Formalization of Reliability Block Diagrams in Higher-order Logic

Waqar Ahmed\textsuperscript{a},\textsuperscript{*}, Osman Hasan\textsuperscript{a}, Sofiène Tahar\textsuperscript{b}

\textsuperscript{a}School of Electrical Engineering and Computer Science
National University of Sciences and Technology, Islamabad, Pakistan
\textsuperscript{b}Department of Electrical and Computer Engineering
Concordia University, Montreal, Canada

Abstract

Reliability Block Diagrams (RBDs) allow us to model the failure relationships of complex systems and their sub-components and are extensively used for system reliability, availability and maintainability analyses. Traditionally, these RBD-based analyses are done using paper-and-pencil proofs or computer simulations, which cannot ascertain absolute correctness due to their inherent limitations. As a complementary approach, we propose to use the higher-order-logic theorem prover HOL to conduct RBD-based analysis. For this purpose, we present a higher-order-logic formalization of commonly used RBD configurations, such as series, parallel, parallel-series and series-parallel, and the formal verification of their equivalent mathematical expressions. A distinguishing feature of the proposed RBD formalization is the ability to model nested RBD configurations, which are RBDs having blocks that also represent RBD configurations. This generality allows us to formally analyze the reliability of many real-world systems.

Keywords: Reliability Block Diagrams (RBDs), Higher-order Logic, Probability Theory, Virtualization Configuration, Virtual Data Centers

\textsuperscript{*}Waqar Ahmed

Email addresses: waqar.ahmad@seecs.nust.edu.pk (Waqar Ahmed), osman.hasan@seecs.nust.edu.pk (Osman Hasan), tahar@ece.concordia.ca (Sofiène Tahar)

1. Reliability Block Diagrams

Reliability Block Diagrams (RBDs) [1] are graphical structures consisting of blocks and connector lines. The blocks usually represent the system components and the connection of these components is described by the connector lines. The system is functional, if at least one path of properly functional components from input to output exists otherwise it fails.

An RBD construction can follow any of these three basic patterns of component connections: (i) series (ii) active redundancy or (iii) standby redundancy. In the series connection, shown in Figure 1(a), all components should be functional for the system to remain functional. Whereas, in an active redundancy all components in at least one of the redundant stages must be functioning in fully operational mode. The components in an active redundancy might be connected in a parallel structure (Figure 1(b)) or a combination of series and parallel structures as shown in Figures 1(c) and 1(d). In a standby redundancy, all components are not required to be active.

Three types of information are necessary to build the RBD of a given system: (i) functional interaction of the system components, (ii) reliability of each component, and (iii) mission times at which the reliability is desired. This information is then utilized by the design engineers to identify the appropriate RBD configuration (series, parallel or series-parallel) in order to determine the overall reliability of the given system. The detail about the commonly used RBD configurations and their corresponding mathematical expressions are as follows:

**Series Reliability Block Diagram.** The reliability of a system with components connected in series is considered to be reliable at time $t$ only if all of its components are functioning reliably at time $t$, as depicted in Figure 1(a). If $A_i(t)$ is a mutually independent event that 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 complete system can be expressed as [1]:

$$R_{series}(t) = Pr\left(\bigcap_{i=1}^{N} A_i(t)\right) = \prod_{i=1}^{N} R_i(t) \quad (1)$$

**Parallel Reliability Block Diagram.** The reliability of a system with parallel connected sub-modules, depicted in Figure 1(b), 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 $i^{th}$ component of a system with $M$ parallel components at time $t$, then the overall reliability of the system can be mathematically expressed as [1]:

$$R_{\text{parallel}}(t) = Pr(\bigcup_{i=1}^{M} A_i) = 1 - \prod_{i=1}^{M} (1 - R_i(t))$$

