

# 4. előadás

# Petri hálók kiterjesztések

ismétlés

# Dinamikus viselkedés

Petri hálók „működésének” egy lépése:

- Állapot megváltozása: tranzíciók „tüzelése”
  - korábbi állapot: kezdeti token eloszlás vektor
  - tüzelés végrehajtása
    1. engedélyezettség vizsgálata
    2. tokenek elvétele a bemeneti helyekről
    3. tokenek kirakása a kimeneti helyekre
  - új állapot: megváltozott token eloszlás vektor

# Állapotátmenet

Tüzelés végrehajtása:

- Engedélyezett tranzíció tetszése szerint tüzelhet vagy nem
  - “fire at will”
- Több tranzíció engedélyezett: konfliktus
  - egy lépésben csak egy engedélyezett tranzíció tüzelhet
  - konfliktusfeloldás véletlen választással
- **$\mathcal{P}$  Nemdeterminisztikus működés**

A tranzíció tüzelése

- elvesz  $w^-(p, t)$  darab tokent a  $p \in \mathbf{I}$   $t$  bemeneti helyekről
  - $w^-(p, t)$  a  $p \rightarrow t$  él súlya
- elhelyez  $w^+(t, p)$  darab tokent a  $p \in \mathbf{tI}$  kimeneti helyekre
  - $w^+(t, p)$  a  $t \rightarrow p$  él súlya

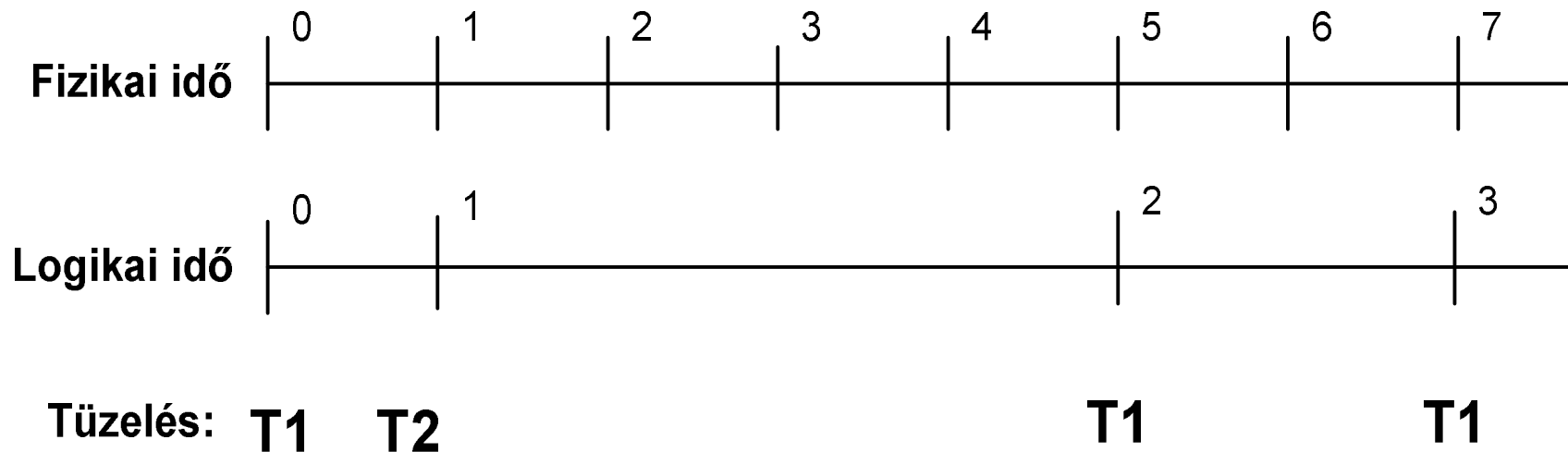
# **A nemdeterminisztikus működés kiküszöbölése**

- A nemdeterminisztikus működés nem mindig engedhető meg
- kiküszöbölésére különböző korlátozásokat vezethetünk be rendszerünkben
- Szimulációs problémák

# Kiterjesztések okai

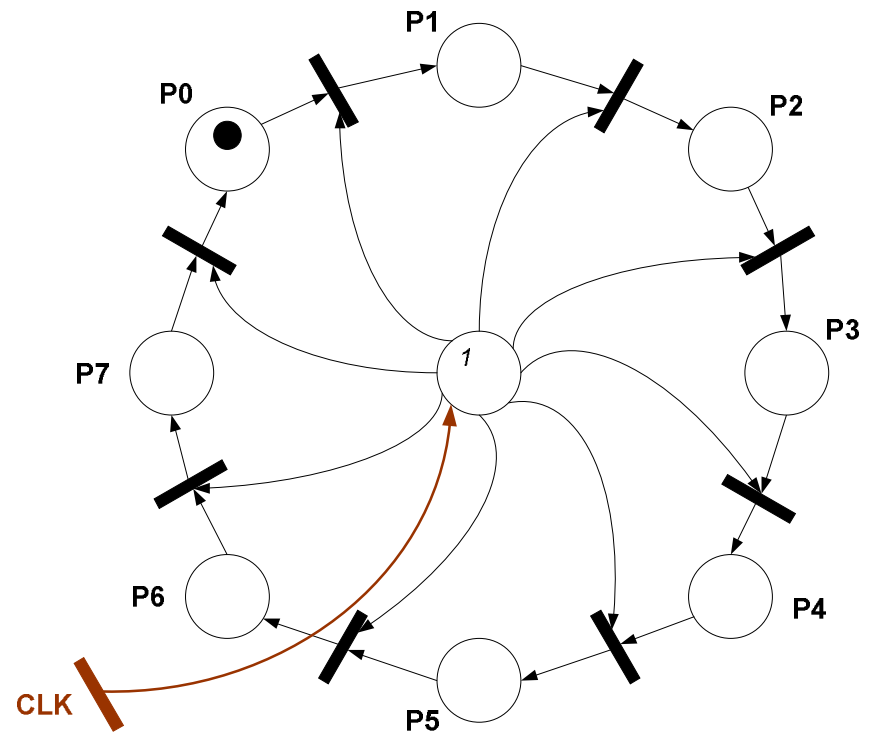
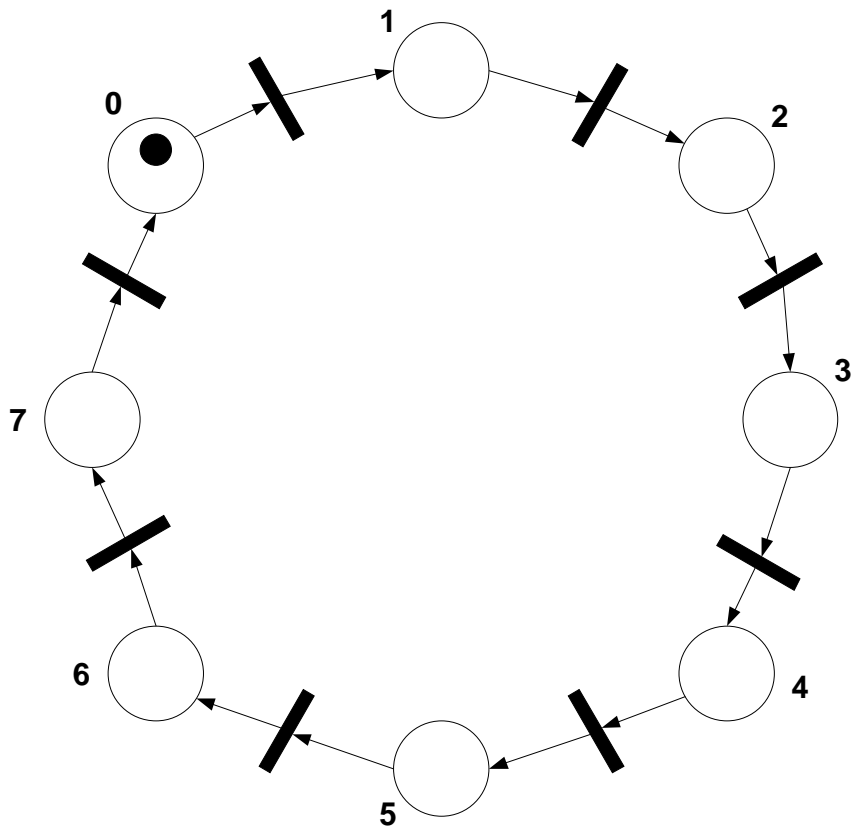
- Időkorlátok (valós és logikai idő, implicit és explicit idő),
- Átláthatóság,
- Végtelen körök,
- Nem-determinisztikus működés...

