

Petri-gráfok alkalmazása véges automaták és protokollok modellezésére

PAPP GYÖRGY

Nemzetbiztonsági Hivatal

Összefoglalás

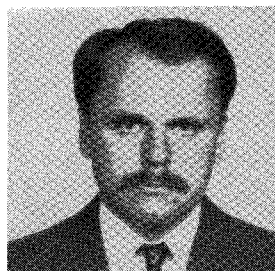
A protokollok implementálása, verifikálása és tesztelése során szükséges egy könnyen programozható, és modulárisan bővíthető matematikai modell alkalmazása. A protokollok leírására használt jelenlegi matematikai modellek nem rendelkeznek ezekkel a tulajdonságokkal. A nemzetközi szabványok például a protokollokat véges automatákkal definiálják. A cikk meg kívánja mutatni, hogyan transzformálható a véges automata modell a Petri-gráfos modellre, ami a fenti tulajdonságai miatt a protokollok egyik legalkalmasabb leíró modellje. Segítségével ugyanis egyszerűen valósítható meg a különböző protokoll funkciók kompozíciója.

Bármilyen rendszer vagy folyamat tervezése, vizsgálata, működésének tesztelése egy alkalmas modell megalkotásával és annak többszintű elemzésével valósul meg. A vizsgálat hatékonysága a modell optimális kiválasztásától függ. A modell létrehozásának elsődleges szempontjai, hogy hűen írják le a vizsgálandó rendszert, a vizsgálathoz felhasznált segédeszközök viszonylag egyszerűen kezelhetők, az adott rendszer paramétereinek bővüléseihez, változásaihoz jól igazodók legyenek. Ilyen megfontolásból a modellek általában matematikai modellek szoktak lenni, melyeknek az elemzése számítástechnikai segédeszközökkel valósítható meg.

A protokollok verifikálása, implementálása és tesztelése során sokszor okoz gondot, hogy kisebb mértékű változtatás a teljes protokoll újraértékelését teszi szükségessé. Ez nagy erő- és idő ráfordítást igényel. Igény van ezért olyan matematikai modell alkalmazására, amelyik egyszerűen kezelhető, könnyen gépesíthető és modulárisan bővíthető illetve szűkíthető.

A protokollok leírásában jelenleg használt matematikai modelleknek a hátránya, hogy azok nem rendelkeznek a moduláris bővíthetőség tulajdonságával és így nehezen verifikálhatók. A nemzetközi szabványok például a protokollokat véges automatákkal definiálják. A véges automata modell dinamikus vizsgálata azonban a moduláris bővíthetőség bonyolultsága miatt nehézkes. A protokollok sajátossága alapján viszont a bővíthetőségre az állandó igény szerinti igazodás érdekében szükség van.

Mégsem kell elvetni a régi matematikai modellt, az felhasználható újabb, az előbb felsorolt tulajdonságokkal rendelkező, alkalmasabb modellbe való átalakításához. Megmutatjuk miképpen valósítható meg a transzformálás. Példákon keresztül illusztráljuk, hogy az egymással kommunikálni tudó rendszerek protokoll-



PAPP GYÖRGY

1979-ben végzett a Budapesti Műszaki Egyetem villamosmérnöki Karán, majd a Bel

ügyminisztériumban kezdett dolgozni. Kutató-fejlesztő mérnökként elsősorban mikroprocesszoros vezérlő berendezések fejlesztésével és számítástechnikai rendszerek kidolgozásával foglalkozott. 1986-ban oklevelet szerzett a BME Távadattfeldolgozó Szakmérnöki Szakon. Azóta intenzív kutató munkát végez a számítógépes hálózatokba történő illegális beavatkozás felfedésének területén. Fő kutatási témája: a Petri háló elemzése és a védelmi modell megteremtése.

jainak lecserélésével hogyan alakul át a modell és így az elemzés tárgya. A Petri-gráfos modell segítségével egyszerűen megvalósítható a különböző típusú protokoll funkciók kompozíciója. Bizonyítjuk, hogy a protokollok egyik legalkalmasabb leíró modellje a Petri-gráf modell. A modell számítógépre viteléhez a Petri háló mátrix alakját használjuk fel. A protokoll modelljének megalkotása során bemutatjuk milyen lépésekkel jutunk el egy már ismert számítógép leíró matematikai modelltől, a véges automaták állapot-átmenet gráfjának transzformálása segítségével az egymással kommunikálni képes számítógépek Petri hálós leírásához.

A számítógépek összekapcsolása során az egymáshoz elküldött adatokat, utasításokat, kéréseket, válaszokat valamilyen kommunikációs folyamatban szabályozzák. Ezt a kommunikációs folyamatot *protokollnak* nevezzük [ISO-84]. A számítógépek egymással történő kommunikációja tehát a számítógépek kompozíciójával valósul meg. Így, ha a számítógép modelljét véges automataként értelmezzük, a protokoll nem más, mint a véges automaták olyan kompozíciója, ahol a véges automaták valamilyen tömegkiszolgálási soron, vagy hálózaton keresztül kommunikálnak egymással.

A számítógépek formális leírásának legalacsonyabb szintjén véges automatával (FSM – Finite State Machine) modellezzük a számítógépet. A véges automata [DEM-85] a nevet onnan kapta, hogy véges számú belső állapottal rendelkezik és a működése során automatikusan átkerül valamilyen külső folyamat hatására az egyik belső állapotból a másik belső állapotba. A véges automatát ezek alapján állapotgépnek is nevezik. A későbbiekben a modell elemeire hivatkozva gyakran használjuk az FSM előtagot, megkülönböz-

Beérkezett: 1990. III. 7. (H)

tetve, hogy véges automata elemről van szó. (Pl.: FSM-állapot.)

Leírásához öt változót definiálnak: $(Q, \Sigma, \Delta, \delta, \Gamma)$, ahol;

Q : az állapotok véges halmaza.

$Q = \{q_1, q_2, \dots, q_m\}$ m az állapotok száma.

Σ : egy véges bemeneti ábécé.

$\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$ n a bemenetek száma.

Δ : egy véges kimeneti ábécé.

$\Delta = \{\Delta_1, \Delta_2, \dots, \Delta_k\}$ k a kimenetek száma

δ : $Q \times \Sigma \rightarrow Q$ következő állapot leíró függvény, ami egy aktuális állapothoz és egy bemenethez rendeli a következő állapotot.

$\delta(q_i, \sigma_k) = \{q_j\}$ $i = 1 \dots m, k = 1 \dots n$

Γ : $Q \times \Sigma \rightarrow \Delta$ kimeneti függvény, ami egy állapothoz és egy bemenethez egyértelműen hozzárendel egy kimenetet.

$\Gamma(q_i, \sigma_k) = \{\Delta_l\}$ $i = 1 \dots m, k = 1 \dots n$

A véges automata segítségével protokollokat is le tudunk írni, azonban nem a legalkalmasabb modell implementálásra és verifikálására. A kommunikáló rendszerek tanulmányozása során eljutottak ahhoz a felismeréshez, hogy más fajta matematikai modellek segítségével lehet a protokollokat tervezni és elemezni. A modellek közül az egyik legalkalmasabbnak tűnő a C. Petri német matematikusról elnevezett Petri-gráf modell, amelyet az 1962-ben benyújtott doktori disszertációjában publikált először [PETRI-62]. A Petri-gráf alapfogalmait jó néhány szerző már ismertette, ezért ezekre nem térünk ki [NAGY-79]. A transzformáció miatt azonban szükség van a gráf matematikai leírására. A Petri-gráf modell elemeihez a jobb megkülönböztetés érdekében néha PN előtagot teszünk. (Pl.: PN-gráfpont, PN-átmenet.)

