Towards the Formal Reliability Analysis of Oil and Gas Pipelines

Waqar Ahmad[1], Osman Hasan[1],
Sofiène Tahar[2], and Mohammad Salah Hamdi[3]

[1] School of Electrical Engineering and Computer Science (SEECS)
National University of Sciences and Technology (NUST)
Islamabad, Pakistan
{12phdwahmad,osman.hasan}@seecs.nust.edu.pk
[2] Electrical and Computer Engineering Department
Concordia University, Montreal, Canada
tahar@ece.concordia.ca
[3] Information Systems Department
Ahmed Bin Mohammed Military College, Doha, Qatar
mshamdi@abmmc.edu.qa

Abstract. It is customary to assess the reliability of underground oil and gas pipelines in the presence of excessive loading and corrosion effects to ensure a leak-free transport of hazardous materials. The main idea behind this reliability analysis is to model the given pipeline system as a Reliability Block Diagram (RBD) of segments such that the reliability of an individual pipeline segment can be represented by a random variable. Traditionally, computer simulation is used to perform this reliability analysis but it provides approximate results and requires an enormous amount of CPU time for attaining reasonable estimates. Due to its approximate nature, simulation is not very suitable for analyzing safety-critical systems like oil and gas pipelines, where even minor analysis flaws may result in catastrophic consequences. As an accurate alternative, we propose to use a higher-order-logic theorem prover (HOL) for the reliability analysis of pipelines. As a first step towards this idea, this paper provides a higher-order-logic formalization of reliability and the series RBD using the HOL theorem prover. For illustration, we present the formal analysis of a simple pipeline that can be modeled as a series RBD of segments with exponentially distributed failure times.

Keywords: Reliability Block Diagrams, Formal Methods, Theorem Proving, Oil and Gas pipeline.

1 Introduction

On April 20, 2010, methane gas leakage on the Deepwater Horizon oil rig operated by Transocean, a subcontractor of British Petroleum (BP), caused a big explosion [1]. This leakage not only killed 11 workers instantly but destroyed and sank the rig, and caused millions of gallons of oil to pour into the Gulf of

The original version of this chapter was revised. The spelling of the author Waqar Ahmad has been corrected. The erratum to this chapter is available at DOI: 10.1007/978-3-319-08434-3_39

S.M. Watt et al. (Eds.): CICM 2014, LNAI 8543, pp. 30–44, 2014.

Mexico. The gushing well, about a mile under the sea, was finally brought under control after more than three months of frenetic attempts. The spill, which is considered to be the largest accidental marine oil spill in the history of the petroleum industry, caused extensive damage to marine and wildlife habitats as well as the Gulf's fishing and tourism industries and its impact still continues. Just like the BP pipeline, there are tens of thousands of miles long oil and gas pipelines around the world. All of these pipelines are aging and are becoming more and more susceptible to failures, which may lead to disasters like the BP one. Hence, it is very important to do rigorous reliability analysis of oil and gas pipelines to detect and rectify potential problems.

The reliability analysis of a pipeline system involves a three-step process: (i) partitioning the given pipeline into segments and constructing its equivalent reliability block diagram (RBD), (ii) assessing the reliability of the individual segments and (iii) evaluating the reliability of the complete pipeline system based on the RBD and the reliability of its individual segments. The reliability of an individual segment is usually expressed in terms of its failure rate λ and a random variable, like exponential [2] or Weibull random variable [3], which models the failure time. A single oil or gas pipeline can be simply modeled as a series RBD [2]. However, in many cases, these pipeline systems have either reserved components or subsystems and such pipeline systems exhibit a combination of series and parallel RBDs [4].

The reliability analysis of oil and gas pipelines has predominantly been accomplished by first gathering data from in-line inspection tools to detect cracks, corrosion or damage [5, 6]. This information is then manipulated using the paper-and-pencil based analytical analysis and computer simulations to deliver diagnostics and insightful pipeline integrity reports (e.g. [2, 4, 7]). However, due to the complex nature of large pipeline system analysis, paper-and-pencil proof methods are error prone and the exhaustive testing of all possible system behaviors using simulation is almost impossible. Thus, these traditional analysis techniques cannot guarantee accurate results, which is a severe limitation in the case of oil and gas pipelines as an uncaught system bug may endanger human and animal life or lead to a significant financial loss.

