

Beszédes Árpád

Code Analysis Techniques for Debugging and Software Maintenance

című MTA doktori értekezésének bírálata

Célkitűzés

Az értekezés célkitűzése, **a szoftver alkalmazások tesztelésének támogatása** az informatika egyik kulcsproblémája.

A digitalizáció mindenütt meglévő igényének hatására ugyanis megjelent az iparszerű szoftvertermelés. A hibák potenciálisan nagy gazdasági, sőt környezeti és életvédelmi hatása miatt a szolgáltatásminőséggel és -biztonsággal szembeni elvárások is folyamatosan nőnek.

Ma az informatikai rendszerek hibás működésében dominálnak a szoftver hibák, dacára annak, hogy már most is a szoftvertermék előállítási folyamata által igényelt erőforrások és idő nagy része a minőség vizsgálatára terjed ki.

Az elmúlt időszak egyik új trendje éppen ezért az, hogy megpróbálják a kódgenerálási folyamatot mindinkább mesterséges intelligenciával megvalósítani, de a technológia jelenlegi fejlettsége mellett a generált kód minősége különösen az extra-funkcionális követelmények figyelembevételével erősen kifogásolható.

Ennek megfelelően minden olyan eredmény, amely a szoftver minőségbiztosítási folyamatot hatékonyabbá tudja tenni, távlatilag pedig automatizálni tudja, alapvető technikai és gazdasági előrelépést jelent. Ugyanakkor minden automatizálásnak előfeltétele az, hogy a terméket részletesen mérni lehessen.

A disszertáció fő célkitűzése a szoftver termékek ellenőrzésének gyakorlatához illeszkedő, ugyanakkor reális számítástechnikai ráfordítással megvalósítható módszerek kidolgozása, az ellenőrzés, hibalokalizáció, illetve minőségmérés folyamatát hatékony támogatására.

A feladat természeténél fogva jelentős mértékben empirikus alapú és következőképpen heurisztika alkalmazását igényli. Az eredményeket jelentős kísérleti munka alapozza meg, így elméletileg megalapozott alkalmazott kutatás körébe sorolható.

Az értekezés felépítése

Az értekezés a tézisek csoportosítását követve három fő részre tagolva tizenegy fejezetből áll. 34 saját és 113 külső hivatkozást tartalmaz.

1. Bevezetés

A rövid bevezető fejezet a kutatási téma felvázolása mellett a saját fő eredményeket ismerteti felsorolásszerűen, 3 téziscsoportra tagolva, összesen hét altézist magában foglalva.

Kérdések: Szerencsés lett volna, ha a fejezet részletesebb szoftvertechnológiai kontextust adott volna, illetve a fő célkitűzéseket itt is összefoglalja.

- **Hol látja az értekezés szerzője az ismertetett módszerek szerepét a szoftverfejlesztési folyamatban (ez a szövegben csak utalásszerűen jelenik meg) és az elérendő céloknak mi a technológia jelentősége?**
- **Milyen méretű és absztrakciójú programok vizsgálata a célkitűzés?**
- **Az ismert egyéb technológiai megoldásokhoz képest milyen előnyöket vár az ismertetendő kutatási eredményektől?**
- **Hogyan viszonyul az értekezés alapját képező technológiai megközelítés az egyéb paradigmákhoz, van-e köztük kiegészítő szerep?**
- **A szűkebb célterületen belül mik azok a tulajdonságai a korábban ismert módszereknek, amelyeket a jelen kutatás meg akar haladni?**

2. Gráfmentes programszeletelés

Az értekezés második fejezete a gráfmentes dinamikus programszeletelés témakörét foglalja össze. Bemutatja az alapvető célkitűzéseket és alkalmazási eseteket. Pseudo-algoritmus formájában összefoglalja a főbb algoritmusokat. Leírásuk szabatos, de rendkívül tömör.

Alkalmazási szempontból a 2.1 táblázat kiemelkedően fontos, részletesebb leírást igényelt volna.

A fejezet fő eredményét a 2.1 altézis foglalja össze, amelyet egy rangos nemzetközi konferencia publikáció támaszt alá. 14 hivatkozása van, és gyakorlati alkalmazását egy amerikai szabadalom is alátámasztja. Az értekezés a társszerzős megalapozó cikkben belül korrektül elhatárolja a jelölt saját munkáját.

Az altézist elfogadom.

3. Dinamikus függvénycsatolás

Az értekezés harmadik fejezete azon algoritmusokat ismerteti, amelyek az egyes függvények változókon keresztüli csatolását határozzák meg. A definíciók és az algoritmusok tiszták.

Kérdés: Felvetődik, hogy a jelentős futási idő, illetve tárbonyolultság kezelésére milyen javítási lehetőségek vannak.

