Petri hálók dinamikus tulajdonságai

dr. Bartha Tamás

dr. Majzik István

dr. Pataricza András

BME Méréstechnika és Információs Rendszerek Tanszék

Petri hálók vizsgálata

Vizsgálati lehetőségek

Az elemzés mélysége szerint:

- Szimuláció — Egy-egy trajektória bejárása
- Állapottér bejárása — Minden trajektória bejárása adott kezdőállapotból (kimerítő bejárás)
  - elérhetőségi gráf analízis
  - dinamikus (viselkedési) tulajdonságok
- Strukturális tulajdonságok — Kezdőállapottól független tulajdonságok (bármely kezdőállapotra)
  - invariáns analízis

ha mindez nem vezet eredményre

- Algebrai közelítés, részleges döntés

Petri hálók kezdőállapot-függő analízise

- Elérhetőségi analízis
  - Az elérhetőségi gráf (állapottér!) konstrukciójával
  - Dinamikus (viselkedési) tulajdonságok:
    - Elérhetőség, fedhetőség, élőség, holtpontmentesség, korlátosság, fairség, megfordíthatóság
  - Kezdőállapot-függő
    - Nem általános érvényű tulajdonságok

- Segíthetnek az analízisben: Redukciós technikák
  - Tulajdonságmegtartó transzformációk
    - A struktúra (és így az állapottér) szisztematikus csökkentése

Petri hálók kezdőállapot-független analízise

- Kezdőállapot-független tulajdonságok
  - Meghatározhatók csak a struktúra alapján
    - Univerzális (minden működésre érvényes), vagy
    - Egzisztenciális (lehetséges ilyen működés)
  - Jellegzetes strukturális tulajdonságok:
    - Strukturális élőség, strukturális korlátosság, vezérelhetőség, konzervativitás, ismételhetőség, konzisztencia
- Invariáns tulajdonságok
  - Tüzelési (T-, transition) invariáns
    - Lehetséges olyan működés, hogy az állapot újra előálljon
  - Hely (P-, place) invariáns
    - Minden állapotban igaz és állandó
    - Súlyozott tokenösszeg $\Rightarrow$ dinamikus egyensúly

Petri háló modellek szimulációja

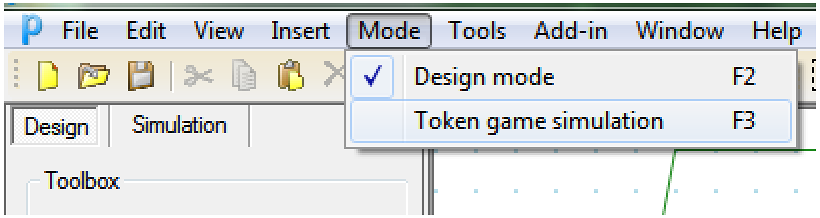
Diszkrét rendszerek szimulációja

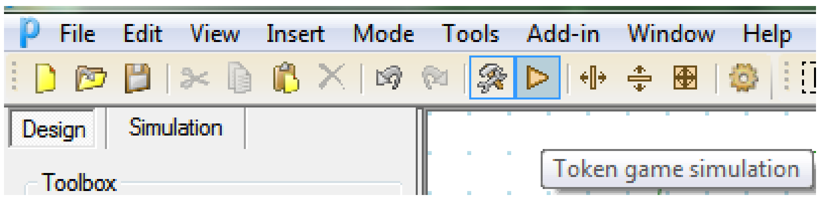
- Cél: a vizsgált rendszer „valósághű” modellezése
  - Valósághű: validálható (modell megfelelőség)
- Szimuláció folyamatmodellek esetén
  - Tevékenységorientált (eseményorientált)
  - Csak az események időpontjait tartjuk nyilván
    - Tevékenységek kezdete és vége (vagy időtartama)
    - Erőforrások lefoglalása és elengedése

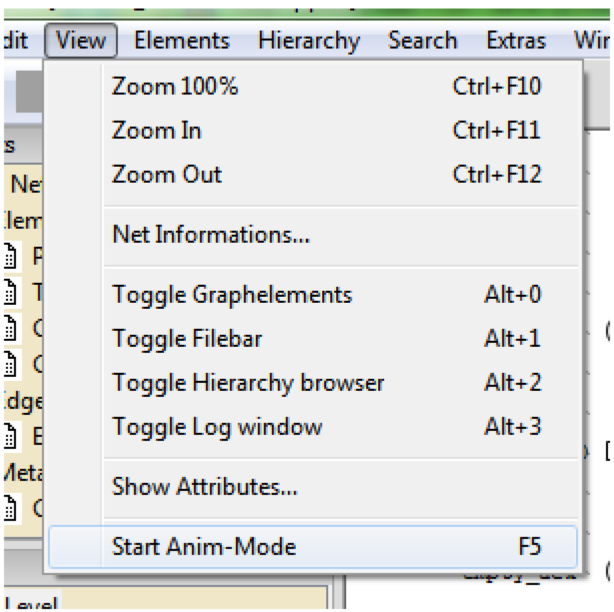
Petri hálók szimulációja

- A rendszer lehetséges trajektóriáinak vizsgálata
- Petri háló állapota: tokeneloszlás (jelölés)
  - Állapotváltás = tüzelés
  - Trajektóriák az állapottérben = tüzelési szekvenciák
- Petri háló nemdeterminisztikus
  - A nemdeterminizmust is modellezni kell
  - Valódi (ál-)véletlen generálásra van szükség
  - Állapottér bejárása: interaktív szimuláció (választás)

Animáció (token játék, token game)

