SZATURÁCIÓ ALAPÚ KORLÁTOS MODELLELLENŐRZÉSI TECHNIKÁK PETRI-HÁLÓK ANALÍZISÉRE

DARVAS Dániel

Abstract
Nowadays the formal verification of software and hardware systems is gaining an even more important role in system design. As the size of the systems grows, their verification becomes an increasingly complex task. This creates many new requirements for verification tools, also for model checking tools. In my work I have investigated the so-called *bounded model checking* techniques. In this paper I present a new saturation based bounded model checking tool. I describe an improved algorithm: the constrained saturation based bounded model checking algorithm, which I developed during my research.
Key words:
Petri Net, bounded model checking, saturation

Összefoglalás
Napjainkban a szoftver és hardver rendszerek formális verifikációja egyre nagyobb szerepet kap a rendszertervezésben. Ahogy az architektúrák mérete növekszik, az ellenőrzésük is egyre komplexebb feladatot jelent, ami új elvárásokat támaszt a verifikációs eszközökkel szemben, így a modellellenőrző eszközökkel szemben is. Munkám során a szaturációs algoritmuson alapuló modellellenőrzés továbbfejlesztését vizsgáltam *korlátos modellellenőrzési technikák* segítségével. E cikkben bemutatom hogyan lehetséges szaturáció alapú korlátos modellellenőrzést készíteni, illetve leírom ennek egy javított, kibővített változatát: a vezérelt szaturációval kiegészített korlátos modellellenőrzést.
Kulcsszavak:
Petri-háló, korlátos modellellenőrzés, szaturáció

1. Bevezetés

A rendszertervezés során a szoftver- és hardverrendszerek helyességének ellenőrzése, azaz *verifikációja* egyre nagyobb szerepet kap. Elég, ha egy atomerőmű irányítórendszerén futó szoftvert tekintünk: ennek hibás működése komoly károkhoz vezethet, így a tényleges használatba vétel előtt szükséges, hogy megbizonyosodjunk a helyes működéséről.

A verifikációval kapcsolatban alapvető elvárás, hogy teljes körű és matematikai precizitású legyen, ami formális módszerek alkalmazását igényli. Komplex rendszerek esetén a formális verifikáció igen számításigényes probléma. Bár a rendelkezésre álló számítási kapacitás növekszik, még mindig komoly kihívást jelent a nagyméretű, összetett architektúrák ellenőrzése.

A verifikáció folyamatában gyakran alkalmazott módszer a modellellenőrzés. Ennek során elkészítjük a rendszer egy modelljét – jelen esetben Petri-hálók segítségével, amely aszinkron rendszerek esetén egy elterjedt módszer –, majd felderítjük annak állapotterét. Utána az állapottérben a specifikációhoz

kötődő kritériumok – jelen esetben temporális logikai kifejezések segítségével megfogalmazott követelmények – teljesülését ellenőrizzük matematikai módszerek alkalmazásával.

2. Szaturációs modellellenőrzés

A modellellenőrzés megvalósítására számos algoritmus ismert, melyek más-más területen hatékonyak. A Petri-hálókkal gyakran modellezett aszinkron rendszerek esetén az egyik leghatékonyabbnak bizonyult ismert algoritmus a *szaturációs modellellenőrzés* algoritmusa [1].

A szaturációs modellellenőrzés két fő fázisra osztható:

1. A modell állapotterének (összes állapotának) felderítése. Ennek során a kezdőállapotból kiindulva az algoritmus végigjárja az összes olyan állapotot, amelyeket a modellezett rendszer elérhet.
2. Az ellenőrizendő kritériumok vizsgálata. Ennek során az algoritmus megvizsgálja, hogy a felderített állapottér alapján a megadott kritériumok a rendszerre teljesülnek-e vagy sem.

A szaturációs algoritmus hatékonyságát a következő tulajdonságai biztosítják:

- *Szimbolikus*, kompakt állapottér-reprezentációt használ, azaz a felderített állapotokat nem egyesével, explicit módon, hanem egy kódolt formában, az ún. *többértékű döntési diagramok* használatával.
- *Speciális iterációs stratégiát* alkalmaz, így nem szükséges ténylegesen a modell összes állapotát megvizsgálni, hanem kihasználható az aszinkron rendszerek azon tulajdonsága, hogy az egyes állapotok nem függenek szorosan össze. Ezért nem kell az összes kombinációt megvizsgálni a modellellenőrzés folyamán.

