e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

USING SMT Z3 SOLVER FOR RISK MANAGEMENT OPTIMIZATION: A PROOF OF CONCEPT

Sherif M. Saif[*1]

[*1]ERI, Computers and System Dept., Egypt.

DOI: https://www.doi.org/10.56726/IRJMETS63240

ABSTRACT

Risk assessment is a fundamental component of information security management systems, involving the identification, evaluation, and prioritization of potential threats and vulnerabilities. The optimal allocation of resources to reduce the effects of these risks is a complex challenge, often requiring careful consideration of various factors. This paper serves as a proof-of-concept showcasing the potential and power of Z3 which is a Satisfiability Modulo Theories (SMT) solver, to resolving risk management optimization. By formulating the risk management problem as a mathematical model, we demonstrate the feasibility of using the Z3 SMT solver to identify optimal risk mitigation strategies, and prove that the solver can be used to determine the most efficient allocation of resources to implement the actionable controls for reducing the effects of risks. Hence the research proves the solver approach for addressing complex risk management challenges and optimizing information security investments.

I. INTRODUCTION

Risk management is a critical process for organizations to identify, assess, and mitigate potential threats. However, as organizations grow in complexity and the volume of data increases, the traditional methods of risk management often become inadequate. To address this challenge, a more systematic and data-driven approach is required.

One such approach involves leveraging formal verification techniques to model and analyze risk scenarios. Satisfiability Modulo Theories (SMT) solvers, powerful tools for automated reasoning, have emerged as a promising solution for tackling complex decision-making problems in various domains, including risk management.

This paper presents a novel application of the Z3 SMT solver to optimize risk management strategies. By formulating the risk management problem as a mathematical model, we demonstrate how SMT solvers can efficiently identify optimal solutions, even in the face of complex constraints and uncertainties. The paper is organized as follows. Section 1 is an Introduction. Section 2 explains the scientific idea behind the approach used in this paper. Hence, it explains the scientific foundation of SAT and SMT Solvers and puts an emphasis on Satisfiability Modulo Theories (SMT) as a powerful tool for automated reasoning. Section 3 focuses on a specific tool: Z3 as a powerful SMT solver and surveys its applications. Section 4 explains the idea of the risk management, and the relevant concepts including risk registers and risk treatment. Section 5 shows the mathematical formulation of the problem and proposes the framework for using Z3 Solver for optimization of risk management. Section 6 provides the results that proves the feasibility of the proposed idea and discusses how this proof-of-concept can be further developed. Section 7 is for conclusion and future work.

1 The Scientific Foundation of SAT and SMT Solvers

1.1 Boolean Satisfiability (SAT): A Fundamental Problem

At the heart of SAT and SMT solvers lies the Boolean satisfiability problem (SAT). Given a Boolean formula, SAT asks whether there exists an assignment of truth values (true or false) to the variables that makes the formula true. This fundamental problem, though simple to state, is NP-complete, meaning that in the worst-case scenario, it requires exponential time to solve [1].

1.2 The DPLL Algorithm: A Core Technique

The Davis-Putnam-Logemann-Loveland (DPLL) algorithm is a fundamental algorithm for solving SAT problems. It employs a systematic search strategy, exploring the space of possible assignments. DPLL uses techniques like unit propagation, pure literal elimination, and backtracking to prune the search space and efficiently find a satisfying assignment or determine that none exists.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

1.3 Extending SAT to SMT

While SAT deals with Boolean formulas, SMT extends this concept by incorporating theories from various domains, such as arithmetic, arrays, and bit vectors. SMT solvers combine the techniques of SAT with specialized decision procedures for these theories.

For instance, when dealing with arithmetic theories, SMT solvers can leverage techniques from linear programming or integer programming. For bit-vector theories, they can employ bit-level reasoning and symbolic execution. By integrating these techniques, SMT solvers can handle more complex problems than traditional SAT solvers.

1.4 Satisfiability Modulo Theories (SMT): A Powerful Tool for Automated Reasoning

SMT is a powerful decision procedure that combines elements of Boolean satisfiability (SAT) and first-order logic. It is used to determine the satisfiability of logical formulas over various theories, such as arithmetic, arrays, and bit vectors [2].