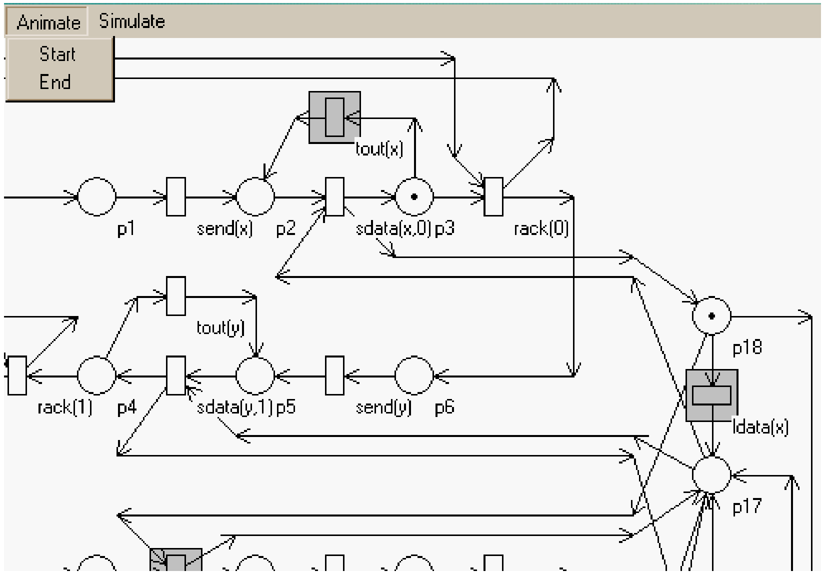


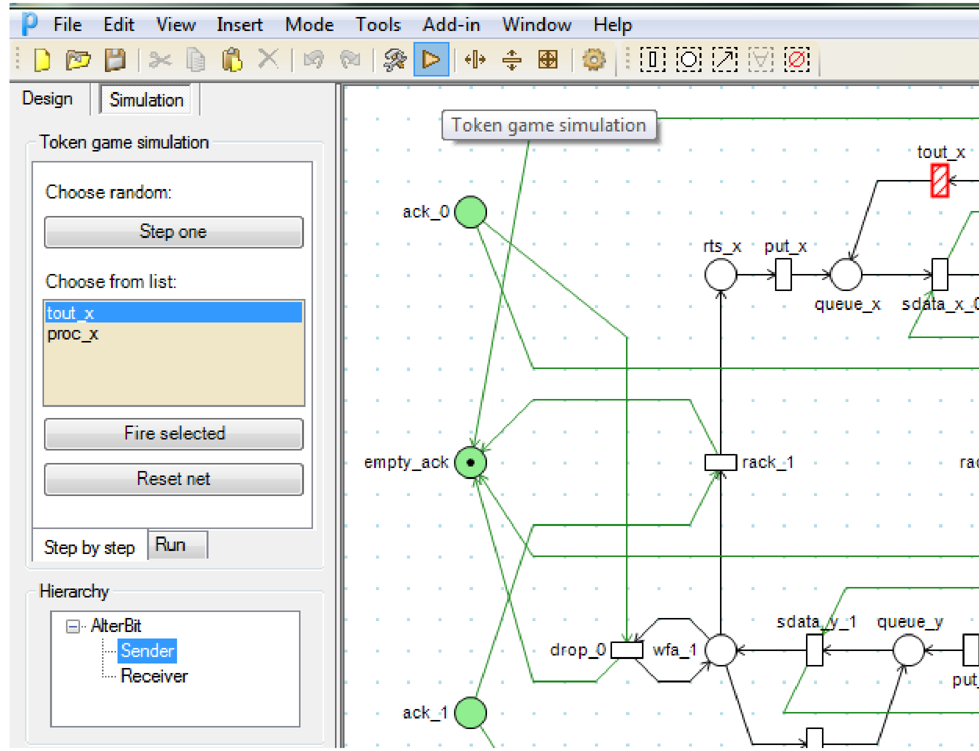


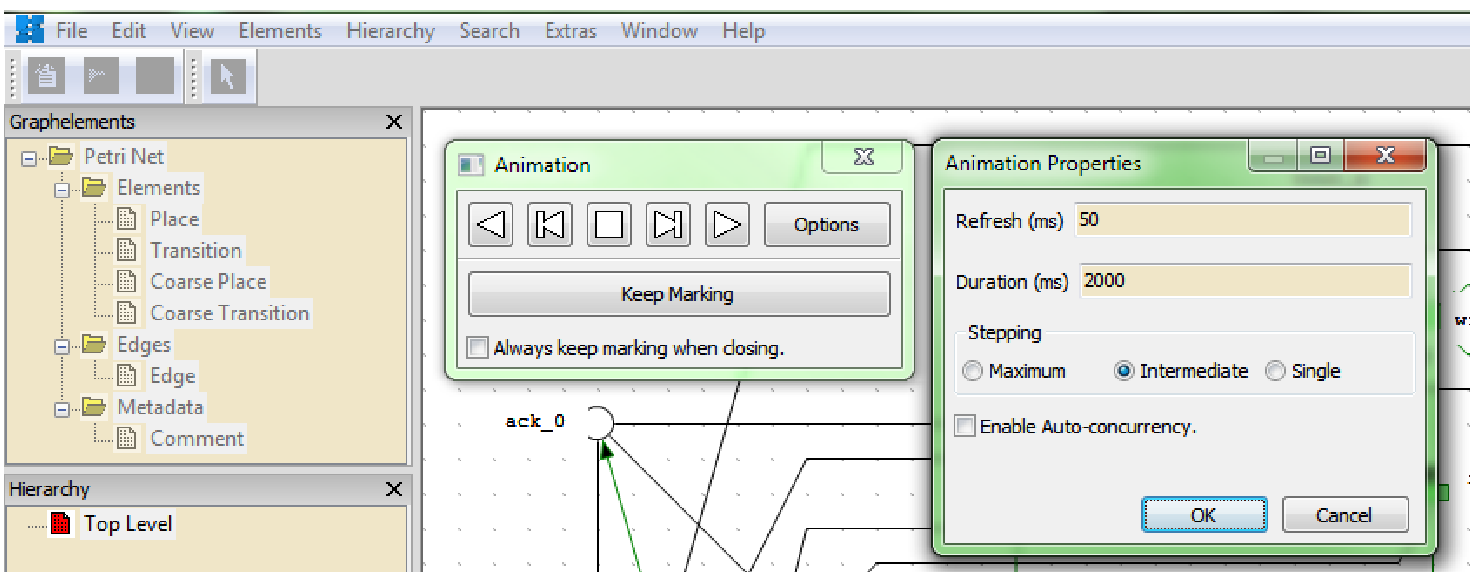


- A modell interaktív ellenőrzése
- Engedélyezett átmenetek jelölve
- Jelölt átmenetre kattintva tüzel
- Előállítja az új tokeneloszlást
- Konkurens átmenetek:
  - Manuálisan
  - Automatikus véletlen választással (pl. PetriDotNet)
- Befejezéskor visszaállítja a kezdeti tokeneloszlást

