Gyakorló feladatok:
Szoftver-modellellenőrzés absztrakcióval.
Modellezés Petri-hálókkal.
Petri-hálók tulajdonságai.
Színezett Petri-hálók.

Majzik István
BME Méréstechnika és Információs Rendszerek Tanszék

Absztrakt állapottér szoftver-modellellenőrzéshez

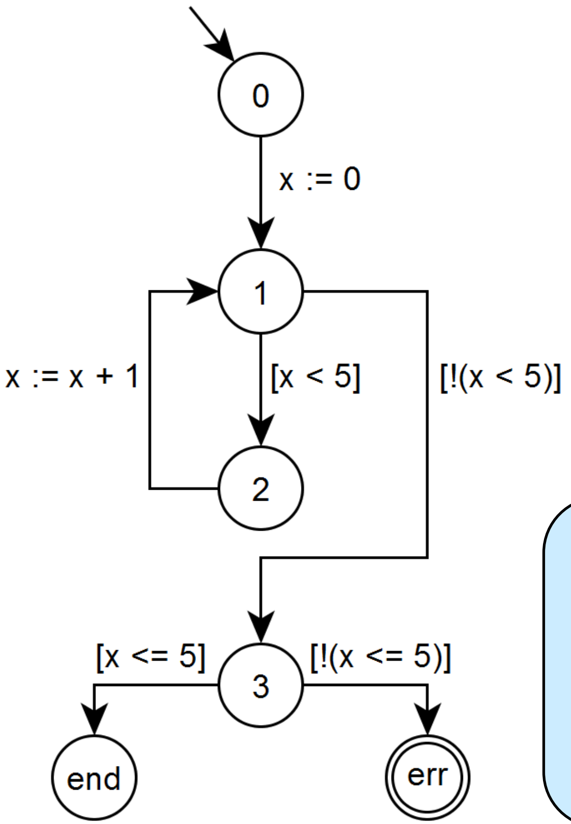
```
    x : int
0:  x = 0
1:  while (x < 5) {
2:      x = x + 1
    }
3:  assert (x <= 5)
```

- Rajzolja le a jobb oldali programrészlethez tartozó Control Flow Automaton (CFA) modellt!
  - A vezérlési helyeket a programsorokhoz írt sorszámokkal (0, 1, 2, …) azonosítsa.
  - Az assert megsértése esetére vegyen fel egy err címkéjű vezérlési helyet, a jó végállapothoz egy end címkéjű vezérlési helyet.
- Rajzolja fel az absztrakt állapotteret, ha vezérlési hely absztrakciót alkalmazunk.
- Rajzolja fel az absztrakt állapotteret (az elérhető absztrakt állapotokat), ha CFA modellellenőrzésére vezérlési hely és predikátum absztrakciót alkalmazunk, ez utóbbihoz egyetlen (x <= 5) predikátumot használunk.

Absztrakt állapottér szoftver-modellellenőrzéshez – Megoldás

```
    x : int
0:  x = 0
1:  while (x < 5) {
2:      x = x + 1
    }
3:  assert (x <= 5)
```

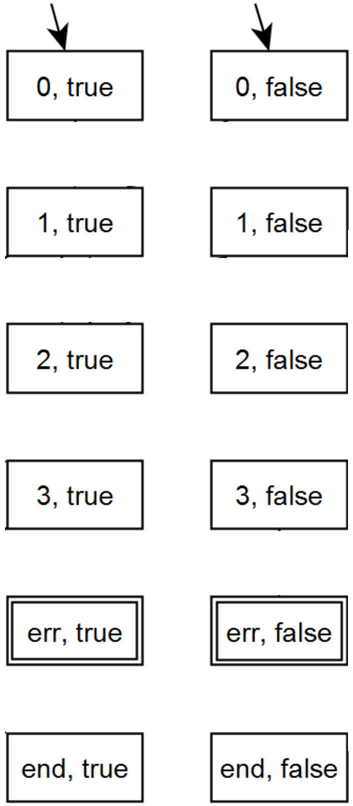
- CFA modell: Balra lent
- Vezérlési hely absztrakció: Ld. CFA struktúra
- Vezérlési hely és predikátum absztrakció:



Absztrakció (x<=5) mellett: 1. fázis

Lehetséges absztrakt állapotok: (vezérlési hely, predikátumérték) kombinációk

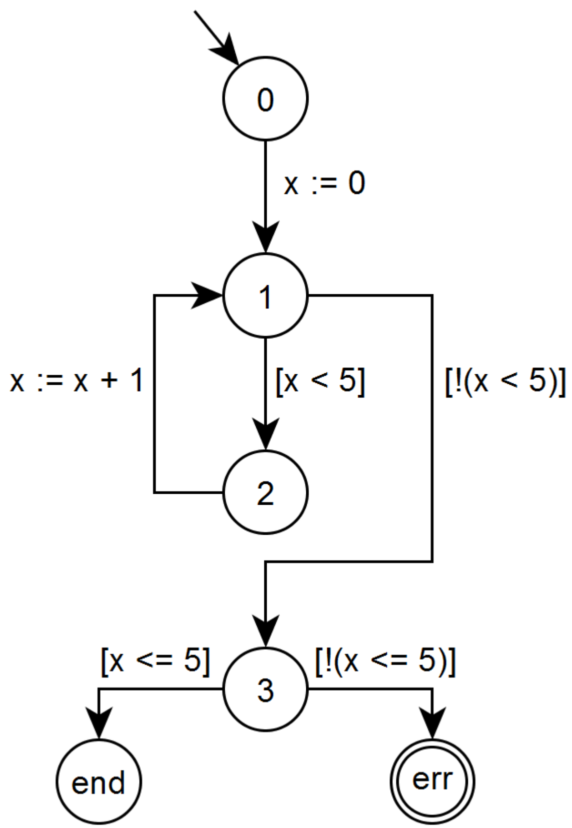
Az absztrakt állapotok közötti átmenetek létezése lokálisan, állapotpáronként ellenőrizendő, a CFA-ban lévő vezérlési helyek közötti átmenetek, utasítások és feltételek alapján



Absztrakt állapottér szoftver-modellellenőrzéshez – Megoldás

```
    x : int
0:  x = 0
1:  while (x < 5) {
2:      x = x + 1
    }
3:  assert (x <= 5)
```

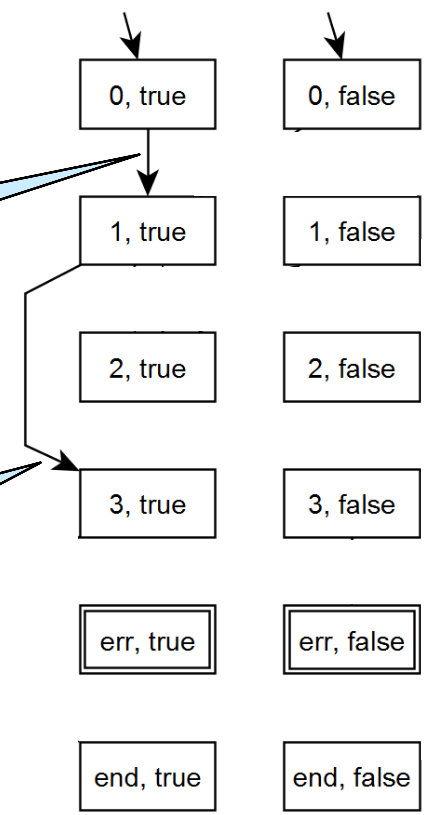
- CFA modell: Balra lent
- Vezérlési hely absztrakció: Ld. CFA struktúra
- Vezérlési hely és predikátum absztrakció:



Absztrakció (x<=5) mellett: 2. fázis

A (0, x<=5) állapotból a 0→1 átmeneten lévő x:=0 utasítás mellett az (1, x<=5) teljesülhet.

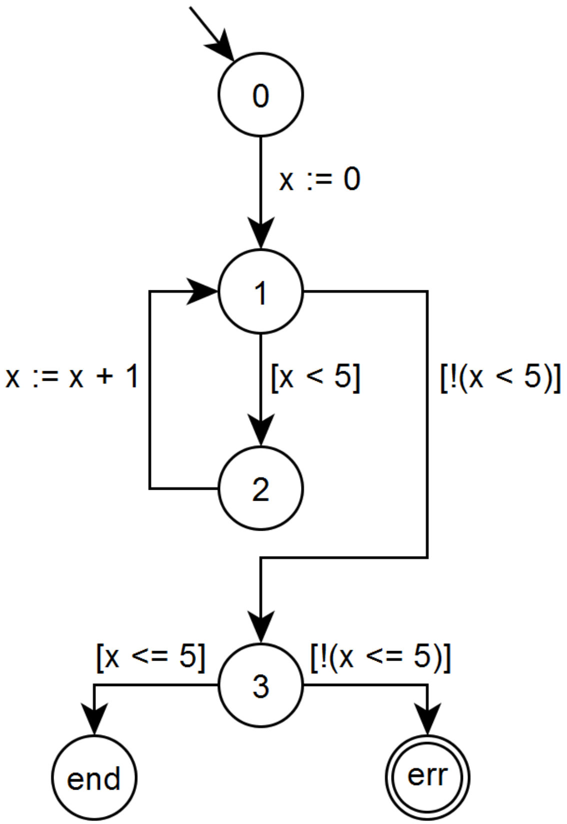
Az (1, x<=5) állapotból az 1→3 átmeneten lévő !(x<5) feltétel mellett (3, x<=5) elérhető, ha x==5.



Absztrakt állapottér szoftver-modellellenőrzéshez – Megoldás

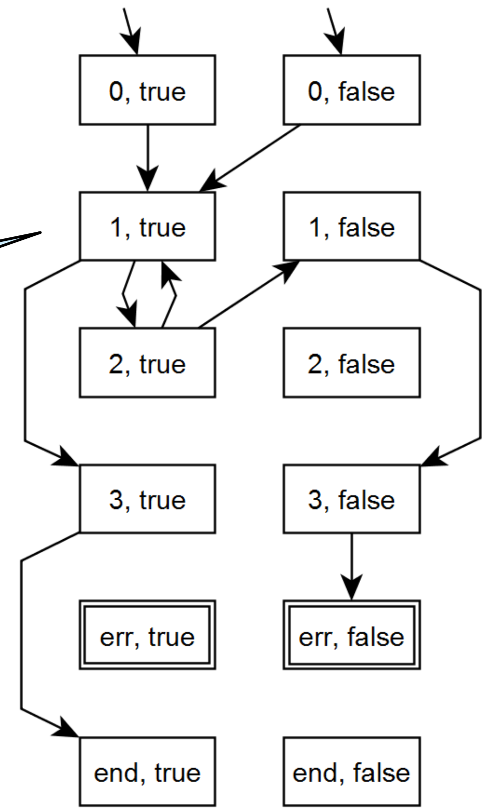
```
    x : int
0:  x = 0
1:  while (x < 5) {
2:      x = x + 1
    }
3:  assert (x <= 5)
```

- CFA modell: Balra lent
- Vezérlési hely absztrakció: Ld. CFA struktúra
- Vezérlési hely és predikátum absztrakció:



