Elérhetőségi probléma egyszerűsítése: Állapottér és struktúra redukció Petri-háló alosztályok

dr. Bartha Tamás

Dr. Pataricza András

BME Méréstechnika és Információs Rendszerek Tanszék

Elérhetőségi probléma egyszerűsítése

- Redukció:
  - Érthető modellből kompakt modell
    - redundancia eliminálása
  - További egyszerűsítés: modell kifejezőereje csökken
    - ellenőrzött változtatások, de a funkcionalitás megváltozik
    - a kiválasztott tulajdonságokat megőrzi!
    - eredeti modellt a tulajdonságok szerint „fedő” modell jön létre
  - Sokféle tulajdonságmegőrző transzformáció létezik

Petri hálók redukciós módszerei

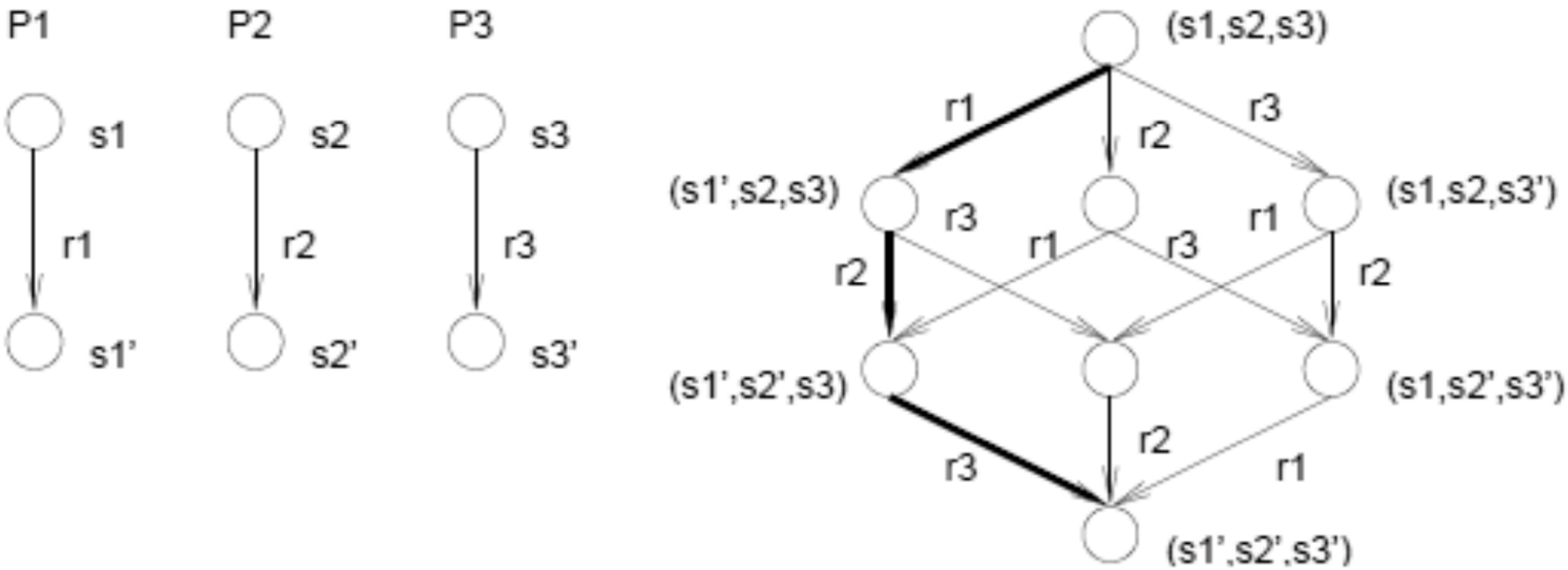
Állapottér redukció

Elérhetőségi probléma egyszerűsítése

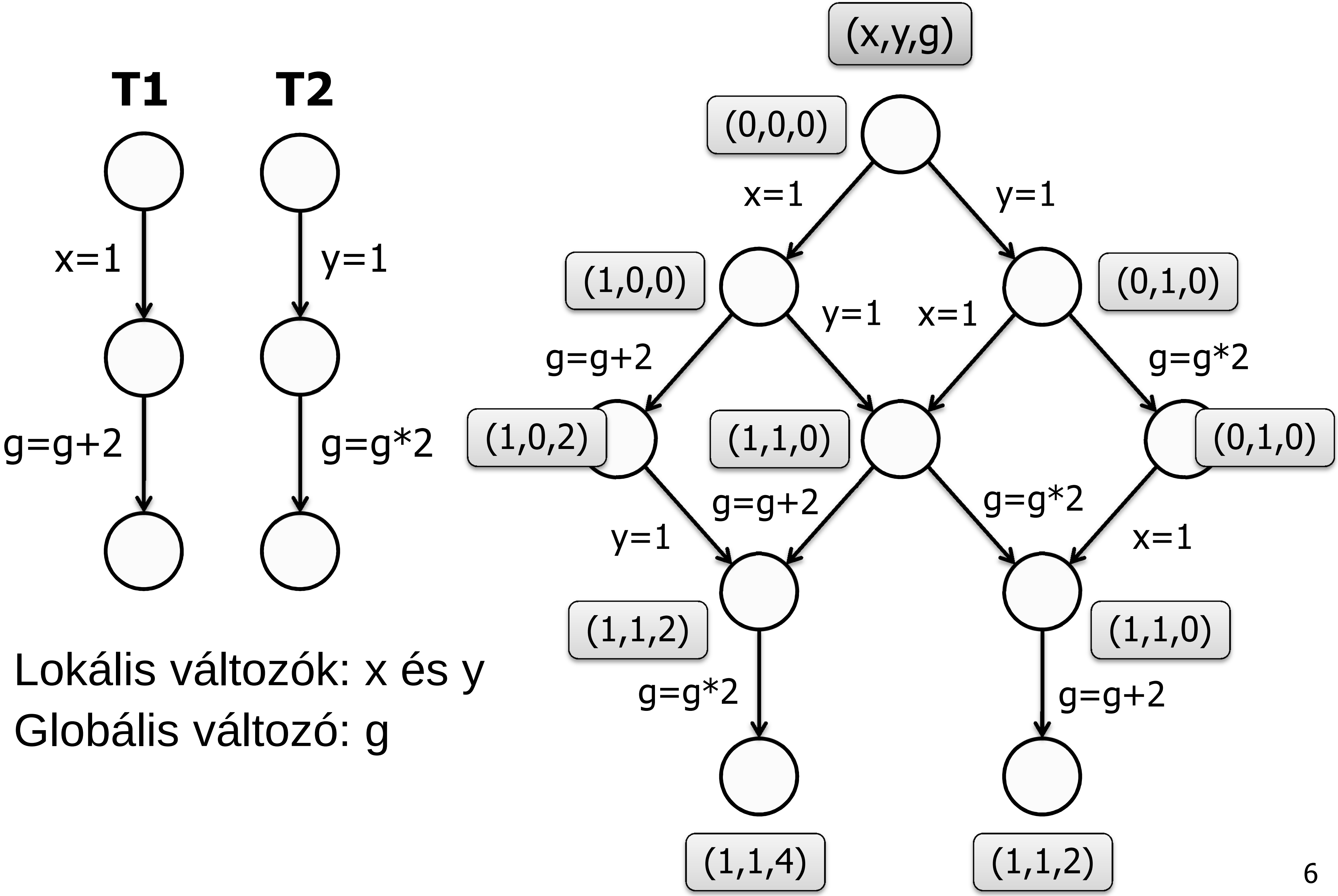
- Szimmetriák kihasználása
  - Azonos hálózatrészek csak egyszer vizsgálva
    - pl. erőforrás csoportok: azonos módon viselkedő tagok
  - Invariancia a ciklikus permutációra nézve
    - Színezett Petri hálók → Jól formált színezett Petri hálók (WFN)
- Állapottér bejárás hatékonyságának növelése
  - Csak az „érdekes” állapotok bejárása
    - Tulajdonságmegőrző redukció
  - Csak a szükséges mennyiségű állapotváltás bejárása
    - Alternatív utak elhagyása

Elérhetőségi probléma egyszerűsítése

- Részleges sorrendezési redukció
  - Elérhető állapotok részlegesen sorrendezett halmazt alkotnak
  - Aszinkron működés: átlapolás → alternatív utak, azonos eredmény
  - Végállapotokat nézve (elérhetőség) az alternatív utak redundánsak



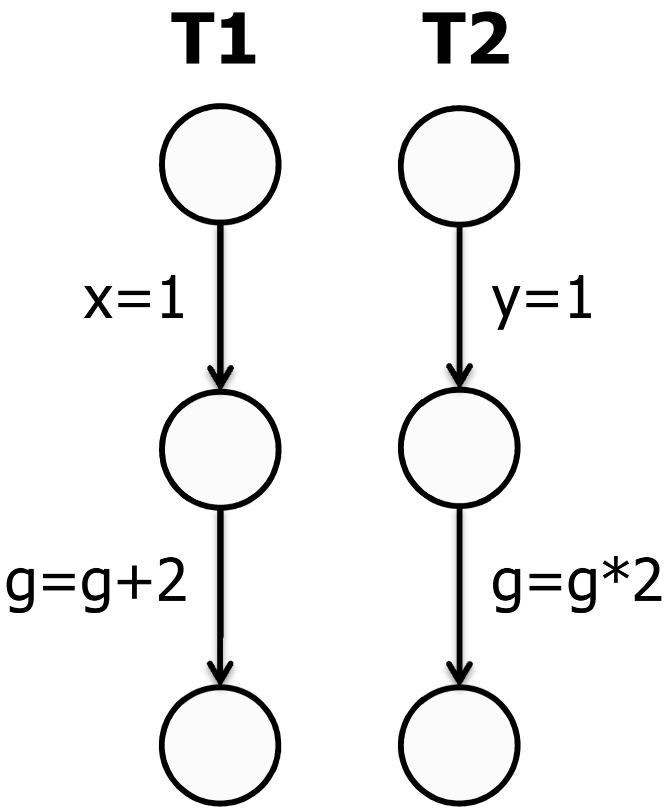
Példa: alternatív utak



Lokális változók: x és y

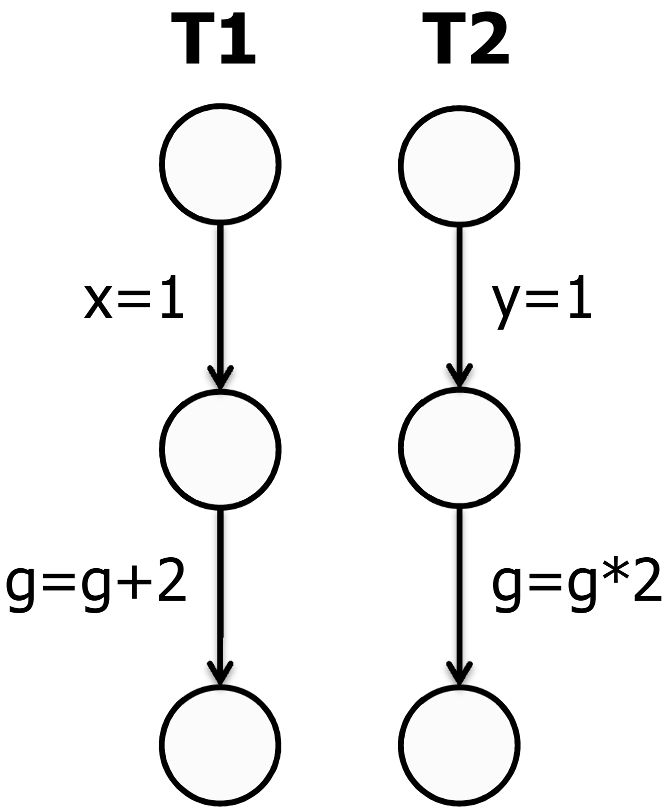
Globális változó: g

Példa: alternatív utak



- Lokális változók:
  - x és y
- Globális változó:
  - g
- 6 lehetséges lefutás:
  1. x=1; g=g+2; y=1; g=g*2
  2. x=1; y=1; g=g+2; g=g*2
  3. x=1; y=1; g=g*2; g=g+2
  4. y=1; g=g*2; x=1; g=g+2
  5. y=1; x=1; g=g*2; g=g+2
  6. y=1; x=1; g=g+2; g=g*2

