Input Sequence Generation for Testing of Communicating Finite State Machines (CFSMs)

Karnig Derderian, Robert M. Hierons, Mark Harman, and Qiang Guo

Brunel University, Department of Information Systems and Computing
Uxbridge UB8 3PH, UK
{karnig.derderian,rob.hierons,mark.harman,qiang.guo}@brunel.ac.uk

1 Introduction

Finite State Machines (FSMs) have been used to model systems in different areas like sequential circuits, software development and communication protocols [1]. FSMs have been an effective method of modelling because a variety of techniques and automated tools exist that work with them.

Testing is an important part of the software engineering process and can account for up to 50% of the total cost of software development [2]. This motivates the study of testing FSMs to ensure the correct functioning of systems.

Some systems are more naturally modelled as a set of FSMs that operate concurrently or use messages to communicate with each other, rather than as a single FSM. FSMs that communicate by passing messages to each other are called Communicating Finite State Machines (CFSMs). CFSMs have input queues and communicate when a CFSM produces an output that is placed in the input queue of another CFSM. When a model M, consisting of CFSMs, receives an input this input triggers a sequence of local transitions within the individual CFSMs forming a global transition in M.

Under certain conditions a set of CFSMs $M_1, ..., M_n$ can be converted into an equivalent single FSM called the product machine [3]. However if n_i denotes the number of states in M_i, the product machine has $O(\prod_i n_i)$ states and hence suffers from a combinatorial explosion. This poster explores new alternatives to state and transition testing of CFSMs using Genetic Algorithm (GA) heuristics, without the use of a product machine.

Local transitions for CFSMs represent an internal transition of individual CFSMs. The testing effort can be reduced by testing all the local transitions instead of all the global transitions. Global transitions of the CFSM set involve at least one local transition in one of the CFSMs. Therefore discovering a fault in a local transition t that is used by a number of global transitions could reduce the testing effort. After passing a transition its final state has to be verified. In order to verify that the current local state of CFSM M_i is s it is sufficient to use an input sequence $u(s)$ such that it distinguishes global state σ, with $\sigma(i) = s$ from all other global states in which only the local state of M_i differs. A transition sequences with such property, given a number of conditions on the global state of M are met is known as a Constrained Identification Sequences (CIS) [4].

K. Deb et al. (Eds.): GECCO 2004, LNCS 3103, pp. 1429–1430, 2004.

Similarly transition sequences called Unique Input/Output (UIO) sequences are sometimes used in state verification of FSMs [5].

This poster extends a previously suggested idea regarding local transitions and their corresponding final states and suggests the use of genetic algorithms to automatically find a test set. This poster shows how the problem of finding a robust set of test data for a set of CFSMs may be represented as an optimisation problem, where GAs are proven to contribute [6].

2 Input Sequence Generation

The focus is on how to use a novel method of finding CISs using genetic algorithms. Before the fitness function can be used a list of all the local transitions are linearly scanned and all the local transitions are ranked according to how common the input output pair is within the rest of the transitions. The penalty of the input and output pair of a transition will be calculated. A unique input output pair will have no penalty, while the most repeated input output pair will have the maximum cost. This notion of ranking the input output pairs will be used in determining the fitness of a potential CIS. The fitness of an input sequence is calculated by adding all the ranks of the transitions involved.

GAs using similar fitness measure have been successful in generating UIOs for up to 62% more of the states for an FSM compared to a random search [7]. Since the problem of generating CISs for CFSMs can be related to the problem of generating UIOs for FSMs [4] GAs might provide a good solution to the problem. The issues of concurrency and the invisibility of local transition remains with CIS, but the fitness functions and verification algorithm attempt to address these complications. Work in [5] also showed how GAs can outperform random search, however using a different fitness measure.

Future work will consider the problem of sequencing a set of CISs in order to minimise the cost of reaching the transition to be tested. This could be also represented as an optimisation problem and a solution attempted using GAs.

References

1. Tanenbaum, A.: Computer Networks. Prentice Hall (1996) 3rd edition.
2. Beizer, B.: Software testing techniques. Technical report (1990) 2nd edition.
3. Petrenko, A., Yavtushenko, N., Dssouli, R.: Testing strategies for communicating fsms. Protocol Test Systems (November 1995) 195–208
4. Hierons, R.: Checking states and transitions of a set of communicating finite state machines. Microprocessors and Microsystems,Special Issue on Testing and testing techniques for real-time embedded software systems, **24** (2001) 443 452,
5. Guo, Q., Hierons, R., Harman, M., Derderian, K.: Computing unique input/output sequences using genetic algorithms. In: LNCS vol. 2931, Springer (2004) 169–184
6. Goldderg, D.E.: Genetic Algorithms in search, optimisation and machine learning. Addison-Wesley Publishing Company (1989)
7. Derderian, K., Hierons, R., Harman, M., Guo, Q.: Generating unique input output sequences for finite state machine testing using genetic algorithms. IEE Electronics Letters (Under consideration) (2004)