Absztrakció (x<=5) mellett: befejezés

A teljes absztrakt állapottér



Szoftver-modellellenőrzés absztrakcióval

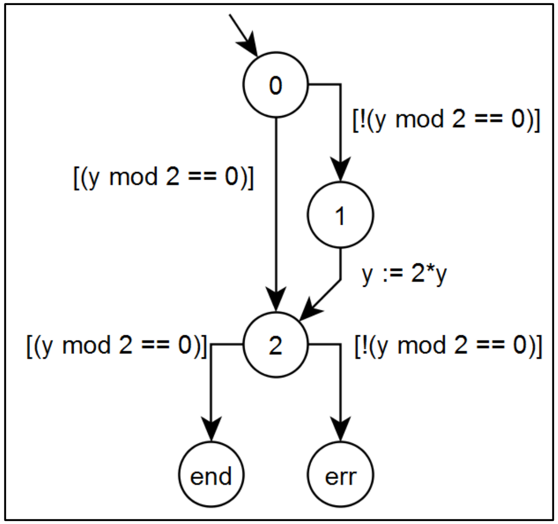
```
        y : int
0:      if !((y mod 2) ==0) {
1:              y := 2*y;
        }
2:      assert((y mod 2) == 0);
```

- Rajzolja le a programrészlethez tartozó Control Flow Automaton (CFA) modellt!
  - A vezérlési helyeket a programsorokhoz írt sorszámokkal (0, 1, 2) azonosítsa.
  - Az assert megsértése esetére vegyen fel egy err címkéjű vezérlési helyet, a jó végállapothoz egy end címkéjű vezérlési helyet.
- A CFA modellellenőrzésére vezérlési hely és predikátum absztrakciót alkalmazunk, ez utóbbihoz egyetlen (y > 0) predikátumot használunk.
  Mik lehetnek az absztrakt állapottérben a kezdőállapotok (vezérlési hely, predikátumérték) alakban megadva, ha a program indulásakor az y egész értékű változó tetszőleges lehet?
- Hamis ellenpéldának tekinthető-e az err vezérlési hely eléréséhez az absztrakt állapottérben található (0, true) → (2, true) → (err, true) útvonal?

Szoftver-modellellenőrzés absztrakcióval – Megoldás

- Rajzolja le a Control Flow Automaton (CFA) modellt!
  - Megoldás: Ld. jobbra

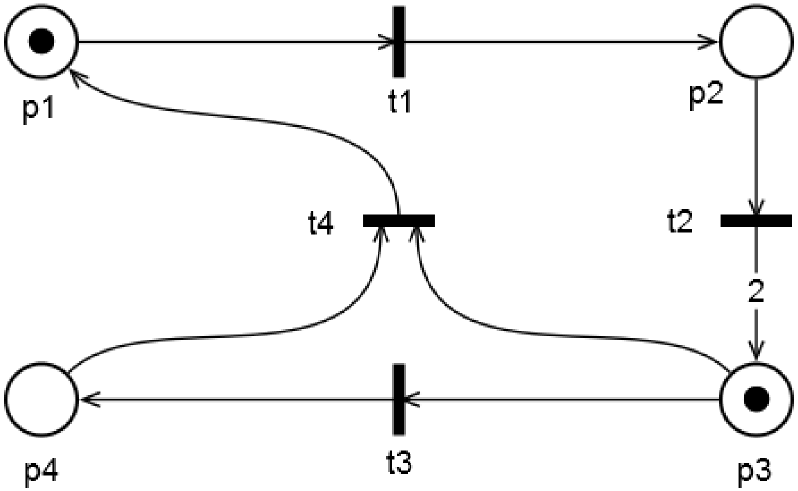
```
      y : int
0:    if !((y mod 2) ==0) {
1:          y := 2*y;
      }
2:    assert((y mod 2) == 0);
```



- Az (y > 0) predikátumot használjuk. Mik lehetnek az absztrakt állapottérben a kezdőállapotok (vezérlési hely, predikátumérték) alakban megadva, ha a program indulásakor az y egész értékű változó tetszőleges lehet?
  - Megoldás: (0, true) és (0, false)
- Hamis ellenpéldának tekinthető-e az err vezérlési hely eléréséhez az absztrakt állapottérben lévő (0, true) → (2, true) → (err, true) útvonal?
  - Megoldás: (0, true) → (2, true) átmenet esetén (y>0) és (y mod 2 == 0) szükséges, előbbi a predikátum, utóbbi az átmenet feltétele.
    A (2, true) → (err, true) átmenet esetén (y>0) és !(y mod 2 == 0) szükséges, ahol az utóbbi az itteni átmenet feltétele. Ez ellentmond az előző átmenet feltételének, tehát az ellenpélda hamis.

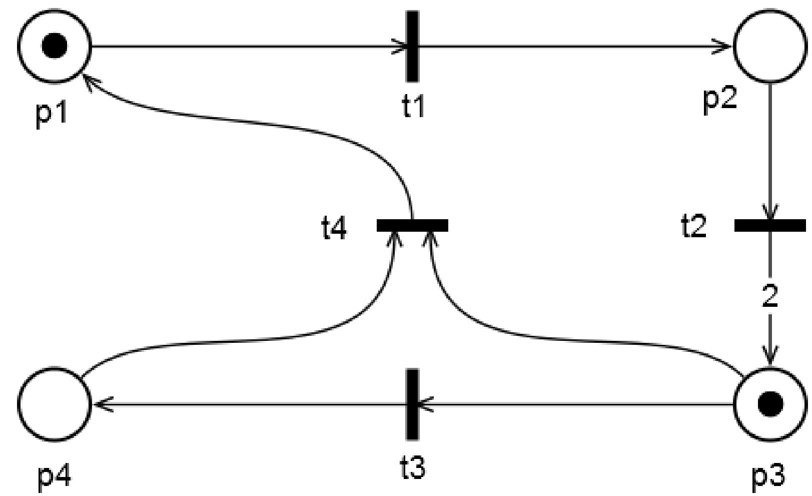
Petri-háló állapotterének felvétele

Készítse el a jobbra látható Petri-háló elérhetőségi gráfját!



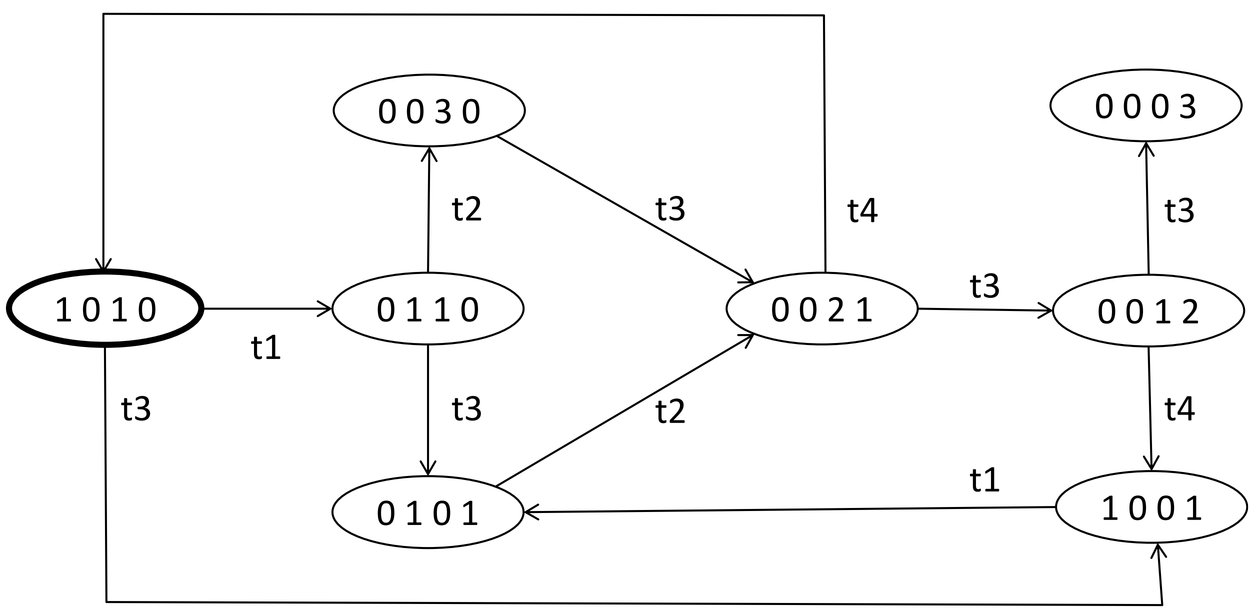
Petri-háló állapotterének felvétele – Megoldás

Készítse el a jobbra látható Petri-háló elérhetőségi gráfját!



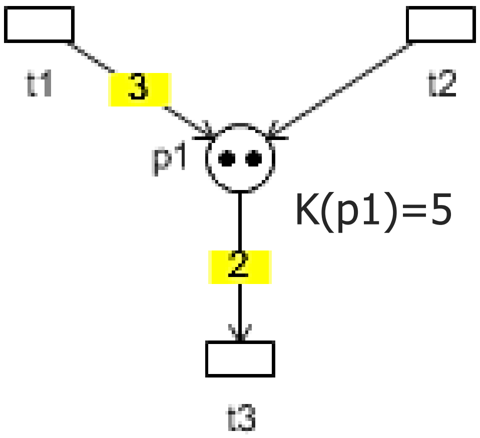
Megoldás:

Minden jelölésből a lehetséges tüzelő tranzíciók és a kialakuló új jelölések felvétele.



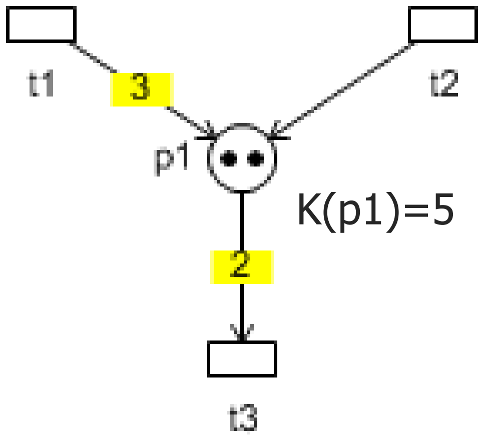
Kapacitáskorlát Petri-hálókban

- Mit jelent az, hogy egy Petri-hálóban egy hely kapacitáskorlátos?
- A mellékelt véges kapacitású háló kiegészítésével rajzoljon ekvivalens, kapacitáskorlát nélküli hálót!

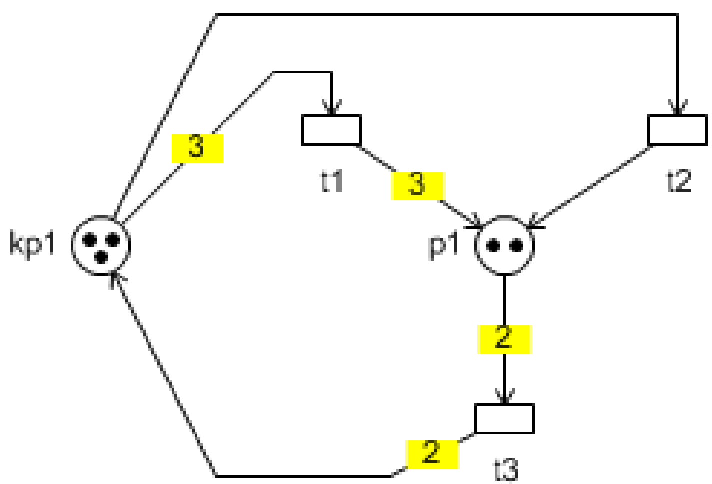


Kapacitáskorlát Petri-hálókban – Megoldás

- Mit jelent az, hogy egy Petri-hálóban egy hely kapacitáskorlátos?
  - Az adott helyen a tokenszám nem lehet nagyobb, mint a kapacitáskorlát értéke.

