Formalization of RBD-Based Cause Consequence Analysis in HOL

Mohamed Abdelghany(✉) and Sofiène Tahar(✉)

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

Abstract. Cause consequence analysis is a safety assessment technique that is traditionally used to model the causes of subsystem failures in a critical system and their potential consequences using Fault Tree and Event Tree (ET) dependability modeling techniques, combined in a graphical Cause-Consequence Diagram (CCD). In this paper, we propose a novel idea of using Reliability Block Diagrams (RBD) for CCD analysis based on formal methods. Unlike Fault Trees, RBDs allow to model the success relationships of subsystem components to keep the entire subsystem reliable. To this end, we formalize in higher-order logic new mathematical formulations of CCD functions for the RBD modeling of generic n-subsystems using HOL4. This formalization enables universal n-level CCD analysis, based on RBDs and ETs, by determining the probabilities of multi-state safety classes, i.e., complete/partial failure and success, that can occur in the entire complex systems at the subsystem level.

Keywords: Cause-Consequence Diagram · Reliability Block Diagram · Event Tree · Higher-order logic · Theorem proving

1 Introduction

Since the late 60's, various types of dependability modeling techniques have been developed to determine the safety assessment of safety-critical systems, such as smart grids and automotive industry. These include predominantly graph theory based approaches such as Fault Trees (FT) [18], Reliability Block Diagrams (RBD) [7] and Event Trees (ET) [14]. FTs mainly provide a graphical model for analyzing the factors causing a complete system failure upon their occurrences. On the other hand, RBDs provide a schematic structure for analyzing the success relationships of system components that keep the entire system reliable. In contrast to FTs and RBDs, ETs provide a tree model for all possible complete/partial failure and success scenarios at the system-level so that one of these possible scenarios can occur [14]. More recently, an approach has been proposed to conduct ET analysis in conjunction with FTs to identify all subsystem failure events in a critical system and their cascading dependencies on the entire system [16]. This analysis method is known as cause-consequence analysis, using a combined hierarchical structure of Cause-Consequence Diagrams (CCD) [16].

F. Kamareddine and C. Sacerdoti Coen (Eds.): CICM 2021, LNAI 12833, pp. 47–64, 2021.
https://doi.org/10.1007/978-3-030-81097-9_4

Traditionally, CCD analysis based on FTs and ETs is carried out by using paper-and-pencil approaches (e.g., [5]) or computer simulation tools (e.g., [20]). The major limitations of the manual analysis approach are its human-error proneness and scalability to handle large complex systems [19]. On the other hand, simulation-based analysis approaches, such as MATLAB Monte-Carlo Simulation (MCS), can be used for CCD analysis for faster computation. They, however, lack the rigor of detailed proof steps and absolute accuracy (i.e., results approximation) due to an explosion of the test cases [20]. A more practical way to remedy the shortcomings of informal reasoning approaches of cause-consequence analysis is to use formal generic mathematical formulations that can analyze large-scale CCD graphs. Only a few works have previously considered using formal methods for cause-consequence analysis. For instance, Ortmeier et al. in [13] developed a formal framework for Deductive Cause-Consequence Analysis (DCCA) using the SMV model checker to formally verify probabilistic properties for CCD analysis. However, according to the authors of [9], there is a scalability problem of showing the completeness of DCCA due to the exponential growth of the number of proof obligations with large complex CCD graphs. For that reason, to overcome the above-mentioned limitations, we endeavor to solve the scalability problem of CCDs by using theorem proving, in particular the HOL4 proof assistant [10], which provides the ability of verifying generic expressions constructed in higher-order logic (HOL).

Prior to this work, there were three notable projects for building formal infrastructures in HOL to formally model and analyze FTs, RBDs and ETs. For instance, Ahmad [4] used the HOL4 theorem prover to formalize ordinary (static) FT and RBD structures. Elderhalli [8] had formalized dynamic versions of FTs and RBDs in HOL4. These formalizations have been used for the reliability analysis of several engineering systems. However, they formally analyze either a critical system static/dynamic failure or static/dynamic success only. Therefore, in [2], we developed a HOL4 theory to reason about ETs considering all failure and success events of system-level components simultaneously. We proposed a new datatype `EVENT_TREE` consisting of ET basic constructors that can build large scale ET diagrams and provides us with the ability to obtain a verified system-level failure/operating expression. Moreover, in [3], we proposed a formal approach for state-of-the-art CCD analysis using the above static FT and ET formalizations, which enables safety analysts to perform formal failure analysis for n-level subsystems of a complex system and obtain all possible complete/partial failure and success consequences events that can occur in HOL4. However, in order to identify potential areas of poor reliability, safety analysts often require a reliability model that is close to the hierarchical structure of the subsystem components. For that reason, we propose in this paper a novel approach to conduct a CCD analysis based on RBDs rather than FTs. In particular, we develop new formulations of CCDs based on RBD and ET theories, and provide their formalization using HOL theorem proving.

Unlike FT-based CCD analysis, RBDs allow to model all success relationships of n-subsystems to keep them reliable and obtain multi-state consequence safety

classes, i.e., complete/partial failure and complete/partial success, that can occur in the entire critical system at the subsystem level. To the best of our knowledge, the idea of using RBD modeling in conjunction with the graph theory of CCDs has not been proposed before. We propose new mathematical formulations that can analyze scalable CCDs associated with different RBD configurations to n-subsystems. In order to check the correctness of the newly-proposed equations, we verified them within the sound environment of the HOL4 theorem prover. To this end, we formalize in HOL4 cause-consequence functions for the formal modeling of the graph theory of RBDs corresponding to generic n-subsystems. Also, our proposed formalization enables the formal probabilistic assessment of large scale n-level CCD structures based on any probabilistic distribution, which makes our work the first of its kind.

The rest of the paper is organized as follows: In Sect. 2, we describe some preliminaries of RBDs and ETs to facilitate understanding of the rest of the paper. Section 3 presents the proposed formalization of CCDs based on RBDs and ETs, including the newly introduced probabilistic formulations and their verification in the HOL4 theorem prover. Lastly, Sect. 4 concludes the paper.

