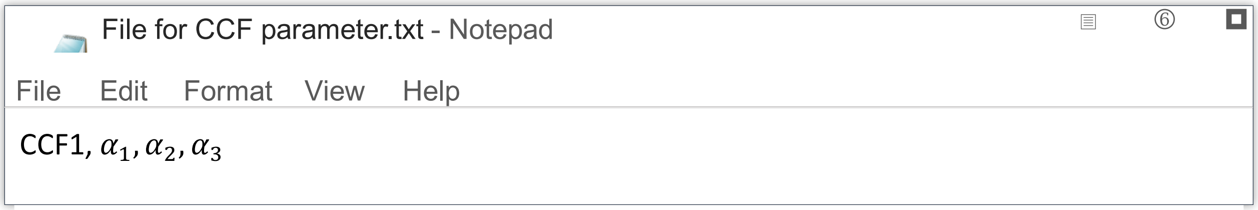
```
CCF1, be1, be2, be3
```

The above row indicates that the common cause failure group CCF1 has three component failure. These components are *be1*, *be2* and *be3*.

File for common cause failure parameter

This file contains the values of common cause failure parameters for alpha factor parameterization. For a nuclear power plant this parameters can be found here https://nrcoe.inl.gov/resultsdb/publicdocs/CCF/ccfparamest2015.pdf

For a common cause failure group of size n, there will be n such CCF parameters. These parameters should be listed in a CSV text files. Each row contains information of one different CCF group. The CCF groups mentioned in this file will be identical as the ones provided in File for common cause failure group. Although the order of CCF groups does not matter. The first element in each row is the name of the CCF group and the rest of the elements in each row are the values of parameters in order as $\alpha_1, \alpha_2, \alpha_3, \ldots\ \alpha_n$, where n is the size of CCF group. Any row in this file will have the following format,

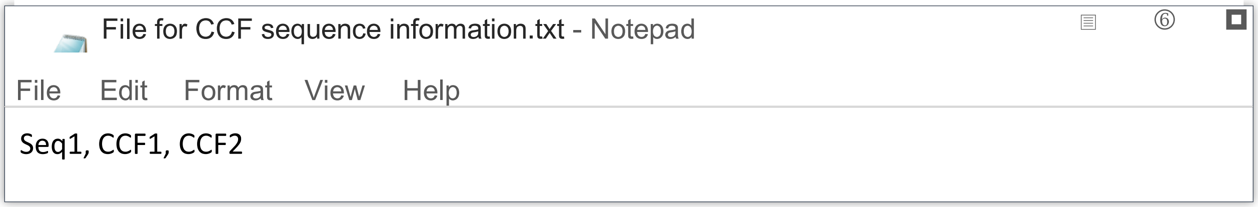


CCF1, $\alpha_1, \alpha_2, \alpha_3$

The above row gives the values of the alpha parameters for common cause failure group CCF1.

File for sequence information

This file gives names of the sequences that contain CCF scenarios. Each row in this CSV text file gives information about a different sequence. It is not necessary that all the sequences in event tree may share some CCF events. The sequence name referred in this file must also exist in File for accident sequences. Each row in this file starts with a sequence name and followed by the names of CCF it shares. The CCFs listed in this file must exist in File for common cause failure group and File for common cause failure parameter. Any row in this file will have the following format,



```
Seq1, CCF1, CCF2
```

Any sequence can share any number of CCF events. It is user's responsibility to make sure that the listed CCFs are indeed shared by the particular sequence.

4.0 Output

Event tree analysis

The ETAUtils.C code provides the total risk estimate calculated using rare event approximation/ upper bound analysis/ min-max approach for each of the failure sequences. All the following importance measures are calculated for the accident sequences: Fussell-vesely, risk reduction ratio. risk increase ratio, risk reduction difference, risk increase difference and birbaum importance.

Fault tree analysis

The QuantificationUtils.C code provides the total risk estimate for top event which is calculated using rare event approximation/ upper bound analysis/ min-max approach. All the following importance measures are calculated for the top event of the fault tree: Fussell-vesely, risk reduction ratio. risk increase ratio, risk reduction difference, risk increase difference and birbaum importance. The FaultTreeUtils.C code provide the set of minimal cutsets for the top event of the fault tree.

