Indonesian Journal of Computer Science
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

A Review of Automated Reasoning And Its Applications in The 21st Century

Rebeccah Ndungi[1], Shofwatul 'Uyun[2]
rebeccahndungi1@gmail.com
Department of Informatics, UIN Sunan Kalijaga Yogyakarta

Article Information

Submitted : 17 Mar 2023
Reviewed: 3 Apr 2023
Accepted : 20 Apr 2023

Keywords

Automatic Reasoning, Artificial Inteligence, F-O-L, Theorem Proving, Superposition and Saturation

Abstract

This article takes a look at the progress and advancement of automated reasoning and its applications in the 21st century. Reasoning refers to the method of reaching logical conclusions. The construction of computing systems that automate this process over some knowledge bases is the focus of automatic reasoning. Automated Reasoning is frequently regarded as a subfield of machine learning. It is also studied in theoretical computer science and philosophy. Some of the applications of automated reasoning include but not limited to Tableau-style systems, Automatic Theorem Proving, Superposition and Saturation, benchmarks and Classical First-Order Logic. The development of formal led to the development of artificial intelligence, which was essential in the development of artificial intelligence for reasoning.

A. Introduction

Certain outstanding topics in formal logic and mathematics have benefited greatly from the use of automated reasoning software [1].

Automated argumentation is most commonly utilized in conjunction with deductive reasoning to locate, check, and verify mathematical theorems through the use of a computing system. When checking proofs with an automated reasoning system, the user can be certain that they have not committed an error in their computations. Automatic argumentation can also be used for implementations in math, engineering, and computer science. It can also be utilized for non-mathematical objectives, such as posing questions in exact philosophical. However, a significant number of these additional topics still require representation in a language that can be comprehended by the software [2].

The Compendium of Automated Rationalization (RV01) provides an overview of computational modeling as a broad subfield of artificial intelligence; however, the questions that will be asked here will be framed in terms of completely automated identity in the figurative heritage, with a focus on classical first-order logic. There is no thought given to contexts in which automated reasoning is used for purposes other than the assertion of theorems in an interactive setting, the discovery of counterexamples, the answering of questions, or the creation of programs. The same goes for special-purpose reasoning, including approaches for autonomous reasoning in planar geometry, and logic that differ greatly from first-order logic, like description, modal, basic structure, and higher-order logic [1]

According to the Stanford Encyclopaedia of Philosophy, "reasoning" is the capacity to make conclusions, and "automated reasoning" refers to the process of constructing computer systems that carry out this process automatically. Even though the overarching objective is to mechanize the many modes of reasoning, this term has been primarily associated with sound deductive reasoning in the sense that it is utilized in mathematics and logic. In this regard, automated reasoning is comparable to the process of mechanically proving theorems. To construct an automated reasoning program, one must first provide an algorithmic description of a formal calculus[3].

That way, the theorems of formal calculus can be proven quickly and easily with the help of a computer implementation of calculus. This task requires defining the problem domain, selecting an appropriate representational language for the data that will be provided to the program as well as any new data that will be deduced by the program, laying out the steps that will be taken by the program to make deductive inferences, and determining the most effective way to carry out the necessary computations. These responsibilities must be met in full [3]. Researchers in the field of computational approaches are now using automated analysis programs to investigate problems in mathematics and logic, create important uses in computing and engineering, zero in on the solution of engineering problems, and find new angles from which to investigate questions in exact philosophy. This field has progressed to the point where automated reasoning projects are being deployed [4], even if fundamental research is still being done to give the necessary theoretical framework.

Observation is the process of engaging one's senses with the external world to gather new data that can be expressed verbally. In contrast hand, reasoning is the mental process of deducing new information from previously known material using only the mind and no external sensory input whatsoever (as opposed to learning, which involves both the senses and the brain). Some generic mappings, which are largely knowledge-type dependent, can be used to accomplish this. Scientific explanations of reality are based on the interplay of these two methods of inquiry, which can be used independently or together [5]. Automated argumentation is the study of creating methods to replace human reasoning with processes that conduct out individual thinking stages structurally and can locate, immediately, suitable action scenarios of reasoning steps for acquiring new information from current information. The study of such methods is known as automated reasoning [6, 7].

