Formal FT-based Cause-Consequence Reliability Analysis using Theorem Proving

Mohamed Abdelghany and Sofiène Tahar

Department of Electrical and Computer Engineering,
Concordia University, Montréal, QC, Canada

{m_eldes,tahar}@ece.concordia.ca

TECHNICAL REPORT

January 2021

Abstract

Cause-consequence Diagram (CCD) is widely used as a deductive safety analysis technique for decision-making at the critical-system design stage. This approach models the causes of subsystem failures in a highly-critical system and their potential consequences using Fault Tree (FT) and Event Tree (ET) methods, which are well-known dependability modeling techniques. Paper-and-pencil-based approaches and simulation tools, such as the Monte-Carlo approach, are commonly used to carry out CCD analysis, but lack the ability to rigorously verify essential system reliability properties. In this work, we propose to use formal techniques based on theorem proving for the formal modeling and step-analysis of CCDs to overcome the inaccuracies of the simulation-based analysis and the error-proneness of informal reasoning by mathematical proofs. In particular, we use the HOL4 theorem prover, which is a computer-based mathematical reasoning tool. To this end, we developed a formalization of CCDs in Higher-Order Logic (HOL), based on the algebraic approach, using HOL4. We demonstrate the practical effectiveness of the proposed CCD formalization by performing the formal reliability analysis of the IEEE 39-bus electrical power network. Also, we formally determine the Forced Outage Rate ($\mathcal{FOR}$) of the power generation units and the network reliability index, i.e., System Average Interruption Duration Index ($\mathcal{SAIDI}$). To assess the accuracy of our proposed approach, we compare our results with those obtained with MATLAB Monte-Carlo Simulation (MCS) as well as other state-of-the-art approaches for subsystem-level reliability analysis.

Keywords— Cause-Consequence Diagram, Event Tree, Fault Tree, Reliability Analysis, Safety, Formal Methods, Theorem Proving, HOL4, Monte-Carlo, FMECA, Electrical Power Network, FOR, SAIDI.

1 Introduction

Nowadays, in many safety-critical systems, which are prevalent, e.g. in smart grids [1] and automotive industry [2], a catastrophic accident may happen due to coincidence of sudden events and/or failures of specific subsystem components. These undesirable accidents may result in loss of profits and sometimes severe fatalities. Therefore, the central inquiry, in many critical-systems, where safety is of the utmost importance, is to identify the possible consequences given that one or more components could fail at a subsystem level on the entire system. For that purpose, the main discipline for safety design engineers is to perform a detailed Cause-Consequence Diagram (CCD) [3] reliability analysis for identifying the subsystem events that prevent the entire system from functioning as desired. This approach models the causes of component failures and their consequences on the entire system using Fault Tree (FT) [4] and Event Tree (ET) [5] dependability modeling techniques.

FTs mainly provide a graphical model for analyzing the factors causing a system failure upon their occurrences. FTs are generally classified into two categories Static Fault Trees (SFT) and Dynamic Fault Trees (DFT) [6]. SFTs and DFTs allow safety-analysts to capture the static/dynamic failure characteristics of systems in a very effective manner using *logic*-gates, such as OR, AND, NOT, Priority-AND (PAND) and SPare (SP) [4]. However, the FT technique is incapable of identifying the possible consequences resulting from an undesirable failure on the entire system. ETs provide risk analysis with all possible system-level operating states that can occur in the system, i.e., success and failure, so that one of these possible scenarios can occur [5]. However, both of these modeling techniques are limited to analyzing either a critical-system failure or cascading dependencies of system-level components only, respectively.

There exist some techniques that have been developed for subsystem-level reliability analysis of safety-critical systems. For instance, Papadopoulos et al. in [7] have developed a software tool called *HiP-HOPS* (Hierarchically Performed Hazard Origin & Propagation Studies) [8] for subsystem-level failure analysis to overcome classical manual failure analysis of complex systems and prevent human errors. HiP-HOPS can automatically generate the subsystem-level FT and perform Failure Modes, Effects, and Critically Analyses (FEMCA) from a given system model, where each system component is associated with its failure rate or failure probability [7]. Currently, HiP-HOPS lacks the modeling of *multi-state* system components and also cannot provide generic mathematical expressions that can be used to predict the reliability of a critical-system based on any probabilistic distribution [9]. Similarly, Jahanian in [10] has proposed a new technique called Failure Mode Reasoning (FMR) for identifying and quantifying the failure modes for safety-critical systems at the subsystem level. However, according to Jahanian [11], the soundness of the FMR approach needs to be proven mathematically.

On the other hand, CCD analysis typically uses FTs to analyze failures at the subsystem or component level combined with an ET diagram to integrate their

cascading failure dependencies at the system level. CCDs are categorized into two general methods for the ET linking process with the FTs [12]: (1) Small ET diagram and large subsystem-level FT; (2) Large ET diagram and small subsystem-level FT. The former one with small ET and large subsystem-level FT is the most commonly used for the probabilistic safety assessment of industrial applications (e.g., in [13]). There are *four* main steps involved in the CCD analysis [14]: (1) *Component failure events*: identify the causes of each component failure associated with their different modes of operations; (2) *Construction of a complete CCD*: construct a CCD model using its basic blocks, i.e., *Decision box*, *Consequence path* and *Consequence box*; (3) *Reduction*: removal of unnecessary decision boxes based on the system functional behavior to obtain a minimal CCD; and lastly (4) *Probabilistic analysis*: evaluating the probabilities of CCD paths describing the occurrence of a sequence of events.

Traditionally, CCD subsystem-level reliability analysis is carried out by using paper-and-pencil-based approaches to analyze safety-critical systems, such as high-integrity protection systems (HIPS) [14] and nuclear power plants [15], or using computer simulation tools based on Monte-Carlo approach, as in [16]. A major limitation in both of the above approaches is the possibility of introducing inaccuracies in the CCD analysis either due to human infallibility or the approximation errors due to numerical methods and pseudo-random numbers in the simulation tools. Moreover, simulation tools do not provide the mathematical expressions that can be used to predict the reliability of a given system based on any probabilistic distributions and failure rates.

A more safe way is to substitute the error-prone informal reasoning of CCD analysis by formal generic mathematical proofs as per recommendations of safety standards, such as IEC 61850 [17], EN 50128 [18] and ISO 26262 [19]. In this work, we propose to use formal techniques based on theorem proving for the formal reliability CCD analysis-based of safety-critical systems, which provides us the ability to obtain a *verified* subsystem-level failure/operating consequence expression. Theorem proving is a formal verification technique [20], which is used for conducting the proof of mathematical theorems based on a computerized proof tool. In particular, we use HOL4 [21], which is an interactive theorem prover with the ability of verifying a wide range of mathematical expressions constructed in higher-order logic (HOL). For this purpose, we endeavor to formalize the above-mentioned *four* steps of CCD analysis using HOL4 proof assistant. To demonstrate the practical effectiveness of the proposed CCD formalization, we conduct the formal CCD analysis of an IEEE 39-bus electrical power network system. Subsequently, we formally determine a commonly used metric, namely Forced Outage Rate ($\mathcal{FOR}$), which determines the capacity outage or unavailability of the power generation units [22]. Also, we evaluate the System Average Interruption Duration Index ($\mathcal{SAIDI}$), which describes the average duration of interruptions for each customer in a power network [22].

The main contributions of the work we describe in this report can be summarized as follows:

- Formalization of the CCD basic constructors, such as *Decision box*, *Consequence path* and *Consequence box*, that can be used to build an arbitrary level of CCDs
- Enabling the formal reduction of CCDs that can remove unnecessary decision boxes from a given CCD model, a feature not available in other existing approaches
- Provide reasoning support for formal probabilistic analysis of scalable CCDs consequence paths with new proposed mathematical formulations
- Application on a real-world IEEE 39-bus electrical power network system and verification of its reliability indexes $\mathcal{FOR}$ and $\mathcal{SAIDI}$
- Development of a Standard Meta Language (SML) function that can numerically compute reliability values from the *verified* expressions of $\mathcal{FOR}$ and $\mathcal{SAIDI}$
- Comparison between our formal CCD reliability assessment with the corresponding results obtained from MATLAB MCS and other notorious approaches

The rest of the report is organized as follows: In Section 2, we present the related literature review. In Section 3, we describe the preliminaries to facilitate the understanding of the rest of the report. Section 4 presents the proposed formalization of CCD and its formal probabilistic properties. In Section 5, we describe the formal CCD analysis of an electrical network system and the evaluation of its reliability indices $\mathcal{FOR}$ and $\mathcal{SAIDI}$. Lastly, Section 6 concludes the report.

2 Related Work

Only a few work have previously considered using formal techniques [20] to model and analyze CCDs. For instance, Ortmeier et al. in [23] developed a framework for Deductive Cause-Consequence Analysis (DCCA) using the SMV model checker [24] to verify the CCD proof obligations. However, according to the authors [23], there is a problem of showing the completeness of DCCA due to the exponential growth of the number of proof obligations with complex systems that need cumbersome proof efforts. To overcome above-mentioned limitations, a more practical way is to verify *generic* mathematical formulations that can perform $\mathcal{N}$-level CCD reliability analysis for real-world systems within a sound environment. Higher-Order-Logic (HOL) [25] is a good candidate formalism for achieving this goal.

Prior to our work, there were two notable projects for building frameworks to formally analyze dependability models using HOL4 theorem proving [21]. For instance, HOL4 has been previously used by Ahmad et al. in [26] to formalize SFTs. The SFT formalization includes a new datatype consisting of `AND`, `OR` and `NOT` FT gates [4] to analyze the factors causing a static system failure. Furthermore, Elderhalli et al. in [27] had formalized DFTs in the HOL4 theorem prover, which can be used to conduct formal dynamic failure analysis. Similarly, we have defined in [28] a new `EVENT_TREE`

datatype to model and analyze all possible system-level success and failure relationships. All these formalizations are basically required to formally analyze either a system static/dynamic failure or cascading dependencies of system-level components only, respectively. On the other hand, CCDs have the superior capability to use SFTs/DFTs for analyzing the static/dynamic failures at the subsystem level and analyze their cascading dependencies at the system-level using ETs. For that purpose, in this work, we provide new formulations that can model mathematically the graphical diagrams of CCDs and perform the subsystem-level reliability analysis of highly-critical systems. Moreover, our proposed new mathematics provides the modeling of *multi-state* system components and is based on any given probabilistic distribution and failure rates, which makes our proposed work the first of its kind. In order to check the correctness of the proposed equations, we verified them within the sound environment of HOL4.

3 Preliminaries

In this section, we briefly summarize the fundamentals of the HOL4 theorem proving approach and existing FT and ET formalizations in HOL4 to facilitate the reader's understanding of the rest of the report.

3.1 HOL4 Theorem Proving

Theorem proving [20] is one of the formal verification techniques that use a computerized proof system for conducting the proof of mathematical theorems. HOL4 [21] is an interactive theorem prover, which is capable of verifying a wide range of safety-critical systems as well as mathematical expressions constructed in HOL. In general, given a safety-critical system to be formally analyzed, we first model its structure mathematically, then using the HOL4 theorem prover, several properties of the system can be verified based on this mathematical model. The main characteristic of the HOL4 theorem prover is that its core consists only of four axioms and eight inference rules. Any further proof or theorem should be formally verified based on these axioms and rules or based on previously proven theorems. This ensured the soundness of the system model analysis, i.e., no wrong proof goal can be proved. Moreover, since the system properties are proven mathematically within HOL4, no approximation is involved in the analysis results. These features make HOL4 suitable for carrying out the CCD-based reliability analysis of safety-critical systems that require *sound verification* results. Table 1 provides the HOL4 symbols and functions that we will use in this report.

3.2 Probability Theory in HOL4

Measure space is defined mathematically as (Ω, Σ, and μ), where Ω represents the sample space, Σ represents a σ-algebra of subsets of Ω, and μ represents a measure with the domain Σ. A probability space is a measure space (Ω, Σ, and Pr), where Ω is the complete sample space, Σ is the corresponding event space containing all the events of interest, and Pr is the probability measure of the sample space as 1. The HOL4 theorem

Table 1: HOL4 Symbols and Functions

HOL4 Symbol	Standard	Meaning
{x | P(x)}	$\{\lambda x.\ P(x)\}$	Set of all x such that $P(x)$
h :: L	*cons*	Add an element h to a list L
MAP (λx. f(x)) X	x $\in$ X $\rightarrow$ (λx. f)	Function that maps each element x in the list X to f(x)
L_1 ++ L_2	*append*	Joins lists L_1 and L_2 together

prover has a rich library of probabilities, including the functions `p_space`, `events`, and `prob`. Given a probability space p, these functions return the corresponding Ω, Σ, and Pr, respectively. The Cumulative Distribution Function (CDF) is defined as the probability of the event where a random variable X has a value less or equal to a value t, i.e., $\mathcal{P}(X \leq t)$. This definition can be been formalized in HOL4 as [29]:

$\vdash$ `CDF p X t = distribution p X {y | y` $\leq$ `t}`

where the function `distribution` takes three inputs: (i) probability space p; (ii) random variable X; and (iii) set of real numbers, then returns the probability of the variable X acquiring all the values of the given set in probability space p.

3.3 FT Formalization

Fault Tree (FT) analysis [4] is one of the commonly used reliability assessment techniques for critical-systems. It mainly provides a schematic diagram for analyzing undesired *top events*, which can cause complete system failure upon their occurrence. An FT model is represented by *logic*-gates, like OR, AND and NOT, where an OR gate models the failure of the output if any of the input failure events occurs alone, while an AND gate models the failure of the output if all of the input failure events occur at the same time, and lastly a NOT gate models the complement of the input failure event. Ahmad et al. [26] presented the FT formalization by defining a new datatype gate, in HOL4 as:

```
Hol_datatype gate = AND of (gate list) |
                    OR of (gate list) |
                    NOT of (gate) |
                    atomic of (event)
```

The FT constructors `AND` and `OR` are recursive functions on `gate`-typed lists, while the FT constructor `NOT` operates on a `gate`-type variable. A semantic function is then defined over the `gate` datatype that can yield an FT diagram as:

Definition 1:

```
⊢ FTree p (atomic X) = X ∧
  FTree p (OR  (h::t)) = FTree p h ∪ FTree p (OR  t) ∧
  FTree p (AND (h::t)) = FTree p h ∩ FTree p (AND t) ∧
  FTree p (NOT X) = p_space p DIFF FTree p X
```

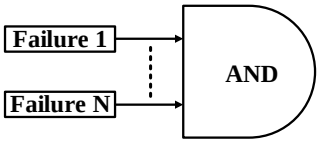
The function `FTree` takes an event `X`, identified by a type constructor `atomic`, and returns the given event `X`. If the function `FTree` takes a list of type `gate`, identified by a type constructor `OR`, then it returns the union of all elements after applying the function `FTree` on each element of the given list. Similarly, if the function `FTree` takes a list of type `gate`, identified by a type constructor `AND`, then it performs the intersection of all elements after applying the function `FTree` on each element of the given list. For the `NOT` type constructor, the function `FTree` returns the complement of the failure event obtained from the function `FTree`.

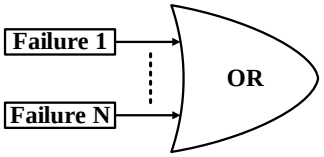
The formal verification in HOL4 for the failure probabilistic expressions of the above-mentioned FT gates is presented in Table 2 [26]. These expressions are verified under the following constrains: (a) $F_{\mathcal{N}} \in$ `events p` ensures that all associated failure events in the given list $F_{\mathcal{N}}$ are drawn from the events space p; (b) `prob_space p` ensures that p is a valid probability space; and lastly (c) `MUTUAL_INDEP p` $F_{\mathcal{N}}$ ensures the independence of the failure events in the given list $F_{\mathcal{N}}$. The function $\prod$ takes a list and returns the product of the list elements while the function `PROB_LIST` returns a list of probabilities associated with the elements of the list. The function `COMPL_LIST` returns the complement of the given list elements.

