JRC Scientific and Technical Reports

# ASTRA 3.x: THEORETICAL MANUAL

**Sergio Contini and Vaidas Matuzas**

JRC
EUROPEAN COMMISSION

iPSc
Institute for the Protection
and Security of the Citizen
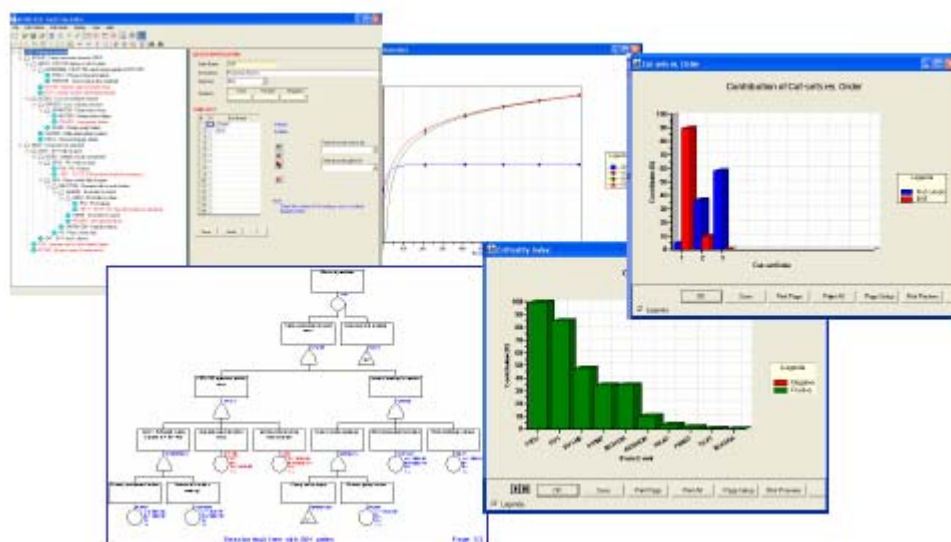
The mission of the JRC-IPSC is to provide research results and to support EU policy-makers in their effort towards global security and towards protection of European citizens from accidents, deliberate attacks, fraud and illegal actions against EU policies.

---

***Europe Direct is a service to help you find answers
to your questions about the European Union***

**Freephone number (*):**

**00 800 6 7 8 9 10 11**

(*) Certain mobile telephone operators do not allow access to 00 800 numbers or these calls may be billed.

---

A great deal of additional information on the European Union is available on the Internet.
It can be accessed through the Europa server http://europa.eu/

*Printed in Italy*

# SUMMARY

This report describes the main algorithms implemented in ASTRA 3.x to analyse coherent and non-coherent fault trees. ASTRA 3.x is fully based on the state-of-the-art of Binary Decision Diagrams (BDD) approach. In case of non-coherent fault trees ASTRA 3.x dynamically assigns to each node of the graph a label that identifies the type of the associated variable in order to drive the application of the most suitable analysis algorithms. The resulting BDD is referred to as Labelled BDD (LBDD). Exact values of the unavailability, expected number of failure and repair are calculated; the unreliability upper bound is automatically determined under given conditions. Several importance measures of basic events are also provided. From the LBDD a ZBDD embedding all MCS is obtained from which a subset of Significant Minimal Cut Sets (SMCS) is determined through the application of the cut-off techniques.

An important issue is related to the analysis of safety related systems according to the IEC 61508 international standard. In order to simplify the fault tree modelling and analysis a new component type has been defined allowing determining, for any configuration, the $PFD_{avg}$ and $PFH_{avg}$ values. The Staggered testing policy is also applicable besides the Sequential testing implicitly considered by the IEC standard.

# TABLE OF CONTENTS

# 1. INTRODUCTION

Fault Tree Analysis (FTA) is the most popular methodology for RAMS studies of complex systems; it allows to systematically describe the system's failure logic for each system failure state (Top event) and to quantify the corresponding occurrence probability / frequency. FTA is applied for system design review to prove that the system is reasonably safe and that it is well protected against both internal failures and external events.

Fault trees containing AND, OR Boolean operators are referred to as *Coherent,* and are characterised by monotonic (non-decreasing) functions with respect to the logical state of all basic events. Non monotonic logical functions, due to the presence of the NOT operator, are also of interest in system analysis. They are referred to as *not-coherent* and are very helpful in modelling e.g. the following types of problems:
- mutually exclusive events;
- event-tree sequences;
- top-events conditioned to the working state of one or more component / subsystem;
- safe maintenance procedures.

ASTRA-FTA allows the user to handle both coherent and non coherent fault trees.

ASTRA-FTA is based on the state of the art approach of Binary Decision Diagrams (BDD). This approach was introduced in the reliability field in the early nineteen. Today a vast literature is available on this subject; see e.g. Ackers (1978), Bryant (1986), Brace et al. (1990), Couder-Madre (1994), Rauzy (1993), Sinnamon-Andrews (1996), Rauzy-Dutuit (1997).

A BDD is a compact graph representation of Boolean functions. The main advantage of the BDD approach with respect to previous approaches is given by the possibility to obtain a compact graph embedding all system failure modes. On this graph it is possible to perform an exact probabilistic quantification. Then, from this graph, another graph can be derived embedding all Minimal Cut Sets (MCS), from which the Significant MCS can easily be extracted using the classical (probabilistic / logical) cut-off techniques. The greater efficiency of the BDD approach with respect to previous approaches is somewhat surprising. Trees analysed in the past with great difficulties have been analysed with the BDD approach in few seconds and without introducing any approximation. This is due to the compact representation of the fault tree and to the high efficiency of the algorithms working on the BDD.

In the case of non-coherent fault trees ASTRA dynamically assigns to each node of the graph a label that identifies the "local" type of the associated variable. This is done in order to drive the application of the most suitable analysis algorithms, since the complexity of these algorithms depends on the type of variable. The resulting BDD is referred to as Labelled BDD (LBDD).

This report describes the main calculation methods implemented in ASTRA for the logical and probabilistic analysis of coherent and non-coherent fault trees. After a brief description of the implemented fault tree analysis procedure, provided in the next section, the logical analysis methods, from the construction of the LBDD to the determination of the BDD embedding all MCS up to the extraction of the significant MCS are briefly described in section 3. Section 4 provides the equations for the quantification of the LBDD and of the Significant Minimal Cut Sets (SMCS).

# 2. OVERVIEW OF THE ASTRA 3.X ANALYSIS PROCEDURE

## 2.1 Types of operators

ASTRA allows the user to analyse fault trees containing the following set of operators: AND, OR, K/N, NOT, XOR and INH. Other operators such as NAND and NOR can be easily represented respectively as NOT-AND and NOT-OR.
Boundary Conditions (BC) can be assigned to basic events.
The first three operators are well known and will not be discussed further, while some comments will be made for the others and for boundary conditions.

The NOT operator allows the analyst to easily model complex failure logic, e.g.:
- top-events conditioned to the good state of one or more components/subsystems;
- sequences of event trees;
- safe maintenance procedures.

The XOR operator is used to model a relationship between $n \geq 2$ events such that one occurs and n-1 do not occur. Note that the representation of a mutually exclusive relationship using the XOR operator should not be confused with the mutual exclusivity of physically disjoint events, such as for instance the different failure states of a multistate component. Consider, for instance, two independent components connected in parallel. If we are interested in the probability of both components failed then we model the system using the AND operator. However, if we are interested in determining the probability of failure of only one component we use the XOR operator because components are independent. In other words both components, being independent, can also be both failed; however, we are not interested in this state, but rather in the probability that exactly one component is failed.

For non-coherent fault trees, represented as non-monotonic logical functions the concept of minimal cut set has to be replaced with that of *Prime Implicant* (PI), i.e. a combination made up by negated and not negated primary events, which are not contained in any other implicant. Non-monotonic functions are more complex to analyse than the monotonic ones.
However, by removing negated events from the set of PI and minimising the result, an approximated coherent form, expressed as a set of MCS, is obtained. This implicitly means to assign unit probability to negated events, which represents a condition frequently met in practice in safety applications. Hence, the probabilistic quantification gives accurate conservative results.
It follows that the need to consider negated events may be limited to the logical analysis only in order to remove impossible combinations, in which an event is present in both forms, negated and not negated. The advantages of applying this simplified approach are twofold, namely:
- Clearer interpretation of system failure modes;
- Significant reduction of computation time and working memory space.

ASTRA analyses non-coherent fault trees as follows:
1) It performs the construction of the LBDD and performs the exact probabilistic analysis;
2) It determines the BDD of the approximated coherent function;
3) It determines the SMCS;

The set of Significant MCS is calculated using the logical and/or probabilistic cut offs, i.e. in ASTRA prime implicants are not determined.

The INHIBIT (INH) operator is useful to model situations in which the output event occurs when the input event occurs and a conditional event is already verified. This operator presents only two input variables: the initiating event and the enabler event. The INH gate is applied to correctly quantify the occurrence probability of a particular type of sequential events.

In developing fault trees of, for instance, chemical installations, situations frequently encountered are those where the occurrence of an event (called initiating event) perturbs one or more system variables and places a demand for the protective system to intervene. The failure of the protection system allows the perturbation to further propagate in the plant which, eventually, may generate a dangerous situation or cause an accident. Obviously, a dangerous situation occurs only if the protective system is already failed when the initiating event occurs.

Consider, for instance, a pressurised tank, and the event "tank rupture" due to: 1) overpressure and 2) the failure of the automatic relief system. The rupture of the tank can occur only if the relief system is already failed at the time the overpressure occurs; the opposite situation is a sequence that may lead to other undesired events, e.g. production loss, but certainly not to the tank rupture.

Summarising:

1. the correct modelling of these types of events requires to account for the sequence of events, since a simple AND gate would give conservative (sometimes even too conservative) results (Demichela et al. 2003);
2. In ASTRA the initiating and enabling events are modelled by means of the INH gate;
3. The two inputs can also be sub-trees not necessarily independent; common event are automatically identified and treated as initiators.
4. Each MCS must contain at least one initiating event.

The INH gate as implemented in ASTRA is a particular type of sequential gate with two inputs in which the first to occur is characterised by its unavailability (enabler event) whereas the second (initiating event) by its unconditional failure frequency.

BOUNDARY CONDITIONS (BC) can be assigned to a subset of basic events. A boundary condition can assume only two values: good / failed, corresponding respectively to states 0, 1. Hence the analysis of a fault tree with boundary conditions allows determining the Top event occurrence probability conditioned to the state of one or more basic events. Events with BC can also be used as "House" events. Events with BC do not appear in any MCS since their value is used to properly remove them before starting the BDD construction.

## 2.2 Analysable fault trees

ASTRA has been developed for the analysis of both coherent and non-coherent fault trees. A description of the possible uses of non-coherent fault trees for safety and security applications can be found in Contini et al. (2004 and 2008).

In ASTRA basic events are classified as:

– *Positive* or normal, representing the failed state of a binary component;
– *Negated* or complemented, representing the working state of a binary component;
– *Double form*, when an event is present in the fault tree in both forms, positive and negated.

For instance, the function $\phi = \bar{a} b + \bar{a} c + b \bar{c}$ contains the negated variable $a$, the positive variable $b$ and the double form variable $c$.

This classification allows constructing a BDD, which is referred to as "Labelled BDD" (LBDD), in which all nodes are dynamically labelled with the variable type. This solution was adopted considering that the degree of complexity of the (logical and probabilistic) analysis algorithms depends on the type of variables.

Basic events are associated with the failed state of components that may be:

– Not repairable;
– On-line maintained;
– Tested/inspected;
– characterised by a constant probability

The distribution of time to failure and time to repair is exponential.

The particular model implemented for tested components allows ASTRA to conform to the IEC 61508 standard (2010 edition) for both low demand mode.

## 2.3 Fault tree analysis procedure

The analysis procedure implemented in ASTRA follows the main phases pictured in Figure 2.1.
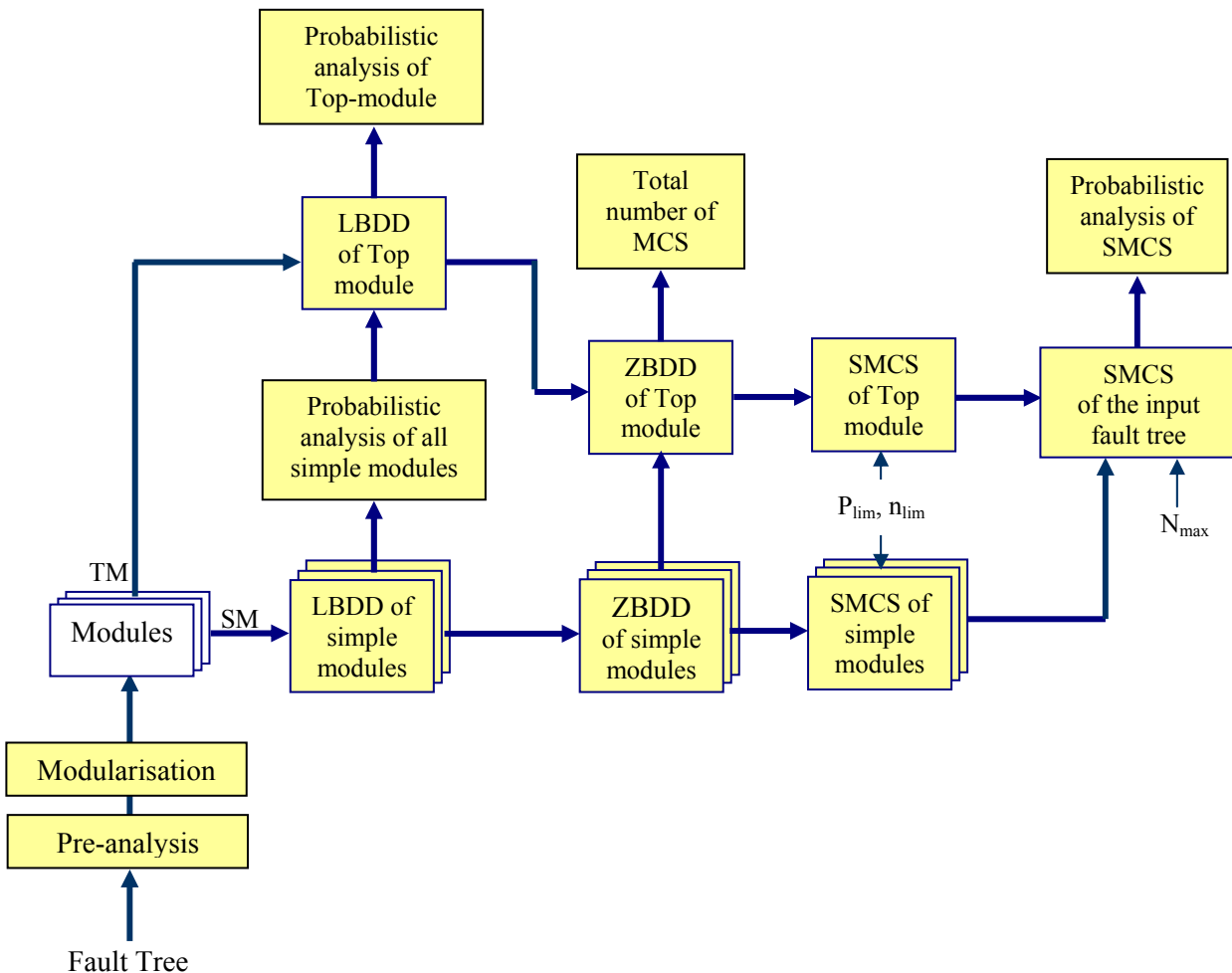


**Figure 2.1 Main phases of the analysis procedure implemented in ASTRA**

Initially a pre-analysis of the fault tree is performed. It consists of:
- complete check of the input data, to avoid analysing a wrong input data set;
- expansion of K/N and XOR gates into AND-OR-NOT equivalent expressions;
- transformation of INH gates into AND gates and labelling of enabling events;
- application of De Morgan rules.

