Formalization of Functional Block Diagrams Using HOL Theorem Proving

Mohamed Abdelghany(✉) and Sofiène Tahar(✉)

Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada
{m_eldes,tahar}@ece.concordia.ca

Abstract. Functional Block Diagrams (FBD) are commonly used as a graphical representation for safety analysis in a wide range of complex engineering applications. An FBD models the stochastic behavior and cascading dependencies of system components or subsystems. Within FBD-based safety analysis, Event Trees (ET) dependability modeling techniques are typically used to associate all possible failure/success events to each subsystem. In this paper, we propose to use higher-order logic theorem proving for the formal modeling and step-analysis of FBDs. To this end, we develop a formalization in HOL4 enabling the mathematical modeling of the graphical diagrams of FBDs and the formal analysis of subsystem-level failure/reliability. The proposed FBD formalization in HOL4 is capable of analyzing *n-level* subsystems with *multi-state* system components and enables the formal FBD probabilistic analysis for any given probabilistic distribution and failure rates.

Keywords: Functional block diagrams · Event trees · Safety analysis · Higher-order logic · Theorem proving · HOL4

1 Introduction

In many safety-critical complex systems, a catastrophic accident may happen due to the coincident occurrence of multiple sudden events in different subsystem components. These undesirable accidents in safety-critical systems may result in huge financial losses and sometimes severe injury or fatalities. Therefore, the central safety inquiry in many complex systems is to identify the possible consequences given that one or more sudden events could happen at a subsystem level. For that purpose, several dependability modeling techniques have been developed for safety analysis of critical-systems, such as Fault Trees (FT) [14], Reliability Block Diagrams (RBD) [21] and Event Trees (ET) [18]. FTs and RBDs are used to either analyze the factors causing a complete system failure or the complete success relation ships of a system only, respectively. In contrast to FTs and RBDs, ETs provide a complete analysis for all possible complete/partial failure and success consequence scenarios that can occur in a system. Moreover, ET analysis can be used to associate failure and success events to all subsystems of a safety-critical system in more complex hierarchical structures, such as

L. Lima and V. Molnár (Eds.): SBMF 2022, LNCS 13768, pp. 22–35, 2022.
https://doi.org/10.1007/978-3-031-22476-8_2

Functional Block Diagrams (FBD) [9]. An FBD is a graphical representation of the detailed system functionality and the functional relationship between all its subsystems that are represented as Functional Blocks (FB). Each FB describes the failure characteristics of a subsystem by modeling its component failure and success relationship in terms of an ET structure [18]. All these subsystem level ETs associated with their corresponding FBs are then composed together to build a complete subsystem-level ET model of a complex system.

Papazoglou [9] was the first researcher to lay down the mathematical foundations of ETs and FBDs in the late 90s, where the analysis is done purely manually using a paper-and-pencil approach. A major limitation in the manual approach is the possibility of human error-proneness. On the other hand, there exist several simulation ET tools, such as ITEM [11], Isograph [10], and EC Tree [20], which have been widely used to determine all possible failure and success consequence scenarios of realistic systems, like electrical power grids [16], nuclear power plants [19] and Electric railways [12]. However, simulation based analysis approaches lack the rigor of detailed proof steps and may not be scalable for large systems due to an explosion of the test cases. To the best of our knowledge, these tools have not been used for FBD modeling and analysis. On the other hand, such simulation approaches generally use approximate random-based algorithms, such as MATLAB Monte-Carlo Simulation (MCS) for ET analysis [13], for faster computation, which could introduce undesirable inaccuracies that can be deemed fatal for safety-critical systems.

Following the recommendations of safety standards, such as IEC 61850 [15], EN 50128 [6], and ISO 26262 [17], we propose to use formal techniques based on theorem proving for the safety analysis of complex systems. In particular, we use the HOL4 theorem prover [8], which provides the ability of verifying probabilistic mathematical expressions constructed in higher-order logic (HOL). Prior to our work, there were two notable projects for building frameworks to formally analyze FTs and RBDs. For instance, HOL4 has been previously used by Ahmad et al. in [5] to formalize Static FTs and RBDs. Furthermore, Elderhalli et al. in [7] had formalized Dynamic FTs and RBDs in the HOL4 theorem prover. All these formalizations are basically required to formally analyze either a system static/dynamic failure or static/dynamic success only. Therefore, in [2], Abdelghany *et al.* developed a HOL4 theory to reason about ETs considering both failure and success states of system components simultaneously. The authors proposed a new datatype `EVENT_TREE` consisting of ET basic constructors that can analyze large scale ET diagrams. Based on [2], Abdelghany *et al.* have also developed the formalizations of cause consequence diagrams (CCD) in HOL4 to enable formal failure analyses combining, respectively, ETs with FTs [3] and ETs with RBDs [4]. These works allow the reasoning about all possible complete/partial failure and success consequences events that can occur at the subsystem level. However, a limitation of CCD analysis is that we can only assign two states to each subsystem (failure or success). While for realistic systems, safety and reliability engineers need to assign *multi-states* to subsystem components (e.g., partial failure, partial success, complete failure, complete

success). To this end, Functional Block Diagrams (FBD) would be the graphical representation of choice for the reliability analysis of *n-level* multi-state critical systems.

