A Formally Verified HOL4 Algebra for Event Trees

Mohamed Abdelghany, Waqar Ahmad, and Sofiène Tahar

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

{m_eldes,waqar,tahar}@ece.concordia.ca

TECHNICAL REPORT

May 2020

Abstract

Event Tree (ET) analysis is widely used as a forward deductive safety analysis technique for decision-making at the critical-system design stage. ET is a schematic diagram representing all possible operating states and external events in a system so that one of these possible scenarios can occur. In this report, we propose to use the HOL4 theorem prover for the formal modeling and step-analysis of ET diagrams. To this end, we developed a formalization of ETs in higher-order logic, which is based on a generic *list*-datatype that can: (i) construct an arbitrary level of ET diagrams; (ii) reduce the irrelevant ET branches; (iii) partition ET paths; and (iv) perform the probabilistic analysis based on the occurrence of certain events. For illustration purposes, we conduct the formal ET stepwise analysis of an electrical power grid and also determine its System Average Interruption Frequency Index ($\mathcal{SAIFI}$), which is an important indicator for system reliability.

1 Introduction

Nowadays, the fulfillment of stringent safety requirements for critical-systems, which are prevalent, e.g., in smart grids and automotive industry, has been encouraging safety design engineers to use formal techniques as per recommendations of safety standards, such as IEC 61850 [1] and ISO 26262 [2]. Therefore, it is required to build necessary formal support for rigorous reliability analysis so that they become an essential step in the design process and ensure the delivery of a trusted service without failures [3]. Several reliability modeling techniques have been developed, such as Fault Trees (FT) [4], Reliability Block Diagrams (RBD) [5] and Event Trees (ET) [6], that describe the behavior of components for a given system. FTs mainly provide a graphical model for analyzing the factors causing a system failure upon their occurrences. On the other hand, RBDs allow us to model the success relationships of complex systems. ETs enumerate all possible operating states and external events in a system in the form of a tree structure represented by an *initiating node* and *branches* [6]. The results of the ET analysis are extremely useful for safety analysts as it provide a more detailed system view compared to FTs and RBDs.

Papazoglou [6] was the first researcher to lay down the mathematical foundations of ET in the late 90s. He described the ET analysis in *four* main steps: (1) *Generation*: construct a complete ET model; (2) *Reduction*: removal of unnecessary ET branches; (3) *Partitioning*: extract a collection of ET paths according to the system failure and success events; and lastly (4) *Probabilistic analysis*: evaluate the probabilities of ET paths based on the occurrence of a certain event. But the analysis of ET proposed in [6] is done purely manually using a paper-and-pencil approach. On the other hand, there exist several commercial tools based on Monte-Carlo Simulation for ET analysis, such as ITEM [7], Isograph [8], and EC Tree [9], which have been widely used to determine sequentially failure and success scenarios of real-world systems, like electrical power grids [10], nuclear power plants [11] and railways [12]. Prior to utilizing these tools for ET analysis, the users must draw a given system actual ET diagram manually, maybe on paper. Both of these approaches may introduce inaccuracies in the ET analysis due to human infallibility and analysis approximations caused by the numerical methods in the simulation tools, respectively. A more efficient and practical way is to define functions describing the pattern of modeling ETs as well as ET probabilistic properties.

In this report, we propose to use HOL theorem proving [13], which provides us the ability to accurately model and also rigorously verify the essential ET properties. For this purpose, we endeavor to formalize the four steps of ET analysis using the HOL4 proof assistant, i.e., generation, reduction, partitioning and probabilistic analysis, as described by Papazoglou [6]. We present two syntactically different, but semantically equivalent formalizations for ET analysis, using *set* and *list*-datatypes, respectively. The former *set*-datatype ET formalization is described by Papazoglou, however, it cannot mimic the graphical model of an ET consisting of an initiating node and branches since the elements in sets are orderless. The ordering is important in *Steps* 3 and 4 (*reduction* and *partitioning* processes) of the ET analysis. In the latter

3

approach, the *list*-datatype inherently preserves the index of its member elements and naturally captures the graphical structure of ETs. Also, from our experience, the reasoning about ET reduction and partitioning properties using the *set*-datatype is quite cumbersome and significantly slow compared to the *list*-datatype especially when the ET diagram becomes tremendously large. Therefore, we use the *list*-datatype to formalize all *four* steps of ET analysis in HOL4. For that purpose, we propose to use the *list*-datatype that inherently preserves the index of its member elements and naturally captures the graphical structure of ETs. For illustration purposes, we conduct the formal ET analysis of a practical power grid system consisting of $\mathcal{N}$ transmission lines and $\mathcal{M}$ customers. Subsequently, we also formally determine the System Average Interruption Frequency Index ($\mathcal{SAIFI}$) [14], which is an important reliability index describing the average frequency of interruptions in an electrical power systems.

The rest of the report is organized as follows: In Section 2, we present the related literature review. Section 3 briefly summarize the fundamentals of ETs. In Section 4, we present the details of our HOL4 formalization of ETs using the *set*-datatype. Section 5 describes the formalization of ETs by developing a new recursive datatype `EVENT_TREE`. In Section 7, we present the formalization of ETs reduction and partitioning. Section 6 describes the formal probabilistic analysis of ETs. In Section 8, we present the formal ET analysis-based of a power grid system and the assessment of its reliability index $\mathcal{SAIFI}$. Lastly, Section 9 concludes the report.

2 Related Work

Only a few work have previously considered using formal methods to model and analyze ETs. For instance, Nỳvlt et al. in [15] used Petri nets to model the cascading failure of sub-systems and their effect on the entire system using the standard FT and ET modeling techniques. The authors proposed a new method based on P-invariants to obtain a model of cascading dependencies in ETs [15]. However, according to the authors, they are not able to obtain verified equations from that model [15]. HOL4 [13] has been previously used by Ahmad et al. in [16] to formalize FTs and RBDs. The FT formalization includes a new datatype consisting of `AND`, `OR` and `NOT` FT constructors [4] to analyze the factors causing a system failure. Similarly, Ahmad et al. in [17] defined a new `RBD` datatype to model and analyze the success relationships of a system using different RBD configurations [5], such as series, parallel and combination of series and parallel. However, both of these formalizations are limited to analyzing either a system failure or its success only. On the other hand, ETs have the superior ability to analyze both failure and success scenarios in a system. For the formalization of ET in HOL4, the existing `treeTheory` in the standard library of HOL4 only allows drawing a specific tree with leaves and nodes manually. To the best of our knowledge, this is the first work, which develops a formal modeling and step-analysis of ETs using HOL4 theorem prover.

