Towards Formal Fault Tree Analysis using Theorem Proving

Waqar Ahmed and Osman Hasan

School of Electrical Engineering and Computer Science (SEECS)
National University of Sciences and Technology (NUST)
Islamabad, Pakistan
{waqar.ahmad,osman.hasan}@seecs.nust.edu.pk

Abstract. Fault Tree Analysis (FTA) is a dependability analysis technique that has been widely used to predict reliability, availability and safety of many complex engineering systems. Traditionally, these FTA-based analyses are done using paper-and-pencil proof methods or computer simulations, which cannot ascertain absolute correctness due to their inherent limitations. As a complementary approach, we propose to use the higher-order-logic theorem prover HOL4 to conduct the FTA-based analysis of safety-critical systems where accuracy of failure analysis is a dire need. In particular, the paper presents a higher-order-logic formalization of generic Fault Tree gates, i.e., AND, OR, NAND, NOR, XOR and NOT and the formal verification of their failure probability expressions. Moreover, we have formally verified the generic probabilistic inclusion-exclusion principle, which is one of the foremost requirements for conducting the FTA-based failure analysis of any given system. For illustration purposes, we conduct the FTA-based failure analysis of a solar array that is used as the main source of power for the Dong Fang Hong-3 (DFH-3) satellite.

Keywords: Higher-order Logic, Probabilistic Analysis, Theorem Proving, Satellite's Solar Arrays

1 Introduction

With the increasing usage of engineering systems in safety-critical domains, their dependability and failure analysis [1] has become a dire need to predict their reliability, availability and safety. One of the most widely used dependability and failure analysis techniques is the Fault Tree Analysis (FTA) method [2]. It is a graphical technique consisting of internal nodes, which are represented by gates like OR, AND and XOR, and the external nodes, that model the events which are associated with the occurrence of faults in sub-systems or components of the given system. The generic nature of these gates and events allows us to construct an efficient and accurate fault tree (FT) model for any given system. This FT can in turn be used to investigate the potential causes of a fault occurrence in a system and the calculation of minimal number of events that contribute

towards the occurrence of a *top event*, i.e., a critical event, which can cause the whole system failure upon its occurrence. Some noteworthy applications of FTA include the failure analysis of transportation systems [3], healthcare systems [4] and aerospace systems [5].

Traditionally, FTA is carried out by using paper-and-pencil proof methods, computer simulations and computer algebra systems. The first step in the paper-and-pencil proof methods is the construction of the FT of the given system on a paper. This is followed by the identification of the Minimal Cut Set (MCS) failure events, which contribute in the occurrence of the top event. These MCS failure events are generally modeled in terms of the exponential or weibull random variables and the Probabilistic Inclusion-Exclusion (PIE) principle [6] is then used to evaluate the exact probability of failure of the given system. However, this method is prone to human errors when it comes to the MCS and failure probability assessment of large safety-critical systems. For instance, in nuclear plants, where a fault tree model involves 50 to 130 levels of logic gates between the top event and the lowest basic events that are contributing to the top event [7]. So, there is a possibility, that many of these basic failure events may be overlooked while calculating MCS and thus not further incorporated in the FTA, which may lead to erroneous designs.

The FTA-based computer simulators, such as Relia-Soft [8] and ASENT Reliability analysis tools [9], provide graphical editors for the construction of FTs and the analysis is carried out by generating samples from the exponential and Weibull random variables that are associated with the events of the FT. These samples are then processed to evaluate the reliability and the failure probability of the complete system using computer arithmetic and numerical techniques. Although, these tools provide a more scalable alternative to the paper-and-pencil proof methods but the computational requirement increases drastically as the size of the FT increases. For example, if there are q terms involves in the MCS of a given FT then the total number of terms in the corresponding PIE principle will be $2q - 1$. In addition, these tools cannot ascertain absolute correctness or error-free analysis because of the involvement of pseudo random numbers and numerical methods and the inherent sampling-based nature of simulation.

Similarly, computer algebra systems (CAS), such as Mathematica [10], provide extensive features for FT-based failure analysis. For instance, the MCS expressions for any given system can be validated with failure distributions, such as Exponential or Weibull, by using symbolic and numerical algorithms. However, due to the presence of these unverified simplification algorithms, the analysis provided by CAS cannot be termed as sound and accurate.

Formal methods can overcome the above-mentioned inaccuracy limitations of the traditional techniques and thus have been used for FTA. The Interval Temporal Logic (ITS), i.e., a temporal logic that supports first-order logic, has been used, along with the Karlsruhe Interactive Verifier (KIV), for formal FTA of a rail-road crossing [11]. The work presented in [12] describes a deductive method for FT construction, in contrast to the intuitive approach followed in [11], by using the Observational Transition Systems (OTS) [12] and then the

formal analysis of this FT is carried out using CafeOBJ [13], which is a formal specification language with interactive verification support. One of the main limitations of all the above-mentioned formal methods based works is the inability to conduct a probability theoretic FTA. The COMPASS tool-set [14], which is developed at RWTH Achen University in collaboration with the European Space Agency (ESA), caters for this problem and supports the formal FTA specifically for aerospace systems using the NuSMV and MRMC model checkers. However, the scope of these tools is somewhat limited in terms of handling failure analysis of large FTs, due to the inherent state-space explosion problem of model checking, and the fact that the computation of probabilities in these methods involve numerical methods, which compromises the accuracy of the results.

