3.előadás

Források:

-Molnár Ágnes: Formális módszerek az informatikában (2) , NetAkadámia Tudástár

-dr. Pataricza András, dr. Bartha Tamás: Petri hálók: alapfogalmak, formális definíció, kiterjesztések

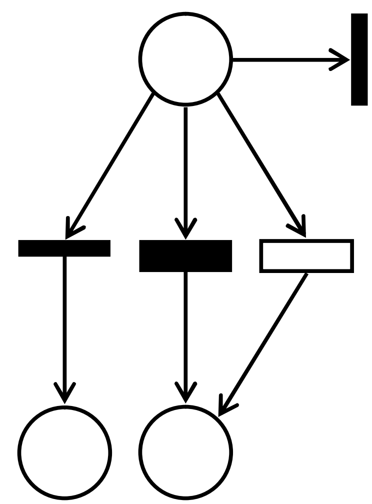
Ismétlés
Petri hálók

- Egyidejűleg:
  - grafikus
  - matematikai reprezentáció
- Struktúrával fejezi ki:
  - vezérlési struktúra
  - adatstruktúra
- Előnyök/hátrányok:
  - más ábrázolásmódok is kiteríthetőek Petri hálóvá
  - egyszerű feladathoz is nagy Petri háló tartozhat

Petri hálók struktúrája

Strukturálisan: irányított, súlyozott, páros gráf

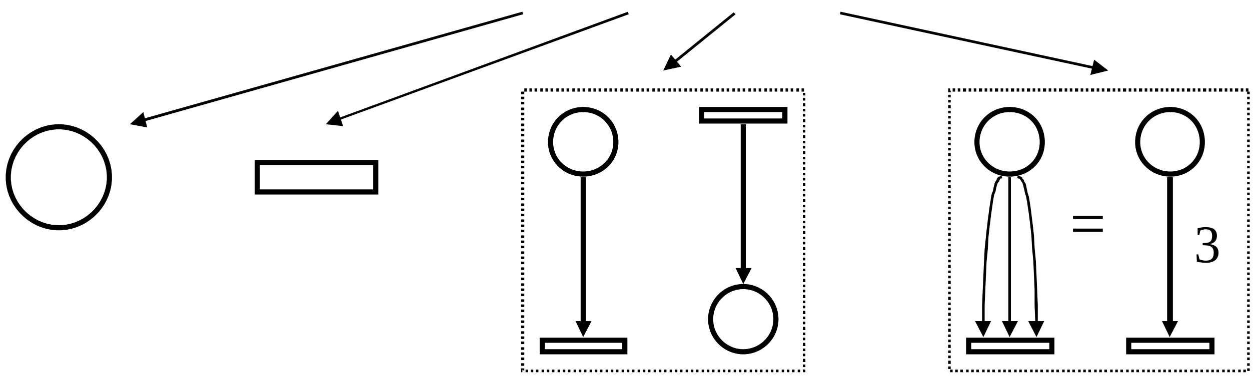
- Két típusú csomópont:
  - hely: $p \in P$ O
  - tranzíció: $t \in T$ □
- Irányított élek (páros gráf):
  - hely $\rightarrow$ tranzíció
  - tranzíció $\rightarrow$ hely
  - $e \in E : (P \times T) \cup (T \times P)$



Petri hálók felépítése

Élsúly:
Bármely $e \in E$ élhez $w^*(e) \in$ **N**+ súlyt lehet rendelni
A $w^*(e)$ súlyú e él ugyanaz, mint w_e darab párhuzamos él
Nem szokás feltüntetni az egyszeres súlyokat

$$N = < P, T, E, W >$$

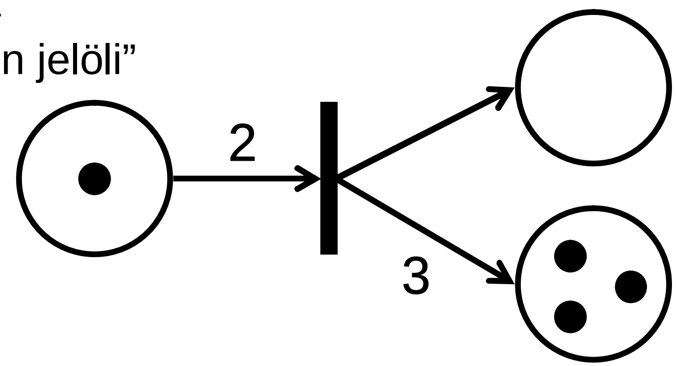


Dinamikus működés: állapotváltozók

Állapot:

- Állapotjelölő: token
- Hely állapota: benne levő tokenek száma
- Hálózat állapota: az egyes helyek állapotainak összessége
  - Állapotvektor: a $\pi = |P|$ komponensű M token eloszlás vektor
  - Az m_i komponense a p_i helyen található tokenek száma
    - „p_i –t m_i token jelöli”

$$M = \begin{bmatrix} m_1 \\ \vdots \\ m_p \end{bmatrix}$$



•Kezdőállapot: $M0$ kezdő token elosztás

Működés

Állapotváltás:

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

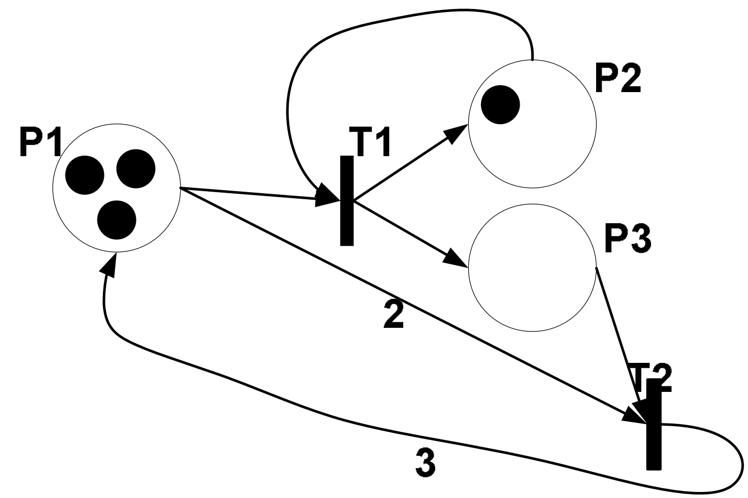
Speciális csomópontok

- a forrás- és a nyelő tranzíció
- A forrásnak nincs bemenete, és mindig képes tüzelni
- a nyelő tranzíciónak pedig nincs kimenete, így a tüzelés során a hozzá érkező tokeneket „elnyeli”.