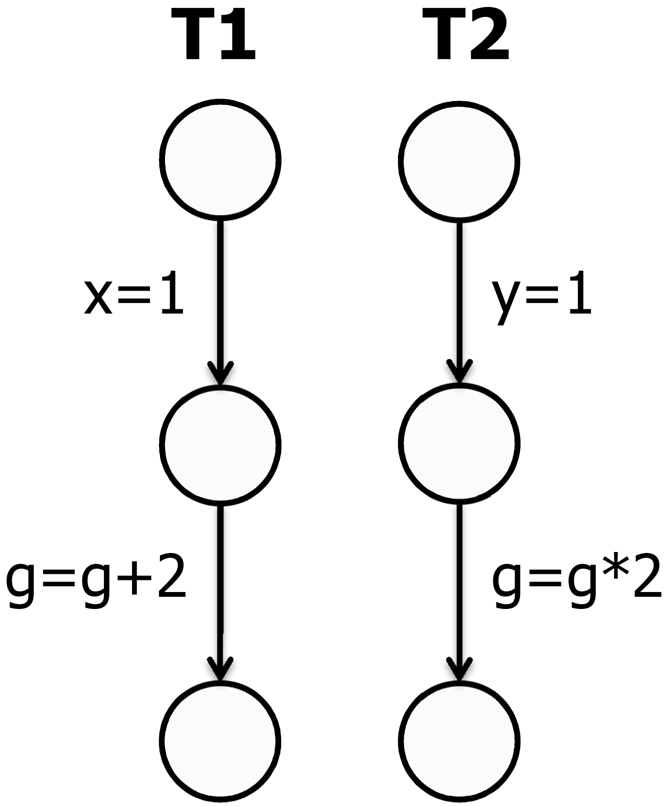
Példa: alternatív utak



F: független
V: vezérlési függőség
A: adatfüggőség

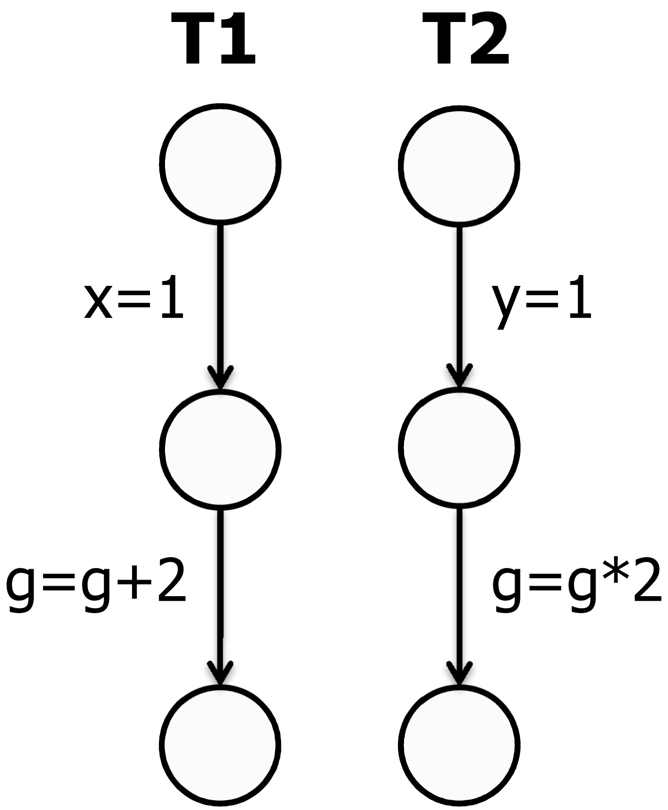
	x=1	y=1	g=g+2	g=g*2
x=1		F	V	F
y=1	F		F	V
g=g+2	V	F		A
g=g*2	F	V	A	

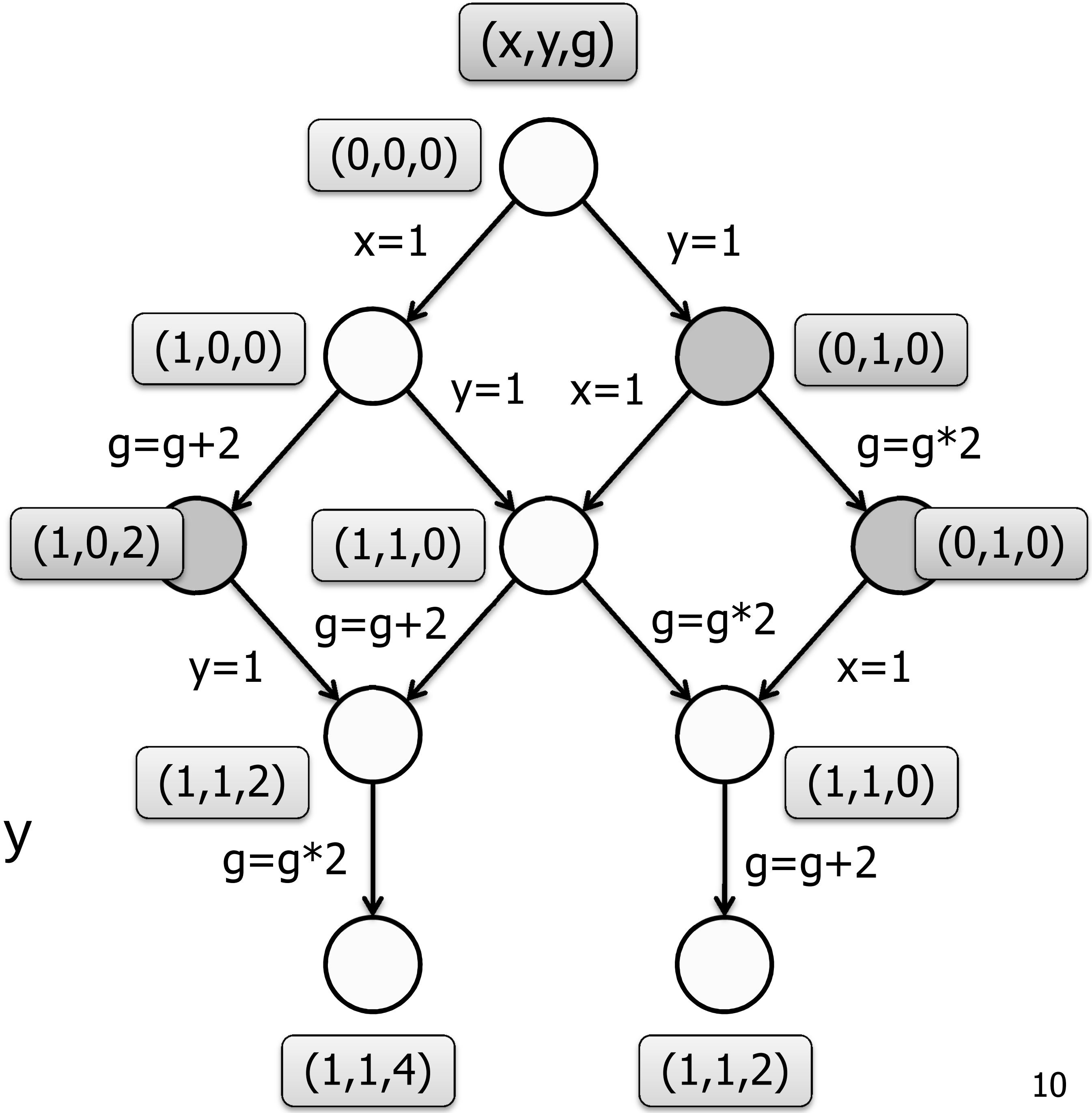
Példa: részleges sorrendezési redukció



1. x=1; g=g+2; y=1; g=g*2
2. x=1; y=1; g=g+2; g=g*2
3. y=1; x=1; g=g+2; g=g*2
4. x=1; y=1; g=g*2; g=g+2
5. y=1; x=1; g=g*2; g=g+2
6. y=1; g=g*2; x=1; g=g+2

Példa: alternatív utak





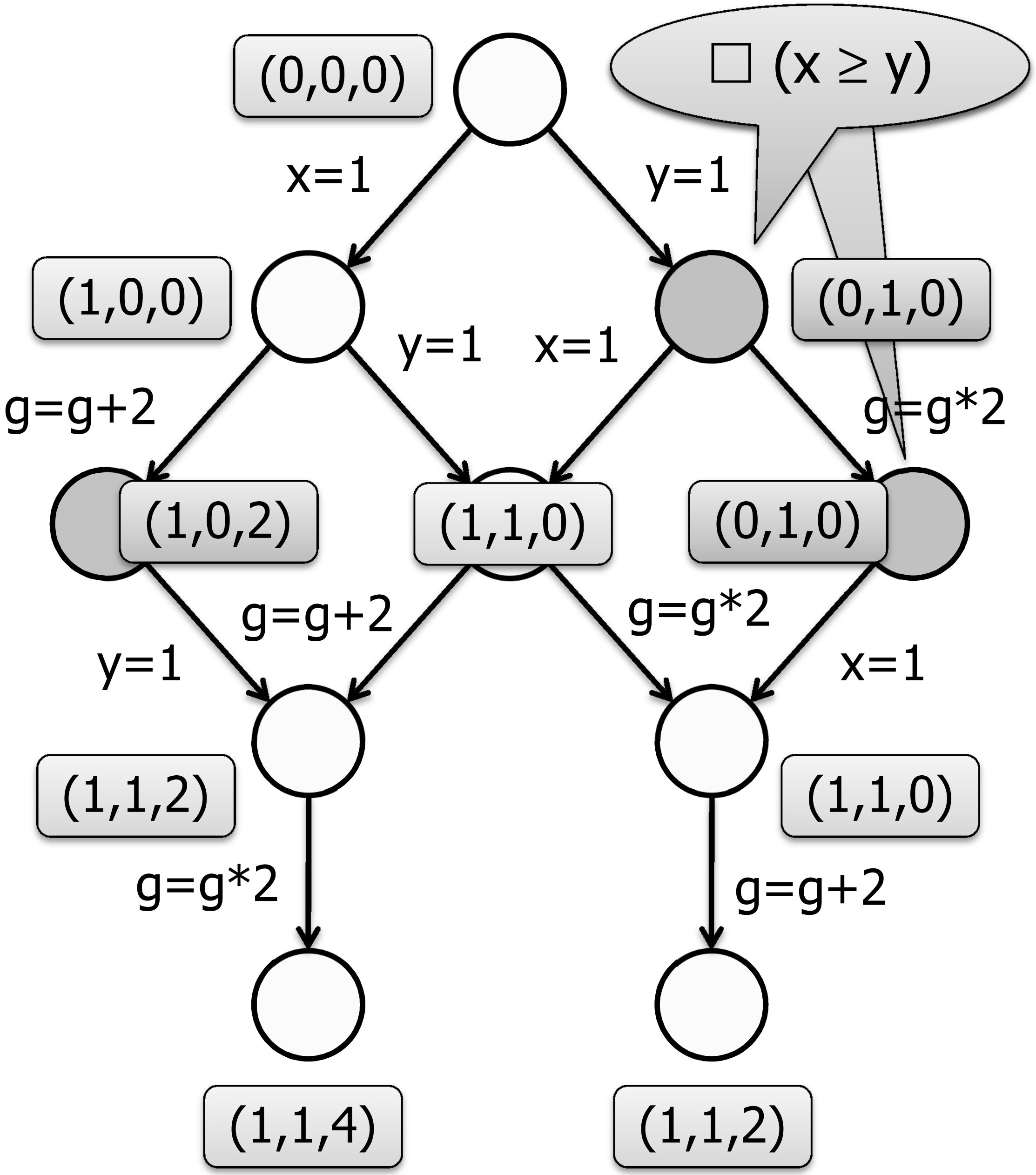
Lokális változók: x és y

Globális változó: g

A részleges sorrendezési redukció alkalmazása

- Két LTS: T1 és T2
  - sorrendezett: $x \rightarrow g$ és $y \rightarrow g$
    - igaz tulajdonság: pl. $\Box\ (g = 0 \vee g > x)$
- Redukált gráf
  - eltávolított élek: szürke
  - független: x=1 és y=1
  - adatfüggő: g+=2 és g*=2
    - igaz tulajdonság: pl. $\Diamond\ (g \geq 2)$
- Redukció
  - redundáns utak eltávolítása
  - független párok:
    - $x=1 \leftrightarrow y=1$
    - $x=1 \leftrightarrow g=g*2$
    - $y=1 \leftrightarrow g=g+2$
  - vigyázat, függ a céltól!
    - pl. $\Box\ (x \geq y)$ tulajdonság a redukáltban igaz, az eredetiben nem igaz

Példa: tulajdonság alapú függőség



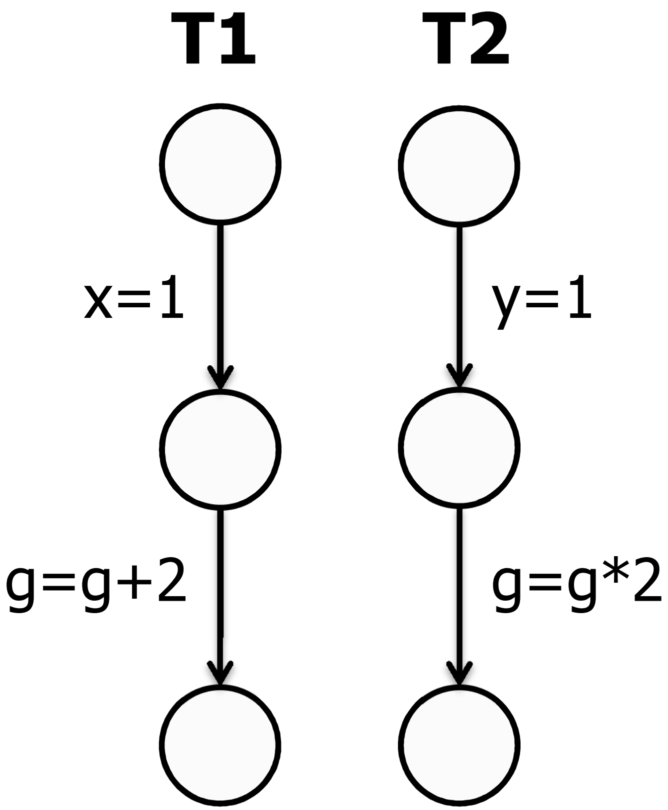
	x=1	y=1	g=g+2	g=g*2
x=1		T	V	F
y=1	T		F	V
g=g+2	V	F		A
g=g*2	F	V	A	