At its core, SMT solvers take a logical formula as input and determine whether there exists an assignment of values to variables that makes the formula true. If such an assignment exists, the solver returns a satisfying assignment; otherwise, it reports that the formula is unsatisfiable. SMT solvers have become increasingly sophisticated, capable of handling complex formulas and theories.

II. Z3: A POWERFUL SMT SOLVER, AND ITS APPLICATIONS

Z3 is a state-of-the-art SMT solver developed by Microsoft Research [1] [3]. It has gained significant popularity due to its efficiency, robustness, and versatility. Z3 supports a wide range of theories, including:

- **Linear arithmetic:** Deals with linear inequalities and equalities over integer and real numbers.
- **Bit vectors:** Represents fixed-size bit vectors and supports operations like bitwise AND, OR, XOR, and shifts.
- **Arrays:** Models arrays with read and write operations.
- **Uninterpreted functions:** Represents functions with unknown behavior.

Z3 employs a variety of techniques to solve SMT problems, including:

- **SAT solving:** Leverages techniques from Boolean satisfiability to handle propositional logic.
- **Theory-specific solvers:** Uses specialized algorithms to handle specific theories like linear arithmetic and bit vectors.
- **Combination of solvers:** Integrates different solvers to handle complex formulas involving multiple theories.

Z3's ability to handle complex formulas and its efficient implementation make it a powerful tool for various applications. It has been used in a wide range of domains, including:

- **Formal verification** [4]**.**
- **Software testing** [5].
- **Model checking** [6].
- **Artificial intelligence** [7].
- **Security** [8].

The increasing complexity of modern systems necessitates automated reasoning techniques to ensure their correctness and security. SMT solvers provide a powerful tool for addressing these challenges, enabling the analysis and verification of complex systems [9].

III. RISK MANAGEMENT

Risk management is a systematic process of identifying, assessing, and controlling risks. It involves a series of steps to minimize the impact of potential threats and maximize opportunities. By proactively addressing risks, organizations can enhance their resilience, protect their assets, and achieve their strategic objectives.

2.1 Key Concepts in Risk Management

- **Risk:** A potential event that may have a positive or negative impact on an organization's objectives.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

- **Risk Assessment:** The process of identifying, analyzing, and evaluating risks.
- **Risk Treatment:** The process of selecting and implementing strategies to modify risk.
- **Risk Monitoring and Review:** The ongoing process of monitoring risks and taking corrective action as needed.

3.1 Risk Management Standards and Frameworks

Several standards and frameworks provide guidance on risk management practices [10] [11] [12]. Some of the most prominent ones include:

- **ISO/IEC 27001:** This international standard specifies the requirements for an information security management system (ISMS). It provides a comprehensive approach to risk management, including risk assessment, treatment, and monitoring.
- **NIST Cybersecurity Framework (CSF):** Developed by the National Institute of Standards and Technology (NIST), the CSF provides a flexible and adaptable approach to managing cybersecurity risk.
- **COBIT 5:** This framework provides a holistic approach to IT governance and management, including risk management. It emphasizes the importance of aligning IT with business goals and mitigating risks.
- **ITIL 4:** This framework provides a comprehensive set of best practices for IT service management, including risk management. It focuses on managing risk throughout the IT service lifecycle.

3.2 Risk Assessment and Risk Registers

Risk assessment is a critical step in the risk management process. It involves:

1. **Risk Identification:** Identifying potential threats and vulnerabilities that could impact the organization.
2. **Risk Analysis:** Evaluating the likelihood and potential impact of each identified risk.
3. **Risk Evaluation:** Prioritizing risks based on their severity and the likelihood of occurrence.

Risk registers are essential tools for documenting and tracking identified risks. They typically include the following information:

- **Risk Description:** A clear and concise description of the risk.
- **Risk Owner:** The individual or team responsible for managing the risk.
- **Risk Rating:** A quantitative or qualitative assessment of the risk's severity.
- **Risk Controls:** The specific measures implemented to mitigate the risk.
- **Risk Monitoring:** A plan for monitoring the effectiveness of risk controls.

