36

Formal Reliability Analysis of Railway Systems Using Theorem Proving Technique

Waqar Ahmad, Osman Hasan, and Sofiène Tahar

CONTENTS

36.1 Introduction

In recent years, high-speed railway has been rapidly developed and deployed around the world including Germany, China, France, and Japan. The continuous endeavor to operate these trains at higher speeds has led to the development of high-speed railways into a new era. For instance, the high-speed railways in China had been operating at speeds of 300 km/h, but the introduction of the Beijing–Shanghai high-speed railway in June 2011 has further ushered China toward superhigh-speed trains that can operate at speeds of 380 km/h [1]. Due to the widespread coverage and continuous operation of the railway systems, the rigorous reliability analysis of these high-speed trains is a dire need. Moreover, a slight malfunctioning in the train components may cause undesirable delays at the arrival stations or even the loss of human lives in extreme cases.

Reliability block diagrams (RBDs) [2] are commonly used to develop reliability models for high-speed railway systems. Traditionally, these reliability models are analyzed by paper-and-pencil proof methods and simulation tools. However, the paper-and-pencil methods are prone to human errors for large systems, and it is often the case that many

key assumptions that are essentially required for the analytical proofs are in the minds of the engineers and, hence, are not properly documented. These missing assumptions are thus not communicated to the design engineers and are ignored in system implementations, which may also lead to unreliable designs. On the other hand, there are numerous simulation tools available, such as ReliaSoft [3] and ASENT reliability analysis tool [4], that offer scalable reliability analysis compared to paper-and-pencil methods. However, these tools cannot ensure accurate analysis due to the involvement of pseudorandom numbers and numerical methods. Additionally, exhaustive verification of systems for all values of the variables is not possible.

To overcome the inaccuracy limitations of traditional techniques mentioned earlier, formal methods have also been proposed as an alternative for the RBD-based analysis using both state-based [5,6] and theorem-proving techniques [7]. The main idea behind the formal analysis of a system is to first construct a mathematical model of the given system using a state machine or an appropriate logic and then use logical reasoning and deduction methods to formally verify that this system model exhibits the desired characteristics, which are also mathematically specified using an appropriate logic. However, state-based approaches cannot be used for verifying generic mathematical expressions for reliability. On the other hand, theorem proving, which is based on the expressive higher-order logic (HOL) [8], allows working with a variety of datatypes, such as lists and real numbers and has been recently used to formalize commonly used RBDs [9] by leveraging upon the probability theory formalization in HOL [10]. This HOL-based RBD formalization provides the formally verified generic reliability expressions that can be used to carry out an accurate and rigorous reliability analysis of high-speed railway systems. In this chapter, we have utilized the recently proposed HOL formalization of RBDs [9] to conduct formal reliability analysis of a railway system designed for the Italian high-speed railways [11] consisting of several critical components, such as traction drive system, induction motors, converters, and transformers.

The rest of the chapter is organized as follows: Section 36.2 presents a review of the related work. Section 36.3 provides an overview of the proposed methodology that has been used to conduct formal reliability analysis of railway systems. To facilitate the understanding of the chapter for nonexperts in theorem proving, we present a brief introduction about theorem proving, the HOL theorem prover, and the formalization of probability and reliability theories in Section 36.4. This is followed by the description of our formalization of the RBD configurations in Section 36.5. The RBD-based formal reliability analysis of the Italian high-speed railway system is presented in Section 36.6, and finally Section 36.7 concludes the chapter.

36.2 Related Work

Many simulation tools, such as DNV-GL [12], ReliaSoft [3], and ASENT [4], support RBD-based reliability analysis and provide powerful graphical editors that can be used to construct the RBD models of the high-speed trains. These tools generate samples from the exponential or Weibull random variables to model the reliabilities of the individual system

components. These samples are then processed by using computer arithmetic and numerical techniques in order to compute the reliability of the complete system. Although these software tools provide more scalable and quick analysis compared to paper-and-pencil based analytical methods, they cannot ascertain the absolute correctness of the system because of their inherent sampling based nature and the involvement of pseudorandom numbers and numerical methods.

Formal methods, such as Petri nets (PNs), have also been used to model RBDs [13] as well as dynamic RBDs [5] that are used to describe the reliability behavior of systems. PN verification tools, based on model checking principles, are then used to verify behavioral properties of the RBD models to identify design flaws [5,13]. Similarly, the probabilistic model checker *Prism* [14] has been used for the quantitative verification of various safety and mission-critical systems, such as failure analysis for an industrial product development workflow [15], an airbag system [6], and the reliability analysis of a global navigation satellite system that enables an aircraft to determine its position (latitude, longitude, and altitude) [16]. However, due to the state-based models, only state-related property verification, such as deadlock checks, reachability, and safety properties, is supported by these approaches, i.e., we cannot verify generic reliability relationships for the given system using the approaches presented in the studies by Robidoux et al. [5], Norman and Parker [6], Signoret et al. [13], Herbert and Hansen [15], and Lu et al. [16].

A number of formalizations of probability theory are available in HOL (e.g., the studied by Mhamdi et al. [10], Hurd [17], and Hölzl and Heller [18]). Hurd's [17] formalization of probability theory has been utilized to verify sampling algorithms of a number of commonly used discrete [19] and continuous random variables [20,21] based on their probabilistic and statistical properties. Moreover, this formalization has been used to conduct the reliability analysis of a number of applications, such as memory arrays [22] and electronic components [23]. However, Hurd's formalization of probability theory only supports having the whole universe as the probability space. This feature limits its scope, and thus, this probability theory cannot be used to formalize more than a single continuous random variable, whereas in the case of reliability analysis of railways systems, multiple continuous random variables are required. The recent formalizations of probability theory by Mhamdi et al. [10] and Hölzl and Heller [18] are based on extended real numbers (including $\pm\infty$) and provide the formalization of Lebesgue integral to reason about advanced statistical properties. These theories also allow using any arbitrary probability space, a subset of the universe, and are thus more flexible than Hurd's formalization. Leveraging upon the high expressiveness of HOL and the inherent soundness of theorem proving, Mhamdi et al.'s [10] formalized probability theory has been recently used for the formalization of RBDs [9], including series [7], parallel [24], parallel–series [24], series–parallel [25], and k-out-*n* [26]. These formalizations have been used for the reliability analysis of many applications, including simple oil and gas pipelines with serial components [7], wireless sensor network protocols [24], logistic supply chains [25], and oil and gas pipelines [26]. Similarly, Mhamdi et al.'s probability theory has also been used for the formalization of commonly used fault tree (FT) gates, such as AND, OR, NAND, NOR, XOR, and NOT, and the probabilistic inclusion–exclusion principle [27]. In addition, the RBD and FT formalizations mentioned earlier have been recently utilized for availability analysis [28]. In this chapter, we utilize recently proposed HOL formalization of RBDs

[9] to carry out the formal reliability analysis of a railway system operated by the Italian high-speed railways.

