# Fault Detection of Reachability Testing with Game Theoretic Approach

S. Preetha
Research Scholar,
Karpagam University,
Coimbatore.

Dr.M. Punithavalli
Director,
Sri Ramakrishna Engineering College,
Coimbatore.

***Abstract*---This paper presents a game theoretic approach to fault detection of reachability testing. Physical systems are susceptible to fault at any time. Due to this reason the problem of determining and reacting to fault detection has received a greater attention in today's era. In this paper we study the fault detection of reachability tesing from a game theoretic point of view. Reachability testing is an approach that combines non-deterministic and deterministic testing. It is based on a technique, called prefix-based testing, which executes a test run deterministically up to a certain point and there after allows the test run to proceed non-deterministically.**

**We use the well known framework of reachability testing in two fold. First, reachability testing adopts a dynamic framework in which SYN-sequences are derived automatically and on-the-fly, without constructing any static models. Second, reachability testing is an interleaving-free approach in which independent events are never totally ordered in a SYN-sequence. Therefore, reachability testing automatically avoids the problem of exercising more than one interleaving of the same partial ordering of events. The methodology controls the evolution of the system and chooses whether and when a fault occurs and the detection of the same. Case study and preliminary experimental results indicate that proves to be a viable approach using game theoretic method.**

## I. INTRODUCTION

The classical mode of semaphore based reachability testing refer to reachability testing distributed across different nodes which addresses the communication efficiency between parallel task programs. Usually the states and transitions of the system under consideration are collected in a transition graph, and the only dynamic aspect is the change of state (via available transitions) which is then captured by the concept of an execution path or (if branching behavior is considered) of a computation tree. A theory of model-checking over dynamic structures is still very much in its beginnings. The objective of this paper is to introduce (and prove initial results on) a simple but quite general model for studying networks which change over time. We consider graphs as models of communication networks. Such a network is assumed to change over time by way of deletion and creation of nodes. The available resources travel through edges to existing nodes, and they may at some node a stores a deleted node b by moving to it if there was an edge (a, b) in the network before the deletion of b as in Fig. 1.


Fig 1. Movement of Nodes

The same way the creation of a node can also be performed by picking out some set of strong nodes by creating a new node S and connect it with each of all the nodes in S by transition as shown in fig 2

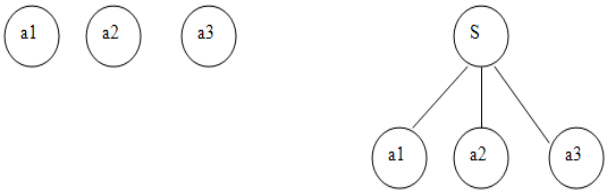


Fig.2. Creation of Nodes

A *fault* is a deviation of the system structure or the system parameters from the nominal situation. This implies that after the occurrence of a fault the system will have a behavior which is different from the nominal one. Hence Fault detection is the property of reacting to faults. In particular the analysis of fault tolerance consists in establishing if a given system is still able to achieve its tasks after the occurrence of a given fault, whereas the synthesis of fault tolerance resides in providing a given system the tools to react to a given faulty situation. Fault detection and fault tolerant systems have been studied by the control community since the late '70s, as in where fault detection for chemical processes is introduced. One of the first surveys on fault detection is, which is dated 1984, and where some methods based on modelling and estimation are introduced. Much later the interesting book collects some results on Fault Detection and Isolation (FDI) methods.

Reachability of states is one of the key problems in the area of automatic verification. Reachability Testing is implementation-based, it cannot by itself detect "missing sequences", i.e., sequences that are valid according to the specification but are not allowed by the implementation. In this respect, Reachability testing is complementary to specification-based testing, which selects valid sequences from a specification and determines whether the sequences are allowed by the implementation. Most safety properties of systems can be reduced to simple reachability properties; a typical example is the mutual exclusion property of mutual exclusion algorithms. While concurrent programs offer some advantages, they also exhibit nondeterministic behavior, making them notoriously difficult to test. One way to deal with nondeterministic behavior during testing is to execute the program with the same input many times and hope that faults will be exposed by at least one of the executions with the help of game theoretic approach. Our proposed work effectively detect the faults inserted in the programs by reachability testing using game theoretic approach.

