A Method to Select Software Test Cases in Consideration of Past Input Sequence

Hee Eun Kim[a], Bo Gyung Kim[a], Hyun Gook Kang[a,]*
[a] Department of Nuclear and Quantum Engineering, Korea Advanced Institute of Science and Technology, 373-1 Guseong-dong,Yuseong-gu, Daejeon 305-701, South Korea
*Corresponding author: hyungook@kaist.ac.kr

1. Introduction

Nowadays, most of instrumentation and control (I&C) systems in the nuclear power plant (NPP) are digitalized. In the Korea Nuclear I&C Systems (KNICS) project, the software for the fully-digitalized reactor protection system (RPS) was developed under a strict procedure [1]. Even though the behavior of the software is deterministic, the randomness of input sequence produces probabilistic behavior of software. To include failures of the software into the reliability models of digital I&C systems, the contribution of software failure to the risk need to be assessed. A software failure occurs when some inputs to the software occur and interact with the internal state of the digital system to trigger a fault that was introduced into the software during the software lifecycle [2]. In this paper, the method to select test set for software failure probability estimation is suggested. This test set reflects past input sequence of software, which covers all possible cases.

2. Development of test sets regarding state of software

In this section, a method for selecting variables of test set and determining their profile is described. The test set reflects the previous input sequence of software as a form of state variables.

2.1 Test sets reflecting the state of software

State and output of running software is determined by the input and current state, and the state represents history of the input sequence. Therefore the state of software need to be considered to include past input sequence in the test of software. Only the variables stored in the memory could affect the program, so those variables are defined as state variables.

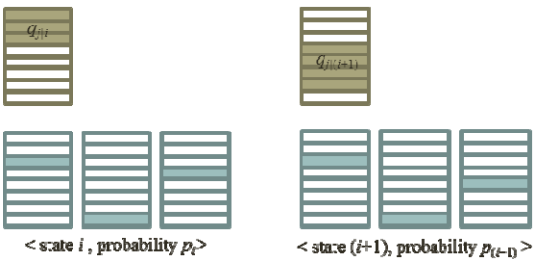

Fig. 1. Relationship of state and the range of input [3].

The test set is composed of state variables which represent past input sequence and current input variables. Among those test sets, the combinations which appear in the real world need to be tested to assess failure probability. If a state is determined, the possible range of inputs are limited (Fig. 1).

2.2 Determining profile of variables

To obtain the realistic test sets, the profile of each variables need to be determined. Some variables are related each other. For example, if a state variable is calculated using other variables, the range of one variable limits that of other variables. The profile of paired variable needs to be obtained along with the other paired variables. On the other hand, there are independent variables which are not related to any other variables. The profile of independent variables can be obtained individually. The combination of the paired variables and independent variables constitute test sets, then the probability of a test set can be obtained using the profile of variables.

There are two issues to be considered when obtaining the profile of the variables. One is dependency introduced by an input from human operator. Input signal from human operator is simple, but the action of human operator has latency and it also affects other variables. An input from human operator is so simple that the actions of the operator can be characterized with two criteria: timing of input from human operator, and repetition and delay of input from human operator. By considering those criteria, the variables related to the human action can be classified into paired and independent variables. The timing of the signal might be described as distributions.

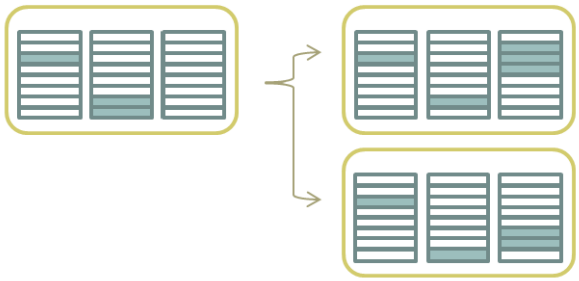

Fig. 2. Determining profile of variables considering dependency among the variables.

