

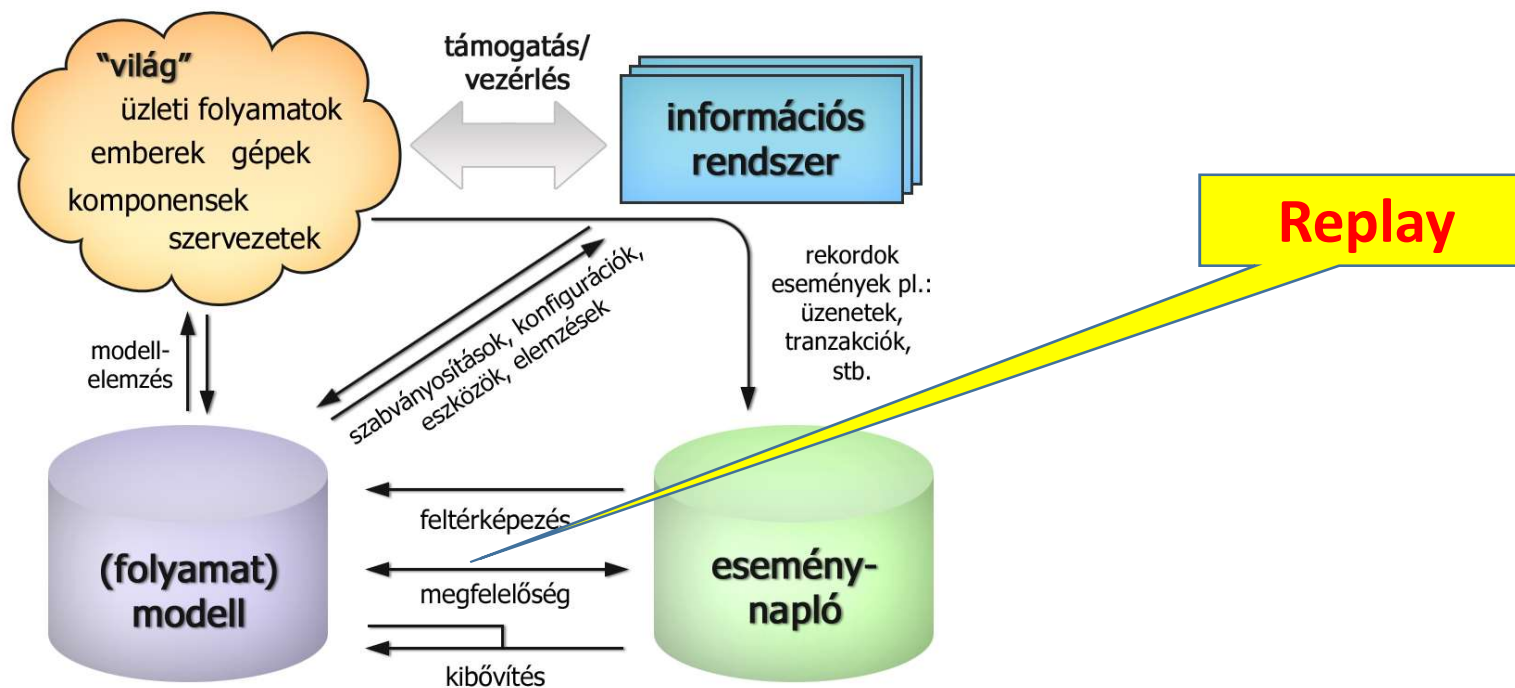
Folyamatbányászat

Konformancia elemzés – megfelelőség elemzés II.

Token alapú és illesztés alapú
megfelelőség ellenőrzés

A konformancia ellenőrzés módszerei

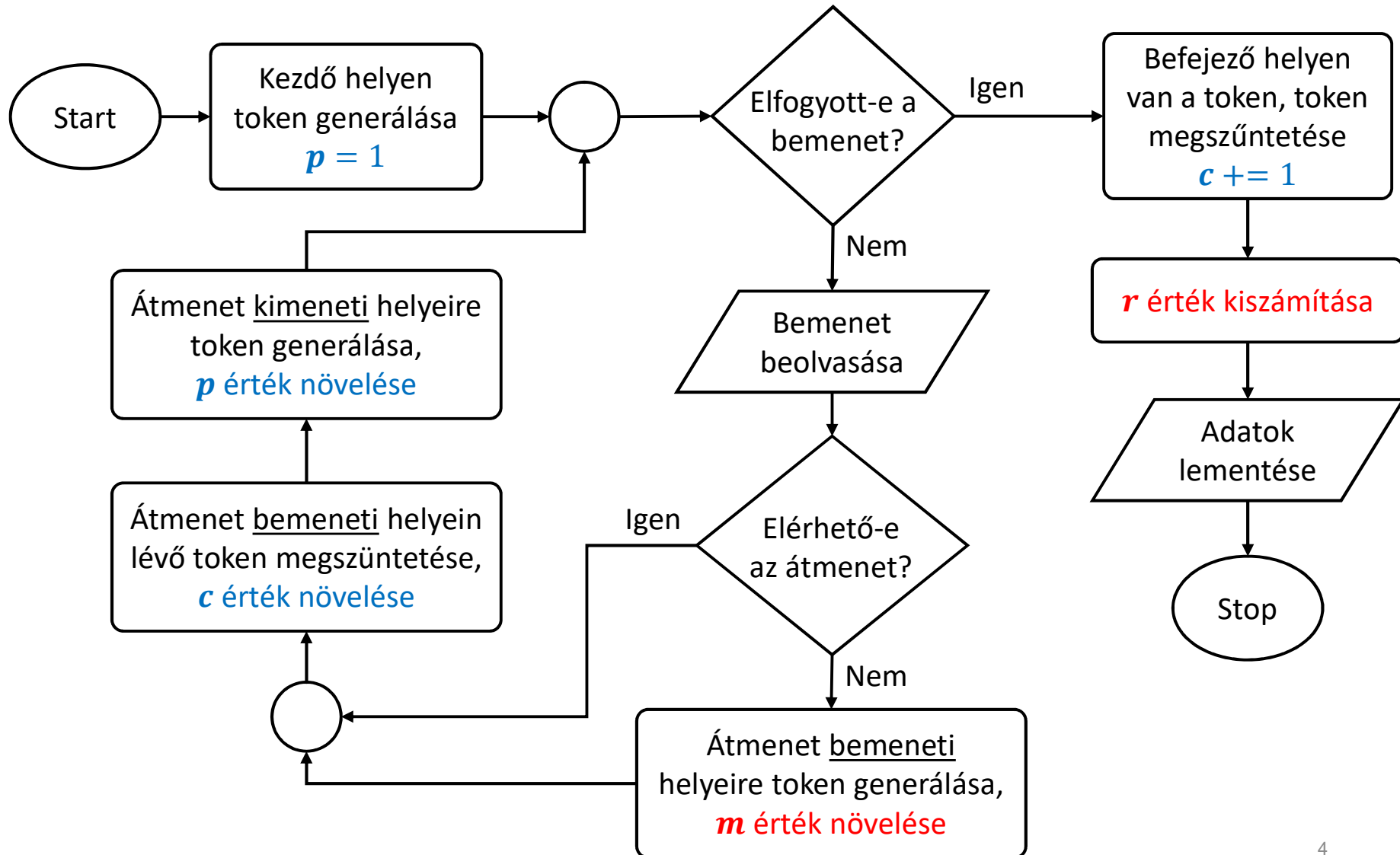
1. Konformancia ellenőrzés **ok-okozati lábnyomokkal**,
2. **Konformancia ellenőrzés token alapú visszajátszással**,
3. **Illesztés alapú** megfelelés ellenőrzés.



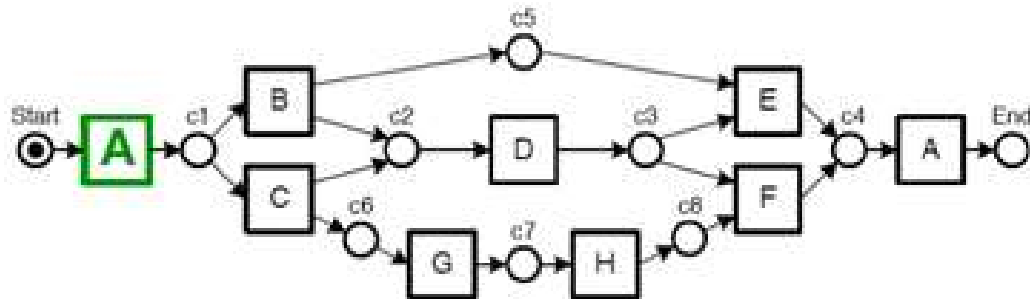
Konformancia token alapú egyezőség-vizsgálata

- Ehhez szükség van a **folymat modelljére**, pl. Petri háló formájában
- A **napló fájlok** rengeteg információt tartalmaznak, melyek sok esetben nem feltétlen szükségesek a vizsgálat elvégzéséhez. Ezért az adatoknak előzetes átalakításon kell átesniük, mely folyamán csak **azon esemény tulajdonságokat tartjuk meg, melyek számunkra szükségesek.**
- Bevezetünk négy különböző mérőszámot:
 1. ***p*** - létrehozott token (produced token),
 2. ***c*** - elhasznált token (consumed token),
 3. ***m*** - hiányzó token (missing token),
 4. ***r*** - megmaradt token (remaining token).

Konformancia token alapú egyezőség-vizsgálatának folyamatábrája

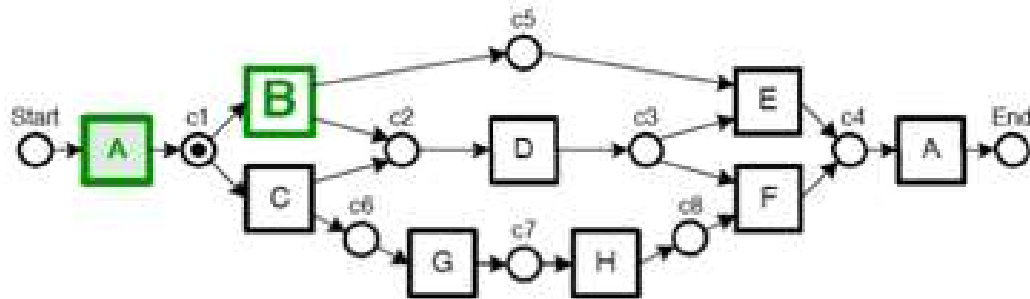


Fitnesz érték figyelése az események végig követésével egy trace-ben



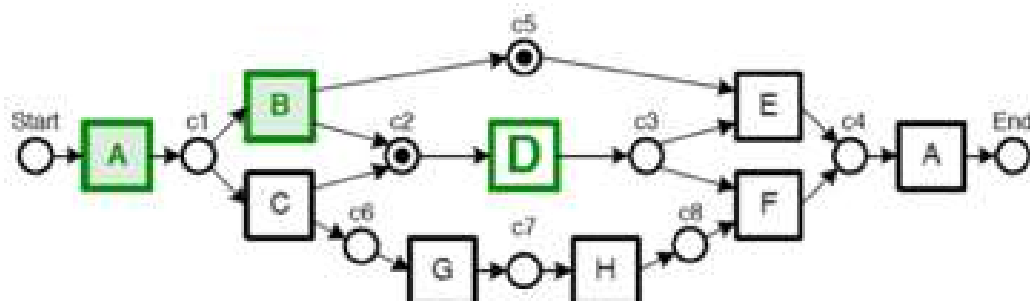
$m = 0$
 $r = 0$
 $c = 0$
 $p = 1$

No. of Instances	Log Traces
1207	→ A BDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA



$m = 0$
 $r = 0$
 $c = 1$
 $p = 2$

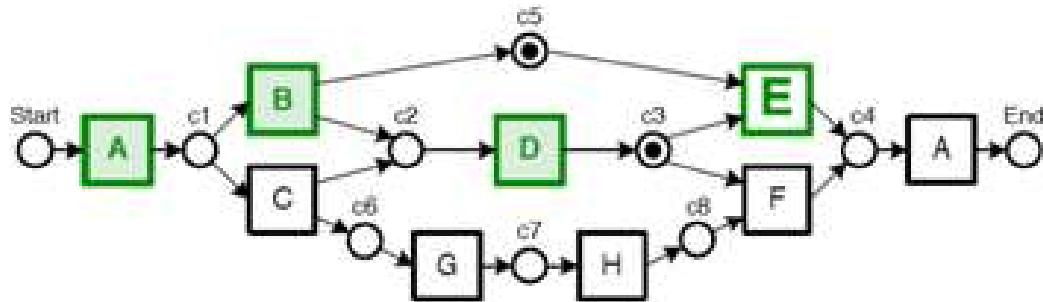
No. of Instances	Log Traces
1207	→ AB DEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA



$m = 0$
 $r = 0$
 $c = 2$
 $p = 4$

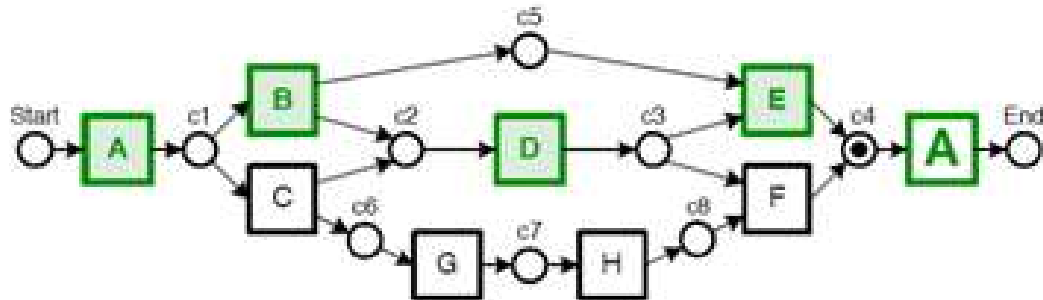
No. of Instances	Log Traces
1207	→ ABD EA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

Ha nincs probléma: $m = 0 \wedge r = 0$



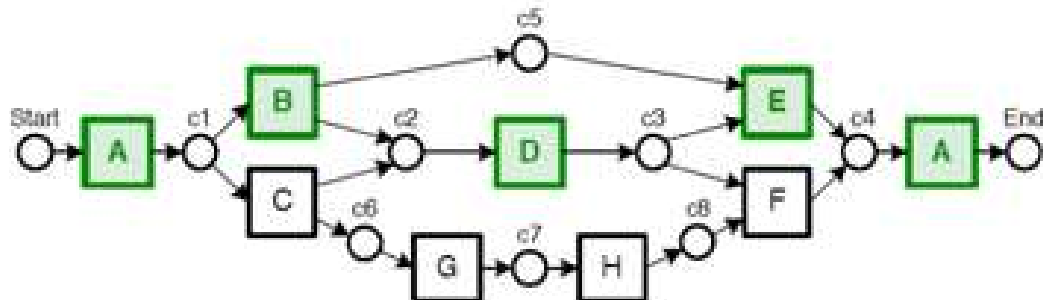
$m = 0$
 $r = 0$
 $c = 3$
 $p = 5$

No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA



$m = 0$
 $r = 0$
 $c = 5$
 $p = 6$

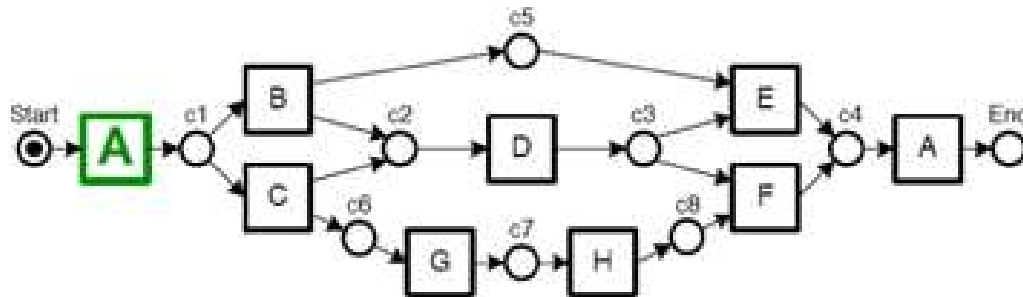
No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA



$m = 0$
 $r = 0$
 $c = 7$
 $p = 7$

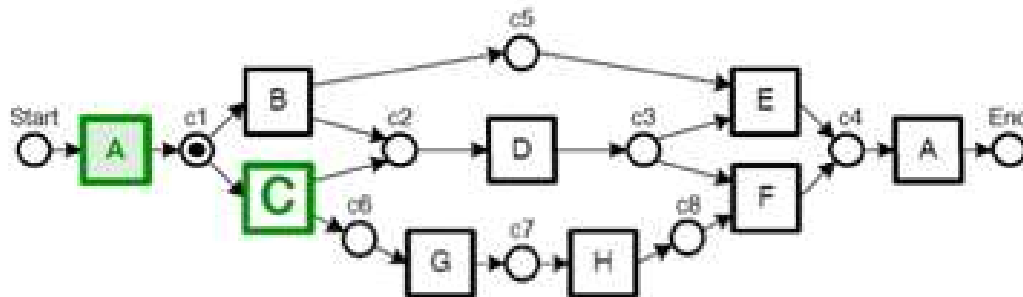
No. of Instances	Log Traces
1207	→ ABDEA
145	ACDGHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

Ha probléma van: $m > 0$ v $r > 0$



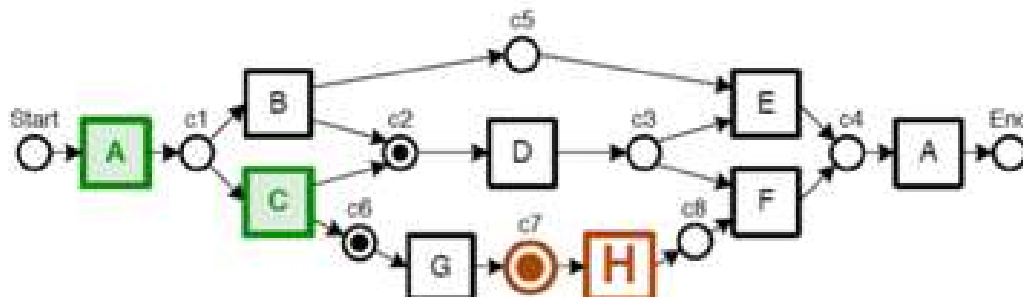
$m = 0$
 $r = 0$
 $c = 0$
 $p = 1$

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA



$m = 0$
 $r = 0$
 $c = 1$
 $p = 2$

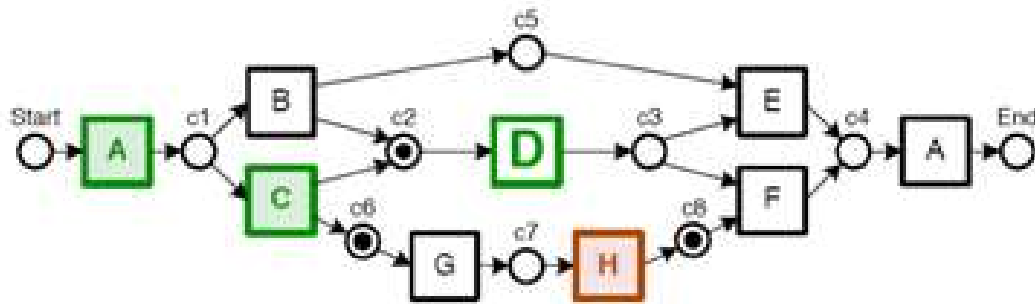
No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA



$m = 1$
 $r = 0$
 $c = 2$
 $p = 4$

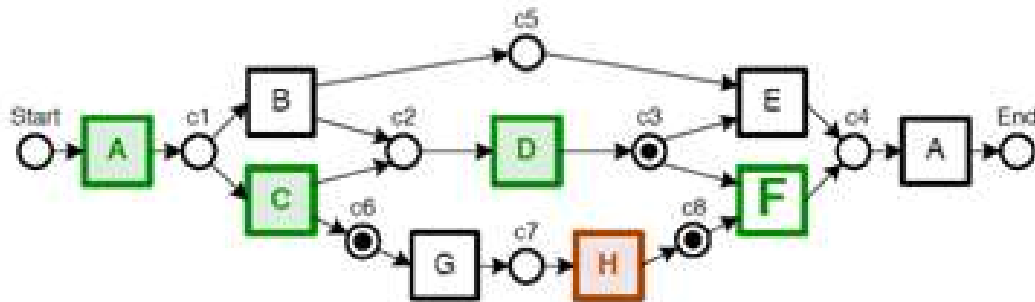
No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

Folytatás



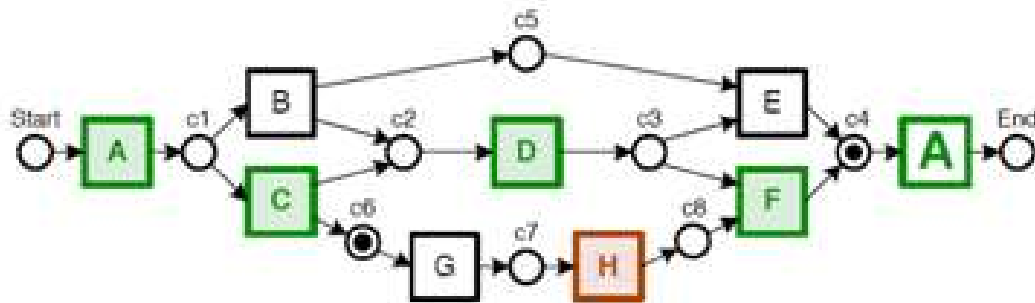
$m = 1$
 $r = 0$
 $c = 3$
 $p = 5$

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA



$m = 1$
 $r = 0$
 $c = 4$
 $p = 6$

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA



$m = 1$
 $r = 0$
 $c = 6$
 $p = 7$

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

Fitness érték számítása

Token alapú egyezőség-vizsgálat esetén

Egy trace (t) **fitness értéke:**

$$f(M, t) = \frac{1}{2} \left(1 - \frac{m}{c} \right) + \frac{1}{2} \left(1 - \frac{r}{p} \right)$$

hiányzó token(ek) m megmaradt token(ek) r
elhasznált token(ek) c létrehozott token(ek) p

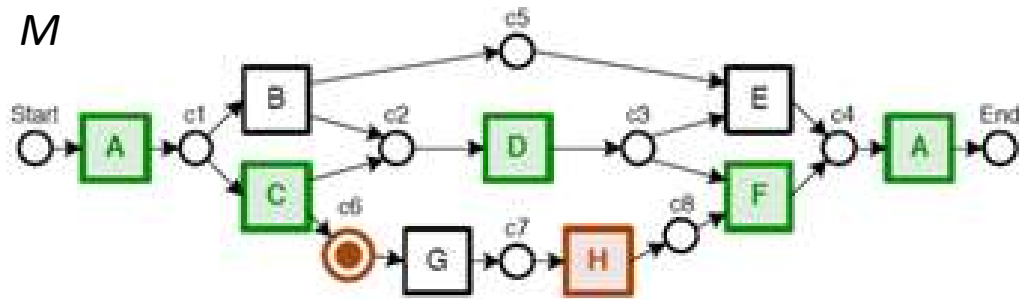
Egy log (L) **fitness értéke:**

$$f(M, L) = \frac{1}{2} \left(1 - \frac{\sum_{i=1}^k n_i m_i}{\sum_{i=1}^k n_i c_i} \right) + \frac{1}{2} \left(1 - \frac{\sum_{i=1}^k n_i r_i}{\sum_{i=1}^k n_i p_i} \right)$$

különböző tracek száma $\sum_{i=1}^k n_i m_i$ azonos tracek száma $\sum_{i=1}^k n_i r_i$