3.3 Factors Considered in Risk Assessment

- **Impact:** The potential consequences of a risk materializing.
- **Probability or Likelihood:** The probability of a risk event occurring.
- **Chances for Detection:** The likelihood of identifying a risk before it occurs.
- **Severity:** A combination of impact and likelihood.
- **Business Impact:** The potential impact on business operations, financial performance, or reputation.

By carefully analyzing these factors, organizations can prioritize risks and allocate resources effectively to mitigate the most significant threats.

3.4 Risk Treatment Strategies

Once risks have been identified and assessed, organizations can employ various strategies to treat them:

- **Risk Avoidance:** Eliminating the risk altogether by avoiding the activity or decision that gives rise to the risk.
- **Risk Reduction:** Implementing controls to reduce the likelihood or impact of a risk.
- **Risk Transfer:** Shifting the risk to a third party, such as through insurance or outsourcing.
- **Risk Acceptance:** Accepting the risk and its potential consequences.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

3.5 Continuous Monitoring and Review

Effective risk management requires ongoing monitoring and review. Organizations should regularly assess the effectiveness of their risk controls and update their risk registers as needed. By staying proactive and adapting to changing circumstances, organizations can minimize the impact of risks and achieve their strategic objectives.

IV. PROPOSAL OF USING Z3 FOR RESOLVING RISK MANAGEMENT PROBLEM

This section begins with problem identification, and then it explains why this problem is classified as NP-Complete, and shows some examples for NP-Complete problems. It shows the algorithm used for brute force implementation and the proposed Z3 system.

4.1 Problem Identification

Organizations can gain a comprehensive understanding of their risk landscape by quantifying the risk factors. Incorporating different parameters within the risk register table establishes a quantifiable framework for assessing each risk's gravity. By synthesizing these factors, organizations gain a clear perspective on their risk environment, enabling informed decisions for managing risks [13]. Different organization use different approaches to calculate the overall risk. They usually multiply two or three factors to calculate the overall risk. These factors can be:

- Risk impact and risk probability.
- Risk impact, risk probability and risk chance of detection.

After forming the risk registers, a significant challenge lies in allocating resources effectively to implement actions, that are referred to as controls, and the purpose of these actions is to reduce the risk value that was calculated for each of the identified risks scenarios. Hence, this requires and mandates calculating what is referred to as risk scores before and after the actionable control implementation. This will require also prioritizing these actionable controls to decide what to start with and what to defer, given that implementation resources are always limited. This is a computationally demanding task. Such a task is often classified as NP-Complete. By addressing this challenge, organizations can optimize resource allocation and minimize overall risk exposure.

4.2 NP Completeness

NP-completeness is a fundamental concept in computer science that classifies problems based on their computational complexity. NP-complete problems are a class of decision problems that are believed to be inherently difficult to solve, meaning that no efficient algorithm is known to solve them for all possible inputs.

A problem is considered NP-complete if it belongs to the class of problems that can be verified in polynomial time (NP) and if every other problem in NP can be reduced to it in polynomial time. This property implies that if a polynomial-time algorithm were found for any NP-complete problem, it could be used to solve all other NP-complete problems efficiently.

4.3 Famous NP-Complete Problems

Several well-known problems have been proven to be NP-complete, including:

- **The Traveling Salesman Problem (TSP):** Given a list of cities and the distances between them, find the shortest possible route that visits each city exactly once and returns to the origin city.
- **The Satisfiability (SAT) Problem:** Given a Boolean formula, determine whether there exists an assignment of truth values to the variables that makes the formula true.
- **The Knapsack Problem:** Given a set of items, each with a weight and a value, determine the combination of items to include in a knapsack of a given weight capacity to maximize the total value.
- **The Vertex Cover Problem:** Given an undirected graph, find the smallest subset of vertices such that each edge in the graph is incident to at least one vertex in the subset.

