Elérhetőségi analízis
Petri hálók dinamikus tulajdonságai

dr. Bartha Tamás

Dr. Pataricza András

BME Méréstechnika és Információs Rendszerek Tanszék

Petri hálók vizsgálata

Vizsgálati lehetőségek

Az elemzés mélysége szerint:

- Szimuláció

  egy trajektória bejárása

- Állapottér bejárása

  minden trajektória bejárása (kimerítő bejárás)

  - elérhetőségi gráf analízis
  - dinamikus (viselkedési) tulajdonságok

  kezdőállapottól független (bármely kezdőállapotra)

- Strukturális tulajdonságok
  - 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

ha az állapottér nem kezelhető

- 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
    - vagy univerzális (minden működésre érvényes)
    - vagy egzisztenciális (lehetséges ilyen működés)
  - **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

Elérhetőség fogalma, tulajdonságai
Elérhetőségi analízis

Elérhetőség

Elérhetőségi analízis

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

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

- Elérhetőségi gráf: állapottér grafikus képe

Elérhetőségi analízis

Az M_0 kiinduló állapotból az N Petri hálóban

– Elérhető állapotok

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

  • Állapot bázisú kérdések

– Végrehajtható tüzelési sorozatok

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

  • Állapotátmenet (esemény) bázisú kérdések

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 elérhető-e M_n állapot az adott helyekre megadott tokeneloszlással

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

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
    - processz algebra, biszimuláció

Petri hálók dinamikus (viselkedési) tulajdonságai

Dinamikus tulajdonságok

- Elérhetőséggel kapcsolatos tulajdonságok
  - Függenek a kiinduló állapottól (kezdő jelöléstől)
    - ⇒ **strukturális** tulajdonságok: kiinduló állapottól függetlenek
  - Nem csak elérhetőségi analízissel határozhatók meg
- Dinamikus tulajdonságok:
  - Korlátosság
  - Élő tulajdonság
    - Holtpontmentesség
  - Megfordíthatóság
  - Visszatérő állapot
  - Fedhetőség
  - Perzisztencia
  - Fair tulajdonság
    - Korlátozott fairség
    - Globális fairség

Dinamikus tulajdonságok: végesség

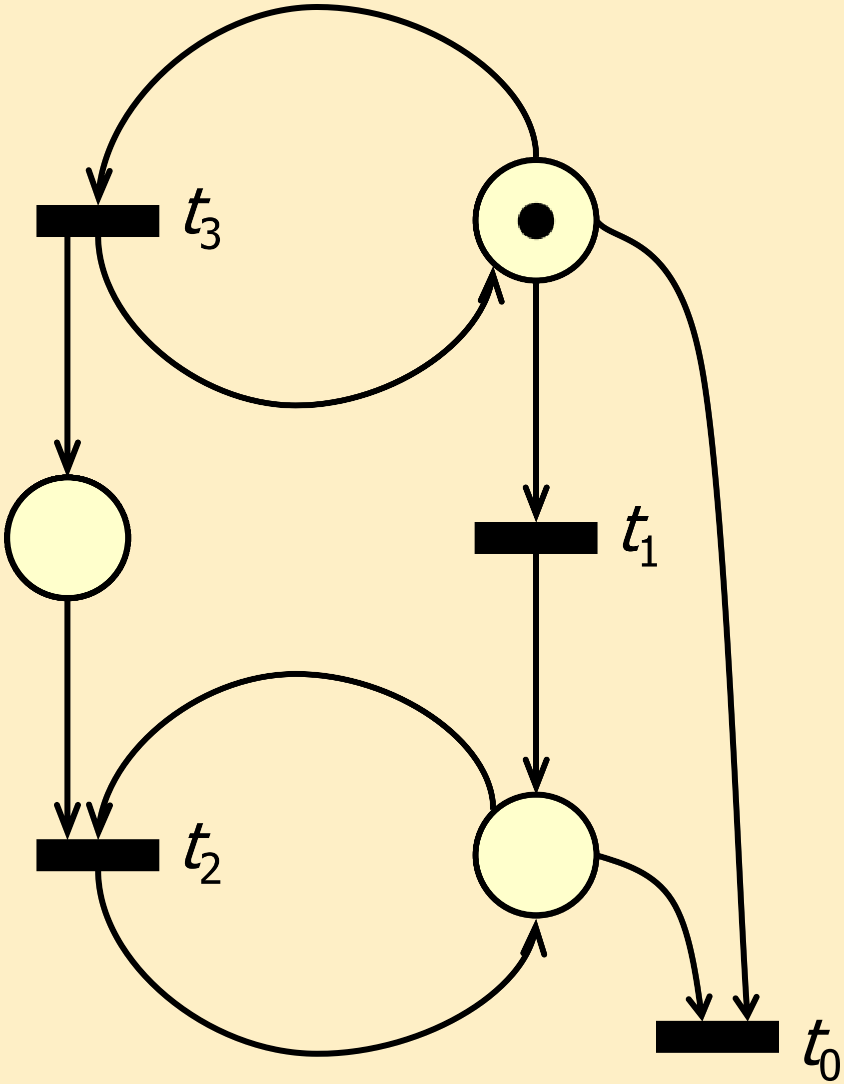
- *k*-korlátosság (korlátosság)
  - bármely állapotban minden helyen helyenként maximum *k* token lehet (M_0 kiinduló állapot függő!)
    - Biztos Petri háló: korlátosság speciális esete ($k = 1$)
  - végesség kifejezése
    - korlátosság $\Leftrightarrow$ véges állapottér
  - erőforrás használat, illetve általános értelemben vett feladatkezelés modellezése
    - (részben) konzervativitás $\overset{?}{\Rightarrow}$ korlátosság
  - a rendszerben a feladatok elvégzése garantált-e?

Dinamikus tulajdonságok: élőség

- Holtpont (deadlock) -mentesség
  - Minden állapotban legalább egy tranzíció tüzelhető
- Élő tulajdonság
  - 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 $M_n \in R(N, M_0)$ állapotban

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

Dinamikus tulajdonságok: élőség (folyt.)

- 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
  - Bejárási úttól függetlenül garantáltan holtpontmentes
    - köztes állapottól függetlenül minden tranzíció újra tüzelhető
    - holtpontmentesség $\Leftarrow$ élőség
  - Bizonyítása költséges lehet
    - szerencsés esetben nem (invariánsok!)
    - ideális rendszert tételez fel

Dinamikus tulajdonságok: ciklikusság

- Megfordíthatóság
  - A kezdőállapot bármely követő állapotból elérhető

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

  - Gyakran ciklikus működésű hálózat
- Visszatérő állapot
  - Van olyan, a kezdőállapotból elérhető állapot, amely bármely őt követő állapotból elérhető

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

  - Gyakran ciklikus (rész)hálózat inicializáló szekvenciával

Dinamikus tulajdonságok: ciklikusság (folyt.)

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

Dinamikus tulajdonságok: kölcsönhatás

- Perzisztencia
  - Két tetszőleges engedélyezett tranzíció közül egyik tüzelése sem tiltja le a másik engedélyezettségét
    - engedélyezett tranzíció engedélyezve marad tüzelésig!
  - Rendszerbeli funkcionális dekompozíció megmarad-e?
  - Párhuzamos működések befolyásolják-e egymást?
- Egy (P, T, M_0) Petri háló perzisztens, ha
  - bármely két $t_1, t_2 \in T$ tranzíciója az összes lehetséges tüzelési szekvenciában perzisztens

Dinamikus tulajdonságok: kölcsönhatás (folyt.)

- Fair tulajdonság
  - Párhuzamos folyamatok nem tartják-e fel egymást?
  - Valamennyi folyamat végbemegy-e (előbb-utóbb)?
- Nem egységes a fairség definíciója
  - Korlátozott fairség (B-fairség)
  - Globális fairség (korlátlan fairség)
- Egy tüzelési szekvencia korlátozottan fair (B-fair)
  - ha bármely tranzíció maximum korlátos sokszor tüzelhet anélkül, hogy egy másik tranzíció tüzelne
- Egy (P, T, M_0) Petri háló korlátozottan fair (B-fair)
  - ha minden $t \in T$ tranzíció az összes lehetséges tüzelési szekvenciában korlátozottan fair (B-fair)

Dinamikus tulajdonságok: kölcsönhatás (folyt.)

