

Automatikus tesztgenerálás modell ellenőrző segítségével

Micskei Zoltán, V. Inf.

**Konzulens: dr. Majzik István,
Mérés-technika és Információs Rendszerek Tanszék**

A szoftverfejlesztés során a tesztelés mindig is a folyamat fontos feladata, mely sok időt és erőforrást igényel. A tesztelés első lépése a tesztek megtervezése, a megfelelő bemenet - elvárt kimenet sorozatok összeállítása. A szoftverfejlesztésre vonatkozó szabványok gyakran előírják a tesztelés során a forráskódra vonatkozó úgynevezett fedettségi kritériumok teljesítését (pl. minden utasítás, vagy minden döntési ág bejárását a tesztelés során). A kritériumoknak megfelelő, részletes és hatékony tesztkészlet előállítása a lehetséges bemeneti kombinációk nagy száma miatt nehéz feladat.

Modell alapú fejlesztés esetén lehetővé válik, hogy a tesztkészletek generálását automatikusan végezzük el. A TDK dolgozat áttekinti ezeket a módszereket, és részletesen bemutat egy kiválasztott technikát, a modell ellenőrző segítségével történő tesztgenerálást [1], ami elsősorban eseményvezérelt beágyazott rendszerek esetén lehet hatékony. Ennek lényege, hogy a teszt fedettségi kritériumot a modell ellenőrző számára mint követelményt fogalmazzuk meg, ez a legtöbb eszköz esetén temporális logikai állításokkal történik. A követelmény rögzíti a tesztelési cél negáltját (pl. azt előírva, hogy egy adott utasítássorozat nem hajtható végre), ezt ellenőrizve a modell ellenőrző egy ellenpéldát talál (itt megmutatja azt a bemeneti sorozatot, amivel az utasítássorozat ténylegesen végrehajtható). Ez az ellenpélda már közvetlenül használható teszt sorozat konstruálására.

A TDK dolgozat bemutatja egy tesztgenerátor program implementációját, amelynek működése a következő részfeladatok automatikus végrehajtásából áll:

1. A specifikáció alapján elkészített UML állapotterkép modellt a SPIN modell ellenőrző [3] bemeneti nyelvére transzformálja. (Ehhez egy már elkészült modell transzformátor programot alkalmaz [2].)
2. A fedési kritériumokat az UML modell alapján temporális logikai kifejezésekkel fogalmazza meg.
3. Ellenőriztetni a kifejezést, és ha létezik ellenpélda, akkor ebből a bemeneti és kimeneti események kiválasztásával tesztet generál.

A tesztgenerálás igényei (minél rövidebb és hatékonyabb tesztek összeállítása) sok esetben eltérnek a klasszikus modell ellenőrzés igényeitől (biztonsági és élőségi feltételek ellenőrzése). Igen fontos tehát a modell ellenőrzőnek azokat a beállításait kidolgozni, amelyekkel a tesztek hatékonysága mellett a rövid futási idő és a mérsékelt erőforrásigény is biztosítható. A TDK dolgozat egy mintapéldán végzett méréseken keresztül bemutatja a SPIN és az SMV modell ellenőrző optimalizációs lehetőségeit és a kapcsolódó futási eredményeket.

A tesztgenerálás alkalmazhatóságát egy valós ipari rendszer modelljén demonstráljuk. A dolgozat tárgyalja azokat a kiterjesztéseket is, amelyek szükségesek valós idejű rendszerek (időzítési kritériumokat is tartalmazó) teszt sorozatainak generálásához.

Irodalom:

1. H.S. Hong, I. Lee, O. Sokolsky, and S.D. Cha, Automatic Test Generation from Statecharts Using Model Checking, Proc. FATES '01, pp. 15-30, Aalborg, Denmark, 2001.
2. D. Latella, I. Majzik, M. Massink: Automatic Verification of a Behavioural Subset of UML Statechart Diagrams Using the SPIN Model-checker, Formal Aspects of Computing, (11)6, pp 637-664, Springer Verlag, 1999.
3. Gerard J. Holzmann: The Model Checker Spin, IEEE TSE (23)5, pp. 279-295, 1997.