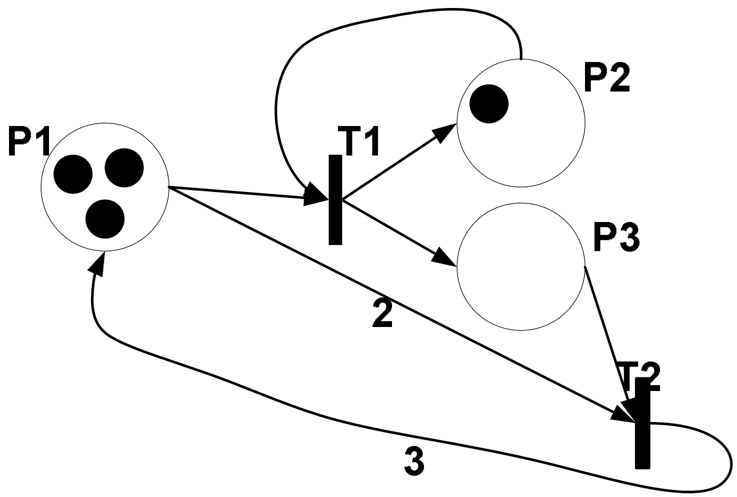
Matematikai formalizmussal:

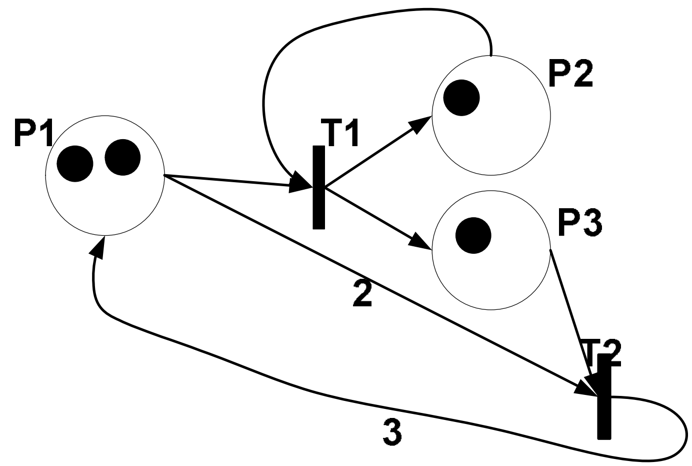
- Egy adott tokeneloszlást az az M vektor jelöl, amelynek dimenziója a P helyek száma ($\pi=|P|$), elemei pedig az adott sorszámú helyen az adott pillanatban található tokenek száma.
- Az előzőekben tekintett kezdeti tokeneloszlás leírása tehát ily módon

(π =3):



$$M_0 = \begin{bmatrix} 3 \\ 1 \\ 0 \end{bmatrix}$$

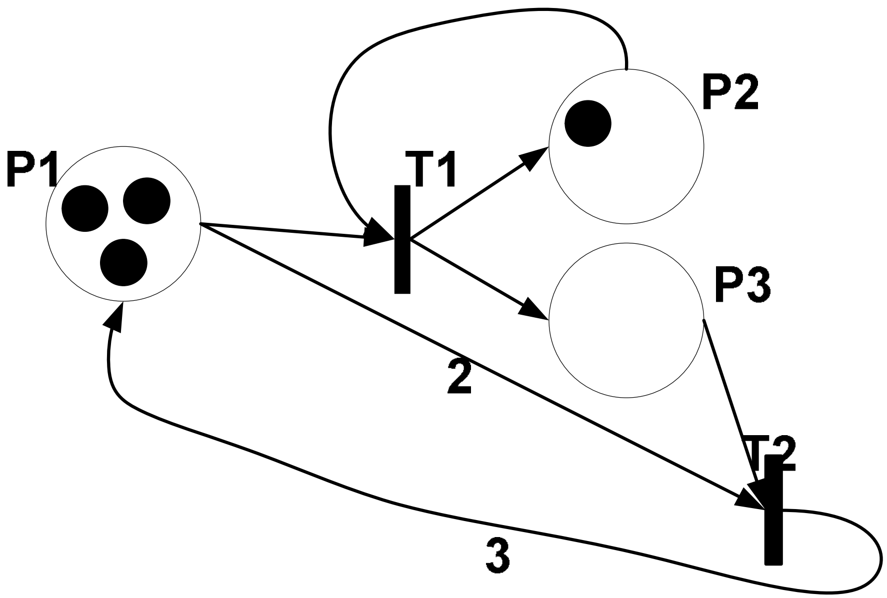


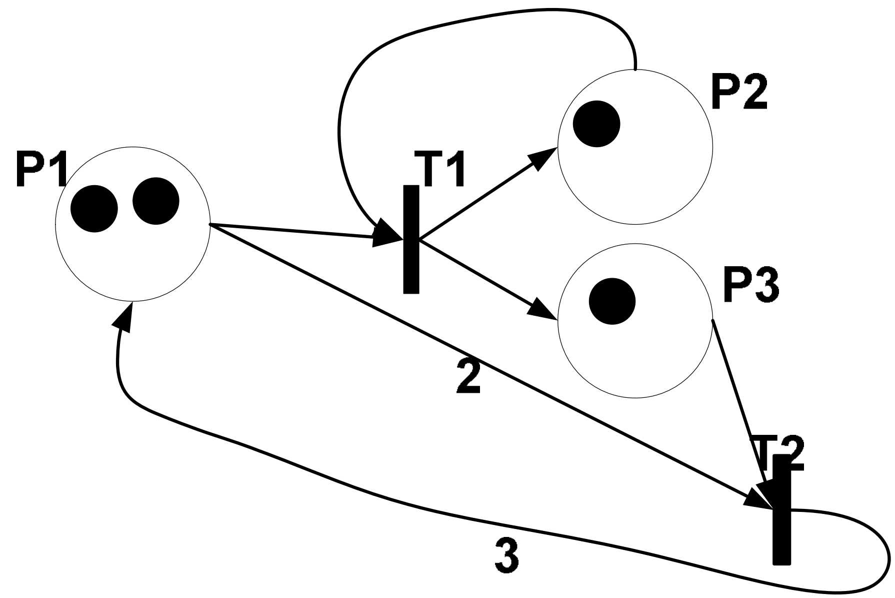


$$M_0 = \begin{bmatrix} 3 \\ 1 \\ 0 \end{bmatrix}$$

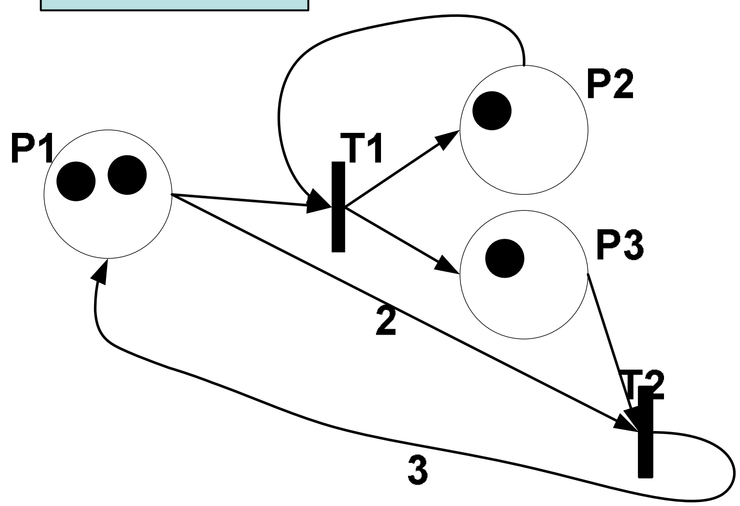
$$\mathrm{M}_{\mathrm{T1}} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$$

Hogyan néz ki a fenti háló a T1 tranzíció tüzelése után?

