

# Hiba lokalizációs algoritmusok és hibaosztályok közötti kapcsolat vizsgálata JavaScript programokban

*Szatmári Attila*

*II. évf. programtervező informatikus MSc*

*Témavezető: Dr. Beszédes Árpád  
SZTE TTIK Szoftverfejlesztés Tanszék*

A szoftverfejlesztés egyik alappillére a hibás kód felderítése és javítása, más néven „debugging”. Habár elengedhetetlen része a minőségi szoftvernek, a mai napig legtöbbször manuálisan történik, amely idő és pénz veszteséghez vezethet.

Erre a problémára nyújtanak megoldást a hiba lokalizációs módszerek. Az automatizált hiba lokalizációs algoritmusok kevés erőbefektetéssel, olcsón találják meg az egyes hibákat a programban ezáltal segítik a fejlesztők munkáját, és meggyorsítják a szoftverfejlesztés életciklusát. Egy Fault Localization (hiba lokalizációs) röviden FL algoritmus bemenete a forráskód, a hozzá tartozó tesztek eredménye és viselkedése (kód lefedettség). A lefuttatása után pedig outputként egy listát ad, melyben az egyes metódusok rangja van aszerint, hogy az FL mennyire találja gyanúsnak. Több fajta FL módszer létezik, a legnépszerűbbek a Specturm-based, Mutation based, Data Mining-based technikák. A Spectrum-Based Fault Localization (SBFL) egy statisztikán alapuló hiba lokalizációs módszer. Számos tanulmány készült már a hatékonyságának vizsgálatáról, azonban nagyrészt Java és C/C++ programokon lettek futtatva az egyes algoritmusok, míg tudomásom szerint JavaScript projekteken aligha.

A kutatásunk során egy viszonylag új bug benchmarkot használtunk a BugsJS-t, melyen kiértékeltek, hogy mennyire jól teljesítenek az egyes SBFL algoritmusok. Megvizsgáltuk, hogy a legnépszerűbb FL algoritmusok (Tarantula, Ochiai és Dstar) milyen pontossággal tudják megjósolni a hibát ezekben a JavaScript projektekben. Továbbá azt vizsgáltuk van-e szignifikáns különbség az egyes algoritmusok hatékonysága között, illetve, hogy van-e valami kapcsolat a hibaosztályok azaz bug-fix típusok és az SBFL performanciája között.

A BugsJS minden egyes bugjára lefuttattuk a három implementált FL algoritmust (Tarantula, Ochiai, Dstar). Ez egy „gyanúsági” listát adott, melyben minden egyes metódus gyanúsági értéke meg van adva. Egy publikált hibaosztály séma alapján manuálisan felcímkeztünk minden egyes metódust a bugot tartalmazó forráskódokban.

Végül, az eredmények alapján az SBFL algoritmusok hasonlóan teljesítenek, azonban az egyes hibaosztályok a fő kategóriák alapján szignifikánsan elkülöníthetők az SBFL algoritmusok hatékonysága alapján. Továbbá ha a fő kategóriákat tovább bontjuk alkategóriákra akkor új eredményeket kapunk.

Ekkor kiderül, hogy a Dstar algoritmus az MC-DAP (Method Call with Different Actual Parameters) kategóriával felcímkeztett bugokat nagyobb eséllyel rakja a gyanúsági sorrend első helyére, mint a többi SBFL algoritmus. Viszont ha a fő kategóriát nézzük, tehát az MC-t (Method Call), akkor nincs különösebb eltérés a három algoritmus kimenetele között.

Ha az IF (bármely if-hez kapcsolódó változás) kategóriát is tovább bontjuk, akkor megjelenik az az eredmény, hogy a top-3 kategóriában az IF-ABR (Addition of an Else Branch) szignifikánsan

különbözik a többi kategóriától (rossz irányban) minden egyes algoritmussal. Ennek következtében ebbe a kategóriába tartozó bugokat nehezebben lokalizálja a három vizsgált algoritmus.