Moreover, if the fault tree is non-coherent and the selected analysis option is "Approximated", then basic events that appear (one or more times and independently of their probability) in negated form only are removed from the fault tree, since they are not necessary for deleting impossible cut sets. Impossible cut sets may be generated because of the presence in the fault tree of events in both forms, positive and negated. This option is useful when dealing with complex fault trees of safety studies containing negated not repeated sub-trees. Indeed, removing unrepeated events allows reducing the fault tree dimension and the computational effort. The impact of such a simplification on the probabilistic results is negligible for safety applications, since negated events have probability very close to 1. Hence, from the probabilistic point of view the removal of unrepeated negated events is equivalent to set their probability to 1.

Then the input fault tree is modularised i.e. it is decomposed into a set of "Simple Modules, SM" and a "Top-module, TM".

A simple module is a sub-tree containing basic events that are not replicated in any other module; however, basic events can be replicated within a simple module.

The Top-module is the module that contains the top event definition and generally is the most complex one; it contains simple modules as fictitious basic events. If the Top-module is non-coherent, all simple modules appear in positive form.

All modules, being independent, are independently analysed.

Simple modules are examined first. Each of them is stored in the form of LBDD. The basic concepts of the LBDD method are briefly described in the next section. The results of the probabilistic quantification of simple modules are used to quantify the Top-module, which is also stored as LBDD. The following parameters are determined for each simple module in its normal or positive form: Unavailability, and Birnbaum importance of basic events.

The quantification of the LBDD of the Top-module allows obtaining the exact values of the Top-event Unavailability, Expected number of failure, characteristic times, and various importance measures of basic events for both coherent and non-coherent fault trees. Moreover the upper bound for the Unreliability and the mean time to first failure are determined when the "Time dependent analysis" option is selected. The equations applied are described in Section 4.

The analysis proceeds with the independent determination of the ZBDD of simple modules and of the Top module. The ZBDD (Minato, 1990) is a compact graph embedding MCS or Prime Implicants (PI). For non coherent fault trees Prime Implicants (PI) should be determined due to the presence of negated events. Since generally a PI contains many negated events (representing working states) and few normal events (representing failed states), when the probability of negated events is very close to 1 it is more convenient to determine the MCS by removing negated events.

In practice all negated events are removed from the LBDD; the resulting MCS are stored, after minimisation, as a ZBDD. At this stage the ZBDD contain simple modules as events.

Given that the probabilistic analysis has already been performed on the LBDD, the determination of MCS can be limited to the most important ones, which in this report are referred to as Significant MCS (SMCS). The SMCS are determined for all simple modules and for the Top module by setting up the thresholds on the order and unavailability of MCS (cut-off values):

– If the *Logical cut-off* $n_{lim}$ is applied a MCS is retained if its order $m \leq n_{lim}$;
– According to the *Probabilistic cut-off* $P_{lim}$ a MCS is retained if its probability $Q \geq P_{lim}$.

As above mentioned simple modules are represented in the ZBDD as fictitious basic events; the significant minimum failure combination of the ZBDD, referred to as Macro MCS (MMCS) do not represent the SMCS of the input fault tree. A further step is necessary to determine the SMCS.

For instance the j-th MMCS say $C_j$ of the Top-module can be represented as:

$$C_j : \bigcap_{i=1}^{w} BM_i \bigcap_{k=1}^{r} BE_k$$

where $w$ and $r$ are respectively the number of simple modules and basic events making up the j-th MMCS; $BM_i$ represents a generic simple module; $BE_k$ denotes a generic basic event. Depending on $w$ and on the number of its failure combinations, $C_j$ may indeed contain a large number of MCS. If $m_i$ is the number of combinations contained in the generic simple module $BM_i$, then the number of MCS of

the input tree embedded in $C_j$ is equal to $\prod_{i=1}^{w} m_i$

An efficient algorithm has been implemented to extract the SMCS by setting up the threshold value $N_{max}$. The *cut off $N_{max}$* defines the maximum number of SMCS to be extracted from a single MMCS. The use of this cut-off, at the end of the analysis of each MMCS, implies the automatic modification of $P_{lim}$ to a value equal to the probability of the least important SMCS extracted. This new $P_{lim}$ value will then be used for the analysis of the next MMCS, and so on. The result is the set of the SMCS of the input tree: their number is generally close to $N_{max}$.

The final phase of the analysis is the determination of the probabilistic parameters of interest, i.e. unavailability, unconditional failure and repair frequencies, expected number of failures and repairs and unreliability for all SMCS.

To summarise, the analysis procedure can be subdivided in two parts:
1) Construction of the LBDD for all modules, exact probabilistic analysis performed on the LBDD and construction of the ZBDD;
2) Use of the cut-off technique for the determination of the SMCS.

ASTRA 3.x represents a significant improvement with respect to version 2, in which .the probabilistic analysis was performed on the set of SMCS.
The software has been developed by G. de Cola from INFOCON under JRC contract.

# 3. BDD-BASED LOGICAL ANALYSIS

The aim of this section is to describe the concept of LBDD and how it is generated. A more detailed description can be found in Contini et al (2006) and Contini-Matuzas (2010).

## 3.1 Classification of variables in non-coherent fault trees

The Boolean function describing the logical relationships among events in fault trees can be monotonic or not monotonic, also commonly referred to as *Coherent* or *Non-Coherent*.

The binary function $\Phi(\mathbf{x})$ of a non-coherent fault tree contains three different types of basic events or binary variables, namely:
1. normal or positive, e.g. x;
2. negated, e.g. $\bar{y}$;
3. events appearing both in positive and negated forms, e.g. z, $\bar{z}$.

In this report the following definitions are used. Variables of type 1 are referred to as *Single form Positive variables* (SP), variables of type 2 as *Single form Negated variables* (SN), whereas variables of the third type as *Double Form variables* (DF).

For instance, in the function $F = a\,\bar{b} + \bar{a}\,c$, the variable *a* is of DF type, *b* is of SN type and *c* of SP type.

Let us consider the following section of a BDD, where Y and Z are the two functions having the variables y and z as roots:
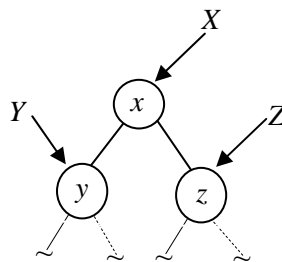


This partial BDD may represent different logical functions, i.e. $X = x\,Y + Z$, or $X = x\,Y + \bar{x}\,Z$ depending on the type of x. In the first function x is of SP type, whereas in the second it is of DF type.

Different types of variables require different algorithms of analysis. Indeed, on nodes with DF variables the determination of the Prime Implicants (PI) and of the failure and repair frequencies require the logical intersection between the left and right descending functions, whereas this is not needed for the other two types (SP, SN) of variables.
Three different types of variables require the labelling of two out of them. In ASTRA variables of SN type are labelled with the symbol $; variables of DF type are labelled with the symbol &.

The information about the type of variables can easily be extracted from the input fault tree and associated to the nodes on the BDD. We shall call this method as "Static Labelling", since the association *node-variable* with the *variable-type* is made after the construction of the BDD. Odeh-Limnios (1996) applied this technique.

However, we can observe that in a BDD a DF variable may be associated with two or more nodes in which it behaves as a positive (SP) or as a negative (SN) variable, the analysis of which require simpler algorithms.

Consider for instance the function $F = a \bar{b} + \bar{a} c$. Using the ordering a<b<c we get the LBDD in Figure 3.1.a in which "a" is represented as a DF variable. Using the inverse ordering c < b < a we get another LBDD in which "a" is represented once as SP and once as SN.
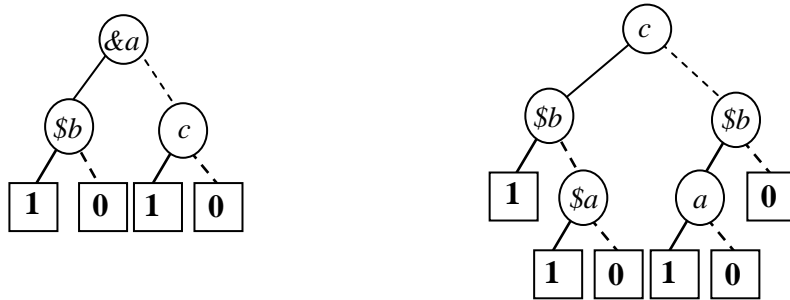


**Figure 3.1 Labelled BDD of the function $F = a \bar{b} + \bar{a} c$ with two different ordering**

According to Figure 3.1.a F can be written, in terms of labelled variables, as $F = a \$b + \overline{\&a} \, a \, c$, from which the following equivalencies can be derived: $\&a = a$, $\overline{\&a} = \bar{a}$, $\$b = \bar{b}$.

The labelling technique applied during the construction of the BDD can be used to characterise each node with the type of the associated variable.

## 3.2 Construction of an LBDD - example

This example is taken from Liu-Pan (1990). Let $\phi(x) = x_2 (x_1 + \bar{x}_3 + \bar{x}_4) + x_3 ( \bar{x}_1 + \bar{x}_2 x_4)$ be the non monotonic function, containing four variables of DF type. Considering the ordering $x_2 < x_1 < x_3 < x_4$ the LBDD is represented in Figure 3.3, which has been obtained as follows.
First of all negated variables are labelled with $, giving:
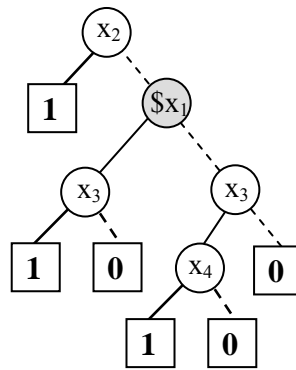
$\phi(x) = [x_2 (x_1 + \$x_3 + \$x_4)] + [x_3 (\$x_1 + \$x_2 x_4)]$.

The LBDD for the two functions are as follows:
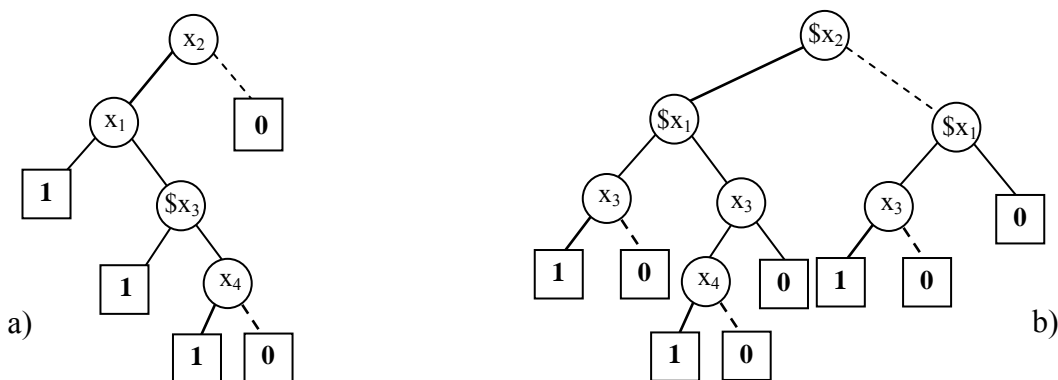


**Figure 2.2 LBDD of the two terms of $\phi(x)$: a) $F = x_2 (x_1 + \$x_3 + \$x_4)$; b) $G = x_3 (\$x_1 + \$x_2 x_4)$**

In Contini-Matuzas (2010) the steps for combining these two LBDD are described. The resulting LBDD is represented in Figure 3.3.

**Figure 2.3 LBDD representation of Top = $x_2$ ($x_1$ + $\overline{x_3}$ + $\overline{x_4}$) + $x_3$ ( $\overline{x_1}$ + $\overline{x_2}$ $x_4$)**

Comparing the LBDD with the BDD the following considerations can be drawn:

-  Dynamic labelling and static labelling have the same number of nodes;

–  The representation of the negated $\overline{x}$ variables as $x allows applying the same algorithms to nodes with SP and SN variable types. Hence we may call SP and SN variables as coherent (i.e. x as monotonic not decreasing; $x as monotonic not increasing) and DF as non-coherent.

–  Variables $x_2$, $x_3$ and $x_4$ are represented in the LBDD as SP variables and only $x_1$ is represented as SN variable. Therefore, in spite of the fact that all variables in the fault tree are of DF type, none of them is represented as such in the LBDD. The number of nodes with DF variables depends on the variable's ordering;

–  The determination of the prime implicants for the BDD requires, for each node, the intersection between the left and right descendants, whereas this is not necessary on the LBDD in Figure 3.3. The same consideration can be applied for the determination of the unconditional failure and repair frequencies.

