Michał Mnich

Streszczenie pracy doktorskiej

Rozprawa doktorska dotyczy wybranych zagadnień inżynierii oprogramowania. Celem rozprawy jest zaprezentowanie nowych metod optymalizacji testowania mutacyjnego bądź wykorzystanie ich w popularnych metodykach wytwarzania oprogramowania. Testowanie mutacyjne jest uznawane za jedną z najefektywniejszych metod testowania kodu oprogramowania. Wadą tego podejścia jest duża złożoność czasowa i pamięciowa. Niniejsza rozprawa dotyczy zagadnień optymalizacji procesu mutacyjnego oraz jego zastosowania w procesach inżynierii oprogramowania. W pracy zaproponowano szereg mechanizmów optymalizacyjnych, mających na celu zmniejszenie czasu trwania procesu mutacji kodu.

Pierwszy mechanizm dotyczy redukcji liczby mutantów na podstawie analizy zmian w kodzie pomiędzy różnymi wersjami oprogramowania. Drugi wykorzystuje podejście bayesowskie w celu optymalizacji prawdopodobieństwa generacji mutantów z określonej grupy operatorów mutacyjnych tak, aby zredukować liczbę mutantów nie zmniejszając jednocześnie znacząco efektywności procesu analizy mutacyjnej. Model trzeci dotyczy generowania wielu mutantów w jednej kompilacji. Przedstawione zostały tu wyniki teoretyczne oraz eksperymenty weryfikujące, czy przy użyciu modelu następuje poprawa wydajności procesu testowania mutacyjnego. W pracy wprowadzono również wersję metodyki Test-Driven Development wzbogaconej o krok zawierający testowanie mutacyjne. Eksperymentalnie potwierdzono, że stosowanie tak wzbogaconej metodyki przyczynia się do podniesienia jakości kodu.

Do przeprowadzania testowania mutacyjnego został wykorzystany model samoadaptacyjnego, rozproszonego, skalowalnego systemu. System ten został zaimplementowany przez autora rozprawy jako klaster obliczeniowy z zaimplementowaną metodą optymalizacji procesu mutacji i testowania. Rozprawa zawiera również opis architektury tego systemu.

W rozprawie podjęto się uzasadnienia następujących tez badawczych:

Teza 1. Efektywność testowania mutacyjnego nie zmniejsza się znacząco, jeśli w nowej wersji oprogramowania mutanty generowane są wyłącznie dla kodu nowego bądź zmodyfikowanego.

Teza 2. Zmiana rozkładu prawdopodobieństwa losowania operatorów mutacyjnych, dokonana przy użyciu podejścia bayesowskiego, może być wykorzystana do istotnego zmniejszenia liczby mutantów przy dopuszczalnej niewielkiej utracie efektywności procesu.

Teza 3. Wzbogacenie podejścia Test-Driven Development o testowanie mutacyjne znacząco podnosi charakterystyki jakościowe kodu i zwiększa niezależność testowania poprzez obniżenie zjawiska stronniczości (ang. bias) autora testowanego kodu.

Teza 4. Podejście polegające na wprowadzeniu wielu mutacji do jednej kompilacji ma swoje ograniczenia i nie każdy operator mutacyjny może współistnieć z wszystkimi innymi.

Teza 5. Podejście polegające na wprowadzeniu wielu mutacji do jednej kompilacji dla języków, w których modyfikacja kodu pośredniego nie jest możliwa, może znacząco przyspieszyć proces testowanie mutacyjnego.

.....