36.3 Proposed Methodology

The proposed methodology for the formal reliability analysis of railway systems, depicted in Figure 36.1, allows us to formally verify the reliability expressions corresponding to the given *railway system description* and thus formally check that the given railway system satisfies its reliability requirements. The core component of this methodology is the HOL formalizations of the notions of probability, reliability, and RBDs.

The given railway system is first partitioned into segments, and the corresponding *RBD model* is constructed. This model can then be formalized in HOL using the core formalizations mentioned earlier, particularly the formalization of commonly used RBD configurations. The next step is to assign failure distributions, such as exponential and Weibull, to individual components of the given railway system. These distributions are also formalized by building upon the formalized probability theory and are used, along with the formal RBD model, to formalize the given reliability requirements as a proof goal in HOL. The user has to reason about the correctness of this *proof goal* using a theorem prover by building upon the core formalizations of probability and reliability theories. If all subgoals are discharged, then we obtain formally verified reliability expressions, which correspond to the given railway system and its reliability requirements of the given railway system. Otherwise, we can use the failing subgoals to debug the formal RBD model and proof goal, which represent the originally specified model and requirements, respectively, as depicted by the dotted line in Figure 36.1.

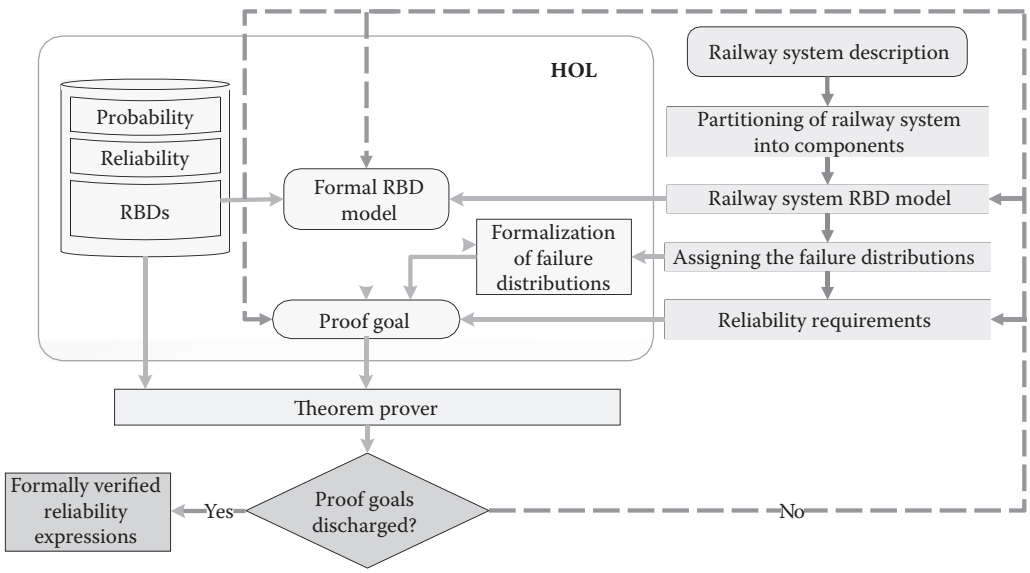

FIGURE 36.1
Methodology for formal railway system reliability analysis.

36.4 Preliminaries

In this section, we give a brief introduction to theorem proving and the HOL theorem prover to facilitate the understanding of the rest of the chapter.

36.4.1 Theorem Proving

Theorem proving [8] is a widely used formal verification technique. The system that needs to be analyzed is mathematically modeled in an appropriate logic, and the properties of interest are verified using computer-based formal tools. The use of formal logics as a modeling medium makes theorem proving a very flexible verification technique as it is possible to formally verify any system that can be mathematically described. The core of theorem provers usually consists of some well-known axioms and primitive inference rules. Soundness is assured as every new theorem must be created from these basic or already proven theorems and primitive inference rules. The verification effort of a theorem in a theorem prover varies from trivial to complex depending on the underlying logic [29].

36.4.2 HOL Theorem Prover

HOL [30] is an interactive theorem prover developed at the University of Cambridge, United Kingdom, for conducting proofs in HOL. It utilizes the simple type theory of Church [31] along with Hindley–Milner polymorphism [32] to implement HOL. HOL has been successfully used as a verification framework for both software and hardware as well as a platform for the formalization of pure mathematics.

The HOL core consists of only five basic axioms and eight primitive inference rules, which are implemented as meta language (ML) functions. The type system of the ML ensures that only valid theorems can be constructed. Soundness is assured as every new theorem must be verified by applying these basic axioms and primitive inference rules or any other previously verified theorems/inference rules.

In the work presented in this chapter, we utilize the HOL theories of Booleans, lists, sets, positive integers, *real* numbers, measure, and probability [10]. In fact, one of the primary motivations of selecting the HOL theorem prover for our work was to benefit from these built-in mathematical theories. Table 36.1 provides the mathematical interpretations of some frequently used HOL symbols and functions, which are inherited from existing HOL theories.

36.4.3 Probability and Reliability in HOL

Mathematically, a measure space is defined as a triple (Ω, Σ, μ), where Ω is a set, called the sample space; Σ represents a σ algebra of subsets of Ω, where the subsets are usually referred to as measurable sets; and μ is a measure with domain Σ. A probability space is a measure space (Ω, Σ, Pr), such that the measure, referred to as the probability and denoted by Pr, of the sample space is 1. In the HOL formalization of probability theory [10], given a probability space p, the functions `space`, `subsets`, and `prob` return the corresponding Ω, Σ, and Pr, respectively. This formalization also includes the formal verification of some of the most widely used probability axioms, which play a pivotal role in formal reasoning about reliability properties.