–  The absence of DF variables in Figure 3.3 assures that all implicants are embedded in the LBDD. This can easily be explained by observing that the Consensus operation, i.e. x y + $\overline{x}$ z = y z is never applied. The Prime Implicants set is found as if the BDD was coherent, i.e. {PI} = {($x_2$) ($x_1$ $x_3$) ($x_3$ $x_4$)}, which is equivalent to {PI} = {($x_2$) ($\overline{x_1}$ $x_3$) ($x_3$ $x_4$)}.

On the LBDD so obtained the probabilistic analysis is performed. The implemented equations are described in Section 4.

## 3.3 Construction of the ZBDD from the LBDD

After the quantification of the LBDD the next step of the analysis is the determination of Minimal Cut Sets (MCS). If the fault tree is not coherent then the negated events must be properly removed from the LBDD giving a new BDD containing all MCSs. This new data structure is referred to as ZBDD.
The removal of negated events is justified by observing that negated variables, in many applications of non-coherent fault trees, e.g. safety analysis, represent the working state of components, having success probability very close to 1.
In practice the reliability analyst is interested in failed components, i.e. in MCS also in the case of non coherent trees.

Using the LBDD the ZBDD can be obtained by:

- Removing the & labels (&x → x);
- Deleting SN variables ($ labelled) by performing the logical OR between the two descending functions.

The elimination of the "$" labels has been implemented as follows.
Let ite($x, F, G) be the node under examination, F = ite(z, $F_1$, $F_0$) its left descendant and G = ite(w, $G_1$, $G_0$) its right descendant. The following relationships are applied:

if z < w then ite($x, F, G) = ite(z, $F_1$, $F_0 \vee G$)

if z > w then ite($x, F, G) = ite(w, $G_1$, $G_0 \vee F$)

if z = w then ite($x, F, G) = ite(z, $F_1 \vee G_1$, $F_0 \vee G_0$)

These operations are then followed by the reduction and minimisation rules typical of monotonic functions. The result is a ZBDD embedding all minimal MCS.

As an example consider again the LBDD in Figure 3.3. To obtain the ZBDD embedding all MCS it is sufficient to remove the $x_1$ node.
In this case (z = $x_3$; w = $x_3$) we have: $F_1 = 1$; $F_0 = 0$; $G_1 = $ ite($x_4$, 1, 0) and $G_0 = 0$.
Since both descending nodes have the same variable $x_3$, i.e. z = w, then:

ite($x_3$, $F_1 \vee G_1$, $F_0 \vee G_0$) = ite($x_3$, $1 \vee$ ite($x_4$,1,0), $0 \vee 0$) = ite($x_3$, 1, 0).

The resulting ZBDD is represented in Figure 3.4, in which the MCS are {($x_2$) ($x_3$)}.
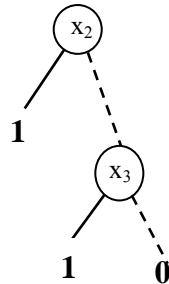


**Figure 2.4. Results of the transformation of the LBDD of Figure 3.3 into the BDD embedding all MCS**

# 4. BDD-BASED PROBABILISTIC ANALYSIS

The quantification of the LBDD allows obtaining the exact values of Unavailability, Expected number of failures and repair and components' importance measures. Moreover the Unreliability upper bound and the MTTFF can be calculated when the unconditional Top-event frequency is not constant. This section provides information about the equations used together with some simple clarification examples. Other examples of application of the probabilistic analysis methods implemented in ASTRA 3.0 are described in the test case report (Contini-Matuzas, 2009). In this section the probabilistic equations will be described for basic event, for the Top event and for minimal cut sets.

## 4.1 Notation.

$\lambda$         Failure rate (constant)
$\mu$         Repair rate (constant)
$\tau$         Repair time ($\tau = 1 / \mu$)
$\theta$         Time between tests
$\theta_0$         First time to test
$t_d$         Test duration
$\omega(t)$         Unconditional failure frequency
$\upsilon(t)$         Unconditional repair frequency
$q(t)$         Basic event unavailability at time t
$q(0)$         Basic event unavailability at time t=0
$\Lambda_T(t)$         Top event conditional failure frequency
$Q_T(t)$         Top event Unavailability at time t
$Q_T(0)$         Top event Unavailability at t=0
$W_T(t)$         Top event Expected number of failures in 0-t
$V_S(t)$         Top event Expected number of repair in 0-t
$F_T(t)$         Top event Unreliability in 0- t
$Q_{Cj}(t)$         Top event Unavailability at time t for the j-th MCS
$Q_{Cj}(0)$         Unavailability at t=0 for the j-th MCS
$W_{Cj}(t)$         Expected number of failures in 0-t for the j-th MCS
$F_{Cj}(t)$         Unreliability in 0- t for the j-th MCS
DC         Detection Coverage
PTC         Proof Test Coverage
EUC         Equipment Under Control
MTBF         Mean Time Between failures
MTTR         Mean Time To Repair
MTTF         Mean Time To failure
MTTFF         Mean Time To First Failure
BE         Basic event
MCS         Minimal Cut Set
SMCS         Significant MCS
Ne         Number of basic events of the fault tree
n         Number of basic events in an MCS/SMCS
T         Mission time
$p^f_x(t)$         Probability of failure critical state for the generic event x
$p^r_x(t)$         Probability of repair critical state for generic event x
$IC_x(t)$         Criticality index at of event x time t
$RAW_x(t)$ Risk Achievement Worth of event x at time t
$RRW_x(t)$ Risk Reduction Worth of event x at time t
$IS_x$         Structural criticality of event x

## 4.2 Probabilistic quantification of basic events

A basic event (BE) represents the failure mode of a component. The terms "*basic event*" and "*component failure*" are synonymous. BEs are binary and statistically independent, e.g. the failure or repair of a component does not have any influence on the failure probability of any other component; the unique exception to the independence is the sequence of events that is considered in the INH gate.

Failure and repair times are exponentially distributed, i.e. failure and repair rate are constant.
− Since $\lambda$ is constant, then the mean time to failure MTTF = 1 / $\lambda$.
− Since $\mu$ is constant, then the mean time to repair MTTR = 1 / $\mu$.
− The repair makes the component as good as new.

ASTRA allows the use of different types of basic events, whose parameters are given in Table 4.1.

**Table 4.1 Basic events and associated parameters**

| Event type | $\lambda$ | $\tau$ | q | $\theta$ | $\theta_0$ | PTC | DC | notes |
|---|---|---|---|---|---|---|---|---|
| Not repairable | • | | • | | | | | |
| On-line maintained | • | • | • | | | | | |
| Acting on demand | | | • | | | | | |
| Periodically tested | • | • | | • | • | • | • | $0 \leq DC < 1$ <br> $0 \leq PTC \leq 1$ |

Parameters DC and PTC are requested for modelling components according to the IEC 61508 standard.
The parameter DC (Detection Coverage) represents the probability that the failure is on-line detected. DC<1 means that the diagnostic system is not able to reveal all failures; consequently the component must also to be tested to identify undetected failures.
The parameter PTC (Proof Test Coverage) represents the probability that the test is able to reveal all failures. PTC<1 means that the test is not complete, i.e. it does not reveal all failures; hence (1-PTC) represents the probability of undetected failures, which will remain hidden until the end of the mission time.

### 4.2.1 Unavailability of basic events

The time specific unavailability Q(t) of an item (component, subsystem, system) is the probability that the item is failed at time t.

### I. Not repairable components.

These are components that, in case of failure, cannot be repaired in the mission time interval either because their failure cannot be revealed or because they are not accessible during the mission.

These components are characterized by their failure rate $\lambda$.

The failure rate is defined as the "*probability that the component fails in the time interval t-t+dt given that it never occurred from 0 to t*".

The unavailability of the component at time t=0 may also be different from 0.

$$q(t) = 1 - e^{-\lambda t} + q(0) \, e^{-\lambda t} \tag{1}$$

The following plot, produced by the ASTRA chart capability, represents the unavailability of three non repairable components with different failure rate.

With constant failure rate it can be proved that MTTF = 1 / λ.
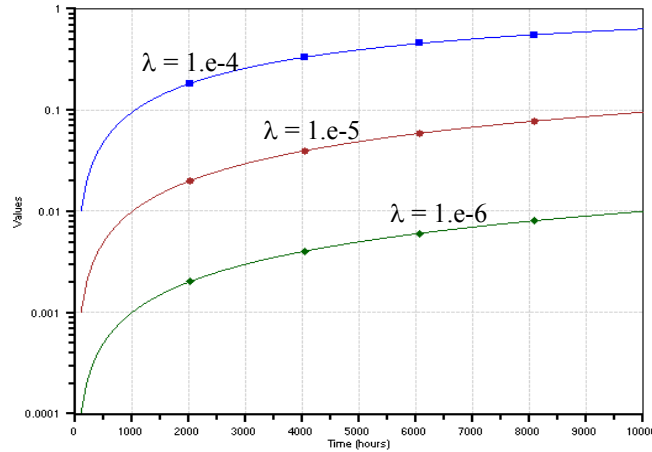


**Figure 4.1 Unavailability of a non-repairable component for different failure rates**


## II. On-line maintained components

The basic hypotheses are that:

−    the failure is revealed with probability 1; and

−    the repair process immediately starts as soon as the component fails.

The repair, which is preformed during the mission time, i.e. while the system works, makes the component as good as new. The required parameters are λ and τ (MTTR). The unavailability of the component at time t = 0 may also be different from 0.

$$q_D(t) = \frac{\lambda}{\lambda + \mu}(1 - e^{-(\lambda+\mu)\,t}) + q(0)\,e^{-(\lambda+\mu)\,t} \tag{2}$$

The following plot represents the unavailability of a repairable component with λ = 1.e-5 and for different values of the mean repair time τ.

Equation 2 tends to the steady-state value $q_\infty = \dfrac{\lambda}{\lambda + \mu}$ after about 4 times the value τ.
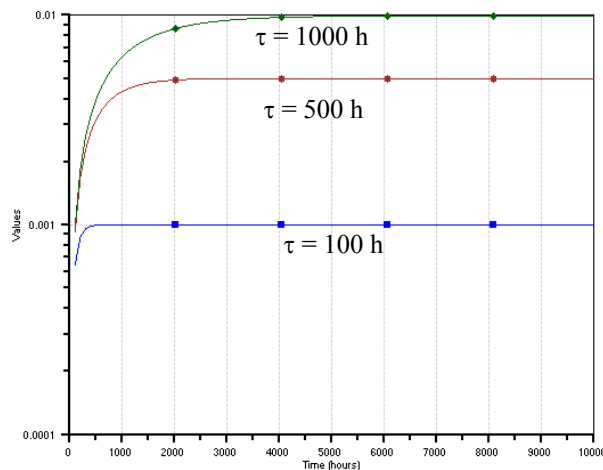


**Figure 4.2 Unavailability of a repairable component for different repair times**


## III. Periodically tested components

In these types of components the failure is not revealed. These are the typical components of safety systems whose failure can be revealed only through test/inspections.

The required parameters are $\lambda$ and $\theta$ (inspection interval), $\theta_0$ (first inspection time), and $\tau$ (mean repair time).

The hypothesis adopted in this model is that the test does not fail the component and that the unavailability due to test is negligible compared with the unavailability between tests.

In ASTRA this model has been enriched with the addition of two parameters to consider the analysis of a safety system according to the standard IEC 61508:
- Detection Coverage DC; and
- Proof Test Coverage PTC.

Moreover, according to the standard, the failure of a component can be "safe" or "dangerous". The total failure rate is therefore $\lambda = \lambda_D + \lambda_S$, where the subscripts D means detected and U undetected.

The dangerous failure rate $\lambda_D$ is further subdivided into $\lambda_{DD}$ and $\lambda_{DU}$,

- Dangerous Undetected $\lambda_{DU} = \lambda_D(1\text{-DC})$

- Dangerous Detected $\lambda_{DD} = \lambda_D\,DC$.

Hence the unavailability of a tested component in which the test is perfect and able to reveal all failures is given by:

$$q(t) = q_U(t) + q_D(t)$$

where $q_D(t)$ and $q_U(t)$ are the contributions to unavailability respectively due to undetected and detected failures

To account for the fact that the test does not reveal all failures the IEC 61508 standard introduced the Proof Test Coverage (PTC) parameter. PTC < 1 means that the proof test does not allow revealing all failures; hence (1-PTC) represents the fraction of failures that remain hidden until the end of the mission.

Therefore, the unavailability of a tested component is given by:

$$q(t) = q_U(t) + q_D(t) + q_N(t) \tag{3}$$

where $q_N(t)$ is the contribution to unavailability due to failures that remain hidden due to non-complete test.
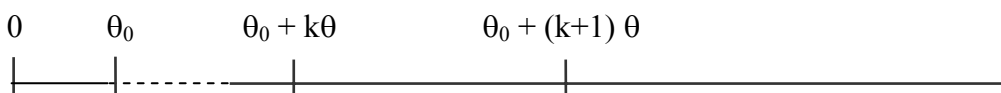
The different unavailability equations implemented in ASTRA are described below.

*Determination of $q_U(t)$.*

In this model $\lambda = \lambda_D(1\text{-DC})\,PTC$

The applied unavailability equation depends on the value of the repair time $\tau$ compared to the test interval $\theta$.