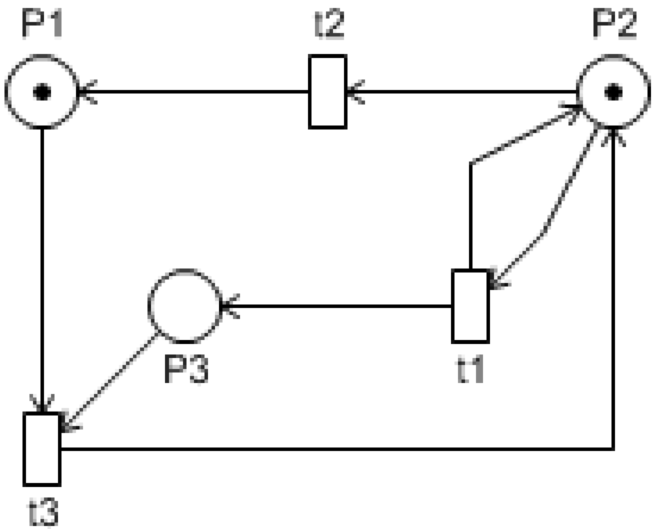


- A mellékelt véges kapacitású háló kiegészítésével rajzoljon ekvivalens, kapacitáskorlát nélküli hálót!
  - A szabad kapacitás megjelenítésére kp1 kiegészítő hely felvétele
  - Szabad kapacitás változásának követése a megfelelő élekkel



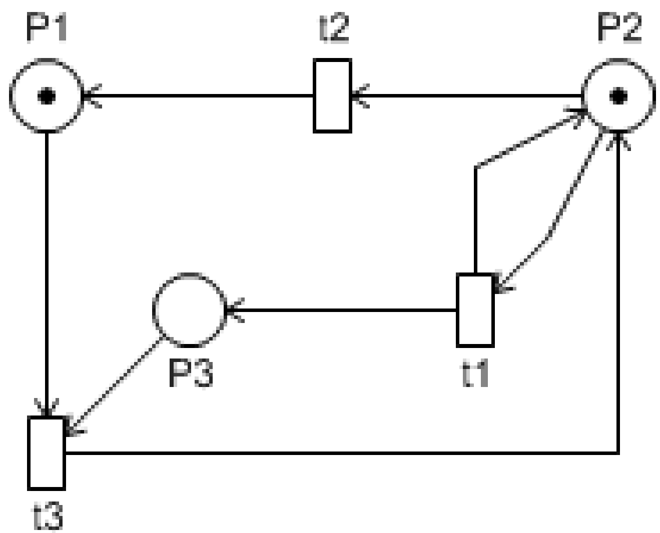
Petri-háló fedési gráfjának felvétele

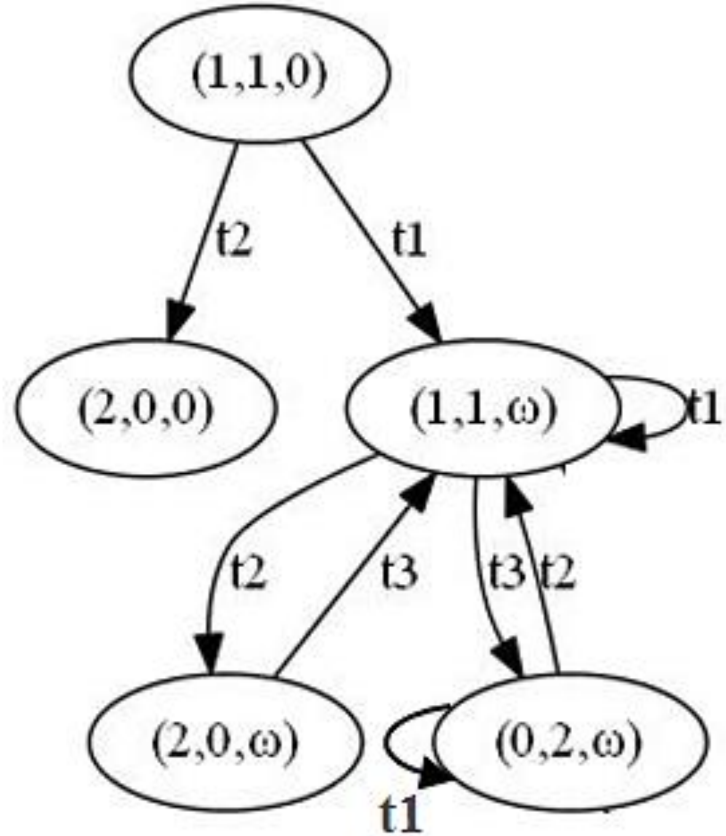
Rajzolja fel az alábbi Petri-háló fedési gráfját!



Petri-háló fedési gráfjának felvétele – Megoldás

Rajzolja fel az alábbi Petri-háló fedési gráfját!

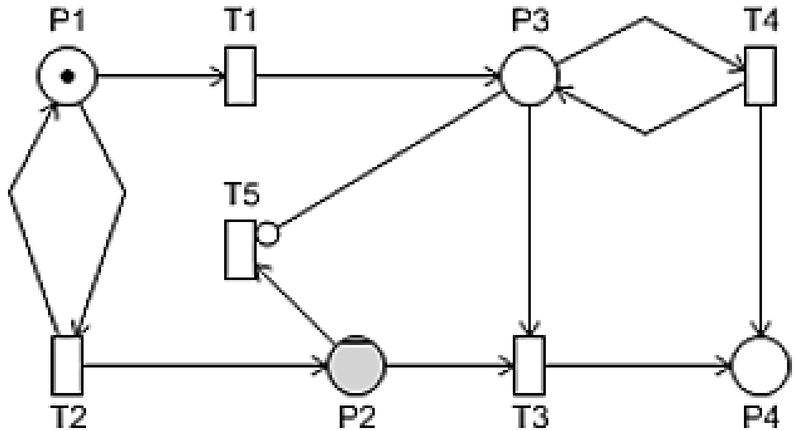




Az előzőt erősen fedő jelölések esetén, az erősen fedő helyeken szaporodhat a token;
ennek jelölésére megjelenik az ω szimbólum.

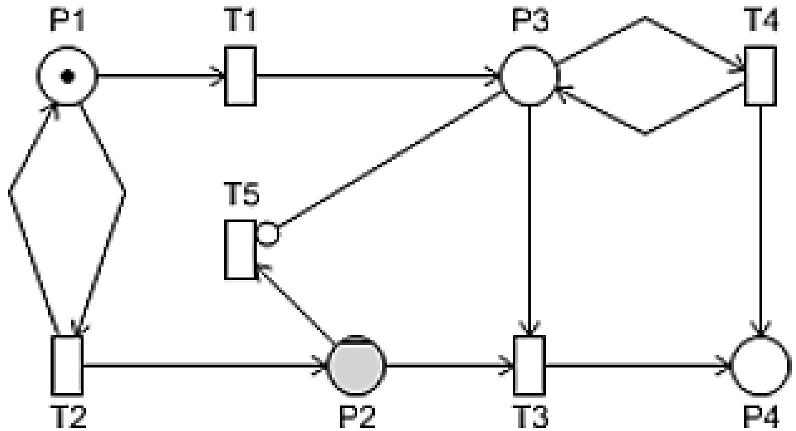
Petri-háló fedési gráfjának felvétele

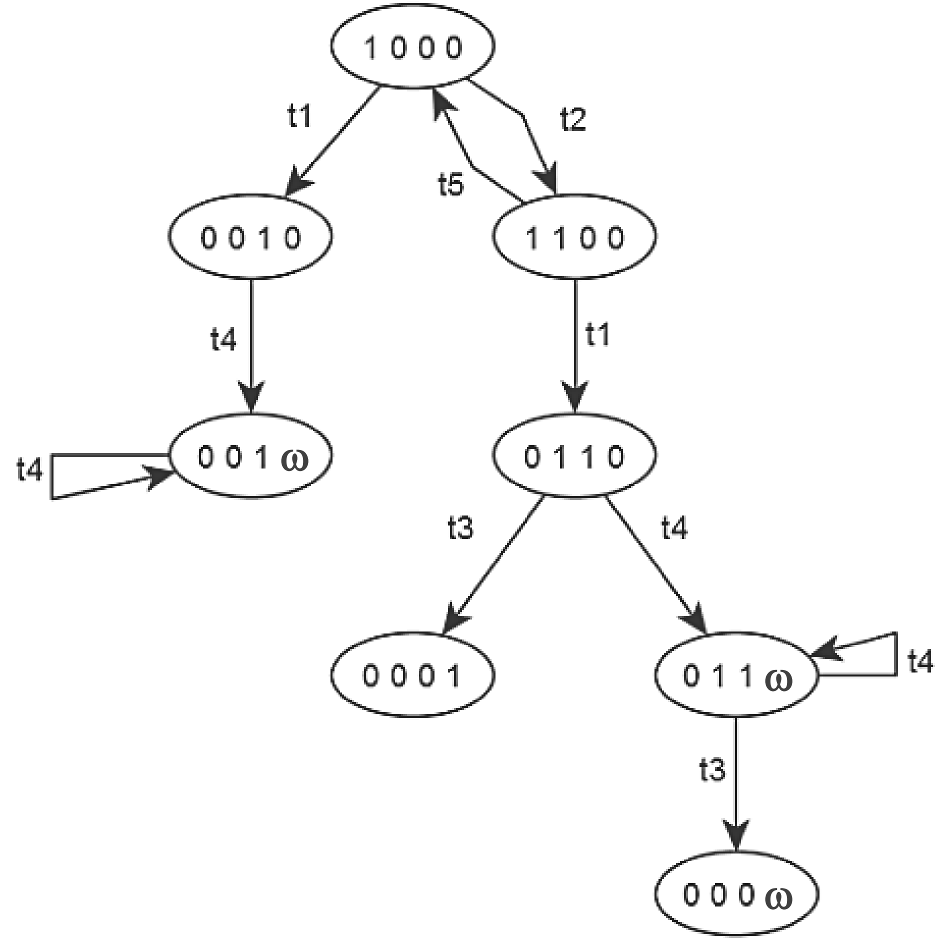
Rajzolja fel az alábbi Petri-háló fedési gráfját!
A Petri-háló P2 helye kapacitáskorlátos, K(P2)=1.



Petri-háló fedési gráfjának felvétele – Megoldás

Rajzolja fel az alábbi Petri-háló fedési gráfját!
A Petri-háló P2 helye kapacitáskorlátos, K(P2)=1.