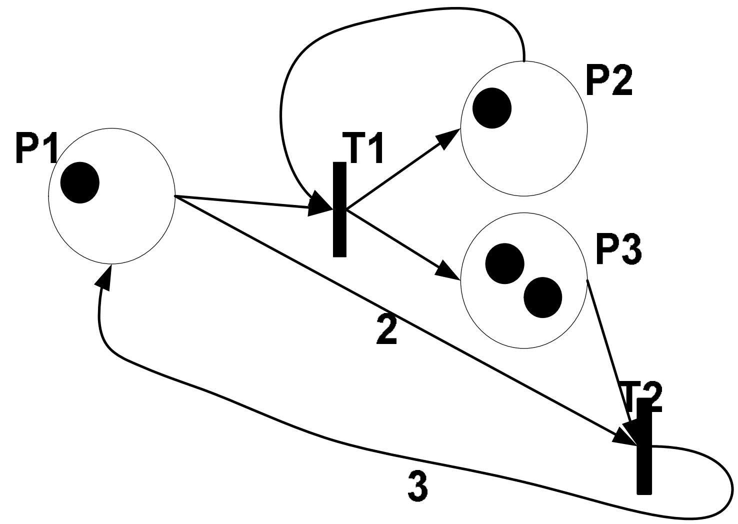




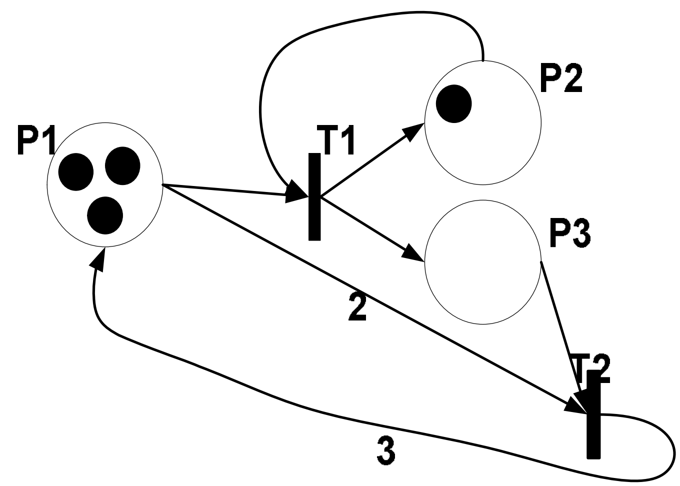
Jól látható, hogy most már mind a T1, mind a T2 állapot tüzelhető.



A T1 tüzelése után



T2 tüzelése után



Összefoglaló

Petri háló:

- Nemdeterminisztikus véges automata
- Állapotvektor: token eloszlás vektor
- Állapotátmeneti függvény: tranzíciók

Felépítés:

- egy-egy hely egy-egy logikai feltétel
- Petri háló struktúrája követi a feladat logikai dekompozícióját

Szintakszis összefoglalása

Petri-háló	$PN = (P, T, E, W, M_0)$
Helyek	$P = \{p_1, p_2, \cdots p_\pi\}$
Tranzíciók	$T = \{t_1, t_2, \cdots t_\tau\}$
	$P \cap T = \emptyset$
Élek	$E \subseteq (P \times T) \cup (T \times P)$
Súlyfüggvény	$w^* : E \longrightarrow \mathbb{N}^+$
Kezdőállapot	$M_0 : P \longrightarrow \mathbb{N}$
PN struktúra	$N = (P, T, E, W)$
PN adott kezdőállapottal	(N, M_0)

Topológia

$n \in (P \cup T)$ csomópont $^{\bullet}n$ ősei és $n^{\bullet}$ utódai:

- $t \in T$ ősei a bemeneti helyei: $^{\bullet}t = \{p \mid (p,t) \in E\}$
- $t \in T$ utódai a kimeneti helyei: $t^{\bullet} = \{p \mid (t,p) \in E\}$
- $p \in P$ ősei a bemeneti tranzíciói: $^{\bullet}p = \{t \mid (t,p) \in E\}$
- $p \in P$ utódai a kimeneti tranzíciói: $p^{\bullet} = \{t \mid (p,t) \in E\}$

Topológia

Csomópontok $P' \subseteq P$ ill. tranzíciók $T' \subseteq T$ részhalmazára:

$$\bullet P' = \bigcup_{p \in P'} \bullet p$$

$$P' \bullet = \bigcup_{p \in P'} p \bullet$$

$$\bullet T' = \bigcup_{t \in T'} \bullet t$$

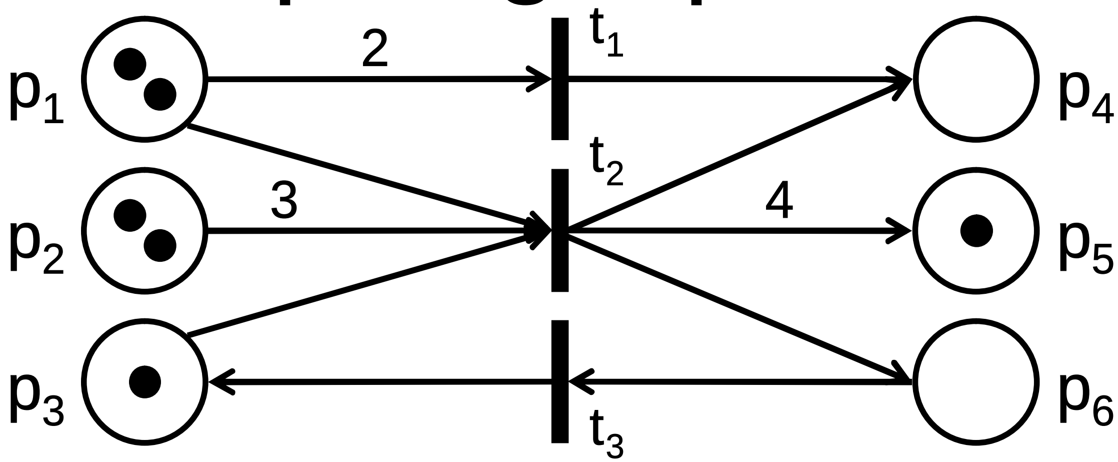
$$T' \bullet = \bigcup_{t \in T'} t \bullet$$

Speciális csomópontok

Forrás ill. nyelő csomópontok

- $t \in T$ forrás (nyelő) tranzíció:
  - Bemenő (kimenő) hely nélküli ($\bullet t = \emptyset$ illetve $t \bullet = \emptyset$)
  - Forrás tranzíció minden esetben tud tüzelni
- PN **tiszta**, ha nincsenek önhurkai, azaz
  - $\forall t \in T: \bullet t \cap t \bullet = \emptyset$