3.4 ET Formalization

Event Tree (ET) [5] analysis is a widely used technique to enumerate all possible combinations of system-level components failure and success states in the form of a

Table 2: FT HOL4 Probabilistic Theorems

FT Gate	Probabilistic Theorem
	`prob p` `(FTree p (AND` $F_{\mathcal{N}}$`)) =` $\prod$ `(PROB_LIST p` $F_{\mathcal{N}}$`)`

	`prob p`
	`(FTree p (OR` $F_{\mathcal{N}}$`)) =`
	`1 -` $\prod$ `(PROB_LIST p (COMPL_LIST p` $F_{\mathcal{N}}$`))`

tree structure. An ET diagram starts by an initiating event called *Node* and then all possible scenarios of an event that can occur in the system are drawn as *Branches*. ETs were formally modeled by using a new recursive datatype `EVENT_TREE`, in HOL4 as [28]:

```
Hol_datatype EVENT_TREE = ATOMIC of (event) |
                          NODE of (EVENT_TREE list) |
                          BRANCH of (event) (EVENT_TREE list)
```

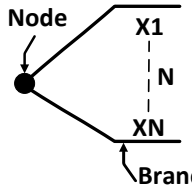
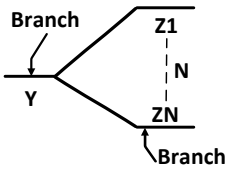
The type constructors `NODE` and `BRANCH` are recursive functions on `EVENT_TREE`-typed lists. A semantic function is then defined over the `EVENT_TREE` datatype that can yield a corresponding ET diagram as:

Definition 2:

```
⊢ ETREE (ATOMIC X) = X ∧
  ETREE (NODE (h::L)) = ETREE h ∪ (ETREE (NODE L)) ∧
  ETREE (BRANCH X (h::L)) = X ∩ (ETREE h ∪ ETREE (BRANCH X L))
```

The function `ETREE` takes an event `X`, identified by a type constructor `ATOMIC` and returns the event `X`. If the function `ETREE` takes a list of type `EVENT_TREE`, identified by a type constructor `NODE`, then it returns the union of all elements after applying the function `ETREE` on each element of the list. Similarly, if the function `ETREE` takes an event `X` and a list of type `EVENT_TREE`, identified by a type constructor `BRANCH`, then it performs the intersection of the event `X` with the union of the head of the list after applying the function `ETREE` and the recursive call of the `BRANCH` constructor. For the formal probabilistic assessment of each path occurrence in the ET diagram, HOL4 probabilistic properties for `NODE` and `BRANCH` ET constructors are presented in Table 3 [28]. These expressions are formally verified under the same FT constrains, i.e., $\mathcal{X}_{\mathcal{N}} \in$ `events p`, `prob_space p` and `MUTUAL_INDEP p` $\mathcal{X}_{\mathcal{N}}$. The function $\sum_{\mathcal{P}}$ is defined to sum the probabilities of events for a list.

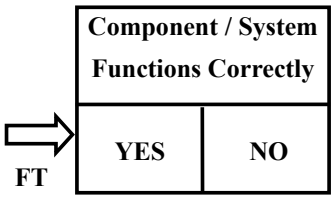
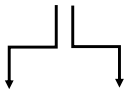
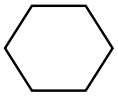
Table 3: ET HOL4 Probabilistic Theorems

ET Constructor	Probabilistic Theorem
	`prob p (ETREE (NODE` $\mathcal{X}_{\mathcal{N}}$`)) =` $\sum_{\mathcal{P}}$ `p` $\mathcal{X}_{\mathcal{N}}$
	`prob p (ETREE (BRANCH Y` $\mathcal{Z}_{\mathcal{N}}$`)) = (prob p Y) ×` $\sum_{\mathcal{P}}$ `p` $\mathcal{Z}_{\mathcal{N}}$

4 Cause-Consequence Diagrams

Cause–Consequence Diagram [15] (CCD) has been developed to analyze the causes of an undesired subsystem failure events, using FT analysis, and from these events obtain all possible consequences on the entire system, using ET analysis [30]. The description of the CCD basic constructors are illustrated in Table 4 [14]. CCD analysis is mainly divided into two categories [31]: (1) *Type I* that combines SFT and ET, as shown in Fig. 1 and Table 5 [12]; and (2) *Type II* that combines DFT and ET without shared events in different subsystems, as shown in Fig. 2 and Table 6 [12]. In this analysis, we focus on the CCD-based reliability analysis at the subsystem level of *Type I*.

Table 4: CCD Symbols and Functions

CCD Symbol	Function
	`Decision Box:` represents the functionality of a component. (1) `NO Box`: describes the component or subsystem failure behavior. A FT of the component is connected to this box that can be used to determine the failure probability ($\mathcal{P}_F$) (2) `YES Box`: represents the correct functioning of the component or reliability, which can be calculated by simply taking the complement of the failure probability determined in the NO Box, i.e., 1 - $\mathcal{P}_F$
	`Consequence Path:` models the next possible consequence scenarios due to a particular event
	`Consequence Box:` models the outcome event due to a particular sequence of events

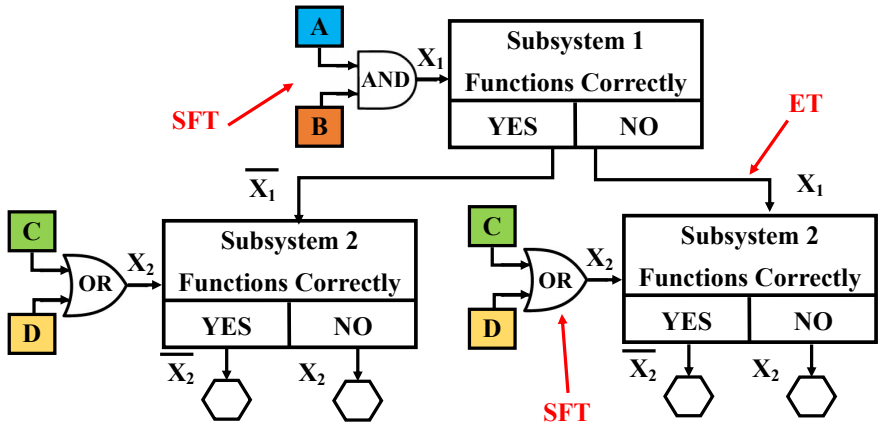
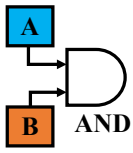
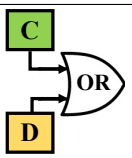


Figure 1: CCD Analysis Type A

Table 5: SFT Symbols and Functions

SFT Symbol	Function
	`AND Gate:` models the failure of the output if all of the input failure events, i.e., A and B, occur at the same time (simultaneously)
	`OR Gate:` models the failure of the output if any of the input failure events, i.e., C or D, occurs alone

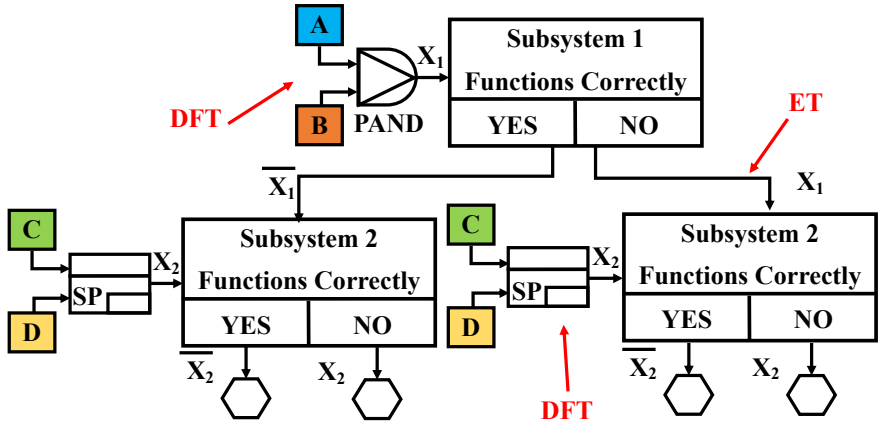


Figure 2: CCD Analysis Type B

Table 6: DFT Symbols and Functions

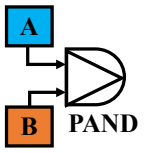
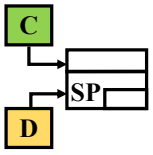
DFT Symbol	Function
	`Priority-AND (PAND) Gate:` models the dynamic behavior of failing the output when all input events occur in a sequence, i.e., A then B
	`SPare (SP) Gate:` models the dynamic behavior of activating the spare input D after the failure of the main input C

Fig. 3 depicts the overview of the *four* steps of CCD analysis [3]: (1) *Components failure events*: identify the causes of the undesired failure events for each subsystem/component in the safety-critical system; (2) *Construction of a complete CCD*: draw a complete system CCD model using its basic constructors considering that the order of components should follow the temporal action of the system; (3) *CCD model reduction*: remove the unnecessary decision boxes in the system to obtain its minimal CCD representing the actual functional behavior of the system; and (4) *CCD probabilistic analysis*: evaluate the probabilities of all CCD consequence paths. The paths in a CCD represent the likelihood of specific sequence scenarios that are possible to occur in a system so that *only one* scenario can occur [30]. This implies that all consequences in a CCD are disjoint (mutually exclusive) [14]. Assuming that all events associated with the decision boxes in a CCD model are mutually independent, then the CCD paths probabilities can be quantified as follows [15]:

1. Evaluate the probabilities of each outgoing branch stemming from a *decision box*, i.e., quantifying the associated FT models

2. Compute the probability of each *consequence path* by multiplying the individual probabilities of all events associated with the decision boxes

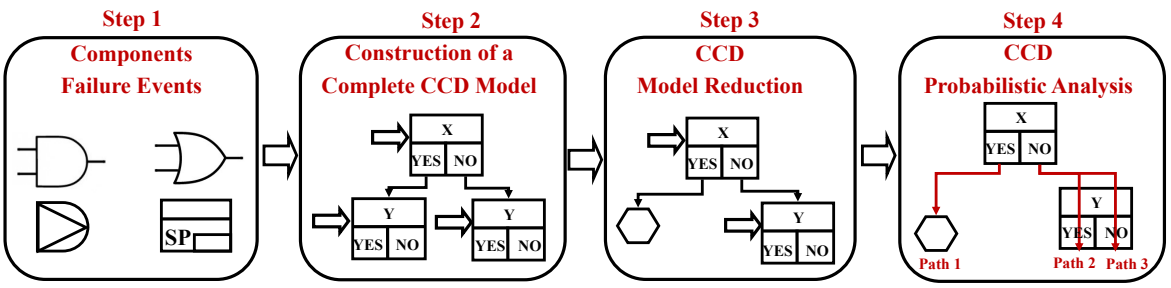


Figure 3: Overview of CCD Analysis

3. Determine the probability of a particular *consequence box* by summing the probabilities of all consequence paths ending with that consequence event

As an example, consider a Motor Control Centre (MCC) [32] consisting of three components *Relay*, *Timer* and *Fuse*, as shown in Fig. 4. The MCC is designed to control an Induction Motor (IM) and let it run for a specific period of time then stops. The IM power circuit is energized by the closure of the Relay Contacts (Rc), as shown in Fig. 4. Rc closes after the user press the Start button that energizes R and at the same time energizes an ON-delay Timer (T). The Timer opens its contacts (Tc) after a specific period of time t and consequently the IM stops. If the IM is overloaded than its design, then the Fuse (F) melts and protects both MCC and IM from damage. Assume that each component in the MCC has two operational states, i.e., operating or failing. The *four* steps of a CCD-based reliability analysis described by Andrews et al. [14] are as follows [30]:

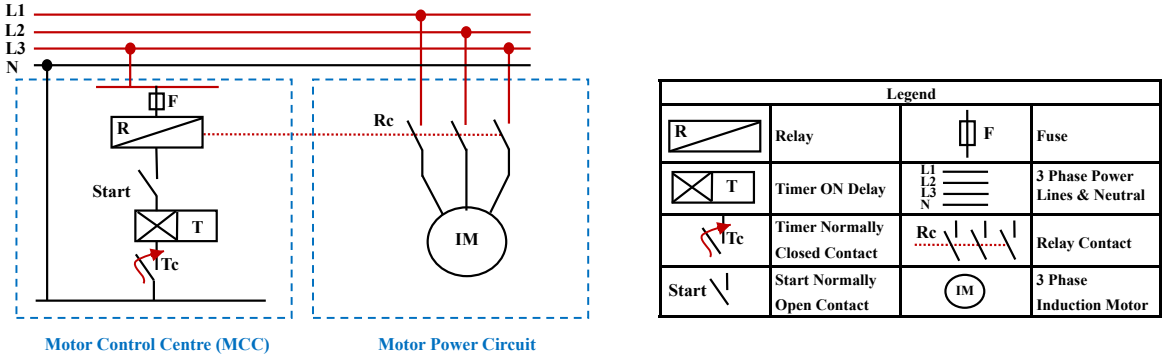


Figure 4: Schematic of an Example MCC

1. *Components failure events*: Assign a FT to each component in the MCC, i.e., FT_{Relay}, FT_{Timer}, FT_{Fuse}.

2. *Construction of a complete CCD*: Construct a complete CCD model of the IM control operation, as shown in Fig. 5. For instance, if the condition of the first decision box is either satisfied or not, i.e., YES or NO, then the next system components are considered in order, i.e., *Timer* and *Fuse*, respectively. Each consequence in the CCD ends with either motor stops (MS) or motor runs (MR).

3. *CCD model reduction*: Apply the reduction process on the obtained complete CCD model. For instance, if the condition of the first decision box (Relay Contacts Open) is satisfied, i.e., YES box, then the IM stops regardless of the status of the rest of the components, as shown in Fig. 6. Similarly, if the condition of the second decision box (Timer Contacts Open) is satisfied, then the IM stops. So, Fig. 6 represents the minimal CCD for the IM control operation.