*Nested Reliability Block Diagrams.* Most safety-critical systems in the real-world contain many reserved stages for backup in order to ensure reliable op-
eration [2, 3]. If the components in these reserved subsystems are connected serially then the structure is called a parallel-series structure, as depicted in Figure 1(c). The parallel-series RBD is a nested form of series RBD in a parallel RBD configuration. 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:

$$R_{parallel-series}(t) = Pr\left(\bigcup_{i=1}^{M} \bigcap_{j=1}^{N} A_{ij}(t)\right) = 1 - \prod_{i=1}^{M} \left(1 - \prod_{j=1}^{N} (R_{ij}(t))\right)$$ (3)

Similarly, if in each serial stage the components are connected in parallel, then the configuration is termed as a series-parallel structure, shown in Figure 1(d). 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:

$$R_{series-parallel}(t) = Pr\left(\bigcap_{i=1}^{N} \bigcup_{j=1}^{M} A_{ij}(t)\right) = \prod_{i=1}^{N} (1 - \prod_{j=1}^{M} (1 - R_{ij}(t)))$$ (4)

In many cases, real-world systems involve sub-components, which themselves form a nested RBD configuration, as shown in Figure 1(e). Such systems can be modeled by nested RBD configurations. For instance, if a system and its components both are modeled by the series-parallel RBDs, then the complete system can be modeled by using a nested series-parallel RBD configuration. The reliability of this kind of nested series-parallel RBD can be expressed mathematically as follows:

$$R_{nested-series-parallel}(t) = Pr\left(\bigcap_{i=1}^{N} \bigcup_{j=1}^{M} \bigcap_{k=1}^{N} \bigcup_{l=1}^{M} A_{ijkl}(t)\right)$$

$$= \prod_{i=1}^{N} (1 - \prod_{j=1}^{M} (1 - \prod_{k=1}^{N} (1 - \prod_{l=1}^{M} (1 - (R_{ijkl}(t)))))))$$ (5)

where, $i$ and $j$ are the indices of the outer series-parallel RBD and the indices $k$ and $l$ refer to the reliability events corresponding to the inner sub-components of the system.
In the next section, we present the formalization of series, parallel and nested RBD configurations. These formalized configurations can then be used in turn to formally model systems behaviors in HOL4 and reason about their reliability, availability and maintainability characteristics.

2. Formalization of the Reliability Block Diagrams

The proposed formalization is primarily based on defining a new polymorphic datatype $rbd$ that encodes the notion of series and parallel configurations. Then a semantic function is defined on that $rbd$ datatype yielding an event for the corresponding RBD configuration. This semantic function allows us to verify the generic reliability expressions of the RBD configurations, that are described in the previous section, by utilizing the underlying probability theory within the sound core of the HOL4 theorem prover. Such a deep embedding considerably simplifies the RBD modeling approach, compared to our previous work [4] (shallow embedding), and also enables us to develop a framework that can deal with arbitrary levels of nested RBD configurations, which can be used to cater for a wide variety of real-world reliability analysis problems.

We verify the reliability expressions for the commonly used RBD configurations, as presented in Section 1, on reliability event lists, where a single event represents the scenario when the given system or component does not fail before a certain time:

**Definition 1:** $\forall \ p \ X \ t. \ rel\_event\ p\ X\ t = PREIMAGE\ X\ \{y \mid\ Normal\ t < y\} \cap\ p\_space\ p$

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 is derived by mapping the function $rel\_event$ on each element of the given random variable list in HOL4 as follows:
Definition 2: \[ \vdash \forall p \, L \, t. \]
\[ \text{rel \_ event \_ list } p \, L \, t = \text{MAP} (\lambda a. \text{rel \_ event } p \, a \, t) \, L \]