Animációs képernyő

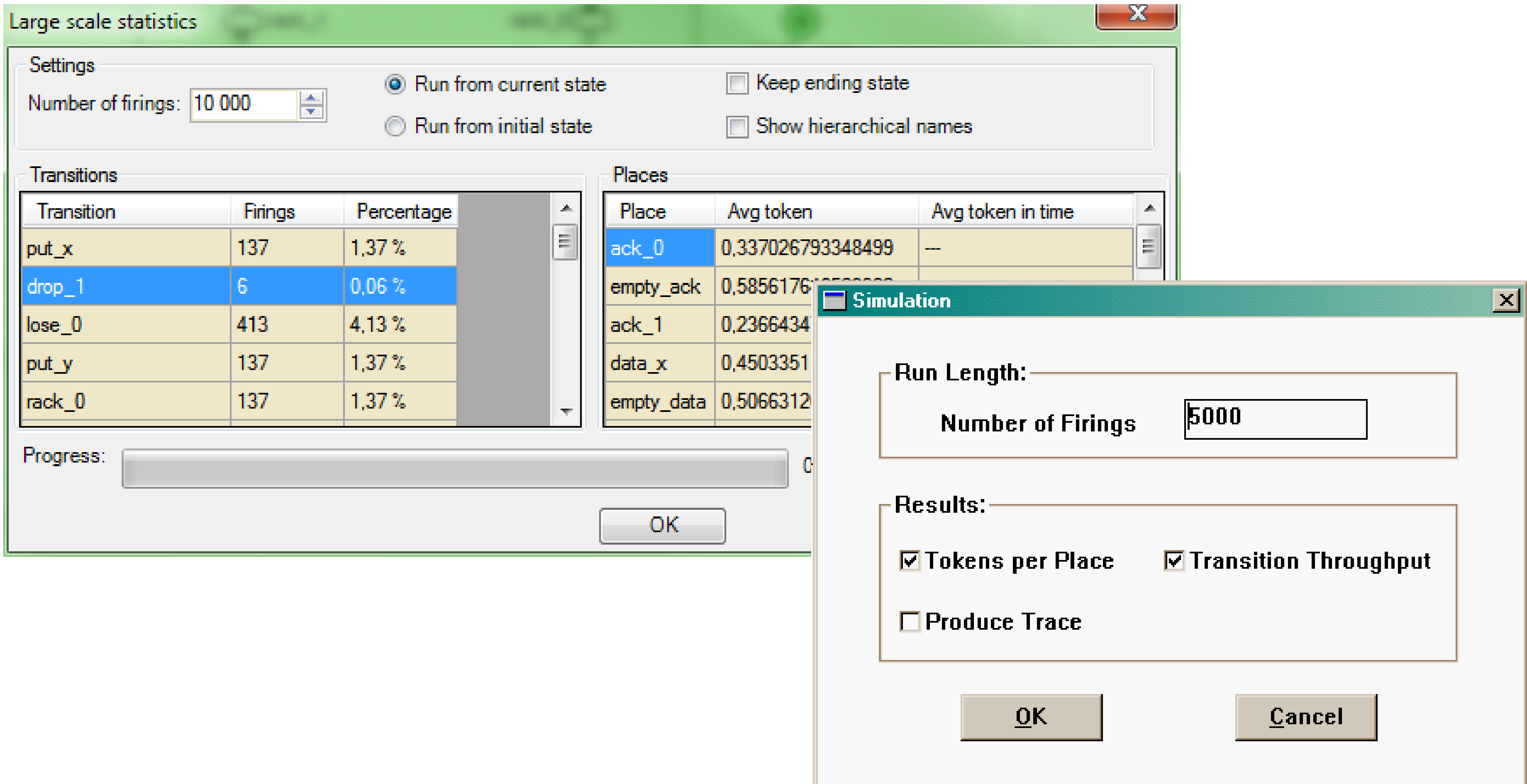






Szimuláció

- Lépések (tüzelések) számának beállítása
- Statisztika gyűjtése



Egyszerű szimulációs algoritmus

```
while (true) do
    Engedélyezett tranzíciók felmérése
    if (Van tüzelhető tranzíció)
      then Végrehajtandó kiválasztása (nemdeterminisztikus)
      else Szimuláció vége.
    Tüzelés
end while
```

Tüzelhető tranzíciók listájának összeállítása

function $collect\_fireable\_transitions(M)$

// Tüzelhető tranzíciók halmaza

$L_{fireable} \leftarrow \varnothing$

for all $t \in T$ **do**

if $enabled(t, M)$ **then** $L_{fireable} \leftarrow L_{fireable} \cup \{t\}$

return $L_{fireable}$

end function

Az állapotváltozás nagysága

Ha t tüzel M állapotban

- Új állapot: $M' = M + \mathbf{W}^T \cdot e_t$
  - ahol e_t a t tranzíciónak megfelelő egységvektor
- Itt **W** a súlyozott szomszédossági mátrix
  - **W** = [w(*t*, *p*)] ← t tüzelése hogyan módosítja p jelölését
  - Dimenziója: $\tau \times \pi = |T| \times |P|$ ← sorok × oszlopok
  - Ha t tüzel, mennyit változik a p -beli tokenszám:

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

Tüzelés

// Inicializálás

$M \leftarrow \mathrm{M}_0$

$L_{fireable} \leftarrow collect\_fireable\_transitions(M)$

while $L_{fireable} \neq \varnothing$ **do**

    // Tüzelés

    $t \leftarrow rnd(L_{fireable})$

    $M' \leftarrow M + \mathbf{W}^{\mathbf{T}} \cdot \underline{\mathrm{e}}_t$

    $L_{fireable} \leftarrow collect\_fireable\_transitions(M')$

    $M \leftarrow M'$

end while

Redundancia

- Miért kellene mindig megnézni az összes tranzíció tüzelhetőségét ($|T|$ számú vizsgálat), ha csak az éppen tüzelt tranzíció környezete ($\bullet t \cup t \bullet$) változik?
  - Letiltódó tranzíciók
  - Engedélyeződő tranzíciók

Letiltódó tüzelések

- Egy t tranzíció tüzelése során letiltódhat
  - Olyan t' tranzíció, melynek bemenete kapcsolódik $\bullet t$-hez
  - Konfliktusban van t-vel: $\bullet t' \cap \bullet t \neq \emptyset$
- Numerikus meghatározás:
  - Az elvett tokenek száma: $M^{-} = \mathbf{W}^{-T} \cdot \underline{e}_t$
  - t ősei (bemeneti helyei): $\bullet$t, itt $\{p \in P: M^{-}(p) > 0\}$
  - t által letiltódhatnak: $T' = \{(\bullet t)\bullet\}$

Engedélyeződő tüzelések

- Egy t tranzíció tüzelése során engedélyeződhet
  - olyan t'' tranzíció, melynek bemenete kapcsolódik $t\bullet$-hez
  - t engedélyezi t''-t: $\bullet t'' \cap t\bullet \neq \emptyset$
- Numerikus meghatározás
  - A hozzátett tokenek száma: $M^{+} = \mathbf{W}^{+\mathbf{T}} \cdot \underline{e}_t$
  - t utódai (kimeneti helyei): $t\bullet$, itt $\{p \in P: M^{+}(p) > 0\}$
  - t által engedélyeződhetnek: $T'' = \{(t\bullet)\bullet\}$
- Elég ezeket a tranzíciókat (ami letiltódhat vagy engedélyeződhet) újraértékelni!

Hatékony algoritmus: inicializálás

- Inicializálási fázis ugyanaz

// Inicializálás

$M \leftarrow \mathrm{M}_0$

$L_{fireable} \leftarrow \varnothing$

// Tüzelhető tranzíciók kezdeti halmaza

for all $t \in T$ **do**

  if $enabled(t, \mathrm{M}_0)$ **then** $L_{fireable} \leftarrow L_{fireable} \cup \{t\}$

Hatékony algoritmus: tüzelési ciklus

```
while L_fireable ≠ ∅ do
    // Tüzelés
    t ← rnd(L_fireable)
    M' ← M + W^T·e_t
    // Letiltottak eltávolítása
    for all t' ∈ {(•t)•} do
        if not(enabled(t', M')) then L_fireable ← L_fireable \ {t'}
    // Engedélyezettek bevonása
    for all t'' ∈ {(t•)•} do
        if enabled(t'', M') then L_fireable ← L_fireable ∪ {t''}
    M ← M'
end while
```

while $L_{fireable} \neq \varnothing$ **do**

// Tüzelés

$t \leftarrow rnd(L_{fireable})$

$M' \leftarrow M + \mathbf{W}^{\mathrm{T}} \cdot \underline{e}_t$

// Letiltottak eltávolítása

for all $t' \in \{(\bullet t)\bullet\}$ **do**

if $not(enabled(t', M'))$ **then** $L_{fireable} \leftarrow L_{fireable} \setminus \{t'\}$

// Engedélyezettek bevonása

for all $t'' \in \{(t\bullet)\bullet\}$ **do**

if $enabled(t'', M')$ **then** $L_{fireable} \leftarrow L_{fireable} \cup \{t''\}$

$M \leftarrow M'$

end while

Prioritás

- Tüzelési szabály kiegészül: t a.cs.a. tüzelhet, ha
  - Engedélyezett és
  - Nincs a $\pi(t)$ prioritásánál nagyobb prioritású engedélyezett
- Következmény:
  - $L_{fireable}$ nem halmaz, hanem halmazok $\pi \in \Pi$ prioritási szintek szerint rendezett $L_{fireable}[\pi]$ vektora
  - Tüzeléskor a legmagasabb prioritású nem üres $L_{fireable}[\pi]$ halmazból választunk nemdeterminisztikusan

Prioritásos algoritmus: inicializálás

// Inicializálás

$M \leftarrow \mathrm{M}_0$

for all $\pi \in \Pi$ **do**

  $L_{fireable}[\pi] \leftarrow \varnothing$

// Tüzelhető tranzíciók kezdeti halmaza

for all $t \in T$ **do**

  if $enabled(t, \mathrm{M}_0)$ **then** $L_{fireable}[\pi(t)] \leftarrow L_{fireable}[\pi(t)] \cup \{t\}$

