# Petri hálók strukturális tulajdonságai
# Invariánsok és számításuk

dr. Bartha Tamás

Dr. Pataricza András

BME Méréstechnika és Információs Rendszerek Tanszék

# Vizsgálati lehetőségek

Az elemzés mélysége szerint:

- Szimuláció
- Állapottér bejárása
  - elérhetőségi gráf analízis
  - dinamikus (viselkedési) tulajdonságok
- Strukturális tulajdonságok
  - invariáns analízis

egy trajektória bejárása

minden trajektória bejárása
(kimerítő bejárás)

kezdőállapottól független
(bármely kezdőállapotra)

ha mindez nem vezet eredményre

- Algebrai közelítés, részleges döntés

# Petri hálók redukciós módszerei

# 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

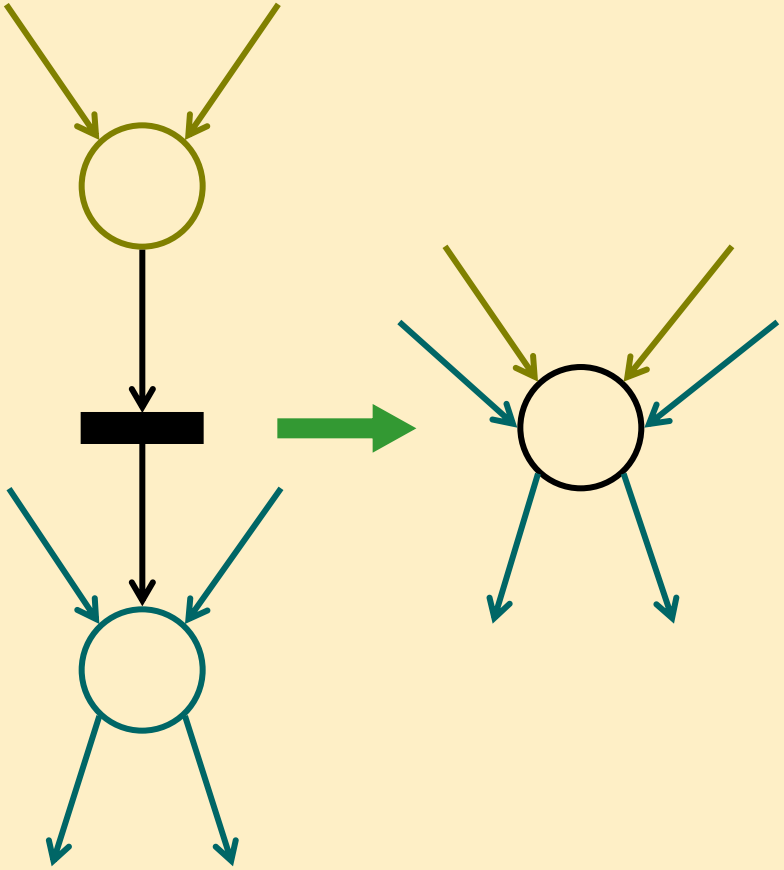
# Elérhetőségi probléma egyszerűsítése

- 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

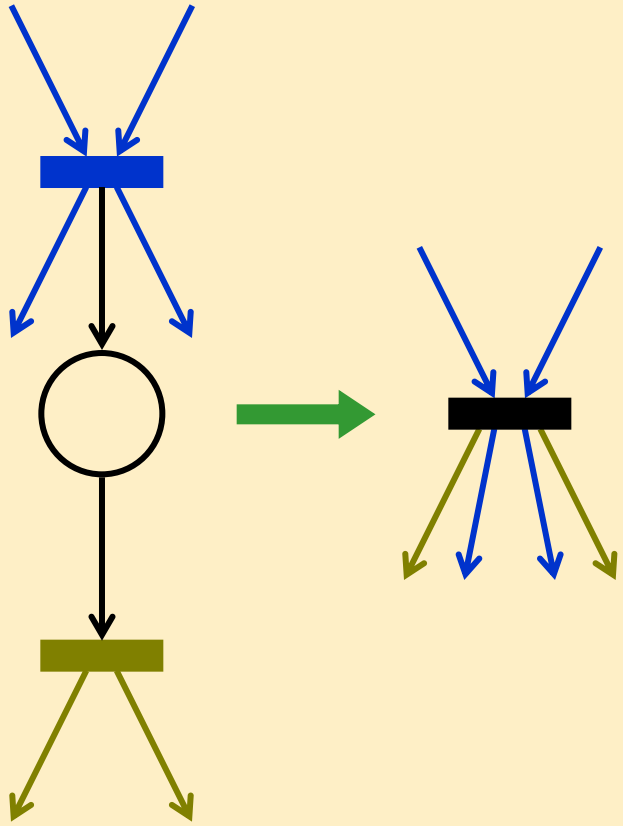
# Transzformációk

- 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

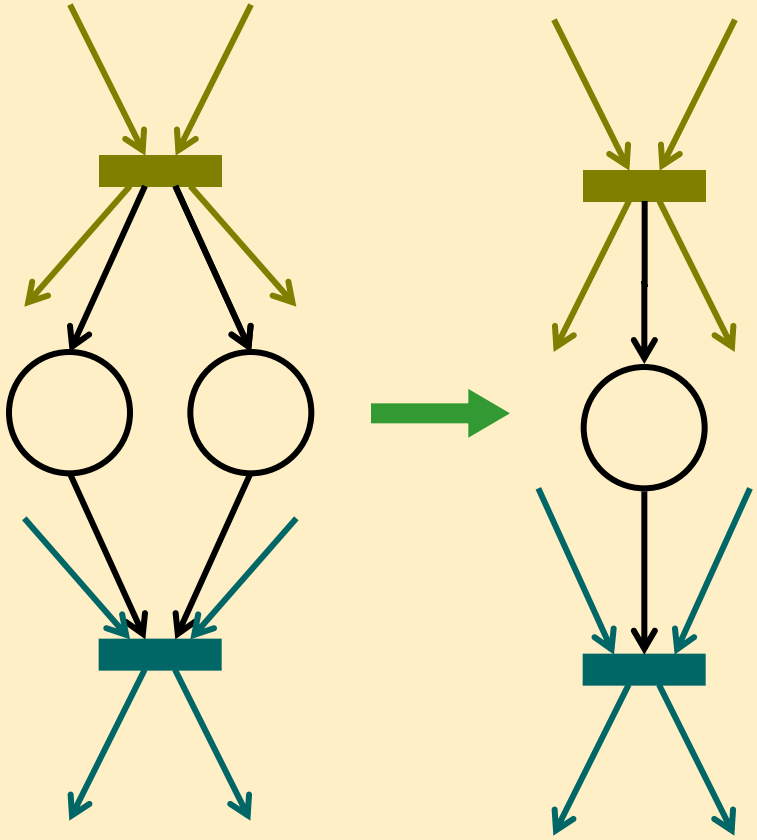


soros helyek összevonása (FSP)

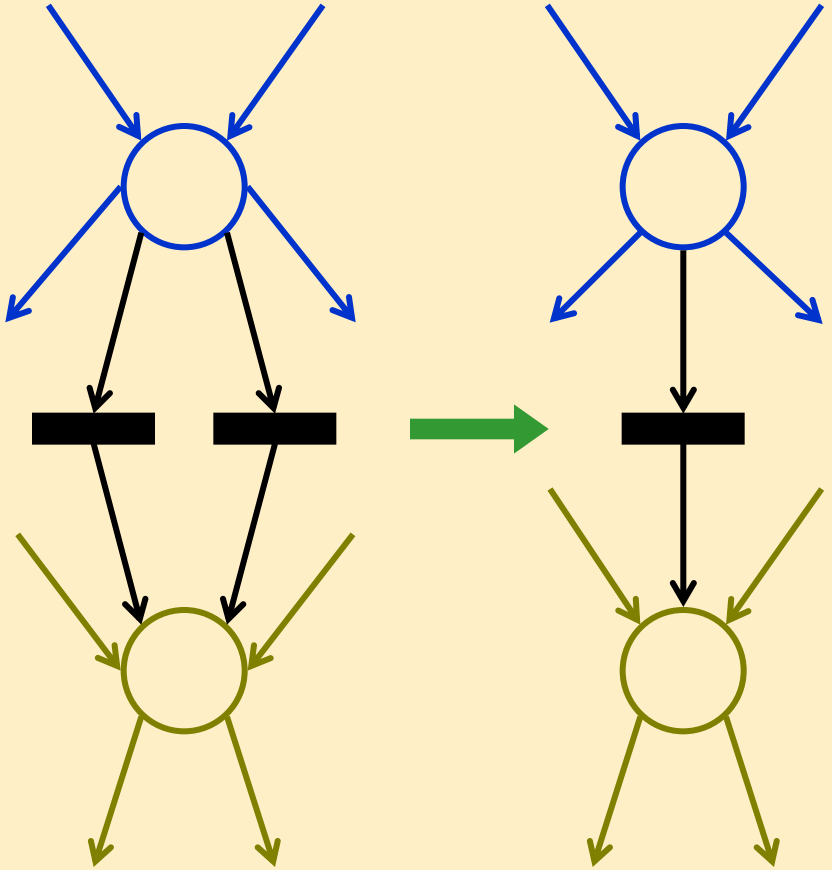


soros tranzíciók összevonása (FST)

# Párhuzamos összevonások

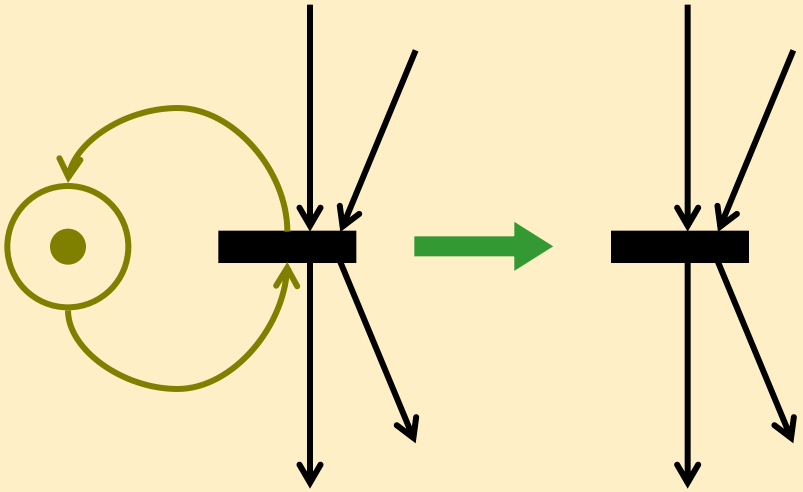


párhuzamos helyek összevonása (FPP)

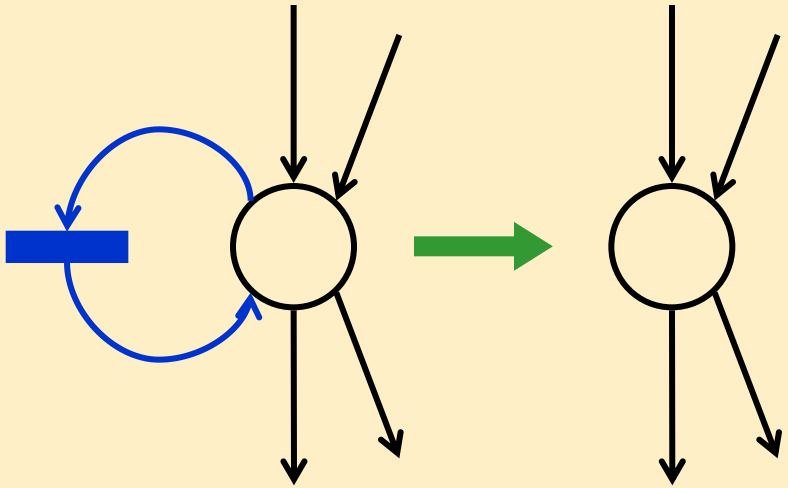


párhuzamos tranzíciók összevonása (FPT)

# Önhurkok törlése

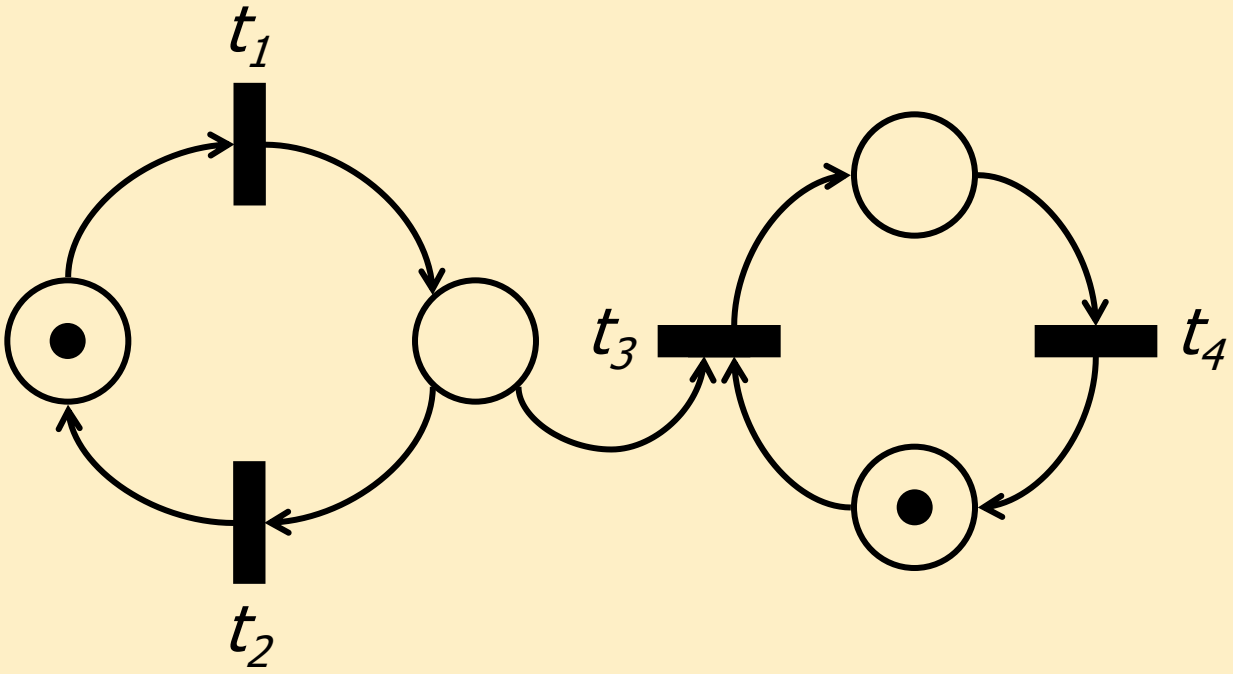


önhurkot alkotó helyek törlése (ESP)



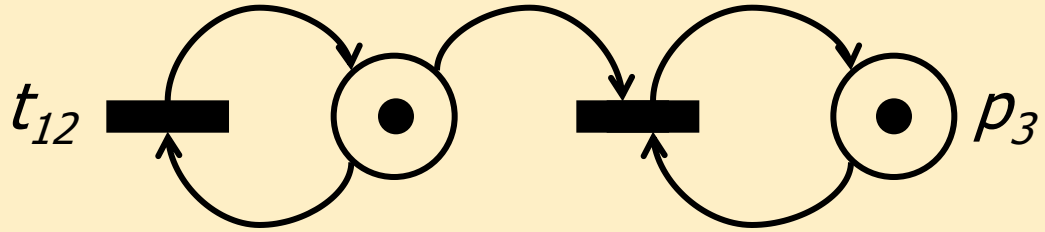
önhurkot alkotó tranzíciók törlése (EST)

# Példa



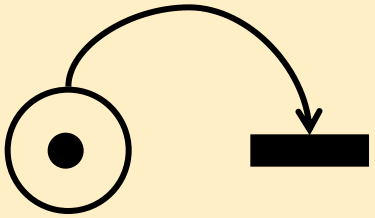

- $t_1$ tüzelése
- $t_1$ és $t_2$ összevonása (soros tranzíciók) $\rightarrow$ $t_{12}$
- $t_3$ és $t_4$ összevonása (soros tranzíciók) $\rightarrow$ $t_{34}$

# 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

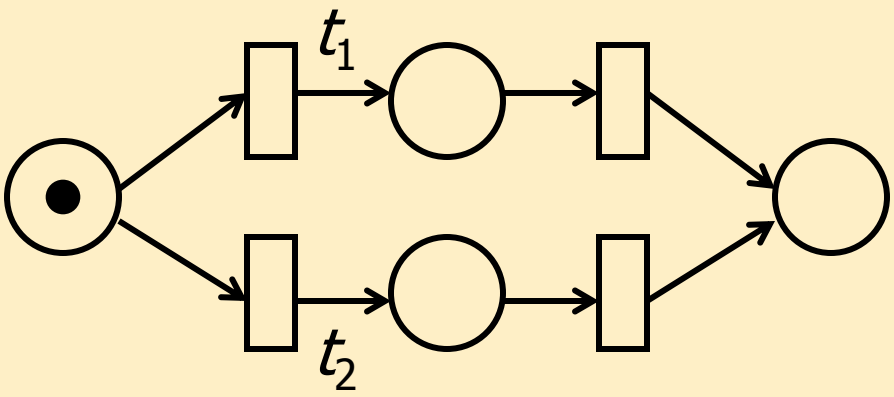
## Konfliktus: kizáró események, választás

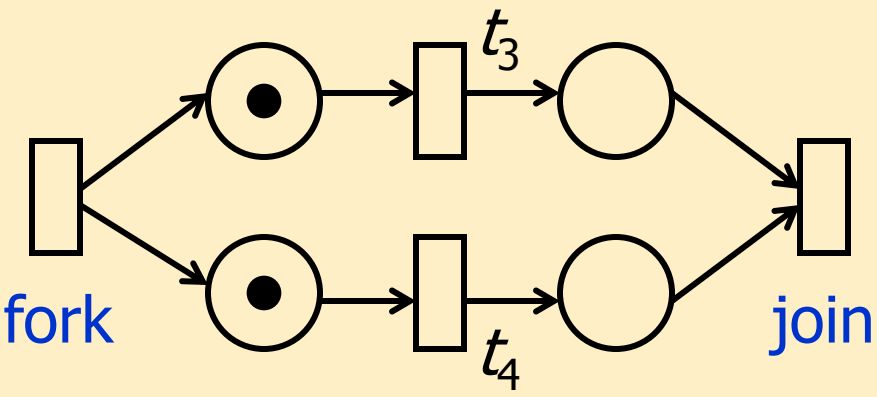
$e_1$ vagy $e_2$ esemény következik be, de csak az egyik

## 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 $e_3$

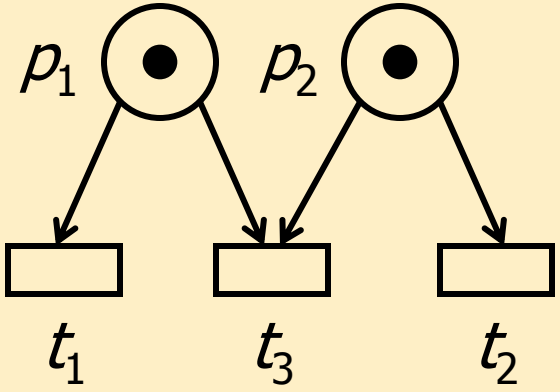


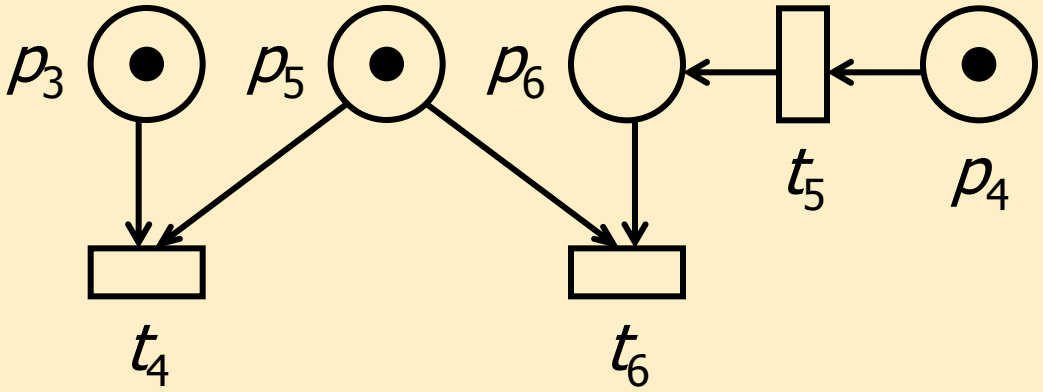



# Konfúzió

## 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

    $\forall t \in T : |\bullet t| = |t \bullet| = 1$

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

    $\forall p \in P : |\bullet p| = |p \bullet| = 1$

  - van szinkronizáció, nincs konfliktus

# Alosztályok (folyt.)

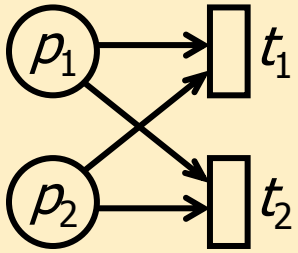
- Szabad választású háló (Free-Choice Net, FC)
  - közönséges Petri háló, melyben minden helyből kifelé mutató él vagy egyedüli kimenő él, vagy egyedüli bemenő él egy átmenetbe

$$\forall p \in P : |p\bullet| \geq 1 \Rightarrow \bullet(p\bullet) = \{p\},$$
$$\forall p_1, p_2 \in P : p_1\bullet \cap p_2\bullet \neq \emptyset \Rightarrow |p_1\bullet| = |p_2\bullet| = 1$$

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

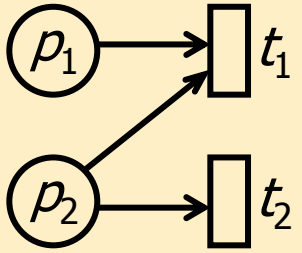
# Alosztályok (folyt.)

