

ВИКОРИСТАННЯ АЛГОРИТМУ ЦУКАМОТО ДЛЯ НЕЧІТКОЇ КЛАСИФІКАЦІЇ ПОМИЛОК У ТЕСТУВАННІ ПРОГРАМ

К.т.н. Ю. Ю. Козіна, к.т.н. О. В. Ткаченко, А.О. Козін

Одеський національний політехнічний університет;

НУ “Одеська морська академія”

Україна, Одеса

yuliyakc@mail.ru

У процесі життєвого циклу розробки програм роль тестування дуже значна. Пошук помилок у програмах та їх класифікація є найважливішою задачею. Невірно класифіковані помилки негативно впливають на подальший цикл її розробки. У роботі запропоновано виконувати класифікацію помилок за допомогою математичного апарату нечіткої логіки, а саме, на основі алгоритму Цукамото. Це дозволило підвищити вибірковість класифікації.

Ключові слова: тестування, нечітка класифікація, програми.

Вступ: В даний час в області розробки програм відбувся перехід від безпосереднього тестування до забезпечення якості, що охоплює цикл розробки в цілому. Вірна класифікація помилок за ступенем їх критичності [1], істотно впливає на подальший їхній життєвий цикл, а також на ефективність роботи системи в цілому. Рішення даної задачі вимагає знань, логічних міркувань, досвіду інженера з контролю якості, тобто знаходиться в площині інтелектуальної роботи людини. Кількість помилок може бути настільки велика, що не завжди буває можливим вірно виконати класифікацію помилок за ступенем їх критичності. Тому завдання розробки алгоритму класифікації помилок у тестуванні програм для підвищення достовірності класифікації є актуальною. Для її вирішення пропонується використовувати математичний апарат нечіткої логіки, як одного з напрямків досліджень в галузі штучного інтелекту, що дозволяє знаходити рішення трудноформалізованих завдань, а саме задач класифікації.

Основна частина: Класифікація помилки у процесі тестування програм передбачає надання їй ступеня критичності. Виходячи з цих отриманих даних, помилка отримує пріоритет для подальшої роботи над її виправленням. Ступені критичності та пріоритет помилки можуть по різному позначатися, в залежності від баг-трекінгової системи, в якій працює виконавець [2]. Наприклад, в баг-трекінгової системі Bugzilla пріоритет помилки позначається від P1 (найбільш важливі) до P2(найменш важливі), також використовуються такі ступені критичності:

- надзвичайна. Блокує подальшу роботу з розробки та/або тестуванню;
- критична. Аварія програми, втрата даних, серйозний витік ресурсів;
- серйозна. Велика втрата функціональності;
- суттєва. Звичайна проблема, деяка втрата функціональності при певних обставинах;
- незначна. Несуттєва втрата функціональності або відомо нескладне рішення, яке можна обійти;
- тривіальна. Косметичні дефекти: опечатка, розташування на екрані, оформлення і т. д.

У роботі запропоновано виконувати класифікацію помилок за ступенем їх критичності на базі алгоритму Цукамото [3]. Алгоритм включає такі кроки:

1) Формування бази правил системи нечіткого виведення здійснюється у вигляді упорядкованого узгодженого списку нечітких продукційних правил у вигляді «IF A THEN B», де антецеденти ядер правил нечіткої продукції побудовані за допомогою логічних зв'язок «I», а консиквенти ядер правил нечіткої продукції прості;

2) Фазифікації вхідних змінних здійснюється аналогічно алгоритму Мамдані [4];

3) Агрегування підумов правил нечіткої продукції здійснюється аналогічно алгоритму Мамдані за допомогою класичної нечіткої логічної операції «I» двох елементарних висловлювань;

4) Активізація підвисновків правил нечіткої продукції проводиться в два етапи. На першому етапі, ступені істинності висновків (консиквенти) нечітких продукційних правил знаходяться аналогічно алгоритму Мамдані, як алгебраїчний добуток вагового коефіцієнта і ступеня істинності антецедента даного нечіткого продукційного правила. На другому етапі, на відміну від алгоритму Мамдані, для кожного з продукційних правил замість побудови функцій приналежності підвисновків вирішується рівняння $\mu(x) = c$ і визначається чітке значення вихідної ω лінгвістичної змінної, де $\mu(x)$ і c – відповідно функції приналежності термів лінгвістичних змінних і ступеня істинності нечітких висловлювань, що утворюють відповідні слідства (консиквенти) ядер нечітких продукційних правил;

5) Акумуляція висновків правил нечіткої продукції не проводиться, оскільки на етапі активізації вже отримані дискретні множини чітких значень для кожної з вихідних лінгвістичних змінних.

6) На етапі дефаззифікації для кожної лінгвістичної змінної здійснюється перехід від дискретної множини чітких значень $\{w_1, \dots, w_n\}$ до єдиного чіткого значення згідно дискретного аналогу методу центру тяжіння

$$y = \sum_{i=1}^n c_i w_i \cdot \sum_{i=1}^n c_i, \quad (1)$$

де n – кількість правил нечіткої продукції, в підвисновках якої фігурує дана лінгвістична змінна, c_i – ступінь істинності підвисновка продукційного правила, w_i – чітке значення даної лінгвістичної змінної, отримане на стадії активізації шляхом вирішення рівняння

$$\mu(x) = c_i, \text{ тобто } \mu(w_i) = c_i, \quad (2)$$

а $\mu(x)$ – функція приналежності відповідного терма лінгвістичної змінної.

Висновки: У роботі запропоновано застосувати алгоритм Цукамото для нечіткої класифікації помилок у тестуванні програм. Експерименти показали підвищення достовірності класифікації помилок у порівнянні з результатами класифікації, виконаної фахівцями з контролю якості в середньому до 1,4 раз. Дані результати рекомендовано використовувати для широкого кола задач класифікації, що розв'язуються в області розробки та тестування програм.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Блэк Р. Ключевые процессы тестирования. Планирование, подготовка, проведение, совершенствование /Р Блэк. – Москва: Лори. – 2006. – 125 с.
2. Козина Ю.Ю. Методика параметризации данных при автоматизации тестирования приложений / Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. – Київ, 2013. – Вип. 44. – С.107 – 110.
3. Круглов В.В. Нечеткая логика и искусственные нейронные сети / В. В. Круглов, М. И. Дли, Р. Ю. Голунов. – М. : ФИЗМАТЛИТ. – 2001. – 224 с.
4. Рыжов А.П. О качестве классификации объектов на основе нечетких правил / А.П. Рыжов // Интеллектуальные системы. – 2005. Т.9.В.1-4. – С. 253 – 264.

Kozina Y.Y., Tkachenko O.V., Kozin A.A.

Usage of Tsukamoto Algorithm for Fuzzy Classification of Defects in Software Testing

The software testing role is very important during the programs development lifecycle. Search for defects and their classification is an important task. Incorrect classified defects adversely affect for future cycle of development. On this thesis the classification of defects to perform is offered using the mathematical apparatus of fuzzy logic, namely, on the basis of the Tsukamoto algorithm. Applying of this algorithm leads to increase of classification reliability.

Keywords: testing, fuzzy classification, programs.