While no efficient algorithm is known for these problems, researchers continue to explore approximation algorithms and heuristics to find near-optimal solutions.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

4.4 Highlighting the Problem's NP Completeness

The order of addressing the identified risks will affect the overall risk throughout the risk whole project life-time. Throughout this time the residual risk has to be minimized. To achieve this goal, we can decide the order of implementation through a brute force solution. The brute force solution examines all the possible permutations. If someone has to construct a 2-Dimensional array, whose number of columns is equal to the number of risks and whose number of rows is equal to the permutations of order of addressing the risks this can be used to resolve this optimization problem. Therefore, if there are M risks, the number of rows will be represented by the equation:

$$\text{Number of risk reduction scenarios} = M! \quad (1)$$

Table 1. Number of risks vs. number of permutations vs. log of base10 of the number of risk handling scenarios.

Number of Risks	Number of Risk Scenarios	Log base10 of the number of risk handling scenarios
3	6	0.8
4	24	1.4
5	120	2.1
6	720	2.9
7	5040	3.7
8	40320	4.6

4.5 Build a Brute Force Resolution Model

To show the brute force approach for resolving this model the following algorithm is used:

Algorithm1:

1. **Initialize Variables:**
   - Define _NoOfRisks as the number of risks (default: 6).
   - Calculate numOfRows as the factorial of _NoOfRisks to represent the total possible permutations of risk indices.
   - Set constant values:
     - N as the total project phases,
     - Store as the initial risk values,
     - Store_final as the final risk values post-mitigation,
     - T as control times for each risk.
2. **Generate Risk Permutations:**
   - Generate all permutations of risk indices using a helper function generatePermutations1006().
3. **Create Google Sheets Setup:**
   - Open the active Google Sheet, clear existing content, and add headers for each risk and total risk scores.
4. **Calculate Risk Scores and Populate Table:**
   - **For each permutation** of risks:
     - Initialize an array G to store risk values and calculated scores.
     - For each risk:
       1. **Calculate Risk Contribution (RC)** by combining R (initial risk), R_final (final risk), T (control times), and N (total duration), with summation adjustments based on control time.
       2. Accumulate RC values to calculate the total risk score for the permutation.
     - Store the permutation indices and total risk score in G.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:11/November-2024 Impact Factor- 8.187 www.irjmets.com

- Output the populated G array to the sheet, starting from row 2 to preserve headers.

5. **Generate Formulas for Dynamic Calculations:**

- **For each row** corresponding to a permutation:
  - Construct formulas to dynamically retrieve Effort, Initial Risk, and Final Risk values from Google Sheets.
  - Calculate each risk's impact on the total risk score using a combined formula and insert the formula in columns of the sheet.
  - Summate calculated risk scores in each row's total risk cell.

6. **Create Summary Table and Statistical Labels:**

- Append a table with initial risk, final risk, and effort values at the end of the main data table.
- Add minimum and maximum total risk calculations, using formulas that reference the main data table.
- Label minimum and maximum total risk values and apply bold formatting to improve table readability.

4.6 Forming a Z3 Model

To resolve this problem, the proposal of this work is to formulate a mathematical structure that represents the identified problem and then input the model to Z3 to find an optimum solution, throughout the following steps:

1- Define cost function.

2- Define z3 constraints.

4.6.1 Cost Function:

We call the function Gross Whole Risk (GWR). The GWR is calculated by calculating the total of all the risk values throughout an agreed upon period P. For M risks, each of them will risk will have a risk share (RC). Calculating the total sum of the RS's will be the GWR.

The RS of any risk will be calculated by the following equation:

$$RS(x) = R(x) \times (\Sigma_{i=1 \text{ to } n} \text{effort}(i)) + R'(x) \times (T - (\Sigma_{i=1 \text{ to } x} \text{effort}(i))) \quad (2)$$

The GWR will be the summation of all RS's. If there are Z risks, the Gross Whole Risk will be calculated by summing all RS's

$$GWR = \Sigma_{i=1 \text{ to } Z} RS(i) \quad (3)$$

The goal is to minimize GWR represented by equation 3.