3 Event Trees

An ET diagram is a graphical model that enumerate all possible combinations of component states and external events in a system in the form of a tree structure. ETs utilize the forward logic [18] starting by an Initiating Event (IE) called *node* and then all possible scenarios of an event are drawn as *branches*. For instance, consider a system consist of three components C_1, C_2 and C_3, each has two operational states, i.e., operating or failing. The ET *four* step-analysis defined by Papazoglou [6] are as follows:

1. *Generation*: Construct a complete ET diagram that draws all possible scenarios, which is well-known as *paths*. Each *path* consists of a unique sequence of events. Fig. 1 depicts 8 paths (0-7) with all possible scenarios that can occur.

2. *Reduction*: Model the accurate functional behavior of the system in the sense that the irrelevant branches should be removed from a complete ET. This can be done by deleting some specific branches corresponding to the occurrence of certain events, which are known as *Complete Cylinders* (CCs) [19]. These cylinders are ET *paths* consisting of $\mathcal{N}$ events and they are conditional on the occurrence of $\mathcal{K}$ *Conditional Events* (CEs) in their respective paths and they are referred to as CCs with respect to $\mathcal{K}$ [19]. For instance, if the critical-component C_1 fails then the whole system fails regardless of the status of the rest of the components, i.e., C_2 and C_3, as shown in Fig. 1. Therefore, paths 4-7 are CCs with respect to C_{1F}.

3. *Partitioning*: partition of an ET diagram is essential as we are only interested in the occurrence of certain events according to the system failure and success events. For instance, suppose we are only focusing on the failure of the system in Fig. 1, then ET paths 3 and 4 are obtained from the reduced ET.

4. *Probabilistic analysis*: Lastly, evaluate the probabilities of ET paths based on the occurrence of a certain event. These probabilities represent the likelihood of each scenario that is possible to occur in a system so that *only one* can occur [6]. This implies that all paths connected to a node are disjoint (mutually exclusive) [6]. Assuming that all events in an ET are mutually independent that the probability of any ET path can be computed by simply multiplying the individual probabilities of all the events associated with it [6]. For example, the probability of the system failure in Fig. 1, i.e., paths 3 and 4, is expressed mathematically as:

$$\mathcal{P}(System_{Failure}) = \mathcal{P}(C_{1S}) \times \mathcal{P}(C_{2F}) \times \mathcal{P}(C_{3F}) + \mathcal{P}(C_{1F}) \tag{1}$$

where $\mathcal{P}(C_{XF})$ is the unreliability function or the probability of failure for a component X and $\mathcal{P}(C_{XS})$ is the reliability function or the probability of operating.

In the next sections, we present, in detail, the formalization of ETs using the *set* and the *list* data-types, respectively. The reason for using the *set* theory is that most of the mathematical foundations of ETs from the work of Papazoglou [6] are built on sets. However, ordering of events in ET paths is important during *Steps 2* and 3 of the ET analysis. Therefore, a sequence-preserving formalization of ETs in the *list* theory should be adopted. In order to ensure the correspondence of the *set* and *list* theory based ET formalizations, we formally verify the equivalence between them.
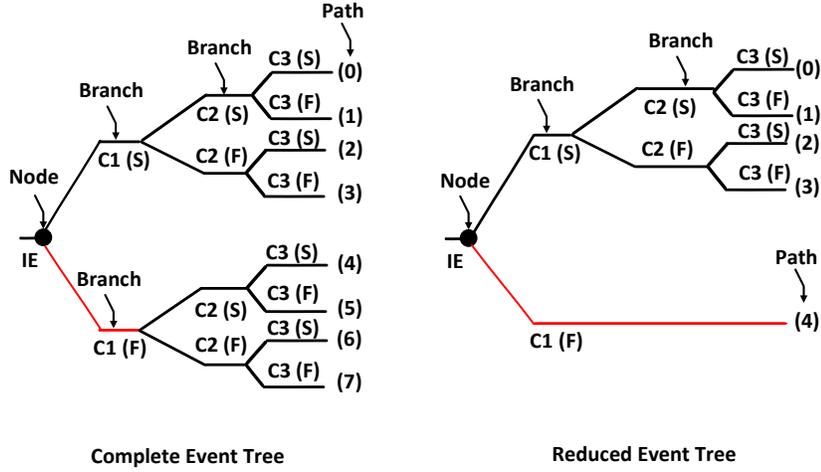
Figure 1: A sample ET diagrams / S (Success state) / F (Failure state)

4   ET Formalization using Sets

An event outcome space ($\mathcal{W}$) is referred to a set of all possible scenarios of an IE or modes of operation of a system critical-component, which must satisfy the following constraints according to Papazoglou [6]:

a) *Distinct*: All outcomes in an event outcome space must be unique.

b) *Disjoint (mutually exclusive)*: Any pair of events from a set of events outcome space cannot occur at the same time.

c) *Complete*: An event outcome space must contain all possible events that can occur.

d) *Finite*: An event outcome space must consist of a finite number of elements.

$$\mathcal{W} = \{\omega_j\} \qquad j = 1, 2, \ldots, \mathcal{N} \tag{2}$$

We formalize the above-mentioned event outcome space ($\mathcal{W}$) constraints in HOL4 as follows:

type_abbrev (``event'', ``:(α -> bool)'')

Definition 1:
⊢ Ω $\mathcal{W}$ = {x | x ∈ $\mathcal{W}$ ∧ ∅ ∈ $\mathcal{W}$ ∧ disjoint $\mathcal{W}$ ∧ FINITE $\mathcal{W}$}

where $\mathcal{W}$ is a set of *events* representing the possibilities resulting from an IE or modes of operation of a system component in HOL4. The elements in a set are intrinsically distinct and thus ensuring the constraint (a). The function disjoint ensures that each pair of elements in a given set is mutually exclusive satisfying constraint (b). The completeness of an event outcome space, constraint (c), means containing all possible events or modes of a system component that can occur. In many practical systems, some components are redundant for improving system reliability and they are only used when required, i.e., in a hold state meaning neither success nor fail. This completeness of the event outcome space

can be ensured by adding an empty set $\varnothing$ representing the default (not in-use) case, i.e., a component is neither in success nor in failure state. The HOL4 function `FINITE` guarantees that the set of event outcome space must consist of a finite number of elements, as indicated by constraint (d).

