SZÍNEZETT PETRI-HÁLÓK SZATURÁCIÓS MODELLELLENŐRZÉSE KONJUNKTÍV DEKOMPOZÍCIÓ ALKALMAZÁSÁVAL

JÁMBOR Attila

Abstract
Nowadays, the verification of software and hardware systems is gaining an even more important role in system design. In my former work I have presented the so-called *saturation algorithm* to discover and store the state space efficiently. Although the amount of available computational resources is increasing, model checking of certain complex systems could not be efficiently accomplished with these methods so far. Such an important problem is that using Petri nets as a modeling language leads to huge and inexpressive models in many cases. My work mainly focuses on elaborating the saturation-based verification of coloured Petri nets. Using this new formalism the models can be constructed in a more compact and scalable way, therefore many new problems can be modeled and analyzed now.

Key words:
Petri Net, model checking, saturation, decomposition

Összefoglalás
A rendszertervezés során a szoftver- és hardverrendszerek helyességének ellenőrzése, azaz verifikációja egyre nagyobb szerepet kap. Korábbi munkáimban bemutattam, hogy az ún. *szaturációs algoritmus* használatával a verifikációhoz szükséges állapottér hatékonyan felderíthető és kompakt formában tárolható. Bár a rendelkezésre álló számítási kapacitás növekszik, összetett rendszerek ellenőrzése során felmerülnek olyan problémák, amelyekre az eddigi algoritmusok nem nyújtottak hatékony megoldást. Ilyen többek között, hogy bizonyos rendszerméret felett az egyszerű Petri-hálók már nem alkalmasak kompakt és szemléletes modellek készítésére.
Munkám során kidolgoztam a színezett Petri-hálók szaturációs vizsgálatát, amely formalizmus segítségével a modellek kompaktabb, skálázható formában készíthetők el, így sok olyan probléma is modellezhető és vizsgálható segítségükkel, amelyekre az egyszerű Petri-hálók nem nyújtottak megfelelő megoldást.

Kulcsszavak:
Petri-háló, modellellenőrzés, szaturáció, dekompozíció

1. Bevezetés

A színezett Petri-hálók az egyszerű Petri-hálók kiterjesztései abból a célból, hogy a meglévő modelleket kompaktabb módon lehessen ábrázolni, továbbá lehetővé váljon olyan rendszerek modellezése is, amelyekhez az egyszerű Petri-hálók nem nyújtanak megfelelő támogatást. Színezett hálók esetén a modellezés során különféle adatszerkezetek használhatók, amelyekkel paraméterezhető modellek készítésére is lehetőség nyílik.

Erre a megváltozott modellező nyelvre viszont nem alkalmazhatók a szaturáció eddig megismert változatai [2] az újonnan bevezetett eleme miatt.E területet érintő munkám során kidolgoztam a színezett Petri-hálók szaturációs ellenőrzésének elméletét.

2. Háttérismeretek

A Petri-háló a rendszermodellezés és rendszeranalízis egyik elterjedt modellezési eszköze [1]. Az egyszerű Petri-háló egy irányított, súlyozott, páros gráf, amelyben a két pontosztály elemei a helyek (P) és a tranzíciók (T). A gráfban egy irányított, súlyozott él egy tranzíciót köt össze egy hellyel vagy egy helyet egy tranzícióval. A Petri-háló állapotát helyeken lévő tokenek segítségével fejezzük ki. A háló tokeneloszlása (állapota) egy $M: P \rightarrow \mathbb{N}$ függvény, amely minden helyhez egy nemnegatív egész számot rendel. Ezek alapján a Petri-háló formálisan egy olyan $PN = (P, T, E, W, M_0)$ struktúra,ahol $P, T \text{ és } E$ rendre a helyek, tranzíciók és élek halmaza, W egy nemnegatív, élekhez tartozó súlyfüggvény, M_0 pedig a kezdeti tokeneloszlás.

A Petri-háló dinamikus viselkedését az egyes állapotokban értelmezett tüzelések határozzák meg. A tüzelések szabályai a következők: a) Egy $t \in T$ tranzíció engedélyezett, ha t-nek minden $p \in P$ bemenő helyén legalább $W(p,t)$ darab token van; b) egy engedélyezett tranzíció tetszése szerint tüzelhet vagy nem tüzelhet, tehát működése nem determinisztikus; c) egy engedélyezett t tranzíció tüzelése $W(p,t)$ darab tokent vesz el t minden p bemenő helyéről és $W(t,p')$ darab tokent helyez el a t tranzíció minden $p' \in P$ kimenő helyére.