- Globális fairség
  - Egy tüzelési szekvencia globálisan fair, ha
    - véges, vagy
    - az összes tranzíció végtelen sokszor szerepel benne
  - Egy (P, T, M_0) Petri háló globálisan (korlátlanul) fair
    - ha a háló összes lehetséges tüzelési szekvenciája globálisan (korlátlanul) fair

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
    - legfeljebb annyi új csomópont, ahány engedélyezett tranzíció
      - kevesebb, ha prioritásos a Petri háló
    - csomópont, amiből nem indul ki él: holtpont
  - Nem korlátos a Petri háló → végtelen sok állapot
    - korlátosság ⇔ véges állapottér
  - Szélességi típusú bejárás az állapotból tüzelések mentén
    - mélységi bejárás nem korlátos állapottérben rossz ötlet…
    - viszont a részleges sorrendezési redukció mélységi kereséssel

Állapottér reprezentációk: fedési gráf

- Petri háló mérete/bonyolultsága és az állapottér mérete közti korreláció?
  - pl. megállási probléma
- Végesség hiányát kezelni kell
- Fedési gráf: végtelen állapottér esetére is
  - Hasonló felépítés: M_0 kezdőállapot, élek: tüzelések
  - Kritikus részek: token „túlszaporodás”
    - trajektória: $M_0 \ldots M'' \ldots M, M'$ és $M'' \leq M'$ $\rightarrow$ 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

$L_{\text{vizsgálandó}} \leftarrow \{ M_0 \}$

`MAIN:` **`if`** $L_{\text{vizsgálandó}} \neq \varnothing$

A következő $M \in L_{\text{vizsgálandó}}$ állapot kiválasztása

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

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

`goto` `MAIN` // ciklus

`if` *M*-ben nincs engedélyezett tüzelés

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

`goto` `MAIN` // ciklus

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

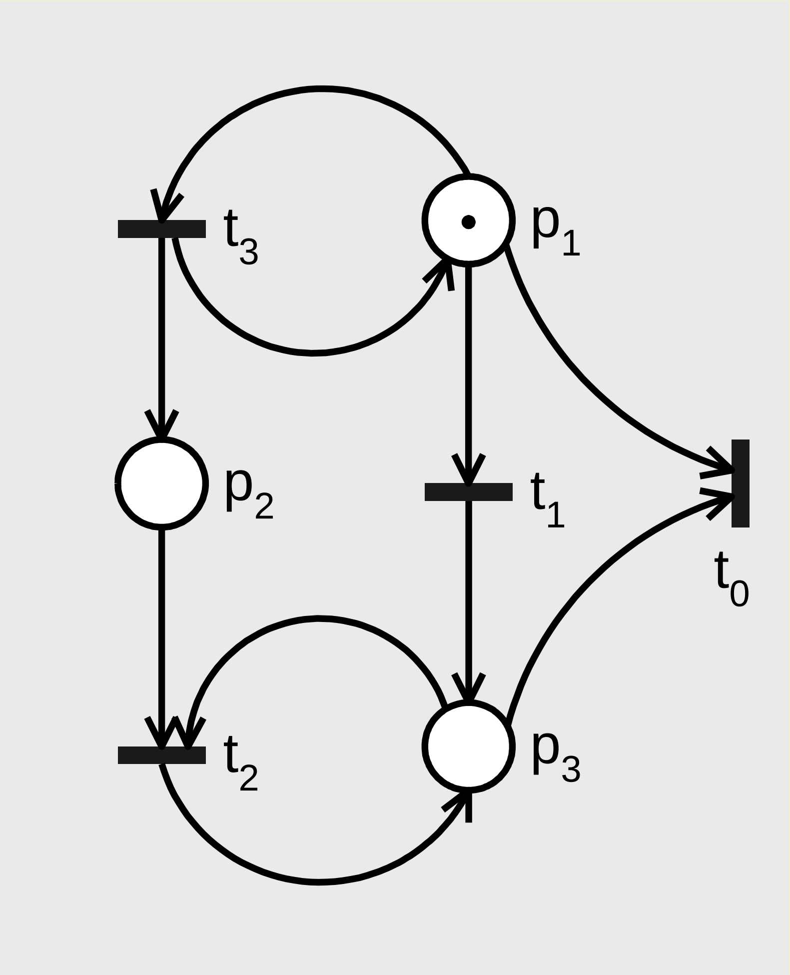
$$\boxed{\forall p \in P : \mathrm{m}'(p) > \mathrm{m}''(p) \rightarrow \mathrm{m}'(p) = \omega}$$

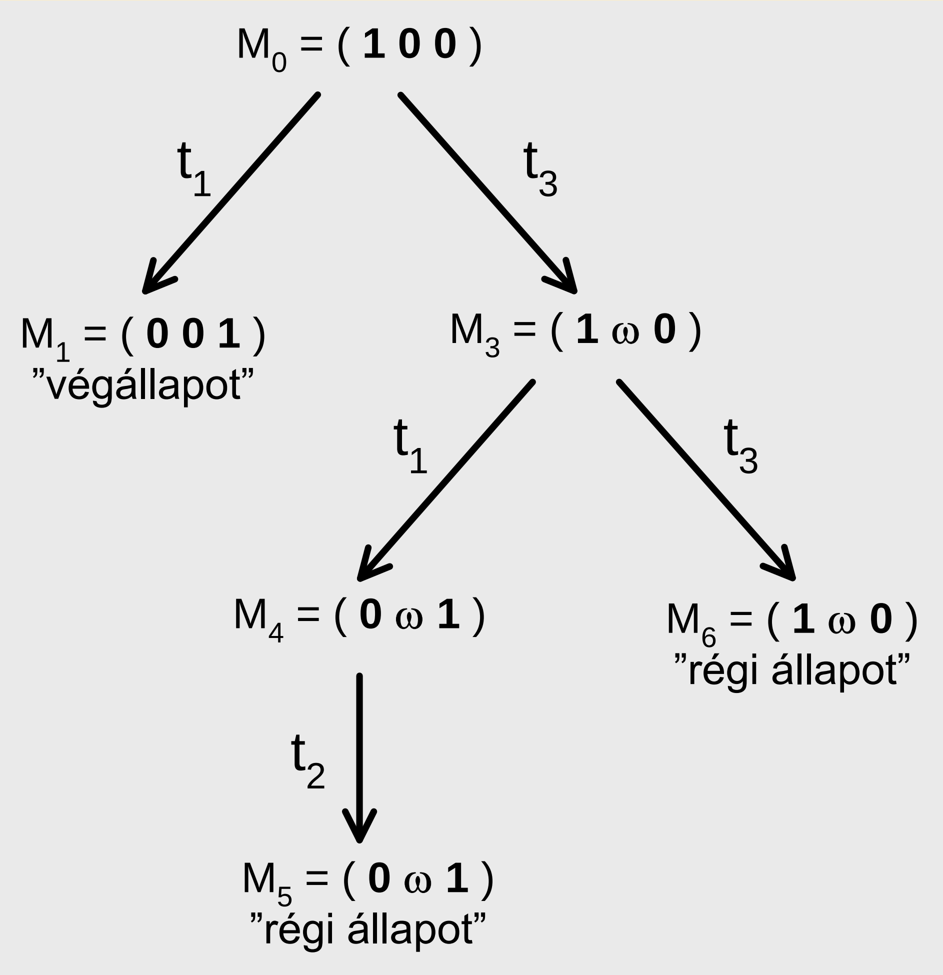
else *M′* új állapot: $L_{vizsgálandó} \leftarrow L_{vizsgálandó} \cup M'$