An accurate MCS calculation and exact failure probability assessment in the FTA is very important specially while dealing with safety-critical systems used in domains like transportation, aerospace or medicine. In order to achieve an accurate and precise FTA, we propose to conduct the formal FTA within the sound core of a higher-order-logic theorem prover [15]. Higher-order logic provides a precise deductive mechanism that can be used to model any mathematically expressive behavior including recursive definitions, random variables, fault tree events, which are the foremost building blocks for modeling FTs. Once the FTs are modeled in higher-order logic, we can deduce an accurate MCS by using formal reasoning based on the set-theoretic foundations. Moreover, FT properties, such as the probability of failure, can be formally verified using interactive theorem provers based on the PIE principles.

The foremost requirement for reasoning about reliability and failure related properties of a system in a theorem prover is the availability of the higher-order-logic formalization of probability theory. Hurd's formalization of measure and probability theories [16] is a pioneering work in this regard. Building upon this formalization most of the commonly-used continuous random variables [17] and some reliability theory fundamentals [18] have been formalized using the HOL theorem prover. However, Hurd's formalization of probability theory [16] only supports the whole universe as the probability space. This feature limits its scope in many aspects [19] and one of the main limitations, related to FTA-based analysis, is the nonability to reason about multiple continuous random variables [17]. Some recent probability theory formalizations [19, 20] allow using any arbitrary probability space that is a subset of the universe and thus are more flexible than Hurd's formalization of probability theory. Particularly, Mhamdi's probability theory formalization [19], which is based on extended-real numbers (real numbers including $\pm\infty$), has been recently used to reason about the Reliability Block Diagram (RBD)-based analysis of a series pipelines structure [21], which involves multiple exponential random variables. The current paper is mainly inspired from this development as we use Mhamdi's formalized probability theory [19] for the formalization of all the commonly used FTA gates and the formal verification of their probabilistic properties. Moreover, we have also formally verified the PIE principle, which provides the foremost foundation for formal reasoning about the accurate failure analysis of any FT.

In order to illustrate the effectiveness of the proposed FTA approach, the paper presents a formal failure analysis, by taking a FT model, of a solar array that has been used in the DFH-3 Satellite, which was launched by the People's Republic of China on May 12, 1997 [5]. Solar arrays are one of the most vital components of the satellites because the mission success heavily depends upon the continuous reliable source of power [22]. Over the last ten years, 12 out of the 117 satellite's solar array anomalies, documented by the Airclaims Ascend SpaceTrak database, led to the total satellite failure [23, 22]. Thus the absolute accuracy of the failure analysis of a solar array is a dire need in satellite missions and, to the best of our knowledge, it is the novelty of the proposed technique to meet this requirement. The satellite's solar array is a mechanical system, which mainly consists of various mechanisms, including: deployable, synchronization, locking and orientation. The FT of the solar array contains the failure events of these mechanisms and their interrelationships regarding the overall system failure. The paper presents the higher-order-logic modeling of this FT and the formal verification of the probability of failure of satellite's solar array system based on the probability of occurrence of the above-mentioned mechanism faults.

2 Probability Theory in HOL

In this section, we provide a brief overview of the HOL4 formalization of the probability theory [19], which we build upon in this paper. Based on the measure theoretic foundations, a probability space is defined as a triple (Ω, Σ, Pr), where Ω is a set, called the sample space, Σ represents a σ-algebra of subsets of Ω, where the subsets are usually referred to as measurable sets, and Pr is a measure with domain Σ and is 1 for the whole sample space. In the HOL4 probability theory formalization [19], given a probability space p, the functions `space` and `subsets` return the corresponding Ω and Σ, respectively. Based on this definition, all the basic probability axioms have been verified. Now, a random variable is a measurable function between a probability space and a measurable space, which essentially is a pair $(S, \mathcal{A})$, where S denotes a set and $\mathcal{A}$ represents a nonempty collection of sub-sets of S. A random variable is termed as discrete if S is a set with finite elements and continuous otherwise.

The cumulative distribution function (CDF) is defined as the probability of the event where a random variable X has a value less than or equal to some value x, i.e., $Pr(X \leq x)$. This definition characterizes the distribution of both discrete and continuous random variables and has been formalized [21] as follows:

```
⊢ ∀ p X x. CDF p X x = distribution p X {y | y ≤ Normal x}
```

The function `Normal` takes a *real* number as its input and converts it to its corresponding value in the *extended-real* data-type, i.e, it is the *real* data-type with the inclusion of positive and negative infinity. The function `distribution` takes three parameters: a probability space p, a random variable X and a set of *extended-real* numbers and returns the probability of the given random variable X acquiring all the values of the given set in probability space p.

The unreliability or the probability of failure $F(t)$ is defined as the probability that a system or component will fail by the time t. It can be described in terms of CDF, known as the failure distribution function, if the random variable X represent a time-to-failure of the component. This time-to-failure random variable X usually exhibits the exponential or weibull distribution.