Topológia példa



$\mathbf{I}\, p_1 = \emptyset$

$\mathbf{I}\, p_2 = \emptyset$

$\mathbf{I}\, p_3 = \{t_3\}$

$\mathbf{I}\, p_4 = \{t_1, t_2\}$

$\mathbf{I}\, p_5 = \{t_2\}$

$\mathbf{I}\, p_6 = \{t_2\}$

$p_1\, \mathbf{I} = \{t_1, t_2\}$

$p_2\, \mathbf{I} = \{t_2\}$

$p_3\, \mathbf{I} = \{t_2\}$

$p_4\, \mathbf{I} = \emptyset$

$p_5\, \mathbf{I} = \emptyset$

$p_6\, \mathbf{I} = \{t_3\}$

$\mathbf{I}\, t_1 = \{p_1\}$

$\mathbf{I}\, t_2 = \{p_1, p_2, p_3\}$

$\mathbf{I}\, t_3 = \{p_6\}$

$t_1\, \mathbf{I} = \{p_4\}$

$t_2\, \mathbf{I} = \{p_4, p_5, p_6\}$

$t_3\, \mathbf{I} = \{p_3\}$

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

Tüzelés feltétele

- Ha egy $t \in T$ tranzíció minden bemeneti helyét legalább $w^-(p, t)$ token jelöli:
  - $w^-(p, t)$ a p -ből t -be vezető $e = (p, t)$ él $w^*(e)$ súlya

  $\Rightarrow$ a tranzíció tüzelése **engedélyezett**, ha

$$\boxed{\forall p \in \bullet t : m_p \geq w^-(p,t)}$$

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
- Þ **Nemdeterminisztikus működés**

A tranzíció tüzelése

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

Nemdeterminizmus és időzítés

Tetszés szerinti tüzelés jelentése:

- implicit időfogalom
- nincs időskála
- a tüzelés a $[0, \infty)$ időintervallumban valahol megtörténhet

A tüzelésekhez tetszőleges konkrét időértéket rendelve:

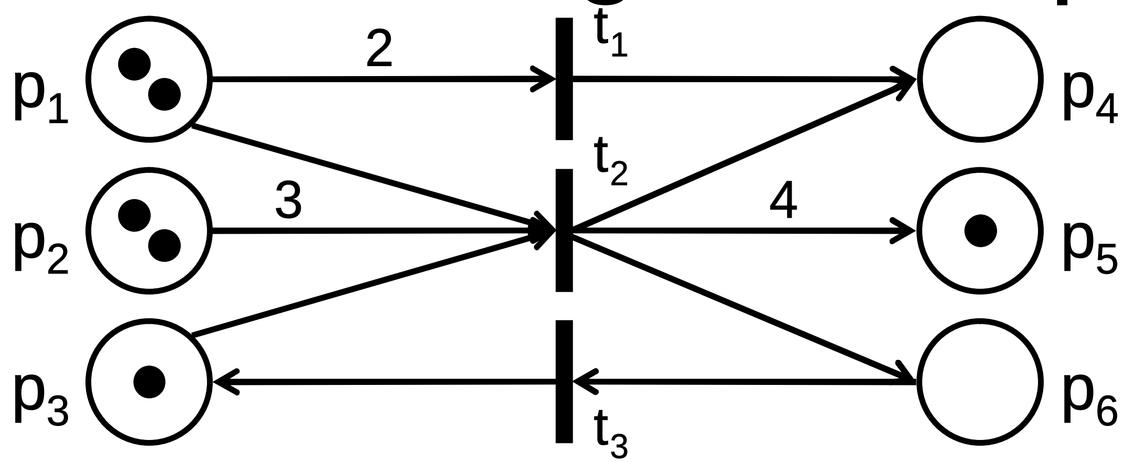
- az azonos struktúrájú és kezdőállapotú nem-determinisztikus időzítetlen Petri háló annak minden lehetséges tüzelési szekvenciáját lefedi.

Szomszédossági mátrix

- Súlyozott szomszédossági mátrix: $\mathbf{W} = [\mathrm{w}(t, p)]$
- Dimenziója: $\tau \times \pi = |T| \times |P|$
- Ha t tüzel, mennyit változik a p-beli tokenszám:

$$\mathrm{w}(t, p) = \begin{cases} 0 & \text{ha } (t, p) \notin E \text{ és } (p, t) \notin E \\ \mathrm{w}^{+}(t, p) - \mathrm{w}^{-}(p, t) & \text{ha } (t, p) \in E \text{ vagy } (p, t) \in E \end{cases}$$

Szomszédossági mátrix példa



$$W^{-}=\begin{bmatrix}2&0&0&0&0&0\\1&3&1&0&0&0\\0&0&0&0&0&1\end{bmatrix}$$

$$W=\begin{bmatrix}-2&0&0&1&0&0\\-1&-3&-1&1&4&1\\0&0&1&0&0&-1\end{bmatrix}$$

$$W^{+}=\begin{bmatrix}0&0&0&1&0&0\\0&0&0&1&4&1\\0&0&1&0&0&0\end{bmatrix}$$

Petri hálók kiterjesztések

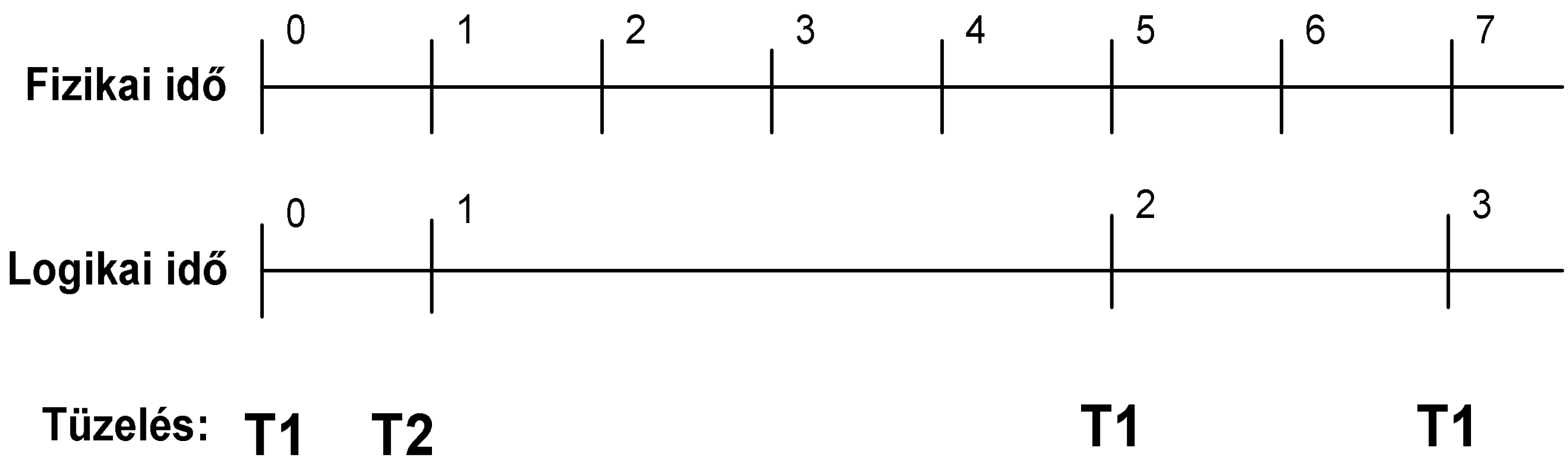
Okok

- Időkorlátok,
- Átláthatóság,
- Végtelen körök,
- Nem-determinisztikus működés…

idő

- **A logikai idő**
- Fizikai idő (óránkkal, a napszakok változásával stb. mérhetünk, tehát ami a rendszertől független), objektív időskálán szemléltethető (többnyire szabályosan periodikus).
- A logikai idő: a rendszer működésétől függ, viszonyítási pontjai a bekövetkezett események, Petri-hálók esetében *a tüzelések*.