## II. RELATED WORKS

The earliest example of a formal game-theoretic analysis is the study of a duopoly by Antoine Cournot in 1838. The mathematician Emile Borel suggested a formal theory of games in 1921, which was furthered by the mathematician John von Neumann in 1928in a "theory of parlor games." We choose to use game theory applied to fault diagnosis of hybrid systems because it allows us not to split the continuous and the discrete behaviours. A hybrid game is a multiplayer structure where the players have both discrete and continuous moves and the game proceeds in a sequence of rounds. In every round each player chooses either a discrete or a continuous move among the available ones [5]. Hybrid games has been successfully applied to solve the controller synthesis problem for timed [1] and hybrid automata [2, 4], and to the fault diagnosis problem for timed automata [3]. In our setting we model the fault diagnosis problem as a game between two players, the environment and the diagnoser.

Reachability testing is an approach that combines non-deterministic and deterministic testing [9] [10] [11]. It is based on a technique called prefix-based testing, which controls a test run up to a certain point, and then lets the run continue non-deterministically. The controlled portion of the execution is used to force the execution of a "prefix SYN-sequence", which is the beginning part of one or more feasible SYN-sequences of the program. The non-deterministic portion of the execution exercises one of these feasible sequences.

The environment controls the evolution of the system and chooses whether and when a fault occurs. The diagnoser observes the external behaviour of the system and announces whether a fault has occurred or not. Existence of a winning strategy for the diagnoser implies that faults can be identified correctly, while computing such a winning strategy corresponds to implement a diagnoser for the system. In contrast with the usual definition of hybrid game, our game is asymmetric, since the environment is more powerful than the diagnoser, and is under partial observability, since the diagnoser is blind to the value of internal variables and to the occurrence of internal events. We define two notions of diagnosability, and we prove that the fault diagnosis problem is solvable for the weakest notion of diagnosability for all classes of hybrid automata that admit a bisimulation with finite quotient that can be effectively computed.

Consistency-based diagnosis [6, 8] considers the faulty behavior as a contradiction between the actual and the nominal behavior of the system. It does not require the possible faults to be known, and it proceeds by dropping

the assumptions on the behavior of each component in turn. If this removes the contradiction, the component is considered a candidate for correction. More recently, applicability of discrete game theory to fault localization and automatic repair of programs have been proposed in [7].

## III. METHODOLOGY

Game Theory provides us with the needed necessary mathematical techniques for analyzing and deciding upon strategic situations. Such a situation comprises of two or more players, all of them having with their own strategies and motivations. By playing a particular strategy a player receives a reward, depending on the strategy chosen by other players.

### *A. Reachability Testing*

The reachability testing is solution concept of game theory specifying optimal strategic choices for all players by reason that none of the players has any motivation to diverge from the reachability testing because one player can not gain greater payoffs by choosing another strategy when all the other players choose the strategies given by the profile. To calculate reachability testing we require N set of players, Si as a finite strategy set and Pi as a payoff function. A reachability testing process is illustrated in fig 2 by way of four threads T1, T2,T3 and T4.

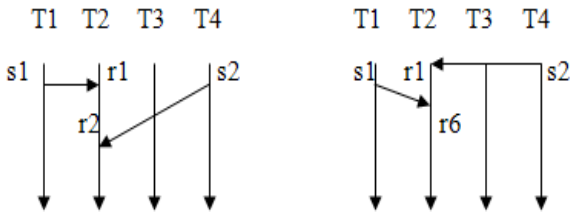


Fig 3. Reachability Testing Scenario

Each variant is used to conduct a prefix-based test run, which forces the events and synchronizations in the variant to be replayed and then allows the test run to proceed non deterministically. As no new variants can be derived so the reachability testing process stops. We present a new reachability testing algorithm for driving the testing process. In order to avoid exercising the same SYN-sequence more than once, all existing reachability testing algorithms need to save the history of SYN-sequences that have been exercised. Our new algorithm using game theoretic approach saves no SYN-sequences, but still guarantees that every partially ordered SYN-sequence is exercised exactly once. This significantly reduces the space and time requirements of reachability testing. Our game theoretic approach is written in Java and makes no modification to the Java virtual machine or to the underlying operating system.

### *B. Reachability Testing with Game Theoretic Approach*

The fig 4. depicts the architecture of the game theoretic approach which results in the minimized fault detection. The key idea of our test method is to adopt a winning strategy. A reachability testing problem is that given a set S and a goal state K , we should find a winning strategy f. A strategy f is a function that during the course of the timed game constantly gives information as to what the player should do in order to win the game. At a given state of the run, the player can be guided either to do a particular controllable action (i.e., to offer an input to the plant), or to do nothing at this point in time and just wait

.

