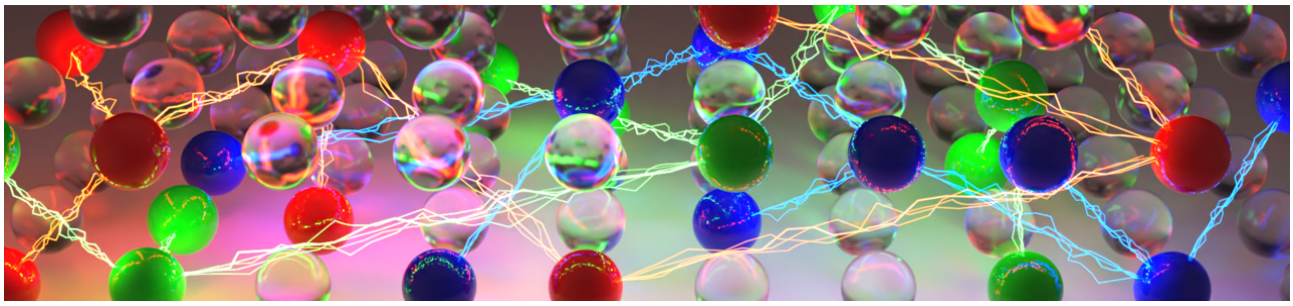
Call for Applications: Scalable Automated Reasoning

Research assistant / doctoral researcher (m/w/d) · German / English · Fall 2024



The young investigator group *Scalable Automated Reasoning* (SAtRes, `https://satres.kikit.kit.edu`) at Karlsruhe Institute of Technology (KIT), Germany, is looking for a research assistant (a.k.a. doctoral researcher, PhD student), beginning as soon as possible.

Introduction. Today's challenges of *automated reasoning* are diverse and manifold. They include concrete industrial applications, such as cooperative path planning of warehouse robots [17]; basic research, such as solving long-standing open problems of mathematics [6]; and some matters that are of relevance to society as a whole, such as verifying correctness or fairness properties of artificial intelligences [3] and of the logic enabling them [18]. Our aim is to study and advance essential tools at the core of automated reasoning, such as the satisfiability of propositional formulas (SAT), with a particular focus on parallel and distributed algorithms. Since tools like SAT solvers are crucial backbones of symbolic AI, they are essential for many relevant use cases such as hardware design [18], cryptography [7], and supporting and complementing subsymbolic AI methods (i.e., machine learning) [12].

The scalable SAT solving engine MallobSat [16] (developed by the thesis advisors) can achieve appealing speedups at thousands of parallel cores distributed over more than 100 machines. Each solver thread attempts to solve the formula at hand with a distinctly configured sequential *Conflict Driven Clause Learning* (CDCL) SAT solver—heuristically searching the space of variable assignments while continuously producing helpful *conflict clauses*. The main driver of scalability is a careful and compact approach to exchanging these conflict clauses across individual solver threads [13, 15].

In order to continue and expand on this fruitful line of research, we have identified a few promising subjects for doctoral research, outlined below. The exact subject to be pursued will be chosen together with the candidate.

Scalable Distributed SAT Solving. Today's sequential SAT solvers are not "pure" CDCL solvers but also perform numerous simplifications and transformations on the input formula, ahead of solving (*preprocessing*) and/or interleaved with solving (*inprocessing*) [2]. In modular distributed solvers like MallobSat, these pre– and inprocessing tasks are not parallelized but rather performed in a fully redundant fashion. This result in superfluous work performed and does not constitute an appropriate use of computational resources.

Our aim is to increase the scalability of distributed SAT solving by reducing the redundant work performed. One major undertaking is to replace the solvers' sequential and redundant pre–/inprocessing techniques with cooperative and scalable pre–/inprocessing. For instance, a significant result would be a distributed algorithm for *Bounded Variable Elimination* (BVE)—one of the most important SAT simplification techniques [2]. Another promising story is to combine tightly integrated shared-memory solvers [4] at the node-level with scalable clause sharing at the distributed level (cooperation with A. Biere). A third avenue worth exploring is to generalize the information exchange across solver threads beyond mere conflict clauses. As far as possible, all novel techniques should produce *LRAT proof information*, which allows to certify a result's correctness [9, 14].