In [4], the series RBD function operates on a single dimension list of random variables, where each random variable in a list is associated with a block of the series RBD configuration. A major limitation of this modeling approach arises when dealing with nested RBD configurations, such as parallel-series and series-parallel, where the blocks themselves are modeled by RBD configurations. To cater for these RBD configurations, we are required to model a random variable that can incorporate the notion of multiple random variables. For instance, to formalize the parallel-series RBD configuration, we need to assign a random variable to each one of the serial stage such that the random variables associated to each parallel stage model all the random variables that are assigned to the corresponding components connected in a serial stage and thus making the RBD formalization of [4] challenging. In order to simplify the formalization of nested RBDs, we propose to distinguish the notion of random variable from the reliability event. We thus propose to formally verify generic RBD reliability relationships on reliability event lists. These formally verified expressions can then be used with the random variables corresponding to each component of the system for analyzing the reliability of systems that can be represented as nested RBDs.

We start the formalization process by type abbreviating the notion of event, which is essentially a set of observations with type \('a\text{-}\text{->}\text{bool}'\) as follows:

\[
\text{type\_abbrev ('event', '\text{``}'\text{'}\text{-}\text{->}\text{bool}')}
\]

We then define a recursive datatype \(rbd\) in the HOL4 system as follows:

\[
\text{Hol\_datatype \text{'}rbd\text{'} = \text{series of rbd list} |}
\]
\[
\text{parallel of rbd list} | \text{atomic of \text{'}a\text{\_}event'}
\]

An RBD can either be a series configuration, parallel configuration or an atomic event. The type constructors \text{series} and \text{parallel} recursively function on \(rbd\)-typed lists and thus enable us to deal with nested RBD configurations. The type constructor \text{atomic} is basically a typecasting operator between \text{event} and \(rbd\)-typed variables.

We define a semantic function \text{rbd\_struct} over the above-defined \(rbd\) datatype that can yield the corresponding event from the given RBD configuration as follows:
Definition 3: \[ \vdash (\forall p. \text{rbd}\_\text{struct} p (\text{series} []) = \text{p}\_\text{space} p) \land \\
(\forall x s x p. \\
\text{rbd}\_\text{struct} p (\text{series} (x::xs)) = \\
\text{rbd}\_\text{struct} p x \cap \text{rbd}\_\text{struct} p (\text{series} xs)) \land \\
(\forall x s x p. \\
\text{rbd}\_\text{struct} p (\text{parallel} []) = \{\}) \land \\
(\forall x s x p. \\
\text{rbd}\_\text{struct} p (\text{parallel} (x::xs)) = \\
\text{rbd}\_\text{struct} p x \cup \text{rbd}\_\text{struct} p (\text{parallel} xs)) \land \\
(\forall p a. \text{rbd}\_\text{struct} p (\text{atomic} a) = a) \]

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

Now using Definition 3, we can formally verify the reliability expression, given in Equation 1, for a series RBD configuration in HOL as follows:

\[ \begin{align*}
\text{Theorem 1:} \quad & \vdash \forall p L. \text{prob}\_\text{space} p \land \\
& \neg \text{NULL} L \land (\forall x'. \text{MEM} x' L \Rightarrow x' \in \text{events} p) \land \\
& \text{mutual}\_\text{indep} p L \Rightarrow \\
& (\text{prob} p (\text{rbd}\_\text{struct} p (\text{series} (\text{rbd}\_\text{list} L)))) = \\
& \text{list}\_\text{prod} (\text{list}\_\text{prob} p L))
\end{align*} \]

The first assumption, in Theorem 1, ensures that \( p \) is a valid probability space based on the probability theory in HOL4 [5]. 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 (1). The function \text{rbd}\_\text{list} generates a list of type \text{rbd} by mapping the function \text{atomic} to each element of the given event list \( L \) to make it consistent with the assumptions of Theorem 1. It can be formalized in HOL4 as:
∀ L. rbd_list L = MAP (λa. atomic a) L

The proof of Theorem 1 is primarily based on a mutual independence lemma mutual_indep_cons, given in Table 1, and some fundamental axioms of probability theory.

