ISSN 2229-5518

An Efficient Approach for Equivalent Mutants Detection using Fuzzy Logic

Megha Jhamb, Abhishek Singhal

Abstract— Mutation testing can consist of a large number of Equivalent Mutants that degrade the capabilities of mutation testing and require more resources, time and efforts. Class level and operator level mutation operators are applied to generate a large number of mutants and are the most critical factor as the time required for mutation testing is totally dependent on these two factors. Constraints based mutation testing approach utilizes these two factors to improve the efficiency of mutation testing by killing most of the equivalent mutants. In this paper Fuzzy Logic is used to improve the quality of the constraints based equivalent mutants detection technique. This paper takes into consideration three metrics reachability, necessity, and sufficiency to increase the efficiency of constraints based mutation testing. The analysis of the study shows that fuzzy based equivalent mutation detection technique improves the efficiency of mutation testing and reduces the time required to perform it.

Index Terms — Mutation Testing, Mutants, Equivalent Mutant, Reachability, Necessity, Sufficiency, Fuzzy Logic

---

1 Introduction

Mutation testing [1] is the process of introducing faults in a program by changing some part of program's code and then comparing the new results with the original results. The changed program is then termed as a mutant and changes to the program are made using mutation operators [2]. But if the output of mutant program and original program are same then we call this mutant as equivalent mutant [3, 4]. In this paper we will study how equivalent mutants can be found using fuzzy logic.

Table 1
An Example of Mutant and Equivalent Mutant Generation

Original program	Mutant I	Mutant II

Int x,y,a=10;
While (x>y)
{
a++;
If(a==20)
{
break;
}
}
	int x,y,a=10;
	while(x<y)
	{
	a++;
	if(a==20)
	{
	break;
	}
	}
		int x,y,a=10;
		while(x<y)
		{
		a++;
		if(a>=20)
		{
		break;
		}
		}

In the above example of mutation testing a slight changes are made in the code of original program a. In Mutant I, > symbol is replaced with < symbol and so the outputs will also differ but in Mutant II , == arithmetic operator is replaced with >= operator and still here the output will not differ so Mutant II will be termed as an equivalent mutant.

In the proposed technique, we have considered three constraints for equivalent mutant detection; three attributes reachability, necessity, and sufficiency are taken into account to predict the equivalent mutants. The following section gives the details of these attributes:

Reachability: The test case must execute the mutated statement. i.e. if t cannot reach S, then t will never kill M [5].

Necessity: To kill a mutant, the test case must cause the mutant to have an incorrect state if it reaches the mutated statement. i.e. for t to kill M, it is necessary that if S is reached, the state of M immediately following some execution of S must be different from the state of P at the same point [5].

Sufficiency: The test case must cause the final state of the mutant to be different from the original program. i.e. the final state of M differs from that of P [5].

The rest of the paper is organized as follows: Section 2 and section 3 contains the concept and proposed Fuzzy based approach for equivalent mutants' predication. Section 4 discusses the comparative analysis of the proposed fuzzy logic based technique for equivalent mutant detection. Section 5 provides the conclusion & future prospects of this technique.

2 Proposed Fuzzy based approach and model for Equivalent Mutants Detection.

Zadeh [6] derived the concept of Fuzzy Logic (FL) to implement vagueness in linguistic variables. The concept of FL implements and simulate the human knowledge as nature implements it in daily life. The important feature of the proposed methodology is the Fuzzy Inference System (FIS). It is a combination of fuzzification system that takes the crisp values an input and transforms them into the fuzzy sets by the use of fuzzification function, fuzzy inference engine it is responsible for matching and validating the inputs with the fuzzy rule base, and fuzzy rule base is a combination of IF-THEN rules and defuzzification system that transforms the fuzzy output into the crisp values. To interpret the if-then rule involves fuzzification of the input and applying the suitable fuzzy operators [7]. The architecture of the model used in the proposed approach is shown in figure 1.