The inaccuracy limitations of traditional analysis techniques can be overcome by using formal methods [8], which use computerized mathematical reasoning to precisely model the system's intended behavior and to provide irrefutable proof that a system satisfies its requirements. Both model checking and theorem proving have been successfully used for the precise probabilistic analysis of a broad range of systems (e.g. [9–13]). However, to the best of our knowledge, no formal analysis approach has been used for the reliability analysis of oil and gas pipelines so far. The foremost requirement for conducting the formal reliability analysis of underground oil and gas pipelines is the ability to formalize RBDs recursively and continuous random variables. Model checking is a state-based formal method technique. The inherent limitations of model checking is the state-space explosion problem and the inability to model complex datatypes such as trees, lists and recursive definitions [14]. On the other hand, higher-order logic [15] is a

system of deduction with a precise semantics and can be used to formally model any system that can be described mathematically including recursive definitions, random variables, RBDs, and continuous components. Similarly, interactive theorem provers are computer based formal reasoning tools that allow us to verify higher-order-logic properties under user guidance. Higher-order-logic theorem provers can be used to reason about recursive definitions using induction methods [16]. Thus, higher-order-logic theorem proving can be used to conduct the formal analysis of oil and gas pipelines.

A number of higher-order-logic formalizations of probability theory are available in higher-order logic (e.g. [17–19]). Hurd's formalization of probability theory [17] has been utilized to verify sampling algorithms of a number of commonly used discrete [17] and continuous random variables [20] based on their probabilistic and statistical properties [21, 22]. Moreover, this formalization has been used to conduct the reliability analysis of a number of applications, such as memory arrays [23], soft errors [24] and electronic components [25]. 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 pipelines, multiple continuous random variables are required. The recent formalizations of probability theory by Mhamdi [18] and Hölzl [19] are based on extended real numbers (including $\pm\infty$) and provide the formalization of Lebesgue integral for reasoning about advanced statistical properties. These theories also allow using any arbitrary probability space that is a subset of the universe and thus are more flexible than Hurd's formalization. However, to the best of our knowledge, these foundational theories have not been used to formalize neither reliability and RBDs nor continuous random variables so far.

In this paper, we use Mhamdi's formalization of probability theory [18], which is available in the HOL theorem prover [26], to formalize reliability and the commonly used series RBD, where its individual segments are modeled as random variables. Our formalization includes various formally verified properties of reliability and series RBD that facilitate formal reasoning about the reliability of some simple pipelines using a theorem prover. To analyze more realistic models of pipelines, it is required to formalize other RBDs, such as parallel, series-parallel and parallel-series [27]. In order to illustrate the utilization and effectiveness of the proposed idea, we utilize the above mentioned formalization to analyze a simple pipeline that can be modeled as a series RBD with an exponential failure time for individual segments.

2 Preliminaries

In this section, we give a brief introduction to theorem proving in general and the HOL theorem prover in particular. The intent is to introduce the main ideas behind this technique to facilitate the understanding of the paper for the reliability analysis community. We also summarize Mhamdi's formalization of probability theory [18] in this section.

2.1 Theorem Proving

Theorem proving [28] is a widely used formal verification technique. The system that needs to be analysed is mathematically modelled in an appropriate logic and the properties of interest are verified using computer based formal tools. The use of formal logics as a modelling medium makes theorem proving a very flexible verification technique as it is possible to formally verify any system that can be described mathematically. 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 proved axioms 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]. For instance, first-order logic [30] utilizes the propositional calculus and terms (constants, function names and free variables) and is semi-decidable. A number of sound and complete first-order logic automated reasoners are available that enable completely automated proofs. More expressive logics, such as higher-order logic [15], can be used to model a wider range of problems than first-order logic, but theorem proving for these logics cannot be fully automated and thus involves user interaction to guide the proof tools. For reliability analysis of pipelines, we need to formalize (mathematically model) random variables as functions and their distribution properties are verified by quantifying over random variable functions. Henceforth, first-order logic does not support such formalization and we need to use higher-order logic to formalize the foundations of reliability analysis of pipelines.

2.2 HOL Theorem Prover

HOL is an interactive theorem prover developed at the University of Cambridge, UK, for conducting proofs in higher-order logic. It utilizes the simple type theory of Church [31] along with Hindley-Milner polymorphism [32] to implement higher-order logic. 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 5 basic axioms and 8 primitive inference rules, which are implemented as ML functions. 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.

