ERCIM "Alain Bensoussan"
Fellowship Scientific Report

Fellow: Valerio SENNI
Visited Location : LORIA-INRIA, Villers-les-Nancy, France
Duration of Visit: 12 months

I - Scientific activity ([…] is a citation of a paper in Sec. II)

My research interests are on automating the analysis of software systems. Software analysis tasks are challenging mainly for two (orthogonal) reasons: (1) the systems of interest are often concurrent (e.g. protocols, reactive systems) and have a very complex behaviour, which motivates the use of automatic techniques, and (2) their verification require solvers that are able to discharge proof obligations involving several different mathematic concepts at once, such as data-structures together with arithmetic constraints and other mathematical abstractions, such as size- or set-based abstractions. My work during the fellowship has focused on both issues.

With respect to the first issue, I extended my previous research on the use of program transformation techniques for the analysis of logic programs. In particular, I focused on the problem of verifying properties of *infinite-state* systems, which have attracted great interest in the literature. In the transformation-based approach (constraint) logic programs are used to model the concurrent systems to be verified and unfold/fold program transformations provide a calculus that allows us to deduce consequences from this logic programs. The analysis of concurrent infinite-state systems is reduced to the analysis of the constraint logic programs used to model these systems. My contributions on this topic have been: (1) the study of new generalization strategies [a] to improve the precision of our (previously developed) verification framework, (2) the formalization of a transformation calculus that uses real relaxations in the analysis of systems that work on integer-valued linear constraints [b], thus allowing the use of fast and efficient solvers on reals, and (3) the adaptation of our transformation technique to act as a preprocessing technique to improve the precision of the analysis performed by other verification tools [c]. The effectiveness of (1) and (3) have been attested by running some large benchmarks and comparing our MAP tool (available online) with other verification tools.

With respect to the second issue, I acquired new scientific skills. I studied the development of satisfiability procedures, to be used in Satisfiability Modulo Theories (SMT) solvers, based on the superposition calculus. In this setting, the problem of designing the satisfiability procedures is often addressed with success by using combination techniques à la Nelson-Oppen, which focus on the construction of satisfiability procedures for smaller theories and on the development of general results to address their compositions into more complex ones. Problems arise when one considers (more interesting) combinations involving theories whose signatures are *non-disjoint* (e.g. when we consider theories sharing some algebraic constraints). To handle non-disjoint unions one needs to rely on powerful results, based on semantic properties of the considered theories, which often require complex checks on the component theories.

My contributions to this topic have been twofold [d]. First, I proved a modular termination result for extending the applicability of the superposition calculus to theories that share a theory of counter arithmetic. This generalizes, to the non-disjoint case, recent results in the literature, where the standard superposition calculus and signature-disjoint theories are considered. This result allows us to drop some of the complex conditions required by the combination frameworks. Second, I proved a general compatibility result that allows us to use superposition-based satisfiability procedures into combination frameworks à la Nelson-Oppen. This result provides less, simpler, and automatically checkable conditions for combinability. As a consequence we are able to obtain satisfiability in two ways: (1) by a uniform approach based on superposition (e.g., for theories of data structures) and by combination with other solvers for theories which are not 'superposition-friendly' (e.g. for theories of arithmetic). We are currently working [e] on the development of further results on *non-convex* theories, that are of great interest when considering commonly used data-structures such as arrays.

Regarding the research performed during the two REPs I will give further details in Sec.IV.

II- Publication(s) during your fellowship

[a]. Fioravanti, F., Pettorossi, A., Proietti, M., Senni, V.: *Generalization Strategies for the Verification of Infinite State Systems*. Accepted for publication in *Theory and Practice of Logic Programming*, to appear.

We present a method for the automated verification of temporal properties of infinite state systems. Our verification method is based on the specialization of constraint logic programs (CLP) and works in two phases: (1) in the first phase, a CLP specification of an infinite state system is specialized with respect to the initial state of the system and the temporal property to be verified, and (2) in the second phase, the specialized program is evaluated by using a bottom-up strategy. The effectiveness of the method strongly depends on the generalization strategy which is applied during the program specialization phase. We consider several generalization strategies obtained by combining techniques already known in the field of program analysis and program transformation, and we also introduce some new strategies. Then, through many verification experiments, we evaluate the effectiveness of the generalization strategies we have considered. Finally, we compare the implementation of our specialization-based verification method to other constraint-based model checking tools. The experimental results show that our method is competitive with the methods used by those other tools.

[b]. Fioravanti, F., Pettorossi, A., Proietti, M., Senni, V.: *Using Real Relaxations During Program Specialization*. Submitted to LOPSTR 2011.

We propose a program specialization technique for locally stratified CLP(Z) programs, that is, logic programs with linear constraints over the set Z of the integer numbers. For reasons of efficiency our technique makes use of a relaxation from integers to reals. We reformulate the familiar unfold/fold transformation rules for CLP programs so that: (i) the applicability conditions of the rules are based on the satisfiability or entailment of constraints over the reals, and (ii) every application of the rules transforms an old program into a new program with the same perfect model constructed over Z. Then, we introduce a strategy which applies the transformation rules for specializing CLP(Z) programs with respect to a given query. Finally, we show that our specialization strategy can be applied for verifying properties of infinite state reactive systems specified by constraints over the integers.