In this paper, we provide a formalization of Functional Block Diagrams that can mathematically model FBDs based on our ET theory in HOL4 to analyze multi-state subsystem components and obtain all possible consequence classes (e.g., partial failure, partial success, etc.) that can occur in the whole system at the subsystem level. The proposed formalization in HOL4 defines a basic FBD constructor *Functional Block* (FB), which can be used to build the mathematical expressions of *n-level* FBDs based on *multi-state* subsystem components. Also, the formalization of FBDs, in this paper, enables a formal probabilistic risk assessment of scalable graphical diagrams of FBDs that provides the reasoning support for formal safety analysis of complex systems at the subsystem-level based on any arbitrary probabilistic distribution and failure rates.

The rest of the paper is organized as follows: In Sect. 2, we review the recently developed ET theory in HOL4. Section 3 introduces the fundamentals of FBDs. In Sect. 4, we detail our proposed HOL4 formalization of FBDs. Lastly, Sect. 5 concludes the paper.

2 Preliminaries

Event Tree (ET) is a well-known probabilistic reliability and risk assessment technique, which provides all possible risk consequence scenarios that can occur in a safety-critical system, i.e., complete/partial failure and reliability [18]. An ET diagram starts by an *Initiating Node* from which all possible consequence scenarios of a sudden event that can occur in the system are drawn as *Branches* connected to *Proceeding Nodes* so that *only one* of these risk scenarios can occur, i.e., all possible ET consequence paths are *disjoint* and *distinct*.

2.1 Formal ET Modeling

The ET constructors are formally modeled using a new recursive datatype `EVENT_TREE`, in HOL4 as follows [1]:

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

The type constructors `NODE` and `BRANCH` are recursive functions on `EVENT_TREE`-typed. Also, a semantic function is defined over the `EVENT_TREE` datatype that can yield a corresponding ET model as [1]:

Definition 1: *Event Tree*

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

The function ETREE takes a success/fail event Y, identified by an ET type constructor ATOMIC and returns the event Y. If the function ETREE takes a list XN 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 given list. Similarly, if the function ETREE takes a success/fail event X and a proceeding ET Z, identified by a type constructor of EVENT_TREE type, then it performs the intersection of the event Y with the ET Z after applying the function ETREE. A complete ET model should draw all possible consequence scenarios, called *paths*. Each *path* consists of a unique consequence of branch events associated with it. A function ET_{PATH} is defined to obtain a specific path in the ET model consisting of M branch events. This was done in HOL4 by using the HOL4 recursive function FOLDL that recursively applies the BRANCH ET constructor on a given list of different M branch events as [2]:

Definition 2: *ET Path of M Events*

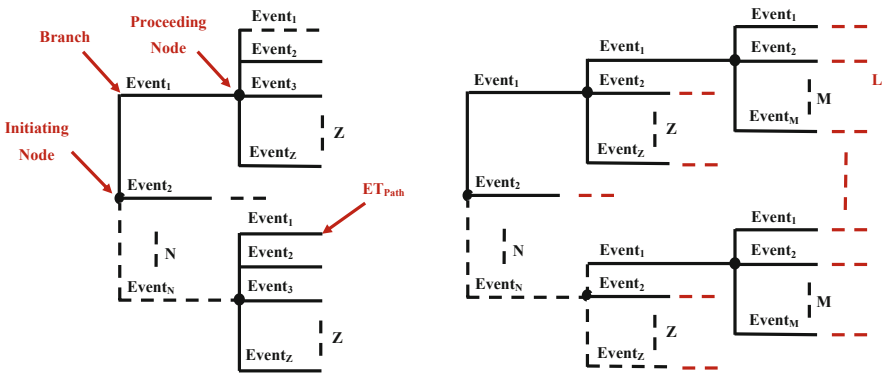
$\vdash \text{ET}_{PATH}$ p (EVENT$_1$::EVENT$_M$) =
FOLDL (λa b. ETREE (BRANCH a b)) EVENT$_1$ EVENT$_M$

A function $\bigotimes_{\text{L}}$ is defined that can model an ET diagram with all possible scenarios for two consecutive node lists L_1 and L_2, as shown in Fig. 1a, based on the mathematical Cartesian product $\bigotimes$ concept, in HOL4 as [2]:

Definition 3: *Two Stair ET Generation*

$\vdash \text{L}_1 \bigotimes_{\text{L}} \text{L}_2$ =
MAP (λa. MAP (λb. ETREE (BRANCH a b)) L$_2$) L$_1$

where the function $\bigotimes_{\text{L}}$ takes two different EVENT_TREE-typed lists and returns an EVENT_TREE-typed list by recursively mapping the BRANCH constructor on each element of the first NODE list paired with the entire second NODE list using the HOL4 mapping function MAP.