A Petri háló-leírása egy számnegyessel történik:

$C = (P, T, I, O)$,

ahol

P : a Petri háló gráfpontjainak halmaza.

$P = \{p_1, p_2, \dots, p_s\}$ s a gráfpontok száma.

T : az átmenetek halmaza.

$T = \{t_1, t_2, \dots, t_r\}$ r az átmenetek száma.

I : a bemenő függvény, ami leírja, melyik átmenetbe melyik gráfpontból van irányított összekötés.

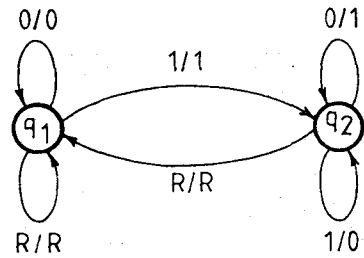
$I(t_i) = \{p_j, \dots, p_h\}$ ahol $i = 1 \dots r$.

O : a kimenő függvény, ami leírja, melyik átmenetből melyik gráfpontba van irányított összekötés.

$O(t_k) = \{p_b, \dots, p_g\}$ ahol $k = 1 \dots r$.

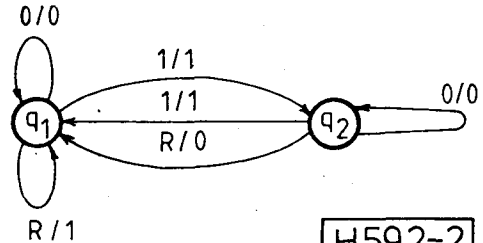
Véges automaták modellezése Petri hálóval

A véges automata leggyakoribb modellezése állapot-átmenet gráffal történik. Példaként nézzük meg a bináris szám kettes komplementének képzését megvalósító véges automata állapot-átmenet gráfját (1. ábra). A gráfban az állapotokat körökkel ábrázoljuk. Az átmenet irányított élén a q_i állapotból q_j állapotba törté-



H592-1

1. ábra. Bináris számok kettes komplementének kiszámítása véges automatán

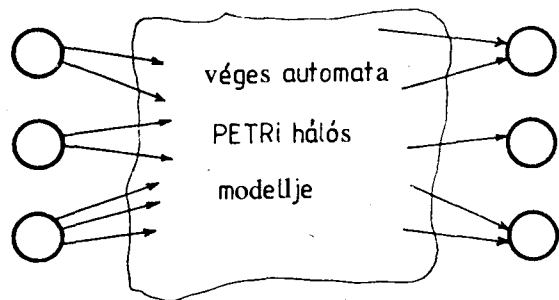


H592-2

2. ábra. Bináris számsorozat paritásának meghatározása véges automatán

bemenetek

kimenetek

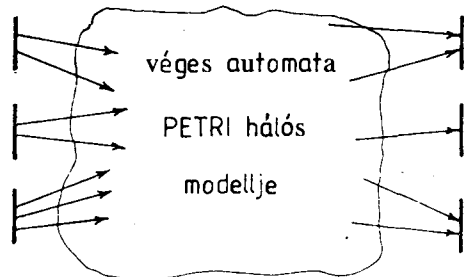


H592-3

3. ábra. A Petri háló külvilággal való kapcsolatának általános megközelítése gráfpontokkal

bemenetek

kimenetek



H592-4

4. ábra. A Petri háló külvilággal való kapcsolatának alternatív megközelítése gráfpontokkal

nő átmeneten szereplő σ_i/Δ_i jelentése, hogy a gép q_i állapotában az „ σ_i ” bemenet hatására „ Δ_i ” kimeneti válasszal a q_j állapotba jut.

Formálisan: $\delta(q_i, \sigma_i) = q_j$ és $\Gamma(q_i, \sigma_i) = \Delta_i$.

A külvilággal való bemenő kapcsolatokat a bemenő ábécé, a kimenő kapcsolatokat a kimeneti ábécé valósítja meg. A másik példában (2. ábra) a bemeneti ábécékre az állapotgép kiszámolja a paritást.

A véges automata modellje egyszerűen átalakítható a Petri hálós modellé [PET-81a]. Először modellezni kell a Petri háló külvilággal való kapcsolatát. A Petri háló általában zárt struktúra, a környezetétől elszigetelt. A külvilággal való kapcsolat ábrázolása két módon történhet. A bemeneteket és a kimeneteket vagy Petri gráfpontokkal (PN-gráfpont) vagy Petri átmenetekkel (PN-átmenet) ábrázolhatjuk (3. és 4. ábra). A két modell egymással egyenértékű. A továbbiakban mi az első típust fogjuk használni [PET-81b]. A transzformáció alap gondolata a Petri háló definíciójából következik [PET-81c]. Az állapot-átmenet gráfnak az összes feltétele a PN-gráfpontokat fogja meghatározni. Azaz a PN-gráfpontokat az FSM-bemenetek, az FSM-gráfpontok és az FSM-kimenetek uniója alkotja.

A PN-átmeneteket az állapot-átmenet gráf eseményei határozzák meg. Esemény pedig gyakorlatilag az egyik FSM-állapotból egy másik FSM-állapotba jutás pillanatában van. Ez az esemény egy PN-átmenettel azonosítható. A PN-átmenet bemeneteit az esemény előfeltételei fogják meghatározni. Az egyik előfeltétele az adott FSM-állapot, a másik előfeltétele pedig egy FSM-bemenet, mivel FSM-átmenet létrejöttéhez egy FSM-bemenet megjelenésére van szükség. A PN-átmenet kimeneteit pedig a véges automata következő állapot függvénye valamint a kimenet függvénye fogja meghatározni.

Ezek alapján megállapítható, hogy a transzformáció hatására a PN-átmenetek száma megegyezik az FSM-átmenetek, az FSM-állapotok és az FSM-kimenetek számainak összegével. A PN-átmenetek számát pedig az FSM-állapotok és az FSM-bemenetek számának szorzata fogja eredményezni, mert az átmeneteket az FSM-állapotok és az FSM-bemenetek Descartes szorzatából kapjuk.

Ezt formailag felírva:

$$(Q, \Sigma, \Delta, \delta, \Gamma) \rightarrow (P, T, I, O)$$

ahol a $(Q, \Sigma, \Delta, \delta, \Gamma)$ a véges automatát (FSM),
a (P, T, I, O) a Petri hálót (PN) írja le.

Q elemeinek száma: m P elemeinek száma: s
 Σ elemeinek száma: n T elemeinek száma: r
 Δ elemeinek száma: k

A jelöléseknél használt nyíl a transzformáció irányát jelzi.

A PN gráfpontjainak összessége az FSM-állapotok, az FSM-bemenetek és az FSM-kimenetek uniója lesz:
 $P = \Sigma \cup Q \cup \Delta$ ahol elemeinek száma: $s = m + n + k$.

A PN átmeneteit az FSM-állapotok és az FSM-bemenetek Descartes szorzata alkotja:

$$T = \{t_{q, \sigma} \mid q \in Q \text{ és } \sigma \in \Sigma\}$$

ahol elemeinek száma: $r = m * n$.