# Időkorlát



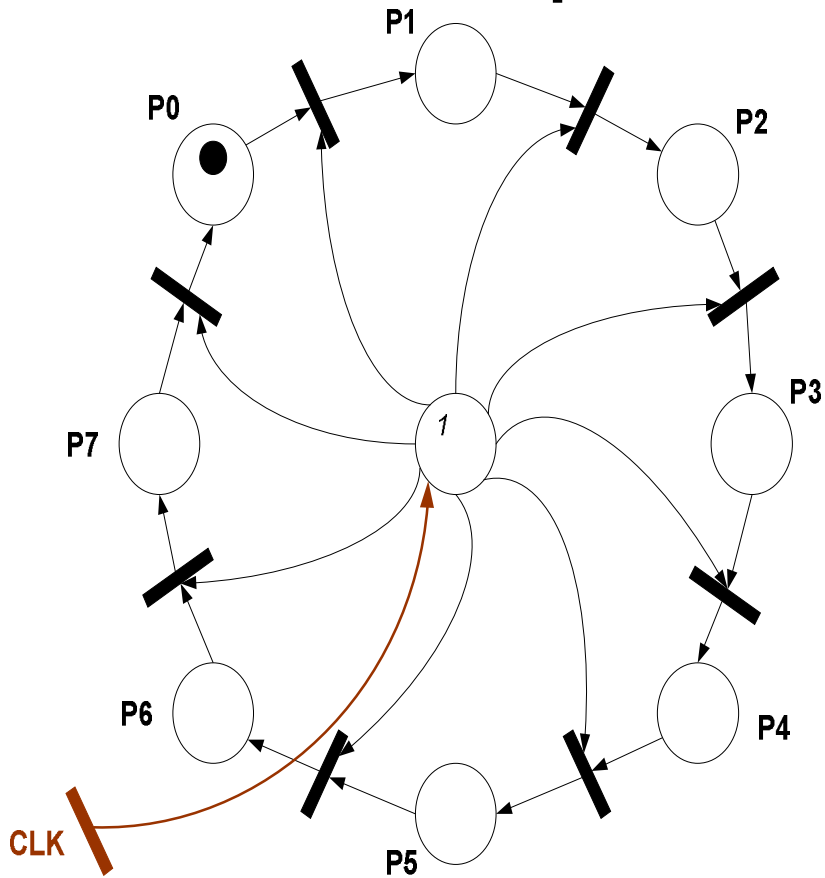
implicit időfogalmat kapcsolunk rendszerünkhöz, amelyben egy engedélyezett tranzíció tüzelése a  $[0, \infty)$  intervallumban bárhol megtörténhet (fire-at-will).

# 1. példa: nyolcas számláló



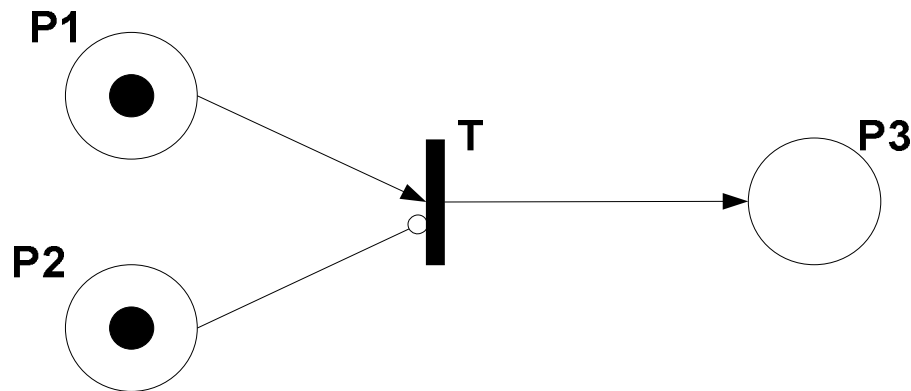


# Vezessünk be órajelet kapacitáskorláttal!



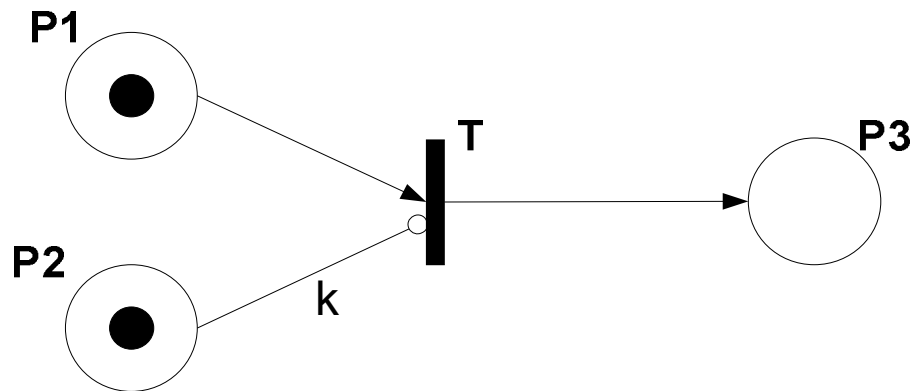
- Korlátozzuk tehát a közepső állapot kapacitását egyetlen tokenre (ezt jelöli a beleírt 1-es). Ez azt jelenti, hogy azon a helyen maximum 1 token lehet egyszerre, tehát bemenő tranzíció nem tüzelhet, amíg a tüzelés túllépné a **kapacitáskorlátot**.
- Így ha esetünkben egyszer már ütött az óra, tehát van órajel-token a rendszerben, akkor mindaddig nem üthet újra az óra, amíg ez el nem tűnik, azaz amíg a számláló nem lép egyet. Így biztosíthatjuk azt, hogy minden órajelre egyet és pontosan egyet lépjen a számlálónk.

# Használjunk tiltó éleket!



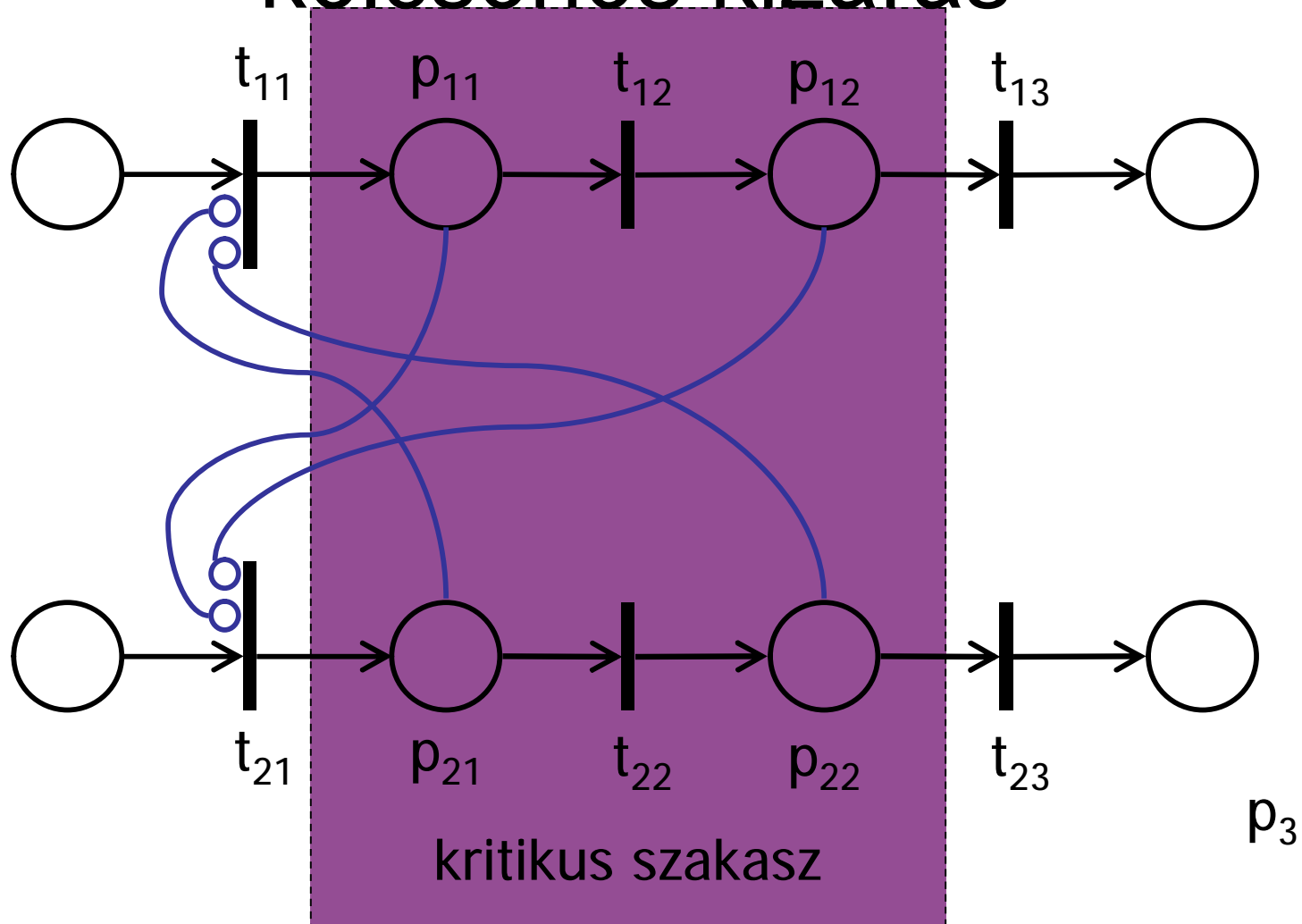
- A kapacitáskorlát helyett **tiltó éleket** veszünk fel a hálóba.
- A tiltó él azt jelöli, hogy a tranzíció *ne* tüzessen, amíg az adott feltétel teljesül. Az alábbi ábrán például addig nem tüzeli a tranzíció, amíg a P2 helyen van token

