A) Research-related Topics with possible internships in research centers

Context: Development of algorithms and tools for the verification of reactive systems and the synthesis of such systems

Reactive systems are computer systems that maintain a continuous interaction with an environment in order to control it and to ensure that some properties are enforced. The controller of a lift system is a typical reactive system: it reacts to stimuli from the environment (e.g. an user push a bottom to call the lift) and it controls the lift in order to fulfill a given specification (e.g. whenever a user asks for the lift at floor i, the lift eventually reach floor i).

Reactive systems are notoriously difficult to develop correctly. Difficulties arise because of concurrency, real-time constraints, parallelism, etc. Because they are difficult to develop and often used in safety critical situations (e.g. control of automatic transportation devices, embedded systems in plane or aeronautics, etc), a large research effort has been devoted to the development of computer aided methods able to certify the correctness of reactive systems. The most successful of those techniques are based on the so-called model-checking approach. Model-checking relies on algorithms that receive as inputs mathematical models (for example in the form of a finite state automaton) of the behaviours of the reactive system and the environment in which it is embedded, a logical formula defining the expected correct behaviour of the system within its environment. Then the algorithm verify that all the possible executions of the model are satisfying the formula.

A more ambitious goal is to synthesis correct reactive systems directly from their specifications. Our research group at ULB has done significant progress recently on this problem (see http://www.antichains.be/). Those techniques are based on ideas coming from automata theory, temporal logics and two-player games on graphs. In this context, we propose the three following master's thesis topics.

Please note that:

- all the proposed master's thesis can be coupled to an internship in a research center abroad

- all the topics are proposed by JF Raskin and G. Geeraerts. In practice, the students will be either co-supervised, or supervised by either of them.

1) Verification of Communicating Systems by Learning and Abstraction Refinement

Asynchronous distributed systems (such as client/server applications) are notoriously difficult to design correctly. This is due to the complex (and sometimes unforeseen) interactions resulting from the asynchronous communication. Therefore, there is a growing need for automatic verification tools capable of analyzing such systems.

This project considers asynchronous distributed systems that are modeled by so-called *FIFO systems*. The latter are given by a set of local, finite-state machines that communicate via reliable and unbounded FIFO channels. FIFO systems have the expressive power of Turing machines and, hence, their verification is undecidable. As for other classes of undecidable models, a way out of this dilemma is provided by the *abstract/check/refine* paradigm: compute an approximation of the system, (b) determine whether the system is correct when the approximation is sound, and (c) refine the approximation otherwise. This paradigm has been instantiated in various ways, such as *counterexample-guided abstraction refinement* [7, 8] and *learning-based verification* [9, 11]. In [12], we proposed new refinement algorithms for counterexample-guided abstraction refinement of FIFO systems, and we implemented them in the McScM tool [13].

This proposal aims at extending McScM with learning-based verification algorithms. The candidate is expected to:

1. Implement a learning algorithm in the automata library (latticeAutomata) used by McScM.
2. Design learning-based verification algorithms (inspired from [9, 10, 11]) and implement them in McScM. Learning may be used to generate invariants for the whole system, or to generate *path invariants* for spurious counterexample paths. If time permits, both approaches will be considered and evaluated.

Requirements: Good programming skills, readiness to acquire basic knowledge on formal verification techniques (automata theory, abstraction refinement, learning algorithms).

Together wih this topic, an internship is offered at the LaBRI, in Bordeaux (with a salary of 436 € per month), under the supervision of Prof. Grégoire Sutre. See http://www.labri.fr/perso/sutre/Master-Proposal-2012-1.php

Contact: Gilles Geeraerts (gigeerae@ulb.ac.be), Alexander Heußner (alexander.heussner@ulb.ac.be) and Jean-François Raskin (jraskin@ulb.ac.be)

2) Implementation of a prototype for the verification of programs written using Apple's Grand Central Dispatch

To make the development of efficient multi-core applications easier, libraries, such as Grand Central Dispatch, have been proposed. When using such a library, the programmer writes so-called blocks, which are chunks of codes, and dispatches them, using synchronous or asynchronous calls, to several types of waiting queues. A scheduler is then responsible for dispatching those blocks on the available cores. Blocks can synchronize via a global memory. This programming paradigm has notably being used in the Apache webserver.

This framework helps the programmer write efficient code in a nutshell. Unfortunately, the runtime behaviour of such programs is hard to grasp, because of the many possible interactions between blocks, queues, and so forth...

In a recent work, we have proposed algorithms to translate GCD programs to more basic models (such as Petri nets), which then allows to verify that those programs respect some properties (such as mutual exclusive access to shared data).