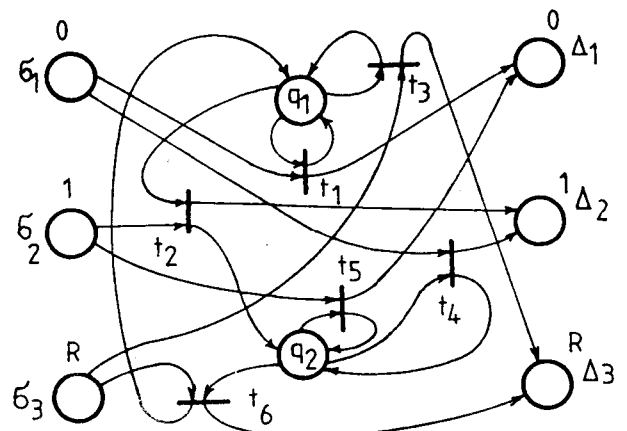
A PN bemenetei a rendre képzett PN-átmenetckhez az FSM-állapotok és az FSM-bemenetek lesznek:

$$I(t_{q, \sigma}) = \{q, \sigma\}.$$

A PN kimenetei a rendre képzett átmenetkből az FSM következő állapot és kimeneti függvényei lesznek:

$$O(t_{q, \sigma}) = \{\delta(q, \sigma), \Gamma(q, \sigma)\}.$$

A transzformációs szabály alkalmazásával így már elkészíthető a véges automata-példák Petri hálós modellje. Az első példa, aminek állapot-átmenet gráfja az 1. ábrán látható, egy olyan Petri gráfot eredményez, amely nyolc PN-gráfpontból és hat PN-átmenetből áll (5. ábra).

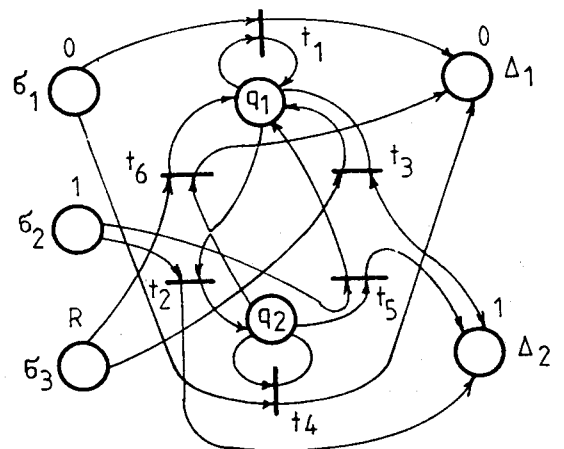


5. ábra. Az 1. ábra Petri hálós változata

H592-5

Hasonlóképpen a 2. ábrán szereplő gráf Petri gráf megfelelője is elkészíthető. Ezt a 6. ábra szemlélteti.

Első látásra szembeötlők a két modell-típus közötti különbség. Sokkal bonyolultabbnak látszik ugyanannak a problémának a megfogalmazása Petri modell-



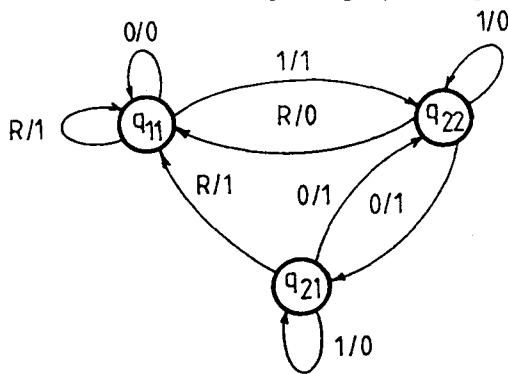
6. ábra. A 2. ábra Petri hálós változata

H592-6

ben, mint állapot-átmenet modellben. Vajon mégis miért előnyösebb a Petri modell?

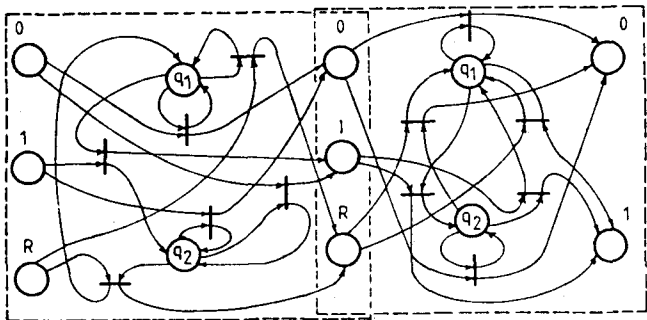
A válaszhoz vizsgáljunk meg egy összetettebb feladatot. Ne tegyünk mást, csupán ötvözzük a fenti két feladatot olyan formában, hogy a bemenő bináris számok kettes komplementének kiszámításán kívül végezzük el a paritás képzést is. Ehhez a feladathoz, ha elkészítjük az állapot-átmenet gráfot, gyakorlatilag előlről kezdjük el az FSM-állapotok létrehozását, és végül megkapunk egy látszólag független modellt (7. ábra).

Ezzel szemben sokkal egyszerűbb a Petri hálóból kiindulva elkészíteni az újabb igényt kielégítő Petri



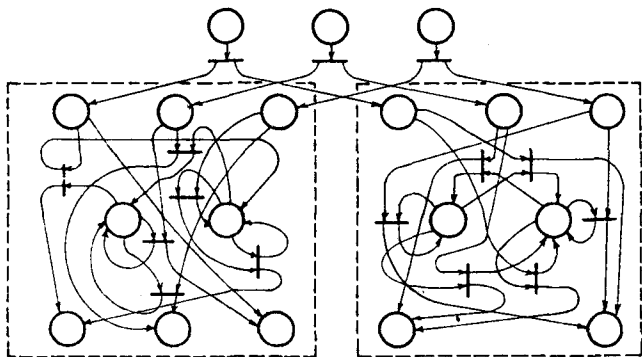
H592-7

7. ábra. A kettes komplement és a paritás egymás utáni megvalósítása véges automatán



H592-8

8. ábra. A kettes komplement és a paritás egymás utáni megvalósítása Petri hálón



H592-9

9. ábra. A kettes komplement és a paritás párhuzamos megvalósítása Petri hálón

modellt, sőt lehetőség van a kettes komplement és a paritás képzésének szekvenciális és párhuzamos megvalósítására is. Ezek figyelhetők meg a 8. és a 9. ábrákon. Bár a példák egyszerűek voltak, mégis látható belőle a Petri háló moduláris jellege, és az ebből származó előnyök. Ez azt jelenti, hogy a Petri modell vizsgálatához a feladatot bármilyen mélységben lehet tárgyalni, részfeladatokra osztani.

Ez a transzformációs szabály egyszerűen gépesíthető. A legegyszerűbb, ha a modellek mátrix alakban felírt formáit alkalmazzuk. A véges automata állapot-átmenet gráfja önmagában szemléletes, de nehezen gépesíthető. Alkalmassá viszont a táblázatban megadott forma. Két táblázat készíthető el (amit gyakran egy közös táblázatban szoktak ábrázolni), az egyik az állapot-átmeneti táblázat, a másik a kimeneti táblázat. Mindkét táblázatnak az oszlopaiba a véges automata állapotait, soraiba pedig a bemeneteit tüntetik fel. Az állapot-átmenet táblázat elemei úgy kerülnek kitöltésre, hogy az adott oszlophoz (FSM-állapot) tartozó sor (FSM-bemenet) keresztező helyére a következő FSM-állapotot írják. Ez gyakorlatilag azt mutatja meg, hogy egy adott FSM-állapotban egy adott bemeneti elem hatására melyik FSM-állapotba fog jutni az állapotgép. A kimeneti táblázat keresztező helyei pedig az előző mintájára a kimeneti elemeket fogják tartalmazni, vagyis adott FSM-állapotban kapott adott FSM-bemenet hatására milyen kimeneti elemet, milyen választ szolgáltat az állapotgép. A közös táblázatban a kitöltendő helyeken közösen szerepel a következő állapot és a kimenet egy "/" jel elválasztásával.