# Használjunk tiltó élet!

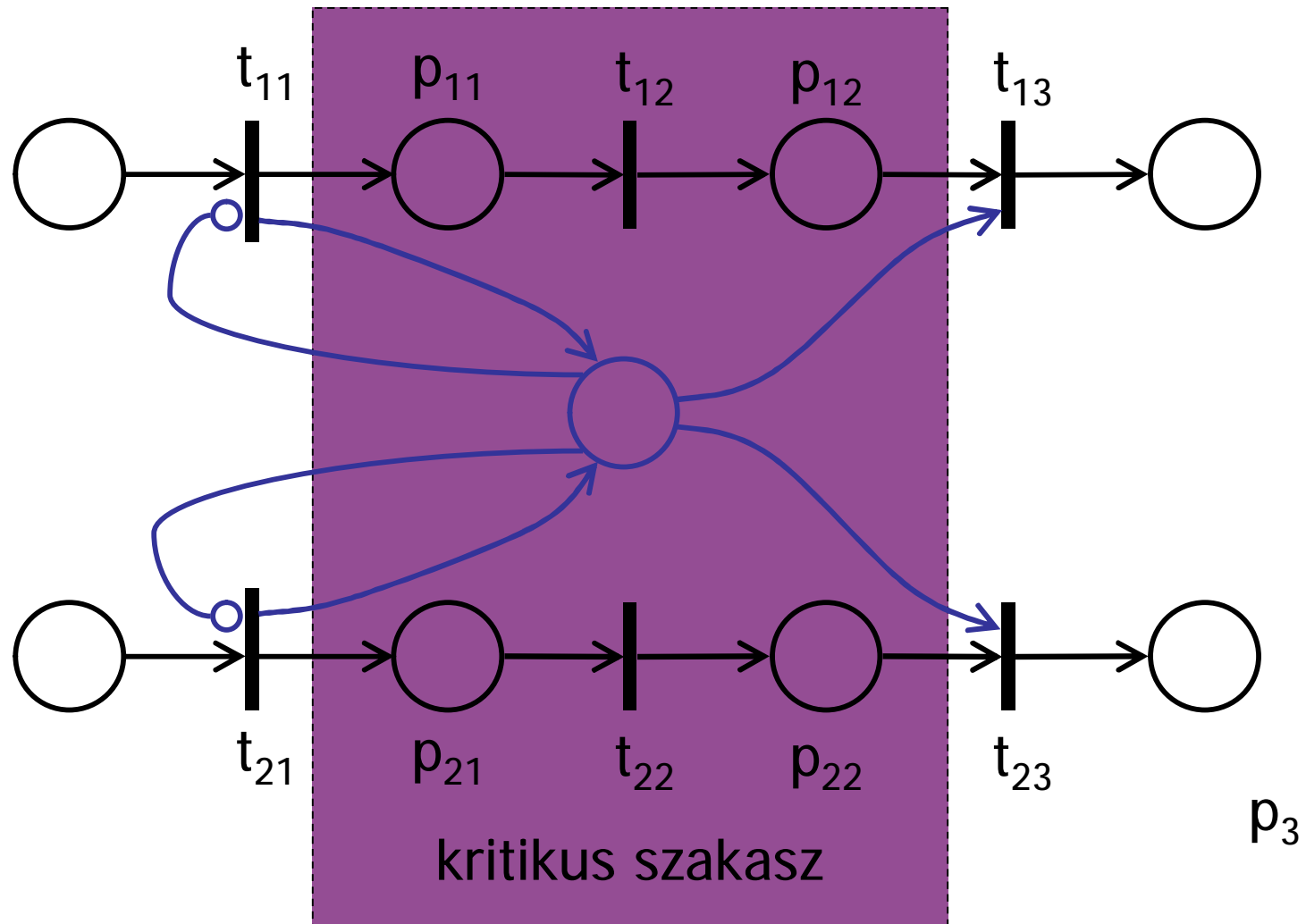


- Ha a tiltó élhez egy  $k$  súlyt is rendelünk, az azt jelenti, hogy ha az él bemeneti helyén az adott  $k$  számú, vagy annál több token van, akkor a tranzíció tiltott, ha  $k$ -nál kevesebb token szerepel a helyen, akkor a tranzíció engedélyezett.
- Ezzel a módszerrel azt köthetjük tehát ki, hogy mindaddig nem jöhet óraütés, amíg van token a megfelelő helyen

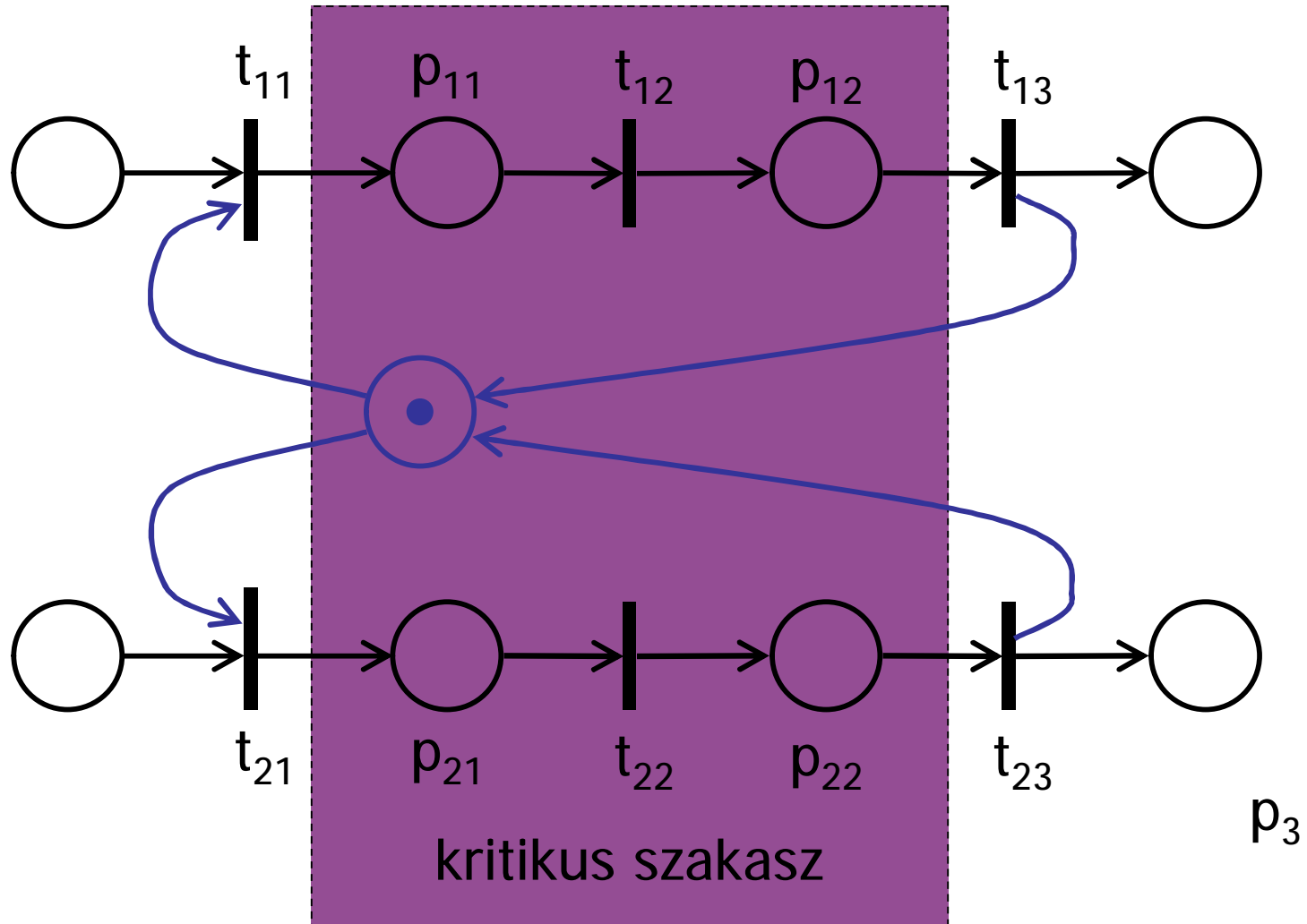
## 2. példa tiltó él alkalmazására: kölsönös kizárás



# Ugyanez másképp:

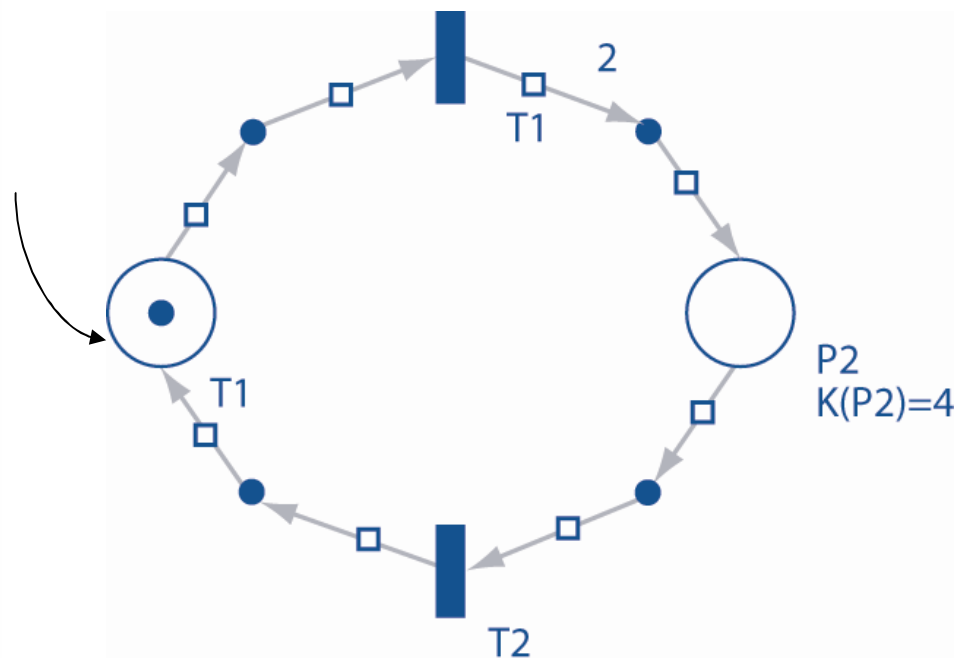
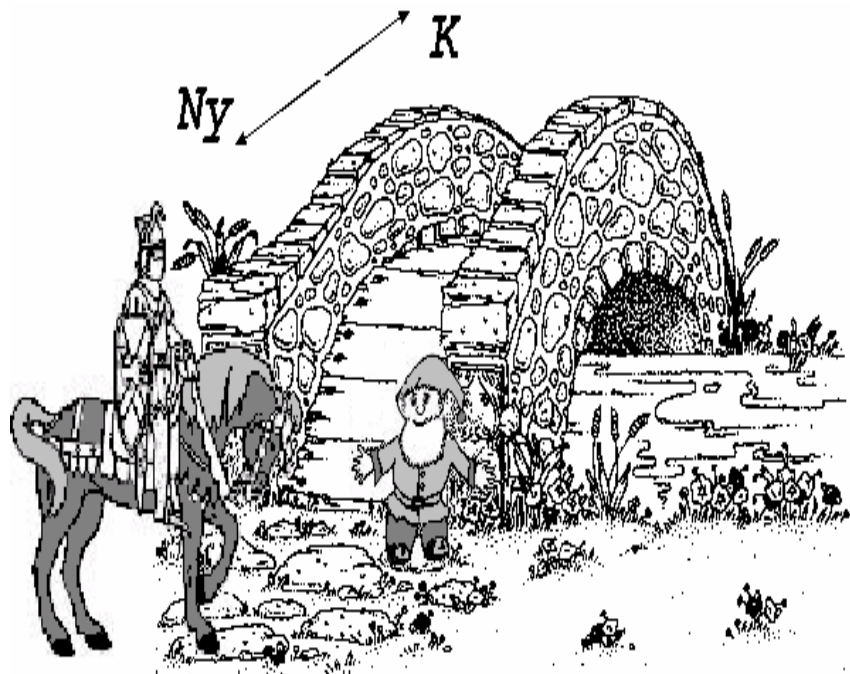