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

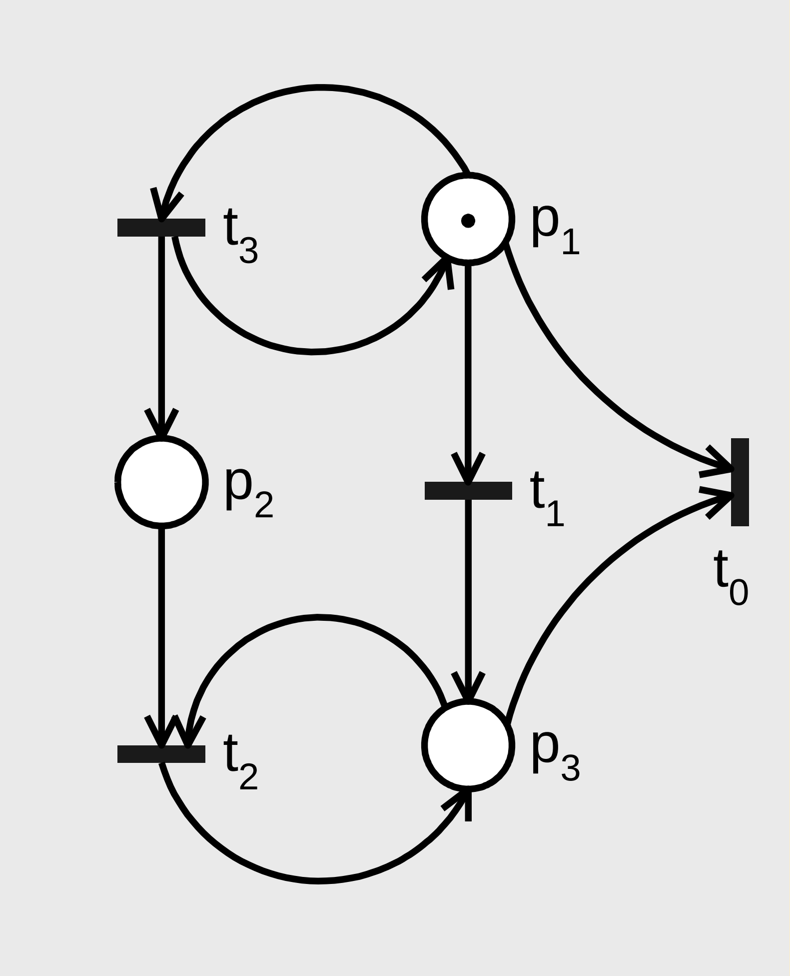
goto `MAIN` // ciklus

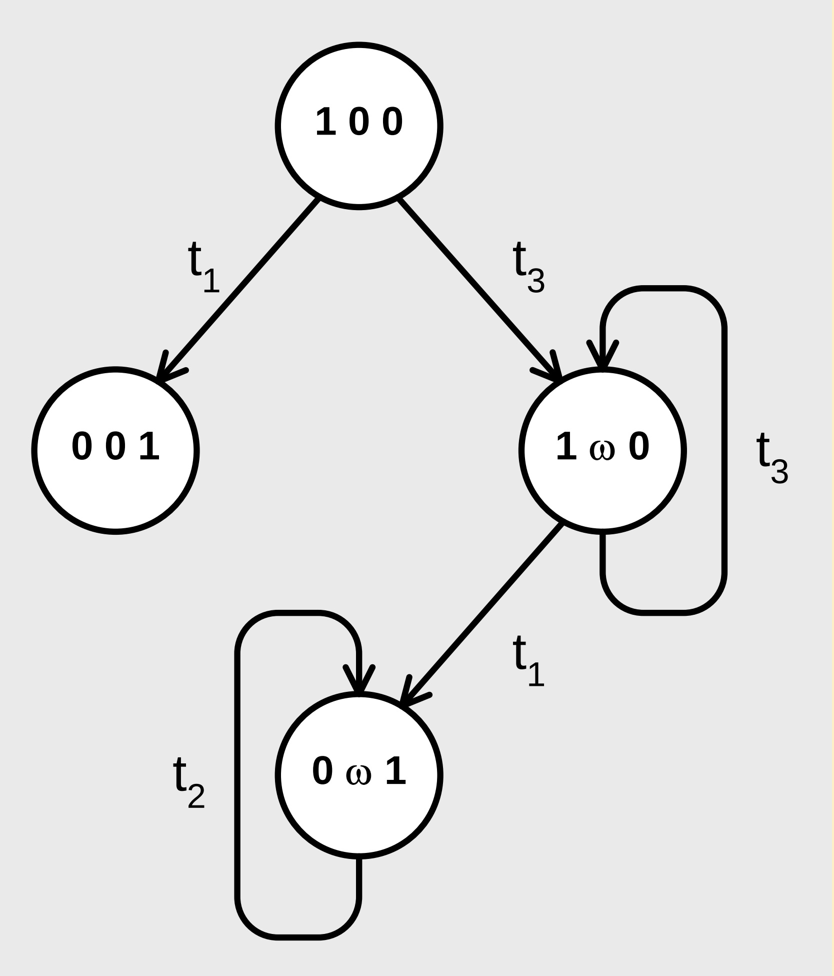
A példa és annak fedési fája





A példa és annak fedési gráfja





Petri hálók fedési fájának analízise

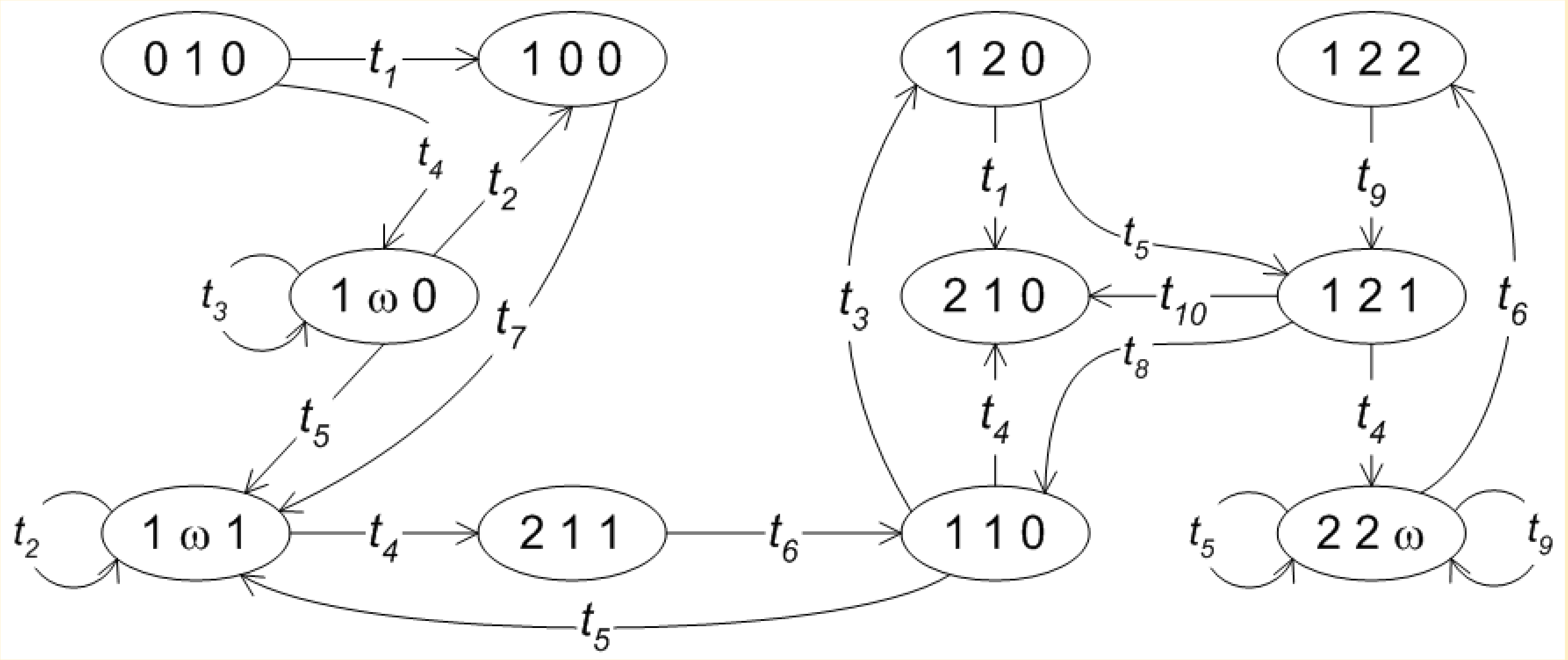
Közvetlenül is leolvasható tulajdonságok:

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

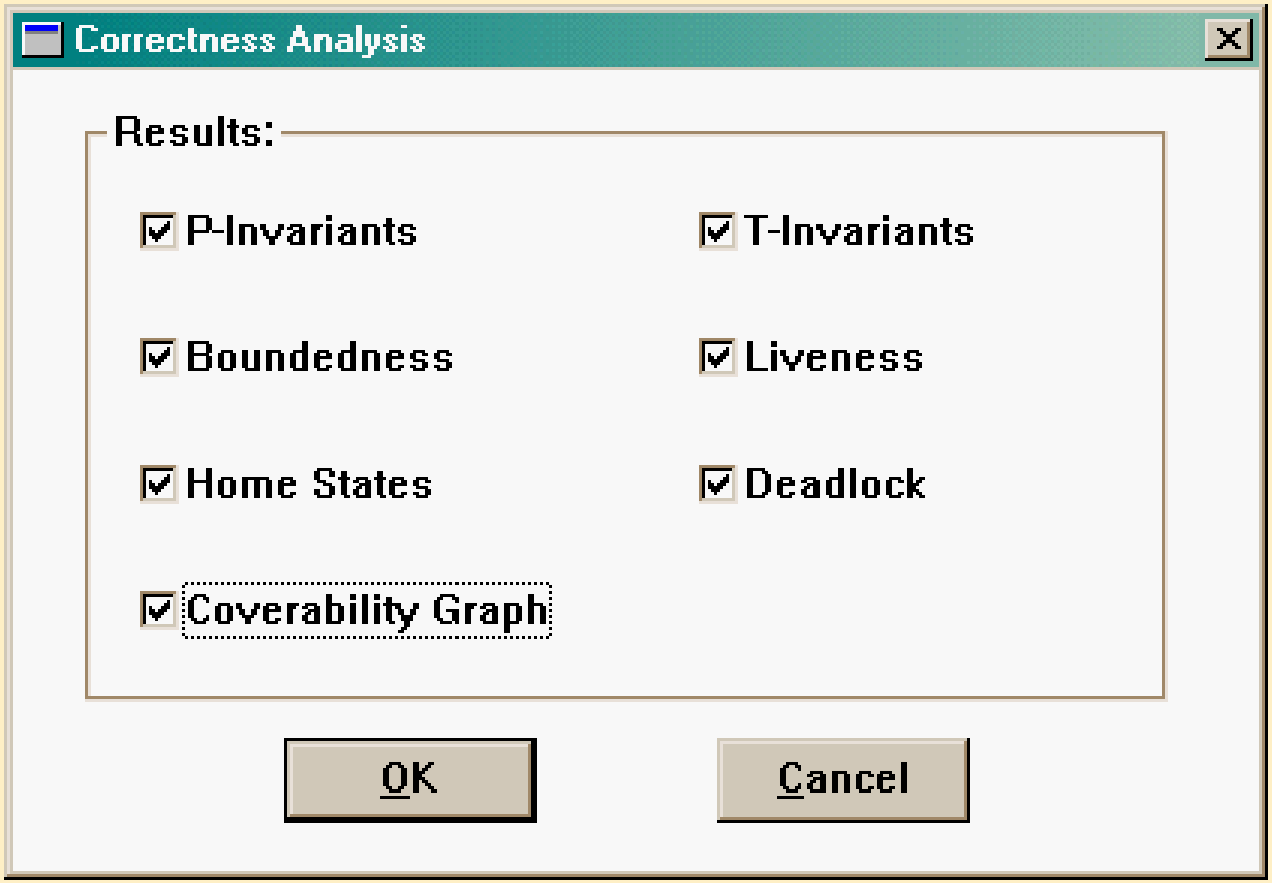
Korlátosság vizsgálata az állapottérben

- 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”