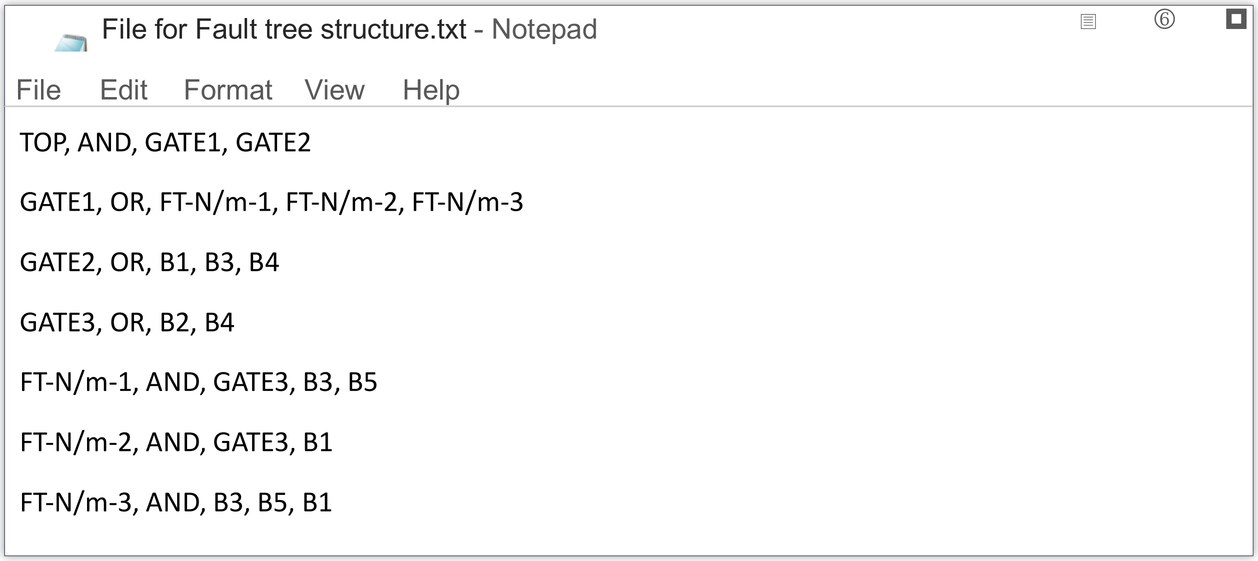
Sometimes the importance measure ratio may lead to invalid values when F_0 is 0. F_0 means that all the cutsets have a common basic event for which the value is set to be zero. In such a case, it is not possible to provide an importance measure estimate for the basic event. A constant value of -1 is provided in results in those cases. The code prints that 'Tree contains a common basic event in each cutset' to indicate such presence of basic events to the users.

5.0 Example problems for validation

The fault tree code is validated through the example provided in the sapphire technical manual. This example has been shown in the following section 5.1. However, there is no validation example for event tree analysis in sapphire technical manual. Hence the event tree code is validated through solving simple event trees using hand calculations.

5.1 Example 1

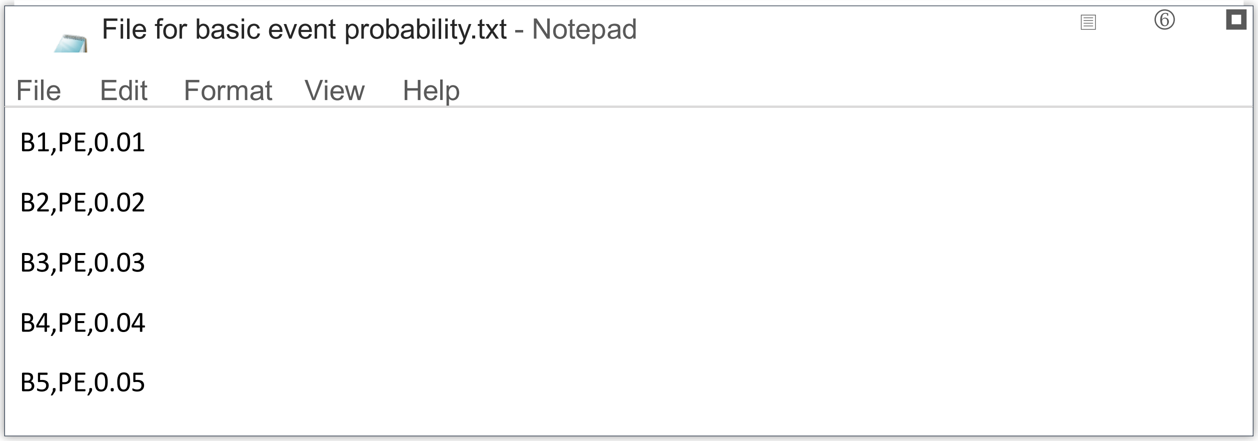
The fault tree structure for example 1 in inputted in FTA modules using the following CSV text file (unit/logic2.txt).


File for Fault tree structure.txt - Notepad
File Edit Format View Help

```
TOP, AND, GATE1, GATE2

GATE1, OR, FT-N/m-1, FT-N/m-2, FT-N/m-3

GATE2, OR, B1, B3, B4

GATE3, OR, B2, B4

FT-N/m-1, AND, GATE3, B3, B5

FT-N/m-2, AND, GATE3, B1

FT-N/m-3, AND, B3, B5, B1
```

The above fault tree structure validated through point estimate of basic event probability.