Similarly by following the above-mentioned approach, used for formalizing the series RBD, we can formally verify the reliability expression for the parallel RBD configuration, given in Equation (2), in HOL4 as follows:

**Theorem 2:** \(\forall p \ L.\) prob_space p \(\land\) (\(\forall x'.\) MEM x' L \(\Rightarrow\) x' \(\in\) events p) \(\land\) ¬NULL L \(\land\) mutual_indep p L \(\Rightarrow\)
\[
prob p (rbd_struct p (parallel (rbd_list L))) = 1 - \text{list_prod} (\text{one_minus_list} (\text{list_prob} p L))
\]

The above theorem is verified under the same assumptions as Theorem 1. The conclusion of the theorem represents Equation (2) where, the function one_minus_list accepts a list of real numbers \([x_1, x_2, x_3, \cdots, x_n]\) 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., \([1 - x_1, 1 - x_2, 1 - x_3, \cdots, 1 - x_n]\).

To verify Theorem 2, we need to verify a lemma that provides an alternate expression for the parallel RBD in terms of the series RBD configuration. As the series and parallel RBD configurations are represented mathematically from the intersection and union of events, respectively. So, this lemma can be expressed mathematically as follows:

\[
P(\bigcup_{i=1}^{N} A_i) = 1 - P(\bigcap_{i=1}^{N} A_i)
\] (6)

The HOL4 formalization of Equation (6) is as follows:

**Lemma 1:** \(\vdash \forall p \ L.\) prob_space p \(\land\) ¬NULL L \(\land\)
(\(\forall x'.\) MEM x' L \(\Rightarrow\) x' \(\in\) events p) \(\land\)
mutual_indep p L \(\Rightarrow\)
\[
prob p (rbd_struct p (parallel (rbd_list L))) = 1 - prob p (rbd_struct p (series (rbd_list (compl_list p L))))
\]

The proof of Theorem 2 is primarily based on Lemma 1 and Theorem 1 along with the fact that given the list of n mutually independent events, the complement of these n events are also mutually independent:
Table 1: Mutual Independence Lemmas

<table>
<thead>
<tr>
<th>Theorem</th>
<th>HOL Formalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>mutual_indep_cons</td>
<td>∀ h p L. mutual_indep p (h::L) ⇒ mutual_indep p L</td>
</tr>
<tr>
<td>mutual_indep_append_sym</td>
<td>∀ p L1 L. mutual_indep p (L1 ++ L) ⇒ mutual_indep p (L ++ L1)</td>
</tr>
<tr>
<td>mutual_indep_front_append</td>
<td>∀ p L1 L. mutual_indep p (L1 ++ L) ⇒ mutual_indep p L</td>
</tr>
<tr>
<td>mutual_indep_append_swap</td>
<td>∀ p L1 L2 L. mutual_indep p (L1 ++ L2 ++ L) ⇒ mutual_indep p (L2 ++ L1 ++ L)</td>
</tr>
<tr>
<td>mutual_indep_cons_append</td>
<td>∀ p h L1 L2 L. mutual_indep p (h::L1 ++ L2) ⇒ mutual_indep p (L1 ++ h::L2)</td>
</tr>
<tr>
<td>mutual_indep_cons_append1</td>
<td>∀ p L1 L. mutual_indep p (h::L1 ++ Q::L) ⇒ mutual_indep p (L1 ++ Q::h::L)</td>
</tr>
<tr>
<td>mutual_indep_cons_append2</td>
<td>∀ h p L1 L. mutual_indep p (L1 ++ h::L) ⇒ mutual_indep p (L1 ++ L)</td>
</tr>
<tr>
<td>mutual_indep_cons_append3</td>
<td>∀ h p L1 L2 L. mutual_indep p (L1 ++ h::(L2 ++ L)) ⇒ mutual_indep p (h::(L1 ++ L))</td>
</tr>
<tr>
<td>mutual_indep_cons_append4</td>
<td>∀ h p L1 L. mutual_indep p (L1 ++ h::L) ⇒ mutual_indep p (L1 ++ L)</td>
</tr>
<tr>
<td>mutual_indep_cons_flat</td>
<td>∀ h p L. mutual_indep p (FLAT (h::L)) ⇒ mutual_indep p (FLAT L)</td>
</tr>
<tr>
<td>mutual_indep_flat_append</td>
<td>∀ h p L1 L2 L. mutual_indep p (FLAT (L1::L2::L)) ⇒ mutual_indep p (L1 ++ L2)</td>
</tr>
<tr>
<td>mutual_indep_flat_append1</td>
<td>∀ h p L1 L. mutual_indep p (FLAT (L1::h::L)) ⇒ mutual_indep p (FLAT ((h ++ L1)::L))</td>
</tr>
<tr>
<td>mutual_indep_flat_append2</td>
<td>∀ h p L1 L2 L. mutual_indep p (FLAT (L1::L2::L)) ⇒ mutual_indep p (L1 ++ L2)</td>
</tr>
<tr>
<td>mutual_indep_cons_flat2</td>
<td>∀ h p L1 L2 L. mutual_indep p (FLAT ([h]:L)) ⇒ mutual_indep p (FLAT ([h]:L))</td>
</tr>
</tbody>
</table>