Példa egy közös állapot-átmenet és kimeneti táblázatra

Σ^Q	q_1	q_2	q_3	q_4	q_5
σ_1	q_1/Δ_1	q_3/Δ_2	q_3/Δ_2	q_1/Δ_5	q_1/Δ_6
σ_2	$q_2/-$	q_2/Δ_1	q_1/Δ_2	q_5/Δ_3	q_3/Δ_4
σ_3	q_4/Δ_6	q_1/Δ_2	q_5/Δ_4	q_3/Δ_3	q_5/Δ_3
σ_4	q_2/Δ_5	q_1/Δ_6	q_3/Δ_4	q_4/Δ_2	q_4/Δ_1

A táblázat első oszlopának és második sorának keresztező helyén szereplő "-" kimenet azt jelenti, hogy az állapotgép az állapot-átmenet során nem ad kimeneti választ.

A két táblázat számítógépre viteléhez képezzünk két mátrixot. Az első mátrix legyen az állapot-átmenet mátrix, az ún. Q gamma mátrix, melynek elemei legyenek egész számok- és értéként vegyék fel az állapot-átmenet táblázatban a megfelelő helyen szereplő következő állapotok indexeit. A második mátrix pedig legyen a kimeneti mátrix, Q delta mátrix, melyet hasonlóan, a kimeneti táblázat alapján képezzünk és elemei a kimeneti táblázatban az adott helyen lévő kimenetek indexei legyenek. Mindkét mátrix mérete, mint ahogy az könnyen belátható $m * n$.

Az állapot átmeneti mátrix és a kimeneti mátrix felépítése a következő:

Q gamma állapot átmeneti mátrix

	q ₁	q ₂	q _i	q _m
σ ₁				
σ ₂				
σ _j			*	
σ _n				

a * helyére a következő állapot sorszáma kerül

Q delta kimeneti mátrix

	q ₁	q ₂	q _i	q _m
σ ₁				
σ ₂				
σ _j			#	
σ _n				

a # helyére a kimenet sorszáma kerül

Most vizsgáljuk meg a Petri háló leírásának mátrix alakját [AND-83]. Defináljunk itt is két mátrixot. Az első legyen a kimeneti mátrix, amit D mínusz mátrixnak hívunk a későbbiekben. A másik mátrix pedig legyen a bemeneti mátrix, amit D plusz mátrixnak nevezünk. Mindkettő mátrix szerkezete ugyanolyan, oszlopai a Petri háló gráfpontjai legyenek, sorai pedig a Petri háló átmenetei. Első megközelítésben a következőt állítjuk. A D mínusz mátrix (D⁻) elemeit képezzük úgy, hogy az adott átmenet sorának mindazon oszlopaiba írjuk -1 értéket, amelyeknek oszlopához tartozó gráfpontokhoz gráféi érkeznek. A mátrix többi eleme legyen nulla. A D plusz mátrixot pedig úgy képezzük, hogy az adott átmenet sorába írjunk a +1 értéket, amelyik oszlophoz tartozó gráfpontból gráféi indul ki.

A többi mátrix elem itt is nulla legyen. Látható, miért nevezzük kimeneti, illetve bemeneti mátrixnak a D mínusz illetve D plusz mátrixot, ugyanis D mínusz mátrix a Petri háló kimeneti függvényét fogja leírni, a D plusz mátrix pedig a bemeneti függvényt.

A -1 és a +1 számérték itt azt jelenti, hogy egy adott átmenet és egy adott gráfpont között csupán egy gráféi létezik. Ez a modell az egyszerű Petri háló modellje. A kiterjesztett Petri háló modellben a gráfélek száma egynél több is lehet. Mivel azonban a dinamikus vizsgálathoz a kiterjesztett Petri háló egyszerűen átalakítható egyszerű Petri hálóbá [HACK-74], mi ezt a modellt fogjuk a továbbiakban alkalmazni.

A bemeneti mátrix és a kimeneti mátrix felépítése a következő:

D plusz bemeneti mátrix

	p ₁	p ₂	p _i	p _m
t ₁				
t ₂				
t _j			*	
t _r				

a * helyére +1 kerül ha a p_i állapotból gráféi indul t_j átmenetbe, egyébként értéke 0

D mínusz kimeneti mátrix

	p ₁	p ₂	p _i	p _m
t ₁				
t ₂				
t _j			#	
t _r				

a # helyére -1 kerül, ha a t_j átmenetből gráféi indul p_i állapotba, egyébként értéke 0

Az egész háló struktúráját a két mátrix összege fogja leírni, ami a későbbiekben a Petri háló viselkedésének számítógépes elemzéséhez alkalmas. A Petri háló dinamikus vizsgálata ennek a mátrixnak a felhasználásával történik, ami csak a háló felépítését, struktúráját jellemzi. A működést a háló töltéseinek áramlása, és annak folyamata fogja leírni. A töltések mozgása, a gyűjtások sorozata minden jellemző pillanatban meghatározható, amiket a modellben az ún. jelölések és a jelölés vektorok írnak le. [MUR-76] Ennek matematikai alapja kidolgozott, az elemzés a modell mátrix alakjának felhasználásával elvégezhető. A Petri háló dinamikus viselkedésének vizsgálatával ebben a cikkben nem foglalkozunk, csupán a Petri háló felépítésének számítógépre viteli lehetőségét elemezzük.

A transzformáció szabályai:

A PN-gráfpontokat úgy kapjuk, hogy rendre vesszük az FSM-bemeneteket, az FSM-állapotokat és az FSM-kimeneteket unióját:

$$P = \Sigma \cup Q \cup \Delta \quad \text{ahol } P \text{ száma: } s = n + m + k.$$

A PN-átmenetek előállításához képezzük az FSM-állapotok és az FSM-bemenetek Descartes szorzatát:

$$T = \{ t_{q, \sigma} \mid q \in Q \text{ és } \sigma \in \Sigma \} \quad \text{ahol } T \text{ száma} = n * m.$$

A PN bemeneti függvénye azon FSM-állapotok és FSM-bemenetek halmaza, amikből a Descartes szorzatokkal képezett PN-átmenetekbe gráféi érkeznek:

$$I(t_{q, \sigma}) = \{ q, \sigma \}.$$

A PN kimeneti függvénye azon FSM-állapot átmeneti függvény értékek és FSM-kimeneti függvény értékek halmaza, amikbe a Descartes szorzattal képezett PN-átmenetekből gráféi indul:

$$O(t_{q, \sigma}) = \{ \delta(q, \sigma), \Gamma(q, \sigma) \}.$$

Tekintsünk egy m állapotból, n bemenetből és k kimenetből álló véges automatát. Így a Q gamma és a O delta mátrix $m * n$ -es lesz.

Q gamma mátrix elemei egész számok, melyek a következő állapot sorszámaival egyeznek meg. Így értékkészlete q elemeinek száma, azaz $1 \dots m$.

O delta mátrix elemei szintén egész számok, de értékei az FSM kimeneteinek sorszámaival egyezik meg. Tehát értékkészlete Δ elemeinek száma, azaz $1 \dots k$.