2 Preliminaries

In this section, we summarize the fundamentals of existing RBD and ET formalizations in HOL4 to facilitate the reader's understanding of the rest of the paper.

2.1 RBD Formalization

Reliability Block Diagram [7] (RBD) analysis is one of the commonly used safety assessment techniques for critical systems. It mainly provides a schematic diagram for analyzing the success relationships of subsystem components that keep the entire subsystem reliable. An RBD structure consists of blocks that represent the subsystem components and connectors that indicate the connections between these components. An RBD has two main types of configuration patterns *series* and *parallel*. The reliability of a subsystem when its components are connected in series configuration is considered to be reliable at time t only if all of the components are functioning reliably at time t, then the overall reliability $\mathcal{R}$ of the subsystem can be mathematically expressed as [7]:

$$\mathcal{R}_{series}(t) = Pr\left(\bigcap_{i=1}^{N} X_i(t)\right) = \prod_{i=1}^{N} \mathcal{R}_i(t) \tag{1}$$

Similarly, the reliability of a subsystem where its components connected in parallel will continue functioning at a specific time t as long as at least one of its components remains functional, which can be mathematically expressed as [7]:

$$\mathcal{R}_{parallel}(t) = Pr\left(\bigcup_{i=1}^{N} X_i(t)\right) = 1 - \prod_{i=1}^{N} (1 - \mathcal{R}_i(t)) \tag{2}$$

Ahmad et al. in [4] presented the RBD formalization by defining a new datatype `rbd`, in HOL4 as:

```
Hol_datatype rbd = series of (rbd list) | parallel of (rbd list) |
                   atomic of (event)
```

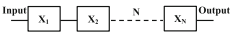
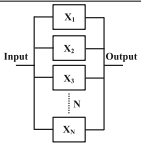
The RBD constructors `series` and `parallel` are recursive functions on `rbd`-typed lists, while the RBD constructor `atomic` operates on an `rbd`-type variable. A semantic function is then defined over the `rbd` datatype that can yield mathematically the corresponding RBD diagram as:

Definition 1

```
⊢ rbd_struct p (atomic X) = X ∧
  rbd_struct p (series []) = p_space p ∧ rbd_struct p (parallel []) = {} ∧
  rbd_struct p (series (X::XN)) =
  rbd_struct p X ∩ rbd_struct p (series XN) ∧
  rbd_struct p (parallel (X::XN)) =
  rbd_struct p X ∪ rbd_struct p (parallel XN)
```

The function `rbd_struct` takes a single event `X`, identified by a basic type constructor `atomic`, and returns the given event `X`. If the function `rbd_struct` takes an arbitrary list of type `rbd`, identified by a type constructor `series`, then it performs the intersection of all elements after applying the function `rbd_struct` on each element of the given list. Similarly, if the function `rbd_struct` takes an arbitrary list of type `rbd`, identified by a type constructor `parallel`, then it returns the union of all elements after applying the function `rbd_struct` on each element of the list X_N. The formal verification in HOL4 for the reliability series and parallel probabilistic expressions Eq. 1 and Eq. 2, respectively, is presented in Table 1 [4]. These mathematical expressions (Theorems 1–2) are verified under the constraints that (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`); and lastly (c) the events in the given list X_N are independent (`MUTUAL_INDEP p` X_N). The function `PROB_LIST` takes an arbitrary list $[Z_1, Z_2, Z_3, \ldots, Z_N]$ and returns a list of probabilities associated with the

Table 1. RBD probabilistic theorems [4]

RBD Connection	Probabilistic Theorem
	**Theorem 1:** `prob p (rbd_struct p (series` X_N`)) =` $\prod$ `(PROB_LIST p` X_N`)`
	**Theorem 2:** `prob p (rbd_struct p (parallel` X_N`)) =` `1 -` $\prod$ `(PROB_LIST p (COMPL_LIST p` X_N`))`

elements of the list [prob p Z_1, prob p Z_2, prob p Z_3, ..., prob p Z_N], while the function COMPL_LIST takes a list $[X_1, X_2, X_3, \ldots, X_N]$ and returns the complement of all elements in the list $[(1 - X_1), (1 - X_2), (1 - X_3), \ldots, (1 - X_N)]$. The function $\prod$ takes a list $[Y_1, Y_2, Y_3, \ldots, Y_N]$ and returns the product of the list elements $Y_1 \times Y_2 \times Y_3 \times \cdots \times Y_N$.

2.2 ET Formalization

Event Tree [14] (ET) is a widely used dependability modeling technique that can model all possible system-level components failure and success states and their cascading dependencies on the entire system in the form of a tree structure. 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 scenarios can occur, i.e., all possible ET consequence paths are disjoint (mutually exclusive) and distinct. These ET constructors were formally modeled using a new recursive datatype EVENT_TREE, in HOL4 as [2]:

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

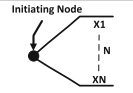
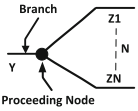
The type constructors NODE and BRANCH are recursive functions on EVENT_TREE-typed. A semantic function is then defined over the EVENT_TREE datatype that can yield a corresponding ET diagram as:

Definition 2

```
⊢ ETREE (ATOMIC Y) = Y ∧ ETREE (NODE []) = {} ∧
  ETREE (NODE (X::XN)) = ETREE X ∪ (ETREE (NODE XN)) ∧
  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 X with the ET Z after applying the function ETREE. 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 2 [2]. These expressions are formally verified under the same RBD constraints, i.e., $X_N \in$ events p, prob_space p, MUTUAL_INDEP p X_N, as well as the ET constraints defined by Papazoglou [14] (*distinct*, *disjoint*, *finite*), i.e., ALL_DISTINCT X_N and disjoint X_N to ensure that each pair of elements in a given list X_N is distinct and mutually exclusive, respectively. The elements in a list are intrinsically finite and thus all ET constraint requirements are satisfied. The function $\sum$ takes a list $[X_1, X_2, X_3, \ldots, X_N]$ and returns the sum of the list elements $X_1 + X_2 + X_3 + \cdots + X_N$.

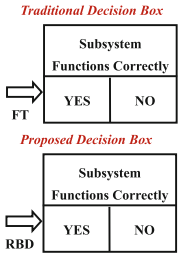
Table 2. ET probabilistic theorems [2]

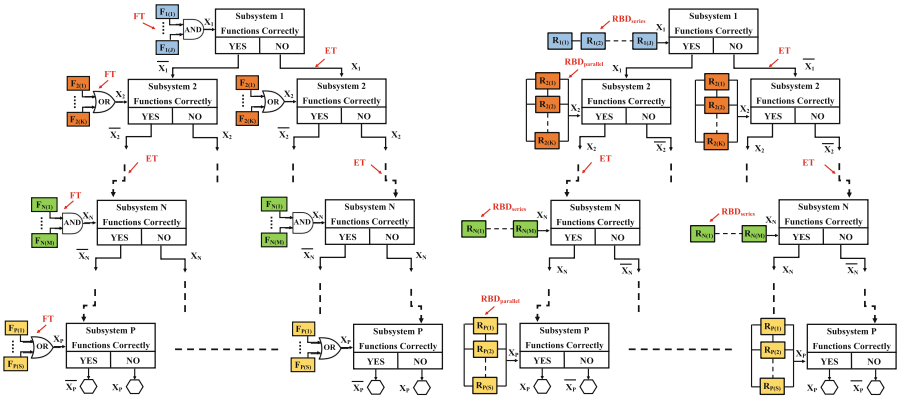
ET Constructor	Probabilistic Theorem
	**Theorem 3:** prob p (ETREE (NODE X_N)) = $\sum$ (PROB_LIST p X_N)
	**Theorem 4:** prob p (ETREE (BRANCH Y (NODE $\mathcal{Z}_\mathcal{N}$)) = (prob p Y) $\times$ $\sum$ (PROB_LIST p Z_N)

3 Cause-Consequence Diagram Formalization

The graph theory of CCDs [21] uses three basic constructors *Decision box*, *Consequence path* and *Consequence box* [6]. The detailed description of the CCD constructors is illustrated in Table 3. To present a clear understanding of these concepts, the traditional FT/ET-based CCD analysis for n-subsystems is described in Fig. 1a. As shown in Fig. 1a, FT *logic*-gates, such as AND (models the complete failure of the subsystem if all of the input failure events occur at the same time) and OR (models the complete failure of the subsystem if any of the input failure events occurs alone), are associated with all decision boxes to model the failure of generic n-subsystems. It can be noticed from Fig. 1a that the output of each `NO BOX` for all decision boxes is equal to the subsystem FT model ($\mathrm{FT_X}$), while the `YES BOX` is the complement of the FT model ($\overline{\mathrm{FT_X}}$). Analogously to

Table 3. CCD symbols and functions [3]

CCD Symbol	Function
	`Decision Box:` represents the status of functionality for a component or subsystem. (1) `NO Box`: describes the subsystem failure operation. An FT or RBD of the subsystem is connected to this box that can be used to determine the failure probability, i.e., $\mathcal{P}_{\mathrm{NO}} = \mathcal{P}_{\mathrm{FT}} = 1 - \mathcal{P}_{\mathrm{RBD}}$ (2) `YES Box`: represents the correct functioning of the subsystem or reliability, which can be calculated by simply taking the complement of the failure operation, i.e., $\mathcal{P}_{\mathrm{YES}} = 1 - \mathcal{P}_{\mathrm{FT}} = \mathcal{P}_{\mathrm{RBD}}$
	`Consequence Path:` models the next possible scenarios due to the occurrence of subsystem failure or reliability
	`Consequence Box:` models the final outcome event due to a particular sequence of events for all connected subsystems



(a) Classical FT/ET-based CCD Analysis [3]

(b) Proposed RBD/ET-based CCD Analysis

Fig. 1. Cause consequence analysis models

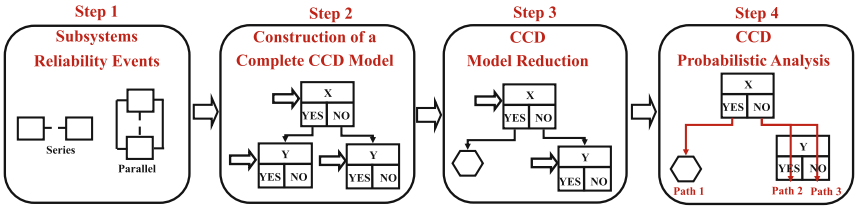


Fig. 2. Overview of RBD-based CCD analysis [3]

Fig. 1a, Fig. 1b illustrates the proposed RBD/ET-based CCD analysis, where different RBD configurations, such as *series* (models the complete success of the subsystem if all of the input success events occur at the same time) and *parallel* (models the complete success of the subsystem if any of the input success events occurs alone), are associated with all CCD decision boxes to model the reliability of generic n-subsystems. As shown in Fig. 1b, the output of each `YES BOX` for all decision boxes is equal to the RBD outcome ($\mathrm{RBD_X}$), while the `NO BOX` is the complement of the RBD model ($\overline{\mathrm{RBD_X}}$).