Consider a system having two events, say E_1 and E_2, with two event outcome spaces $\mathcal{W}_1$ and $\mathcal{W}_2$, respectively. The Cartesian product ($\bigotimes$) of these event outcome spaces returns a set of ($\mathcal{N}_1 \times \mathcal{N}_2$) pairs containing all possible outcome pairs for the occurrence of E_1 and E_2 together (i.e., $\mathcal{W}_1 \bigotimes \mathcal{W}_2$). In ET, an intersection operation is performed on each member of these pairs to obtain a valid event outcome space. In other words, the resulting event outcome space from the Cartesian product of two event outcome spaces also satisfies the above-mentioned constraints. We formalize this concept in HOL4 as follows:

We first construct a set by taking each element from the event outcome spaces $\mathcal{W}_1$ and $\mathcal{W}_2$ and then performing an intersection operation on these elements as:

Definition 2:
$\vdash \mathcal{W}_1 \bigcap^{\otimes} \mathcal{W}_2 = \{x \cap y \mid x \in \Omega\ \mathcal{W}_1 \wedge y \in \Omega\ \mathcal{W}_2\}$

Next, we ensure that the obtained duets from Definition 2 are mutually exclusive. For instance, consider two arbitrary outcomes ($\omega_{1m} \cap \omega_{2n}$) and ($\omega_{1k} \cap \omega_{2l}$) at least ($m \neq k$) or ($n \neq l$) must be true.

Definition 3:
$\vdash \mathcal{W}_1 \bigotimes \mathcal{W}_2 = \{x \mid x \in \mathcal{W}_1 \bigcap^{\otimes} \mathcal{W}_2 \wedge \texttt{disjoint}\ (\mathcal{W}_1 \bigcap^{\otimes} \mathcal{W}_2)\}$

To ensure the validity of an event outcome space, as described in Eq. 2, we define a *predicate* function in HOL4 as follows:

Definition 4:
$\vdash \Omega^{\vDash} \mathcal{W} \Leftrightarrow \varnothing \in \mathcal{W} \wedge \texttt{disjoint}\ \mathcal{W} \wedge \texttt{FINITE}\ \mathcal{W}$

Using the above definitions, we formally verify that the function $\bigotimes$ forms a valid event outcome space.

Theorem 1: (Cartesian product fulfilling the event outcome space constraints)
$\vdash \Omega^{\vDash} \mathcal{W}_1 \wedge \Omega^{\vDash} \mathcal{W}_2 \Rightarrow \Omega^{\vDash} (\mathcal{W}_1 \bigotimes \mathcal{W}_2)$

Now, we can define a generic function as defined by Papazoglou [6] that can take an arbitrary set of event outcome spaces and generate the corresponding ET diagram (i.e., $\mathcal{W}_1 \bigotimes \mathcal{W}_2 \bigotimes \cdots \bigotimes \mathcal{W}_{\mathcal{N}}$). For this purpose, we use the HOL4 function `ITSET` that can recursively apply $\bigotimes$ on a given set of event outcome spaces as follows:

Definition 5:
$\vdash \mathcal{S} \bigotimes^{\mathcal{N}} \mathcal{W}_{\mathcal{N}} = \texttt{ITSET}\ (\lambda \mathcal{W}_1\ \mathcal{W}_2.\ \ \mathcal{W}_1 \bigotimes \mathcal{W}_2)\ \mathcal{S}\ \mathcal{W}_{\mathcal{N}}$

where $\mathcal{S}$ is a *set* containing all event outcome spaces till $\mathcal{N}-1$ (i.e., $\mathcal{S} = \{\mathcal{W}_1; \mathcal{W}_2; \ldots; \mathcal{W}_{\mathcal{N}-1}\}$) and $\mathcal{W}_{\mathcal{N}}$ represents the last event outcome space. In order to reason about essential properties of above-mentioned ET model, we formally verify the following properties, by utilizing the properties of the HOL4 function `ITSET`, on a given set of event outcome spaces as:

7

Theorem 2:
$\vdash (\mathcal{W}_1 \text{ INSERT } \mathcal{S}) \bigotimes^{\mathcal{N}} \mathcal{W}_{\mathcal{N}} = (\mathcal{S} \text{ DELETE } \mathcal{W}_1) \bigotimes^{\mathcal{N}} (\mathcal{W}_1 \bigotimes \mathcal{W}_{\mathcal{N}})$

Theorem 3:
$\vdash (\mathcal{W}_1 \text{ INSERT } \mathcal{S}) \bigotimes^{\mathcal{N}} \mathcal{W}_{\mathcal{N}} = \mathcal{W}_1 \bigotimes ((\mathcal{S} \text{ DELETE } \mathcal{W}_1) \bigotimes^{\mathcal{N}} \mathcal{W}_{\mathcal{N}})$

The order of events in a path is irrelevant when evaluating the probabilities of a given path [19], i.e., the probability of path (C_{1S}, C_{2F}, C_{3F}) in Eq. 1 is exactly equivalent to the probability of path (C_{3F}, C_{2F}, C_{1S}) due to the commutative property of intersection and the events independence. However, it is important to preserve the order of events in ET paths during *Steps* 2 and 3 (*reduction* and *partitioning*) of the ET analysis [6] while the elements in sets are orderless. A possible way to resolve the problem of ordering in the *set*-datatype is by assigning a unique number to each set element representing a branch during the ET modeling. However, when the ET diagram becomes tremendously large, the set-based reasoning is quite cumbersome and significantly slow compared to the *list*-datatype. For that purpose, in the next three sections, we present the formalization of all *four* ET analysis steps using the *list* datatype, which inherently preserves the order of elements.