TABLE 36.1

HOL Symbols and Functions

HOL Symbol	Standard Symbol	Meaning
∧	*and*	Logical *and*
∨	*or*	Logical *or*
¬	*not*	Logical *negation*
::	*cons*	Adds a new element to a list
++	*append*	Joins two lists together
HD L	*head*	Head element of list *L*
TL L	*tail*	Tail of list *L*
EL n L	*element*	*n*th element of list *L*
MEM a L	*member*	True if *a* is a member of list *L*
λ x.t	$\lambda x.t$	Function that maps *x* to *t*(*x*)
SUC n	$n+1$	Successor of a *num*
lim(λ n.f(n))	$\lim_{n\to\infty} f(n)$	Limit of a *real* sequence *f*

A random variable is a measurable function between a probability space and a measurable space. The measurable functions belong to a special class of functions, which preserves the property that the inverse image of each measurable set is also measurable. A measurable space refers to a pair (*S*, *A*), where *S* denotes a set and *A* represents a nonempty collection of subsets of *S*. Now, if *S* is a set with finite number of elements, then the corresponding random variable is termed as discrete; otherwise, it is known as a continuous random variable.

The probability that a random variable *X* is less than or equal to some value *t*, $Pr(X \le t)$ is called the cumulative distribution function (CDF), and it characterizes the distribution of both discrete and continuous random variables. The CDF has been formalized in HOL as follows [7]:

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

where the variables *p*: (α → *bool*)#((α → *bool*) → *bool*)#((α → *bool*) → *real*), *X*: (α → *extreal*), and *t*: *real* represent a probability space, a random variable, and a *real* number, respectively. The function `Normal` takes a *real* number as its input and converts it to its corresponding value in the *extended real* data type, i.e., it is the *real* data type with the inclusion of positive and negative infinity. The function `distribution` takes three parameters: a probability space *p*, a random variable *X*, and a set of *extended real* numbers and outputs the probability of a random variable *X* that acquires all values of the given set in probability space *p*.

Now, reliability *R*(*t*) is stated as the probability of a system or component performing its desired task over a certain interval of time *t*:

$$R(t) = Pr(X > t) = 1 - Pr(X \le t) = 1 - F_X(t) \tag{36.1}$$

where $F_X(t)$ is the CDF. The random variable *X*, in the preceding definition, models the time to failure of the system and is usually modeled by the exponential random

variable with parameter λ, which corresponds to the failure rate of the system. Based on the HOL formalization of probability theory [10], Equation 36.1 has been formalized as follows [7]:

```
⊢ ∀ p X t. Reliability p X t = 1 − CDF p X t.
```

The series RBD, presented by Ahmad et al. [7], is based on the notion of mutual independence of random variables, which is one of the most essential prerequisites for reasoning about the mathematical expressions for all RBDs. If N reliability events are mutually independent, then

$$Pr\left(\bigcap_{i=1}^{N} A_i\right) = \prod_{i=1}^{N} Pr(A_i). \tag{36.2}$$

This concept has been formalized as follows [7]:

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

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

36.5 Formalization of the Reliability Block Diagrams

Commonly used RBD configurations for the reliability analysis of the railway system include series, parallel, and a combination of both and are depicted in Figure 36.2. In this chapter, we present their formalization, which, in turn, can then be used to formally model the structures of a railway system in HOL and reason about their reliability, availability, and maintainability characteristics.

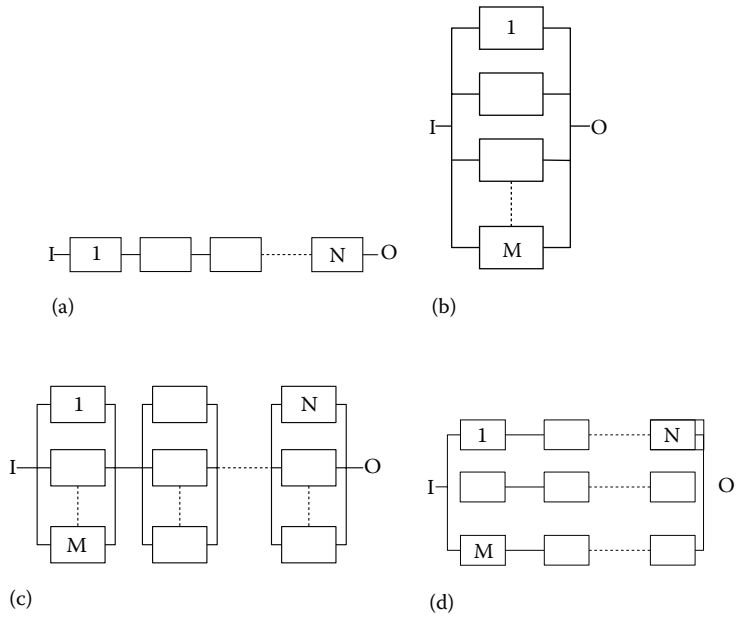

FIGURE 36.2
RBDs: (a) series; (b) parallel; (c) series–parallel; (d) parallel–series.

36.5.1 Formalization of Reliability Event

We describe the formally verified reliability expressions for the commonly used RBD configurations using reliability event lists, where a single event represents the scenario when the given system or component does not fail before a certain time. The HOL formalization of this concept is as follows:

Definition 36.1: `⊢ ∀ p X t.`

```
rel_event p X t = PREIMAGE X {y | Normal t < y} ∩ p_space p
```

The function `PREIMAGE` takes two arguments, a function *f* and a set *s*, and returns a set, which is the domain of the function *f* operating on a given range set *s*. The function `rel _ event` accepts a probability space *p*; a random variable *X*, representing the failure time of a system or a component; and a real number *t*, which represents the time index at which the reliability is desired. It returns an event representing the reliable functioning of the system or component at time *t*.

Similarly, a list of reliability events can be derived by mapping the function `rel _ event` on each element of the given random variable list in HOL as follows:

Definition 36.2: `⊢ ∀ p L t.`

```
rel_event_list p L t = MAP (λa. rel_event p a t) L,
```

where the HOL function `MAP` takes a function *f* and a list and returns a list by applying the function *f* on each element of the given list.

Now, we describe the formalization process by type abbreviating the notion of event, which is essentially a set of observations with type/$a \rightarrow bool$ as follows:

```
type_abbrev ("event" , ``:'a ->bool'')
```

We then define a recursive datatype *rbd* in the HOL system as follows:

```
Hol_datatype 'rbd = series of rbd list |
                parallel of rbd list |
                atomic of 'a event'
```

An RBD can be either a series configuration, a parallel configuration, or an atomic event. The type constructors `series` and `parallel` recursively function on *rbd*-typed lists and thus enable us to deal with nested RBD configurations. The type constructor `atomic` is basically a typecasting operator between *event* and *rbd*-typed variables. Typically, a new datatype is defined in HOL as $(\alpha_1, \alpha_2, ..., \alpha_n)op$, where $(\alpha_1, \alpha_2, ..., \alpha_n)$ represent the arguments taken by the HOL datatype *op* [30]. For instance, the `atomic` type constructor is defined with the arbitrary type α, which is taken by the already defined type `events`. On the other hand, the type constructors `series` and `parallel` are defined without any arguments because the datatype `rbd` is not defined at this point.