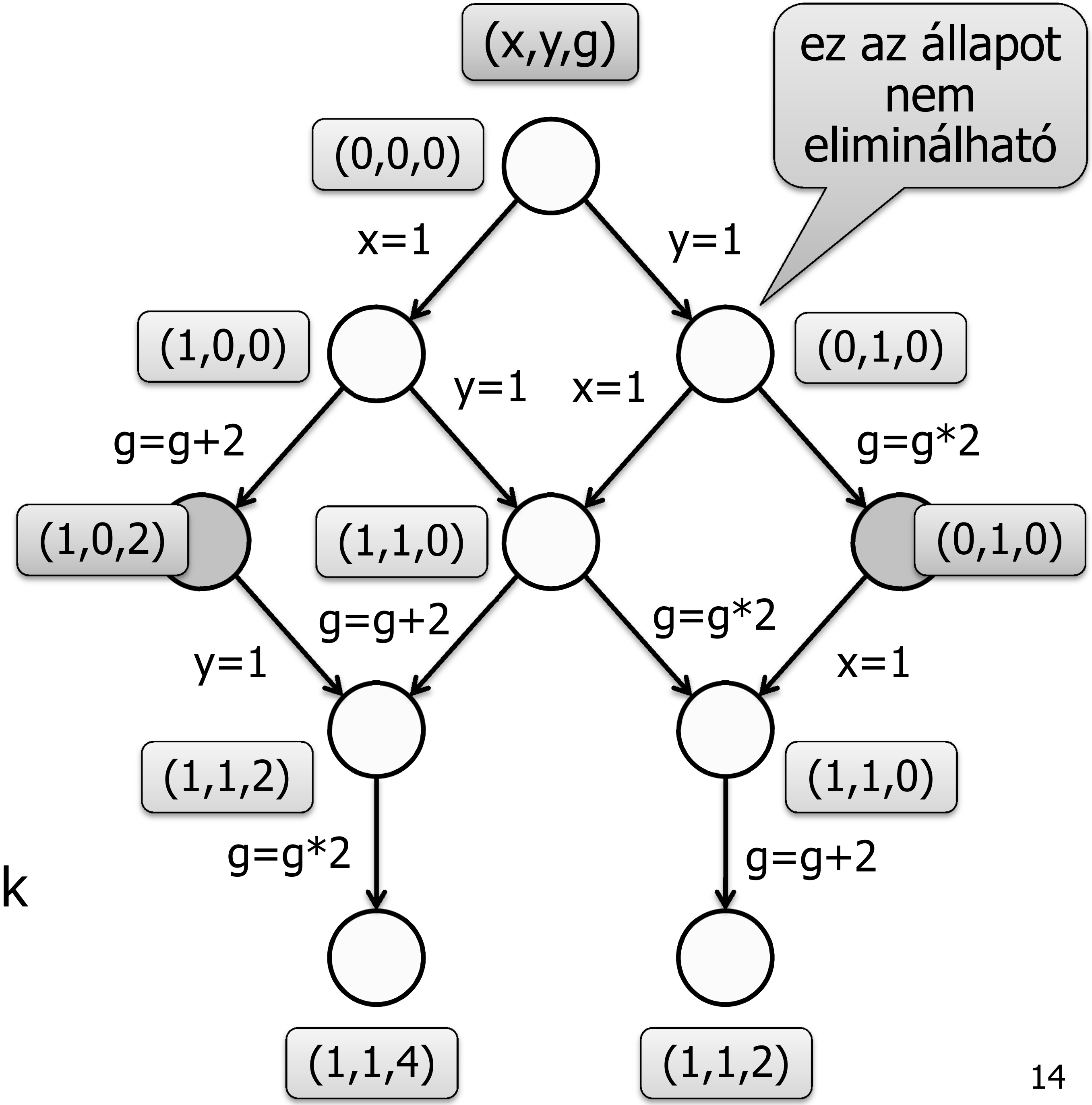
Példa: tulajdonságmegőrző redukció

	x=1	y=1	g=g+2	g=g*2
x=1		T	V	F
y=1	T		F	V
g=g+2	V	F		A
g=g*2	F	V	A	

1. x=1; g=g+2; y=1; g=g*2
2. x=1; y=1; g=g+2; g=g*2
3. y=1; x=1; g=g+2; g=g*2
4. x=1; y=1; g=g*2; g=g+2
5. y=1; x=1; g=g*2; g=g+2
6. y=1; g=g*2; x=1; g=g+2

Példa: tulajdonságmegőrző redukció





x és y → tulajdonság szempontból függenek

Részleges sorrendezési redukció alapja

- Két állapotátmenet független egy s állapotban, ha
  - mindkettő engedélyezett az s állapotban
  - egyikük végrehajtása sem tiltja le a másikat (nincs vezérlési függőség – perzisztencia)
  - a két állapotátmenet együttes hatása független a végrehajtási sorrendtől (sem adat, sem tulajdonság függőség)
- Erős függetlenség
  - két állapotátmenet erősen független, ha függetlenek minden olyan állapotban, amelyben mindkettő engedélyezett

Petri hálók redukciós módszerei

Struktúra redukció

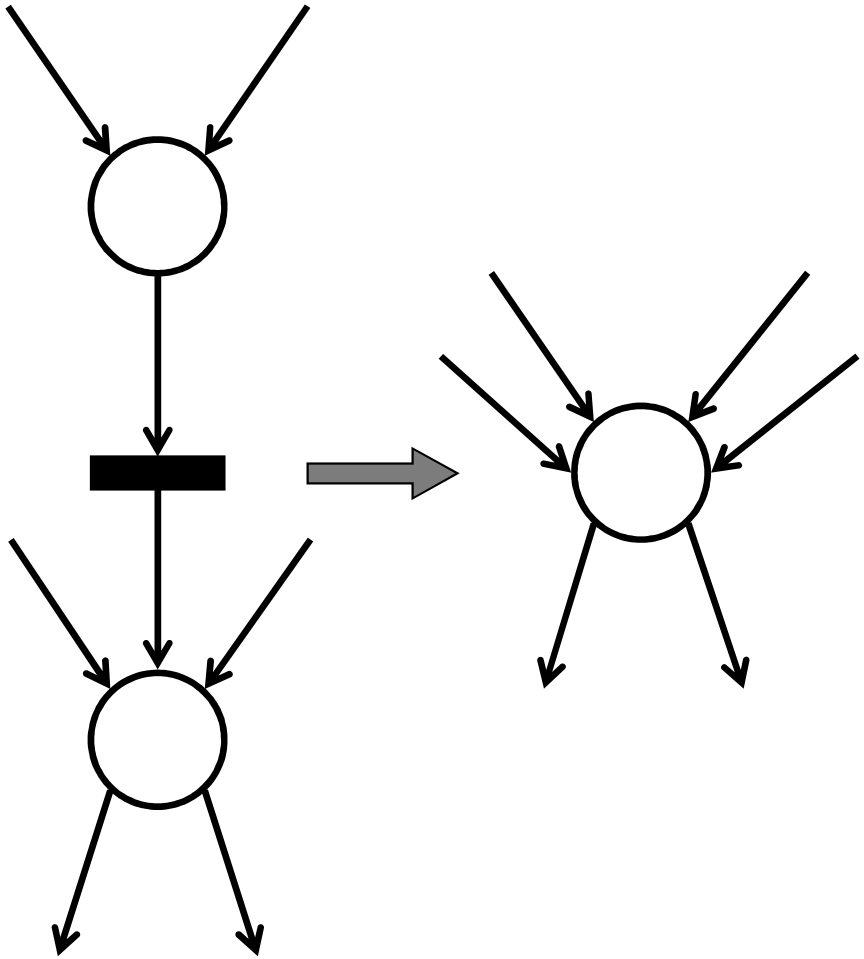
Elérhetőségi probléma kezelése

- Struktúra redukálása
  - Tulajdonságmegtartó transzformáció redukált modellre
- Hierarchikus modellezés
  - Részhálózatok összevonása egyetlen csomóponttá
  - Petri hálók nemdeterminizmusa $\rightarrow$ modellabsztrakció
    - Keresési tér behatárolása durvább modellen
    - Részletes analízis egy finomított modellen
- Kompozícionális verifikáció
  - Rendszerek $\Leftarrow$ részrendszerek + interfészek + együttműködés
  - Részrendszerek analízise és az együttműködések vizsgálata
  - A teljes rendszer analízise a részrendszerekre kapott eredményekből

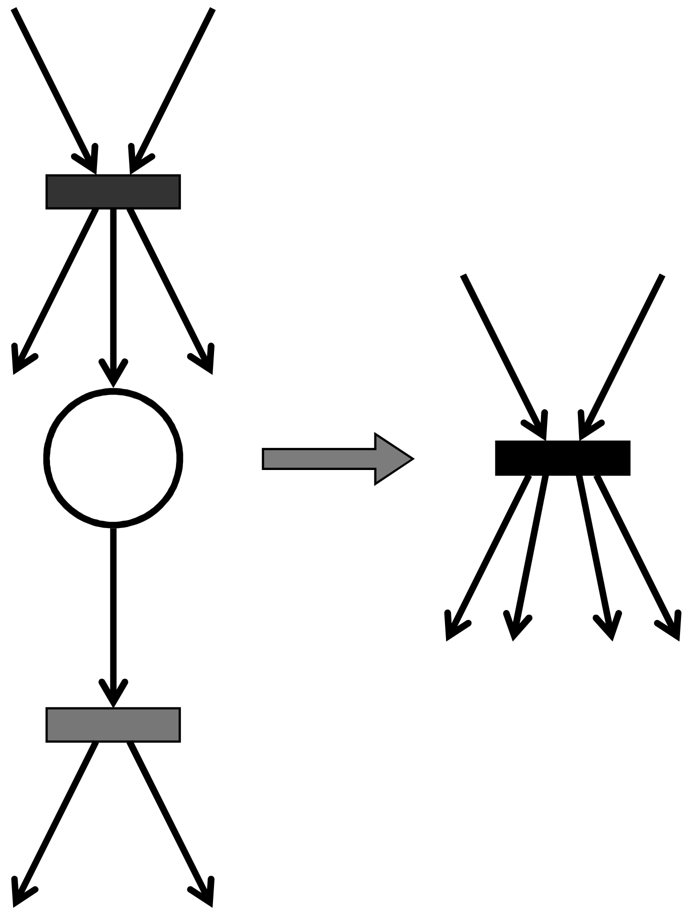
Transzformációk

- Egyszerű tulajdonságmegőrző transzformációk:
  - soros helyek összevonása
  - soros tranzíciók összevonása
  - párhuzamos helyek összevonása
  - párhuzamos tranzíciók összevonása
  - önhurkot alkotó helyek törlése
  - önhurkot alkotó tranzíciók törlése
- Megőrzik az élő, korlátos és biztos tulajdonságot

Soros összevonások

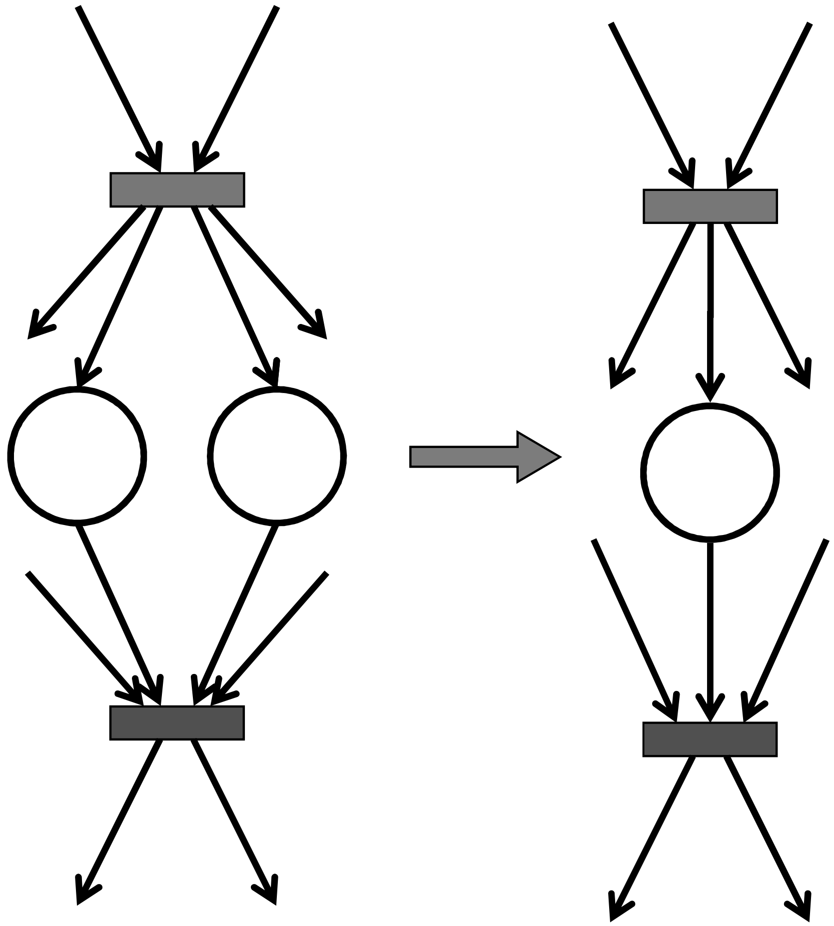


soros helyek összevonása (FSP)