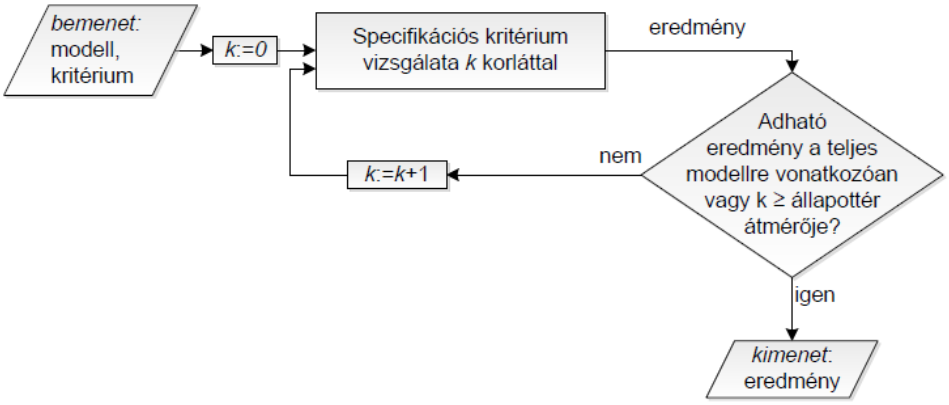
3. Korlátos modellellenőrzés

Bár a szaturáció előnyös tulajdonságai miatt gyors működésre képes, bizonyos modellkomplexitás felett már a klasszikus megközelítés sebessége nem kielégítő. Több lehetőség is van a szaturációs modellellenőrzés továbbfejlesztésére, amelyek közül az egyik a *korlátos modellellenőrzés* felhasználása.

Tételezzük fel, hogy egy rendszer helyességét kívánjuk eldönteni. Ennek egy lehetséges módszere az, hogy hibás állapotokat keresünk a modell állapotterében. A vizsgálatot addig kell folytatni, amíg nem találunk egy elérhető hibás állapotot, vagy be nem jártuk a teljes állapotteret. Ha a teljes állapotteret bejártuk és nem találtunk hibásnak tekintett állapotot, akkor a rendszer helyes, különben hibás.

Klasszikus szaturáció esetén először a teljes állapotteret fel kell deríteni, csak ezután lehet az ellenőrzést elvégezni. Abban az esetben viszont, ha már a teljes állapottér egy részében találhatunk hibás állapotot, a felderítést felesleges folytatni, a rendszer biztosan hibás. A korlátos modellellenőrzés ezt használja ki: először felderíti az állapottér egy kis részét, majd ezen elvégzi az ellenőrzéseket. Amennyiben nincs elegendő információja ahhoz, hogy eldönthető legyen a kritérium teljesülése vagy

meghiúsulása, akkor az állapottér egyre nagyobb részeit felderítve is elvégzi a vizsgálatot, amíg a kritérium kiértékelhető nem lesz. Ezt a folyamatot mutatja be az 1. ábra.



1. ábra. *Iteratív korlátos modellellenőrzési folyamat*

Korábban a szakirodalomban csak a teljes korlátos modellellenőrzési folyamat állapottér-felderítési fázisának elméleti megvalósításáról olvashattunk. Munkám során bemutattam [2], hogy az elméleti alapok átültethetők a gyakorlatba: a korlátos állapottér-felderítő algoritmus integrálható a már korábban elkészített, klasszikus szaturáción alapuló nemkorlátos kritériumellenőrzővel, és erre alapozva készíthető egy teljes szaturációs korlátos modellellenőrző. Ennek köszönhetően jelentős gyorsulást sikerült elérni olyan kritériumok ellenőrzése terén, amelyek az állapottér kis részének bejárása alapján kiértékelhetők.

4. Vezérelt korlátos szaturáción alapuló modellellenőrzés

Az előző fejezetben leírtak szerint elkészült az első működő korlátos szaturációs modellellenőrző, azonban bizonyos esetekben az egyszerű korlátos szaturációs algoritmus a vártnál gyengébb eredményeket ér el. A probléma vizsgálata rávilágított arra, hogy bár az állapotok egy adott halmazát nem szükséges felderíteni az eredmény meghatározásához, mégis a modellellenőrzés során felhasznált nemkorlátos klasszikus kritériumellenőrző mellékhatásként ezek egy részét is bejárja, megvizsgálja. Bár ez nem okoz hibás eredményt, az algoritmus teljesítményét jelentősen csökkentheti. Ez vezetett el az ún. *vezérelt szaturációs kritériumellenőrző* algoritmus [3] használatához. Az algoritmus korábban is ismert volt, azonban korlátos modellellenőrzésben először jelen munkám során került felhasználásra.