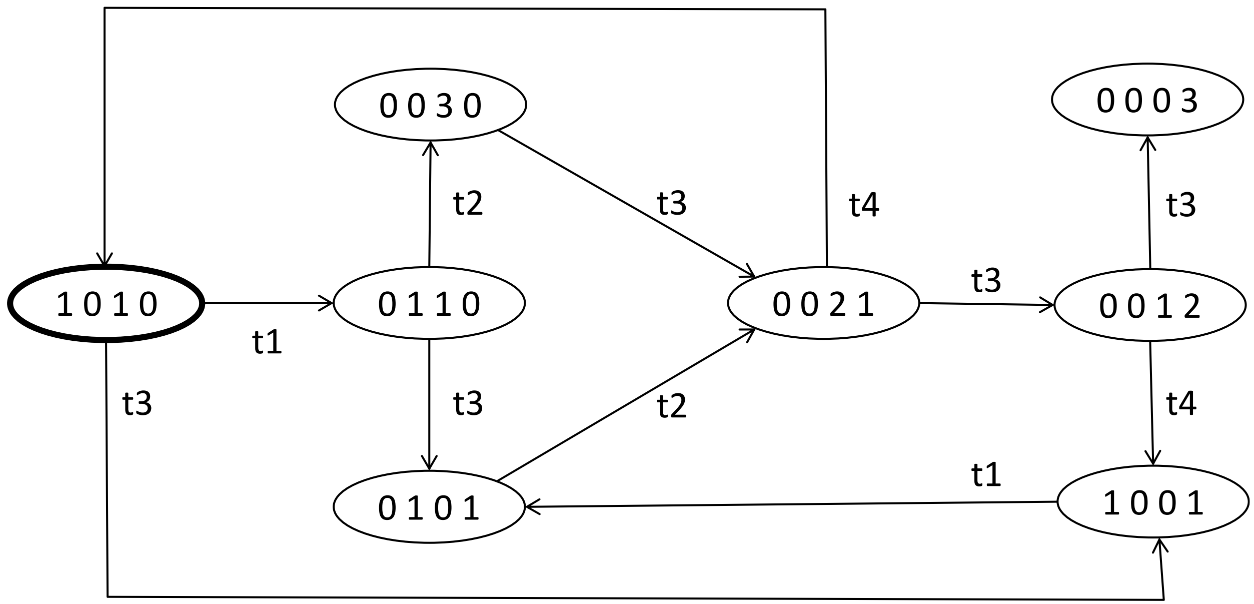


Általában figyelni kell:
Kapacitáskorlátos helyen nem jelenhet meg ω szimbólum.

Lásd itt: (1 0 0 0) után (1 1 0 0) erősen fedő, de a P2 helyen nem jelenik meg ω.

Petri-hálók dinamikus tulajdonságai

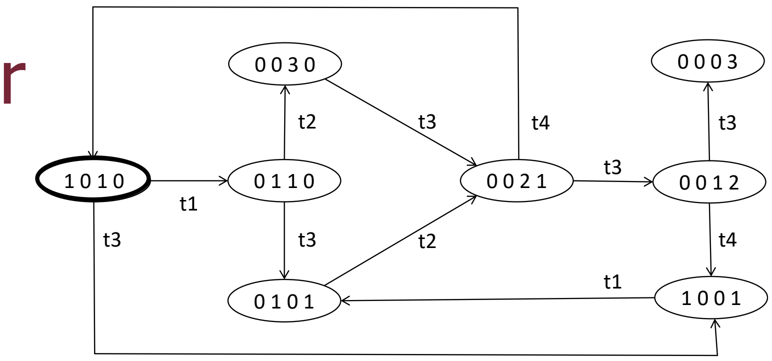
Egy Petri-háló jobb oldali állapottere (kezdőállapot az (1 0 1 0) állapot) alapján jelölje be az alábbi dinamikus tulajdonságok teljesülését:



		igaz	hamis	nem dönthető el
(a)	A háló elérhetőségi és fedési gráfja azonos	☐	☐	☐
(b)	A háló nem perzisztens	☐	☐	☐
(c)	A hálóban van holtpont (deadlock)	☐	☐	☐
(d)	*t3* tranzíció L2-élő	☐	☐	☐

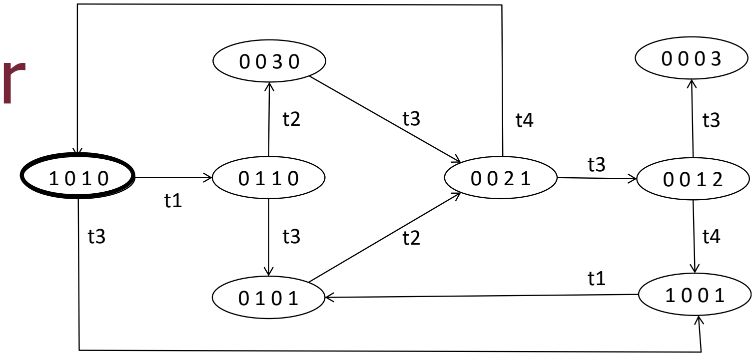
		igaz	hamis	nem dönthető el
(e)	A *t1*, *t2*, *t4* tüzelési szekvencia egy T-invariánst alkot	☐	☐	☐
(f)	*t3* és *t4* tranzíció korlátos fair	☐	☐	☐
(g)	A (0 1 0 1) állapot visszatérő állapot	☐	☐	☐

Dinamikus tulajdonságok – Háttér



- Korlátosság
  - Véges számú token jelenik-e meg minden helyen (normál háló esetén nincs erősen fedő jelölés)
  - Biztosság: Az elérhetőségi gráfban csak 0 és 1 jelölések vannak-e
- Megfordíthatóság
  - Minden elérhető állapotból vissza lehet-e jutni a kezdőállapotba?
- Visszatérő állapot
  - Egy visszatérő állapotba az abból elérhető minden állapotból vissza lehet jutni.
- Perzisztens tranzíció
  - A tranzíció engedélyezetté válva az is marad-e, amíg nem tüzel. Ellenpélda: Egy állapotban több tranzíció engedélyezett, és ha nem az adott tranzíció tüzelt, akkor az így elért állapotban nem marad engedélyezett (azaz nem jelenik meg kimenő él címkéjeként)

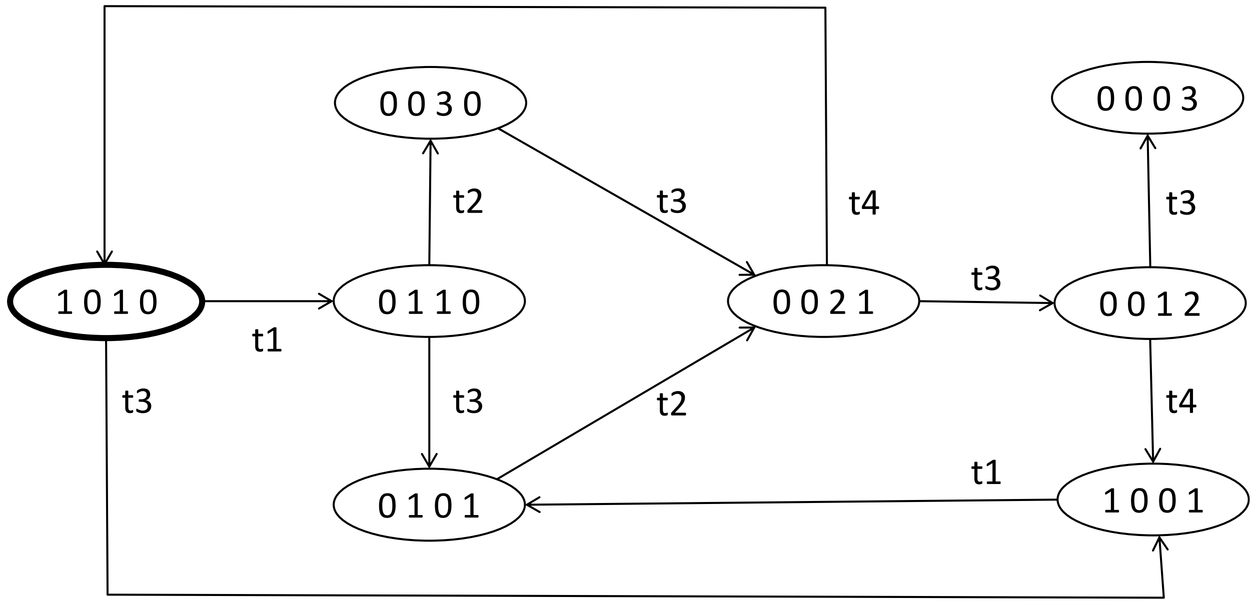
Dinamikus tulajdonságok – Háttér



- Korlátozott fair tüzelési szekvencia
  - Egy engedélyezett tranzíció korlátosan sokszor tüzelhet-e, mielőtt a másik engedélyezett tranzíció tüzelne? Ellenpélda: Olyan ciklus, amiben az egyik tranzíció benne van, de a másik engedélyezett tranzíció nincs (mindig kimarad a tüzelésből).
- Tranzíció L1, L2, L3 élősége
  - Elég egy trajektóriát találni, ahol teljesül, hogy a tranzíció 1-szer (L1), k-szor (L2), végtelenszer (L3) tüzelhet (véges állapottérben: ciklus).
- Tranzíció L4 élősége
  - A tranzíció minden állapotból előbb-utóbb mindig tüzelhetővé válik-e?
- A háló élő
  - A háló akkor élő, ha minden tranzíciója L4-élő. Ellenpélda: Olyan tranzíció, ami egy adott állapotból folytatva a működést már nem válik előbb-utóbb tüzelhetővé.
  - Ha van holtpont, akkor a háló biztosan nem élő.
  - Ha holtpontmentes, akkor még nem biztos, hogy élő.