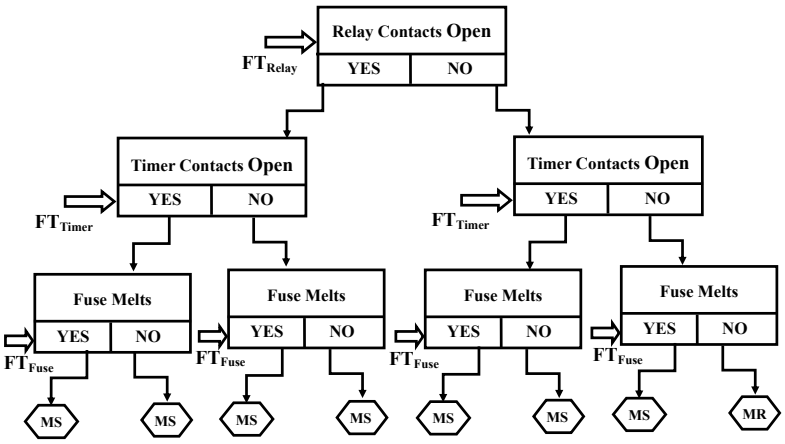


Figure 5: Complete CCD Model of the MCC

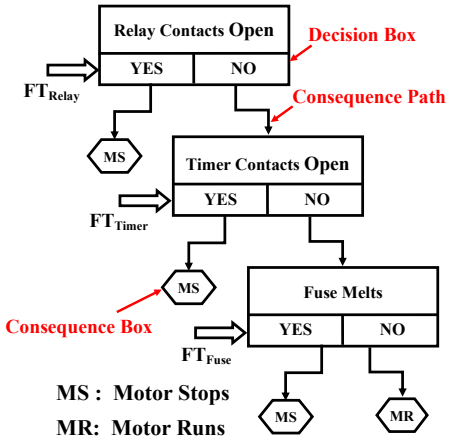


Figure 6: Reduced CCD Model of the MCC

4. *CCD probabilistic analysis*: The probabilities of the two consequence boxes MS and MR in Fig. 6 can be expressed mathematically as:

$$\begin{aligned}\mathcal{P}(Consequence\_Box_{MS}) = \mathcal{P}(Relay_S) + \mathcal{P}(Relay_F) \times \mathcal{P}(Timer_S) + \\ \mathcal{P}(Relay_F) \times \mathcal{P}(Timer_F) \times \mathcal{P}(Fuses_S)\end{aligned} \tag{1}$$

$$\mathcal{P}(Consequence\_Box_{MR}) = \mathcal{P}(Relay_F) \times \mathcal{P}(Timer_F) \times \mathcal{P}(Fuse_F) \tag{2}$$

where $\mathcal{P}(\mathcal{X}_F)$ is the unreliability function or the probability of failure for a component $\mathcal{X}$, i.e., $\text{FT}_{\mathcal{X}}$ model, and $\mathcal{P}(\mathcal{X}_S)$ is the reliability function or the probability of operating, i.e., the complement of the $\text{FT}_{\mathcal{X}}$ model.

In the next section, we present, in detail, the formalization of CCDs in the HOL4 theorem prover to analyze the failures at the subsystem level of a given safety-critical complex system and determine all their possible cascading dependencies of complete/partial reliability and failure events that can occur at the system level.

4.1 Formal CCD Modeling

We start the formalization of CDDs by formally model its basic symbols, as described in Table 4 in HOL4 as follows:

Definition 3:

```
⊢ DEC_BOX p X Y = if X = 1 then FST Y else if X = 0 then SND Y else p_space p
```

where `Y` is an ordered pair (`FST Y`, `SND Y`) representing the reliability and unreliability functions in a decision box, respectively. The condition `X = 1` represents the `YES Box` while `X = 0` represents the `NO Box`. If `X` is neither `1` nor `0`, for instance, `X = 2`, this represents the irrelevance of the decision box, which returns the probability space p to be used in the reduction process of CCDs.

Secondly, we define the CCD *Consequence path* by recursively applying the `BRANCH` ET constructor on a given $\mathcal{N}$ list of decision boxes ($\texttt{DEC\_BOX}_{\mathcal{N}}$) using the HOL4 recursive function `FOLDL` as:

Definition 4:

$$\vdash \texttt{CONSEQ\_PATH p (DEC\_BOX}_1\texttt{::DEC\_BOX}_{\mathcal{N}}\texttt{) =}$$
$$\texttt{FOLDL (}\lambda\texttt{a b. ETREE (BRANCH a b)) DEC\_BOX}_1\ \texttt{DEC\_BOX}_{\mathcal{N}}$$

Finally, we define the CCD *Consequence box* by mapping the function `CONSEQ_PATH` on a list using the HOL4 function `MAP`, then applies the `NODE` ET constructor:

Definition 5:

$$\vdash \texttt{CONSEQ\_BOX p L}_{\mathcal{M}}\texttt{ = ETREE (NODE (MAP (}\lambda\texttt{a. CONSEQ\_PATH p a) L}_{\mathcal{M}}\texttt{))}$$

Using the above definitions, we can construct a complete CCD model (*Step 2* in Fig. 3) for the MCC system shown in Fig. 5, in HOL4 as:

$$\begin{aligned}
&\vdash \texttt{MCC_COMPLETE_CCD FT}_R\ \texttt{FT}_T\ \texttt{FT}_F\ \texttt{=}\\
&\texttt{CONSEQ_BOX p}\\
&\texttt{[[DEC_BOX p 1 (}\overline{\texttt{FT}_R}\texttt{,FT}_R\texttt{);DEC_BOX p 1 (}\overline{\texttt{FT}_T}\texttt{,FT}_T\texttt{);DEC_BOX p 1 (}\overline{\texttt{FT}_F}\texttt{,FT}_F\texttt{)];}\\
&\texttt{ [DEC_BOX p 1 (}\overline{\texttt{FT}_R}\texttt{,FT}_R\texttt{);DEC_BOX p 1 (}\overline{\texttt{FT}_T}\texttt{,FT}_T\texttt{);DEC_BOX p 0 (}\overline{\texttt{FT}_F}\texttt{,FT}_F\texttt{)];}\\
&\texttt{ [DEC_BOX p 1 (}\overline{\texttt{FT}_R}\texttt{,FT}_R\texttt{);DEC_BOX p 0 (}\overline{\texttt{FT}_T}\texttt{,FT}_T\texttt{);DEC_BOX p 1 (}\overline{\texttt{FT}_F}\texttt{,FT}_F\texttt{)];}\\
&\texttt{ [DEC_BOX p 0 (}\overline{\texttt{FT}_R}\texttt{,FT}_R\texttt{);DEC_BOX p 1 (}\overline{\texttt{FT}_T}\texttt{,FT}_T\texttt{);DEC_BOX p 0 (}\overline{\texttt{FT}_F}\texttt{,FT}_F\texttt{)];}\\
&\texttt{ [DEC_BOX p 0 (}\overline{\texttt{FT}_R}\texttt{,FT}_R\texttt{);DEC_BOX p 0 (}\overline{\texttt{FT}_T}\texttt{,FT}_T\texttt{);DEC_BOX p 1 (}\overline{\texttt{FT}_F}\texttt{,FT}_F\texttt{)];}\\
&\texttt{ [DEC_BOX p 0 (}\overline{\texttt{FT}_R}\texttt{,FT}_R\texttt{);DEC_BOX p 0 (}\overline{\texttt{FT}_T}\texttt{,FT}_T\texttt{);DEC_BOX p 0 (}\overline{\texttt{FT}_F}\texttt{,FT}_F\texttt{)]]}
\end{aligned}$$

In CCD analysis [30], *Step 3* in Fig. 3 is used to model the accurate functional behavior of systems in the sense that the irrelevant decision boxes should be removed from a complete CCD model. Upon this, the actual CCD model of the MCC system after reduction, as shown in Fig. 6, can be obtained by assigning `X` with neither `1` nor `0`, for instance, `X = 2`, which represents the irrelevance of the decision box, in HOL4 as:

$$\begin{aligned}
&\vdash \texttt{MCC_REDUCED_CCD FT}_R\ \texttt{FT}_T\ \texttt{FT}_F\ \texttt{=}\\
&\texttt{CONSEQ_BOX p}\\
&\texttt{[[DEC_BOX p 1 (}\overline{\texttt{FT}_R}\texttt{,FT}_R\texttt{);DEC_BOX p 2 (}\overline{\texttt{FT}_T}\texttt{,FT}_T\texttt{);DEC_BOX p 2 (}\overline{\texttt{FT}_F}\texttt{,FT}_F\texttt{)];}\\
&\texttt{ [DEC_BOX p 0 (}\overline{\texttt{FT}_R}\texttt{,FT}_R\texttt{);DEC_BOX p 1 (}\overline{\texttt{FT}_T}\texttt{,FT}_T\texttt{);DEC_BOX p 2 (}\overline{\texttt{FT}_F}\texttt{,FT}_F\texttt{)];}\\
&\texttt{ [DEC_BOX p 0 (}\overline{\texttt{FT}_R}\texttt{,FT}_R\texttt{);DEC_BOX p 0 (}\overline{\texttt{FT}_T}\texttt{,FT}_T\texttt{);DEC_BOX p 1 (}\overline{\texttt{FT}_F}\texttt{,FT}_F\texttt{)];}\\
&\texttt{ [DEC_BOX p 0 (}\overline{\texttt{FT}_R}\texttt{,FT}_R\texttt{);DEC_BOX p 0 (}\overline{\texttt{FT}_T}\texttt{,FT}_T\texttt{);DEC_BOX p 0 (}\overline{\texttt{FT}_F}\texttt{,FT}_F\texttt{)]]}
\end{aligned}$$

Also, we can formally *verify* the above reduced CCD model of the MCC system, in HOL4 as:

```
⊢ MCC_REDUCED_CCD FT_R FT_T FT_F =
  CONSEQ_BOX p
  [[DEC_BOX p 1 (¬FT_R,FT_R)];
   [DEC_BOX p 0 (¬FT_R,FT_R);DEC_BOX p 1 (¬FT_T,FT_T)];
   [DEC_BOX p 0 (¬FT_R,FT_R);DEC_BOX p 0 (¬FT_T,FT_T);DEC_BOX p 1 (¬FT_F,FT_F)];
   [DEC_BOX p 0 (¬FT_R,FT_R);DEC_BOX p 0 (¬FT_T,FT_T);DEC_BOX p 0 (¬FT_F,FT_F)]]
```

where $\overline{\mathrm{FT}_X}$ for a component X is the complement of FT_X.

4.2 Formal CCD Analysis

The important step in the CCD analysis is to determine the probability of each consequence path occurrence in the CCD [14]. For that purpose, we formally verify the following CCD *generic* probabilistic properties, in HOL4 as follows:

Property 1: The probability of a consequence path for *one* decision box assigned with a *generic* FT model, i.e., OR or AND, as shown in Fig. 7, under the assumptions described in Table 2, respectively as follows:

Theorem 1:

```
⊢ prob_space p ∧ F_N ∈ events p ∧ MUTUAL_INDEP p F_N ⇒
   prob p
     (CONSEQ_PATH p [DEC_BOX p X (FTree p (NOT (OR F_N)),FTree p (OR F_N))]) =
      if X = 1 then ∏ (PROB_LIST p (COMPL_LIST p F_N))
else if X = 0 then 1 - ∏ (PROB_LIST p (COMPL_LIST p F_N)) else 1
```

For example, consider a system `X` consists of two components C_1 and C_2. Assuming the failure of either one them causes the system failure, i.e., C_{1F} or C_{2F}, We can formally model the FT of the system (FT_{system}), in HOL4 as:

```
⊢ FT_system p C_1F C_2F = FTree p (OR [C_1F;C_2F])
```

Using Theorem 1, we can obtain the probability of a decision box `YES/NO` outcomes connected to the above FT model, respectively, in HOL4 as:

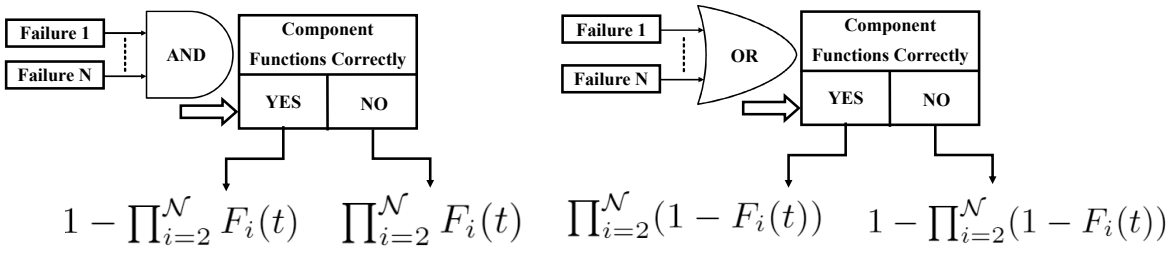


Figure 7: Decision Boxes with FT Gates

$\vdash$ prob p (CONSEQ_PATH p [DEC_BOX p 1 ($\overline{\text{FT}_{system}}$,$\text{FT}_{system}$))]) =
  (1 - prob p C_{1F}) $\times$ (1 - prob p C_{2F})

$\vdash$ prob p (CONSEQ_PATH p [DEC_BOX p 0 ($\overline{\text{FT}_{system}}$,$\text{FT}_{system}$))]) =
  1 - (1 - prob p C_{1F}) $\times$ (1 - prob p C_{2F})

Theorem 2:

$\vdash$ prob_space p $\wedge$ $F_{\mathcal{N}}$ $\in$ events p $\wedge$ MUTUAL_INDEP p $F_{\mathcal{N}}$ $\Rightarrow$
 prob p
  (CONSEQ_PATH p
          [DEC_BOX p X (FTree p (NOT (AND $\text{F}_{\mathcal{N}}$)),FTree p (AND $\text{F}_{\mathcal{N}}$))]) =
    if X = 1 then 1 - $\prod$ (PROB_LIST p $\text{F}_{\mathcal{N}}$)
 else if X = 0 then $\prod$ (PROB_LIST p $\text{F}_{\mathcal{N}}$) else 1

For instance, in the above example, assume the failure of both components simultaneously only causes the system failure, i.e., C_{1F} and C_{2F}. We can formally model the FT of the system, in HOL4 as:

$\vdash$ FT_{system} p C_{1F} C_{2F} = FTree p (AND[C_{1F};C_{2F}])

Using Theorem 2, we can obtain the probability of a decision box YES/NO outcomes connected to the above FT model, respectively, in HOL4 as:

$\vdash$ prob p (CONSEQ_PATH p [DEC_BOX p 1 ($\overline{\text{FT}_{system}}$,$\text{FT}_{system}$))]) =
  1 - prob p C_{1F} $\times$ prob p C_{2F}

$\vdash$ prob p (CONSEQ_PATH p [DEC_BOX p 0 ($\overline{\text{FT}_{system}}$,$\text{FT}_{system}$))]) =
  prob p C_{1F} $\times$ prob p C_{2F}

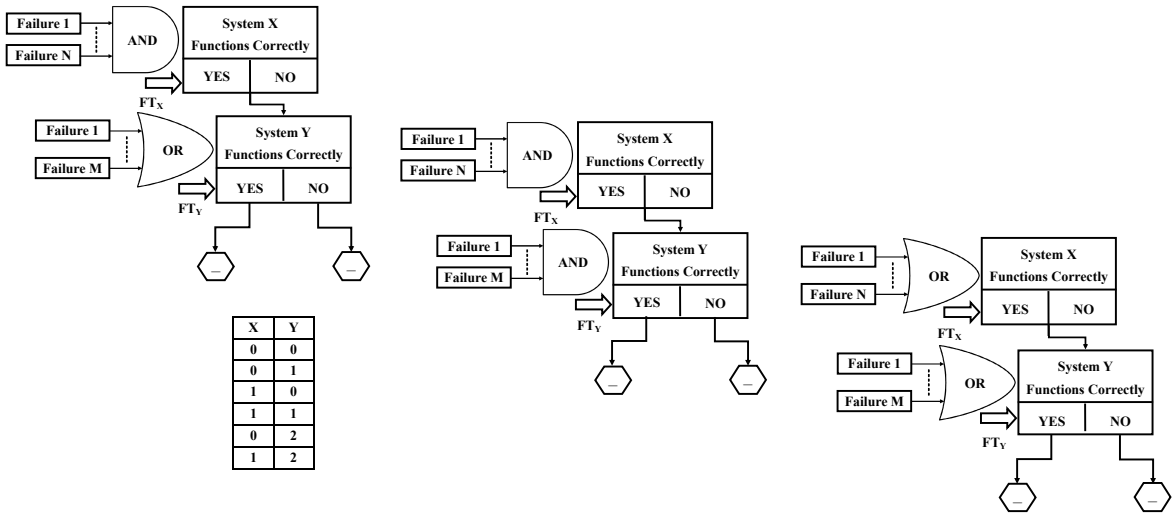


Figure 8: Two-level Decision Boxes for CCD Analysis

Property 2: The probability of *two*-level decision boxes assigned to a CCD path with all combinations of FT gates (AND-OR/OR-AND , AND-AND and OR-OR), as shown in Fig. 8. Each combination has 4 possible operating scenarios that can occur (0-0, 0-1, 1-0 and 1-1) and 2 other possible reduction scenarios that may be required in *Step* 3 (0-2 and 1-2), which represents the removal of the decision box Y from the path. The basic idea is to select different combinations of decision boxes to achieve the desired system behavior and also select the reduction combination (> 1) to remove irreverent decision boxes. This probabilistic expressions can be formally verified, in HOL4 as:

Theorem 3:

```
⊢ prob_space p ∧ (∀y.  y ∈ (F_N++F_M) ⇒ y ∈ events p) ∧
  MUTUAL_INDEP p (F_N++F_M) ⇒
   prob p (CONSEQ_PATH p
         [DEC_BOX p X (FTree p (NOT (AND F_N)),FTree p (AND F_N));
          DEC_BOX p Y (FTree p (NOT (OR F_M)),FTree p (OR F_M))]) =
        if X = 0 ∧ Y = 0 then
     ∏ (PROB_LIST p F_N) × (1 - ∏ (PROB_LIST p (COMPL_LIST p F_M)))
  else if X = 0 ∧ Y = 1 then
     ∏ (PROB_LIST p F_N) × ∏ (PROB_LIST p (COMPL_LIST p F_M))
  else if X = 1 ∧ Y = 0 then
    (1 - ∏ (PROB_LIST p F_N)) × (1 - ∏ (PROB_LIST p (COMPL_LIST p F_M)))
  else if X = 1 ∧ Y = 1 then
    (1 - ∏ (PROB_LIST p F_N)) × ∏ (PROB_LIST p (COMPL_LIST p F_M))
  else if X = 0 ∧ Y = 2 then ∏ (PROB_LIST p F_N)
  else if X = 1 ∧ Y = 2 then (1 - ∏ (PROB_LIST p F_N)) else 1
```

Theorem 4:

```
⊢ prob p (CONSEQ_PATH p
         [DEC_BOX p X (FTree p (NOT (AND F_N)),FTree p (AND F_N));
          DEC_BOX p Y (FTree p (NOT (AND F_M)),FTree p (AND F_M))]) =
        if X = 0 ∧ Y = 0 then
     ∏ (PROB_LIST p F_N) × ∏ (PROB_LIST p F_M)
  else if X = 0 ∧ Y = 1 then
     ∏ (PROB_LIST p F_N) × (1 - ∏ (PROB_LIST p F_M))
                        ⋮
  else if X = 1 ∧ Y = 2 then (1 - ∏ (PROB_LIST p F_N)) else 1
```

Theorem 5:

```
⊢ prob p (CONSEQ_PATH p
         [DEC_BOX p X (FTree p (NOT (OR F_N)),FTree p (OR F_N));
          DEC_BOX p Y (FTree p (NOT (OR F_M)),FTree p (OR F_M))]) =
        if X = 0 ∧ Y = 0 then
  (1 - ∏ (PROB_LIST p (COMPL_LIST p F_N))) ×
  (1 - ∏ (PROB_LIST p (COMPL_LIST p F_M)))
  else if X = 0 ∧ Y = 1 then
  (1 - ∏ (PROB_LIST p (COMPL_LIST p F_N))) ×
  ∏ (PROB_LIST p (COMPL_LIST p F_M))
                        ⋮
  else if X = 1 ∧ Y = 2 then ∏ (PROB_LIST p (COMPL_LIST p F_N)) else 1
```

Property 3: A *generic* probabilistic property for a consequence path consisting of complex *four*-level decision boxes associated with different combination of FTs and each one consisting of $\mathcal{N}$ components (AND-OR-AND-OR/OR-AND-OR-AND/AND-AND-OR-OR/OR-OR-AND-AND), which has 16 possible operating scenarios that can occur and 14 other possible reduction possibilities, as shown in Fig. 9, in HOL4 as:

Theorem 6:

```
⊢ Let
W_F = ∏ (PROB_LIST p F_N);
W̄  = 1 - W_F;
X_F = 1 - ∏ (PROB_LIST p (COMPL_LIST p F_K)); X̄  = 1 - X_F;
Y_F = ∏ (PROB_LIST p F_M);
Ȳ  = 1 - Y_F;
Z_F = 1 - ∏ (PROB_LIST p (COMPL_LIST p F_J)); Z̄  = 1 - Z_F
in
  prob p
    (CONSEQ_PATH p
        [DEC_BOX p W (FTree p (NOT (AND F_N)),FTree p (AND F_N));
         DEC_BOX p X (FTree p (NOT (OR F_K)),FTree p (OR F_K));
         DEC_BOX p Y (FTree p (NOT (AND F_M)),FTree p (AND F_M));
         DEC_BOX p Z (FTree p (NOT (OR F_J)),FTree p (OR F_J))]) =
       if W = 0 ∧ X = 0 ∧ Y = 0 ∧ Z = 0
  then W_F × X_F × Y_F × Z_F
  else if W = 0 ∧ X = 0 ∧ Y = 0 ∧ Z = 1
  then W_F × X_F × Y_F × Z̄
  else if W = 0 ∧ X = 0 ∧ Y = 1 ∧ Z = 0
  then W_F × X_F × Ȳ × Z_F
                          ⋮
  else if W = 1 ∧ X = 1 ∧ Y = 2 ∧ Z = 2
  then W̄ × X̄
  else if W = 1 ∧ X = 2 ∧ Y = 2 ∧ Z = 2
  then W̄ else 1
```

For complex systems consisting of $\mathcal{N}$-level decision boxes, where each decision box is associated with an AND/OR gate consisting of an arbitrary list of failure events, we define *three* types A, B and C of possible CCD outcomes, as shown in Fig. 10, with a new proposed mathematics as:

Property 4 (N Decision Boxes of Type A): The probability of n decision boxes assigned to a consequence path corresponding to n subsystems, where all decision boxes are associated with FT AND models consisting of arbitrary lists of k events, can be expressed mathematically at a specific time t for *three* cases as:

(A1) All outcomes of n decisions boxes are NO

$$\mathcal{F}_{A1}(t) = \prod_{i=1}^{n} \prod_{j=2}^{k} \mathcal{F}_{ij}(t) \tag{3}$$

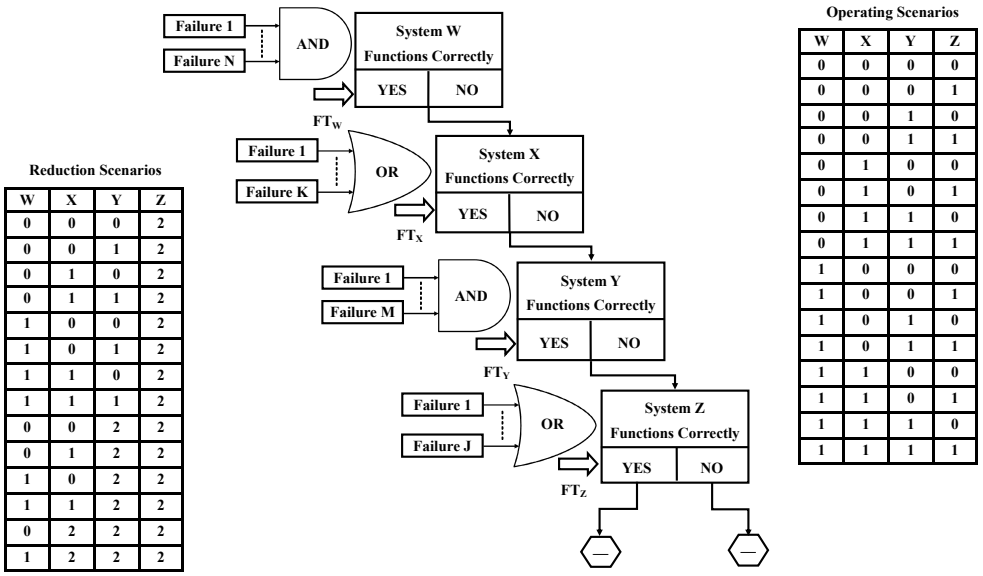


Figure 9: Four-level Decision Boxes for CCD Analysis

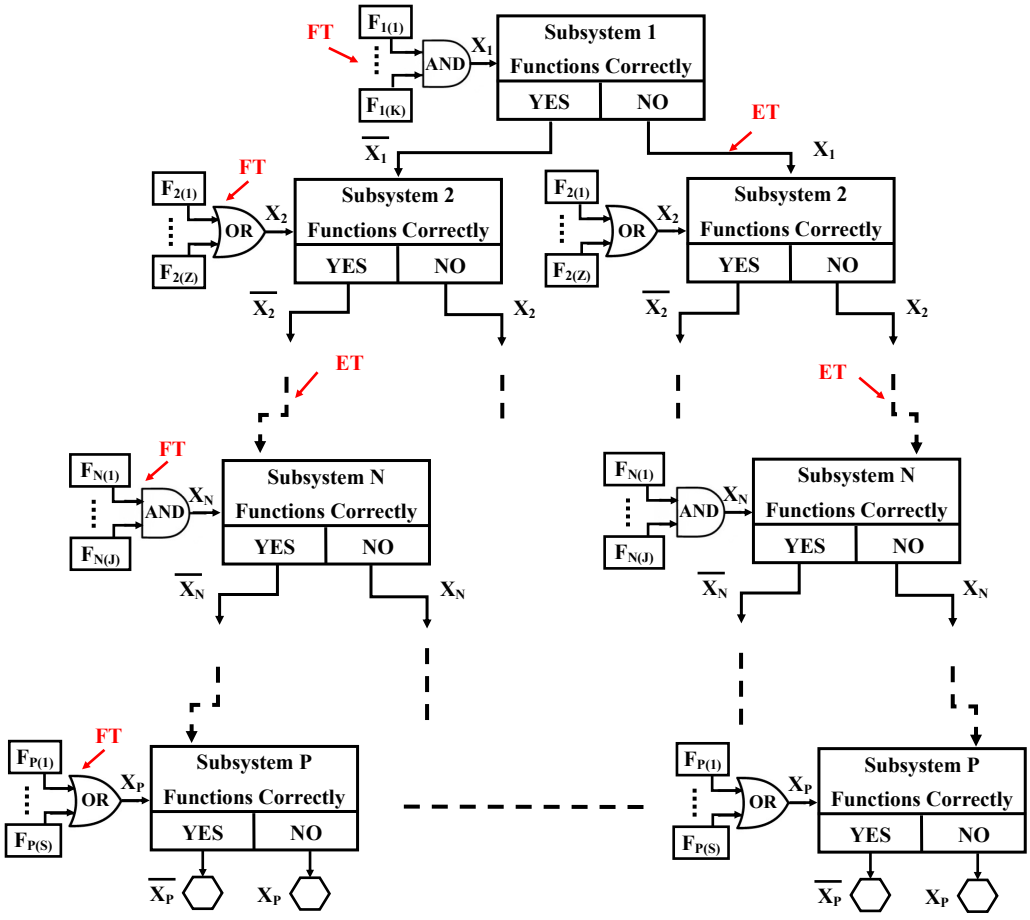


Figure 10: Generic $\mathcal{N}$-level CCD Analysis

(A2) All outcomes of n decisions boxes are YES

$$\mathcal{F}_{A2}(t) = \prod_{i=1}^{n}(1 - \prod_{j=2}^{k}\mathcal{F}_{ij}(t)) \tag{4}$$

(A3) Some outcomes of m decisions boxes are NO and the rest outcomes of p decisions boxes are YES

$$\mathcal{F}_{A3}(t) = \left(\prod_{i=1}^{m}\prod_{j=2}^{k}\mathcal{F}_{ij}(t)\right) \times \left(\prod_{i=1}^{p}(1 - \prod_{j=2}^{k}\mathcal{F}_{ij}(t))\right) \tag{5}$$

To verify the correctness of the above-proposed new safety analysis mathematical formulations in the HOL4 theorem prover, we define two *generic* CCD functions $\mathcal{SS}^{YES}_{AND}$ and $\mathcal{SS}^{NO}_{AND}$ that can recursively generate the outcomes YES and NO of the function `FTree`, identified by `gate` constructors `AND` and `NOT`, for a given arbitrary list of all subsystems failure events (`SSN`), respectively, in HOL4 as:

Definition 6:

$\vdash$ $\mathcal{SS}^{YES}_{AND}$ `p (SS::SSN) = FTree p (NOT (AND SS1))::`$\mathcal{SS}^{YES}_{AND}$ `p SSN`

Definition 7:

$\vdash$ $\mathcal{SS}^{NO}_{AND}$ `p (SS1::SSN) = FTree p (AND SS1)::`$\mathcal{SS}^{NO}_{AND}$ `p SSN`

Using above defined functions, we can verify three *two-dimensional* and *scalable* probabilistic properties corresponding to the above-mentioned safety equations Eq. 3, Eq. 4, and Eq. 5, respectively, in HOL4 as:

Theorem 7:

$\vdash$ `prob p (CONSEQ_PATH p (`$\mathcal{SS}^{NO}_{AND}$ `p SSN)) =`
$\prod$ `(MAP (`λ `a.` $\prod$ `(PROB_LIST p a)) SSN)`

Theorem 8:

$\vdash$ `prob p (CONSEQ_PATH p (`$\mathcal{SS}^{YES}_{AND}$ `p SSN)) =`
$\prod$ `(MAP (`λ `b. (1 -` $\prod$ `(PROB_LIST p b))) SSN)`

Theorem 9:

$\vdash$ `prob p`
`(CONSEQ_PATH p`
`[CONSEQ_PATH p (`$\mathcal{SS}^{NO}_{AND}$ `p SSm);`

`CONSEQ_PATH p (`$\mathcal{SS}^{YES}_{AND}$ `p SSp)]) =`
$\left(\prod$ `(MAP (`λ `a.` $\prod$ `(PROB_LIST p a)) SSm)$\right)$ $\times$
$\left(\prod$ `(MAP (`λ `b. 1 -` $\prod$ `(PROB_LIST p b)) SSp)$\right)$

Property 5 (N Decision Boxes of Type B): The probabilistic assessment of n decision boxes assigned to a CCD consequence path, where all decision boxes are associated with *generic* FT OR models consisting of arbitrary lists of k events, can be expressed mathematically for *three* cases:

(B1) All outcomes of n decisions boxes are NO

$$\mathcal{F}_{B1}(t) = \prod_{i=1}^{n}(1 - \prod_{j=2}^{k}(1 - \mathcal{F}_{ij}(t))) \tag{6}$$

(B2) All outcomes of n decisions boxes are YES

$$\mathcal{F}_{B2}(t) = \prod_{i=1}^{n}\prod_{j=2}^{k}(1 - \mathcal{F}_{ij}(t)) \tag{7}$$

(B3) Some outcomes of m decisions boxes are NO and some outcomes of p decisions boxes are YES

$$\mathcal{F}_{B3}(t) = \left(\prod_{i=1}^{m}(1 - \prod_{j=2}^{k}(1 - \mathcal{F}_{ij}(t)))\right) \times \left(\prod_{i=1}^{p}\prod_{j=2}^{k}(1 - \mathcal{F}_{ij}(t))\right) \tag{8}$$

To verify the correctness of the above-proposed new CCD mathematical formulas in HOL4, we define two *generic* functions $\mathcal{SS}_{OR}^{YES}$ and $\mathcal{SS}_{OR}^{NO}$ to recursively generate the outcomes YES and NO of the function `FTree`, identified by `gate` constructors `OR` and `NOT`, for a given list of subsystems events.