A vezérelt szaturációs algoritmus alapelve megegyezik a klasszikus szaturáció alapelvével, azonban egy más megvalósítást alkalmaz. A klasszikus szaturáció a *„lép és vág”* elvet követi, azaz a kritériumellenőrzés során minden lépést a lépés során létrejövő, de elhagyandó állapotok levágása követ. A vezérelt szaturáció ezzel szemben a *„vizsgál és lép”* elvet követi, azaz minden lépés előtt megvizsgálja, hogy kiléphetünk-e ezzel a lépéssel egy olyan állapothalmazba, amely a lépés után törlendő lenne, és csak akkor hajtja végre a lépést, ha vágásra nem lesz szükség. Mivel a vágáshoz

képest a vizsgálat jelentősen kisebb költségű művelet, ezért a vezérelt szaturáció minden esetben jobban teljesít, mint a klasszikus szaturáció, és többnyire sokkal hatékonyabb működésre képes.
Ennek köszönhetően a vezérelt szaturáció felhasználásával sikerült olyan új algoritmust készíteni, amely nem engedi, hogy a kritériumellenőrzés során olyan állapotok is vizsgálatra kerüljenek, amelyek az állapottér-felderítési fázisban nem lettek bejárva. Ez a klasszikus szaturációval csak hatalmas költséggel tehető meg, viszont a vezérelt szaturáció esetén az általa okozott többlet vizsgálati teher jelentősen kisebb, mint a kritériumellenőrzésben tapasztalható időnyereség.
Jelen cikkben a bemutatott fejlesztések mérésekkel történő alátámasztására terjedelmi korlátok miatt nincs lehetőség, azonban korábbi munkámban [4] részletes mérési eredmények olvashatók.

5. Összefoglaló

Munkám során sikerült elkészíteni az első korlátos modellellenőrző algoritmust szaturációs alapokon. A mérési eredmények alapján felismertem, hogy a hatékonyság tovább növelhető a vezérelt szaturációval történő kiegészítéssel. Ezzel megmutattam, hogy lehetséges szaturációs alapokon hatékony korlátos modellellenőrzőt készíteni, illetve létrehoztam integrált modellező és modellellenőrző eszközt is ezek alapján, amely a http://petridotnet.inf.mit.bme.hu oldalon érhető el.

Irodalom

[1] Zhao, Y., Ciardo, G.: *Symbolic CTL Model Checking of Asynchronous Systems using Constrained Saturation,* Springer-Verlag, Proceedings of the 7th International Symposium on Automated Technology for Verification and Analysis (ATVA'09), Berlin, 2009, pages 368–381.

[2] Vörös, A., Darvas, D., Bartha, T.: *Bounded Saturation Based CTL Model Checking,* Tallinn University of Technology, Institute of Cybernetics, Proceedings of the 12th Symposium on Programming Languages and Software Tools (SPLST'11), Tallinn, 2011, pages 149–160.

[3] Ciardo, G., Lüttgen, G., Siminiceanu, R.: *Saturation: an efficient iteration strategy for symbolic state space generation,* Springer-Verlag, Proceedings of Tools and Algorithms for the Construction and Analysis of Systems (TACAS), LNCS 2031, 2001, pages 328–342.

[4] Darvas, D., Jámbor, A.: *Komplex rendszerek modellezése és verifikációja*, TDK-dolgozat, Budapesti Műszaki és Gazdaságtudományi Egyetem, Villamosmérnöki és Informatikai Kar, Budapest, 2011.

Darvas Dániel, MSc hallgató

Budapesti Műszaki és Gazdaságtudományi Egyetem, VIK, Méréstechnika és Inf. Rendszerek Tanszék

Cím: H-1117 Magyarország, Budapest, Magyar tudósok körútja 2.

Telefon / Fax: +36-1-463-2057, +36-1-463-4112

E-mail: darvas.daniel@gmail.com

A dolgozat létrejöttét támogatta az MFB Magyar Fejlesztési Bank Zrt.