Figure 2 depicts the overview of the developed *four* steps of cause-consequence safety analysis for complex systems [5]: (1) *Subsystems reliability events*: identify the success events for all subsystems using RBD models that keep the subsystems reliable in a complex system; (2) *Construction of a complete CCD*: build a full CCD diagram using its basic constructors (see Table 3) 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 model representing the actual functional behavior of the complex system and reduce the number of test cases; and (4) *CCD probabilistic analysis*: determine the probabilities of all CCD

consequence paths, which represent the likelihood of specific sequence scenarios that are possible to occur in a system so that *only one* scenario can occur [19]. This implies that all consequences in a CCD are mutually exclusive [6]. As an example, consider a Wind Turbine system [15] consisting of two main subsystems: Induction Generator (IG) and Power Converter (PC), as shown in Fig. 3a [11]. An IG consists of three components *Stator*, *Rotor* and *Brushes* [12], while a PC consists of four components *Rotor Side AC/DC Converter* (RSC), *DC Filter*, *Grid Side DC/AC Converter* (GSC) and *Control Unit* (CU) [17]. The *four* main steps of the above-mentioned RBD/ET-based cause-consequence analysis for the wind turbine system can be done as follows:

1. *Components reliability events*: Assign an RBD series configuration to each subsystem in the wind turbine, i.e., $\mathcal{R}_{\text{IG}}$, $\mathcal{R}_{\text{PC}}$, as shown in Fig. 3b [11], which can be expressed mathematically as:

$$\mathcal{R}_{\text{IG}} = \mathcal{R}_{\text{stator}} \times \mathcal{R}_{\text{rotor}} \times \mathcal{R}_{\text{brushes}} \tag{3}$$

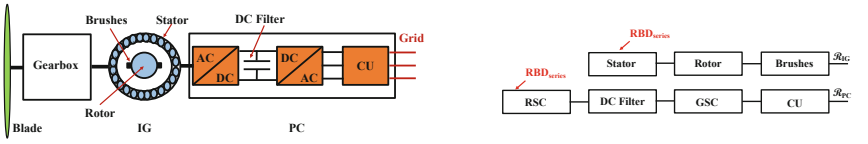
$$\mathcal{R}_{\text{PC}} = \mathcal{R}_{\text{RSC}} \times \mathcal{R}_{\text{filter}} \times \mathcal{R}_{\text{GSC}} \times \mathcal{R}_{\text{CU}} \tag{4}$$

2. *Construction of a complete CCD*: Draw a complete CCD model of the wind turbine system, as shown in Fig. 4a. For instance, if the condition of the IG decision box is either YES or NO, then the next subsystem PC is taken into consideration. Each consequence path in the CCD analysis ends with either a wind turbine success (WT_S) or a wind turbine failure (WT_F).
3. *CCD model reduction*: Apply the reduction operation on the constructed complete CCD model. For instance, if the condition of the IG decision box (IG functions correctly) is not satisfied, i.e., NO box, then the wind turbine fails regardless of the status of PC. Figure 4b represents the minimal RBD/ET-based cause consequence analysis of the wind turbine operation.
4. *CCD probabilistic analysis*: The probabilistic assessment of the two consequence boxes WT_S and WT_F in Fig. 4b can be expressed mathematically as:

$$\mathcal{P}(Consequence\_Box_{WT_S}) = \mathcal{P}(\text{IG}_{\text{YES}}) \times \mathcal{P}(\text{PC}_{\text{YES}}) \tag{5}$$

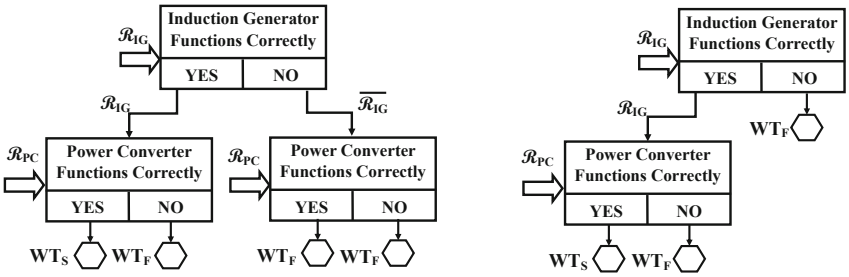
$$\mathcal{P}(Consequence\_Box_{WT_F}) = \mathcal{P}(\text{IG}_{\text{YES}}) \times \mathcal{P}(\text{PC}_{\text{NO}}) + \mathcal{P}(\text{IG}_{\text{NO}}) \tag{6}$$

where $\mathcal{P}(\text{X}_{\text{YES}})$ is the reliability function outgoing from a subsystem decision box, i.e., $\mathcal{R}_{\text{X}}$ model, and $\mathcal{P}(\text{X}_{\text{NO}})$ is the unreliability function or the probability of failure, i.e., the complement of the $\mathcal{R}_{\text{X}}$ model ($\overline{\mathcal{R}_{\text{X}}}$).



(a) WT Structure (b) RBD Models of WT Subsystems

Fig. 3. Wind turbine system [11]



(a) WT Complete CCD Model (b) WT Reduced CCD Model

Fig. 4. Wind turbine cause consequence analysis

3.1 Formal CCD Modeling

The CCD basic constructors *Decision box*, *Consequence path* and *Consequence box*, as described in Table 3, were formally developed, in HOL4, respectively, as [3]:

Definition 3