Point estimate (unit/logic2_bas_events_PE.txt):


File for basic event probability.txt - Notepad
File Edit Format View Help

```
B1,PE,0.01

B2,PE,0.02

B3,PE,0.03

B4,PE,0.04

B5,PE,0.05
```

Result obtained from FTA code is as follows,

Top event failure probability estimate

- Min- max: 0.000694
- Upper bound: 0.000705
- Rare event: 0.000705

Importance measures

	B1	B2	B3	B4	B5
FV	0.872395	0.3263	0.148963	0.652584	0.148963
RRR	7.831866	1.483999	1.174913	2.877066	1.174913
RIR	86.1111	16.96027	5.808755	16.63774	3.826894
RRD	0.000615	0.00023	0.000105	0.00046	0.000105
RID	0.059991	0.01125	0.003389	0.011022	0.001993

The results for example 1 are also replicated using QuantificationUtils.C and the results are validated in TestQuantificationUtils.C.

5.2 Example 2

An event tree with an initiating event and two top events has been shown in Figure 2. The total risk due to unsafe accident sequences 2, 3 and 4 has been evaluated. The accident sequence 2 consists of an initiating event, success of System 1 followed by a failure of System 2. The accident sequence 3 consists of an initiating event, failure of System 1 followed by a success of System 2. The accident sequence 4 consists of an initiating event, failure of System 1 followed by a failure of System 2. In any of the sequence, only two failure events are possible. Hence, the success system fault tree or failure system fault tree will consist a maximum of two top events. The overall minimal cut sets will be determined by eliminating the common cut set between success system fault tree and failed system fault tree from the cut sets of failed system fault tree.

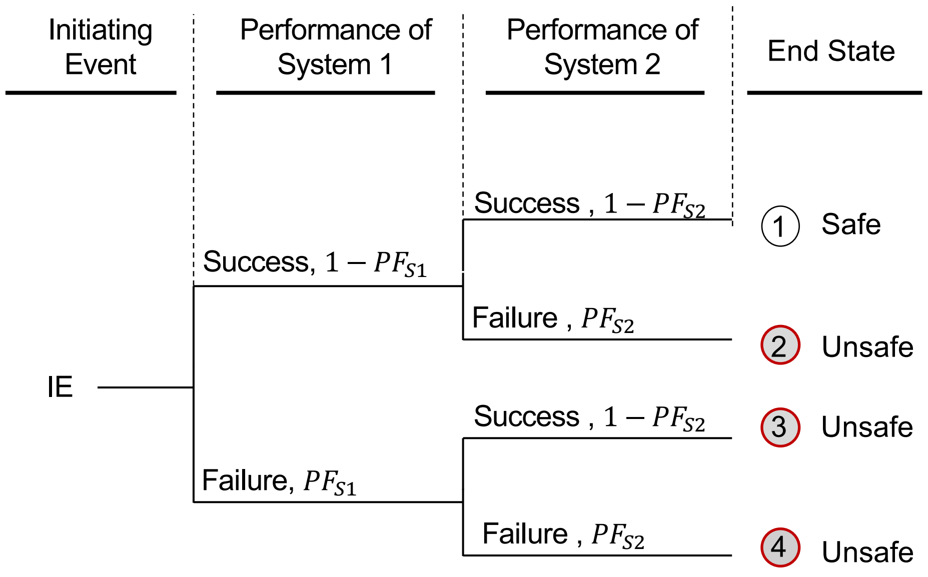


Figure 2: Simple event tree example

The fault trees for system 1 (success) and 2 (failure) are shown in Figure 3. The minimal cut sets for fault tree 1 (FT1) are (A B, C, G) and the minimal cut set for fault tree 2 (FT 2) is (ED, FD).