- 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

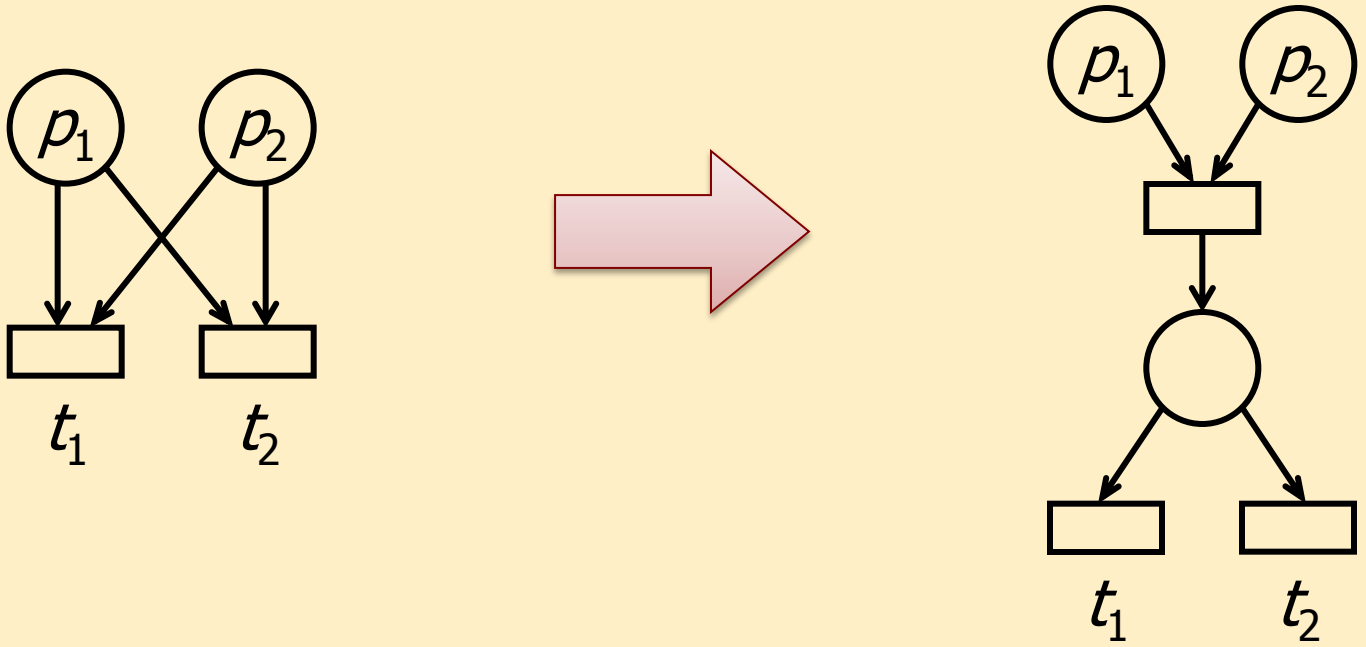



$$\forall p_1, p_2 \in P : p_1\bullet \cap p_2\bullet \neq \emptyset \Rightarrow p_1\bullet \subseteq p_2\bullet \vee p_2\bullet \subseteq p_1\bullet$$

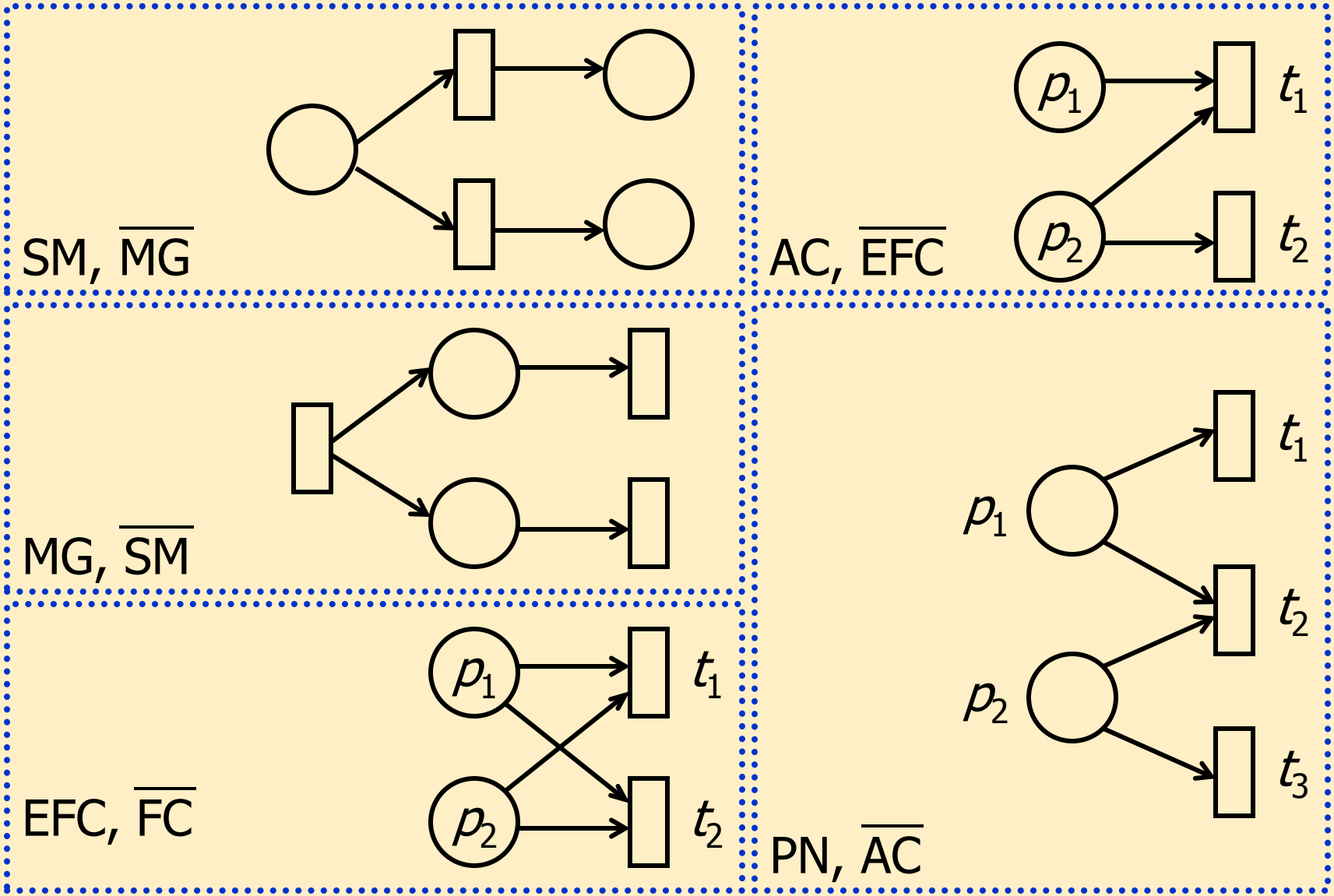
- Nincs szimmetrikus választású $\rightarrow$ a többi PN

# EFC transzformációja FC-re

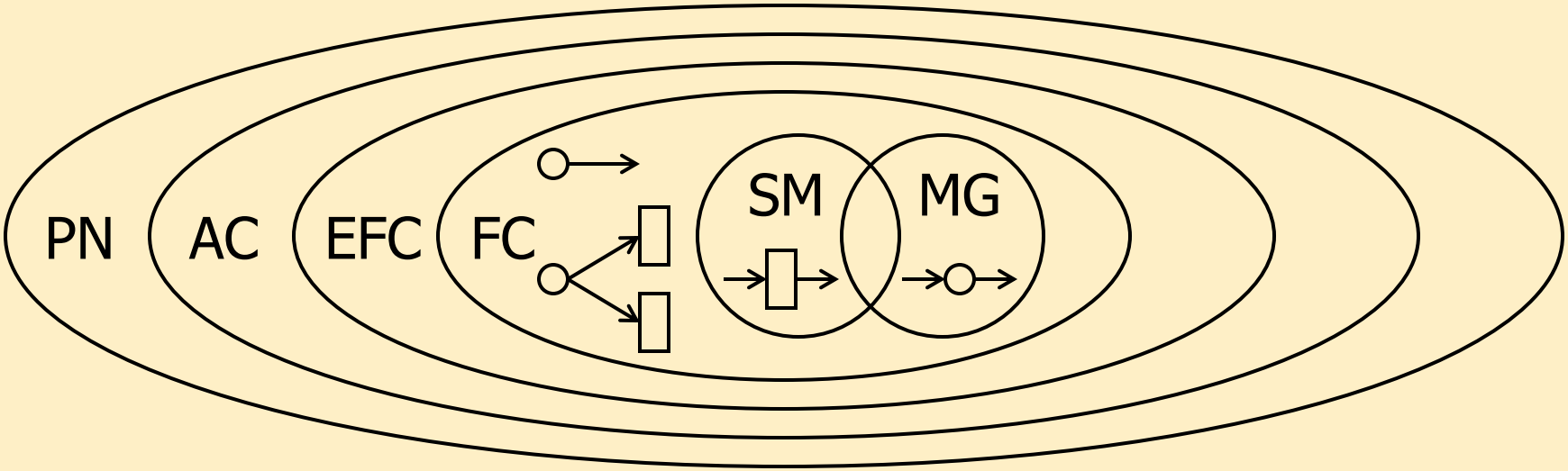
- FC és EFC közös tulajdonsága:
  - ha $t_1$ és $t_2$ $\exists$ közös bemeneti helye, akkor nincs olyan állapot, melyben az egyik engedélyezett és a másik nem
  - EFC transzformálható tulajdonságmegtartó módon FC-re



# 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

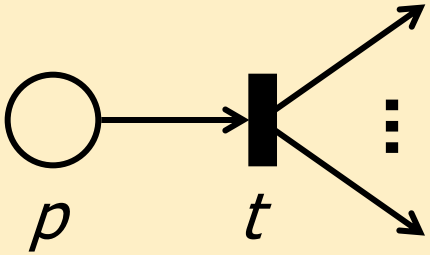
# Élő és bizt(onság)os tulajdonság kritériumai

## Szükséges feltétel általános esetben:

- Ha egy ($N$, $M_0$) Petri háló élő és 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

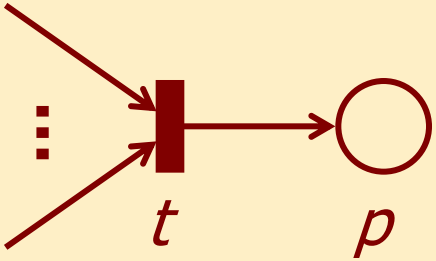
# Miért nem lehetnek forrás és nyelő csomópontok?

$\bullet p = \varnothing$
(forrás hely)



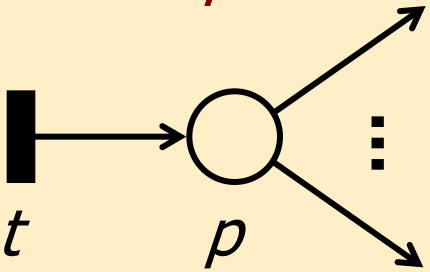

$t$ nem élő

$p\bullet = \varnothing$
(nyelő hely)



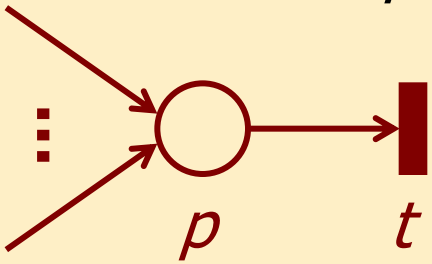

$p$ nem biztos, ha $t$ élő

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



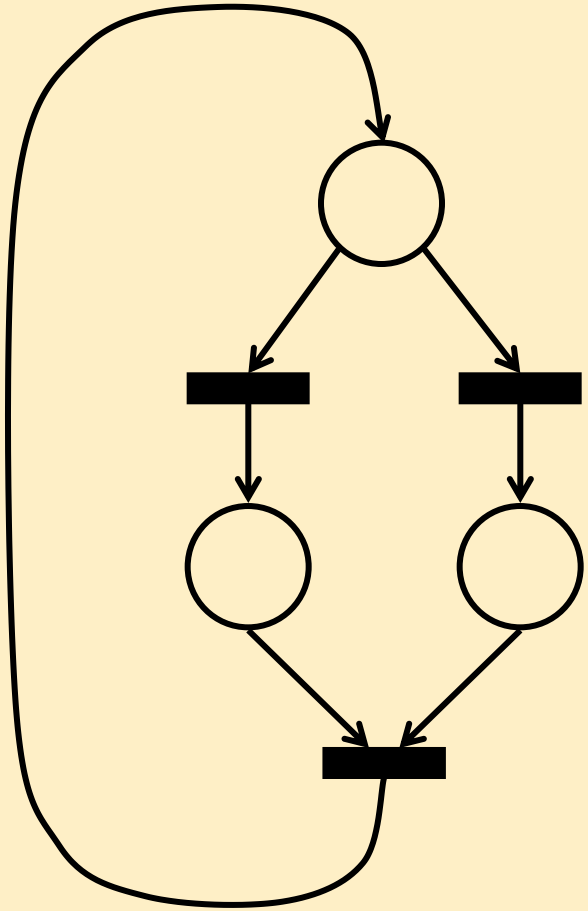

$p$ nem biztos

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

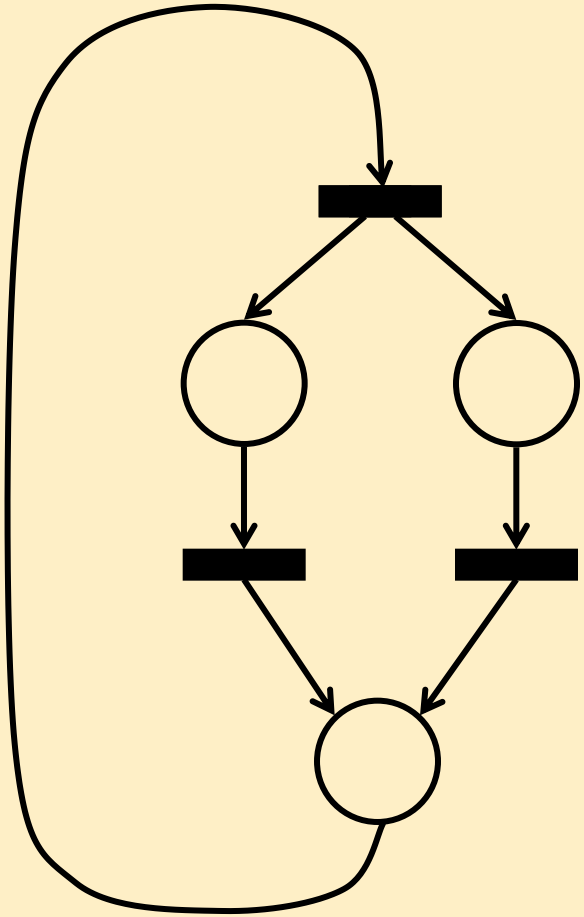



$t$ nem élő, ha $p$ biztos

# Miért nem elégséges az összekötöttség?



nincs élő jelölése

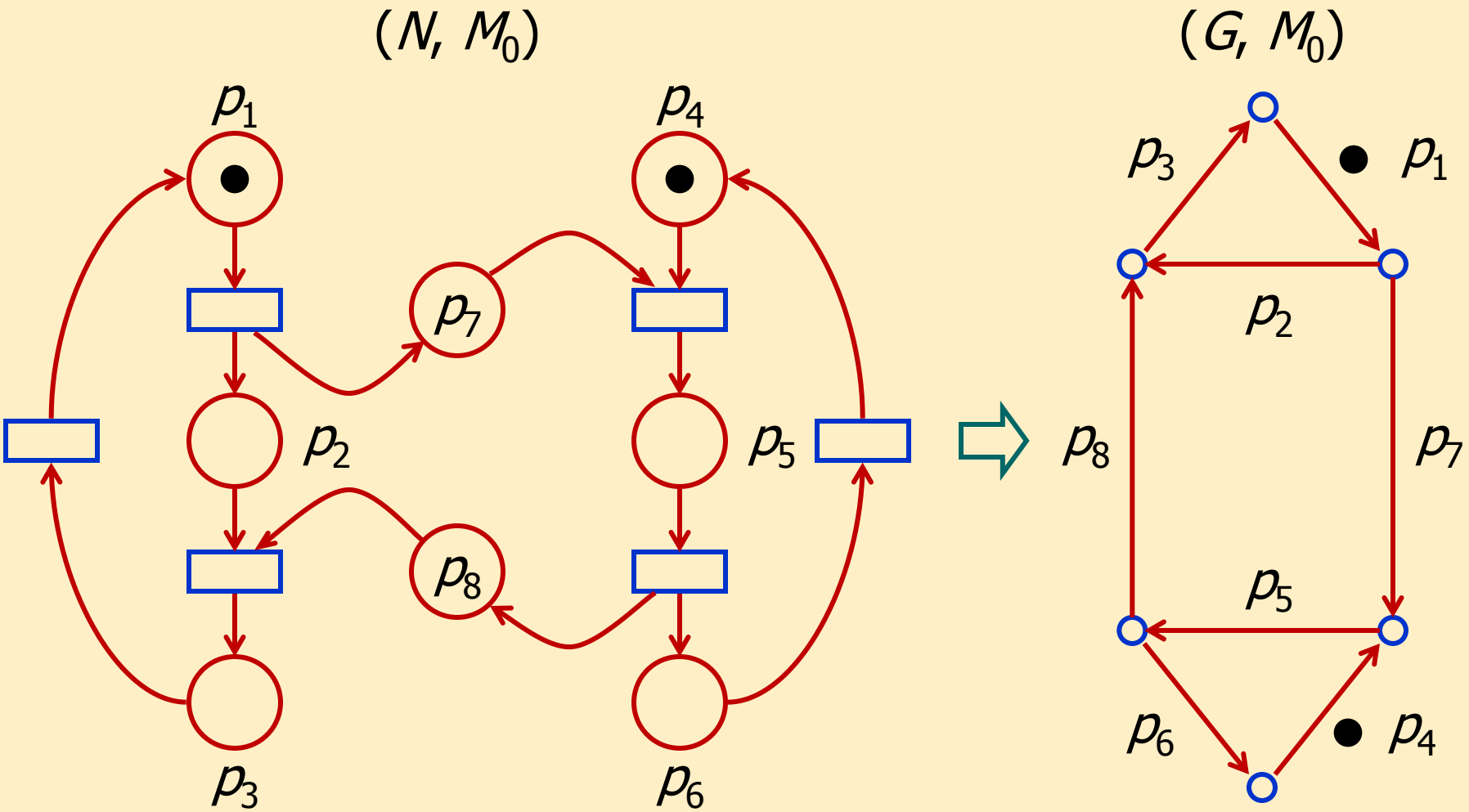


nincs nemüres biztos jelölése

# 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

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

   - 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

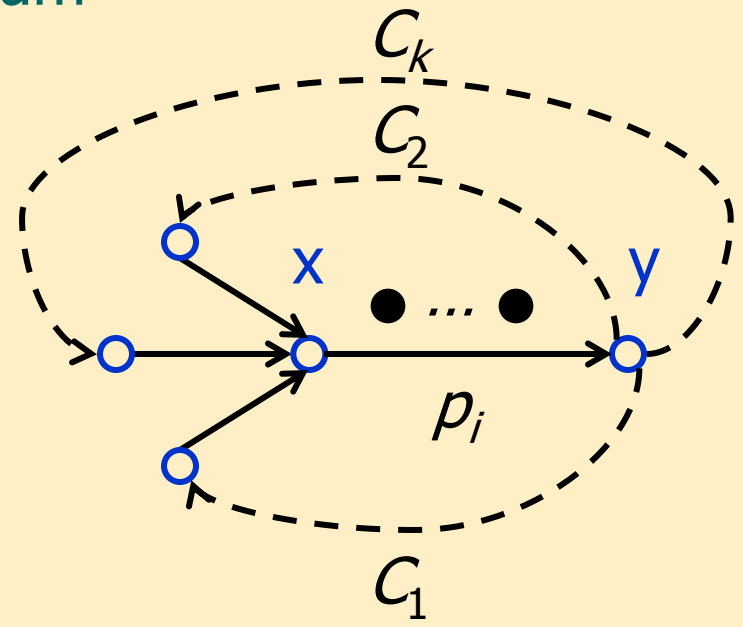
# Biztos tulajdonság az MG hálókban

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 erősen összekötött élő $(G, M_0)$ 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

# Szifon és csapda

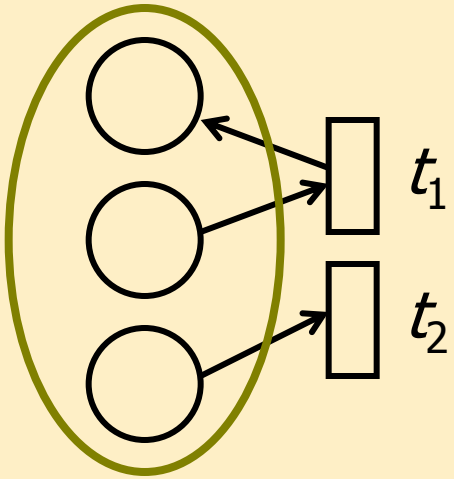
Egy S helyhalmaz **szifon**, ha $\bullet S \subseteq S\bullet$

- minden S-beli kimeneti helyhez bemeneti hely S-ben
- egy állapotban tokenmentes → követő állapotokban is

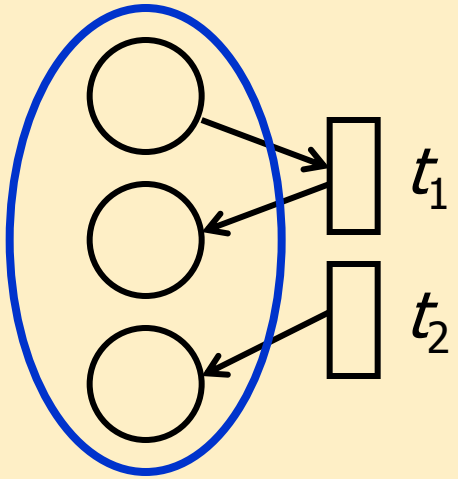
Egy Q helyhalmaz **csapda**, ha $Q\bullet \subseteq \bullet Q$

- minden Q-beli bemeneti helyhez kimeneti hely Q-ban
- egy állapotban jelölt → követő állapotokban is

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




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



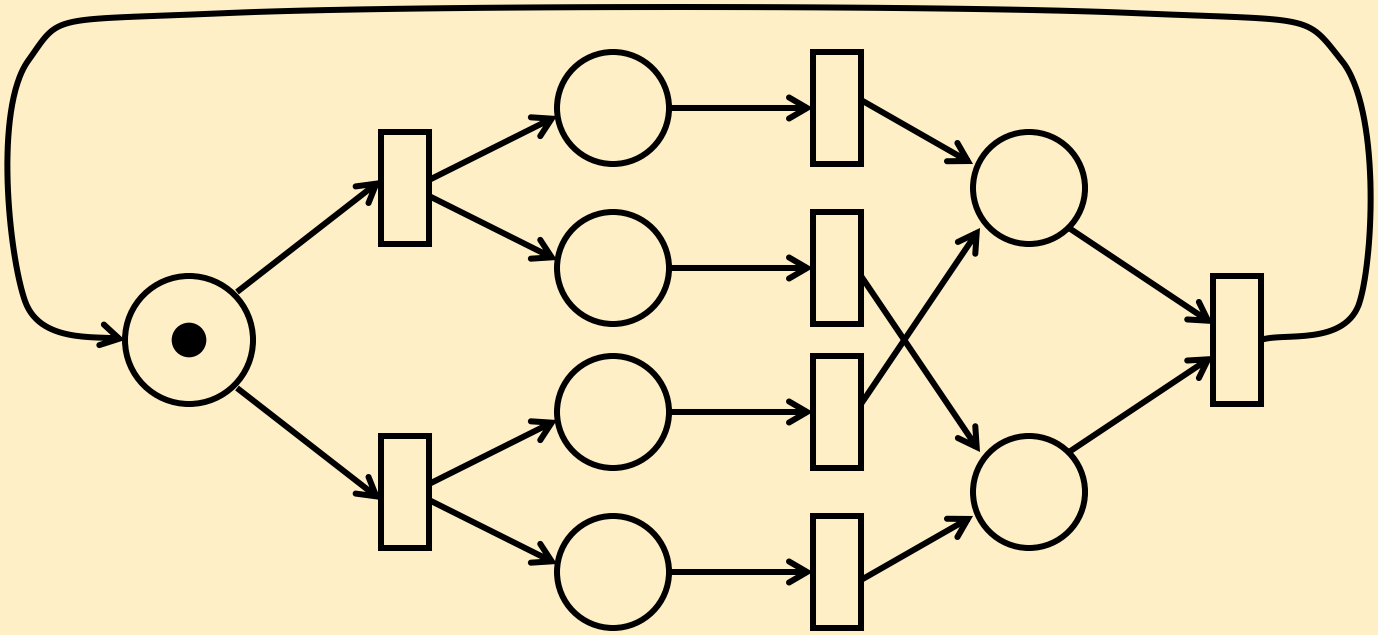
# Élő és biztos tulajdonság az FC hálókban

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

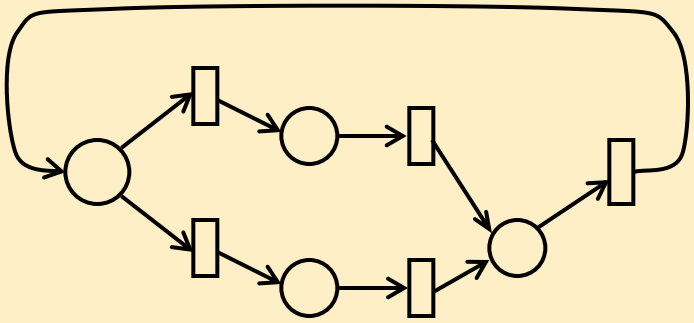
Élő és biztos a jelölt kezdőállapottal, mert

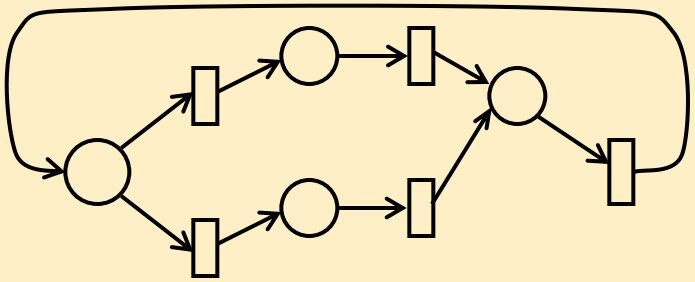
- minden SM komponens erősen összekötött, egy tokent tartalmaz, és a komponensek lefedik a teljes hálót
- minden MG komponens erősen összekötött, egy tokent tartalmaz, és a komponensek lefedik a teljes hálót