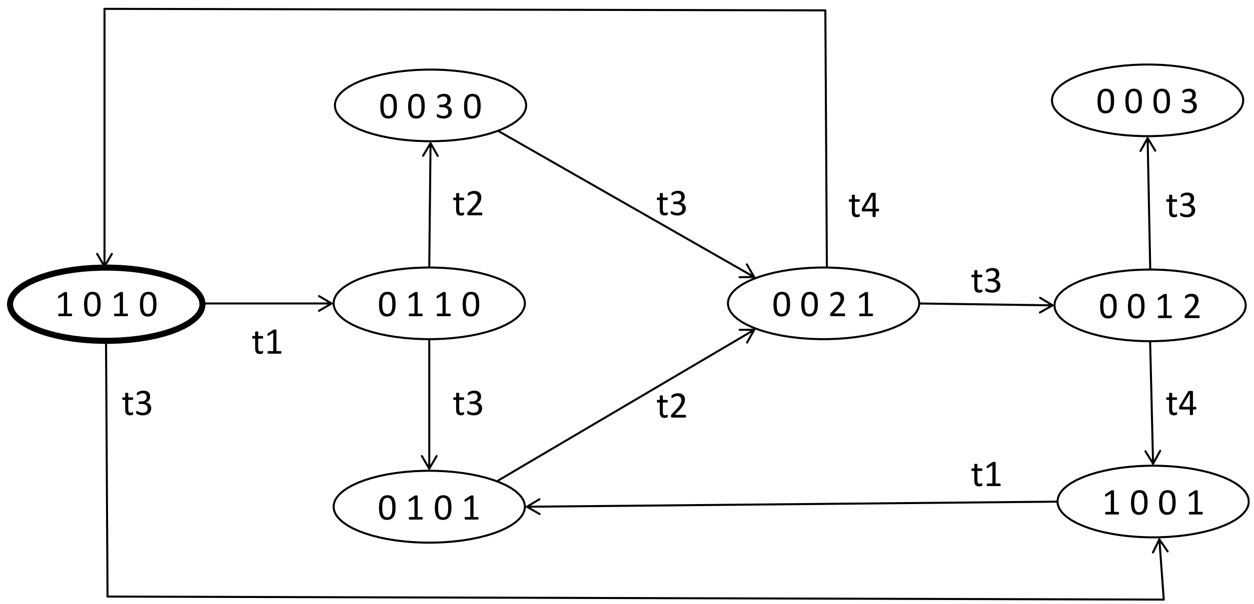
Petri-hálók dinamikus tulajdonságai

Egy Petri-háló jobb oldali állapottere (kezdőállapot az (1 0 1 0) állapot) alapján jelölje be az alábbi dinamikus tulajdonságok teljesülését:



		igaz	hamis	nem dönthető el			igaz	hamis	nem dönthető el
(a)	A háló elérhetőségi és fedési gráfja azonos	☐	☐	☐	(e)	A *t1*, *t2*, *t4* tüzelési szekvencia egy T-invariánst alkot	☐	☐	☐
(b)	A háló nem perzisztens	☐	☐	☐	(f)	*t3* és *t4* tranzíció korlátos fair	☐	☐	☐
(c)	A hálóban van holtpont (deadlock)	☐	☐	☐	(g)	A (0 1 0 1) állapot visszatérő állapot	☐	☐	☐
(d)	*t3* tranzíció L2-élő	☐	☐	☐					

Petri-hálók dinamikus tulajdonságai – Megoldás



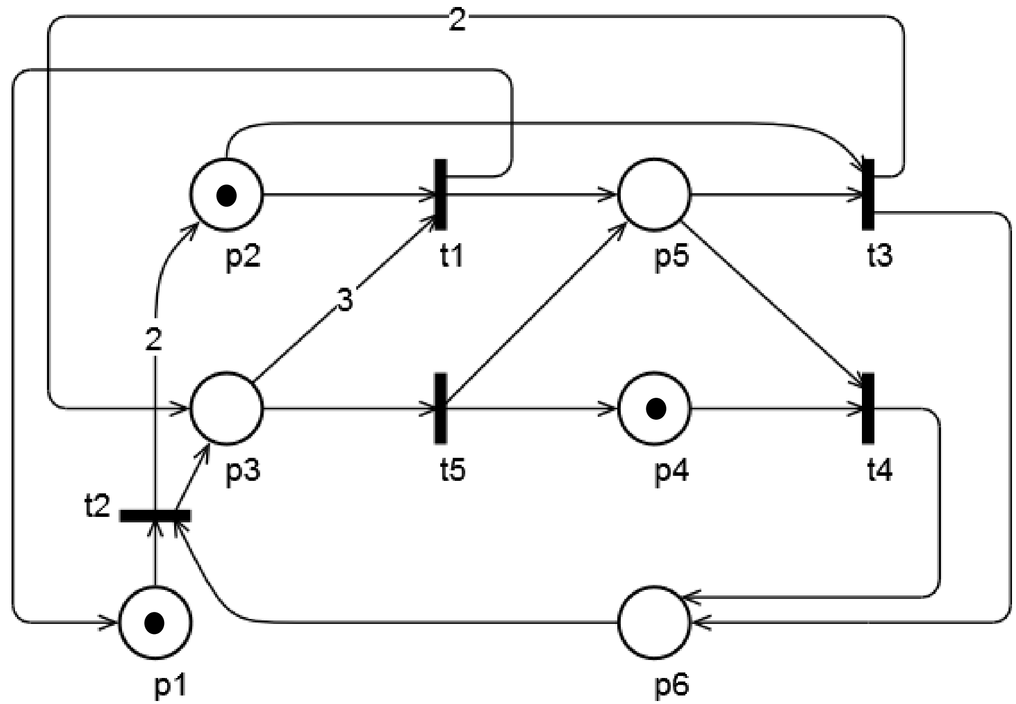
a) Elérhetőségi és fedési gráf azonos: Igaz, mert korlátos.
b) A háló nem perzisztens: Igaz, lásd (0 0 1 2) esetén t3 és t4.
c) A hálóban van holtpont: Igaz, lásd (0 0 0 3).
d) t3 L2 élő: Igaz, mert t3-at tartalmazó ciklus van, de ebből kilépve holtpont.
e) t1, t2, t3 tüzelési szekvencia T-invariáns: Hamis, ilyen ciklus nincs.
f) t3 és t4 korlátos fair: Igaz, mert egymás nélkül nem szerepelnek ciklusban.
g) (0 1 0 1) visszatérő állapot: Hamis, mert (0 0 0 3)-ból nem elérhető.

Strukturális tulajdonságok (1)

Adott az ábrán látható Petri-háló és a hozzátartozó W^T szomszédossági mátrix.

Milyen számokat kell a mátrixban a betűvel jelölt kitöltetlen helyekre írni?

- A =
- B =
- C =
- D =



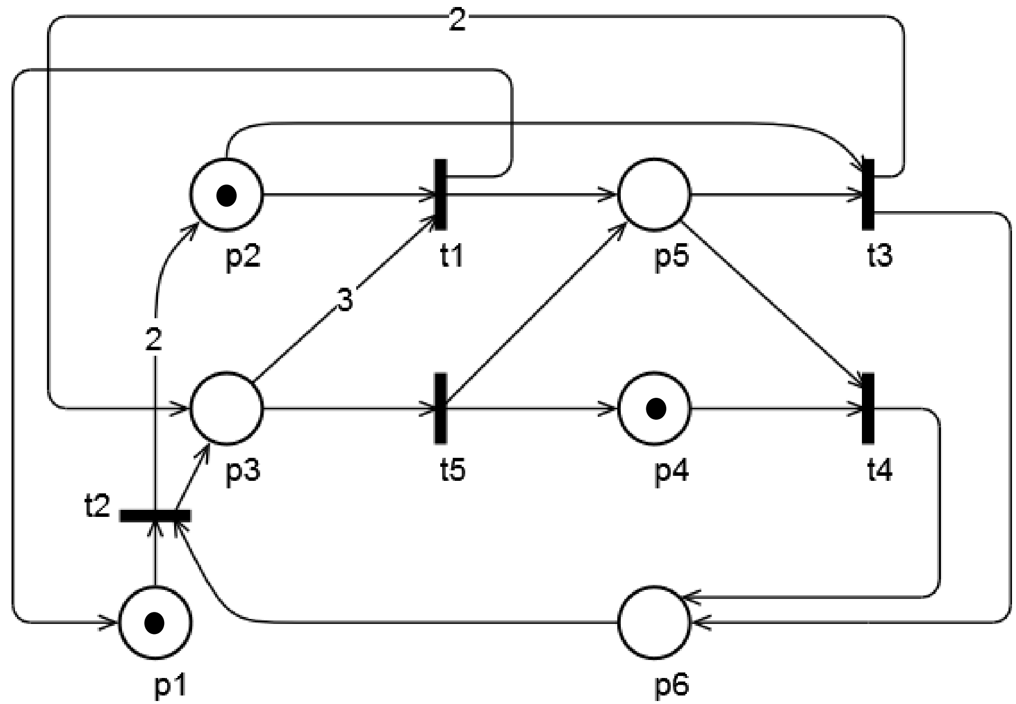
$$W^T = \begin{array}{c|ccccc} & t_1 & t_2 & t_3 & t_4 & t_5 \\ \hline p_1 & 1 & -1 & 0 & 0 & 0 \\ p_2 & -1 & 2 & \mathbf{B} & 0 & 0 \\ p_3 & \mathbf{A} & 1 & 2 & 0 & -1 \\ p_4 & 0 & 0 & 0 & \mathbf{C} & 1 \\ p_5 & 1 & 0 & -1 & -1 & 1 \\ p_6 & 0 & -1 & 1 & 1 & \mathbf{D} \end{array}$$

Strukturális tulajdonságok (1) – Megoldás

Adott az ábrán látható Petri-háló és a hozzátartozó W^T szomszédossági mátrix.

Milyen számokat kell a mátrixban a betűvel jelölt kitöltetlen helyekre írni?

- A = -3
- B = -1
- C = -1
- D = 0



$$W^T = \left[\begin{array}{cccccc} & t_1 & t_2 & t_3 & t_4 & t_5 \\ p_1 & 1 & -1 & 0 & 0 & 0 \\ p_2 & -1 & 2 & \mathbf{B} & 0 & 0 \\ p_3 & \mathbf{A} & 1 & 2 & 0 & -1 \\ p_4 & 0 & 0 & 0 & \mathbf{C} & 1 \\ p_5 & 1 & 0 & -1 & -1 & 1 \\ p_6 & 0 & -1 & 1 & 1 & \mathbf{D} \end{array}\right]$$

Strukturális tulajdonságok (2): T-invariánsok

Ellenőrizze az állapot-egyenlet alapján, hogy az alábbiak közül melyek T-invariánsai a Petri-hálónak!

$$W^T = \begin{array}{c|ccccc} & t_1 & t_2 & t_3 & t_4 & t_5 \\ \hline p_1 & 1 & -1 & 0 & 0 & 0 \\ p_2 & -1 & 2 & \mathbf{B} & 0 & 0 \\ p_3 & \mathbf{A} & 1 & 2 & 0 & -1 \\ p_4 & 0 & 0 & 0 & \mathbf{C} & 1 \\ p_5 & 1 & 0 & -1 & -1 & 1 \\ p_6 & 0 & -1 & 1 & 1 & \mathbf{D} \end{array}$$

A = -3, B = -1, C = -1, D = 0

- $(2,2,2,0,0)^T$
- $(0,1,0,1,3)^T$

Strukturális tulajdonságok (3): P-invariánsok

Ellenőrizze az állapot-egyenlet alapján, hogy az alábbiak közül melyek P-invariánsai a Petri-hálónak!