Szemléletesen a [T1, T2, T1, T1] tüzelési szekvencia a fizikai és a logikai időben :

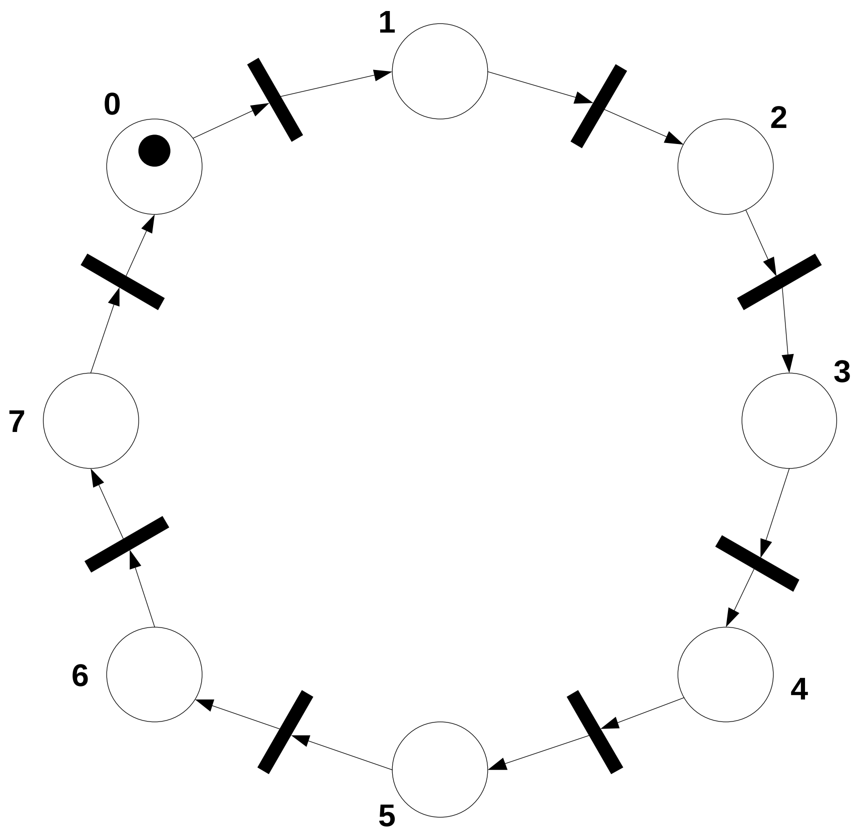


implicit időfogalmat kacsolunk rendszerünkhöz, amelyben egy engedélyezett tranzíció tüzelése a [0, ∞) intervallumban bárhol megtörténhet (fire-at-will).

A nemdeterminisztikus működés kiküszöbölése

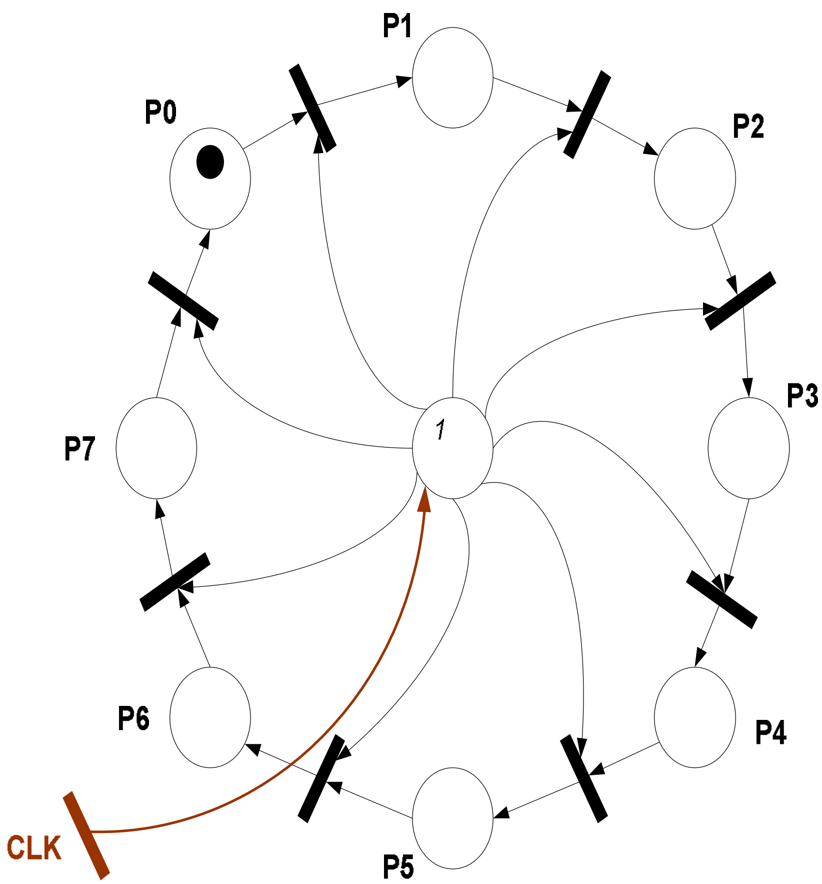
- A nemdeterminisztikus működés nem mindig engedhető meg
- kiküszöbölésére különböző korlátozásokat vezethetünk be rendszerünkben
- Szimulációs problémák