5 ET Formalization using Lists

We start the formalization of ETs by developing a new recursive datatype `EVENT_TREE` in HOL4 as follows:

Hol_datatype EVENT_TREE =   ATOMIC of (α event)
                             | NODE of EVENT_TREE list
                             | BRANCH of (α event) $\Rightarrow$ 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 6:
$\vdash$ ETREE (ATOMIC X) = X $\wedge$
  ETREE (NODE (h::t)) = ETREE h $\cup$ (ETREE (NODE t)) $\wedge$
  ETREE (BRANCH X (h::t)) = X $\cap$ (ETREE h $\cup$ ETREE (BRANCH X t))

The function `ETREE` takes a set `X`, identified by a type constructor `ATOMIC` and returns the given set `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 given list. Similarly, if the function `ETREE` takes a set `X` and a list of type `EVENT_TREE`, identified by a type constructor `BRANCH`, then it performs the intersection of the set `X` with the union of the head of the given list after applying the function `ETREE` and the recursive call of the `BRANCH` constructor.

To formally define a function that can model a complete ET for $\mathcal{N}$ lists, similar to Definition 5, we start by defining a function that can model an ET for two lists, say L_1 and L_2, in HOL4 as:

Definition 7:
$\vdash$ (h::t) $\bigotimes_{\text{L}}$ L$_2$ = BRANCH h L$_2$::t $\bigotimes_{\text{L}}$ L$_2$

where the function $\bigotimes_{\mathrm{L}}$ takes two different `EVENT_TREE`-typed lists and returns an `EVENT_TREE`-typed list by recursively applying the `BRANCH` constructor on each element of the first list paired with the entire second list.

Now, we can define a generic function that takes an arbitrary list of event outcome spaces and generates a corresponding complete ET diagram, i.e., *Step 1 (Generation)* of ET analysis [6]. For this purpose, we utilize the HOL4 function `FOLDR` that recursively applies $\bigotimes_{\mathrm{L}}$ on a given list of event outcome spaces as:

Definition 8:
$\vdash$ L $\bigotimes_{\mathrm{L}}^{\mathcal{N}}$ L$_{\mathcal{N}}$ = FOLDR (λL$_1$ L$_2$. L$_1$ $\bigotimes_{\mathrm{L}}$ L$_2$) L$_{\mathcal{N}}$ L

where L is a *list* of all event outcome spaces till $\mathcal{N}-1$ (i.e., L = $[[\mathcal{W}_1]; [\mathcal{W}_2];\ldots; [\mathcal{W}_{\mathcal{N}-1}]]$) and L$_{\mathcal{N}}$ = $[\mathcal{W}_{\mathcal{N}}]$.

In order to ensure the correspondence of the *list* and *set* theory based ET formalizations, we formally verify the equivalence between Definitions 3 and 7 and Definitions 5 and 8, in HOL4 as:

Theorem 4:
$\vdash$ $\Omega_{\mathrm{L}}^{\models}$ [L$_1$;L$_2$] $\Rightarrow$ ETREE (NODE (L$_1$ $\bigotimes_{\mathrm{L}}$ L$_2$)) = $\bigcup$ ((set L$_1$) $\bigotimes$ (set L$_2$))

Theorem 5:
$\vdash$ $\Omega_{\mathrm{L}}^{\models}$ (L$_{\mathcal{N}}$::L) $\Rightarrow$ ETREE (NODE (L $\bigotimes_{\mathrm{L}}^{\mathcal{N}}$ L$_{\mathcal{N}}$)) = $\bigcup$ ((set L) $\bigotimes^{\mathcal{N}}$ (set L$_{\mathcal{N}}$))

where the predicate function $\Omega_{\mathrm{L}}^{\models}$ covers all constraints of event outcome spaces (*distinct, disjoint, complete* and *finite*) on the given lists.

6 ET Reduction and Partitioning Formalization

In ET analysis [6], *Step 2 (Reduction)* is used to model the accurate functional behavior of systems in the sense that the irrelevant branches should be removed from a complete ET of a system. To perform the reduction process, we first need to extract all possible paths from a given ET and then apply the deletion operation. For this purpose, we define the following functions in HOL4:

Definition 9:
$\vdash$ L $\bigotimes_{\mathrm{paths}}^{\mathcal{N}}$ L$_{\mathcal{N}}$ = FOLDR (λL$_1$ L$_2$. L$_1$ $\bigotimes_{\mathrm{paths}}$ L$_2$) L$_{\mathcal{N}}$ L

where the function $\bigotimes_{\mathrm{paths}}$ takes two different lists and returns a list containing all possible ET paths in a list. To ensure consistency, we also formally verify the equivalence between Definitions 8 and 9, i.e., complete ET paths, in HOL4 as:

Lemma 1:
$\vdash$ ETREE (NODE (L $\bigotimes_{\mathrm{L}}^{\mathcal{N}}$ (EVENT_TREE_LIST L$_{\mathcal{N}}$))) =
  ETREE (NODE (EVENT_TREE_LIST (L $\bigotimes_{\mathrm{paths}}^{\mathcal{N}}$ L$_{\mathcal{N}}$)))

where the function `EVENT_TREE_LIST` is used to type-cast the normal list to `EVEN_TREE`-typed list.