The notion of mutual independence of n random variables is a major requirement for reasoning about the failure analysis of most of the FT gates. According to this notion, if we have N mutually independent failure events then

$$Pr(\bigcap_{i=1}^{N} L_i) = \prod_{i=1}^{N} Pr(L_i) \tag{1}$$

This concept has been formalized as follows [21]:

```
⊢ ∀ p L. mutual_indep p L = ∀ L1 n. PERM L L1 ∧
     1 ≤ n ∧ n ≤ LENGTH L ⇒
      prob p (inter_list p (TAKE n L1)) =
      list_prod (list_prob p (TAKE n L1))
```

The function `mutual_indep` accepts a list of events L and probability space p and returns $True$ if the events in the given list are mutually independent in the probability space p. The predicate `PERM` ensures that its two list arguments form a permutation of one another. The function `LENGTH` returns the length of the given list. The function `TAKE` returns the first n elements of its argument list as a list. The function `inter_list` performs the intersection of all the sets in its argument list of sets and returns the probability space if the given list of sets is empty. The function `list_prob` takes a list of events and returns a list of probabilities associated with the events in the given list of events in the given probability space. Finally, the function `list_prod` recursively multiplies all the elements in the given list of real numbers. Using these functions, the function `mutual_indep` models the mutual independence condition such that for any 1 or more events n taken from any permutation of the given list L, the property $Pr(\bigcap_{i=1}^{N} L_i) = \prod_{i=1}^{N} Pr(L_i)$ holds.

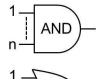
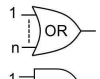
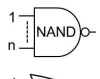
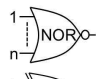
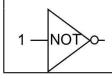
3 Formalization of Fault Tree Gates

In this section, we describe a generic formalization of commonly used FT gates given in Table 1. Our formalizations are generic in terms of the number of inputs n, i.e., our definitions can be used to model arbitrary-input FT gates.

3.1 Formal Definitions of Fault Tree Gates

If the occurrence of the output failure event is caused by the occurrence of all the input failure events then this kind of behavior can be modeled by using the AND FT gate. The function `AND_FT_gate`, given in Table 1, models this behavior as

Table 1: HOL4 Formalization of Fault Tree Gates

Fault Tree Gates	HOL Formalization
	⊢ ∀ p L. AND_FT_gate p L = inter_list p L
	⊢ ∀ L. OR_FT_gate L = union_list L
	⊢ ∀ p L1 L2. NAND_FT_gate p L1 L2 = inter_list p (compl_list p L1) ∩ inter_list p L2
	⊢ ∀ p L. NOR_FT_gate p L = p_space p DIFF (OR_gate L)
	⊢ ∀ p A B. XOR_FT_gate p A B = ((p_space p DIFF A ∩ B) ∪ (A ∩ p_space p DIFF B))
	⊢ ∀ p A. NOT_FT_gate p A = (p_space p DIFF A)

it accepts an arbitrary probability space p and returns the intersection of input failure events, given in the list L, by using the recursive function `inter_list`.

In the OR FT gate, the occurrence of the output failure event depends upon the occurrence of any one of its input failure event. The function `OR_FT_gate`, given in Table 1, models this behavior as it returns the union of the input failure list L by using the recursive function `union_list`. The NOR FT gate can be viewed as the complement of the OR FT gate and its output failure event occurs if none of the input failure event occurs.

The NAND FT gate models the behavior of the occurrence of an output failure event when at least one of the failure events at its input does not occur. This type of gate is used in FTs when the non-occurrence of the failure event in conjunction with the other failure events causes the top failure event to occur. This behavior can be expressed as the intersection of complementary and normal events [1], where the complementary events model the non-occurring failure events and the normal events model occurring failure events. It is important to note that the behavior of the NAND FT gate is usually not captured by the complement of the AND FT gate in the FTA literature [1]. The function `NAND_FT_gate` accepts a probability space p and two list of failure events $L1$ and $L2$. The function returns the intersection of non-occurring failure events, which in turn is modeled by passing the list of failure events $L1$ to the recursive function `compl_list`, and occurring failure events, which are given in the list $L2$, by utilizing the recursive function `inter_list`. The function `compl_list` returns a list of events such that each element of this list is the difference between the probability space p and the corresponding element of the given list.

The output failure event occurs in the 2-input XOR FT gate if only one, and not both, of its input failure events occur. The HOL representation of the behaviour of the XOR FT gate is presented in Table 1. The function `NOT_FT_gate`

accepts an arbitrary failure event A along with probability space p and returns the complement to the probability space p of the given input failure event A.

3.2 Formal Verification of Failure Probability of Fault Tree Gates

The function `AND_FT_gate`, given in Table 1, can be used to evaluate the failure probability of the output failure event of the AND FT gate. If A_i represents the i^{th} failure event with failure probability F_i at time t among the n mutually independent failure events of the AND FT gate then the generic mathematical expression for the failure probability of a n-input AND FT gate is as follows:

$$F_{AND\_gate}(t) = Pr(\bigcap_{i=2}^{N} A_i(t)) = \prod_{i=2}^{N} F_i(t) \tag{2}$$

We formally verified this expression as the following theorem in HOL4:

Theorem 1: `⊢ ∀ p L. prob_space p ∧`
`2 ≤ LENGTH L ∧ mutual_indep p L ⇒`
`(prob p (AND_gate p L) = list_prod (list_prob p L))`

The first assumption ensure that p is a valid probability space based on the probability theory in HOL4 [19]. The next two assumptions guarantee that the list of failure events must have at least two failure event and the failure events are mutually independent, respectively. The conclusion of the theorem represents Equation (2). The proof of Theorem 1 is primarily based on some probability theory axioms and the mutual independence definition.

