

Automatikus tesztgenerálás modell ellenőrző segítségével

Micskei Zoltán
Konzulens: Dr. Majzik István
2003/04. I. félév

A szoftverfejlesztés során a tesztelés mindig is a folyamat jelentős eleme volt, mely sok időt és erőforrást igényel. A tesztelés első lépései a tesztek megtervezése, majd generálása. Meghatározzuk, hogy milyen funkciók működését és tulajdonságokat ellenőrizzük vagy mely fedettségi kritériumokat szeretnénk elérni. Ezután elkészítjük ezekhez a bemenet-várt kimenet sorozatokból álló teszteseteket. Részletes, hatékony tesztkészlet előállítására nehéz feladat, a lehetséges bemeneti kombinációk nagy száma miatt pedig könnyű lényeges eseteket kifelejteni.

Modell alapú fejlesztés esetén lehetővé válik, hogy a tesztek generálását automatikusan végezzük el, modell ellenőrző eszközök felhasználásával. Többféle módszert javasoltak már, az egyik egyszerűbb alaplépései:

- A specifikációból vagy az implementációból elkészített, pl. UML-ben megadott, modellt valamilyen modell ellenőrző bemeneti nyelvére átírjuk.
- A fedési kritériumokat temporális logikai kifejezésekkel fogalmazzuk meg.
- Egy olyan futást szeretnénk kapni, ami teljesíti ezt a kritériumot, ezért a kifejezés negáltját ellenőriztetjük, és ha létezik teszteset, akkor azt az eszköz kiadja, mint a negált formula ellenpéldája.
- A kiadódó futás leírásából ki kell választani a bemeneti és kimeneti eseményeket, amik a tesztesetet alkotják.

A félév első részében a kapcsolódó elméleti háttér megismerése, a témában írt cikkek elolvasása a célom. A fent leírt módszer sok kérdést vet fel:

- A klasszikus adat és vezérlés alapú fedési kritériumokat hogyan lehet megfogalmazni temporális logika segítségével, CTL szükséges-e hozzá, vagy megfelel-e a lineáris temporális logika is?
- Könnyen teszteset-robbanás következhet be, így heurisztikákat kell alkalmazni, hogy a generált tesztesetek számát csökkentsük, miközben a hibafedés minél nagyobb. A probléma, hogy a tesztkészlet elemszámának és hosszának meghatározása is NP-teljes feladat.
- A modelltől hogyan kapjuk meg a modell ellenőrzős leírását? Például UML Statechartok hiányos szemantikája miatt nehéz egy formális leírásra transzformálni.
- A teljes tesztelési folyamatba hogyan illeszthető be a módszer? Meg kell ismerni a modell alapú tesztelés megvalósítására javasolt keretrendszereket.

A félév második felében gyakorlatban is kipróbálom a módszert:

- A fent leírt elméleti módszer lépéseit végigvizsgálok egy kisebb feladaton.
- Több kereskedelmi és akadémiai program is képes automatikus tesztgenerálásra, ezek használhatóságát vizsgálom meg.