The significance of automated argumentation arises from the fact that the vocabulary of reasoning on which modern automated reasoners operate may convey almost all information relevant to modern technologies and science. This language can be used to describe systems (including their hardware and software specifications and implementations), internet resources, and scientifically-generated facts and data [1]. For understanding and further developing our working and living surroundings through the use of science and technology, the methods of spontaneous reasoning are becoming increasingly significant, and in some circumstances indispensable. This is because data gathered by the observational sciences is becoming increasingly complex. Humans have, over the course of thousands of years, developed tools for boosting and aggravating their physical power, and have begun developing techniques (e.g., devices in physics) for helping to improve the direct observation power; therefore, it has become the organic follow-up to create latest technology for trying to enhance and aggravating the reasoning power of the living person [8].

In the same way that, over the course of millennia, humans have developed tools for enhancing and exacerbating their physical power and eventually developed tools [4], [5] the primary goal of automated reasoning, which also encompasses automated deduction and automated theorem proving, is to develop computer programs that use logic or reason for the solution of a variety of problems, including open questions. This can be accomplished by creating computer systems that use automated deduction and automated theorem proving [6]. The subfield of computer science known as automated reasoning aims to apply reasoning in the form of logic to computer systems as its primary research topic. An automated reasoning system should be able to automatically draw logical conclusions in the direction of a given goal when presented with a set of assumptions and the goal itself as inputs. Activities such as proving theorems, checking proofs, and designing circuits can all benefit from the automation and application of logical reasoning that is made possible by computers that use automated reasoning [3], [6], [7].

The application of logic inside the form of reasoning by analogy, induction, abduction, and other forms of non-monotonic reasoning can also be utilized by automated reasoning. Nevertheless, deductive approach in mathematics and logic is the most common context in which the term "automated reasoning" is [8] The

category of challenges that are dealt with by an automated machine learning program is referred to as the "problem domain" in the industry. The automated reasoning system is provided with problem assumptions, which are assertions that provide important information to the system, and problem implications, which are the questions that are being asked of the system. Problem domains have both of these [9] [10], [11]
The problem domain will be provided to the reasoning software as an input, and the computer will then provide a solution, such as the accuracy of proof, as output. When a solution is identified or when all of the resources are used up, a program that uses automated reasoning will come to an end [12]The most typical application of automated reasoning programs is to prove theorems, which can be done by providing algorithmic descriptions of the calculus that is being employed. Users are also obligated to specify the class of problems that the automated reasoning initiative will need to solve, the language that the program will use to symbolize given information, and the methods the program will use to implement inductive approach inferences. These requirements can be found in the automated reasoning program documentation. Although the phrase "automated deduction" can also be used to refer to "automated reasoning," the word "automated deduction" is often reserved for referring to the application of deduction logic in mathematical contexts[1].
Leibniz shows that there are two fundamental Steps involved in automatic reasoning[1].
1. Provide an explicit statement of theorems using a language suitable for academic purposes. (universalis developed by Leibniz.)
2. Perform adjustments on the formal statements using automated algorithmic processes. (Leibniz's calculus ratiocinator)

B. Theorem Proving

i. Classical First-Order Logic

ii. Statements concerning persons, functions & predicates over individual persons, propositional connectives, and quantifiers, over individuals all have syntax and semantics in first-order logic (see, for example, First-Order Logic [Smu95]). It is possible to axiomatize the inclusion of the bi equivalence criterion s = t across words; or to use it explicitly in the logic. You may know it as hypothesis logic or first-order proposition logic. For the purpose of developing data about things and expressing their relations, 1st logic is a highly efficient language [13]–[15].