```
⊢ DECISION_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`, then this represents the irrelevance of the decision box, which returns the probability space p to be used in the CCD reduction process.

Secondly, the CCD *Consequence path* is defined by recursively applying the `BRANCH` ET basic constructor (see Sect. 2.2) on a given n-list of decision boxes (`DECISION_BOX`$_{\mathcal{N}}$) using the HOL4 recursive list function `FOLDL` as:

Definition 4

```
⊢ CONSEQ_PATH p (DECISION_BOX1::DECISION_BOXN)
  = FOLDL (λa b. ETREE (BRANCH a (ATOMIC b))) DECISION_BOX1 DECISION_BOXN
```

Finally, the CCD *Consequence box* is defined by mapping the function `CONSEQ_PATH` on a given *two-dimensional* list of consequence paths $\mathrm{L}_{\mathcal{M}}$ using the HOL4 mapping function `MAP`, then apply the `NODE` ET constructor:

Definition 5

```
⊢ CONSEQ_BOX p LM = ETREE (NODE (MAP (λa. CONSEQ_PATH p a) LM))
```

Using the above-mentioned CCD *generic* definitions, we can formally construct a complete CCD model (*Step 2* in Fig. 2) for the wind turbine shown in Fig. 4a, in HOL4 as:

```
⊢ Wind_Turbine_Complete_CCD R_IG R_PC =
  CONSEQ_BOX p
    [[DECISION_BOX p 1 (R_IG,¬R_IG); DECISION_BOX p 1 (R_PC,¬R_PC)];
     [DECISION_BOX p 1 (R_IG,¬R_IG); DECISION_BOX p 0 (R_PC,¬R_PC)];
     [DECISION_BOX p 0 (R_IG,¬R_IG); DECISION_BOX p 1 (R_PC,¬R_PC)];
     [DECISION_BOX p 0 (R_IG,¬R_IG); DECISION_BOX p 0 (R_PC,¬R_PC)]]
```

In cause-consequence safety analysis [19], *Step 3* in Fig. 2 is to minimize the complete CCD model in the sense that the unnecessary decision boxes should be eliminated to decrease the number of test cases and model the accurate functional behavior of systems. Upon this, the reduced CCD model that actually represents the wind turbine system, as shown in Fig. 4b, can be constructed formally by assigning `X` with neither `1` nor `0` options, for instance, `X = 2`, which represents the irrelevance of the decision box, in HOL4 as:

```
⊢ Wind_Turbine_Reduced_CCD R_IG R_PC =
  CONSEQ_BOX p
    [[DECISION_BOX p 1 (R_IG,¬R_IG); DECISION_BOX p 1 (R_PC,¬R_PC)];
     [DECISION_BOX p 1 (R_IG,¬R_IG); DECISION_BOX p 0 (R_PC,¬R_PC)];
     [DECISION_BOX p 0 (R_IG,¬R_IG); DECISION_BOX p 2 (R_PC,¬R_PC)]]
```

Also, we can formally verify the above minimal CCD model of the wind turbine system after reduction, in HOL4 as:

```
⊢ Wind_Turbine_Reduced_CCD R_IG R_PC =
  CONSEQ_BOX p
    [[DECISION_BOX p 1 (R_IG,¬R_IG); DECISION_BOX p 1 (R_PC,¬R_PC)];
     [DECISION_BOX p 1 (R_IG,¬R_IG); DECISION_BOX p 0 (R_PC,¬R_PC)];
     [DECISION_BOX p 0 (R_IG,¬R_IG)]]
```

3.2 Formal CCD Analysis

The last step in the cause-consequence analysis is to evaluate the probability of each path occurrence in the CCD model [6]. For that purpose, we propose the following novel CCD probabilistic mathematical formulations, based on RBD and ET modeling techniques, which have the capability to determine the probability of *n-level* CCD paths corresponding to n-subsystems in a critical system, where each subsystem consists of an arbitrary list of RBD events. Then, we provide the formalization of the proposed new formulas in HOL4.

One Decision Box: Figure 5 depicts a single CCD decision box associated with either a series or a parallel RBD pattern. It can be observed that the `YES BOX` of the former CCD diagram with a series RBD model is the outcome of Eq. 1 and its `NO BOX` is the complement of Eq. 1. Similarly, the `YES BOX` of the later CCD diagram with a parallel RBD model is the outcome of Eq. 2 and its `NO BOX` is the complement of Eq. 2. The probability of a consequence path for each CCD decision box assigned with a *generic* RBD model consisting of n-events, i.e., series or parallel, as shown in Fig. 5, is verified under the constraints described in Table 1 (Sect. 2.1), respectively, in HOL4 as:

Theorem 5

```
⊢ let RBD_series = rbd_struct p (series X_N)
   in prob_space p ∧ X_N ∈ events p ∧ MUTUAL_INDEP p X_N ⇒
   prob p (CONSEQ_PATH p [DECISION_BOX p J (RBD_series,COMPL p (RBD_series))])
      = if J = 1 then ∏ (PROB_LIST p X_N)
   else if J = 0 then 1 - ∏ (PROB_LIST p X_N) else 1
```

Theorem 6

```
⊢ let RBD_parallel = rbd_struct p (parallel Y_M)
   in prob_space p ∧ Y_M ∈ events p ∧ MUTUAL_INDEP p Y_M ⇒
  prob p(CONSEQ_PATH p [DECISION_BOX p K (RBD_parallel,COMPL p(RBD_parallel))])
      = if K = 1 then 1 - ∏ (PROB_LIST p (COMPL_LIST p Y_M))
   else if K = 0 then ∏ (PROB_LIST p (COMPL_LIST p Y_M)) else 1
```

where the function `COMPL` is defined to take a set X, which is the output of the RBD function `rbd_struct`, and returns the complement of the set X in the probability space p. For a complex graph of CCDs consisting of n-level decision boxes, where each decision box is associated with a series/parallel RBD model consisting of an arbitrary list of success events, we define *three* types A, B and C with all possible CCD consequence scenarios that can occur.

N Decision Boxes (Type A): The probability of n-level decision boxes assigned to a consequence path corresponding to n-subsystems of a complex system, where each decision box is associated with a *generic* RBD model consisting of an arbitrary list of k-events in a *series* connection, can be expressed mathematically for *three* cases as:

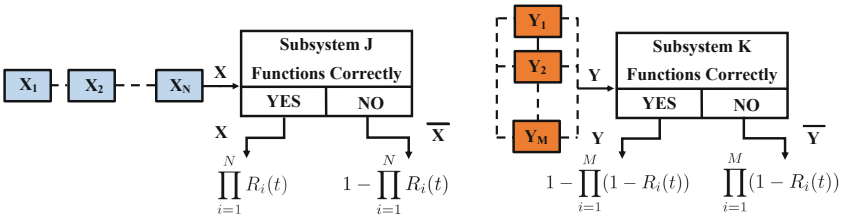


Fig. 5. CCD decision boxes with RBD connections

(A1) All outcomes of n decisions boxes are YES

$$\mathcal{R}_{A1}(t) = \prod_{i=1}^{n}\prod_{j=1}^{k}\mathcal{R}_{ij}(t) \tag{7}$$

(A2) All outcomes of n decisions boxes are NO

$$\mathcal{R}_{A2}(t) = \prod_{i=1}^{n}(1-\prod_{j=1}^{k}\mathcal{R}_{ij}(t)) \tag{8}$$