A transzformáció alapján létre hozzuk a D plusz és a D mínusz mátrixot. A D plusz mátrix a PN bemeneti függvényét írja le. Oszlopai a PN gráfpontjai, sorai a PN átmenetei. Így $n + m + k$ számú oszlopból és $n * m$ számú sorból áll. Elemei mínusz előjellel szereplő annyi számérték, amennyi az adott sorban levő átmenethez tartozó, adott oszlopban szereplő gráfpontokból kiinduló éleinek száma. Tehát mind a D plusz, mind a D mínusz mátrix $s * r$ -es, ahol $s = n + m + k$ és $r = n * m$.

A mátrixot úgy alkotjuk meg a transzformáció alapján, hogy a PN gráfpontjai rendre az FSM bemenetei, az FSM állapotai és az FSM kimenetei legyenek: $P_1 \cdot P_n, P_{n+1} \cdot P_{n+m}, P_{n+m+1} \cdot P_{n+m+k}$. A PN átmeneteit pedig képezzük a Descartes szorzatnak megfelelően úgy, hogy elsőnek az FSM bemeneteit rendre rendeljük az FSM állapotaihoz: $t_1 = \{\sigma_1, q_1\}, t_2 = \{\sigma_2, q_1\}, \dots, t_n = \{\sigma_n, q_1\}, t_{n+1} = \{\sigma_1, q_2\}, t_{n+2} = \{\sigma_2, q_2\}, \dots, t_{i,n+1} = \{\sigma_1, q_i\}, \dots, t_{n,m} = \{\sigma_n, q_m\}$.
 A D plusz mátrix sor és oszlop képzése teljesen hasonlóan történik.

A D mínusz mátrix elemeit meghatározó algoritmus:

(a mátrix sor indexe - i, oszlop indexe - j)

$$D^- [i, j] = \begin{cases} -1 & i = e * n + j \text{ esetén, ahol} \\ & e \in (0..m-1), j \in (1..n) \\ & \text{és } i = e * n + f; \\ & j = e + n + 1 \text{ esetén, ahol} \\ & e \in (0..m-1), f \in (1..n) \\ 0 & \text{egyébként} \end{cases}$$

formálisan: $d^-(t_{\sigma_1, q_j}, \sigma_1 \text{ vagy } q_j) = -1$ egyébként 0

A D plusz mátrix elemeit meghatározó algoritmus:

$$D^+ [i, j] = \begin{cases} +1 & i = (e-1) * n + f; \\ & j = n + \delta(\zeta, \epsilon) \text{ ahol} \\ & e \in (1..m), f \in (1..n) \\ & \text{és } i = (e-1) * n + f; \\ & j = n + m + \Gamma(f, e), \text{ ahol} \\ & e \in (1..m), f \in (1..n) \\ 0 & \text{egyébként} \end{cases}$$

formálisan: $d^+(t_{\sigma_1, q_j}, \delta(\sigma_i, q_j) \text{ vagy } \Gamma(\sigma_i, q_j)) = 1$ egyébként 0

Protokoll modellezése és transzformálása

Ezek után tekintsünk egy transzformációs példát. Példánkban vegyünk egy egyszerű protokollnak az állapot-átmenet gráfját (10. ábra). Az ábrán feltüntetjük a véges automata bemeneti- és kimeneti ábécé elemeit, valamint az egyszerű protokoll állapotait. A 11. ábrán pedig a protokoll állapot-átmenet táblázata figyelhető meg.

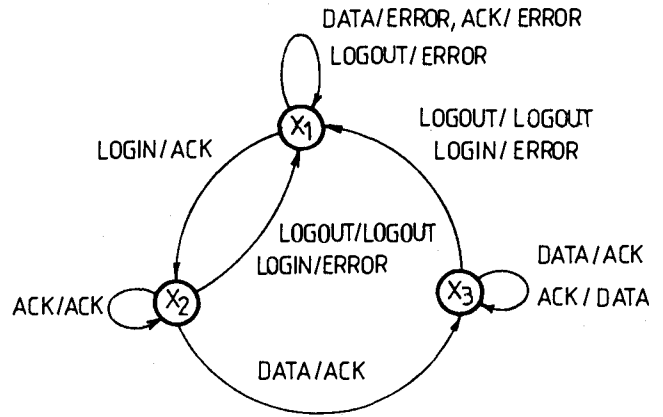
Az egyszerű protokoll FSM változói:

Az FSM-állapotok:

- $q_1 = \text{off}$
- $q_2 = \text{wait}$
- $q_3 = \text{data_transfer}$

Az FSM-bemenetek:

- $\sigma_1 = \text{login}$
- $\sigma_2 = \text{data}$
- $\sigma_3 = \text{ack}$
- $\sigma_4 = \text{logout}$



Bemeneti ábécé: LOGIN, DATA, ACK, LOGOUT

Kimeneti ábécé: DATA, ACK, LOGOUT, ERROR

Állapotok halmaza: OFF, WAIT, DATA, TRANSFER

H592-10

10. ábra. Egy egyszerű protokoll véges automatán

State \ Input	LOGIN	DATA	ACK	LOGOUT
OFF X1	X2 / ACK	X1 / ERROR	X1 / ERROR	X1 / ERROR
WAIT X2	X1 / ERROR	X3 / ACK	X2 / ACK	X1 / LOGOUT
DATA TRANSFER X3	X1 / ERROR	X3 / ACK	X3 / DATA	X1 / LOGOUT

H592-11

11. ábra. A kommunikáló véges automaták állapot-átmenet táblázata

Az FSM-kimenetek:

- $\Delta_1 = \text{ack}$
- $\Delta_2 = \text{error}$
- $\Delta_3 = \text{logout}$
- $\Delta_4 = \text{data}$

az FSM állapot-átmeneti mátrixa (Q delta):

	q_1	q_2	q_3
σ_1	2	1	1
σ_2	1	3	3
σ_3	1	2	3
σ_4	1	1	1

a mátrix belsejében levő számok q azonosító indexei

az FSM kimeneti mátrixa (Q gamma):

	q_1	q_2	q_3
σ_1	1	2	2
σ_2	2	1	1
σ_3	2	1	4
σ_4	2	3	3

a mátrix belsejében levő számok Δ azonosító indexei

A transzformáció után PN mátrixai:

A D plusz, PN-bemeneti mátrix:

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
t_1	0	0	0	0	0	1	0	1	0	0	0
t_2	0	0	0	0	1	0	0	0	1	0	0
t_3	0	0	0	0	1	0	0	0	1	0	0
t_4	0	0	0	0	1	0	0	0	1	0	0
t_5	0	0	0	0	1	0	0	0	1	0	0
$D^+ = t_6$	0	0	0	0	0	0	1	1	0	0	0
t_7	0	0	0	0	0	1	0	1	0	0	0
t_8	0	0	0	0	1	0	0	0	0	1	0
t_9	0	0	0	0	1	0	0	0	1	0	0
t_{10}	0	0	0	0	0	0	1	1	0	0	0
t_{11}	0	0	0	0	0	0	1	0	0	0	1
t_{12}	0	0	0	0	1	0	0	0	0	1	0

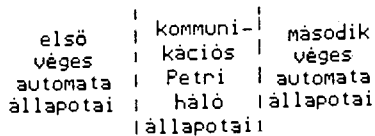
A D mínusz, PN-kimeneti mátrix:

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
t_1	-1	0	0	0	-1	0	0	0	0	0	0
t_2	0	-1	0	0	-1	0	0	0	0	0	0
t_3	0	0	-1	0	-1	0	0	0	0	0	0
t_4	0	0	0	-1	-1	0	0	0	0	0	0
t_5	-1	0	0	0	0	-1	0	0	0	0	0
$D^- = t_6$	0	-1	0	0	0	-1	0	0	0	0	0
t_7	0	0	-1	0	0	-1	0	0	0	0	0
t_8	0	0	0	-1	0	-1	0	0	0	0	0
t_9	-1	0	0	0	0	0	-1	0	0	0	0
t_{10}	0	-1	0	0	0	0	-1	0	0	0	0
t_{11}	0	0	-1	0	0	0	-1	0	0	0	0
t_{12}	0	0	0	-1	0	0	-1	0	0	0	0