# A legegyszerűbben:

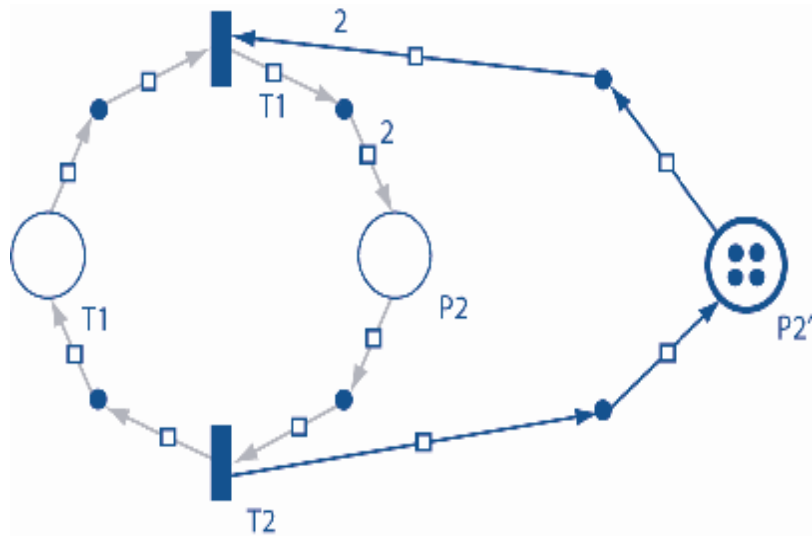


# 3. példa

## A kapacitáskorlát kiküszöbölése



- a kiindulási helyzetben egyetlen tokenünk van, a  $P1$  helyen. A  $T1$  tranzakció tüzelési feltétele, hogy  $P1$ -en legyen token, ezúttal ez teljesül. A  $T2$  tranzakció jelenleg nem engedélyezett, hiszen  $P2$ -ben nincs token.



- Ha azt akarjuk elérni, hogy a  $P_2$  helyen maximum  $k=4$  darab token legyen, be kell vezetnünk egy **adminisztrációs helyet** ( $P_2'$ ), ahol azt tartjuk számon, hogy a  $P_2$  helyre hány token fér még el (nyugat). Így az adott helyen lévő tokenek számát  $m$ -mel jelölve mindig igaz lesz az alábbi összefüggés:  $m(P_2') + m(P_2) = k$ .



# A nemdeterminisztikus működés kiküszöbölése Prioritással

- Tranzíciókhoz rendelt **prioritás**
- Az engedélyezett tranzíciók közül egy alacsonyabb prioritású mindaddig nem tüzelhet, amíg van
  - engedélyezett ÉS
  - magasabb prioritású tranzíció
- **Prioritási szinten belül** továbbra is **nemdeterminisztikus választás!**

# Petri hálók bővített formális definíciója

- Petri háló:
- Helyek halmaza:
- Tranzíciók halmaza:
- **Prioritás:**
- Élek halmaza:
- Súlyfüggvény:
- Kezdőállapot:
- PN struktúra:
- PN adott kezdőállapottal

$$PN = (P, T, \Pi, E, W, M_0)$$

$$P = \{p_1, p_2, \dots, p_\pi\}$$

$$T = \{t_1, t_2, \dots, t_\tau\}$$

$$P \cap T = \emptyset$$

$$\Pi : T \longrightarrow \mathbb{N}$$

$$E \subseteq (P \times T) \cup (T \times P)$$

$$w^* : E \longrightarrow \mathbb{N}^+$$

$$M_0 : P \longrightarrow \mathbb{N}^{|P|}$$

$$N = (P, T, \Pi, E, W)$$

$$(N, M_0)$$

# **Szimuláció Petri-hálókkal**

- A Petri-háló tevékenységeit, akcióit olyan elemi (atomi) eseményekre bontjuk, amelyek tovább már nem oszthatóak.
- Esetünkben atomi eseménynek mondunk egy tranzíció tüzelését, így az összetett események a tüzelési szekvenciákat jelentik (az egymás után végrehajtható tüzelések sorozatát).
- Az eseményeket a rendszerben szereplő állapotváltozókkal szimuláljuk.

Petri-hálók számítógépes szimulációjához több dolgot  
figyelembe kell venni:

A nemdeterminisztikus viselkedés miatt az  
időzítéseket (ál)véletlen módon kell megoldani,  
tehát ha több tüzelhető tranzíció is van egy adott  
időpillanatban, véletlenszerűen válasszuk ki azt,  
amelyik a következő alkalommal valóban tüzelni  
is fog.

(Ilyenkor azonban pl. fennáll annak a veszélye,  
hogy ha egy tranzíció tüzel, olyan tranzíciók is  
tiltottá válhatnak, amelyek az előző lépésben  
még engedélyezettek, illetve olyanok válhatnak  
engedélyezetté, amelyek korábban tiltottak  
voltak)

# Tüzelési szekvencia

- Állapotátmeneti trajektória
  - egymást követő tüzelések hatására felvett állapotok
- Tüzelési szekvencia
- $\underline{\sigma} = \langle M_{i_0} t_{i_1} M_{i_1} \dots t_{i_n} M_{i_n} \rangle \rightarrow \langle t_{i_1} \dots t_{i_n} \rangle$
- Ha az összes tranzíció kielégíti a tüzelési szabályt:
  - $M_{i_n}$  állapot  $M_{i_0}$ -ból elérhető a  $\underline{\sigma}$  tüzelési szekvencia által:
- $M_{i_0} [\underline{\sigma} > M_{i_n}$

# A tüzelési szemantika módosítása

- Petri hálók működési nemdeterminizmusának korlátozása
  - Prioritás rendelése a tranzíciókhoz
  - Kapacitás rendelése a helyekhez
  - Tiltó élek bevezetése

# Diszkrét rendszermodellezés

-dr. Pataricza András, dr. Bartha Tamás:  
Diszkrét rendszermodellezés alapjai  
c. előadása alapján

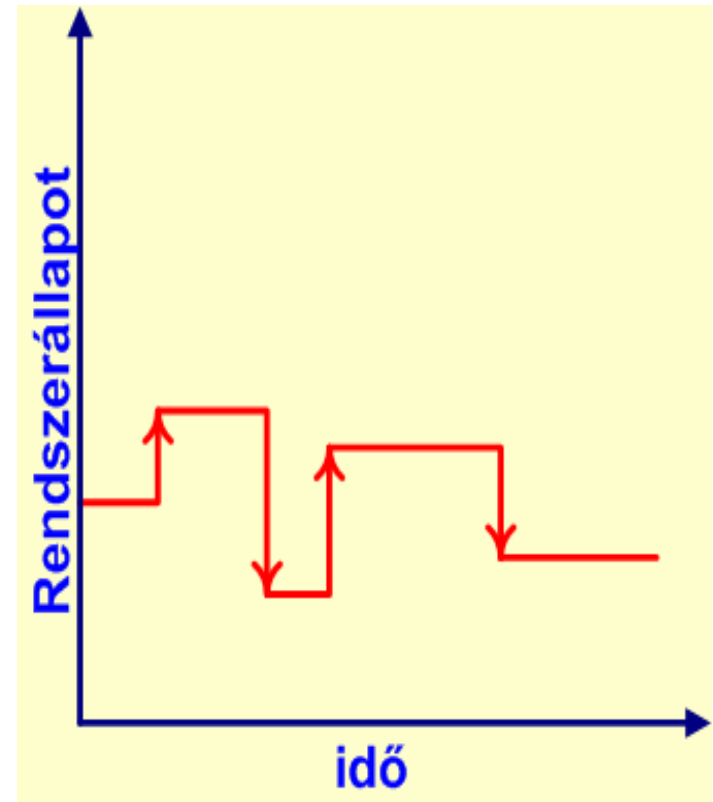


# Diszkrét rendszermodellezés célja

- Informatikai rendszerek: jól tagoltak
  - rendszerépítés a komponensek integrációjával
  - elemi komponensek kapcsolata
    - explicit logikai kapcsolat: sorrendiség, ok-okozati függőség
    - implicit függőség: pl. osztott erőforrás használata
- Célkitűzés: minőségi vagy/és mennyiségi analízis