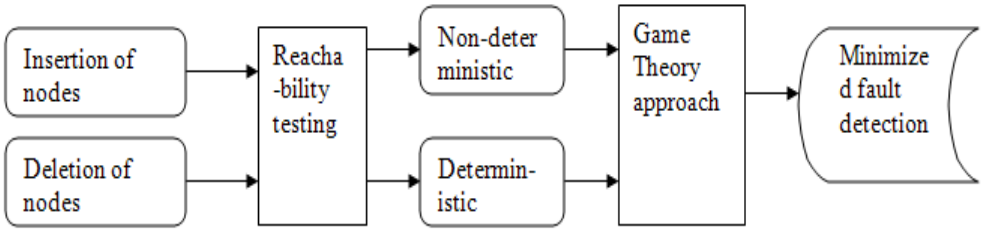


Fig 4. Architecture of Game Theoretic Approach

***Mathematical Model***

The fundamental principles of game theory is that

1) Each node makes the best possible move.
2) Each node knows that his or her opponent is also making the best possible move

In old program new features are added which may create side effects after a long period of time. The reachability testing avoids the side effects with the help of game theory approach. The old program tries to restrain the new program from getting enter into it. But the new features maximizes to attain an equilibrium state and tries to reduce the error rate using the following formulation:

The strategic form of game is defined by three objects:

1) the set N = {1,2,….n} players
2) the sequence S1,….Sn of strategy sets
3) the sequence p1(s1,…sn), p2(s2..sn)….payoff functions of players

A game is zero sum if

$$\sum pi\ (s1,s2,\ldots sn) = 0 \quad (1)$$

where i = 1,2,..n and s1 € S1 ,s2 € S2 ,sn € Sn.

For each player i and alternate strategy S we have that

$$pi(s1,s2\ldots sn) >= pi'(s1',s2'\ldots.sn') \quad (2)$$

***To attain Equilibrium (Nash)***

Consider a strategy R for old program. Let v be the expected value. The expected payoffs for different strategies of new feature add on using game theory will be pR. The best strategy can be obtained by

$$R >= 0 \quad (3)$$

$$\sum Ri = 1 \quad (4)$$

$$(pR)j >= v \text{ for all } j \quad (5)$$

## IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

Simulations performed on this work shows us that by adding on new features to the old program there is a possibility of occurrence of side effects. By using the reachability testing with game theory the side effects can be avoided. Tests have been conducted in java to utilize its multi threading capabilities and shows us that our proposed model outperforms the existing one in terms of execution time and space. Our classical reachability testing model took 58 minutes to perform the job whereas our proposed reachability testing model with game theory approach completed the same job in 15 minutes.

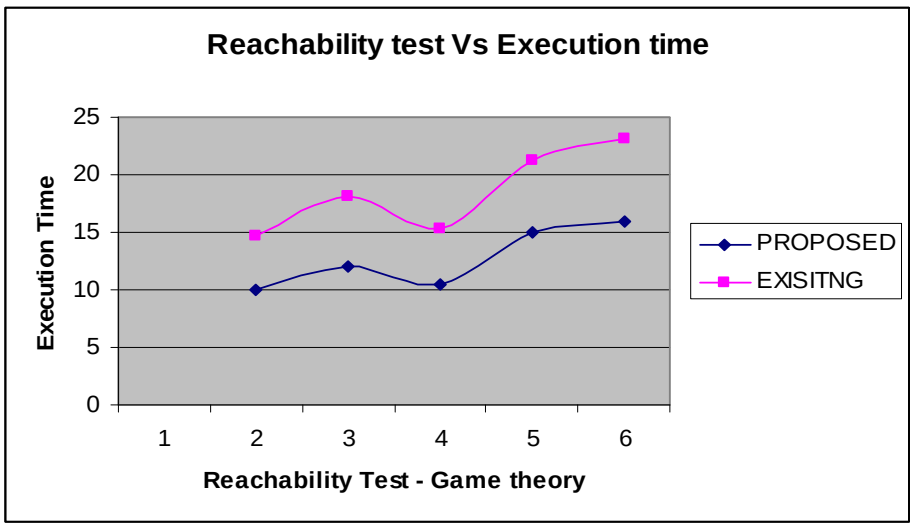


Fig 5 Reachability Test Vs Execution Time

The fig 5 shows the performance metric of reachability testing using game theory. When compared to the existing model, our proposed work executes the job efficiently with minimum amount of time taken to complete the job.