- **Milyen korlátot jelent a futási idő és tárbonyolultság a gyakorlati elemezhetőség szempontjából a kódméretre?**
- **Bár az értekezés erre vonatkozóan adatot nem közöl, de az intuíció azt sejteti, hogy a relációs gráfok ritkán szövöttek. Ennek megfelelően felmerül, hogy a ritka adat-szerkezetek speciális technikái alkalmazhatók-e a tárbonyolultság csökkentésére?**
- **Van-e olyan gráfelméleti modell, amelyik a 3.3.2 alfejezetben bemutatott konvergenciát elméletben is alátámasztja (Az esetleges gráfelméleti reprezentáció nemcsak prezentációs kérdés, hanem ez utóbbi terület könyvtárnyi irodalma segíthet a hatékony algoritmusok kidolgozásában az értekezés további részeiben is)?**
- **Hogyan viselkedik az algoritmus akkor, ha a rendszerben globális változókon keresztül csatolt egyébként konkurens függvények vannak, hiszen ezek hívási és változóhivatkozási kapcsolata véletlenszerű lehet?**

A fejezet az 1.2 altézist alapozza meg. Ez egy rangos nemzetközi konferencia publikáció támasztja alá.

Az altézist elfogadom.

4. Függőségek számításának alapjai

A rendkívül rövid, összesen két oldalas fejezet a programszeletelés alapjait foglalja össze, precízen, az érthetőség határán tömören, de matematikai formalizálás, ábra és példa nélkül.

5. Függőségi klaszterek számítása és analízise

A fejezet bevezetője helyesen, informálisan mutatja be azt, hogy a nagyméretű program-klaszterek fejlesztés és minőség szempontjából egyaránt problematikusak.

A fejezet elsőként a statikus, végrehajtási sorrendet alapul vevő függőségi klaszterek vizsgálatát veszi végig. Szellemes vizuális és megalapozott metrika alapú vizsgálati módszert dolgoz ki. Azok hasznosságát kísérleti alapon mutatja be. A minták statisztikai kiértékelése korrekt. Csak említi, de eredményeket nem közöl a Kendall-korreláció számításának eredményéről.

A következő kérdéskör, amelyet a fejezet tárgyal, azon speciális elemek keresése, amelyek erős klaszterezést okoznak.

A kidolgozott eredmények alapján már a gyakorlati hasznosításhoz közeleső módon mutatja be a függőséganalízis és a hatásanalízis kapcsolatát.

A fejezet a 2. tézist alapozza meg. A tézist két rangos nemzetközi folyóiratcikk és egy konferencia cikk publikálta.

A tézist elfogadom.

6. Spektrumalapú hibalokalizáció alapjai

A rövid két oldalas bevezető fejezet a hibadetektálási gyakoriságon alapuló hibalokalizációs technikák háttérét mutatja be. Számos származtatott metrikát mutat be azok formulán túlmenő ismertetése nélkül.

7. Megbízható kódfedési mérés technika

A fejezet egy a gyakorlat szempontjából lényeges kérdésnek a technikai metrológiai vizsgálatát tűzi célul. A jelölt a kísérleti vizsgálatokat Java programokon, két mérőeszköz segítségével végezte, amelyek forráskód, illetve byte kód instrumentációján alapulnak. Az adott környezetben mindkét eljárás intruzív volt és az adott technológiák eltérése miatt esetenként érdemileg eltérő eredményeket szolgáltatott.

Az értekezés ezeket az eredményeket, illetve kiváltó okaikat mélyen elemzi, majd kvalitatív elemzéssel bemutatja azok potenciális hatását.

Ez a fejezet kimagasló színvonalú szoftvermérnöki munkát tükröz, más hasonló feladatok számára is útmutató. Ugyanakkor a szoftver karbantartási folyamatra való kitekintés is erősen kötődik a tárgy- és eszköztechnológiához.

A leírtak és a következtetések erős technológiai kötődésük miatt azonban általánosított és újrahasználatos tudományos eredményt ez az egyébként szakmailag értékes gondolatmenet nem hozott.

A fentiek miatt az egyébként két rangos publikációval alátámasztott **3.1 altézist a kellően megalapozott általánosítás hiányában nem fogadom el.**

8. Hívási láncok használata a spektrumalapú hibalokalizálásban

A fejezetben ismertetett eredmények kiinduló gondolata az, hogy a teszt-hívott függvény vizsgálati kapcsolatok eredményének pusztán GO/NOGO alapú tárolása és kiértékelése túlságosan nagy információvesztéssel jár.