In the context of classical first-order logic, the so-called "rule of excluded middle," which stipulates that a proposition must either be true or false, is accepted. First-order logic can be embedded in or interpreted into other logics; "theories" like integer arithmetic can be inserted in a slightly orthogonal fashion, and it is sufficiently expressive for "everyday mathematics." However, it is not so descriptive that it evolves into utterly intractable [KV13]. Classical 1st logic is a popular reasoning for a wide range of applications [13], [14]. It is syntactically coherent, semantic information comprehensive, closed under negation, as well as admits powerful fixed points and cut-elimination, which are all theoretical niceties that aid both mechanization and expressivity. This logic in particular possesses a variety of theoretical niceties that aid both digitization and expressivity. The conclusion reached by Lindstrom's theorem is this is the most robust line of reasoning that nonetheless possesses desirable qualities [Lin69][1].

The developed system is capable of much greater expressiveness after the addition of propositions and categorical urgencies to propositional logic, as is done in first-order logic. With the application of Robinson's resolution approach (for further background information, see the chapter that follows), it is now feasible to automate the process of looking for evidence in first-order logic. Although the basic inference phase of the resolution is simple, the heuristics that are necessary to make this search process more accurate in practice are considerably more complicated than those that are used by SAT solvers [6]

Equivalence is an essential component of many hypotheses that are inspired by problems that exist in the real world. Researchers have developed equational proposition provers to include equality immediately into the logic as a result of the discovery that enhancing 1st logic with parity axioms is an inefficient method. Even though it is feasible to automate such triangle inequality appreciate the ability, the search techniques that are used are complicated, and different heuristics perform much better on different problems [6], [16].

This particular kind of prover is the focus of the work that is addressed in the dissertation. The reason for this is that the products that such subsequent processes produce are robust and may one day be amenable to automation, but for the time being, they require some human condition to function correctly. This reliance on humans could be drastically decreased or possibly done away with entirely if we are effective in building reliable automated processes using machine learning [6].

iii. Automatic Theorem Proving

Given a fully implemented inference system, it is known that finitely long proofs of true assertions in a first-order logic system exist. This characteristic supports the existence of automated theorem provers (ATPs)—computer programs that, given a claim, seek to prove it by probing a search space generated by an inference system [17], [18].

Unfortunately, it is known that proof discovery in first logic is semidecidable, both in concept and in reality, and that it is computationally intensive. Nonetheless, the creators of these technologies have continued working on them [19], [20].

Automatic thinking in this context has a lengthy history [Dav01] and several instances of its application leading to successful outcomes. Throughout this period, one of the primary goals of research has been to achieve concluding up to redundancy. For example, normal forms for mathematical formulas [BEL01] avert superfluous reasoning up to invariance in the second normal; the service produced [Rob65] tries to avoid generating components of the Herbrand world unless it is necessary; the paramodulation and afterward quantum mechanical calcification [NR01] start reducing the space of possibilities equality rationale steps; and so on. These methods can lessen the scope of the solution space that needs to be investigated, which in turn speeds up the process of obtaining proofs and makes it possible to solve more challenging problems [17], [21].

Others include the (provisional) viewpoint of problems voiced in other structures to the initial logic (e.g. [MP08, Urb06]), the assimilation of the outside rationale toolkits including such SAT/SMT methods for solving problems, and the creation of effective proof calculi and acquiring knowledge algorithms based upon those technics (such as the density and tableau households, below) [19].

a) Benchmarks

According to the lawsuit, progress in robotic theory is being propelled to a significant degree by benchmark mathematical problems and competitions. Hundreds of Theorem Problems The principal source of issues for the annual CASC competition [Sut16] comes from Understand one thing (TPTP) [SSY94], a regulated set of puzzles from a variety of topics given in a variety of logics and styles. Contrarily, MPTP [Urb06] is a translation of the theorems from the Mizar Mathematical Library [GKN10] to the first logical with equivalence. We frequently employ the M40k and M2k sets for evaluations like the one mentioned in [KUMO18]. This is just one example of many other benchmarks with varying purposes; for example, the SMT-LIB institution's (BST+10) set of issues based on uses and gratification theories [28].