A = -3, B = -1, C = -1, D = 0

- $(4,2,1,0,1,1)^T$
- $(3,1,0,2,0,1)^T$

$$
W^T = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{array}
\begin{array}{c} \begin{array}{ccccc} t_1 & t_2 & t_3 & t_4 & t_5 \end{array} \\
\begin{bmatrix}
1 & -1 & 0 & 0 & 0 \\
-1 & 2 & \mathbf{B} & 0 & 0 \\
\mathbf{A} & 1 & 2 & 0 & -1 \\
0 & 0 & 0 & \mathbf{C} & 1 \\
1 & 0 & -1 & -1 & 1 \\
0 & -1 & 1 & 1 & \mathbf{D}
\end{bmatrix} \end{array}
$$

Strukturális tulajdonságok – Háttér

T-invariánsok:

- Definíció:
  A σ_T tüzelési szám vektor T-invariáns, ha az általa megadott tüzelések végrehajtása nem változtatja meg a tokeneloszlást.
- Számítás, ellenőrzés alapja:

$$\boxed{\mathbf{W}^{\mathrm{T}}\sigma_T = 0}$$

P-invariánsok:

- Definíció:
  A μ_P nem-negatív súlyvektor P-invariáns, ha az által megadott súlyozott tokenösszeg nem változik a háló működése során.
- Számítás, ellenőrzés alapja:

$$\boxed{\mathbf{W}\mu_P = 0}$$

Strukturális tulajdonságok (2) – Megoldás

Ellenőrizze az állapot-egyenlet alapján, hogy az alábbiak közül melyek T-invariánsai a Petri-hálónak!

$$\mathbf{W}^{\mathrm{T}} = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{array} \begin{array}{c} \begin{array}{ccccc} t_1 & t_2 & t_3 & t_4 & t_5 \end{array} \\ \left[\begin{array}{ccccc} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & \mathbf{B} & 0 & 0 \\ \mathbf{A} & 1 & 2 & 0 & -1 \\ 0 & 0 & 0 & \mathbf{C} & 1 \\ 1 & 0 & -1 & -1 & 1 \\ 0 & -1 & 1 & 1 & \mathbf{D} \end{array}\right] \end{array}$$

A = -3, B = -1, C = -1, D = 0

Ellenőrzés az állapotegyenlet alapján: $\mathbf{W}^{\mathrm{T}}\sigma_T = 0$

- $(2,2,2,0,0)^T$ T-invariáns
- $(0,1,0,1,3)^T$ Nem T-invariáns

Strukturális tulajdonságok (3) – Megoldás

Ellenőrizze az állapot-egyenlet alapján, hogy az alábbiak közül melyek P-invariánsai a Petri-hálónak!

$$\mathbf{W}^T = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{array} \begin{bmatrix} t_1 & t_2 & t_3 & t_4 & t_5 \\ 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & \mathbf{B} & 0 & 0 \\ \mathbf{A} & 1 & 2 & 0 & -1 \\ 0 & 0 & 0 & \mathbf{C} & 1 \\ 1 & 0 & -1 & -1 & 1 \\ 0 & -1 & 1 & 1 & \mathbf{D} \end{bmatrix}$$

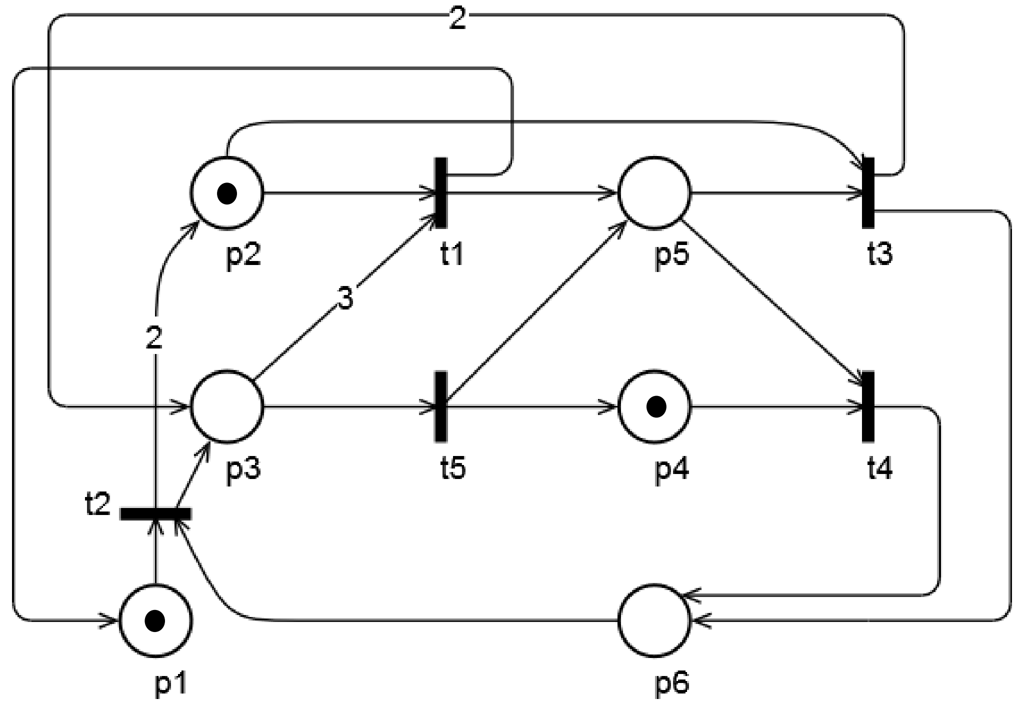
A = -3, B = -1, C = -1, D = 0

Ellenőrzés az állapotegyenlet alapján: $\mathbf{W}\mu_P = 0$

- $(4,2,1,0,1,1)^T$ P-invariáns
- $(3,1,0,2,0,1)^T$ Nem P-invariáns

Temporális tulajdonságok

A jobb oldali Petri-hálóra az M(1,1,0,1,0,0) kezdőállapotból igazak-e a következő, CTL temporális logikával megadott állítások:

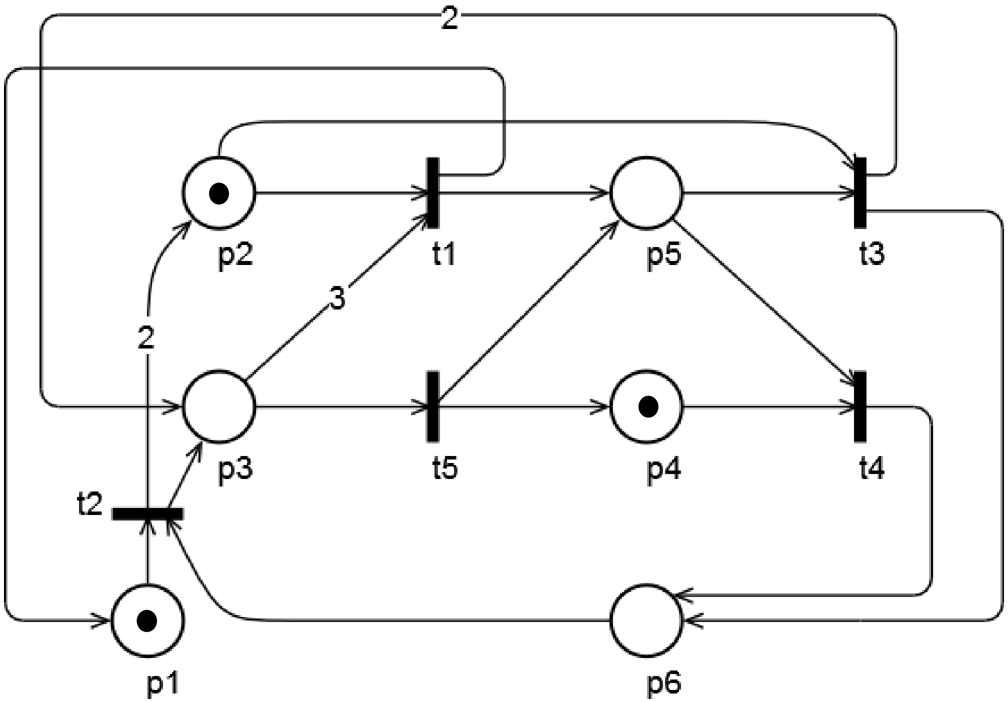


AG (4*m(p1) + 2*m(p2) + m(p3) + m(p5) + m(p6) = 6)

EF (4*m(p1) + 2*m(p2) + m(p3) + m(p5) + m(p6) = 3)

Temporális tulajdonságok – Megoldás (1)

A jobb oldali Petri-hálóra az M(1,1,0,1,0,0) kezdőállapotból igazak-e a következő, CTL temporális logikával megadott állítások:

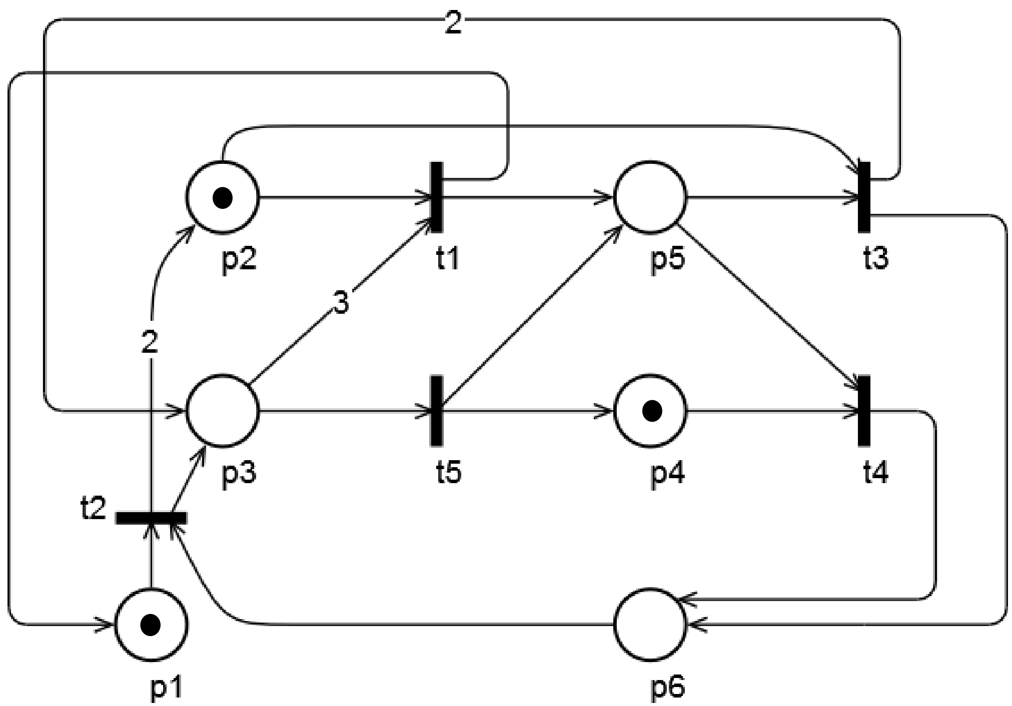


AG (4*m(p1) + 2*m(p2) + m(p3) + m(p5) + m(p6) = 6)

- A kezdőállapotra teljesül: a tokenösszeg 6
- A tokenösszeg súlyai alapján felvett $(4,2,1,0,1,1)^T$ egy P-invariáns, ez $\mathbf{W} \cdot \mu_p = 0$ alapján ellenőrizhető
- Tehát a fenti tokenösszeg állandó az állapottérben (**AG**)

Temporális tulajdonságok – Megoldás (2)

A jobb oldali Petri-hálóra az M(1,1,0,1,0,0) kezdőállapotból igazak-e a következő, CTL temporális logikával megadott állítások:



EF (4*m(p1) + 2*m(p2) + m(p3) + m(p5) + m(p6) = 3)

- A kezdőállapotra nem teljesül: a tokenösszeg 6 és nem 3
- Később sem teljesülhet, mivel a tokenösszeg súlyai alapján felvett $(4,2,1,0,1,1)^T$ egy P-invariáns (ez $\mathbf{W} \cdot \mu_p = 0$ alapján ellenőrizhető), így állandó a 6 tokenösszegként.