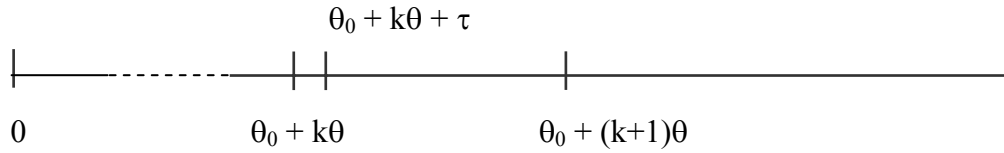1) The repair time $\tau$ is <u>negligible</u>, i.e. $\tau < 10^{-3}\,\theta$ and $\theta_0 \neq \theta$



for $0 \leq t < \theta_0$ then $q(t) = 1 - e^{-\lambda t}$ $\tag{4}$

for $\theta_0 + k\theta \le t < \theta_0 + (k+1)\,\theta$ and $k = 0,1,2,\dots$

$$q_U(t) = 1 - e^{-\lambda(t-\theta_0 + k\,\theta)} \tag{5}$$

2) The repair time $\tau$ is <u>not negligible</u>, i.e. $\tau \ge 10^{-3}\,\theta$

When the test reveals that the component is failed, the basic hypothesis in this case is that the repair starts immediately after the test and lasts for $\tau$ time units. The repair makes the component as good as new.

$$\theta_0 + k\theta + \tau$$

```
|————————————————|  |————————————|—————————————————————————
0              θ₀ + kθ        θ₀ + (k+1)θ
```

The unavailability is given by:

for $0 \le t < \theta_0$ then $q_U(t) = 1 - e^{-\lambda t}$

for $\theta_k^* \le t < \theta_k^* + \tau$ then $q_U(t) = q(\theta) + (1 - q(\theta))\,(1 - e^{-\lambda(t-\theta^*)})$  (6)

with $q(\theta) = 1 - e^{-\lambda\theta}$

for $\theta_k^* + \tau \le t < \theta_{k+1}^*$ then $q_U(t) = 1 - e^{-\lambda(t-(\theta_k^* + \tau))}$  (7)

where $\theta_k^* = \theta_0 + k\theta$ and $\theta_{k+1}^* = \theta_0 + (k+1)\theta$ with $k = 0,1,2,\dots$

The following plot shows the unavailability of a tested component with $\lambda$ = 1.e-4, DC = 0, test interval of 250 h, and with negligible – not negligible repair time $\tau$.
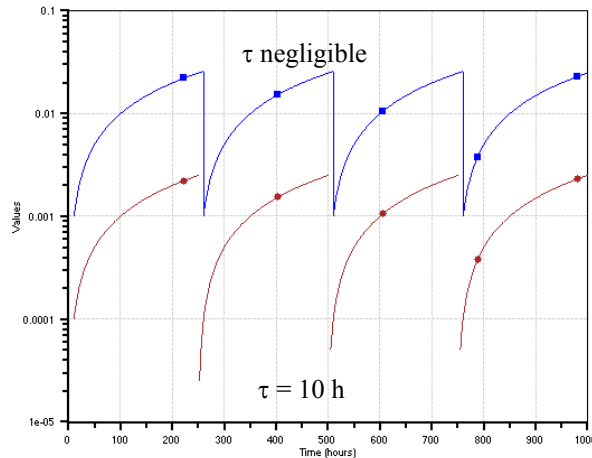


**Figure 4.3 Unavailability of a tested component for different repair times**

*Determination of $q_D(t)$.*

<u>In this model $\lambda = \lambda_D\,DC$</u>

The model is the same as the on-line maintained with $q(0) = 0$, i.e.:

$$q_D(t) = \frac{\lambda}{\lambda + \mu}(1 - e^{-(\lambda + \mu)\,t}) \tag{8}$$

Determination of $q_N(t)$.
In this model $\lambda = \lambda_D (1-DC)(1-PTC)$

The model is the same as the one for non-repairable component with $q(0) = 0$, i.e:
$$q_P(t) = 1 - e^{-\lambda t} \tag{9}$$

Finally, the $PFD_{avg}$ of a single component (1oo1) is given by the mean value of the unavailability $q(t)$:

$$PFD_{avg} = \frac{1}{T} \int_0^T [q_U(t) + q_D(t) + q_N(t)]\, dt \tag{10}$$

As an example the following table compares the content of Table B.2 of the standard (Vol. 6, page 36) with the results of ASTRA.
The small difference between IEC and ASTRA is due to numerical approximations and to the different equations used: ASTRA applies exact equations whereas IEC uses simplified conservative equations.

**Table 4.2 Comparison of ASTRA with IEC 61508: $PFD_{avg}$ for a single channel in low demand mode**

| Architecture | DC | IEC 61508 $\lambda_D = 0.5e\text{-}05$ | ASTRA $\lambda_D = 0.5e\text{-}05$ |
|---|---|---|---|
| 1oo1 | 0% | 1.1e02 | 1.09e-2 |
| | 60% | 4.4e-3 | 4.39e-3 |
| | 90% | 1.1e-3 | 1.13e-3 |
| | 99% | 1.5e-4 | 1.49e-4 |
| Note 1: the test period considered is six months (4380 h). | | | |
| Note 2: the mission time considered is equal to 10 times the proof test interval | | | |
| Note 3: PTC is assumed equal to 1. | | | |

IV. Components acting on demand
The unavailability is given by $q(t) = q(0) = $ const.

### 4.2.2 Unconditional failure and repair frequencies of basic events

The time specific unconditional failure frequency $\omega(t)\, dt$ is defined as:
*the probability that the component fails in (t,t+dt) given that it was working at time 0.*

It is important not to confuse the unconditional failure frequency with the failure rate. In the case of $\omega(t)$ the component was good at t=0 and may have failed before t, whereas $\lambda(t)$ requires that the component has never failed between 0 and t.
In the exponential case it can be proved that the following equation holds (Kumamoto-Henley):

$$\omega(t) = [1 - q(t)]\,\lambda$$
where $q(t)$ is the component unavailability and $\lambda$ the constant failure rate.

The time specific unconditional repair frequency $\nu(t)$ is defined as:
*the probability that the item is repaired in (t,t+dt) given that it was working at time 0.*
In the exponential case ($\lambda$, $\mu$ constants) it can be proved that the following equation holds:

$$v(t) = q(t)\,\mu$$

where $q(t)$ is the component unavailability and $\mu$ the repair rate ($h^{-1}$).

For non-repairable components $v(t) = 0$ since $\mu = 0$. Also events characterised by means of a constant unavailability value have $\omega(t) = 0$ and $v(t) = 0$.

For tested components $\omega(t) \cong \lambda$ and $v(t) \cong \lambda$.

The following figure shows the failure and repair frequencies of a repairable component.
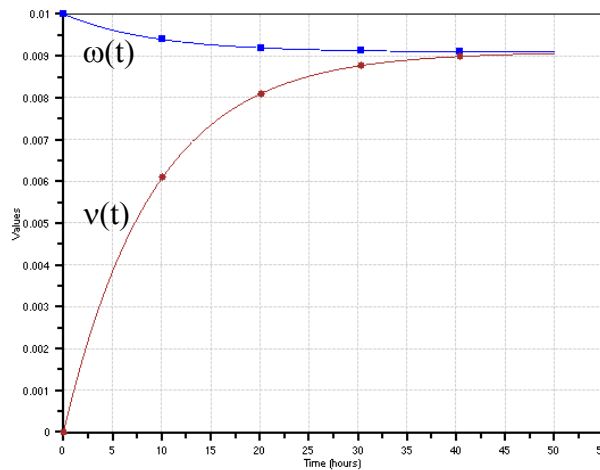


**Figure 4.4 Unconditional failure and repair frequencies of a generic repairable component**

The failure frequency decreases and the repair frequency increases following the variation of the unavailability with time. They tend to the constant value $\omega_\infty = v_\infty = \dfrac{\lambda\,\mu}{\lambda + \mu}$ when the unavailability reaches the steady state condition.

The standard IEC 61508 considers, in the case of high demand or continuous mode of operation, the Probability of Failure per Hour $PFH_{avg}$. This is the unconditional failure frequency $\omega(t)$ previously defined.
$PFH_{avg}$ is given by:

$$PFH_{avg} = \omega_{avg} = \frac{1}{T}\int_0^T \omega(t)\,dt = \frac{W(T)}{T} \qquad (11)$$

where:
$$\omega(t) = PTC\,\omega_U(t) + (1 - PTC)\,\omega_N(t) \qquad (12)$$
$\omega_U(t) = (1-q_U(t))\,\lambda_D\,(1- DC)$
$\omega_P(t) = (1-q_N(t))\,\lambda_D\,(1- DC)$
$W(T)$ is the expected number of failures at the mission time $T$

In equation (12) the failure frequency due to revealed failures is zero since the hypothesis adopted in IEC 61508 (Vol. 6, Appendix B page 43) states that the *revealed faults put the system in a safe*

*condition*. Hence the unrevealed dangerous failure of a safety system may occur between tests only; however, if PTC < 1, there is a probability that the failure remains hidden until the end of the mission.

For a single component (1oo1) Table 4.3 shows that the ASTRA results are equivalent to those of the standard.

**Table 4.3 Comparison of ASTRA with IEC 61508: PFH$_{avg}$ for a single channel in high demand mode**

| Architecture | DC | IEC 61508 $\lambda D = 5.0e\text{-}6$ | ASTRA frequency $= \dfrac{W(T)}{T}$ |
|---|---|---|---|
| 1oo1 | 0% | 5.0e-06 | 4.94e-06 |
| | 60% | 2.0e-06 | 1.98e-06 |
| | 90% | 5.0e-07 | 4.94e-07 |
| | 99% | 5.0e-08 | 4.94e-08 |
| Note 1: the test period considered is six months (8760 h). MTTR = 8 h. $\beta$ = 10% | | | |
| Note 2: the mission time considered was 10 times the proof test interval. | | | |
| Note 3: PTC is assumed equal to 1. | | | |

## 4.3 System Unavailability analysis

The quantification of the Top-event is articulated in the following steps:

Analysis of all basic events by means of the equations previously described
Analysis of simple modules
Analysis of the Top-module

In this section the equations for unavailability analysis are described together with simple examples. The analysis methods are applied first to all simple modules and then to the Top-module, all represented as LBDD.

The following figure represents the generic node of an LBDD; the associated variable x can be of any type (SP, SN, DF).
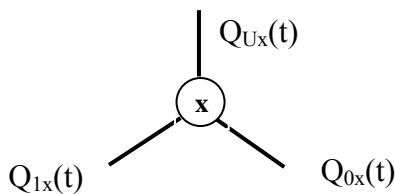


Independently of the variable type the exact value of the unavailability $Q_{Ux}(t)$ is given by:

$$Q_{Ux}(y) = q_x(t)\, Q_{1x}(t) + [1 - q_x(t)]\, Q_{0x}(t) \qquad (13)$$

$Q_{1x}(t)$ and $Q_{0x}(t)$ are respectively the unavailability of the left and right branches of the node and $q_x(t)$ is the unavailability of the event x.
Equation (13) is recursively applied to all nodes of the LBDDs of all simple modules and of the Top-module by visiting them according to the Depth-first mode (Bottom-up approach).
For terminal nodes:
Node 1: $Q_1(t) = 1$;
Node 0: $Q_0(t) = 0$.

NOTES:

The unavailability Q(t) is calculated as a function of time due to the need to correctly take into account the discontinuities of the unavailability function due to the presence of tested components.

Calculating Q(t) simply means that equation (13) is applied to all LBDD nodes at least as many times as the number of time points in which the unavailability function has discontinuities.

If all components have unavailability at t = 0 different from zero, then the top event unavailability Q(0) > 0.

The mean value of the unavailability is important when the system unavailability function contains discontinuities due to the presence of tested components. In these cases in fact the unavailability at the mission time T can be misleading (value higher or lower than the mean). Hence, if tested events are present ASTRA calculates, esides the unavailability at mission time T, i.e. $Q_{Top}(t)$ for $0 \le t \le T$, also

the mean value $Q_{Tmean} = \dfrac{1}{T} \displaystyle\int_0^T Q_{Top}(\tau)\, d\tau$ and the peak value $Q_{Tmax}$.

*Example 1*

As an example of the application of the above equations consider the determination of the unavailability of the following function, Top = $[x_2 ( x_1 + \overline{x}_3 + \overline{x}_4 )] + [x_3 ( \overline{x}_1 + \overline{x}_2 x_4 )]$, containing all variables of DF type. For this system the prime implicants are: $(x_2)$, $(\overline{x}_1 x_3)$, $(x_3 x_4)$.

Basic events' data are shown in Table 4.4 and their plot vs. time is given in Figure 4.6.
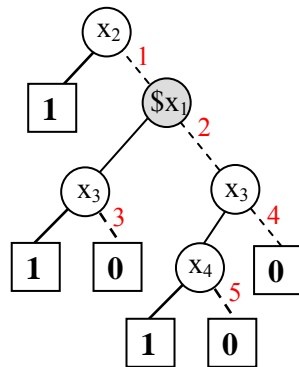


**Figure 4.5 LBDD of the function Top = $[x_2 (x_1 + \overline{x}_3 + \overline{x}_4 )] + [x_3 ( \overline{x}_1 + \overline{x}_2 x_4 )]$**

**Table 4.4 Data characterising the basic events of the LBDD represented in Figure 4.5.**

| Var | Type | λ | τ | q |
|-----|------|---|---|---|
| $x_1$ | Not repairable | 1.e-5 | | |
| $x_2$ | On-line maintained | 1.e-6 | 200 | |
| $x_3$ | On-line maintained | 1.e-4 | 20 | |
| $x_4$ | On demand | | | 0.001 |

The expressions of the unavailability for the different nodes of the BDD are given in Table 4.5 (bottom up visit). The first column contains the number of the node represented in figure 4.5; the sequence from the first row to the last row represents the LBDD visiting order. The last column of the last node contains the expression of the Top event unavailability.

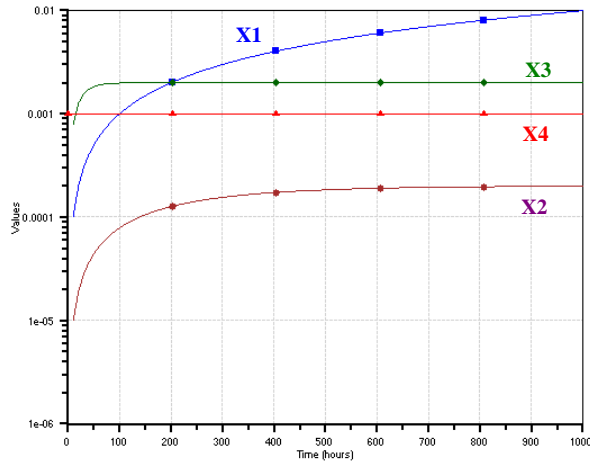Figure 4.7 represents the unavailability of the function in fig. 4.5.

