



Tesztgeneráló eszközöket összehasonlító keretrendszer fejlesztése



Cseppentő Lajos, I. évf., MSc mérnök inf. szakos hallgató
Konzulens: Dr. Micskei Zoltán Imre, adjunktus, MIT
Szolgáltatásbiztos rendszertervezés szakirány
Önálló laboratórium 2 összefoglaló
2014/15. I. félév

Az utóbbi évtizedben egyre több módszer és eszköz jelent meg, aminek a segítségével forráskódból lehet tesztbemeneteket automatikusan előállítani és így hibákat találni a vizsgált szoftverben. Az egyik ilyen technika a szimbolikus végrehajtás, aminek a segítségével a program futtatása nélkül felderíthetőek annak lefutási útjai és az azokat aktiváló bemenetek. Több prototípus eszközt is készítettek már, ami ezt a technikát használja Java nyelvű és más platformra készült kódok vizsgálatához. Azonban sok programrészlettel még nem tudnak ezek az eszközök megbirkózni.

Korábban elkezdtem fejleszteni egy olyan keretrendszert, amely segítségével Java platformra készített (nem csak szimbolikus végrehajtással működő) tesztbemenet-generáló eszközöket lehet hatékonyan vizsgálni. Az elkészült SETTE (Symbolic Execution based Test Tool Evaluator) eszköznek bemenetként egy kódrészletet halmazt kell megadni és metaadatokkal ellátni, amelyek segítségével vizsgálni szeretnénk a tesztbemenet-generáló eszközök képességeit és korlátait. A tesztbemenet-generáló eszközök feladata az, hogy ezekhez a kódrészletekhez generáljon olyan bemeneteket, amelyek maximális fedettséget érnek el. A keretrendszer fő feladatai a következők:

- Eszközök számára szükséges fájlok generálása a kódrészletek alapján (konfigurációs fájlok, test-driver; eszközönként eltérnek)
- Eszközök utasítása arra, hogy bemenetet generáljanak a kódrészletekhez, illetve a futtatás kimenetének a rögzítése
- Futtatási eredmények értelmezése és elemzése
- A generált bemenetből futtatható tesztesetek előállítása
- A tesztesetek segítségével az elért fedettség mérése, a futtatások eredményeinek eldöntése és rögzítése

A félév során a hangsúlyt a keretrendszer befejezésére, közzétételére, dokumentációjára és korábban nem vizsgált eszközök kiértékelésére helyeztettem. Az elkészült keretrendszer képes különböző eszközök automatizált kiértékelésére, bővíthető, illetve cserélhető az általa használt kódrészlet készlet, valamint felkészíthető új, általa korábban nem ismert eszközök kiértékelésére. A keretrendszert GitHub-on publikáltam, valamint a fő GitHub repository-hoz készítettem Wiki oldalakat dokumentáció céljából. Emellett a kiértékelési eredményeket is feltöltésre kerültek.

Korábban három Java platformra készített, szimbolikus végrehajtást használó tesztbemenet-generáló eszköz került összehasonlításra (CATG, jPET, Symbolic PathFinder). A félévben két új eszközt vettem vizsgálat alá. Az egyik a Java platformra készült EvoSuite, ami nem szimbolikus végrehajtást használ, hanem elsősorban mutáció vezérelt hibrid módon működő eszköz. Az EvoSuite meglepően jól teljesített, a korábbi eszközökhöz képest erőssége, hogy mindig sikeresen, a rendelkezésre álló időn belül befejezi a tesztbemenet-generálást, emellett kiválóan kezeli az objektumokat, a generikus adatszerkezeteket és az osztálykönyvtári függőségeket. Legnagyobb hátránya talán az, hogy egy szimbolikus végrehajtással működő eszközhöz képest valamennyivel kevesebb kényszert tud kielégíteni.

A másik vizsgált eszköz a .NET platformra készített Pex volt, melyhez a kódrészleteket át kellett alakítani C#.NET nyelvre. Ez nem volt teljesen triviális, mivel a Java és C# nyelvi elemek nem feleltethetőek meg egyértelműen egymásnak (ahogyan a Java bajtkód és az IL kód sem). A Pex sem kerül végtelen ciklusba és sikeresen leáll a rendelkezésre álló időn belül. Fő erőssége a kényszerek kielégítése, objektumok és osztálykönyvtárak használata, azonban a genericitást, kollekciókat és bonyolultabb osztálykönyvtári függőségeket tekintve nem adott az összes kódrészlethez maximális fedettséget elérő bemeneteket.