A kibontott eredeti gondolat az, hogy az egyszerű, aggregált függőségen túl a hívási láncokhoz való rendelés érdemileg javítja a megőrzött információ tartalmát. A sorrendezésen alapuló kiértékelés és hibalokalizálás információvesztés okozta bizonytalanságának csökkentésére pedig két metrika származtató módszer kombinációját alkalmazza, ami a hibatűró rendszerekben használatos két verziós redundáns tervezési minta egy speciális válfajának felel meg.

A részletesen elvégzett és kiértékelt vizsgálatok igazolják a módszer eredményességét. Ez a fejezet igen jól követhetően, tisztán van megírva.

A fejezet a 3.2 altézist alapozza meg, amelyet egy rangos nemzetközi konferencia publikáció támasz alá.

Az altézist elfogadom.

Kitekintő kérdés:

- **Jól érzékelhető módon az absztrakcióval járó információ tömörítés mértékének csökkentése a kapott metrikák tulajdonságaira pozitív hatást gyakorol. Van-e reálitása annak, hogy az egyes tesztek eredményeit nemcsak bináris Go/NoGo értékkel vegyük figyelembe, hanem a tényleges hibajelzéseket vagy súlyosságukat is használja az algoritmus?**

9. Hívási frekvenciák használata spektrumalapú hibalokalizálásban

A fejezet célkitűzése az hogy a korábbi fejezetekben ismertetett módszerek nagymértékű információtömörítés és -vesztés miatti bizonytalanságát redukálja a hibajelzés tényén túl annak gyakoriságának figyelembevételével. Az egyszerű számláláson alapuló naív módszer bizonyos elemi programozási konstrukciók hatására is torz eredményeket szolgáltat, ezért a fejezet javasolja az egyedi előfordulások számának mértékként használatát. Erre a programok mérésénél jól használható praktikus metodikát is kidolgoz. Az ötlet hatékonyságát egy szokásos benchmarkon mutatja be.

A gondolatmenet logikus, az ötlet eredeti, ezért az arra épülő 3.3. altézist elfogadom.

Kitekintő kérdés:

- ***A metrikák származtatásának megfelelően azok nem csak a vizsgált programtól, hanem a tesztkészlettől is függenek. A tesztkészlet létrehozása elsősorban a hibadetektálási képességért veszi alapul. A szokásos fedési metrikák fényében hogyan alakul a hibalokalizációs folyamat? Jelentene-e például különbséget az, hogy a tesztkészlettel csomópont vagy útfedést célozd meg annak tervezője?***

10. Dinamikus szeletelés használata spektrumalapú hibalokalizálásban

Az alapvető módszereknél, mint a pusztán fedési alapú vizsgálatoknál a nagymértékű információvesztés nyilvánvalóan rontja a becslések hatékonyságát. A fejezet ezért azt vizsgálja, hogy a dinamikus programszeletelés alkalmazása hogyan javítja a spektrum alapú hibalokalizáció hatékonyságát. A friss eredmények rámutatnak a megközelítés ígéretességére, de természetesen finomhangolása további vizsgálatokat igényel. Ugyanakkor az eredeti és tiszta alapgondolat és a megközelítés tartós eredménnyé formálja.

A kapcsolódó 3.4 altézist ezért elfogadom.

11. Konklúzió

A záró fejezet kevesebb, mint egy oldal terjedelemben informálisan összefoglalja az értekezés tartalmát, valamint az azt támogató projekteket. **Szívesen láttam volna a jövőbeli lehetséges kutatási irányok felvázolását.**

Az értekezés általános értékelése

Igazoló kísérletek

Az értekezésben összefoglalt munkát jelentős és nívós kísérleti tevékenység támasztja alá. A rendkívül körültekintő és a szóbajövő hibaforrásokat is elemző metodika meggyőző. A szakterületen szokásos benchmark alapú kutatás keretei között garantálja az eredmények megalapozottságát.

A kísérleti jelleg egyúttal magával hozza azt is, hogy a tapasztalati tényeken alapuló általánosítás nem bizonyítható egzakt módon, de egy jól reprodukálható metodika igazolhatja azok potenciális létjogosultságát. Ennek megfelelően tudományos eredménynek értékeltem mindazon megfogalmazott tapasztalatokat és általánosításokat, amelyeknél a további kutatások majd alkalmazási területük létjogosultságát jobban körvonalazzák, illetve érvényességi körüket behatárolják.

A kísérletek prezentációja némiképpen túlrészletezett. A kísérletek egyes technikai elemei publikációban indokoltan szerepelnek, ugyanakkor egy értekezésben szerencsésebb lett volna ezeket függelékbe gyűjteni.

Megjegyzés: A metodika alapossága minimális formalizáltság (például egy kauzális függőségi gráfra történő felfűzés után) önmagában is megalapozhatott volna egy altézist.