(a) Two Stair ET Generation (b) N Stair ET Generation

Fig. 1. Generic ET model generation

Also, a function $\bigotimes_{\mathrm{L}}^{N}$ is defined that generates a sequential and a complex ET model (see Fig. 1b) consisting of N components of a given system and each component is represented by a different M *multi-state* model for reliability studies (i.e., 2-state model, 3-state model, ..., M-state model), as shown in Fig. 2. in HOL4 as follows [2]:

Definition 4: *N Stair ET Generation*

$\vdash$ `L` $\bigotimes_{\mathrm{L}}^{N}$ L_N `= FOLDR (`λ`L`$_1$ `L`$_2$`. L`$_1$ $\bigotimes_{\mathrm{L}}$ `L`$_2$`) L`$_N$ `L`

where L is a *list* of all component states till $N-1$ (i.e., $\mathrm{L} = [[\mathcal{C}_1]; [\mathcal{C}_2]; \ldots; [\mathcal{C}_{N-1}]]$) and $\mathrm{L}_N = [\mathcal{C}_N]$.

Moreover, a reduction function $\boxtimes^{N}$ is defined in [2] to reduce the generated complete ET model. Lastly, a partitioning function $\boxplus$ is defined to extract a collection of ET paths specified in the index list N from the reduced ET model L representing the possibilities of an accident event, in HOL4 as [2]:

Definition 5: *ET Paths Partitioning*

$\vdash$ `N` $\boxplus$ `L = MAP (`λ`a. EL a L) N`

where the HOL4 function `EL` extracts a specific element from the given list.

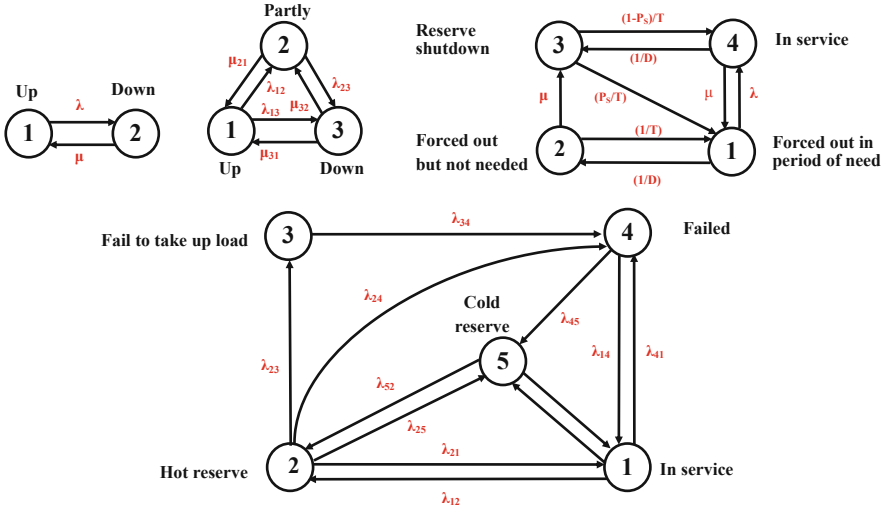
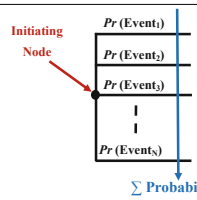
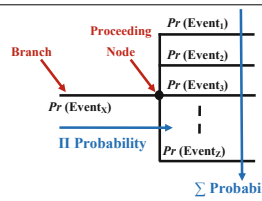
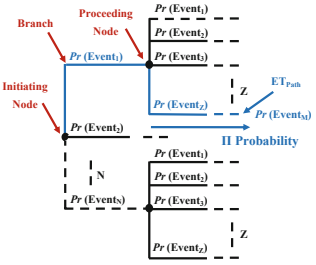
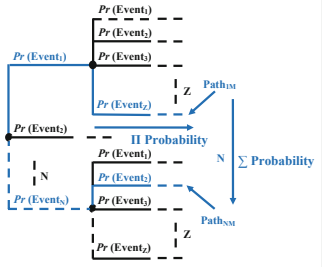


Fig. 2. Multi-state models for safety studies

2.2 Formal ET Probabilistic Analysis

For the formal ET probabilistic analysis, Abdelghany *et al.* verified, in [2], several mathematical ET probabilistic formulations, as presented in Table 1.

Table 1. ET HOL4 probabilistic theorems [2]

EventTrees	ProbabilisticTheorems
	**Theorem1:** probp (ETREE (NODE(E_1::E_N))) = $\sum$ (Pr_L p (E_1::E_N))
	**Theorem2:** probp (ETREE (BRANCH $Event_X$ (NODE(E_1::E_Z)))) =(probp E_X) × $\sum$ (Pr_L p(E_1::E_Z))
	**Theorem3:** probp (ET_{PATH} p(E_1::E_M)) = $\prod$ (Pr_L p(E_1::E_M))
	**Theorem4:** probp (ETREE (NODE (MAP(λa. ET_{PATH} pa) ($Path_{1M}$::$Path_{NM}$)))) = $\sum$ (MAP (λa. $\prod$ (Pr_L pa))($Path_{1M}$::$Path_{NM}$))