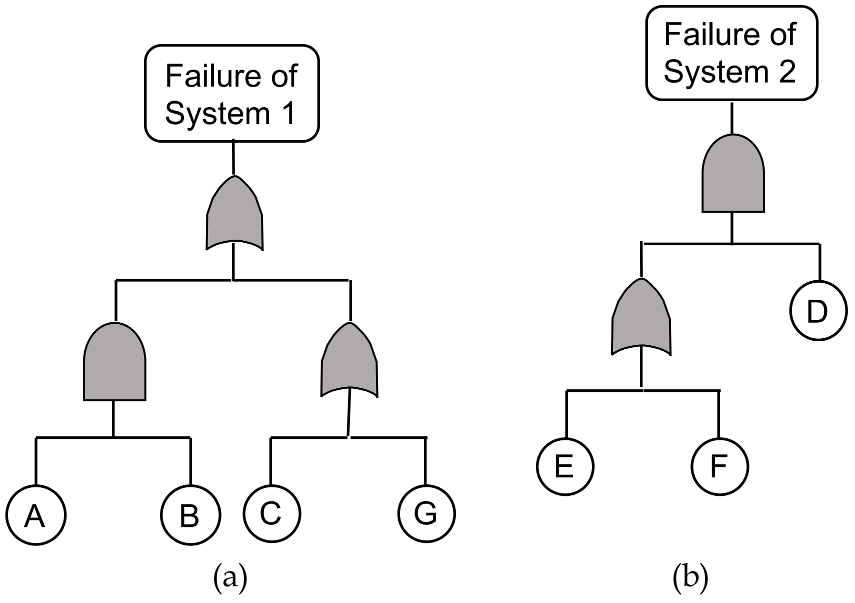


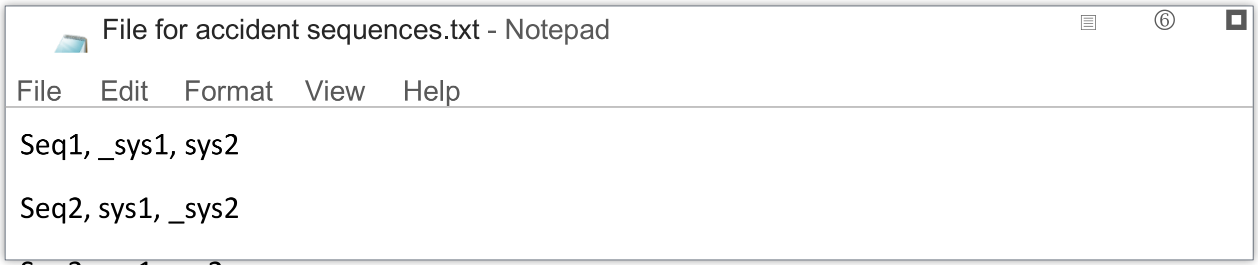
Figure 3: Example 2 Fault tree for (a) System 1 & (b) System 2

Based on the sequences, Sequence 2, 3 and 4 will have the expression: $FT1^c \cap FT2$, $FT1 \cap FT2^c$, $FT1 \cap FT2$.

First step is to identify the list of minimal cutsets for Sequence 2, 3 and 4. The following table provide the list of cutsets for the three sequences.

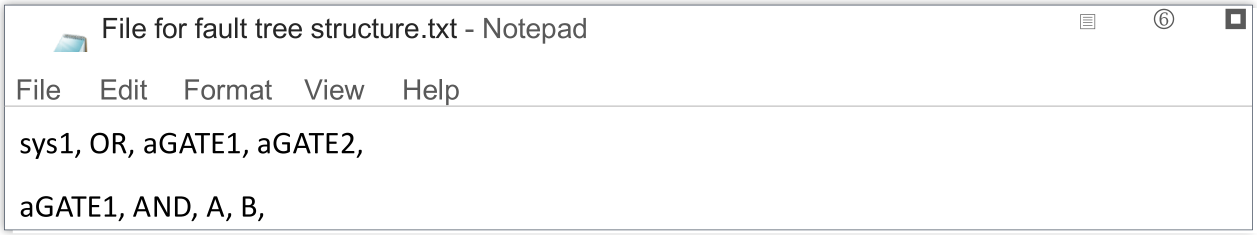
Seq 2	$D \cap E,\ D \cap F$
Seq 3	$\mathrm{A} \cap B,\ C, G$
Seq 4	$(D \cap A \cap B \cap E),\ (D \cap A \cap B \cap F),\ (D \cap C \cap E),\ (D \cap C \cap F),\ (D \cap G \cap E),\ (D \cap G \cap F)$

Based on the above minimal cutsets, all the quantification can be performed easily as described in section 2.1. The input files for the event are as follows (unit/E1_TE.txt),


File for accident sequences.txt - Notepad
File Edit Format View Help

```
Seq1, _sys1, sys2

Seq2, sys1, _sys2

Seq3, sys1, sys2
```

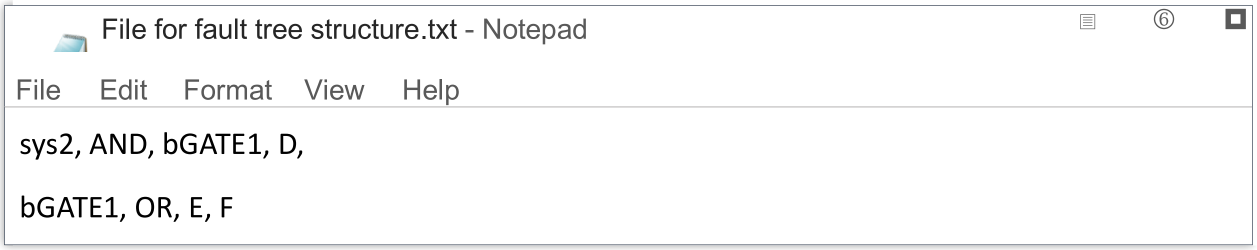
System 1 (unit/E1_sys1.txt)


File for fault tree structure.txt - Notepad
File Edit Format View Help

```
sys1, OR, aGATE1, aGATE2,

aGATE1, AND, A, B,

aGATE2, OR, C, G
```

Here, aGATE1 and aGATE2 are the dummy intermediate gates.