Prioritásos algoritmus: tüzelési ciklus

```
while ⋃_{π∈Π} L_fireable[π] ≠ ∅  do
    for π = π_max to π_min step -1  do  // Tüzelés
        if L_fireable[π] ≠ ∅ then
            t ← rnd(L_fireable[π])
            M' ← M + W^T·e_t
            exit for
        end if

    for all π ∈ Π do  // Engedélyezett tranzíciók
        for all t' ∈ {(•t)•} do
            if not(enabled(t', M')) then L_fireable[π(t')] ← L_fireable[π(t')] \ {t'}
        for all t'' ∈ {(t•)•} do
            if enabled(t'', M') then L_fireable[π(t'')] ← L_fireable[π(t'')] ∪ {t''}
    end for

    M ← M'
end while
```

$$\text{while } \bigcup_{\pi\in\Pi} L_{fireable}[\pi] \neq \emptyset \text{ do}$$

Elérhetőségi analízis

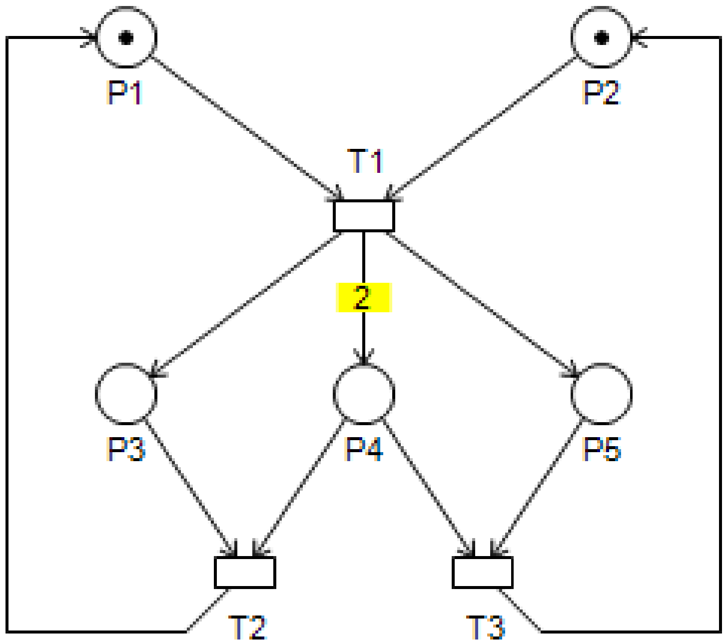
Elérhetőség

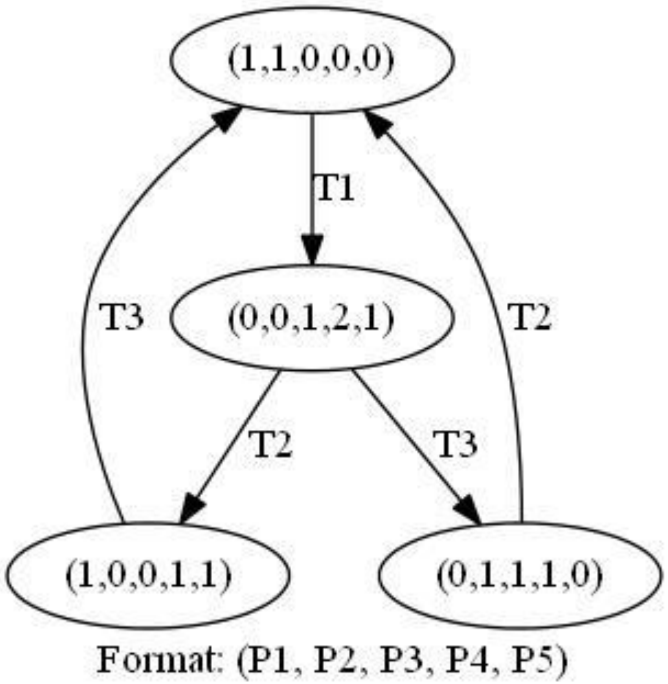
- Elérhetőségi analízis
  - Kezdőállapotfüggő dinamikus viselkedés
    - Jelölés (marking) = állapot
    - Tokeneloszlás = állapotváltozó (értéke)
    - Tüzelés = állapotátmenet
    - Tüzelési sorozatok hatására $M_0, M_1, \ldots, M_n$ állapotsorozat
  - Állapotsorozat: trajektória az állapottérben
  - M_n állapot *elérhető* az M_0 kiinduló állapotból, ha

$$\boxed{\exists \vec{\sigma} : M_0 [\vec{\sigma} > M_n}$$

  - Elérhetőségi gráf: állapottér grafikus képe

Példa: Elérhetőségi gráf





Egyszerű Petri-háló és elérhetőségi gráfja
(a PetriDotNet eszközből)

Elérhetőségi analízis

- Az M_0 kiinduló állapotból az N Petri hálóban
  - Elérhető állapotok

$$\boxed{R(N,M_0)=\left\{M \mid \exists\vec{\sigma}: M_0[\vec{\sigma}>M\right\}}$$

Állapot alapú kérdésekre lehet ez alapján válaszolni

  - Végrehajtható tüzelési sorozatok

$$\boxed{L(N,M_0)=\left\{\vec{\sigma} \mid \exists M: M_0[\vec{\sigma}>M\right\}}$$

Állapotátmenet (esemény) alapú kérdésekre lehet válaszolni

Elérhetőségi probléma

- Petri hálók elérhetőségi problémája:
  - M_n állapot elérhető-e valamilyen M_0 kiinduló állapotból

$$\boxed{M_n \overset{?}{\in} R(N, M_0)}$$

- Rész-tokeneloszlási probléma:
  - Helyek egy $P' \subset P$ részhalmazára korlátozva a kérdést, elérhető-e M_n állapot az adott helyekre megadott tokeneloszlással

$$\boxed{\overset{?}{\exists} M \in R(N, M_0) : \forall p \in P' : M(p) = M_n(p)}$$

Elérhetőségi probléma megoldhatósága

- Az elérhetőségi probléma eldönthető
  - De exponenciális (hely) komplexitású általános esetben

- Míg az egyenlőségi probléma eldönthetősége általános esetben bizonyítottan nem lehetséges
  - Feladat: két Petri háló (N, N’) lehetséges tüzelési sorozatai azonosságának eldöntése

$$\boxed{L(N, M_0) \stackrel{?}{=} L(N', M_0')}$$

  - 1-korlátos (biztos) Petri hálók esetén exponenciális
    - Biszimuláció: egymást szimulálni képesek

Petri hálók dinamikus (viselkedési) tulajdonságai

Dinamikus tulajdonságok

- Elérhetőséggel kapcsolatos tulajdonságok
  - Függenek a kiinduló állapottól (kezdő jelöléstől)

    (Ld.: kiinduló állapottól függetlenek a strukturális tulajdonságok!)
  - Nem csak elérhetőségi analízissel határozhatók meg
- Dinamikus tulajdonságok (áttekintés):

1. Korlátosság
2. Élőség
   - Holtpontmentesség
3. Megfordíthatóság
4. Visszatérő állapot
5. Fedhetőség
6. Perzisztencia
7. Fair tulajdonság
   - Korlátozott fairség
   - Globális fairség