The probability of N events in an ET initiating node is verified as the sum of probabilities associated with the events of the given list. The probability of a branch success/fail event connected to a proceeding node is verified as

the multiplication of the branch event probability with the sum of the probabilities for the next proceeding node events. Also, the probability of ET_{PATH} consisting of M ET branch events is verified as the multiplication of the individual probabilities of all the branch events associated with it. Moreover, a *two-dimensional* probabilistic formulation is verified for extracting a collection of N paths and each of M success/fail events from an ET model, where each path consists of an arbitrary list of events, as the sum of the individual probabilities of all the paths associated with it. These mathematical expressions (Theorems 1–4) are verified under the ET constraints defined by Papazoglou [18] (a) all associated events in the given list X_N are drawn from the events space p ($X_N \in$ `events p`); (b) p is a valid probability space (`prob_space p`); (c) the events in the given list X_N are independent (`MUTUAL_INDEP p` X_N); (d) each pair of elements in a given list X_N is distinct (`ALL_DISTINCT` X_N); and lastly (e) each pair of elements in the given list X_N is mutually exclusive (`disjoint` X_N). The elements in a list are intrinsically finite and thus all ET constraint requirements are satisfied. The function $\mathrm{Pr_L}$ takes an arbitrary list $[Z_1, Z_2, Z_3, \ldots, Z_N]$ and returns a list of probabilities associated with the elements of the list $[Pr(Z_1), Pr(Z_2), , \ \ldots, Pr(Z_{N-1}), Pr(Z_N)]$, while the function $\prod$ takes a list $[Y_1, Y_2, Y_3, \ldots, Y_N]$ and returns the product of the list elements $Y_1 \times Y_2 \times Y_3 \times \cdots \times Y_N$. The function $\sum$ takes a list $[X_1, X_2, X_3, \ldots, X_N]$ and returns the sum of the list elements $X_1 + X_2 + X_3 + \cdots + X_N$.

3 Functional Block Diagrams

Functional Block Diagrams (FBDs) are a probabilistic risk assessment technique that can construct hierarchical ET structures to perform subsystem-level reliability analysis for complex systems. A Functional Block (FB) is the basic constructing element of an FBD graph that represents the stochastic behavior of each subsystem in a safety-critical system. To present a clear understanding of FBD-based safety analysis, consider a turbine governor system of a steam power plant that controls the position of a steam inlet valve (V), which in turn regulates the steam flow to the turbine and thus controls the output power. The valve operates with an induction motor (IM) that is energized by a power supply (PS), as shown in Fig. 3. The main objective of the valve is to control the Steam Flow (SF) at point B given the flow situation at point A and a command signal C that dictates the required function of the valve, i.e. open or close. The FBD *six* step-wise analysis, defined by Papazoglou [9], are as:

1. *FBD Construction*: A system FBD (decomposed into FBs) is constructed based on the engineering knowledge to describe the subsystem-level behavior, as shown in Fig. 4.
2. *ET Generation*: Construct a complete ET model corresponding to each subsystem FB. Assuming each subsystem component is represented by two operating states only, i.e., Success (S) or Fail (F). Figure 5 depicts the subsystem complete ETs, i.e., $\mathrm{ET}_{1(\mathrm{Complete})}$, $\mathrm{ET}_{2(\mathrm{Complete})}$ and $\mathrm{ET}_{3(\mathrm{Complete})}$ corresponding to FB_1, FB_2 and FB_3, respectively, of the steam-turbine governor.