[c]. Fioravanti, F., Pettorossi, A., Proietti, M., Senni, V.: *Improving Reachability Analysis of Infinite State Systems by Specialization*. Submitted to Reachability Problems 2011.

We consider infinite state reactive systems specified by linear constraints over the integers and we address the problem of verifying safety properties of these systems by applying reachability analysis techniques. We propose a method based on program specialization, for preprocessing the given reactive system so that the backward and forward reachability analyses become more effective. For backward reachability our method consists in: (i) specializing the reactive system with respect to the constraints characterizing the initial states, and then (ii) applying to the specialized system a fixpoint computation technique which works backwards from the property to be proved. Symmetrically, for forward reachability our method consists in: (i) specializing the reactive system with respect to the constraints characterizing the property to be proved, and then (ii) applying to the specialized system a fixpoint computation technique which works forwards from the initial states. We prove the correctness of our method and we describe an implementation based on the MAP transformation system and the

ALV verification system. Through various experiments performed on several infinite state protocols, we show that our specialization-based verification technique considerably increases the number of successful verifications, without significantly deteriorating the overall verification time.

[d]. Ringeissen, C., Senni, V. *Modular Termination and Combinability for Superposition Modulo Counter Arithmetics*. Submitted to FroCoS 2011.

Modularity is a very desirable property in the development of satisfiability procedures. In this paper we are interested in using a dedicated superposition calculus to develop satisfiability procedures for (unions of) theories sharing counter arithmetic. In the first place, we are concerned with the termination of this calculus for theories representing data structures and their extensions. To this purpose, we prove a modularity result for termination which allows us to turn our superposition calculus into a satisfiability procedure for combinations of data structures. In addition, we present a general combinability result that permits us to use our satisfiability procedures into a non-disjoint combination method à la Nelson-Oppen without loss of completeness. This latter result is useful whenever data structures are combined with theories of arithmetic for which superposition is not applicable.

[e] Ringeissen, C., Senni, V. *Untitled Manuscript.* 2011.

We are studying the extension of the results in [d] to the case of non-equational and non-convex theories. The major and motivating examples are theories modelling arrays and sets. The first, is a widely used data-structure and we are interested into developing satisfiability procedures that allow to take into account properties of the arrays involving numerical constraints. The second, is a widely used mathematical abstraction in software verification.

III -Attended Seminars, Workshops, and Conferences

FLoC 2011, Edinburgh, UK, July 9-21, 2010.
FroCos 2011, Saarbrücken, Germany, October 5 - 7, 2011.
Several seminars at the LORIA-INRIA laboratory.

IV – Research Exchange Programme (12 month scheme)

1st REP: Visit to professor Viktor Kuncak, head of the LARA (Lab for Automated Reasoning and Analysis) group at EPFL (École Polytechnique Fédérale de Lausanne), Lausanne, Switzerland, from 20 to 31 of August, 2010. I gave an introductory talk on program transformation where I summarized the applications of this techniques I am interested in. Namely: Model Checking, Theorem Proving, and Program Synthesis. I had some interesting feedback on the use of program transformation for: (1) inductive theorem proving (a classical application of transformation), (2) optimization of test generation programs (optimization is another classical application of transformation), and (3) synthesis of theory-specific solvers (synthesis is an application that I previously explored, but this context of use is not much explored). I worked on point (2) and I have some interesting results regarding the speed-up of data-structure generating programs (trees, graphs, etc.). These results have not been yet published and deserve some further investigation to make the approach systematic. Points (1) and (2) are extremely interesting and I expect to have further interaction with people at EPFL in the next future. A copy of the slides of the talk can be retrieved at my home page:

http://www.disp.uniroma2.it/users/senni/program-transformation_EPFL-2010.pdf

2nd REP: Visit to professor John Gallagher, Roskilde University, Roskilde, Denmark, from March 26 to April 2, 2011. The group hosting me in Roskilde is very expert in the topics regarding my research on program transformation, therefore my visit in Roskilde was much more technical and focused on comparing the respective approaches for the verification of infinite-state reactive systems. Professor Gallagher organized for me a talk, hosted by professor Flemming Nielson at DTU (Technical University of Denmark), Copenhagen, where I could illustrate the recent advances developed in papers [a] and [c]. I had very interesting feedback both in Roskilde and in Copenhagen. Namely: (1) the request of using our analysis techniques for verifying cryptographic protocols, (2) a request of further joint study on the useful polyvariant behaviour of out verification tool, and (3) some technical questions related to abstract-interpretation and the encoding of mu-calculus in logic programming. I worked on (2) while I was in Roskilde and I am still working on the topic. Regarding (3) I have current interaction with people in Copenhagen. Point (1) is very interesting and will be considered in the following months. A copy of the slides of the talk can be retrieved at my home page:
http://www.disp.uniroma2.it/users/senni/DTU_march2011.pdf