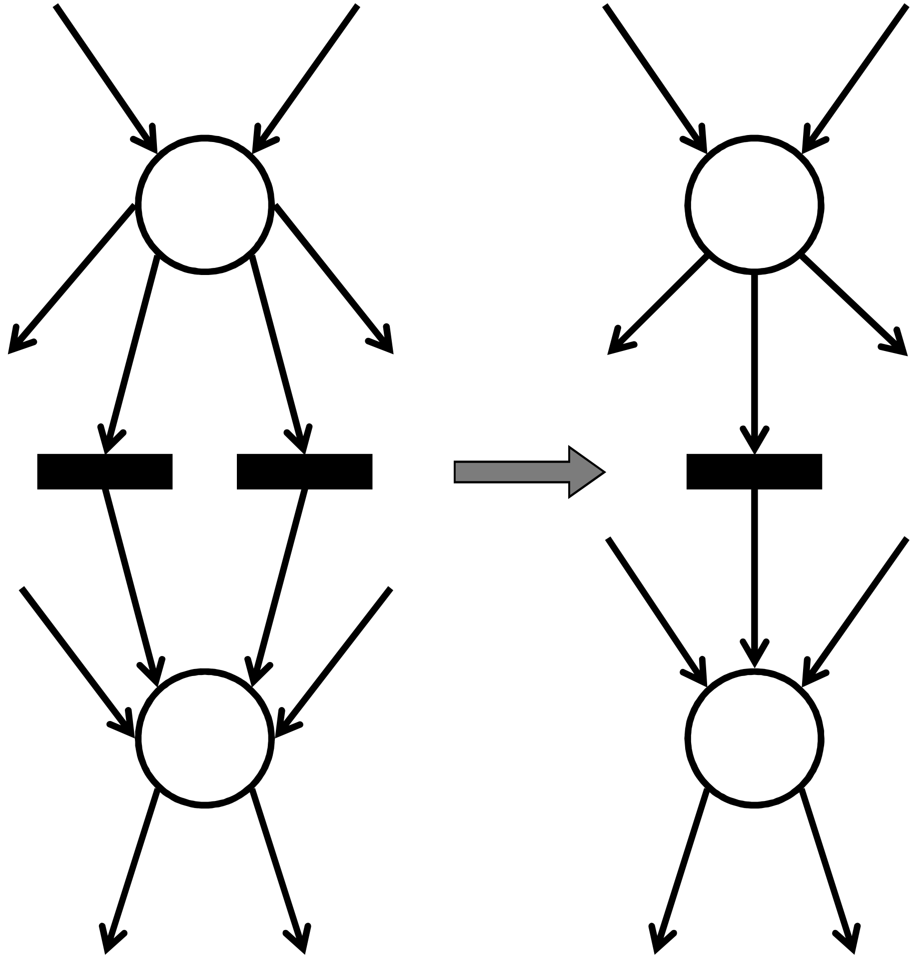


soros tranzíciók összevonása (FST)

Párhuzamos összevonások

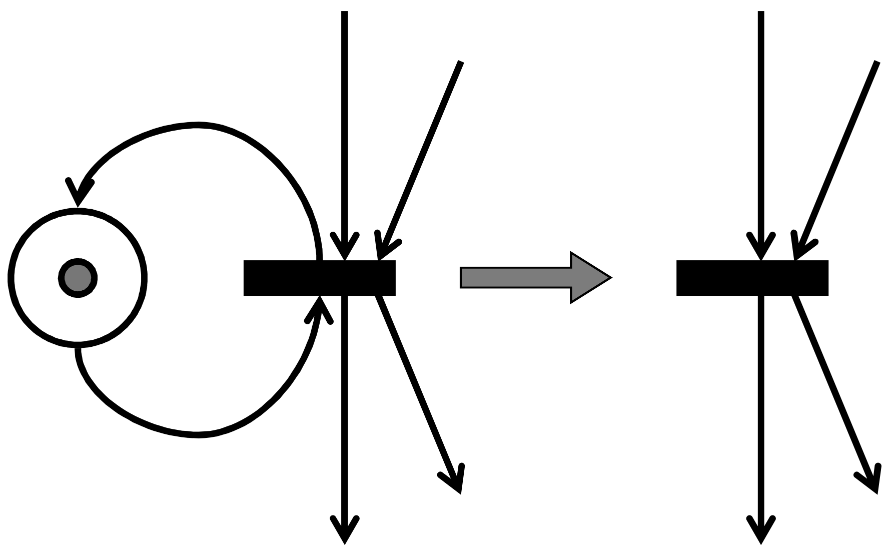


párhuzamos helyek összevonása (FPP)

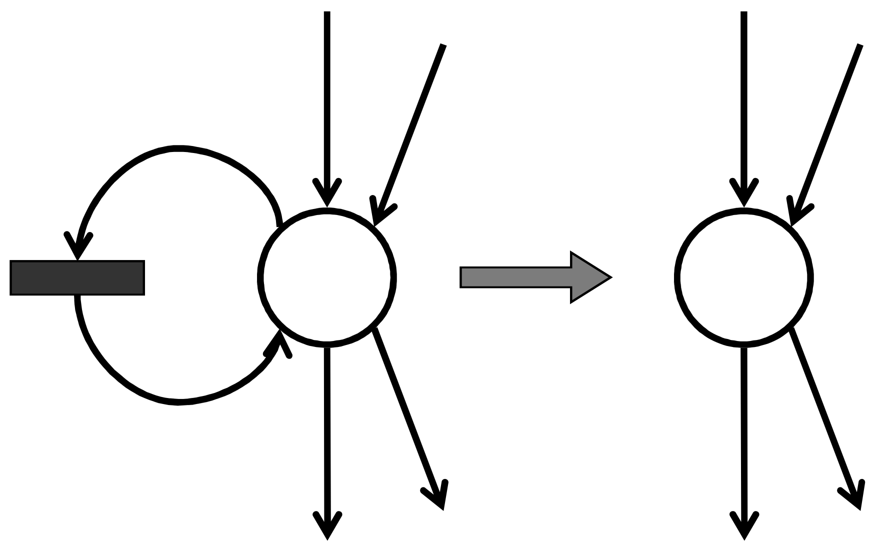


párhuzamos tranzíciók összevonása (FPT)

Önhurkok törlése

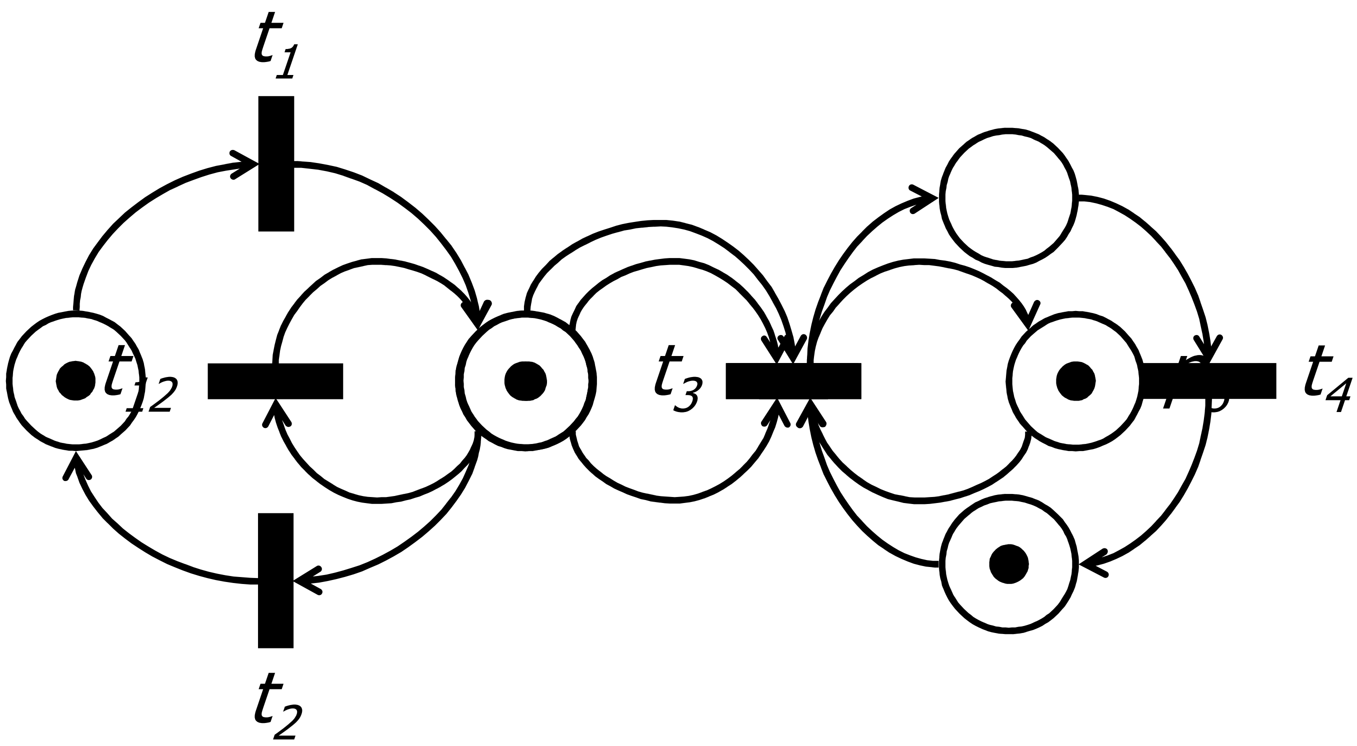


önhurkot alkotó helyek törlése (ESP)



önhurkot alkotó tranzíciók törlése (EST)

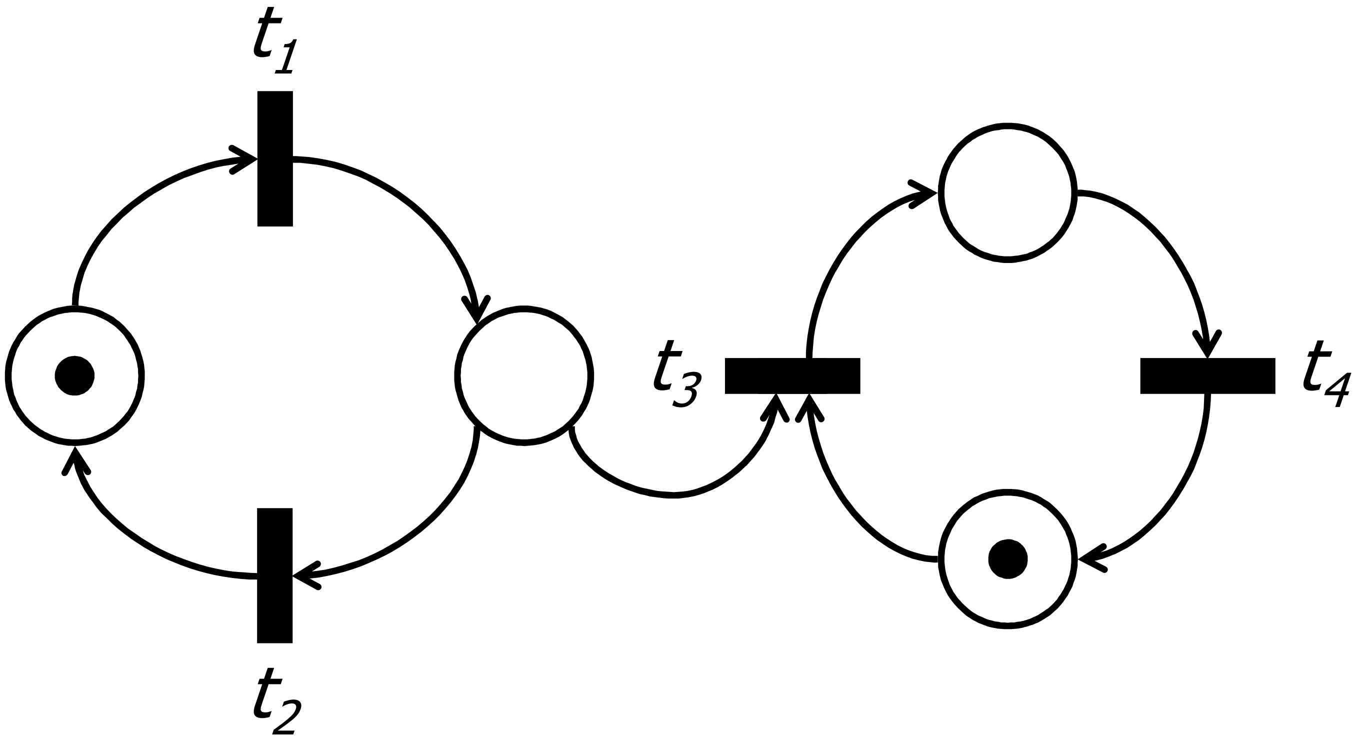
Példa



a példa korlátos, de nem élő (és nem megfordítható)