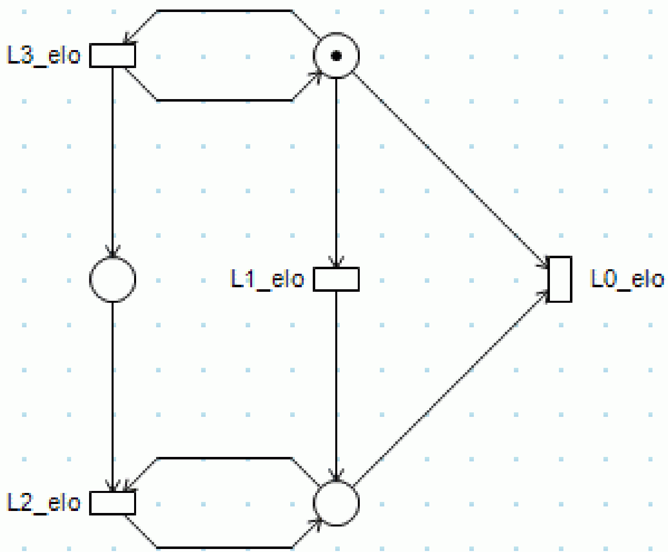
További dinamikus tulajdonságok

- 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?
    - →van: ellenpélda, hiszen létezik tüzelési szekvencia, amiben korlátlan sokszor tüzel
- Perzisztencia:
  - „A tranzíció mindaddig engedélyezett marad, amíg nem tüzel”
    - →ha több engedélyezett és nem ő tüzelt, akkor a következő állapotban is engedélyezett marad
    - →ha engedélyezett maradt, akkor meg is jelenik élcímkeként (prioritás?)

DNAnet: Ellenőrzés



PetriDotNet: dinamikus tulajdonságok



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)

Petri hálók redukciós módszerei

Állapottér redukció

Elérhetőségi probléma kezelése

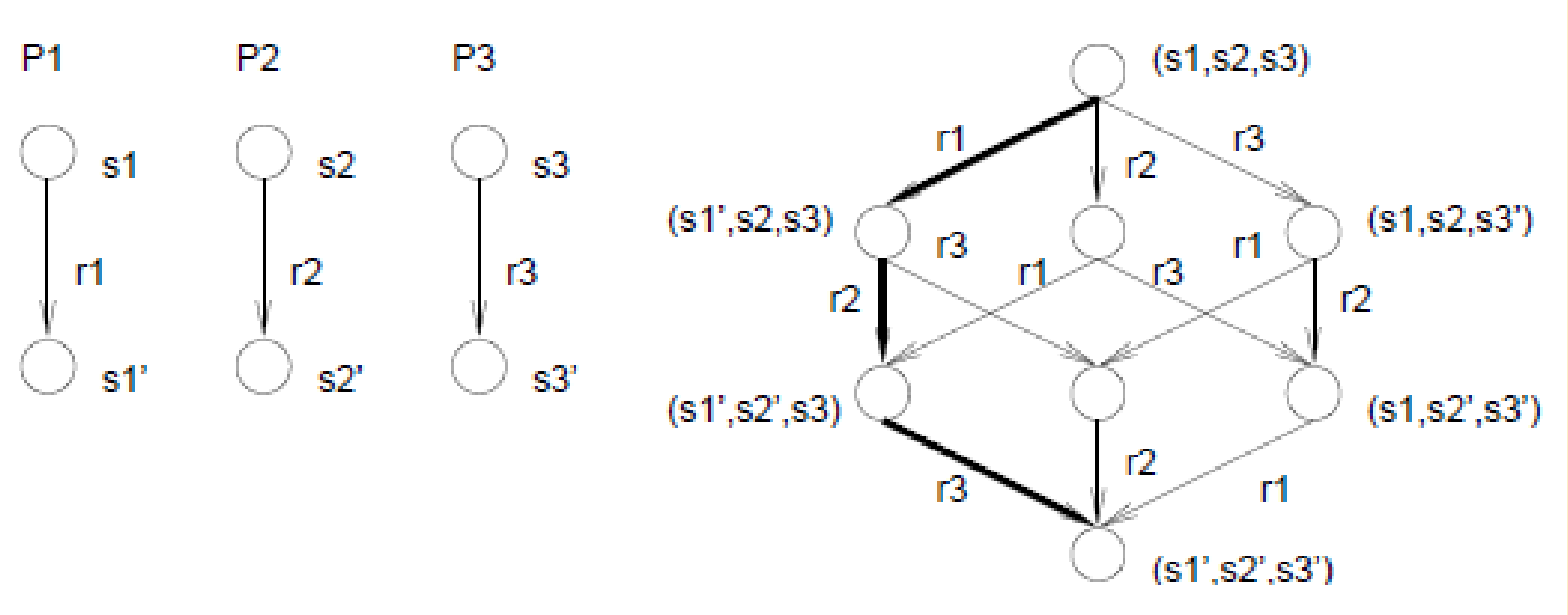
- Struktúra redukálása
  - Tulajdonságmegtartó transzformáció redukált modellre
- Hierarchikus modellezés
  - Részhálózatok összevonása egyetlen csomóponttá
  - Petri hálók nemdeterminizmusa $\rightarrow$ modellabsztrakció
    - Keresési tér behatárolása durvább modellen
    - Részletes analízis egy 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

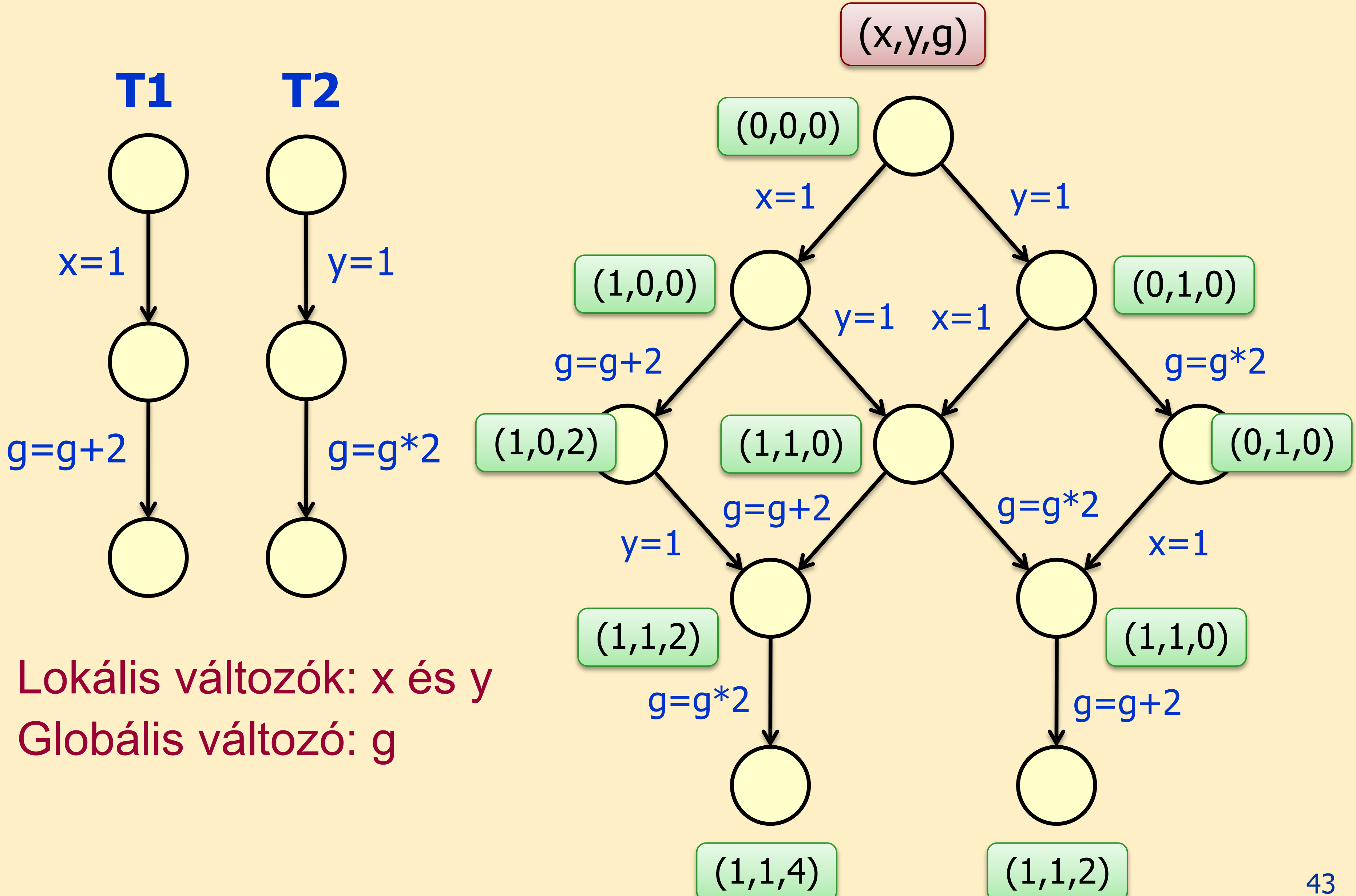
- Szimmetriák kihasználása
  - Azonos hálózatrészek csak egyszer vizsgálva
    - pl. erőforrás csoportok: azonos módon viselkedő tagok
  - Invariancia a ciklikus permutációra nézve
    - Színezett Petri hálók → Jól formált színezett Petri hálók (WFN)
- Állapottér bejárás hatékonyságának növelése
  - Csak az „érdekes” állapotok bejárása
    - Tulajdonságmegőrző redukció
  - Csak a szükséges mennyiségű állapotváltás bejárása
    - Alternatív utak elhagyása

