

Önálló Laboratórium 1.

Összefoglaló

A félév során az előző szemeszterekben kidolgozott modellellenőrző algoritmusommal foglalkoztam. Az eredeti program C++11 nyelven íródott, és két féléven keresztül (BSc. önálló laboratórium, TDK dolgozat, Szakdolgozat keretein belül) sorozatos átdolgozásokon és változtatásokon ment keresztül, ennek eredményeképp kód nehezen érthető és rosszul karbantartható lett. A félév egyik célja a kód átláthatóságának növelése volt. Ezen kívül a C++11-es implementáció saját adatszerkezettel és belső logikával működött, ezáltal a tanszéken zajló más fejlesztésekhez nehezen lehetett illeszteni. Az önálló laboratórium 1. keretein belül az elkészült algoritmusomat a kutatócsoport saját fejlesztésű Zeta Model Checker projektjének Petri hálós részébe kellett beintegrálnom.

A feladat során kihívást jelentett az excesszív pointeraritmetikát használó C++ kódot magas szintű .NET kódra átírni, hiszen egy összetett, bonyolult algoritmusról van szó, és az alapvető program-könyvtárak C++-ban és .NET-ben jelentősen eltérő logikán alapulnak. Kihívást jelentett még, a Zeta Model Checker adatszerkezete (ami fölött kellett működnie az algoritmusomnak) hiszen ez az adatszerkezet eddig csak szimbolikus modellellenőrzésre volt alkalmas, az explicit funkciók (mivel eddig nem volt rájuk szükség) hiányoztak, ezeket nekem kellett implementálni. Ilyen funkciók például egy adott tranzíció tüzelése, vagy az aktuális állapot lekérdezése, felhasználása, vagy az explicit állapotter kezelése. A feladat további nehézségeit az előző féléves gyors prototipizálási módszerem adta, a kevésbé átlátható kód miatt. Az előző félévek fókuszja az elméleti háttér és működés optimalizálása volt, ennek az lett a következménye, hogy a kód karbantarthatósága háttérbe szorult.

A félév során a projekt „Experimental” részlegében dolgoztam, és ott hoztam létre az osztályokat, melyeknek részletesebb dokumentációja külön olvasható. Az önálló labor 1. keretein belül kb. 1000 sor kódot írtam, ezek hat lényegesen eltérő osztályt tesznek ki, ezek közül egy (a diszjunkt normál formára alakító vizitor) nem a kísérleti részlegben, hanem a közösen használt logikai kifejezésfeldolgozó osztályok között található.

Az elkészült program a következő, jól elkülöníthető részekből épül fel:

- **Főprogram**
 - Feladatok indítása
 - Felhasználó értesítése az eredményről
- **Kifejezésfeldolgozó**
 - Boolean kifejezések negált és diszjunkt normál formára hozása
 - Részkifejezésekre bontás
- **LPN Checker**
 - explicit modellellenőrzés (útgenerálásra fókuszálva)

A félév eredménye egy jobban átlátható, a kutatócsoport többi munkájához könnyebben felhasználható, útgenerálásra optimalizált modellellenőrző program .NET platformra, ami a Zeta Model Checker adatszerkezetét használja. A projekt megtalálható az FTSRG csoport SVN tárolójában a projekt pontos neve: *ZetaModelChecker.Experimental.EloD*.