Fitness érték számítása

Token alapú egyezőség-vizsgálat esetén



$m = 1$
 $r = 1$
 $c = 8$
 $p = 8$

L

No. of Instances	Log Traces
1207	ABDEA
145	ACDGHFA
56	ACGDHFA
23	→ ACHDFA
28	ACDHFA

A fenti trace **fitness értéke:**

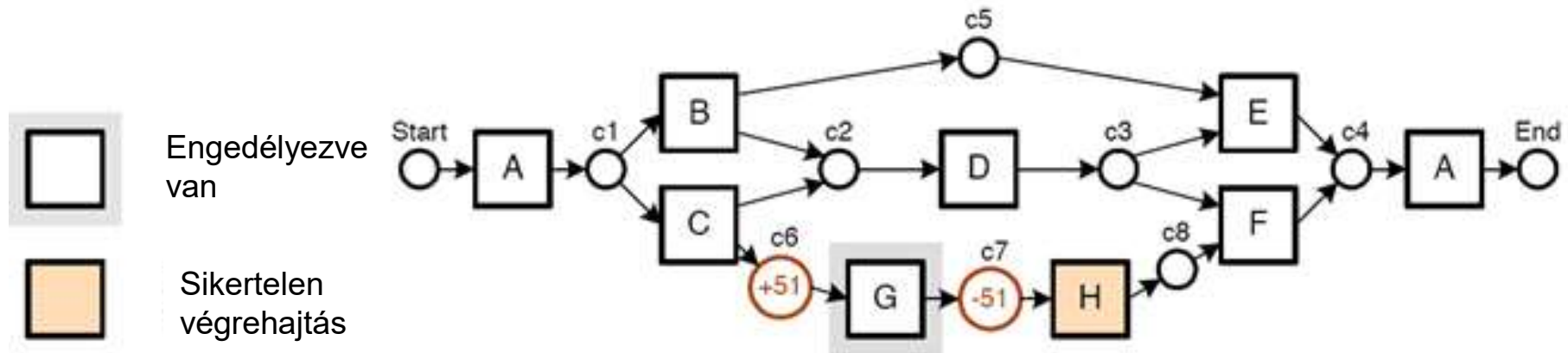
$$f(M, t) = \frac{1}{2} \left(1 - \frac{m}{c} \right) + \frac{1}{2} \left(1 - \frac{r}{p} \right) = \frac{1}{2} \left(1 - \frac{1}{8} \right) + \frac{1}{2} \left(1 - \frac{1}{8} \right) = \frac{7}{8} = \mathbf{0,875}$$

A fenti **log fitness értéke:**

$$f(M, L) = \frac{1}{2} \left(1 - \frac{\sum_{i=1}^k n_i m_i}{\sum_{i=1}^k n_i c_i} \right) + \frac{1}{2} \left(1 - \frac{\sum_{i=1}^k n_i r_i}{\sum_{i=1}^k n_i p_i} \right) = \dots =$$

$$= \frac{1}{2} \left(1 - \frac{51}{10666} \right) + \frac{1}{2} \left(1 - \frac{51}{10666} \right) \approx \mathbf{0,995}$$

Hogyan használhatjuk a módszert diagnosztikához



A **G** művelet valamilyen oknál fogva nem került végrehajtásra.
Ki kell deríteni, hogy mi volt az oka!

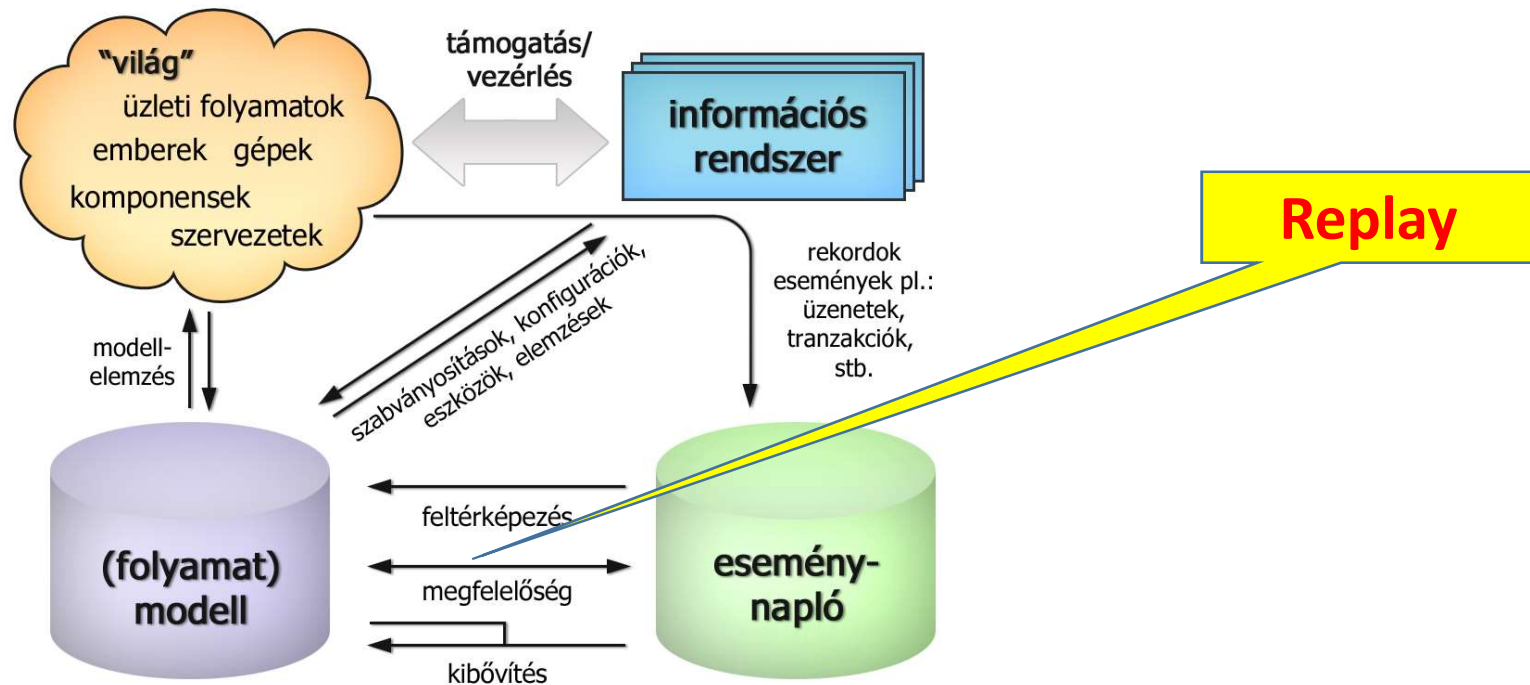


A probléma lehetséges forrásai:

- **végrehajtási probléma** → **G** nem lett végrehajtva
- **rögzítési probléma** → **G** végre lett hajtva, csak nem került rögzítésre a logban
- **modellezési probléma** → **G** végrehajtása opcionális vagy (az adott modellben nem megjelenő) feltételekhez kötött

A konformancia ellenőrzés módszerei

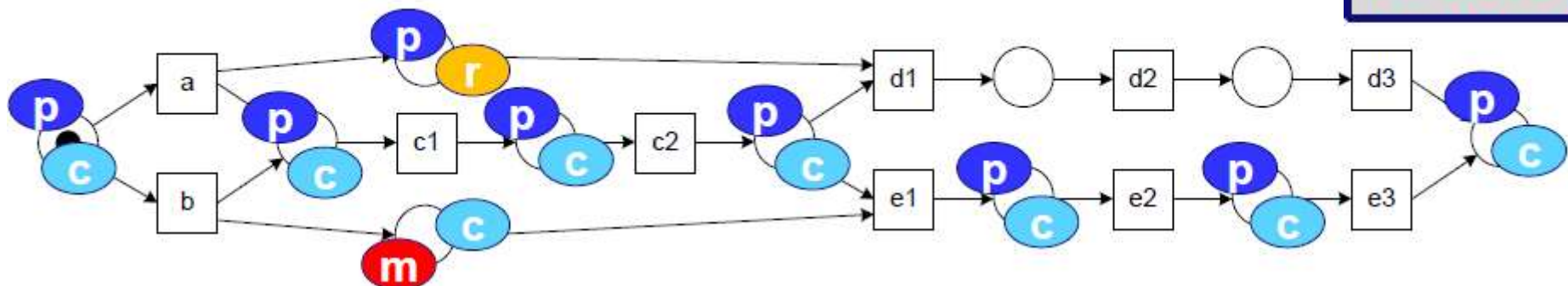
1. Konformancia ellenőrzés **ok-okozati lábnyomokkal**,
2. Konformancia ellenőrzés **token alapú visszajátszással**,
3. **Illesztés alapú** megfelelés ellenőrzés. ←



A token alapú egyezőség módszer korlátai

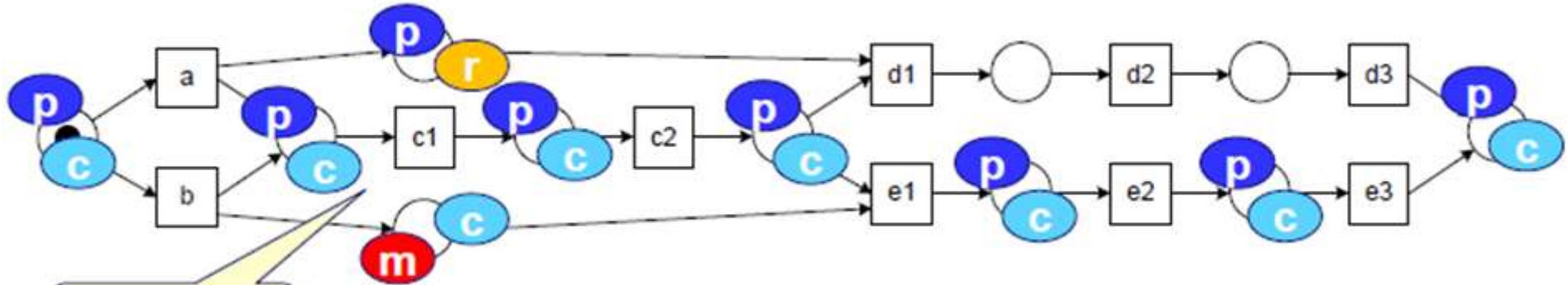
- A módszer feltételezi a láthatóságot és az egyediséget az átmenetek azonosítójának,
- A megfelelőségi értékek bizonyos esetekben túl optimisták,
- A helyi döntés nem mindig helyes,
- A replay technika a modellen keresztül nem feltétlenül ad helyes útvonalat, pedig ez lényeges a megfelelőség vizsgálathoz, a teljesítmény elemzéshez és más diagnosztikákhoz,
- **Szeretnénk látni a „legzártabb útvonalat”, pl. $\langle b, c1, c2, e1, e2, e3 \rangle$**

$\langle a, c1, c2, e1, e2, e3 \rangle$



Illesztés alapú megfelelés ellenőrzés

$\langle a, c1, c2, e1, e2, e3 \rangle$



???

