Bachelor and Master Theses

Specialization: All Bachelor and Master Specializations

Remarks:

1. All theses must be written in English.
2. Usage of Latex (Beamer) is mandatory.
3. If conclusive results are obtained:
    a. they will be sent for publication at students' symposia, workshops, conferences
    b. teams of students will be encouraged to participate in innovation programs for students (e.g. Innovation Labs https://www.innovationlabs.ro)
4. **To work with me:**
    a. you must show and prove outstanding academic record (current GPA greater than 8.50)
    b. you must show disponibility in meeting regularly (weekly or bi-weekly) and tackling research problems
    c. be open to check the business opportunity of the problem you are working on, so you should show interest in entrepreneurship and innovation
5. I also supervise projects proposed by students, but they should be related to my interests and to the topics proposed:
    a. Formal Methods, in particular Static Software Verification;
    b. Automated Theorem Proving, in particular First-Order Theorem Proving;
    c. Software Engineering
    d. Symbolic Computation, in particular Polynomial Algebra;
    e. Distributed Computing, in particular Cloud and Big Data Computing.

Remark! Based on previous experience, many students did not show up, without announcing or having childish excuses, at the meetings in which we discuss the progress of their thesis. To avoid any misunderstandings, I state already that at 3 such happenings you will have to find another supervisor.

Nr	Topic	Observations
1.	Symmetry Breaking for the Cloud Resource Allocation Problem (1-2 theses)	Suppose you want to buy, at the lowest cost, virtual machines (VM) with certain CPU, memory, storage, from cloud providers which are geographically distributed. This is an NP-hard problem which can be formalized as a constraint satisfaction problem and solved using exact algorithms. The problem exhibits symmetries which makes the search for solution to consider already visited solutions, as well as parts of the search tree which are symmetric to already visited parts. The aim of this project is to implement symmetry breaking methods from the paper [1] in the MANeUveR framework (https://merascu.github.io/links/MANeUveR.html, [2]) in order to make the problem above amenable to be solved in practice. **Difficulty**: medium/high **Requirements**: *Programming*: Python; *Math*: computational logic, in particular the notions taught in the lectures Logic for Computer Science, Formal Methods in Software Development, Special Topics in Artificial Intelligence.
3.	Graph Neural Networks for combinatorial optimization problems (1-2 theses)	The project aims to apply Graph Neural Networks for optimization problems coming from cloud resource provisioning (see topic 1 above) [3]. **Difficulty**: high **Requirements**: *Programming*: Python; *Math*: computational logic, in particular the notions taught in the lectures Logic for Computer Science, Formal Methods in Software Development, Special Topics in Artificial

		Intelligence; graph theory, machine learning, operational research (optimization).
5.	Binarized Neural Networks. Training and Verification (2 theses; preferably students who worked together during university projects)	Deep learning is everywhere. It has been shown its practical application in a variety of fields, image recognition, natural language processing, recommendation systems, autonomous driving, just to name a few. Deep learning algorithms are mainly used as a black-box and hence difficult to debug. In fact, the main criticisms to deep learning algorithms are *uncertainty* and unexpected behavior on *adversarial examples*. When we talk about safety-critical systems, it is important that correctness guarantees exist. This leads to the application of *formal verification* to deep neural networks (DNNs), that is, given a DNN and a specification, is there a proof that the DNN satisfies the specification for all inputs? Not surprisingly, the main challenge of applying formal methods to the verification of DNNs is *scalability*. This is because verification is a non-trivial problem: DNNs are large (high number of neurons and layers) and involve activation functions which are non-linear and non-convex. These make the problem NP-complete. We offer three theses for studying three different verification approaches. The theses should contain a comprehensive state-of-the-art as well demo with at least one of the tools from the state-of-the-art. The demo will ensure reproducibility of the results obtained by state-of-the-art [4], [5]. **Difficulty**: high **Requirements**: *Programming*: Python; *Math*: Logic, linear algebra and statistics

References

[1] J. C. Régin and M. Rezgui, “Discussion about constraint programming bin packing models,” in *AAAI Workshop - Technical Report*, 2011.

[2] M. Eraşcu, F. Micota, and D. Zaharie, “Scalable optimal deployment in the cloud of component-based applications using optimization modulo theory, mathematical programming and symmetry breaking,” *J. Log. Algebr. Methods Program.*, vol. 121, 2021.

[3] Q. Cappart, D. Chételat, E. B. Khalil, A. Lodi, C. Morris, and P. Velickovic, “Combinatorial optimization and reasoning with graph neural networks,” *CoRR*, vol. abs/2102.09544, 2021.

[4] N. Narodytska, “Formal analysis of deep binarized neural networks,” in *IJCAI International Joint Conference on Artificial Intelligence*, 2018.

[5] N. Narodytska, S. P. Kasiviswanathan, L. Ryzhyk, M. Sagiv, and T. Walsh, “Verifying Properties of Binarized Deep Neural Networks,” in *Proceedings of the Thirty-Second {AAAI} Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New*, 2018, pp. 6615–6624.