Elérhetőségi probléma egyszerűsítése

- Részleges sorrendezési redukció
  - Elérhető állapotok részlegesen sorrendezett halmazt alkotnak
  - Aszinkron működés: átlapolás → alternatív utak, azonos eredmény
  - Végállapotokat nézve (elérhetőség) az alternatív utak redundánsak



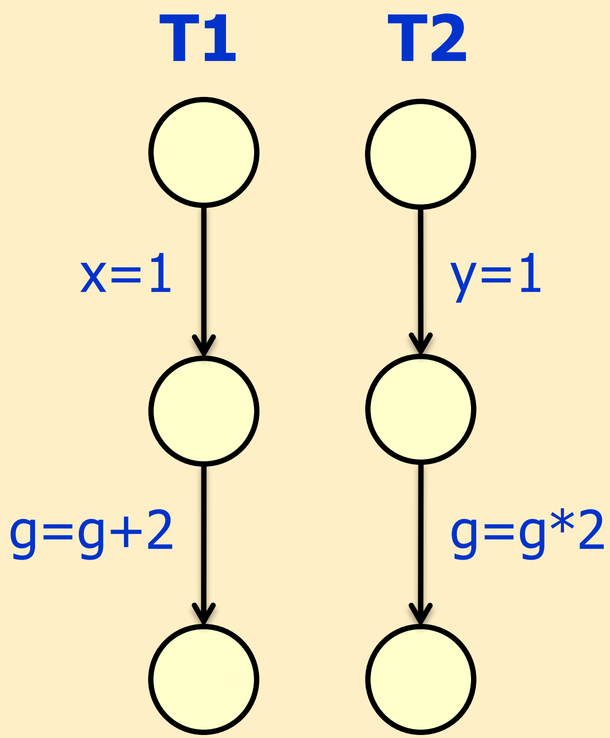
Példa: alternatív utak



Lokális változók: x és y

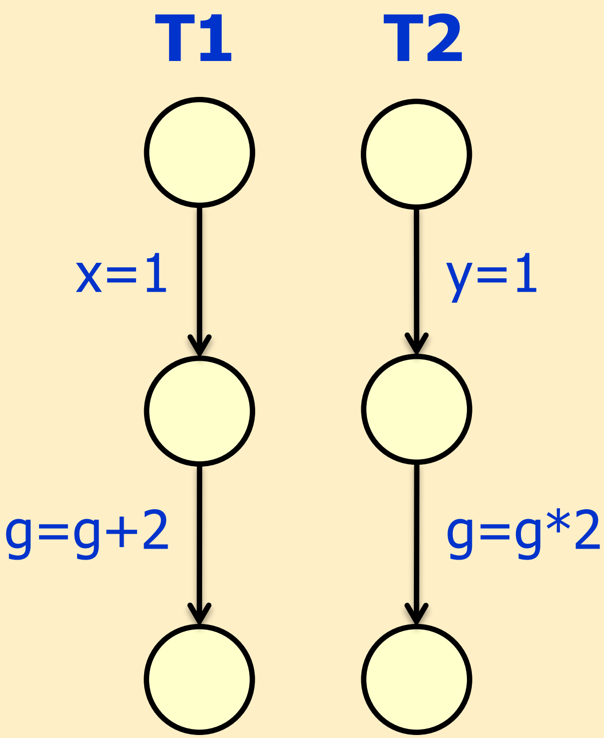
Globális változó: g

Példa: alternatív utak



- Lokális változók:
  - x és y
- Globális változó:
  - g
- 6 lehetséges lefutás:
  1. x=1; g=g+2; y=1; g=g*2
  2. x=1; y=1; g=g+2; g=g*2
  3. x=1; y=1; g=g*2; g=g+2
  4. y=1; x=1; g=g*2; g=g+2
  5. y=1; x=1; g=g+2; g=g*2
  6. y=1; g=g*2; x=1; g=g+2

Példa: alternatív utak



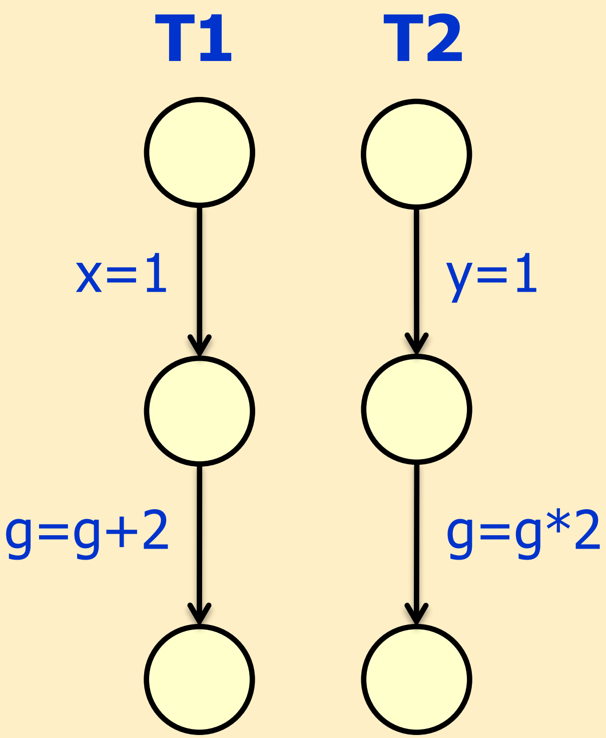
	x=1	y=1	g=g+2	g=g*2
x=1		F	V	F
y=1	F		F	V
g=g+2	V	F		A
g=g*2	F	V	A	