The other issue is related to the dependency of variables. As stated above, the range of one variable limits that of other variables (Fig. 2), so the order of variables need to be determined. The state and corresponding range of each variable can be obtained by reflecting plant dynamics, digital system's characteristics and the relationships among the variables.

3. Case study

Among the 19 trip signals of the KNICS RPS logic, "PZR_PR_Lo Trip" (pressurizer pressure low trip) is chosen as the target logic, which has variable TSP and operator bypass function [4]. The range of values can be obtained in the same way as previous study [5], in which plant dynamics, digital system's characteristics, ADC resolution, and scan timing are considered.

3.1 Variables of a test set

By inspecting source code of PZR-PR-Lo-Trip logic, we can find 3 state variables, TSP, Previous-pressure and Reset-delay-time; and 5 input variables, Current-pressure, Bypass-from-MCR, Bypass-from-RSR, Reset-from-MCR, and Reset-from-RSR. Among those variables, Current-pressure, TSP and Previous-pressure are paired variables.

Current pressure and previous pressure are paired because of plant dynamics. TSP is calculated by using the value of current pressure, therefore current pressure, previous pressure and TSP are paired variable. Reset-delay-time is related to the input from human operator, but the target logic deals with a random event (accident), input from human operator is independent variable. Other variables are related to the random input from human operator and they are independent of other variables.

3.2 Profile of variables

As stated above, the state should be determined first. For a full power operation, the TSP those not changes and the operator also those not pushes the reset or bypass button, so the state is represented as previous pressure only. In the case of start-up procedure, the TSP changes, so the state is represented as TSP and corresponding previous pressure. The TSP is reset when the plant is in shut-down process, the profile of independent variable, Reset-delay-time, also should be considered.

Among the paired variables, current pressure is input variable so the range of current pressure is determined lastly, when the other state variables are obtained. The TSP is changed according to the pressure, so the profile of TSP need to be determined first.

The possible states and input sets for full power operation is shown as an example (Table I). The probability of a set is obtained by multiplying each probabilities of paired variables and other variables. As the previous study [5] suggested, the tested portion is identified as error-free portion, and from that result, the software failure probability can be derived. The error-free portion is obtained based on the plant operation mode and their fraction of accident frequency.

Table I: Possible states and input sets for full power operation

State	Input	Error-free Portion	SW failure prob.
State 1	Input 1	9.666.E-01	3.335.E-02
	Input 2	6.009.E-05	3.329.E-02
	Input 3		
	Input 4	3.245.E-06	3.329.E-02
	Input 5		
State 2	Input 1	6.009.E-05	3.323.E-02
	Input 2		
	Input 3	3.245.E-06	3.323.E-02
	Input 4		
State 3	Input 1	6.009.E-05	3.317.E-02
	Input 2	3.245.E-06	3.316.E-02
	Input 3		
State 4	Input 1	3.245.E-06	3.316.E-02
	Input 2		
State 5	Input 1	3.245.E-06	3.316.E-02

4. Summary

In this study, the method to select test cases for software failure probability quantification was suggested. To obtain profile of paired state variables, relationships of the variables need to be considered. The effect of input from human operator also have to be considered. As an example, test set of PZR-PR-Lo-Trip logic was examined. This method provides framework for selecting test cases of safety-critical software.

REFERENCES

[1] J. H. Park, D. Y. Lee, C. H. Kim, Development of KNICS RPS Prototype, Proceedings of ISOFIC Probabilistic Safety Applications 237 2005, Session 6, pp.160-161, Tongyeong, Korea

[2] BNL-94047-2010, Review of Quantitative Software Reliability Methods, 2010.

[3] H. E. Kim, et al., A Profile-based Method to Select Test Cases for Safety-critical Software, Transactions of the Korean Nuclear Society Autumn Meeting, 2013.

[4] G. Y. Park, et al., Fault Tree Analysis of Knics Rps Software. Nucl Eng Technol;40:397–408, 2008.

[5] H.G Kang, et al., Input-profile-based Software Failure Probability Quantification for Safety Signal Generation Systems, Reliability Engineering and System Safety, Vol. 94, pp1542-1546