**Figure 4.6 Components' unavailability for the example in Figure 5.1**

**Table 4.5 Nodes- unavailability expression for the BDD in Figure 5.1**

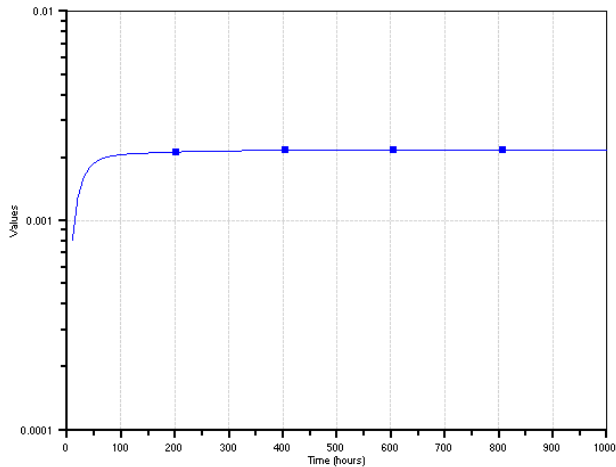| Node | Var | $q_x$ | $Q_{1x}$ | $Q_{0x}$ | $Q_{tot} = q_x Q_{1x} + p_x \ Q_{0x}$ |
|------|-----|-------|----------|----------|----------------------------------------|
| 3 | $x_3$ | $q_3$ | 1 | 0 | $q_3$ |
| 5 | $x_4$ | $q_4$ | 1 | 0 | $q_4$ |
| 4 | $x_3$ | $q_3$ | $q_4$ | 0 | $q_3 q_4$ |
| 2 | $\$x_1$ | $p_1$ | $q_3$ | $q_3 q_4$ | $p_1 q_3 + q_1 q_3 q_4$ |
| 1 | $x_2$ | $q_2$ | 1 | $p_1 q_3 + q_1 q_3 q_4$ | $q_2 + p_2 (p_1 q_3 + q_1 q_3 q_4)$ |



**Figure 4.7 Top event Unavailability for the example in Figure 5.1.**

*Example 2*

The second example deals with the determination of the unavailability of a 2 out of 3 system made up by equal components: Top = a b + a c + b c.
Components are characterized by the data provided in Table 4.6.

**Table 4.6  Data of basic events in Top = a b + a c + b c.**

| Event | $\lambda$ | $\tau$ | q | $\theta$ | $\theta_0$ | PTC | DC |
|-------|-----------|--------|---|----------|------------|-----|-----|
| a | 1.e-4 /h | 0.6h | 0 | 300h | 0 | 0 | 0 |
| b | 1.e-4 /h | 0.6h | 0 | 300h | 100h | 0 | 0 |
| c | 1.e-4 /h | 0.6h | 0 | 300h | 200h | 0 | 0 |

Components are tested one after the other at regular intervals of time (staggered testing policy).
If $\theta$ is the test interval, then the first component is tested at t = 0   (first test at $\theta_0 = 0$),    the second at t = $\theta$ / 3 ($\theta_0 = 100$) and finally the third at t = 2 $\theta$ /3 ($\theta_0 = 200$).

The unavailability of the three components is represented in Figure 4.8, whereas the system unavailability and its mean value are provided in Figure 4.9.
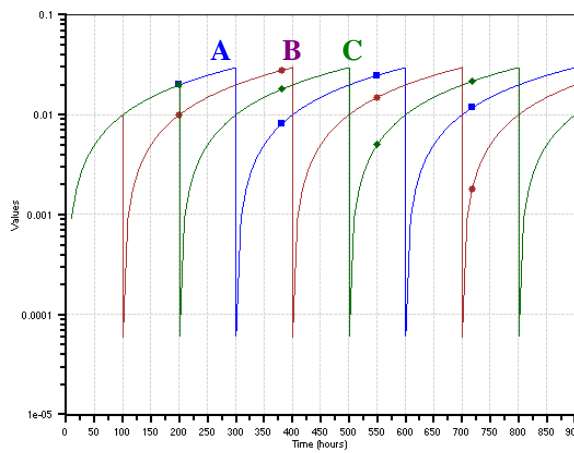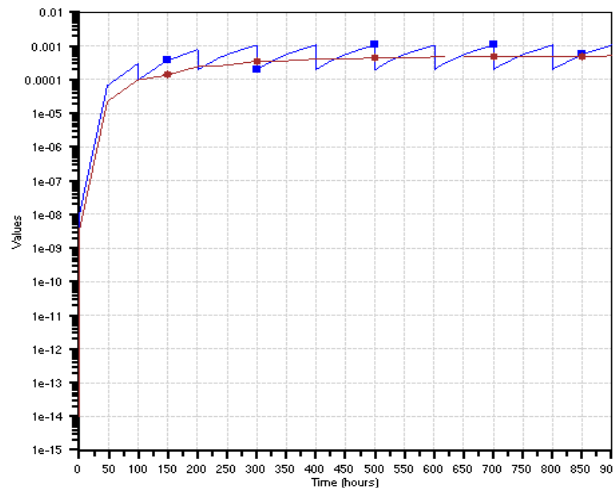


**Figure 4.8 Components' unavailability for staggered testing**

The mean value is given by:

$$Q_{mean} = \frac{1}{t}\int_0^t Q(x)\, dx \qquad\qquad (14)$$

For the example under consideration the mean value of the system unavailability is equal to $Q_{mean}$ = 5.14e-4.



**Figure 4.9 Time dependent and mean system unavailability for staggered testing of 2/3**

*Example 3*

Some of the redundant configurations considered in the IEC 61508 standard are shown in Table 4.7, represented in terms of both functional block diagrams and fault trees. In this table the configurations refer to working conditions, i.e. 1oo2 means that one channel (a component or a series of components) is sufficient to perform the safety function; hence both channels must be failed to lead to system failure. Analogously 2oo2 means that both channels are necessary, i.e. that the failure of one channel is sufficient to lead the system to a failed condition.

In order to avoid confusions we use the notation 2oo2:G meaning that the working state is determined by the working state of two components out of two. Consequently 1oo2:F means that the failure of one component determines the system failure.

In Table 4.7 each channel is described as a tested event with DC > 0 and PTC = 0.

For each configuration ASTRA calculates the $PFD_{avg}$ (Mean Probability of Failure on Demand) by integrating the time dependent unavailability function of the involved components as defined by equation (3). The integration is applied to the BDD representation of the fault tree. Hence exact results are obtained. For instance for the 1oo2:G and 2002:G, the $PFD_{avg}$ equations are respectively:

$$PFD_{avg} = \frac{1}{T} \int_0^T q_1(t) \cdot q_2(t) \; dt \tag{15}$$

$$PFD_{avg} = \frac{1}{T} \int_0^T [q_1(t) + q_2(t) - q_1(t)\, q_2(t)] \; dt \tag{16}$$

where $q_1(t)$ and $q_2(t)$ are determined by means of the equation (3).

**Table 4.7 Different redundant configuration considered in IEC 61508**

| Configuration | Functional Block Diagram | Fault tree representation |
|---|---|---|
| 1oo2:G (equivalent to 2002:F) |  |  |
| 2oo2:G (equivalent to 1002:F) |  |  |
| 2oo3:G (equivalent to 2003:F) |  |  |

Notes:

The dangerous failure rate $\lambda_D$ of channels must be multiplied by $(1-\beta)$.

The dangerous failure rate $\lambda_D$ of the CCF element must be multiplied by $\beta$.

In the IEC 61508 standard the beta factor to account for common cause failures has the value $\beta$ for undetected failures and $\beta_D = \beta/2$ for detected failures; in ASTRA it is conservatively assumed that $\beta_D = \beta$. The numerical approximation is good for all configurations.

The standard does not deal with staggered testing (as a measure to reduce the $PFD_{avg}$) for redundant tested components, whereas ASTRA does. The staggered testing policy reduces the $PFD_{avg}$ by a factor of 2.

Table 4.8 contains the results provided by ASTRA for a set of cases considered by the standard.

Note that in Table 4.8:

the test period considered is six months (4380 h);

the repair time is 8 h;

the mission time considered was 10 times the proof test interval.

The content of the table shows the correct results determined by ASTRA. The highest difference is in the case with DC = 99%. This difference (about 14%) is due to the hypothesis on $\beta$, i.e. that $\beta_D = \beta_U$. The values between brackets in the last column refer to the calculation using the staggered testing policy, something that cannot be found in the IEC standard in which only the sequential testing is considered. The staggered testing could be used to nearly half the mean unavailability value.

**Table 4.8 Comparison of results of ASTRA applied to the basic configurations**

| Architecture | DC | IEC 61508 $\lambda_D = 5.0\text{e-}06$ $\beta = 10\%$; $\beta_D = 5\%$ | ASTRA $\lambda_D = 5.0\text{e-}6$ $\beta_D = \beta = 10\%$ |
|---|---|---|---|
| 1oo1 | 0% | 1.1e02 | 1.08e-2 |
| | 60% | 4.4e-3 | 4.40e-3 |
| | 90% | 1.1e-3 | 1.13e-3 |
| | 99% | 1.5e-4 | 1.49e-4 |
| 1oo2 with CCF | 0% | 1.2e-3 | 1.19e-3   (6.30e-4) |
| | 60% | 4.6e-4 | 4.63e-4   (2.36e-4) |
| | 90% | 1.1e-4 | 1.15e-4   (5.96e-5) |
| | 99% | 1.3e-5 | 1.48e-5   (9.49e-6) |
| 2oo2 | 0% | 2.2e-2 | 2.16e-2 |
| | 60% | 8.8e-3 | 8.78e-3 |
| | 90% | 2.3e-3 | 2.26e-3 |
| | 99% | 3.0e-4 | 2.98e-4 |
| 2oo3 with CCF | 0% | 1.5e-3 | 1.48e-3 |
| | 60% | 5.1e-4 | 5.05e-4 |
| | 90% | 1.2e-4 | 1.17e-4 |
| | 99% | 1.3e-5 | 1.48e-5 |

*Example 4*

As a second example of application of the IEC 61508 equations let us consider the following 2oo3 system (Dutuit et al, 2006).

Given a Protection System configured as 2oo3 logic with two actuators the aim is to determine the Probability of Failure on Demand ($PFD_{avg}$) considering both random failures and Common Cause Failures. In this case it is assumed that DC = 0 and PTC = 0.
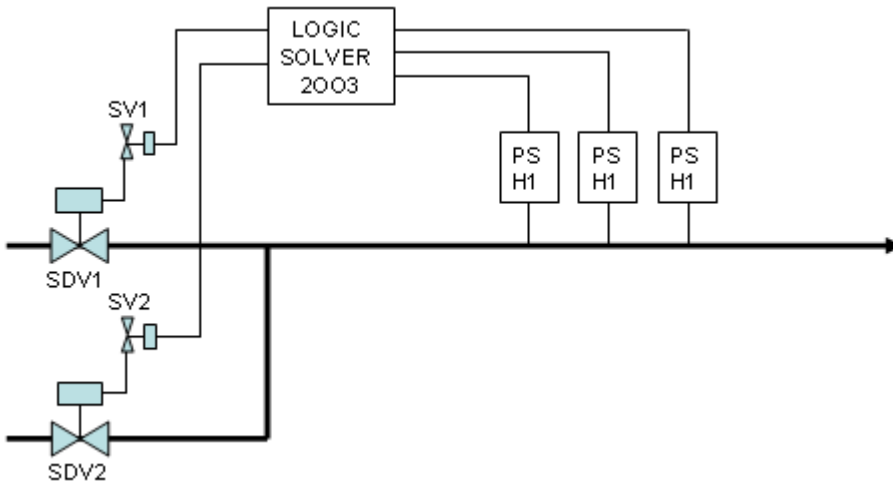


**Figure 4.10 Simplified pressure protection system**

The fault tree and the basic events data are as follows:
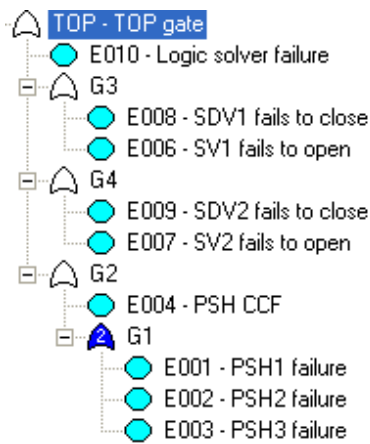


**Figure 4.11 Fault tree representation of the system in Figure 4.10**

**Table 4.9 Fault tree and data for the simple system in Figure 4.10**

| Event | λ | τ | q | θ | θ₀ | PTC | DC | β |
|-------|------|-----|---|------|------|-----|----|-----|
| E001 | 7.0e-07 | | | 720 | | | | 0.9 |
| E002 | 7.0e-07 | | | 720 | 1 | | | 0.9 |
| E003 | 7.0e-07 | | | 720 | 2 | | | 0.9 |
| E004 | 3.5e-08 | | | 720 | 720 | | | 0.1 |
| E006 | 1.3e-06 | | | 1440 | | | | |
| E007 | 1.3e-06 | | | 1440 | 1 | | | |
| E008 | 2.1e-06 | | | 2160 | | | | |
| E009 | 2.1e-06 | | | 2160 | 1 | | | |
| E010 | 1.0e-06 | 10 | | | | | | |

**(missing values mean 0)**

These authors determined, for a mission time of 23,800h the value $PFD_{avg} = 6.374 \times 10^{-3}$ corresponding to a SIL 2 level.

The analysis of the same fault tree with ASTRA gives the same result: $PFD_{avg} = 6.370 \times 10^{-3}$, together with the $PFD_{max} = 1.276 \times 10^{-2}$
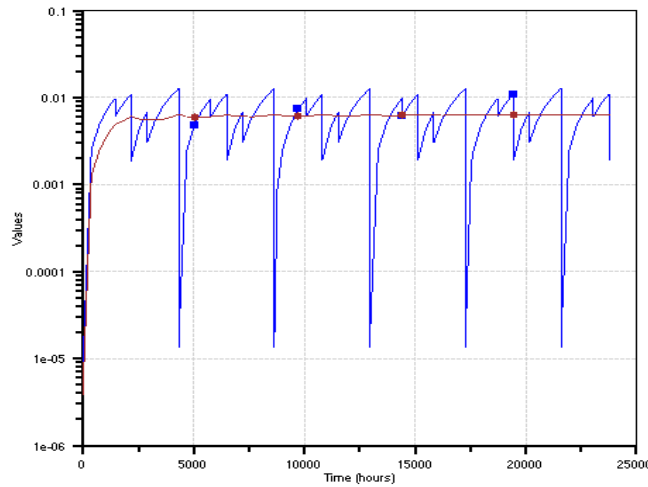


**Figure 4.12 Plot of the unavailability function for the simple system in Figure 4.10**

As pointed out by Dutuit et al., the mean value of the $PFD_{avg}$ is not sufficient to state that the protective system presents a given SIL level. Indeed, the unavailability function has the classical saw-tooth behaviour with peaks that may enter into a worst SIL level region, as can be seen in Figure 4.12. For the present example the peak value of the unavailability on demand was greater that $10^{-2}$, corresponding to SIL1, for about 12% of the mission time, corresponding to 2865 h, a value that cannot be considered negligible.

