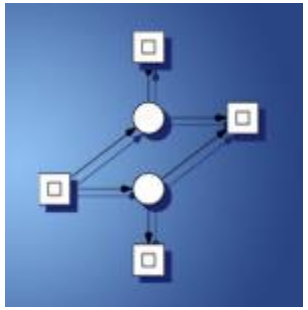
Petri-hálók analízise

dr. Bartha Tamás
dr. Majzik István
dr. Pataricza András
BME Méréstechnika és Információs Rendszerek Tanszék

Modellező és analízis eszközök: DNAnet, Snoopy, PetriDotNet

UNIVERSITY OF CAPE TOWN
Department of Computer Science



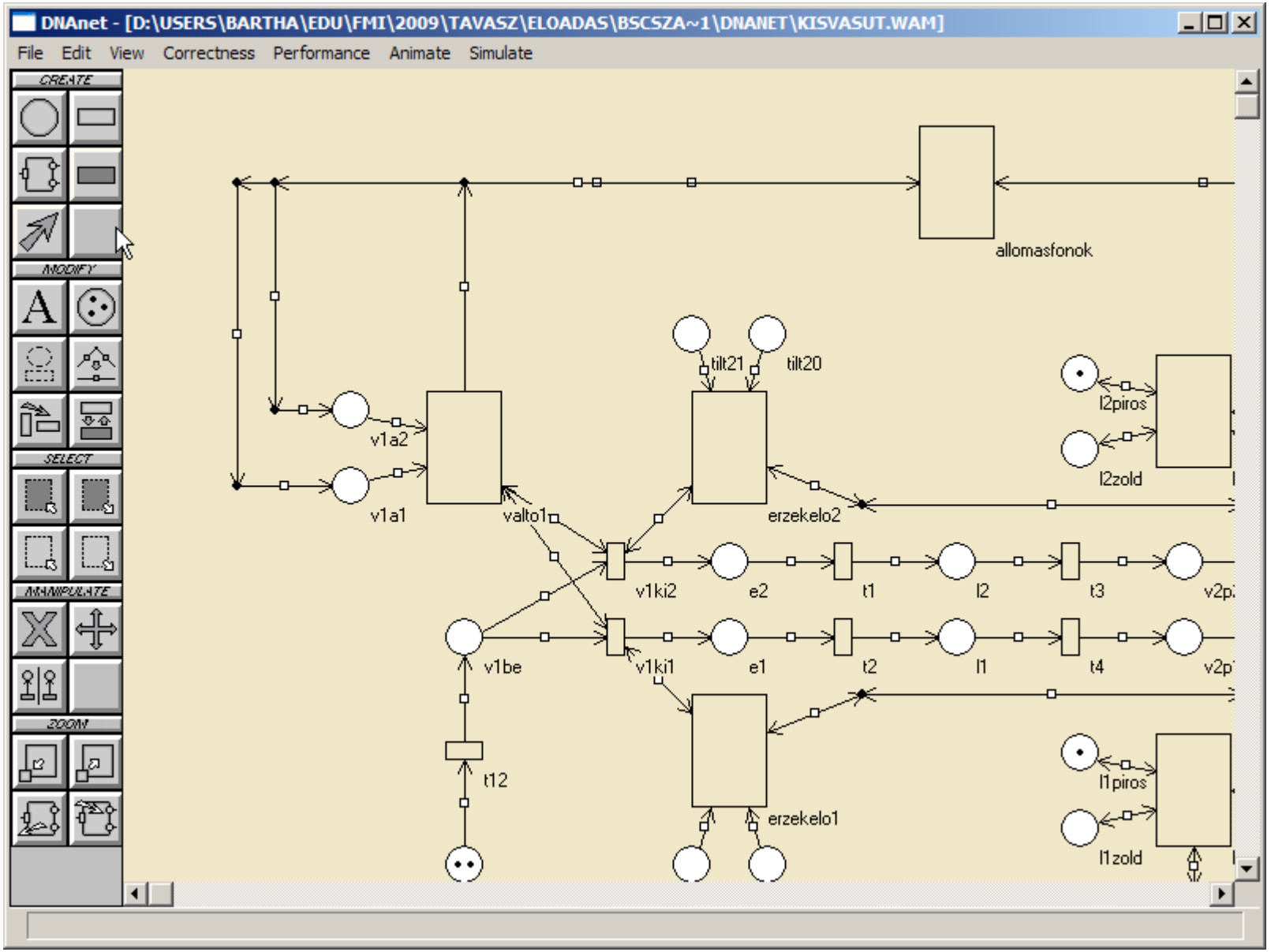
b-tu
Brandenburgische
Technische Universität
Cottbus

petridotnet
from BME-MIT

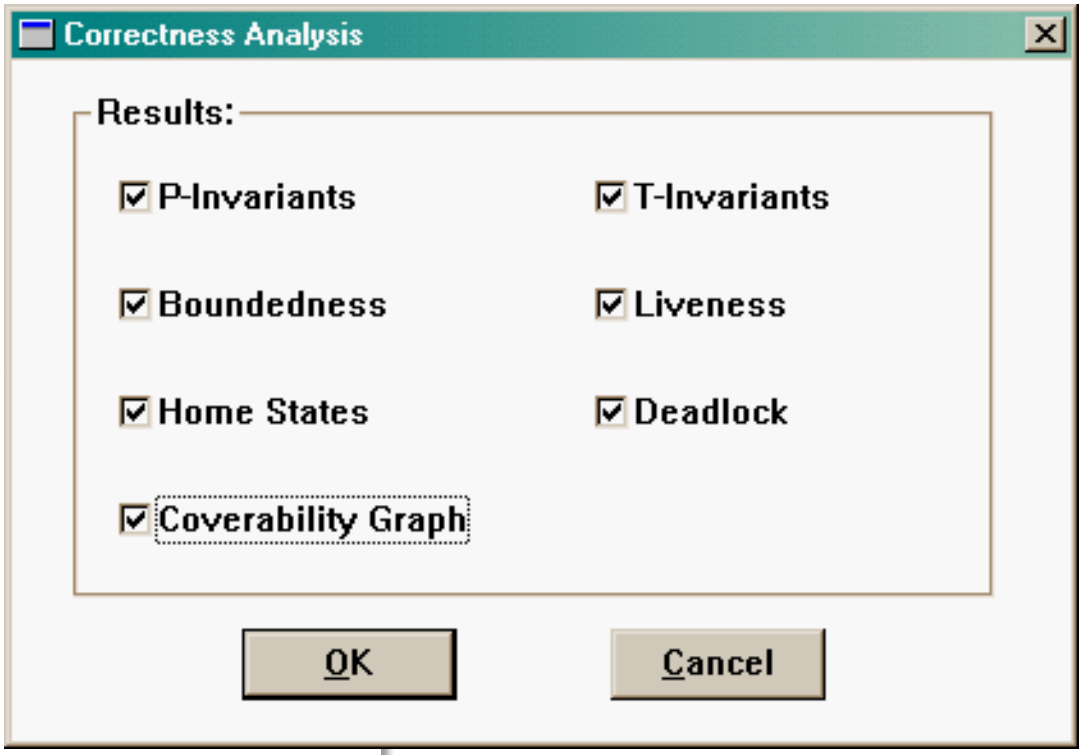
A DNAnet modellező program

- Képességei
  - Grafikus szerkesztő (régi Windowsra)
  - Interaktív animáció (token game)
  - Nem interaktív szimuláció (teljesítmény analízishez)
  - Analízisek: egyes dinamikus és statikus tulajdonságok ellenőrzése
- Előnyei
  - Kicsi, kompakt, gyors, egyszerűen kezelhető
  - Méretéhez képest sokat tud
  - Ingyenes, szabad felhasználású
- Hátránya
  - Stabilitás, új Windowson futtatás problémás