Similarly, if A_i represents the i^{th} with failure event failure probability F_i at time t among the n mutually independent failure events of an OR FT gate then its failure probability expression is as follows:

$$F_{OR\_gate}(t) = Pr(\bigcup_{i=2}^{N} A_i(t)) = 1 - \prod_{i=2}^{N}(1 - F_i(t)) \tag{3}$$

In order to formally verify the above equation, we first formally verify the following lemma that provides an alternate expression for the failure probability of an OR FT gate in terms of the failure probability of an AND FT gate:

Lemma 1:
```
⊢ ∀ L p. (prob_space p) ∧
(∀ x'. MEM x' L ⇒ x' ∈ events p) ⇒
  (prob p (OR_gate L) =
  1 - prob p (AND_gate p (compl_list p L))
```

Now, we can formally verify Equation (3) in HOL4 as follows:

Theorem 2:
```
⊢ ∀ p L. (prob_space p) ∧
(2 ≤ LENGTH L) ∧ (mutual_indep p L) ∧
 (∀ x'. MEM x' L ⇒ x' ∈ events p) ⇒
  (prob p (OR_gate L) =
  1 - list_prod (one_minus_list (list_prob p L)))
```

Where the function `one_minus_list` accepts a list of *real* numbers $[x_1, x_2, \cdots, x_n]$ and returns the list of *real* numbers such that each element of this list is 1 minus the corresponding element of the given list, i.e., $[1-x_1, 1-x_2, \cdots, 1-x_n]$. The proof of Theorem 2 is primarily based on Lemma 1 and Theorem 1 along with the fact that given the list of n mutually independent events, the complement of these n events are also mutually independent.

Similarly, we also verified the failure probability theorems for other FT gates, given in Table 1, and the corresponding mathematical expressions and theorems are given in Table 2. All these results are verified under the same assumptions as the ones used in Theorems 1 and 2.

Table 2: Probability of Failure of Fault Tree Gates

Fault Tree Gates	Theorem's Conclusion

$F_{NOR}(t) = 1 - F_{OR}(t)$
 $= \prod_{i=2}^{N}(1 - F_i(t))$
	`(prob p (NOR_FT_gate p L) =`
	 `list_prod (one_minus_list`
	 `(list_prob p L)))`

$F_{NAND}(t) = Pr(\bigcap_{i=2}^{k} \overline{A}_i(t) \cap \bigcap_{j=k}^{N} A_i(t))$
 $= \prod_{i=2}^{k}(1 - F_i(t)) * \prod_{j=k}^{N}(F_j(t))$
	`(prob p (NAND_FT_gate p L1 L2) =`
	 `list_prod ((list_prob p`
	 `(compl_list p L1))) *`
	 `list_prod (list_prob p L2))`

$F_{XOR}(t) = Pr(\bar{A}(t)B(t) \cup A(t)\bar{B}(t))$
 $= (1 - F_A(t))F_B(t) +$
 $F_A(t)(1 - F_B(t))$
	`(prob p (XOR_FT_gate p A B) =`
	 `(1- prob p A)*prob p B +`
	 `prob p A*(1 - prob p B)`

$F_{NOT}(t) = Pr(A(t))$ $= (1 - F_A(t))$	`prob p (NOT_FT_gate p A) =` `(1 - prob p A)`

The proof script [24] of the above-mentioned formalization is composed of 4000 lines of HOL script and took about 200 man-hours. The main outcome of this exercise is that the definitions, given in Table 1, can be used to capture the behavior of most of the FTs in higher-order logic and the Theorems of Table 2 can then be used in conjunction with the formalization of the PIE principle, explained next, to formally verify the corresponding failure probabilities.

4 Formalization of Probabilistic Inclusion-Exclusion Principle

The probabilistic inclusion-exclusion principle (PIE) forms an integral part of the reasoning involved in verifying the failure probability of a FT. In FTA, firstly all the basic fault events are identified that can cause the occurrence of the system failure event. These fault events are then combined to model

the overall fault behavior of the given system by using the fault gates. These combinations of basic failure events, called cut sets, are then reduced to minimal cut sets (MCS) by using some set-theory rules, such as idempotent, associative and commutative [25]. At this point, the PIE principle is used to evaluate the overall failure probability of the given system based on the MCS events.

If A_i represent the i^{th} basic failure event or a combination of failure event then the failure probability of the given system can be expressed in terms of the probabilistic inclusion-exclusion principle as follows:

$$\mathbb{P}(\bigcup_{i=1}^{n} A_i) = \sum_{t \neq \{\}, t \subseteq \{1,2,\ldots,n\}} (-1)^{|t|+1} \mathbb{P}(\bigcap_{j \in t} A_j) \tag{4}$$

The above equation can be formalized in HOL4 is as follows:

Theorem 3: ⊢ ∀ p L1 L2. prob_space p ∧
(∀ x. MEM x L ⇒ x ∈ events p) ⇒
    (prob p (union_list L) =
    sum_set {t | t ⊆ set L ∧ t ≠ {} }
        (λt. -1 pow (CARD t + 1) * prob p (BIGINTER t)))

The assumptions of the above theorem are the same as the ones used in Theorem 1. The function `sum_set` takes an arbitrary set s with element of type α and a real-valued function f. It recursively sums the return value of the function f, which is applied on each element of the given set s. In the above theorem, the set s is represented by the term $\{x|C(x)\}$ that contains all the values of x, which satisfy condition C. Whereas, the λ abstraction function `(λt. -1 pow (CARD t + 1) * prob p (BIGINTER t))` models $(-1)^{|t|+1}\mathbb{P}(\bigcap_{j \in t} A_j)$, such that the functions `CARD` and `BIGINTER` return the number of elements and the intersection of all the elements of the given set, respectively. Thus, the conclusion of the theorem represents Equation (4).