A D mátrixból látszik, szerkezetük élesen elhatárolható, kilenc almátrixra bontható. Az almátrixok oszlop határait az FSM-bemeneteknek megfelelő p_1, p_2, p_3 és p_4 PN-gráfpontok, az FSM-állapotoknak megfelelő p_5, p_6 és p_7 PN-gráfpontok, valamint az FSM-kimeneteknek megfelelő p_8, p_9, p_{10} és p_{11} PN-gráfpontok húzzák meg. Az almátrixok sor határait pedig az első FSM-állapothoz rendelt FSM-bemenetek alapján létrejött t_1, t_2, t_3 és t_4 PN-átmenetek, a második FSM-állapothoz rendelt FSM-bemenetek alapján létrejött p_5, p_6, p_7 és p_8 PN-átmenetek, valamint a harmadik FSM-állapothoz rendelt FSM-bemenetek alapján kialakult p_9, p_{10}, p_{11} és p_{12} PN-átmenetek határozzák meg. A D mínusz mátrixban így egységes almátrixok és sormátrixok alakulnak ki, gyakorlatilag teljesen függetlenül a modell felépítésétől. A modell leírásában a D plusz mátrix vesz részt. Ez a megállapítás a későbbiekben használható

Kommunikáló véges automaták modellezése

A protokoll két véges automata kompozíciója, amelyek valamilyen tömegkiszolgálási soron keresztül kommunikálnak. (Általában a számítógép-hálózat így modellezhető – 13. ábra). Ez azt jelenti, hogy a két



$$D = \begin{bmatrix} [PN1] & [0] & [0] \\ [A] & [C] & [B] \\ [0] & [0] & [PN2] \end{bmatrix}$$

első véges automata
 átmenetei
 kommunikációs Petri
 háló átmenetei
 második véges
 automata átmenetei

H592-12

12. ábra. A kommunikáló véges automaták Petri háló mátrixának felépítése, almátrixai

véges automata összekapcsolása úgy valósul meg, hogy az egyik automata egy tömegkiszolgálási soron keresztül adja át adatait és nyugtáit a másik automatának és fordítva. A protokoll leírása ennek a tömegkiszolgálási folyamatnak a szabályait tartalmazza.

Az egymással ilyenformán kommunikáló automaták modellezése könnyen megvalósítható Petri háló segítségével, ugyanis ahogy ezt már láttuk a bináriszám kettes komplementjét eredményező Petri modell és a paritást képző Petri modell példák kompozícióján, a független modellek egyszerűen illeszthetők össze. Az egyik modell kimenete megfelel a másik modell bemenetének és fordítva. A Petri hálós modell az ilyen illesztés elvégzésére kiválóan alkalmas. Ebből az is következik, hogy nagyon egyszerűen lehet módosítani a protokollt. De ugyanilyen egyszerűen lehet módosítani a modellt újabb protokoll funkció bevezetésekor.

A kommunikáló véges automaták ebben a modellben három független modell alapján jönnek létre. A három modell közül kettő a két véges automata modellje, a harmadik pedig a tömegkiszolgálást végző egység modellje. A teljes rendszer Petri hálójának megvalósításához lépésenként hozzáláthatunk. Egyrészt a független modellek Petri hálós modelljeinek képzésével, másrészt a külön Petri modellek összekapcsolásával.

Korábban azt mutattuk be, hogy a véges automata állapot-átmenet struktúrája hogyan transzformálható Petri hálóba. Most megmutatjuk, hogy az így külön-külön transzformált és létrejött Petri hálók hogyan egyesíthetők (12. ábra).

A véges automata Petri hálós transzformációjának mátrix alakjában – mint ahogy már említettük – a mátrix oszlopait a véges automata bemenetei, állapotai és kimenetei alkotják. Az egyik kommunikáló véges automata meghatározható számú kimenetét átadja a másik véges automatának és fordítva.

A Petri hálóban azonban két PN-gráfpont közvetlenül nem kapcsolódhat össze, ezért az összetett modell mindenképpen kiegészül újabb PN-gráfpontokkal és PN-átmenetekkel. Ez a gyakorlatban annyit jelent, hogy a D^- és a D^+ mátrixok megalkotásában a követ-

kezőképpen néz ki a mátrixok almátrixokra bontott struktúrája:

$$T.. \rightarrow \begin{matrix} & P.. \\ & \downarrow \\ \begin{bmatrix} PN_1 \\ \underline{A} \\ \underline{0} \end{bmatrix} & \begin{bmatrix} \underline{0} \\ \underline{C} \\ \underline{0} \end{bmatrix} & \begin{bmatrix} \underline{0} \\ \underline{B} \\ PN_2 \end{bmatrix} \end{matrix}$$

- ahol $[PN_1]$ az első véges automata Petri almátrixa,
 $[PN_2]$ a második véges automata Petri almátrixa,
 $P..$ a kommunikáció PN-gráfpontjai,
 $T..$ a kommunikáció PN-átmenetei és
 $[\underline{0}]$ 0 elemű mátrix.
 $[\underline{A}]$ a PN_1 -el kapcsolatot tartó kommunikációs almátrix
 $[\underline{C}]$ a kommunikációt leíró PN almátrix
 $[\underline{B}]$ a PN_2 -el kapcsolatot tartó kommunikációs almátrix

Látható, hogy hiányzó almátrix (\underline{A} , \underline{B} , \underline{C}) elem csupán a kommunikáció PN-átmeneteinek soraiban van, azaz csak a kommunikáció PN-átmenetei kapcsolódnak a véges automaták bemeneti és kimeneti PN-gráfpontjaihoz. Ez természetesen abból következik, hogy a transzformáció során a véges automata bemeneteit és kimeneteit PN-gráfpontokkal modelleztük.

A továbbiakban tekintsünk meg néhány példát az ilyen mátrix létrehozására konkrét protokollok modellezésével.

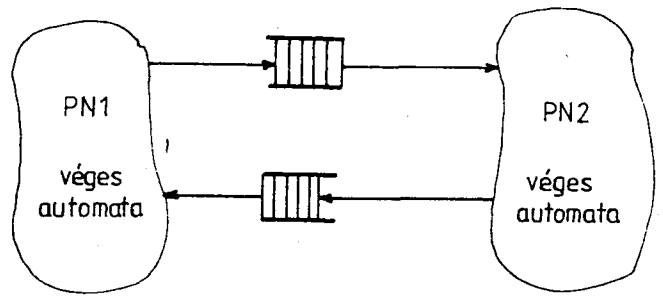
Először vizsgáljunk meg egy egyszerű protokollt a kommunikáló véges automatákkal. Majd a protokoll, a tömegkiszolgáló sor legyen egy FIFO sor. Az újabb példa pedig szemafor rendszeres protokoll legyen. Mindezek folyamatosan bemutatják, hogy ezzel a lépésenkénti megvalósítással egyre bonyolultabb és bonyolultabb protokoll funkciók is adaptálhatók.