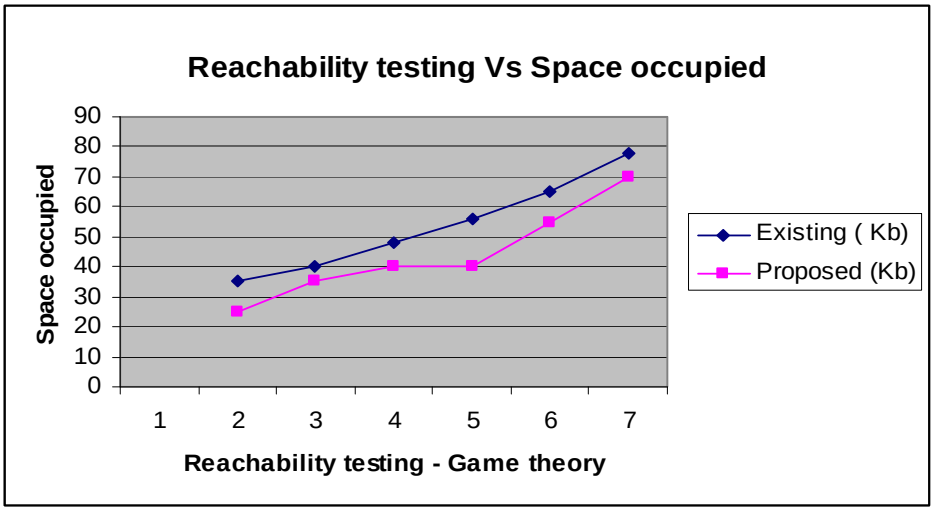


Fig 6 Reachability Testing Vs Space Occupied

Fig 6 depicts the reachability testing Vs space occupied in terms of Kb. From the graph it is clear that our proposed model using game theory occupies less space when compared to the existing model that results in the lower error rate.

## V. CONCLUSION

The Fault detection of the reachability testing is obtained by inserting the faults into the programs and program is tested using a game theoretic technique. In this paper we studied the fault-detection of reachability testing from a game-theoretical point of view. The proposed work is analysed and the performance is measured in terms of execution time and space. We used the formalism of reachability testing for fault detection. The inclusion of a winning strategy implies that the fault detection can be identified correctly using game theory. Finally, we have shown how to determine the existence of winning strategy using execution time and space as the parameters.

## REFERENCES

[1] E. Asarin, O.Maler, A. Pnueli& J. Sifakis (1998): Controller Synthesis For Timed Automata. In: Proceedings of the IFAC Symposium on System Structure and Control, Elsevier Science Publishers, pp. 469–474.

[2] P. Bouyer, T. Brihaye & F. Chevalier (2010): O-Minimal Hybrid Reachability Games. Logical Methods in Computer Science 6(1:1), pp. 1–48.

[3] P. Bouyer, F. Chevalier & D. D'Souza (2005): Fault Diagnosis Using Timed Automata. In: Foundations of Software Science and Computational Structures, Lecture Notes in Computer Science 3441, Springer Berlin / Heidelberg, pp. 219–233, doi:10.1007/978-3-540-31982-5 14.

[4] T. A. Henzinger, B. Horowitz & R. Majumdar (1999): Rectangular Hybrid Games. In: Proceedings of the 10th International Conference on Concurrency Theory, LNCS 1664, Springer-Verlag, pp. 320–335.

[5] C.J. Tomlin, J. Lygeros & S.S. Sastry (2000): A game theoretic approach to controller design for hybrid systems. Proceedings of the IEEE 88(7), pp. 949–970, doi:10.1109/5.871303.

[6] G. Fey, S. Staber, R. Bloem & R. Drechsler (2008): Automatic Fault Localization for Property Checking. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 27(6), pp. 1138 –1149.

[7] B. Jobstmann, S. Staber, A. Griesmayer & R. Bloem (2011): Finding and Fixing Faults. Journal of Computer and System Sciences (JCSS)

[8] R. Reiter (1987): A theory of diagnosis from first principles. Artificial Intelligence 32(1), pp. 57–95, doi:10.1016/0004-3702(87)90062-2.

[9] G. H. Hwang, K. C. Tai, and T. L. Huang. Reachability testing: An approach to testing concurrent software. International Journal of Software Engineering and Knowledge Engineering, 5(4):493-510, 1995.

[10] Yu Lei and Kuo-Chung Tai, Efficient reachability testing of asynchronous messagepassing programs, Proc. 8th IEEE Int'l Conf. on Engineering for Complex Computer Systems, pp. 35-44, Dec. 2002.

[11] K. C. Tai. Reachability testing of asynchronous message-passing programs. Proc. of the 2nd International Workshop on Software Engineering for Parallel and Distributed Systems, pp. 50-61, 1997.