Példa: nyolcas számláló



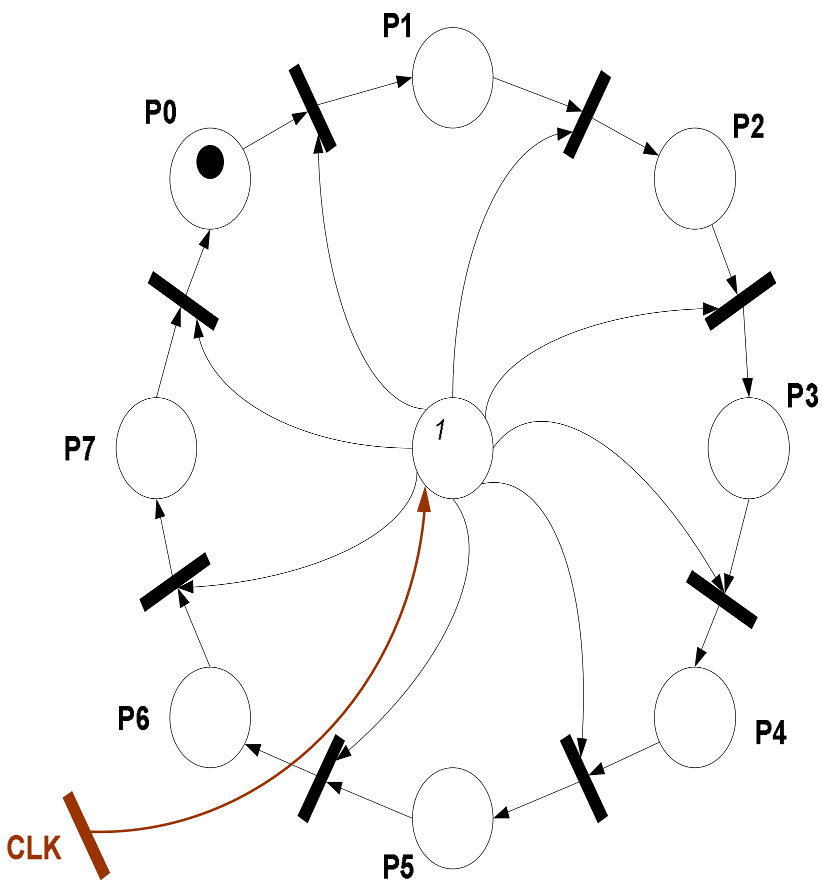
- egyetlen token kering a rendszerben, amelynek minden ugrása a számláló egy ugrását jelenti: kezdetben a token a *P0* helyen van.
- A következő tüzeléskor átkerül az *P1*-es állapotba, majd a *P2*-be stb. A kör végén a *P7*-es helyrők ismét a *P0*-ba ugrunk, és kezdődik az egész számlálás elölről.
- DE: a tüzelésekhez kötött logikai idő semmit nem mond arról, hogy fizikailag mikor következik be a következő ugrás, mi pedig szeretjük a határidőket, szeretjük tudni, mi mikor következik be.

Vezessünk be órajelet!



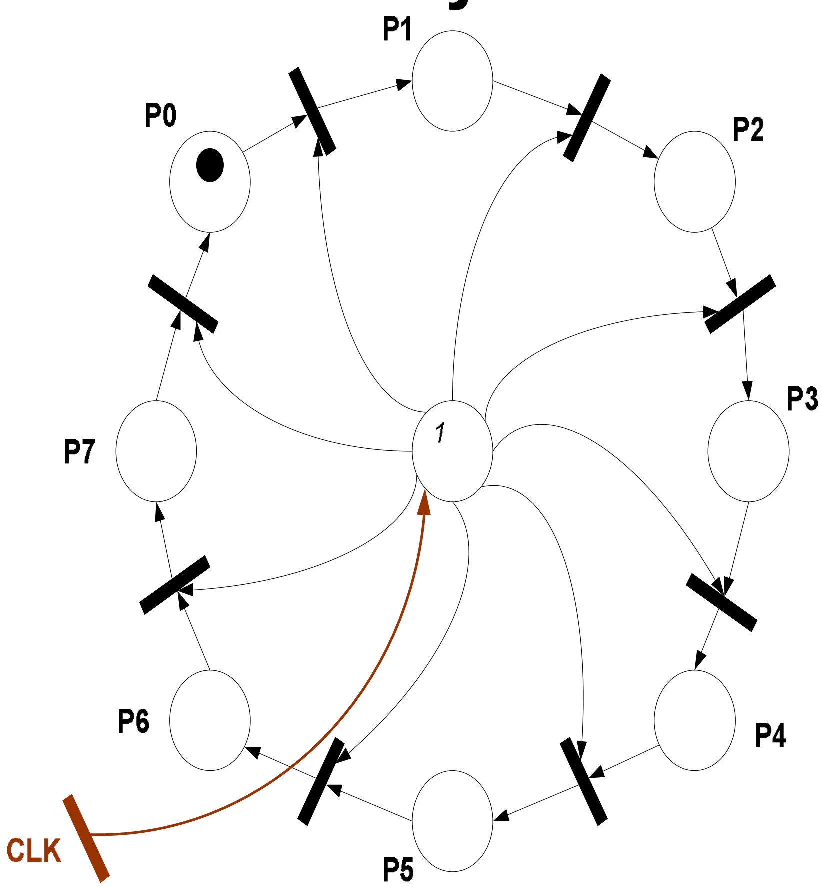
- Első megközelítésben ez egyetlen forrás-tranzíciót jelent, amely tokeneket juttat a rendszerbe az alábbi módon:
- A CLK bizonyos időközönként tokent juttat a rendszerbe. Az újabb élek felvételével a tranzíciók már csak akkor tüzelhetnek, ha a középső állapotban van token, azaz „ütött az óra”.

Vezessünk be órajelet!



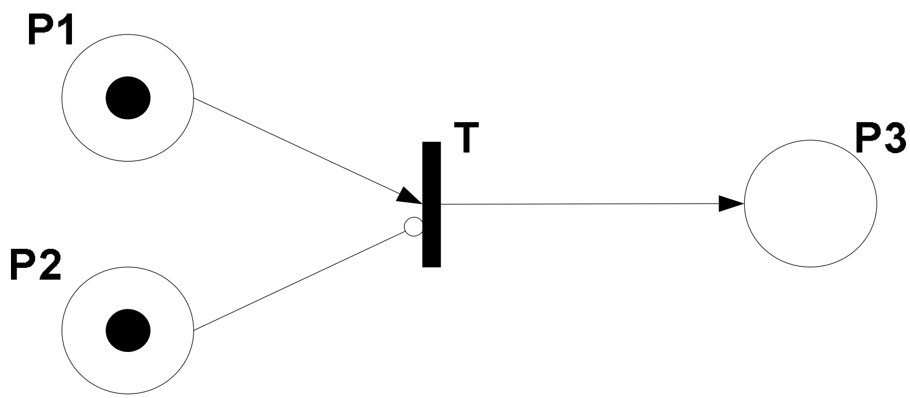
- Első megközelítésben ez egyetlen forrás-tranzíciót jelent, amely tokeneket juttat a rendszerbe az alábbi módon:
- A CLK bizonyos időközönként tokent juttat a rendszerbe. Az újabb élek felvételével a tranzíciók már csak akkor tüzelhetnek, ha a középső állapotban van token, azaz „ütött az óra”.

Vezessünk be órajelet!



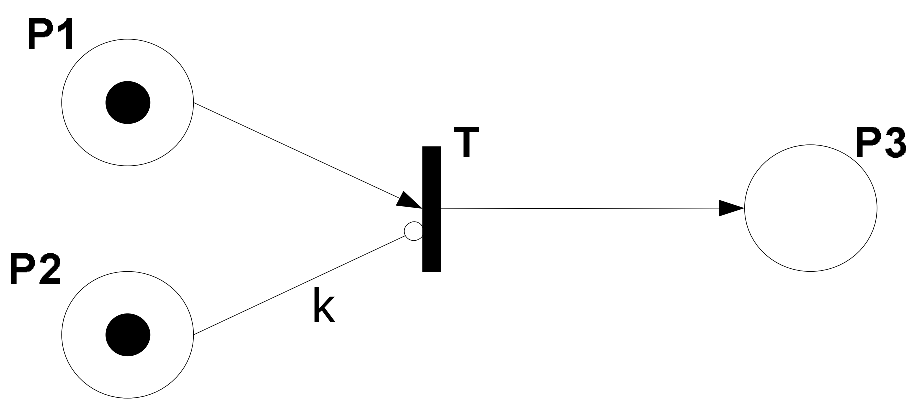
- ha van is órajel-tokenünk, akkor sem garantált, hogy a soron következő tranzíció tüzel a következő óraütés előtt.
- Sajnos a Petri-hálók sajátosságai miatt ezt nem tudjuk garantálni, de azt igen, hogy az óra ne üthessen addig, amíg az előző órajelre nem történt ugrás a számlálóban.
- Korlátozzuk tehát a középső 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 élet!



