



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

Az IntelliTest tesztgeneráló eszköz megismerése

Csonka Lóránt, mérnökinformatikus szak
Konzulens: Honfi Dávid, MIT

Rendszertervezés szakirány
Önálló laboratórium összefoglaló
2015/16. II. félév

A szoftverek, és programozott rendszerek ma már teljesen az életünk részévé váltak. Ott vannak a zsebünkben, az asztalunkon, az autókban, Egy szoftver helyes működésén így adott esetben sok pénz, idő, vagy akár emberélet is múlhat.

Ma a szoftver egy komoly mérnöki munka eredménye. A "kódolás" a fejlesztési folyamat egy szelete. Rengeteg szoftverfejlesztési modell és módszer létezik (vízesés, inkrementális, spirál, agilis, stb.), melyek célja, hogy a tervezés a fejlesztés és végül a tesztelés egy, az igényeknek megfelelő, jól, és megbízhatóan működő szoftvert eredményezzen.

Az IntelliTest tesztgeneráló eszköz kipróbálásához előbb a tesztelés alapvető fogalmaival, majd az egységesítéssel kellett megismerkednem. Az egységesítést (unit teszt) szoftver legkisebb egységeire irányuló teszt. A teszt során egy jól meghatározott egységet, függőségeitől izolálva, csupán a belső logikára koncentrálni tesztelünk, így biztosítva azt, hogy csupán az adott egység működését ellenőrizzük.

Unit tesztet automatikusan is lehet generálni, ilyenkor a forráskódból vagy a specifikációból állítunk elő teszteteket. Ezek a tesztetek két dolgot lehetnek hivatottak tesztelni, magát a kódot (white-box tesztelés), vagy a specifikáció által megadott funkcionálisitást (black-box tesztelés)

Az IntelliTest egy szimbolikus végrehajtást alkalmazó, kód alapú tesztgeneráló eszköz. Használat során, a tesztgenerálás és futtatás után egy kiértékelő ablakban összesíti a generált teszteteket, kilistázza, hogy milyen paraméterekkel hajtotta végre a tesztet, és milyen kimenetek keletkeztek, továbbá egy "nézőkében" betekintést nyújt a konkrét tesztet implementációkra.

Amennyiben nem vagyunk elégedettek a generált tesztekkel, megkötéseket tehetünk a bemeneti paraméterekre, és a tesztelt metódus kimenetére a PexAssert illetve a PexAssume programsorokkal.

Az IntelliTest-et kipróbáltam egy konkrét projekten is (Popocorn). Futtatás során az egy osztályra generált tesztetek mennyisége átlagosan 15 db a tesztek lefedettsége pedig nagyjából 70%-os volt.

A projektben helyet kaptak a komparátor és konverter osztályokra vonatkozó kézzel írt egységesítések is. Ezekről általában véve elmondható, hogy próbálták a valós funkcionálisitást, logikát tesztelni.

Megpróbáltam összehasonlítani az IntelliTest által készített teszteteket a projektben lévővel, és egy feltűnő problémával találkoztam. Előfordult, hogy amennyiben csupán a generált tesztet nézzük, és nem módosítunk azokon (nem adunk hozzá elő és utó feltételeket), a tesztek túl egyszerű bemenetekre jönnek létre, így sok irreleváns tesztet jön létre. Ez a white-box tesztgenerálásból, és a szimbolikus végrehajtásból adódik.

Így látván a projektben lévő kézzel írt teszteteket arra jutottunk, hogy egy jó cél lenne, ha megpróbálnánk azokból kinyerni a tesztadatokat (bemenetek, elvárások, létrehozott osztályok), és beadni az IntelliTest-nek. Így egy, a specifikáció elvárásaival sokkal jobban összefüggő tesztet tudnánk generálni.

Ahhoz, hogy kinyerjük a tesztadatokat, forráskód analízist kell végeznünk. Ehhez egy jó eszköz .NET környezetben a Roslyn, így a félév végén ezzel az eszközzel próbáltam megismerkedni.