(A3) Some outcomes of m decisions boxes are YES and the rest outcomes of p decisions boxes are NO, as shown in Fig. 6a, respectively, as follows:

$$\mathcal{R}_{A3}(t) = \left(\prod_{i=1}^{m}\prod_{j=1}^{k}\mathcal{R}_{ij}(t)\right) \times \left(\prod_{i=1}^{p}(1 - \prod_{j=1}^{k}\mathcal{R}_{ij}(t))\right) \tag{9}$$

To formalize the above-proposed new cause-consequence mathematical formulations in HOL4, we formally define two *generic* functions $\mathcal{SS}^{YES}_{series}$ and $\mathcal{SS}^{NO}_{series}$ that can recursively generate the outcomes YES and NO of the RBD function `rbd_struct`, identified by the RBD basic constructor `series`, for a given arbitrary list of subsystems (SS) events, respectively as:

Definition 6

```
⊢ SS^YES_series p (SS1::SSN) =
  rbd_struct p (series (rbd_list SS1))::SS^YES_series p SSN
```

Definition 7

```
⊢ SS^NO_series p (SS::SSN) =
  COMPL p (rbd_struct p (series (rbd_list SS1)))::SS^NO_series p SSN
```

Using the above defined functions, we can verify *two-dimensional* and *scalable* CCD probabilistic properties corresponding to the proposed formulas Eq. 7, Eq. 8 and Eq. 9, respectively, in HOL4 as:

Theorem 7

```
⊢ prob_space p ∧ MUTUAL_INDEP p SSN ∧ ∀y. y ∈ SSN ⇒ y ∈ events p ∧ ⇒
  prob p (CONSEQ_PATH p (SS^YES_series p SSN)) =
         ∏ (MAP (λ a. ∏ (PROB_LIST p a)) SSN)
```

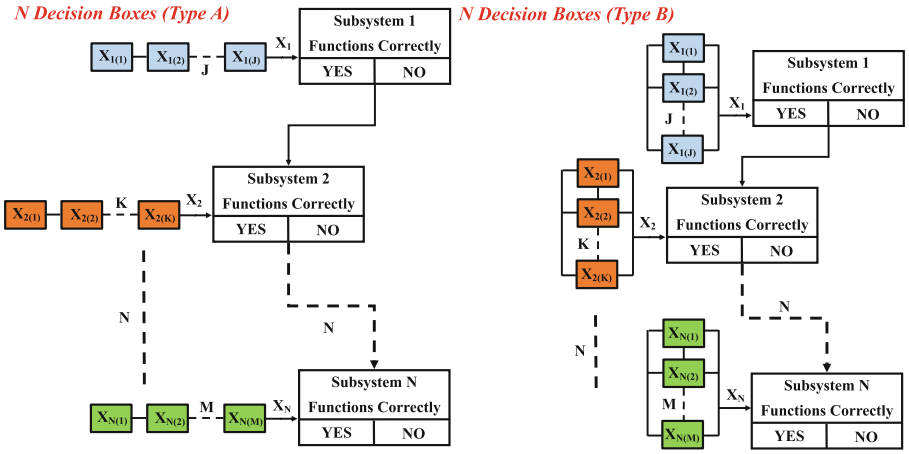
Theorem 8

```
⊢ prob_space p ∧ MUTUAL_INDEP p SSN ∧ ∀y. y ∈ SSN ⇒ y ∈ events p ∧ ⇒
  prob p (CONSEQ_PATH p (SS^NO_series p SSN)) =
         ∏ (MAP (λ b. (1 - ∏ (PROB_LIST p b))) SSN)
```

Theorem 9

```
⊢ prob_space p ∧ MUTUAL_INDEP p (SSM ++ SSP) ∧
  ∀y. y ∈ (SSM ++ SSP) ⇒ y ∈ events p ∧ ⇒
  prob p (CONSEQ_PATH p [CONSEQ_PATH p (SS^YES_series p SSM);
                         CONSEQ_PATH p (SS^NO_series p SSP)]) =
         ∏ (MAP (λ a. ∏ (PROB_LIST p a)) SSM) ×
         ∏ (MAP (λ b. (1 - ∏ (PROB_LIST p b))) SSP)
```

where the assumptions of Theorems 7–9 are similar to the ones used in Theorems 1–4 (see Sect. 2).



(a) N-level CCD Analysis of Type A (b) N-level CCD Analysis of Type B

Fig. 6. Proposed N-level decision boxes for CCD analysis

N Decision Boxes (Type B): Similarly, the probabilistic assessment of n-level decision boxes assigned to a CCD consequence path, where each decision box is associated with a *generic* RBD model consisting of k-events connected in *parallel*, can be expressed mathematically for *three* cases: (B1) All outcomes of n decisions boxes are YES; (B2) All outcomes of n decisions boxes are NO; and (B3) Some outcomes of m decisions boxes are YES and some outcomes of p decisions boxes are NO, as shown in Fig. 6b, respectively, as follows:

$$\mathcal{R}_{B1}(t) = \prod_{i=1}^{n}(1 - \prod_{j=1}^{k}(1 - \mathcal{R}_{ij}(t))) \tag{10}$$

$$\mathcal{R}_{B2}(t) = \prod_{i=1}^{n}\prod_{j=1}^{k}(1 - \mathcal{R}_{ij}(t)) \tag{11}$$

$$\mathcal{R}_{B3}(t) = \left(\prod_{i=1}^{m}(1 - \prod_{j=1}^{k}(1 - \mathcal{R}_{ij}(t)))\right) \times \left(\prod_{i=1}^{p}\prod_{j=1}^{k}(1 - \mathcal{R}_{ij}(t))\right) \tag{12}$$

To verify the correctness of the above-proposed new CCD mathematical formulas in HOL4, we define two *generic* functions $\mathcal{SS}^{YES}_{parallel}$ and $\mathcal{SS}^{NO}_{parallel}$ to recursively generate the outcomes YES and NO of the function `rbd_struct`, identified by the RBD constructor `parallel`, for a given list of subsystems events.