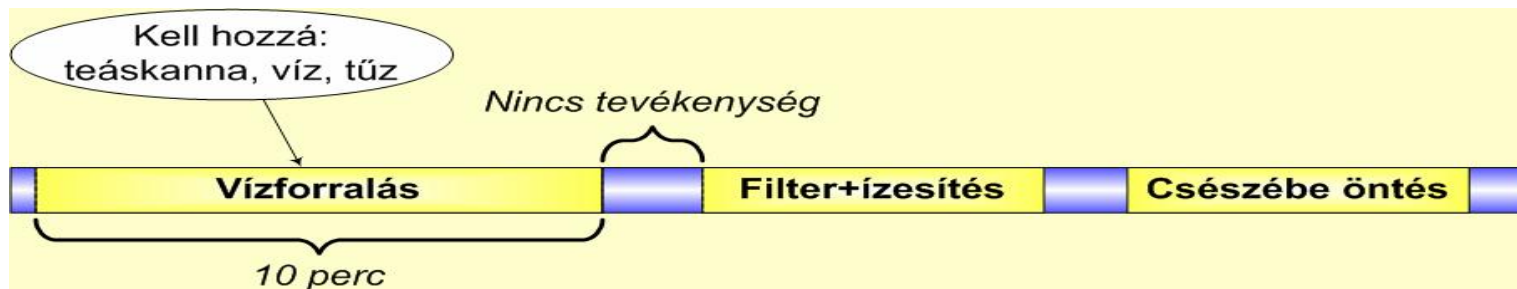
# Rendszermodellek felbontása

- Modellezés célja
  - Korreláció: modell  $\leftrightarrow$  „valóság”
  - Érthetőség, áttekinthetőség
    - Nem cél a minimális modell
  - A teljes rendszer dinamikája
    - Komponensek modellje egyszerű
    - Bonyolult kölcsönhatások
- Csoportosítás
  - Folytonos
  - Diszkrét
  - értékben
  - időben



# Folyamatorientált modellezés

- Elemi tevékenységekből áll
  - Jellemzői:
    - Erőforrást igényel
    - Mennyi ideig tart
  - Folyamat: tevékenységek logikailag rendezett sora
- + járulékos információk:
- erőforrások
  - kölcsönhatások, pl.
  - erőforrás használat



# Tevékenységek

- Modellezés hierarchikus és funkcionális
  - (rész)folyamatok
  - tovább nem bontott, elemi lépések → tevékenységek
- Tevékenység
  - elemi lépés
  - felhasznált erőforrások
  - végrehajtási idő
    - determinisztikus
    - sztochasztikus

# Tevékenységek modellezése Petri hálóokban

- elemi lépés  $\rightarrow$  tranzíció tüzelése
- felhasznált erőforrások  $\rightarrow$  bemeneti / kimeneti helyek
- végrehajtási idő
  - determinisztikus  $\left. \begin{array}{l} \rightarrow \blacksquare \rightarrow \\ \rightarrow \square \rightarrow \end{array} \right\}$  determinisztikus időzítésű tranzíció
  - sztochasztikus  $\left. \begin{array}{l} \rightarrow \blacksquare \rightarrow \\ \rightarrow \square \rightarrow \end{array} \right\}$  exponenciális időzítésű tranzíció

Az engedélyezettséggel kapcsolatban felmerülő kérdések:

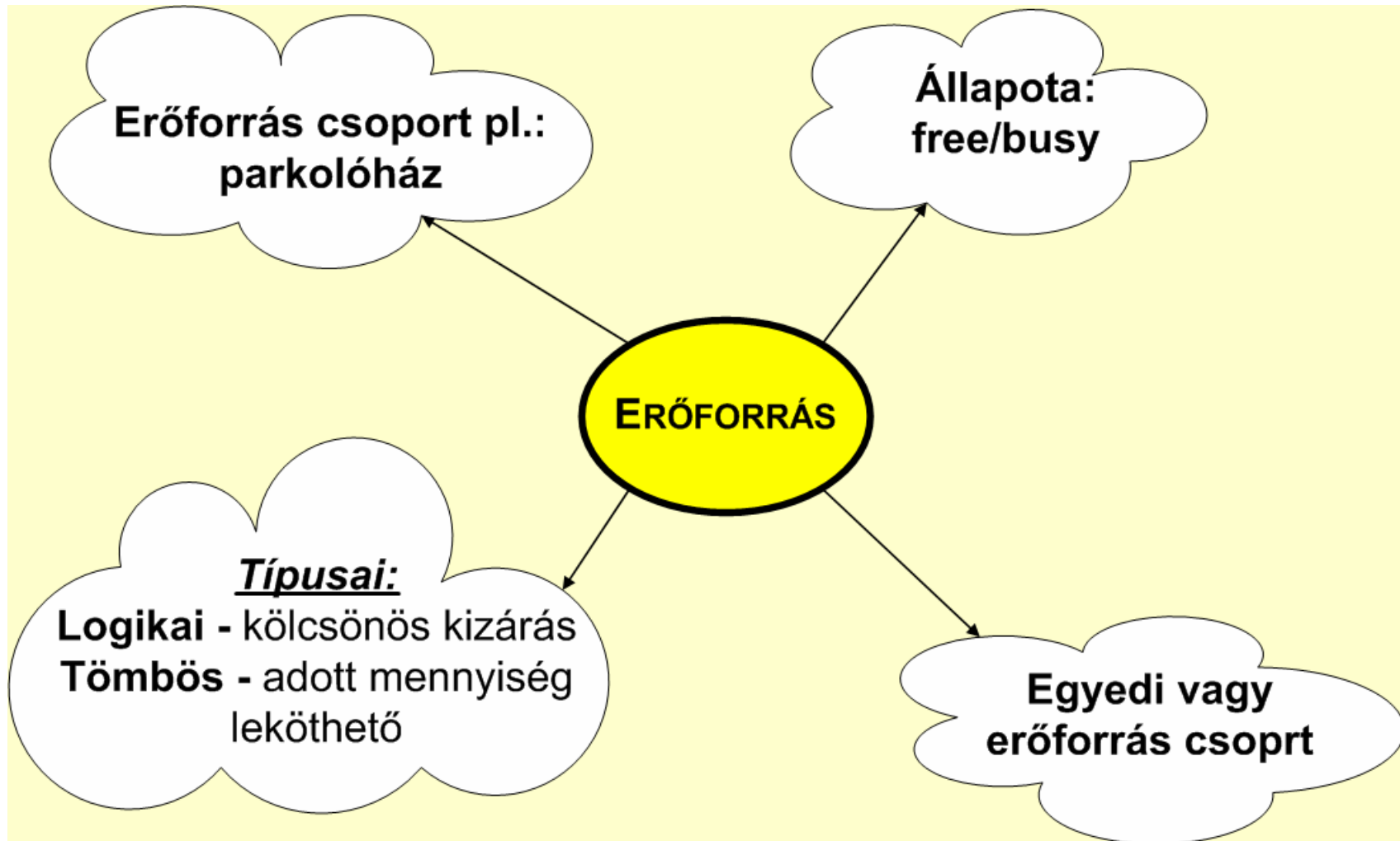
- Időzítetlen tranzíciók „prioritása” nagyobb: előbb tüzelnek
- Engedélyezettség megszűnése: mi lesz a gyújtási idővel?
  - újrainduló (újra sorsol)
  - megőrződő (korábban sorsoltból fennmaradó idő folytatódik)

## Sztochasztikus Petri hálók

$$\mathbf{SPN} = (P, T, E, W, M_0, \Lambda)$$

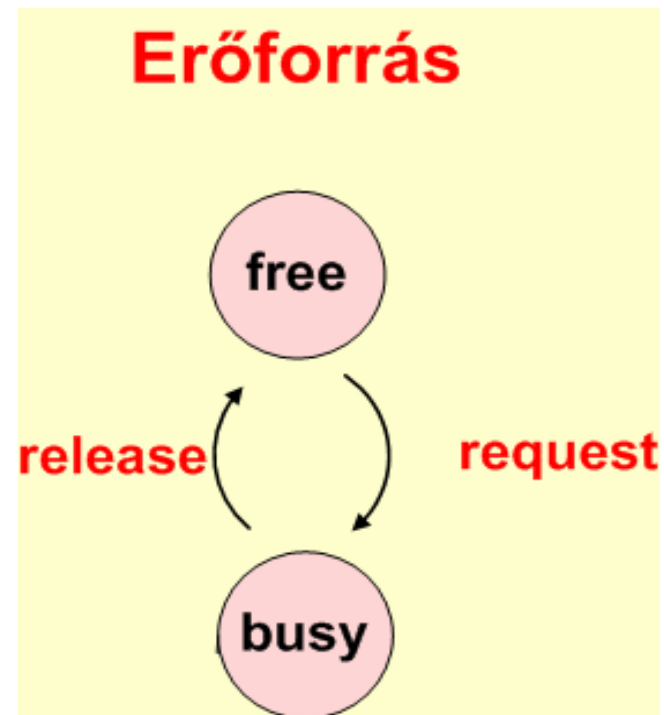
- tüzelési intenzitás  $\Lambda : T \rightarrow \mathbb{R}$
- időzítés:  $\tau$  valószínűségi változó
  - eloszlásfüggvény:  $P\{t < t\} = F(t) = 1 - e^{-\Lambda t}$
  - sűrűségfüggvény:  $f(t) = \Lambda e^{-\Lambda t}$
- tüzelési szabály változik: intenzitás  $\sim$  prioritás
- két exponenciális eloszlás minimuma is az:  $\lambda_1 + \lambda_2$
- PT és SPN elérhetőségi gráfja azonos
- időzített elérhetőségi gráf: folytonos Markov lánc