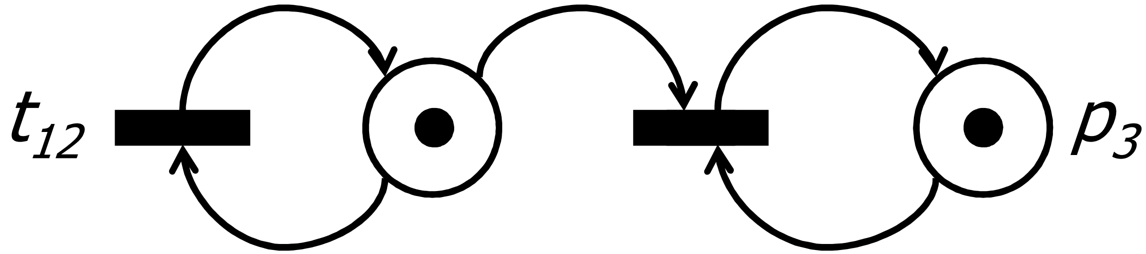
- t_1 és t_2 összevonása (soros tranzíciók) → t_{12}
- t_3 és t_4 összevonása (soros tranzíciók) → t_{34}
- t_{12} törlése (önhurkot alkotó tranzíció)
- p_3 törlése (önhurkot alkotó hely)

Példa



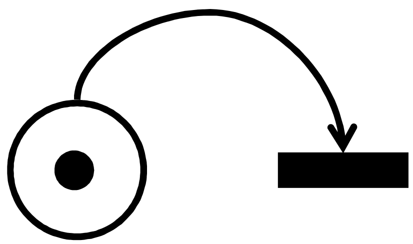
- t_1 tüzelése
- t_1 és t_2 összevonása (soros tranzíciók) $\rightarrow$ t_{12}
- t_3 és t_4 összevonása (soros tranzíciók) $\rightarrow$ t_{34}

Példa: 2. lépés



- t_{12} törlése (önhurkot alkotó tranzíció)
- p_3 törlése (önhurkot alkotó hely)

Példa: eredmény



a példa korlátos, de nem élő (és nem megfordítható)

Egyszerű struktúrájú Petri hálók viselkedése

Konfliktus, konkurencia

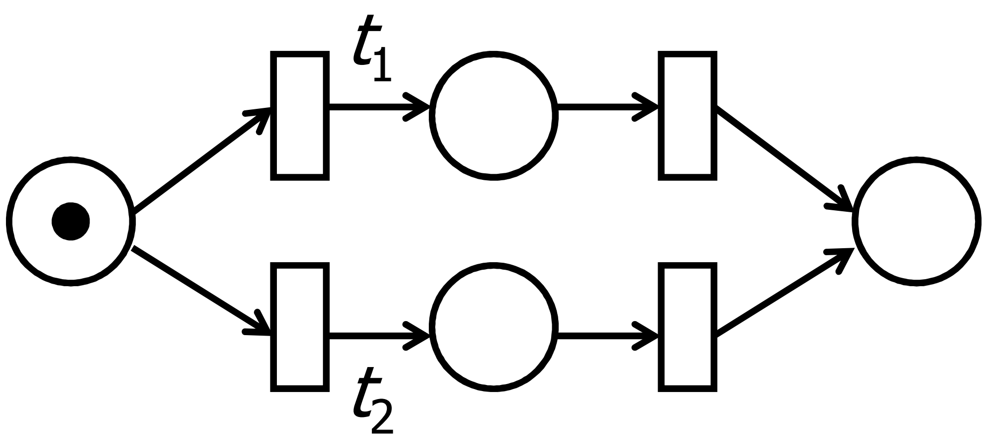
Konfliktus (strukturális): kizáró események, választás

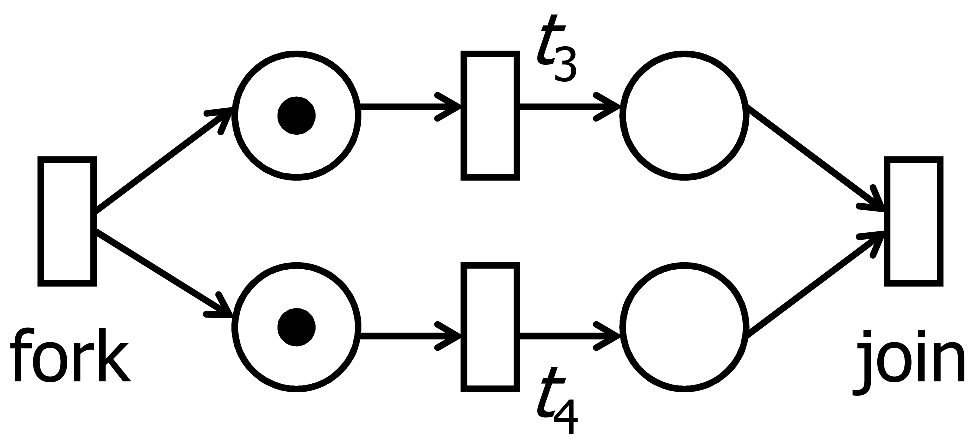
e_1 vagy e_2 esemény következik be, de csak az egyik

Konkurencia: párhuzamos események

e_1, e_2 esemény kauzálisan független: $e_1 \not\rightarrow e_2 \wedge e_2 \not\rightarrow e_1$

- reflexív: e_1 co e_1 és e_2 co e_2
- szimmetrikus: e_1 co $e_2 \Rightarrow e_2$ co e_1
- nem tranzitív: e_1 co $e_2 \wedge e_2$ co $e_3 \not\Rightarrow e_1$ co e_3

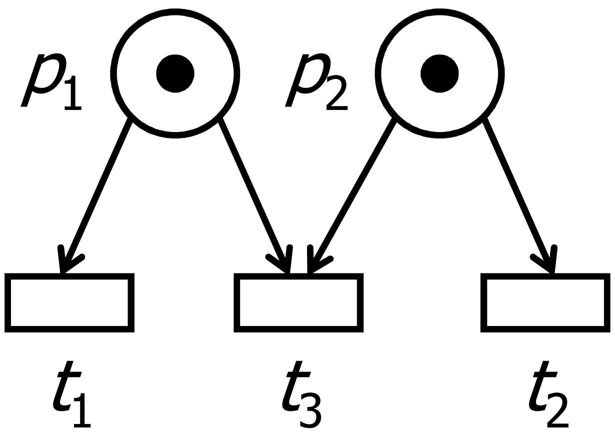


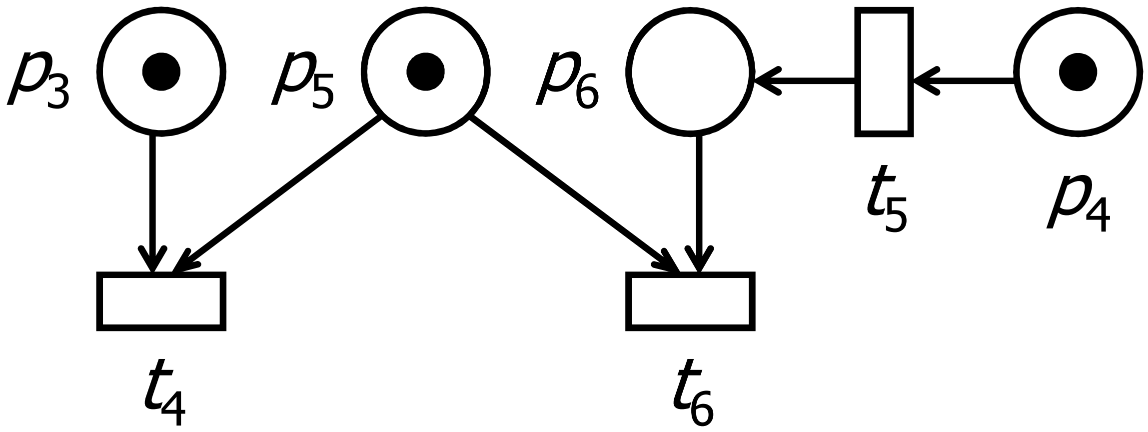


Konfúzió

Konfúzió: konkurencia és konfliktus is jelen van

- Szimmetrikus: egyaránt konkurens és konfliktusos
  - t_1 és t_2 konkurens, mindkettő konfliktusban van t_3 átmenettel
- Aszimmetrikus: tüzelési szekvenciától függően
  - t_4 és t_5 konkurens, de ha t_5 tüzel előbb, akkor t_4 konfliktusba kerül t_6 átmenettel





Petri háló alosztályok

A továbbiakban végig
közönséges Petri hálókról
beszélünk!

Alosztályok

- Állapotgép (State Machine, SM)
  - minden átmenetnek pontosan egy be- és kimeneti helye

$$\boxed{\forall t \in T : |\bullet t| = |t \bullet| = 1}$$

  - van konfliktus, nincs szinkronizáció
- Jelölt gráf (Marked Graph, MG)
  - minden helynek pontosan egy be- és kimeneti tüzelése

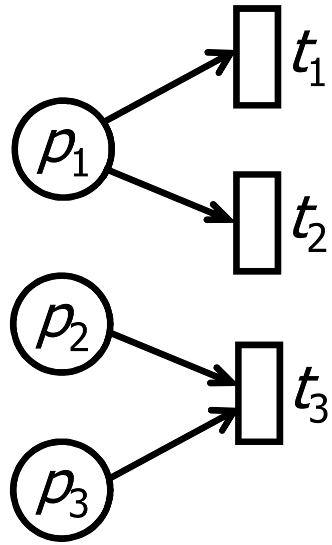
$$\boxed{\forall p \in P : |\bullet p| = |p \bullet| = 1}$$

  - van szinkronizáció, nincs konfliktus

Alosztályok (folyt.)

- Szabad választású háló (Free-Choice Net, FC)
  - közönséges Petri háló, melyben minden helyből kifelé mutató él vagy egyedüli kimenő él, vagy egyedüli bemenő él egy átmenetbe

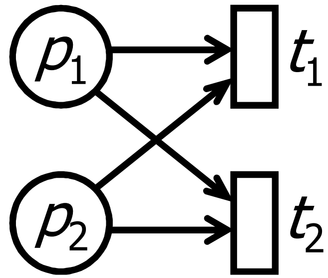
$$\boxed{\begin{array}{l}\forall p \in P : \left|p\bullet\right| \geq 1 \Rightarrow \bullet\left(p\bullet\right) = \left\{p\right\}, \\ \forall p_1, p_2 \in P : p_1\bullet \cap p_2\bullet \neq \emptyset \Rightarrow \left|p_1\bullet\right| = \left|p_2\bullet\right| = 1\end{array}}$$



  - lehet konkurencia és konfliktus is, de nem egyszerre
    - nincs benne konfúzió
  - dekomponálható MG és SM komponensekre

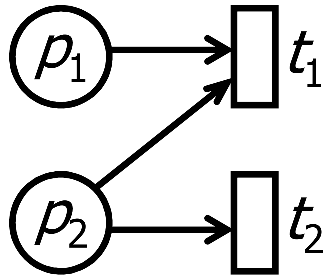
Alosztályok (folyt.)