A DNAnet modellező program képe

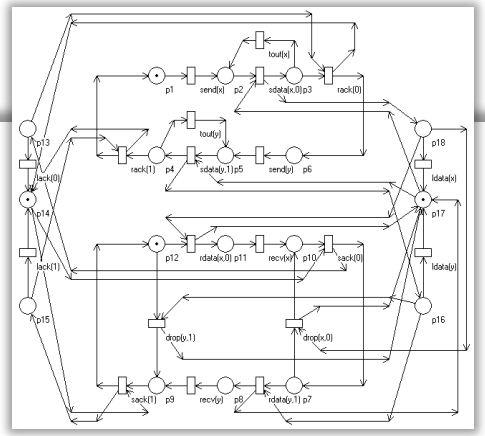


A DNAnet analízis képességei




```
Net is bounded.

Deadlock is not possible.

Net is live.

Net has home states.

Coverability graph generation statistics:
108 unique markings
1 strongly connected components
108 hash table entries used
1 was the longest hash list length
1 was the average hash list length
27 was the maximum stack height
3 was the maximum component stack height
```

A DNAnet invariáns keresése



P-invariants:

1	0	0	0	(main.p1)
1	0	0	0	(main.p2)
1	0	0	0	(main.p3)
1	0	0	0	(main.p4)
1	0	0	0	(main.p5)
1	0	0	0	(main.p6)
0	1	0	0	(main.p7)
0	1	0	0	(main.p8)
0	1	0	0	(main.p9)
0	1	0	0	(main.p10)
0	1	0	0	(main.p11)
0	1	0	0	(main.p12)
0	0	1	0	(main.p13)
0	0	1	0	(main.p14)
0	0	1	0	(main.p15)
0	0	0	1	(main.p16)
0	0	0	1	(main.p17)
0	0	0	1	(main.p18)

ie.

M(main.p1) + M(main.p2) + M(main.p3) + M(main.p4) + M(main.p5) + M(main.p6)
M(main.p7) + M(main.p8) + M(main.p9) + M(main.p10) + M(main.p11) + M(main.p12)
M(main.p13) + M(main.p14) + M(main.p15)
M(main.p16) + M(main.p17) + M(main.p18)

All places are covered by P-invariants.

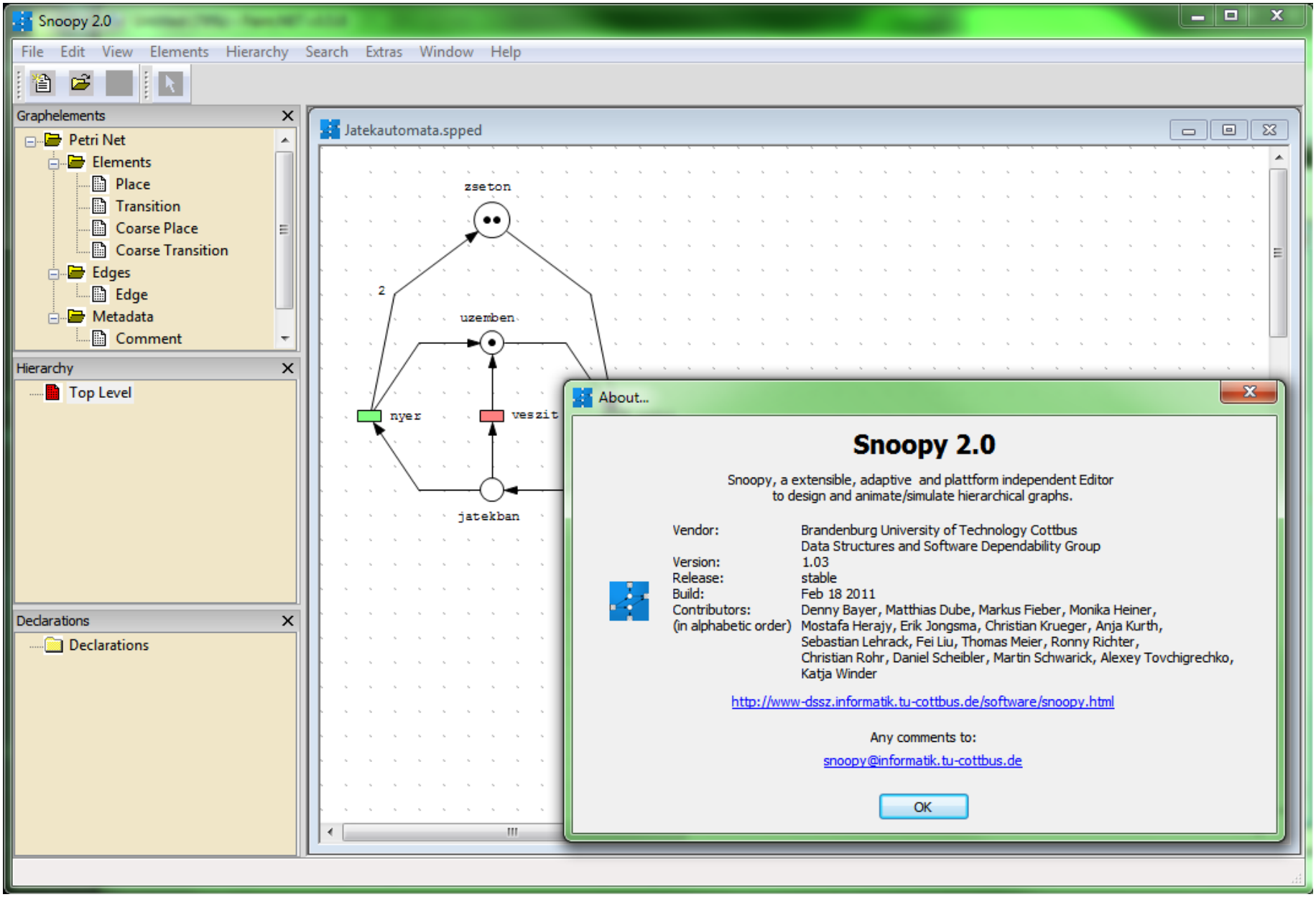
T invariants:

1	0	0	0	0	0	1	(main.send(x))
1	0	1	1	1	0	1	(main.sdata(x,0))
1	0	0	0	0	0	1	(main.rack(0))
1	0	0	0	0	0	1	(main.send(y))
1	1	0	0	1	1	1	(main.sdata(y,1))
1	0	0	0	0	0	1	(main.rack(1))
0	0	1	1	1	0	0	(main.tout(x))
0	1	0	0	1	1	0	(main.tout(y))
1	0	0	0	1	1	1	(main.sack(1))
1	0	0	0	1	0	0	(main.recv(y))
1	0	0	0	1	0	0	(main.rdata(y,1))
1	0	0	1	1	0	1	(main.sack(0))
1	0	0	0	1	0	0	(main.recv(x))
1	0	0	0	1	0	0	(main.rdata(x,0))
0	0	0	1	1	0	0	(main.lack(0))
0	0	0	0	1	1	0	(main.lack(1))
0	1	0	0	0	0	0	(main.ldata(y))
0	0	1	0	0	0	0	(main.ldata(x))
0	0	0	1	0	0	1	(main.drop(x,0))
0	0	0	0	0	1	1	(main.drop(y,1))

A Snoopy modellező program

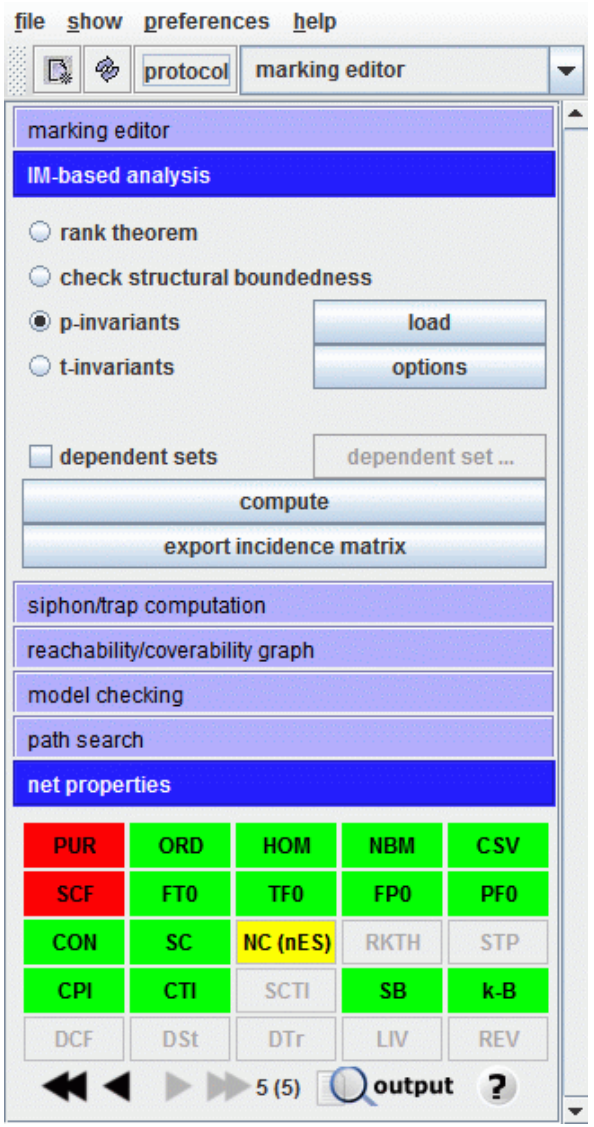
- Snoopy (Windows, Linux)
  - Grafikus szerkesztő + token game (animált)
  - Egyszerűen kezelhető
  - Kényelmi funkciók: copy / paste, undo / redo
  - Kiterjesztések: tiltó él, olvasó él, reset él, egyenlőség él
  - Számos háló típus, többek között színezett háló is
  - Támogatja hierarchikus Petri-hálók készítését
  - Elemek színezése, méretezése, élsúlyok kijelzése
  - On-line help
  - Külső analízis eszköz: Charlie (Java)