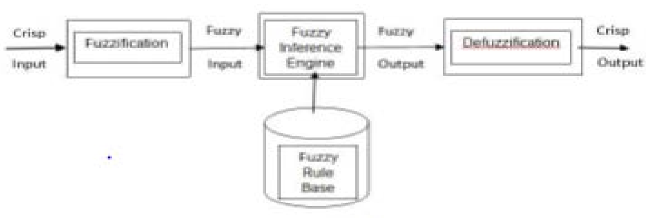

Fig 1: Architecture of FIS

3 FUZZY BASED EQUIVALENT MUTANTS PREDICATION MODEL FOR MUTATION TESTING

In the proposed approach, a fuzzy based system to predict the equivalent mutants for the mutation testing in a software testing environment is given. To detect & predicate the equivalent mutant in mutation testing, sum of all the constraints like reachability, necessity, sufficiency have been taken for the entire mutants program. Mathematically the relation between the equivalent mutants and their probability to be chosen as equivalent to the other mutants can be given as follow:
Let R_n, N_n, S_n be the values of the node energy, stability and buffer occupancy respectively then

$$Q_f = F_m [R_n, N_n, S_n]$$
$$T(n) = P/1\text{-}P^*(r \bmod 1/P), \text{ if } n \in N \;\&\; Q_f > 0 \quad (1)$$
$$T(n) = 0, \text{ otherwise}$$

In the proposed approach there are three crisp inputs in the form of reachability, necessity and sufficiency. These crisp inputs are passed to the inference engine after the fuzzification process. Inferences engine consists of a rule base that are used to predict the mutants that can be selected as equivalent. In the proposed approach Mamdani's method is used for the inference process. The crisp input values are provided to the system and antecedents are evaluating using the fuzzy rule base and a conclusion is obtained that is used to predict the equivalent mutants. The system architecture of the used model is shown in figure 2.

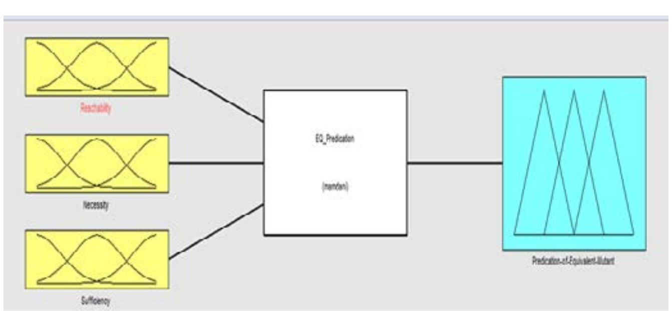

Fig 2: System Architecture of Equivalent Mutants detection based on FIS approach

3.1 Fuzzification Process

In this approach reachability, necessity, sufficiency these three constraints have been considered to predict the mutants that can be considered as equivalent mutants. These three attributes are used to form a fuzzy set and then the membership function for the each attribute is determined. Three membership functions have been used to determine the degree of membership of the input functions. In the output function 6 membership functions have been used to predict the equivalent mutants.

Table 2
Input Functions

Input	Membership		
Degree of Membership	(0)	(1)	(2)
Reachability	Low	Medium	High
Necessity	Low	Medium	High
Sufficiency	Low	Medium	High