Kérdések:

- **Miközben a kísérletek eredményeinek kiértékelése részletes, jó lett volna, ha az észlelt kulcselemek esetében megtörténik a vizsgált objektumok tulajdonságaira való visszavezetés. Például az 5. fejezetben bemutatott linchpin elemeknél a gyakorlat számára is érdekes lenne, hogy a beazonosított részeknek mi a funkcionalitása?**
- **Mennyiben tekinthetők a benchmarkként használt szoftver elemek reprezentatívnak?**
- **A szoftver területen számos komplexitási metrika létezik. Lehet-e ezeket figyelembe venni a becsléseknél?**
- **Az értekezés alapvetően bemenő paraméterként kezeli a tesztkészletet. ezzel kapcsolatosan két kérdés is felmerül kitekintő jelleggel. Az alkalmazott**

módszerek a tesztelés melyik fázisában (modul, integrációs tesztelés) hatékonyak vagy hatékonyabbak? Származtathatóak-e olyan tervezési kritériumok a teszt tervezés számára, amelyek betartása esetén a bemutatott módszerek hatékonysága nő?

- ***Az ilyen komplexitású problémák esetén az utolsó időben robbanásszerűen terjed a gépi tanulás alapú módszerek használata. Lehet-e ezeknek ennél a problémakörnél érdemi szerepük?***

Prezentáció

A disszertáció összességében nagyon jelentős anyagot foglal magában. A terjedelmi korlátok miatt ugyanakkor szokatlanul tömör, ami az olvasást és a bírálat elkészítését is megnehezíti.

Az értekezés a saját eredmények közzlésére fókuszál, is de a környezet és az egyéb kutatási eredmények bemutatása túlságosan szűkös.

Szerkesztési szempontból szerencsésebb lett volna az algoritmikus megalapozást szolgáló részeket az értekezés elején, egyetlen összevont, áttekintő fejezetben megadni.

Nem szerencsés az sem, hogy a szöveg az egyes megközelítések tulajdonságairól állításokat mond ki, de sokszor még informális indoklást sem ad, csak irodalmi hivatkozást.

Az értekezés nagyszámú irodalmi hivatkozást tartalmaz, azonban azok helyenként nincsenek kibontva (alkalmanként egy-egy hivatkozás csoport öt-hat hivatkozást is tartalmaz felsorolásszerűen, anélkül, hogy ezek a lényegét kiemelné és kritikának vetné alá), ami az olvasótól irodalmazást tesz szükségessé. Ugyanakkor a III rész lényegesen jobban taglalja ezeket.

Az értekezés olvashatóságát nagymértékben javította volna, ha az első két részben több magyarázó ábra, és esetenként a technológiai kötet bemutató rövid példa jelent volna meg, akár függelékben.

Az értekezés angol nyelven íródott, kifogástalan nyelvezettel.

A fő eredmények megfogalmazása matematikailag szabatos és precíz.

A korábbiakhoz hasonló követhetőségi kérdés az, hogy az algoritmusok és eredmények diszkussziójánál célszerű lett volna a matematikai változók mellett azok korábban megadott szemantikus jelentését is befoglalni a szövegbe, vagy legalább a függelékben egy jelölésgyűjtemény megadása.

Ugyanakkor kifejezetten zavaró az értekezésen végigvonuló egységes jelölésrendszer hiánya, azaz egyes fogalmakat a különböző fejezetek különféleképpen jelölnek. Például a program jelölésére megjelenik az egyik fejezetben P, a másikban pedig X. Hasonlóan például a spektrum mátrix jelölése és használata eltér a 6. és a 9. fejezetben, bár ez utóbbi visszahivatkozta.

Értékelés

Az értekezést 34 saját publikáció támasztja alá, amelyek szinte kivétel nélküli többszerzősek. Ezek nemzetközileg is kiemelten rangos fórumokon jelentek meg zömmel. Hivatkozási számuk megfelelő, összességében kiemelkedő. Az informatikai terület sajátosságainak megfelelően a nemzetközi konferencián megjelent publikációk dominálnak.

Az egyes fejezetek összefoglalói összekötik a szerződés saját publikációit és altéziseket, kellőképpen kiemelve a társszerzős publikációkban a jelölt szerepét. Hangsúlyozom, hogy a munka jellegénél fogva a publikációk zömének többszerzős volta természetes.

Összefoglalás

Beszédes Árpád MTA doktori értekezését alkalmasnak tartom nyilvános vitára bocsátásra. Az elfogadásra javasolt tézések új tudományos eredményeket fogalmazzanak meg és kellőképpen publikáltak.

Budapest, 2025. 10.14.



Pataricza András

Az MTA doktora