- Kiterjesztett szabad választású háló (EFC)
  - többszörös szinkronizáció is lehetséges



$$\forall p_1, p_2 \in P : p_1 \bullet \cap p_2 \bullet \neq \emptyset \Rightarrow p_1 \bullet = p_2 \bullet$$

- Aszimmetrikus választású háló (AC)
  - lehetséges az (aszimmetrikus) konfúzió is

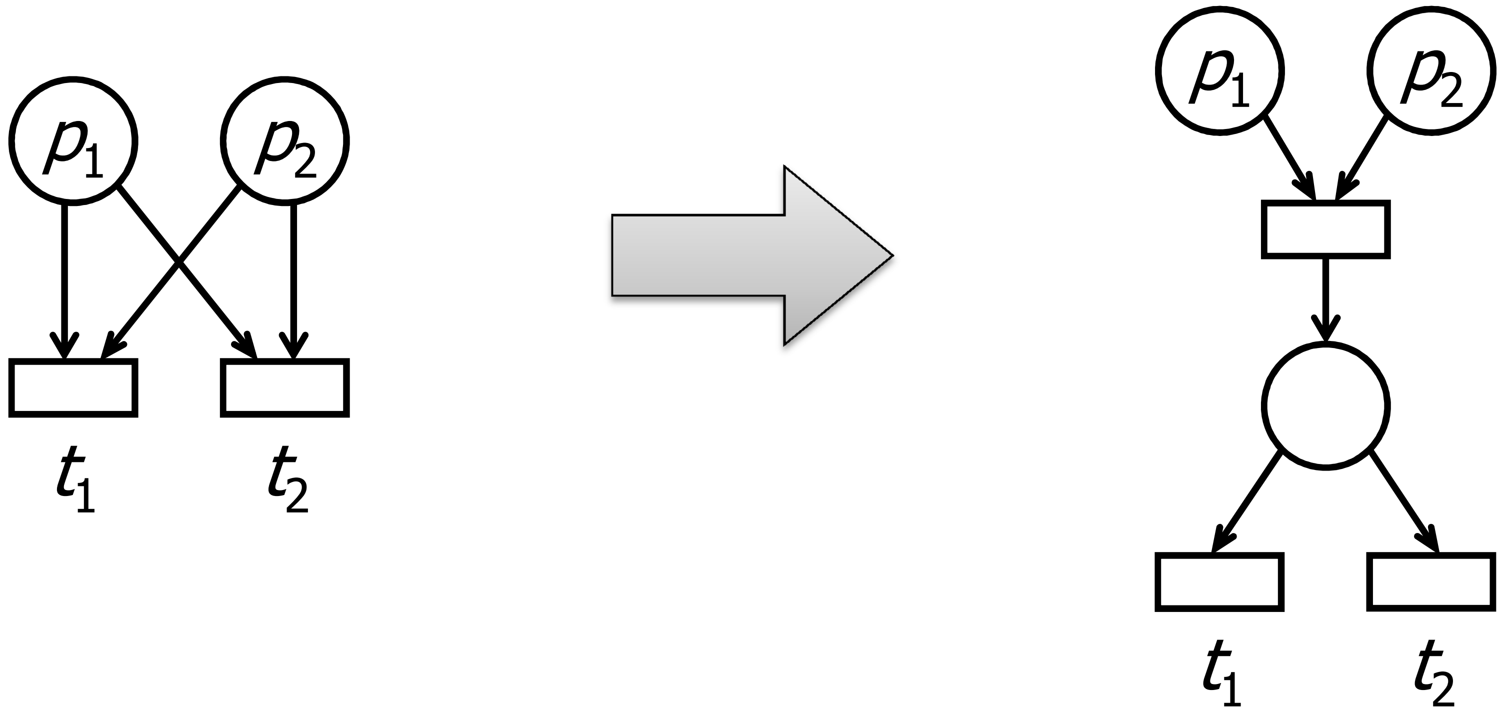


$$\forall p_1, p_2 \in P : p_1 \bullet \cap p_2 \bullet \neq \emptyset \Rightarrow p_1 \bullet \subseteq p_2 \bullet \vee p_2 \bullet \subseteq p_1 \bullet$$

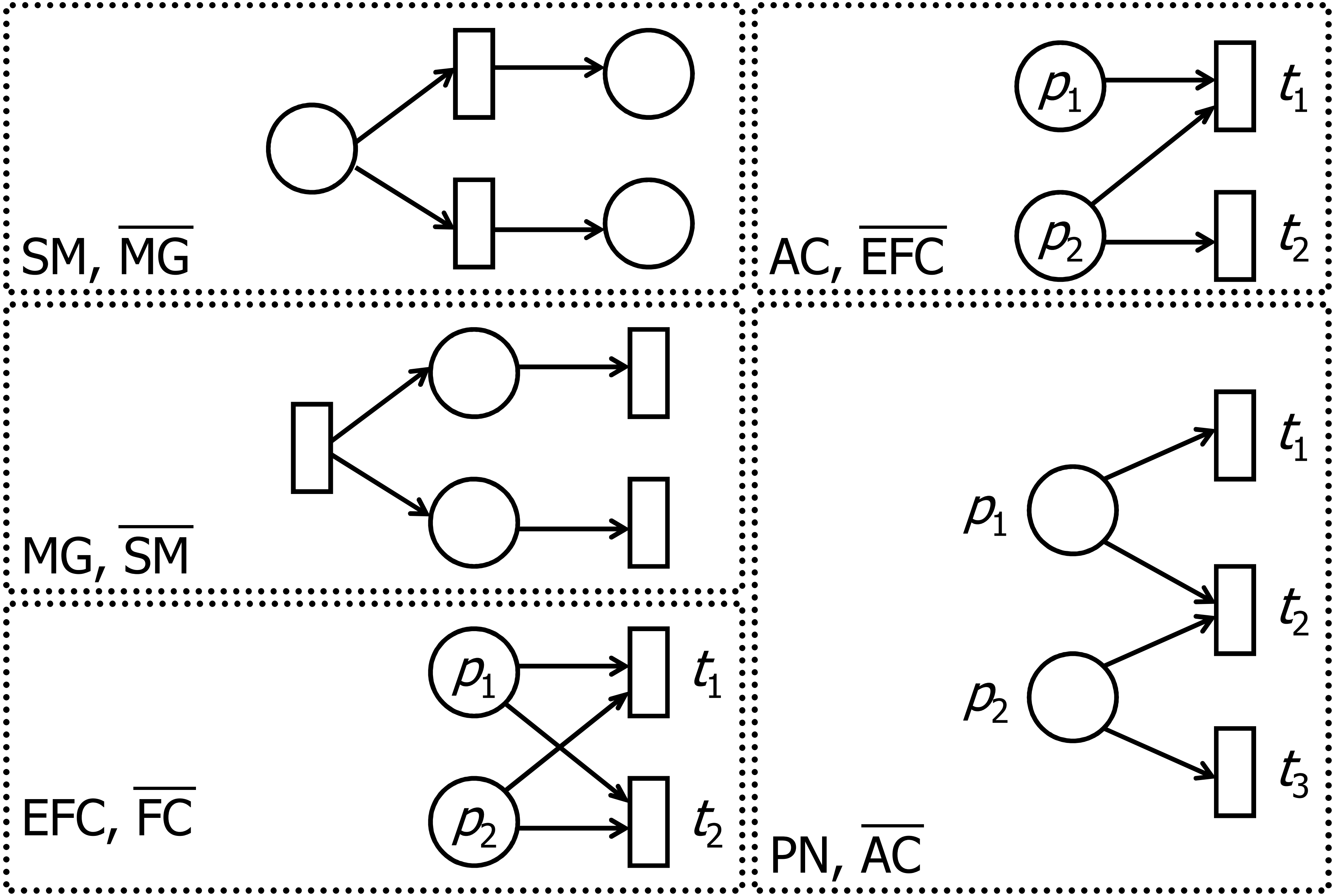
- Nincs szimmetrikus választású $\rightarrow$ a többi PN

EFC transzformációja FC-re

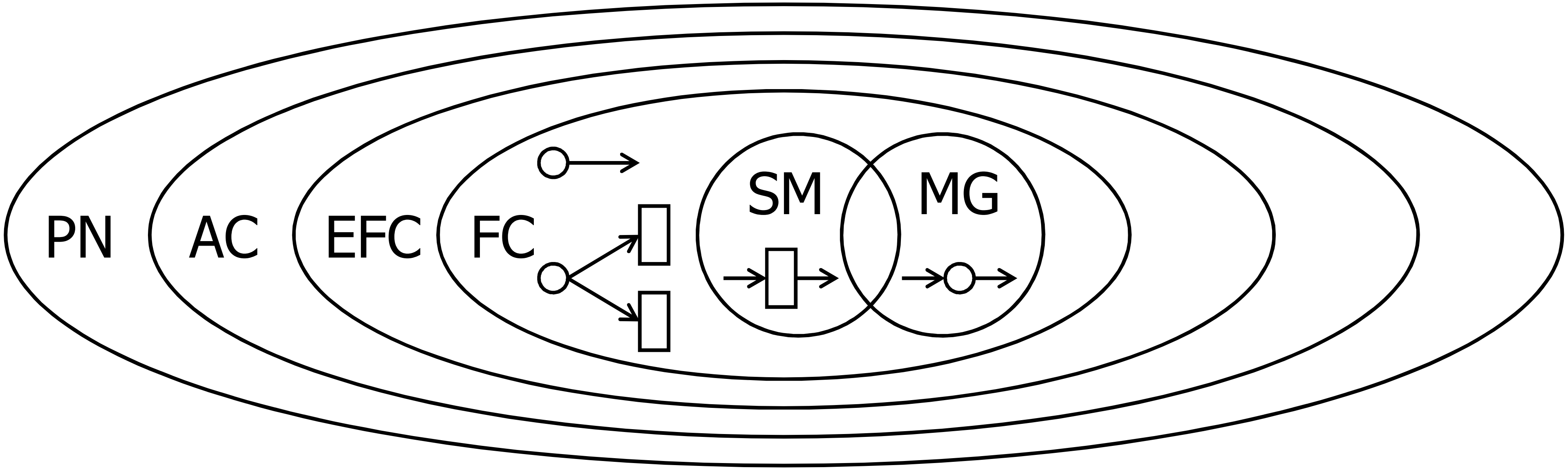
- FC és EFC közös tulajdonsága:
  - ha t_1 és t_2 $\exists$ közös bemeneti helye, akkor nincs olyan állapot, melyben az egyik engedélyezett és a másik nem
  - EFC transzformálható tulajdonságmegtartó módon FC-re



Alapstruktúrák



Az alosztályok viszonya



- Állapotgép (SM) nem enged meg szinkronizációt
- Jelölt gráf (MG) nem enged meg választást
- Szabad választású háló (FC) nem enged meg konfúziót
- Aszimmetrikus vál. háló (AC) megenged aszimmetrikus konfúziót, de nem enged meg szimmetrikus konfúziót

Élő és bizt(onság)os tulajdonság kritériumai

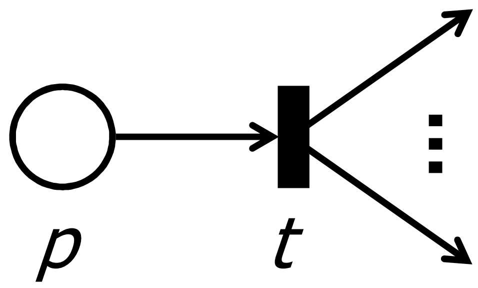
Szükséges feltétel általános esetben:

– Ha egy (N, M_0) Petri háló élő <u>és</u> biztos, akkor N erősen (szigorúan) összekötött gráf topológia
  - bármely csomópontból bármely másikba vezet irányított út
  - sem forrás, sem nyelő csomópontok: $\forall n \in P \cup T : \bullet n \neq \emptyset \neq n \bullet$

– Elégséges feltételek alosztályokra fogalmazhatók meg

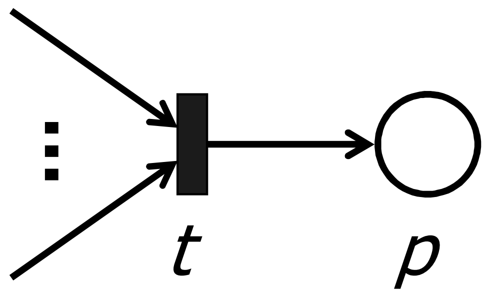
Miért nem lehetnek forrás és nyelő csomópontok?

$\bullet p = \emptyset$
(forrás hely)



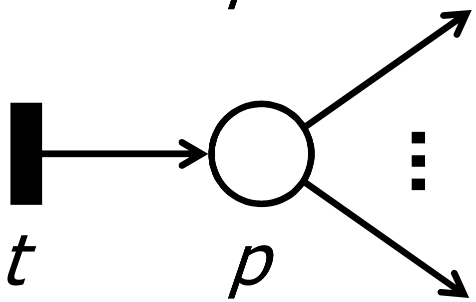
t nem élő

$p\bullet = \emptyset$
(nyelő hely)



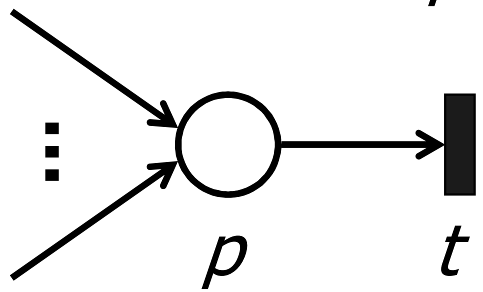
p nem biztos, ha t élő

$\bullet t = \emptyset$
(forrás tranzíció)