A Snoopy modellező program képe



Analízis eszközök Snoopy-hoz

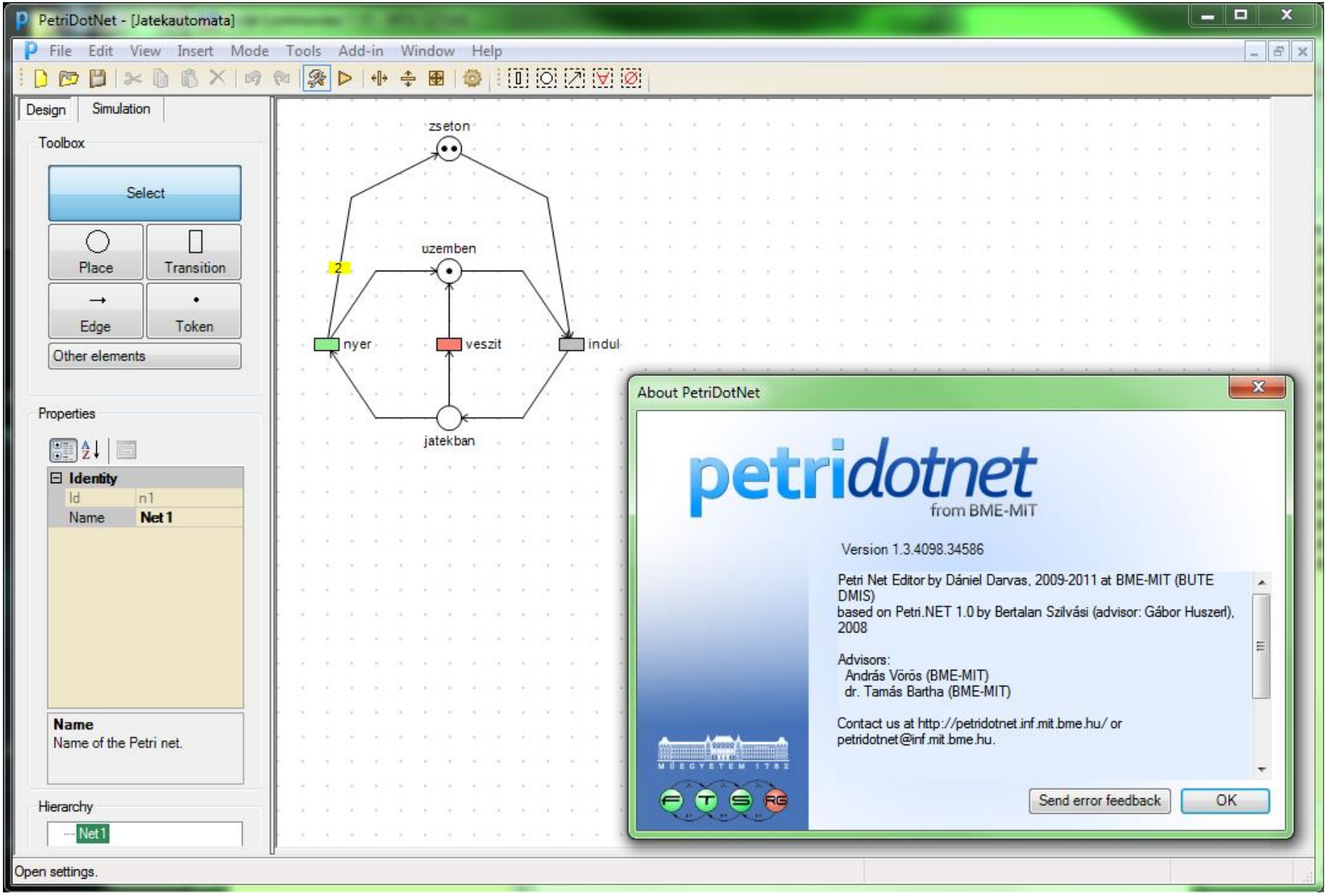
- Charlie (Java)
  - Dinamikus tulajdonságok, elérhetőség
  - Strukturális tulajdonságok, invariánsok
  - CTL és LTL modellellenőrző
- INA (Windows, Linux)
  - Szöveges felületű parancssori program
  - Invariáns analízis, elérhetőségi gráf generálás
  - Strukturális tulajdonságok ellenőrzése
  - Szimulációs képességei nincsenek



A PetriDotNet modellező program

- Alapképességei
  - Grafikus szerkesztő + token game + szimuláció
  - Kiterjesztések: tiltó él, időzítés, színezett háló
  - Szabványos PNML fájlformátum, van hozzá INA kimenet
  - Támogatja hierarchikus Petri-hálók készítését
- Analízis képességei (modulárisan bővíthető)
  - Dinamikus és strukturális tulajdonságok
  - CTL modellellenőrző
  - Elérhetőségi és fedési gráf
- BME MIT fejlesztés: petridotnet.inf.mit.bme.hu