- A kapacitáskorlát helyett **tiltó éleket** veszünk fel a hálóba.
- A tiltó él azt jelöli, hogy a tranzíció *ne* tüzeljen, amíg az adott feltétel teljesül. Az alábbi ábrán például addig nem tüzel 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

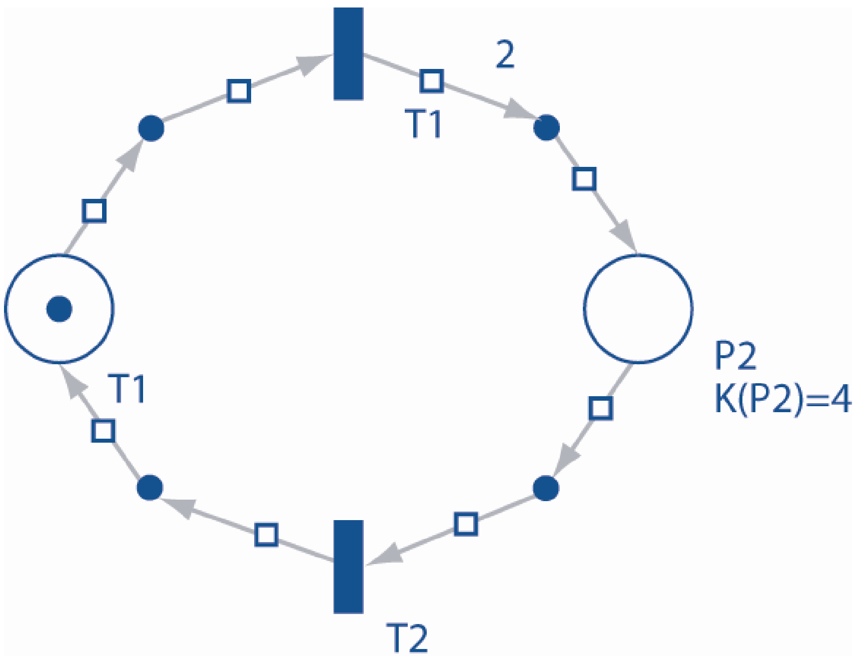
További lehetőségek

- **A kapacitáskorlát illetve kiküszöbölése,**
- Tiltó élek bevezetése
- Prioritás rendelése a tranzíciókhoz

Prioritás

- 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!

A kapacitáskorlát kiküszöbölése



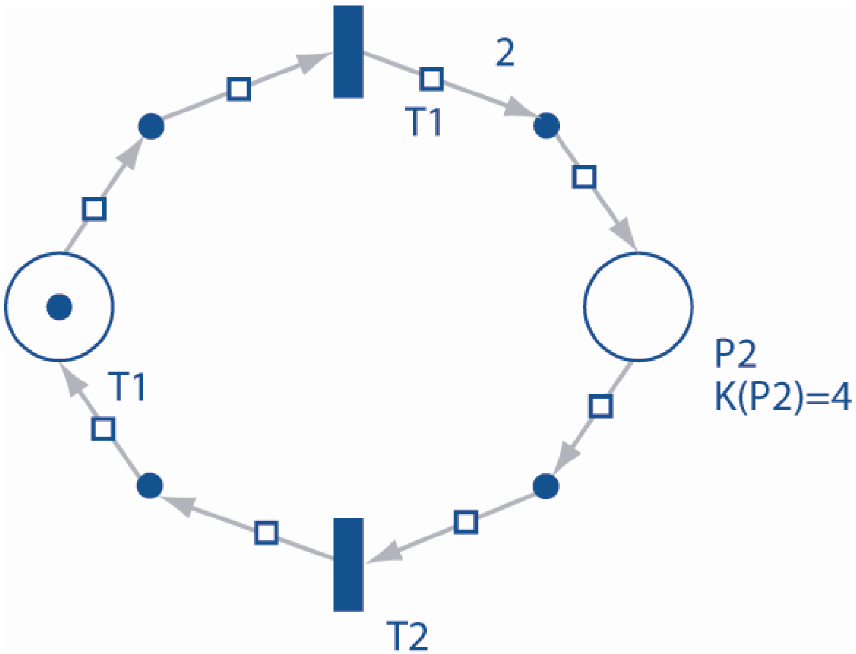
példa.

- kapacitáskorlátos Petri-háló

működése könnyen leírható:

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.

A kapacitáskorlát kiküszöbölése

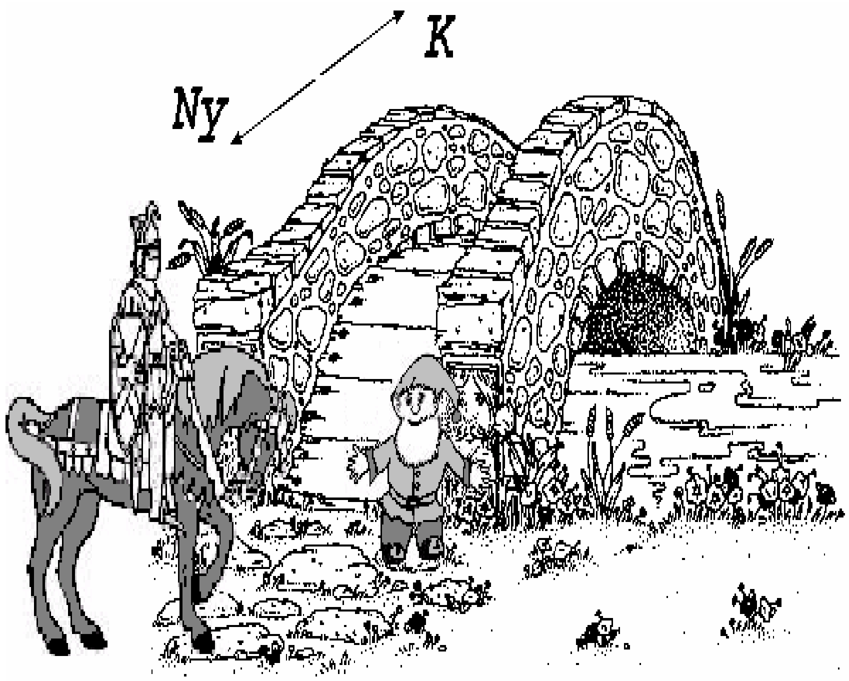


T1 minden egyes tüzelése eggyel megnöveli a tokenek számát, vagyis nincs felső korlát arra, hogy a rendszerben mennyi token lesz, ha tetszőlegesen hosszú ideig magára hagyjuk futás közben.

