Bachelor and Master Theses

Specialization: IA, IR, IE, AIDC, IACD, IASTE

Remarks:

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

Nr	Tema	Detalii
1.	Detecting fake news	The word *post-truth i*s considered by Oxford Dictionaries Word of the Year 2016. The word is an *adjective relating to or denoting circumstances in which objective facts are less influential in shaping public opinion than appeals to emotion and personal belief*. There is no doubt that fake news influence our daily life in a subtle way all of us being subject to manipulation. This thesis aims to study and develop methods and tools for detecting fake news. More precisely: (1) we will do literature review (2) propose a methodology for detecting fake news (3) implement a platform which allows reporting, respectively detecting fake news/sites. **Requirements**: Java/C#/Python
2.	Cloud Computing Resources Crawler	This thesis aims at the design and implementation of a tool which provides information about Cloud Providers and provided resources. It must provide saving and updating information in a database. It will get from the providers site the available default configurations for offered systems (e. g. CPUs number, available memory and disk space, price) as well as other information like OS installed, number of IP addresses, transfer rate, etc. A motivation of chosen technologies for the implementation will also be provided. **Requirements**: Java/C#/Python
3.	Application Description Module	This must design and implement a tool which provides the end user the capability to describe two types of application constraints: (1) virtual machine (VMs) configuration constraints (e.g. my application needs VMs with 2vCPU and maximum 2GB storage); (2) inter-dependency constraints (e.g. load balancing component should not be placed on the same machine as the gateway component and the SQL server). The constraints of type (1) will be used to lease VMs from certain Cloud Providers (CPs) and the information about configuration of acquired VMs together with constraints of type (2) will stored in order to be consulted other times and to obtain a profile of the application. **Requirements**: Java/C#/Python
4.	Recommendation Engine	A recommendation system seeks to predict user preferences, possibly based on some information that the user provided and under certain constraints. We aim at the design and implementation of a recommendation engine dedicated to cloud resources retrieval but also other interesting application

		modules can be considered. The backend of the engine will be an optimization engine which will use different techniques: pattern matching, exact methods, heuristic methods, etc.
5.	Timekeeping system for Department of Computer Science, West University Timisoara	Every employer of our department has to introduce his working hours (per week) specifying the time spent on different activities (teaching, preparing classes, research, administrative matters). Currently, this is done manually by adding information in an Excel file. This thesis should make this process automated. The user authenticates himself, adds hours in different categories every week. Updating/deleting information should be available also. At the end a pretty-printed file should be generated. **Requirements**: Java/C#/Python
6.	Transforming informal text to formal text. Applications to software specification and general first-order logic structures.	Program documentation (specification) is typically written in plain text. Specifications are useful for program verifiers if they are written in a formal language, most notably first-order logic. This thesis aims to bridge this gap by using natural language preprocessing and natural language understanding techniques. **Requirements**: Java/C#/Python
7.	Using Symbolic Computation to speed-up Satisfiability Checkers	Symbolic Computation and Satisfiability Checking both develop powerful algorithms for determining exact solutions for complex problems but using different algorithmic and technological approaches. Although the two communities solve very similar problems they rarely interact. A recent initiative[1] intents to make the two initiatives collaborate. In this thesis, we try to bridge the gap between the methods used in the two approaches by incorporating the simplification rules proposed in [1], [2], which speed-up the quantifier elimination algorithms by Cylindrical Algebraic Decomposition from QEPCAD-B [3] and Mathematica [4], into state-of-the-art Satisfiability Modulo Theories (SMT) solvers, e.g, [5]. *Bibliography*: [1], [2], [6].
8.	Optimization techniques with priorities in constraints using exact or heuristic methods	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. The optimization techniques should be designed using (1) SMT solvers (exact results); (2) heuristics (approximate results). On this thesis should work two students, one for the SMT part,

[1] http://www.sc-square.org/

		the other on the heuristics. *Bibliography*: [7], [8].
9.	Privacy by design	Security plays a central role in the development of distributed software systems. The integration of security engineering into a model-driven software development approach has many advantages, e.g. security requirements can be formulated and integrated into system designs at a high level of abstraction or the model information can be used to detect and to correct design errors or to verify the correctness of the mapping between requirements and their realization in a design. The role of this thesis is to use SecureUML, an UML-based modelling language for model-driven security, to model and prevent errors during the realization of access control policies and enable the technology independent development of secure systems. The benefits of this language will be exemplified on an industrial case study (tax fraudster application) from the EU project H2020-DICE (http://www.dice-h2020.eu/). *Bibliography*: [9], [10], https://www.enisa.europa.eu/publications/privacy-and-data-protection-by-design

Bibliography

[1] 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.

[2] M. E. and H. Hong, "Real quantifier elimination for the synthesis of optimal numerical algorithms (Case study: Square root computation)," *J. Symb. Comput.*, vol. 75, pp. 110–126, 2016.

[3] C. W. Brown, "QEPCAD B: A Program for Computing with Semi-algebraic Sets Using CADs," *SIGSAM Bull.*, vol. 37, no. 4, pp. 97–108, Dec. 2003.

[4] S. Wolfram, *The Mathematica book 5. ed.* Wolfram-Media, 2003.

[5] L. De Moura and N. Bjørner, "Z3: An Efficient SMT Solver," in *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'08/ETAPS'08)*, 2008, pp. 337–340.

[6] E. Ábrahám, "Proceedings of the 2015 {ACM} on International Symposium on Symbolic and Algebraic Computation, {ISSAC} 2015, Bath, United Kingdom, July 06 - 09, 2015," in *Proceedings of the 2015 {ACM} on International Symposium on Symbolic and Algebraic Computation, {ISSAC} 2015, Bath, United Kingdom, July 06 - 09, 2015*, 2015, pp. 1–6.

[7] M. E. and F. M. and D. Zaharie, "A Scalable Hybrid Approach for Applications Placement in the Cloud," in *Grid, Cloud High Performance Computing in Science (ROLCG), 2015 Conference*, 2015, pp. 1–4.

[8] V. Casola, A. De Benedictis, M. Erascu, J. Modic, and M. Rak, "Automatically Enforcing Security SLAs in the Cloud," *IEEE Trans. Serv. Comput.*, no. Special Issue on Security and Dependability of Cloud Systems and Services, 2016.

[9] T. Lodderstedt, D. A. Basin, and J. Doser, “SecureUML: {A} UML-Based Modeling Language for Model-Driven Security,” in *{UML} 2002 - The Unified Modeling Language, 5th International Conference, Dresden, Germany, September 30 - October 4, 2002, Proceedings*, 2002, vol. 2460, pp. 426–441.

[10] D. Basin, M. Clavel, M. Egea, M. A. G. De Dios, and C. Dania, “A model-driven methodology for developing secure data-management applications,” *IEEE Trans. Softw. Eng.*, vol. 40, no. 4, pp. 324–337, 2014.