# Erőforrások



# Erőforrások modellje

- **Állapota**
  - Free
  - Busy
- **Mennyisége**
  - Egy
  - Több



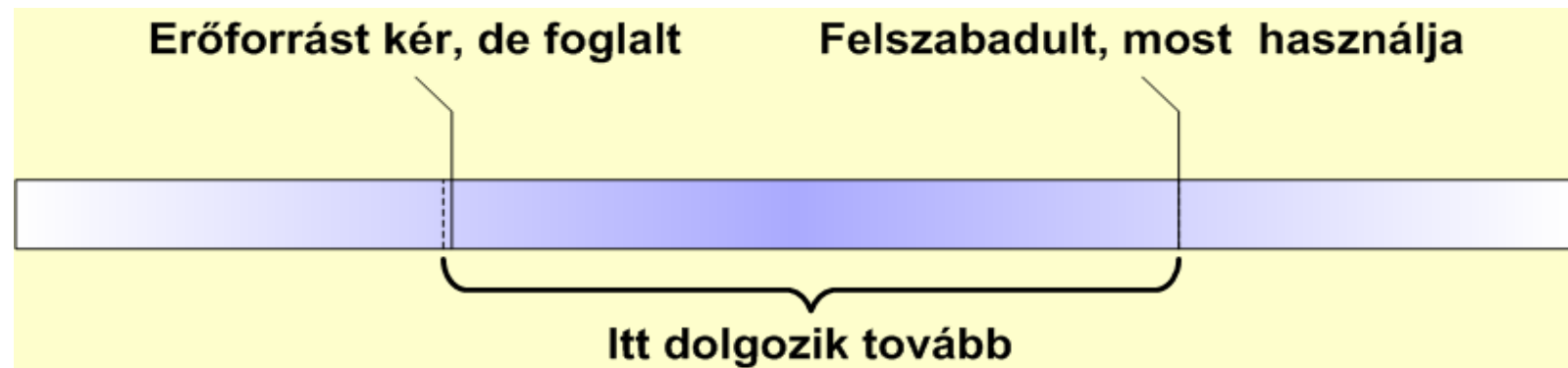


# Blokkoló / nem blokkoló erőforrás

- Blokkoló

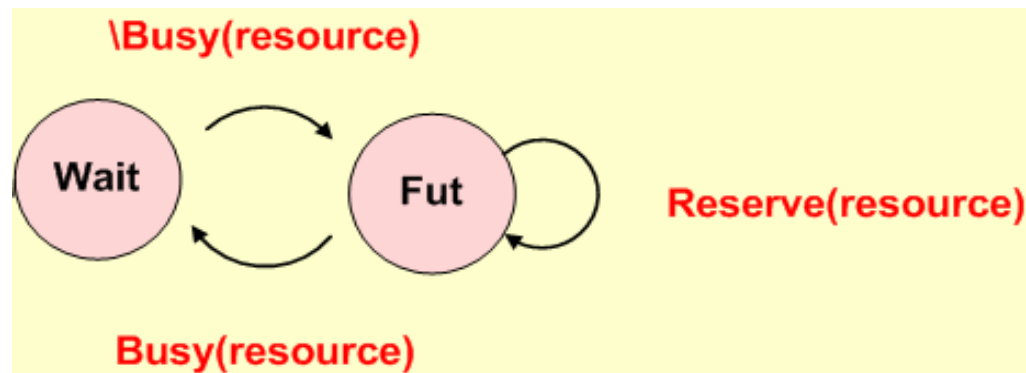


- 



# Események

- Rendszerszintű modellezésben eseménynek tekinthető:
  - Valamely tevékenység kezdete, illetve vége
  - Az erőforrások állapotváltozása
- Tevékenység mit tud csinálni?
  - Erőforrást foglal: Reserve (resource\_list)
  - Erőforrásra vár: Wait (usage\_time)
  - Erőforrást elenged: Release (other\_resources\_list)



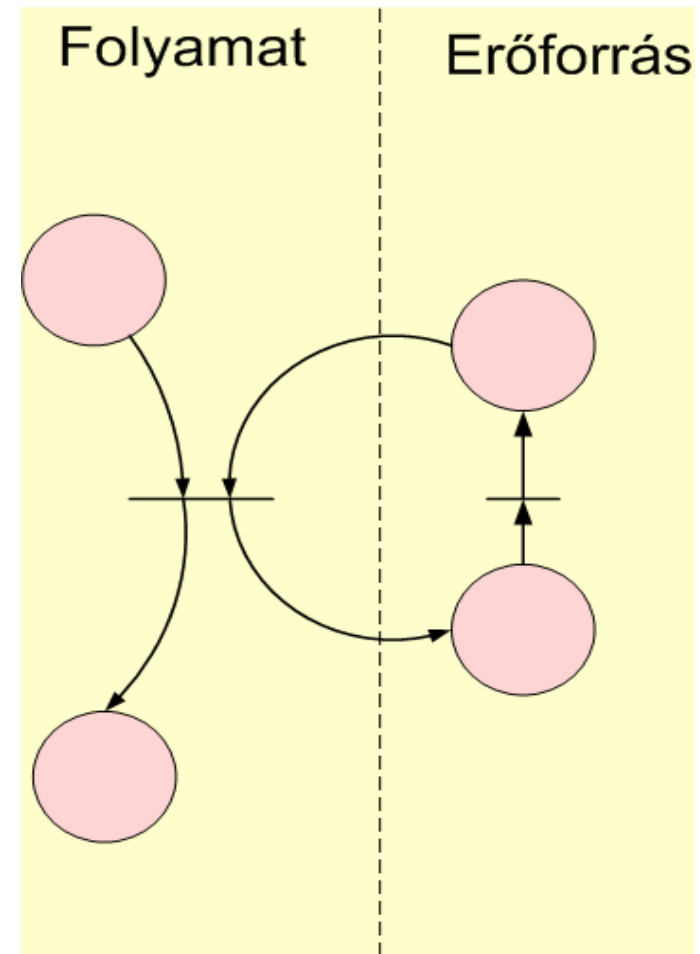
# Erőforrás allokáció Petri hálókbán

- Kölcsönös kizárás
- Több darab lefoglalása

Cél:

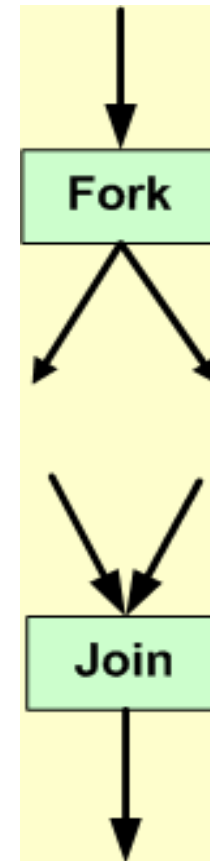
nem a minimális, hanem az

**ÉRTHETŐ PETRI HÁLÓ!**

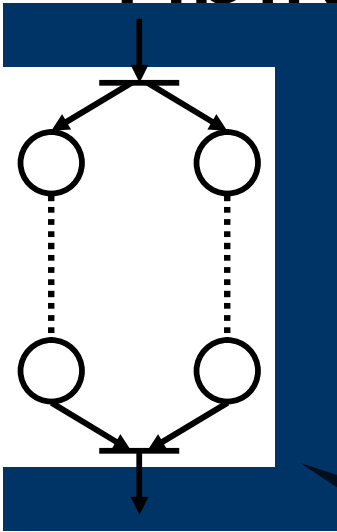


# Üzenetek

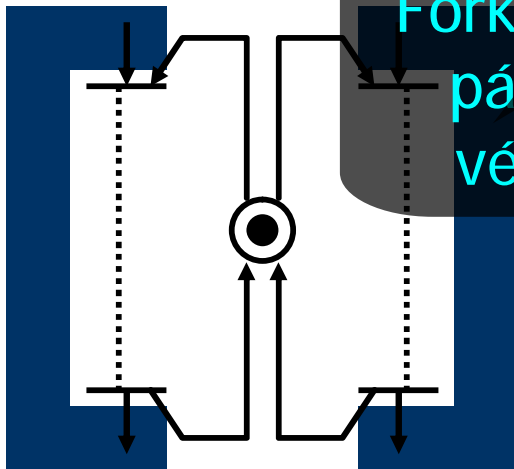
- Szoftverben párhuzamosság
  - FORK - elágazás
  - JOIN - visszatérés
- Kommunikáció biztosítása
  - üzenetekkel
- Wait – egymásra várás



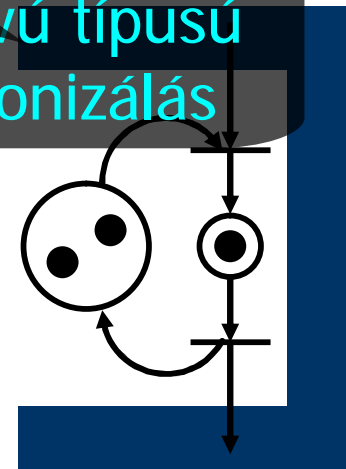
# Tipikus modellek konstrukciói



Körkorlátozott erőforrás állapotváltozó kapacitás modellezése



Fork-Join Szemafor típusú devú típusú párhuzamos végrehajtás szinkronizálás



# Rendszermodellezés

Modellépítés folyamata:

- a három fő modellelem-fajta:
  - folyamatok, illetve tevékenységek
  - erőforrások
  - a logikailag csatolt folyamatok közti esetleges üzenetek
- felépítése először egyedileg
- majd ezekből a modell összeintegrálása

# Rendszermodellezés

Modellépítés elemei:

1. a folyamat modellje (az erőforrás használat, illetve üzenetváltás részletes feltüntetése nélkül)
2. minden egyes erőforráshoz fel építeni
  - a foglalt/szabad állapotot jellemző kétállapotú véges automata modellrészt,
  - valamint az üzenetek pufferjének modelljét
3. a folyamat és erőforrás (illetve az üzenetek pufferje) modelljében a megfelelő állapotátmenetek összevonása