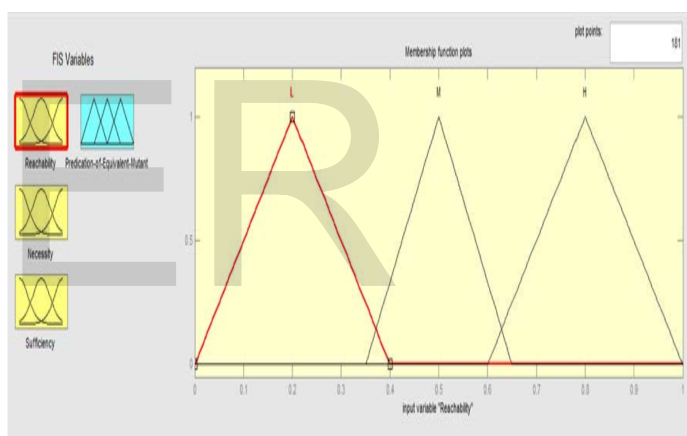

Fig 3(a): Fuzzification function for Reachability

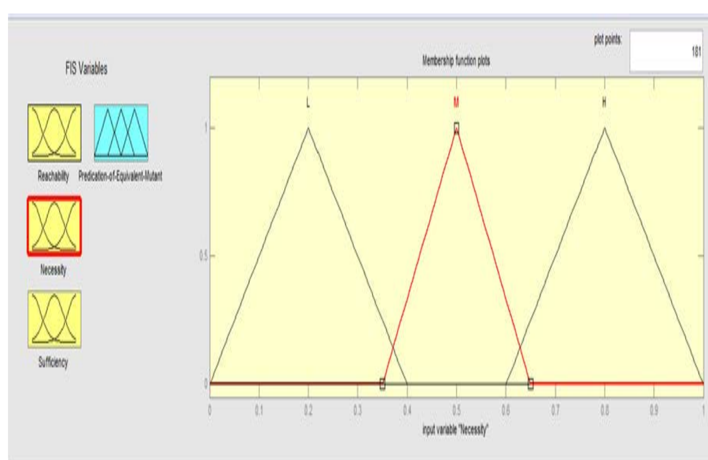

Fig 3(b): Fuzzification function for Necessity

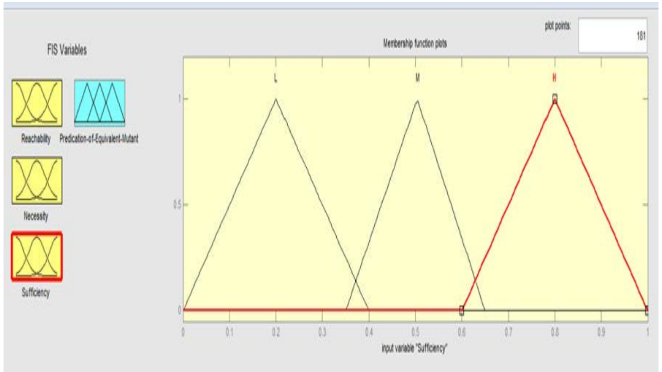

Fig 3(c): Fuzzification function for Sufficiency

Table 3
Output Functions

Output	Membership Function					
Degree of Membership	(1)	(2)	(3)	(4)	(5)	(6)
Probability of a mutant to be an Equivalent Mutant	Very Low	Low	Medium	Lower High	High	Very-High

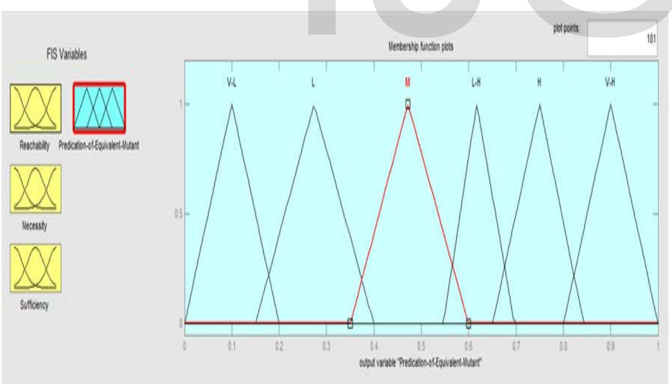

Fig 3(d): Fuzzification function for Probability of Mutant to be an Equivalent Mutant

3.2 Fuzzy Rule Base

In the proposed system a rule base is created for the fuzzy inference engine. There are 27 rules that are used by the inference engine. Some of the proposed rules are shown as:

- If the reachability of a mutant is High, necessity is high and sufficiency is high then the probability of the mutant to be selected as equivalent will be very high.
- If the reachability is mutant medium, necessity is medium and sufficiency is medium then the probability of the mutant to be selected as equivalent will be medium.
- If the reachability of a mutant is low, necessity is low and sufficiency is low then the probability of the mutant to be selected as equivalent will very low.

3.3 Defuzzification Process

The proposed system uses a defuzzification process to obtain the crisp values from the conclusions. These conclusions are obtained by fuzzy inference engine using the fuzzy rule base. Here centroid method is used for the defuzzification process. After the defuzzification process the below formula is used to predict the equivalent mutants:-

$$G(i)=\sum_{i=1}^{n} x_i * u(x_i) / \sum_{i=1}^{n} u(x_i) \qquad (2)$$

4. Comparative Analysis

The proposed approach is verified using the Matlab 7.12. The result clearly shows that the reachability, necessity and sufficiency play an important role to make the mutation testing a quality one. If these constraint metrics are considered for the selection and prediction of equivalent mutants, the quality and time required for mutation testing will be improved.

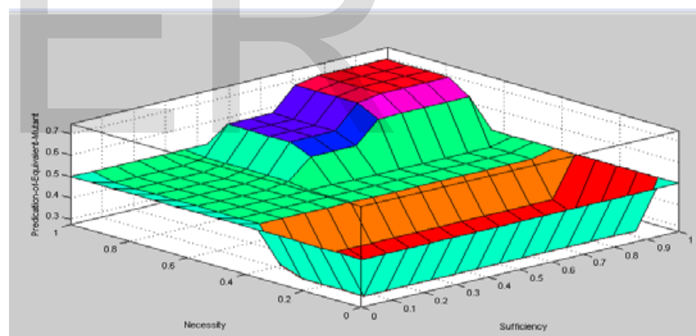

Fig 4(a): Surface View of necessity and sufficiency with mutant probability to be an Equivalent Mutant

