



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

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 1 összefoglaló
2013/14. II. 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 kódok vizsgálatához. Azonban sok programrészlettel még nem tudnak ezek az eszközök megbirkózni.

A félév során elkészítettem egy olyan keretrendszert, amely 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 kódrészlet halmazt kell megadni, amelyek segítségével vizsgálni szeretnénk a tesztbemenet-generáló eszközök képességeit és korlátait. A kódrészleteket annotációkkal kell ellátni, melyekben nem csak azok csoportosítása, hanem egyéb információk is megadhatók, például maximálisan elérhető (utasítási) lefedettség vagy azon hívott metódusok listája, melyekre ki kell terjeszteni a kódfedettség-vizsgálatot.

A keretrendszer minden vizsgálandó eszköz számára először létrehoz egy ún. futtatóprojektet, ami tartalmazza az annotációktól és egyéb metaadatoktól mentes kódrészleteket valamint az eszköz számára szükséges egyéb elemeket (pl. konfigurációs fájlok, test driver-ek). Ezután az eszközöket arra utasítja, hogy a kódrészletekhez generáljon bemeneteket, majd az eszközök által előállított nyers kimenetet egy feldolgozza, és egy egységes XML formátumba transzformálja. Ennek eredményeként már megállapítható, hogy az eszközök mely kódrészletekre fejezték be sikeresen a futást. A SETTE az elért fedettségeket a JaCoCo eszköz segítségével ellenőrzi. Ennek a fázisnak a bemenete a generált kimeneti XML fájlok valamint a kódrészlet-halmaz, így minden eszköz eredményét azonos módon értékeli, így ezek összehasonlíthatók. A fedettség-mérés eredményét szintén egy egységes XML formátumba menti a keretrendszer.

A futtatások után minden vizsgált eszközre részletesen, egységes formátumban rendelkezésre áll, hogy pontosan milyen eredménnyel zárult a futás, milyen bemeneteket generált valamint azokkal milyen fedettséget ért el. Így minden eszközre részletes visszajelzést ad az általam elkészített keretrendszer, valamint lehetőségünk nyílik arra is, hogy az eszközök eredményeit (és így az eszközöket is) összehasonlítsuk.

A megvalósítás során cél volt, hogy a SETTE képes legyen a Java SE 7 JDK-val működni, más platformoktól (pl. Java EE, Eclipse) ne függjön. Ahhoz, hogy az eszköz a feladatát el tudja végezni, nagymértékben használ reflectiont (elsősorban annotációk értelmezése és kódrészletek validálása), valamint Java forráskódot is elemez és generál. Az eszköz platformfüggetlen, azonban maga az eszközfuttatás fázisnak feltétele, hogy az adott eszköz támogassa a platformot (pl. a jPet csak Linux-on működik). Azonban lehetőség van arra, hogy a fázisokat teljesen eltérő gépeken, ill. platformokon végezzük el.