Parallel and Distributed Model Checking. A principal application of SAT solving is *bounded model checking* (BMC), a central approach to hardware and software verification [18]. BMC aims to check correctness or safety properties of a system (e.g., a circuit, a piece of code, or some more abstract model) by exploring

the underlying transition system for up to k steps, where k is increased as far as possible or until a path to an offending state (*counter-example*) is found. We aim to conduct an extensive case study by integrating parallel and distributed SAT solving into state-of-the-art model checkers. Since verification can involve many independent or near-independent proof obligations, BMC is a prime application for on-demand scalable SAT solving with dynamic load balancing of many formulas [11]. In terms of applications, we anticipate at least one cooperation with affiliated researchers on one of the many applications of BMC such as hardware verification and electronic design automation, research software engineering, software verification, and/or security.

Parallel and Distributed SMT Solving. *Satisfiability Modulo Theories* (SMT) is an invaluable framework for formal verification and theorem proving. SMT extends SAT solving by the full power of First-Order Logic plus application-specific *theories* such as integer arithmetic or bit vectors. Most prior SMT parallelizations make use of a *portfolio* of few orthogonal approaches coupled with limited clause sharing across the integrated SAT solvers (e.g., [8]). Our aim is to transfer distributed SAT solving technology to parallel and distributed SMT solving. Important preparatory work for this endeavor is our previous work towards distributed *incremental* SAT solving [13] – a popular extension of SAT that is important for SMT solvers – as well as flexible load balancing of SAT tasks that run in parallel [11]. Since SMT-based verification tools by affiliated researchers like KeY [1] and KeYmaera X [5] are important parts of our overarching agenda, we expect according cooperations in order to engineer and transfer our improvements to actual applications.

Requirements and Expectations. A degree qualifying for doctoral studies in computer science (M.Sc. Computer Science or similar) with good grades is required. Prior specialization in formal methods, algorithms, and/or parallel and distributed computing is desirable. Good programming skills, in particular with C++, are highly recommended or need to be acquired rapidly. The candidate should be proficient in academic writing (e.g., via a high-quality master's thesis in English) and willing to work together with other researchers.

In line with the overarching methodology of Algorithm Engineering [10], important cross-sectional activities throughout the doctoral research are careful consideration of practical inputs and applications, theoretical analyses, practical performance engineering, and thorough experimental evaluations. Publications at high-profile international conferences or journals should be pursued. Significant involvement in teaching activities is not intended (but possible if desired). We do expect the candidate to supervise student qualification theses (roughly 1–2 in parallel) on topics that are directly relevant to the doctoral studies.

Organization. The candidate will be supervised by young investigator group (YIG) leader Dr. Dominik Schreiber and by Prof. Dr. Peter Sanders (in German and/or English). The position is full-time and compensated according to TVÖD E13. Initial funding is secured until 12/2027. An extension may be possible.

What we offer. We are fostering a positive and welcoming work environment where colleagues support and encourage each other. You will be able to focus on your doctoral studies and work together with other researchers to conduct highly visible research at an international level on topics of high practical relevance. Cooperations are planned both with KIT colleagues as well as with high-profile international researchers. As a part of your research, you will be able to interact with cutting-edge high-performance computing (HPC) environments, running your software on thousands of cores at once.

The Karlsruhe Institute of Technology (KIT) features Germany's oldest computer science faculty and is one of Germany's highest regarded universities in the area of computer science. Moreover, the KIT and our project in particular are part of the Helmholtz Association, Germany's largest scientific organization. Karlsruhe is a modestly sized university city (300k people) with a high number of students, offering a large variety of activities, in a sunny climate and near several remarkable landscapes (Black Forest, North Vosges, Palatinate Forest).

Contact. Please direct your informal application or any questions you might have to Dominik Schreiber ⟨ dominik.schreiber@kit.edu ⟩. See also: `https://satres.kikit.kit.edu`
You can share this advertisement easily via the following URL: `https://s.kit.edu/satres-phd`

References.

[1] Wolfgang Ahrendt et al. “The KeY tool: integrating object oriented design and formal verification”. In: *Software & Systems Modeling* 4 (2005), pp. 32–54. DOI: 10.1007/s10270-004-0058-x.

[2] Armin Biere, Matti Järvisalo, and Benjamin Kiesl. “Preprocessing in SAT Solving”. In: *Handbook of Satisfiability*. IOS Press, 2021, pp. 391–435. DOI: 10.3233/faia200987.

