



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

Programszeletelő algoritmusok összehasonlítása

Sallai Gyula, I. évf, (MSc) mérnök inf. szakos hallgató
Konzulensek: Tóth Tamás doktorandusz, MIT
Hajdu Ákos doktorandusz, MIT
Kritikus rendszerek szakirány
Önálló laboratórium 1 összefoglaló
2016/17. II. félév

Ahogy a beágyazott rendszerek egyre inkább életünk szerves részévé válnak, biztonságos és hibamentes működésük egyre kritikusabb a felhasználók és a gyártók számára egyaránt. A tesztelési módszerekkel ellentétben a formális verifikációs technikák nem csak a hibák jelenlétét, hanem hiányát is képesek bizonyítani, ezáltal kiváló eszközzé téve őket biztonságkritikus rendszerek verifikációjához. Az ilyen bizonyítások egy eszköze a formális modellek használata. Az elkészítendő rendszer viselkedését formális modellek segítségével leírjuk, majd az épített modellen különböző lekérdezések futtatásával megtudjuk, hogy lehetséges-e egy hibás állapot. Bár ezek a módszerek képesek kielégítő bizonyítékot szolgáltatni a rendszer helyes működéséről, használatuk valós ipari projektben az idő és költségkorlátok miatt nem mindig valósul meg.

Egy megoldási mód a már elkészült forráskód formális modellé alakítása és a modell ellenőrzése a hibás állapotok bekövetkezhetősége szempontjából. Sajnos a forráskódból formális modellt előállító eszközök gyakran állítanak elő kezelhetetlenül nagy és komplex modelleket, így téve ellenőrzésüket rendkívül bonyolulttá és időigényessé. A problémára orvosolható bizonyos optimalizációs algoritmusok használatával, melyek csökkentik az átalakítás során kapott modell méretét.

Munkámban olyan különböző optimalizációs algoritmusokat mutatok be, melyek képesek az előállított modell méretét jelentősen csökkenteni. A munka legfőképp a különböző *programszeletelő* technikák összehasonlítására fókuszál, köztük a *vissza-*, a *vékony-*, és az *értékszeletelő* algoritmusokat mutatja be és hasonlítja össze.