1. Korlátosság

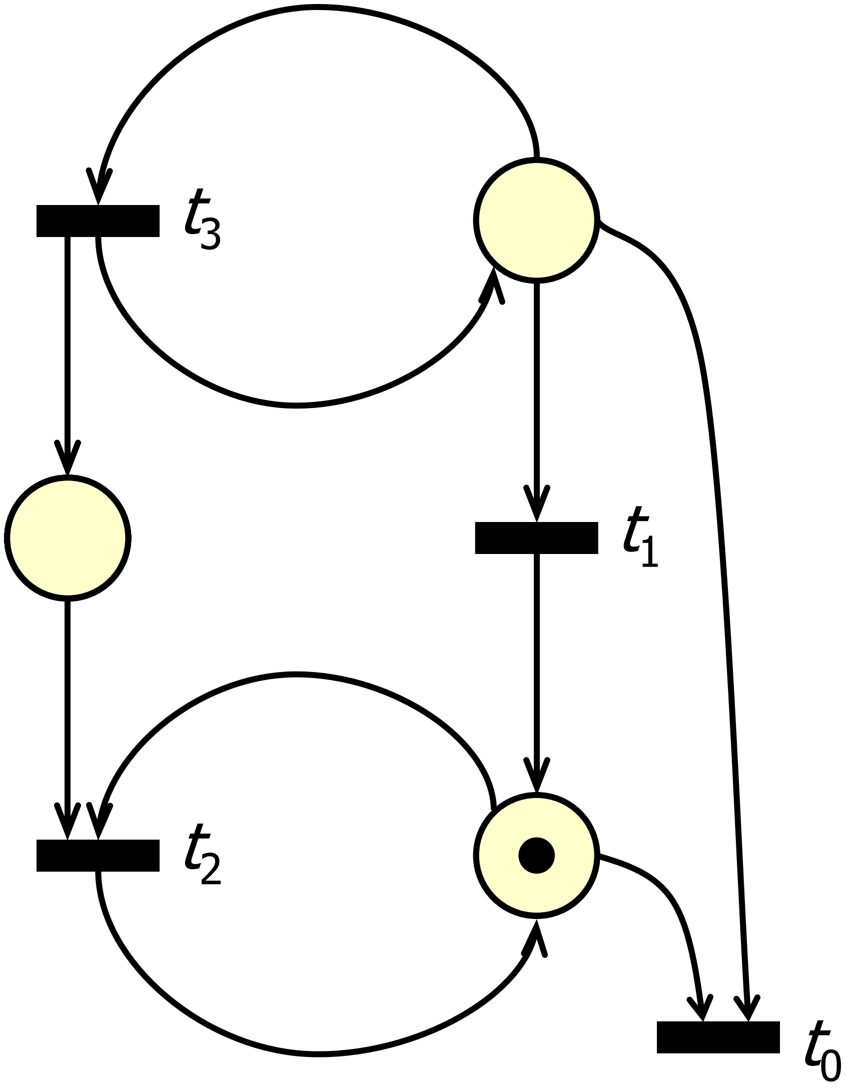
- k-korlátosság (korlátosság)
  - Bármely állapotban minden helyen helyenként maximum k token lehet (M_0 kiinduló állapot függő!)
  - Biztos Petri háló: korlátosság speciális esete ($k = 1$)
  - „Végesség” kifejezése
    - Korlátosság ⇔ véges állapottér
- Kapcsolódó gyakorlati kérdések
  - A rendszerben felgyűlnek-e a feladatok?
  - Megvalósul-e az üzenetek rendszeres feldolgozása?

2. Élőség

- Háló holtpont (deadlock)-mentessége
  - Minden állapotban legalább egy tranzíció tüzelhető
- Élő tulajdonság: Általánosabb ennél
  - Tranzíció egyszer/többször/végtelenszer tüzelhet-e?
  - Gyenge élő tulajdonságok egy t tranzícióra:
    - L_0-élő (halott): t sohasem tüzelhető egyetlen
    - L_1-élő: t legalább egyszer tüzelhető valamely
    - L_2-élő: bármely véges $k>1$ egészre t legalább k–szor tüzelhető valamely
    - L_3-élő: t végtelen sokszor tüzelhető valamely

    $\vec{\sigma} \in L(N, M_0)$ állapot-trajektóriában
  - L_4-élő: t L_1-élő bármely $M_n \in R(N, M_0)$ állapotból

Élő tulajdonság: példa



- t_0 tranzíció: L_0-élő (halott)
- t_1 tranzíció: L_1-élő
- t_2 tranzíció: L_2-élő
- t_3 tranzíció: L_3-élő

Az élőség Petri hálókra

- Egy (P, T, M_o) Petri háló L_x-élő
  - Ha minden $t \in T$ tranzíció L_x-élő
  - L_4-től L_1-ig az élő tulajdonságok tartalmazzák egymást
- Egy (P, T, M_o) Petri háló élő
  - Ha L_4-élő, azaz minden $t \in T$ tranzíció L_4-élő
    - L_4-élő: L_1-élő (azaz legalább egyszer tüzelhető valamely trajektória mentén) bármely elérhető állapotból
  - Bejárási úttól függetlenül garantáltan holtpontmentes
    - Köztes állapottól függetlenül minden tranzíció újra tüzelhető
    - Holtpontmentesség $\Leftarrow$ élőség
  - Bizonyítása költséges lehet
    - Szerencsés esetben nem (ld. majd invariánsok!)

3. Megfordíthatóság

- Megfordíthatóság
  - A kezdőállapot bármely követő állapotból elérhető

$$\boxed{\forall M \in R(N, M_0) : M_0 \in R(N, M)}$$

- Gyakorlati példák:
  - Ciklikus működésű hálózat, a kezdőállapoton keresztül
  - „Reset” jellel a kezdőállapotba vihető rendszer
  - Biztonságos kezdőállapot mindenhonnan elérhető

4. Visszatérő állapot

- Visszatérő állapot
  - Van olyan, a kezdőállapotból elérhető állapot, amely bármely őt követő állapotból elérhető

$$\boxed{\exists M_n \in R(N, M_0) : \forall M \in R(N, M_n) : M_n \in R(N, M)}$$

- Gyakorlati példák:
  - Inicializáló szekvencia után ciklikus működés
  - Inicializálás után bárhonnan elérhető biztonságos állapot

5. Fedhetőség

- Fedhetőség
  - Létrejön-e korábbi működést magában foglaló állapot?
  - M′ állapot fedi M állapotot, ha $M' \in R(N, M_0) \wedge M' \geq M$
    - Fordított megfogalmazás: *M* állapot fedhető *M′* állapottal
    - *M′* ≥ *M* jelentése: $\forall p \in P : \mathrm{m}'(p) \geq \mathrm{m}(p)$
      - Gyenge fedhetőség esetén az azonos állapot is fed, ha elérhető
      - Erős fedhetőség: $\exists p \in P : \mathrm{m}''(p) > \mathrm{m}(p)$
- Kapcsolat az élőséggel
  - Ha μ a *t* tranzíciót engedélyező minimális tokeneloszlás
    - *t* akkor és csak akkor nem L_1-élő, ha μ nem fedhető le
    - fordítva: μ lefedhetősége garantálja *t* L_1-élő voltát

6. Perzisztencia

- Perzisztencia tranzíciókra
  - Egy tranzíció perzisztens, ha engedélyezetté válva engedélyezve is marad tüzelésig
  - Azaz nincs olyan engedélyezett tranzíció, amelynek tüzelése letiltja egy másik tranzíció engedélyezettségét
- Perzisztencia Petri-hálókra
  - Egy (P, T, M_0) Petri háló perzisztens, ha bármely két $t_1, t_2 \in T$ tranzíciója az összes lehetséges tüzelési szekvenciában perzisztens
- Gyakorlati példák:
  - Rendszerbeli funkcionális dekompozíció megmarad-e?
  - Párhuzamos működések befolyásolják-e egymást?