F: független

V: vezérlési függőség

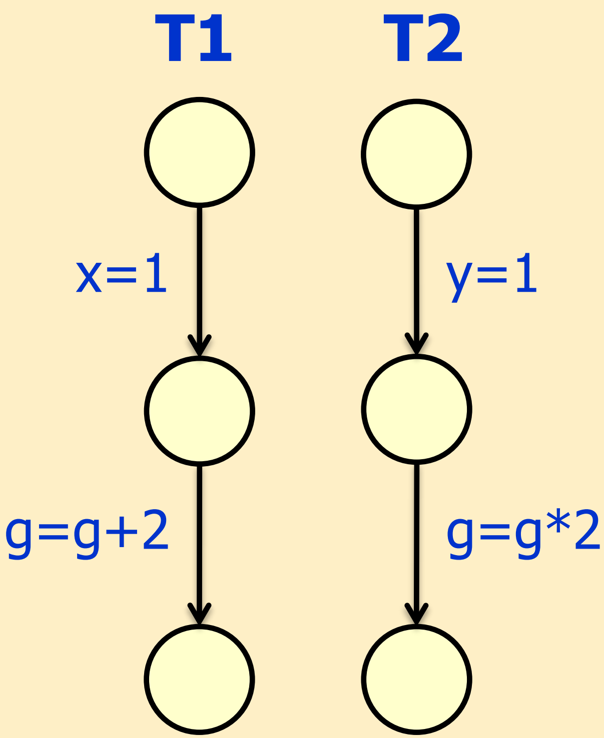
A: adatfüggőség

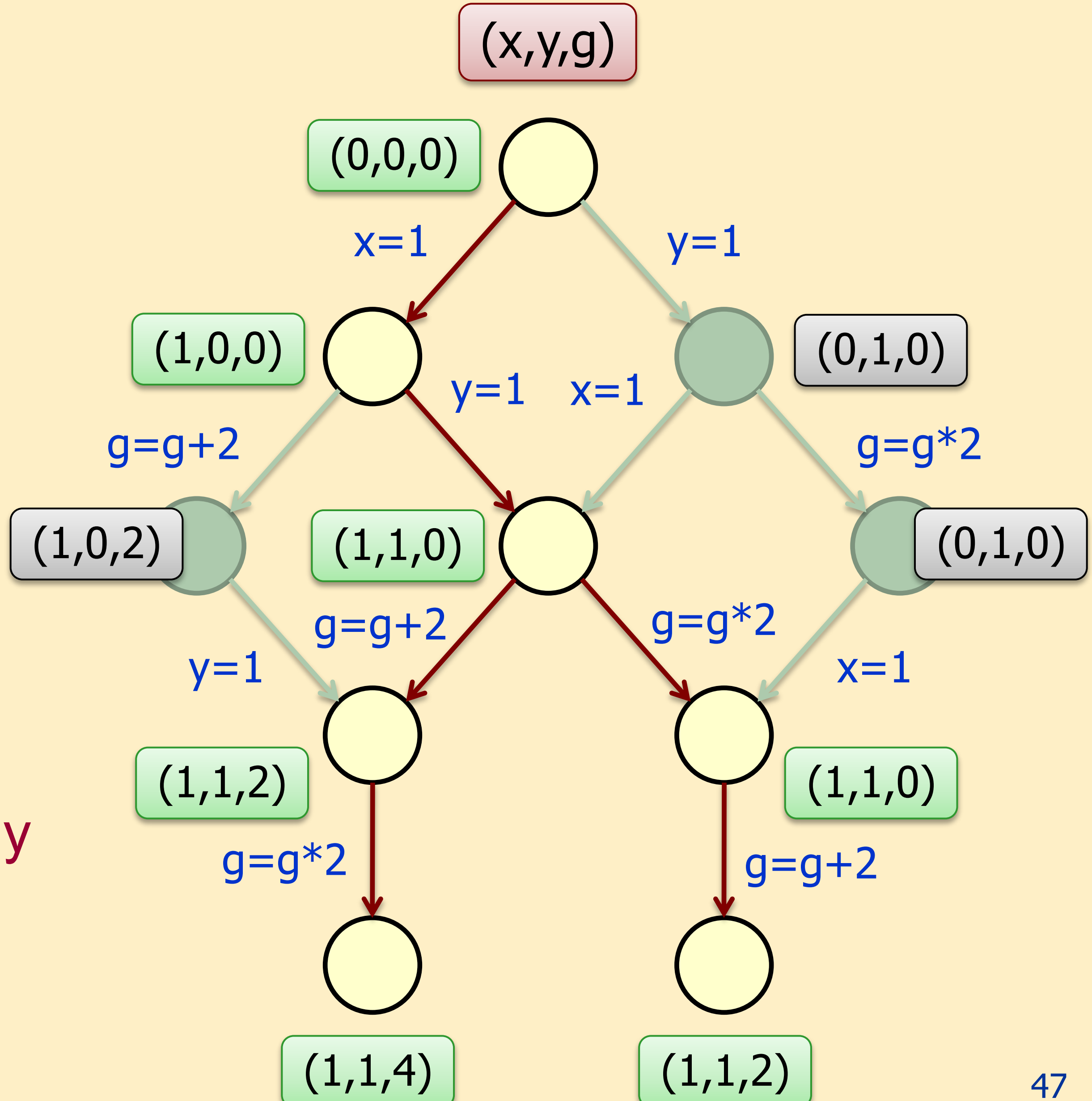
Példa: részleges sorrendezési redukció



1. x=1; g=g+2; y=1; g=g*2
2. x=1; y=1; g=g+2; g=g*2
3. x=1; y=1; g=g*2; g=g+2
4. y=1; x=1; g=g*2; g=g+2
5. y=1; x=1; g=g+2; g=g*2
6. y=1; g=g*2; x=1; g=g+2

Példa: alternatív utak





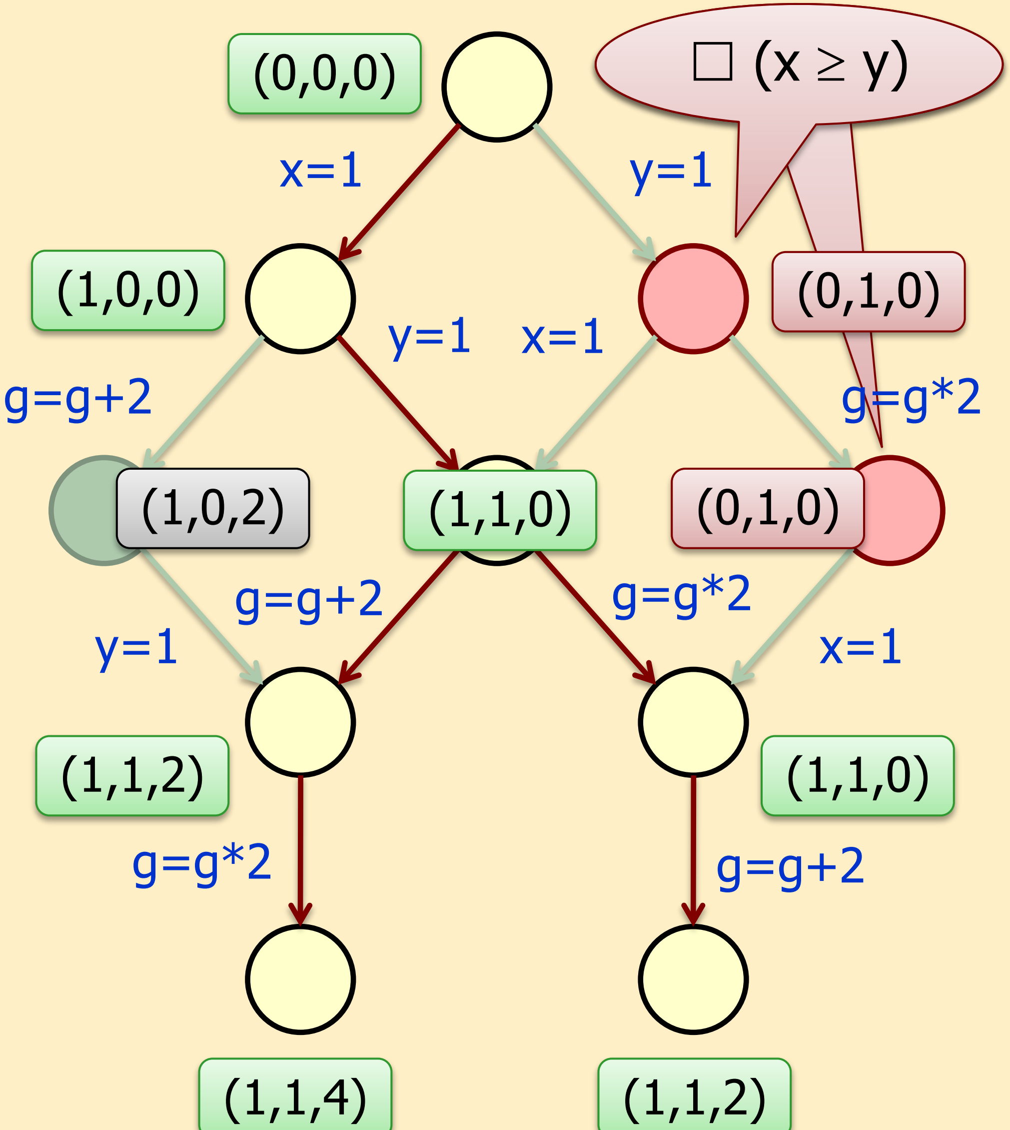
Lokális változók: x és y

Globális változó: g

A részleges sorrendezési redukció alkalmazása

- Két LTS: T1 és T2
  - sorrendezett: $x \rightarrow g$ és $y \rightarrow g$
    - igaz tulajdonság: pl. $\Box\ (g = 0 \vee g > x)$
- Redukált gráf
  - eltávolított élek: szürke
  - független: x=1 és y=1
  - adatfüggő: g+=2 és g*=2
    - igaz tulajdonság: pl. $\Diamond\ (g \geq 2)$
- Redukció
  - redundáns utak eltávolítása
  - vigyázat, függ a céltól!
    - pl. $\Box\ (x \geq y)$ tulajdonság a redukáltban igaz, az eredetiben nem igaz

Példa: tulajdonság alapú függőség



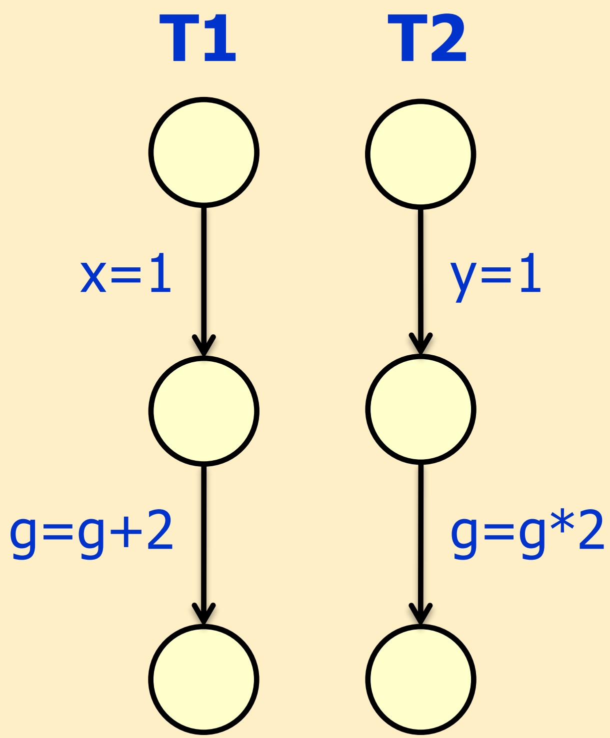
	x=1	y=1	g=g+2	g=g*2
x=1		T	V	F
y=1	T		F	V
g=g+2	V	F		A
g=g*2	F	V	A	