A színezett Petri-háló formálisan egy $\text{CPN} = (P, T, E, \Sigma, C, G, AE, M_0)$ struktúra, ahol Σ a színosztályok halmaza, C egy színfüggvény, amely megadja az egyes helyeken érvényesnek tekintett tokenek típusát, G a tranzíciókhoz rendelt őrfeltételek halmaza, AE pedig az élekhez tartozó élkifejezések halmaza. Az őrfeltételek változókon értelmezett logikai kifejezések, amelyek a hozzájuk tartozó tranzíció engedélyezettségét befolyásolják. Egy t tranzíció engedélyezett, ha létezik a hozzá kapcsolódó e élekre írt $AE(e)$ élkifejezésekhez olyan változó-érték hozzárendelés, amely kielégíti a t tranzíció $G(t)$ őrfeltételét, valamint a tranzíció bemenő éleire írt élkifejezések által meghatározott tokenek a megfelelő bemeneti helyekről elvehetők és hasonlóan a kimenő élekre írt élkifejezések szerinti tokenek a kimeneti helyekre kirakhatók.

A verifikáció folyamatában gyakran alkalmazott módszer a *modellellenőrzés*. Ennek során a megtervezett rendszer modelljének először felderítjük az állapotterét, esetünkben a lehetséges tokeneloszlásokat, majd azon ellenőrizzük a specifikációs kritériumok teljesülését.

A modell elérhető állapotainak felderítésére számos módszer létezik. Legegyszerűbb esete az explicit állapottér-generálás, amikor kiindulunk egy kezdeti tokeneloszlásból, majd tüzelésekkel újabb állapotokat derítünk fel. Ezt mindaddig folytatjuk, amíg már egyik állapotból sem lehetséges tüzeléssel új állapotba eljutni. A 2000-es évek elején egy új módszert javasoltak a hatékony állapottér-generálásra: az ún. *szaturációs algoritmust* [2].

Az új algoritmus tulajdonságai:

- a lehetséges állapotok halmazát kódoltan tárolja döntési diagramok segítségével, ellentétben az explicit technikával, ahol az állapotokat egyenként tároljuk;
- a Petri-hálót dekomponáljuk, azaz részmodellekre bontjuk, a részmodellek lokális állapotait fogjuk szimbolikusan kódolni az állapottér bejárás során;
- az új állapotok felderítése egy speciális iterációs stratégia révén valósul meg. Ennek lényege, hogy egy speciális mélységi-jellegű bejárást hajtunk végre az állapottéren. A bejárás során az egyes tranzíciókat kimerítően tüzeljük el.

Az állapottér bejárása során a szaturációnak szüksége van az állapotátmenetek tárolására is. Egyszerű Petri-hálók esetén a mátrix alapú állapotátmenet-tárolás hatékony megoldást nyújt. Ezzel ellentétben a színezett Petri-hálóknál az állapotátmeneteket szimbolikusan, döntési diagramokat használva tudjuk eltárolni a bonyolult funkcionális függőségek miatt.

3. Konjunktív dekompozíció

Az állapottérfelderítés hatékonyságát az alkalmazott dekompozíció minősége erősen befolyásolja. Ha a bejárás során új állapotot derítünk fel, akkor abban az állapottérrészben, ahol azt megtaláltuk, az összes lehetséges új állapotátmenetet meg kell határozni. Ezért minél kisebb részekre tudjuk bontani a vizsgált Petri-háló állapotterét, annál hatékonyabb felderítést tudunk megvalósítani.

Ha a vizsgált modell állapotátmeneti függvényét $\mathcal{N}$-nel jelöljük, akkor a tranzíciók mentén történő diszjunktív felbontás a következő:

$$\mathcal{N} = \bigcup_{t \in T} \mathcal{N}_t \tag{1}$$

ahol $\mathcal{N}_t$ csak a t tranzíció tüzeléseinek hatását tárolja. A hatékonyság növelése érdekében a célunk, hogy az $\mathcal{N}_t$ relációkat további konjunktív részekre bontsuk fel [3]:

$$\mathcal{N}_t = \bigcap_{\forall i} \mathcal{N}_{t,i} \tag{2}$$

ahol i az egyes részek sorszáma.