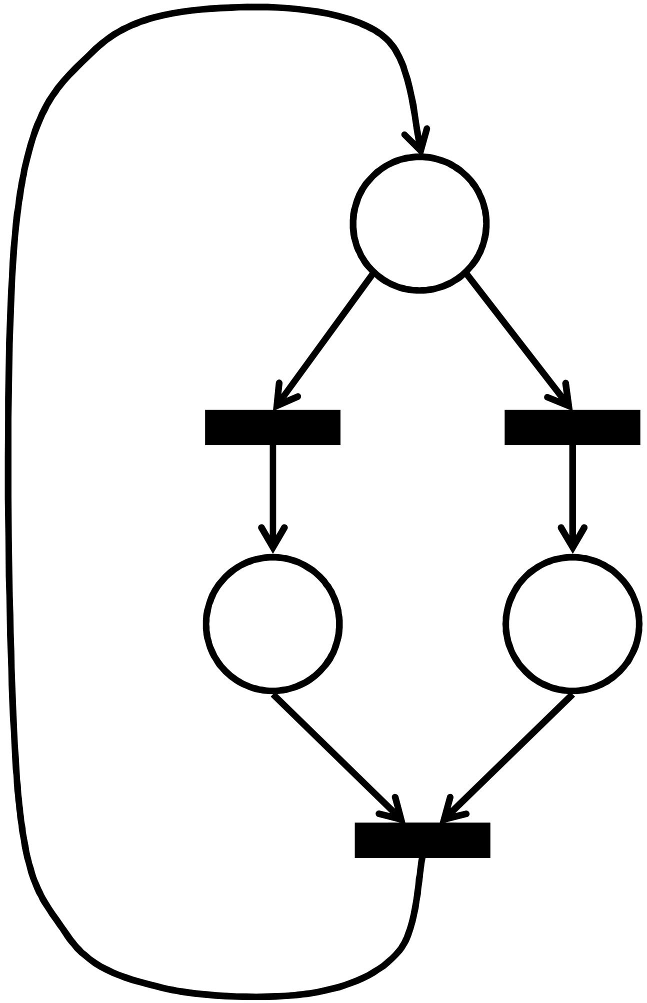
p nem biztos

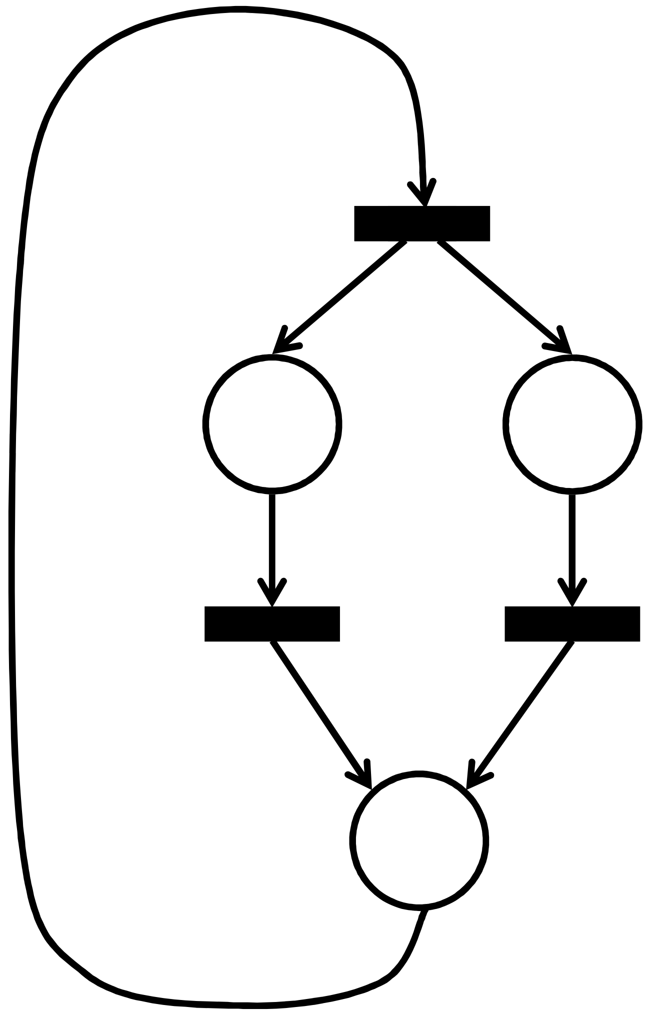
$t\bullet = \emptyset$
(nyelő tranzíció)



t nem élő, ha p biztos

Miért nem elégséges az összekötöttség?



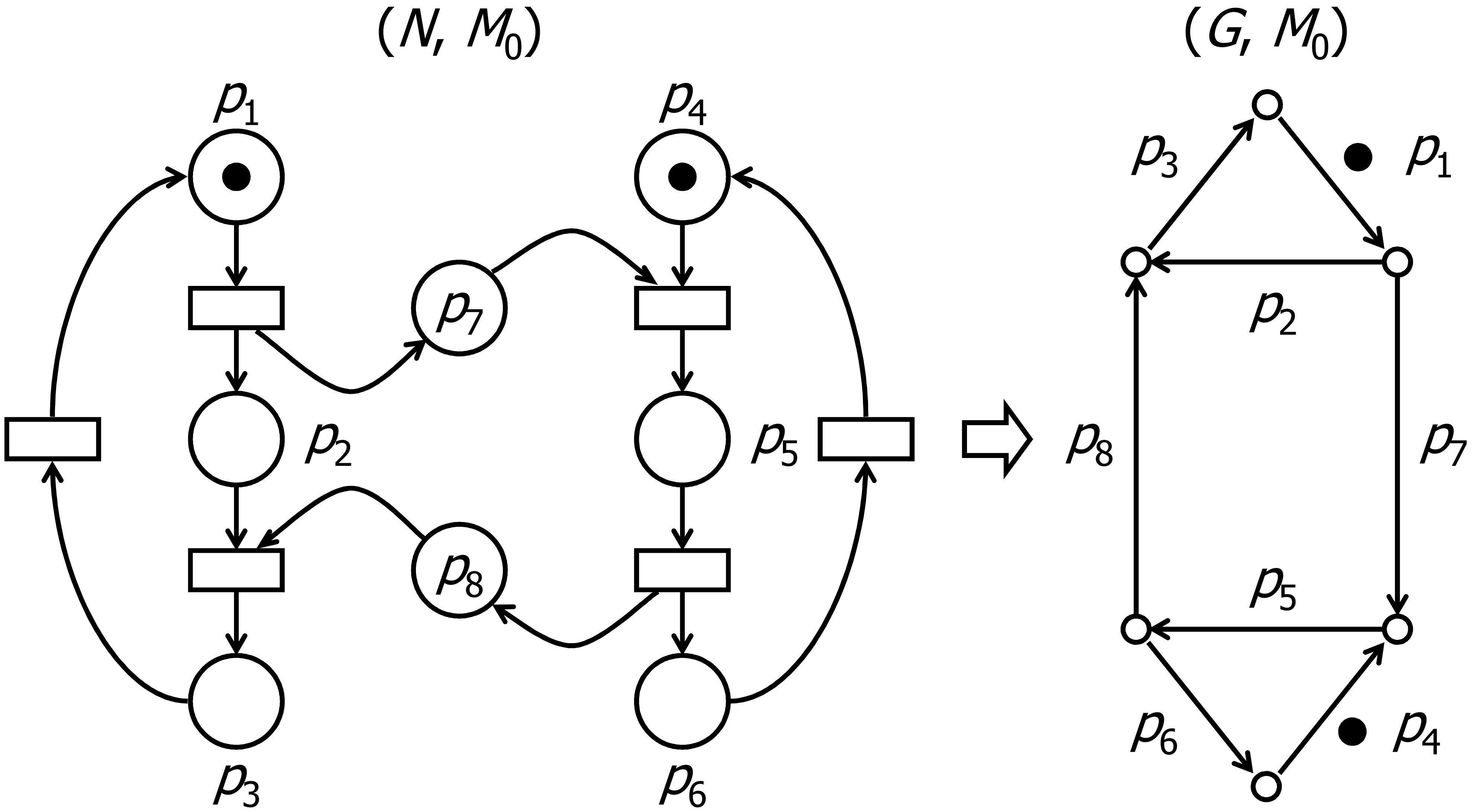


nincs élő jelölése   nincs nemüres biztos jelölése

Élő és biztos tulajdonság az SM hálókban

1. Egy (N, M_0) állapotgép a.cs.a. **élő**, ha N erősen összefüggő és M_0-ban van legalább egy token
   - triviális, hiszen minden tüzelés csak egy tokent mozgat
2. Egy (N, M_0) állapotgép a.cs.a. **biztos**, ha M_0-ban van legfeljebb egy token
3. Egy élő (N, M_0) állapotgép a.cs.a. **biztos**, ha M_0-ban pontosan egy token van

Jelölt gráf reprezentáció



Élő tulajdonság az MG hálókban

4. Egy (G, M_0) jelölt gráfban a tokenek száma minden C irányított körben állandó

$$\boxed{\forall M \in R(N, M_0), \forall C \subseteq G : M(C) = M_0(C)}$$

   - közönséges Petri háló: egyszeres élek; minden csomóponthoz a körben egy bemenő és egy kimenő él

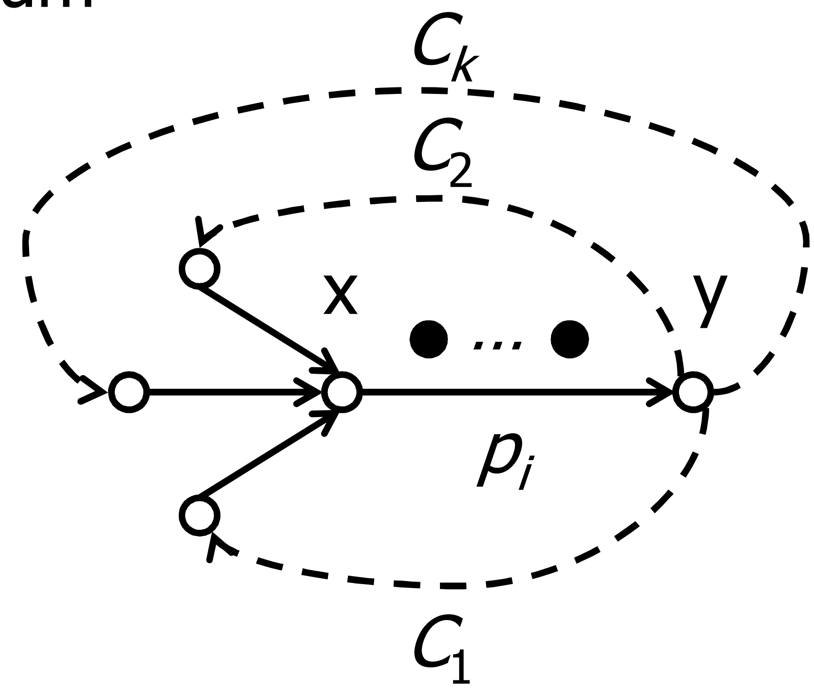
5. Egy (G, M_0) jelölt gráf a.cs.a. **élő**, ha M_0 állapotban minden G-beli irányított körben van legalább egy token

Biztos tulajdonság az MG hálókban

6. Egy (G, M_0) jelölt gráfban egy élt jelölő tokenek maximális száma egyenlő az élt tartalmazó bármely irányított körön az M_0 állapotban levő tokenek minimális számával
   - y elsütése nélkül x-et maximum

     $$\min_{\forall C_j \subseteq G | p_i \in C_j} \left(M_0(C_j) \right)$$

     alkalommal lehet elsütni



Biztos tulajdonság az MG hálókban

7. Egy élő (G, M_0) jelölt gráf a.cs.a. **biztos**, ha minden él (hely) olyan C irányított körben van, amelyre $M_0(C) = 1$

8. Egy G irányított gráfban a.cs.a. létezik **élő** és **biztos** jelölt gráfot létrehozó M_0 állapot, ha G erősen összefüggő gráf
   - a feltétel triviálisan szükséges
   - elégséges is, hiszen van legalább egy irányított kör, és minden irányított körbe elég egy tokent tenni

Biztos tulajdonság az MG hálókban

- Visszacsatoló élhalmaz (Feedback Arc Set, FAS)
  - Egy E′ élhalmaz visszacsatoló élhalmaz, ha elhagyásával a G erősen összefüggő gráf irányított kör mentessé válik, azaz a $G' = (V, E - E')$ körmentes
    - minimális FAS: egyetlen valódi részhalmaza sem FAS
    - minimum FAS: egyetlen más FAS sem tartalmaz kevesebb élt

9. Egy <u>erősen összekötött élő</u> (G, M_0) jelölt gráf a.cs.a. **biztos**, ha az M_0 kezdőállapotból elérhető minden $M \in R(G, M_0)$ állapotban a jelölt élek halmaza minimális visszacsatoló élhalmaz

FC hálók vizsgálata: szifon és csapda

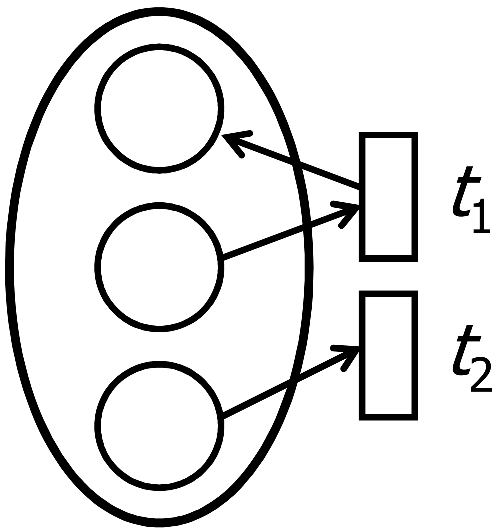
Egy S helyhalmaz **szifon**, ha $\bullet S \subseteq S \bullet$

- minden S-beli kimeneti helyhez bemeneti hely S-ben
- egy állapotban tokenmentes $\rightarrow$ követő állapotokban is