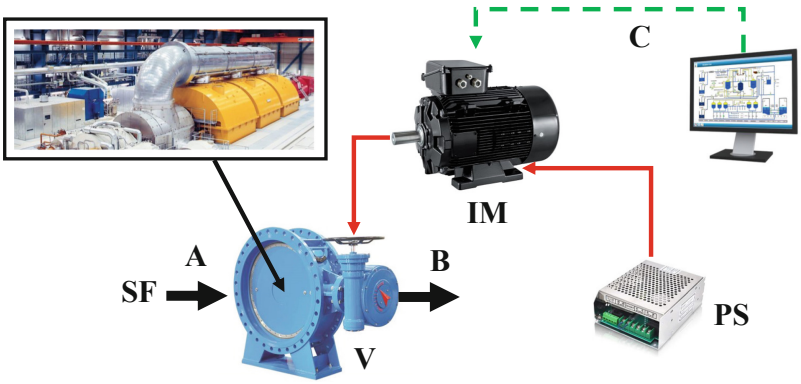


Fig. 3. Steam-turbine governor of a power plant

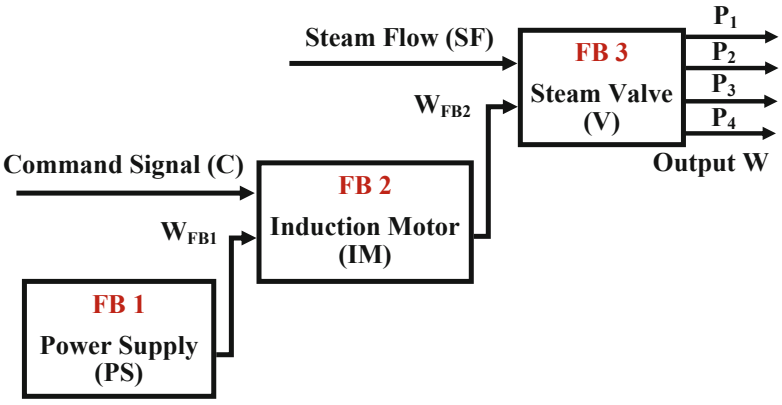


Fig. 4. FBD of steam-turbine governor

3. *ET Composition*: All ETs associated with their corresponding FBs are composed together considering the functional behavior of the governor system to form a complete subsystem-level ET model. For instance, $ET_{1(Complete)}$, $ET_{2(Complete)}$ and $ET_{3(Complete)}$ are composed to form the subsystem-level $ET_{Governor}$, as shown in Fig. 5, with all possible complete/partial failure and reliability ET consequence paths that can occur.
4. *Probabilistic Analysis*: Lastly, evaluate the probabilities of the system complete ET paths based on the occurrence of a certain event. These probabilities represent the likelihood of each unique sequence at the component-level that is possible to occur in a system so that *only one* can occur. For example, the probability of IM Complete Failure (CF) and Governor Complete Success (CS) shown in Fig. 5, i.e., $\sum_{probability(Paths\ 4-31)}$ and $Path_0$, respectively, can be expressed mathematically after shorthand as:

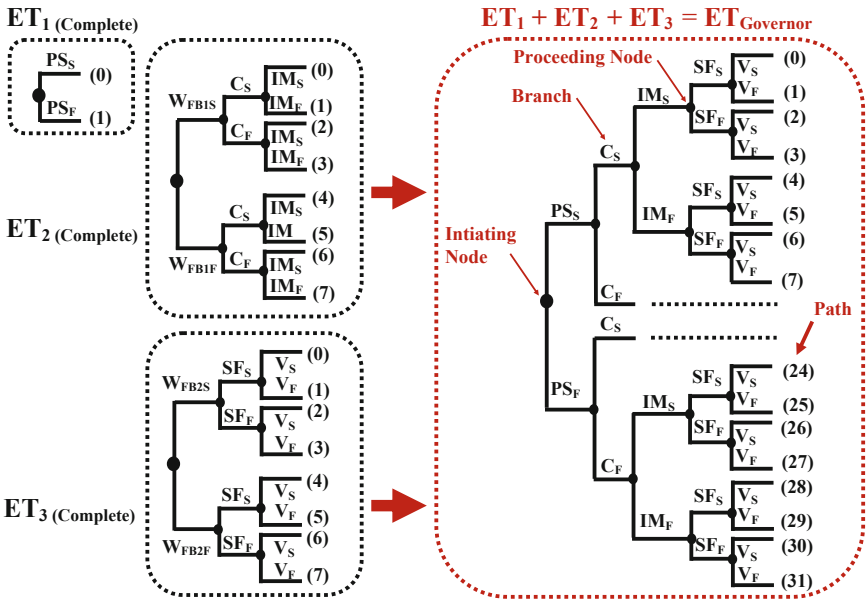


Fig. 5. Steam-turbine governor ET diagrams

$$
\begin{aligned}
Pr(\mathrm{IM}_{CF}) = Pr(\mathrm{PS}_S) \times Pr(\mathrm{C}_S) \times Pr(\mathrm{IM}_F) + \\
Pr(\mathrm{PS}_S) \times Pr(\mathrm{C}_F) \; + \; Pr(\mathrm{PS}_F) \\
Pr(\mathrm{Governor}_{CS}) = Pr(\mathrm{PS}_S) \times Pr(\mathrm{C}_S) \times Pr(\mathrm{IM}_S) \times \\
Pr(\mathrm{SF}_S) \times Pr(\mathrm{V}_S)
\end{aligned} \tag{1}
$$

where $Pr(X_F)$ is the probability of failure for a component X and $Pr(X_S)$ represents the correct functioning of the component, i.e., 1 – $Pr(X_F)$.

4 FBD Formalization

In this section, we describe, in detail, our proposed FBD formalization in the HOL4 theorem prover.

4.1 Formal FBD Modeling

We start the formalization of FBDs by defining a modeling function for its basic element FB, using Definition 4, as shown in Fig. 6, in HOL4 as follows:

Definition 6: *Functional Block*

$\vdash \mathcal{FB}\ (\mathcal{S}::\mathcal{I}_N) = \mathcal{I}_N \bigotimes_{\mathrm{L}}^{N} \mathcal{S}$

where $\mathcal{S}$ is a list of all subsystem internal components failure and success states and $\mathcal{I}_N$ is a *two-dimensional* list of all inputs states that affect the subsystem FB, i.e., $\mathcal{I}_N = [[\mathcal{I}_1]; [\mathcal{I}_2]; [\mathcal{I}_3]; \ldots; [\mathcal{I}_n]]$. Also, we can obtain the ET model of a specific functional block $\mathcal{FB}_j$ by defining a function $\mathcal{FB}_{ET}$, in HOL4 as follows:

Definition 7: *Functional Block ET*

$\vdash \mathcal{FB}_{ET}\ \mathcal{FB}_j$ = ETREE (NODE $\mathcal{FB}_j$)