Egyszerű Petri-hálók esetén a strukturális tulajdonságok lehetővé teszik, hogy a Petri-háló dekompozíciójával megegyezően alakítsuk ki az $\mathcal{N}_{t,i}$ részrelációkat. Színezett Petri-hálók esetén azonban funkcionális függőségek vannak jelen, amelyek megakadályozzák, hogy az előbbi módszerrel particionáljunk.

Munkám során kidolgoztam egy olyan módszert a színezett Petri-hálók állapotátmeneteinek kódolására, amely révén az eddigieknél hatékonyabb kezelés valósítható meg. A kidolgozott konjunktív particionálás a következő:

$$\mathcal{N}_t = \left(\bigcap_{\forall v} \mathcal{N}_{t,v} \right) \cap \mathcal{N}_{t,\ddot{o}} \tag{3}$$

A képletben szereplő v a t tranzícióhoz tartozó élváltozókat jelenti, $\mathcal{N}_{t,v}$ pedig az állapotátmenet-reláció t tranzícióhoz és v változóhoz tartozó részét. Ezek mellett további konjunktot hozok létre a tranzícióhoz tartozó őrfeltétel kényszereinek nyilvántartására. A részrelációkat szimbolikusan, döntési diagramok segítségével kódolom. Az eredeti $\mathcal{N}_t$ reláció visszaállításához a metszetképzést döntési diagram műveletek segítségével valósítom meg. A felbontás veszteségmentességét az biztosítja, hogy a döntési diagramokban a lehetséges állapotátmenetek mellett a tranzíció tüzeléséhez szükséges tokenváltozó-lekötéseket is eltárolom. A módszer biztosítja, hogy a v változók mentén finom dekompozíciót alkalmazhassunk, így egy új állapot felderítésekor csupán kis rész frissítése szükséges. A felbontás további előnye, hogy az őrfeltétel kényszereit kódoló $\mathcal{N}_{t,\text{ö}}$ reláció előre, offline kiszámolható, amely tovább gyorsítja a teljes állapottér-felderítés folyamatát.

4. Összefoglalás

A dolgozatban bemutattam a Petri-háló alapú modellek szaturációs modellellenőrzésének alapjait, valamint új módszert javasoltam a színezett Petri-hálók állapottér-relációjának kezelésére. A kidolgozott módszer az eddigieknél finomabb dekompozíciót tesz lehetővé, ezáltal a szaturációs algoritmus hatékonyan alkalmazható a színezett Petri-hálókra is. Megoldásom előnye, hogy lokálisan vizsgálhatóvá teszi a változólekötések helyekre vett hatásait, valamint segítségével a tranzíciók bizonyos kényszerei offline számolhatóak.

Irodalom

[1] Darvas Dániel, Jámbor Attila: *Komplex rendszerek modellezése és verifikációja*, Tudományos Diákköri Konferencia, Budapesti Műszaki és Gazdaságtudományi Egyetem, Villamosmérnöki és Informatikai Kar, 2011.

[2] Gianfranc oCiardo, Gerald Lüttgen, and Radu Siminiceanu: *Saturation: an efficient iteration strategy for symbolic state space generation,* Proc. Tools and Algorithms for the Construction and Analysis of Systems (TACAS), LNCS 2031, pages 328–342., Springer-Verlag, 2001.

[3] Gianfranco Ciardo and Andy Jinqing Yu: *Saturation-based symbolic reachability analysis using conjunctive and disjunctive partitioning,* CHARME'05, pages 146–161, 2005.

Jámbor Attila, MSc hallgató

Munkahely: Budapesti Műszaki és Gazdaságtudományi Egyetem, Villamosmérnöki és Informatikai Kar, Méréstechnika és Információs Rendszereke Tanszék

Cím: 1117 Budapest, Magyar Tudósok körútja 2.

E-mail: attila.jambor@gmx.com

A dolgozat létrejöttét támogatták: Magyar Fejlesztési Bank Zrt., Huawei Technologies Hungary Kft.