We define a semantic function `rbd _ struct` (α *event* # α *event event* # (α *event* → *real)* → α *rbd* → α *event*) inductively over the *rbd* datatype. It extracts the corresponding event from the given RBD configuration as follows:

Definition 36.3: ⊢(∀p. rbd _ struct p (series []) = p _ space p) ∧

```
(∀ xs x p.
  rbd_struct p (series (x::xs)) =
  rbd_struct p x ⋂ rbd_struct p (series xs)) ∧
(∀ p. rbd_struct p (parallel []) = {}) ∧
(∀ xs x p.
  rbd_struct p (parallel (x::xs)) =
  rbd_struct p x ∪ rbd_struct p (parallel xs)) ∧
(∀ p a. rbd_struct p (atomic a) =  a)
```

The preceding function decodes the semantic embedding of an arbitrary RBD configuration by extracting the corresponding reliability event, which can then be used to determine the reliability of a given RBD configuration. The function `rbd _ struct` takes an *rbd*-typed list identified by a type constructor `series` and returns the whole probability space if the given list is empty and, otherwise, returns the intersection of the events that is obtained after applying the function `rbd _ struct` on each element of the given list in order to model the series RBD configuration behavior. Similarly, to model the behavior of a parallel RBD configuration, the function `rbd _ struct` operates on an *rbd*-typed list encoded by a type constructor `parallel`. It then returns the union of the events after applying the function `rbd _ struct` on each element of the given list or an empty set if the given list is empty. The function `rbd _ struct` returns the reliability event using the type constructor `atomic`.

In the subsequent sections, we present the HOL formalization of RBDs on any reliability event list of arbitrary length [24,25]. The notion of reliability event is then incorporated in the formalization while carrying out the reliability analysis of a real railway system, as it will be described in Section 36.6.

36.5.2 Formalization of Series Reliability Block Diagrams

The reliability of a system with components connected in series is considered to be reliable at time t only if all its components are functioning reliably at time t, as depicted in Figure 36.2a. If $A_i(t)$ is a mutually independent event that represents the reliable functioning of the ith component of a serially connected system with N components at time t, then the overall reliability of the complete system can be expressed as [33]

$$R_{\text{series}}(t) = Pr\left(\bigcap_{i=1}^{N} A_i(t)_{i=1}\right) = \prod_{i=1}^{N} R_i(t). \tag{36.3}$$

Now using Definition 36.3, we can formally verify the reliability expression, given in Equation 3, for a series RBD configuration in HOL as follows:

Theorem 36.1: ⊢ ∀ p L. prob space p ∧

```
¬NULL L ∧ (∀x'. MEM x' L ⇒ x' ∈ events p) ∧
mutual_indep p L ⇒
(prob p (rbd_struct p (series (rbd_list L))) =
 list_prod (list_prob p L))
```

The first assumption, in Theorem 36.1, ensures that p is a valid probability space based on the probability theory in HOL [10]. The next two assumptions guarantee that the list of events L, representing the reliability of individual components, must have at least one event and the reliability events are mutually independent. The conclusion of the theorem represents Equation 36.3. The function `rbd _ list` generates a list of type `rbd` by mapping the function `atomic` to each element of the given event list L to make it consistent with the assumptions of Theorem 36.1. It can be formalized in HOL as

```
∀ L. rbd_list L = MAP (λa. atomic a) L.
```

The proof of Theorem 36.1 is primarily based on mutual independence properties and some fundamental axioms of probability theory.

36.5.3 Formalization of Parallel Reliability Block Diagrams

The reliability of a system with parallel connected submodules, depicted in Figure 36.2b, mainly depends on the component with the maximum reliability. In other words, the system will continue functioning as long as at least one of its components remains functional. If the event $A_i(t)$ represents the reliable functioning of the ith component of a system with N parallel components at time t, then the overall reliability of the system can be mathematically expressed as [33]

$$R_{\text{parallel}}(t) = Pr\left(\bigcup_{i=1}^{N} A_i(t)\right) = 1 - \prod_{i=1}^{N}\left(1 - R_i(t)\right). \tag{36.4}$$

Similarly, by following the formalization approach of series RBD mentioned earlier, we can formally verify the reliability expression for the parallel RBD configuration, given in Equation 36.4, in HOL as follows:

Theorem 36.2: ⊢ ∀ p L.

```
prob_space p ∧ (∀x'. MEM x' L ⇒ x' ∈ events p) ∧
¬NULL L ∧ mutual_indep p L ⇒
 (prob p (rbd_struct p (parallel (rbd_list_L))) =
 1 - list_prod (one_minus_list (list_prob_p L)))
```

The preceding theorem is verified under the same assumptions as Theorem 36.1. The conclusion of the theorem represents Equation 36.4, where the function `one _ minus _ list` accepts a list of *real* numbers [*x*1, *x*2, *x*3, …, *xn*] and returns the list of *real* numbers such that each element of this list is 1 minus the corresponding element of the given list, i.e., [1 – *x*1, 1 – *x*2, 1 – *x*3,…, 1 – *xn*].

The preceding formalization described for series and parallel RBD configurations builds the foundation to formalize the combination of series and parallel RBD configurations. The type constructors `series` and `parallel` can take the argument list containing other *rbd* type constructors, such as `series`, `parallel`, or `atomic`, allowing the function `rbd _ struct` o yield the corresponding event for an RBD configuration that is composed of a combination of series and parallel RBD configurations.

36.5.4 Formalization of Series–Parallel Reliability Block Diagrams

If in each serial stage the components are connected in parallel, as shown in Figure 36.2c, then the configuration is termed as a *series–parallel structure*. If $A_{ij}(t)$ is the event corresponding to the proper functioning of the *j*th component connected in an *i*th subsystem at time index *t*, then the reliability of the complete system can be expressed mathematically as follows [33]:

$$R_{\text{series–parallel}}(t) = Pr\left(\bigcap_{i=1}^{N}\bigcup_{j=1}^{M} A_{ij}(t)\right) = \prod_{i=1}^{N}\left(1 - \prod_{j=1}^{M}\left(1 - R_{ij}(t)\right)\right). \tag{36.5}$$

By extending the RBD formalization approach, presented in Theorems 36.1 and 36.2, we formally verify the generic reliability expression for series-parallel RBD configuration, given in Equation 36.6), in HOL as follows:

Theorem 36.3: ⊢ ∀ p L. prob _ space p ∧

```
(∀z. MEM z L ⇒ ¬NULL z) ∧
(∀x'. MEM x' (FLAT L) ⇒ x'∈ events p) ∧
 mutual_indep p (FLAT L) ⇒
  (prob p
     (rbd_struct p ((series of (λa. parallel (rbd_list a))) L)) =
(list_prod of
  (λa. 1 - list_prod (one_minus_list (list_prob p a)))) L)
```

The first assumption in Theorem 36.3 is similar to the one used in Theorem 36.2. The next three assumptions ensure that the sublists corresponding to the serial substages are not empty, and the reliability events corresponding to the subcomponents of the parallel–series configuration are valid events of the given probability space p and are mutually independent. The HOL function `FLAT` is used to flatten the two-dimensional list, i.e., to transform a list of lists, into a single list. The conclusion models the right-hand side of Equation 36.5). The function `of` is defined as an infix operator [30] in order to connect the two *rbd*-typed constructors by using the HOL `MAP` function and thus facilitates the natural readability of complex RBD configurations. It is formalized in HOL as follows:

```
⊢ ∀ g f. f of g = (f o (λa. MAP g a))
```

36.5.5 Formalization of Parallel–Series Reliability Block Diagrams

If $A_{ij}(t)$ is the event corresponding to the reliability of the j^{th} component connected in a i^{th} subsystem at time t, then the reliability of the complete system can be expressed as follows:

$$R_{\text{parallel-series}}(t) = Pr\left(\bigcup_{i=1}^{M}\bigcap_{j=1}^{N} A_{ij}(t)\right) = 1 - \prod_{i=1}^{M}\left(1 - \prod_{j=1}^{N}\left(R_{ij}(t)\right)\right). \tag{36.6}$$

Similarly, the generic expression of the parallel–series RBD configuration, given in Equation 36.6, is formalized in HOL as follows:

Theorem 36.4: `⊢ ∀ p L. prob_space p ∧`

```
(∀z. MEM z L ⇒ ¬NULL z) ∧
(∀x'. MEM x' (FLAT L) ⇒ x' ∈ events p) ∧
 mutual_indep p (FLAT L) ⇒
(prob p
  (rbd_struct p ((parallel of (λa. series (rbd_list a))) L)) =
1 - (list_prod o (one_minus_list) of
  (λa. list_prod (list_prob p a))) L)
```

The assumptions of Theorem 36.4 are similar to those used in Theorem 36.3. The conclusion models the right-hand side of Equation 36.6.

To verify Theorems 36.3 and 36.4, it is required to formally verify various structural independence lemmas, for instance, given the list of mutually independent reliability events, an event corresponding to the series or parallel RBD structure is independent, in probability, with the corresponding event associated with the parallel–series or series–parallel RBD configurations.

36.6 Traction Drive System of High-Speed Trains

In order to illustrate the practical effectiveness of the RBD-based formal reliability analysis using theorem proving, we consider a multivoltage railway system, specifically designed for the Italian high-speed railways [11]. The overall railway system consists of three identical

modules, i.e., A, B, and C, as depicted in Figure 36.3. Each module represents a traction drive system and two boogies that are composed of two bearings and one reduction gear. The most critical part in the railway system is the traction drive system because a slight malfunctioning in its key components may lead to train delay, affect the operation order, and endanger the safe operation of the train. A traction drive system in each module consists of a transformer, a filter, an inverter, two four-quardent converters and four induction motors that are connected with two boogies. The RBD diagram of the overall railway system is shown in Figure 36.3. The HOL formalization of the given train RBD is as follows:

Definition 36.4: ⊢ ∀ p T1 FQC1 FQC2 F1 I1 IM1 IM2 B1 IM3 IM4 B2 T2 FQC3 FQC4 F2 I2 IM5 IM6 B3 IM7 IM8 B4 T3 FQC5 FQC6 F3 I3 IM9 IM10 B5 IM11 IM12 B6.

```
 railway_RBD p T1 FQC1 FQC2 F1 I1 IM1 IM2 B1 IM3 IM4 B2 T2 FQC3 FQC4
F2 I2 IM5 IM6 B3 IM7 IM8 B4 T3 FQC5 FQC6 F3 I3 IM9 IM10 B5 IM11 IM12
B6 =
 rbd_struct p (parallel
  [MA_RBD T1 FQC1 FQC2 F1 I1 IM1 IM2 B1 IM3 IM4 B2;
   MB_RBD T2 FQC3 FQC4 F2 I2 IM5 IM6 B3 IM7 IM8 B4;
   MC_RBD T3 FQC5 FQC6 F3 I3 IM9 IM10 B5 IM11 IM12 B6])
```

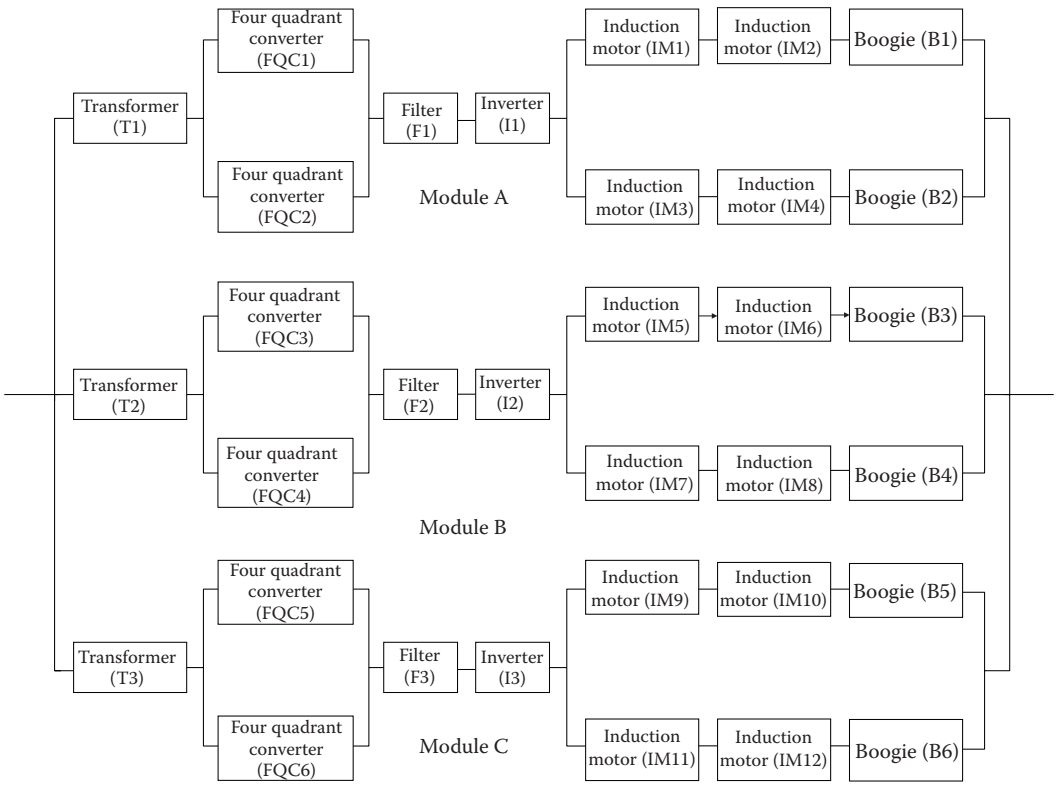

FIGURE 36.3
Railway system RBD.