7. Fair tulajdonság: korlátozott fairség

- Kétféle fairség definíció
  - Korlátozott fairség (B-fairség)
  - Globális fairség (korlátlan fairség)

- Korlátozott fairség
  - Egy tüzelési szekvencia korlátozottan fair (B-fair)
    - ha bármely tranzíció maximum korlátos sokszor tüzelhet anélkül, hogy egy másik tranzíció tüzelne
  - Egy Petri háló korlátozottan fair (B-fair)
    - ha az összes lehetséges tüzelési szekvenciája korlátozottan fair

Fair tulajdonság: globális fairség

- Globális fairség
  - Egy tüzelési szekvencia globálisan (korlátlanul) fair, ha
    - véges, vagy
    - az összes tranzíció végtelen sokszor szerepel benne
  - Egy Petri háló globálisan (korlátlanul) fair
    - Ha a háló összes lehetséges tüzelési szekvenciája globálisan (korlátlanul) fair
- Gyakorlati példák:
  - Párhuzamos folyamatok nem tartják-e fel egymást?
  - Valamennyi folyamat végbemegy-e (előbb-utóbb)?
  - Kérés kiszolgálása előbb-utóbb megtörténik-e?

Dinamikus tulajdonságok (összefoglalás)

- Korlátosság
- Holtpontmentesség
- Élő tulajdonság
  - L0 élő (halott)
  - L1 élő (1-szer tüzelhető)
  - L2 élő (k-szor tüzelhető)
  - L3 élő (∞-szer tüzelhető)
  - L4 élő (∀ állapotban L1)
- Megfordíthatóság
- Visszatérő állapot
- Fedhetőség
  - Gyenge fedhetőség
  - Erős fedhetőség
- Perzisztencia
- Fair tulajdonság
  - Korlátozott fairség
  - Globális fairség

Állapottér reprezentációk:
az elérhetőségi és fedési gráf

Állapottér reprezentációk: Elérhetőségi gráf

- Elérhetőségi gráf
  - M_0 kezdőállapotból induló állapotgráf
    - Csomópontok: állapotok;   címkézés: tokeneloszlások
    - Állapotátmenetek: irányított élek;   címkézés: tüzelések
    - Egy csomópont esetén legfeljebb annyi rákövetkező csomópont (kimenő él), ahány engedélyezett tranzíció
      - Kevesebb, ha prioritásos a Petri háló
    - Csomópont, amiből nem indul ki él: holtpont
  - Nem korlátos a Petri háló $\rightarrow$ végtelen sok állapot
    - Korlátosság $\Leftrightarrow$ véges állapottér
  - Vizsgálat: Szélességi típusú bejárás az állapotból a tüzelések mentén
    - Mélységi bejárás nem korlátos állapottérben rossz ötlet…

Állapottér reprezentációk: Fedési gráf

- Végtelen állapotgráf: token „túlszaporodás”
  - Hol, „milyen módon” lesz végtelen?
  - Milyen analízisre ad lehetőséget?
- Fedési gráf: végtelen állapottér esetére is
  - Hasonló felépítés: M_0 kezdőállapot, élek: tüzelések
  - Trajektória: $M_0 \ldots M'' \ldots M'$
    ha itt $M'' \leq M'$ $\rightarrow$ fedett állapotok
  - $p \in P : \mathrm{m}'(p) > \mathrm{m}''(p)$ $\rightarrow$ fedett helyek (erős fedhetőség)
  - Fedett helyekre speciális szimbólum:
    ω a végtelenség kifejezője

Fedési fa generáló algoritmus

$\mathrm{L}_{\text{vizsgálandó}} \leftarrow \{\ M_0\ \}$

`MAIN:` **`if`** $\mathrm{L}_{\text{vizsgálandó}} \neq \emptyset$

A következő $M \in \mathrm{L}_{\text{vizsgálandó}}$ állapot kivétele

`if` *M* a gyökértől idáig vezető úton már szerepelt

`then` *M* -et „régi állapotként" jelöljük

`goto` `MAIN` // ciklus

`if` *M*-ben nincs engedélyezett tranzíció

`then` *M* -et „végállapotként" (halott állapot) jelöljük

`goto` `MAIN` // ciklus

(folytatás a következő lapon)

Fedési fa generáló algoritmus (folyt.)

else // (van M -ben engedélyezett tranzíció)

for all *t* engedélyezett tranzícióra:

Az *M′* rákövetkező állapot meghatározása: $M\left[\vec{e}_t > M'\right.$

if létezik az M_0-tól *M* -ig vezető úton olyan *M″*, amelyet *M′* fed

$$\boxed{M' \neq M'' \wedge \forall p \in P : \mathrm{m}'(p) \geq \mathrm{m}''(p) \wedge \exists p \in P : \mathrm{m}'(p) > \mathrm{m}''(p)}$$

then *M″* fedett állapot:

az *M′* állapotot jelölő tokeneloszlásban
a fedett helyek jelöléseit ω-val helyettesítjük

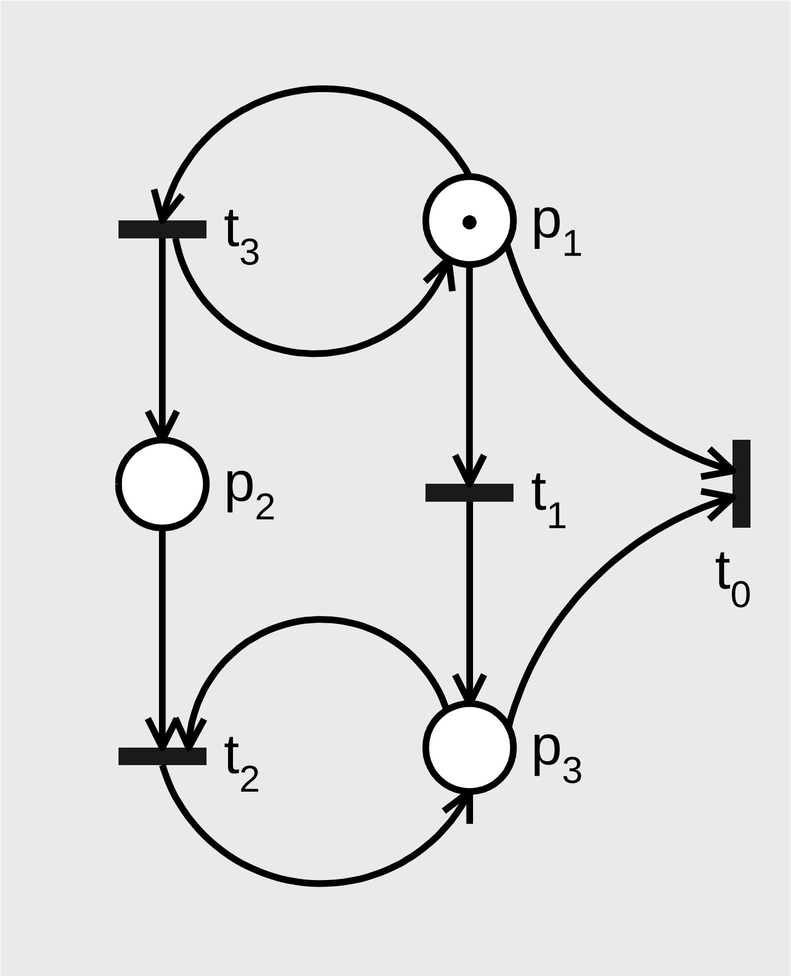
$$\boxed{\forall p \in P : \mathrm{m}'(p) > \mathrm{m}''(p) \rightarrow \mathrm{m}'(p) = \omega}$$