The formal reasoning about Theorem 3 is based upon the following lemma:

Lemma 2: ⊢ ∀ P. (∀ n. (∀ m. m < n ⇒ P m) ⇒ P n) ⇒ ∀ n. P n

Where n in our case is the length of the list L and m represent another list whose length is less then the length of the list L. The predicate P represents the conclusion of Theorem 3. The above property brings an important hypothesis in the assumption list, which has the same form as that of the conclusion of Theorem 3. Then, by utilizing induction and some properties of the function `sum_set` along with some fundamental axioms of probability, we can verify Theorem 3.

The proof script [24] for Theorem 3 is composed of 1000 lines of HOL code and involved 50 man-hours of proof effort. To the best of our knowledge, this is the first formal verification of the probabilistic inclusion exclusion principle, which, besides being used in FTA, is a widely used mathematical result in analyzing various bio-informatics [26] and telecommunication [27] systems.

5 Application: Satellite's Solar Array

The solar arrays used in satellite missions are usually in a folded position during the launch phase [5]. Once the satellite is deployed in the corresponding orbit then the solar arrays are unfolded and the goal is to keep them oriented towards the sun all the time to maximize the power generation for the satellite [5]. The faults in the solar array are mainly caused by the mechanical components that drive these mechanisms associated with the driving, deployment, synchronization, locking and orientation. For example, the solar array is usually driven by using a torsion spring [5]. Whereas, the closed cable loop (CCL) and the stepping or servo motors are used during the synchronization and orientation phases [5]. A FT can thus be constructed by considering the faults in these mechanical components, which are the fundamental causes of satellite' solar array mechanisms failure The FT for the solar array of the DFH-3 Satellite that was launched by the People's Republic of China on May 12, 1997 [28] is depicted in Figure 1 and we formally analyze this FT in this paper.

The failure events, *A, B, C, D* in Figure 1, represent the failures in the unlock mechanism, deployment process, locking process and orientation process, respectively. Whereas, the failure event E represents the failures in the corresponding mechanical parts of the system. These failure events are combined either by using the OR or AND FT gates by considering the behavior of the faults.

In order to formalize the solar array FT of Figure 1, we first present the formal modeling of list of failure events that are associated with each corresponding fault of the solar array FT.

Definition 1: ⊢ ∀ p x. fail_event_list p [] x = [] ∧
∀ p x h t. fail_event_list p (h::t) x =
PREIMAGE h {y | y ≤ Normal x } ∩ p_space p ::
fail_event_list p t x

The function `fail_event_list` accepts a probability space p, a list of random variables, representing the failure time of individual components, and a real number x, which represents the time index at which the failure of the component occurs. It returns a list of events, representing the failure of all the individual components at time x. The formal definitions of FT gates, given in Section 3, along with Definition 1 can be utilized to formally represent the FT of satellite's solar array in terms of its cut-set failure events. The HOL4 formalization of satellite's solar array FT is as follows:

Definition 2:
```
⊢ ∀ p x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11
                        x12 x13 x14 t.
Solar_FT p x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 t =
OR_FT_gate [OR_FT_gate (fail_event_list p [x1; x2] t);
            OR_FT_gate [OR_FT_gate (fail_event_list p [x3; x4] t);
                AND_FT_gate p (fail_event_list p [x5; x6] t); OR_FT_gate
(fail_event_list p [x3; x7; x8] t)];
            OR_FT_gate (fail_event_list p [x3; x9] t);
            OR_FT_gate (fail_event_list p [x10; x11] t);
```

```
OR_FT_gate [PREIMAGE x12 {y | y ≤ Normal t };
            PREIMAGE x13 {y | y ≤ Normal t };
            OR_FT_gate (fail_event_list p[x3; x14]t)]]
```

Where the random variables $x1 - x14$ model the time-to-failure of the solar array processes and components as depicted in Figure 1. However, the cut-set