Definition 8:

```
⊢ SS_OR^YES p (SS::SSN) = FTree p (NOT (OR SS1))::SS_OR^YES p SSN
```

Definition 9:

```
⊢ SS_OR^NO p (SS1::SSN) = FTree p (OR SS1)::SS_OR^NO p SSN
```

Using above defined functions, we can formally *verify* three *scalable* probabilistic properties corresponding to Eq. 6, Eq. 7, and Eq. 8, respectively, in HOL4 as:

Theorem 10:

```
⊢ prob p (CONSEQ_PATH p (SS_OR^NO p SSN)) =
  ∏
   (MAP
    (λ a.
     (1 - ∏ (PROB_LIST p (compl_list p a)))) SSN)
```

Theorem 11:

```
⊢ prob p (CONSEQ_PATH p (SS_OR^YES p SSN)) =
  ∏
   (MAP
    (λ b.
     ∏ (PROB_LIST p (compl_list p b))) SSN)
```

Theorem 12:

```
⊢ prob p
     (CONSEQ_PATH p
         [CONSEQ_PATH p (SS_OR^NO p SSm);

          CONSEQ_PATH p (SS_OR^YES p SSp)]) =
  ∏
  (MAP
    (λ a.
     (1 - ∏ (PROB_LIST p (compl_list p a)))) SSm)
× ∏
  (MAP
    (λ b.
     ∏ (PROB_LIST p (compl_list p b))) SSp)
```

Property 6 (N Decision Boxes of Type C): The probabilistic assessment of n decision boxes assigned to a consequence path for a very complex system, where some m decision boxes are associated with *generic* FT AND models consisting of k-events, while other p decision boxes are associated with *generic* FT OR models consisting of z-events, as shown in Fig. 10, is proposed to be expressed mathematically for *nine* cases as:

(C1) All outcomes of m and p decisions boxes are NO.

$$\mathcal{F}_{C1}(t) = \left(\prod_{i=1}^{m} \prod_{j=2}^{k} \mathcal{F}_{ij}(t) \right) \times \left(\prod_{i=1}^{p} (1 - \prod_{j=2}^{z} (1 - \mathcal{F}_{ij}(t))) \right) \tag{9}$$

Theorem 13:

```
⊢ prob p
     (CONSEQ_PATH p
         [CONSEQ_PATH p (SS_AND^NO p SSm);

          CONSEQ_PATH p (SS_OR^NO p SSp)]) =
  ∏ (MAP (λ a.  ∏ (PROB_LIST p a)) SSm)
× ∏
  (MAP
    (λ b.
     (1 - ∏ (PROB_LIST p (compl_list p b)))) SSp)
```

(C2) All outcomes of m and p decisions boxes are YES.

$$\mathcal{F}_{C2}(t) = \left(\prod_{i=1}^{m} (1 - \prod_{j=2}^{k} \mathcal{F}_{ij}(t)) \right) \times \left(\prod_{i=1}^{p} \prod_{j=2}^{z} (1 - \mathcal{F}_{ij}(t)) \right) \tag{10}$$

Theorem 14:

```
⊢ prob p
     (CONSEQ_PATH p
         [CONSEQ_PATH p (SS^YES_AND p SSm);

          CONSEQ_PATH p (SS^YES_OR p SSp)]) =
  ∏ (MAP (λ a.  1 - ∏ (PROB_LIST p a)) SSm)
× ∏
  (MAP
    (λ b.
     ∏ (PROB_LIST p (compl_list p b))) SSp)
```

(C3) All outcomes of m decisions boxes are NO and all outcomes of p decisions boxes are YES.

$$\mathcal{F}_{C3}(t) = \left(\prod_{i=1}^{m}\prod_{j=2}^{k}\mathcal{F}_{ij}(t)\right) \times \left(\prod_{i=1}^{p}\prod_{j=2}^{z}(1-\mathcal{F}_{ij}(t))\right) \tag{11}$$

Theorem 15:

```
⊢ prob p
     (CONSEQ_PATH p
         [CONSEQ_PATH p (SS^NO_AND p SSm);

          CONSEQ_PATH p (SS^YES_OR p SSp)]) =
  ∏ (MAP (λ a.  ∏ (PROB_LIST p a)) SSm)
× ∏
  (MAP
    (λ b.
     ∏ (PROB_LIST p (compl_list p b))) SSp)
```

(C4) All outcomes of m decisions boxes are YES and all outcomes of p decisions boxes are NO.

$$\mathcal{F}_{C4}(t) = \left(\prod_{i=1}^{m}(1-\prod_{j=2}^{k}\mathcal{F}_{ij}(t))\right) \times \left(\prod_{i=1}^{p}(1-\prod_{j=2}^{z}(1-\mathcal{F}_{ij}(t)))\right) \tag{12}$$

Theorem 16:

```
⊢ prob p
     (CONSEQ_PATH p
         [CONSEQ_PATH p (SS^YES_AND p SSm);

          CONSEQ_PATH p (SS^NO_OR p SSp)]) =
  ∏ (MAP (λ a.  1 - ∏ (PROB_LIST p a)) SSm)
× ∏
  (MAP
    (λ b.
     (1 - ∏ (PROB_LIST p (compl_list p b)))) SSp)
```

(C5) Some outcomes of s out of m decisions boxes are NO, some outcomes of u out of m decisions boxes are YES and all outcomes of p decisions boxes are NO.

$$\mathcal{F}_{C5}(t) = \left(\prod_{i=1}^{s}\prod_{j=2}^{k}\mathcal{F}_{ij}(t)\right) \times \left(\prod_{i=1}^{u}(1-\prod_{j=2}^{k}\mathcal{F}_{ij}(t))\right) \times \left(\prod_{i=1}^{p}(1-\prod_{j=2}^{z}(1-\mathcal{F}_{ij}(t)))\right) \quad (13)$$

Theorem 17:

```
⊢ prob p
     (CONSEQ_PATH p
         [CONSEQ_PATH p (SS^NO_AND p SSs);

          CONSEQ_PATH p (SS^YES_AND p SSu);

          CONSEQ_PATH p (SS^NO_OR p SSp)]) =
  ∏ (MAP (λ a.  ∏ (PROB_LIST p a)) SSs)
× ∏ (MAP (λ b.  1 - ∏ (PROB_LIST p b)) SSu)
× ∏
  (MAP
   (λ c.
    (1 - ∏ (PROB_LIST p (compl_list p c)))) SSp)
```

(C6) Some outcomes of s out of m decisions boxes are NO, some outcomes of u out of m decisions boxes are YES and all outcomes of p decisions boxes are YES.

$$\mathcal{F}_{C6}(t) = \left(\prod_{i=1}^{s}\prod_{j=2}^{k}\mathcal{F}_{ij}(t)\right) \times \left(\prod_{i=1}^{u}(1-\prod_{j=2}^{k}\mathcal{F}_{ij}(t))\right) \times \left(\prod_{i=1}^{p}\prod_{j=2}^{z}(1-\mathcal{F}_{ij}(t))\right) \quad (14)$$

Theorem 18:

```
⊢ prob p
     (CONSEQ_PATH p
         [CONSEQ_PATH p (SS^NO_AND p SSs);

          CONSEQ_PATH p (SS^YES_AND p SSu);

          CONSEQ_PATH p (SS^YES_OR p SSp)]) =
  ∏ (MAP (λ a.  ∏ (PROB_LIST p a)) SSs)
× ∏ (MAP (λ b.  1 - ∏ (PROB_LIST p b)) SSu)
× ∏
  (MAP
   (λ c.
    ∏ (PROB_LIST p (compl_list p c))) SSp)
```

(C7) Some outcomes of s out of p decisions boxes are NO, some outcomes of u out of p decisions boxes are YES and all outcomes of m decisions boxes are NO.

$$\mathcal{F}_{C7}(t) = \left(\prod_{i=1}^{m}\prod_{j=2}^{k}\mathcal{F}_{ij}(t)\right) \times \left(\prod_{i=1}^{u}\prod_{j=2}^{z}(1-\mathcal{F}_{ij}(t))\right) \times \left(\prod_{i=1}^{s}(1-\prod_{j=2}^{z}(1-\mathcal{F}_{ij}(t)))\right) \quad (15)$$

Theorem 19:

```
⊢ prob p
     (CONSEQ_PATH p
         [CONSEQ_PATH p (SS^NO_AND p SSm);

          CONSEQ_PATH p (SS^YES_OR p SSu);

          CONSEQ_PATH p (SS^NO_OR p SSs)]) =
  ∏ (MAP (λ a.  ∏ (PROB_LIST p a)) SSm)
× ∏
  (MAP
    (λ b.
     ∏ (PROB_LIST p (compl_list p b))) SSu)
× ∏
  (MAP
    (λ c.
     (1 - ∏ (PROB_LIST p (compl_list p c)))) SSs)
```

(C8) Some outcomes of s out of p decisions boxes are NO, some outcomes of u out of p decisions boxes are YES and all outcomes of m decisions boxes are YES.

$$\mathcal{F}_{C8}(t) = \left(\prod_{i=1}^{m}(1-\prod_{j=2}^{k}\mathcal{F}_{ij}(t))\right)\times\left(\prod_{i=1}^{u}\prod_{j=2}^{z}(1-\mathcal{F}_{ij}(t))\right)\times\left(\prod_{i=1}^{s}(1-\prod_{j=2}^{z}(1-\mathcal{F}_{ij}(t)))\right) \tag{16}$$

Theorem 20:

```
⊢ prob p
     (CONSEQ_PATH p
         [CONSEQ_PATH p (SS^YES_AND p SSm);

          CONSEQ_PATH p (SS^YES_OR p SSu);

          CONSEQ_PATH p (SS^NO_OR p SSs)]) =
  ∏ (MAP (λ a.  1 - ∏ (PROB_LIST p a)) SSm)
× ∏
  (MAP
    (λ b.
     ∏ (PROB_LIST p (compl_list p b))) SSu)
× ∏
  (MAP
    (λ c.
     (1 - ∏ (PROB_LIST p (compl_list p c)))) SSs)
```

(C9) Some outcomes of s out of m decisions boxes are NO, some outcomes of u out of m decisions boxes are YES, some outcomes of v out of p decisions boxes are NO and some outcomes of w out of p decisions boxes are YES.

$$\mathcal{F}_{C9}(t) = \left(\prod_{i=1}^{s}\prod_{j=2}^{k}\mathcal{F}_{ij}(t)\right) \times \left(\prod_{i=1}^{v}(1-\prod_{j=1}^{z}(1-\mathcal{F}_{ij}(t)))\right) \times \left(\prod_{i=1}^{u}(1-\prod_{j=2}^{k}\mathcal{F}_{ij}(t))\right) \times \left(\prod_{i=1}^{w}\prod_{j=2}^{z}(1-\mathcal{F}_{ij}(t))\right) \tag{17}$$

Theorem 21:

```
⊢ prob p
    (CONSEQ_PATH p
        [CONSEQ_PATH p (SS_AND^NO p SSs);

         CONSEQ_PATH p (SS_AND^YES p SSu);

         CONSEQ_PATH p (SS_OR^NO p SSv);

         CONSEQ_PATH p (SS_OR^YES p SSw)]) =
  ∏ (MAP (λ a.  ∏ (PROB_LIST p a)) SSs)
× ∏ (MAP (λ b.  1 - ∏ (PROB_LIST p b)) SSu)
× ∏
  (MAP
   (λ c.
    (1 - ∏ (PROB_LIST p (compl_list p c)))) SSv)
× ∏
  (MAP
   (λ d.
    ∏ (PROB_LIST p (compl_list p d))) SSw)
```

Therefore, by verifying all the above-mentioned theorems in HOL4, we showed the completeness of our proposed formal approach and thereupon solving the scalability problem of CCD analysis for any given large engineering complex system at the subsystem level [33].

Property 7: A *generic* probabilistic expression of `CONSEQ_BOX` for a certain event occurrence in the entire system as the sum of all individual probabilities of all $\mathcal{M}$ `CONSEQ_PATH` ending with that event:

Theorem 22:

```
⊢ Let
  CONSEQ_PATHS L_M = MAP (λa.  CONSEQ_PATH p a) L_M)
  in
  prob_space p ∧ MUTUAL_INDEP p L_M ∧
  disjoint (CONSEQ_PATHS L_M) ∧ ALL_DISTINCT (CONSEQ_PATHS L_M) ⇒
  prob p (CONSEQ_BOX p L_M) = ∑ (PROB_LIST p (CONSEQ_PATHS L_M))
```

where the HOL4 function `disjoint` ensures that each pair of elements in a given list is mutually exclusive while the function `ALL_DISTINCT` ensures that each pair is distinct.

The function $\sum$ is defined to sum the events for a given list. Remark that all above-mentioned CCD new formulations have been *formally verified* in HOL4, where the proof-script amounts to about 16,000 lines of HOL4 code, which can be downloaded for use from [33]. Also, this code can be extended, with some basic knowhow about HOL4, to perform dynamic failure analysis of dynamic subsystems where no dependencies exist in subsystems using DFTs, such as PAND and SP, i.e, CCD reliability analysis of *Type II* (see Fig. 2).

To illustrate the applicability of our proposed approach, in the next section, we present the formal CCD step-analysis of the standard IEEE 39-bus electrical power network and verify its reliability indexes ($\mathcal{FOR}$ and $\mathcal{SAIDI}$), which are commonly used as reliability indicators by electric power utilities.