[3] Sumon Biswas and Hridesh Rajan. “Fairify: Fairness verification of neural networks”. In: *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE. 2023, pp. 1546–1558. DOI: 10.1109/icse48619.2023.00134.

[4] Mathias Fleury and Armin Biere. “Scalable Proof Producing Multi-Threaded SAT Solving with Gimsatul through Sharing instead of Copying Clauses”. In: *Pragmatics of SAT*. 2022.

[5] Nathan Fulton et al. “KeYmaera X: An Axiomatic Tactical Theorem Prover for Hybrid Systems”. In: *Automated Deduction - CADE-25*. Springer, 2015, pp. 527–538. ISBN: 978-3-319-21401-6. DOI: 10.1007/978-3-319-21401-6_36.

[6] Marijn J. H. Heule, Oliver Kullmann, and Victor Marek. “Solving and verifying the boolean pythagorean triples problem via cube-and-conquer”. In: *Theory and Applications of Satisfiability Testing (SAT)*. Springer. 2016, pp. 228–245. DOI: 10.1007/978-3-319-40970-2_15.

[7] Frédéric Lafitte, Jorge Nakahara Jr, and Dirk Van Heule. “Applications of SAT solvers in cryptanalysis: finding weak keys and preimages”. In: *Journal on Satisfiability, Boolean Modeling and Computation* 9.1 (2014), pp. 1–25. DOI: 10.3233/sat190099.

[8] Matteo Marescotti, Antti E. J. Hyvärinen, and Natasha Sharygina. “SMTS: Distributed, Visualized Constraint Solving.” In: *International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR)*. 2018, pp. 534–542. URL: http://easychair.org/publications/download/k7BQ.

[9] Dawn Michaelson et al. “Unsatisfiability proofs for distributed clause-sharing SAT solvers”. In: *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. Springer. 2023, pp. 348–366. DOI: 10.1007/978-3-031-30823-9_18.

[10] Peter Sanders. “Algorithm engineering–an attempt at a definition”. In: *Efficient Algorithms: Essays Dedicated to Kurt Mehlhorn on the Occasion of His 60th Birthday* (2009), pp. 321–340. DOI: 10.1007/978-3-642-03456-5_22.

[11] Peter Sanders and Dominik Schreiber. “Decentralized online scheduling of malleable NP-hard jobs”. In: *Int. European Conf. on Parallel Processing (Euro-Par)*. Springer. 2022, pp. 119–135. DOI: 10.1007/978-3-031-12597-3_8.

[12] André Schidler and Stefan Szeider. “SAT-based decision tree learning for large data sets”. In: *AAAI Conference on Artificial Intelligence*. Vol. 35. 5. 2021, pp. 3904–3912. DOI: 10.1609/aaai.v35i5.16509.

[13] Dominik Schreiber. “Scalable SAT Solving and its Application”. https://doi.org/10.5445/IR/1000165224. PhD thesis. Karlsruhe Institute of Technology, 2023.

[14] Dominik Schreiber. “Trusted Scalable SAT Solving with on-the-fly LRAT Checking”. In: *Int. Conf. on Theory and Applications of Satisfiability Testing (SAT)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 2024, 25:1–25:19. DOI: 10.4230/LIPIcs.SAT.2024.25.

[15] Dominik Schreiber and Peter Sanders. “MallobSat: Scalable SAT Solving by Clause Sharing”. In: *Journal of Artificial Intelligence Research (JAIR)* (2024). Presented at Pragmatics of SAT (PoS) 2024. DOI: 10.1613/jair.1.15827.

[16] Dominik Schreiber and Peter Sanders. “Scalable SAT Solving in the Cloud”. In: *Theory and Applications of Satisfiability Testing (SAT)*. Springer. 2021, pp. 518–534. DOI: 10.1007/978-3-030-80223-3_35.

[17] Pavel Surynek. “Unifying search-based and compilation-based approaches to multi-agent path finding through satisfiability modulo theories”. In: *International Symposium on Combinatorial Search (SoCS)*. Vol. 10. 1. 2019, pp. 202–203. URL: https://ojs.aaai.org/index.php/SOCS/article/download/18491/18282.

[18] Yakir Vizel, Georg Weissenbacher, and Sharad Malik. “Boolean Satisfiability Solvers and Their Applications in Model Checking”. In: *Proc. IEEE*. Vol. 103. 11. 2015, pp. 2021–2035. DOI: 10.1109/JPROC.2015.2455034.