Két véges automata egyszerű kommunikációja

A legegyszerűbb esetben a protokoll (14. ábra) csupán abból áll, hogy az egyik véges automata kimeneti eleme - Δ_k^1 , a t_1 jelű PN-átmeneten keresztül kapcsolódik a másik véges automata σ_j^2 jelű bemeneti eleméhez, illetve a második véges automata Δ_n^2 jelű kimeneti eleme a t_2 jelű PN-átmenettel kötődik az első automata σ_i^1 bemeneti eleméhez.

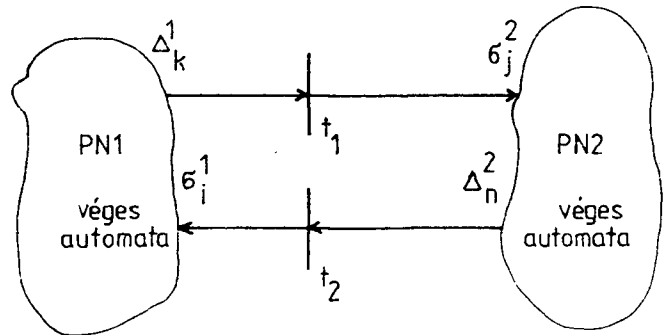
Hogyan írható fel az eredő D plusz és D mínusz mátrix? Ahogy azt már az előbb bemutatottuk, elegendő csak az \underline{A} , a \underline{B} , és a \underline{C} almátrixokat meghatározni, mert a többi almátrix vagy a transzformációból nyerhető vagy nulla elemű:

$$\underline{D} = \begin{bmatrix} [PN_1] & [\underline{0}] & [\underline{0}] \\ [\underline{A}] & [\underline{C}] & [\underline{B}] \\ [\underline{0}] & [\underline{0}] & [PN_2] \end{bmatrix}$$



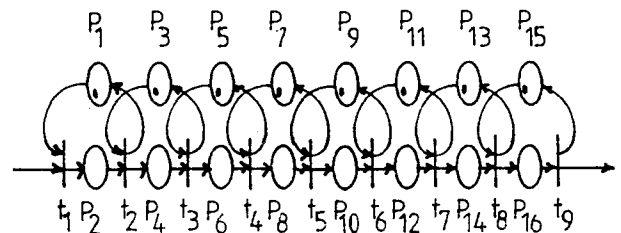
H592-13

13. ábra. A kommunikáló véges automaták, protokoll értelmezése



H592-14

14. ábra. Két véges automata egyszerű kommunikációjának Petri hálós modellje



15. ábra. FIFO sor Petri hálós modellje

H592-15

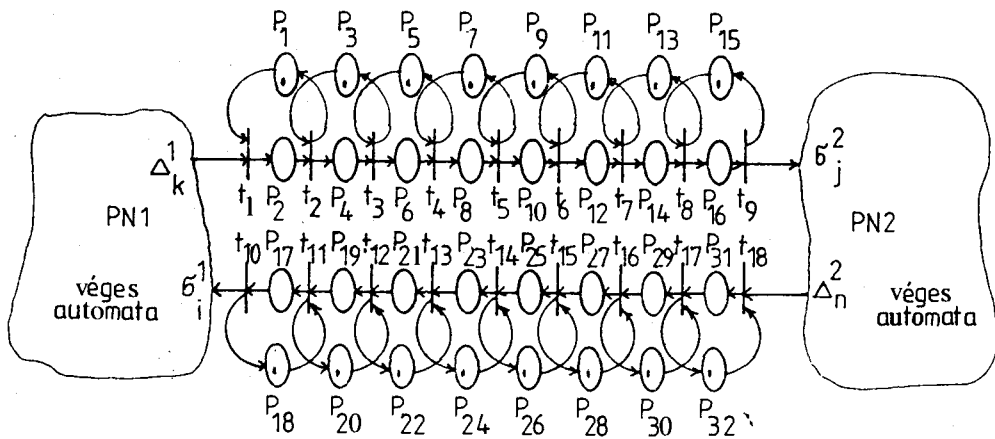
A \underline{C} almátrix üres, mert a kommunikációs rész nem tartalmaz egyetlen gráfpontot sem. Így természetesen a középső oszlop almátrixai sem léteznek. Az ábra jelöléseit alkalmazva az \underline{A} és a \underline{B} mátrixok a következőképpen néznek ki:

Az \underline{A}^+ és \underline{A}^- almátrix:

$$\underline{A}^+ = \begin{bmatrix} \cdot \sigma_i^1 \cdots \Delta_k^1 \\ t_1 \cdot 1 \cdots 0 \\ t_2 \cdot 0 \cdots 0 \end{bmatrix}; \quad \underline{A}^- = \begin{bmatrix} \cdot \sigma_j^2 \cdots \Delta_n^2 \\ t_1 \cdot 0 \cdots 0 \\ t_2 \cdot 0 \cdots -1 \end{bmatrix};$$

A \underline{B}^+ és \underline{B}^- almátrix:

$$\underline{B}^+ = \begin{bmatrix} \cdot \sigma_j^2 \cdots \Delta_n^2 \\ t_1 \cdot 0 \cdots 0 \\ t_2 \cdot 1 \cdots 0 \end{bmatrix}; \quad \underline{B}^- = \begin{bmatrix} \cdot \sigma_i^1 \cdots \Delta_k^1 \\ t_1 \cdot 0 \cdots -1 \\ t_2 \cdot 0 \cdots 0 \end{bmatrix};$$



16. ábra. Két véges automata FIFO regiszteren keresztüli kommunikációjának Petri hálós modellje

Az \underline{A} almátrixok:

$$\underline{A}^+ = \begin{array}{c|ccc} & \sigma_i^1 & \Delta_k^1 & \\ \hline t_1 & \dots & 0 & \dots & 0 & \dots \\ \vdots & & \vdots & & \vdots & \\ t_{10} & \dots & 1 & \dots & 0 & \dots \\ \hline \end{array} ; \underline{A}^- = \begin{array}{c|ccc} & \sigma_j^2 & \Delta_n^2 & \\ \hline t_1 & \dots & 0 & \dots & -1 & \dots \\ \vdots & & \vdots & & \vdots & \\ t_{10} & \dots & 0 & \dots & 0 & \dots \\ \hline \end{array}$$

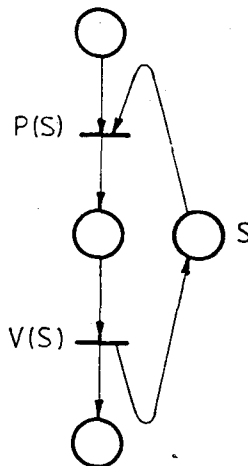
A \underline{B} almátrixok:

$$\underline{B}^+ = \begin{array}{c|ccc} & \sigma_j^2 & \Delta_n^2 & \\ \hline t_9 & \dots & 1 & \dots & 0 & \dots \\ \vdots & & \vdots & & \vdots & \\ t_{18} & \dots & 0 & \dots & 0 & \dots \\ \hline \end{array} ; \underline{B}^- = \begin{array}{c|ccc} & \sigma_j^2 & \Delta_n^2 & \\ \hline t_9 & \dots & 0 & \dots & 0 & \dots \\ \vdots & & \vdots & & \vdots & \\ t_{18} & \dots & 0 & \dots & -1 & \dots \\ \hline \end{array}$$

Két véges automata szemaforokkal való összekapcsolásának modellezése

A szemafor rendszer egy olyan szinkronizálási mechanizmus, ahol egy irányító egység szabályozza az adat és nyugtázás forgalmat. A szemafor rendszerben két művelet, a P és V művelet, valamint egy szemafor lé-