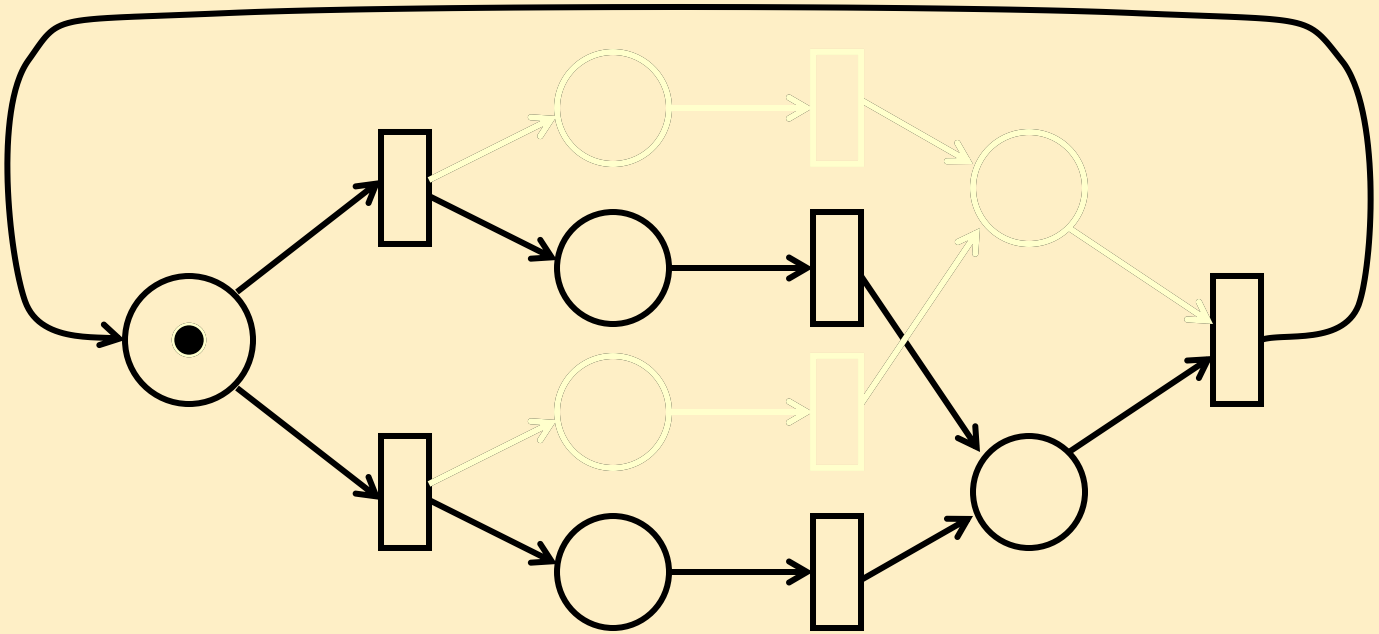
# SM dekompozíció

- 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



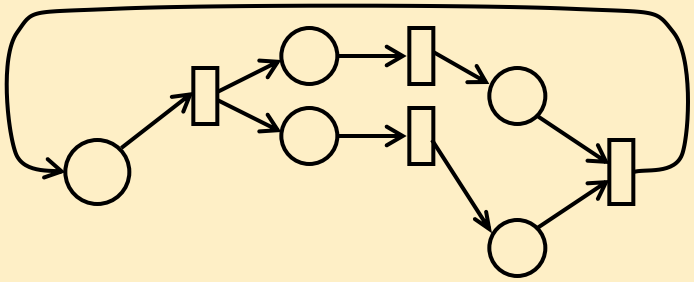


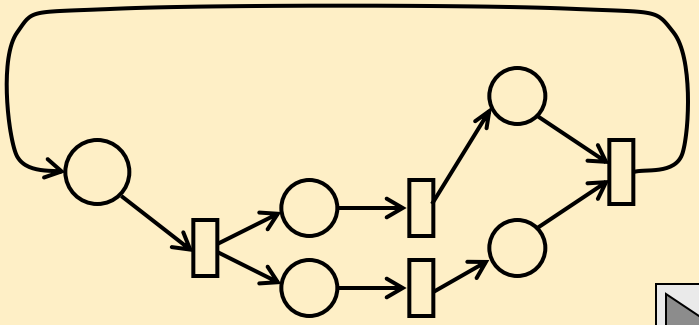
# A példa FC háló SM dekompozíciója



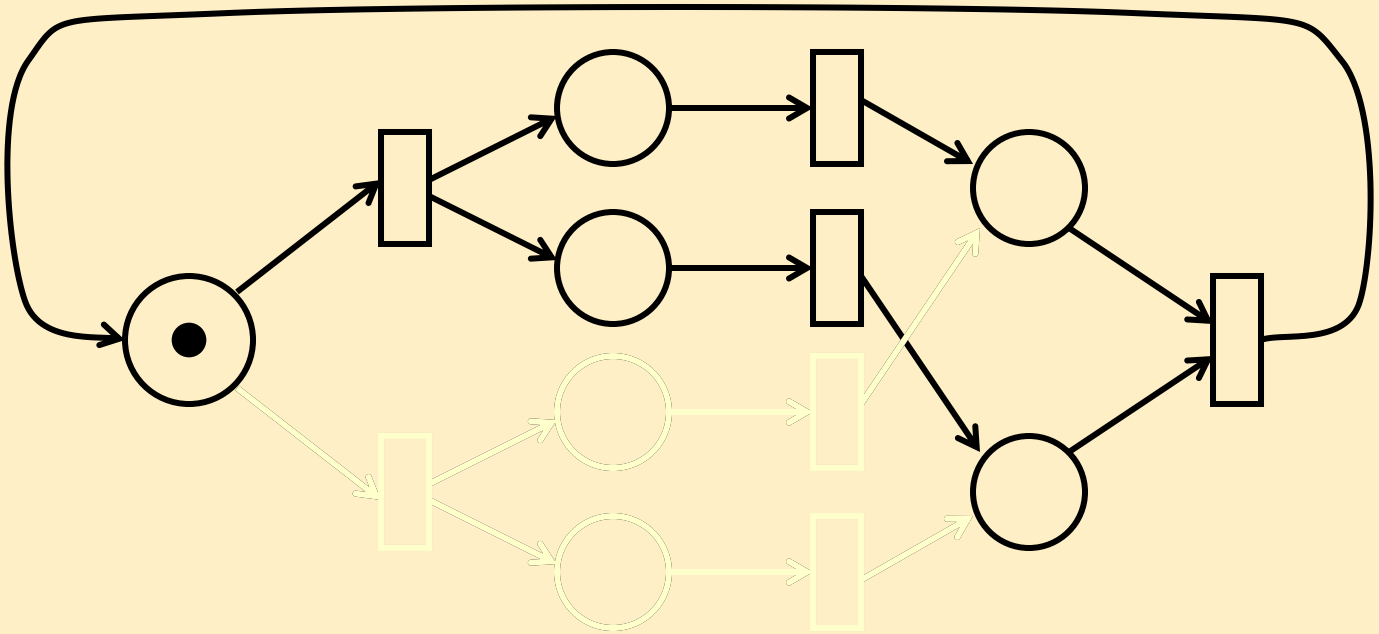
# MG dekompozíció

- 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



# Petri hálók strukturális tulajdonságai

# Strukturális tulajdonságok

## Petri hálók kezdőállapot-független tulajdonságai:

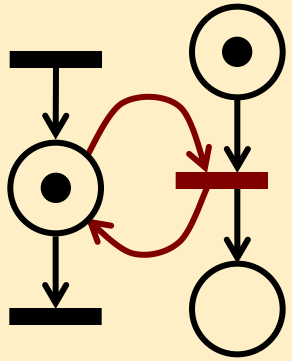
- Strukturális korlátosság
- Vezérelhetőség
- Konzervativitás
  - Hely invariáns,
    P- (place) invariáns
- Strukturális élőség
- Ismételhetőség
- Konzisztencia
  - Tüzelési invariáns,
    T- (transition) invariáns

Csak a háló struktúra határozza meg őket:

- vagy $\forall$ korlátos kezdő tokeneloszlásra igazak,
- vagy $\exists$ olyan korlátos kezdő tokeneloszlás, amire igazak

# Állapotegyenlet

- Petri háló dinamikája: tokenek áramlanak
  - Kirchhoff egyenletekhez hasonló egyensúlyi egyenletek
- Előfeltétel (egyértelműség): tiszta Petri háló
  - Nincs olyan átmenet, amely egyazon helynek egyszerre bemenő és kimenő átmenete: $\forall t \in T : {\bullet}t \cap t{\bullet} = \varnothing$
  - Hurokél
    - a tüzeléskor a tokeneloszlás nem változik, de
    - a tüzelési feltételben szerepet játszik



# Állapotegyenlet

- Tüzelési szekvencia: $M_{i_0}\left[\vec{\sigma} > M_{i_n}\right.$

$$\vec{\sigma} = \left\langle M_{i_0} t_{i_1} M_{i_1} \ldots t_{i_n} M_{i_n} \right\rangle = \left\langle t_{i_1} \ldots t_{i_n} \right\rangle$$

- **Engedélyezés:**
  - $t_{i,j}$ átmenet minden $p \in \bullet t_{i,j}$ bemenő helyén **elég** token

$$\forall t_{i_j} \in \vec{\sigma}, \forall p \in \bullet t_{i_j} : M_{i_{j-1}}(p) \geq w^{-}(p, t_{i_j}) = \mathbf{W}^{-^{\mathrm{T}}} \vec{e}_{i_j}$$

# Állapottrajektóriák

- **Állapotátmenet:**
  - $t_{i,j}$ átmenet engedélyezett $\rightarrow$ tüzel
    - minden $p \in \bullet t_{i,j}$ bemenő helyéről $w^{-}(p, t_{i,j})$ tokent vesz el
    - minden $p \in t_{i,j} \bullet$ kimenő helyére $w^{+}(p, t_{i,j})$ tokent tesz ki

$$M_{i_j} = M_{i_{j-1}} - \mathbf{W}^{-^{\mathrm{T}}} \vec{e}_{i_j} + \mathbf{W}^{+^{\mathrm{T}}} \vec{e}_{i_j} = M_{i_{j-1}} + \mathbf{W}^{\mathrm{T}} \vec{e}_{i_j}$$

    - összeadva és átrendezve:

$$M_{i_0} \left[ \vec{\sigma} > M_{i_n} \Rightarrow M_{i_n} - M_{i_0} = \mathbf{W}^{\mathrm{T}} \vec{\sigma}_T \right.$$

- **Tüzelési szám vektor**: az egyes tranzíciók tüzeléseinek száma a tüzelési szekvenciában

# Állapotegyenlet levezetése

$$M_1 = M_0 + \mathbf{W}^{\mathrm{T}}\vec{e}_{t_1}$$

$$M_2 = M_1 + \mathbf{W}^{\mathrm{T}}\vec{e}_{t_2} = \overbrace{M_0 + \mathbf{W}^{\mathrm{T}}\vec{e}_{t_1}}^{M_1 \text{ behelyettesítésével}} + \mathbf{W}^{\mathrm{T}}\vec{e}_{t_2}$$

$$\dots$$

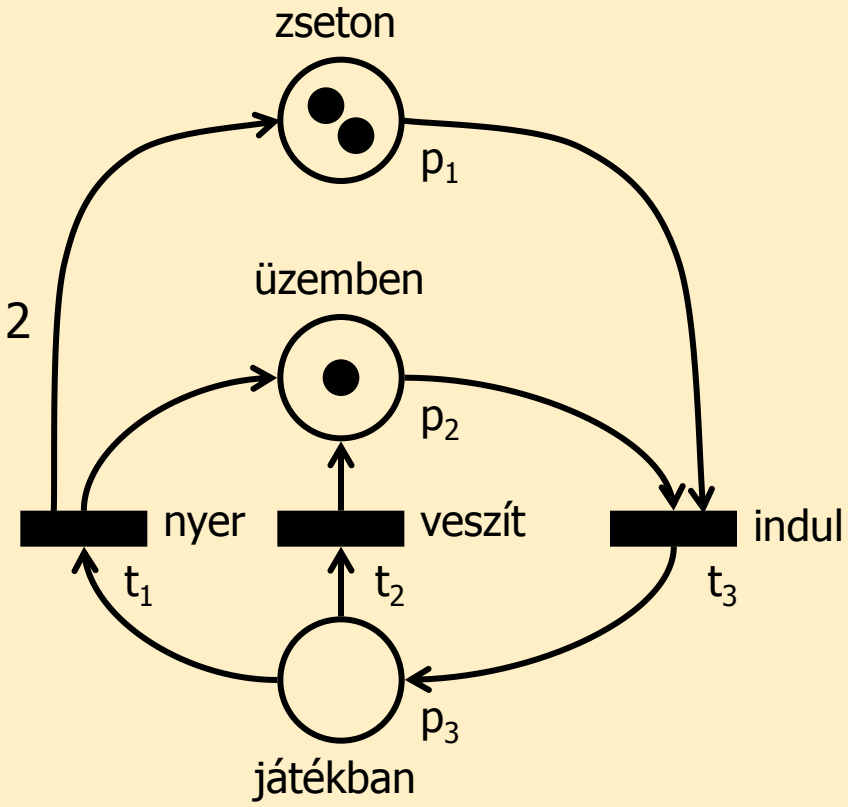
$$M_{n+1} = M_n + \mathbf{W}^{\mathrm{T}}\vec{e}_{t_{n+1}} = M_0 + \mathbf{W}^{\mathrm{T}}\vec{e}_{t_1} + \mathbf{W}^{\mathrm{T}}\vec{e}_{t_2} + \dots + \mathbf{W}^{\mathrm{T}}\vec{e}_{t_{n+1}}$$

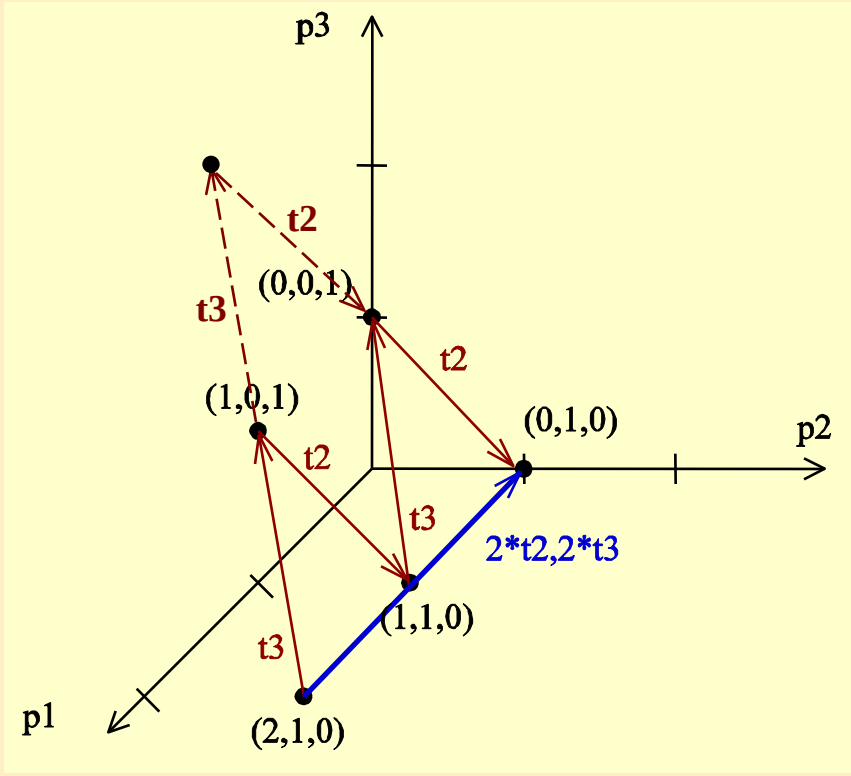
$$\dots$$

$$M_m = M_0 + \underbrace{\mathbf{W}^{\mathrm{T}}\vec{e}_{t_1} + \mathbf{W}^{\mathrm{T}}\vec{e}_{t_2} + \dots + \mathbf{W}^{\mathrm{T}}\vec{e}_{t_m}}_{\text{összevonva}} = M_0 + \mathbf{W}^{\mathrm{T}}\sum_{i=1}^{m}\vec{e}_{t_i}$$

$$M_m = M_0 + \mathbf{W}^{\mathrm{T}}\vec{\sigma}_T \Rightarrow \boxed{M_m - M_0 = \mathbf{W}^{\mathrm{T}}\vec{\sigma}_T}$$

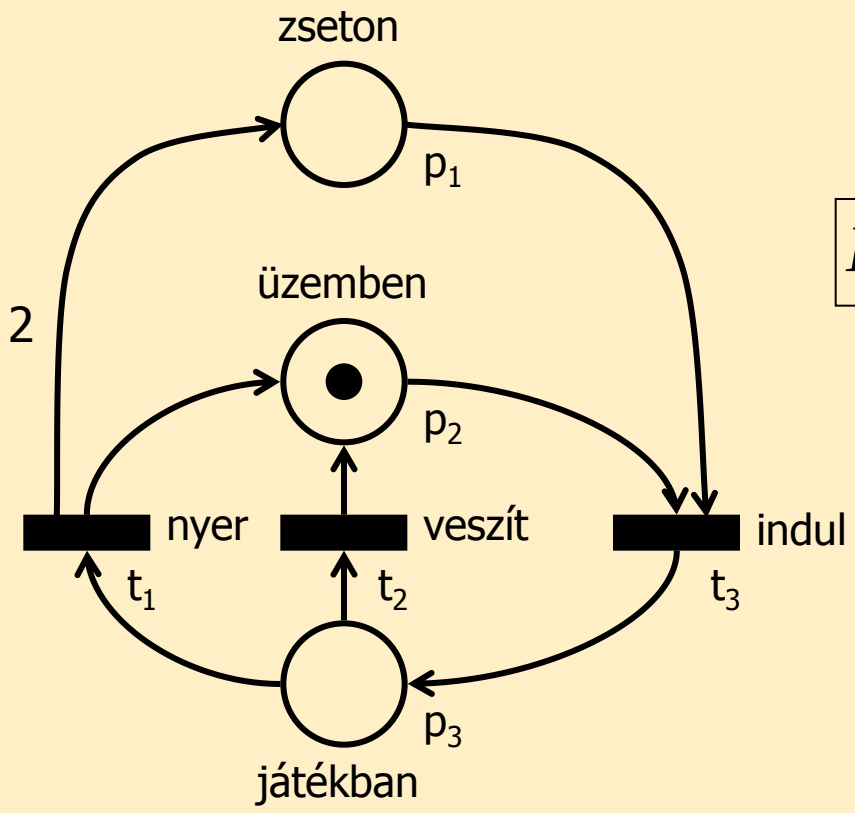
# Állapotegyenlet és elérhetőség




- A **tüzelési szám vektor**ban kevesebb az információ, mint a **tüzelési szekvenciá**ban!

# Állapotegyenlet és elérhetőség




- Tokenmérleg teljesülése a tüzelésnek csak a **szükséges** feltétele!

$$M_{i_0}\left[\vec{\sigma} > M_{i_n} \Rightarrow M_{i_n} - M_{i_0} = \mathbf{W}^{\mathrm{T}}\vec{\sigma}_T\right.$$

$$\mathbf{W}^{\mathrm{T}} = \begin{array}{c} \\ \mathbf{p_1} \\ \mathbf{p_2} \\ \mathbf{p_3} \end{array} \begin{array}{c} \begin{array}{ccc} \mathbf{t_1} & \mathbf{t_2} & \mathbf{t_3} \end{array} \\ \begin{pmatrix} \mathbf{2} & \mathbf{0} & \mathbf{-1} \\ \mathbf{1} & \mathbf{1} & \mathbf{-1} \\ \mathbf{-1} & \mathbf{-1} & \mathbf{1} \end{pmatrix} \end{array}$$

$$(1,1,0)^{\mathrm{T}} - (0,1,0)^{\mathrm{T}} = \mathbf{W}^{\mathrm{T}} \cdot (1,0,1)^{\mathrm{T}}$$

$t_1$ és $t_3$ sem nem tüzelhető a (0,1,0) kezdőállapotban!

# Invariánsok fogalma
# Tüzelési és hely invariáns

# Tüzelési, avagy T-invariáns

A $\sigma$ tüzelési szekvencia végrehajtása nem változtatja meg a tokeneloszlást:

$$\mathbf{W}^{\mathrm{T}}\vec{\sigma}_T = 0$$

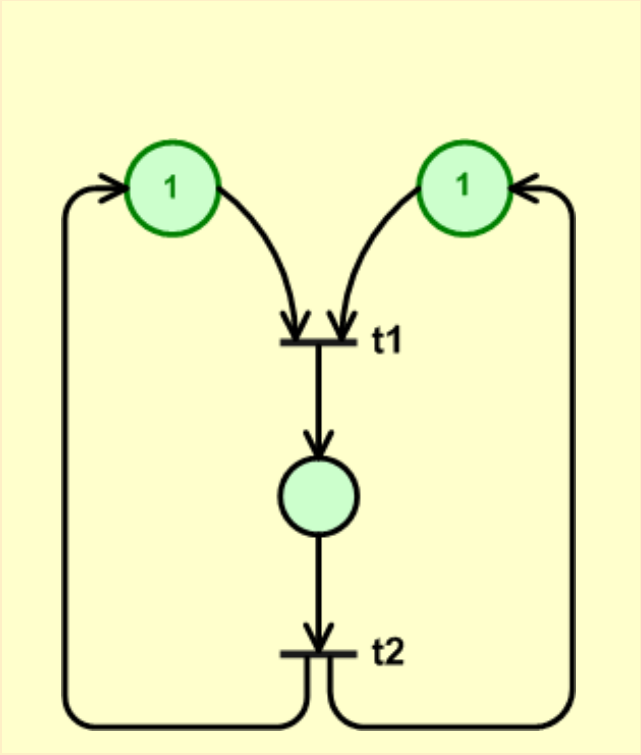
- Ciklus az állapottérben: $M_i \left[\vec{\sigma}_T > M_i\right.$
  - ha $\sigma_T$ szekvencia az $M_i$ állapotból végrehajtható!

$$\forall t_{i_j} \in \vec{\sigma}, \forall p \in \left\{\bullet t_{i_j}\right\} : m_{i_{j-1}}\left(p\right) \geq w^{-}\left(p, t_{i_j}\right) = \mathbf{W}^{-\mathrm{T}} \cdot \vec{e}_{i_j}$$

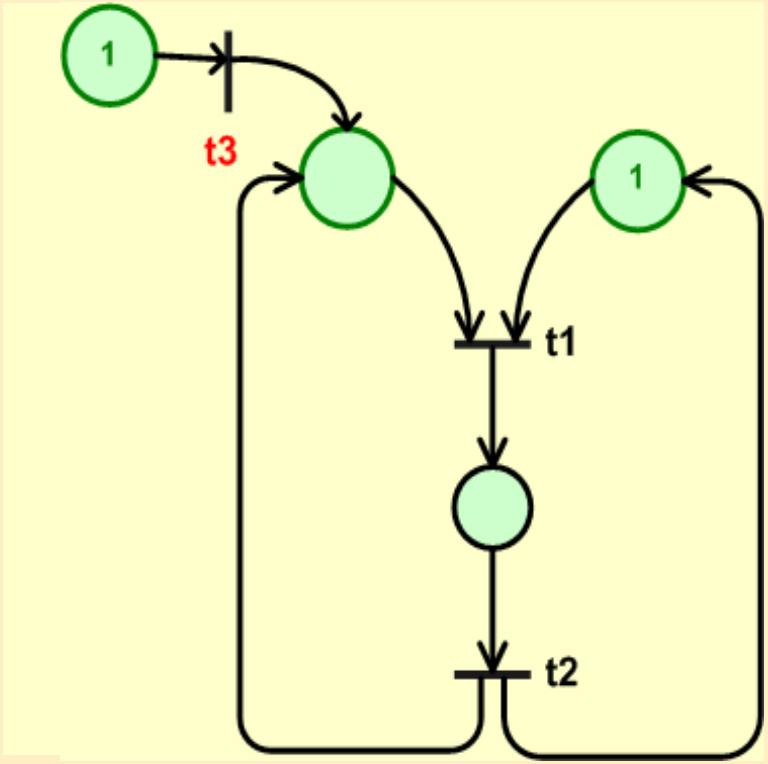
- Megjegyzés: bármely $\sigma$ tüzelési szekvenciához található olyan $M_0$ kezdőállapot, amelyből $\sigma$ végrehajtható
  - Pl. $M_0 \geq \mathbf{W}^{-\mathrm{T}}\vec{\sigma}$ esetén induláskor annyira „teletömött”, hogy a $\sigma$ tüzelési szekvencia által termelt tokenekre már nincs szükség!

