Tesztek és tesztelt osztályok felderítése szövegfeldolgozási módszerekkel

Kicsi András
II. évf. programtervező informatikus MSc

Témavezetők: Dr. Vidács László, Dr. Gyimóthy Tibor
MTA-SZTE Mesterséges Intelligencia Kutatócsoport, SZTE TTIK Szoftverfejlesztés Tanszék

A szoftverminőség javításának és fenntartásának érdekében a szoftverrendszerek fejlesztése során rengeteg kód készül a programkód tesztelésére. Napjainkban a teszt-rendszerek akár több tízezer tesztből is állhatnak. A teszteket készítők jelenléte, illetve megfelelő konvenciók híján a tesztesetekről sokszor nehéz és időigényes feladat eldönteni, hogy melyik kódrészlet tesztelésére íródtak. E dolgozat készítése során ezt a problémát a korábban legyakrabban alkalmazott módszerekkel szemben szövegfeldolgozási eszközökkel próbáltuk megoldani. A kódok egymáshoz rendelését *Latent Semantic Indexing* segítségével végeztük.

A feldolgozáshoz a statikus elemzés segítségével a szoftver kódját tokenekre bontjuk, amelyekben nemcsak a metódusok, hanem azok kommentjei is szerepelnek. Ettől a ponttól kezdve szövegként kezeljük a teljes programkódot is. A tesztkódot és a többi programkódot szétválasztjuk, és egy szöveges előfeldolgozás után ezen két halmaz elemei között keresünk szemantikai hasonlóságot LSI felhasználásával, ami más alkalmazási területeken már jól bevált. Az elkészült rendszer ajánlórendszerként működik, a legvalószínűbb N darab kapcsolatot adja eredményül. Ezen kimenet jó kiindulópontul szolgálhat a tesztelt programkód kereséséhez.

Munkánk során négy, nyílt forrású szoftverre próbáltuk meghatározni az egyes tesztesetekkel legnagyobb valószínűséggel kapcsolatban álló programkód-osztályokat. Ezek az Apcahe Commons *Commons Lang* és *Commons Math* komponensei, valamint az *ArgoUML* és *Mondrian* rendszerek. Munkánkat Java nyelven írt szoftverekhez optimalizáltuk.

Az eredmények ellenőrzéséhez egy, a témában alapvető műnek számító, 2009-ben megjelent cikk eredményeit vettük alapul, amely szerint elnevezési szabályok alapján teljes precizitással felfedezhetőek a tesztesetek és programkódok közötti kapcsolatok. Ez alapján egy kiértékelő modult is készítettünk, amely az elnevezések és elérési utak közötti hasonlóság alapján meghatározza a valósnak ítélhető kapcsolatokat, amelyeket összehasonlít a rendszerünk kimenetével. Habár a módszert a betartott névkonvenciókhoz mérve értékeltük ki, attól teljesen független megoldást adtunk, ami a névkonvenciók nélkül is alkalmazható. A tesztek és a hozzájuk tartozó kódrészletek összekötésének egyik fontos alkalmazási területe lehet a hibák lokalizációja, melynek irányába tervezzük folytatni a kutatást.