Ezáltal arra sem tudunk korlátot adni, hogy egy-egy állapotban hány tokenünk lesz. Ezt hivatott megoldani a kapacitáskorlát,

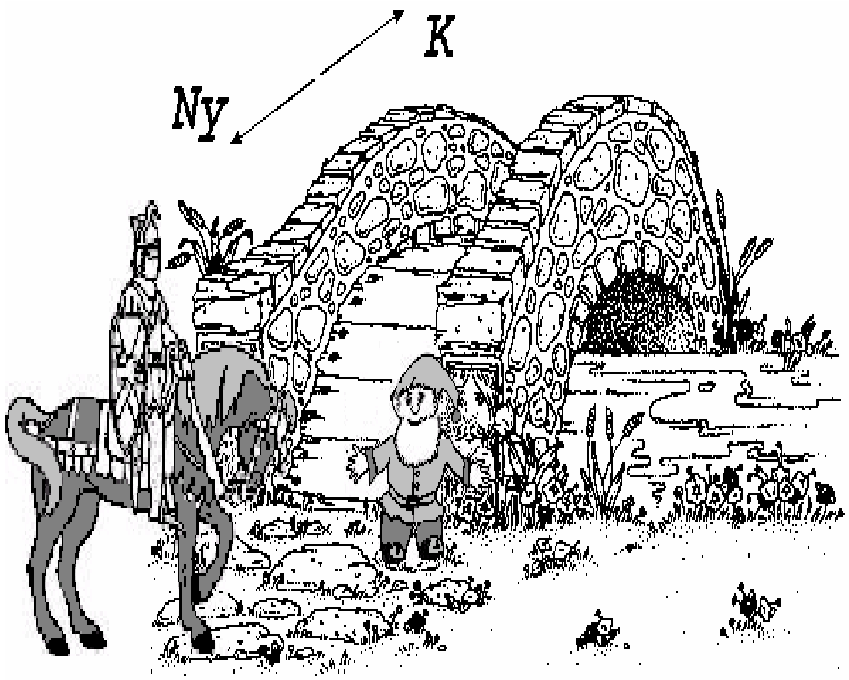
Ha bevezetünk egy K(*P2*)=4 kapacitáskorlátot, az azt jelenti, hogy a *P2* helyen maximum 4 token lehet egyszerre, s ez a *T1* tranzíció tüzeléséhez is egy újabb, korlátozó feltételt jelent (ha *P2*-n már 4 token tartózkodik, akkor a *T1* tüzelése nem engedélyezett, hiába van token *P1*-en).

funkcionalitásukban kapacitáskorlátos helyek

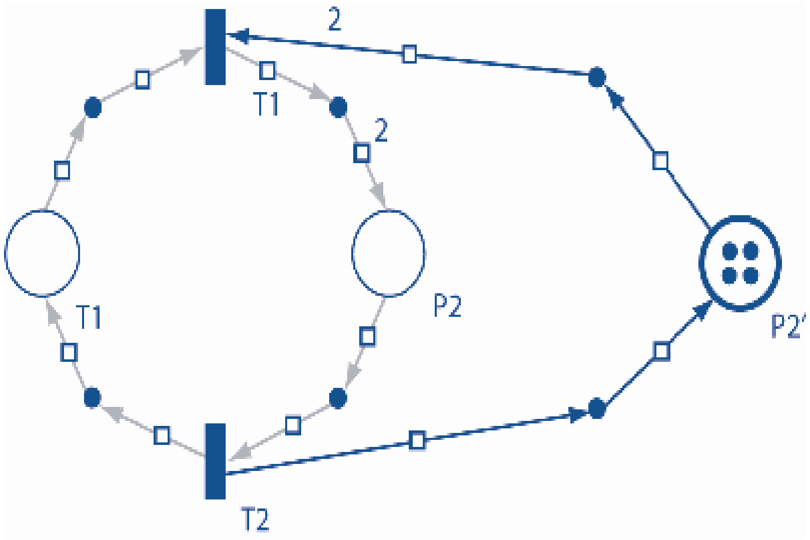


- konkrét kapacitáskorlát nélkül.
- Mi történik a *P2* hellyel, amikor elveszünk onnan, illetve teszünk oda tokent?
- Tegyük fel, hogy *P2* egy **kapacitáskorlát nélküli** (végtelen korlátos) hely, és kezdetben nincs ott token (ez a helyzet a fenti példában is). Képzeljük el, hogy a *P2* helyen (a híd lábánál) egy törpe ücsörög, akinek üres az erszénye. Ha arra jár egy lovag (és mondjuk keleti irányba tart), egy garas vámot szed tőle, ha viszont ugyanez a lovag visszafelé jön, visszakapja a garasát. Így a törpénél átmenetileg lehetnek a garasok, és legfeljebb annyi, ahány lovag kelet felé elhaladt.

Kapacitáskorlátos helyek funkcionalitásukban



- Ha a törpe zsebébe *k* darab garas fér el, egyszerre *k* lovag lehet a hídtól keletre (tegyük fel, hogy a világ azon felén eredetileg nem éltek lovagok). Ebben az esetben csak akkor mehet át a hídon a következő lovag, amikor egy másik visszatér. (*Formálisan*: a *Lovag_átlép* tranzíció csak akkor engedélyezett, ha a Kelet állapotban *k*-nál kevesebb token van.)
- ha mesebeli törpénk zsebe feneketlen, elvileg akárhány lovag lehet a keleti, és a nyugati oldalon egyaránt. Hogyan oldható meg ekkor, hogy a sok lovag ne árassza el a keleti oldalt?
- Ennek egyetlen módja van: kevés embert kell lovaggá ütni. Ha az Óperenciás tengeren túli király gondoskodik arról, hogy országában összesen *k* lovag legyen (és semmiképpen sem több), akkor egészen biztos, hogy a keleti oldalon sohasem lesz *k*-nál több lovag. Ebben az esetben, ha keleten *x* számú lovag van, akkor nyugaton egészen pontosan *k-x*.



- Ha azt akarjuk elérni, hogy a *P2* (kelet) helyen maximum *k=4* darab token legyen, be kell vezetnünk egy **adminisztrációs helyet** (*P2'*), ahol azt tartjuk számon, hogy a *P2* helyre hány token fér még el (nyugat). Így az adott helyen lévő tokenek számát *m*-mel jelölve mindig igaz lesz az alábbi összefüggés: m(*P2'*)+m(*P2*)=*k*.
- Hogyan felügyelhető, hogy ez minden esetben így legyen?
- Kezdetben a *P2'* helyen legyen *k* darab token, *P2*-n pedig 0. A *P2*-vel szomszédos tranzakciókhoz vegyünk fel új éleket az alábbiaknak megfelelően: ha a tranzíció *a* darab tokent vesz el a *P2* helyről, akkor az új élen adjon *a* darabot a *P2'*-höz. Ha pedig *b* tokent ad *P2*-höz, akkor ugyanennyit vegyen el *P2'*-ből.
- A módosított *T1* tranzíció tehát nemcsak hozzáad *P2*-höz 2 db. tokent, hanem ugyanennyit el is vesz a *P2'* adminisztrációs helyről. Hasonlóan *T2* elvesz 1 tokent *P2*-ből, és hozzáad egyet *P2'*-höz.

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 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