# Példa T-invariánsra

$t_1 – t_2$ után a tokeneloszlás ugyanez




$t_3 – t_1 – t_2$ tüzelési szekvencia nem ismételhető



# T-invariánsok halmaza

$$\mathbf{W}^{\mathrm{T}}\vec{\sigma}_T = 0$$

**Homogén, lineáris** egyenletrendszer

- egy megoldás többszöröse is megoldás
  - ha tüzelhető, akkor többször is befutja a ciklust
- megoldás összege is megoldás
  - ha tüzelhető, akkor több ciklus kombinációját futja be
- megoldások lineáris kombinációi is megoldások

Keressünk **BÁZIST**!

- az összes megoldást előállító minimális halmaz

# Minimális T-invariáns

- A $\sigma$ tüzelési szekvencia sup($\sigma$) **alapja**
  - azon átmenetek $T' = \{t_i \mid \sigma_i > 0\}$ részhalmaza, amelyek $\sigma$ szekvenciában előfordulnak
- A $\sigma_T$ tüzelési invariáns **minimális alapú**
  - ha nincs olyan T-invariáns, amelynek alapja $\sigma_T$ alapjának valódi részhalmaza, vagy
  - ha részhalmaza azonos, annak tüzelési számai kisebbek

$$\forall \sigma_T^1 : \mathbf{W}^{\mathrm{T}} \sigma_T^1 = 0 \Rightarrow \left(\sigma_T^1 \geq \sigma_T\right) \vee \left(\mathrm{sup}(\sigma_T) \not\subseteq \mathrm{sup}(\sigma_T^1)\right)$$

# Hely, avagy P-invariáns

A $\mu_P$ súlyvektor által kijelölt helyeken a tokenek súlyozott összege nem változik:

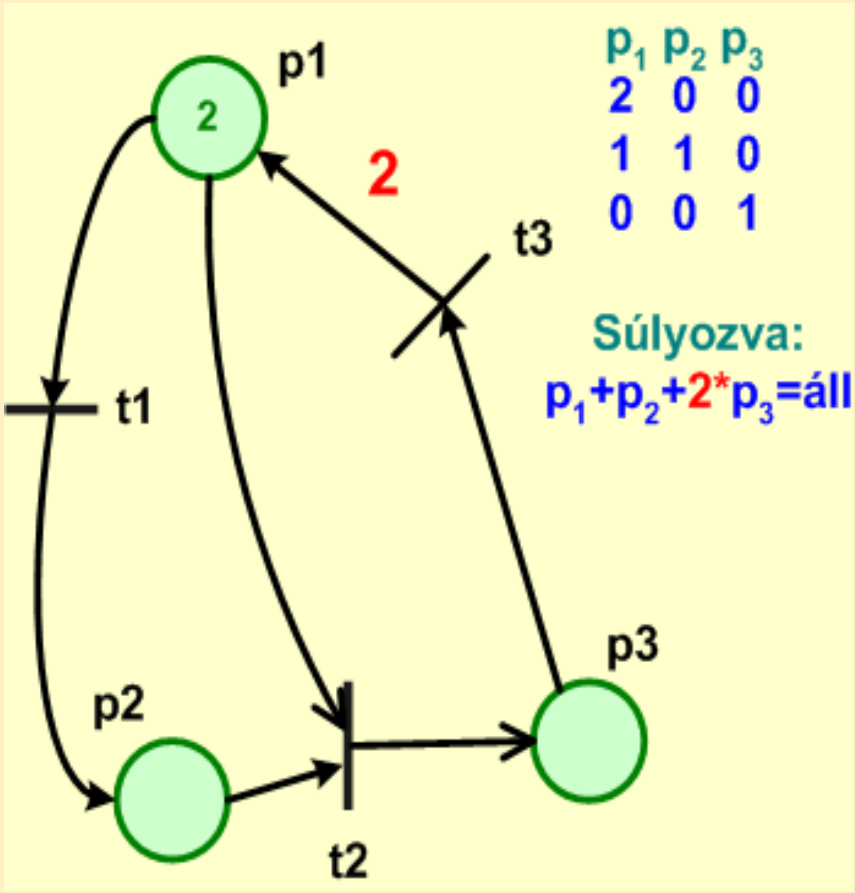
$$\boxed{\vec{\mu}_P^{\mathrm{T}} M = \text{állandó}}$$

– A tokenek (egy része) a helyek egy részhalmazában kering (pl. erőforrások nem fogynak, nem keletkeznek)

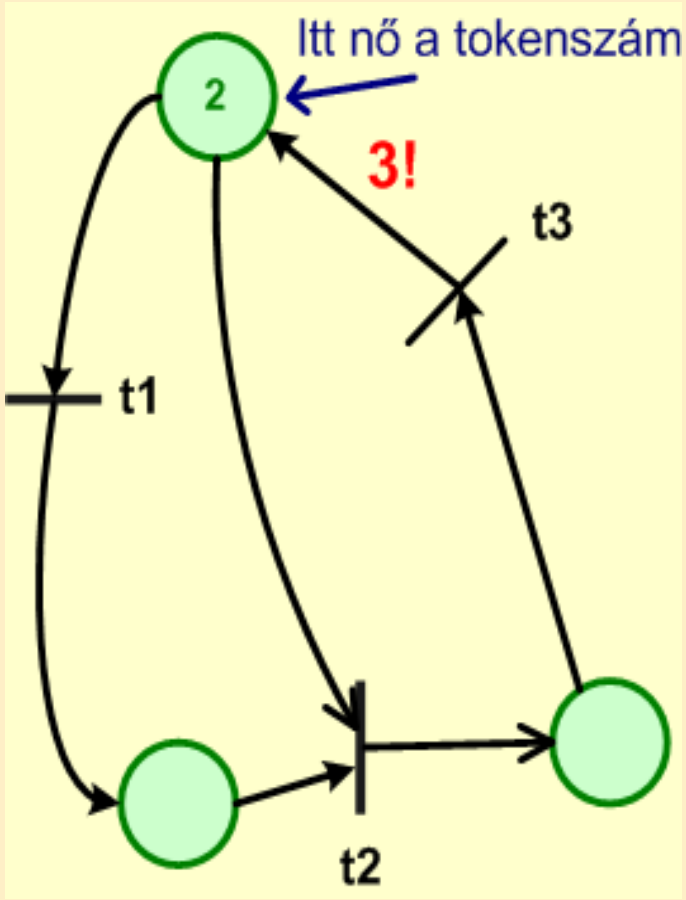
$$\left.\begin{aligned} M &= M_0 + \mathbf{W}^{\mathrm{T}}\vec{\sigma} \\ \vec{\mu}_P^{\mathrm{T}} M &= \underbrace{\vec{\mu}_P^{\mathrm{T}} M_0}_{\vec{\mu}_P^{\mathrm{T}} M = \vec{\mu}_P^{\mathrm{T}} M_0 = \text{állandó}} + \vec{\mu}_P^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\vec{\sigma} \\ \vec{\mu}_P^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\vec{\sigma} = 0 &\Rightarrow \vec{\mu}_P^{\mathrm{T}}\mathbf{W}^{\mathrm{T}} \underset{\forall\vec{\sigma}}{\equiv} 0 \end{aligned}\right\} \quad \boxed{\mathbf{W}\vec{\mu}_P = 0}$$

# Példa P-invariánsra

P-invariáns $p_1$, $p_2$, $p_3$-ra:




NEM P-invariáns:



# Invariánsok alkalmazásai

- T-invariánsok alkalmazásai
  - Termelési folyamat meghatározása
  - Szabályalapú rendszerek, diagnosztikai problémák
  - Dinamikus tulajdonságok
    - ciklikusan tüzelhető ➲ megfordíthatóság, visszatérő állapot
    - később is tüzelhető ➲ élő tulajdonság, holtpontmentesség
- P-invariánsok alkalmazásai
  - Véges automaták keresése $\rightarrow$ dekompozíció
  - Dinamikus tulajdonságok
    - token nem vész el ➲ élő tulajdonság, holtpontmentesség
    - token nem termelődik ➲ korlátosság

# Invariánsok számítása

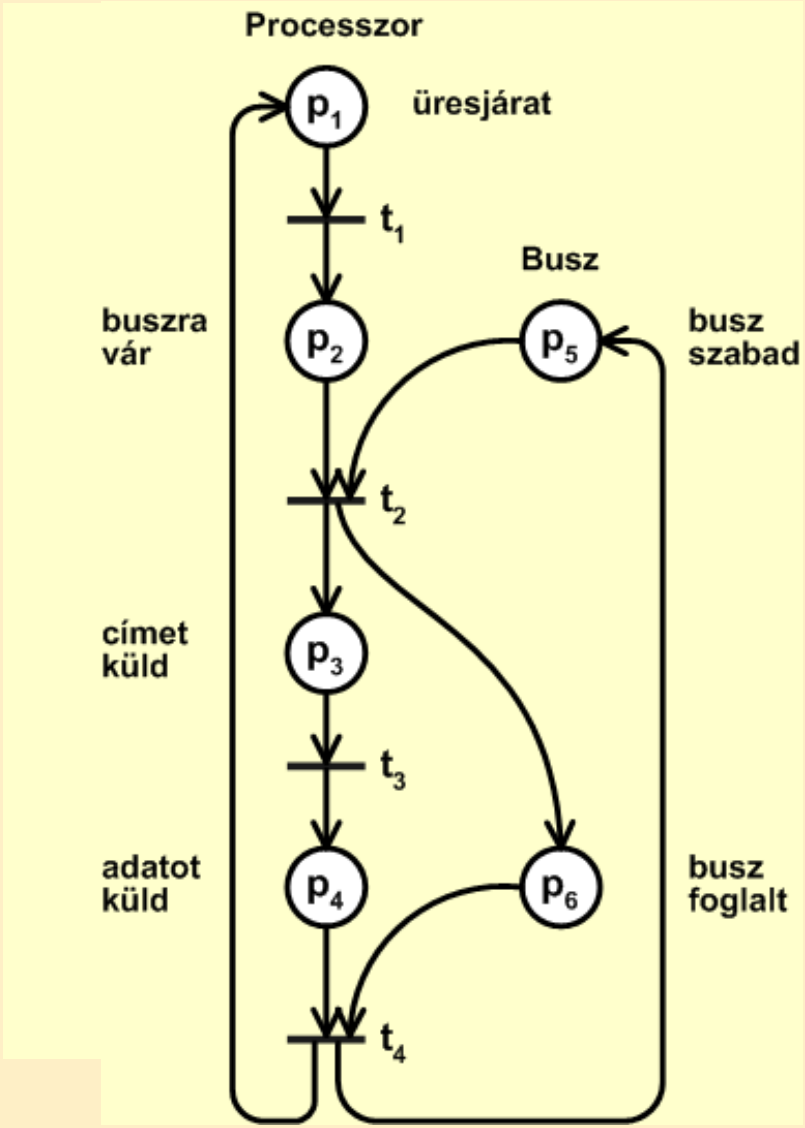
# Megoldási módszerek

Kérdések:

- $\sigma_T$ bázis komponenseinek értelmezési tartománya?
- a lineáris kombinációk együtthatóinak értelmezési tartománya?

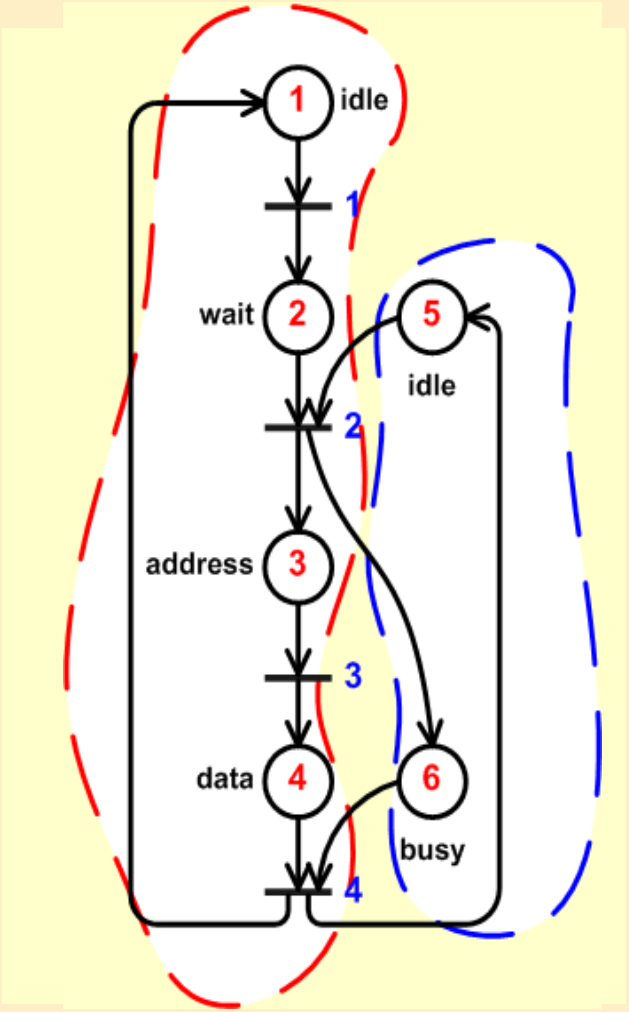
| Szint | Tartomány | Együttható | Lineárisan független? | Egyértelmű? | Algoritmus |
|---|---|---|---|---|---|
| 1 | $x \in \mathbf{Z}$ | $\mathbf{Q}$ | Igen | Nem | Gauss elimináció |
| 2 | $x \in \mathbf{Z}$ | $\mathbf{Z}$ | Igen | Nem | Hermite redukció |
| 3 | $x \in \mathbf{N}_0$ | $\mathbf{Q}_0$ | Nem biztos | Igen | Martinez-Silva |
| 4 | $x \in \mathbf{N}_0$ | $\mathbf{N}_0$ | Nem biztos | Igen | Pascoletti |
| 5 | $x \in \mathbf{B}$ | $\mathbf{B}$ | Nem biztos | Igen | Jaxy |

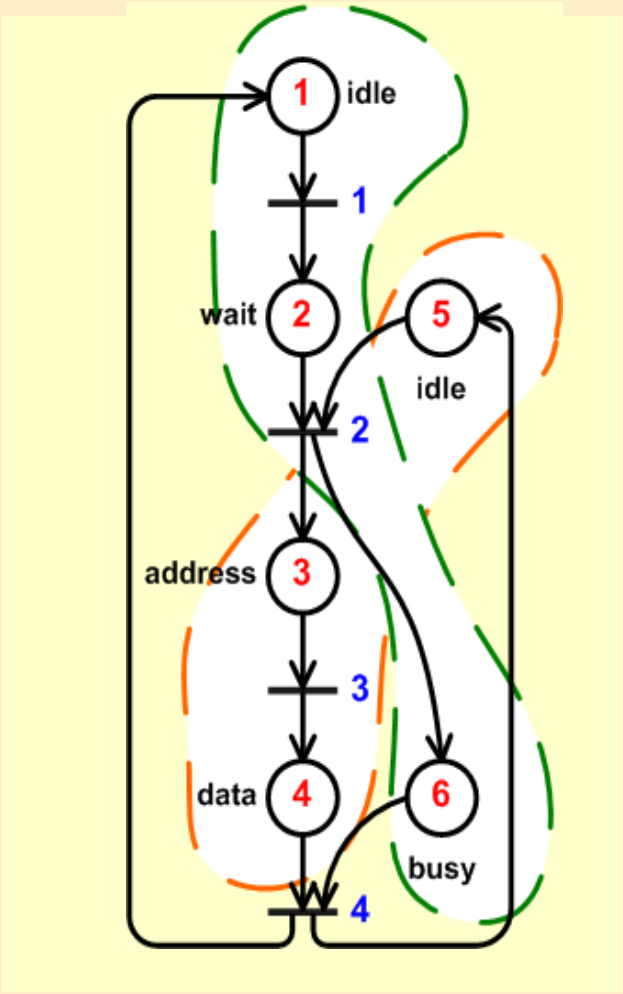
# Példa: processzor adatátvitel




- Processzor
  - várakozik (idle - üresjárat)
  - busz hozzáférési jogot kér
  - címet tesz ki a címbuszra
  - adatot tesz ki az adatbuszra
- Busz(ok)
  - szabad (nem használja senki)
  - foglalt (processzor/periféria)
- Petri háló
  - $n = 4$ darab átmenet
  - $m = 6$ darab hely

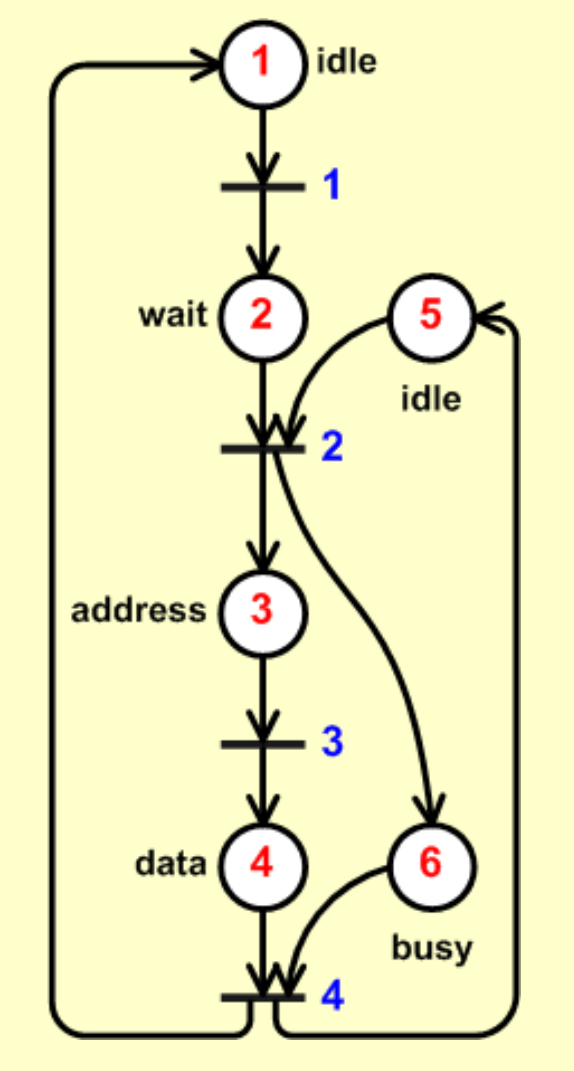
# Keressük meg fejben a megoldást!




Négy P invariáns található

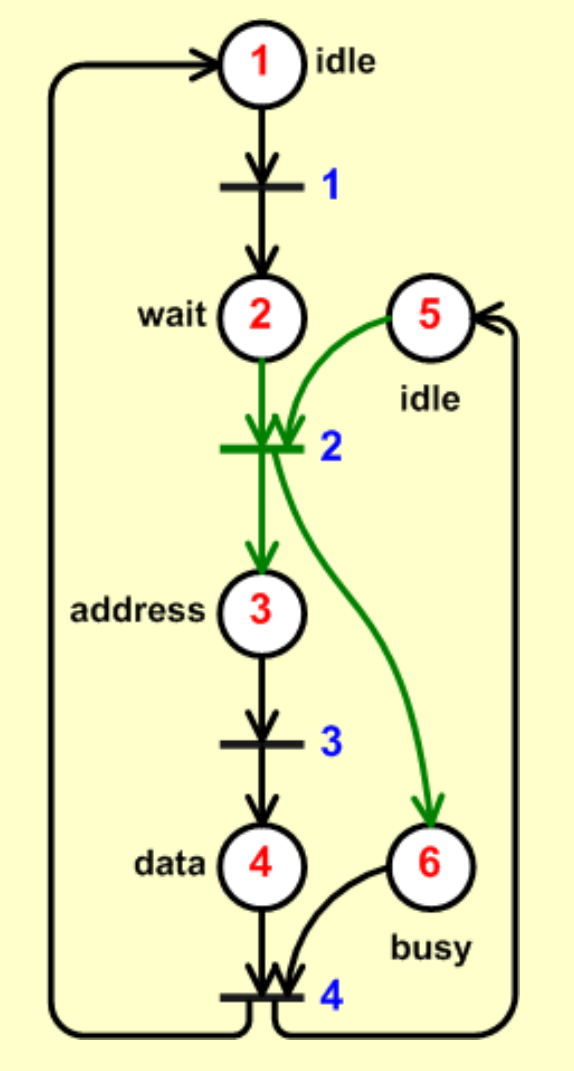
# Szomszédossági mátrixok




$$W^- = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & \\ 1 & 0 & 0 & 0 & 0 & 0 & t_1 \\ 0 & 1 & 0 & 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & 0 & 0 & 0 & t_3 \\ 0 & 0 & 0 & 1 & 0 & 1 & t_4 \end{bmatrix}$$

$$W^+ = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & \\ 0 & 1 & 0 & 0 & 0 & 0 & t_1 \\ 0 & 0 & 1 & 0 & 0 & 1 & t_2 \\ 0 & 0 & 0 & 1 & 0 & 0 & t_3 \\ 1 & 0 & 0 & 0 & 1 & 0 & t_4 \end{bmatrix}$$

# Szomszédossági mátrixok




$$\mathbf{W}^T = \begin{bmatrix} t_1 & t_2 & t_3 & t_4 & \\ -1 & 0 & 0 & 1 & p_1 \\ 1 & -1 & 0 & 0 & p_2 \\ 0 & 1 & -1 & 0 & p_3 \\ 0 & 0 & 1 & -1 & p_4 \\ 0 & -1 & 0 & 1 & p_5 \\ 0 & 1 & 0 & -1 & p_6 \end{bmatrix}$$

# Martinez-Silva algoritmus: inicializálás

$i \leftarrow 1$

$T_i \leftarrow \{\ t \in T\ \}$

$\mathbf{A} \leftarrow \mathbf{W}^{\mathsf{T}}$, $\mathbf{D} \leftarrow \mathbf{1}_n$ // $n = |P|$

$\mathbf{Q}_i \leftarrow [\mathbf{D} \mid \mathbf{A}]$ // egységmátrix + szomszédossági mátrix

$L_p \leftarrow$ a $\mathbf{Q}_i$ mátrix $p$. sora

$T_1 = \{\ t_1, t_2, t_3, t_4\ \}$

$\mathbf{Q_1} =$

| $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ | $t_1$ | $t_2$ | $t_3$ | $t_4$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 1 | $p_1$ |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | 0 | $p_2$ |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | $p_3$ |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | -1 | $p_4$ |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | -1 | 0 | 1 | $p_5$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | -1 | $p_6$ |

# Martinez-Silva algoritmus: ciklus

**while** $\mathbf{A_i} \neq 0$

**if** $t_j \in T_i$ // válasszunk egy eddig nem vizsgált oszlopot

$T_{i+1} \leftarrow T_i \setminus \{ t_j \}$

$L_{delete} \leftarrow \emptyset$

$\mathbf{Q}_{i+1} \leftarrow \mathbf{Q}_i$

**for all** $u, v : \mathrm{A_i}(u, j) \neq 0 \wedge \mathrm{A_i}(v, j) \neq 0 \wedge$
$\exists \lambda_u, \lambda_v \in \infty^+ : \lambda_u \mathrm{A_i}(u, j) + \lambda_v \mathrm{A_i}(v, j) = 0$

