1. Introduction

Software safety testing focuses on whether safety-critical software under test fulfills corresponding safety specifications. But unfortunately, it is very difficult to test software safety by using traditional operational reliability demonstration testing methods. The main reason is that different failures may have different impacts on safety.

Damage size of a failure is a classified measurement of harm severity of the failure under the most adverse circumstances. Harm severity of a failure could be obtained through harm analysis. A classified software safety weight is the probability that failures in a subset of the corresponding damage size do not emerge. Software safety vector is a vector of classified software safety weights.

2. Methodology of Software Safety Demonstration Stress Testing

In the methodology of software safety demonstration stress testing, probabilities of test inputs in the test profile are adjusted according to damage sizes of their possible failures. Namely, for those test inputs which may cause software produce failures of large damage sizes, we could increase their probabilities in the test profile to realize stress testing, and for those test inputs which may cause software produce failures of small damage sizes, we could decrease their probabilities in the test profile. Let’s suppose $D$ be the input domain of software under test, its operational profile is:

$\{ <p_i, i >; i \in D \}, \sum_{i \in D} p_i = 1$

and its test profile is:

$\{ <d_i, i >; i \in D \}, \sum_{i \in D} d_i = 1$

Among them, $d_i = z_i \cdot p_i$, and $z_i$ is an adjustment factor used to increase or decrease probabilities of test inputs according to damage sizes of their possible failures. For those test inputs of same damage size, a same adjustment factor is taken. In every input subdomain $C_j$, $n_j$ test inputs will be randomly selected. Then, in the test profile,

$n = \sum n_j$ and $n_j = \sum_{i \in C_j} p_i \cdot Z_j \cdot n$

If in the operational profile, $n_j$ test inputs of the input subdomain $C_j$ are also randomly selected, the total $n'$ test inputs will be needed. That is

$n_j = \sum_{i \in C_j} p_i \cdot n'$

For every input subdomain $C_j$, the actually required total number of test inputs $n'$ in the operational profile is

$n' = Z_j \cdot n$

Therefore, for those test inputs of larger damage sizes, we could magnify test samples by making $Z_j >> 1$, and for those test inputs of smaller damage sizes, we could reduce test samples by making $0 \leq Z_j < 1$. In this way, we can change the number of test inputs in the test profile according to their damage sizes, so that inputs of larger damage sizes could be more adequately tested and the safety testing efficiency could be much improved.