⊢ ∀ p L. prob_space p ∧ mutual_indep p L ∧
¬NULL L ∧ (∀x’. MEM x’ L ⇒ x’ ∈ events p) ⇒ mutual_indep p (compl_list p L1)
The function \texttt{compl\_list} returns a list of events such that each element of this list is the difference between the probability space \( p \) and the corresponding element of the given list. The proof process of the above lemma utilizes mutual independence properties of Table 1 as well as various other probability independence lemmas that can be found in [6].

The above formalization described for series and parallel RBD configurations builds the foundation to formalize the combination of series and parallel RBD configurations. The type constructors \texttt{series} and \texttt{parallel} can take the argument list containing other \texttt{rbd} type constructors, such as \texttt{series}, \texttt{parallel} or \texttt{atomic}, allowing the function \texttt{rbd\_struct} to yield the corresponding event for an RBD configuration that is composed of a combination of series and parallel RBD configurations.

By extending the RBD formalization approach, presented in Theorems 1 and 2, we formally verified the generic reliability expression for parallel-series RBD configuration, given in Equation (3), in HOL4 as follows:

\textbf{Theorem 3:}\quad \vdash \forall p \ L. \ prob\_space \ p \land \\
\quad (\forall z. \ MEM \ z \ L \Rightarrow \neg \text{NULL} \ z) \land \\
\quad (\forall x'. \ MEM \ x' \ (FLAT \ L) \Rightarrow x' \in events \ p) \land \\
\quad \text{mutual\_indep} \ p \ (FLAT \ L) \Rightarrow \\
\quad (\text{prob} \ p \\
\quad \quad (\text{rbd}\_\text{struct} \ p \ ((\text{parallel} \ of \ (\lambda a. \ \text{series} \ (\text{rbd}\_\text{list} \ a))) \ L)) = \\
\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad (1 - \text{list}\_\text{prod} \ (\text{one}\_\text{minus}\_\text{list}) \ of \\
\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad (\lambda a. \ \text{list}\_\text{prod} \ (\text{list}\_\text{prob} \ p \ a))) \ L)

The first assumption in Theorem 3 is similar to the one used in Theorem 2. The next three assumptions ensure that the sub-lists corresponding to the serial sub-stages are not empty and the reliability events corresponding to the sub-components of the parallel-series configuration are valid events of the given probability space \( p \) and are also mutually independent. The HOL4 function \texttt{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 (3). The infixr function \texttt{of} connects two \texttt{rbd} type-constructors by using the HOL4 \texttt{MAP} function and thus facilitates the natural readability of complex RBD configurations. It is formalized in HOL4 as follows:

\[ \vdash \forall g \ f. \ f \ of \ g = (f \ o \ (\lambda a. \ \text{MAP} \ g \ a)) \]