A Petri.NET modellező program képe



A PetriDotNet analízis képességei

A(z) AlterBit háló tulajdonságai

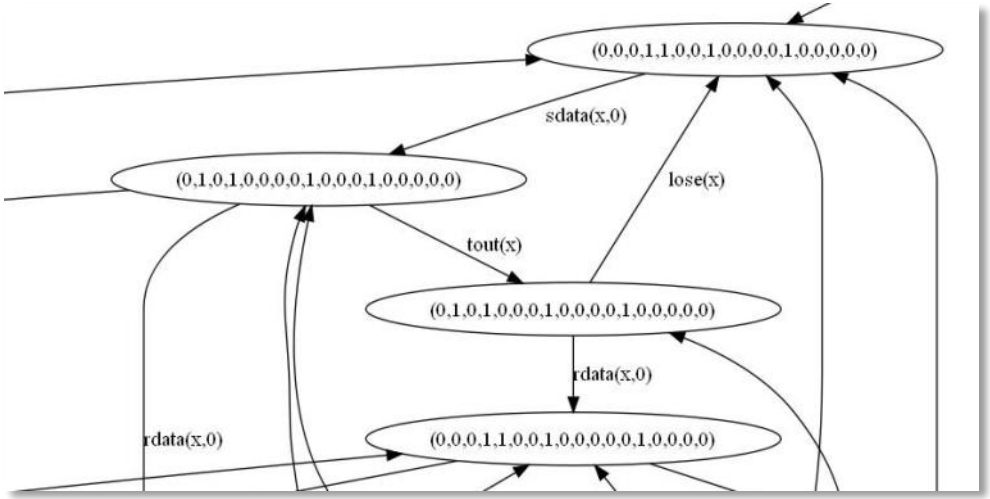
Dinamikus tulajdonságok

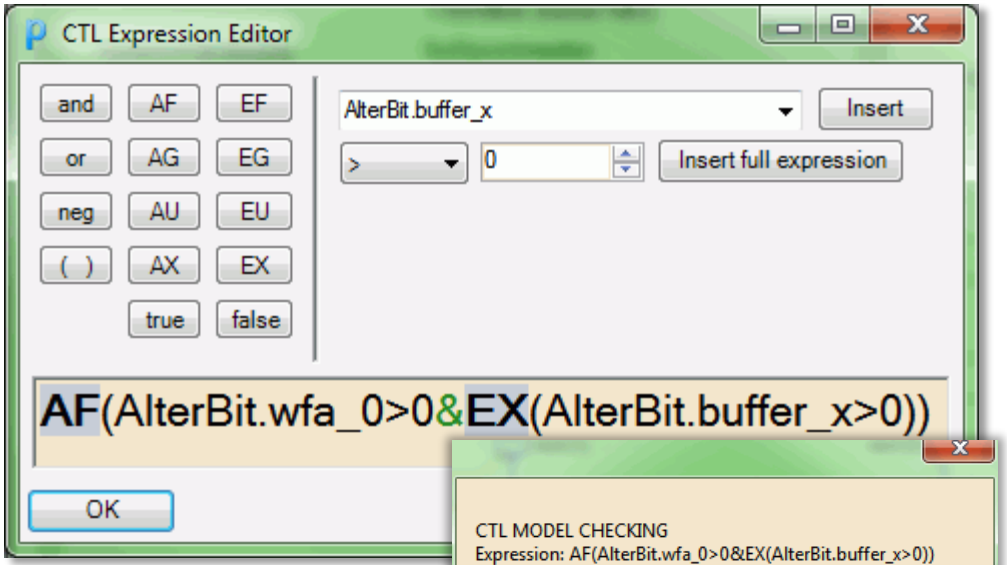
Állapotok száma:	108
Korlátosság:	**korlátos**
	1-korlátos (biztos háló)
Holtpontmentesség:	**holtpontmentes**
Megfordíthatóság:	**megfordítható**
Perszisztencia:	**nem perzisztens**

Strukturális tuladjonságok

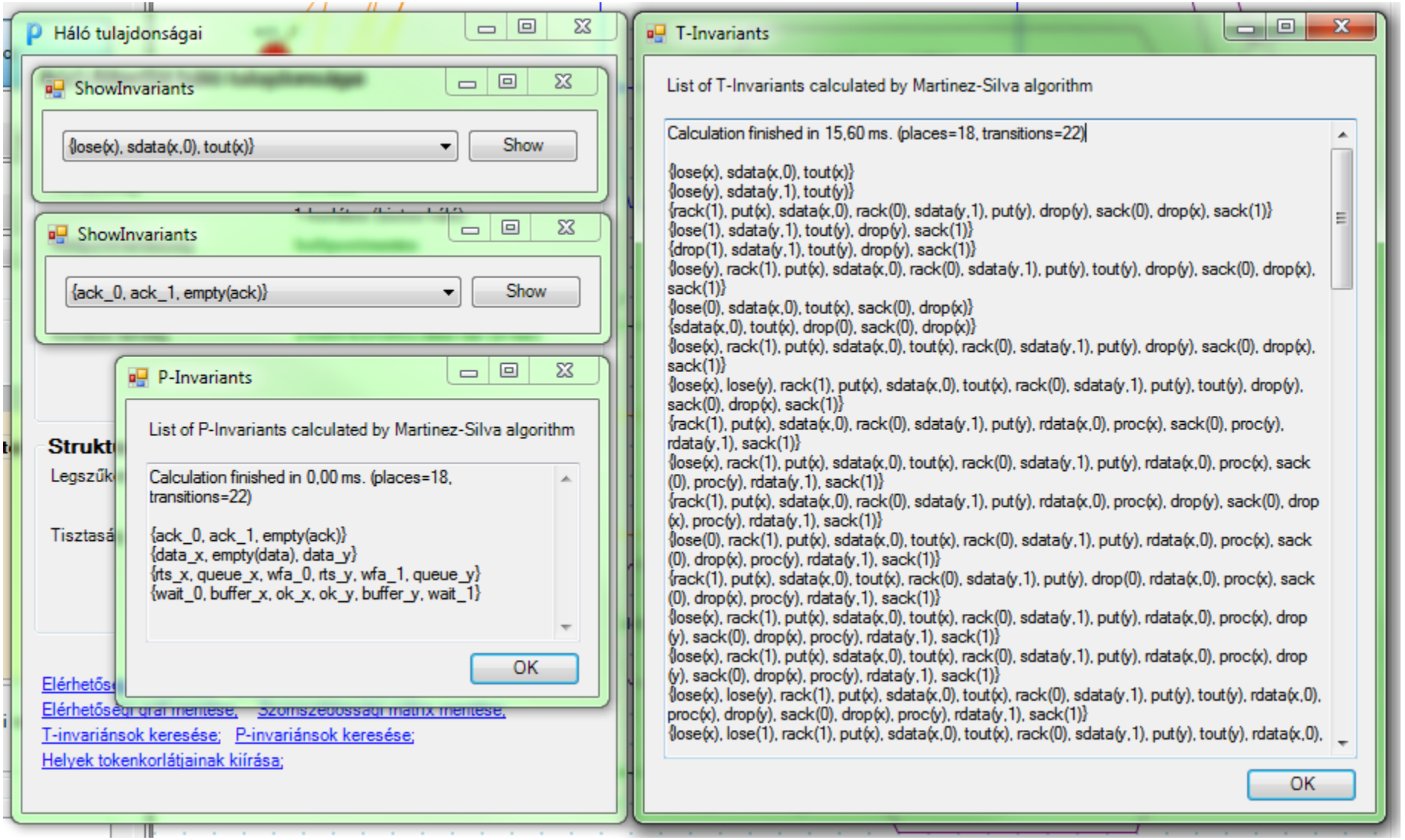
Legszűkebb alosztály:	Petri-háló
Tisztaság:	**nem tiszta (van hurokél)**

Elérhetőség vizsgálata; CTL-kifejezés vizsgálata;
Elérhetőségi gráf mentése; Szomszédossági mátrix mentése;
T-invariánsok keresése; P-invariánsok keresése;
Helyek tokenkorlátjainak kiírása;





A PetriDotNet invariáns analízis



Példa modell:
Az alternáló bit protokoll

A modellezési feladat

Alternating Bit Protocol

- Átviteli protokoll veszteséges csatornához
  - Üzenet elveszhet (véges számú alkalommal)
  - Üzenet tartalma nem változhat (ha változik, az detektálható és az üzenet elveszettnek tekinthető)
- Cél: a protokoll biztosítsa, hogy minden üzenet (véges számú próbálkozással) eljusson a vevőhöz

Küldő folyamat

- A küldő az üzenetekhez egy ellenőrző bitet kapcsol
- Az üzenetek megérkezését a vevőtől nyugta jelzi, ugyanazzal az ellenőrző bittel
- Hibakezelés a küldőnél
  - Ha a küldő időtúllepéssel észleli a nyugta hiányát $\rightarrow$ újra küldi az üzenetet
  - Ha a küldő egy b bittel ellátott nyugtát vár és ilyet kap, akkor a következő üzenethez ¬b bitet csatol
  - Ha a küldő egy b bittel ellátott nyugtát vár de nem ilyet kap, akkor egyszerűen eldobja a nyugtát
    (majd időtúllépés lesz a nyugta hiánya miatt)

Fogadó folyamat

- Az üzenet vételét nyugtázza a kapott ellenőrző bitet visszaküldve a nyugtával
- Hibakezelés a fogadónál:
  - Ha egy üzenetben az előző üzenethez képest nem negált ellenőrző bit érkezik, akkor
    - az üzenetet eldobja (nem dolgozza fel),
    - nyugtát küld a bejött ellenőrző bittel
      (arra számítva, hogy bizonyára az üzenet újraküldése történt az előző nyugta elvesztése miatt)