Now, we define a reduction function $\boxtimes$ in HOL4 on event outcome spaces that takes a list L, which is the output of Definition 9, a list of ET path numbers N to be reduced and their K conditional events CE and returns a reduced ET list as:

Definition 10:
⊢ L ⊠ N CE p =
  LUPDATE (PATH p CE) (LAST N) (DELETE_N L (TAKE (LENGTH N-1) N))

where the functions LUPDATE, LAST, and TAKE are the HOL4 *list* theory functions to update an element, extract the last element and take a collection of elements, respectively. The function PATH takes a list of events from a probability space p and extracts an intersection between the elements of the list. The function DELETE_N recursively deletes N elements from a given list corresponding to the branches that should be removed from a complete ET of a system in order to model the accurate functional behavior of systems. To ensure that the reduced ET is consistent, we formally verify the following reduction properties:

We first ensure that the length of ET after reduction is equal to the length of complete ET minus the number of paths that were deleted, in HOL4 as:

Lemma 2:
⊢ (INDEX_LT_LEN N (L $\bigotimes_{\text{paths}}^{\mathcal{N}}$ L$_{\mathcal{N}}$)) ∧ (LENGTH N ⩾ 1) ⇒
  LENGTH ((L $\bigotimes_{\text{paths}}^{\mathcal{N}}$ L$_{\mathcal{N}}$) ⊠ N CE p) = LENGTH (L $\bigotimes_{\text{paths}}^{\mathcal{N}}$ L$_{\mathcal{N}}$) - LENGTH N + 1

where the function INDEX_LT_LEN ensure that each index in the given list N is less than the length of the reduced ET list, respectively.

Next, we ensure that the paths that were not reduced still exist in the reduced ET, in HOL4 as:

Lemma 3:
⊢ (∀x. x ∈ N ⇒ i < x) ∧ (SORTED (λa b. a > b) N) ∧ (LENGTH N ⩾ 1) ∧
  (INDEX_LT_LEN N (L $\bigotimes_{\text{paths}}^{\mathcal{N}}$ L$_{\mathcal{N}}$)) ∧ (i ≠ LAST N) ⇒
  EL i ((L $\bigotimes_{\text{paths}}^{\mathcal{N}}$ L$_{\mathcal{N}}$) ⊠ N CE p) = EL i (L $\bigotimes_{\text{paths}}^{\mathcal{N}}$ L$_{\mathcal{N}}$)

where the function EL, from the *list* theory, extracts a specific element from a list. The function SORTED ensure that the index list N is sorted in descending order.

To perform multiple reduction operations on a given ET model, we define the following recursive function, using Definition 10, in HOL4 as:

Definition 11:
⊢ L ⊠$^{\mathcal{N}}$ (N::Ns) (CE::CEs) p = (L ⊠ N CE p) ⊠$^{\mathcal{N}}$ Ns CEs p

After the ET reduction process, the next step is the partitioning of the reduced ET paths space according to the system failure and success events [6]. Since the output of the function ⊠$^{\mathcal{N}}$ is a list, we can define a partitioning function ⊞ to extract a collection of ET paths specified in the index list N, in HOL4 as:

Definition 12:
⊢ N ⊞ L = MAP (λa. EL a L) N

To ensure the correctness of the above function, we formally verify the following commutative property with the functions ⊞ and REVERSE, using Definitions 11 and 12, in HOL4 as:

Lemma 4:
$\vdash$ (REVERSE M) $\boxplus$ (L $\bigotimes_{\text{paths}}^{\mathcal{N}}$ L$_{\mathcal{N}}$) $\boxtimes^{\mathcal{N}}$ Ns CEs p) =

  REVERSE (M $\boxplus$ (L $\bigotimes_{\text{paths}}^{\mathcal{N}}$ L$_{\mathcal{N}}$) $\boxtimes^{\mathcal{N}}$ Ns CEs p)

where the HOL4 function REVERSE returns a list in reverse order.

7   ET Probabilistic Analysis Formalization

The last step in the ET analysis [19] is to determine the probability of each path occurrence in the whole ET diagram. For that purpose, we formally verify *generic* probabilistic properties for NODE, BRANCH, PATH and $\bigotimes_{\text{L}}^{\mathcal{N}}$ as follows:

Theorem 6:
$\vdash$ prob_space p $\wedge$ $\Omega_{\text{L}}^{\vDash}$ L $\wedge$ ($\forall$y. y $\in$ L $\Rightarrow$ y $\in$ events p) $\Rightarrow$
  prob p (ETREE (NODE L)) = $\sum_{\mathcal{P}}$ p L

The first assumption in the above theorem ensures that p is a valid probability space. The next assumption is quite similar to the one described in Theorem 4. The last assumption ensures that all component states list belongs to the events space. The function $\sum_{\mathcal{P}}$ is defined to sum the probabilities of events for a given list.

Similarly, the probability of events in branches is the multiplication of each branch event probability with the sum of the probabilities for the next events. This can be verified in HOL4 as:

Theorem 7:
$\vdash$ prob_space p $\wedge$ $\Omega_{\text{L}}^{\vDash}$ L $\wedge$ MUTUAL_INDEP p (X::L) $\wedge$
  ($\forall$y. y $\in$ (X::L) $\Rightarrow$ y $\in$ events p) $\Rightarrow$
  prob p (ETREE (BRANCH X L)) = (prob p X) $\times$ $\sum_{\mathcal{P}}$ p L

where the predicate function MUTUAL_INDEP ensures that all events in each path of an ET are mutually independent.

Also, the probability of an ET path can be verified as the multiplication of the individual probabilities of all the events associated with it, in HOL4 as:

Theorem 8:
$\vdash$ prob_space p $\wedge$ MUTUAL_INDEP p L $\wedge$ ($\forall$y. y $\in$ L $\Rightarrow$ y $\in$ events p) $\Rightarrow$
  prob p (PATH p L) = $\prod$ (PROB_LIST p L)

where the function $\prod$ takes a list and extracts the multiplication of the list elements while the function PROB_LIST returns a list of probabilities associated with the elements of the list.