$\mathbf{Q}_{i+1}$-hez adjuk hozzá a $\lambda_u L_u + \lambda_v L_v$ sort

$L_{delete} \leftarrow L_{delete} \cup \{ L_u, L_v \}$

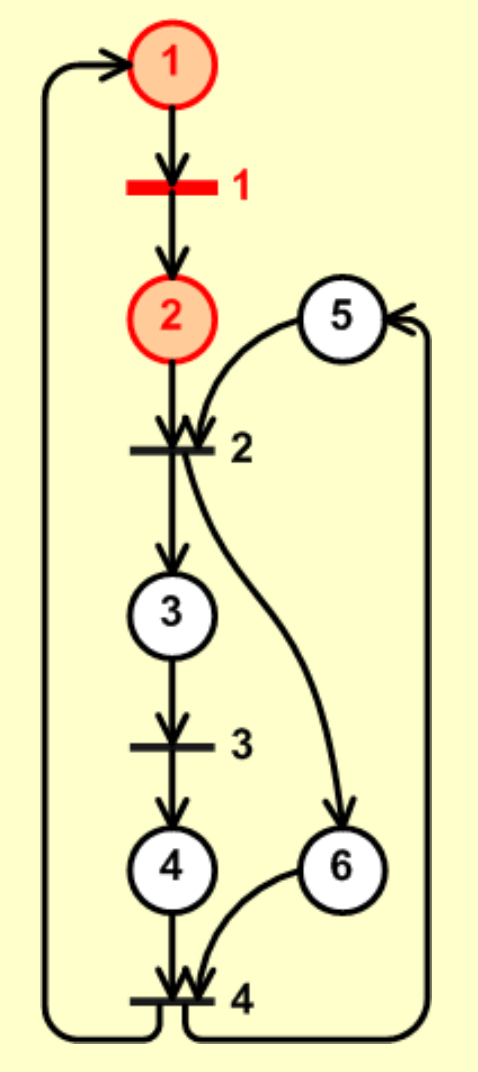
**end for**

$\mathbf{Q}_{i+1}$-ből töröljük az $L_{delete}$ halmazbeli sorokat

$i \leftarrow i + 1$

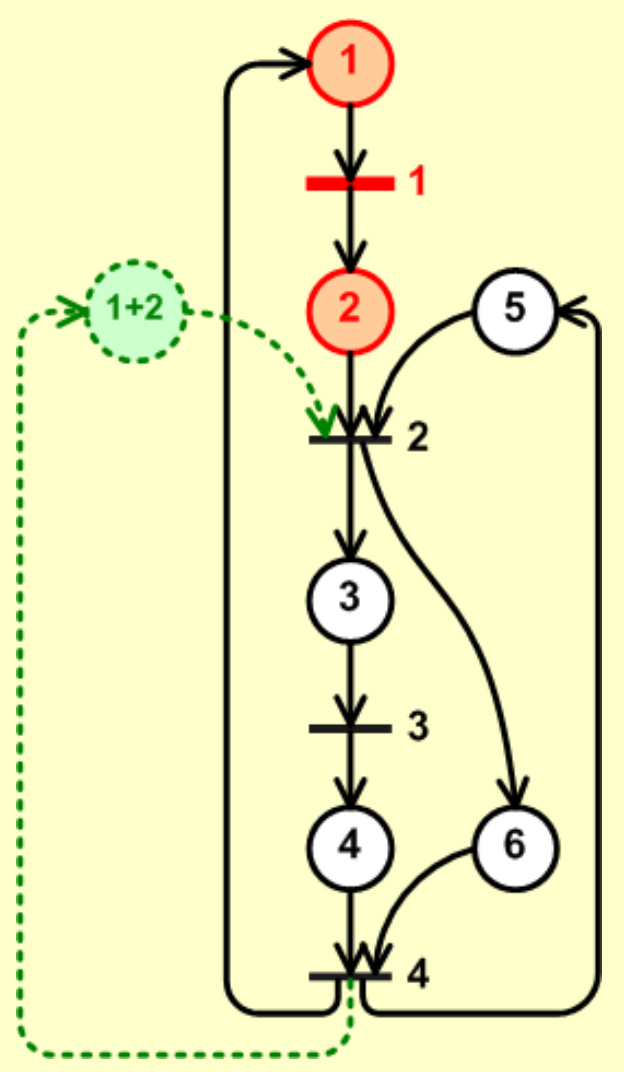
end while

# Martinez-Silva algoritmus: 1-1. lépés




$$Q_1 = \left[\begin{array}{cccccc|cccc|c} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & t_1 & t_2 & t_3 & t_4 & \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & p_1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & p_2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & p_3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & p_4 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & p_5 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 & p_6 \end{array}\right]$$

# Martinez-Silva algoritmus: 1-2. lépés



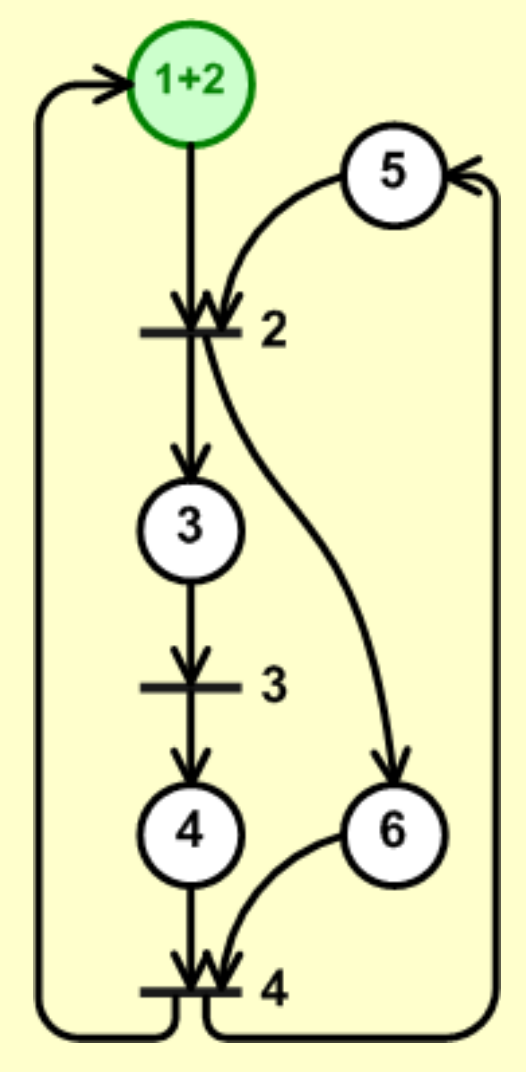

$Q_1 =$

| $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ | $t_1$ | $t_2$ | $t_3$ | $t_4$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 1 | $p_1$ |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | 0 | $p_2$ |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | $p_3$ |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | -1 | $p_4$ |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | -1 | 0 | 1 | $p_5$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | -1 | $p_6$ |

$Q_1' =$

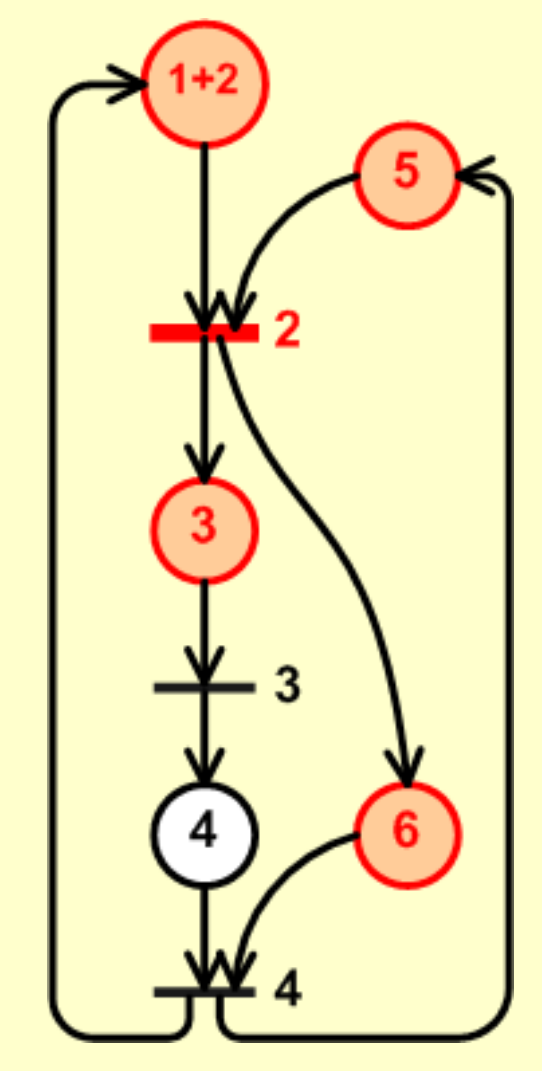
| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 1 | $p_1$ |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | 0 | $p_2$ |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | $p_3$ |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | -1 | $p_4$ |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | -1 | 0 | 1 | $p_5$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | -1 | $p_6$ |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 1 | $p_{1+2}$ |

# Martinez-Silva algoritmus: 1. részeredmény




$$Q_1'' = \left[\begin{array}{cccccc|cccc|c} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & t_1 & t_2 & t_3 & t_4 & \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & p_3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & p_4 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & p_5 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 & p_6 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & p_{1+2} \end{array}\right]$$

# Martinez-Silva algoritmus: 2-1, 2-2. lépés



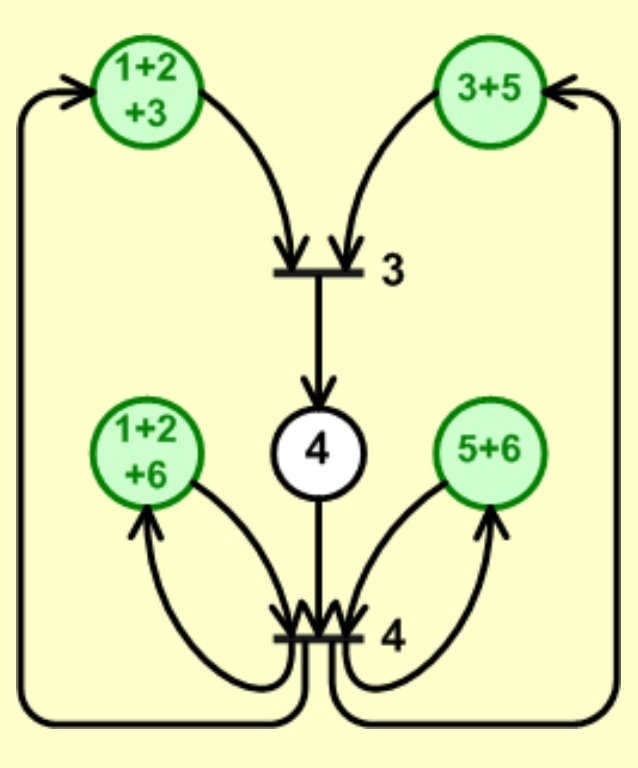

$Q_2 =$

| $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ | $t_1$ | $t_2$ | $t_3$ | $t_4$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | $p_3$ |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | -1 | $p_4$ |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | -1 | 0 | 1 | $p_5$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | -1 | $p_6$ |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 1 | $p_{1+2}$ |

$Q_2' =$

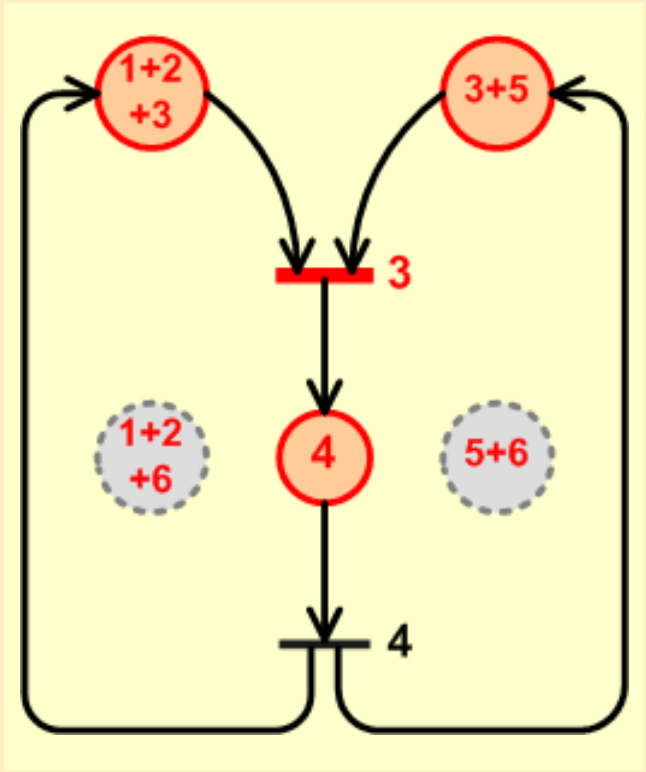
| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | $p_3$ |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | -1 | $p_4$ |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | -1 | 0 | 1 | $p_5$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | -1 | $p_6$ |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 1 | $p_{1+2}$ |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | -1 | 1 | $p_{1+2+3}$ |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | -1 | 1 | $p_{3+5}$ |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | $p_{1+2+6}$ |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | $p_{5+6}$ |

# Martinez-Silva algoritmus: 2. részeredmény




$$Q_2'' = \left[\begin{array}{cccccc|cccc|l} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & t_1 & t_2 & t_3 & t_4 & \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & p_4 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & p_{1+2+3} \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & -1 & 1 & p_{3+5} \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & p_{1+2+6} \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & p_{5+6} \end{array}\right]$$

# Martinez-Silva algoritmus: 3-1, 3-2. lépés



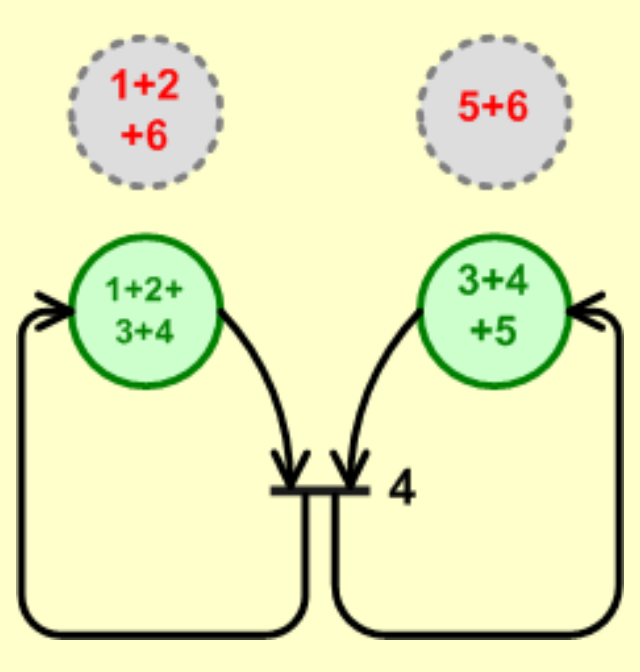

$Q_3 =$

| $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ | $t_1$ | $t_2$ | $t_3$ | $t_4$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | -1 | $p_4$ |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | -1 | 1 | $p_{1+2+3}$ |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | -1 | 1 | $p_{3+5}$ |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | $p_{1+2+6}$ |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | $p_{5+6}$ |

$Q_3' =$

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | -1 | $p_4$ |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | -1 | 1 | $p_{1+2+3}$ |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | -1 | 1 | $p_{3+5}$ |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | $p_{1+2+6}$ |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | $p_{5+6}$ |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | $p_{1+2+3+4}$ |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | $p_{3+4+5}$ |

# Martinez-Silva algoritmus: végeredmény




$$Q_3'' = \left[\begin{array}{cccccc|cccc|l} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & t_1 & t_2 & t_3 & t_4 & \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & p_{1+2+6} \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & p_{5+6} \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & p_{1+2+3+4} \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & p_{3+4+5} \end{array}\right]$$

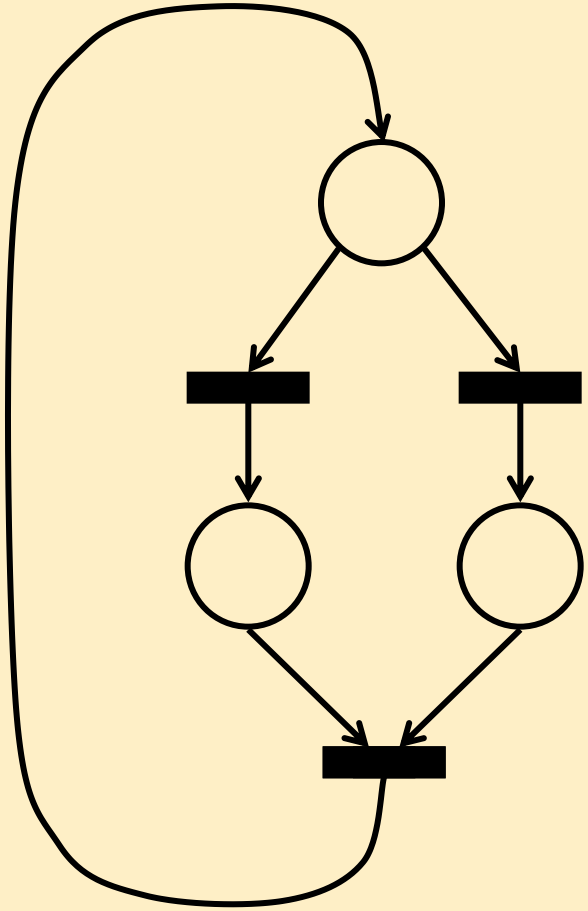
- Invariánsok:
  - a végső $Q_m = [D_m|0]$ mátrix alapján a $D_m$ mátrix soraiban található együtthatók
- Kiszámított P (hely) -invariánsok:
  1. $m(p_1)+m(p_2)+m(p_6) = 1$
  2. $m(p_5)+m(p_6) = 1$
  3. $m(p_1)+m(p_2)+m(p_3)+m(p_4) = 1$
  4. $m(p_3)+m(p_4)+m(p_5) = 1$

# További strukturális tulajdonságok

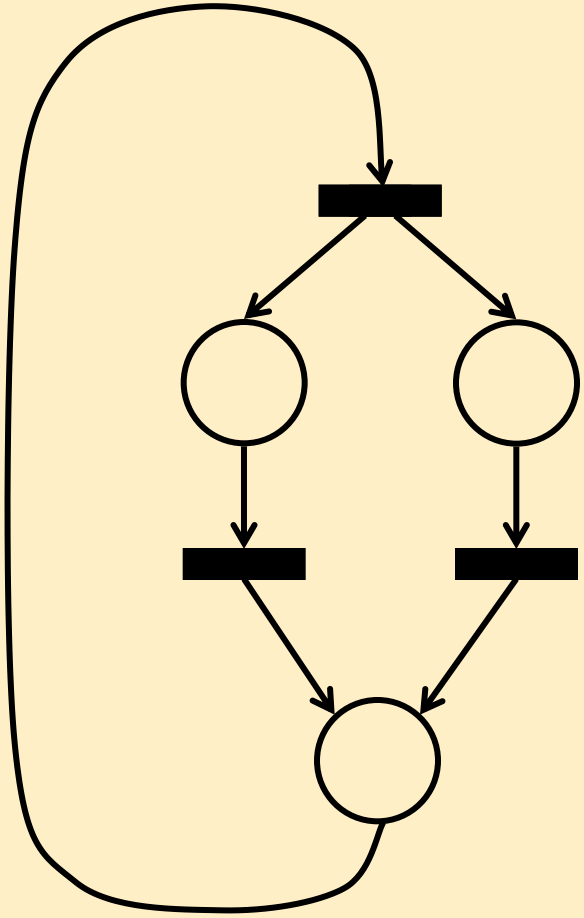
# Strukturális élőség

- Egy $N$ Petri háló strukturálisan élő, ha létezik olyan $M_0$ kezdőállapota, amelyben ($N$, $M_0$) ($L_4$-)élő
  - Szükséges feltétel: erősen összekötött gráf struktúra
  - Jelölt gráfok: egy ($G$, $M_0$) jelölt gráf a.cs.a. élő, ha $M_0$ állapotban minden G-beli irányított körben van legalább egy token $\rightarrow$ minden jelölt gráf strukturálisan élő
  - FC hálók: egy szabad választású háló strukturálisan élő, ha minden $N$–beli szifon tartalmaz csapdát
  - Általános (közönséges) Petri hálókra a strukturálisan élőség jellemzése (még) nem ismert

# Strukturális élőség, korlátosság?



nincs élő jelölése



nincs nemüres biztos jelölése

# Vezérelhetőség

- Egy $N$ Petri háló teljesen vezérelhető, ha bármely korlátos $M_0$ kezdőállapot esetén:

$$\forall M_i, M_j : M_i, M_j \in R(N, M_0) \Rightarrow M_i \in R(N, M_j) \wedge M_j \in R(N, M_i)$$

  - azaz bármely állapot elérhető bármely más állapotból

- Elégséges feltétel: $\boxed{\text{rang}(\mathbf{W}^{\mathrm{T}}) = m}$
  - mert $M_1 = M_0 + \mathbf{W}^{\mathrm{T}}\vec{\sigma} \rightarrow \mathbf{W}^{\mathrm{T}}\vec{\sigma} = \Delta M$
  - rangfeltétel: $\text{rang}(\mathbf{W}^{\mathrm{T}}) = \text{rang}(\mathbf{W}^{\mathrm{T}} \mid \Delta M) = m$

    ahol $m$ a helyek száma.

- Ue. szükséges feltétel is jelölt gráfok esetén

# Strukturális korlátosság

- Egy $N$ Petri háló strukturálisan korlátos, ha bármely korlátos $M_0$ kezdőállapotra korlátos marad
- Feltétele: létezik egy $m$ pozitív komponensű $\vec{\mu}$

$$\boxed{\vec{\mu} > 0, \mathbf{W}\vec{\mu} \leq 0}$$

- Szükségesség: $M \in R(N, M_0) \rightarrow M = M_0 + \mathbf{W}^{\mathrm{T}}\vec{\sigma}, \vec{\sigma} \geq 0$
  - átrendezve: $M^{\mathrm{T}}\vec{\mu} = M_0^{\mathrm{T}}\vec{\mu} + \underbrace{\vec{\sigma}^{\mathrm{T}}\mathbf{W}\vec{\mu}}_{\mathbf{W}\vec{\mu}\leq 0, \vec{\sigma}\geq 0}$ ← felhasználjuk a feltételt

    (belső szorzat)
  - felső korlát: $M^{\mathrm{T}}\vec{\mu} \leq M_0^{\mathrm{T}}\vec{\mu} \Rightarrow M(p) \leq \dfrac{M_0^{\mathrm{T}}\vec{\mu}}{\mu_p}$

# Strukturális korlátosság: elégségesség

- $\vec{\mu} > 0, \mathbf{W}^{\mathrm{T}} \vec{\mu} \leq 0$ feltétel elégséges is, mert
  - egyébként $\exists \vec{\sigma} \geq 0 : \mathbf{W}^{\mathrm{T}} \vec{\sigma} \underset{\neq}{\geq} 0$

$$\exists M, M_0 : M - M_0 = \mathbf{W}^{\mathrm{T}} \vec{\sigma} \underset{\neq}{\geq} 0 \rightarrow M \underset{\neq}{\geq} M_0$$

  - ekkor megfelelő $M_0$ választásával $\vec{\sigma}$ tetszőlegesen sokszor végrehajtható és $N$ nem korlátos

# Lineáris mátrixegyenlőtlenségek

vagy az egyik, vagy a másik megoldható

| *Lemma* | Rendszer$_1$ $\oplus$ | Rendszer$_2$ |
|---|---|---|
| Minkowski-Farkas | $\mathbf{W}^{\mathrm{T}}\vec{\sigma} \underset{\neq}{\geq} \vec{b},\ \vec{\sigma}$ tetszőleges | $\mathbf{W}\vec{\mu} = 0,\ \vec{\mu} \geq 0,\ \vec{\mu}^{\mathrm{T}}\vec{b} > 0$ |
| Minkowski-Farkas | $\mathbf{W}^{\mathrm{T}}\vec{\sigma} \underset{\neq}{\geq} \vec{b},\ \vec{\sigma} \geq 0$ | $\mathbf{W}\vec{\mu} \leq 0,\ \vec{\mu} \geq 0,\ \vec{\mu}^{\mathrm{T}}\vec{b} > 0$ |
| Stiemke | $\mathbf{W}^{\mathrm{T}}\vec{\sigma} \underset{\neq}{\geq} 0,\ \vec{\sigma}$ tetszőleges<br>$\rightarrow$ nem konzervatív | $\mathbf{W}\vec{\mu} = 0,\ \vec{\mu} > 0$<br>$\rightarrow$ konzervatív |
| Farkas | $\mathbf{W}^{\mathrm{T}}\vec{\sigma} \underset{\neq}{\geq} 0,\ \vec{\sigma} \geq 0$<br>$\rightarrow$ nem strukturálisan korlátos | $\mathbf{W}\vec{\mu} \leq 0,\ \vec{\mu} > 0$<br>$\rightarrow$ strukturálisan korlátos |

# Konzervativitás

- Egy *N* Petri háló (részlegesen) konzervatív, ha bármely korlátos $M_0$ és $M \in R(N, M_0)$ állapotra minden (néhány) $p \in P$ helyhez található egy $\mu_p$ pozitív egész súlytényező, hogy $M\vec{\mu} = M_0\vec{\mu} = \text{állandó}$
- Szükséges és elégséges feltétel: $\exists \vec{\mu} \underset{\neq}{\geq} 0 : \mathbf{W}\vec{\mu} = 0$

# Ismételhetőség

- Egy *N* Petri háló (részlegesen) ismételhető, ha létezik olyan $M_0$ kezdőállapot és $M_0$-ből induló $\sigma$ tüzelési szekvencia, hogy minden (néhány) $t \in T$ tranzíció végtelen sokszor tüzel $\sigma$-ban
- Szükséges és elégséges feltétel: $\exists \vec{\sigma} \underset{\neq}{\geq} 0 : \mathbf{W}^{\mathrm{T}} \vec{\sigma} \geq 0$
  - Bizonyítás: $\exists \vec{\sigma} > 0 : \mathbf{W}^{\mathrm{T}} \vec{\sigma} \geq 0$

    $$\exists M, M_0 : M - M_0 = \mathbf{W}^{\mathrm{T}} \vec{\sigma} \geq 0 \rightarrow M \geq M_0$$
  - ekkor megfelelő $M_0$ választásával $\vec{\sigma}$ tetszőlegesen sokszor végrehajtható