### 4.4 Accident Frequency analysis

If the Top-event describes a catastrophic situation, then the parameter of interest is the Reliability R(t), i.e. the probability that the system works from 0 to the mission time T *without* failure. In fault tree analysis we work on failure events, so the parameter of interest becomes the Unreliability F(t). Whereas the unreliability of systems with not repairable components is equal, by definition, to the unavailability, the unreliability of systems with repairable components cannot be exactly determined using the fault tree methodology (see e.g. Clarotti, 1981). However, several bounds giving conservative values of system unreliability can be found in literature.

In ASTRA the following two bounds have been implemented:
Expected Number of Failures (ENF);
Vesely equation.

The application of these bounds requires the determination of the unconditional failure and repair frequencies.

It can be shown that the unconditional failure frequency of a fault tree containing events in different form (SP, SN, DF) is given by:

$$\omega_S(t) = \sum_{x=1}^{N}[p_x^f(t)\,\omega_x(t) + p_x^r(t)\,v_x(t)] \qquad (17)$$

where:

$p_x^f(t)$ and $p_x^r(t)$ are respectively the probability of critical states for the failure and repair of x.

$\omega_x(t)$ and $\nu_x(t)$ are respectively the unconditional failure and repair frequencies of the event x.

The term $p_x^f(t)$ represents the probability that the *failure* of the generic event $x \in \mathbf{x}$ is critical, i.e. the Top-event is verified (Top=1) if x = 1 and it is not verified (Top=0) if x = 0. For positive events $p_x^f(t)$ is nothing but the Birnbaum importance index, which is given by:

$$p_x^f(t) = \Pr\{\Phi|_{x=1}\} - \Pr\{\Phi|_{x=0}\} \qquad (18)$$

The term $p_x^r(t)$ represents the probability that the *repair* of the generic event $x \in \mathbf{x}$ is critical, i.e. the Top-event is not verified (Top=0) if x = 1 and is verified (Top=1) if x = 0. For negated events $p_x^r(t)$ is given by:

$$p_x^r(t) = \Pr\{\Phi|_{x=0}\} - \Pr\{\Phi|_{x=1}\} \qquad (19)$$

Therefore, according to equation (17), the Top-event is verified in the time interval t-t+dt if it is in a critical state for the failure of x (represented by $p_x^f(t)$) and fails in t+dt (represented by $\omega_x(t)$) plus the probability that the Top-event is in a critical state for the repair of x (represented by $p_x^r(t)$) and is repaired in t+dt (represented by $\nu_x(t)$).

Equation (17) is of general character and is applicable to events in DF form. It can be simplified for events of type SP and SN. Indeed:

For SP events $\omega_S(t) = \sum_{x=1}^{N} p_x^f(t)\, \omega_x(t)$ since $p_x^r(t) = 0$

For SN events $\omega_S(t) = \sum_{x=1}^{N} p_x^r(t)\, \nu_x(t)$ since $p_x^f(t) = 0$

Finally it remains to determine $p_x^f(t)$ and $p_x^r(t)$ for basic events in simple modules in order to obtain their contribution to the Top-event frequency.

In ASTRA a module contains only events of SP type. Therefore $p_x^r(t) = 0$. Moreover it can be shown that:

$$p_x^f(t) = p_x^M(t)\, p_M^f(t) \qquad (20)$$

This equation states that the probability that x is critical with respect to the Top-event is given by the product of two terms: 1) the probability that x is critical with respect to the module M; and 2) the probability that M is critical with respect to the Top-event.

Analogously the unconditional repair frequency is given by equation (21) with obvious meaning of the symbols:

$$\nu_T(t) = \sum_{x=1}^{N} p_x^r(t)\, \omega_x(t) + \sum_{x=1}^{N} p_x^f(t)\, \nu_x(t) \qquad (21)$$

*Top event Expected number of failures $W_T(t)$*

The Expected Number of Failures $W_T(t)$ is obtained as:

$$W_T(t) = \int_0^t \omega_T(\tau)\,d\tau + Q_T(0) \qquad (22)$$

The ENF is also a good upper bound for the unreliability $F_T(t)$ provided that its value is less than 0.1.

$$\text{If } \omega_T(t) \approx \text{constant then } MTBF_T = \frac{1}{\omega_T(t)}; \qquad (23)$$

$$\text{if } Q_T(t) \approx \text{constant then } MTTR_T = Q_{Tmean} * MTBF_T \qquad (24)$$

### *Top event Expected number of repairs $V_T(t)$*

If the fault tree does not contain INH gates nor tested events then also the Expected Number of Repair $V_T(t)$ is calculated as:

$$V_T(t) = \int_0^t v_T(\tau)\,d\tau \qquad (25)$$

### *Top event Unreliability upper bound $F_T(t)$*

For safety applications the Expected number of failure is generally a very good upper bound for the top-event unreliability $F_T(t)$. ASTRA does not calculate the MTBF if the failure frequency is not constant. In these cases the *time dependent option* of ASTRA allows calculating the Vesely bound for $F(t)$, which is better than $W(t)$, but more time consuming.

To this purpose the conditional failure frequency $\Lambda_T$ at Top level is determined on the basis of the unconditional failure frequency $\omega_T(t)$ and unavailability $Q_T(t)$, (Vesely, 1970) i.e.:

$$\Lambda_T(\tau) = \frac{\omega_T(\tau)}{1 - Q_T(\tau)} \qquad (26)$$

Then,

$$F_T(t) = 1 - [1 - Q_T(0)]\, e^{-\int_0^t \Lambda_T(\tau)\,d\tau} \qquad (27)$$

Mean time to first failure is calculated, according to its definition, as:

$$MTTFF_T = \int_0^\infty (1 - F_T(t))\,dt \qquad (28)$$

In practice, to save time, it is more convenient to determine $W(t)$ and then to pass to $F(t)$ if deemed necessary.
$F(t)$ is calculated by activating the *time dependent option* when the ENF is greater than the threshold (e.g. 0.1) defined by the user.

NOTE: Due to the time consuming operation used to calculate equations (27) and (28) the Vesely equation is applied only when the time dependent analysis option is selected.

*Mean  Probability of Failure per Hour according to IEC 61508*

In the  case of high demand  or  continuous  mode  of operation of the safety system the  standard IEC 61508  requires the determination of the average value of the Probability of Failure per Hour $PFH_{avg}$ which is given by:

$$PFH_{avg} = \frac{W(T)}{T}$$

where W(T) is the expected number of failures at the mission time T.

The following Table contains the results provided by ASTRA for a set of cases considered by the standard, in which the test period considered is six months (4380 h), mean repair time  of 8h and mission time equal to 10 times the proof test interval.

The content of the table shows the correct results determined by ASTRA.

| Architecture | DC | IEC 61508 $\lambda_D = 5.0e\text{-}06$ $\beta= 10\%;\ \beta_D=5\%$ | ASTRA $\lambda_D = 5.0e\text{-}6$ $\beta_D= \beta= 10\%$ |
|---|---|---|---|
| 1oo2 with CCF | 0% | 5.9E-7 | 5.86E-7 |
| | 60% | 2.1E-7 | 2.14E-7 |
| | 90% | 5.1E-8 | 5.09E-8 |
| | 99% | 5.0E-9 | 5.01E-9 |
| 2oo2 | 0% | 1.0E-5 | 9.78E-6 |
| | 60% | 4.0E-6 | 3.96E-6 |
| | 90% | 1.0E-6 | 9.97E-7 |
| | 99% | 1.0E-7 | 9.99E-8 |
| 2oo3 with CCF | 0% | 7.7E-7 | 7.56E-7 |
| | 60% | 2.4E-7 | 2.42E-7 |
| | 90% | 5.3E-8 | 5.27E-8 |
| | 99% | 5.0E-9 | 5.03E-9 |

*Example 1.*

Let us consider the bridge network represented in terms of reliability block diagram and fault tree in Figure 4.13. The coherent fault tree describes the events leading to the top-event "no output signal". The analysis is performed considering $\lambda = 0.01$ and $\mu = 0.1$ for all events.
The results of the analysis are graphically represented in Figure 4.14 and Figure 4.15.

At the mission time T = 50 h the results are:

– Unavailability $Q_T(50)$              = 1.755459 e-2
– Unconditional Failure frequency $\omega_T(50)$ = 3.612036e -3
– Unconditional Repair frequency $\nu_T(50)$ = 3.595805e -3
– Expected Number of Failures $W_T(50)$    = 0.1490059
– Expected Number of Repair $V_T(50)$     = 0.1314516

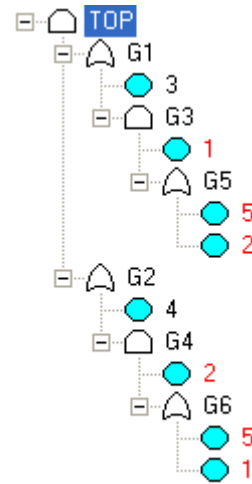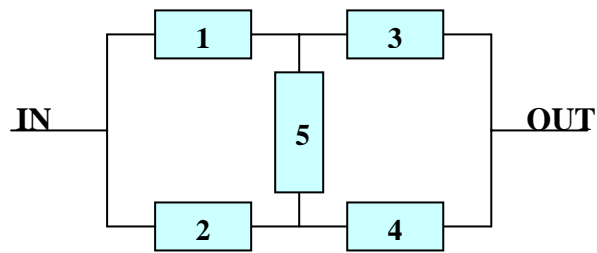- Unreliability $F_T(50)$ = 0.1403110



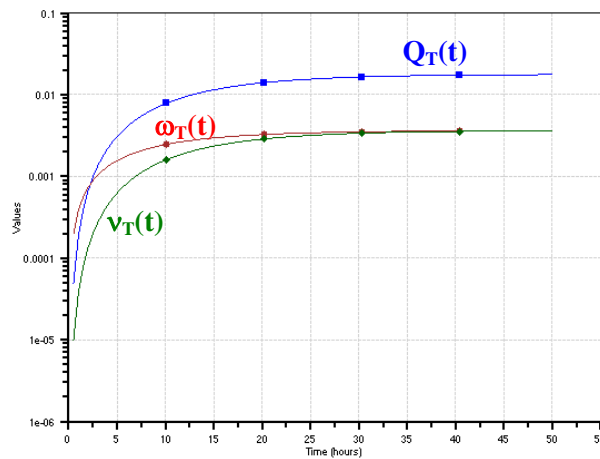**Figure 4.13 Bridge network of repairable components and associated fault tree**



**Figure 4.14 Unavailability, failure and repair frequencies for the bridge system**
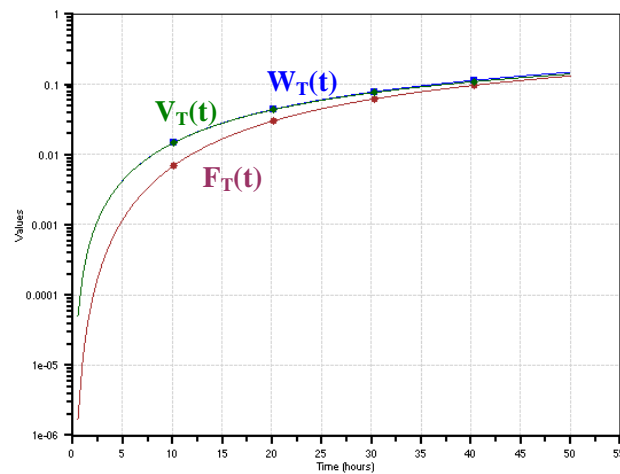


**Figure 4.15 Expected number of failures, Expected number of repair and
Unreliability upper bound for the bridge system**

*Example 2*

This example concerns the analysis of a non-coherent fault tree.
Let us consider again the function Top = $[x_2 ( x_1 + \bar{x}_3 + \bar{x}_4 )] + [x_3 ( \bar{x}_1 + \bar{x}_2 \ x_4 )]$ previously considered for the determination of the unavailability.
Basic events data are as follows:

**Tabella 4.10 Data of basic events for Top = $[x_2 ( x_1 + \bar{x}_3 + \bar{x}_4 )] + [x_3 ( \bar{x}_1 + \bar{x}_2 \ x_4 )]$**

| Var | Type | $\lambda$ | $\tau$ | q |
|-----|------|-----------|--------|---|
| $x_1$ | Not repairable | 1.e-5 | | |
| $x_2$ | On-line maintained | 1.e-6 | 200 | |
| $x_3$ | On-line maintained | 1.e-4 | 20 | |
| $x_4$ | On demand | | | 0.001 |

The frequency analysis results are plotted in Figure 4.16 for a mission time of 10,000 h.
It can be seen that the unavailability reaches the steady state condition very rapidly and consequently Ws(t) and Vs(t) are very close each other.

At T = 10,000 h the following values have been calculated by ASTRA:

- Qs(T)  = 2.005851E-03
- ωs(t)  = 9.129257E-05
- vs(t)  = 9.131060E-05
- Ws(t)  = 9.595718E-01
- Vs(t)  = 9.575631E-01
- Fs(t)  = 6.177130E-01
- MTBF  = 10953h
- MTTR  = 21.97h
- MTTF  = 19951h
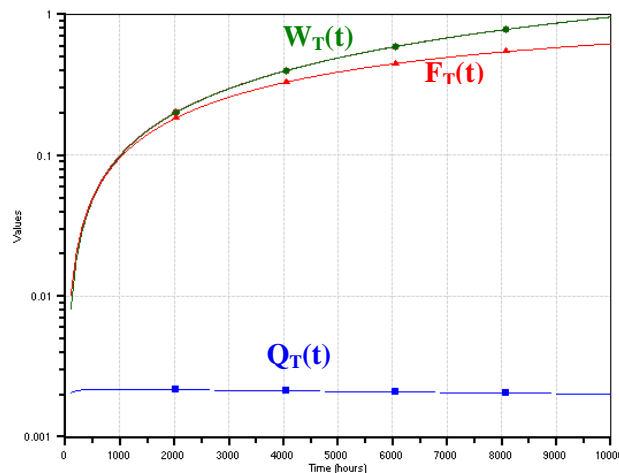


**Figure 4.16 Unavailability, Expected number of failures and unreliability for the LBDD in Figure 3.3**

The non coherent function can also be approximated to a coherent function by setting all negated events to 1. In our example the coherent tree is represented in Figure 3.4.
The results of the analysis of the coherent tree, performed considering the mission tome T = 10000h are as follows:

Qs(T) = 2.195569E-03     ωs(t) = 1.007782E-04     vs(t) = 1.007782E-04
Ws(t) = 1.007852        Vs(t) = 9.762612E-01     Fs(t) = 6.35781E-01
MTBF = 9922.7h          MTTR = 21.787h        MTTF = 9901h

Note that all results are conservative, as expected.