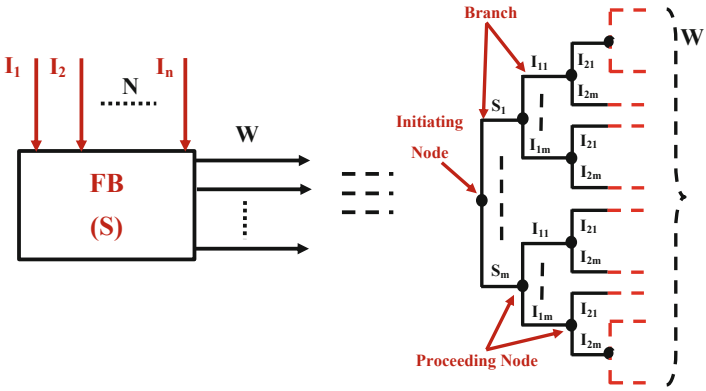


Fig. 6. An FB equal to a complete ET model

To construct multiple consecutive N FBs, we define the following recursive function $\mathcal{FB}_{ET}^{N}$, in HOL4 as follows:

Definition 8: *Multiple Functional Block ET*

$\vdash \mathcal{FB}_{ET}^{N}\ (\mathcal{FB}_1 :: \mathcal{FB}_N)$ = $(\mathcal{FB}_{ET}\ \mathcal{FB}_1) :: (\mathcal{FB}_{ET}^{N}\ \mathcal{FB}_N)$

In order to verify the correctness of the above-mentioned functions, we formalize the following FBD modeling properties, in HOL4 as follows:

Property 1: An ET diagram of an FB model having N input lists $\mathcal{I}_N$ and an internal state list $\mathcal{S}$ can be *split* as connected individual FBs for all lists associated with the FB model, as shown in Fig. 7, in HOL4 as:

Theorem 5: *Splitting Single Functional Block*

$\vdash \mathcal{FB}_{ET}\ (\mathcal{FB}\ (\mathcal{S} :: \mathcal{I}_N))$ = ET_{PATH} p $(\mathcal{FB}_{ET}^{N}\ (\mathcal{S} :: \mathcal{I}_N))$

Property 2: The *commutativity* and *associativity* properties of two consecutive FBs consisting of N input lists $\mathcal{I}_N$, as shown in Fig. 8, in HOL4 as:

Theorem 6: *Commutativity and Associativity of Two FBs*

$\vdash \mathcal{FB}_{ET}\ (\mathcal{FB}\ (\mathcal{I}_1 :: \mathcal{I}_N) \bigotimes_{\mathrm{L}} \mathcal{I}_2)$ = $\mathcal{FB}_{ET}\ (\mathcal{I}_1 \bigotimes_{\mathrm{L}} (\mathcal{FB}\ (\mathcal{I}_2 :: \mathcal{I}_N)))$

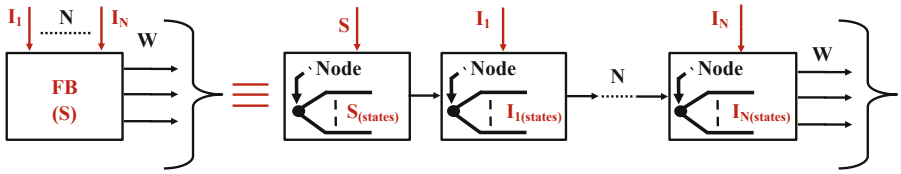


Fig. 7. An FB of N inputs split into individual FBs

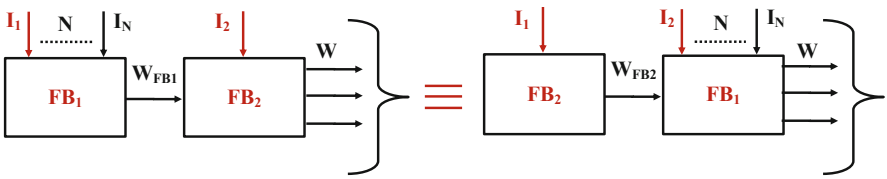


Fig. 8. Commutativity and associativity of two FBs

Now, we can define a *three-dimensional* function $\mathcal{FB}_N$ that takes N FBs, where each FB takes an arbitrary list of n-inputs and then generates the corresponding complete FBD model to obtain all possible risk consequences of failure and reliability, as shown in Fig. 9, in HOL4 as:

Definition 9: *Three Dimensional N Functional Blocks*

$\vdash \mathcal{FB}_N\ (\mathcal{SI}_1::\mathcal{SI}_2::\mathcal{SI}_N) = \mathcal{FB}\ (\texttt{MAP}\ (\lambda \texttt{a}.\ \mathcal{FB}\ \texttt{a})\ (\mathcal{SI}_1::\mathcal{SI}_2::\mathcal{SI}_N))$

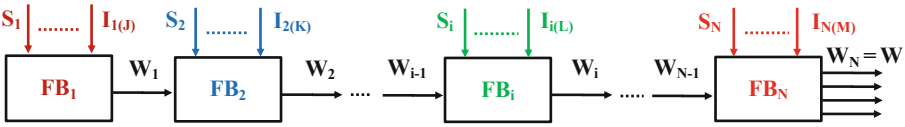


Fig. 9. Complete FBD model of multi-level FBs connected together

The next steps of the FBD analysis are to reduce and partition the ET model for each FB. Since the outcome of $\mathcal{FB}$ is a list of all risk events, we can use the same reduction function $\boxtimes^N$ and partitioning function $\boxplus$ for ET analysis to reduce the ET model and partition a collection of consequence events that end with the same risk events.