Additionally, we can formally verify a generic probabilistic property for the function $\bigotimes_{\text{L}}^{\mathcal{N}}$, in HOL4 as:

Theorem 9:
$\vdash$ prob_space p $\wedge$ $\Omega_{\text{L}}^{\vDash}$ (L$_{\mathcal{N}}$::L) $\wedge$ MUTUAL_INDEP p (L$_{\mathcal{N}}$::L) $\wedge$
  ($\forall$y. y $\in$ (L$_{\mathcal{N}}$::L) $\Rightarrow$ y $\in$ events p) $\Rightarrow$
  prob p (ETREE (NODE (L $\bigotimes_{\text{L}}^{\mathcal{N}}$ L$_{\mathcal{N}}$))) = $\prod$ ($\sum_{\mathcal{P}}^{\text{2D}}$ p (L$_{\mathcal{N}}$::L))

where the function $\sum_{\mathcal{P}}^{2D}$ is used to recursively apply the function $\sum_{\mathcal{P}}$ on a given two dimensional list.

Using the above theorems, we can formally verify in HOL4 that the probability of the function $\bigotimes_{L}^{\mathcal{N}}$ is equal to 1, which returns a complete space of failure and success events:

Theorem 10:
⊢ `prob_space p` ∧ `MUTUAL_INDEP p` (↑↓ ($L_{\mathcal{N}}$::L)) ∧ $\Omega_{L}^{\models}$ (↑↓ ($L_{\mathcal{N}}$::L)) ∧
  (∀y. y ∈ (↑↓ ($L_{\mathcal{N}}$::L)) ⇒ y ∈ `events p`) ⇒
  `prob p` (`ETREE` (`NODE` ((↑↓ L) $\bigotimes_{L}^{\mathcal{N}}$ (↑↓ [$L_{\mathcal{N}}$])))) = 1

where the function ↑↓ takes a list of components and assigns both ↑ and ↓ events to each component in the given list representing operating and failing events, respectively.

The prime purpose of the above-mentioned formalization of ETs is to build a reasoning support for the formal analysis of reliability aspects of real-world safety-critical systems within the sound environment of HOL4. In the next section, we present the formal ET analysis of an electrical power grid and determine its reliability index to illustrate the applicability of our proposed approach.

8   Electrical Power Grid System

An electrical power grid is an interconnected network for delivering electricity from producers to consumers. The power grid system [20] mainly consists of: (i) generating stations that produce electric power; (ii) electrical substations for stepping voltage up for transmission or down for distribution; (iii) high voltage transmission lines that carry power from distant sources to demand-centers; and (iv) distribution lines that connect individual customers. With respect to the power-outage-causes study domain, the majority of the outages in the power grid are the result of events that occur on the grid transmission and distribution sides [21]. Therefore, a rigorous formal ET step-analysis of the power grid is essential in order to reduce the risk situation of a blackout and back-up decisions to be taken. Using our proposed ET formalization, we can model the ET for any power grid consisting of $\mathcal{N}$ transmission lines and $\mathcal{M}$ customers. Also, we can determine the System Average Interruption Frequency Index ($\mathcal{SAIFI}$), which is used by design engineers to indicate the average frequency of customers experience a sustained outage. $\mathcal{SAIFI}$ is defined as the total number of customer interruptions over the total number of customers served [22]:

$$\mathcal{SAIFI} = \frac{\sum_{\mathcal{P}(\mathcal{X}_{Fail}) \times \text{CN}_{\mathcal{X}}}}{\sum \text{CN}_{\mathcal{X}}} \qquad (3)$$

where $\text{CN}_{\mathcal{X}}$ is the number of customers for a certain location $\mathcal{X}$. We define a generic function $\mathcal{SAIFI}$ in HOL4 by dividing the sum of multiplying the probabilities of a collection of ET paths after reduction with the number of customers that are affected by them over the total number of customers served as:

Definition 13:
⊢ $\sum_{F}$ $L_{\mathcal{N}}$ L Ns CEs (E::Es) (CN::CNs) p =
(λa b. `prob p` (`ETREE` (`NODE` (a ⊞ ((L $\bigotimes_{paths}^{\mathcal{N}}$ $L_{\mathcal{N}}$) ⊠$^{\mathcal{N}}$ Ns CEs)))) × b) E CN
+ $\sum_{F}$ $L_{\mathcal{N}}$ L Ns CEs Es CNs p

12

Definition 14:

$\vdash \mathcal{SAIFI}$ L$_\mathcal{N}$ L Ns CEs Es CNs p = ($\sum_F$ L$_\mathcal{N}$ L Ns CEs Es CNs p) / $\sum$ CNs

where

L   : list of transmission lines (TL) modes;   L$_\mathcal{N}$: Last TL modes;
Ns : list of complete cylinders;   Es : list of events partitioning paths;
CEs : list of conditional events;   CNs : list of customer numbers

For instance, consider a power grid consisting of three main transmission lines (M), two lateral transmission lines (L), two generators (G), three (two step-up and one step-down) substations (S/S) and three different loads A, B and C with the number of customers served X, Y and Z, respectively, as shown in Fig. 2. Assume that each TL (M/L) has two operational states, i.e., operating or failing. Using our ET formalization, we can formally verify the complete ET model (32 paths) for the 5 TLs that mainly affect the reliability of the power grid, in HOL4 as:

Lemma 5:
```
⊢ ETREE (NODE (↑↓ [M1; M2; M3; L1]) ⊗_L^𝒩 (↑↓ [L2])) =
  ETREE (NODE
   [BRANCH (M1 ↑) [BRANCH (M2 ↑) [BRANCH (M3 ↑)...; BRANCH (M3 ↓)...];
                   BRANCH (M2 ↓) [BRANCH (M3 ↑)...; BRANCH (M3 ↓)...]];
    BRANCH (M1 ↓) [BRANCH (M2 ↑) [BRANCH (M3 ↑)...; BRANCH (M3 ↓)...];
                   BRANCH (M2 ↓) [BRANCH (M3 ↑)...; BRANCH (M3 ↓)...]]])
```