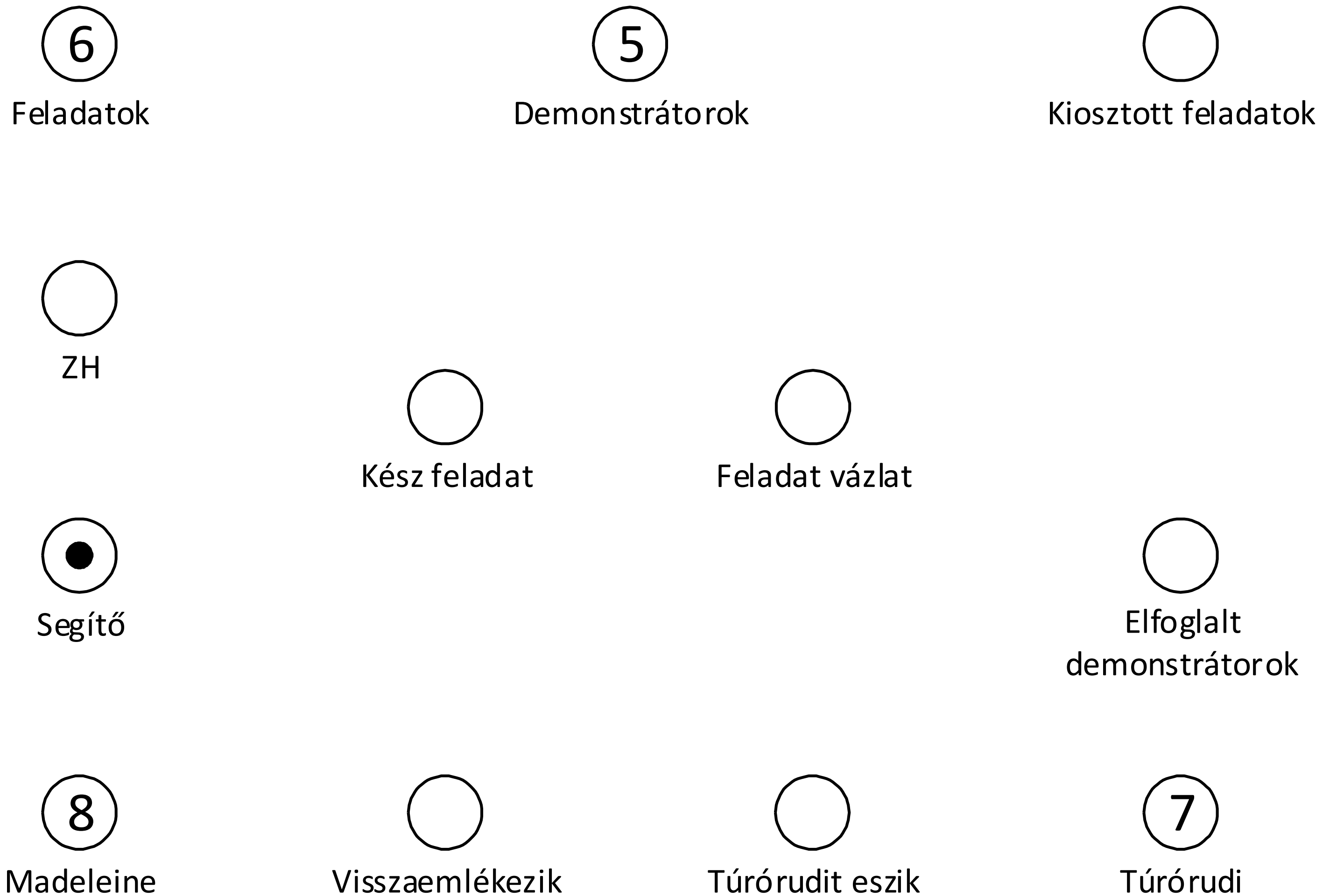
Modellezés színezetlen Petri-hálókkal (1/2)

Készítse el a következő folyamat színezetlen Petri-háló modelljét!

1. A Formális módszerek ZH 6 feladatból áll. A ZH előkészítésén az előadó, 1 segítő és 5 demonstrátor dolgozik.
2. Az 6 feladat mindegyikéhez az előadó vagy maga állítja össze a feladat vázlatot, vagy kiosztja egy demonstrátornak. Egy demonstrátor, ha feladatot kap, elfoglalt lesz, és nem kaphat több feladatot.
3. Az elkészült feladat vázlatokat a segítő egyenként átnézi. Az átnézés után a feladat kész.
4. Ha az összes feladat kész, az előadó összeállítja a ZH-t.
5. A tanszéken 7 túrórudi és 8 madeleine-sütemény van.
6. Az elfoglalt demonstrátorok néha megéheznek, és elkezdenek túrórudit enni (ha van túrórudi). Ekkor nem tudnak a feladatok összeállításán dolgozni, amíg be nem fejezik az evést.
7. A segítő inkább a madeleine-süteményt szereti, tehát időnként ezt eszi. A sütemény hatására elkezd visszaemlékezni saját diákkorára, és amíg be nem fejezi, nem néz át feladatot.

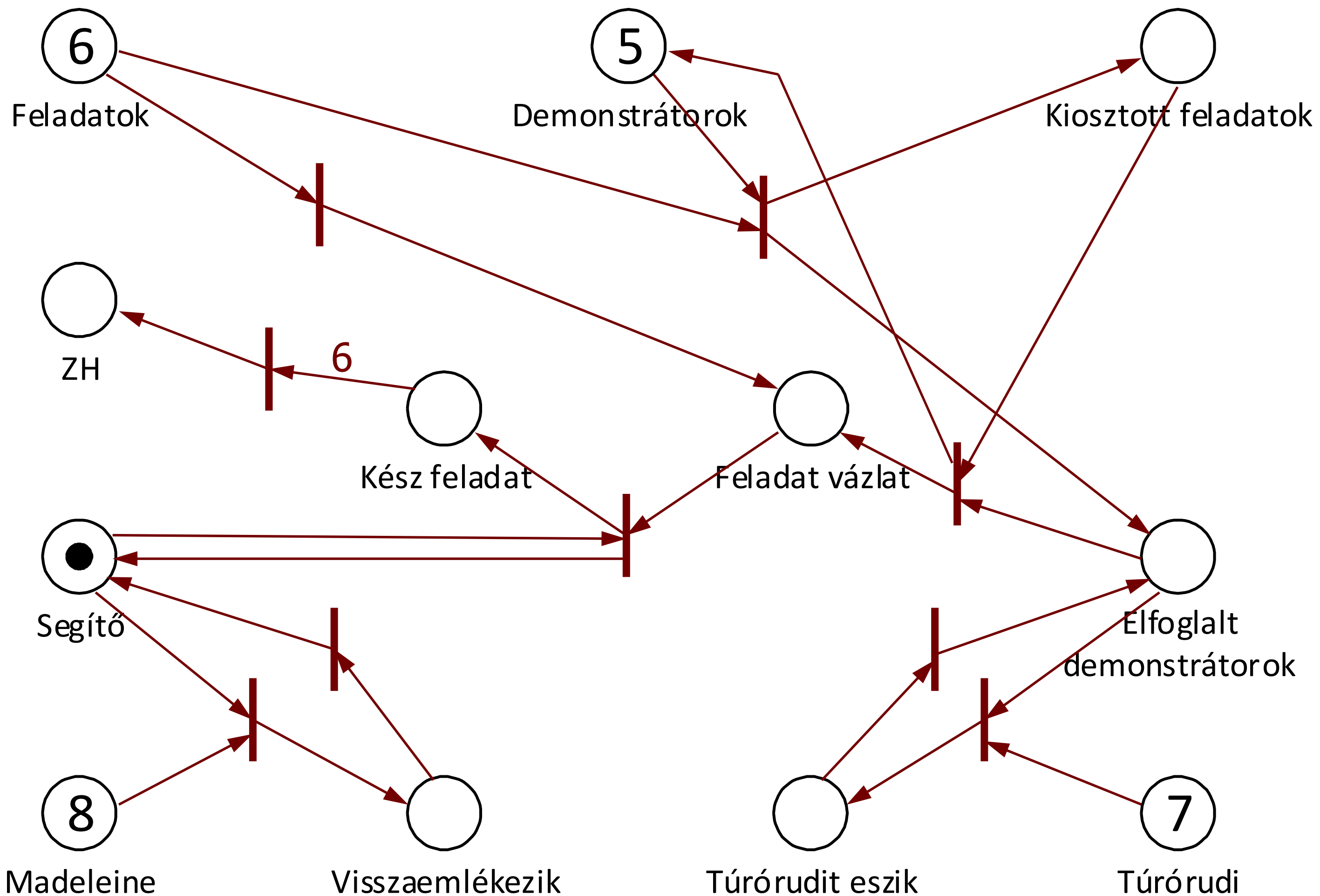
Modellezés színezetlen Petri-hálókkal (2/2)

Az előző leírás szerinti modellt a lenti a lenti modell-részletet kiegészítve készítse el!



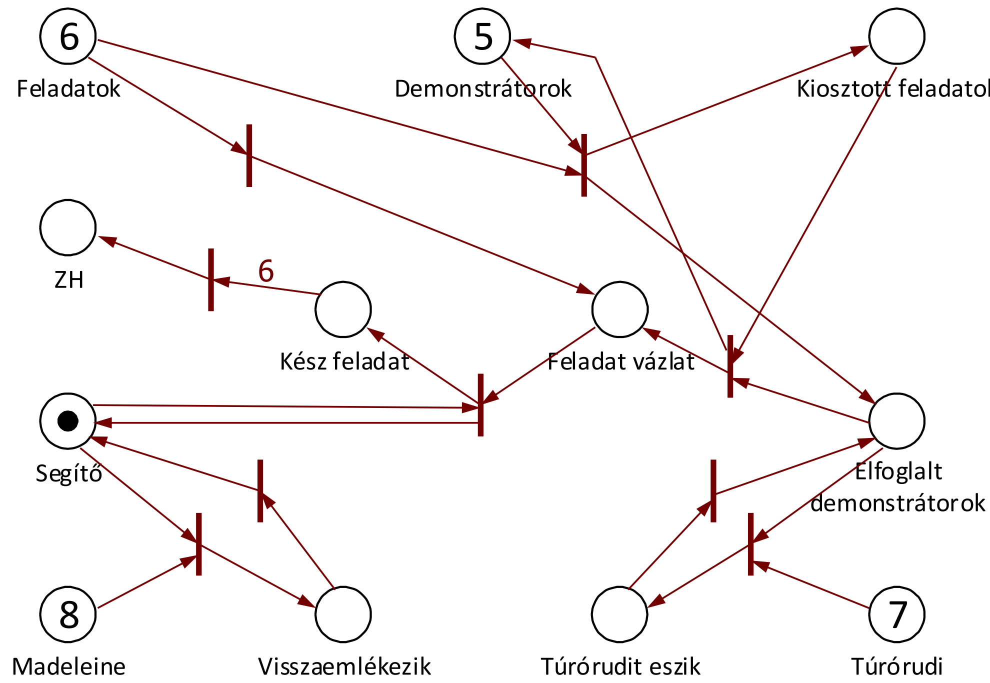
Modellezés színezetlen Petri-hálókkal – Megoldás

A megadott modell-részlet kiegészítése:



Modellezés színezetlen Petri-hálókkal – Megoldás

- Az 6 feladat mindegyikéhez az előadó vagy maga állítja össze a feladat vázlatot, vagy kiosztja egy demonstrátornak. Egy demonstrátor, ha feladatot kap, elfoglalt lesz, és nem kaphat több feladatot.
- Az elkészült feladat vázlatokat a segítő egyenként átnézi. Az átnézés után a feladat kész.
- Ha az összes feladat kész, az előadó összeállítja a ZH-t.
- Az elfoglalt demonstrátorok néha megéheznek, és elkezdenek túrórudit enni (ha van túrórudi). Ekkor nem tudnak a feladatok összeállításán dolgozni, amíg be nem fejezik az evést.
- A segítő inkább a madeleine-süteményt szereti, tehát időnként ezt eszi. A sütemény hatására elkezd visszaemlékezni saját diákkorára, és amíg be nem fejezi, nem néz át feladatot.