5 Electrical Power 39-bus Network System

An electrical power network is an interconnected grid for delivering electricity from producers to customers. The power network system consists of three main zones [1]: (i) generating stations that produce electric power; (ii) transmission lines that carry power from sources to loads; and (iii) distribution lines that connect individual consumers. Due to the complex and integrated nature of the power network, failures in any zone of the system can cause widespread catastrophic disruption of supply [1]. Therefore a rigorous formal cause-consequence analysis of the grid is essential in order to reduce the risk situation of a blackout and enable back-up decisions [34]. For power network safety assessment, reliability engineers have been dividing the power network into three main hierarchical levels [12]: (a) generation systems; (b) composite generation and transmission (or bulk power) systems; and (c) distribution systems. We can use our proposed CCD formalization for the formal modeling and analysis of any hierarchical level in the power network. In this case study, we focus on the generation part only, i.e., hierarchical level I. Also, we can evaluate the Force Outage Rate ($\mathcal{FOR}$) for the generation stations, which is defined as the probability of the unit unavailability to produce power due to unexpected equipment failure [34]. Additionally, we can determine the System Average Interruption Duration Index ($\mathcal{SAIDI}$), which is used to indicate the average duration for each customer served to experience a sustained outage. $\mathcal{SAIDI}$ is defined as the sum of all customer interruption durations (probability of load failures $\lightning$ multiplying by the mean-time-to-repair the failures and the number of customers that are affected by these failures) over the total number of customers served [34]:

$$\mathcal{SAIDI} = \frac{\sum_{\mathcal{P}(\mathcal{X}_{\lightning})\times \mathrm{MTTR}_{\mathcal{X}} \times \mathrm{CN}_{\mathcal{X}}}}{\sum_{\mathrm{CN}_{\mathcal{X}}}} \tag{18}$$

where $\mathrm{CN}_{\mathcal{X}}$ is the number of customers for a certain location $\mathcal{X}$ while $\mathrm{MTTR}_{\mathcal{X}}$ is the mean-time-to-repair the failure that occurred at $\mathcal{X}$. We formally define a function $\sum^{T}_{\lightning}$ in HOL4 to sum all customer interruption durations. Also, we formally define a generic function $\mathcal{SAIDI}$ by dividing the output of $\sum^{T}_{\lightning}$ over the total number of customers served, in HOL4 as:

Definition 10:

$\vdash \sum_{\lightning}^{T}$ (L::$\mathtt{L}_{\mathcal{M}}$) (MTTR::$\mathtt{MTTR}_{\mathcal{M}}$) (CN:$\mathtt{CN}_{\mathcal{M}}$) p =
prob p (CONSEQ_BOX p $\mathtt{L}_{\mathcal{M}}$) $\times$ MTTR $\times$ CN + $\sum_{\lightning}^{T}$ $\mathtt{L}_{\mathcal{M}}$ $\mathtt{MTTR}_{\mathcal{M}}$ $\mathtt{CN}_{\mathcal{M}}$ p

Definition 11:

$$\vdash \mathcal{SAIDI}\ \mathtt{L}_{\mathcal{M}}\ \mathtt{MTTR}_{\mathcal{M}}\ \mathtt{CN}_{\mathcal{M}}\ \mathtt{p} = \frac{\sum_{\lightning}^{T} \mathtt{L}_{\mathcal{M}}\ \mathtt{MTTR}_{\mathcal{M}}\ \mathtt{CN}_{\mathcal{M}}\ p}{\sum \mathtt{CN}_{\mathcal{M}}}$$

where $\mathtt{L}_{\mathcal{M}}$ is the list of CCD paths, $\mathtt{MTTR}_{\mathcal{M}}$ is the list of meantime to repairs, and $\mathtt{CN}_{\mathcal{M}}$ is the list of customer numbers. The function $\sum_{\lightning}^{T}$ (Definition 10) models the numerator of Eq. 18, which is the sum of all customer interruption durations at different locations in the electrical power grid. Each probability of failure is obtained by evaluating a CONSEQ_BOX consisting of a list of $\mathcal{M}$ CONSEQ_PATH, which cause that failure. Definition 11 represents the division of output of Definition 10 over the total number of customers at all those locations as described in Eq. 18.

Consider a standard *IEEE 39-bus* electrical power network test system consisting of 10 generators (G), 12 substations (S/S), 39 Buses (Bus), and 34 transmission lines (TL), as shown in Fig. 11 [35]. Assuming the generators G1-G10 are of two types: (i) solar photo-voltaic (PV) power plants G1-G5; and (ii) steam power plants G6-G10. Using the Optimal Power Flow (OPF) optimization [36], we can determine the flow of electricity from generators to consumers in the power network. Typically, we are only interested in evaluating the duration of certain failure events occurrence for specific loads in the grid. For instance, if we consider the failure of load A, which according to the OPF is supplied from G9 and G5 only, as shown in Fig. 11, then the failure of either one or both power plants will lead to a partial or a complete blackout failure at that load, respectively. Assuming the failure of two consecutive power plants causes a complete blackout of the load. Hence, considering the disruption cases of *only one* supply generator, then different partial failures for loads A, B, C and D, as shown in Fig. 11, can be obtained by observing different failures in the power network as:

a. $\mathcal{P}(\text{Load}_{A\lightning}) = (1 - \mathcal{FOR}_{G_9}) \times \mathcal{FOR}_{G_5} + \mathcal{FOR}_{G_9} \times (1 - \mathcal{FOR}_{G_5})$

b. $\mathcal{P}(\text{Load}_{B\lightning}) = (1 - \mathcal{FOR}_{G_7}) \times \mathcal{FOR}_{G_9} + \mathcal{FOR}_{G_7} \times (1 - \mathcal{FOR}_{G_9})$

c. $\mathcal{P}(\text{Load}_{C\lightning}) = (1 - \mathcal{FOR}_{G_1}) \times \mathcal{FOR}_{G_2} + \mathcal{FOR}_{G_1} \times (1 - \mathcal{FOR}_{G_2})$

d. $$\begin{aligned}\mathcal{P}(\text{Load}_{D\lightning}) = {} & (1 - \mathcal{FOR}_{G_6}) \times (1 - \mathcal{FOR}_{G_3}) \times (1 - \mathcal{FOR}_{G_8}) \times \mathcal{FOR}_{G_4} \\ & + (1 - \mathcal{FOR}_{G_6}) \times (1 - \mathcal{FOR}_{G_3}) \times \mathcal{FOR}_{G_8} \times (1 - \mathcal{FOR}_{G_4}) \\ & + (1 - \mathcal{FOR}_{G_6}) \times \mathcal{FOR}_{G_3} \times (1 - \mathcal{FOR}_{G_8}) \times (1 - \mathcal{FOR}_{G_4}) \\ & + \mathcal{FOR}_{G_6} \times (1 - \mathcal{FOR}_{G_3}) \times (1 - \mathcal{FOR}_{G_8}) \times (1 - \mathcal{FOR}_{G_4})\end{aligned}$$

Therefore, the assessment of $\mathcal{SAIDI}$ for the Grid (G) shown in Fig. 11, including an evaluation for the $\mathcal{FOR}$ of all its power plants, can be written mathematically as:

$$\mathcal{SAIDI}_G = \frac{\mathcal{P}(\text{Load}_{A\lightning}) \times \text{MTTR}_{\text{Load}_A} \times \text{CN}_{\text{Load}_A} + \ldots}{\text{CN}_{\text{Load}_A} + \text{CN}_{\text{Load}_B} + \text{CN}_{\text{Load}_C} + \text{CN}_{\text{Load}_D}} \tag{19}$$

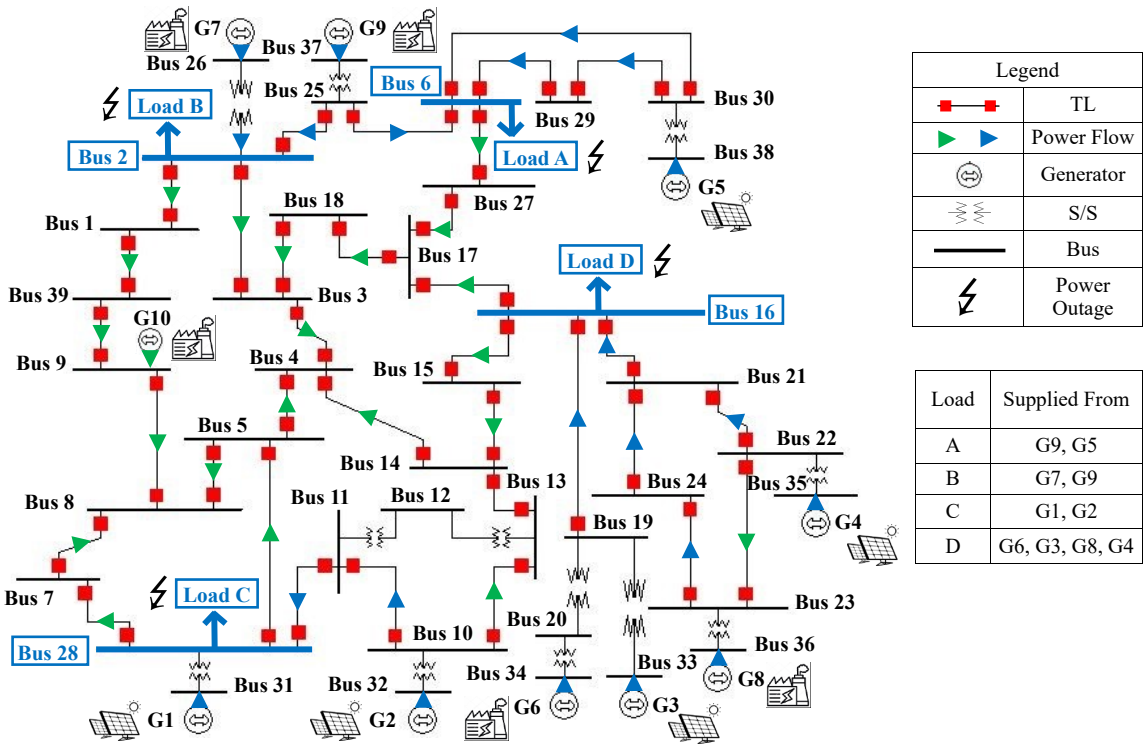


Figure 11: IEEE 39-bus Electrical Power Network [35]

5.1 Formal CCD Analysis in HOL4

We can apply our *four* steps of CCD formalization to verify the expression of $\mathcal{SAIDI}$ in terms of the power plant generator components, in HOL4 as:

Step 1 (Component failure events):
The schematic FT models of a typically PV power plant consisting of 2 solar farms [37] and a steam power plant consisting of 3 generators [34] are shown in Fig. 12 and Fig. 13, respectively. Using the formal FT modeling, we can formally define the FT models of both plants, in HOL4 as:

Definition 12:

```
⊢ FT_PV p [LF1;LF2] [DC_DC1;DC_DC2] [SA1;SA2] [DC_AC1;DC_AC2] =
  FTree p (OR [OR [LF1;DC_DC1;DC_AC1;SA1]; OR [LF2;DC_DC2;DC_AC2;SA2]])
```

Definition 13:

```
⊢ FT_STEAM p [BO1;BO2;BO3] [TA1;TA2;TA3] =
  FTree p (AND [AND [BO1;TA1]; AND [BO2;TA2]; AND [BO3;TA3]])
```

Steps 2 and 3 (Construction of a CCD and Reduction):
Construct a formal complete CCD for all loads in our case study (Fig. 11), i.e., A, B, C, and D, then remove the irrelevant decision boxes according to the electrical power network functional behavior. For instance, we can model the CCD models for loads A and D, as shown in Fig. 14, respectively, in HOL4 as:

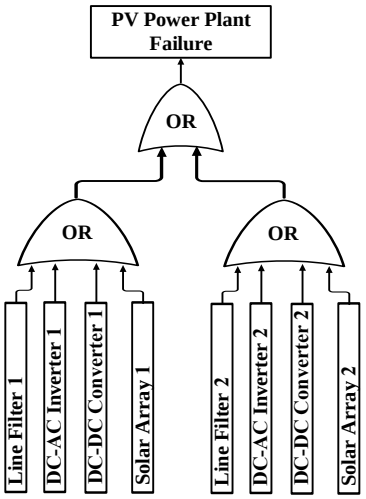


Figure 12: FT Model of a PV Power Plant

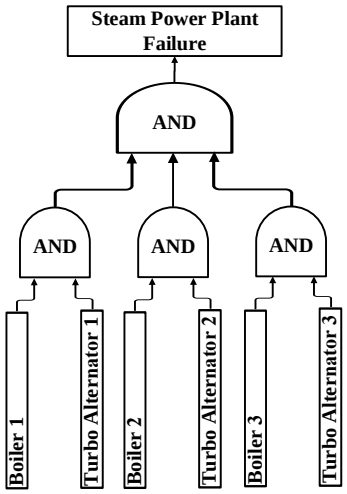


Figure 13: FT Model of a Steam Power Plant

Definition 14:

$$
\begin{aligned}
&\vdash \texttt{CCD\_LOAD\_A} = \\
&\quad \texttt{CONSEQ\_BOX p} \\
&\qquad [[\texttt{DEC\_BOX p 1 (}\overline{\mathrm{FT}_{STEAM}}\texttt{,}\mathrm{FT}_{STEAM}\texttt{);DEC\_BOX p 1 (}\overline{\mathrm{FT}_{PV}}\texttt{,}\mathrm{FT}_{PV}\texttt{)}]; \\
&\qquad\ [\texttt{DEC\_BOX p 1 (}\overline{\mathrm{FT}_{STEAM}}\texttt{,}\mathrm{FT}_{STEAM}\texttt{);DEC\_BOX p 0 (}\overline{\mathrm{FT}_{PV}}\texttt{,}\mathrm{FT}_{PV}\texttt{)}]; \\
&\qquad\ [\texttt{DEC\_BOX p 0 (}\overline{\mathrm{FT}_{STEAM}}\texttt{,}\mathrm{FT}_{STEAM}\texttt{);DEC\_BOX p 1 (}\overline{\mathrm{FT}_{PV}}\texttt{,}\mathrm{FT}_{PV}\texttt{)}]; \\
&\qquad\ [\texttt{DEC\_BOX p 0 (}\overline{\mathrm{FT}_{STEAM}}\texttt{,}\mathrm{FT}_{STEAM}\texttt{);DEC\_BOX p 0 (}\overline{\mathrm{FT}_{PV}}\texttt{,}\mathrm{FT}_{PV}\texttt{)}]]
\end{aligned}
$$

Definition 15:

$$
\begin{aligned}
&\vdash \texttt{CCD\_LOAD\_D} = \\
&\quad \texttt{CONSEQ\_BOX p} \\
&\qquad [[\texttt{DEC\_BOX p 1 (}\overline{\mathrm{FT}_{STEAM}}\texttt{,}\mathrm{FT}_{STEAM}\texttt{);DEC\_BOX p 1 (}\overline{\mathrm{FT}_{PV}}\texttt{,}\mathrm{FT}_{PV}\texttt{);} \\
&\qquad\ \ \texttt{DEC\_BOX p 1 (}\overline{\mathrm{FT}_{STEAM}}\texttt{,}\mathrm{FT}_{STEAM}\texttt{);DEC\_BOX p 1 (}\overline{\mathrm{FT}_{PV}}\texttt{,}\mathrm{FT}_{PV}\texttt{)}]; \\
&\qquad\ [\texttt{DEC\_BOX p 1 (}\overline{\mathrm{FT}_{STEAM}}\texttt{,}\mathrm{FT}_{STEAM}\texttt{);DEC\_BOX p 1 (}\overline{\mathrm{FT}_{PV}}\texttt{,}\mathrm{FT}_{PV}\texttt{);} \\
&\qquad\ \ \texttt{DEC\_BOX p 1 (}\overline{\mathrm{FT}_{STEAM}}\texttt{,}\mathrm{FT}_{STEAM}\texttt{);DEC\_BOX p 0 (}\overline{\mathrm{FT}_{PV}}\texttt{,}\mathrm{FT}_{PV}\texttt{)}]; \\
&\qquad\qquad\ \ \vdots \\
&\qquad\ [\texttt{DEC\_BOX p 0 (}\overline{\mathrm{FT}_{STEAM}}\texttt{,}\mathrm{FT}_{STEAM}\texttt{);DEC\_BOX p 0 (}\overline{\mathrm{FT}_{PV}}\texttt{,}\mathrm{FT}_{PV}\texttt{)}]]
\end{aligned}
$$

Step 4 (Probabilistic analysis):
We can use our proposed formal approach to express subsystem-level failure/reliability probabilistic expressions of electrical power grids, which enable us to analyze the cascading dependencies with many subsystem levels, based on any probabilistic distribution. In this work, we assumed that the failure of each component is exponentially distributed (i.e., `CDF p X t = 1 −` $\mathrm{e}^{(-\lambda_X t)}$, where λ_X is the failure rate of the variable X and t is a time index).