The complete ET, obtained above, can be reduced from 32 paths (0-31) to 14 paths (0-13), in the sense that the irrelevant nodes and branches are removed to model the exact logical behavior of the power grid. For instance, the paths from 31 to 24, where both M1 and M2 fail, then the likelihood of occurrence of these paths is equal to the probabilities of M1 and M2 failures only regardless of the status of other TLs. We formally verify the following reduction property to obtain the actual ET of TLs, as shown in Fig. 3, in HOL4 as:
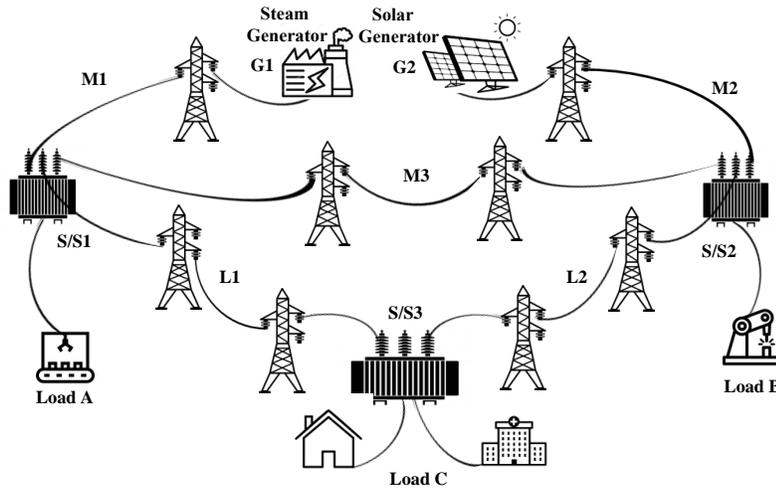

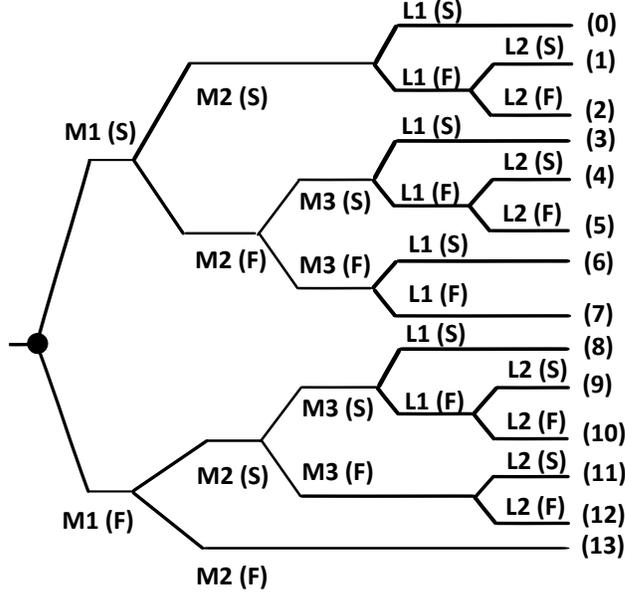
Figure 2: Electrical power grid

Figure 3: Reduced ET of the electrical power grid

Lemma 6:
```
⊢ ETREE (NODE (EVENT_TREE_LIST ((↑↓ [M1; M2; M3; L1]) ⊗ᴺₚₐₜₕₛ (↑↓ [L2]))
              ⊠ᴺ [[31;30;29;28;27;26;25;24];...]  [[M1 ↓; M2 ↓];...])) =
  ETREE (NODE
    [BRANCH (M1 ↑)
        [BRANCH (M2 ↑) [L1 ↑; BRANCH (L1 ↓) [L2 ↑; L2 ↓]];
         BRANCH (M2 ↓) [BRANCH (M3 ↑) [L1 ↑; BRANCH (L1 ↓) [L2 ↑; L2 ↓]];
                        BRANCH (M3 ↓) [L1 ↑; L1 ↓]]];
     BRANCH (M1 ↓)
        [BRANCH (M2 ↑) [BRANCH (M3 ↑) [L1 ↑; BRANCH (L1 ↓) [L2 ↑; L2 ↓]];
                        BRANCH (M3 ↓) [L2 ↑; L2 ↓]]; M2 ↓]])
```

Typically, we are only interested in the occurrence of certain events in ET that affect certain paths. For instance, if we consider the failure of load A, then paths 11, 12 and 13 are obtained. Similarly, a different set of paths can be obtained by observing different failures in the power grid as: (i) $\mathcal{P}(A_{Fail}) = \sum_{\mathcal{P}(paths11,12,13)}$; (ii) $\mathcal{P}(B_{Fail}) = \sum_{\mathcal{P}(paths6,7,13)}$; and (iii) $\mathcal{P}(C_{Fail}) = \sum_{\mathcal{P}(paths2,5,7,10,12,13)}$.

Therefore, the assessment of $\mathcal{SAIFI}$ can be done informally as:

$$\mathcal{SAIFI} = \frac{\mathcal{P}(A_{Fail}) \times X + \mathcal{P}(B_{Fail}) \times Y + \mathcal{P}(C_{Fail}) \times Z}{X + Y + Z} \tag{4}$$

In this work, we assumed that the failure and success states of each TL is exponentially distributed. This can be formalized in HOL4 as:

14

Definition 15:
⊢ `EXP_ET_DISTRIB p X` λ_X `=` ∀ `t.` `0` ≤ `t` ⇒ `(CDF p X t = 1 - exp (`$-\lambda_X$ × `t))`

where 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., $\mathcal{P}(X \leq t)$.