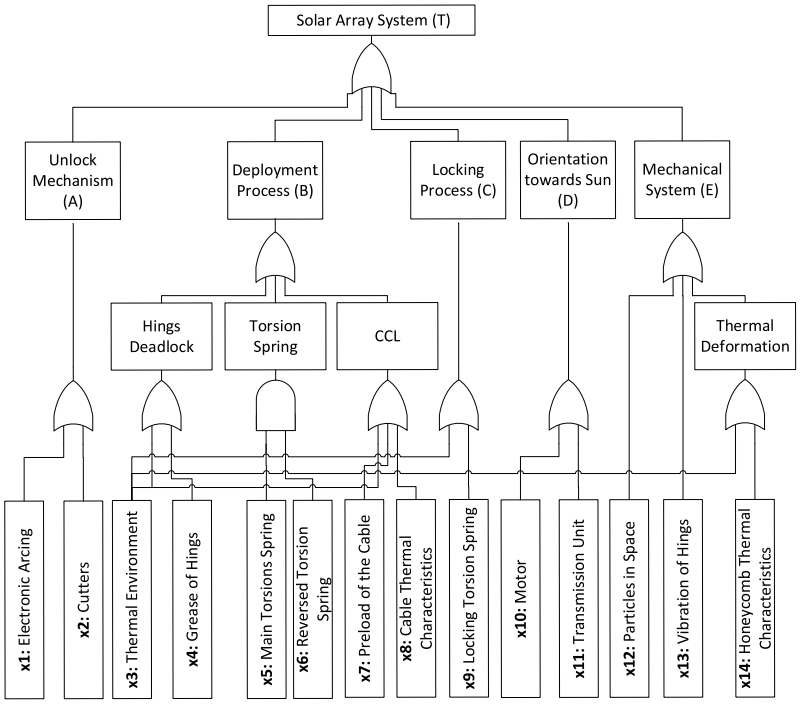


Fig. 1: FT of the Solar Array of the DFH-3 Satellite [5]

failure events in the above definition is not minimal [5], i.e., there are some redundant failure events. For example, $x3$ is part of more than one OR FT gates. These kind of redundant failure events can be removed by verifying an accurate equivalent but reduced representation, i.e., the MCS, by using set theory laws, like idempotent, commutative and associative, as follows:

Lemma 2:
```
⊢ ∀ p x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 t.
      prob_space p ⇒
(Solar_FT p x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 t =
OR_FT_gate [OR_FT_gate (fail_event_list p [x1; x2; x3; x4] t);
            AND_FT_gate p (fail_event_list p [x5; x6] t);
            OR_FT_gate
             (fail_event_list p [x7; x8; x9; x10; x11; x12; x13; x14]t)])
```

We consider that random variables, associated with the failure events of the solar array FT, exhibit the exponential distribution, which can be formalized in HOL4 as follows:

Definition 3: ⊢ ∀ p X l. exp_dist p X l =
∀ x. (CDF p X x = if 0 ≤ x then 1 - exp (-l * x) else 0)

The function exp_dist guarantees that the CDF of the random variable X is that of an exponential random variable with a failure rate l in a probability space p. We classify a list of exponentially distributed random variables based on this definition as follows:

Definition 4: ⊢ ∀ p L. list_exp p [] L = T ∧
∀ p h t L. list_exp p (h::t) L =
exp_dist p (HD L) h ∧ list_exp p t (TL L)

The function list_exp accepts a list of failure rates, a list of random variables L and a probability space p. It guarantees that all elements of the list L are exponentially distributed with the corresponding failure rates, given in the other list, within the probability space p. For this purpose, it utilizes the list functions HD and TL, which return the *head* and *tail* of a list, respectively. Now, the failure probability of satellite's solar array can be verified as the following theorem:

Theorem 4:
```
⊢ ∀ p x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 t c1
c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14.
(0 ≤ t) ∧ (prob_space p) ∧
(∀ x'. MEM x' (fail_event_list p
 ([x1; x2; x3; x4; x5;
   x6; x6; x7; x8; x9; x10; x11; x12; x13; x14]) t)) ⇒ x' ∈ events p) ∧
(mutual_indep p ((fail_event_list p
  ([x1; x2; x3; x4; x5; x6; x7; x8; x9; x10; x11; x12; x13; x14]) x))) ∧
list_exp p
 ([c1; c2; c3; c4; c5; c6; c7; c8; c9; c10; c11; c12; c13; c14])
    ([x1; x2; x3; x4; x5; x6; x7; x8; x9; x10; x11; x12; x13; x14]) ⇒
  (prob p (Solar_FT p
    x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 t ) =
  (1 - (exp -(t*(list_sum [c1;c2;c3;c4])))) +
    list_prod(one_minus_exp t [c5;c6;c7]) +
    (1 - (exp -(t*(list_sum
      [c7; c8; c9; c10; c11; c12; c13; c14])))) -
    (1 - list_prod(one_minus_exp_prod t
      [[c1;c5;c6];[c2;c5;c6];[c3;c5;c6];[c4;c5;c6]])) -
    (1 - (exp -(t*(list_sum [c1;c2;c3;c4])))) *
      (1 - (exp -(t*(list_sum
        [c7; c8; c9; c10; c11; c12; c13; c14])))) -
    (1 - list_prod(one_minus_exp_prod t
      [[c5;c6;c7];[c5;c6;c8];[c5;c6;c9];[c5;c6;c10];
        [c5;c6;c11];[c5;c6;c12];[c5;c6;c13];[c5;c6;c14]])) +
    (1 - list_prod(one_minus_exp_prod t
      [[c1;c5;c6];[c2;c5;c6];[c3;c5;c6];[c4;c5;c6]])) *
```

```
(1 - (exp -(t*
  (list_sum [c7; c8; c9; c10; c11; c12; c13; c14])))))
```