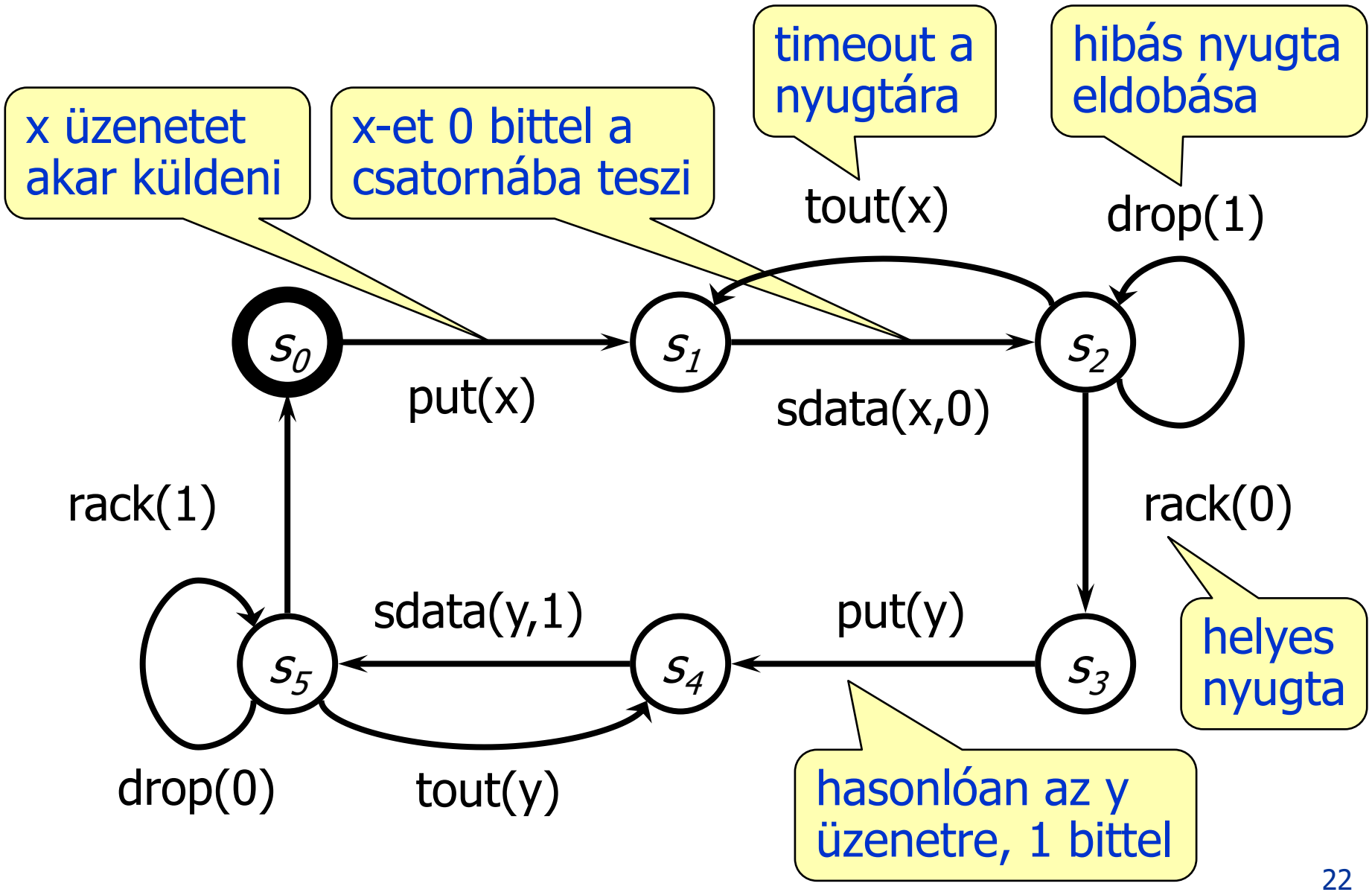
A modellalkotás lépései

1. A feladat felbontása folyamatokra és erőforrásokra
2. Folyamatok állapotainak meghatározása
3. Erőforrások állapotainak meghatározása
4. Állapot alapú modellekből Petri-háló modell készítése
5. Folyamatok és erőforrások modelljeinek integrálása
   - Folyamat lépése módosítja az erőforrás állapotát
   - Erőforrás állapota feltétel egy folyamat lépéséhez
6. Integrált modell helyességének ellenőrzése
7. Modell felhasználása a feladat megoldására

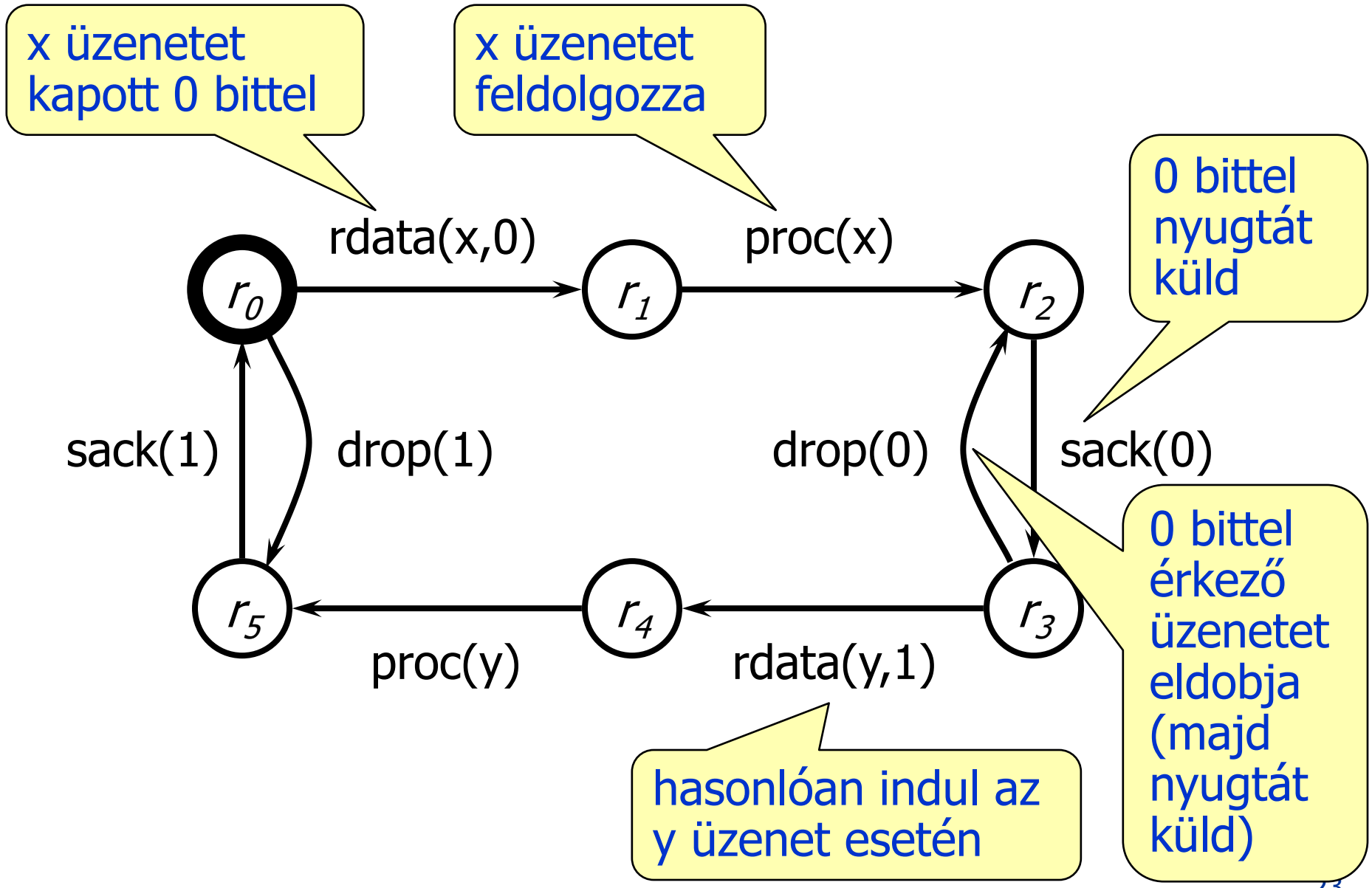
Komponensek és állapotaik

- Komponensek (alrendszerek)
  - Folyamatok: küldő folyamat, fogadó folyamat
  - Erőforrások: adat csatorna, nyugtázó csatorna
- Minden komponens saját állapotokkal rendelkezik
  - Informális állapotgráf: állapotok körökkel, állapotátmenetek az eseményekkel címkézett nyilakkal
- Azonos események egy időben mennek végbe: szinkronizáció
  - Pl. a fogadó megkapja az üzenetet és ez egyúttal a csatornából is kikerül