Similarly, the generic expression of the series-parallel RBD configuration, given in Equation (4), is formalized in HOL4 as follows:
Theorem 4: \[ \vdash \forall p \ L. \text{prob\_space\_p} \land \\
(\forall z. \text{MEM}\ z\ L \Rightarrow \neg\text{-NULL}\ z) \land \\
(\forall x'. \text{MEM}\ x'\ (\text{FLAT}\ L) \Rightarrow x' \in \text{events}\ p) \land \\
\text{mutual\_indep}\ p\ (\text{FLAT}\ L) \Rightarrow \\
(\text{prob}\ p \\
(\text{rbd\_struct}\ p\ ((\text{series\ of } (\lambda a. \text{parallel}\ (\text{rbd\_list}\ a)))\ L)) = \\
(\text{list\ prod}\ of \\
(\lambda a. 1 - \text{list\ prod}\ (\text{one\ minus\ list}\ (\text{list\ prob}\ p\ a))))\ L)\] 

The assumptions of Theorem 4 are similar to those used in Theorem 3. The conclusion models the right-hand-side of Equation (4). To verify Theorems 3 and 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. Some of the foundational structural independence lemmas are presented in Table 2. These lemmas are verified under the following assumptions: (i) prob\_space\_p ensures that p is a valid probability space; (ii) (\forall x. \text{MEM}\ x\ (L1::L) \Rightarrow \neg\text{NULL}\ x) guarantees that the given list must not be empty; (iii) (\forall x. \text{MEM}\ x\ (\text{FLAT}\ (L1::L)) \Rightarrow x \in \text{events}\ p) makes sure that each event in a given list is a valid event in a probability space p; and (iv) mutual\_indep\ p\ (\text{FLAT}\ (L1::L)) ensures that the given list of events are mutually independent in probability. The proof of these lemmas are primarily based on the mutual independence lemmas, given in Table 1, and many fundamental probability theory axioms, for instance, Probabilistic Inclusion-exclusion (PIE) Principle for two events, which can be found in [6].

Now, using Theorem 4, we can formally model and verify the reliability relationship of a nested series-parallel RBD configuration as well, given in Equation (5), in HOL4 as follows:

Theorem 5: \[ \vdash \forall p\ L. \\
\text{prob\_space\_p} \land (\forall z. \text{MEM}\ z\ (\text{FLAT}\ (\text{FLAT}\ L)) \Rightarrow \neg\text{NULL}\ z) \land \\
(\forall x'. \text{MEM}\ x'\ (\text{FLAT}\ (\text{FLAT}\ (\text{FLAT}\ L)))) \Rightarrow x' \in \text{events}\ p) \land \\
\text{mutual\_indep}\ p\ (\text{FLAT}\ (\text{FLAT}\ (\text{FLAT}\ L))) \Rightarrow \\
(\text{prob}\ p\ (\text{rbd\_struct}\ p \\
((\text{series\ of } \text{parallel}\ (\text{series\ of} \\
(\lambda a. \text{parallel}\ (\text{rbd\_list}\ a)))\ L))) = \\
(\text{list\ prod\ of} (\lambda a. 1 - \text{list\ prod}\ (\text{one\ minus\ list}\ (\text{list\ prob}\ p\ a))) of \text{L}))\]
Table 2: Independence of RBD Configurations Lemmas