Definition 8

$\vdash$ $\mathcal{SS}^{YES}_{parallel}$ `p (SS1::SSN) =`
`rbd_struct p (parallel (rbd_list SS1))::`$\mathcal{SS}^{YES}_{parallel}$ `p SSN`

Definition 9

```
⊢ SS^NO_parallel p (SS::SSN) =
  COMPL p (rbd_struct p (parallel (rbd_list SS1)))::SS^NO_parallel p SSN
```

Using above defined functions, we can formally verify three *scalable* properties corresponding to Eq. 10, Eq. 11, and Eq. 12, respectively, in HOL4 as:

Theorem 10

```
⊢ prob_space p ∧ MUTUAL_INDEP p SSN ∧ ∀y. y ∈ SSN ⇒ y ∈ events p ∧ ⇒
  prob p (CONSEQ_PATH p (SS^YES_parallel p SSN)) =
         ∏ (MAP (λ a. (1 - ∏ (PROB_LIST p (compl_list p a)))) SSN)
```

Theorem 11

```
⊢ prob_space p ∧ MUTUAL_INDEP p SSN ∧ ∀y. y ∈ SSN ⇒ y ∈ events p ∧ ⇒
  prob p (CONSEQ_PATH p (SS^NO_parallel p SSN)) =
         ∏ (MAP (λ b. ∏ (PROB_LIST p (compl_list p b))) SSN)
```

Theorem 12

```
⊢ prob_space p ∧ MUTUAL_INDEP p (SSM ++ SSP) ∧
  ∀y. y ∈ (SSM ++ SSP) ⇒ y ∈ events p ∧ ⇒
  prob p (CONSEQ_PATH p [CONSEQ_PATH p (SS^YES_parallel p SSM);
                         CONSEQ_PATH p (SS^NO_parallel 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)
```

N Decision Boxes (Type C): The probabilistic assessment of n-level decision boxes assigned to a consequence path for a very complex system, where some m decision boxes are associated with *generic* RBD models consisting of k-events connected in *series*, while other p decision boxes are associated with *generic* RBD models consisting of z-events connected in *parallel*, as shown in Fig. 1b, can be expressed mathematically for *nine* cases as:
(C1) All outcomes of m and p decisions boxes are YES.

$$\mathcal{R}_{C1}(t) = \left(\prod_{i=1}^{m}\prod_{j=1}^{k}\mathcal{R}_{ij}(t)\right) \times \left(\prod_{i=1}^{p}(1-\prod_{j=1}^{z}(1-\mathcal{R}_{ij}(t)))\right) \tag{13}$$

(C2) All outcomes of m and p decisions boxes are NO.

$$\mathcal{R}_{C2}(t) = \left(\prod_{i=1}^{m}(1-\prod_{j=1}^{k}\mathcal{R}_{ij}(t))\right) \times \left(\prod_{i=1}^{p}\prod_{j=1}^{z}(1-\mathcal{R}_{ij}(t))\right) \tag{14}$$

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

$$\mathcal{R}_{C3}(t) = \left(\prod_{i=1}^{m}\prod_{j=1}^{k}\mathcal{R}_{ij}(t)\right) \times \left(\prod_{i=1}^{p}\prod_{j=1}^{z}(1-\mathcal{R}_{ij}(t))\right) \tag{15}$$

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

$$\mathcal{R}_{C4}(t) = \left(\prod_{i=1}^{m}(1 - \prod_{j=1}^{k}\mathcal{R}_{ij}(t))\right) \times \left(\prod_{i=1}^{p}(1 - \prod_{j=1}^{z}(1 - \mathcal{R}_{ij}(t)))\right) \tag{16}$$

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

$$\mathcal{R}_{C5}(t) = \left(\prod_{i=1}^{s}\prod_{j=1}^{k}\mathcal{R}_{ij}(t)\right) \times \left(\prod_{i=1}^{u}(1 - \prod_{j=1}^{k}\mathcal{R}_{ij}(t))\right) \times \left(\prod_{i=1}^{p}(1 - \prod_{j=1}^{z}(1 - \mathcal{R}_{ij}(t)))\right) \tag{17}$$

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

$$\mathcal{R}_{C6}(t) = \left(\prod_{i=1}^{s}\prod_{j=1}^{k}\mathcal{R}_{ij}(t)\right) \times \left(\prod_{i=1}^{u}(1 - \prod_{j=1}^{k}\mathcal{R}_{ij}(t))\right) \times \left(\prod_{i=1}^{p}\prod_{j=1}^{z}(1 - \mathcal{R}_{ij}(t))\right) \tag{18}$$

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

$$\mathcal{R}_{C7}(t) = \left(\prod_{i=1}^{m}\prod_{j=1}^{k}\mathcal{R}_{ij}(t)\right) \times \left(\prod_{i=1}^{u}\prod_{j=1}^{z}(1 - \mathcal{R}_{ij}(t))\right) \times \left(\prod_{i=1}^{s}(1 - \prod_{j=1}^{z}(1 - \mathcal{R}_{ij}(t)))\right) \tag{19}$$

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

$$\begin{aligned}\mathcal{R}_{C8}(t) = &\left(\prod_{i=1}^{m}(1 - \prod_{j=1}^{k}\mathcal{R}_{ij}(t))\right) \times \left(\prod_{i=1}^{u}\prod_{j=1}^{z}(1 - \mathcal{R}_{ij}(t))\right) \\ &\times \left(\prod_{i=1}^{s}(1 - \prod_{j=1}^{z}(1 - \mathcal{R}_{ij}(t)))\right)\end{aligned} \tag{20}$$

Using Theorems 5–12, we formally *verify* in HOL4 all the above-newly proposed formulas from Eq. 13 to Eq. 20 for RBD/ET-based cause consequence safety analysis (see Theorems 13–20, respectively, in [1]), which is evidence for the correctness of the proposed mathematical formulations.