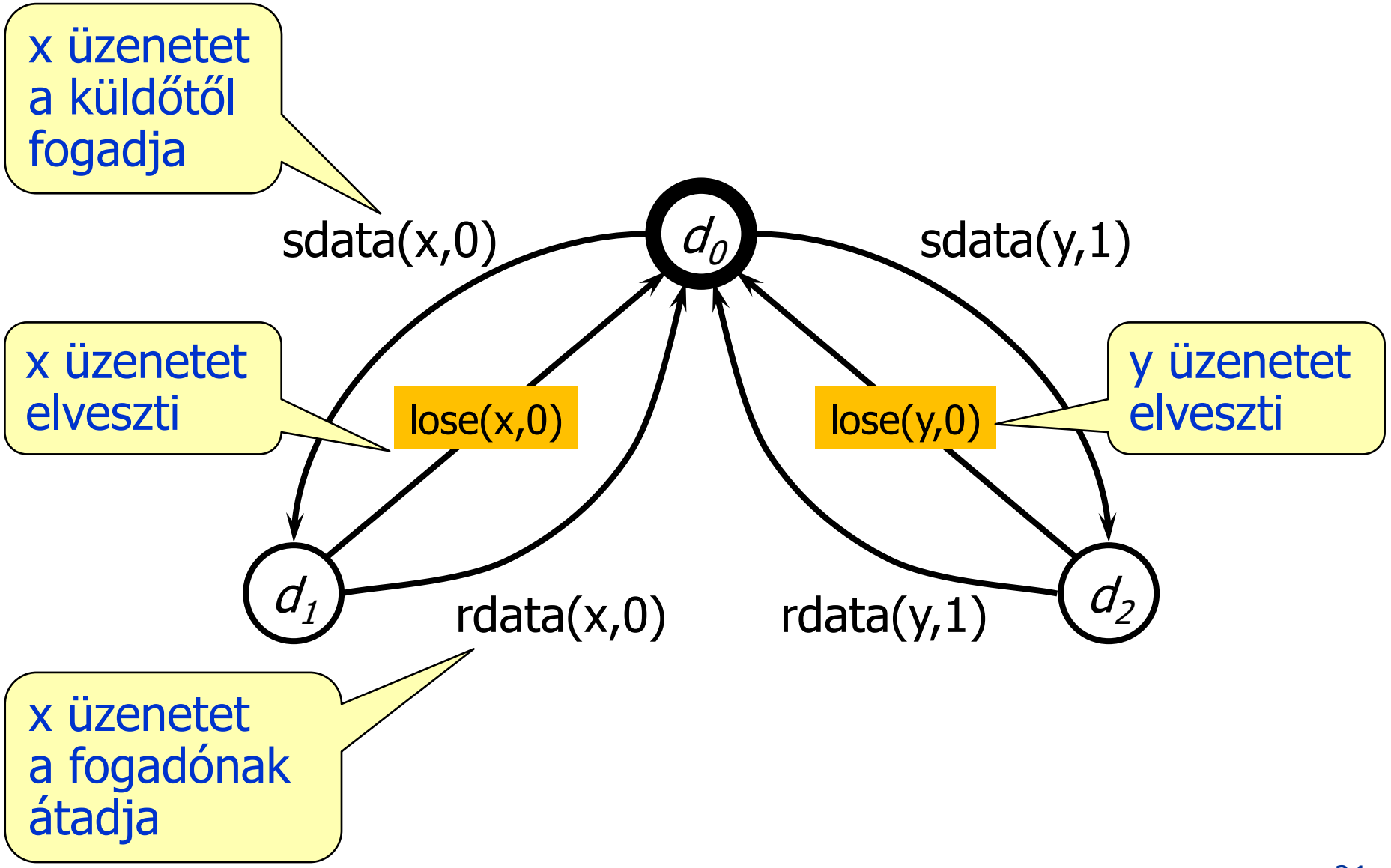
Küldő folyamat állapotgráfja



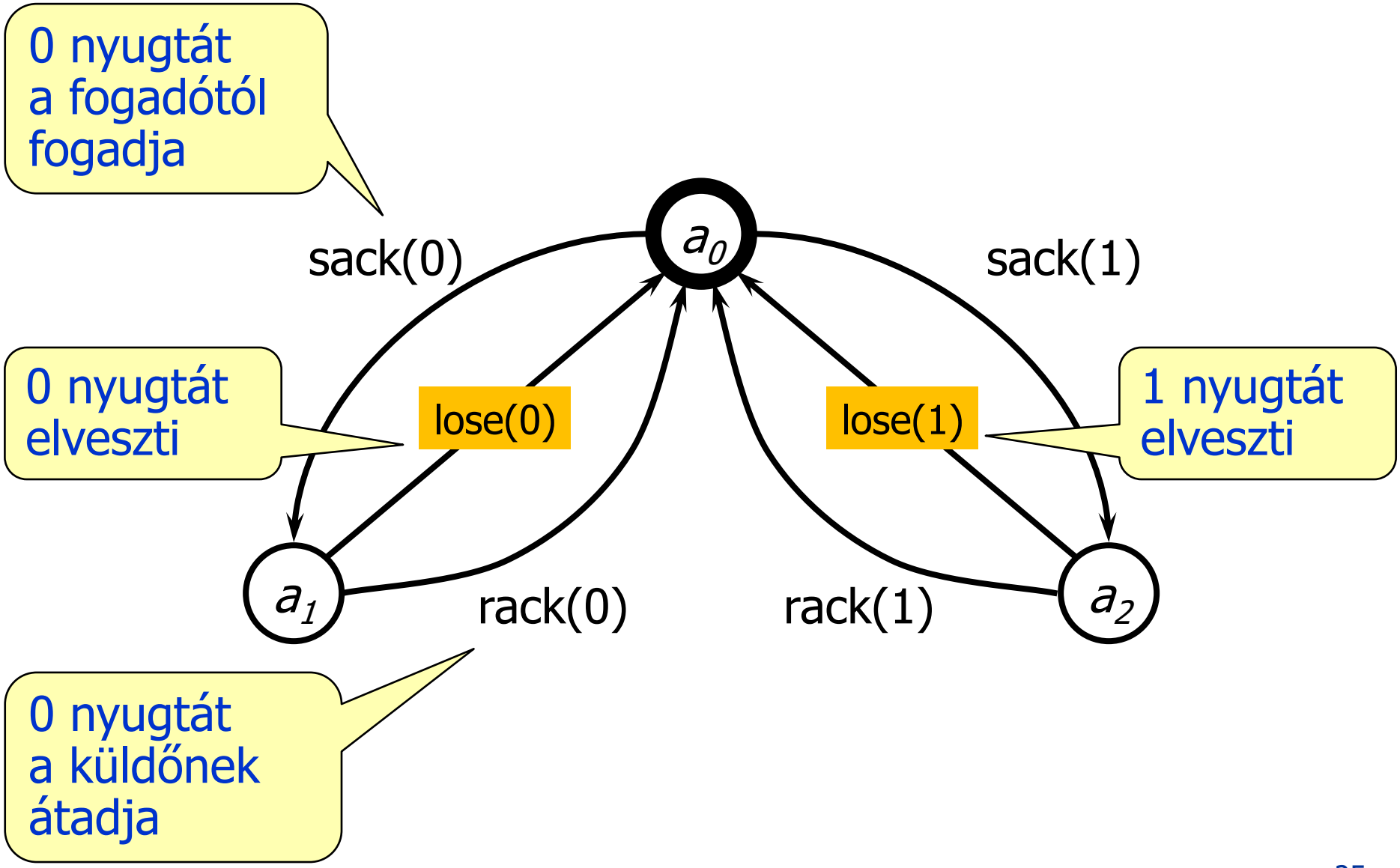
Fogadó folyamat állapotgráfja



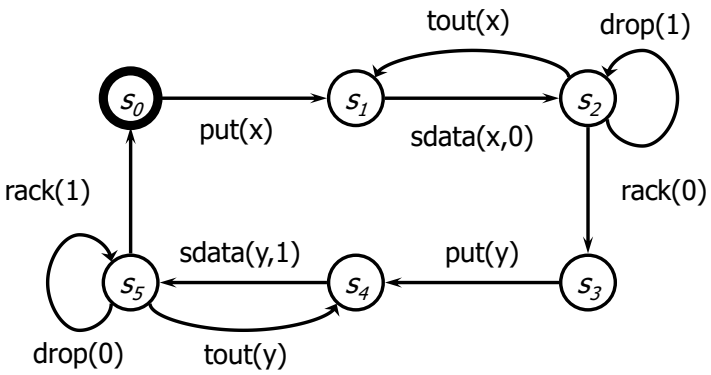
Adat csatorna állapotgráfja

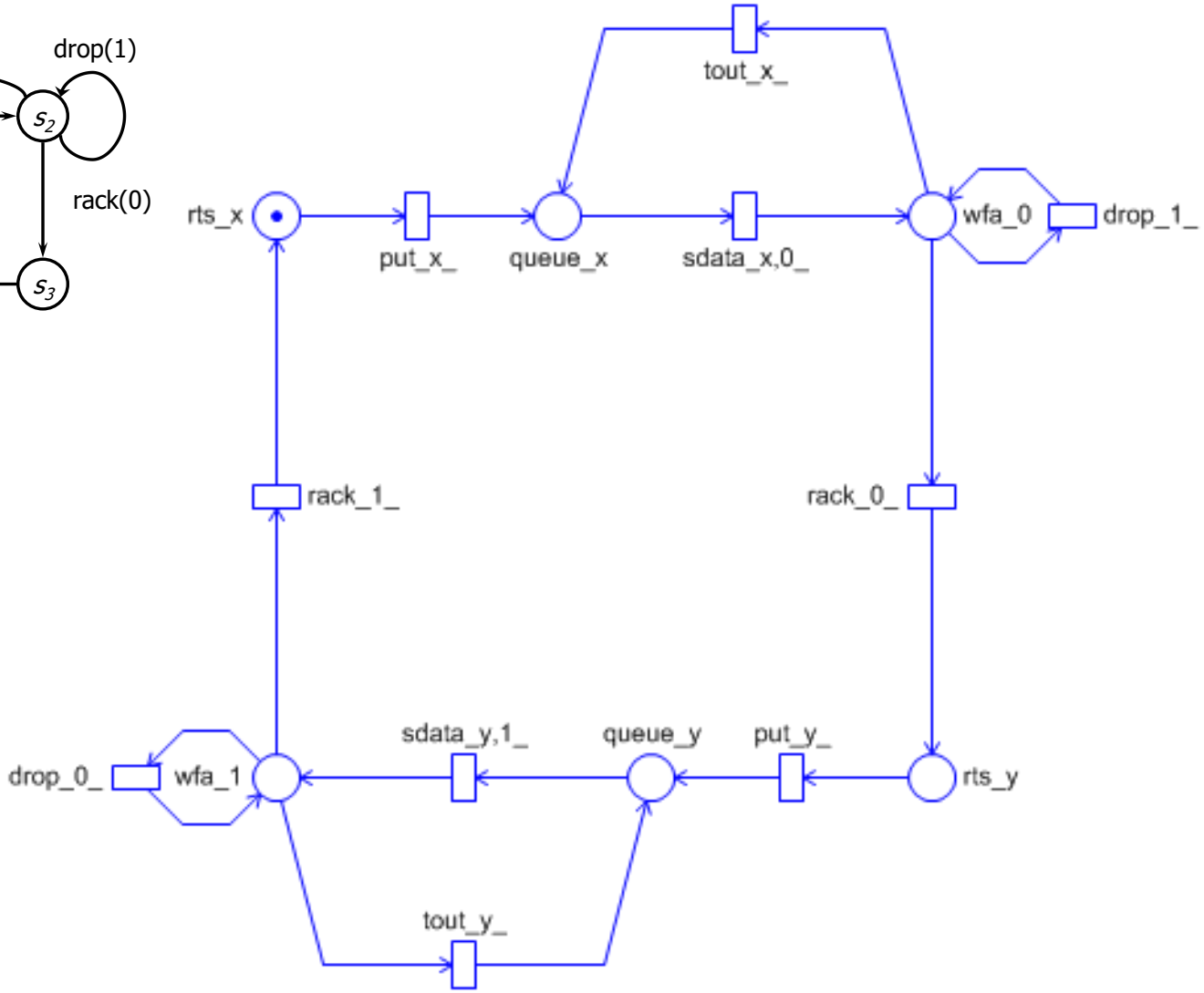


Nyugtázó csatorna állapotgráfja



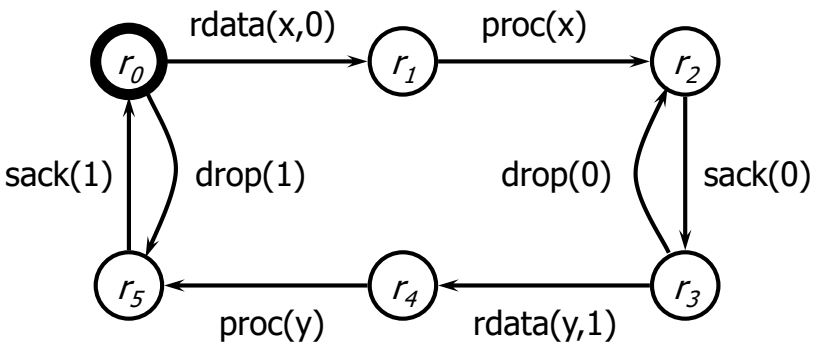