## 4.5 Frequency analysis using the extended INH gate

In the analysis of catastrophic Top-events it is important to model situations in which a failure occurs only if the direct causes occur in a given sequence. Consider for instance the following example. The overpressure in a tank triggers the intervention of a shut-down system; if this system does not operate then the tank rupture occurs. Hence the event "tank rupture" is due to the occurrence of the initiating events I "overpressure" and the enabling event E "shut-down does not intervene". However, the tank rupture can occur *only if* E occurs *before* I or at the same time. If E occurs *after* I there would be simply a trip of the plant.
The simple AND of the two input variables I and E cannot represent this situation, since the sequence of occurrence is not taken into consideration.

These situations can be modelled in ASTRA using the Inhibit (INH) gate as shown in Figure 4.18.
This *extended definition of the INH gate* is based on the distinction between *initiating* and *enabling* events.
*An Initiating event is an event whose occurrence triggers the intervention of the Enabler event.*
The output is true when, at the time the input is true, the condition defined by the enabler event is *already* true.
The method implemented in ASTRA identifies the events as either *initiator* or *enabler* depending on the sub-tree they belong to. Common events are flagged as initiators.
The differentiation of the type of events has an impact on the calculation of W(t), since initiating events are characterised by their failure frequency ω(t), whereas enabler events, associated with components of the protective system, are characterised by their on-demand unavailability q(t).
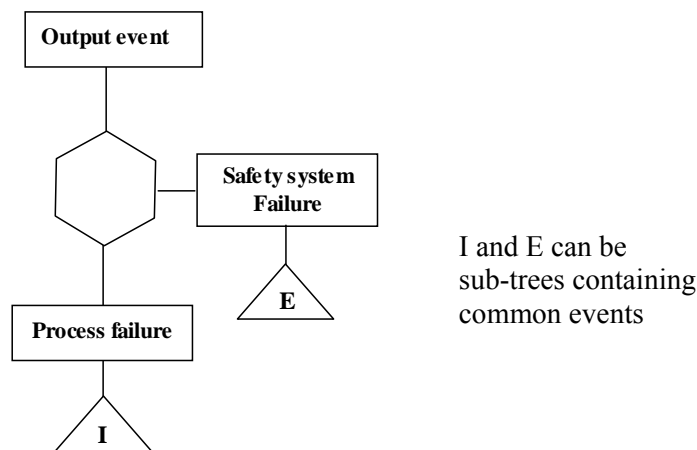


**Figure 4.18 The INH gate used for modelling the relationship between initiator and enabler events**

The failure and repair frequencies of the output event are given by:

$$\omega_O(t) = \omega_I(t)\, q_E(t) \tag{29}$$

$$v_O(t) = v_I(t)\, q_E(t)) \tag{30}$$

Therefore in ASTRA when the parameter of interest is the frequency of the catastrophic Top event modelled by means of the INH gate, enabler events are characterised by their on-demand unavailability $q_x(t)$ only, i.e. their unconditional failure and repair frequencies are set to zero.

*Example*

A system is comprised of two components: A monitors the operation of the component B. System failure occurs if both fail, but only if A fails before B.
Data about components are as follows.

A : $\lambda_A = 1.e-6$      $\mu_A = 0$
B : $\lambda_B = 1.e-7$      $\mu_B = 0$

The system is not repairable. Table 4.12 shows the comparison of the unavailability values $Q(t)$ obtained using a Markovian approach (Ericson, 2005) with those calculated using the ASTRA method above described for different mission times. As can be seen the agreement is very good at all values of the mission time.

Let us consider now the same problem with the following parameters:

A : $\lambda_A = 1.e-4$      $\mu_A = 1.e-2$
B : $\lambda_B = 1.e-5$      $\mu_B = 1.e-2$

The Markov state diagram and the fault tree are as shown in Figure 4.19.

**Table 4.12 Comparison between Markov analysis and ASTRA 3.x on sequential events (non-repairable case)**

| Mission time (h) | Markov | ASTRA |
|---|---|---|
| 100 | 4.99980E-10 | 4.99980E-10 |
| 1,000 | 4.99800E-8 | 4.99800E-8 |
| 10,000 | 4.98006E-6 | 4.98005E-6 |
| 100,000 | 4.80542E-4 | 4.80542E-4 |
| 1,000,000 | 3.45145E-2 | 3.45144E-2 |
| 10,000,000 | 5.41213E-1 | 5.41208E-1 |

Table 4.13 contains the results of the analysis for a mission time T = 1,000 h.

The comparison of the results between the two programs XSMKA (De Cola, 2005) and ASTRA shows good agreement.

In practice it is common to deal with a cascade of INH operators as shown in Figure 4.20a), in which the previously considered tank is supposed to have two levels of protection. In this case two INH gates are used to model the pressure increase. We can notice that in this case there is no need to consider the sequence of intervention of the safety devices, since both must be failed to produce the tank rupture.
Therefore the cascade of INH gates is equivalent to a single INH gate in which all enabling functions are grouped under an AND gate of the enabler branch as shown Figure 4.20b).
These considerations are applicable to the cascade of any number of INH gates.

| Parameter | XSMKA | ASTRA |
|---|---|---|
| Unreliability | 8.901275E-05 | 8.911964E-05 |
| Unavailability | 4.940146E-06 | 4.96708E-06 |
| ENF – Expected Number of Failures | 8.904747E-05 | 8.912281E-05 |
| ENR – Expected Number of Repairs | 8.407051E-05 | 8.415573E-05 |
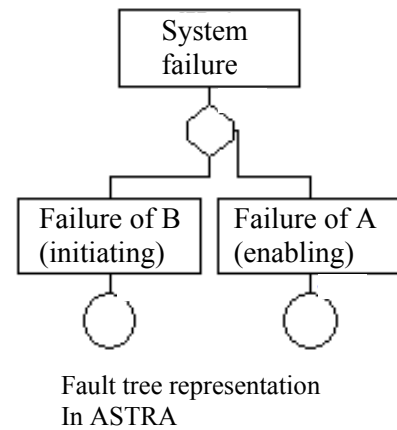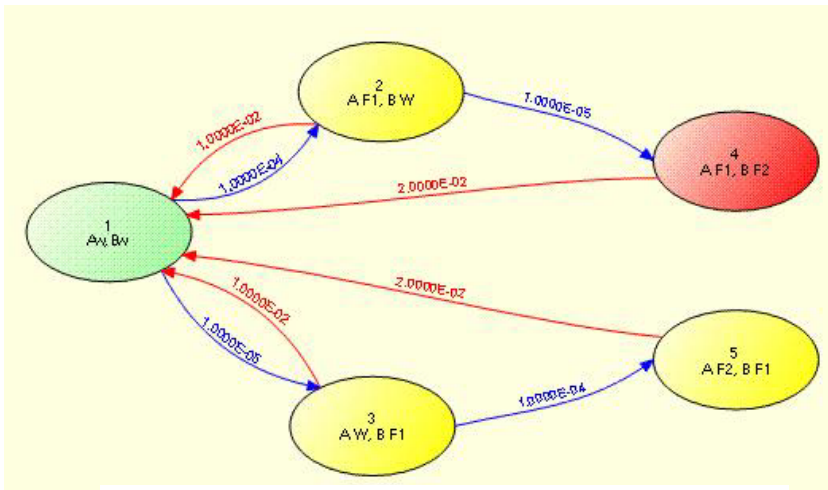


A, B components' names;
Subscripts:
W = Working
F = Failed;
N = state number from which the transition comes from.

r sequential repairable events

This extended implementation of the INH gate should not be confused with the sequential AND gate that may have more than two inputs. In the cascade of INH gates the grouped enabler events are independent. In the latter we may have more than two events which must fail according to a given sequence.
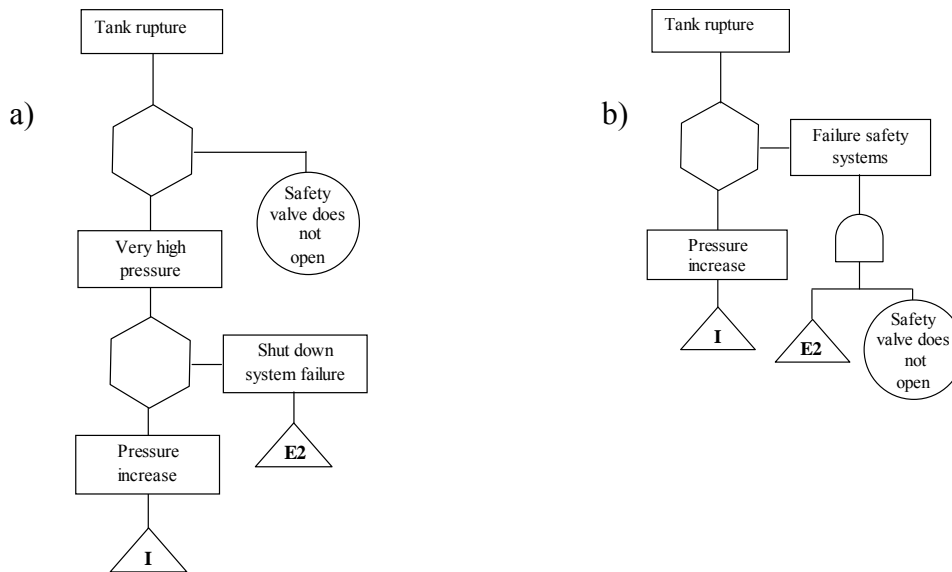


**Figure 4.20 a) Cascade of INH gates; b) equivalent fault tree representation**

## 4.6 Importance analysis

Components' importance measures play a very important role in system reliability analysis. They are used to identify the weakest parts of the system for design improvement, failure diagnosis and maintenance.
In ASTRA the importance analysis module allows to get the ranking of components' importance based on both unavailability and expected number of failures.
Importance measures can be determined either at a specific time (generally at the mission time) or as time dependent.

For each importance measure (IM) two contributions, positive and negative, are determined for non-coherent variables.

Besides the probabilistic IM, ASTRA also calculates the *Structural importance index IS$_x$*, which is a measure based only on how events are located in the set of MCS. The Structural importance is determined by applying the equation $p^f_x(t)$ and $p^r_x(t)$ in which all events have probability 0.5 (Lambert 1975).

All equations described below for importance analysis are applied as function of time. They are listed without any proof or justification. Interested readers can refer to Contini et al. (2008), and to Van der Borst-Shoonaker (2001) for details and further references.

### *4.6.1 Importance measures based on Unavailability*

We have seen that a modularized fault tree contains:
- one or more independent simple modules;
- the Top-module (module containing the top event).

A generic basic event x can be in one of the simple modules or in the top module.
A simple module contains only events in positive form; also a simple module in the Top-module can appear only in positive form. Consequently, events of type \$ (SN) or & (DF) can appear only in the Top-module.

*Probability of critical states $p^f_x(t)$ and $p^r_x(t)$*

For basic events of a monotonic function $\Phi(\mathbf{x})$ (AND-OR fault trees) the first importance measure was proposed by Birnbaum. The Birnbaum importance of a generic component *x* is defined as follows:
$$p^f_x(t) = \partial P\{\Phi, t\} / \partial P\{x, t\} = P(\Phi(x = 1, \mathbf{x}, t) \quad P(\Phi(x = 0, \mathbf{x}, t)$$
where $P\{\Phi, t\}$ is the unavailability of $\Phi(\mathbf{x})$ at time *t* and $P\{x, t\}$ the unavailability of *x* at time t.

The partial derivative of $P\{\Phi, t\}$ with respect to $P\{x, t\}$ means that the Birnbaum measure can be interpreted as the probability that the Top-event is critical with respect to the failure of x, i.e. the Top-event is verified as soon as x fails.

The Birnbaum index does not depend on the failure probability of component *x*. However, it is important in that:
− It gives the maximum variation of the Top-event unavailability when the component changes its state from perfectly working to failed;
− It is useful when used in connection with other indexes;
− Other indexes can be expressed as a function of it as e.g. Criticality, Risk Achievement Worth, Risk Reduction Worth;

– It is used for determining the system unconditional failure frequency.

For *non-coherent variables* the Birnbaum index as defined above loses its meaning, since it can assume negative values.

The generalization of the Birnbaum index for *non-coherent* variables can be expressed as:

$$p_x^f(t) = \partial P\{\Phi, t\} / \partial P\{x, t\} = P(\Phi_{1x}, t) \wedge \overline{P(\Phi_{ox}, t)} \tag{31}$$

for the positive contribution, and

$$p_x^r(t) = \partial P\{\Phi, t\} / \partial P\{\bar{x}, t\} = \overline{P(\Phi_{1x}, t)} \wedge P(\Phi_{ox}, t) \tag{32}$$

for the negative contribution.

These two values are used to determine other importance indexes.

The generic event x considered can be part of a module. Its Birnbaum importance with respect to the Top-event is determined by multiplying the importance of x with respect to the Module with the importance of the module with respect to the Top-event, i.e.:

$$p_x^f(t) = p_x^M(t)\, p_M^f(t) \tag{33}$$

In order to simplify the notation in the following we will represent $P\{\Phi, t\}$ as $Q_T(t)$ and $P\{x, t\}$ as $q_x(t)$.