# Konzisztencia

- Egy $N$ Petri háló (részlegesen) konzisztens, ha létezik olyan $M_0$ kezdőállapot és $M_0$-ből induló és $M_0$-ba visszavezető $\sigma$ tüzelési szekvencia, hogy minden (néhány) $t \in T$ tranzíció legalább egyszer tüzel $\sigma$-ban
- Szükséges és elégséges feltétel: $\exists \vec{\sigma} \underset{\neq}{\geq} 0 : \mathbf{W}^{\mathrm{T}} \vec{\sigma} = 0$
  - Bizonyítás: ismételhetőség feltételénél látott módon

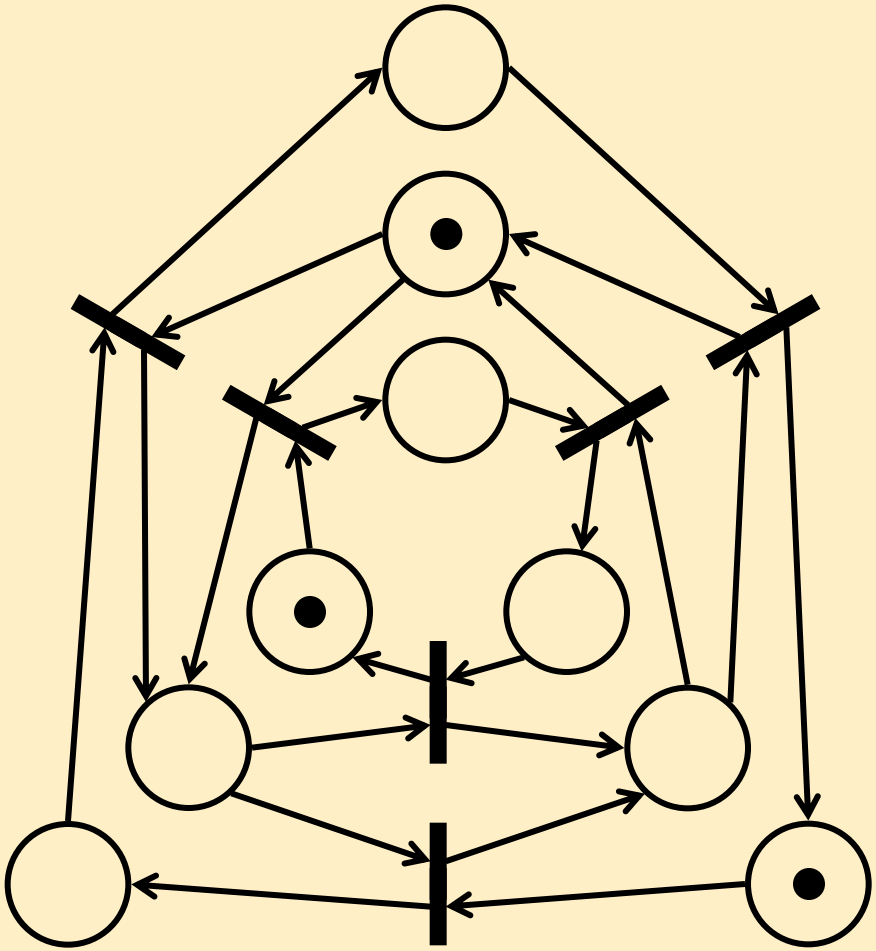
# Strukturális B-fairség

- Két tranzíció strukturálisan B-fair, ha bármely $M_0$ kezdőállapot esetén B-fair (korlátos fair) relációban állnak.
- Egy $N$ Petri háló B-fair, ha bármely két tranzíciója esetén a B-fair reláció teljesül
- Egy $N$ Petri háló strukturálisan B-fair, ha bármely $M_0$ kezdőállapotra a háló B-fair
  - B-fair reláció ekvivalencia reláció $\rightarrow$ tranzíciókat ekvivalencia osztályokba csoportosítja
  - Strukturális B-fair reláció $\rightleftharpoons$ B-fair reláció

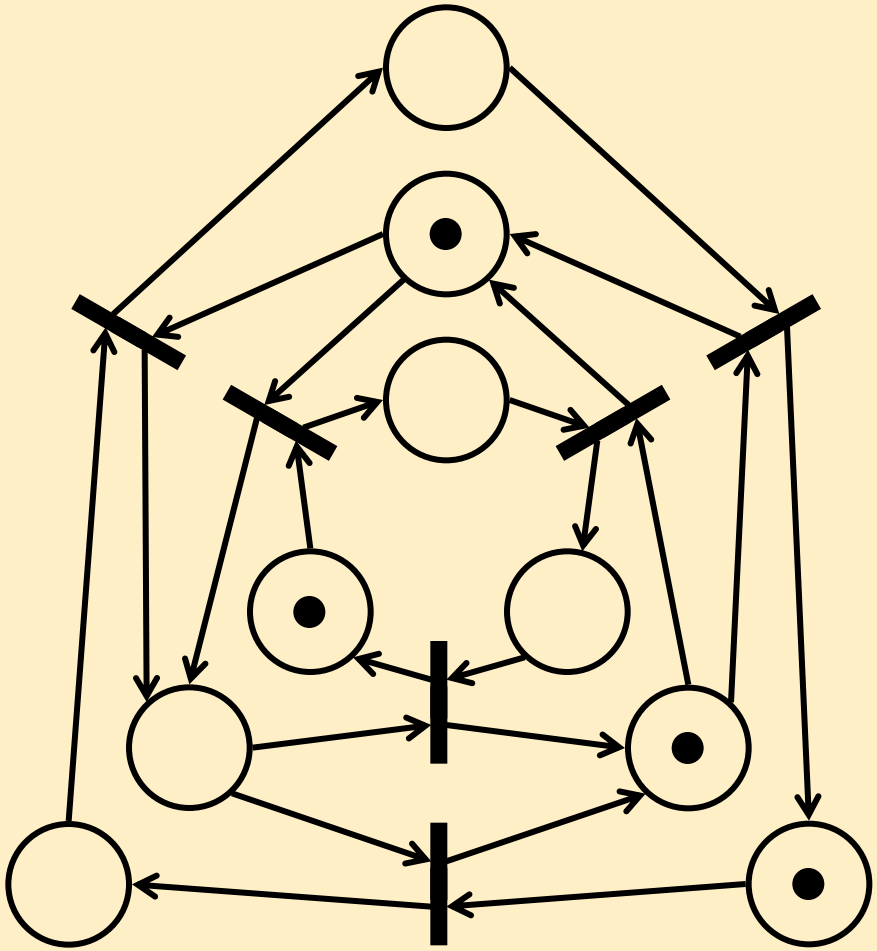
# Strukturális B-fairség feltétele

- Egy strukturálisan korlátos Petri háló a.cs.a. strukturálisan B-fair, ha
  - konzisztens és csak egy minimális nemnegatív T-invariánsa van, vagy
  - nem konzisztens és nincs minimális nemnegatív T-invariánsa
- Minden erősen összekötött jelölt gráf strukturálisan B-fair

# B-fair, de nem strukturálisan B-fair háló



élő és B-fair $M_0$



élő, de nem B-fair $M_0$

# Összefoglalás

| | Tulajdonság | Szükséges és elégséges felt. |
|---|---|---|
| SB | Strukturálisan korlátos | $\exists \vec{\mu} > 0, \mathbf{W}\vec{\mu} \leq 0$ (vagy $\nexists \vec{\sigma} > 0, \mathbf{W}^{\mathrm{T}}\vec{\sigma} \underset{\neq}{\geq} 0$) |
| CN | Konzervatív | $\exists \vec{\mu} > 0, \mathbf{W}\vec{\mu} = 0$ (vagy $\nexists \vec{\sigma}, \mathbf{W}^{\mathrm{T}}\vec{\sigma} \underset{\neq}{\geq} 0$) |
| PCN | Részlegesen konzervatív | $\exists \vec{\mu} \underset{\neq}{\geq} 0, \mathbf{W}\vec{\mu} = 0$ |
| RP | Ismételhető | $\exists \vec{\sigma} > 0, \mathbf{W}^{\mathrm{T}}\vec{\sigma} \geq 0$ |
| PRP | Részlegesen ismételhető | $\exists \vec{\sigma} \underset{\neq}{\geq} 0, \mathbf{W}^{\mathrm{T}}\vec{\sigma} \geq 0$ |
| CS | Konzisztens | $\exists \vec{\sigma} > 0, \mathbf{W}^{\mathrm{T}}\vec{\sigma} = 0$ (vagy $\nexists \vec{\mu}, \mathbf{W}\vec{\mu} \underset{\neq}{\geq} 0$) |
| PCS | Részlegesen konzisztens | $\exists \vec{\sigma} \underset{\neq}{\geq} 0, \mathbf{W}^{\mathrm{T}}\vec{\sigma} = 0$ |

# További strukturális tulajdonságok

| Ha | Akkor |
|---|---|
| *N* strukturálisan korlátos és strukturálisan élő | *N* konzervatív és konzisztens. |
| $\exists \vec{\mu} \geq 0, \mathbf{W}\vec{\mu} \underset{\neq}{\leq} 0$ | Létezik nem élő $M_0$ *N*–hez. *N* nem konzisztens. |
| $\exists \vec{\mu} \geq 0, \mathbf{W}\vec{\mu} \geq 0$ | (*N*, $M_0$) nem korlátos egy élő $M_0$ esetén. *N* nem konzisztens. |
| $\exists \vec{\sigma} \geq 0, \mathbf{W}^{\mathrm{T}}\vec{\sigma} \underset{\neq}{\leq} 0$ | Létezik nem élő $M_0$ strukturálisan korlátos *N*–hez. *N* nem konzisztens. |
| $\exists \vec{\sigma} \geq 0, \mathbf{W}^{\mathrm{T}}\vec{\sigma} \geq 0$ | *N* nem strukturálisan korlátos. *N* nem konzervatív. |

# Példa Petri háló modell analízisére
# Alternáló bit protokol

# A modellezési feladat

## Alternating Bit Protocol

- Átviteli protokoll veszteséges csatornához
  - üzenet elveszhet (véges számú alkalommal)
  - üzenet tartalma nem változhat
- Cél: a protokoll biztosítsa, hogy minden üzenet véges időn belül eljusson a vevőhöz

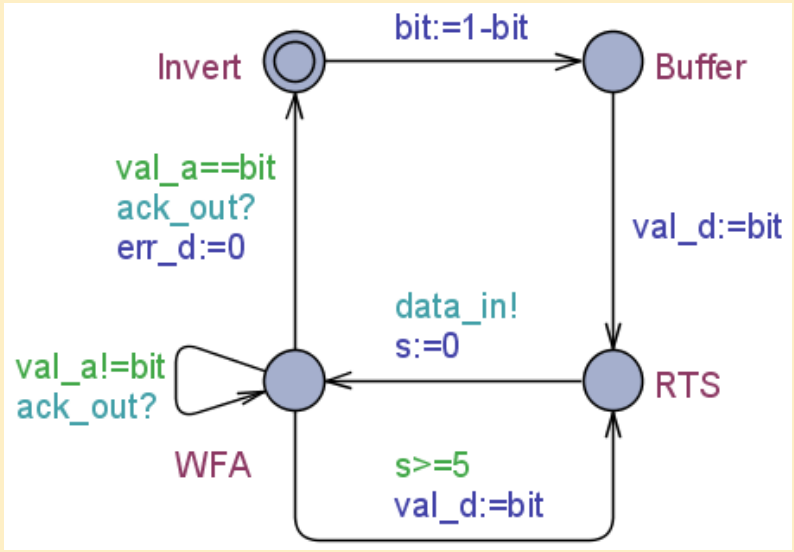
# Küldő folyamat

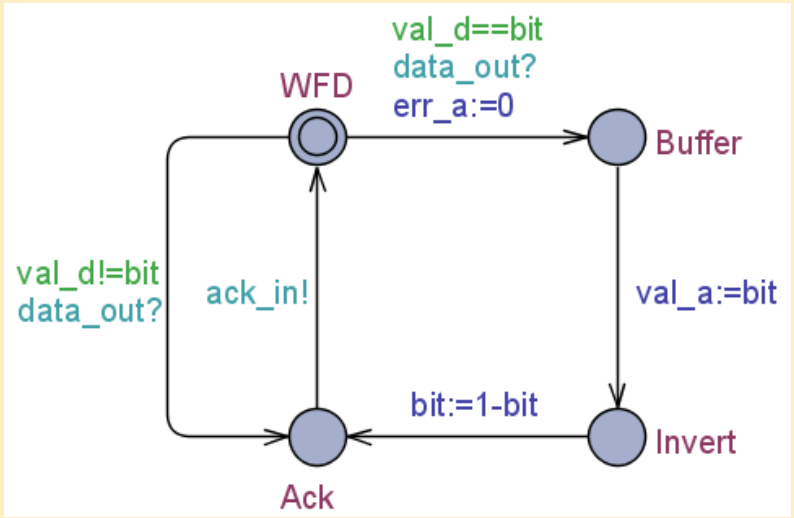
- Üzenetekhez egy ellenőrző bitet kapcsol
- Az üzenetek megérkezését nyugta jelzi
- A nyugta tartalmazza az ellenőrző bitet
- Első üzenethez csatolt bit: $\mathbf{b^0}$
  - ha az üzenet elvész, a folyamat időtúllepéssel észleli a nyugta hiányát $\rightarrow$ újra küldi
  - ha a folyamat $\mathbf{b^0}$ bittel ellátott nyugtát kap (ilyet várt), akkor a következő üzenethez $\mathbf{b^1} = \neg\, \mathbf{b^0}$ bitet köt
  - ha a folyamat $\mathbf{b^X}$ bittel ellátott nyugtát vár és $\mathbf{b^Y}$ bittel ellátott nyugtát kap $\rightarrow$ egyszerűen eldobja

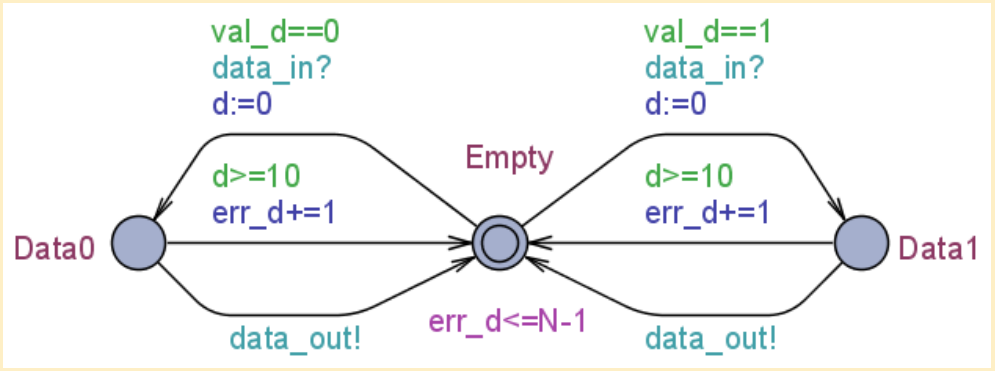
# Fogadó folyamat

- Első vétel: $b^0$ ellenőrző bittel jelölt üzenetet kap
- Az üzenetet feldolgozza, a vételt a kapott bit visszaküldésével nyugtázza
  - ha a következő üzenetben az ellenőrző bit értéke $b^1$ (helyesen), akkor az új üzenetet is feldolgozza és a $b^1$ bit visszaküldésével nyugtázza
  - ha a következő üzenetben az ellenőrző bit értéke $b^0$ (nem megfelelő), akkor az üzenetet eldobja (korábban már feldolgozta), de nyugtát küld

# UPPAAL modell

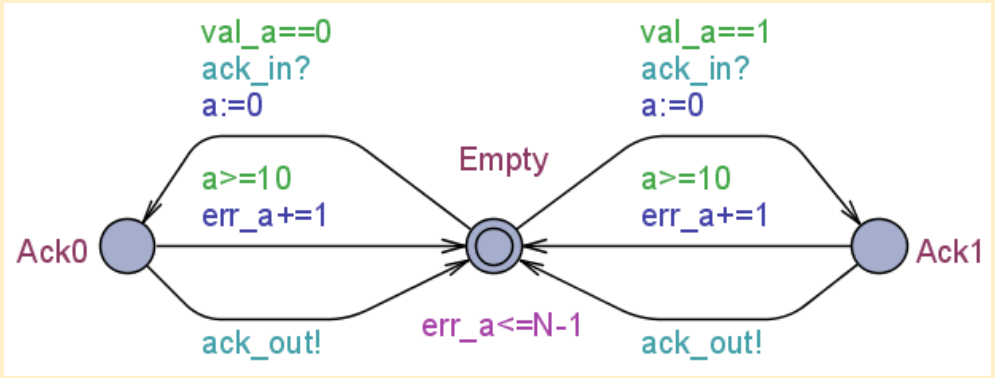


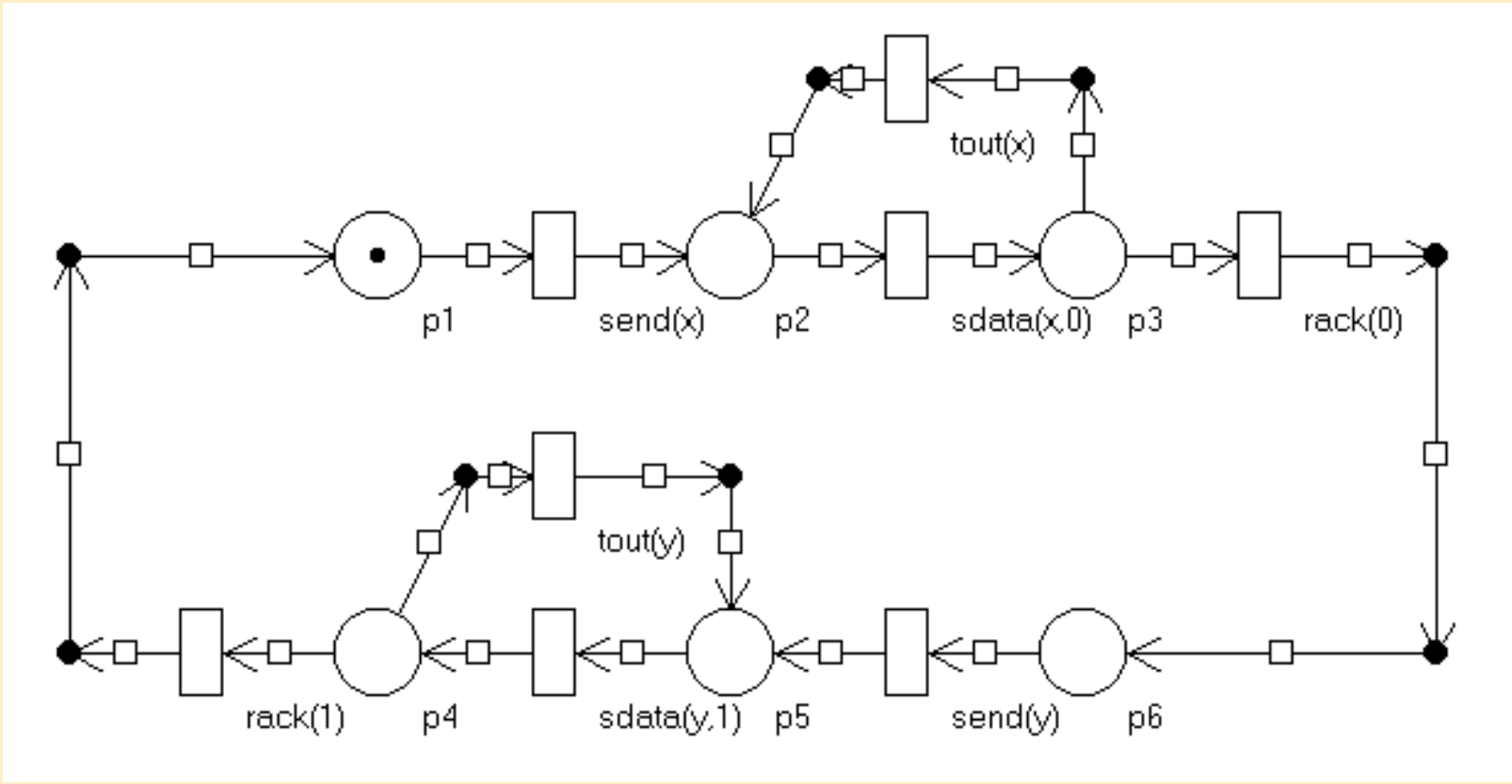




```
const int N=10;

typedef int[0,1] bit_t;
typedef int[0,2] value_t;
typedef int[0,N] error_t;
```

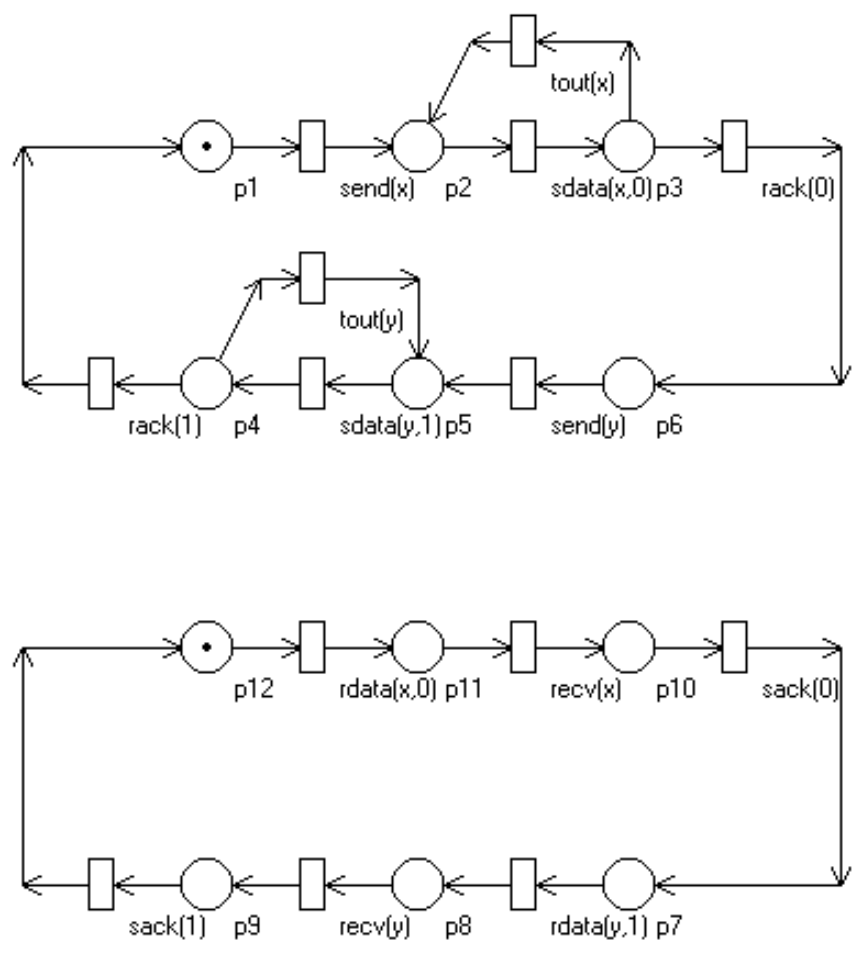
```
value_t val_d, val_a;
error_t err_d, err_a;

chan data_in, ack_in;
urgent chan data_out, ack_out;
```