b) Superposition and Saturation

Saturated proof search is used to investigate the inferences drawn from the superposition calculus, which is the foundation for a significant number of the most prominent and cutting-edge current systems. Despite the fact that the two are not intrinsically related to one another, it is an extremely natural and common pairing. The goal of saturation algorithms is to produce a set of equations that is saturated; that is, a set that has been constructed in such a way that all deductions from the set up to duplication are included inside the set [BG01a]. [29]. Saturation can be achieved by a variety of different algorithms, including Given-clause methods are a popular approach taken at the clause level. In this type of algorithm, a "given clause" is chosen in some fashion and then added to a "processed" set that was initially empty. This was accomplished by carrying out all of the required to generate inferences with the members of the packaged set [KV13] [30].

There are many different ways to further classify not-yet-processed clauses; nonetheless, for this discussion, it is sufficient to point out that this group of clauses quickly expands as the processed set grows. At any step in the search process, simplifying and removing inferences can likewise have the effect of simplifying and deleting available clauses [31]. This framework enables powerful approaches for eliminating repetition, such as subsumption. It also has obvious advantages over other methods, such as the fact that it never explores any area of the search area twice. In return, the memory utilization of saturating systems can be large; however, when combined with the elimination of redundancy and the use of contemporary hardware, this issue is significantly less of a concern than it was in the past. Furthermore, the majority of realizations of saturation, such as given-clause algorithms, enforce a temporally linearisation of inferences [32]. This linearization makes certain advances difficult, such as parallelization at the level of proof search or learnt heuristics guiding [22].

c) Tableau-style systems

Although nothing stops a tableau system from using superposition1 - or saturation-style reasoning [DV98, Gie06], it is typically portrayed as different from, or even in opposition to, saturation/superposition systems [Hah01]. [1]. The goal of these frameworks is to construct closed tableaux, which are like a rooted tree and show a contradiction as evidence. If a conflict is found between two data gathered at different times on the same branch or its ancestors, the branch is considered to be dead. Tableau calculi come in a wide variety of forms and levels of sophistication. A powerful refinement[1] is connection tableau [LS01], which requires that any axioms added to a tableau must be specifically linked to the leaf literal of the current branch. In this way, we can begin with the (negated) hypothesis and proceed backward toward the axioms that disprove it, allowing for a highly distinctive goal-directed proof search. Since search in connectors is often retracing in nature, such as through iterative deepening[28], the fact that Tableau with the connectivity upgrade is not a concrete evidence system is not a fatal flaw. Current (connection) tableau techniques are often weaker than current quantum state systems, at least as tested by performance on big benchmark datasets like TPTP[13]. The lack of specialized equivalence handling, the lack of evidence confluence, or plain underdevelopment in comparison to state-of-the-art systems could all contribute to the performance disparity. Yet, when combined with other systems, they offer compelling advantages that hint at unrealized potential. To achieve its goals, proof search does not rely on sequential inference[28], can easily accommodate at least modal and fundamental structure logics [Waa01], and is goal-oriented.

C. Conclusion

It is increasingly common to use automated reasoning systems to solve difficult problems in mission-critical application domains. Even though the thinker has been built and thoroughly tested, there is still a great deal of work to be done in incorporating it and creating the right interfaces. Finally, we'd like to stress the great gap between being an intellectual prototype and a final solution when it comes to the development of industrial real-world applications. Although this will require a considerable investment of time, it has the potential to advance scholarly inquiry and development. To successfully bridge the gap between theory and practice, the field of automated reasoning has flourished. Various theorem-proving strategies are currently in use in automated deduction, including resolution, sequent calcium, natural deduction, matrix link approaches, term retraining, and mathematics induction. The strategies are implemented using a broad variety of logic formalisms, including first-order logic, type theories and larger logic, phrase or Horn reasoning, para logic, and so on

D. References

[1] F. Portoraro, "Automated Reasoning".

