Speeding up the Heuristics for Generating Synchronizing Sequences

by

Kamer Kaya

Date and Time: April 26th, 2017 (Wednesday), 13:00
Place: Room 207, Computer Engineering Building, GTU

All interested are cordially invited.

ABSTRACT:

Model Based Testing (MBT) uses formal models of system requirements to generate effective test cases. Most MBT techniques use state-based models, where the behaviour of the model is described in terms of states and state transitions. Common to most FSM based testing methods is the need to bring the system under test (SUT) to a particular state. When there is a trusted reset input in the SUT, this is quite easy. However, sometimes such a reset input is not available, or even if it is available, it may be time consuming to apply the reset input. Therefore there are cases where the use of a reset input is not preferred. A synchronizing sequence for an FSM M is a sequence of inputs such that no matter at which state M currently is, if this sequence of inputs is applied, M is brought to a particular state. Therefore a synchronizing sequence is in fact a compound reset input, and can be used as such to simulate a reset input in the context of FSM based testing. For an FSM M with n states and alphabet size p, checking if M has a synchronizing sequence can be decided in time O(pn^2). Since a synchronizing sequence will possibly be used many times in a test sequence, computing a shortest one for an FSM is of interest, but this problem is known to be NP-hard. There exist a number of heuristics, called synchronizing heuristics, to compute short synchronizing sequences, such as Greedy and Cycle both with time complexity O(n^3 + pn^2), and SynchroP/SynchroPL with time complexity O(n^5 + pn^2).

There exist recent studies in the literature on finding short(er) synchronizing sequences. To evaluate the proposed heuristics with respect to their runtime performance and sequence lengths, the existing heuristics, such as Greedy and SynchroP, have been frequently used. These studies show that improvements on the synchronizing sequence length are possible with a reasonable tradeoff on runtime. In this talk, we will discuss that the existing heuristics are in fact much more competitive since with correct implementations, their performance can be improved by 10-100x. We also believe that the techniques we tailor for the existing heuristics can also be adapted for the recent ones yielding better, higher-quality and faster, heuristics to generate synchronizing sequences.

BIOGRAPHY:

Prof. Kaya is an Assistant Professor at the Faculty of Engineering and Natural Sciences at Sabancı University and Adjunct Research Assistant Professor at the Dept. of Biomedical Informatics of the Ohio State University. Before that, he received his PhD from Dept. of Computer Engineering at Bilkent University and worked at CERFACS, France, as a post-graduate researcher in the Parallel Algorithms Project for two years. Then he joined the Ohio State University in September 2011 as a postdoctoral researcher, and in December 2013, he became a Research Assistant Professor in the Dept. of Biomedical Informatics. His current research interests include Parallel Programming, High Performance Computing, and Cryptography.