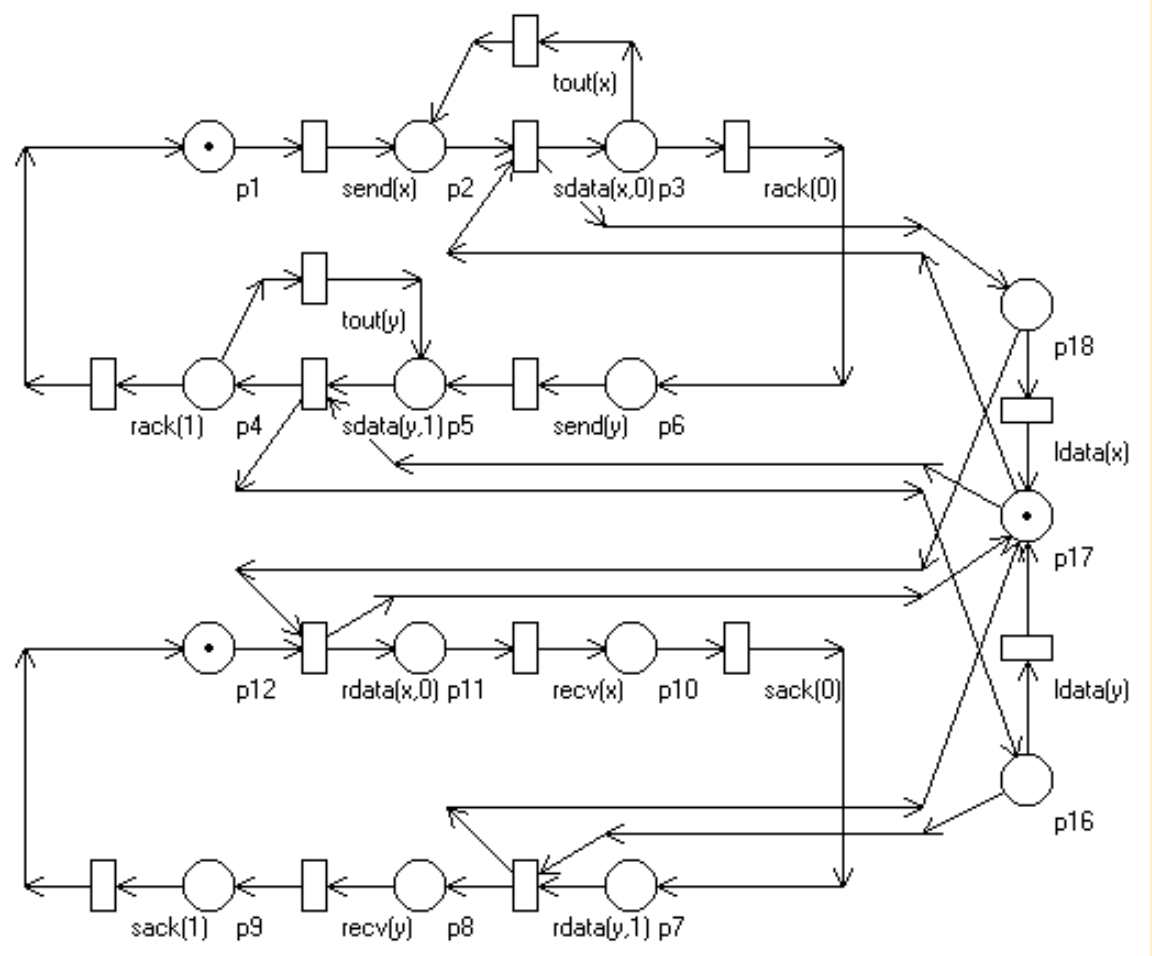
# Küldő folyamat Petri háló modellje



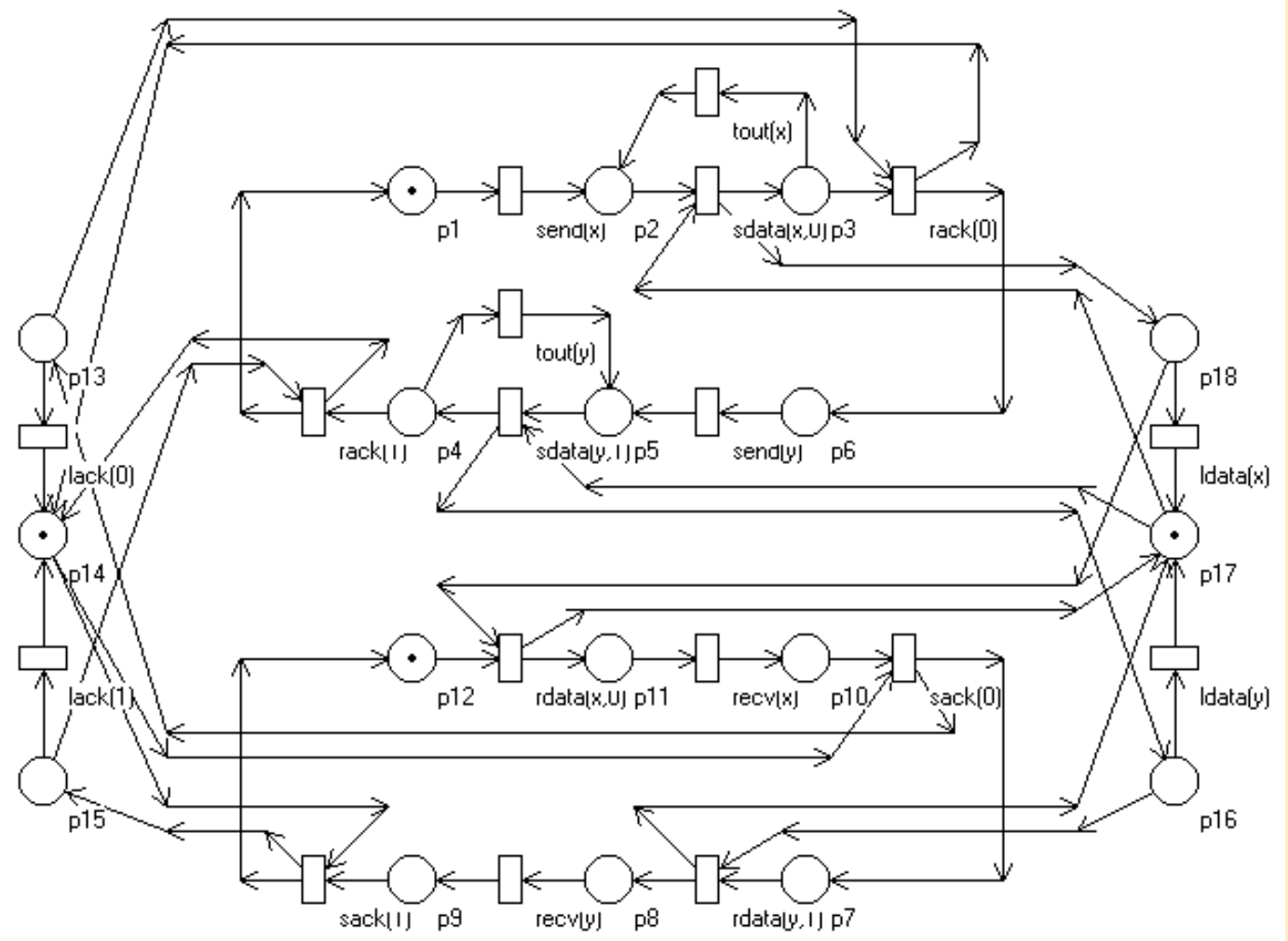
# Fogadó folyamat Petri háló modellje



# Adat csatorna és az átvitel

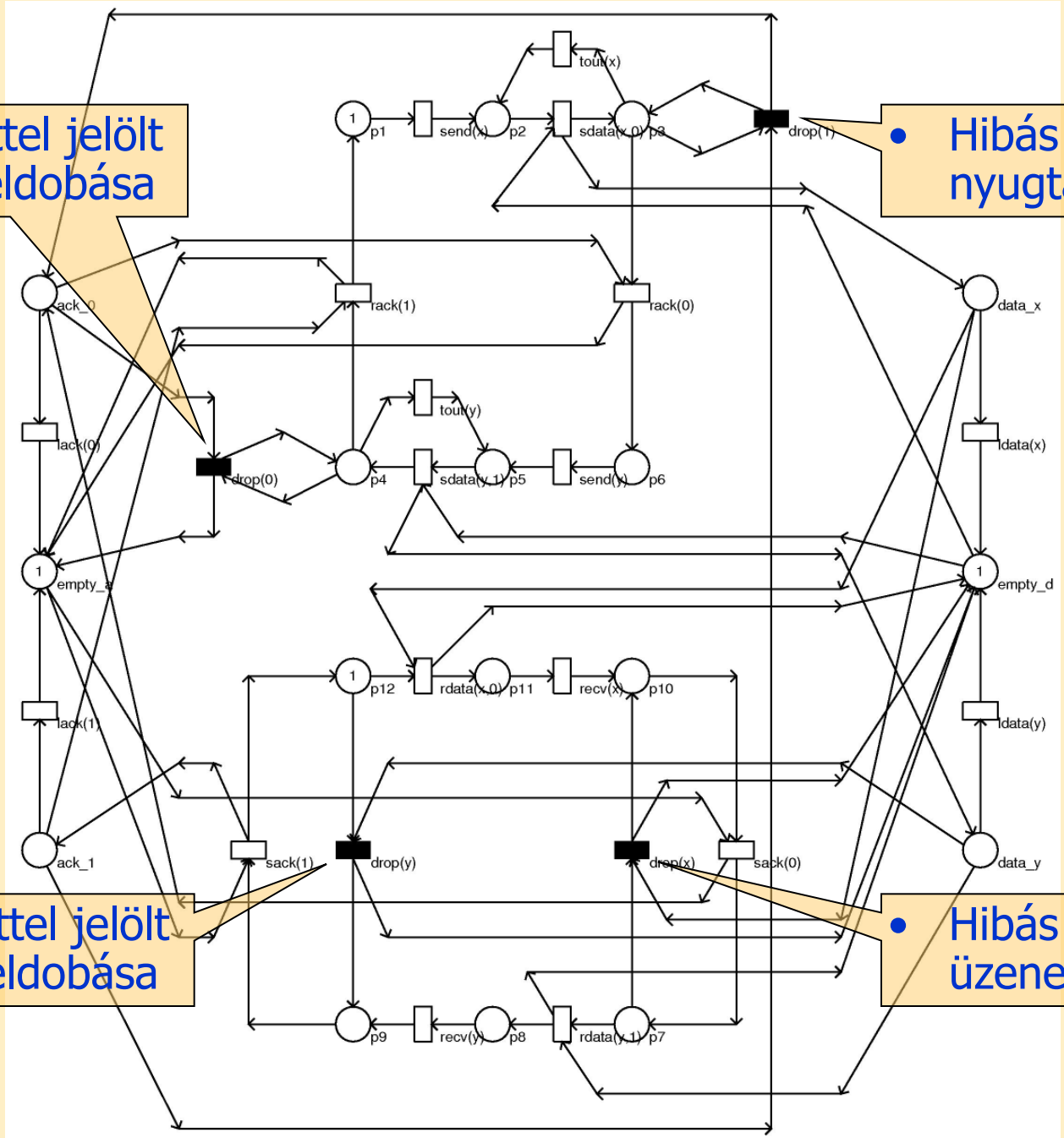


# Nyugtázó csatorna



- Hibás bittel jelölt nyugta eldobása

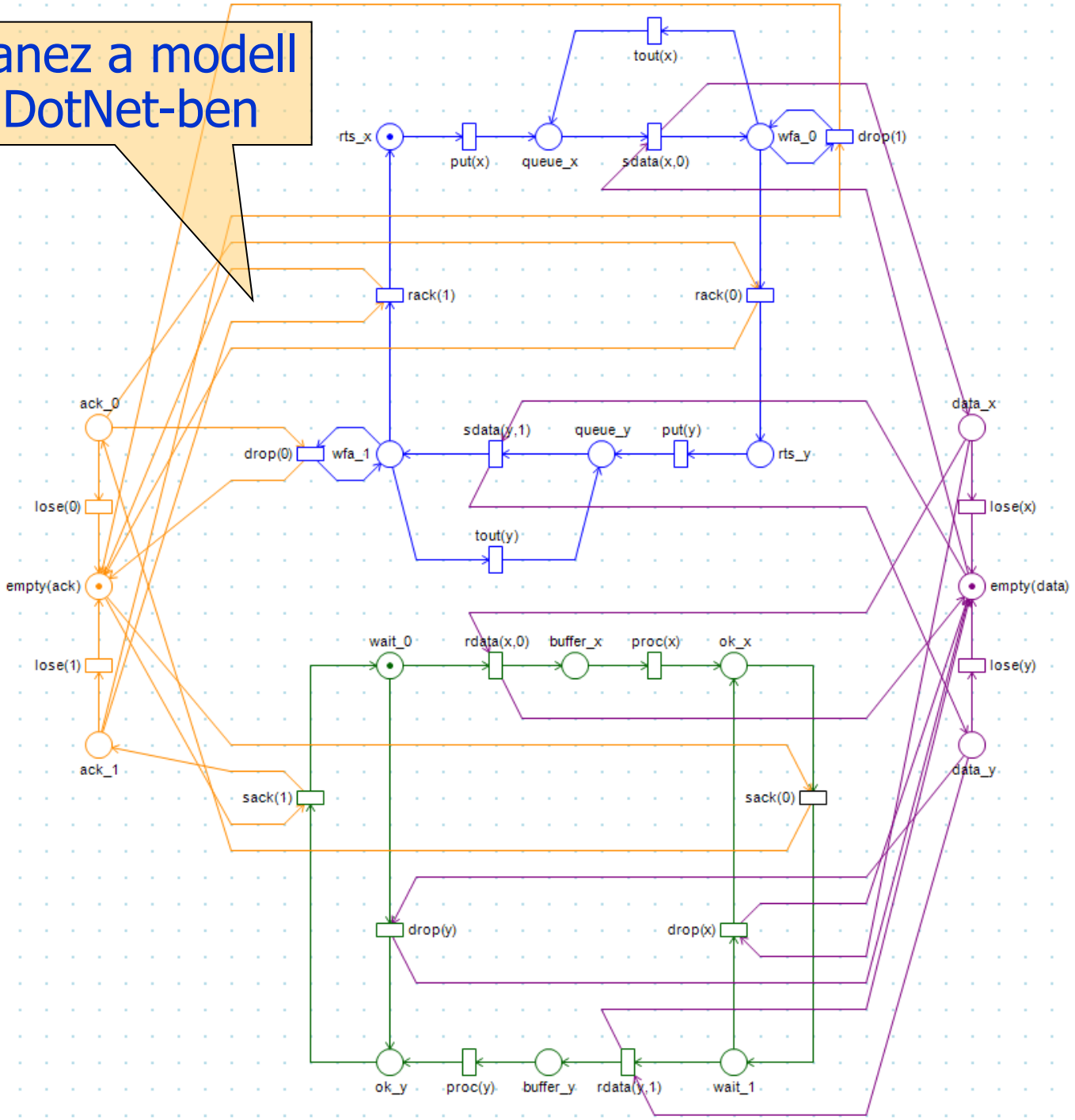
- Hibás bittel jelölt nyugta eldobása



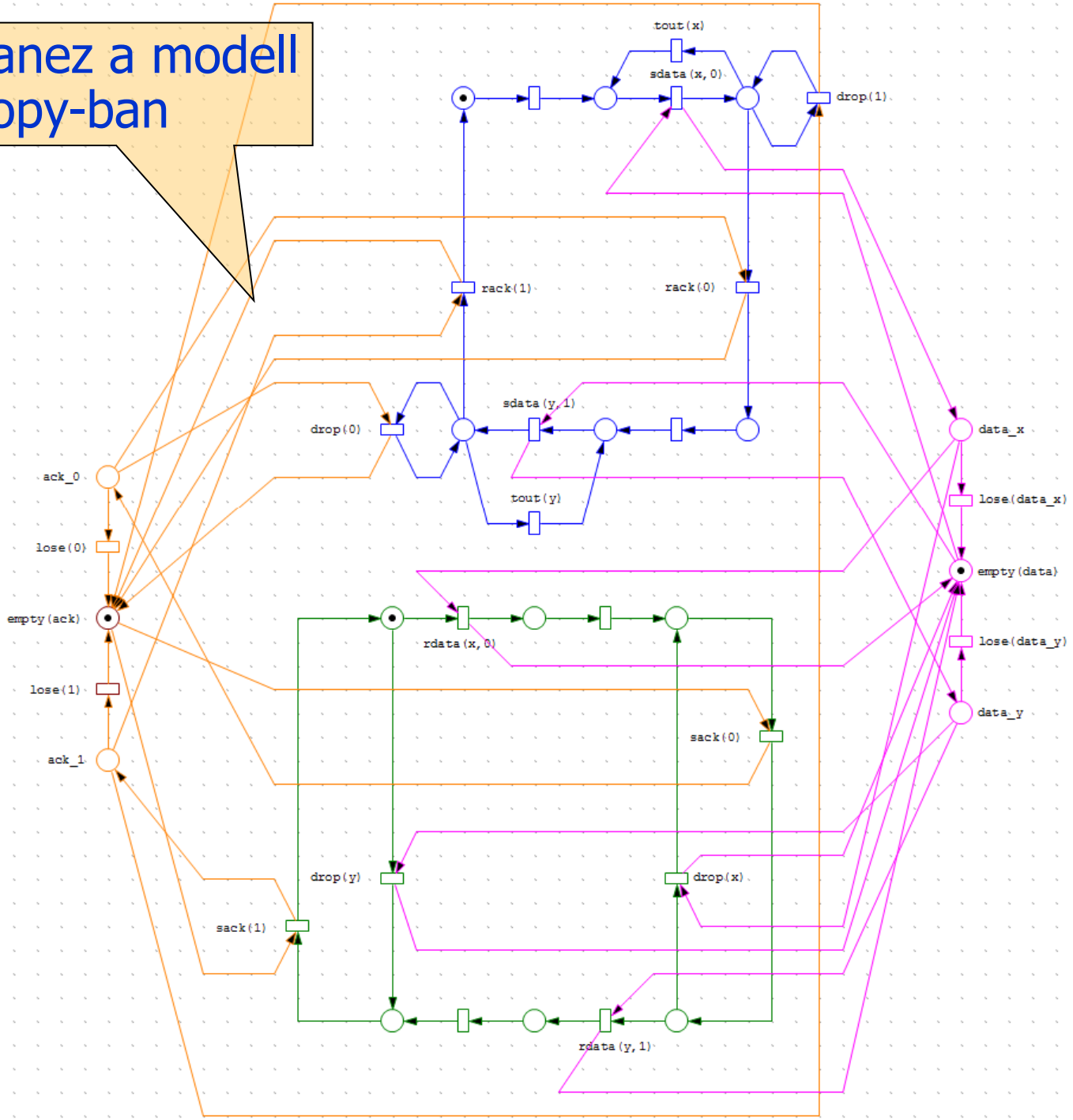

- Hibás bittel jelölt üzenet eldobása

- Hibás bittel jelölt üzenet eldobása

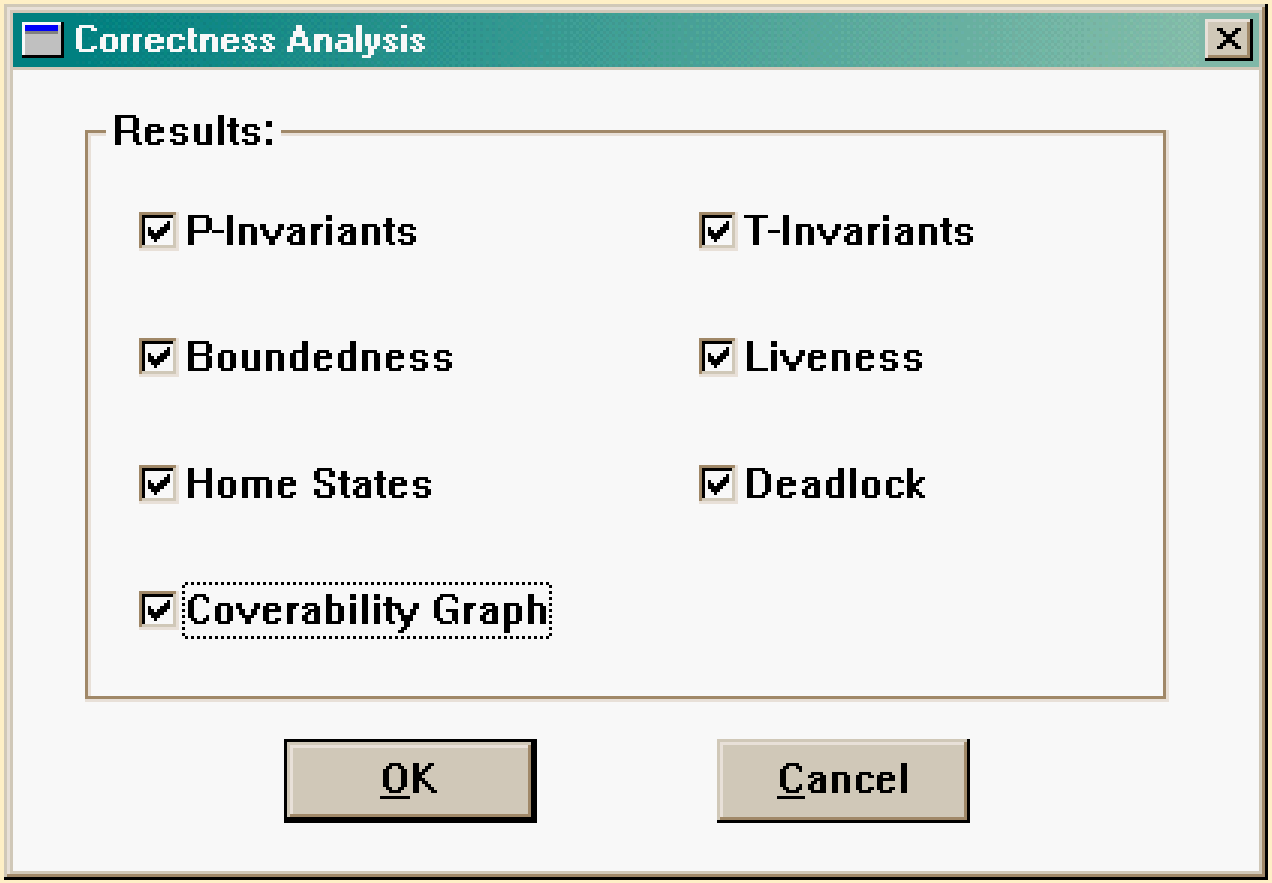
- Ugyanez a modell PetriDotNet-ben



- Ugyanez a modell Snoopy-ban



# DNAnet analízis eszköztár



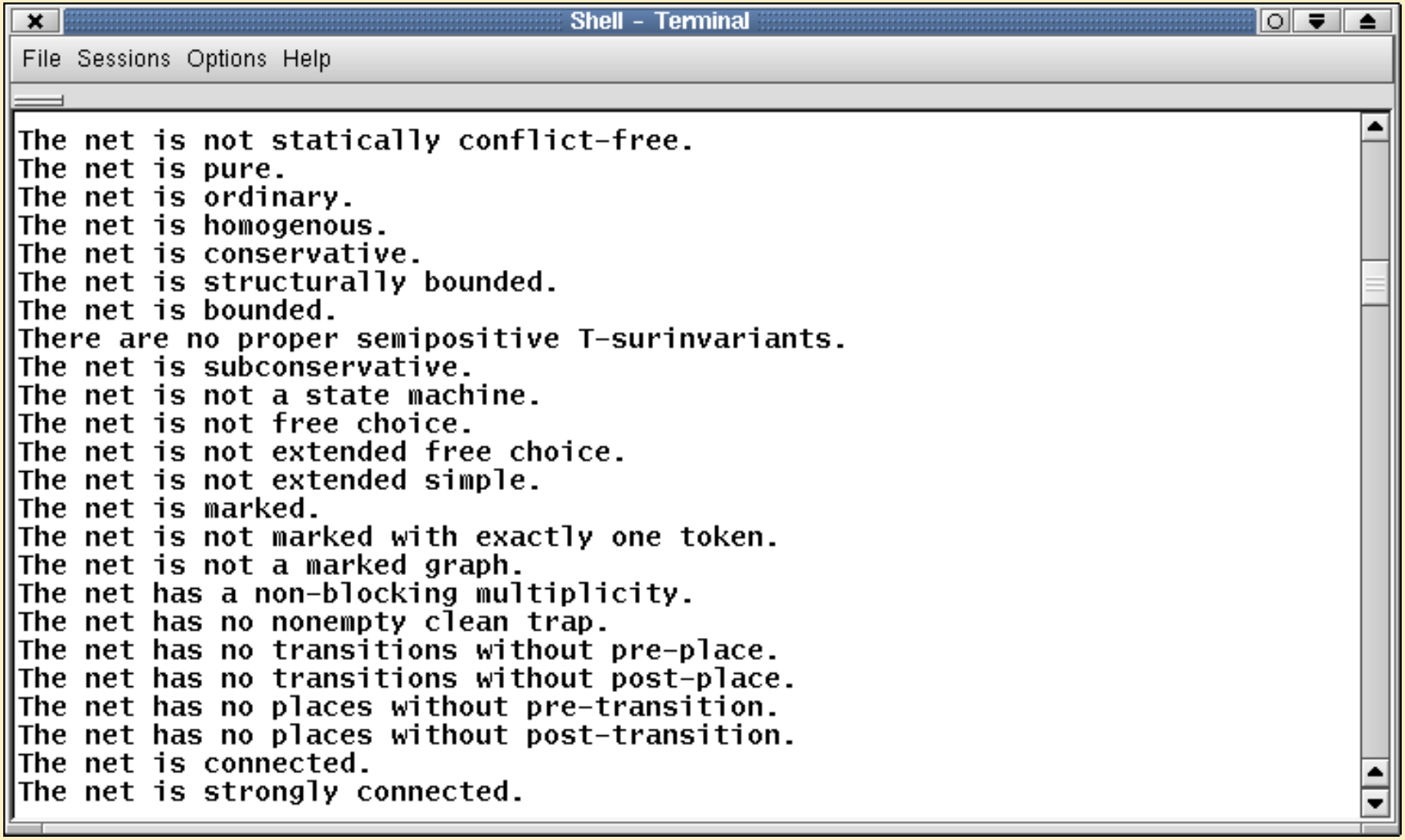
# DNAnet: Dinamikus tulajdonságok

- Korlátosság (Boundedness)
- Élő ($L_4$-élő) tulajdonság (Liveness)
- Holtpont (Deadlock) felderítése
- Visszatérő állapotok (Home States)
- Fedési gráf (Coverability Graph)
- Hely invariánsok (P-Invariants)
- Tüzelési invariánsok (T-Invariants)