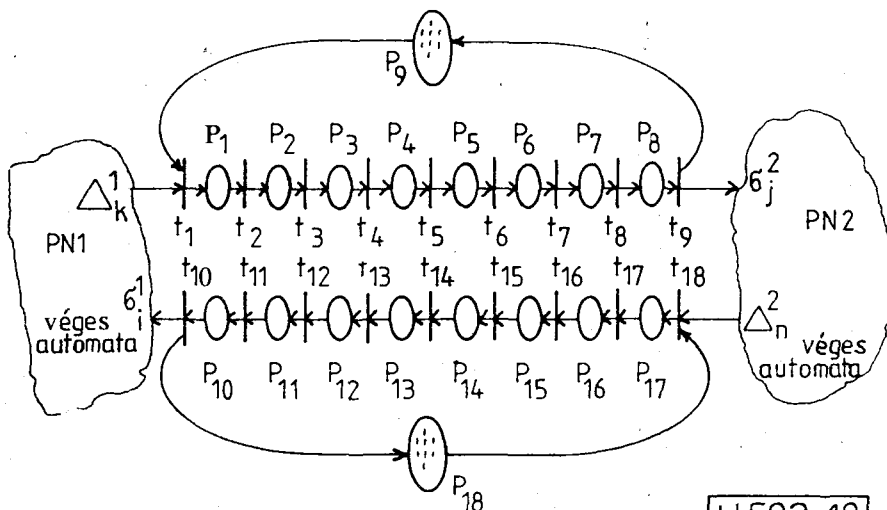
H592-16



H592-17

17. ábra. P és V szemafor rendszer Petri hálós modellje

tezik, ami nem negatív egész értéket vehet fel. A V művelet eggyel növeli, a P művelet eggyel csökkenti ennek a számnak az értékét. P művelet akkor követ-



H592-18

18. ábra. Két véges automata szemaforos összekapcsolásának Petri hálós modellje

kezhet be, ha a szemafor értéke pozitív. Ha a szemafor értéke nulla, a P műveletnek mindaddig várnia kell, amíg nem következett be egy V művelet. A szemafor értékét pedig semmi más nem módosíthatja. A 17. ábrán bemutatjuk ennek Petri hálós modelljét. A szemafor az S jelű gráfpont, aminek értéke a benne lévő töltések számával egyezik meg, a P és V művelet pedig egy-egy átmenet.

Ha a tömeg kiszorgálást regisztersorral valósítjuk meg, akkor a szemaforos szinkronizálás egy lehetséges módja a 18. ábrának megfelelően realizálható. A nyolc regiszteres megoldásnál az egyik szemafor a p_9 -dik gráfpont, a másik a p_{18} -dik, melyeknek pontosan nyolc kezdeti töltéssel kell rendelkezni, amennyi a regiszterek száma. A P műveleteket a t_1 és a t_{18} átmenetek a V műveleteket pedig a t_9 és a t_{10} átmenetek jelzik. A kommunikációs Petri háló így 18x18-as almátrixot eredményez. A véges automatákhoz való kapcsolódás pedig hasonlóan az előbbi példákhoz csak két-két átmenetnél valósul meg: t_1 , t_9 , t_{10} , és t_{18} .

Az almátrixok tehát a következő módon néznek ki:

A \underline{C}^+ almátrix:

$$\underline{C}^+ = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

A \underline{C}^- almátrix:

$$\underline{C}^- = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

Az \underline{A} almátrixok:

$$\underline{A}^+ = \begin{pmatrix} \sigma_i^1 & \Delta_k^1 \\ \vdots & \vdots \\ t_1 & \vdots \\ \vdots & \vdots \\ t_{10} & \vdots \\ \vdots & \vdots \end{pmatrix}; \underline{A}^- = \begin{pmatrix} \sigma_i^1 & \Delta_k^1 \\ \vdots & \vdots \\ t_1 & \vdots \\ \vdots & \vdots \\ t_{10} & \vdots \\ \vdots & \vdots \end{pmatrix};$$

A \underline{B} almátrixok:

$$\underline{B}^+ = \begin{pmatrix} \sigma_j^2 & \Delta_n^2 \\ \vdots & \vdots \\ t_9 & \vdots \\ \vdots & \vdots \\ t_{18} & \vdots \end{pmatrix}; \underline{B}^- = \begin{pmatrix} \sigma_i^2 & \Delta_n^2 \\ \vdots & \vdots \\ t_9 & \vdots \\ \vdots & \vdots \\ t_{18} & \vdots \end{pmatrix};$$

A kutatás további célja, hogy elemzési módszereket dolgozzunk ki az ilyen formában számítógépre vitt protokollok dinamikai vizsgálatához, a Petri hálós modell gyújtási sorozatainak hatékony elemzéséhez. Kibővíve a kiterjesztett Petri hálós modellek vizsgálatával a numerikus és az idő Petri hálók analizálhatók. Ezen kívül célja, hogy további lényeges protokoll funkciók moduláris beépítéséhez, újabb előre nem látható problémák értékeléséhez dolgozzon ki eljárásokat. Ne csak kizárólag a mátrix algebra alkalmazásával teremtsen meg ezeket a vizsgálatokat, hanem új, alkalmas programozási nyelvek keresésével, felhasználásával.

Protokollok segítségével, azok Petri hálós modelljeinek elemzése alkalmas teremt más kommunikáló rendszer modellezésére is, például a számítógépes hálózat modelljének vizsgálatára. Így alkalom kínálkozik a hierarchikus rendszerek Petri modelljének megalkotásával és annak vizsgálatával a számítógépes hálózatok biztonsági és védelmi tulajdonságainak elemzésére is.

IRODALOM

- [AGE-79] T. Agerwala: „Putting Petri Nets to Work”, Computer Vol. 12, No. 12. December 1979. pp. 85-94.
- [AND-83] B. Andrásfai: The Theory of Graphs, Flows and Matrices. Akadémia Kiadó, Budapest, 1983.
- [DEM-85] Demetronics J., Jordan D., Radiszlav P.: A számítástudomány matematikai alapjai, Tankönyvkiadó, Budapest, 1985. pp. 170-181.
- [HACK-74] M. Hack: „Decision Problems for Petri Nets and Vector Addition Systems”, Computation Structures Group Memo 95, Project MAC., March 1974. pp. 79
- [ISO-84] International Standard Information Processing Systems: Open System Interconnection - Basic Reference Model, ISO 7498, 1984.
- [MUR-76] T. Murata: „A Method for Synthesizing Marked Graphs from Given Markings”, Proceedings of the Tenth Annual Asilomar Conference on Circuits, Systems and Computers, November, 1976. pp. 202-206.
- [NAGY-79] Dr. Nagy Á.: „Modellezés Petri-gráfokkal”, Informatió-Elektronika, 1979/3. pp. 158-161.
- [PETRI-62] C. Petri: „Kommunikation mit Automaten”, Ph. D. Thesis, University of Bonn, 1962.
- [PET-77] J. L. Peterson: „Petri Nets”, Ass. Comput. Mach. Computing Surveys, Vol. 9, Sept. 1977. pp. 223-251.
- [PET-81a] J. L. Peterson: „Petri Net Theory and the Modeling of Systems”, Prentice-Hall, Inc., Englewood Cliffs, N. J. 1981. pp. 43-44.
- [PET-81b] J. L. Peterson: „Petri Net Theory and the Modeling of Systems”, Prentice-Hall, Inc., Englewood Cliffs, N. J. 1981. pp. 42.
- [PET-81c] J. L. Peterson: „Petri Net Theory and the Modeling of Systems”, Prentice-Hall, Inc., Englewood Cliffs, N. J. 1981. pp. 44-45.