We utilized the HOL theories of Booleans, lists, sets, positive integers, *real* numbers, measure and probability in our work. 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 1 provides the mathematical interpretations of some frequently used HOL symbols and functions, which are inherited from existing HOL theories, in this paper.

Table 1. HOL Symbols and Functions

HOL Symbol	Standard Symbol	Meaning
$\wedge$	*and*	Logical *and*
$\vee$	*or*	Logical *or*
$\neg$	*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
λx.t	$\lambda x.t$	Function that maps x to $t(x)$
SUC n	$n+1$	Successor of a *num*
lim(λn.f(n))	$\lim_{n\to\infty} f(n)$	Limit of a *real* sequence f

2.3 Probability Theory and Random Variables 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 Mhamdi's formalization of probability theory [18], given a probability space p, the functions `space` and `subsets` return the corresponding Ω and Σ, 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.

Mathematically, a random variable is a measurable function between a probability space and a measurable space. A measurable space refers to a pair $(S, \mathcal{A})$, where S denotes a set and $\mathcal{A}$ represents a nonempty collection of sub-sets of S. Now, if S is a set with finite elements, then the corresponding random variable is termed as a discrete random variable and else it is called a continuous one. The probability that a random variable X is less than or equal to some value x, $Pr(X \leq x)$ is called the cumulative distribution function (CDF) and it characterizes the distribution of both discrete and continuous random variables. Mhamdi's formalization of probability theory [18] also includes the formalization of random variables and the formal verification of some of their classical properties using the HOL theorem prover.

3 Reliability

In reliability theory [27], reliability $R(t)$ of a system or component is defined as the probability that it performs its intended function until some time t.

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

where $F_X(t)$ is the CDF. The random variable X, in the above definition, models the time to failure of the system. Usually, this time to failure is modeled by the exponential random variable with parameter λ that represents the failure rate of the system. Now, the CDF can be modeled in HOL as follows:

Definition 1: *Cumulative Distributive Function*

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

where p represents the probability space, X is the random variable and x represents a *real* number. The function `Normal` converts a *real* number to its corresponding value in the *extended−real* data-type, i.e, the *real* data-type including the positive and negative infinity. The function `distribution` accepts a probability space p, a random variable X and a set and returns the probability of X acquiring all the values of the given set in the probability space p. Now, Definition 1 can be used to formalize the reliability definition, given in Equation 1, as follows:

Definition 2: *Reliability*

```
⊢ ∀ p X x. Reliability p X x = 1 - CDF p X x
```

We used the above mentioned formal definition of reliability to formal verify some of the classical properties of reliability in HOL. The first property in this regard relates to the fact that the reliability of a good component is 1, i.e., maximum, prior to its operation, i.e., at time 0. This property has been verified in HOL as the following theorem.

Theorem 1: *Maximum Reliability*

```
⊢ ∀ p X. prob_space p ∧ (events p = POW (p_space p)) ∧
        (∀ y. X y ≠ NegInf ∧ X y ≠ PosInf) ∧
        (∀ z. 0 ≤ z ⇒ (λx. CDF p X x) contl z) ∧
        (∀ x. Normal 0 ≤ X x) ⇒
        (Reliability p X 0 = 1)
```

The first two assumptions of the above theorem ensure that the variable p represents a valid probability space based on the formalization of Mhamdi's probability theory [18]. The third assumption constraints the random variable to be well-defined, i.e., it cannot acquire negative or positive infinity values. The fourth assumption states that the CDF of the random variable X is a continuous function, which means that X is a continuous random variable. This assumption utilizes the HOL function `contl`, which accepts a lambda abstraction function and a real value and ensures that the function is continuous at the given value. The last assumption ensures that the random variable X can acquire positive values only since in the case of reliability this random variable always models time, which cannot be negative. The conclusion of the theorem represents our desired property that reliability at *time=0* is *1*.

The proof of the Theorem 1 exploits some basic probability theory axioms and the following property according to which the probability of a continous random variable at a point is zero.

The second main characteristic of the reliability function is its decreasing monotonicity, which is verified as the following theorem in HOL:

Theorem 2: *Reliability is a Monotone Function*

```
⊢ ∀ p X a b. prob_space p ∧ (events p = POW (p_space p)) ∧
       (∀ y. X y ≠ NegInf ∧ X y ≠ PosInf) ∧
       (∀ x. Normal 0 ≤ X x) ∧ a ≤ b ⇒
       (Reliability p X (b)) ≤ (Reliability p X (a))
```

The assumptions of this theorem are the same as the ones used for Theorem 1 except the last assumption, which describes the relationship between variables a and b. The above property clearly indicates that the reliability cannot increase with the passage of time.