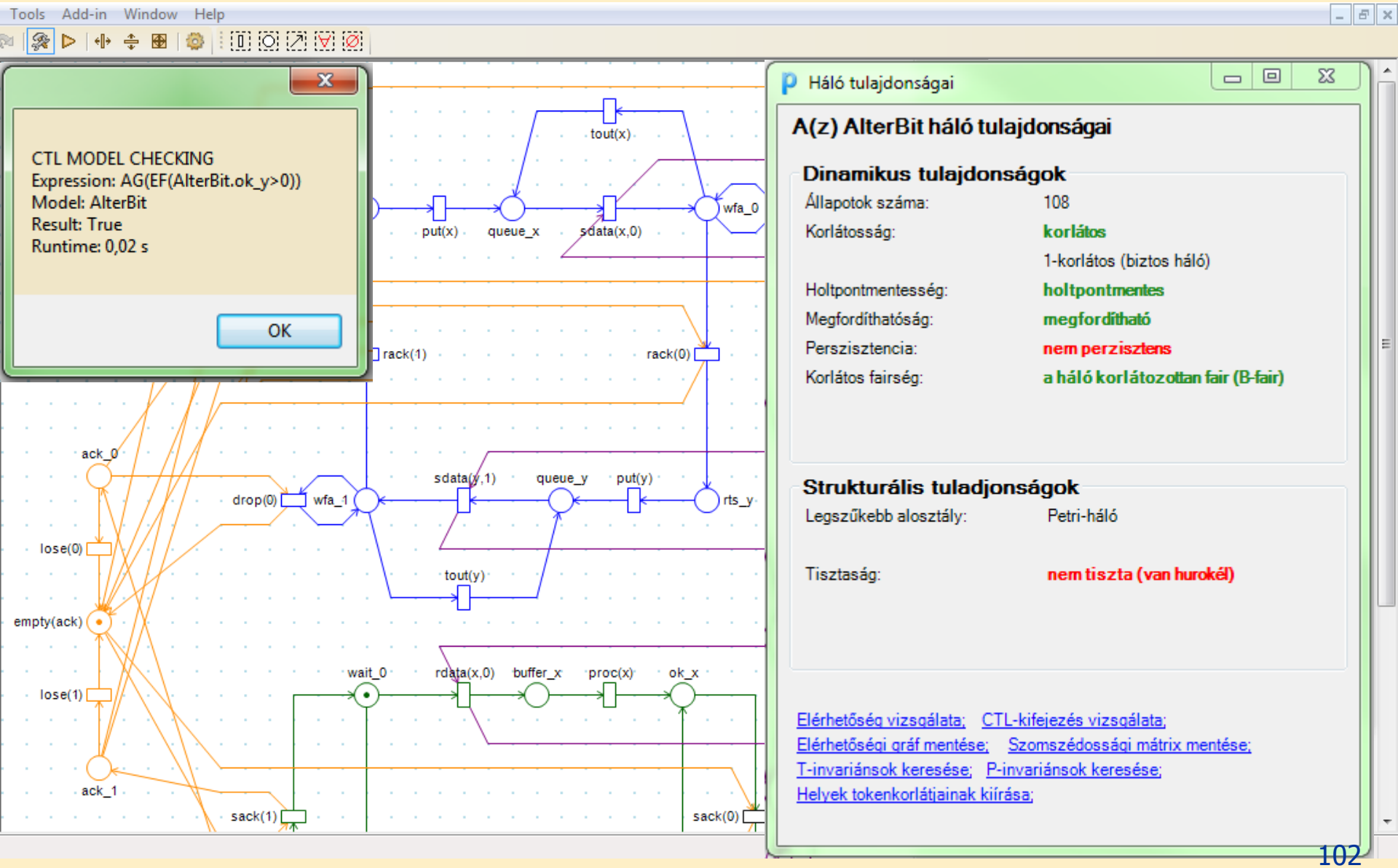
# DNAnet: A modell dinamikus tulajdonságai

```
Net is bounded.

Deadlock is not possible.

Net is live.

Net has home states.

Coverability graph generation statistics:
108 unique markings
1 strongly connected components
108 hash table entries used
1 was the longest hash list length
1 was the average hash list length
27 was the maximum stack height
3 was the maximum component stack height
```

# INA: Petri háló tulajdonságok



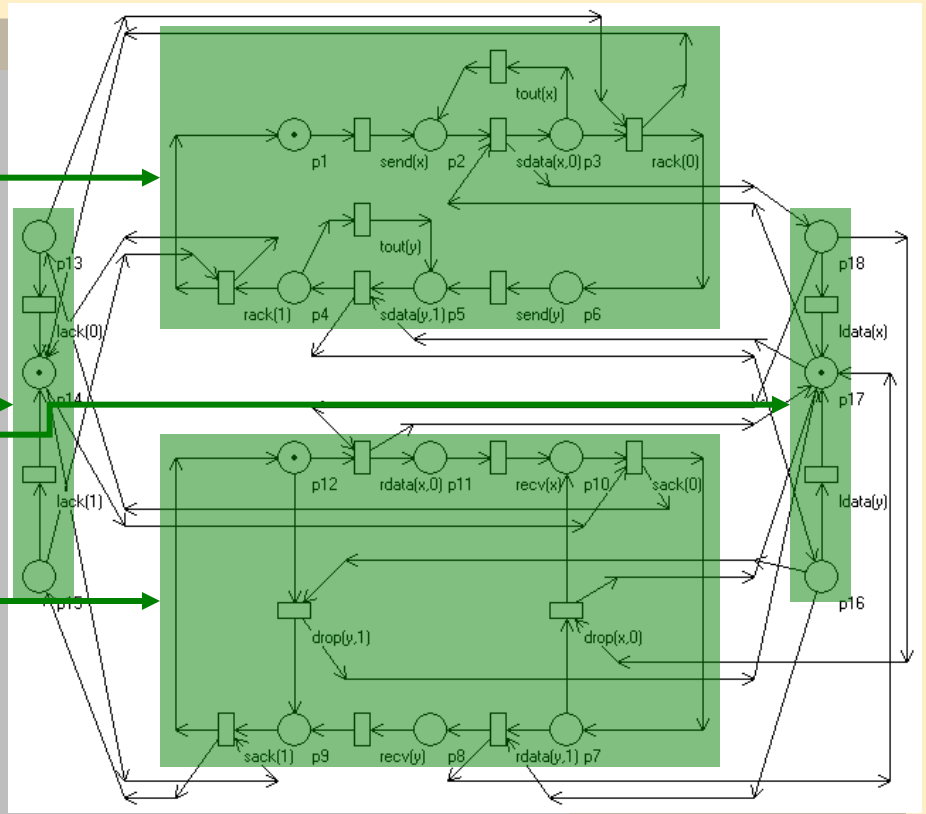
# PetriDotNet analízis eredmények



# DNAnet: Hely invariánsok

P-invariants:

| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | (main.p1) |
| 1 | 0 | 0 | 0 | (main.p2) |
| 1 | 0 | 0 | 0 | (main.p3) |
| 1 | 0 | 0 | 0 | (main.p4) |
| 1 | 0 | 0 | 0 | (main.p5) |
| 1 | 0 | 0 | 0 | (main.p6) |
| 0 | 1 | 0 | 0 | (main.p7) |
| 0 | 1 | 0 | 0 | (main.p8) |
| 0 | 1 | 0 | 0 | (main.p9) |
| 0 | 1 | 0 | 0 | (main.p10) |
| 0 | 1 | 0 | 0 | (main.p11) |
| 0 | 1 | 0 | 0 | (main.p12) |
| 0 | 0 | 1 | 0 | (main.p13) |
| 0 | 0 | 1 | 0 | (main.p14) |
| 0 | 0 | 1 | 0 | (main.p15) |
| 0 | 0 | 0 | 1 | (main.p16) |
| 0 | 0 | 0 | 1 | (main.p17) |
| 0 | 0 | 0 | 1 | (main.p18) |




ie.

M(main.p1) + M(main.p2) + M(main.p3) + M(main.p4) + M(main.p5) + M(main.p6)
M(main.p7) + M(main.p8) + M(main.p9) + M(main.p10) + M(main.p11) + M(main.p12)
M(main.p13) + M(main.p14) + M(main.p15)
M(main.p16) + M(main.p17) + M(main.p18)

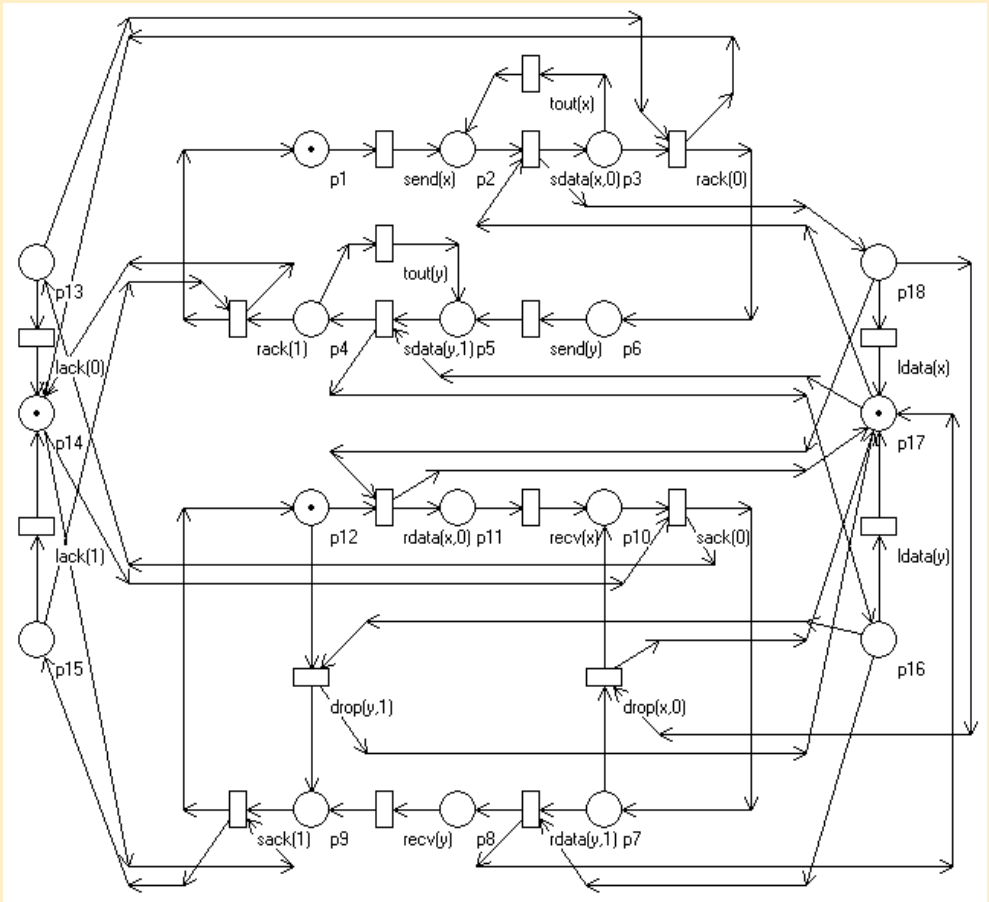
All places are covered by P-invariants.

# DNAnet: Tüzelési invariánsok

hibás működés (nem tüzelhető)

T invariants:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | (main.send(x)) |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | (main.sdata(x,0)) |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | (main.rack(0)) |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | (main.send(y)) |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | (main.sdata(y,1)) |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | (main.rack(1)) |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | (main.tout(x)) |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | (main.tout(y)) |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | (main.sack(1)) |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | (main.recv(y)) |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | (main.rdata(y,1)) |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | (main.sack(0)) |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | (main.recv(x)) |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | (main.rdata(x,0)) |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | (main.lack(0)) |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | (main.lack(1)) |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | (main.ldata(y)) |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | (main.ldata(x)) |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | (main.drop(x,0)) |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | (main.drop(y,1)) |



# INA: Invariánsok

alterbit.pin

File Search Options Help

```
place invariants basis of net 0.alterbit        :
=================
Nr.          0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
   1 |       1   1   1   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0 +
   2 |       0   0   0   0   0   0   1   1   1   1   1   1   0   0   0   0   0   0 +
   3 |       0   0   0   0   0   0   0   0   0   0   0   0   1   1   1   0   0   0 +
   4 |       0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1   1 +

@
```

alterbit.inv

File Search Options Help

```
transition sub/sur/invariants for net 0.alterbit        :

semipositive transition invariants =

Nr.          0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  1 |       1   1   1   0   1   1   1   0   1   1   1   1   1   1   0   0   0   0   0   0
  2 |       0   0   0   0   0   1   0   1   0   0   0   0   0   0   0   1   0   0   0   0
  3 |       0   0   0   0   0   1   0   1   1   0   0   0   0   0   0   0   1   0   1   0
  4 |       1   1   1   0   1   1   1   0   1   0   0   1   0   0   0   0   0   0   1   1
  5 |       0   1   0   1   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0
  6 |       0   1   0   1   0   0   0   0   0   0   0   1   0   0   0   0   0   1   0   1
  7 |       0   1   0   1   0   1   0   1   1   1   1   1   1   1   0   0   1   1   0   0

@
```

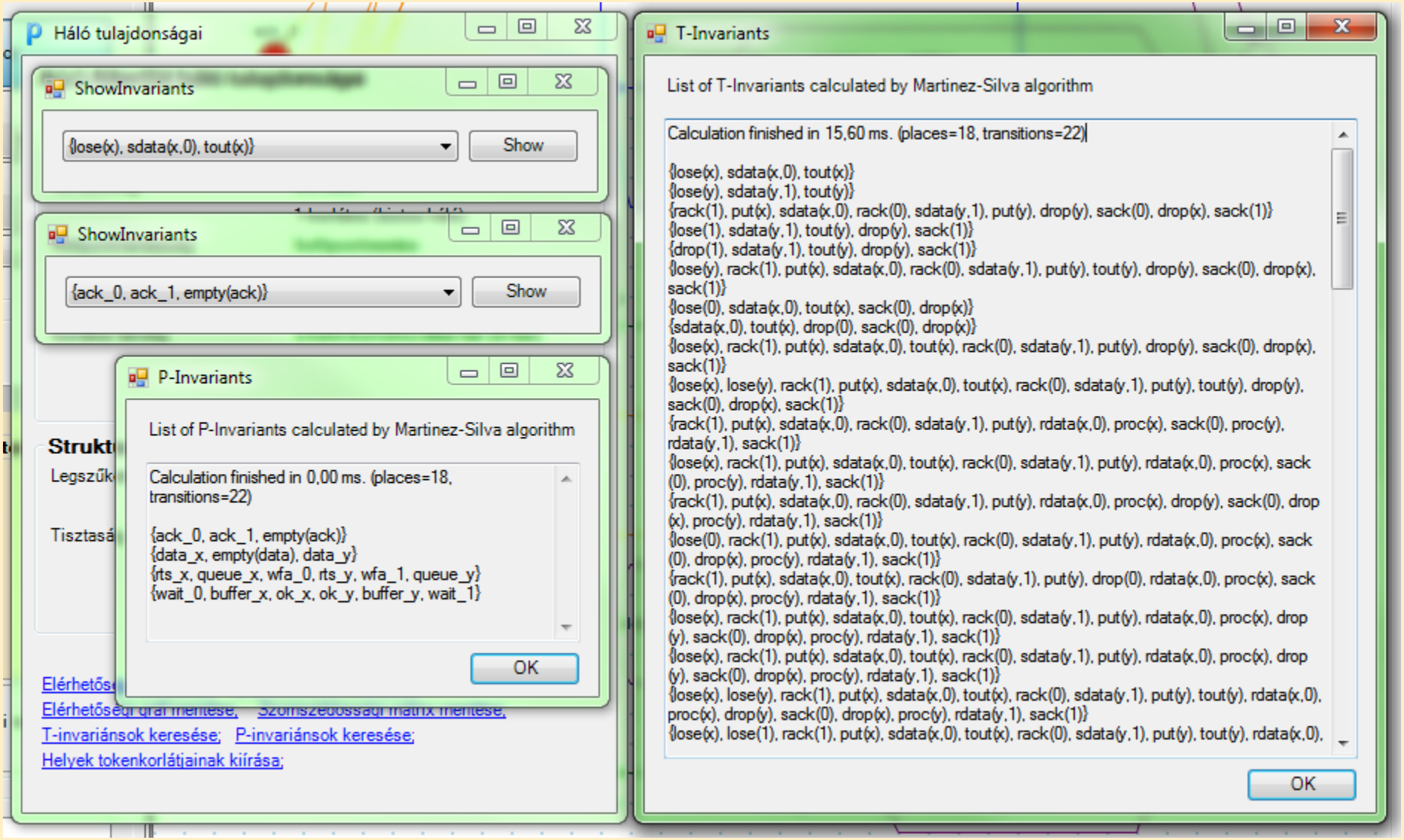
Name: /home/bartha/petrinet/ped/alterbit.inv Size: 940 Bytes

# INA: Egy másfajta megjelenítési formátum

```
semipositive place invariants =

  1 |     12.empty_ack_      :  1,
    |     13.ack_1           :  1,
    |     14.ack_0           :  1
  2 |      0._place_0        :  1,
    |      1._place_1        :  1,
    |      2._place_2        :  1,
    |      3._place_3        :  1,
    |      4._place_4        :  1,
    |      5._place_5        :  1
  3 |     15.empty_data_     :  1,
    |     16.data_y          :  1,
    |     17.data_x          :  1
  4 |      6._place_6        :  1,
    |      7._place_7        :  1,
    |      8._place_8        :  1,
    |      9._place_9        :  1,
    |     10._place_10       :  1,
    |     11._place_11       :  1
```

```
semipositive transition invariants =

  1 |      0._transition_0   :  1,
    |      1._transition_1   :  1,
    |      4.sack_0_         :  1,
    |      5.rdata_x_0_      :  1,
    |      6.sack_1_         :  1,
    |      7.rdata_y_1_      :  1,
    |     12._transition_12  :  1,
    |     13._transition_13  :  1,
    |     14.sdata_y_1_      :  1,
    |     15.rack_1_         :  1,
    |     16.rack_0_         :  1,
    |     17.sdata_x_0_      :  1
  2 |     10.tout_y_         :  1,
    |     14.sdata_y_1_      :  1,
    |     20.lose_data_y_    :  1
  3 |      2.drop_y_         :  1,
    |      6.sack_1_         :  1,
    |      9.drop_1_         :  1,
    |     10.tout_y_         :  1,
    |     14.sdata_y_1_      :  1
```

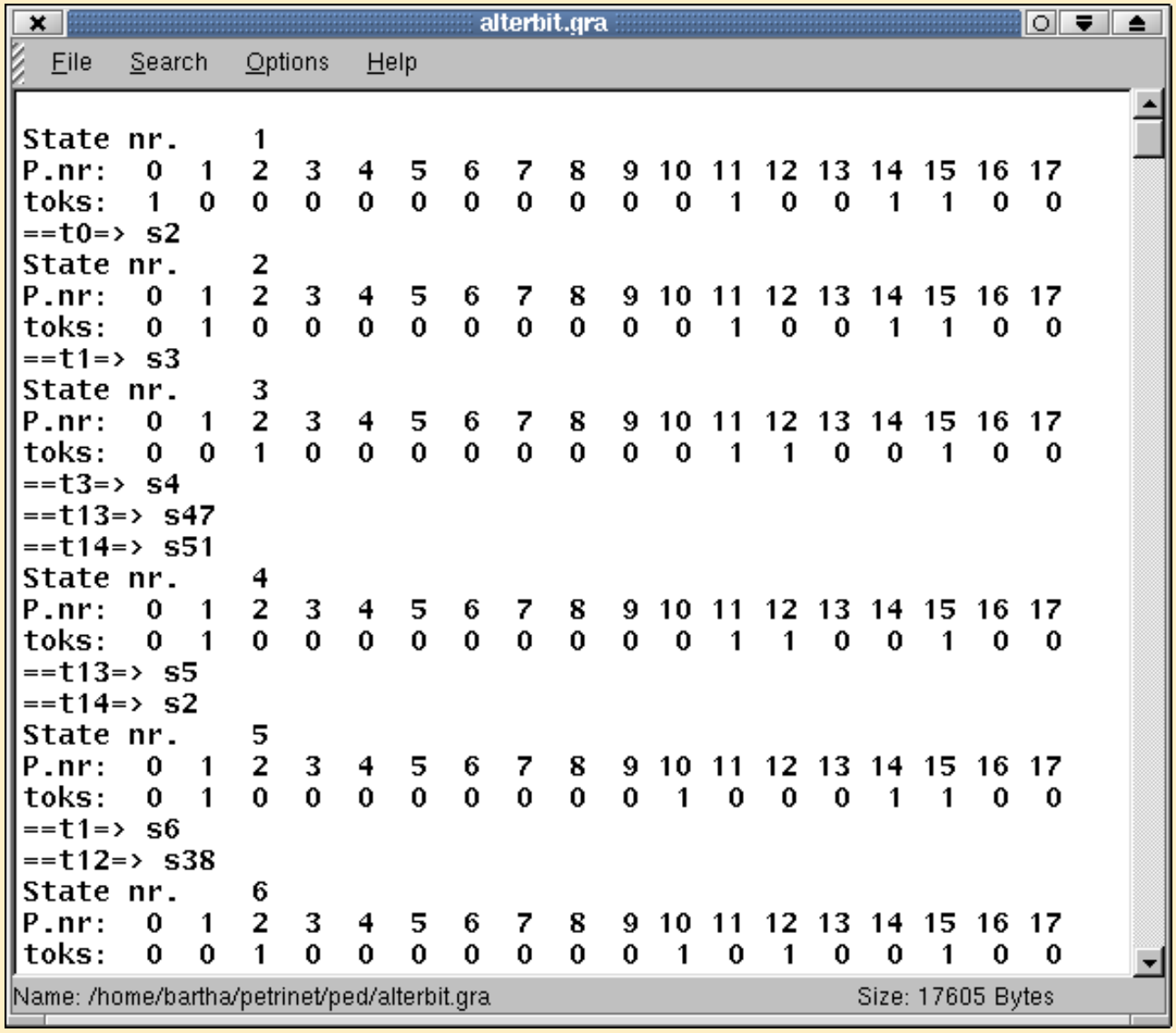
# PetriDotNet: Invariáns analízis



# DNAnet: Fedési (elérhetőségi) gráf (részlet)

```
Current: 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0
Child: 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0  (main.send(x))

Current: 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0
Child: 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1  (main.sdata(x,0))

Current: 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1
Child: 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1  (main.tout(x))
Child: 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0  (main.rdata(x,0))
Child: 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0  (main.ldata(x))

Current: 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0
Child: 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0  (main.tout(x))

Current: 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0
Child: 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0  (main.tout(x))
Child: 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0  (main.recv(x))

Current: 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0
Child: 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0  (main.tout(x))
Child: 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0  (main.sack(0))

Current: 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0
Child: 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 1 0  (main.rack(0))
Child: 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0  (main.tout(x))
Child: 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0  (main.lack(0))
```

# INA: Elérhetőségi gráf



# PetriDotNet: Elérhetőségi gráf rajzolása (GraphViz)