[2] "(PDF) The Automation of Mathematical Reasoning." https://www.researchgate.net/publication/346425938_The_Automation_of_Mathematical_Reasoning (accessed Nov. 12, 2022).

[3] J. Urban and J. Vyskočil, "Theorem proving in large formal mathematics as an emerging AI Field," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7788, pp. 240–257, 2013, doi: 10.1007/978-3-642-36675-8_13.
[4] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," SN Comput Sci, vol. 2, no. 3, pp. 1–21, May 2021, doi: 10.1007/S42979-021-00592-X/FIGURES/11.
[5] G. Lupyan, "Cognitive Penetrability of Perception in the Age of Prediction: Predictive Systems are Penetrable Systems," Rev Philos Psychol, vol. 6, no. 4, pp. 547–569, Dec. 2015, doi: 10.1007/S13164-015-0253-4.
[6] J. P. Bridge, "Number 792 Machine learning and automated theorem proving," 2010, Accessed: Nov. 12, 2022. [Online]. Available: http://www.cl.cam.ac.uk/
[7] A. Steen, D. Fuenmayor, T. Gleißner, G. Sutcliffe, and C. Benzmüller, "Automated Reasoning in Non-classical Logics in the TPTP World," Feb. 2022, doi: 10.48550/arxiv.2202.09836.
[8] "(PDF) ABox Abduction in the Description Logic." https://www.researchgate.net/publication/220532480_ABox_Abduction_in_the_Description_Logic (accessed Nov. 12, 2022).
[9] G. Singh, "Scalable Automated Reasoning for Programs and Deep Learning", doi: 10.3929/ethz-b-000445921.
[10] "Automated lazy metalearning in introspective reasoning systems".
[11] "(PDF) A Computational Model of Knowledge-Intensive Learning and Problem Solving." https://www.researchgate.net/publication/2612942_A_Computational_Model_of_Knowledge-Intensive_Learning_and_Problem_Solving (accessed Nov. 12, 2022).
[12] "A gentle introduction to automated reasoning - Amazon Science." https://www.amazon.science/blog/a-gentle-introduction-to-automated-reasoning (accessed Nov. 12, 2022).
[13] "First-order logic in Artificial Intelligence - Javatpoint." https://www.javatpoint.com/first-order-logic-in-artificial-intelligence (accessed Nov. 12, 2022).
[14] A. Fern, "First-Order Logic: Syntax and Semantics," 2020.
[15] Z. Manna and A. Pnueli, "Temporal Verification of Reactive Systems," Temporal Verification of Reactive Systems, 1995, doi: 10.1007/978-1-4612-4222-2.
[16] J. P. Bridge, S. B. Holden, and L. C. Paulson, "Machine learning for first-order theorem proving: Learning to select a good Heuristic," J Autom Reason, vol. 53, no. 2, pp. 141–172, 2014, doi: 10.1007/S10817-014-9301-5.
[17] M. Saqib Nawaz, M. Malik, Y. Li, M. Ikram, and U. Lali, "A Survey on Theorem Provers in Formal Methods".
[18] G. Reger, "Some Thoughts About FOL-Translations in Vampire", Accessed: Nov. 12, 2022. [Online]. Available: https://vprover.
[19] C. Benzmüller, X. Parent, and L. van der Torre, "Designing Normative Theories for Ethical and Legal Reasoning: LogiKEy Framework, Methodology, and Tool Support," 2020.

[20] A. Steen, “Extensional Paramodulation for Higher-Order Logic and its Effective Implementation Leo-III-Preprint”, Accessed: Nov. 12, 2022. [Online]. Available: https://alexandersteen.de/phd/.
[21] N. Balacheff, “PROOF TECHNOLOGY AND LEARNING IN MATHEMATICS: COMMON ISSUES AND PERSPECTIVES,” Perspectives (Montclair), pp. 349–365.