Küldő folyamat Petri-háló modellje

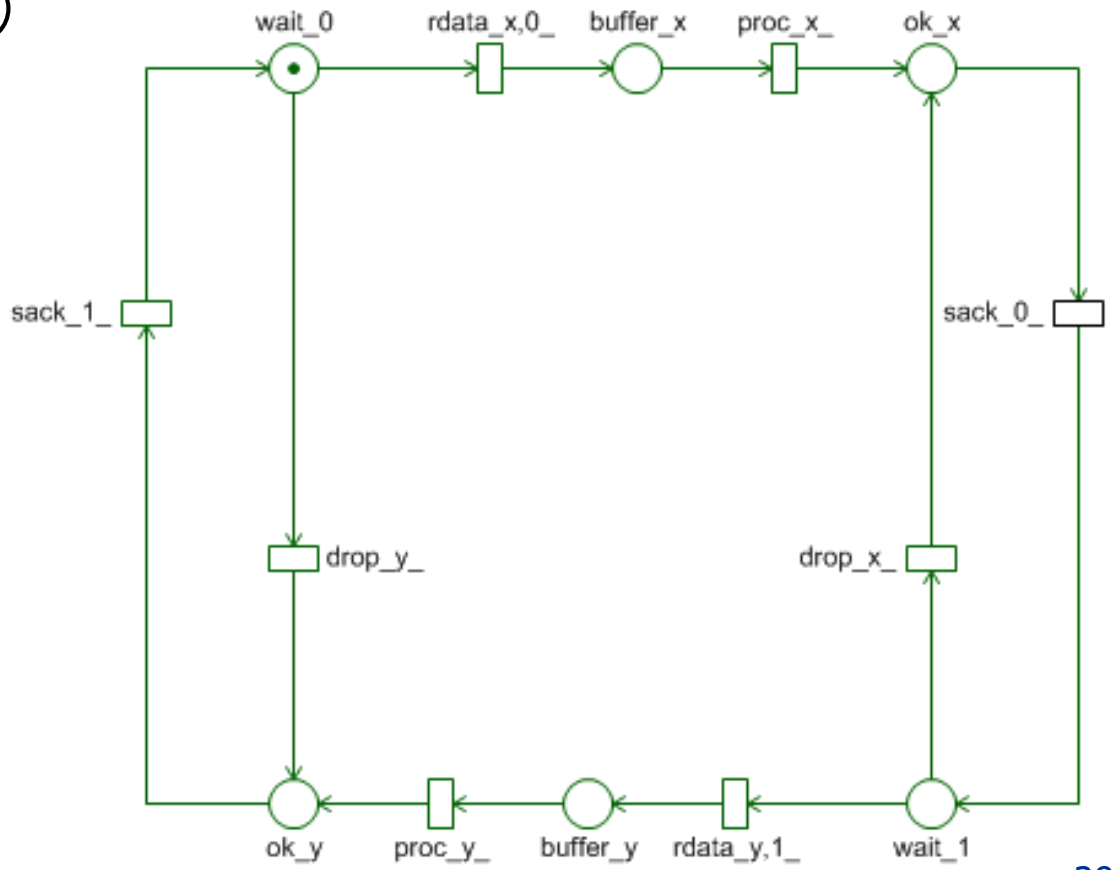




Zárójelek helyett ’_’ van a modellben

Fogadó folyamat Petri-háló modellje





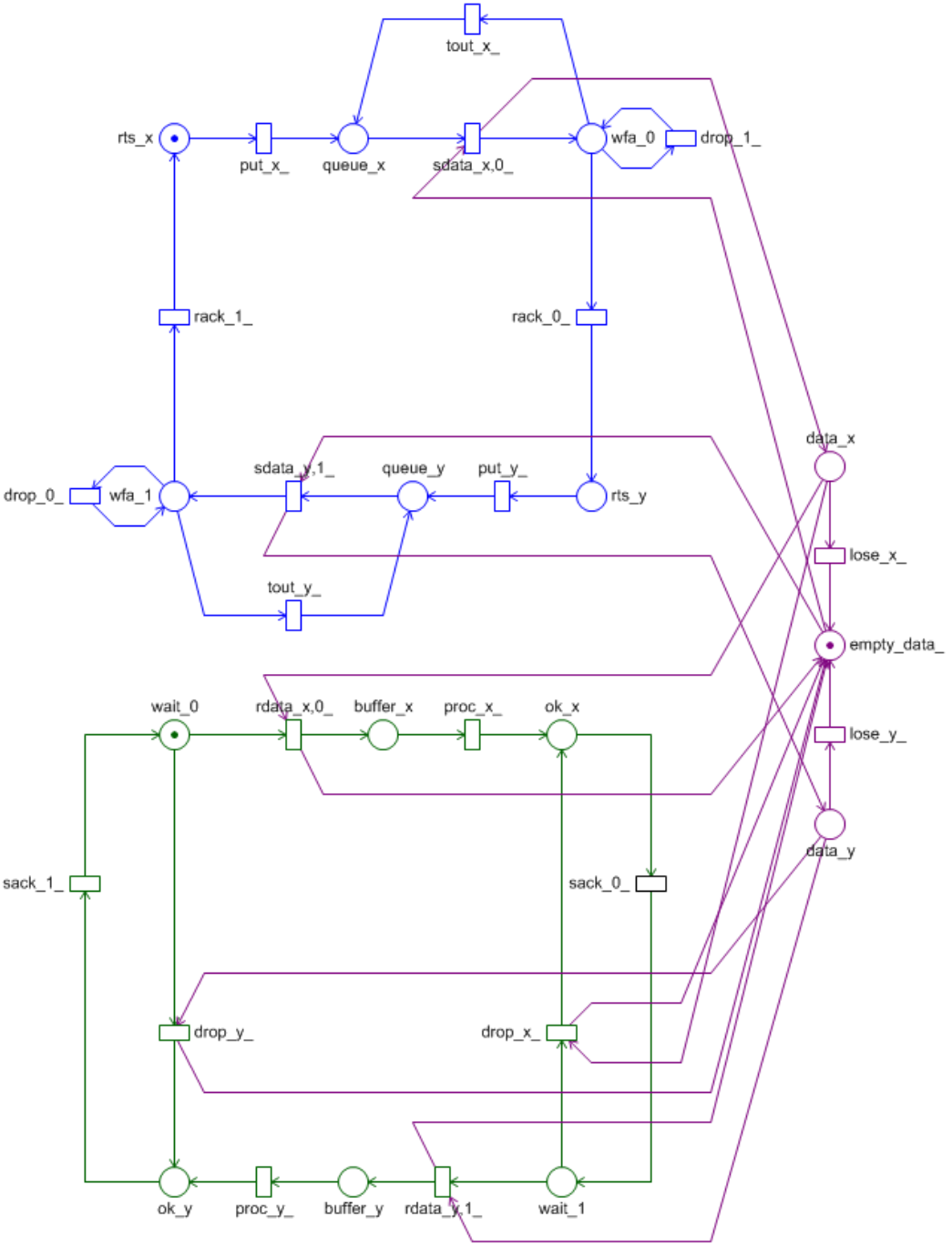
Adat csatorna és adatátvitel

Üzenet normál fogadása és átadása a folyamatok megfelelő tranzícióival