*Criticality importance measure, $IC_x$*

The criticality index represents the probability that the event *x* is critical and its occurrence leads to system failure.
This index can also be interpreted as the relative variation of the Top-event occurrence probability vs. the relative variation of the occurrence probability of the basic event *x*, i.e.:

$$IC_x^+(t) = \frac{\partial Q_T(t)/Q_T(t)}{\partial q_x(t)/q_x(t)} = p_x^f(t)\frac{q_x(t)}{Q_T(t)} \tag{34}$$

For negated variables:

$$IC_x^-(t) = p_x^r(t)\frac{1-q_x(t)}{Q_T(t)} \tag{35}$$

*Risk Achievement Worth, $RAW_x$*

The RAW is defined as a measure of the increase of the system failure probability when *x* is supposed failed or removed e.g. for test/maintenance operations. In calculating the RAW it is important to consider all other components that are dependent by the failure / removal of *x*. According to the definition it is proved that:

$$RAW_x^+(t) = 1 + p_x^f(t)\frac{1-q_x(t)}{Q_T(t)} \tag{36}$$

For negated variables:

$$RAW_x^-(t) = 1 + p_x^r(t) \frac{q_x(t)}{Q_T(t)} \qquad (37)$$

*Risk Reduction Worth RRW$_x$*

The RRW is defined as a measure of the decrease of the system failure probability when *x* is supposed to be perfectly working:

$$RRW_x^+ = \frac{Q_T}{Q_T - p_x^f(t)\, q_x(t)} \qquad (38)$$

For negated variables:

$$RRW_x^- = \frac{Q_T}{Q_T - p_x^r(t)[1 - q_x(t)]} \qquad (39)$$

### 4.6.2 Importance measures based on failure frequency (Unreliability analysis))

In case of unreliability analysis it is necessary to subdivide events as *initiators* and *enablers* because they have different role in the system and consequently they are treated differently. These events have already been defined in section 4.5.
Moreover:
− Initiating events are characterised by their failure frequency;
− Enabling events are characterised by their on-demand unavailability;
− Initiating events may cause system failure if the system is in a critical state for the initiating event;
− Enabling events contribute to system failure but do not cause it.

An initiating event that is in a MCS with other initiating events has an enabling contribution.
Let (A B) be a system failure combination, i.e. a minimal cut set of a fault tree.
The unconditional failure frequency $\Omega$(A B) that the combination (A B) occurs (enter into the failed state A=1, B=1) in the time interval *dt* is given by the probability that A occurs in *t-t+dt* (represented by $\omega_A(t)$ dt) with B already failed at *t* (represented by $q_B(t)$) or that B occurs in *t-t+dt* with A already failed at *t*, i.e.:
$\Omega$(A B) = $q_B(t)\, \omega_A(t) + q_A(t)\, \omega_B(t)$
Now, consider for instance event A. In the first term of the right hand side A behaves as initiator because it is characterised by $\omega_A(t)$, whereas in the second term - as enabler, characterised by $q_A(t)$.

In order to determine the importance of a basic event it is necessary to consider its type: initiator or enabler. It is proved in Contini-Matuzas (2011) a method for determining such importance measures. Their implementation in ASTRA 3.x is planned for the next release.

## 4.7 Probabilistic quantification of SMCS

The determination of the generic SMCS is followed by the probabilistic quantification of each of them for the determination of: Unavailability, Unconditional failure frequency, Expected number of failures, and Unreliability. A MCS of order *n* is the parallel failure configuration of *n* basic events.

*Unavailability of a MCS*

The unavailability of a generic minimal cut set $C_j$ of order $n$ is the probability that all $n$ events are verified at time t. Since events are independent then:

$$Q_{Cj}(t) = \prod_{i=1}^{nj} q_i(t) \tag{40}$$

*Expected number of failures of a MCS*

The ENF of a MCS is obtained by integrating, over the mission time interval, the unconditional failure frequency of the SMCS given by:

$$\omega_{Cj}(t)\,dt = \sum_{i=1}^{nj} \omega_i(t) \prod_{\substack{k=1 \\ k \neq i}}^{nj} q_k(t)\,dt \tag{41}$$

The above equation expresses the concept that the SMCS occurs in a time interval t, t+dt if:

- *n-1* events already occurred at t, given by $\displaystyle\prod_{\substack{k=1 \\ k \neq i}}^{nj} q_k(t)$

- the last one occurs in *dt*, expressed as $\omega_i(t)\,dt$

The last event to occur may be the first, the second, and so on, that's why the use of the summation.

Hence: $\displaystyle W_{Cj}(t) = \int_0^t \omega_{Cj}(\tau)\,d\tau + Q_{Cj}(0) \tag{42}$

*Unreliability upper bound of a MCS*

As for the Top event, the unreliability of a MCS is calculated by means of the conditional failure frequency of the MCS `determined from $\omega_T(t)$ and $Q_T(t)$.

$$\Lambda_{Cj}(\tau) = \frac{\omega_{Cj}(\tau)}{1 - Q_{Cj}(\tau)}$$

Then,

$$F_{Cj}(t) = 1 - [1 - Q_{Cj}(0)]\,e^{-\int_0^t \Lambda_{Cj}(\tau)\,d\tau} \tag{43}$$

If the number of MCS is very high the integration operation, necessary to determine the Expected number of failures and the Unreliability, becomes time consuming. Moreover, the above equations are applied to determine the importance ranking of MCS, since the quantification has already been performed on the LBDD. Therefore it is interesting to give the user the possibility to apply simplified equations which do not require any integration.
The following equation gives an approximated result of the Expected number of failures:

$$W_{Cj}(t) = \sum_{i=1}^{nj} \left[ \frac{\lambda_i}{h+r} \prod_{\substack{k=1 \\ k \neq j}}^{nj} Q_{Cj}(t) \right] t \qquad (44)$$

where:

$$r \begin{cases} 1 \text{ if } \lambda_i \text{ is the failure rate of an on-line maintained or tested component} \\ \\ 0 \text{ if } \lambda_i \text{ is the failure rate of a not repairable component} \end{cases}$$

k   is the number of not repairable components belonging to the MCS.
If the MCSs contains tested component, the product in (5.13) is multiplied by the coefficient $C_n$ given in table 5.1.

$$G_z = \frac{2^z}{z+1} \qquad (45)$$

where z is the number of tested events in the MCS under consideration.

The $G_z$ equation is valid for components in a MCS which are tested in sequential mode and have equal test interval.

# 5. CONCLUSIONS AND FURTHER DEVELOPMENTS

In this report we have described the main algorithms implemented in ASTRA 3.x for performing the analysis of both coherent and non-coherent fault trees. More detailed information is given in the referenced documentation.

Since non-coherent fault trees contain different types of variables (SP, SN, and DF) for which algorithm of different cost (in terms of computational resources) are required, an algorithm for dynamically labelling each BDD node with the variables' type was implemented. The experimental results showed the advantage of the dynamic labelling operation. The reduction of the number of nodes with DF variables implies a reduction of the working memory due to the consequent reduction of computational effort for determining the failure and repair frequencies.

From the LBDD the ZBDD embedding all MCS is obtained from which the SMCS are finally extracted.

The probabilistic analysis of the fault tree is performed on the LBDD. Besides the Unavailability, the unconditional failure and repair frequencies, the Expected Number of Failures and the Vesely equation for the unreliability, ASTRA performs the importance analysis of basic events; equations used depend on the type of analysis: unavailability or unreliability. In the second case the calculations are based on a set of new equations that will be implemented in the next version.

Another important implementation refers to the analysis of safety systems performed according to the methods defined in the standard IEC 61508 for the low and high demand modes.

In order to analyse configurations containing multiple channels the application of the fault tree requires representing each "basic event" as three events descending from the same OR gate. This is due to the fact that the standard considers the component failure rate $\lambda_D$ (D stands for *dangerous*) as given by the Detected part ($\lambda_{DD}$) and by the Undetected part ($\lambda_{DU}$). The first accounts for failures that can be on-line detected and repaired whereas the second by those that can be detected only at test time intervals.

Moreover, the standard considers also the Proof Test Coverage (PTC) parameters to take into account non-complete tests. Hence, PTC < 1 means that there may be non-tested and non revealed failures which remain hidden during the whole mission time interval.

To simplify the fault tree construction a solution has been implemented in ASTRA which avoids the triple-event representation by defining a new tested event having DC and PTC as additional parameters.

# REFERENCES

Akers S. B. (1978), "Binary Decision Diagrams", *IEEE Transactions on Computers,* Vol C-27.

Remenyte-Prescoll R., Andrews J. (2008), "Analysis of non-coherent fault trees using ternary decision diagrams", *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* June 1, vol. 222 no. 2 127-138

Brace K, Rudell R, Briant R. (1990), "Efficient Implementation of a BDD Package", *27th ACM/IEEE Design Automation Conference,* IEEE 0738.

Bryant R.E (1986), "Graph Based Algorithms for Boolean Functions Manipulation", *IEEE Transactions on Computers*, Vol. C-35.

Clarotti C.A. (1981), "Limitations of minimal cut-set approach in evaluating reliability of systems with repairable components", *IEEE Transactions on Reliability,* Vol. 30.

Contini S. Cojazzi G.G.M., de Cola G., (2006), "On the exact analysis of non-coherent fault trees: the ASTRA package", PSAM, International Conference on Probabilistic Safety Assessment and Management, New Orleans, Louisiana, USA.

Contini S. Cojazzi G.G.M., Renda G. (2008), "On the use of non-coherent fault trees in safety and security studies", *Reliability Engineering and System Safety,* Vol. 93.

Contini S. Matuzas V. (2009), "ASTRA 3.0: Test Case Report", EUR Technical Report, EUR 24124 EN.

Contini S., Matuzas V., de Cola G., (2012) "An efficient FT-based implementation of the IEC 61508 safety analysis methods", Paper to be presented at the CISAP5 Conference, Milan, Chemical Engineering Transaction, Vol 26, 2012.

Contini S. Matuzas V. (2010), "Reduced ZBDD construction algorithm for large fault tree analysis" ESREL, European Safety & Reliability Conference, Rhodes, Greece.

Coudert J.C, Madre P. (1994), "Metaprime: an Interactive Fault Tree Analyser with Binary Decision Diagrams", *IEEE Transactions on Reliability*, Vol. 43.

De Cola G., (2005), "XS Toolset – MKA Module for Markovian Analysis", User Guide.

Demichela M., Piccinini N., Ciarambino I., Contini S., (2003), "On the numerical solution of fault trees", *Reliability Engineering and System Safety,* Vol. 82.

Dutuit Y., Rauzy A., Signoret J.P., (2006), "Probabilistic assessment in relationship with Safety Integrity Levels by using Fault Trees", *ESREL 2006 Conference*, Guedes Soares & Zio ed.

Ericson, C.A. (2005), "Hazard Analysis Techniques for System Safety", Wiley Interscience, ISBN: 0.471-72019-4

Kumamoto H., Henley E.J., (1981), "Probabilistic Risk assessment and Management for Engineers and Scientists", Prentice Hall, Englewood Cliffs, NY.

IEC 61508 standard (2010), "Functional safety of electrical/electronic/programmable electronic safety-related systems", Parts 1 to 7, 1988-2000, International Electrotechnical Commission, Geneva, Switzerland. S+ IEC 61508 standard, Commented version.

Lambert H. E., (1975), "Measures of importance of events and cut sets in fault tree analysis", in *Theoretical and Applied Aspects of System Reliability and Safety Assessment*, Barlow, Fussell, Singpurwalla ed. SIAM, Philadelphia, USA.

Liu J.C, Pan Z.J. (1990), "A New Method to Calculate the Failure Frequency of Non-Coherent Systems", I*EEE Transactions on Reliability,* Vol. 39.

Minato S. (1990), "Zero-suppressed BDDs for set manipulation in combinatorial problems", *Proc. ACM/IEEE Design Automation Conference*.

Odeh K, Limnios N. (1996), "A New Algorithm for Fault Trees Prime Implicants Computation", *Probabilistic Safety Assessment and Management, ESREL 96 – PSAM III Conference*, P.C. Cacciabue, I.A. Papazoglou, Creete, Greece.

Rauzy (1993), A, "New Algorithms for Fault Trees Analysis", *Reliability Engineering and System Safety*, Vol 40, 203-211,

Rauzy A, Dutuit Y. (1997), "Exact and Truncated Computation of Prime Implicants of Coherent and Non-Coherent Fault Trees within Aralia", *Reliability Engineering and System Safety*, Vol 58.

Sinnamon R.M., Andrews J.D. (1996), "Quantitative fault tree analysis using Binary Decision Diagrams", *Journal Europèen des Systèmes Automatisès*,

Van der Borst M. and Schoonakker H. (2001), "An overview of PSA importance measures," *Reliability Engineering & System Safety*, vol. 72, no. 3

Vesely W.E. (1970)*, "*A time dependent methodology for fault tree evaluation", *Nuclear Engineering and Design, Vol 13.*

## ACKNOWLEDGEMENTS

**Abstract**

This report describes the main algorithms implemented in ASTRA 3.x to analyse coherent and non-coherent fault trees. ASTRA 3.x is fully based on the state-of-the-art of Binary Decision Diagrams (BDD) approach. In case of non-coherent fault trees ASTRA 3.x dynamically assigns to each node of the graph a label that identifies the type of the associated variable in order to drive the application of the most suitable analysis algorithms. The resulting BDD is referred to as Labelled BDD (LBDD). Exact values of the unavailability, expected number of failure and repair are calculated; the unreliability upper bound is automatically determined under given conditions. Five different importance measures of basic events are also provided. From the LBDD a ZBDD embedding all MCS is obtained from which a subset of Significant Minimal Cut Sets (SMCS) is determined through the application of the cut-off techniques.
An important issue is related to the analysis of safety related systems according to the IEC 61508 international standard. In order to simplify the fault tree modelling and analysis a new component type has been defined allowing determining, for any configuration, the $PFD_{avg}$ and $PFH_{avg}$ values. The Staggered testing policy is also applicable besides the Sequential testing implicitly considered by the IEC standard.

**How to obtain EU publications**

Our priced publications are available from EU Bookshop (http://bookshop.europa.eu), where you can place an order with the sales agent of your choice.

The Publications Office has a worldwide network of sales agents. You can obtain their contact details by sending a fax to (352) 29 29-42758.

The mission of the JRC is to provide customer-driven scientific and technical support for the conception, development, implementation and monitoring of EU policies. As a service of the European Commission, the JRC functions as a reference centre of science and technology for the Union. Close to the policy-making process, it serves the common interest of the Member States, while being independent of special interests, whether private or national.

JRC
EUROPEAN COMMISSION

Publications Office