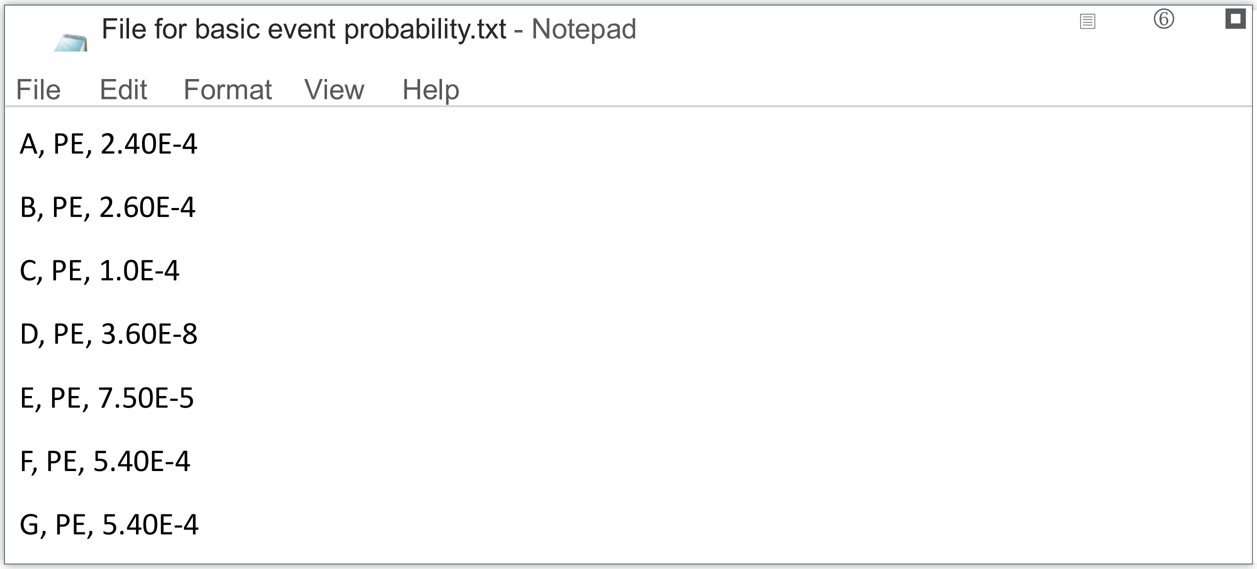
System 2 (unit/E1_sys2.txt)


File for fault tree structure.txt - Notepad
File Edit Format View Help

```
sys2, AND, bGATE1, D,

bGATE1, OR, E, F
```

Here, bGATE1 and bGATE2 are the dummy intermediate gates.

(unit/be_prob_E1E2.txt)


File for basic event probability.txt - Notepad
File Edit Format View Help

```
A, PE, 2.40E-4

B, PE, 2.60E-4

C, PE, 1.0E-4

D, PE, 3.60E-8

E, PE, 7.50E-5

F, PE, 5.40E-4

G, PE, 5.40E-4
```

The importance measures and failure probability for the three sequences are evaluated by hand calculation and are shown in the tables below.

Failure probability:

	Seq 2	**Seq 3**	**Seq 4**
Rare app.	2.21E-11	6.4E-04	1.417E-14

Min- max	2.21E-11	6.4E-04	1.417E-14
Upper bound	2.21E-11	6.4E-04	1.417E-14

The importance measure estimate for sequence 2 are as follows,

Seq 2	***D***	***E***	***F***
RRR	-1	1.138886351	8.200131584
RRI	2.214E-11	2.69995E-12	1.944E-11
RIR	27776003.04	1626.897502	1626.141401
RII	6.14959E-04	3.59973E-8	3.59806E-8

The importance measure estimate for sequence 3 are as follows,

Seq 3	***A***	***B***	***C***	***G***
RRR	1.000097446	1.000097446	1.1850638	6.396092838
RRI	6.23601E-8	6.23601E-8	9.9946E-05	5.39946E-04
RIR	1.405887282	1.374657689	1562.47959	1562.47959
RID	2.59771E-04	2.39784E-04	0.99936	0.99936

5.3 Example 3

This example demonstrates the process to include the contribution of Common Cause Failures (CCF) in event tree analysis. Saphire utilizes Basic Parameter Model (BPM) and alpha factor model to evaluate the probabilities of failure corresponding to CCF. An event tree diagram for this example is same as the one used in example 2 in Figure 2. The fault trees corresponding to the failure of System 1 and System 2 are shown in Figure 4.