The first assumption ensures the variable `t` that models time can acquire positive values only. The second assumption ensure that `p` is a valid probability space based on the probability theory in HOL4 [29]. The next two assumptions ensure that the events corresponding to the failures modeled by the random variables `x1` to `x14` are valid events from the probability space `p` and they are mutually exclusive. Finally, the last assumption characterizes the random variables `x1` to `x14` as exponential random variables with failure rates `c1` to `c14`, respectively. The conclusion of the Theorem 4 represents the failure probability of the given solar array in terms of the failure rates of its components as follows:

$$
\begin{aligned}
&(1 - e^{-(c1+c2+c3+c4)t}) + \prod_{i=5}^{6}(1 - e^{-(c_i t)}) + \\
&(1 - e^{-(c7+c8+c9+c10+c11+c12+c13+c14)t}) - (1 - \prod_{i=1}^{4}(1 - \prod_{j=5}^{6}[(1 - e^{-c_i t})(1 - e^{-c_j t})])) - \\
&(1 - e^{-(c1+c2+c3+c4)t}) * (1 - e^{-(c7+c8+c9+c10+c11+c12+c13+c14)t}) - \\
&(1 - \prod_{i=7}^{14}(1 - \prod_{j=5}^{6}[(1 - e^{-c_i t})(1 - e^{-c_j t})])) + \\
&(1 - \prod_{i=1}^{4}(1 - \prod_{j=5}^{6}[(1 - e^{-c_i t})(1 - e^{-c_j t})])) * (1 - e^{-(c7+c8+c9+c10+c11+c12+c13+c14)t})
\end{aligned}
\tag{5}
$$

where the function `exp` represents a exponential function, the function `list_sum` is used to sum all the element of the given list of failure rates, the function `one_minus_exp` accepts a list of failure rates and returns a one minus list of exponentials and the function `one_minus_exp_prod` accepts a two dimensional list of failure rates and returns a list with one minus product of one minus exponentials of every sub-list. For example, `one_minus_exp_prod`$[[c1; c2; c3]; [c4; c5]; [c6; c7; c8]]$ $x = [1 - ((1 - e^{-(c1)x}) * (1 - e^{-(c2)x}) * (1 - e^{-(c3)x}))$; $(1 - (1 - e^{-(c4)x}) * (1 - e^{-(c5)x}))$; $(1 - (1 - e^{-(c6)x}) * (1 - e^{-(c7)x}) * (1 - e^{-(c8)x}))]$.

The proof of the above theorem utilizes the failure probabilities of AND and OR FT gates, given in Table 2, along with Lemma 2 and Theorem 3 and some fundamental facts and axioms of probability theory. Due to the universally quantified variables in Theorem 3, the proof of Theorem 4 is quite straightforward (about 800 lines of HOL code) as compared to that of Theorem 3. The distinguishing features of the formally verified Theorem 4 includes its generic nature, i.e., all the variables are universally quantified and thus can be specialized to obtain the failure probability for any given failure rates, and its guaranteed correctness due to the involvement of a sound theorem prover in its verification, which ensures that all the required assumptions for the validity of the result are accompanying the theorem.

A fuzzy reasoning Petri Net (FRPN), which is a combination of fuzzy logic [30] and Petri Nets [31], based failure analysis for the above-mentioned solar array is presented in [5]. In this work, the FT of Figure 1 is first represented as a Petri Net such that the gates are represented by transitions and the failure events are modeled as places. The possibility of fault occurrence is then evaluated by using fuzzy degree of truth on the basis of petri nets transitions. However, the truth degree values evaluated using these FRPN models cannot be regarded as precise and sound as the formally verified expression using the HOL theorem prover due to the involvement of numerical techniques and pseudo randomness. On the other hand, our analysis result, i.e., Theorem 4, is based on a probability theoretic formal reasoning, verified in a sound theorem prover and is valid for all possible values of the failure rates. These features constitute the main motivations of the work presented in this paper.

6 Conclusion

The accuracy of failure analysis is a dire need for safety and mission-critical applications, where an incorrect failure analysis may lead to disastrous situations including the loss of human lives or heavy financial setbacks. In this paper, we presented an accurate FTA approach, based on higher-order-logic theorem proving, to tackle the analysis of such critical systems. In particular the paper presents a formalization of commonly used FT gates and the PIE principle, which are the foremost foundations for formal reasoning about FTA within a sound core of theorem prover. As a case-study, the paper also presents the formal failure analysis of a satellite's solar array.

Building upon the results, presented in this paper, other FT gates, such as priority AND and voting OR gate, can also be formally modeled and thus the scope of FTA-based formal reliability analysis [32] can be further enhanced. Some interesting real-world applications that can benefit from our work include transportation systems [3], healthcare systems [4] and avionics [33]. Moreover, we also plan to further facilitate the formal FT-based failure analysis by incorporating the automatic simplification capabilities of CAS, such as Mathmatica, for MCS calculation. This obtained MCS can then be validated within the sound environment of the HOL theorem prover.