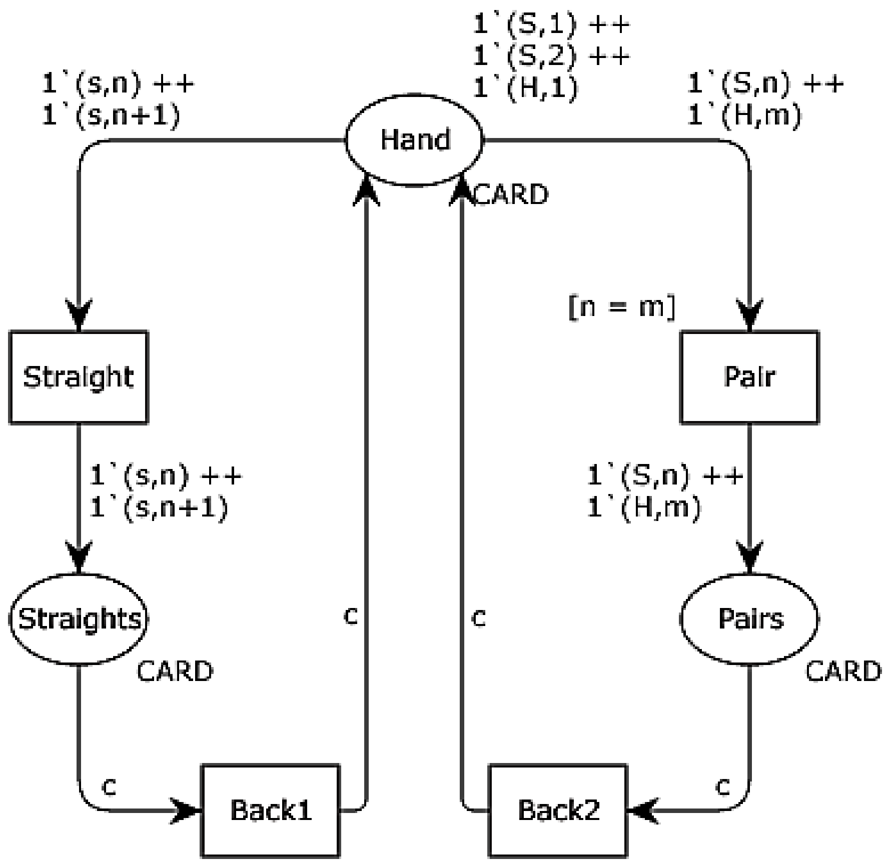


Modellezés színezett Petri-hálókkal

Adott az ábrán látható színezett Petri-háló modell és a hozzá tartozó definíciós mező.

1. Mely tranzíciók és milyen lekötéssel engedélyezettek az adott állapotban?
2. Tüzelés után mik lehetnek a háló következő jelölései? Válasszon ki egyet ezek közül és adja meg az ezután következő lehetséges lekötéseket!
3. Korlátos-e a háló az adott kezdőállapottal?
4. Holtpontmentes-e a háló az adott kezdőállapottal?
5. Van-e a hálóban T-invariáns?

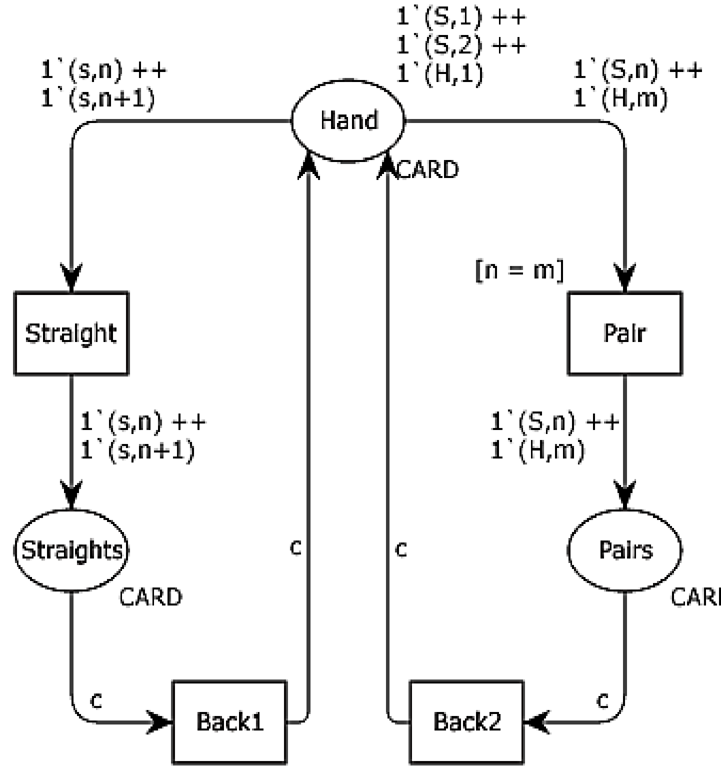
```
colset SUIT = with S | H;
colset NUM = int with 0..12;
colset CARD = product SUIT * NUM;
var s : SUIT;
var n, m : NUM;
var c : CARD;
```



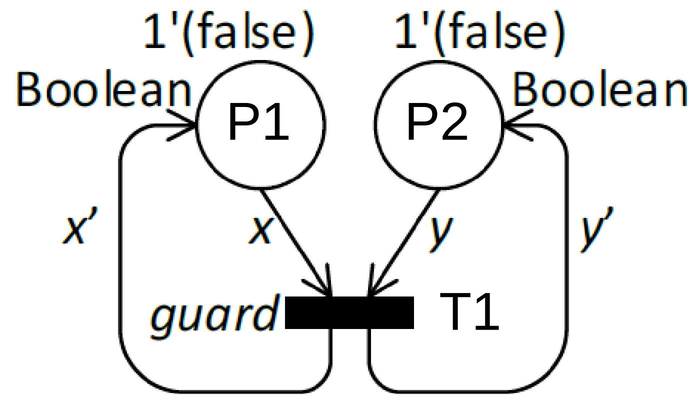
Modellezés színezett Petri-hálókkal – Megoldás

1. Engedélyezett:
   - **Straight**, s=S, n=1 lekötéssel
   - **Pair**, n=1, m=1 lekötéssel
2. Következő jelölések, majd lekötések:
   - **Straight** tüzel: **Hand** lesz 1′(H,1),
     Straights lesz 1′(S,1)++1′(S,2).
     Ezután engedélyezett:
     Back1, c=(S,1) vagy c=(S,2) lekötéssel.
   - **Pair** tüzel: **Hand** lesz 1′(S,2),
     Pairs lesz 1′(S,1)++1′(H,1).
     Ezután engedélyezett:
     Back2, c=(S,1) vagy c=(H,1) lekötéssel.
3. Korlátos-e:
   Korlátos: Tranzíciók nem „termelhetnek” tokeneket; a tokenek száma a hálóban nem változik.
4. Holtpontmentes-e:
   Igen: A **Back1** illetve **Back2** mindig vissza tudja (egyenként) rakni, amit a **Straight** illetve a **Pair** elvett, tehát mindig ciklikus a működés.
5. T-invariánsok „kézzel” is megkereshetők:
   - **Straight, Back1, Back1**
   - **Pair, Back2, Back2**

```
colset SUIT = with S | H;
colset NUM = int with 0..12;
colset CARD = product SUIT * NUM;
var s : SUIT;
var n, m : NUM;
var c : CARD;
```



Színezett Petri-hálók széthajtogatása



Adott az ábrán látható színezett Petri-háló modell és a hozzá tartozó definíciós mező:

```
var x, y, x',y': Boolean;
```

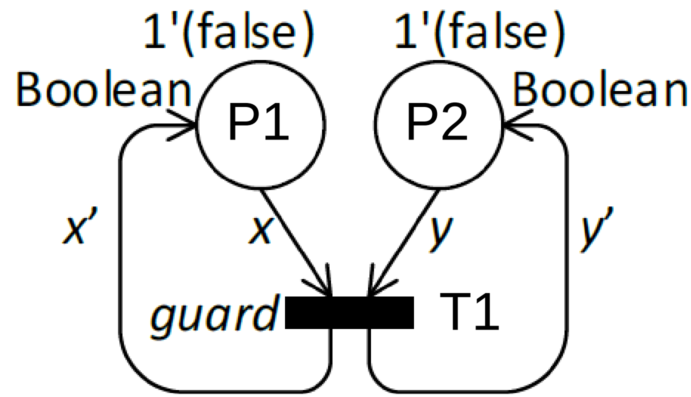
Az őrfeltétel: guard =

$$(\neg x \wedge \neg y \wedge x' \wedge \neg y') \vee (x \wedge \neg y \wedge \neg x' \wedge y') \vee (\neg x \wedge y \wedge x' \wedge y')$$

- Készítse el a színezett Petri-háló struktúrával ekvivalens működésű színezetlen Petri-háló struktúrát, azaz a színezett Petri-háló széthajtogatását!
- Élő-e és/vagy korlátos-e a fenti színezett háló és az ekvivalens működésű széthajtogatott színezetlen háló az adott (vagy bármilyen korlátos) kezdőállapottal?

Színezett Petri-hálók széthajtogatása – Megoldás

Adott az ábrán látható színezett Petri-háló modell és a hozzá tartozó definíciós mező:




```
var x, y, x’,y’: Boolean;
```

Az őrfeltétel: guard =

$$(\neg x \wedge \neg y \wedge x' \wedge \neg y') \vee (x \wedge \neg y \wedge \neg x' \wedge y') \vee (\neg x \wedge y \wedge x' \wedge y')$$

- Petri-háló széthajtogatása:
- Élő-e és/vagy korlátos-e a fenti színezett háló?
  - Nem élő (t1 t2 t3 után holtpont)
  - Korlátos: Egyik tranzíció sem szaporíthatja a tokeneket.