4.2 Formal FBD Probabilistic Analysis

The last step in the FBD analysis is to determine the probability of each ET consequence possible scenario at the subsystem-level that could occur in the complex system. Based on the ET probabilistic theorems (Theorems 1–4 in Table 1) and the formal FBD modeling theorems (Theorems 5 and 6), we have verified some FBD probabilistic theorems, in HOL4 as:

Property 3: The probability of the Cartesian product function $\bigotimes_{\mathrm{L}}$ for two $\mathcal{FB}$ lists X_N and Y_N, as shown in Fig. 10a, is verified as the multiplication of the sum of the individual probabilities of all the events associated with each list, in HOL4 as:

Theorem 7: *Two FBs of One Inputs*

$$\vdash \texttt{prob p}\ \left(\mathcal{FB}_{ET}\ (\mathrm{X}_N \bigotimes_{\mathrm{L}} \mathrm{Y}_M)\right) = \sum\ (\mathrm{Pr}_{\mathrm{L}}\ \texttt{p}\ \mathrm{X}_N) \times \sum\ (\mathrm{Pr}_{\mathrm{L}}\ \texttt{p}\ \mathrm{Y}_M)$$

where the function $\sum$ takes a list Y_M and returns the sum of the elements of a list, i.e., $Y_1+Y_2+Y_3+Y_4+\cdots+Y_{N-1}+Y_N$ while the function Pr_{L} returns the probabilities of the elements of a list, i.e., $[Pr(Z_1), Pr(Z_2),, \ \ldots, Pr(Z_{N-1}), Pr(Z_N)]$.

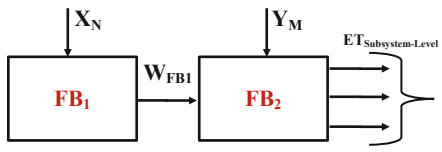
Property 4: A generic probabilistic formulation for one $\mathcal{FB}$ associated with N component multi-state lists, as shown in Fig. 10b, is verified as the product of the sum of each component list probabilities, in HOL4 as:

Theorem 8: *One FB of N Inputs*

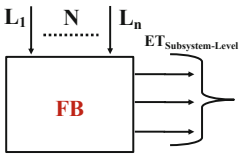
$$\vdash \texttt{prob p}\ \left(\mathcal{FB}_{ET}\ (\mathcal{FB}\ (\mathrm{L}_1 :: \mathrm{L}_N))\right) = \prod\ (\textstyle\sum_{\mathrm{prob}}\ \texttt{p}\ (\mathrm{L}_1 :: \mathrm{L}_N))$$

where the function $\sum_{\mathrm{prob}}$ is used to recursively apply the functions Pr_{L} and $\sum$ on a given *two-dimensional* list L_N, i.e., $[[\mathrm{L}_1]; [\mathrm{L}_2]; [\mathrm{L}_3]; \ldots; [\mathrm{L}_n]]$.

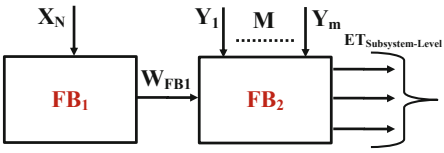
Property 5: A probabilistic formulation for two FBs of one list and N lists, as shown in Fig. 10c, is verified as the multiplication of their probabilities, in HOL4 as:



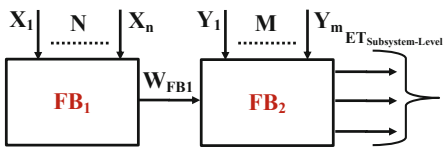
(a) 2 FBs of 1 Inputs



(b) 1 FB of N Inputs



(c) 2 FBs of 1 and N Inputs



(d) 2 FBs of 2 N Inputs

Fig. 10. Different configurations of connected FBs

Theorem 9: *Two FBs of One Input and N Inputs*

$$\vdash \texttt{prob p}\left(\mathcal{FB}_{ET}\ (\mathrm{X}_N \bigotimes_{\mathrm{L}} (\mathcal{FB}\ (\mathrm{Y}_1::\mathrm{Y}_m)))\right) = \sum (\mathrm{Pr}_{\mathrm{L}}\ \texttt{p}\ \mathrm{X}_N) \times \prod (\sum_{\mathrm{prob}}\ \texttt{p}\ (\mathrm{Y}_1::\mathrm{Y}_m))$$

Property 6: A probabilistic formulation for two FBs of N input lists, as shown in Fig. 10d, is verified as the multiplication of both probabilities, in HOL4 as:

Theorem 10: *Two FBs of Two N Inputs*

$$\vdash \texttt{prob p}\left(\mathcal{FB}_{ET}\ \left(\mathcal{FB}\ (\mathrm{X}_1::\mathrm{X}_m) \bigotimes_{\mathrm{L}} \mathcal{FB}\ (\mathrm{Y}_1::\mathrm{Y}_m)\right)\right) = \prod (\sum_{\mathrm{prob}}\ \texttt{p}\ (\mathrm{X}_1::\mathrm{X}_n)) \times \prod (\sum_{\mathrm{prob}}\ \texttt{p}\ (\mathrm{Y}_1::\mathrm{Y}_m))$$

The prime purpose of the above-developed formalization of FBDs is to build a reasoning support for the subsystem-level formal safety analysis of complex systems within the sound environment of HOL4. Our proposed formalization is capable of enabling the verification of safety properties of complete/partial failure of critical systems of any size and compute their reliability events simultaneously. For instance, our FBD formalization framework can handle systems consisting of multi-level decomposition subsystems, where each subsystem is composed of multiple components and each component is associated with multi-state failure and success consequence events [1].

5 Conclusions

In this paper, we described the formalization of FBDs step-analysis in HOL theorem proving using a generic list data-type. Our proposed formalization provides the mathematical verification of the graphical FBDs diagrams of complex systems associated with multi-state components and based on any given probabilistic distribution. The proposed formal approach enables safety engineers to perform FBD-based safety analysis of n-level complex systems within the sound environment of HOL4. We believe that our work will help safety design engineers to meet the desired quality requirements. As future work, we plan to apply the proposed the FBD formalization in the safety analysis of real world case studies. We also intend to develop an integrated framework with a GUI for FBD modeling and linking ET tools with the FBD formalization in HOL4.

References

1. Abdelghany, M.: Formal probabilistic risk assessment using theorem proving with applications in power systems. Ph.D. thesis, Concordia university, Montreal, QC, Canada (2021)
2. Abdelghany, M., Ahmad, W., Tahar, S.: Event tree reliability analysis of safety-critical systems using theorem proving. IEEE Syst. J. **16**(2), 2899–2910 (2022)

3. Abdelghany, M., Tahar, S.: Cause-consequence diagram reliability analysis using formal techniques with application to electrical power networks. IEEE Access **9**, 23929–23943 (2021)
4. Abdelghany, M., Tahar, S.: Formalization of RBD-based cause consequence analysis in HOL. In: Kamareddine, F., Sacerdoti Coen, C. (eds.) CICM 2021. LNCS (LNAI), vol. 12833, pp. 47–64. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-81097-9_4
5. Ahmad, W., Hasan, O., Tahar, S.: Formal reliability and failure analysis of ethernet based communication networks in a smart grid substation. Formal Aspects Comput. **31**, 321–351 (2019)
6. Boulanger, J.L.: CENELEC 50128 and IEC 62279 Standards. Wiley, Hoboken (2015)
7. Elderhalli, Y., Hasan, O., Tahar, S.: A framework for formal dynamic dependability analysis using HOL theorem proving. In: Benzmüller, C., Miller, B. (eds.) CICM 2020. LNCS (LNAI), vol. 12236, pp. 105–122. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-53518-6_7
8. HOL Theorem Prover. https://hol-theorem-prover.org
9. Papazoglou, I.: Functional block diagrams and automated construction of event trees. Reliab. Eng. Syst. Saf. **61**(3), 185–214 (1998)
10. Isograph (2022). https://www.isograph.com
11. ITEM (2021). https://itemsoft.com/eventtree.html
12. Ku, B.H., Cha, J.M.: Reliability assessment of catenary of electric railway by using FTA and ETA analysis. In: Environment and Electrical Engineering, pp. 1–4. IEEE (2011)
13. Li, W.: Reliability Assessment of Electric Power Systems Using Monte Carlo Methods. Springer, Heidelberg (2013)
14. Limnios, N.: Fault Trees. Wiley, Hoboken (2013)
15. Mackiewicz, R.E.: Overview of IEC 61850 and benefits. In: Power Systems Conference and Exposition, pp. 623–630. IEEE (2006)
16. Muzik, V., Vostracky, Z.: Possibilities of event tree analysis method for emergency states in power grid. In: Electric Power Engineering Conference, pp. 1–5. IEEE (2018)
17. Palin, R., Ward, D., Habli, I., Rivett, R.: ISO 26262 safety cases: compliance and assurance. In: IET Conference on System Safety, pp. 1–6 (2011)
18. Papazoglou, I.: Mathematical foundations of event trees. Reliab. Eng. Syst. Saf. **61**(3), 169–183 (1998)
19. Peplow, D.E., Sulfredge, C.D., Sanders, R.L., Morris, R.H., Hann, T.A.: Calculating nuclear power plant vulnerability using integrated geometry and event/fault-tree models. Nucl. Sci. Eng. **146**(1), 71–87 (2004)
20. Sen, D.K., Banks, J.C., Maggio, G., Railsback, J.: Rapid development of an event tree modeling tool using COTS software. In: Aerospace Conference, pp. 1–8. IEEE (2006)
21. Trivedi, K., Bobbio, A.: Reliability block diagrams. In: Reliability and Availability Engineering: Modeling, Analysis, and Applications, pp. 105–149. Cambridge University Press (2017)