The formal reasoning about the proof of Theorem 2 involves some basic axioms of probability theory and a property that the CDF is a monotonically increasing function.

Finally, we verified that the reliability tends to 0 as the time approaches infinity. This property is verified under the same assumptions that are used for Theorem 1.

Theorem 3: *Reliability Tends to Zero As Time Approaches Infinity*

```
⊢ ∀ p X. prob_space p ∧ (events p = POW (p_space p)) ∧
   (∀ y. X y ≠ NegInf ∧ X y ≠ PosInf) ∧ (∀ x. Normal 0 ≤ X x) ⇒
       (lim (λn. Reliability p X (&n)) = 0)
```

The HOL function `lim` models the limit of a real sequence. The proof of Theorem 3 primarily uses the fact that the CDF approches to 1 as its argument approaches infinity.

These three theorems completely characterize the behavior of the reliability function on the positive real axis as the argument of the reliability is time and thus cannot be negative. The formal verification of these properties based on our definition ensure its correctness. Moreover, these formally verified properties also facilitate formal reasoning about reliability of systems, as will be demonstrated in Section 5 of this paper. The proof details about these properties can be obtained from our proof script [33].

4 Formalization of Series Reliability Block Diagram

In a serially connected system [27], depicted in Figure 1, the reliability of the complete system mainly depends upon the failure of a single component that has the minimum reliability among all the components of the system. In other words, the system stops functioning if any one of its component fails. Thus, the operation of such a system is termed as reliable at any time t, if all of its components are functioning reliably at this time t. If the event $A_i(t)$ represents the reliable functioning of the i^{th} component of a serially connected system

with N components at time t then the overall reliability of the system can be mathematically expressed as [27]:

$$R_{series}(t) = Pr(A_1(t) \cap A_2(t) \cap A_3(t) \cdots \cap A_N(t)) \tag{2}$$

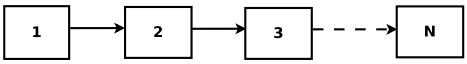

Fig. 1. System with a Series Connection of Components

Using the assumption of mutual independence of individual reliability events of a series system [27], the above equation can be simplified as:

$$R_{series}(t) = \prod_{i=1}^{N} R_i(t) \tag{3}$$

Moreover, an intrinsic property of a series system is that its overall reliability is always less than or equal to the reliability of the sub-component with the least reliability.

$$R_{series}(t) \leq min(R_i(t)) \tag{4}$$

We proceed with the formalization of the series RBD by first formalizing the notion of mutual independence of more than two random variables, which is one of the most essential prerequisites for reasoning about the simplified expressions for RBD. Two events A and B are termed as mutually independent iff $Pr(A \cap B) = Pr(A)Pr(B)$. All the events involved in reliability modeling are generally assumed to be mutually independent. Since we often tackle the reliability assessment of systems with more than two components, we formalize the mutual independence of a list of random variables in this paper as follows:

Definition 3: *Mutual Independence of Events*

```
⊢ ∀ p L. mutual_indep p L =
  ∀ L1 n. PERM L L1 ∧ 2 ≤ n ∧ n ≤ LENGTH L ⇒
      prob p (inter_set p (TAKE n L1)) =
      list_prod (list_prob p (TAKE n L1))
```

The function `mutual_indep` takes a list of events or sets L along with the probability space p as input and returns True if the given list of events are mutually independent in p. The formal definitions for the HOL functions used in the above definition are given in Table 1. The predicate `PERM` ensures that its two list arguments form a permutation of one another, the function `LENGTH` returns the length of a list, the function `TAKE` returns a list that contains the first n elements of its argument list, the function `inter_set` performs the intersection of all the sets in a list of sets and returns the probability space in case of an

empty list argument, the function `list_prob` returns a list of probabilities associated with the given list of events in the given probability space and the function `list_prod` recursively multiplies all the elements of its argument list of real numbers. Thus, using these functions the function `mutual_indep` ensures that for any 2 or more elements n, taken in any order, of the given list of events L, the property $Pr(\bigcap_{i=0}^{n} L_i) = \prod_{i=0}^{n} Pr(L_i)$ holds.

