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. In order to work with me, you have to show and prove disponibility in meeting regularly (weekly) (and tackling research problems).
5. I also supervise projects proposed by students. These should be related to my interests:
   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.

Nr	Topic	Observations
1.	Symmetry Breaking for the Cloud Resource Allocation Problem (1 thesis)	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.
2.	Benchmark problems for the constraints satisfaction problems (CSP) repository (1 thesis)	The project involves preparing and submitting existing constraints satisfaction problem for Cloud resource provisioning to the constraints satisfaction problems repository [3]. **Difficulty**: medium **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 topics 1 and 2 above) [4]. **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).

4.	Predicting the fastest method for constrained satisfaction/ optimization problems (2 theses)	Constrained optimization/satisfaction problems can be encoded in different logical theories (propositional logic, integers, reals, or combinations). The encoding influences the running time of the algorithms/tools solving the problem [5], [6]. We propose two theses: • One investigates the best algorithm/tool, from the computational time point of view, for solving the problem. • The other studies, implements, and performs experiments with incremental techniques for SAT/SMT solving for speeding up the existing algorithms/tools. **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.
4.	Semi-automatic generation of university course timetabling using hybrid methods. Case study Faculty of Mathematics and Informatics (West University of Timisoara) (1 thesis)	Curriculum-Based Course Timetabling (CB-CTT) consists of finding the best weekly assignment of university course lectures to rooms and time periods. A feasible schedule must satisfy a set of hard constraints and must also take into account a set of soft constraints, whose violation produces penalty terms to be minimized in the objective function [7]–[10], [11].
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 [12], [13]. **Difficulty**: high **Requirements**: *Programming*: Python; *Math*: Logic, linear algebra and statistics
6.	Synthesis of optimal numerical algorithms (1-2 theses)	Program synthesis is the automatic construction of software that provably satisfies a given specification (input and output condition). Given a specification of what a program should do, the synthesizer generates an implementation that satisfies this specification. The aim of the thesis is to study the possibility of the synthesis of algorithms (e.g. reciprocal, square root, reciprocal square root of numbers) suitable for hardware implementations. The main characteristic of these algorithms is that they do not contain the division operation, which is expensive. The experiments will be conducted in Mathematica [14]. **Difficulty**: high **Requirements**: *Programming*: Mathematica; *Math*: computational logic, computer algebra

7.	Comparative study of formal analysis methods for biological networks involved in the development of resistance of microorganisms to antibiotics. (1-2 theses).	Formal analysis of biological networks has the potential of developing reliable and efficient methods and tools for patterns (motifs) identification which could help in *understanding the mechanisms behind complex phenomena* (e.g. antimicrobial resistance) [15]. **Difficulty**: high **Requirements**: *Programming*: Python; *Math*: basic abstract algebra, computational logic, in particular the notions taught in the lecture Formal Methods in Software Development. *Interest* in bioinformatics.
8.	Investigation of symmetry breaking methods for formal analysis methods for biological networks involved in the development of resistance of microorganisms to antibiotics. (1-2 theses).	Formal analysis of biological networks has the potential of developing reliable and efficient methods and tools for patterns (motifs) identification which could help in *understanding the mechanisms behind complex phenomena* (e.g. antimicrobial resistance). As it is an intractable task, we aim to study the usability of symmetry breaking methods for speeding it up. **Difficulty**: high **Requirements**: *Programming*: Python; *Math*: basic abstract algebra, computational logic, in particular the notions taught in the lecture Formal Methods in Software Development. *Interest* in bioinformatics.
9.	Construction waste reduction (2-3 theses)	The construction industry consumes about half of all resources extracted from the earth and is responsible for more than a third of global energy consumption and emissions, excluding cement production accounts for 7% of CO2 emissions. Reducing these environmental damage requires a radical approach to the perception of construction primarily by the beneficiaries. We aim to build a community around an application / platform to assist people in the construction or renovation of a building, apartment, etc. This should help the beneficiaries to understand the construction stages, to calculate the quantities of necessary materials, to present them with environmentally friendly materials, to bring together experts and beneficiaries but also owners and future owners. The result would be a reduction in the consumption of materials (by precise calculation of quantities, putting beneficiaries in contact with distributors who also offer environmentally friendly materials) and a sustainable living space. **Difficulty**: medium **Requirements**: *Programming*: Python; *appetite for learning* new technologies (e.g. Angular, Node.js, Firebase, Machine/Deep Learning)

References

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

[2] M. Erașcu, 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] A. Iovescu, “Benchmark Problems for the Constraints Satisfaction Problems Repository,” *STUDMath-IT 2020*, 2020.

[4] 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.

[5] M. Erascu, F. Micota, and D. Zaharie, “Influence of Variables Encoding and Symmetry Breaking on the Performance of Optimization Modulo Theories Tools Applied to Cloud Resource Selection,” in *LPAR-22 Workshop and Short Paper Proceedings*, 2018, vol. 9, pp. 1–14.

[6] V. Scott, Joseph and Poupart, Pascal and Ganesh, “An algorithm selection approach for qf fp solvers,” *17th Int. Work. Satisf. Modul. Theor.*, 2019.

[7] A. Bettinelli, V. Cacchiani, R. Roberti, and P. Toth, “An overview of curriculum-based course timetabling,” *TOP*, vol. 23.

[8] A. Lemos, P. T. Monteiro, and I. Lynce, “Minimal Perturbation in University Timetabling with Maximum Satisfiability,” in *Integration of Constraint Programming, Artificial Intelligence, and Operations Research - 17th International Conference, {CPAIOR} 2020, Vienna, Austria, September 21-24, 2020, Proceedings*, 2020, pp. 317–333.
[9] A. Lemos, P. T. Monteiro, and I. Lynce, “Disruptions in timetables: a case study at Universidade de Lisboa,” *J. Sched.*
[10] A. Lemos, F. S. Melo, P. T. Monteiro, and I. Lynce, “Room usage optimization in timetabling: A case study at Universidade de Lisboa,” *Oper. Res. Perspect.*
[11] H. Babaei, J. Karimpour, and A. Hadidi, “A survey of approaches for university course timetabling problem,” *Comput. Ind. Eng.*, vol. 86, pp. 43–59, 2015.
[12] N. Narodytska, “Formal analysis of deep binarized neural networks,” in *IJCAI International Joint Conference on Artificial Intelligence*, 2018.
[13] 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.
[14] M. Erascu and H. Hong, “Synthesis of optimal numerical algorithms using real quantifier elimination (Case study: Square root computation),” in *International Symposium on Symbolic and Algebraic Computation, ISSAC ’14, Kobe, Japan, July 23-25, 2014*, 2014, pp. 162–169.
[15] H. Kugler, S. J. Dunn, and B. Yordanov, “Formal Analysis of Network Motifs,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018.