Regression Verification: Proving the Equivalence of Similar Programs
(Invited Talk)

Ofer Strichman

Information Systems Engineering, IE, Technion, Haifa, Israel
ofers@ie.technion.ac.il

The ability to prove equivalence of successive, closely-related versions of a program can be useful for maintaining backward compatibility. This problem has the potential of being easier in practice than functional verification for at least two reasons: First, it circumvents the problem of specifying what the program should do; Second, in many cases it is computationally easier, because it offers various opportunities for abstraction and decomposition that are only relevant in this context.

I will begin the talk by defining six notions of input/output equivalence between programs, and then show Hoare-style proof rules that can be used for proving the equivalence of recursive functions according to these definitions. I will then show a decomposition algorithm that, given a mapping between the recursive functions in the two programs, attempts to reduce the equivalence verification problem into verification of many smaller verification problems corresponding to pairs of mapped functions. Callees of these functions that were already proven equivalent are abstracted with uninterpreted functions. I will conclude the talk by describing a regression verification tool for C programs – built by Benny Godlin – that, based on these rules and decomposition algorithm, was able to prove automatically the equivalence of some nontrivial programs.

The talk is based on [GS08, GS09].

References

[GS08] Godlin, B., Strichman, O.: Inference rules for proving the equivalence of recursive procedures. Acta Informatica 45(6), 403–439 (2008)

[GS09] Godlin, B., Strichman, O.: Regression verification. In: 46th Design Automation Conference (DAC) (2009) (to be published)

A. Bouajjani and O. Maler (Eds.): CAV 2009, LNCS 5643, p. 63, 2009.