Table 2. HOL Functions used in Definition 3

Function Name	HOL Definition
PERM	⊢ ∀ L1 L2. PERM L1 L2 = ∀ x. FILTER ($= x) L1 = FILTER ($= x)L2
LENGTH	⊢ (LENGTH [] = 0) ∧ ∀ h t. LENGTH (h::t) = SUC (LENGTH t)
TAKE	⊢ (∀ n. TAKE n [] = []) ∧ ∀ n x xs. TAKE n (x::xs) = if n = 0 then [] else x::TAKE (n - 1) xs
inter_set	⊢ (∀ p. inter_set p [] = p_space p) ∧ ∀ p h t. inter_set p (h::t) = h ∩ inter_set p t
list_prod	⊢ (∀ list_prod [] = 1) ∧ ∀ h t. list_prod (h::t) = h * list_prod t
list_prob	⊢ (∀ p. list_prob p [] = []) ∧ ∀ p h t. list_prob p (h::t) = prob p (h ∩ p_space p) * list_prob p t
min	⊢ ∀ x y. min x y = if x ≤ y then x else y
min_rel	⊢ (∀ f. min_rel f [] = 1) ∧ ∀ f h t. min_rel f (h::t) = min (f h) (min_rel f t)

Next, we propose to formalize the RBDs in this paper by using a list of events, where each event models the proper functioning of a single component at a given time based on the corresponding random variable. This list of events can be modeled as follows:

Definition 4: *Reliability Event List*

```
⊢ ∀ p x. rel_event_list p [] x = [] ∧
  ∀ p x h t. rel_event_list p (h::t) x =
PREIMAGE h {y | Normal x < y} ∩ p_space p :: rel_event_list p t x
```

The function `rel_event_list` accepts a list of random variables, representing the time to failure of individual components of the system, and a *real* number x, which represents the time index where the reliability is desired, and returns a list of sets corresponding to the events that the individual components are functioning properly at the given time x. This list of events can be manipulated, based on the structure of the complete system, to formalize various RBDs.

Similarly, the individual reliabilities of a list of random variables can be modeled as the following recursive function:

Definition 5: *Reliability of a List of Random Variables*

```
⊢ ∀ p x . rel_list p [] x = [] ∧
  ∀ p h t x. rel_list p (h::t) x =
  Reliability p h x :: rel_list p t x
```

The function `rel_list` takes a list of random variables and a *real* number x, which represents the time index where the reliability is desired, and returns a list of the corresponding reliabilities at the given time x. It is important to note that all the above mentioned definitions are generic enough to represent the behavior of any RBD, like series, parallel, series-parallel and parallel-series.

Now, using Equation (2), the reliability of a serially connected structure can be defined as:

Definition 6: *System with a Series Connection of Components*

```
⊢ ∀ p L. rel_series p L = prob p (inter_set p L)
```

The function `rel_series` takes a list of random variables L, representing the failure times of the individual components of the system, and a probability space p as input and returns the intersection of all the events corresponding to the reliable functioning of these components using the function `inter_set`, given in Table 2. Based on this definition, we formally verified the result of Equation (2) as follows:

Theorem 4: *Reliability of a System with Series Connections*

```
⊢ ∀ p L x. prob_space p ∧ (events p = POW (p_space p)) ∧
     0 ≤ x ∧ 2 ≤ LENGTH (rel_event_list p L x) ∧
   mutual_indep p (rel_event_list p L x) ⇒
 (rel_series p (rel_event_list p L x) = list_prod (rel_list p L x))
```

The first two assumptions ensure that p is a valid probability space based on Mhamdi's probability theory formalization [18]. The next one ensures that the variable x, which models time, is always greater than or equal to 0. The next two assumptions of the above theorem guarantee that we have a list of at least two mutually exclusive random variables (or a system with two or more components). The conclusion of the theorem represents Equation (2) using Definitions 4 and 6. The proof of Theorem 4 involves various probability theory axioms, the mutual independence of events and the fact that the probability of any event that is in the returned list from the function `rel_event_list` is equivalent to its reliability. More proof details can be obtained from our proof script [33].