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

$$\mathcal{R}_{C9}(t) = \left(\prod_{i=1}^{s}\prod_{j=1}^{k}\mathcal{R}_{ij}(t)\right) \times \left(\prod_{i=1}^{u}(1-\prod_{j=1}^{k}\mathcal{R}_{ij}(t))\right) \times \left(\prod_{i=1}^{v}(1-\prod_{j=1}^{z}(1-\mathcal{R}_{ij}(t)))\right) \times \left(\prod_{i=1}^{w}\prod_{j=1}^{z}(1-\mathcal{R}_{ij}(t))\right) \tag{21}$$

Theorem 21

```
⊢ prob p (CONSEQ_PATH p [CONSEQ_PATH p (SS^YES_series p SSs);
                         CONSEQ_PATH p (SS^NO_series p SSu);
                         CONSEQ_PATH p (SS^YES_parallel p SSv);
                         CONSEQ_PATH p (SS^NO_parallel 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)
```

A Consequence Box: Lastly, we verify a *generic* probabilistic formulation of a CCD `CONSEQ_BOX` for a certain event occurrence in the given system as the sum of all individual probabilities of all $\mathcal{M}$ CCD paths ending with that event:

Theorem 22

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

where the assumptions of the above-theorem are quite similar to those used in Theorems 3 and 4 (see Sect. 2.2). The verification of all the above-mentioned theorems was a bit challenging as we are dealing with all four types of different RBD configurations, i.e., series, the complement of series, parallel, and the complement of parallel, where each type is consisting of *generic* n-decision boxes and each decision box is associated with *generic* m-events, simultaneously in HOL4. The proof-script of the formalization work presented in this section amounts to about 5,500 lines of HOL4 code and can be downloaded from [1].

4 Conclusion

In this paper, we proposed novel formulations of cause-consequence analysis, based on RBDs and ETs dependability modeling techniques, for the safety assessment of large systems. We provided a HOL4 formalization for the proposed equations that enables the formal probabilistic assessment of scalable CCD models associated with different RBD configurations and based on any probabilistic distribution and failure rates. Moreover, the proposed RBD/ET-based CCD formalization in HOL4 solves the scalability problem of n-level CCD analysis. Our proposed new formulations provide the *first mechanical computation* of complex *n-level* cause-consequence probabilistic analysis ever, augmented with the rigor

of the HOL4 theorem prover. As future work, we plan to use the proposed CCD formalization in performing the formal RBD/ET-based cause consequence analysis of real-world complex systems, such as a smart grid or a nuclear power plant system, to verify their probabilistic expressions for all possible safety classes of consequence events at the subsystem level.

References

1. RBD/ET based Cause-Consequence Formalization in HOL4 (2021). https://github.com/hvg-concordia/CCD_RBD
2. Abdelghany, M., Ahmad, W., Tahar, S.: Event tree reliability analysis of safety critical systems using theorem proving. IEEE Syst. J. (2021). https://doi.org/10.1109/JSYST.2021.3077558
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. Ahmad, W.: Formal dependability analysis using higher-order-logic theorem proving. Ph.D. thesis, National University of Sciences & Technology, Pakistan (2017)
5. Andrews, J., Ridley, M.: Reliability of sequential systems using the cause consequence diagram method. Part E J. Process Mech. Eng. **215**(3), 207–220 (2001)
6. Andrews, J., Ridley, M.: Application of the cause-consequence diagram method to static systems. Reliab. Eng. Syst. Saf. **75**(1), 47–58 (2002)
7. Brall, A., Hagen, W., Tran, H.: Reliability block diagram modeling-comparisons of three software packages. In: Reliability and Maintainability Symposium, pp. 119–124 (2007)
8. Elderhalli, Y.: Dynamic dependability analysis using HOL theorem proving with application in multiprocessor systems. Ph.D. thesis, Concordia University, Canada (2019)
9. Güdemann, M., Ortmeier, F., Reif, W.: Using deductive cause-consequence analysis (DCCA) with SCADE. In: Saglietti, F., Oster, N. (eds.) SAFECOMP 2007. LNCS, vol. 4680, pp. 465–478. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75101-4_44
10. HOL Theorem Prover (2021). https://hol-theorem-prover.org
11. Jaiswal, S., Pahuja, G.: Effect of reliability of power converters in productivity of wind turbine. In: Conference on Power Electronics, pp. 1–6. IEEE (2014)
12. Muller, S., Deicke, M., De Doncker, R.: Doubly fed induction generator systems for wind turbines. Ind. Appl. Mag. **8**(3), 26–33 (2002)
13. Ortmeier, F., Reif, W., Schellhorn, G.: Deductive cause-consequence analysis. IFAC Proc. Vol. **38**(1), 62–67 (2005)
14. Papazoglou, I.A.: Mathematical foundations of event trees. Reliab. Eng. Syst. Saf. **61**(3), 169–183 (1998)
15. Porté-Agel, F., Bastankhah, M., Shamsoddin, S.: Wind-turbine and wind-farm flows: a review. Bound.-Layer Meteorol. **174**(1), 1–59 (2020)
16. Ridley, M.: Dependency modelling using fault-tree and cause-consequence analysis. Ph.D. thesis, Loughborough University, UK (2000)
17. Shepherd, W., Zhang, L.: Power Converter Circuits. CRC Press, Boca Raton (2004)
18. Towhidnejad, M., Wallace, D.R., Gallo, A.M.: Fault tree analysis for software design. In: NASA Goddard Software Engineering Workshop, pp. 24–29 (2002)

19. Vyzaite, G., Dunnett, S., Andrews, J.: Cause-consequence analysis of non-repairable phased missions. Reliab. Eng. Syst. Saf. **91**(4), 398–406 (2006)
20. Wadi, M., Baysal, M., Shobole, A., Tur, R.: Reliability evaluation in smart grids via modified Monte Carlo simulation method. In: International Conference on Renewable Energy Research and Applications, pp. 841–845. IEEE (2018)
21. Xin, B., Wan, L., Yu, J., Dang, W.: Basic event probability determination and risk assessment based on cause-consequence analysis method. J. Phys. **1549**, 052094 (2020)