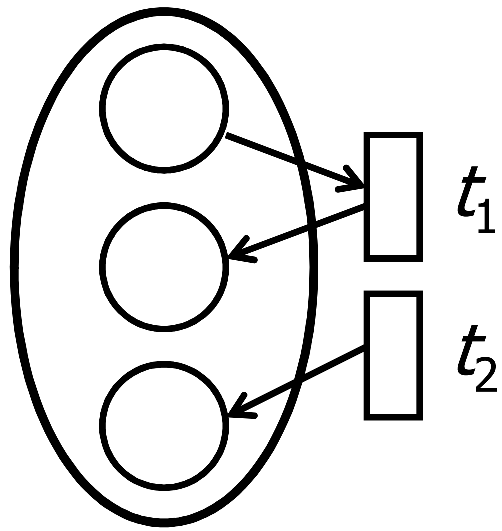
Egy Q helyhalmaz **csapda**, ha $Q\bullet \subseteq \bullet Q$

- minden Q-beli bemeneti helyhez kimeneti hely Q-ban
- egy állapotban jelölt $\rightarrow$ követő állapotokban is

$\bullet S = \{t_1\}$
$S\bullet = \{t_1, t_2\}$
$\bullet S \subseteq S\bullet$



$\bullet Q = \{t_1, t_2\}$
$Q\bullet = \{t_1\}$
$\bullet Q \supseteq Q\bullet$



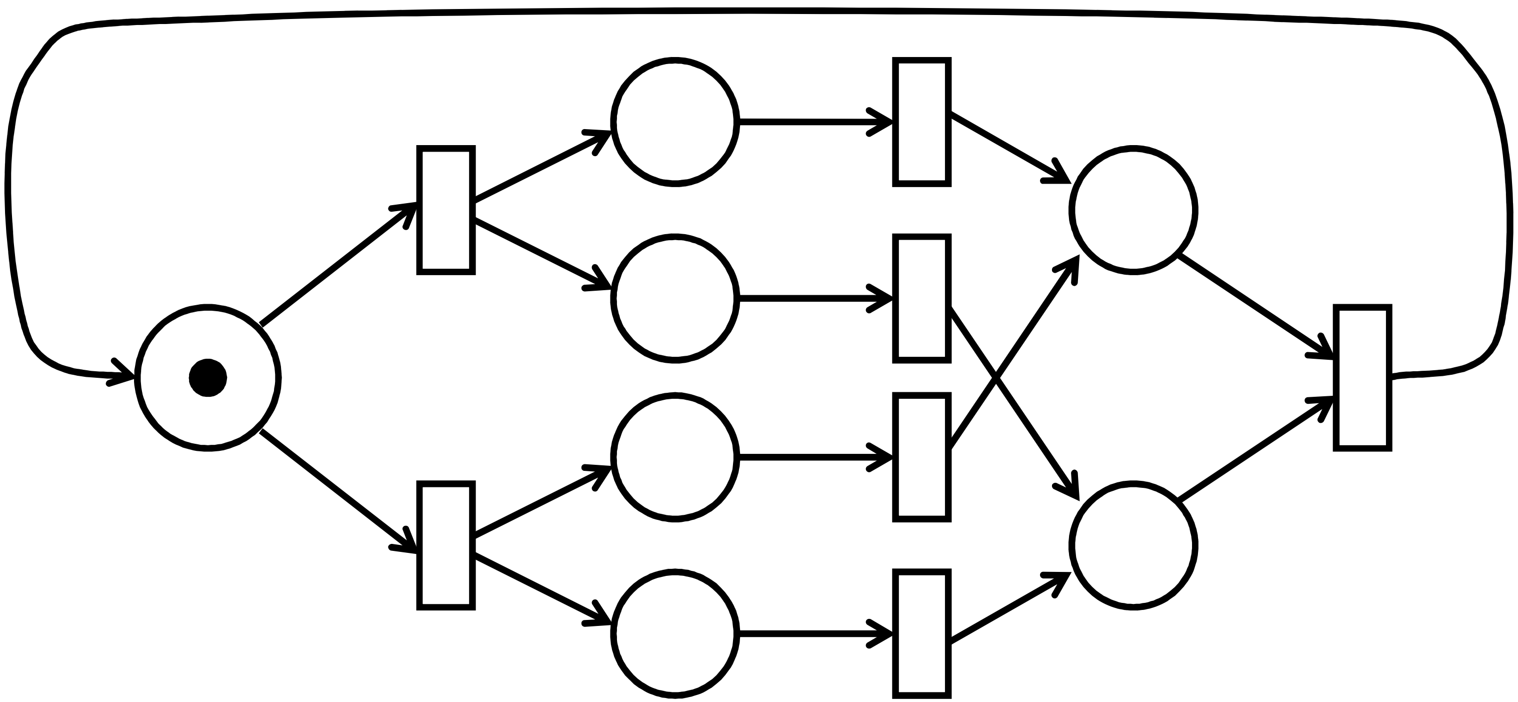
Élő és biztos tulajdonság az FC hálókban

10. Egy (N, M_0) szabad választású háló a.cs.a. **élő**, ha minden N–beli szifon tartalmaz jelölt csapdát
11. Egy élő (N, M_0) szabad választású háló a.cs.a. **biztos**, ha N lefedhető egy tokent tartalmazó erősen összekötött SM komponensekkel
12. Ha (N, M_0) élő és biztos szabad választású háló, akkor N lefedhető erősen összekötött MG komponensekkel. Létezik olyan $M \in R(N, M_0)$, hogy minden (N_1, M_1) komponens élő és biztos MG háló, ahol M_1 az M-nek N_1-re vett rész tokeneloszlása

Példa: élő és biztos FC háló

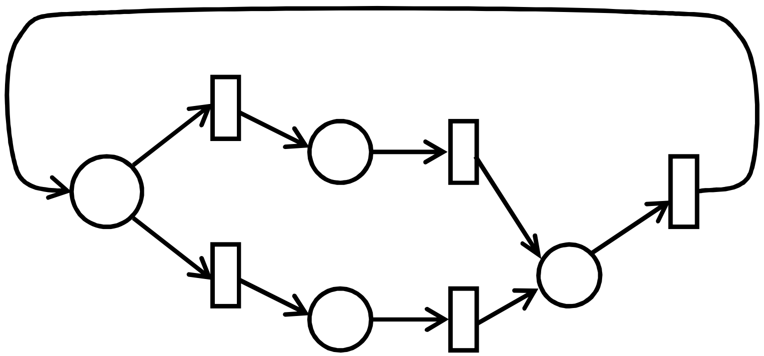
Élő és biztos a jelölt kezdőállapottal, mert

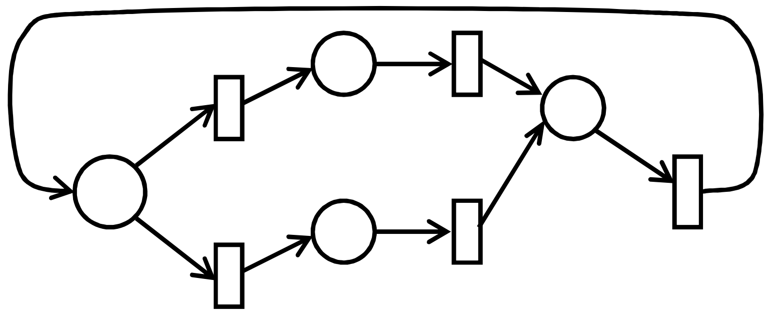
– minden SM komponens erősen összekötött, egy tokent tartalmaz, és a komponensek lefedik a teljes hálót
– minden MG komponens erősen összekötött, egy tokent tartalmaz, és a komponensek lefedik a teljes hálót



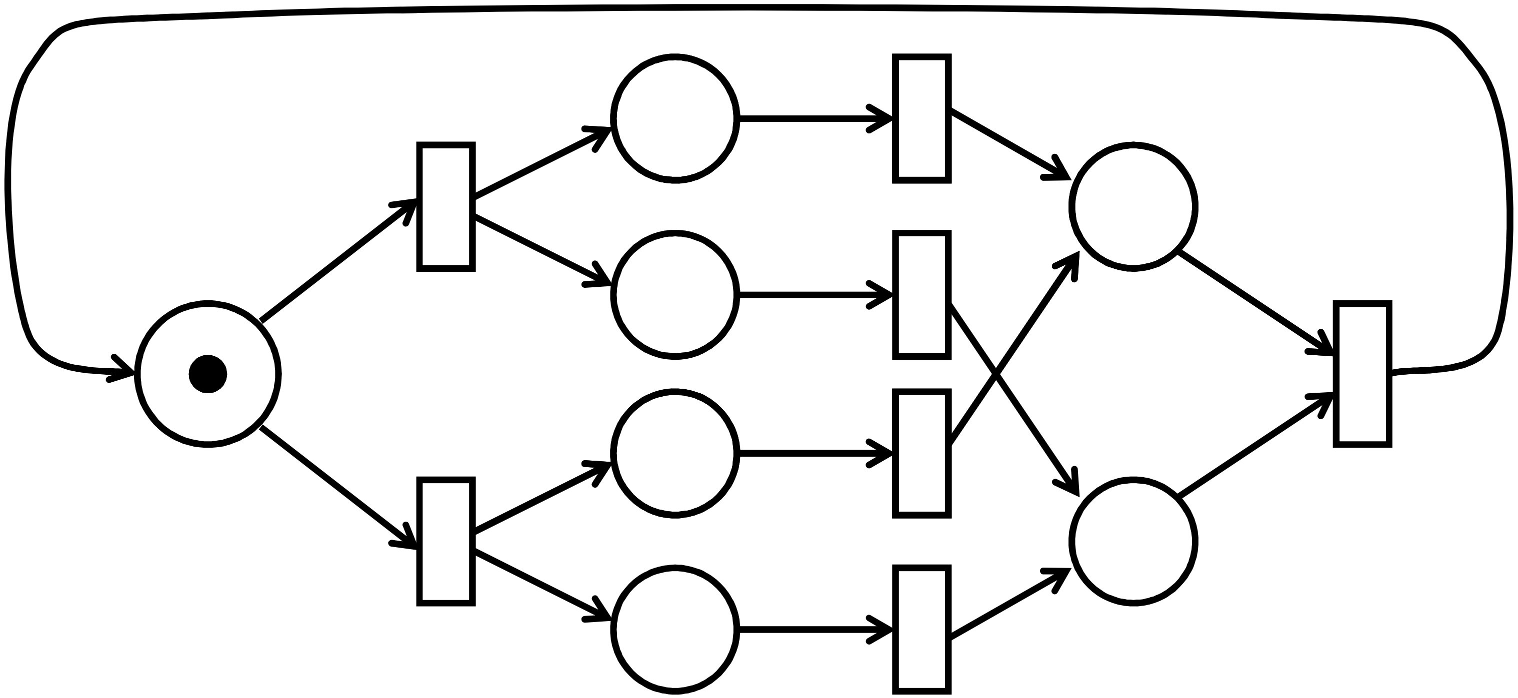
SM dekompozíció

- Allokáció: választás a szinkronizációban részt vevő tevékenységek közt
- Redukció menete
  1. Allokáció választása
  2. A nem allokált helyek törlése
  3. Minden kimeneti hely nélküli átmenet törlése
  4. Minden törölt kimeneti tüzeléssel bíró hely törlése



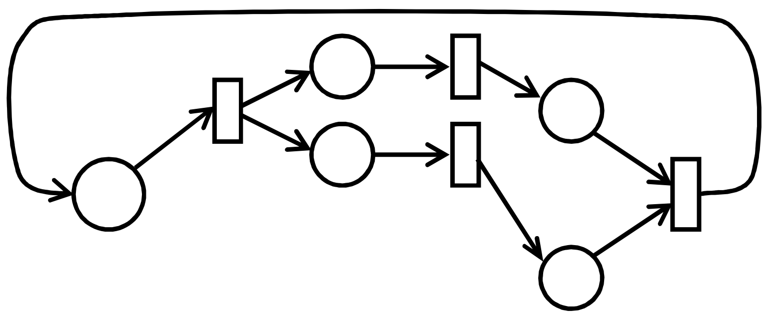


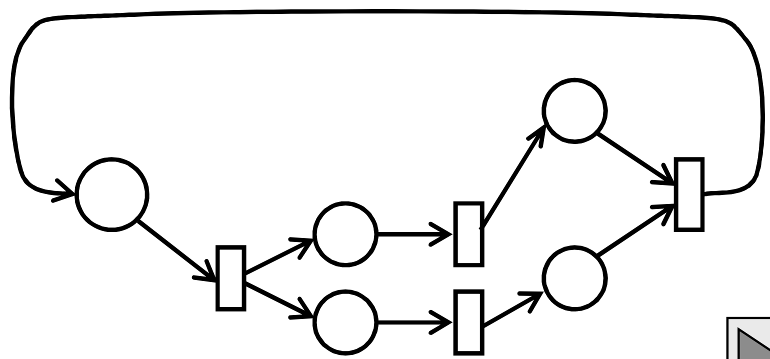
A példa FC háló SM dekompozíciója



MG dekompozíció

- Allokáció: választás a konfliktusban levő átmenetek közt
- Redukció menete
  1. Allokáció választása
  2. A nem allokált átmenetek törlése
  3. Minden bemeneti tüzelés nélküli hely törlése
  4. Minden törölt bemeneti hellyel bíró átmenet törlése





A példa FC háló MG dekompozíciója