Similarly, we verified Equation (4) as the following theorem in HOL:

Theorem 5: *Reliability of a System depends upon the minimum reliability of the connected components*

```
⊢ ∀ p L x. prob_space p ∧ (events p = POW (p_space p)) ∧
     0 ≤ x ∧ 2 ≤ LENGTH (rel_event_list p L x) ∧
     mutual_indep p (rel_event_list p L x) ⇒
      (rel_series p (rel_event_list p L x) ≤
        min_rel (λ L. Reliability p L x) L)
```

The proof of the Theorem 5 uses several probability theory axioms and the fact that any subset of a mutually independent set is also mutually independent.

The definitions, presented in this section, can be used to model parallel RBD [27] and formally verify the corresponding simplified reliability relationships as well. The major difference would be the replacement of the function `inter_set` in Definition 6 by a function that returns the union of a given list of events.

5 Reliability Analysis of a Pipeline System

A typical oil and gas pipeline can be partitioned into a series connection of N segments, where these segments may be classified based on their individual failure times. For example, a 60 segment pipeline is analyzed in [2] under the assumption that the segments, which exhibit exponentially distributed failure rates, can be sub-divided into 3 categories according to their failure rates (λ), i.e., 30 segments with $\lambda = 0.0025$, 20 segments with $\lambda = 0.0023$ and 10 segments with $\lambda = 0.015$. The proposed approach for reliability analysis of pipelines allows us to formally verify generic expressions involving any number of segments and arbitrary failure rates. In this section, we formally verify the reliability of a simple pipeline, depicted in Figure 2, with N segments having arbitrary exponentially distributed failure times.

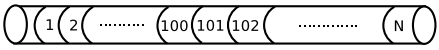

Fig. 2. A Simple Pipeline

We proceed with the formal reliability analysis of the pipeline, shown in Figure 2, by formalizing the exponential random variable in HOL.

Definition 7: *Exponential Distribution Function*

```
⊢ ∀ p X l. exp_dist p X l =
   ∀ x. (CDF p X x = if 0 ≤ x then 1 - exp (-l * x) else 0)
```

The predicate `exp_dist` ensures that the random variable X exhibits the CDF of an exponential random variable in probability space p with failure rate l. We classify a list of exponentially distributed random variables based on this definition as follows:

Definition 8: *List of Exponential Distribution Functions*

```
⊢ ∀ p L. list_exp p [] L = T ∧
  ∀ p h t L. list_exp p (h::t) L =
  exp_dist p (HD L) h ∧ list_exp p t (TL L)
```

The `list_exp` function accepts a list of failure rates, a list of random variables L and a probability space p. It guarantees that all elements of the list L are

exponentially distributed with 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.

Next, we model the pipeline, shown in Figure 2, as a series RBD as follows:

Definition 9: *Reliability of Series Pipeline System*

```
⊢ ∀ p L . pipeline p L = rel_series p L
```

Now, we can use Definition 8 to guarantee that the random variable list argument of the function `pipeline` contains exponential random variables only and thus verify the following simplified expression for the pipeline reliability.

Theorem 6: *Series Pipeline System*

```
⊢ ∀ p L x C. prob_space p ∧ (events p = POW (p_space p)) ∧
     0 ≤ x ∧ 2 ≤ LENGTH (rel_event_list p L x) ∧
     mutual_indep p (rel_event_list p L x) ∧
      list_exp p C L ∧ (LENGTH C = LENGTH L) ⇒
      (pipeline p (rel_event_list p L x) = exp (-list_sum C * x))
```

The first five assumptions are the same as the ones used in Theorem 5. The sixth assumption `list_exp p C L` ensures that the list of random variable L contains all exponential random variables with corresponding failure rates given in list C. The next assumptions guarantees that the lengths of the two lists L and C are the same. While the conclusion of Theorem 6 represents desired reliability relationship for the given pipeline model. Here the function `list_sum` recursively adds the elements of its list argument and is used to add the failure rates of all exponentially distributed random variables, which are in turn used to model the individual segments of the series RBD of the pipeline. The proof of Theorem 6 is based on Theorem 4 and some properties of the exponential function `exp`. The reasoning was very straightforward (about 100lines of HOL code) compared to the reasoning for the verification of Theorem 4 [33], which involved probability-theoretic guidance. This fact illustrates the usefulness of our core formalization for conducting the reliability analysis of pipelines.