Log mozgás:
csak a logban mozog

Szinkron mozgás:
mindkettőben mozog

A felső sor megfelel egy **trace**-nek a **logban**

Az alsó sor megfelel egy **útvonal**nak a **modellben**

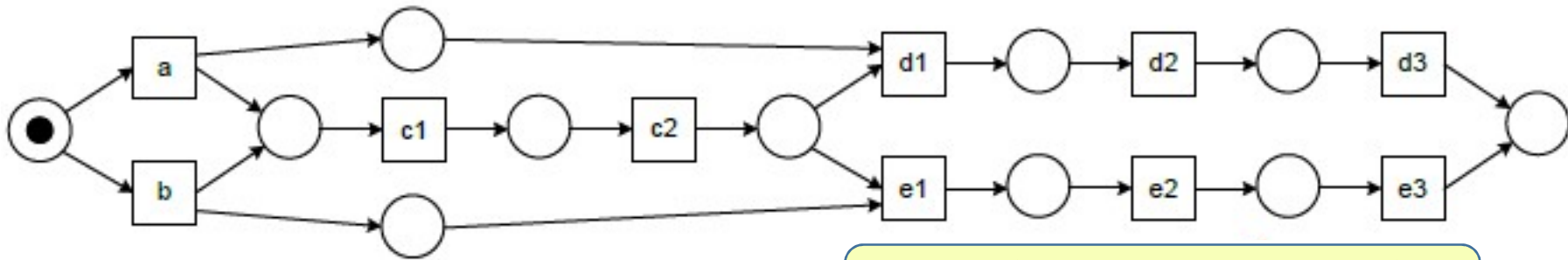
a	»	c1	c2	e1	e2	e3
»	b	c1	c2	e1	e2	e3

Modell mozgás:
csak a modellben mozog

>> - nem mozog

Trace a logban: $\langle a, c1, c2, e1, e2, e3 \rangle$

Összehasonlítjuk a **log**beli lefutást a **modell**beli lehetséges útvonalakkal.



Optimális, nincs másik illesztés, amelynek alacsonyabb a költsége

a	»	c1	c2	e1	e2	e3
»	b	c1	c2	e1	e2	e3

2

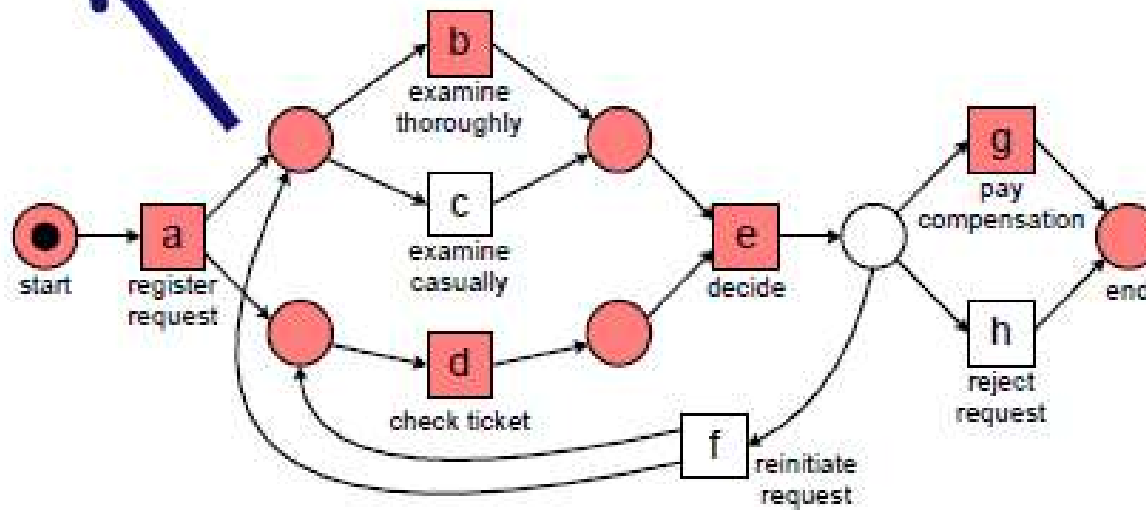
a	c1	c2	»	»	»	e1	e2	e3
a	c1	c2	d1	d2	d3	»	»	»

6

Másik példa: Illesztés vizsgálata

a	b	d	e	g
a	b	d	e	g

← **<a,b,d,e,g>**



#	trace
455	acdeh
191	abdeg
177	acdeh
144	abdeh
111	acdeg
82	adbeg
56	adbeg
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefddeh
6	adcefdbeh
5	adcefddeg
3	acdefdbefdbeg
2	adcefdbeg
2	adcefdbefdbeg
1	adcefdbefodeh
1	adbefdbefdbeg
1	adcefdbefcdefdbeg
1391	

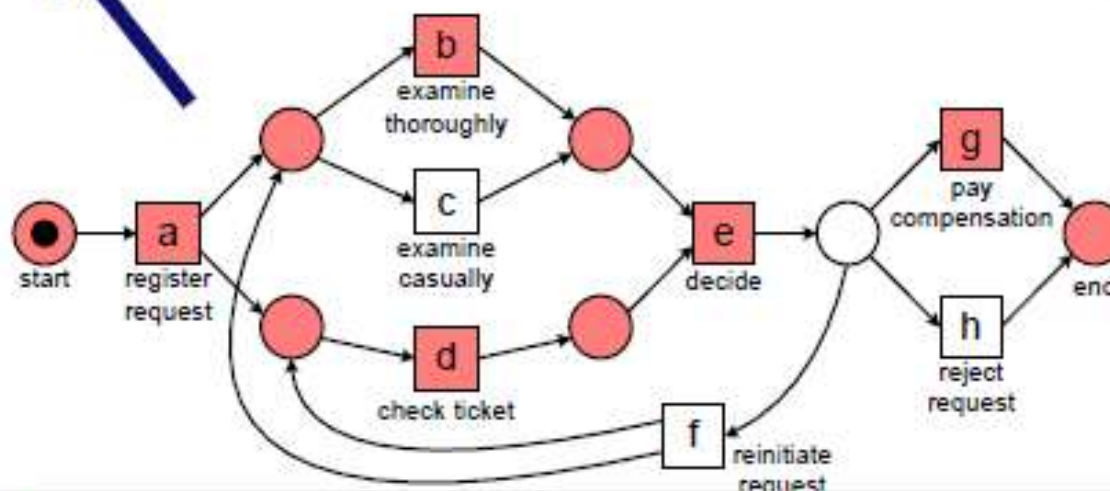
Illesztés vizsgálata

Problémák:

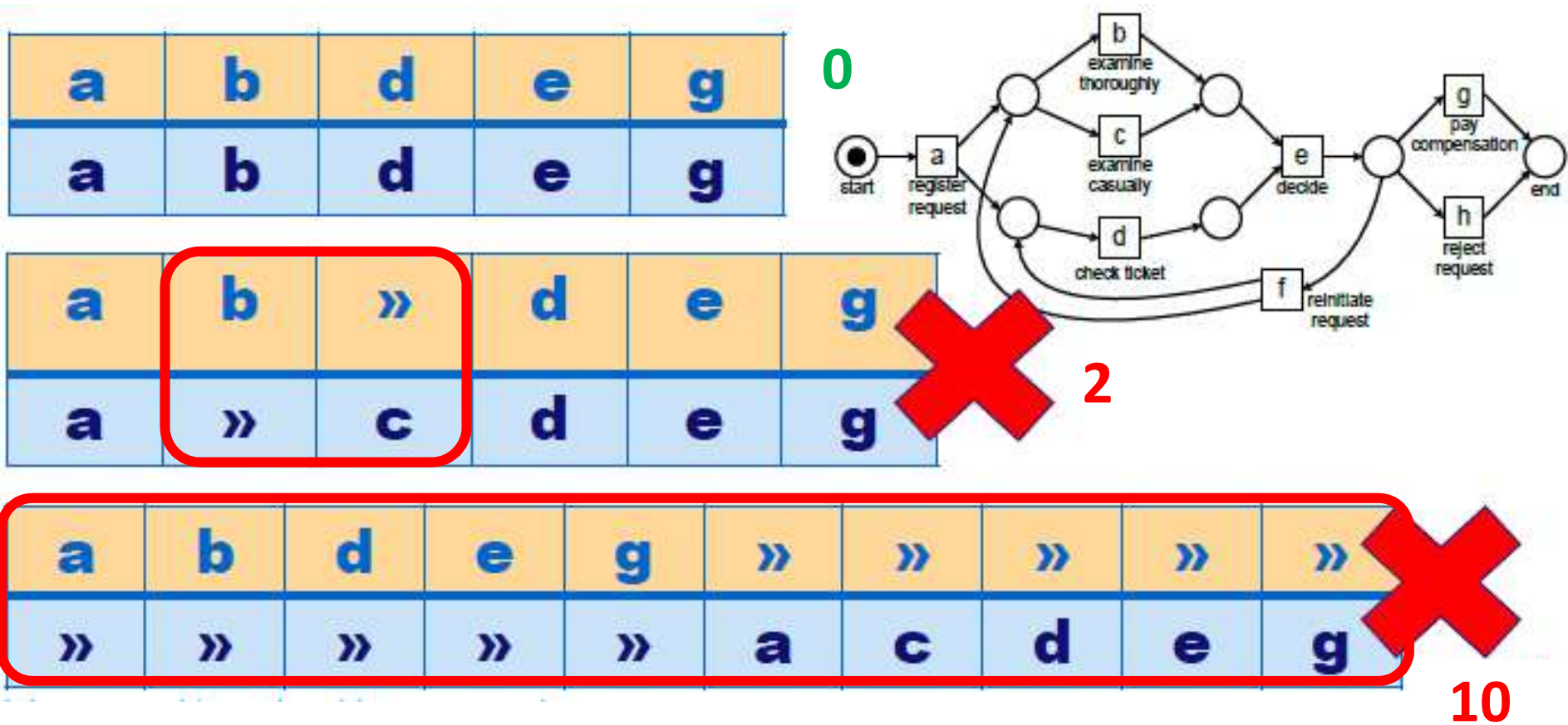
- **b** vagy **c** hiányzik a lefutás során
- **g** és **h** közül csak az egyik hajtódhat végre



← **⟨a,d,e,g,h⟩**



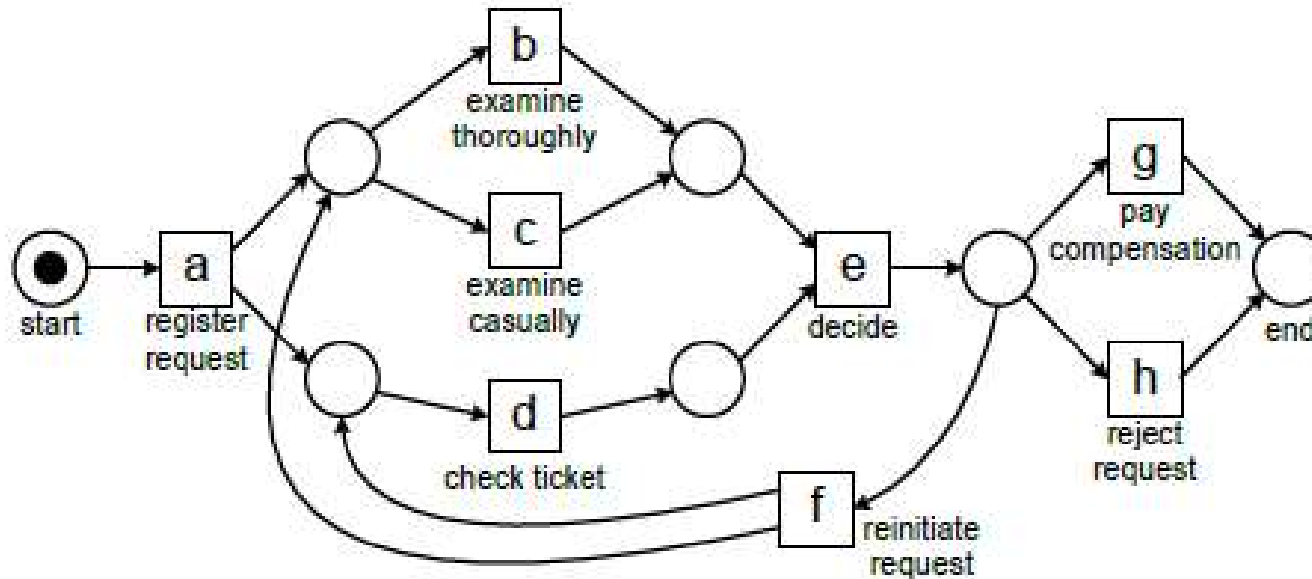
Optimális illesztés $\langle a, b, d, e, g \rangle$ -re



Egy másik optimális illesztés $\langle a, b, e, f, d, e, g \rangle$ -re

a	b	»	e	f	d	»	e	g	2
a	b	d	e	f	d	b	e	g	

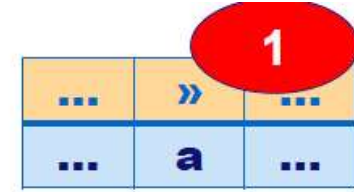
A ciklust tekintjük, így a **d** és a **b** esemény hiányzik a **logban**!