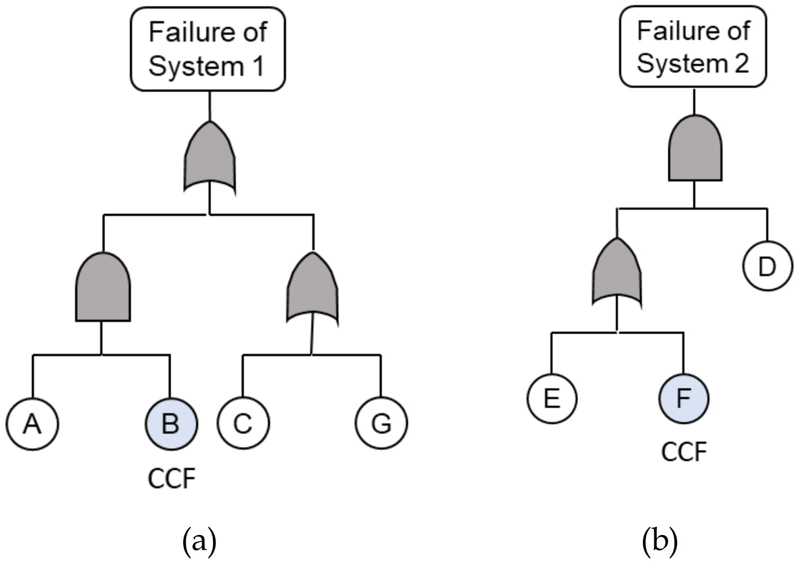


Figure 4: Example 3 Fault tree for (a) System 1 & (b) System 2

In Figure 4, it is assumed that the basic events *B* and *F* share a common cause failure. The BPM expands the failure probability of common cause components into independent failure of the component and combination of CCFs. Hence, the total failure of components, designated as B and F would have the following expansions:

$$B_t = B_i \cup C_{BF}$$

$$F_t = F_i \cup C_{BF}$$

8

Where, B_t is the total failure of B from all causes,
F_t is the total failure of F from all the causes,
B_i is the failure of B from independent causes,
F_i is the failure of F from independent causes,
C_{BF} is the failure of B and F from common causes.

As per the BPM, the fault tree diagrams for system 1 and system 2 are modified as shown in Figure 5.

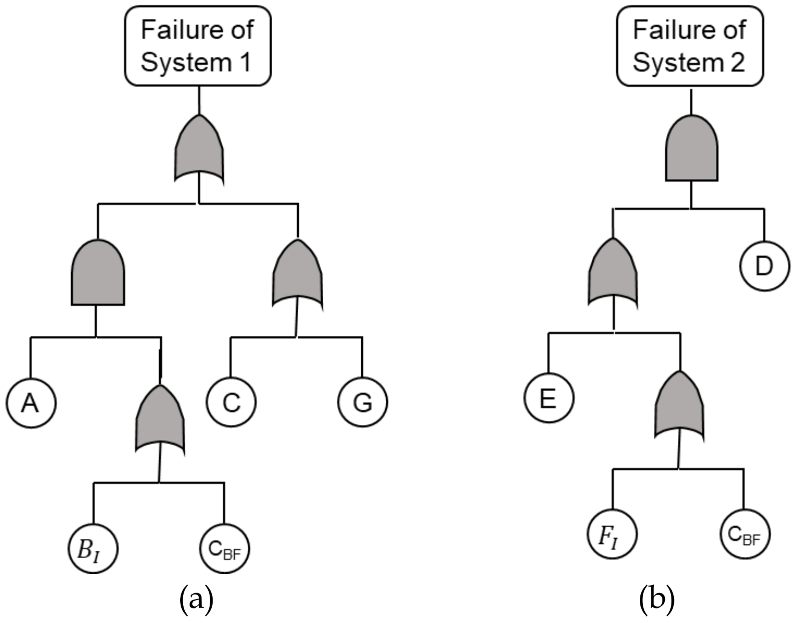


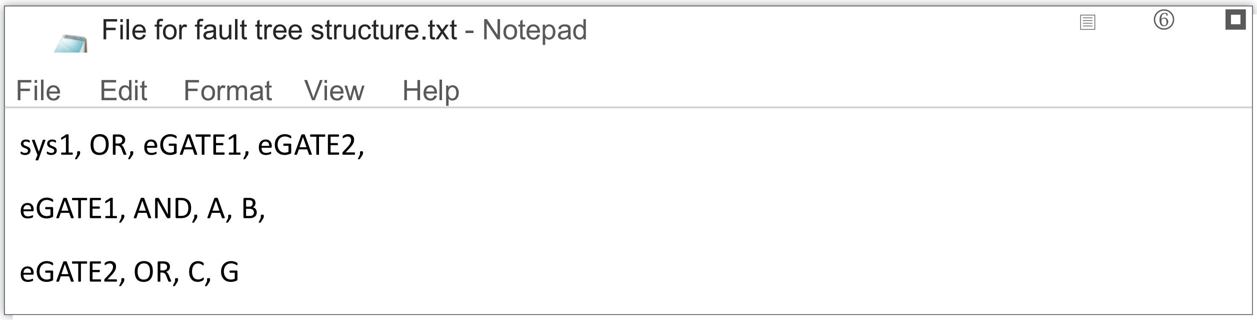
Figure 5: Example 3 Fault tree for (a) System 1 & (b) System 2

Again, the first step is to identify the list of minimal cutsets for Sequences. The following table provide the list of cutsets for the sequence2 and 3. The risk quantification can be made based on the list of minimal cutsets.