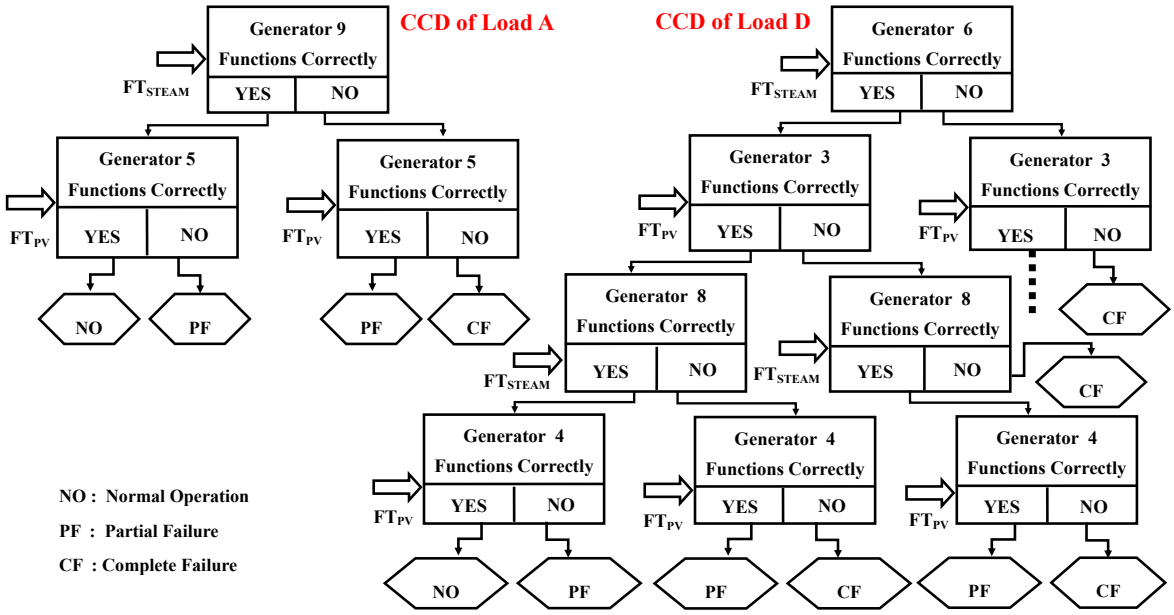


Figure 14: CCD Analysis of Loads A and D

5.1.1 $\mathcal{FOR}$ Analysis

Using Definitions 12 and 13 with the assumption that the failure states of components are exponentially distributed, we can formally specify the probabilistic $\mathcal{FOR}$ expression for both PV and steam power plants, in HOL4 as:

Definition 16:

$\vdash$ $\mathcal{FOR}_{PV}$ p [LF1;LF2] [DC_DC1;DC_DC2] [SA1;SA2] [DC_AC1;DC_AC2] =
prob p (FT_{PV} p ($\downarrow$ [LF1;LF2]) ($\downarrow$ [DC_DC1;DC_DC2])
($\downarrow$ [SA1;SA2]) ($\downarrow$ [DC_AC1;DC_AC2]))

Definition 17:

$\vdash$ $\mathcal{FOR}_{STEAM}$ p [BO1;BO2;BO3] [TA1;TA2;TA3] =
prob p (FT_{STEAM} p ($\downarrow$ [BO1;BO2;BO3]) ($\downarrow$ [TA1;TA2;TA3])

where the function $\downarrow$ takes a list of $\mathcal{N}$ components and assigns an exponential failing event to each component in the list.

We can formally *verify* the above-expressions of $\mathcal{FOR}_{PV}$ and $\mathcal{FOR}_{STEAM}$, in HOL4 as:

Theorem 23:

$\vdash$ $\mathcal{FOR}_{PV}$ p [LF1;LF2] [DC_DC1;DC_DC2] [SA1;SA2] [DC_AC1;DC_AC2] =
$1 - \text{e}^{(-\lambda_{LF1}t)} \times \text{e}^{(-\lambda_{LF2}t)} \times \text{e}^{(-\lambda_{DC\_DC1}t)} \times \text{e}^{(-\lambda_{DC\_DC2}t)} \times \text{e}^{(-\lambda_{SA1}t)} \times \text{e}^{(-\lambda_{SA2}t)} \times$
$\text{e}^{(-\lambda_{DC\_AC1}t)} \times \text{e}^{(-\lambda_{DC\_AC2}t)}$

Theorem 24:

$\vdash$ $\mathcal{FOR}_{STEAM}$ p [BO1;BO2;BO3] [TA1;TA2;TA3] =
$(1 - \text{e}^{(-\lambda_{BO1}t)}) \times (1 - \text{e}^{(-\lambda_{BO2}t)}) \times (1 - \text{e}^{(-\lambda_{BO3}t)}) \times (1 - \text{e}^{(-\lambda_{TA1}t)}) \times$
$(1 - \text{e}^{(-\lambda_{TA2}t)}) \times (1 - \text{e}^{(-\lambda_{TA3}t)})$

5.1.2 $\mathcal{SAIDI}$ Analysis

Using Theorems 1-24 with the assumption that the failure states of components are exponentially distributed, we can formally verify $\mathcal{SAIDI}_G$ (Eq. 19), in HOL4 as:

Theorem 25:

```
⊢ SAIDI
 [[CONSEQ_PATH p
   [DEC_BOX p 1
      (FTree p (NOT (FT_STEAM p (↓ [BO1;BO2;BO3]) (↓ [TA1;TA2;TA3]))),
                    FT_STEAM p (↓ [BO1;BO2;BO3]) (↓ [TA1;TA2;TA3]));
    DEC_BOX p 0
      (FTree p (NOT (FT_PV p (↓ [LF1;LF2]) (↓ [DC_DC1;DC_DC2])
                             (↓ [SA1;SA2]) (↓ [DC_AC1;DC_AC2]))),
                    FT_PV p (↓ [LF1;LF2]) (↓ [DC_DC1;DC_DC2])
                             (↓ [SA1;SA2]) (↓ [DC_AC1;DC_AC2]))];
   [DEC_BOX p 0
      (FTree p (NOT (FT_STEAM p (↓ [BO1;BO2;BO3]) (↓ [TA1;TA2;TA3]))),
                    FT_STEAM p (↓ [BO1;BO2;BO3]) (↓ [TA1;TA2;TA3]));
    DEC_BOX p 1
      (FTree p (NOT (FT_PV p (↓ [LF1;LF2]) (↓ [DC_DC1;DC_DC2])
                             (↓ [SA1;SA2]) (↓ [DC_AC1;DC_AC2]))),
                    FT_PV p (↓ [LF1;LF2]) (↓ [DC_DC1;DC_DC2])
                             (↓ [SA1;SA2]) (↓ [DC_AC1;DC_AC2]))]];
   ...]
[MTTR_LoadA;MTTR_LoadB;MTTR_LoadC;MTTR_LoadD]
[CN_LoadA; CN_LoadB; CN_LoadC; CN_LoadD] p =
```

$$
\begin{aligned}
&((1-(1-\mathrm{e}^{(-\lambda_{BO1}t)})\times(1-\mathrm{e}^{(-\lambda_{BO2}t)})\times(1-\mathrm{e}^{(-\lambda_{BO3}t)})\times\\
&\quad(1-\mathrm{e}^{(-\lambda_{TA1}t)})\times(1-\mathrm{e}^{(-\lambda_{TA2}t)})\times(1-\mathrm{e}^{(-\lambda_{TA3}t)}))\times\\
&(1-\mathrm{e}^{(-\lambda_{LF1}t)}\times\mathrm{e}^{(-\lambda_{LF2}t)}\times\mathrm{e}^{(-\lambda_{DC\_DC1}t)}\times\mathrm{e}^{(-\lambda_{DC\_DC2}t)}\times\\
&\quad\mathrm{e}^{(-\lambda_{DC\_AC1}t)}\times\mathrm{e}^{(-\lambda_{DC\_AC2}t)}\times\mathrm{e}^{(-\lambda_{SA1}t)}\times\mathrm{e}^{(-\lambda_{SA2}t)})+\\
&(1-\mathrm{e}^{(-\lambda_{BO1}t)})\times(1-\mathrm{e}^{(-\lambda_{BO2}t)})\times(1-\mathrm{e}^{(-\lambda_{BO3}t)})\times\\
&(1-\mathrm{e}^{(-\lambda_{TA1}t)})\times(1-\mathrm{e}^{(-\lambda_{TA2}t)})\times(1-\mathrm{e}^{(-\lambda_{TA3}t)})\times\\
&\mathrm{e}^{(-\lambda_{LF1}t)}\times\mathrm{e}^{(-\lambda_{LF2}t)}\times\mathrm{e}^{(-\lambda_{DC\_DC1}t)}\times\mathrm{e}^{(-\lambda_{DC\_DC2}t)}\times\\
&\mathrm{e}^{(-\lambda_{DC\_AC1}t)}\times\mathrm{e}^{(-\lambda_{DC\_AC2}t)}\times\mathrm{e}^{(-\lambda_{SA1}t)}\times\mathrm{e}^{(-\lambda_{SA2}t)})\times\\
&\mathrm{MTTR\_LoadA}\times\mathrm{CN\_LoadA}+\ldots)\\
&\hline
&\mathrm{CN\_LoadA}+\mathrm{CN\_LoadB}+\mathrm{CN\_LoadC}+\mathrm{CN\_LoadD}
\end{aligned}
$$

To further facilitate the exploitation of our proposed approach for power grid reliability engineers, we defined a Standard Meta Language (SML) functions [33] that can numerically evaluate the above-*verified* expressions of $\mathcal{FOR}_{PV}$, $\mathcal{FOR}_{STEAM}$, and $\mathcal{SAIDI}$. Subsequently, we compared our results with MATLAB CCD algorithm based on Monte-Carlo Simulation (MCS) and also with other existing subsystem-level reliability analysis techniques, such as HiP-HOPS and FMR, to ensure the accuracy of our computations, which is presented in the next section.

5.2 Experimental Results and Discussion

Considering the failure rates of the power plant components λ_{BO}, λ_{TA}, λ_{LF}, $\lambda_{\text{DC\_DC}}$, $\lambda_{\text{DC\_AC}}$ and λ_{SA} are 0.91, 0.84, 0.96, 0.67, 0.22, and 0.56 per year [38], respectively. Also, assuming that $\text{MTTR}_{\text{Load}_A}$, $\text{MTTR}_{\text{Load}_B}$, $\text{MTTR}_{\text{Load}_C}$, and $\text{MTTR}_{\text{Load}_D}$ are 12, 20, 15, and 10 hours/interruption [39] and $\text{CN}_{\text{Load}_A}$, $\text{CN}_{\text{Load}_B}$, $\text{CN}_{\text{Load}_C}$, and $\text{CN}_{\text{Load}_D}$ are 500, 1800, 900, and 2500 customers, respectively. The reliability study is undertaken for 1 year, i.e., $t = 8760$ hours. Based on the given data, we can evaluate $\mathcal{FOR}$ and $\mathcal{SAIDI}$ for the electrical power network (Fig. 11) using following techniques:

1. Our proposed SML functions to evaluate the *verified* expressions of $\mathcal{FOR}_{PV}$, $\mathcal{FOR}_{STEAM}$, and $\mathcal{SAIDI}$ in HOL4 (Theorems 23-25), as shown in Fig. 15.

```
> FOR_PV  = 0.991933212861 /year
> FOR_STEAM  = 0.0388700719343 /year
> SAIDI  = 6.37276953475 (Hours / System Customer)
> val it = (): unit
*** Emacs/HOL command completed ***
```

Figure 15: SML Functions: $\mathcal{FOR}$ and $\mathcal{SAIDI}$ Results

2. MATLAB MCS-based toolbox that uses a random-based algorithm to obtain $\mathcal{FOR}$ and $\mathcal{SAIDI}$ for the electrical grid. The steps followed in this technique are as follows [40]:

   - Read the values of failure rate λ in *f/hours* and repair time r in hours for each component
   - Generate a random number U
   - Calculate the predicted next Time to Fail (TTF) and Time to Repair (TTR) from the equations

   $$TTF = \frac{-\ln U}{\lambda} \quad TTR = \frac{-\ln U}{r} \tag{20}$$

   - Repeat the above iterative process till the number of iterations exceeds 1e5

   Based on the above-mentioned MCS steps, we obtain different results of $\mathcal{FOR}$ and $\mathcal{SAIDI}$ every run of the algorithm depending on the generated random number with a tolerance error between 4-9%. So, we present in Table 7 the best-estimated results of $\mathcal{FOR}$ and $\mathcal{SAIDI}$ in MATLAB based on the MCS approach with the least errors. Subsequently, we take the mean average of all the obtained $\mathcal{FOR}$ and $\mathcal{SAIDI}$ results for the power grid.

Table 7: MATLAB MCS: $\mathcal{FOR}$ and $\mathcal{SAIDI}$ Results

Run	$\mathcal{FOR}_{PV}$	$\mathcal{FOR}_{STEAM}$	$\mathcal{SAIDI}$
1	88.55e-2	36.18e-3	5.8023
2	107.19e-2	40.03e-3	6.5045
3	93.52e-2	36.35e-3	6.0222
5	110.17e-2	43.03e-3	7.0495
4	95.24e-2	38.66e-3	6.3960
Average	98.93e-2	38.85e-3	6.3549

3. The Failure Mode Reasoning (FMR) approach, which identifies all the failure modes of safety-critical system inputs that can result in an undesired state at its output. The FMR process consists of four main stages [10]:

   (a) *Composition*: Failure mode variables are defined and a set of logical implication statements is generated that express local failure modes.

   (b) *Substitution*: Local statements will be combined to create a single global implication statement between the critical-system inputs and outputs.

   (c) *Simplification*: The complex formula is simplified, where we trim off any redundant statements.

   (d) *Calculation*: The probability of failure is evaluated using the component failure rates.

   Based on the above-mentioned FMR procedures, we can express the component-level failure analysis of the PV power plant (Fig. 12) as:

$$(\hat{o} = \dot{f}) \Rightarrow (\hat{x_1} = \dot{f} \vee \hat{x_2} = \dot{f}) \tag{21}$$

   The above equation means that if the output o is *False* by fault then either one of its inputs to the OR gate, i.e., x_1 or x_2, must be *False* by fault. We now need to determine what can cause $\hat{x_1} = \dot{f}$ and $\hat{x_2} = \dot{f}$. Similar to Eq. 6, we can write:

$$(\hat{x_1} = \dot{f}) \Rightarrow (\hat{x_3} = \dot{f} \ \vee \hat{x_4} = \dot{f} \ \vee \hat{x_5} = \dot{f} \ \vee \hat{x_6} = \dot{f}) \tag{22}$$

$$(\hat{x_2} = \dot{f}) \Rightarrow (\hat{x_7} = \dot{f} \ \vee \hat{x_8} = \dot{f} \ \vee \hat{x_9} = \dot{f} \ \vee \hat{x_{10}} = \dot{f}) \tag{23}$$

   where x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10} are LF_1, DC_DC_1, DC_AC_1, SA_1, LF_2, DC_DC_2, DC_AC_2, SA_2, respectively. Similarly, we can express the component-level failure analysis of the steam power plant (Fig. 13) as:

$$(\hat{o} = \dot{f}) \Rightarrow (\hat{x_{11}} = \dot{f} \wedge \hat{x_{12}} = \dot{f} \wedge \hat{x_{13}} = \dot{f}) \tag{24}$$

$$(\hat{x_{11}} = \dot{f}) \Rightarrow (\hat{x_{14}} = \dot{f} \wedge \hat{x_{15}} = \dot{f}) \tag{25}$$

$$(\hat{x_{12}} = \dot{f}) \Rightarrow (\hat{x_{16}} = \dot{f} \wedge \hat{x_{17}} = \dot{f}) \tag{26}$$

$$(\hat{x_{13}} = \dot{f}) \Rightarrow (\hat{x_{18}} = \dot{f} \wedge \hat{x_{19}} = \dot{f}) \tag{27}$$

where x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}, are BO_1, TA_1, BO_2, TA_2, BO_3, TA_3, respectively. Table 8 shows the results of $\mathcal{FOR}_{PV}$, $\mathcal{FOR}_{STEAM}$, and $\mathcal{SAIDI}$ based on FMR analysis using the assumed failure rates of the power plant components.