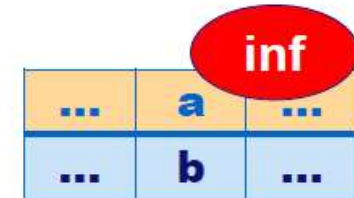
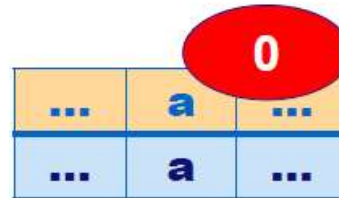
Az optimális illesztés **költségfüggvény** függő!

Az alap költségfüggvény a „>>” jel alapján számol

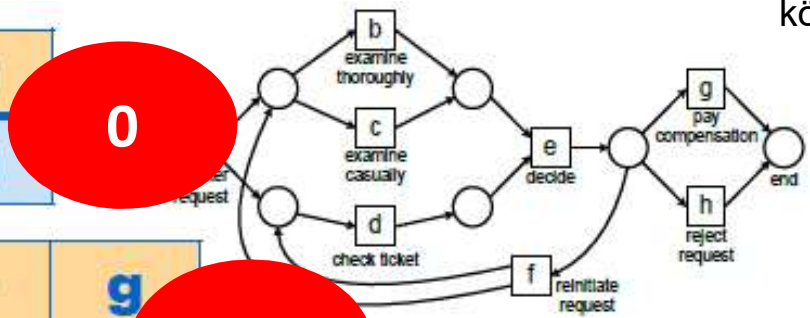
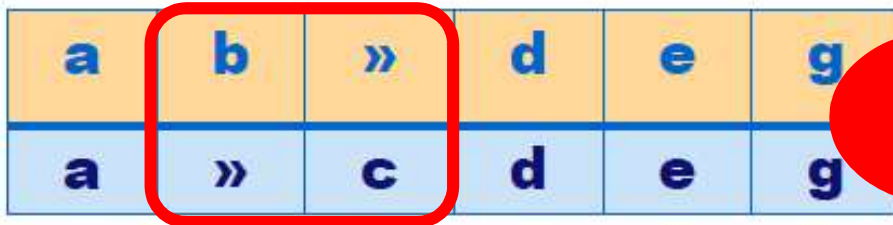
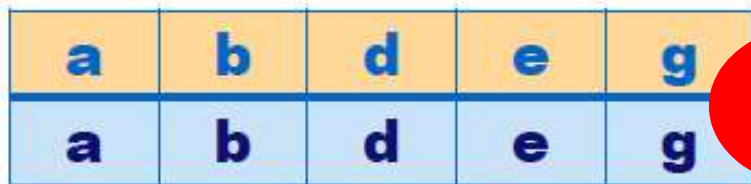
- A **log mozgás** és **modell mozgás** költsége **1**.



- A **szinkron mozgás** költsége **0** (ha a művelet megegyezik).



azt mondjuk, hogy a költség végtelen

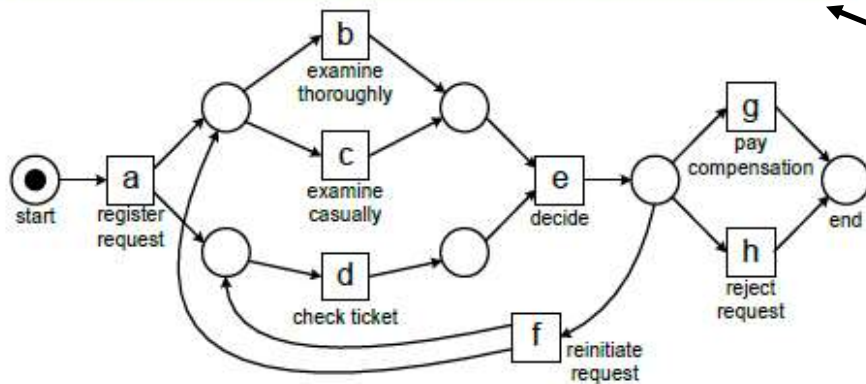
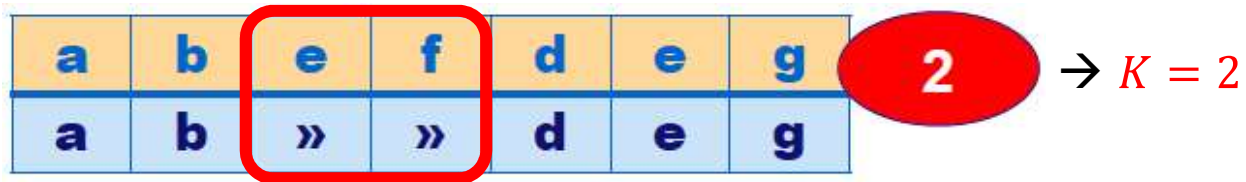


Fitness érték számítása

Illesztés alapú egyezőség-vizsgálat esetén

Példa egy trace **fitness értékének** kiszámítására:

$$t = \langle a, b, e, f, d, e, g \rangle \rightarrow M = 7$$

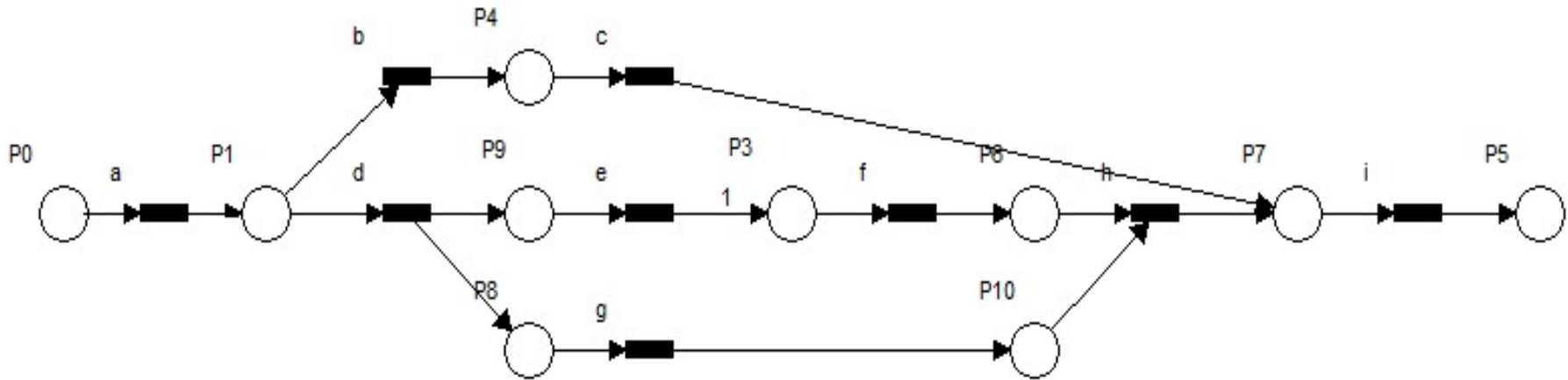


legrövidebb útvonal a modellben → N = 5

$$f(M, t) = 1 - \frac{K}{M + N} = 1 - \frac{2}{7 + 5} = 0,833$$

Számolja ki az **egyezőség mértékét** az illesztés alapú módszer segítségével az alábbi log és modell alapján az egyes tracekre és a teljes logra is!

$$L = [\langle a, b, c, h, i \rangle^5, \langle a, b, e, f, g, h, i \rangle^4, \langle a, d, e, g, f, h, i \rangle]$$

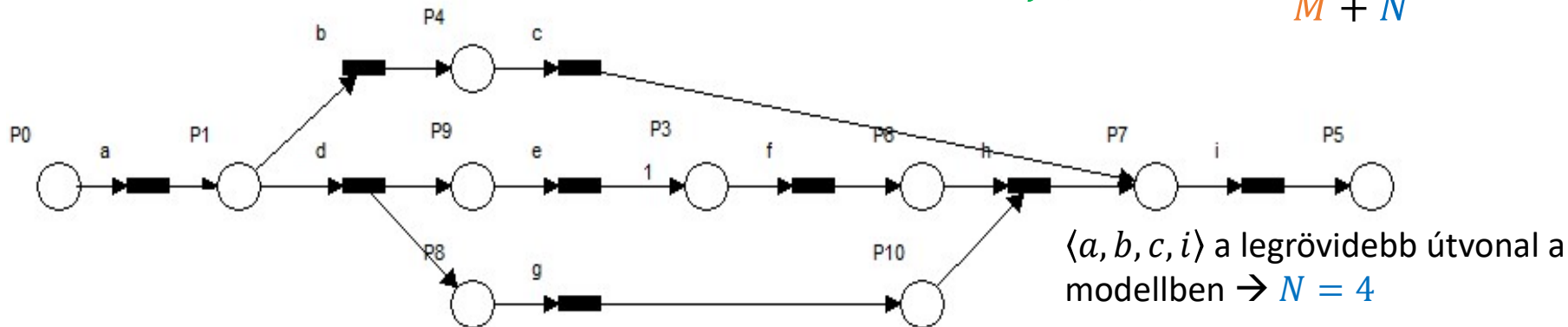


$$f(M, t) = 1 - \frac{K}{M + N}$$

$$f(M, L) = 1 - \frac{\sum_{i=1}^k n_i K_i}{\sum_{i=1}^k n_i (M_i + N_i)}$$

Számolja ki az **egyezőség mértékét** az illesztés alapú módszer segítségével az alábbi log és modell alapján!