where MA _ RBD, MB _ RBD, and MC _ RBD are RBDs corresponding to the train modules A, B, and C, as depicted in Figure 36.3. For instance, the HOL formalization of MA _ RBD is as follows:

Definition 36.5: ⊢ ∀ T1 FQC1 FQC2 F1 I1 IM1 IM2 B1 IM3 IM4 B2.

```
MA_RBD T1 FQC1 FQC2 F1 I1 IM1 IM2 B1 IM3 IM4 B2 =
series
 [atomic T1; parallel [atomic FQC1; atomic FQC2];
  series [atomic F1; atomic I1];
  parallel
   [series [atomic IM1; atomic IM2; atomic B1];
    series [atomic IM3; atomic IM4; atomic B2]]]
```

where T1, FQC1, FQC2, F1, I1, IM1, IM2, IM3, IM4, B1, and B2 are the events corresponding to transformer, four-quadrant converter, filter, inverter, induction motors and boogies, respectively.

In the same way, we can formalize the functions MB _ RBD and MC _ RBD in HOL, and the formal definitions can be found in the study by Ahmed [34]. It can be observed that the railway _ RBD does not seem to be directly mapping to any of the commonly used RBDs, which are described in Section 36.5. However, we can mathematically verify that this configuration is equivalent to one of those generic RBDs, i.e., the parallel–series RBD configuration in this case. The following lemma formally describes this relationship:

Lemma 36.1: ⊢ ∀ p T1 FQC1 FQC2 F1 I1 IM1 IM2 B1 IM3 IM4 B2 T2

```
FQC3 FQC4 F2 I2 IM5 IM6 B3 IM7 IM8 B4 T3 FQC5 FQC6 F3 I3 IM9 IM10 B5
IM11 IM12 B6.
 railway_RBD p T1 FQC1 FQC2 F1 I1 IM1 IM2 B1 IM3 IM4 B2 T2 FQC3
FQC4 F2 I2 IM5 IM6 B3 IM7 IM8 B4 T3 FQC5 FQC6 F3 I3 IM9 IM10 B5
IM11 IM12 B6 =
 rbd_struct p ((parallel of (λa. series (rbd_list a)))
[[IM1; IM2; B1; T1; F1; I1; FQC1];
 [IM1; IM2; B1; T1; F1; I1; FQC2];
 [IM3; IM4; B2; T1; F1; I1; FQC1];
 [IM3; IM4; B2; T1; F1; I1; FQC2];
 [IM5; IM6; B3; T2; F2; I2; FQC3];
 [IM5; IM6; B3; T2; F2; I2; FQC4];
 [IM7; IM8; B4; T2; F2; I2; FQC3];
 [IM7; IM8; B4; T2; F2; I2; FQC4];
 [IM9; IM10; B5; T3; F3; I3; FQC5];
 [IM9; IM10; B5; T3; F3; I3; FQC6];
 [IM11; IM12; B6; T3; F3; I3; FQC5];
 [IM11; IM12; B6; T3; F3; I3; FQC6]])
```

Each component of a railway system is exponentially distributed, as described by Dazi et al. [11], so we can express the reliability of the railway system, as shown in Figure 36.3, mathematically as follows:

$$
\begin{aligned}
R_{\text{railway\_system}} = 1 - &\left(1 - e^{-(\lambda_{IM1}+\lambda_{IM2}+\lambda_{B1}+\lambda_{T1}+\lambda_{F1}+\lambda_{I1}+\lambda_{FQC1})t}\right) \\
&\times \left(1 - e^{-(\lambda_{IM1}+\lambda_{IM2}+\lambda_{B1}+\lambda_{T1}+\lambda_{F1}+\lambda_{I1}+\lambda_{FQC1})t}\right) \\
&\times \left(1 - e^{-(\lambda_{IM1}+\lambda_{IM2}+\lambda_{B1}+\lambda_{T1}+\lambda_{F1}+\lambda_{I1}+\lambda_{FQC2})t}\right) \\
&\times \left(1 - e^{-(\lambda_{IM3}+\lambda_{IM4}+\lambda_{B2}+\lambda_{T1}+\lambda_{F1}+\lambda_{I1}+\lambda_{FQC1})t}\right) \\
&\times \left(1 - e^{-(\lambda_{IM3}+\lambda_{IM4}+\lambda_{B2}+\lambda_{T1}+\lambda_{F1}+\lambda_{I1}+\lambda_{FQC2})t}\right) \\
&\times \left(1 - e^{-(\lambda_{IM5}+\lambda_{IM6}+\lambda_{B3}+\lambda_{T2}+\lambda_{F2}+\lambda_{I2}+\lambda_{FQC3})t}\right) \\
&\times \left(1 - e^{-(\lambda_{IM5}+\lambda_{IM6}+\lambda_{B3}+\lambda_{T2}+\lambda_{F2}+\lambda_{I2}+\lambda_{FQC4})t}\right) \\
&\times \left(1 - e^{-(\lambda_{IM7}+\lambda_{IM8}+\lambda_{B4}+\lambda_{T2}+\lambda_{F2}+\lambda_{I2}+\lambda_{FQC3})t}\right) \\
&\times \left(1 - e^{-(\lambda_{IM7}+\lambda_{IM8}+\lambda_{B4}+\lambda_{T2}+\lambda_{F2}+\lambda_{I2}+\lambda_{FQC4})t}\right) \\
&\times \left(1 - e^{-(\lambda_{IM9}+\lambda_{IM10}+\lambda_{B5}+\lambda_{T3}+\lambda_{F3}+\lambda_{I3}+\lambda_{FQC5})t}\right) \\
&\times \left(1 - e^{-(\lambda_{IM9}+\lambda_{IM10}+\lambda_{B5}+\lambda_{T3}+\lambda_{F3}+\lambda_{I3}+\lambda_{FQC6})t}\right) \\
&\times \left(1 - e^{-(\lambda_{IM11}+\lambda_{IM12}+\lambda_{B6}+\lambda_{T3}+\lambda_{F3}+\lambda_{I3}+\lambda_{FQC5})t}\right) \\
&\times \left(1 - e^{-(\lambda_{IM11}+\lambda_{IM12}+\lambda_{B6}+\lambda_{T3}+\lambda_{F3}+\lambda_{I3}+\lambda_{FQC6})t}\right)
\end{aligned}
\tag{36.7}
$$