The distinguishing features of this formally verified result include its generic nature, i.e., all the variables are universally quantified and thus can be specialized to obtain the reliability of the given pipeline for any given parameters, and its guaranteed correctness due to the involvement of a sound theorem prover in its verification, which ensures that all the required assumptions for the validity of the result are accompanying the theorem. Another point worth mentioning is that the individual failure rates of the pipeline segments can be easily provided to the above theorem in the form of a list, i.e., C. The above mentioned benefits are not shared by any other computer based reliability analysis approach for oil and gas pipelines and thus clearly indicate the usefulness of the proposed approach.

6 Conclusions

Probabilistic analysis techniques have been widely utilized during the last two decades to assess the reliability of oil and gas pipelines. However, all of these probability theoretic approaches have been utilized using informal system analysis methods, like simulation or paper-and-pencil based analytical methods, and thus do not ensure accurate results. The precision of results is very important in the area of oil and gas pipeline condition assessment since even minor flaws in the analysis could result in the loss of human lives or heavy damages to the environment. In order to achieve this goal and overcome the inaccuracy limitation of the traditional probabilistic analysis techniques, we propose to build upon our proposed formalization of RBDs to formally reason about the reliability of oil and gas pipelines using higher-order-logic theorem proving.

Building upon the results presented in this paper, the formalization of other commonly used RBDs, including parallel, series-parallel and parallel-series, and the Weibull random variable is underway. These advanced concepts are widely used in the reliability analysis of pipelines. However, their formalization requires some advanced properties of probability theory. For example, for formalizing the reliability block diagrams of the series-parallel and parallel-series structures, we need to first formally verify the principle of inclusion exclusion [34]. We also plan to formalize the underlying theories to reason about more realistic series pipeline systems, such as multi-state variable piping systems, where each subcomponent of the pipeline system consists of many irreversible states from good to worst. We also plan to investigate artificial neural networks in conjunction with theorem proving to develop a hybrid semi-automatic pipeline reliability analysis framework. Besides the pipeline reliability analysis, the formalized reliability theory foundation presented in this paper, may be used for the reliability analysis of a number of other applications, including hardware and software systems.

Acknowledgments. This publication was made possible by NPRP grant # [5 - 813 - 1 134] from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the author[s].

References

1. BP Leak the World's Worst Accidental Oil Spill, London Telegraph (August 03, 2010), http://www.telegraph.co.uk/finance/newsbysector/energy/oilandgas/7924009/bp-leak-the-worlds-worst-accidental-oil-spill.html (2014)
2. Zhang, Z., Shao, B.: Reliability Evaluation of Different Pipe Section in Different Period. In: Service Operations and Logistics, and Informatics, pp. 1779–1782. IEEE (2008)
3. Kolowrocki, K.: Reliability and Risk Analysis of Multi-State Systems With Degrading Components. Electronic Journal of International Group on Reliability 2(1), 86–104 (2009)