Table 8: FMR: $\mathcal{FOR}$ and $\mathcal{SAIDI}$ Results

$\mathcal{FOR}_{PV}$	$\mathcal{FOR}_{STEAM}$	$\mathcal{SAIDI}$
99.19e-2	38.87e-3	6.3728

According to Jahanian et al. [11], the soundness of the obtained FMR equations (Eq. 21 to Eq. 27) needs to be proven mathematically.

4. The HiP-HOPS software for failure analysis, which can perform FMECA analysis by given architectural blocks that hierarchically describe a safety-critical system

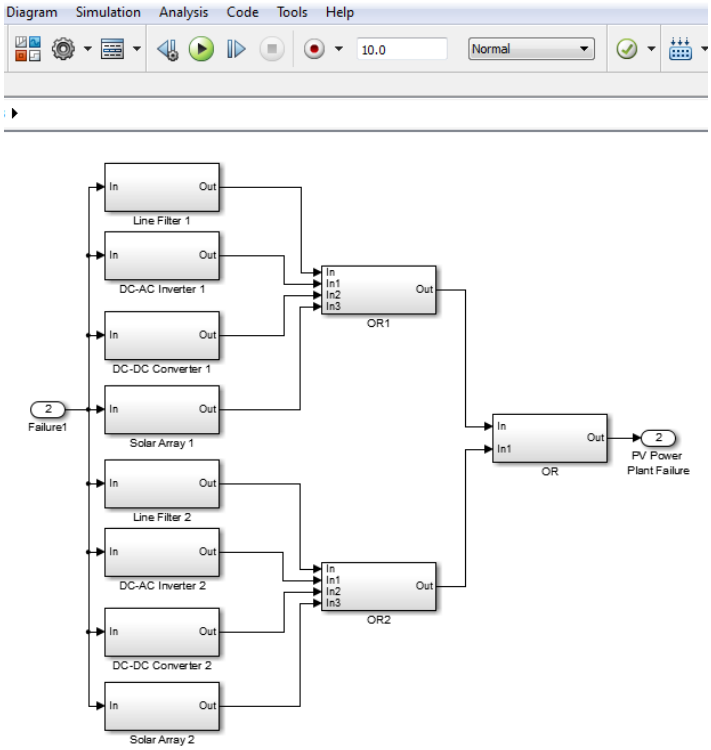


Figure 16: HiP-HOPS: PV Plant FMECA Analysis

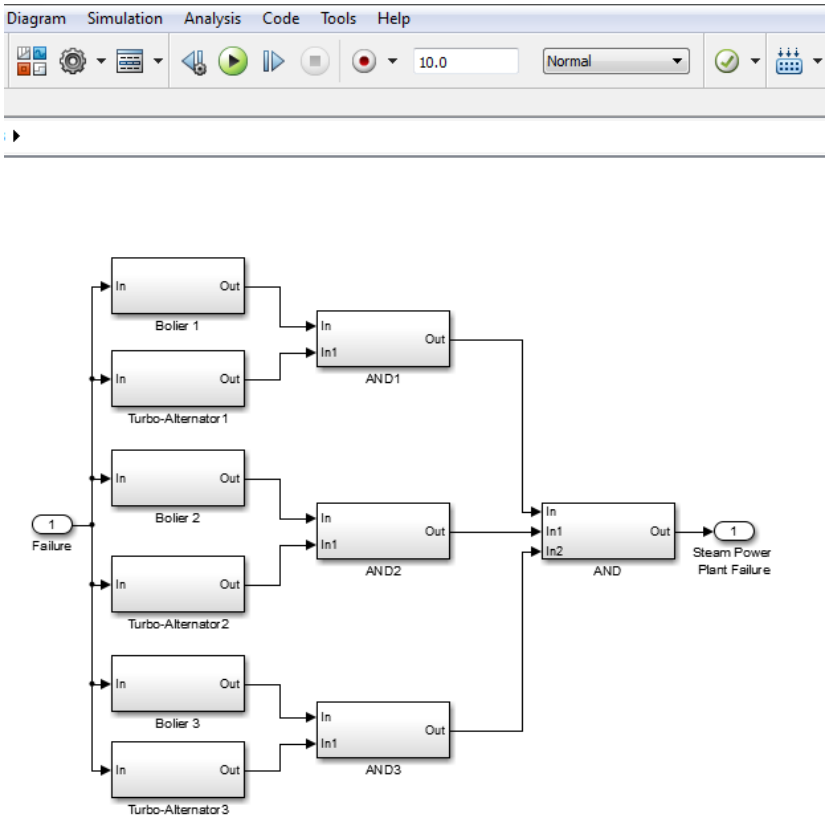


Figure 17: HiP-HOPS: Steam Plant FMECA Analysis

at the subsystem level. Fig. 16 and Fig. 17 depict the FMECA analysis of the PV and steam power plants using the HiP-HOPS software, respectively. The probabilistic results of $\mathcal{FOR}_{PV}$, $\mathcal{FOR}_{STEAM}$, and $\mathcal{SAIDI}$ based on HiP-HOPS analysis are equivalent to the FMR analysis results presented in Table 8.

It can be observed that $\mathcal{SAIDI}$ result obtained from our formal HOL4 analysis are approximately equivalent to the corresponding ones calculated using FMR and HiP-HOPS approaches. On the other hand, MATLAB MCS-based uses a random-based algorithm, which estimates different results of $\mathcal{FOR}$ and $\mathcal{SAIDI}$ every generation of a random number with errors between 4-9%. This clearly demonstrates that our analysis is not only providing the correct result but also with a *formally proven* reliability expressions (Theorems 23-25) compared to simulation tools, i.e., the soundness of subsystem-level reliability analysis. By performing the formal CCD step-analysis of a real-world 39-bus electrical power network, we demonstrated the practical effectiveness of the proposed CCD formalization in HOL4, which will help design engineers to meet the desired quality requirements. Also, our proposed formal approach can be used to analyze larger scale CCD models of other complex electrical power system applications, such as Smartgrids [1].

6 Conclusions

In this work, we developed a formal approach for Cause-Consequence Diagrams (CCD), which enables safety engineers to perform $\mathcal{N}$-level CCD analysis of safety-critical systems within the sound environment of the HOL4 theorem prover. Our proposed approach provides new CCD mathematical formulations, which their correctness was verified in the HOL4 theorem prover. These formulations are capable of performing CCD analysis of *multi-state* system components and based on any given probabilistic distribution and failure rates. These features are not available in any other existing approaches for subsystem-level reliability analysis. The proposed formalization is limited to perform CCD-based reliability analysis at the subsystem level that integrates static dependability analysis. However, this formalization is *generic* and can be extended to perform dynamic failure analysis of dynamic subsystems where no dependencies exist in different subsystems. We demonstrated the practical effectiveness of the proposed CCD formalization by performing the formal CCD step-analysis of a standard *IEEE 39-bus* electrical power network system and also formally verified the power plants Force Outage Rate ($\mathcal{FOR}$) and the System Average Interruption Duration Index ($\mathcal{SAIDI}$). Eventually, we compared the $\mathcal{FOR}$ and $\mathcal{SAIDI}$ results obtained from our formal CCD-based reliability analysis with the corresponding ones using MATLAB based on Monte-Carlo Simulation (MCS), the HiP-HOPS software tool, and the Failure Mode Reasoning (FMR) approach. As future work, we plan to integrate Reliability Block Diagrams (RBDs) [41] as reliability functions in the CCD analysis, which will enable us to analyze hierarchical systems with different component success configurations, based on our CCD formalization in the HOL4 theorem prover.

References

[1] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart Grid—The New and Improved Power Grid: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 944–980, 2011.

[2] M. Rahman, "Power Electronics and Drive Applications for the Automotive Industry," in *Conference on Power Electronics Systems and Applications*. IEEE, 2004, pp. 156–164.

[3] J. D. Andrews and L. M. Ridley, "Reliability of Sequential Systems Using the Cause—Consequence Diagram Method," *Part E: Journal of Process Mechanical Engineering*, vol. 215, no. 3, pp. 207–220, 2001.

[4] M. Towhidnejad, D. R. Wallace, and A. M. Gallo, "Fault Tree Analysis for Software Design," in *NASA Goddard Software Engineering Workshop*, 2002, pp. 24–29.

[5] I. A. Papazoglou, "Mathematical Foundations of Event Trees," *Reliability Engineering & System Safety*, vol. 61, no. 3, pp. 169–183, 1998.

[6] O. Bäckström, Y. Butkova, H. Hermanns, J. Krčál, and P. Krčál, "Effective Static and Dynamic Fault Tree Analysis," in *Computer Safety, Reliability, and Security*, ser. LNCS, vol. 9922. Springer, 2016, pp. 266–280.

[7] Y. Papadopoulos, M. Walker, D. Parker, E. Rüde, R. Hamann, A. Uhlig, U. Grätz, and R. Lien, "Engineering Failure Analysis and Design Optimisation with HiP-HOPS," *Engineering Failure Analysis*, vol. 18, no. 2, pp. 590–608, 2011.

[8] HiP-HOPS, 2020. [Online]. Available: https://hip-hops.co.uk/

[9] S. Kabir, K. Aslansefat, I. Sorokos, Y. Papadopoulos, and Y. Gheraibia, "A Conceptual Framework to Incorporate Complex Basic Events in HiP-HOPS," in *Model-Based Safety and Assessment*, ser. LNCS, vol. 11842. Springer, 2019, pp. 109–124.

[10] H. Jahanian, "Failure Mode Reasoning," in *International Conference on System Reliability and Safety*. IEEE, 2019, pp. 295–303.

[11] H. Jahanian, D. Parker, M. Zeller, A. McIver, and Y. Papadopoulos, "Failure Mode Reasoning in Model Based Safety Analysis," 2020. [Online]. Available: https://arxiv.org/abs/2005.06279

[12] M. Čepin, *Assessment of Power System Reliability: Methods and Applications*. Springer Science & Business Media Springer, 2011.

[13] T. Liu and J. Tong, J.and Zhao, "Probabilistic Risk Assessment Framework Development for Nuclear Power Plant," in *International Conference on Industrial Engineering and Engineering Management*. IEEE, 2008, pp. 1330–1334.

[14] J. D. Andrews and L. M. Ridley, "Application of the Cause-Consequence Diagram Method to Static Systems," *Reliability Engineering & System Safety*, vol. 75, no. 1, pp. 47–58, 2002.

[15] L. M. Ridley, "Dependency Modelling Using Fault-Tree and Cause-Consequence Analysis," Ph.D. dissertation, Loughborough University, UK, 2000.

[16] M. Bevilacqua, M. Braglia, and R. Gabbrielli, "Monte Carlo Simulation Approach for a Modified FMECA in a Power Plant," *Quality and Reliability Engineering International*, vol. 16, no. 4, pp. 313–324, 2000.

[17] R. E. Mackiewicz, "Overview of IEC 61850 and Benefits," in *Power Engineering Society General Meeting*. IEEE, 2006, pp. 623–630.

[18] B. Gallina, E. Gómez-Martínez, and C. B. Earle, "Deriving Safety Case Fragments for Assessing MBASafe's Compliance with EN 50128," in *Conference on Software Process Improvement and Capability Determination*. Springer, 2016, pp. 3–16.

[19] R. Palin, D. Ward, I. Habli, and R. Rivett, "ISO 26262 Safety Cases: Compliance and Assurance," in *Conference on System Safety*, 2011, pp. 1–6.

[20] O. Hasan and S. Tahar, "Formal verification methods," in *Encyclopedia of Information Science and Technology, Third Edition*. IGI Global, 2015, pp. 7162–7170.

[21] HOL Theorem Prover, 2020. [Online]. Available: https://hol-theorem-prover.org

[22] J. J. Grainger and W. D. Stevenson, *Power System Analysis*. McGraw-Hill, 2003.

[23] F. Ortmeier, W. Reif, and G. Schellhorn, "Deductive Cause-Consequence Analysis," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 62–67, 2005.

[24] SMV, 2020. [Online]. Available: http://www.cs.cmu.edu/~modelcheck/smv.html

[25] D. Miller and G. Nadathur, *Programming with higher-Order Logic*. Cambridge University Press, 2012.

[26] W. Ahmad and O. Hasan, "Towards Formal Fault Tree Analysis Using Theorem Proving," in *Intelligent Computer Mathematics*, ser. LNCS, vol. 9150. Springer, 2015, pp. 39–54.

[27] Y. Elderhalli, O. Hasan, and S. Tahar, "A Methodology for the Formal Verification of Dynamic Fault Trees using HOL Theorem Proving," *IEEE Access*, vol. 7, pp. 136 176–136 192, 2019.

[28] M. Abdelghany, W. Ahmad, and S. Tahar, "A Formally Verified HOL4 Algebra for Event Trees," 2020. [Online]. Available: http://arxiv.org/abs/2004.14384

[29] O. Hasan, N. Abbasi, B. Akbarpour, S. Tahar, and R. Akbarpour, "Formal Reasoning About Expectation Properties for Continuous Random Variables," in *Formal Methods*, ser. LNCS, vol. 5850. Springer, 2009, pp. 435–450.

[30] G. Vyzaite, S. Dunnett, and J. Andrews, "Cause-Consequence Analysis of Non-Repairable Phased Missions," *Reliability Engineering & System Safety*, vol. 91, no. 4, pp. 398–406, 2006.

[31] H. Xu and J. Dugan, "Combining Dynamic Fault Trees and Event Trees for Probabilistic Risk Assessment," in *Symposium Reliability and Maintainability*. IEEE, 2004, pp. 214–219.

[32] L. R. Olsen, J. A. Kay, and M. Van Krey, "Enhanced Safety Features in Motor Control Centers and Drives for Diagnostics and Troubleshooting," in *IAS Electrical Safety*. IEEE, 2015, pp. 1–9.

[33] M. Abdelghany, "Cause-Consequence Diagrams Formalization in HOL4," 2020. [Online]. Available: https://github.com/hvg-concordia/CCD

[34] R. N. Allan, *Reliability Evaluation of Power Systems*. Springer Science & Business Media, 2013.

[35] G. Bhatt and S. Affljulla, "Analysis of Large Scale PV Penetration Impact on IEEE 39-Bus Power System," in *Riga Technical University Conference on Power and Electrical Engineering*. IEEE, 2017, pp. 1–6.

[36] D. Gan, R. J. Thomas, and R. D. Zimmerman, "Stability-Constrained Optimal Power Flow," *IEEE Transactions on Power Systems*, vol. 15, no. 2, pp. 535–540, 2000.

[37] A. Alferidi and R. Karki, "Development of Probabilistic Reliability Models of Photo-Voltaic System Topologies for System Adequacy Evaluation," *Applied Sciences*, vol. 7, no. 2, p. 176, 2017.

[38] W. Li *et al.*, *Reliability Assessment of Electric Power Systems Using Monte Carlo Methods*. Springer Science & Business Media, 2013.

[39] G. J. Anders and A. Vaccaro, *Innovations in Power Systems Reliability*. Springer, 2011.

[40] A. K. Pradhan, S. K. Kar, P. Dash *et al.*, "Implementation of Monte Carlo Simulation to the Distribution Network for Its Reliability Assessment," in *Innovation in Electrical Power Engineering, Communication, and Computing Technology*. Springer, 2020, pp. 219–228.

[41] W. Ahmed, O. Hasan, and S. Tahar, "Formalization of Reliability Block Diagrams in Higher-Order Logic," *Journal of Applied Logic*, vol. 18, pp. 19–41, 2016.