In order to formally verify the preceding equation, we first formalize the notion of exponentially distributed random variable in HOL as follows:

Definition 36.6: `⊢ ∀ p X c. exp _ dist p X c =`

```
∀ t. (CDF p X t = if 0 ≤ t then 1 - exp (-c * t) else 0)
```

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

Definition 36.7: `⊢ (∀ p L. exp _ dist _ list p L [] = T) ∧`

```
∀ p h t L. exp_dist_list p L (h::t) =
        exp_dist p (HD L) h ∧ exp_dist_list p (TL L) t
```

where the symbol *T* stands for logical *True*. The function `exp _ dist _ list` accepts a list of random variables *L*, a list of failure rates and a probability space *p*. It guarantees that all elements of the random variable list *L* are exponentially distributed with the corresponding failure rates, given in the other list, within the probability space *p*. For this purpose, it utilizes the list functions `HD` and `TL`, which return the *head* and *tail* of a list, respectively.

By using the definitions mentioned earlier, we can formally verify the reliability expression of the railway system, given in Equation 36.7, in HOL as follows:

Theorem 36.5: ⊢ ∀ p X _ T1 X _ FQC1 X _ FQC2 X _ F1 X _ I1
X _ IM1 X _ IM2 X _ B1 X _ IM3 X _ IM4

```
X_B2 X_T2 X_FQC3 X_FQC4 X_F2 X_I2 X_IM5 X_IM6 X_B3 X_IM7 X_IM8 X_B4 X_T3
X_FQC5 X_FQC6 X_F3 X_I3 X_IM9 X_IM10 X_B5 X_IM11 X_IM12 X_B6 C_FQC1
C_FQC2
C_F1 C_I1 C_IM1 C_IM2 C_B1 C_IM3 C_IM4 C_B2 C_T2 C_FQC3 C_FQC4 C_F2 C_I2
C_IM5 C_IM6 C_B3 C_IM7 C_IM8 C_B4 C_T3 C_FQC5 C_FQC6 C_F3 C_I3 C_IM9
C_IM10
C_B5 C_IM11 C_IM12 C_B6.
(A1): 0 ≤ t ∧ (A2): prob_space p ∧
(A3): in_events p [X_T1; X_FQC1; ···; X_B6] t ∧
(A4): mutual_indep p
     (rel_event_list p [X_T1; X_FQC1; ···; X_B6] t)) ∧
(A5): exp_dist_list p
     [X_T1; X_FQC1; ···; X_B6] [C_FQC1; C_FQC2; ···; C_B6] ⇒
  (prob p (railway_RBD p (rel_event p T1 t) (rel_event p FQC1 t)
    (rel_event p FQC2 t) ··· (rel_event p B6 t)) =
  1 - (list_prod o one_minus_list of
  (λa. list_prod (exp_func_list a t)))
   [[C_IM1;C_IM2;C_B1;C_T1;C_F1;C_I1;C_FQC1];
   [C_IM1;C_IM2;C_B1;C_T1;C_F1;C_I1;C_FQC2];
   [C_IM3;C_IM4;C_B2;C_T1;C_F1;C_I1;C_FQC1];
   [C_IM3;C_IM4;C_B2;C_T1;C_F1;C_I1;C_FQC2];
   [C_IM5;C_IM6;C_B3;C_T2;C_F2;C_I2;C_FQC3];
   [C_IM5;C_IM6;C_B3;C_T2;C_F2;C_I2;C_FQC4];
   [C_IM7;C_IM8;C_B4;C_T2;C_F2;C_I2;C_FQC3];
   [C_IM7;C_IM8;C_B4;C_T2;C_F2;C_I2;C_FQC4];
   [C_IM9;C_IM10;C_B5;C_T3;C_F3;C_I3;C_FQC5];
   [C_IM9;C_IM10;C_B5;C_T3;C_F3;C_I3;C_FQC6];
   [C_IM11;C_IM12;C_B6;C_T3;C_F3;C_I3;C_FQC5];
   [C_IM11;C_IM12;C_B6;C_T3;C_F3;C_I3;C_FQC6]])
```

In the preceding theorem, we have replaced the events that are associated with the railway components in function `railway _ RBD` with their corresponding random variable form by using Definition 36.1. It allows us to assign the exponential failure distribution with the random variables that correspond to the railway components. The assumptions A1 and A2 ensure that the time index must be positive, and *p* is a valid probability space. The assumptions A3 and A4 guarantee that the events associated with the railway components are in events space *p* and mutually independent in the probability space *p*. The predicate `in _ events` takes a probability space *p*, a list of random variables, and a time index *t* and makes sure that each element in the given random variable list is in event space *p*. The last assumption (A5) ensures that the random variables that are exponentially distributed are assigned the corresponding failure rates. The conclusion of Theorem 36.5 models Equation 36.7. The proof of Theorem 36.5 utilizes Theorem 36.4, Lemma 36.1, and some fundamental axioms of probability theory.

The distinguishing features of the formally verified Theorem 36.5, compared to simulation-based reliability analysis of the railway system [11], include its generic nature and guaranteed correctness. All variables in Theorem 36.5 are universally quantified and can thus be specialized to obtain the reliability of any railway system for any given failure rates. The correctness of our results is guaranteed thanks to the involvement of a sound

theorem prover in their verification, which ensures that all required assumptions for the validity of the results are accompanying the theorem. Unlike the work presented by Dazi et al. [11], the formally verified reliability result of Theorem 36.5 is sound and obtained through a rigorous reasoning process during the mechanization of their proofs. To the best of our knowledge, the benefits mentioned earlier are not shared by any other computer-based railway system reliability analysis approach.

36.7 Conclusion

The safe operation of high-speed trains has been the highest priority of railway companies around the world. However, their reliability analysis has been carried out using informal system analysis methods, such as simulation or paper-and-pencil, which do not ensure accurate results. The accuracy of the reliability results for railway systems is very critical since even minor flaws in the analysis could lead to the loss of many human lives or cause heavy financial setbacks. In order to achieve this goal and overcome the inaccuracy limitations of the traditional reliability analysis techniques, we propose to build upon the recent formalization of RBDs to formally reason about the reliability of high-speed railway systems using HOL theorem proving. As an application, we formally verified the reliability expressions of the a railway system designed for the Italian high-speed railways.