$$L = [\langle a, b, c, h, i \rangle^5, \langle a, b, e, f, g, h, i \rangle^4, \langle a, d, e, g, f, h, i \rangle] \quad f(M, t) = 1 - \frac{K}{M + N}$$



t_1	a	b	c	h	i
M	a	b	c	>>	i

$$K = 1 \quad M = 5$$

$$f(M, t_1) = 1 - \frac{1}{5 + 4} = 1 - \frac{1}{9} = \frac{8}{9}$$

t_2	a	b	>>	e	f	g	h	i
M	a	>>	d	e	f	g	h	i

$$K = 2 \quad M = 7$$

$$f(M, t_2) = 1 - \frac{2}{7 + 4} = 1 - \frac{2}{11} = \frac{9}{11}$$

t_3	a	d	e	g	f	h	i
M	a	d	e	g	f	h	i

$$K = 0 \quad M = 7$$

$$f(M, t_3) = 1 - \frac{0}{7 + 4} = 1 - \frac{0}{11} = 1$$

$$f(M, L) = 1 - \frac{\sum_{i=1}^k n_i K_i}{\sum_{i=1}^k n_i (M_i + N_i)} = 1 - \frac{5 * 1 + 4 * 2 + 1 * 0}{5 * 9 + 4 * 11 + 1 * 11} = 1 - \frac{13}{100} = \mathbf{0,87}$$

Összehasonlíthatjuk az eseményeket nagyon finom szinten, ha a műveleteken kívül más adatok is rendelkezésre állnak az eseményhez kapcsolódóan

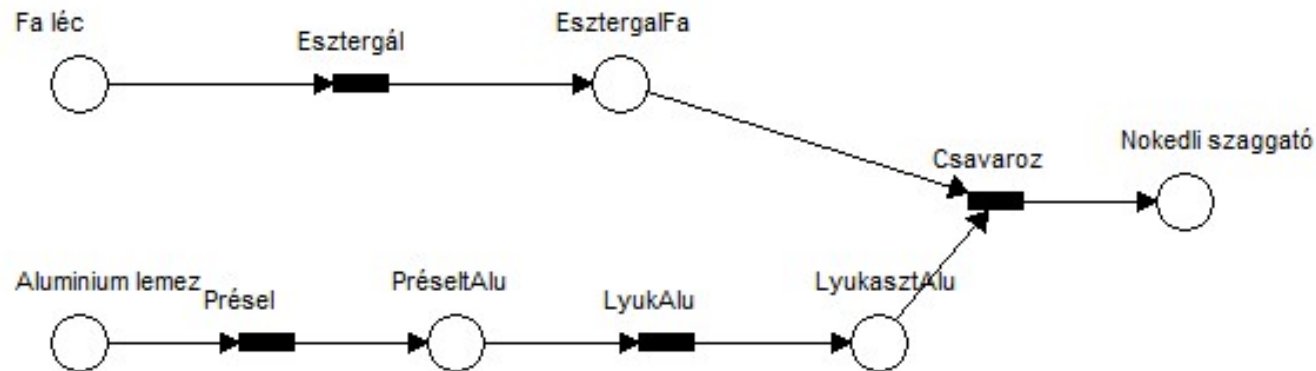
- Bármilyen költséget számolhatunk
- Számolhatunk kockázatot

...	Levél küldés (János, március. 8., 50e Ft)	...
...	Email küldés (Patrícia, március. 22., 20e Ft)	...

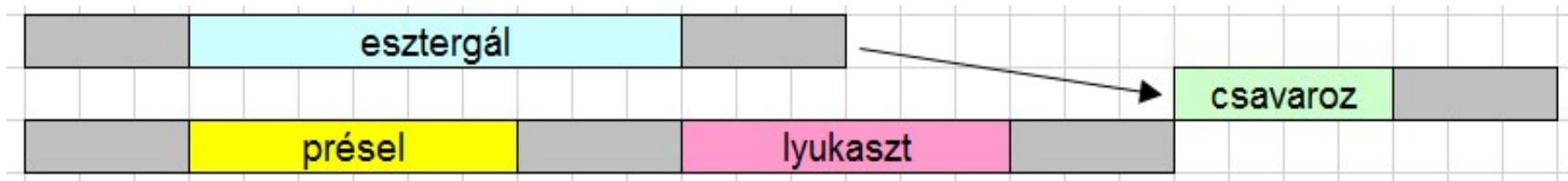
- Pl. megállapíthatjuk, hogy rossz személy hajtotta végre a tevékenységet, nem emailt küldött, hanem postai levelet, rossz összeget közölt az ügyféllel
- stb.

A konformancia elemzés alternatív megközelítései

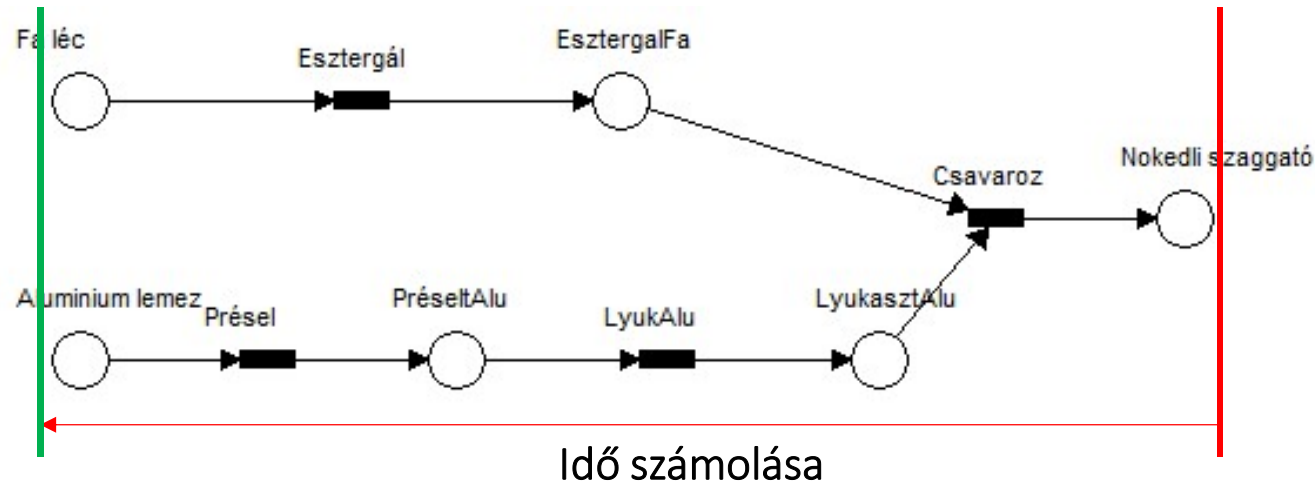
- Tekintsük az alábbi **gyártási folyamatot**:



- A folyamat lényege, hogy mindkét részfolyamatnak el kell készülnie mielőtt összecsavarozásra kerülne a termék.
- Másfajta szemléltetése a folyamatnak: A **Gantt-diagramon** a szürke részek az egyes műveletek közötti szállítást jelölik, ez a Petri-hálóba nem jelenik meg.



Petri háló kiértékelése

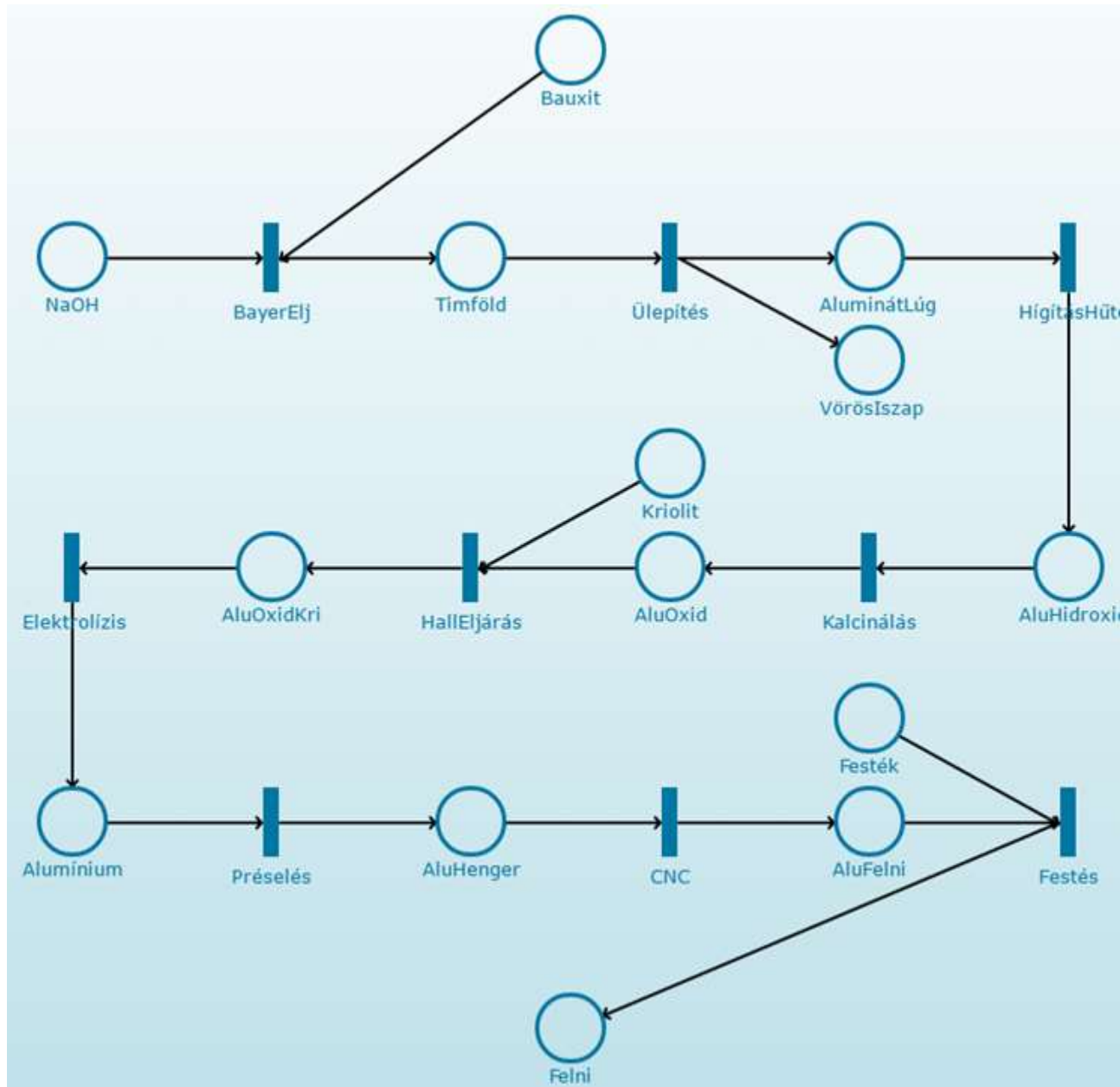


- Az algoritmus a készterméktől indulva számolja ki, hogy az egyes félkész munkadaraboknak a kezdéstől számítva **mennyi idő alatt** kell készen lenniük.
- A célállapotból kiindulva **rekurzívan** kiszámoljuk az addig eltelt időt.
- Amennyiben egy átmenetbe több helyről jön él (pl.: csavaroz), akkor a bejövő élekből a **legtöbb időt igénylőt választjuk**, mert nem lehet elindítani a műveletet, amíg a legtöbb időt igénylő előfeltétel nem készült el.