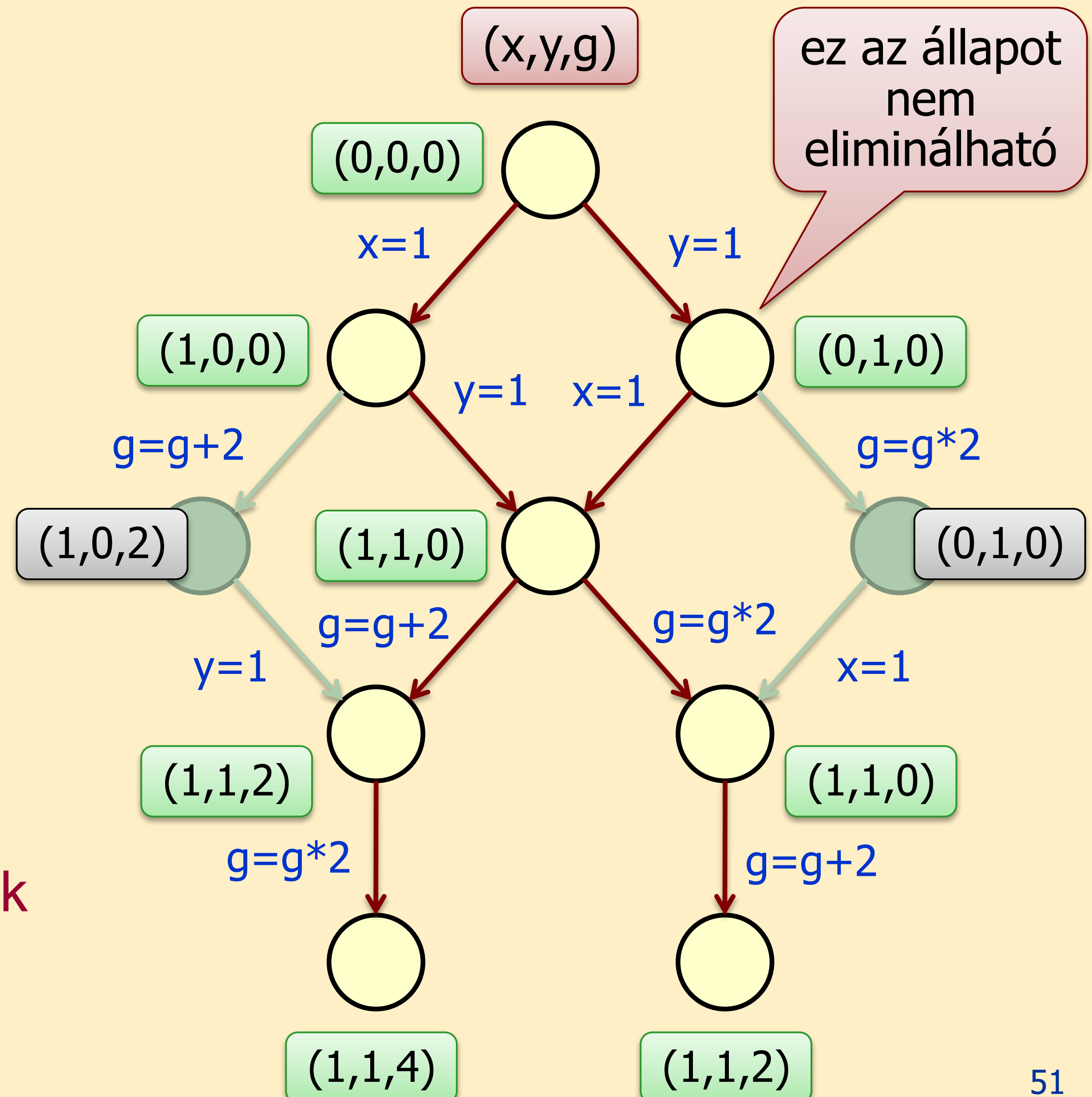
Példa: tulajdonságmegőrző redukció

	x=1	y=1	g=g+2	g=g*2
x=1		T	V	F
y=1	T		F	V
g=g+2	V	F		A
g=g*2	F	V	A	

1. x=1; g=g+2; y=1; g=g*2
2. x=1; y=1; g=g+2; g=g*2
3. x=1; y=1; g=g*2; g=g+2
4. y=1; x=1; g=g*2; g=g+2
5. y=1; x=1; g=g+2; g=g*2
6. y=1; g=g*2; x=1; g=g+2

Példa: tulajdonságmegőrző redukció





x és y → tulajdonság szempontból függenek

Részleges sorrendezési redukció alapja

- Két állapotátmenet független egy s állapotban, ha
  - mindkettő engedélyezett az s állapotban
  - egyikük végrehajtása sem tiltja le a másikat (nincs vezérlési függőség – perzisztencia)
  - a két állapotátmenet együttes hatása független a végrehajtási sorrendtől (sem adat, sem tulajdonság függőség)
- Erős függetlenség
  - két állapotátmenet erősen független, ha függetlenek minden olyan állapotban, amelyben mindkettő engedélyezett