References

1. IEC: International Electrotechnical Commission, 61025 Fault Tree Analysis (2006)
2. Roberts, N.H., Vesely, W.E.: Fault Tree Handbook. Government Printing (1987)
3. Huang, H.Z., Yuan, X., Yao, X.S.: Fuzzy Fault Tree Analysis of Railway Traffic Safety. In: Conference on Traffic and Transportation Studies, American Society of Civil Engineers (2000) 107–112
4. Hyman, W.A., Johnson, E.: Fault Tree Analysis of Clinical Alarms. Journal of Clinical Engineering **33**(2) (2008) 85–94

5. Wu, J., Yan, S., Xie, L.: Reliability Analysis Method of a Solar Array by using Fault Tree Analysis and Fuzzy Reasoning Petri Net. Acta Astronautica **69**(11) (2011) 960–968
6. Trivedi, K.S.: Probability and Statistics with Reliability, Queuing and Computer Science Applications. 2nd edn. John Wiley and Sons Ltd. (2002)
7. Epstein, S., Rauzy, A.: Can we trust PRA? Reliability Engineering & System Safety **88**(3) (2005) 195–205
8. ReliaSoft: http://www.reliasoft.com/ (2015)
9. ASENT: https://www.raytheoneagle.com/asent/rbd.htm (2015)
10. Long, W., Sato, Y., Horigome, M.: Quantification of Sequential Failure Logic for Fault Tree Analysis. Reliability Engineering & System Safety **67**(3) (2000) 269–274
11. Ortmeier, F., Schellhorn, G.: Formal Fault Tree Analysis-Practical Experiences. Volume 185., Elsevier (2007) 139–151
12. Xiang, J., Futatsugi, K., He, Y.: Fault Tree and Formal Methods in System Safety Analysis. In: Computer and Information Technology, IEEE (2004) 1108–1115
13. Futatsugi, K., Nakagawa, A.T., Tamai, T.: CAFE: An Industrial-Strength Algebraic Formal Method. Elsevier (2000)
14. Bozzano, M., Cimatti, A., Katoen, J.P., Nguyen, V.Y., Noll, T., Roveri, M.: The COMPASS Approach: Correctness, Modelling and Performability of Aerospace Systems. In: Computer Safety, Reliability, and Security. Volume 5775 of LNCS. Springer (2009) 173–186
15. Harrison, J.: Handbook of Practical Logic and Automated Reasoning. Cambridge University Press (2009)
16. Hurd, J.: Formal Verification of Probabilistic Algorithms. PhD Thesis, University of Cambridge, UK (2002)
17. Hasan, O., Tahar, S.: Formalization of the Continuous Probability Distributions. In: Automated Deduction. Volume 4603 of LNAI. Springer (2007) 3–18
18. Abbasi, N., Hasan, O., Tahar, S.: An Approach for Lifetime Reliability Analysis using Theorem Proving. Journal of Computer and System Sciences **80**(2) (2014) 323–345
19. Mhamdi, T., Hasan, O., Tahar, S.: On the Formalization of the Lebesgue Integration Theory in HOL. In: Interactive Theorem Proving. Volume 6172 of LNCS. Springer (2011) 387–402
20. Holzl, J., Heller, A.: Three Chapters of Measure Theory in Isabelle/HOL. In: Interactive Theorem Proving. Volume 6172 of LNCS. Springer (2011) 135–151
21. Ahmed, W., Hasan, O., Tahar, S., Hamdi, M.S.: Towards the Formal Reliability Analysis of Oil and Gas Pipelines. In: Intelligent Computer Mathematics. Volume 8543 of LNCS. Springer (2014) 30–44
22. Brandhorst Jr, H.W., Rodiek, J.A.: Space Solar Array Reliability: A Study and Recommendations. Volume 63. Elsevier (2008) 1233–1238
23. Airclaims Ascend SpaceTrak Database: www.ascendspacetrak%20com/home (2014)
24. Ahmad, W.: Formal Fault Tree Analysis of Satellite's Solar Array. http://save.seecs.nust.edu.pk/projects/fta.html (2014)
25. Halmos, P.R.: Naive set theory. Springer (1960)
26. Todor, A., Gabr, H., Dobra, A., Kahveci, T.: Large Scale Analysis of Signal Reachability. Bioinformatics **30**(12) (2014) 96–104
27. Gao, F., Liu, X., Liu, H.: A rapid algorithm for computing st reliability of radiocommunication networks. In Apolloni, B., Howlett, R., Jain, L., eds.: Knowledge-Based Intelligent Information and Engineering Systems. Volume 4693 of LNCS. Springer (2007) 167–174

28. Jianing, W., Shaoze, Y.: Reliability Analysis of the Solar Array based on Fault Tree Analysis. In: Journal of Physics. Volume 305., IOP Publishing (2011) 012006
29. Mhamdi, T., Hasan, O., Tahar, S.: On the Formalization of the Lebesgue Integration Theory in HOL. In: Interactive Theorem Proving. LNCS. Springer (2010) 387–402
30. Zadeh, L.A.: Toward a Theory of Fuzzy Information Granulation and its Centrality in Human Reasoning and Fuzzy logic. Volume 90. Elsevier (1997) 111–127
31. Peterson, J.L.: Petri Net Theory and the Modeling of Systems. Prentice Hall (1981)
32. Volkanovski, A., Čepin, M., Mavko, B.: Application of the Fault Tree Analysis for Assessment of Power System Reliability. Reliability Engineering & System Safety, Elsevier, **94**(6) (2009) 1116–1127
33. Lefebvre, A., Simeu-Abazi, Z., Derain, J.P., Glade, M., et al.: Diagnostic of the Avionic Equipment Based on Dynamic Fault Tree. In: IFAC-CEA conference. (2007)