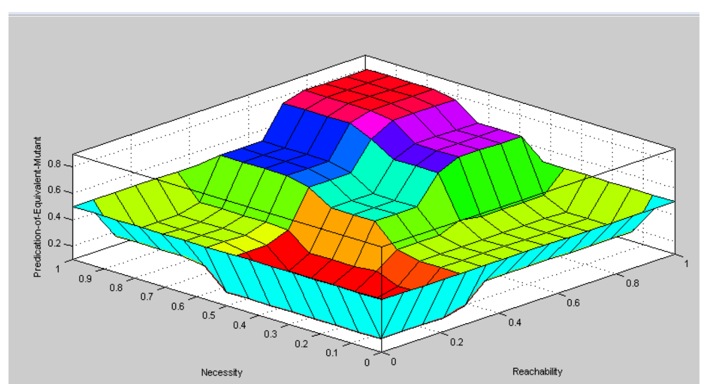

Fig 4(b): Surface View of Reachability and necessity with

mutant probability to be an Equivalent Mutant

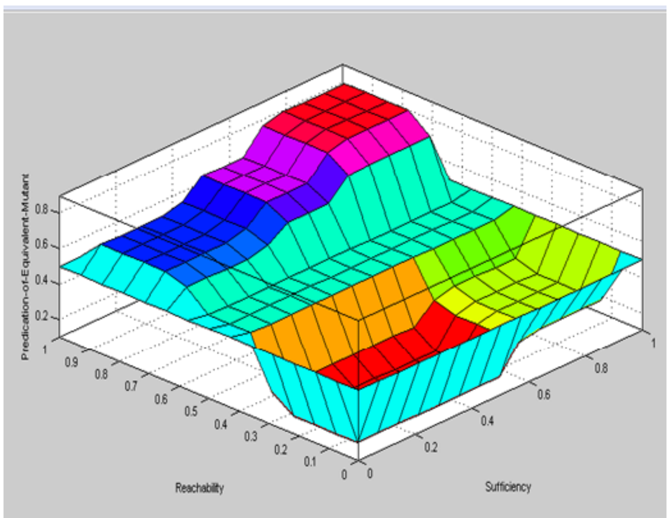

Fig 4(c): Surface View of Reachability and sufficiency with mutant probability to be an Equivalent Mutant

It is shown that a mutant having higher reachability and necessity condition has the highest probability to be elected as an equivalent mutant. As the reachability of the mutant degrades its probability to be selected as equivalent also decreased. Mutant reachability with the necessity is the most crucial factors for the equivalent mutants' selection and to the Mutation testing a quality one.

5. Conclusion

The fuzzy logic based equivalent mutation detection is proposed to make the constraints based mutation testing a quality one to produce the quality software in the fixed time frames. The efficiency of mutation testing is increased by selecting the quality mutants and killing the equivalent ones. The proposed approach helps in increasing the throughput & reducing the time frame for mutation testing. In future different set of metrics can be used to improve the efficiency of the proposed fuzzy logic based approach for equivalent mutant detection.

References

[1] Y. Jia & Mark Harman: An Analysis and Survey of the Development of Mutation Testing. IEEE, 2010.

[2] Offutt, A. J., Lee, A., Rothermel, G., Untch, R. & Zapf, C. "An Experimental Determination of Sufficient Mutation Operators". ACM Transactions on Software Engineering, (Baltimore, MD), IEEE Computer Society Press, pp. 100 –107. May 1993.

[3] K. Adamopoulos, M. Harman, and R. Hierons, "How to overcome the equivalent mutant problem and achieve tailored selective mutation using co-evolution," in Proc. Genetic and Evol. Comp. Conf. (GECCO). ACM, 2004, pp. 1338–1349.

[4] D. Schuler and A. Zeller, "(un-)covering equivalent mutants," in Proc. Intl. Conf. on Softw. Testing, Verification, and Validation (ICST). IEEE Computer Society, 2010, pp. 45–54.

[5] Offutt, J. & Pan, J. "Automatically Detecting Equivalent Mutants and Infeasible Paths". The Journal of Software Testing, Verification, and Reliability, Vol 7, No. 3, pages 165--192, September 1997

[6] Zadeh, L. A.: 1965. Fuzzy Sets, Journal of Information and Control, Vol. 8, pp: 338–353.

[7] Abraham, A.: Rule Based Expert Systems, Handbook for Measurement Systems Design, Peter Sydenham and Richard Thorn (Eds.), John Wiley and Sons Ltd., London, pp. 909-919, 2005.