Bachelor and Master Theses

Specialization: All Bachelor and Master Specializations

Remarks:

1. All theses must be written in English.
2. Usage of Latex is mandatory.

Nr	Tema	Detalii
1.	Optimization techniques using constraint programming solvers (Gurobi, CPLex, Google OR-Tools) for the deployment of component-based applications in the Cloud.	Suppose you want to deploy your component-based application in the Cloud at the lowest price This can be formalized as an optimization problem and solved usinng constraint programming solvers. The solution will be included into a reccomedation engine. **Requirements**: *Programming*: Python; *Math*: Logic
2.	Optimization techniques with priorities in constraints SMT solvers	Suppose you want to buy, at the lowest cost, virtual machines (VM) with certain CPU, memory, storage, from cloud providers which are geographically distributed. You don't know preciselly the chacteristics of the machines you want to buy but you know you need a machine for installing games and storing your collection of movies and music. Hence it's most likely you give priority to memory and storage VMs. But what preciselly the characteristics of these VMs and from which Cloud Provider should you buy them in order that you pay the lowest price? In this thesis we aim to ask these questions by developing optimization techniques in which the constraints might not be fully specified but have a certain priority. These will be integrated into a reccomendation engine. **Requirements**: *Programming*: Python; *Math*: Logic
3.	Predicting the fastest method for optimization	As seen above, there might be different optimization techniques for a problem. But which one should be used on optimization problem? We will try to solve the problem using machine learning techniques. **Requirements**: *Programming*: Python; *Math*: Basic linear algebra and statistics
4.	Verification of Deep Neural Networks	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. New results are also welcomed. **Requirements**: *Programming*: Python; *Math*: Logic, linear algebra and statistics
5.	Synthesis of optimal numerical algorithms	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. **Requirements**: *Programming*: Mathematica; *Math*: Logic
6.	Deep learning techniques in autonomous driving	We will try to give solutions to the following tasks of self-driving Car: (1) *Localization and Mapping* (Where am I?); (2) *Scene Understanding* (Where is everyone else?); (3) *Movement Planning* (How do I get from A to B?); (4) *Driver State* (What's the driver up to?). A variate number of theses are offered. **Requirements**: *Programming*: Python/TensorFlow; *Math*: Statistics, Linear Algebra