Petri hálók redukciós módszerei

Struktúra redukció

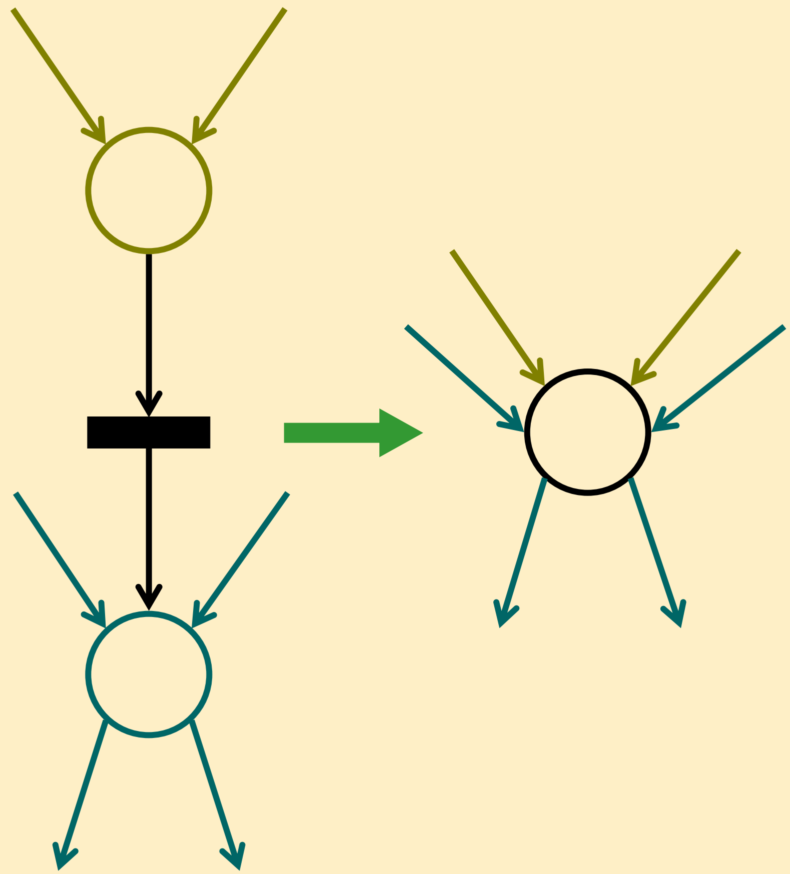
Elérhetőségi probléma egyszerűsítése

- 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

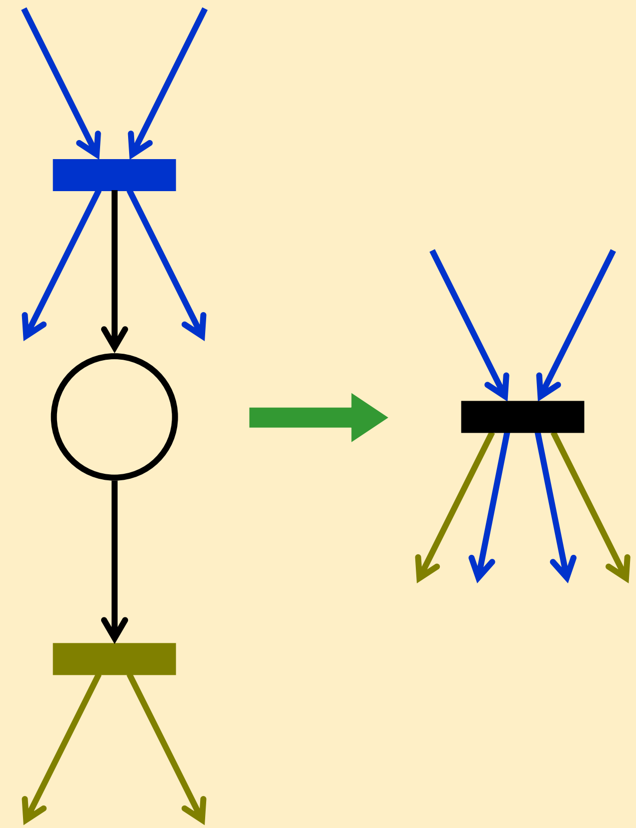
Transzformációk

- 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

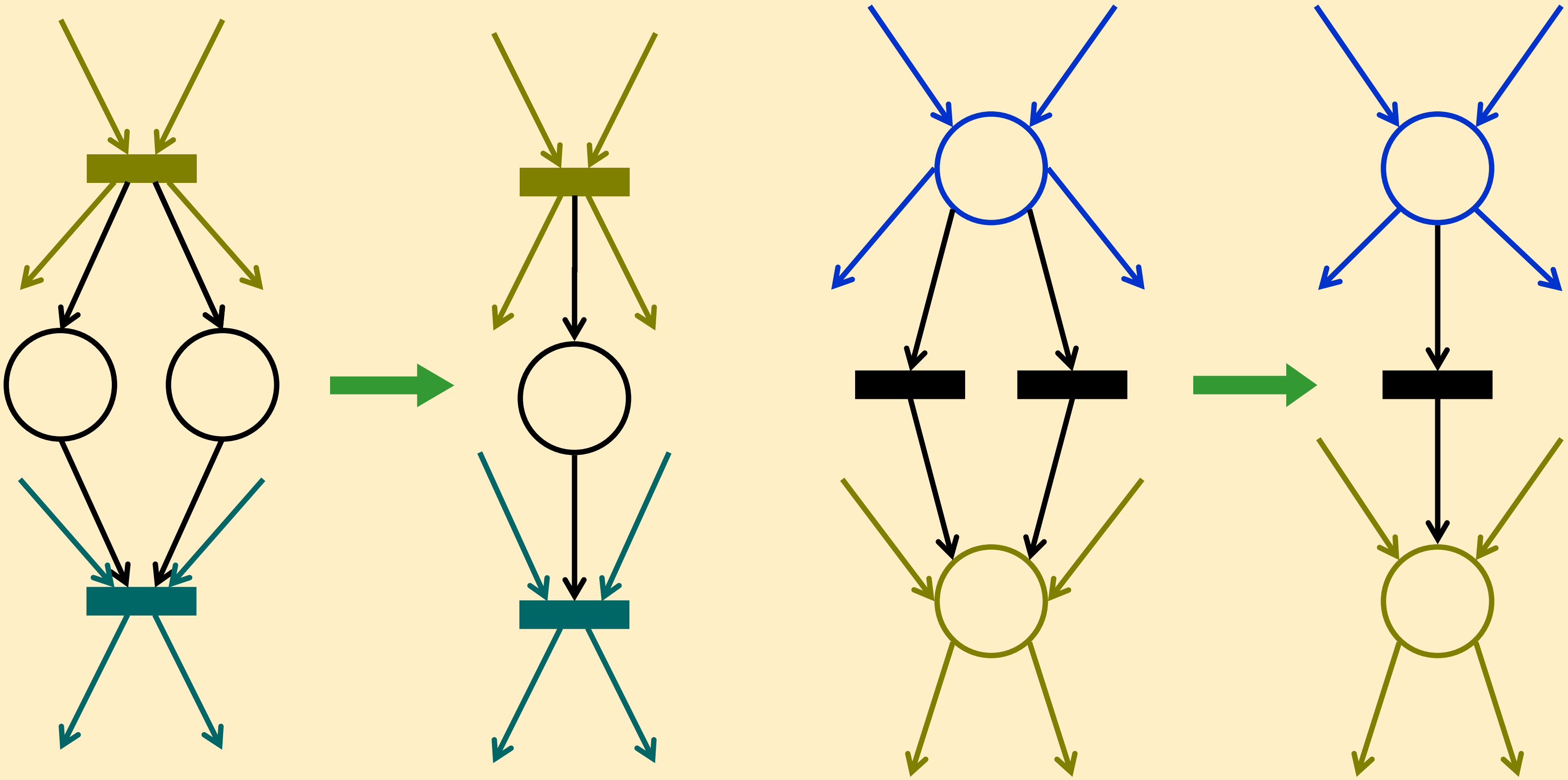


soros helyek összevonása (FSP)



soros tranzíciók összevonása (FST)

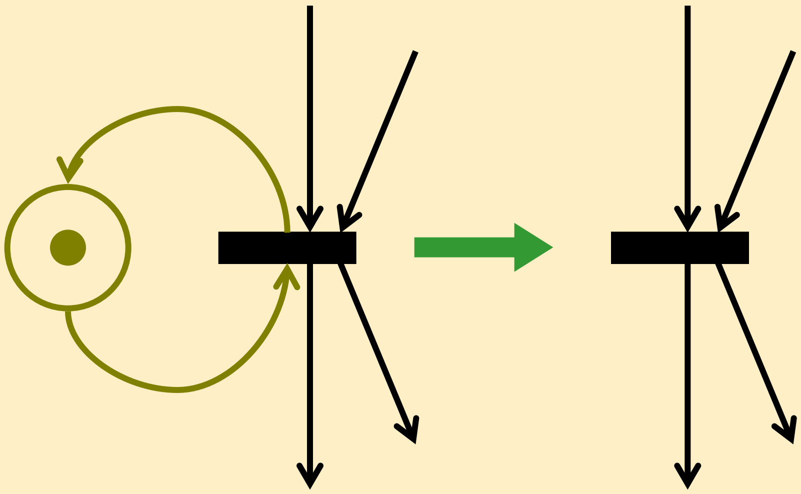
Párhuzamos összevonások



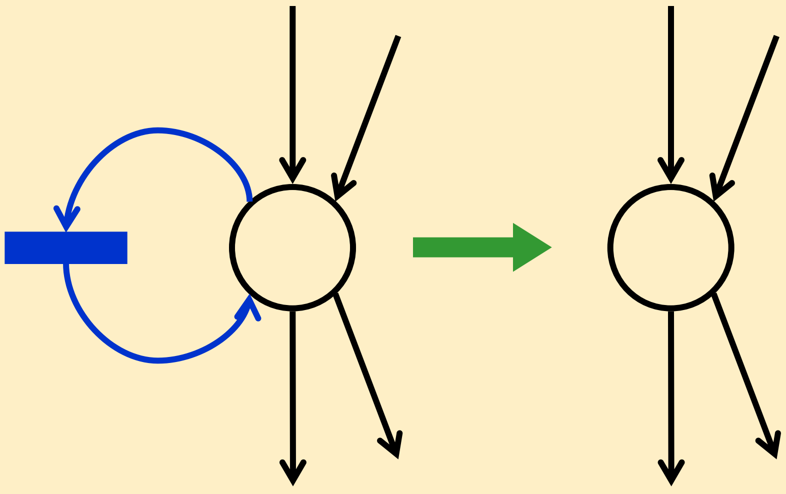
párhuzamos helyek összevonása (FPP)

párhuzamos tranzíciók összevonása (FPT)

Önhurkok törlése

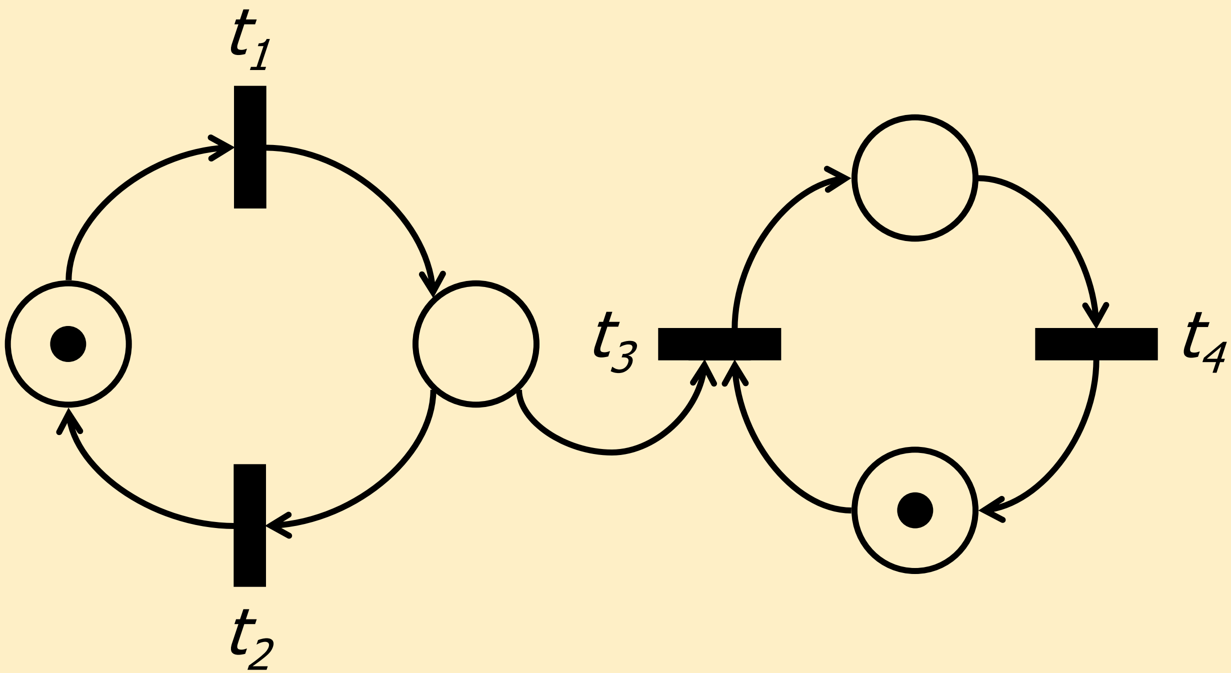


önhurkot alkotó helyek törlése (ESP)



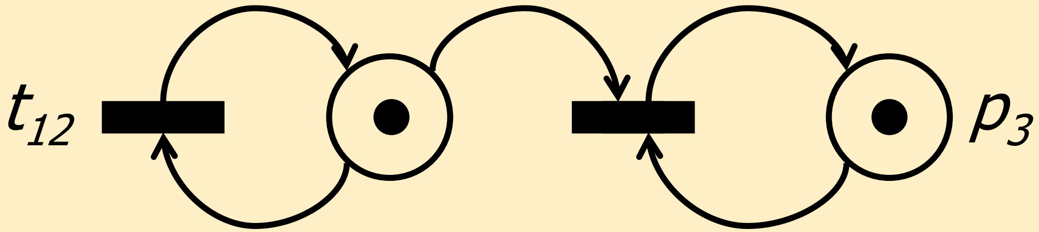
önhurkot alkotó tranzíciók törlése (EST)

Példa



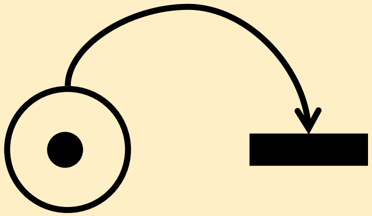
- t_1 tüzelése
- t_1 és t_2 összevonása (soros tranzíciók) $\rightarrow$ t_{12}
- t_3 és t_4 összevonása (soros tranzíciók) $\rightarrow$ t_{34}

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)