

# SZOFTVERHIBÁK AUTOMATIKUS JAVÍTÁSI LEHETŐSÉGEINEK TOVÁBBALAKÍTÁSA

*Csuvik Viktor*

*II. évf. programtervező matematikus MSc*

*Témavezető: Dr. Vidács László*

*SZTE TTIK Szoftverfejlesztés Tanszék*

*MTA-SZTE Mesterséges Intelligencia Kutatócsoport*

Bár a szoftver minőségmérés egy igen jól kutatott terület, nem tudjuk biztosítani, hogy egy program hibamentes legyen. A minőség meghatározása persze elég szubjektív, mégis manapság egy jól bevett gyakorlatnak számít a szoftvertesztelés a kitűzött cél eléréséhez. A tesztelés elsődleges célja a defektusok felismerése, hogy később ezeket a hibákat ki tudjuk javítani. Mindazonáltal a hibák javítása emberi erőforrást igényel és általában időigényes feladat, így igen költséges. Az automatikus programjavítás pontosan erre próbál megoldást kínálni azáltal, hogy a tesztelés során lokalizált hibákat emberi beavatkozás nélkül javítja ki. A javítás általános menete három részből áll: (1) tesztesetek alapján a hiba lokalizálása, (2) javítás elkészítése, és (3) a potenciálisan javított program ellenőrzése, melyek közül az (1) és (3) pontok a TDK dolgozat fő témái. A tesztesetek kritikus jelentőségűek az egész folyamatban, hiszen egyrészt ezek segítségével lehet megtalálni a hibát, másrészt nagy szerepet játszanak a javítás elkészítése során is. Egy nagy szoftverrendszer több ezer tesztesettel is rendelkezhet, melyek célja az idő során feledésbe merülhet. A dolgozatban bemutatok egy megközelítést, mely kideríti a teszt- és kódosztályok közötti kapcsolatokat, ezáltal azok szerepe érthetőbbé válik. A mai programjavító eszközök kimenetei még elég bizonytalanok, abban az értelemben, hogy a javított programok gyakran kézi ellenőrzésre szorulnak. Mivel a cél eleve az emberi tényező kivonása volt a folyamatból, ez nem túl előnyös. A dolgozat másik felében azt mutatom be, hogy a potenciálisan javított programok közül hogyan lehet kiszűrni a nagy valószínűséggel helytelen szoftvereket. A két megközelítés nagyban épít a forráskódon mért szöveges hasonlóságra, melynek mérésére több technikát is bemutatok. A dolgozatban három gépi tanulási algoritmus kerül bemutatásra, melyek mindegyike a programok három különböző reprezentációján dolgozik. A két technikát nyílt forráskódú adatbázisokon értékeltem ki, ahol igen jó eredményeket értek el. Az eredmények újszerűségét bizonyítja a témában írt nemzetközi publikáció is, mely elfogadásra került és jelenleg megjelenés alatt áll.

**Kulcsszavak:** automatikus programjavítás, gépi tanulás, mesterséges intelligencia, szöveges hasonlóság