Seq 2	$E \cap D,\ \ D \cap F_I,\ \ D \cap C_{BF}$
Seq 3	$A \cap B_I,\ A \cap C_{BF}, C, G$

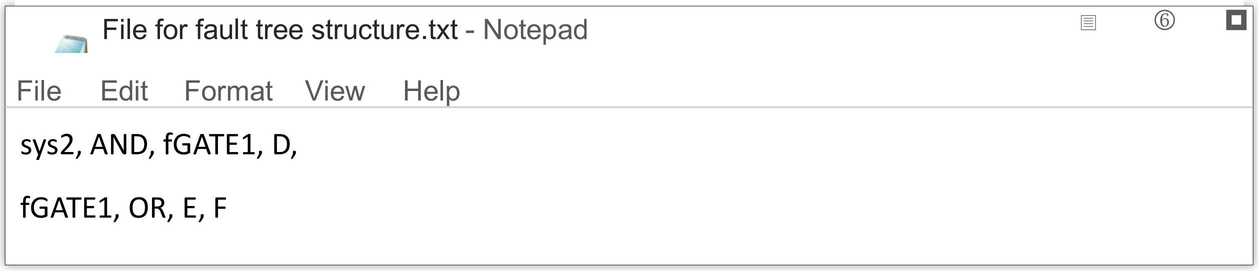
The input files for the event tree in this example will have the same format as the example 2 event tree. However, the fault tree structures for the top events would be different.

System 1 (unit/E3_sys1.txt)


File for fault tree structure.txt - Notepad
File Edit Format View Help

```
sys1, OR, eGATE1, eGATE2,

eGATE1, AND, A, B,

eGATE2, OR, C, G
```

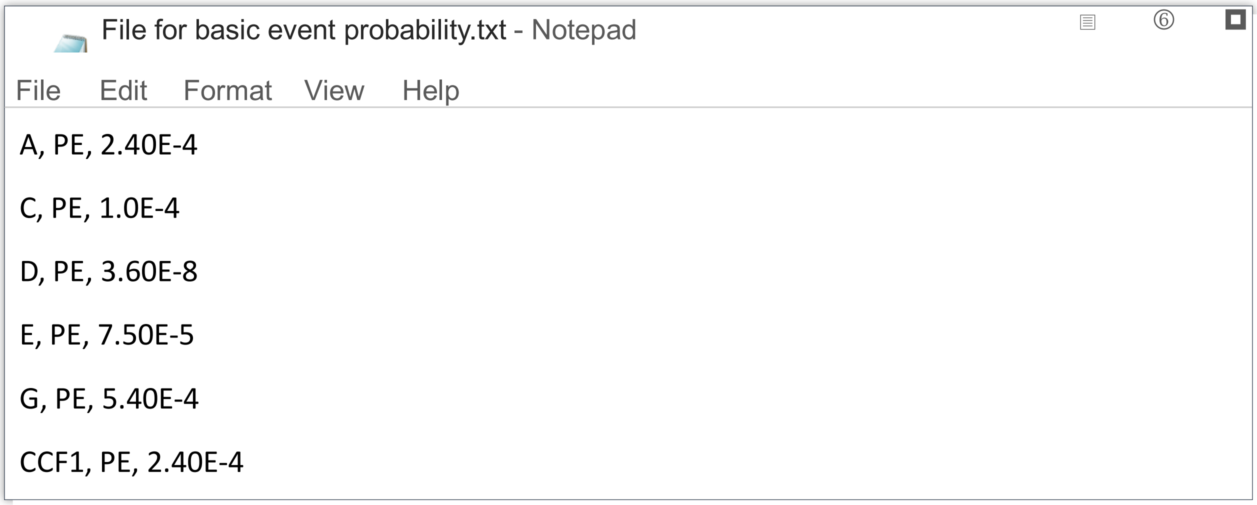
System 2 (unit/E3_sys2.txt)


File for fault tree structure.txt - Notepad
File Edit Format View Help

```
sys2, AND, fGATE1, D,

fGATE1, OR, E, F
```

Since B and F share a common cause failure event, their probability of failure should be same based on symmetry assumption of alpha factor parametrization. Hence, a common probability of failure is provided by variable named CCF. Instead of providing the probability of failure for individual basic events in CCCG, a common probability of failure for the all the events in CCCG can be provided by its name, i.e. CCF1.