<table>
<thead>
<tr>
<th>Lemma Description</th>
<th>HOL Formalization</th>
</tr>
</thead>
</table>
| Probability Independence of Series and Parallel-Series RBD Configurations | \[
\forall p \ L1 \ L. \\
(prob \ p \ (rbd\_{struct} \ p \ (series \ (rbd\_{list} \ L1))) \cap rbd\_{struct} \ p \ ((parallel \ of (\lambda a. \ series \ (rbd\_{list} \ a))) \ L)) = \\
prob \ p \ (rbd\_{struct} \ p \ (series \ (rbd\_{list} \ L1)))) \ast \\
prob \ p \ (rbd\_{struct} \ p \ ((parallel \ of (\lambda a. \ series \ (rbd\_{list} \ a))) \ L)))
\] |
| Probability Independence of Parallel and Parallel-Series RBD Configurations | \[
\forall p \ L1 \ L. \\
(prob \ p \ (rbd\_{struct} \ p \ (parallel \ (rbd\_{list} \ L1))) \cap rbd\_{struct} \ p \ ((parallel \ of (\lambda a. \ series \ (rbd\_{list} \ a))) \ L)) = \\
prob \ p \ (rbd\_{struct} \ p \ (parallel \ (rbd\_{list} \ L1)))) \ast \\
prob \ p \ (rbd\_{struct} \ p \ ((parallel \ of (\lambda a. \ series \ (rbd\_{list} \ a))) \ L)))
\] |
| Probability Independence of Series and Series-Parallel RBD Configurations | \[
\forall p \ L1 \ L. \\
(prob \ p \ (rbd\_{struct} \ p \ (series \ (rbd\_{list} \ L1))) \cap rbd\_{struct} \ p \ ((series \ of (\lambda a. \ parallel \ (rbd\_{list} \ a))) \ L)) = \\
prob \ p \ (rbd\_{struct} \ p \ (series \ (rbd\_{list} \ L1)))) \ast \\
prob \ p \ (rbd\_{struct} \ p \ ((series \ of (\lambda a. \ parallel \ (rbd\_{list} \ a))) \ L)))
\] |
| Probability Independence of Parallel and Series-Parallel RBD Configurations | \[
\forall p \ L1 \ L. \\
(prob \ p \ (rbd\_{struct} \ p \ (parallel \ (rbd\_{list} \ L1))) \cap rbd\_{struct} \ p \ ((series \ of (\lambda a. \ parallel \ (rbd\_{list} \ a))) \ L)) = \\
prob \ p \ (rbd\_{struct} \ p \ (parallel \ (rbd\_{list} \ L1)))) \ast \\
prob \ p \ (rbd\_{struct} \ p \ ((series \ of (\lambda a. \ parallel \ (rbd\_{list} \ a))) \ L)))
\] |

\((\lambda a. \ list\_prod \ a) \ of \\
(\lambda a. \ 1 - list\_prod \ (one\_minus\_list \ (list\_prob \ p \ a))))) \ L)\)

Most of the assumptions of the above theorem are very similar to those used in Theorem 4 and the remaining ones are used to ensure that the reliability
event lists are not empty and their corresponding reliability events are mutually independent with respect to the given probability space. The proof of above theorem uses the results of Theorems 1 and 3 and various lemmas, like the ones presented in Table 2, stating that given the mutually independent reliability events list, a reliability event associated with the sub-component of the inner series-parallel RBD configuration is independent of the reliability event associated with the nested series-parallel RBD configuration.

The formalization, reported in this paper, took about 3500 lines of HOL4 proof script and took about 400 man-hours. It is worthwhile to mention that the generic formalization of lemmas, presented in Tables 2 and 3, related to mutual independence and probability independence of RBD configurations significantly reduced the HOL4 proof script for RBD formalization, which is available for download at [6]. The formal reasoning was primarily based on probabilistic, set-theoretic and arithmetic simplification and some parts of the proofs were also handled automatically using the various built-in automatic provers and advanced simplifiers in HOL4. The main benefit of the formalization, presented in this section, is the ability to formally analyze the reliability aspects of safety-critical systems within the sound environment of a theorem prover, as will be demonstrated using a Virtual Data Center example in the next section.

References