4.6.2 Constraints:

The following constraints will be fed to the solvers:

- Implementations of controls are non-overlapping since they will be implemented one-at-a-time.
- Time will be represented by positive integers.
- Start times will in sequence.
- Start time depend on order and control time
- Define the solver's goal to **minimize** the GWR
- Request Z3 check satisfiability
- Request Z3 to find a solution

V. EXPERIMENTAL RESULTS AND DISCUSSION

When an example was fed to the solver it yielded the following output:

Recommended Order of Actionable Control Implementation	Order of implementation	Start time of implementation
defined_risk_number3	1	1
defined_risk_number4	2	2
defined_risk_number5	3	5
defined_risk_number2	4	7

defined_risk_number1	5	10

This same output was obtained through the brute force approach.

VI. CONCLUSION

The work was centered on the understanding of the risk management problem and understanding it as a decision problem that can be formulated in a structure that is apt to be resolved by the Z3 solver. The proposal aimed to get and define an optimum implementation for actionable controls by ordering them in a specific order that reduces the overall cost function. The cost function that was presented is named GWR and its goals is to minimize the total risk which is the summation of the addressed risk and the residual risk throughout the time of the implementation project.

A brute force version was implemented and it was followed by an implementation of the Z3 solver version. For a small number of risks both approaches yielded the same results. This served as a proof-of-concept that this approach can be used for large number of risks with factorial complexity.

More work can be done in the future by introducing a synthetic risk generator that can be used for extensive testing of the framework. In addition, an automatic wrapper can be added to the system to allow its flexible usage instead of hardcoding the problems in this work. Another enhancement would be increasing the complexity of the system to by involving more resources in the actionable controls implementation to addresses several controls simultaneously.

VII. REFERENCES

[1] L. &. B. N. de Moura, "Z3: An efficient SMT solver. Tools and Algorithms for the Construction and Analysis of Systems," pp. 337-340, 2008.

[2] N. P. A.-D. &. F. L. Bjorner, "νZ - An optimizing SMT solver," Tools and Algorithms for the Construction and Analysis of Systems, pp. 194-199, 2015.

[3] NikolajBjorner, "Z3 Prover, Z3," https://github.com/Z3Prover/z3, GitHub.

[4] S. B. S. J. T. T. &. S. C. Bak, "Hybrid automata: From verification to implementation," Proceedings of the IEEE, vol. 108, no. 4, pp. 657-682, 2020.

[5] Y. W. Y. Z. Z. &. L. D. Liu, "Formal verification of security protocols in big data systems using SMT solvers," International Journal of Information Security, vol. 17, no. 6, pp. 671-683, 2018.

[6] S. S. B. C. C. &. B. D. Meier, "The TAMARIN prover for the symbolic analysis of security protocols," Computer Aided Verification (CAV), pp. 696-701, 2013.

[7] E. K. L. E. &. V. M. Y. Plaku, "Motion planning with dynamics by a synergistic combination of layers of planning," IEEE Transactions on Robotics, vol. 26, no. 3, pp. 469-482, 2013.

[8] R. &. T. P. Sebastiani, "OptiMathSAT: A tool for optimization modulo theories," Tools and Algorithms for the Construction and Analysis of Systems, pp. 447-454, 2015.

[9] N. B. B. M. M. R. R. r. M.C. Leva, "Structuring data collection to develop risk intelligence," Safety Science, vol. 100, no. Part B, p. 2017, 143-156.

[10] N. I. o. S. a. Technology., "Risk Management Guide for Information Technology Systems. NIST Special Publication 800-30. 2012.," https://www.nist.gov/.

[11] ISACA, "COBIT, Control Objectives for Information and Related Technology," https://www.isaca.org/resources/cobit.

[12] "ITIL: Information Technology Infrastructure Library," Information Technology Infrastructure Library.

[13] N. U. A. R. L. D. A. Y. H. R. N. S. T. R. B. e. a. Singh, "Assessment of Profitability and Sustainability in Integrated Farming Systems: A Case Study in Meghalaya, India," vol. 55, no. 01, 2024.