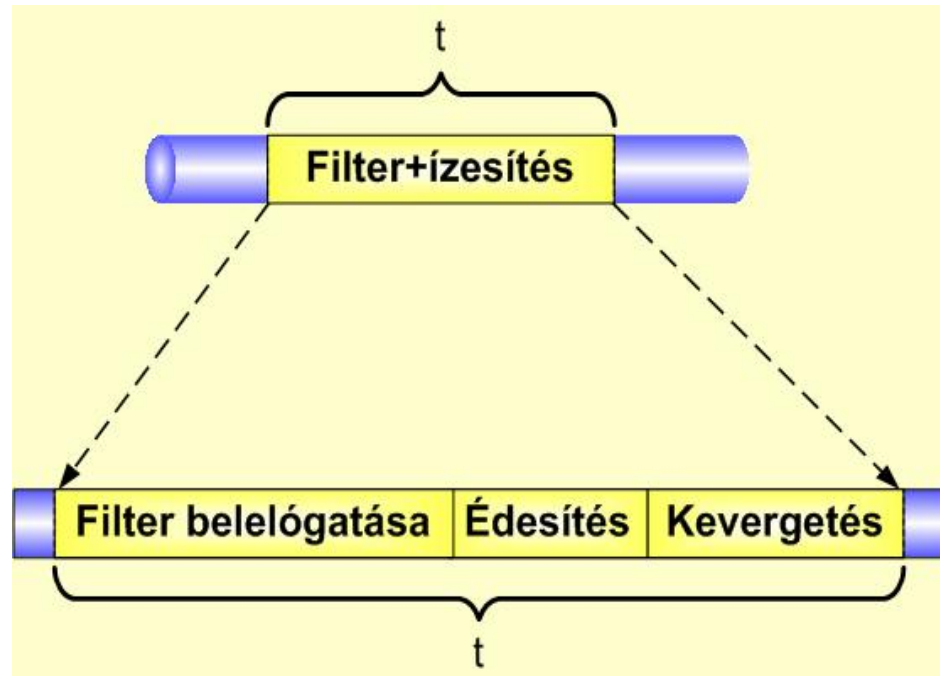
# Hierarchikus modellfinomítás

- Elemi eseményeket több részeseményre bontunk fel
- Az új események összideje = a régi esemény ideje

Kibontás

„egy az egyben”  
behelyettesíthető

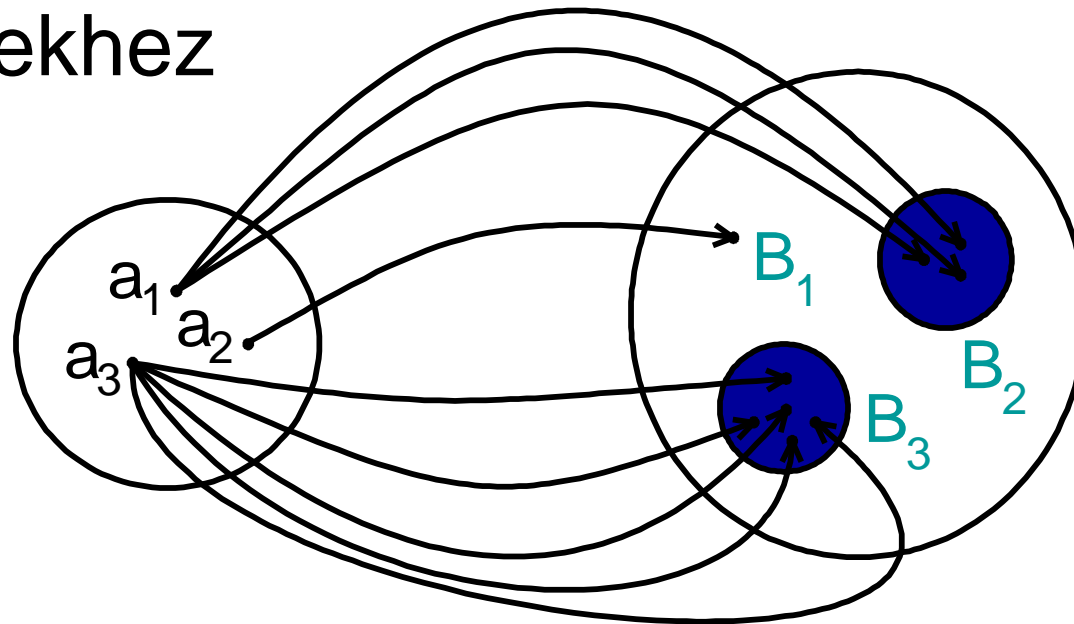
**KOMPOZÍCIONALITÁS**





# Halmazfinomítás

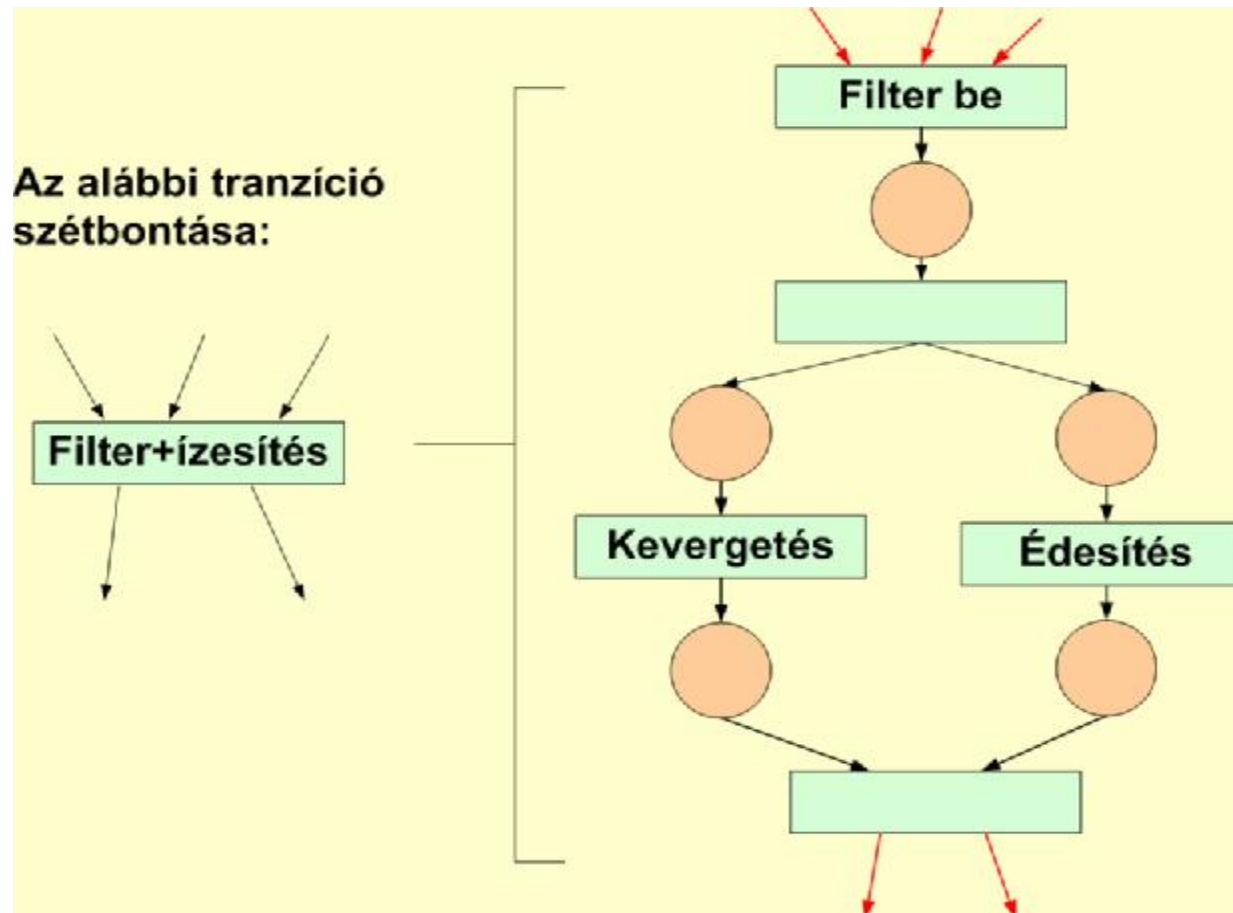
Diszjunkt részhalmazok hozzárendelése  
elemekhez



# Kompozicionalitás (Petri hálók)

- Egy tranzíciót helyettesíthetünk
- A behelyettesítendő gráf
  - Tranzícióval kezdődjön és végződjön
  - Az eredeti tranzíció be/kimenő élei ezekbe menjenek
- Csak akkor használjuk, ha szükséges
  - Kanalat csak a kevergetésre foglalom
    - eredeti modellben nem részletezett tevékenység
  - Komplexitás megnő

