

BLACK BOX CHECKING¹

DORON PELED

*University of Warwick
Coventry CV4 7AL, United Kingdom
e-mail: doron@dcs.warwick.ac.uk*

MOSHE Y. VARDI²

*Department of Computer Science, Rice University
Houston, TX 77005, USA
e-mail: vardi@cs.rice.edu*

and

MIHALIS YANNAKAKIS

*Computer Science Department, Stanford University
Stanford CA 94305-9025, USA
e-mail: mihalis@cs.stanford.edu*

ABSTRACT

Two main approaches are used for increasing the quality of systems: in *model checking*, one checks properties of a known design of a system; in *testing*, one usually checks whether a given implementation, whose internal structure is often unknown, conforms with an abstract design. We are interested in the combination of these techniques. Namely, we would like to be able to test whether an implementation with unknown structure satisfies some given properties. We propose and formalize this problem of *black box checking* and suggest several algorithms. Since the input to black box checking is not given initially, as is the case in the classical model of computation, but is learned through experiments, we propose a computational model based on games with incomplete information. We use this model to analyze the complexity of the problem. We also address the more practical question of finding an approach that can detect errors in the implementation before completing an exhaustive search.

Keywords: Formal methods, model checking, specification, testing, verification

¹Full version of a submission presented at the workshop *Logic and Algebra for Concurrency* held at Dresden University of Technology (Germany), September 13–16, 2000.

²Supported in part by NSF grants CCR-9628400 and CCR-9700061, and by a grant from the Intel Corporation. Part of this work was done when this author was a Varon Visiting Professor at the Weizmann Institute of Science