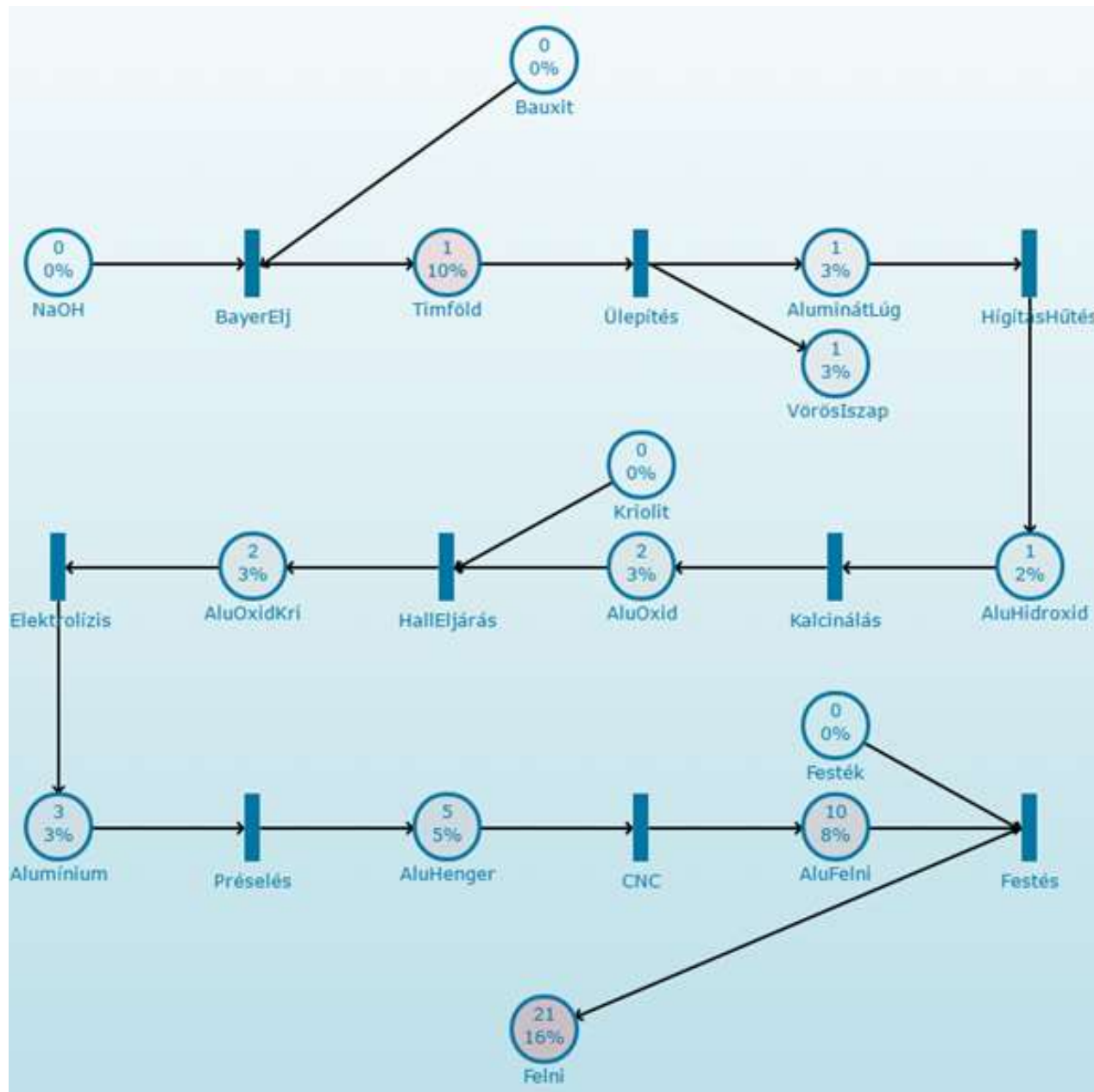
Eltérések keresése

- **Időbeli eltéréseket** keresünk jelen állapotban (vizsgálható lehet még a költség vagy az erőforrások).
- Az eltérések kereséséhez egy olyan **log fájlt** kell feldolgozni, amely **idő adatokat** is tartalmaz. A fájl az egyes **műveletek kezdeti és befejezési idejét** tartalmazza.
- Ezeket az időket megfeleltetjük a **hálóban** az **átmenet előtti és az azt követő hely idejével**, majd **letranszformáljuk** úgy, hogy a folyamat nullával kezdődjön.
- **Például**, ha a log fájlban az esztergálás 2021-01-13 11:39:32-től 2021-01-13 11:39:56-ig (**24 másodpercig**) tart, akkor a transzformálás után a *Fa* hely 0 időértéket, az *EsztergaltFa* hely pedig **24 értéket** kap.
- Ha ezt az egész **háló minden helyére** elvégeztük, akkor hasonló adatszerkezetet kapunk mint a Petri-háló feldolgozásakor kaptunk. Ezen két adatszerkezetet összehasonlítva megkapjuk, hogy **a folyamat egyes állomásai időben hogy viszonyulnak a modellhez**.
- A rendszer ezeket **az eltéréseket színkódok segítségével jelölheti a Petri-hálón**, ezzel vizualizálva, hogy hol vannak komolyabb időbeli eltérések, illetve hol sikerült lefaragni a modell szerinti időből.

Alumínium felni készítés modellje



Az elemzés eredménye



- Az egyes helyeken megjelennek az **időbeli eltérések értékei és százalékos arányuk**.
- Bizonyos erősségű **vörös háttérzínnel** vannak a **helyek** a Petri-hálóban jelölve.
- **PI.** meg tudjuk nézni, hogy az adott műveletet **ki végezte** abban az időben és meg lehet vizsgálni, hogy **mi a lassulás oka**.

Többperspektívás és online megfelelés ellenőrzés

- A megfelelés ellenőrzési módszerek túlnyomó többsége **offline** és csak a **control-flow perspektívára** (azaz csak az események sorrendjére) fókuszál.
- Az **online megfelelés ellenőrzés** alkalmazásával az eltéréseket hamarabb lehet detektálni, így a helyreállítási intézkedéseket akár a folyamat lefutásának befejeződése előtt végre lehet hajtani.

Bemenetei: **eseményfolyam** és folyamat modell

- Mivel nem csak az események sorrendjében lehet eltérés, ezért szükség van **többperspektívás megfelelés ellenőrzési** megoldásokra is, amelyek más perspektívákat (pl.: adat vagy erőforrás) is figyelembe vesz.

Bemenetei: log és **többperspektívát figyelembe vevő folyamat modell** (pl. adat Petri-háló)

- Még nincs kiforrott egyszerre **többperspektívás** és **online megfelelés ellenőrzési** algoritmus

Bemenetei: **eseményfolyam** és **többperspektívát figyelembe vevő folyamat modell**

Eseményfolyam

- csak **control-flow** perspektíva

eset azonosító × tevékenység azonosító

..., (1352, *a*), (1351, *b*), (1351, *c*), ...



- **control-flow** perspektíva + más perspektívák

eset azonosító × tevékenység azonosító × az esemény attribútumainak értékei

..., (1352, *a*, (Bas, 20/10/26 10:45)), (1351, *b*, (Daniel, 20/10/26 11:11)), ...

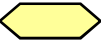
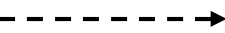






Naplófájlba rögzíthető:

Eset	Tevékenység	Erőforrás	Időbélyeg
...
13152	create account (<i>a</i>)	Bas	20/10/26 10:45
13151	submit order (<i>b</i>)	Daniel	20/10/26 11:11
13151	request quote (<i>c</i>)	Daniel	20/10/26 11:39
...

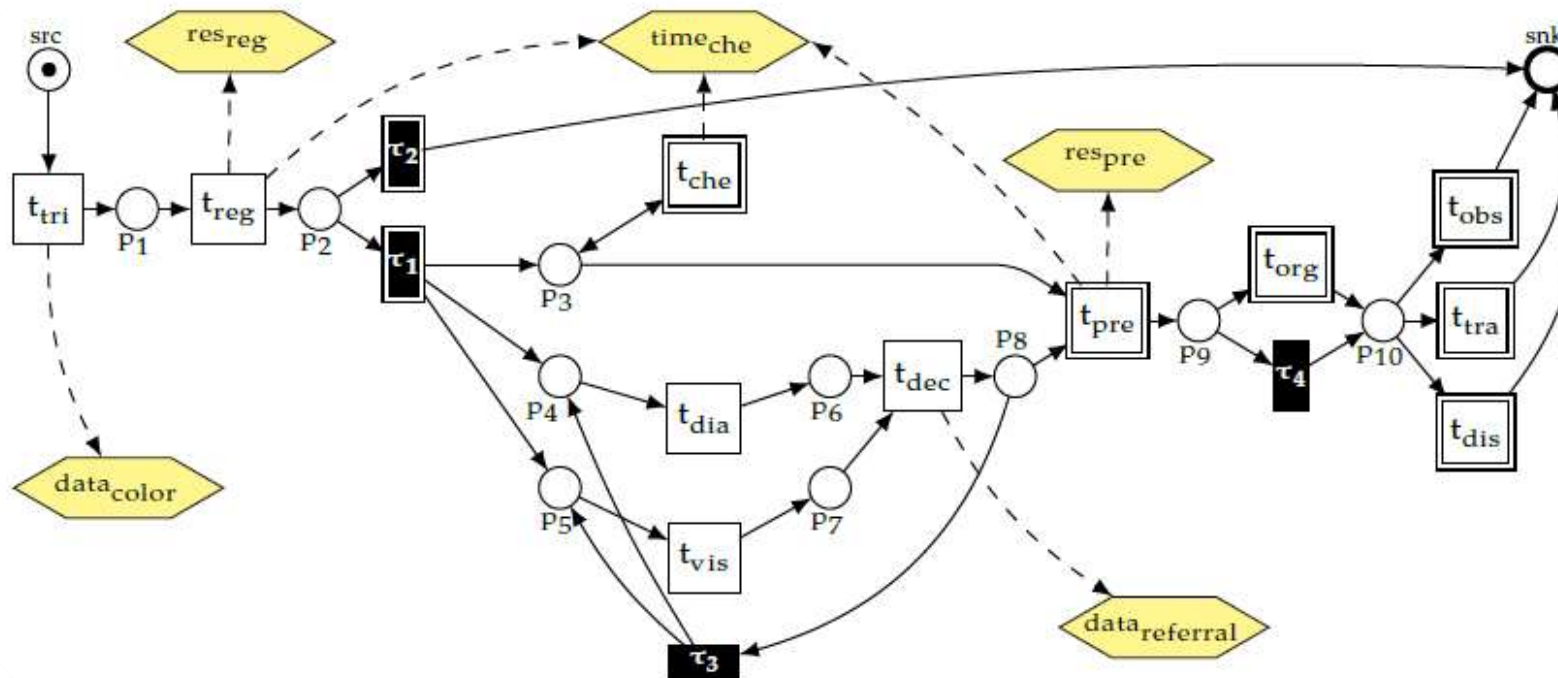
Adat Petri-háló (Data Petri Net, DPN)

Fontos fogalmak:

- változó (v), prime változó (v') 
- írás művelet 
- átmenet adatfüggő őrrrel   
- láthatatlan átmenet 