The main objective of this master's thesis is to implement some or all of those algorithms and to benchmark them on realistic GCD examples. To this aim, the student will have to rely on compiling tools such as GNU LLVM to extract the flow graph of the code, and so forth...

Requirements: good programming skills, good command of compiler writing techniques. Basic knowledges in formal methods and verification is an advantage.

An internship in a research centre is possible (to be discussed).

Contact: Gilles Geeraerts (gigeerae@ulb.ac.be) Alexander Heußner (alexander.heussner@ulb.ac.be) and Jean-François Raskin (jraskin@ulb.ac.be)

3) Synthesis of a representative system using LTL synthesis tools

Several recent research works are tackling the synthesis problem for reactive systems (see [1,2,3] and references there). In this project, the candidate will try to apply several tools developed by research teams around the world and at ULB on a case study. The synthesis tools will be used to synthesize a reactive system from its specification. The result of the synthesis is a finite state transducer. This finite state transducer needs to be translated into a program. In our case study, we will use lego Mindstorms as a representative platform for embedded systems. The translation will be automated by a program to be developed within the project.

Requirements: good programming skills, readiness to acquire the basic knowledge underlying the synthesis algorithms (automata theory, temporal logics and two-player

games).

Contact: Gilles Geeraerts (gigeerae@ulb.ac.be) and Jean-François Raskin (jraskin@ulb.ac.be)

References

[1] Emmanuel Filiot, Naiyong Jin, and Jean-François Raskin. An Antichain Algorithm for LTL Realizability. In CAV'09, LNCS 5643, Springer, pp.263-277, 2009.

[2] Emmanuel Filiot, Nayiong Jin, and Jean-François Raskin. Compositional Algorithms for LTL Synthesis. To appear in ATVA'10, LNCS, Springer, 15 pages, 2010.

[3] Nir Piterman, Amir Pnueli, Yaniv Sa'ar: Synthesis of Reactive(1) Designs. VMCAI 2006: 364-380

[4] Gilles Geeraerts, Gabrial Kalyon, Tristan Le Gall, Nicolas Maquet and Jean-François Raskin. Lattice-Valued Binary Decision Diagrams. In the Proceedings of ATVA 2010, Lecture Notes in Computer Science, volume 6252, Springer Verlag.

[5] Giorgio Delzanno, Arnaud Sangnier, Gianluigi Zavattaro: Parameterized Verification of Ad Hoc Networks. CONCUR 2010: 313-327

[6] Geeraerts, Heußner, Raskin, Queue-Dispatch Asynchronous Systems. Downloadable at http://arxiv.org/abs/1201.4871v1

[7] E. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith. *Counterexample-guided Abstraction Refinement for Symbolic Model Checking*. Journal of the ACM, 50(5):752–794, 2003.

[8] T. A. Henzinger, R. Jhala, R. Majumdar, and G. Sutre. *Lazy Abstraction*. In Proc. POPL'02, pages 58–70. ACM Press, 2002.

[9] A. Vardhan, K. Sen, M. Viswanathan, and G. Agha. *Learning to Verify Safety Properties*. In Proc. ICFEM'04, LNCS 3308, pages 2747–289. Springer, 2004.

[10] A. Vardhan, K. Sen, M. Viswanathan, and G. Agha. *Actively Learning to Verify Safety for FIFO Automata*. In Proc. FSTTCS 2004, LNCS 3328, pages 494–505. Springer, 2004.

[11] P. Habermehl and T. Vojnar. *Regular Model Checking Using Inference of Regular Languages*. Electr. Notes Theor. Comput. Sci. 138(3): 21–36, 2005.

[12] A. Heussner, T. Le Gall, and G. Sutre. *Extrapolation-based Path Invariants for Abstraction Refinement of Fifo Systems*. In Proc. SPIN'09, LNCS 5578, pages 107–124. Springer, 2009.

[13] A. Heussner, T. Le Gall, and G. Sutre. *McScM: A General Framework for the Verification of Communicating Machines*. In Proc. TACAS'12, LNCS 7214, Springer, 2012. To appear.

B) Industry related topics

The “formal methods and verification” group has contact with the following companies that are active in the field of embedded systems in Belgium:

- Macq – Development of control systems for traffic monitoring – see http://www.macqel.be

- Nuance – GPS and voice synthesis/recognition applications - see http://www.nuance.com/

Students who are interested in an internship in one of those companies (with a potential master's thesis topic) should contact Gilles Geeraerts (gigeerae@ulb.ac.be).