Doctoral Thesis Report

Student: David Hauzar

Thesis title: Towards Static Analysis of Languages with Dynamic Features

Reviewer: Lukáš Holík, Brno University of Technology, FIT

There is a high demand for methods of software analysis and the research in the area is very lively. Dynamic languages such as PHP are widely used, often in security critical applications where functional correctness is also a security issue. However, these languages have not received much attention yet in the software analysis community, perhaps due to the additional complexity caused by the dynamic features. The topic of the thesis is therefore well motivated.

Dynamic programming languages allows certain information defined statically in classical languages to be obtained at runtime. In the case of PHP, which is the focus of the thesis, this information may be (1) types of variables, (2) control flow, (3) definitions of objects, or (4) dimensions and sizes of arrays. Classical analyses that count on this information being available statically are of a little use here. The thesis has a practical goal: to define a static analysis framework that determines the dynamic information and makes it ready to use for classical user-defined analyses. For this to be possible, it is needed to resolve method calls, definitions of objects, include statements, and accesses to dynamic data structures. The solution proposed in the thesis is a combination of heap, value, and declaration analysis. The proposed heap analysis is specialised to track states of multidimensional arrays (and objects). It uses a value domain that tracks values of primitive types, and communicates with a declaration analysis, which keeps track of, e.g., which variant of a function is used. The analysis works with imprecise information such as "unknown values" or "unknown index of an array". The output of the analysis is some kind of a control a flow graph, called Intermediate Representation, with resolved dynamic function calls, includes of scripts, etc., together with a heap invariant. It allows the user analysis to access values stored in data structures. With all this information provided, standard user-defined analyses can used easily.

The problem is complex and has many levels of difficulty. The thesis solves them as a whole, leading to a consistent framework which is ready to use. It is innovative and, at the same time, still reasonably simple and comprehensible. It seems to be a good balance between precision and computational complexity, and I believe it is practical.

The framework has been implemented in the form of a prototype tool WeVerca. My impression is that the tool is quite mature and usable. It has been evaluated on a number of user-defined analyses such as taint analysis. A comparison with two state of the art tools gave very positive results. Most importantly, the precision of the analysis framework is high, leading to a great reduction of false positives and only a small number of missed errors. The weaknesses of the approach and current implementation are thoroughly analysed, which should be appreciated since this is not a common practice in our field. The discussion provides a lot of insight in the method. The discussed shortcomings are mainly moderate efficiency and occasional problems with imprecision of the analysis. Directions towards solving the identified problems are proposed, and I believe that they are feasible. Many of them can be probably solved by a larger amount of engineering effort.

One weakness of the thesis that I see is the style of writing. On one hand, I had problems understanding dense formal definitions of the analyses. I failed in some cases. I would need the informal explanation of pages such as page 50 even more detailed. On the other hand, in contrast to this dense formal style of presentation, there are other parts which would deserve a more formal approach (for instance, the presentation of functions joinToValue and joinToState on page 65). A minor shortcoming is occasional typos and inadequate formating, such as Section 5.6., which is 4 lines long.

The work was published in a sufficient way at several renowned international conferences and workshops. One journal publication is accepted. The tool WeVerca is also a valuable outcome.

Overall, I find the achieved results of a high quality and value. The author has proved to be able to independently conduct research in the given area. The thesis satisfies the common requirements expected to be met by Ph.D. theses in the area of computer science. Therefore, I recommend the thesis to be accepted for the defense and upon its successful completion, David Hauzar to be assigned the Ph.D. degree.

I would be interested in answers to the following questions:

1. There is a number formalisms that can be used as quite precise abstract domains for arrays and strings. Does it make sense to use them in your framework? What would be the properties of such an abstract domain which would be useful (what kind of information should it keep track of)?

2. Do you have any specific ideas on how to solve the efficiency problems of your tool by employing more data sharing?

3. Is it possible to use your method for verifying pre/post conditions and invariants? Could they be written in some logic? If yes, would it be possible to make use of the power of an SMT solver in your framework?

Brno, August 17, 2014

Mgr. Lukáš Holík, Ph.D.
Faculty of Information Technology
Brno University of Technology
Božetěchova 2, 612 66, Brno