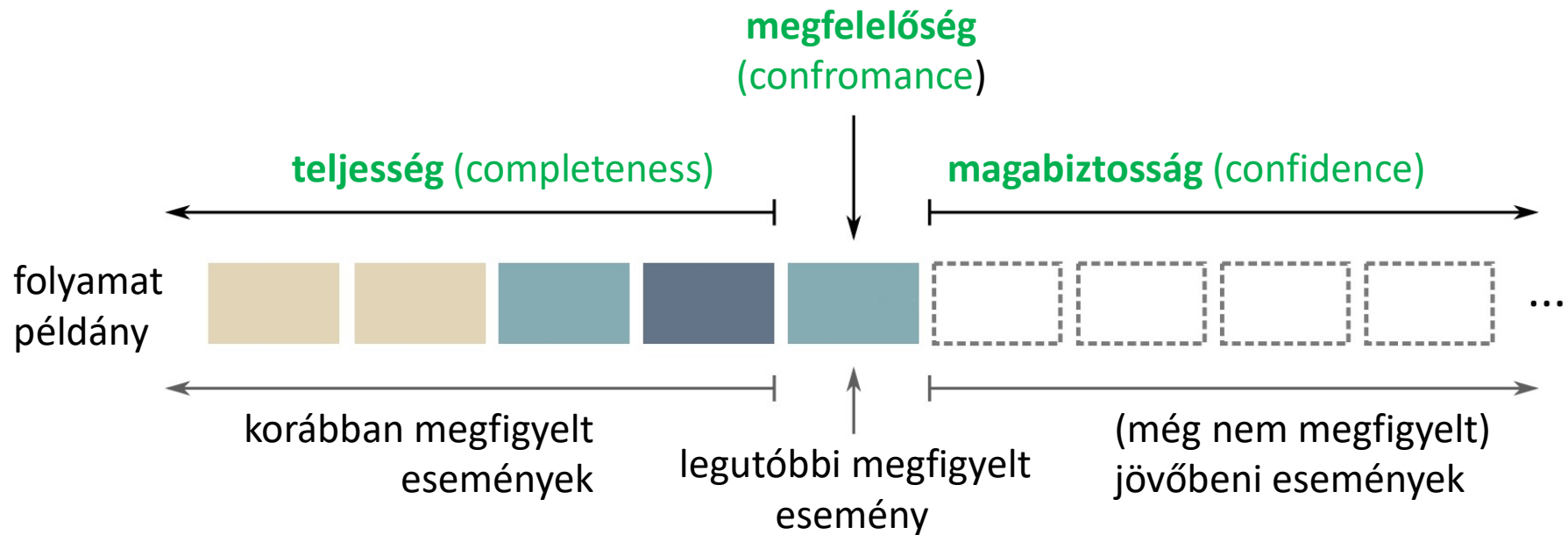
Transition	Guard expression
τ_1	$data_{color} \neq White$
τ_2	$data_{color} = White$
t_{che}	$time'_{che} \leq (time_{che} + 1h)$
t_{pre}	$time'_{che} \leq (time_{che} + 1h) \wedge res_{reg} = res'_{pre}$
t_{org}	$data_{referral} = Tertiary$
t_{obs}	$data_{referral} \neq Home$
t_{tra}	$data_{referral} \neq Home$
t_{dis}	$data_{referral} = Home$

örkifejezések



Online megfelelés ellenőrzés

Fő kihívás: a lefutás a vizsgálatkor még nem befejezett, az eleje és/vagy a vége hiányozhat



3 megfeleléségi mutató:

- **megfelelés:** a megfigyelt jó viselkedések aránya az összes viselkedéshez képest
- **teljesség:** azt jelzi, hogy az adott folyamat végrehajtás a kezdettől lett-e megfigyelve
- **magabiztosság:** azt jelzi, hogy az adott folyamat végrehajtás milyen közel van a befejezéshez

Többperspektívás online megfelelés ellenőrzés

Többperspektívás illesztés alapú módszert használ:

- Lehetséges mozgások:
 - **helyes szinkron mozgás** → a modell változó értékei megegyeznek az esemény attribútum értékeivel
 - **helytelen szinkron mozgás** → a modell változó értékei nem egyeznek meg az esemény attribútum értékeivel
 - **log mozgás**
 - **modell mozgás**
- Két költségfüggvényt használ:
 - tevékenység-illesztés költségfüggvénye
 - változó-illesztés költségfüggvénye → az egyes változó/attribútum értékeknek különböző súlyt lehet adni (pl. az súlyosabb probléma lehet, ha nem megfelelő személy végezte el a tevékenységet, mint az, hogy az esemény hosszabb ideig tartott az elvártnál)

Kihívások:

- a lefutások a vizsgálatkor még nem befejezettek
- a többperspektívás illesztés alapú módszer használata számítás igényes lehet
→ cache használata

Többperspektívás online megfelelés ellenőrzés

Gyártási folyamat példa

Több lefutásra is a következő kimenetet kapjuk:

	művelet	t_{s10}	t_{s20}	...	t_{s70}	t_{s80}
log rész	attribútumok	-	-	...	-	qib = 1 tray_row = 0 tray_column = 0
	átmenet	t_{s10}	t_{s20}	...	t_{s70}	t_{s80}
folyamat-modell rész	írt változók	-	-	...	-	qib = 1 tray_row = 1 tray_column = 1



Jelentése: A késztermékek egy része nem kerül tálcára a 8. állomás végén.

Kérdés: A késztermékek eldobásra kerülnek? Vagy csak hibásan érzékeli és/vagy rögzíti a rendszer? Ez helyszíni vizsgálatot igényel!

Megfelelőség ellenőrzés a ProM-ben

Online megfelelés ellenőrzés

- Nincs elérhető plugin!
(A megvalósított verzió pm4py folyamatbányászati platformban érhető el.)

Több perspektívás megfelelés ellenőrzés

- Kiegyensúlyozott több perspektívás megfelelés ellenőrzési módszer
(Balanced Multi-perspective Conformance Checking, rövidítve: BMCC)
- A BMCC megvalósított verziója “**Conformance checking of DPN**” néven érhető el ProM 6.9-es pluginként, a “**DataAwareReplayer**” nevű csomag részeként.

Példa, ami betölthető a ProM-be: Lfull.mxml betöltése

The screenshot shows the ProM UI with the 'Workspace' area. A resource named 'L-conf XLog' is selected and highlighted in green. A red arrow points to this resource. A tooltip is visible over the resource, displaying the following log content:

```
455x Case1 a c d e h
191x Case2 a b d e g
177x Case3 a d c e h
144x Case4 a b d e h
111x Case5 a c d e g
82x Case6 a d c e g
56x Case7 a d b e h
47x Case8 a c d e f d b e h
38x Case9 a d b e g
33x Case10 a c d e f b d e h
14x Case11 a c d e f b d e g
11x Case12 a c d e f d b e g
9x Case13 a d c e f c d e h
8x Case14 a d c e f d b e h
5x Case15 a d c e f b d e g
3x Case16 a c d e f b d e f d b e g
2x Case17 a d c e f d b e g
2x Case18 a d c e f b d e f b d e g
1x Case19 a d c e f d b e f b d e h
1x Case20 a d b e f b d e f d b e g
1x Case21 a d c e f d b e f c d e f d b e g
```

Ez a log tartalma

N1.tpn betöltése

ProM UITopia

ProM

Workspace

import...

sort by [refresh] [play] [stop] ABC

All

Favorites

Imported

Selection

N1.tpn
Petri net

L-conf
XLog

L-conf
XLog
created about a mi...
imported

Show parents

Show children

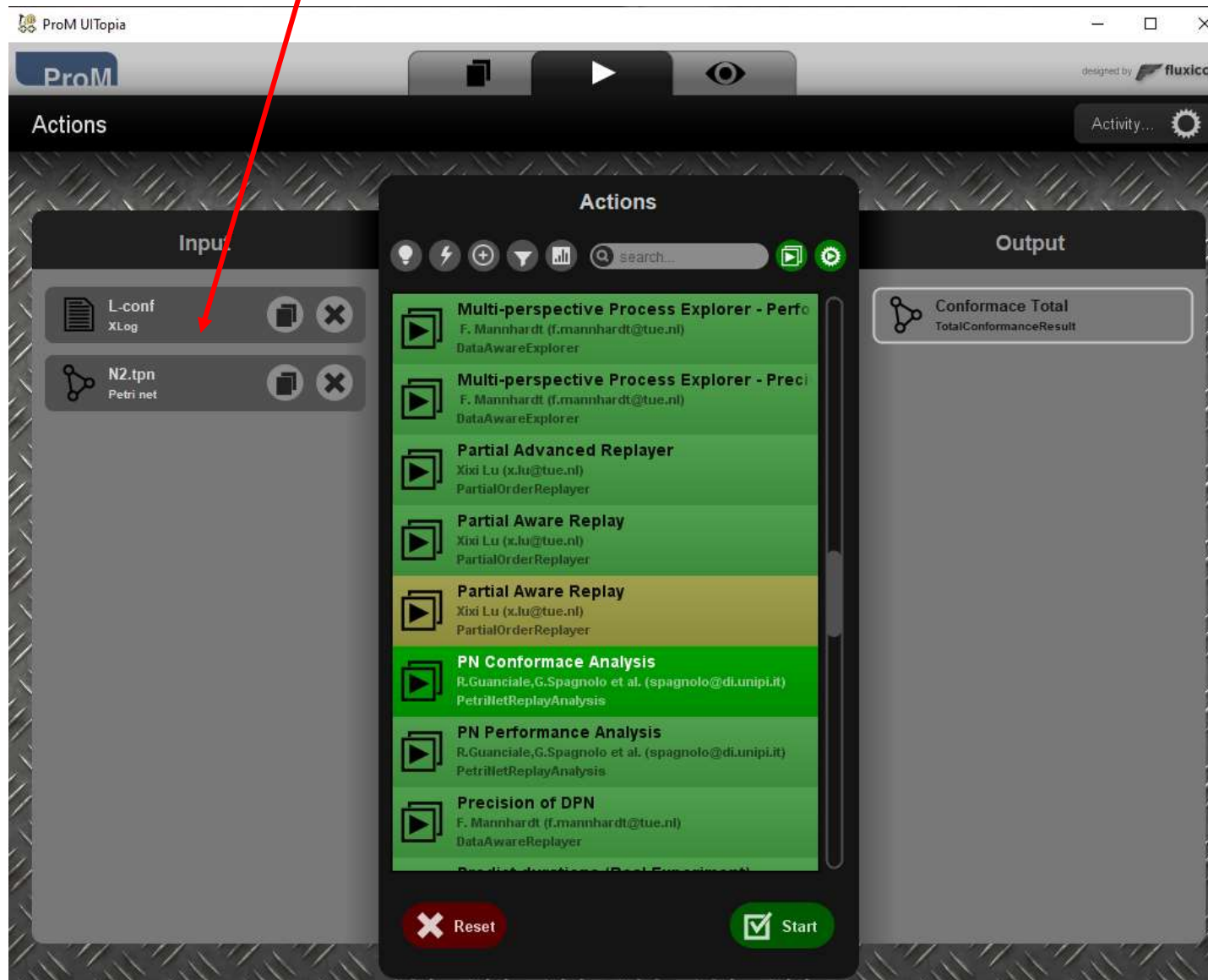
Rename resource

Export to disk

```
place "start" init 1;
place "p1";
place "p2";
place "p3";
place "p4";
place "end";
trans "a"~"a\ncomplete" in "start" out "p1";
trans "b"~"b\ncomplete" in "p1" out "p2" ;
trans "c"~"c\ncomplete" in "p1" out "p2" ;
trans "d"~"d\ncomplete" in "p2" out "p3" ;
trans "e"~"e\ncomplete" in "p3" out "p4" ;
trans "f"~"f\ncomplete" in "p4" out "p1" ;
trans "g"~"g\ncomplete" in "p4" out "end" ;
trans "h"~"h\ncomplete" in "p4" out "end" ;
```

Ez a modell felépítése

Ha mind a két fájl betöltésre került, akkor a **PN Conformance Analysis** plugint használhatjuk



Eredmények:
f=0,87
d 443 trace-nél

```
place "start" init 1;  
place "p1";  
place "p2";  
place "p3";  
place "p4";  
place "end";  
trans "a~"~"a\ncomplete" in "start" out "p1";  
trans "b~"~"b\ncomplete" in "p1" out "p2" ;  
trans "c~"~"c\ncomplete" in "p1" out "p2" ;  
trans "d~"~"d\ncomplete" in "p2" out "p3" ;  
trans "e~"~"e\ncomplete" in "p3" out "p4" ;  
trans "f~"~"f\ncomplete" in "p4" out "p1" ;  
trans "g~"~"g\ncomplete" in "p4" out "end" ;  
trans "h~"~"h\ncomplete" in "p4" out "end" ;
```

ProM UITopia