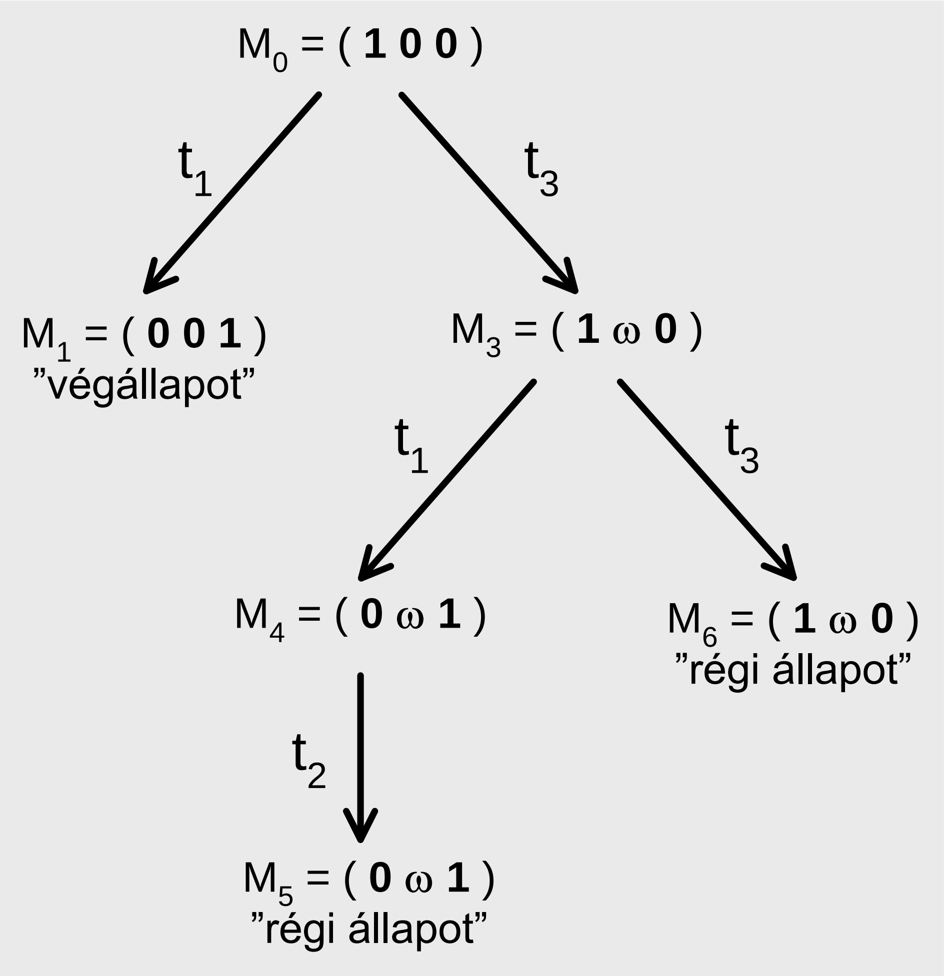
else *M′* új állapot: $L_{vizsgálandó} \leftarrow L_{vizsgálandó} \cup M'$