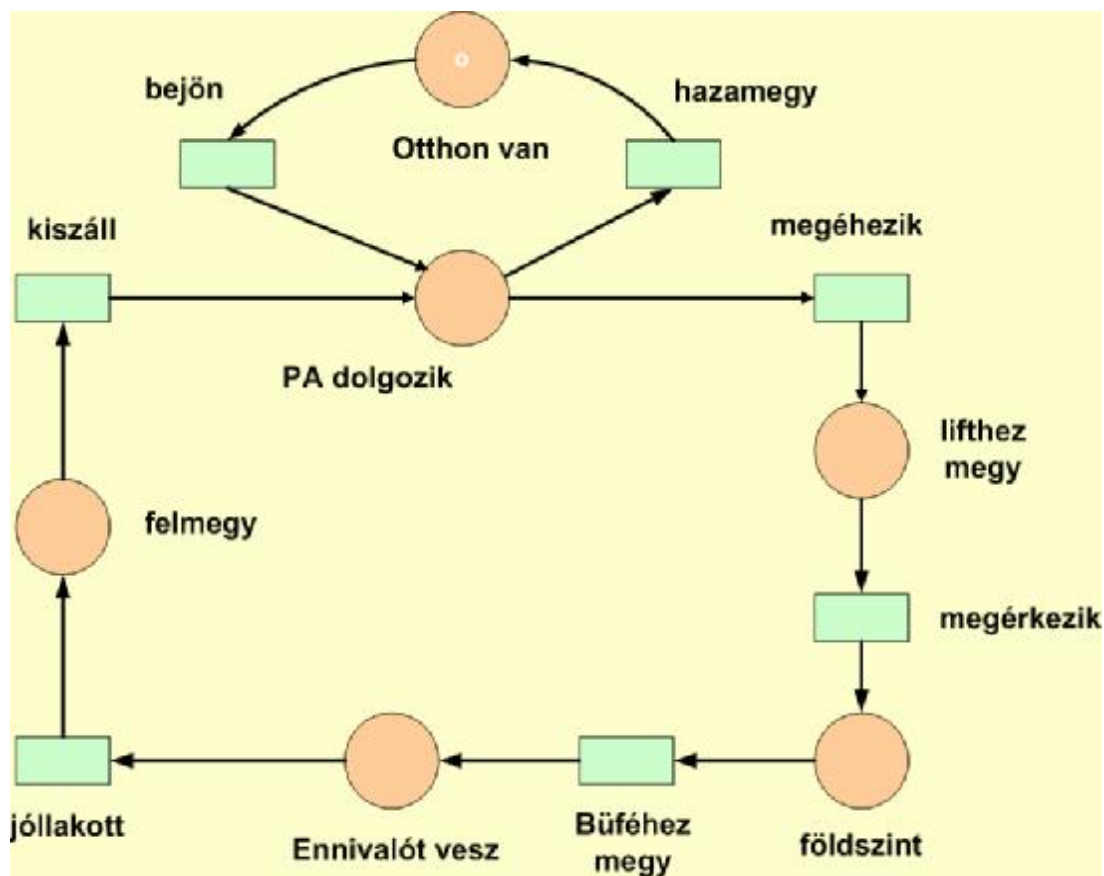
# Finomítás Petri hálónál



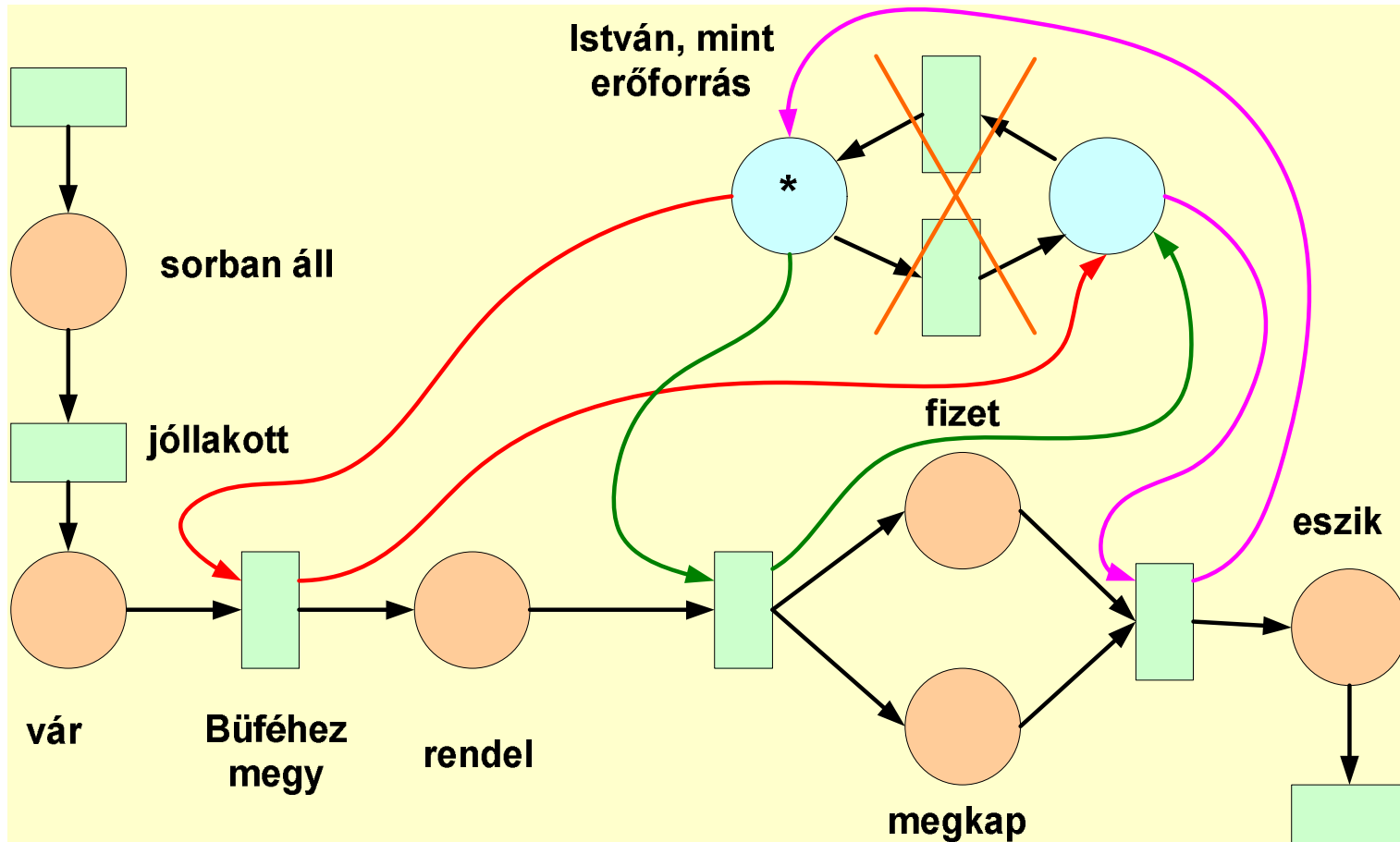
# Példa

## Uzsonnázás

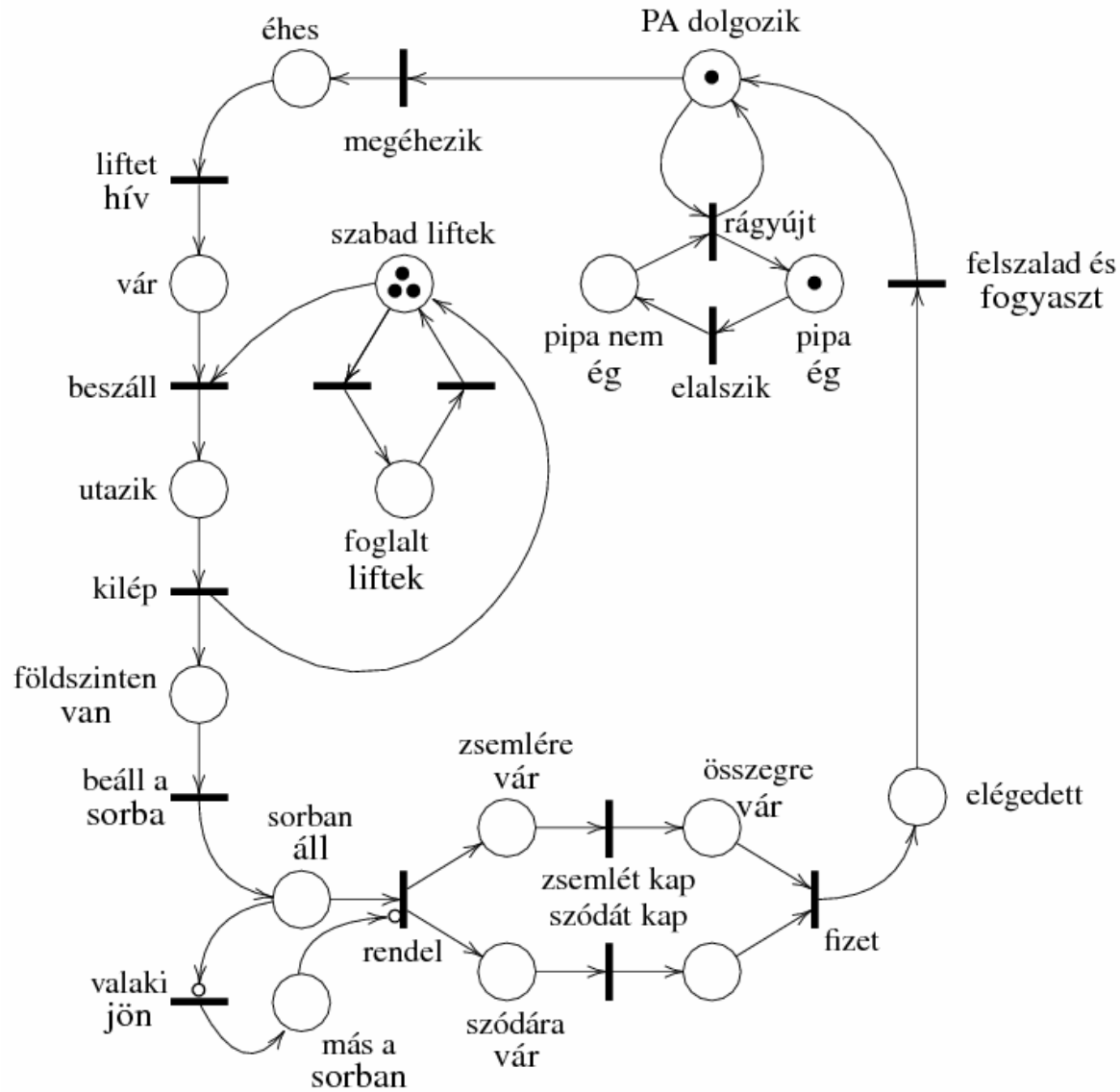
- Erőforrás modellezés
  - professzor (PA)
  - büfés (István)
- Kompozicionalitás
  - étkezés
  - fizetés
- Petri hálónak transzformálása
  - erőforrások kezelése



# Erőforrás modellezés



# Részletesebb modell



# Tiltó él kiváltása