Using Theorems 6-8 with the assumption that the failure and success states of each TL are exponentially distributed, we can formally verify the expression of $\mathcal{SAIFI}$ in HOL4 as follows:

Theorem 11:
⊢ $\mathcal{SAIFI}$ `(↑↓ [L2]) (↑↓ [M1; M2; M3; L1])`
　　　`[[31;30;29;28;27;26;25;24];...]` `[[M1 ↓; M2 ↓];...]`
　　　`[[11;12;13];[6;7;13];[2;5;7;10;12;13]] [X; Y; Z] p =`
　`(((1 - exp (`$-\lambda_{M1}$ × `t)) × (exp (`$-\lambda_{M2}$ × `t)) × (1 - exp (`$-\lambda_{M3}$ × `t)) ×`
　　`(exp (`$-\lambda_{L2}$ × `t)) + (1 - exp (`$-\lambda_{M1}$ × `t)) × (exp (`$-\lambda_{M2}$ × `t)) ×`
　　`(1 - exp (`$-\lambda_{M3}$ × `t)) × (1 - exp (`$-\lambda_{L2}$ × `t)) + ...)` × `X +`
　`((exp (`$-\lambda_{M1}$ × `t)) × (1 - exp (`$-\lambda_{M2}$ × `t)) × (1 - exp (`$-\lambda_{M3}$ × `t)) ×`
　　`(exp (`$-\lambda_{L1}$ × `t)) + ...)` × `Y +`
　`((exp (`$-\lambda_{M1}$ × `t)) × (exp (`$-\lambda_{M2}$ × `t)) × (1 - exp (`$-\lambda_{L1}$ × `t)) ×`
　　`(1 - exp (`$-\lambda_{L2}$ × `t)) + ...)` × `Z) / (X + Y + Z)`

To further facilitate the utilization of our proposed approach for safety engineers, we define an *Auto_$\mathcal{SAIFI}$_ML* Standard Meta Language (SML) function that can numerically compute the above-verified expression of $\mathcal{SAIFI}$. Assume that λ_{M1}, λ_{M2}, λ_{L1}, λ_{L2}, and λ_{L3} are 3, 2, 1, 4, 5 per year and X, Y, and Z are 250, 100, and 50 customers, respectively, then the result obtained by evaluating the $\mathcal{SAIFI}$ using *Auto_$\mathcal{SAIFI}$_ML* is *0.916173800938* interruptions/system customer. We also compared our computed result with the state-of-the-art reliability analysis tool Isograph [8], which is evaluated to *0.92* interruptions/system customer. It is quite evident that our proposed HOL4-based formalization approach provides the required rigor for ET analysis compared to existing simulation based approaches for system level reliability analysis. By conducting the formal ET analysis of an electrical power grid system and determining its reliability index $\mathcal{SAIFI}$, we demonstrated the practical effectiveness of the proposed ET formalization in the HOL4 theorem prover, which will help design engineers to meet the desired quality requirements. The proof-script of our proposed ET formalization and case study amounts to about 5000 lines of HOL4 code and can be downloaded from [23].

9 Conclusions

In this report, we described the HOL4 formalization of ETs step-analysis using a generic *list* data-type. We defined the `NODE` and `BRANCH` concepts, which can be used to model an arbitrary level of ET diagram consisting of $\mathcal{N}$ system components. We developed a formal approach to reduce ET branches, partition ET paths, and perform the probabilistic analysis based on the occurrence of certain events. For illustration purposes, we conducted the formal ET analysis of a power grid and also verified its system reliability index $\mathcal{SAIFI}$. As a future work, we plan to formalize the cascading dependencies in ETs [15], which will enable us to analyze hierarchical systems with many sub-system levels, based on our proposed ET formalization in the HOL4 theorem prover.

References

[1] R. E. Mackiewicz, "Overview of IEC 61850 and Benefits," in *Power Systems Conference and Exposition.* IEEE, 2006, pp. 623–630.

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

[3] M. Bozzano and A. Villafiorita, *Design and Safety Assessment of Critical Systems.* Auerbach Publications, 2010.

[4] M. Towhidnejad, D. R. Wallace, and A. M. Gallo, "Fault Tree Analysis for Software Design," in *27th NASA Goddard Software Engineering Workshop*, 2002, pp. 24–29.

[5] A. Brall, W. Hagen, and H. Tran, "Reliability Block Diagram Modeling-Comparisons of Three Software Packages," in *Reliability and Maintainability Symposium*, 2007, pp. 119–124.

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

[7] ITEM, 2020. [Online]. Available: https://itemsoft.com/eventtree.html

[8] Isograph, 2020. [Online]. Available: https://www.isograph.com

[9] D. K. Sen, J. C. Banks, G. Maggio, and J. Railsback, "Rapid Development of an Event Tree Modeling Tool Using COTS Software," in *Aerospace Conference.* IEEE, 2006, pp. 1–8.

[10] V. Muzik and Z. Vostracky, "Possibilities of Event Tree Analysis Method for Emergency States in Power Grid," in *Electric Power Engineering Conference.* IEEE, 2018, pp. 1–5.

[11] D. E. Peplow, C. D. Sulfredge, R. L. Sanders, R. H. Morris, and T. A. Hann, "Calculating Nuclear Power Plant Vulnerability Using Integrated Geometry and Event/Fault-Tree Models," *Nuclear Science and Engineering*, vol. 146, no. 1, pp. 71–87, 2004.

[12] B. H. Ku and J. M. Cha, "Reliability Assessment of Catenary of Electric Railway by Using FTA and ETA Analysis," in *Environment and Electrical Engineering.* IEEE, 2011, pp. 1–4.

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

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

[15] O. Nỳvlt and M. Rausand, "Dependencies in Event Trees Analyzed by Petri Nets," *Reliability Engineering & System Safety*, vol. 104, pp. 45–57, 2012.

[16] 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.

[17] 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.

[18] Y. Hu and M. Modarres, "Evaluating System Behavior Through Dynamic Master Logic Diagram (DMLD) Modeling," *Reliability Engineering & System Safety*, vol. 64, no. 2, pp. 241–269, 1999.

[19] I. A. Papazoglou, "Functional Block Diagrams and Automated Construction of Event Trees," *Reliability Engineering & System Safety*, vol. 61, no. 3, pp. 185–214, 1998.

[20] 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.

[21] E. C. Portante, S. F. Folga, J. A. Kavicky, and L. T. Malone, "Simulation of The September 8, 2011, San Diego Blackout," in *Winter Simulation Conference*, 2014, pp. 1527–1538.

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

[23] M. Abdelghany, "Formalization of Event Trees: HOL4 Script," 2020. [Online]. Available: https://github.com/hvg-concordia/Event-Trees-Formalization