M -ből *M′*-höz egy *t* -vel jelölt élet húzunk

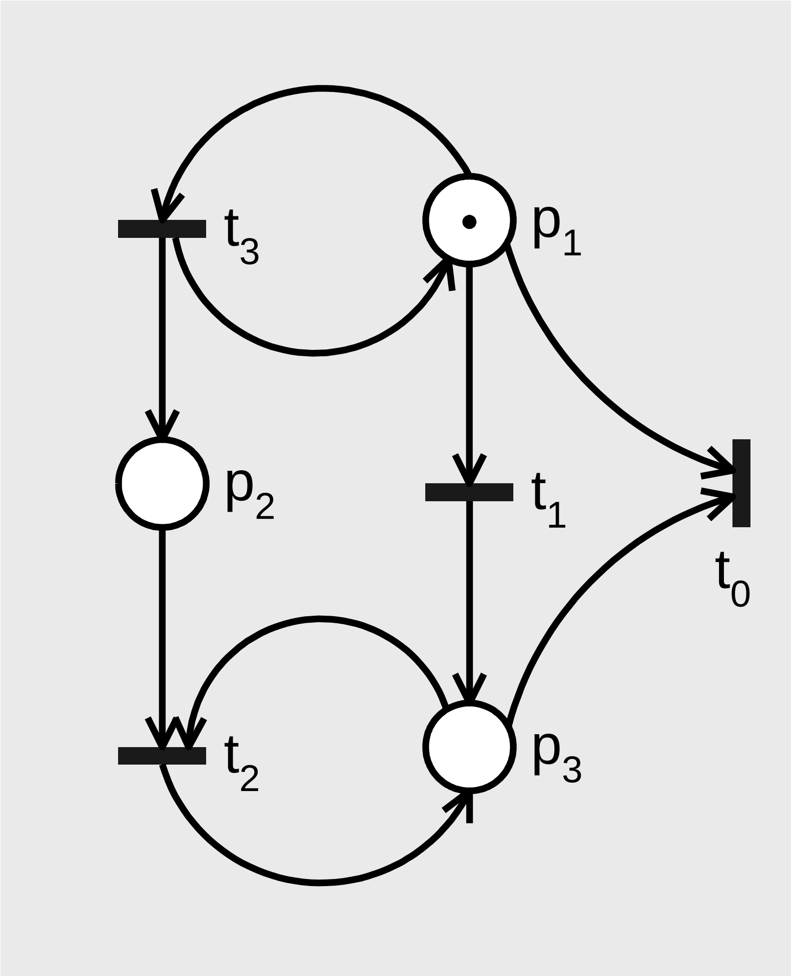
goto `MAIN` // ciklus

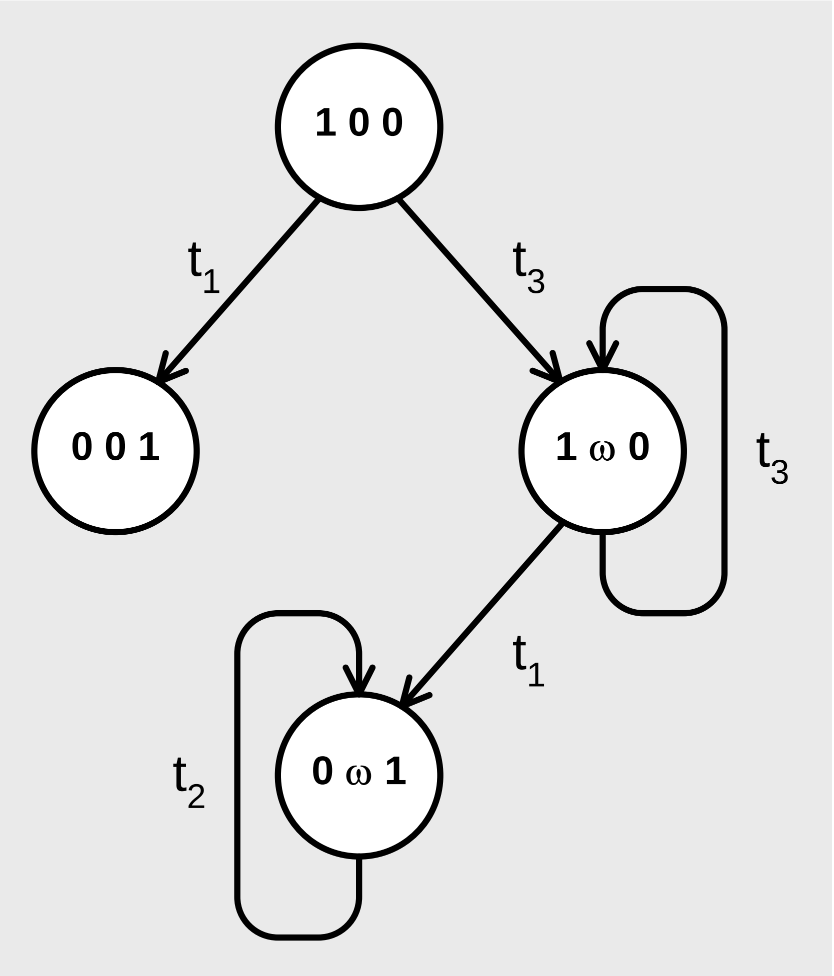
Egy példa és annak fedési fája





Egy példa és annak fedési gráfja





Petri hálók fedési fájának analízise

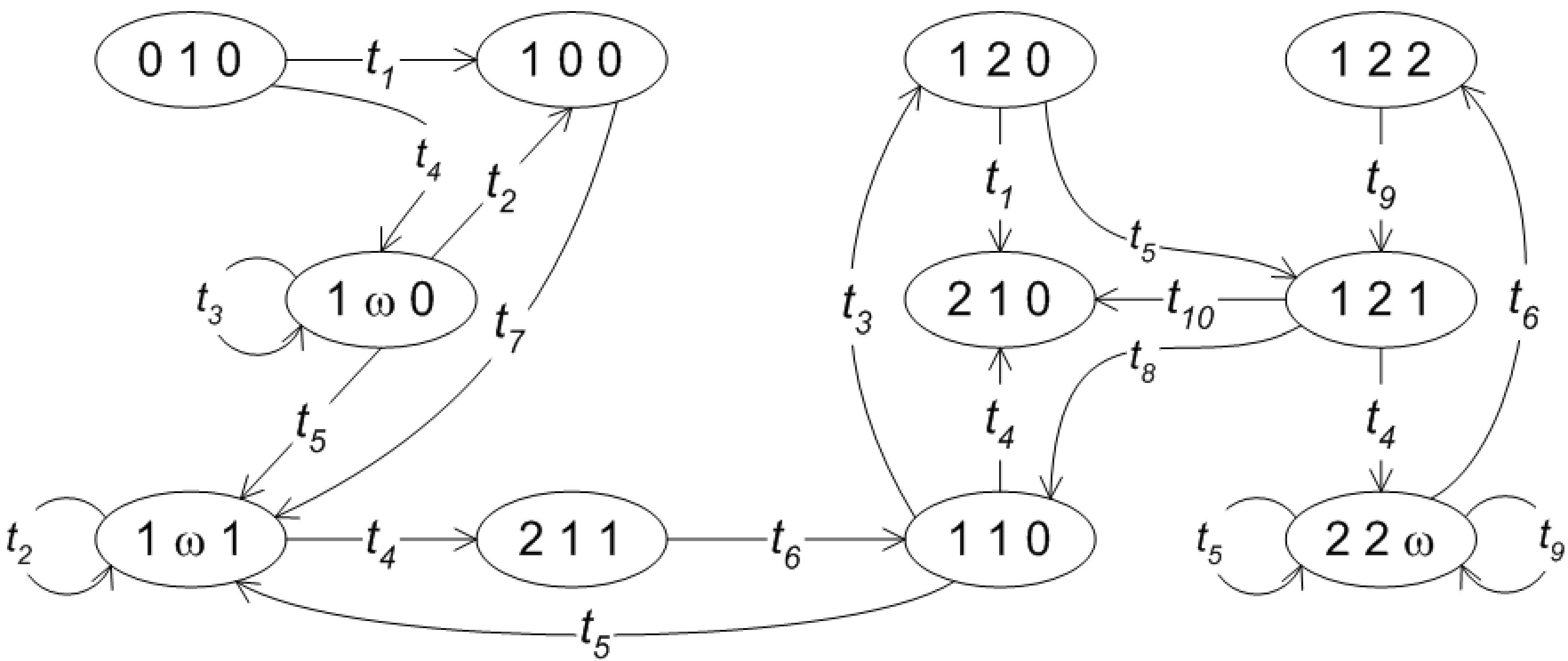
Közvetlenül is leolvasható tulajdonságok:

- Petri háló korlátos ⇔ $R(N, M_0)$ elérhetőségi gráfja véges ⇔ Fedési fában ω nem jelenik meg címkeként
- Petri háló biztonságos ⇔ Csak 0 és 1 jelenik meg csomópont címkeként a fedési fában
- Petri háló egy tranzíciója halott ⇔ tranzícióhoz tartozó tüzelés nem jelenik meg élcímkeként a fedési fában

Dinamikus tulajdonságok vizsgálata az állapotérben

Tipikus feladat

Az ábra egy Petri háló állapotterét mutatja be fedési gráf alakban. A hálóban 10 darab tranzíció található, amelyeket t_1, ..., t_{10} címkékkel jelölünk. Az állapotokat a token eloszlás vektorral címkéztük meg, tehát 0 1 0 jelentése: $m(p_1) = 0$, $m(p_2) = 1$ és $m(p_3) = 0$.



Tipikus kérdések

1. A Petri háló élő?
2. A háló (deadlock) holtpontmentes?
3. t_6 tranzíció L_3-élő?
4. t_7 tranzíció L_2-élő?
5. A (2 2 1) állapot fedhető?
6. A (2 1 0) állapot fedhető?
7. A háló perzisztens?
8. A háló korlátos?
9. A háló megfordítható?
10. A hálóban létezik visszatérő állapot?
11. t_4 és t_6 tranzíció korlátos fair tulajdonságú?
12. t_5 és t_8 tranzíció korlátos fair tulajdonságú?
13. Létezik P-invariáns?
14. Létezik T-invariáns?

Élőség vizsgálata az állapottérben

- Ellenpéldát találni (szinte mindig) a leggyorsabb és legegyszerűbb megoldás!
- Élőség:
  - L_4-élőség
    - Végig kell nézni, hogy mindig tüzelhetővé válik-e?
    - Minden trajektóriára teljesülnie kell!
  - L_3-élőség (és a többi)
    - Elég egy trajektóriát találni, ahol teljesül!
  - A háló akkor élő, ha minden tranzíciója élő!
    - Ha találunk akár egy tranzíció esetében ellenpéldát $\rightarrow$ nem élő
  - Ha holtpontmentes, akkor még nem biztos, hogy élő is!

Korlátosság vizsgálata az állapottérben

- Korlátosság:
  - Lásd a fedési gráfnál tanultakat!
    - „Petri háló korlátos $\Leftrightarrow$ $R(N, M_0)$ elérhetőségi gráfja véges $\Leftrightarrow$ fedési fában ω nem jelenik meg állapot címkében”
  - Biztosság:
    - „Petri háló biztos $\Leftrightarrow$ csak 0 és 1 jelenik meg állapot címkében a fedési fában”

További dinamikus tulajdonságok

- Megfordíthatóság:
  - Az elérhetőségi gráf egyetlen erősen összekötött komponens?
- Visszatérő állapot:
  - Van az elérhetőségi gráfban erősen összekötött komponens?
  - Az adott állapot része ennek?
- Fairség:
  - „Az egyik tranzíció korlátos sokszor tüzelhet, mielőtt a másik tüzelne”
  - Van-e olyan ciklus, amiben az egyik tranzíció benne van és a másik nincs?
    - $\rightarrow$Van: ellenpélda, hiszen létezik tüzelési szekvencia, amiben korlátlan sokszor tüzel
- Perzisztencia:
  - „A tranzíció mindaddig engedélyezett marad, amíg nem tüzel”
    - $\rightarrow$Ha több engedélyezett és nem ő tüzelt, akkor a következő állapotban is engedélyezett marad
    - $\rightarrow$Ha engedélyezett maradt, akkor meg is jelenik élcímkeként (prioritás?)

Dinamikus tulajdonságok analízis eszközökben

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