Journal of Applied Logic 18 (2016) 19–41

Contents lists available at ScienceDirect

Journal of Applied Logic

www.elsevier.com/locate/jal

Formalization of Reliability Block Diagrams in Higher-order Logic

Waqar Ahmed[a,*], Osman Hasan[a], Sofiène Tahar[b]

[a] *School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan*
[b] *Department of Electrical and Computer Engineering, Concordia University, Montreal, Canada*

ARTICLE INFO

Article history:
Received 26 May 2015
Received in revised form 15 February 2016
Accepted 25 May 2016
Available online 15 June 2016

Keywords:
Reliability Block Diagrams (RBDs)
Higher-order logic
Probability theory
Virtualization configuration
Virtual Data Centers

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 inaccuracy 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. For illustration purposes, we formally analyze the reliability of a generic Virtual Data Center (VDC) in a cloud computing infrastructure exhibiting the nested series-parallel RBD configuration.

1. Introduction

Reliability Block Diagrams (RBDs) [6] are used to assess various failure-related characteristics, such as reliability [18], availability [13] and maintainability [7], of a wide range of engineering systems. An RBD is primarily a graphical structure that consists of blocks and connectors (lines) representing the functional behavior of the system components and their interconnectivity with each other, respectively. For example, while assessing the reliability of a computational software, the blocks may represent the computational elements, with some given failure rate, and the connectors between them may be used to describe various alternative paths required for a successful computation using the given software [1]. Now, based on this

[*] Corresponding author.
E-mail addresses: waqar.ahmad@seecs.nust.edu.pk (W. Ahmed), osman.hasan@seecs.nust.edu.pk (O. Hasan), tahar@ece.concordia.ca (S. Tahar).

http://dx.doi.org/10.1016/j.jal.2016.05.007
1570-8683/© 2016 Elsevier B.V. All rights reserved.

RBD, the failure characteristics of the overall system can be judged based on the failure rates of individual components, whereas the overall system failure happens if all paths for successful execution fail. The RBD analysis enables us to evaluate the impact of component failures on the overall system reliability and thus is widely used for assessing the trade-offs of various possible system configurations, such as series, parallel or a combination of both, at the system design stage.

Traditionally, RBD-based analysis is carried out using paper-and-pencil proof methods and computer simulations. The first step in the paper-and-pencil proof methods is to express the reliability of each component of the system in terms of its failure rate λ and a random variable, like exponential [31] or Weibull [16], which models the failure time. This information, along with the RBD of the system, is then used to analytically derive mathematical expressions for the system-level failure characteristics. Due to the involvement of manual manipulation and simplification, this kind of analysis is prone to human errors and the problem becomes more sever when analyzing large systems. Moreover, it is possible, and in fact a common occurrence, that many key assumptions required for the analytical proofs are in the mind of the specialist assisting the system engineers in the analysis of the system and they are hence not 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, computer simulators, such as ReliaSoft [24] and ASENT reliability analysis tool [4], have been extensively used for the RBD analysis of the various real-world systems. However, they cannot ensure absolute correctness as well due to the involvement of pseudo-random numbers and numerical methods.

To overcome the above-mentioned inaccuracy problems, formal methods have also been proposed for the RBD-based analysis using both state-based [21,25] and theorem proving techniques [3]. However, state-based approaches can neither be used to reason about continuous elements and nor for verifying generic reliability expressions. These limitations can be overcome by using theorem proving, given the high expressiveness of higher-order logic and inherent soundness of the provers, and thus generic mathematical expressions involving continuous elements can be verified.

In [3], we presented a formalization of the series RBD using the HOL4 theorem prover [27]. This formalization was then successfully used to verify the reliability of an oil and gas pipeline [3]. However, most of the RBDs for real-world systems involve a combination of series and parallel configurations. Moreover, another limitation of the series RBD formalization of [3] is that the series RBD function takes a single-dimension list of random variables as an argument, where each element of this list models a single component of the structure. This fact limits the usage of this function to model the case when the system as well as its components are also modeled by the RBD configurations, or in other words, this formalization does not cater for nested RBD configurations. The ability to handle such nested RBD configurations requires assigning the random variables to each block or sub-stage of the system-level RBD.

To overcome the above-mentioned limitations, we propose a deep embedding approach to formalize the commonly used RBD configurations, such as series, parallel, parallel-series and series-parallel. In particular, we introduce a recursive datatype *rbd* to formalize RBD configurations consisting of type constructors, such as `series`, `parallel` and `atomic`. Then, a semantic function over the *rbd* datatype is defined with the ability to decode the RBD configuration encoded by these type-constructors to yield the corresponding reliability event, which corresponds to the scenario when the given system or component does not fail before a certain time. This proposed formalization approach is compositional in nature and can be easily extended to cater for any combination of series and parallel RBD configurations. Also, it allows us to verify the generic reliability expressions for RBDs on any reliability event list of arbitrary length and thus overcomes the above-mentioned limitations of series RBD formalization [3]. To elaborate the compositional ability of the proposed RBD formalization, we also present a higher-order logic formalization of a nested series-parallel RBD, which is a series-parallel RBD having each block itself modeled by a series-parallel RBD configuration.

To illustrate the practical effectiveness of our work, we utilize our proposed RBD formalization to conduct the formal reliability analysis of a generic Virtual Data Center (VDC) system in a cloud computing