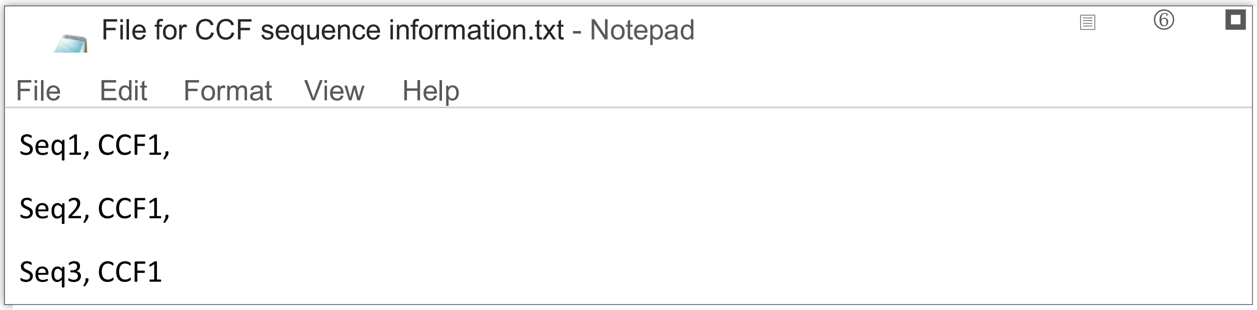
(unit/be_prob_E3.txt)


File for basic event probability.txt - Notepad
File Edit Format View Help

```
A, PE, 2.40E-4
C, PE, 1.0E-4
D, PE, 3.60E-8
E, PE, 7.50E-5
G, PE, 5.40E-4
CCF1, PE, 2.40E-4
```

All the sequences share common cause failure event, hence,

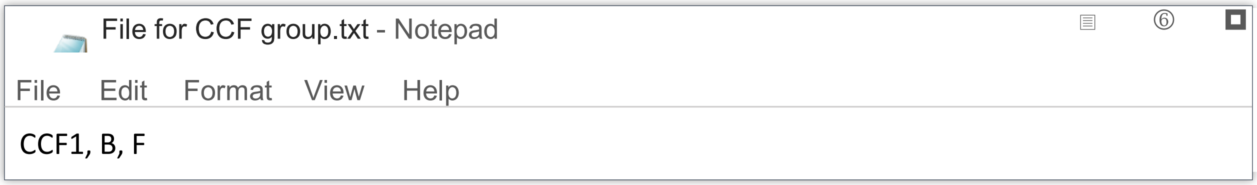
(unit/E3_CCF_Ft_Details.txt)


File for CCF sequence information.txt - Notepad
File Edit Format View Help

```
Seq1, CCF1,
Seq2, CCF1,
Seq3, CCF1
```

The common cause failure group, CCF1, shares two basic event *B* and *F*, hence,

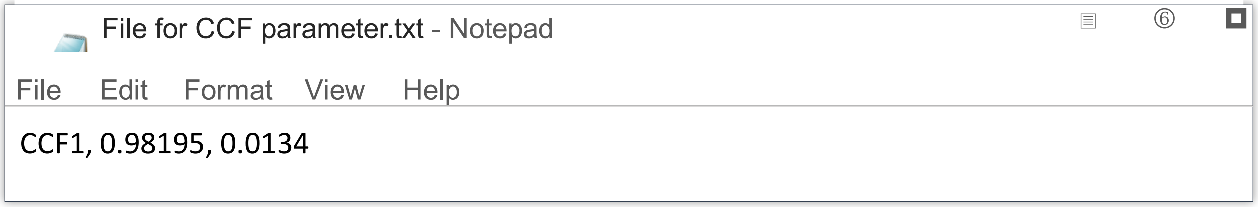
(unit/E3_CCF_List.txt)



```
CCF1, B, F
```

The two parameters corresponding to the CCF1 are,

(unit/E3_CCF_alpha.txt)


File for CCF parameter.txt - Notepad
File Edit Format View Help

```
CCF1, 0.98195, 0.0134
```

The importance measures and failure probability for the sequence 2 and sequence 3 are evaluated by hand calculation and are shown in the tables below.

Failure probability:

	Seq 2	**Seq 3**
Rare app.	1.127E-11	6.4E-04
Min- max	1.127E-11	6.4E-04
Upper bound	1.127E-11	6.4E-04

The importance measure estimate for sequence 2 are as follows,

Seq 2	***D***	***E***	$\boldsymbol{B_I}$ **and** $\boldsymbol{F_I}$	$\boldsymbol{C_{BF}}$
RRR	-1	1.314833515	4.004534343	1.010376045
RRI	1.12758E-11	2.69995E-12	8.46001E-12	1.15796E-13
RIR	27776170.36	3193.450538	3192.939702	3193.679716
RID	0.000313197	3.59973E-08	3.59915E-08	3.59999E-08

The importance measure estimate for sequence 3 are as follows,

Seq 2	A	C	G	B_I **and** F_I	C_{BF}
RRR	1.000089281	1.185065592	6.396374793	1.000088076	1.000001205
RRI	5.71353E-08	9.9946E-05	0.000539946	5.63639E-08	7.71346E-10
RIR	1.371882028	1562.492346	1562.492346	1.374670116	1.374756958
RID	0.000238006	0.999359997	0.999359997	0.00023979	0.000239846

The results for example 2 and example 3 are also replicated using ETAUtils.C and the results are validated in TestETAUtils.C.