References

1. Liu, J., Li, S., Jiang, Y., and Krishnamurthy, M.: Reliability Evaluating for Traction Drive System of High-speed Electrical Multiple Units. In: *Transportation Electrification Conference and Expo*, Institute of Electrical and Electronics Engineers, Piscataway, NJ (2013) 1–6.
2. Trivedi, K. S.: *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. John Wiley & Sons, Hoboken, NJ (2002).
3. ReliaSoft: http://www.reliasoft.com/ (2014).
4. ASENT: https://www.raytheoneagle.com/asent/rbd.htm (2016).
5. Robidoux, R., Xu, H., Xing, L., Zhou, M.: Automated modeling of dynamic reliability block diagrams using colored Petri nets. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* **40** (2) (2010) 337–351.
6. Norman, G., and Parker, D.: Quantitative Verification: Formal Guarantees for Timeliness, Reliability and Performance. The London Mathematical Society and the Smith Institute (2014) http://www.prismmodelchecker.org/papers/lms-qv.pdf.
7. Ahmad, W., Hasan, O., Tahar, S., and Hamdi, M. S.: Towards the formal reliability analysis of oil and gas pipelines. *In: Intelligent Computer Mathematics*. Volume 8543 of LNCS. Springer, Berlin (2014) 30–44.
8. Gordon, M. J. C.: Mechanizing programming logics in higher-order logic. In: *Current Trends in Hardware Verification and Automated Theorem Proving*. Springer, Berlin (1989) 387–439.
9. Ahmed, W., Hasan, O., and Tahar, S.: Formalization of reliability block diagrams in higher order logic. *Journal of Applied Logic* **18** (2016) 19–41.
10. Mhamdi, T., Hasan, O., and Tahar, S.: On the formalization of the Lebesgue integration theory in HOL. In: *Interactive Theorem Proving*. Volume 6172 of LNCS. Springer, Berlin (2011) 387–402.

11. Dazi, G., Savio, S., and Firpo, P.: Estimate of components reliability and maintenance strategies impact on trains delay. In: *European Conference on Modelling and Simulation*. (2007) 447–452.
12. DNV-GL: http://www.dnvgl.com/oilgas/ (2015).
13. Signoret, J. P., Dutuit, Y., Cacheux, P. J., Folleau, C., Collas, S., and Thomas, P.: Make your Petri nets understandable: Reliability block diagrams driven Petri nets. *Reliability Engineering and System Safety* **113** (2013) 61–75.
14. PRISM: www.cs.bham.ac.uk/ dxp/prism (2015).
15. Herbert, L. T., and Hansen, Z. N. L.: Restructuring of workflows to minimise errors via stochastic model checking: An automated evolutionary approach. *Reliability Engineering and System Safety* **145** (2016) 351–365.
16. Lu, Y., Peng, Z., Miller, A. A., Zhao, T., and Johnson, C. W.: How reliable is satellite navigation for aviation? Checking availability properties with probabilistic verification. *Reliability Engineering and System Safety* **144** (2015) 95–116.
17. Hurd, J.: *Formal Verification of Probabilistic Algorithms*. PhD Thesis, University of Cambridge, Cambridge, UK (2002).
18. Hölzl, J., and Heller, A.: Three chapters of measure theory in Isabelle/HOL. In: *Interactive Theorem Proving*. Volume 6172 of LNCS. Springer, Berlin (2011) 135–151.
19. Hasan, O., and Tahar, S.: Formal verification of tail distribution bounds in the HOL theorem prover. Mathematical *Methods in the Applied Sciences* **32** (4) (2009) 480–504.
20. Hasan, O., and Tahar, S.: Formalization of the standard uniform random variable. *Theoretical Computer Science* **382** (1) (2007) 71–83.
21. Hasan, O., and Tahar, S.: Formalization of continuous probability distributions. In: *Automated Deduction*. Volume 4603 of LNCS. Springer, Berlin (2007) 2–18.
22. Hasan, O., Tahar, S., and Abbasi, N.: Formal reliability analysis using theorem proving. *IEEE Transactions on Computers* **59** (5) (2010) 579–592.
23. Abbasi, N., Hasan, O., and Tahar, S.: An approach for lifetime reliability analysis using theorem proving. *Journal of Computer and System Sciences* **80** (2) (2014) 323–345.
24. Ahmed, W., Hasan, O., and Tahar, S.: Formal reliability analysis of wireless sensor network data transport protocols using HOL. In: *Wireless and Mobile Computing, Networking and Communications*, Institute of Electrical and Electronics Engineers, Piscataway, NJ (2015) 217–224.
25. Ahmed, W., Hasan, O., and Tahar, S.: Towards formal reliability analysis of logistics service supply chains using theorem proving. In: *Implementation of Logics* Volume 40, EPiC Series in Computing. Suva, Fiji (2015) 111–121.
26. Ahmad, W., Hasan, O., Tahar, S., and Hamdi, M. S.: Formal reliability analysis of oil and gas pipelines. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* (2017) 1–15.
27. Ahmed, W., and Hasan, O.: Towards the formal fault tree analysis using theorem proving. In: *Intelligent Computer Mathematics*. Volume 9150 of LNAI. Springer, Berlin (2015) 39–54.
28. Ahmad, W., and Hasan, O.: Formal availability analysis using theorem proving. In: I*nternational Conference on Formal Engineering Methods*. Volume 10009 of LNCS. Springer, Berlin (2016) 1–16.
29. Harrison, J.: *Formalized Mathematics*. Technical Report 36, Turku Centre for Computer Science, Turku (1996).
30. Gordon, M. J., and Melham, T. F.: *Introduction to Hol: A Theorem Proving Environment for Higher Order Logic*. Cambridge University Press, Cambridge, UK (1993).
31. Church, A.: A formulation of the simple theory of types. *Journal of Symbolic Logic* **5** (1940) 56–68.
32. Milner, R.: A theory of type polymorphism in programming. *Journal of Computer and System Sciences* 17 (1977) 348–375.
33. Narasimhan, K.: Reliability engineering: Theory and practice. *The TQM Magazine* **17** (2) (2005) 209–210.
34. Ahmed, W.: *Formal Reliability Analysis of Railway Systems using Theorem Proving Technique* http:// save.seecs.nust.edu.pk/train/ (2016).