4. Soszynska, J.: Reliability and Risk Evaluation of a Port Oil Pipeline Transportation System in Variable Operation conditions. International Journal of Pressure Vessels and Piping 87(2-3), 81–87 (2010)
5. Pipeline Integrity Solution GE-Energy (2014), http://www.ge-energy.com/products_and_services/services/pipeline_integrity_services/
6. Pipecheck - Pipeline Integrity Assessment Software (2014), http://www.creaform3d.com/en/ndt-solutions/pipecheck-damage-assessment-software
7. Pandey, D.: Probabilistic Models for Condition Assessment of Oil and Gas Pipelines. Independent Nondestructive Testing and Evaluation International 31(3), 349–358 (1998)
8. Boca, P., Bowen, J., Siddiqi, J.: Formal Methods: State of the Art and New Directions. Springer (2009)
9. Hasan, O., Tahar, S.: Performance Analysis of ARQ Protocols using a Theorem Prover. In: International Symposium on Performance Analysis of Systems and Software, pp. 85–94. IEEE Computer Society (2008)
10. Kwiatkowska, M., Norman, G., Parker, D.: Probabilistic Model Checking for Systems Biology. In: Symbolic Systems Biology, pp. 31–59. Jones and Bartlett (2010)
11. Elleuch, M., Hasan, O., Tahar, S., Abid, M.: Formal Analysis of a Scheduling Algorithm for Wireless Sensor Networks. In: Qin, S., Qiu, Z. (eds.) ICFEM 2011. LNCS, vol. 6991, pp. 388–403. Springer, Heidelberg (2011)
12. Hasan, O., Patel, J., Tahar, S.: Formal Reliability Analysis of Combinational Circuits using Theorem Proving. J. Applied Logic 9(1), 41–60 (2011)
13. Fruth, M.: Formal Methods for the Analysis of Wireless Network Protocols. PhD thesis, Oxford University, UK (2011)
14. Kaufman, M.: Some Key Research Problems in Automated Theorem Proving for Hardware and Software Verification. Revista de la Real Academia de Ciencias Exactas, Físicas y Naturales. Serie A: Matemáticas 98(1), 181 (2004)
15. Brown, C.: Automated Reasoning in Higher-order Logic. College Publications (2007)
16. Kapur, D., Subramaniam, M.: Lemma Discovery in Automating Induction. In: McRobbie, M.A., Slaney, J.K. (eds.) CADE 1996. LNCS, vol. 1104, pp. 538–552. Springer, Heidelberg (1996)
17. Hurd, J.: Formal Verification of Probabilistic Algorithms. PhD Thesis, University of Cambridge, UK (2002)
18. Mhamdi, T., Hasan, O., Tahar, S.: On the Formalization of the Lebesgue Integration Theory in HOL. In: Kaufmann, M., Paulson, L.C. (eds.) ITP 2010. LNCS, vol. 6172, pp. 387–402. Springer, Heidelberg (2010)
19. Hölzl, J., Heller, A.: Three Chapters of Measure Theory in Isabelle/HOL. In: van Eekelen, M., Geuvers, H., Schmaltz, J., Wiedijk, F. (eds.) ITP 2011. LNCS, vol. 6898, pp. 135–151. Springer, Heidelberg (2011)
20. Hasan, O., Tahar, S.: Formalization of Continuous Probability Distributions. In: Pfenning, F. (ed.) CADE 2007. LNCS (LNAI), vol. 4603, pp. 3–18. Springer, Heidelberg (2007)
21. Hasan, O., Tahar, S.: Verification of Tail Distribution Bounds in a Theorem Prover. In: Numerical Analysis and Applied Mathematics, vol. 936, pp. 259–262. American Institute of Physics (2007)
22. Hasan, O., Abbasi, N., Akbarpour, B., Tahar, S., Akbarpour, R.: Formal Reasoning about Expectation Properties for Continuous Random Variables. In: Cavalcanti, A., Dams, D.R. (eds.) FM 2009. LNCS, vol. 5850, pp. 435–450. Springer, Heidelberg (2009)

23. Hasan, O., Tahar, S., Abbasi, N.: Formal Reliability Analysis using Theorem Proving. IEEE Transactions on Computers 59(5), 579–592 (2010)
24. Abbasi, N., Hasan, O., Tahar, S.: Formal Analysis of Soft Errors using Theorem Proving. In: Symbolic Computation in Software Science. EPTCS, vol. 122, pp. 75–84 (2013)
25. Abbasi, N., Hasan, O., Tahar, S.: An Approach for Lifetime Reliability Analysis using Theorem Proving. Journal of Computer and System Sciences 80(2), 323–345 (2014)
26. Slind, K., Norrish, M.: A Brief Overview of HOL4. In: Mohamed, O.A., Muñoz, C., Tahar, S. (eds.) TPHOLs 2008. LNCS, vol. 5170, pp. 28–32. Springer, Heidelberg (2008)
27. Bilintion, R., Allan, R.: Reliability Evaluation of Engineering System. Springer (1992)
28. Gordon, M.: Mechanizing Programming Logics in Higher-Order Logic. In: Current Trends in Hardware Verification and Automated Theorem Proving, pp. 387–439. Springer (1989)
29. Harrison, J.: Formalized Mathematics. Technical Report 36, Turku Centre for Computer Science (1996)
30. Fitting, M.: First-Order Logic and Automated Theorem Proving. Springer (1996)
31. Church, A.: A Formulation of the Simple Theory of Types. Journal of Symbolic Logic 5, 56–68 (1940)
32. Milner, R.: A Theory of Type Polymorphism in Programming. Journal of Computer and System Sciences 17, 348–375 (1977)
33. Ahmad, W.: Formalization of Reliability Block Diagram for Analyzing Oil and Gas Pipelines (2014), `http://save.seecs.nust.edu.pk/wahmad/frsaogp.html`
34. Trivedi, K.S.: Probability and Statistics with Reliability, Queuing and Computer Science Applications, 2nd edn. John Wiley and Sons Ltd., Chichester (2002)