együtt (élekkel bekötve a küldőtől illetve fogadótól):

- sdata()
- rdata()
- drop()

Üzenetvesztés:

- lose()



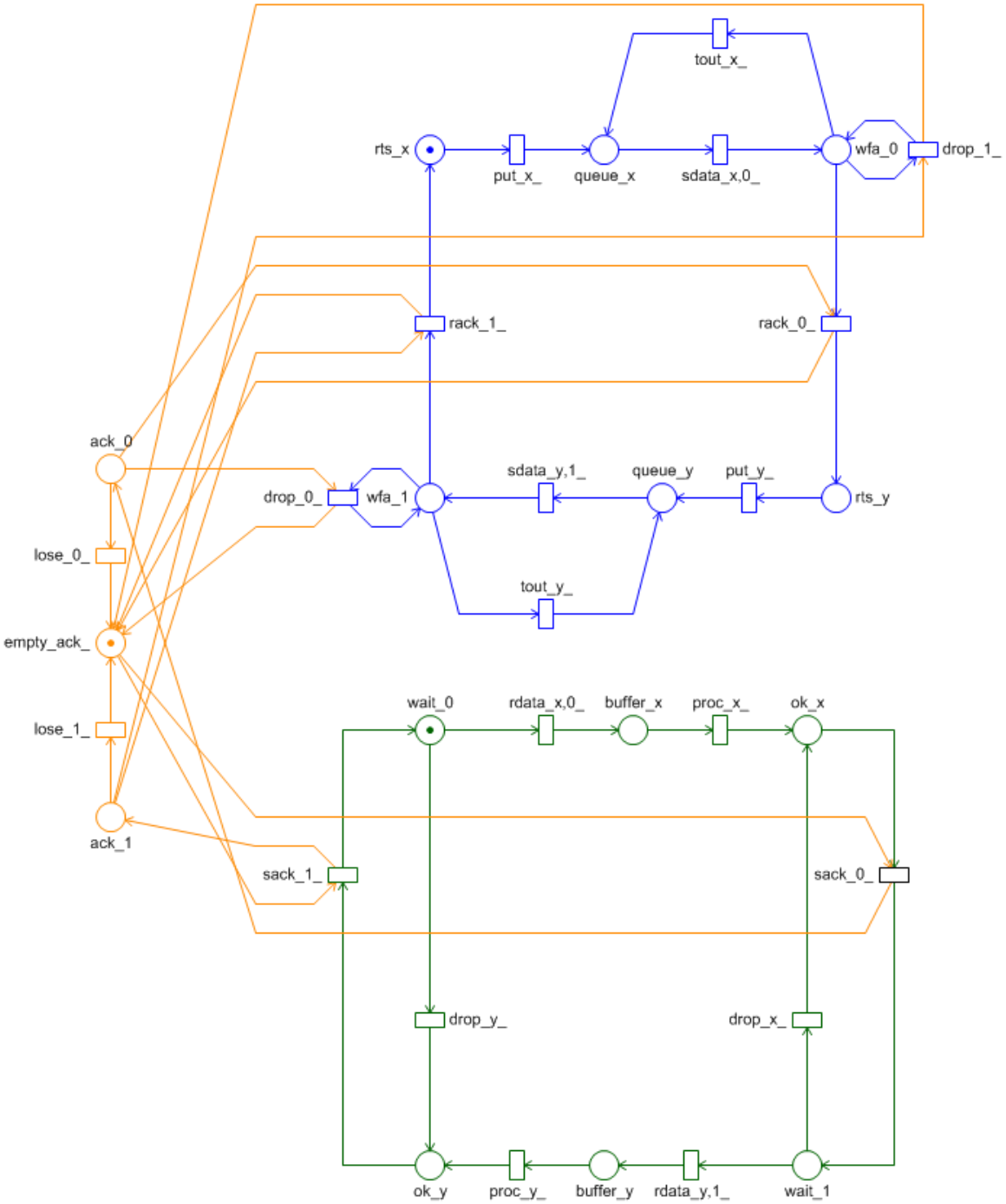
Nyugtázó csatorna és nyugtázás

Nyugta normál fogadása és átadása a folyamatok megfelelő tranzícióival együtt (élekkel bekötve a küldőtől illetve fogadótól):

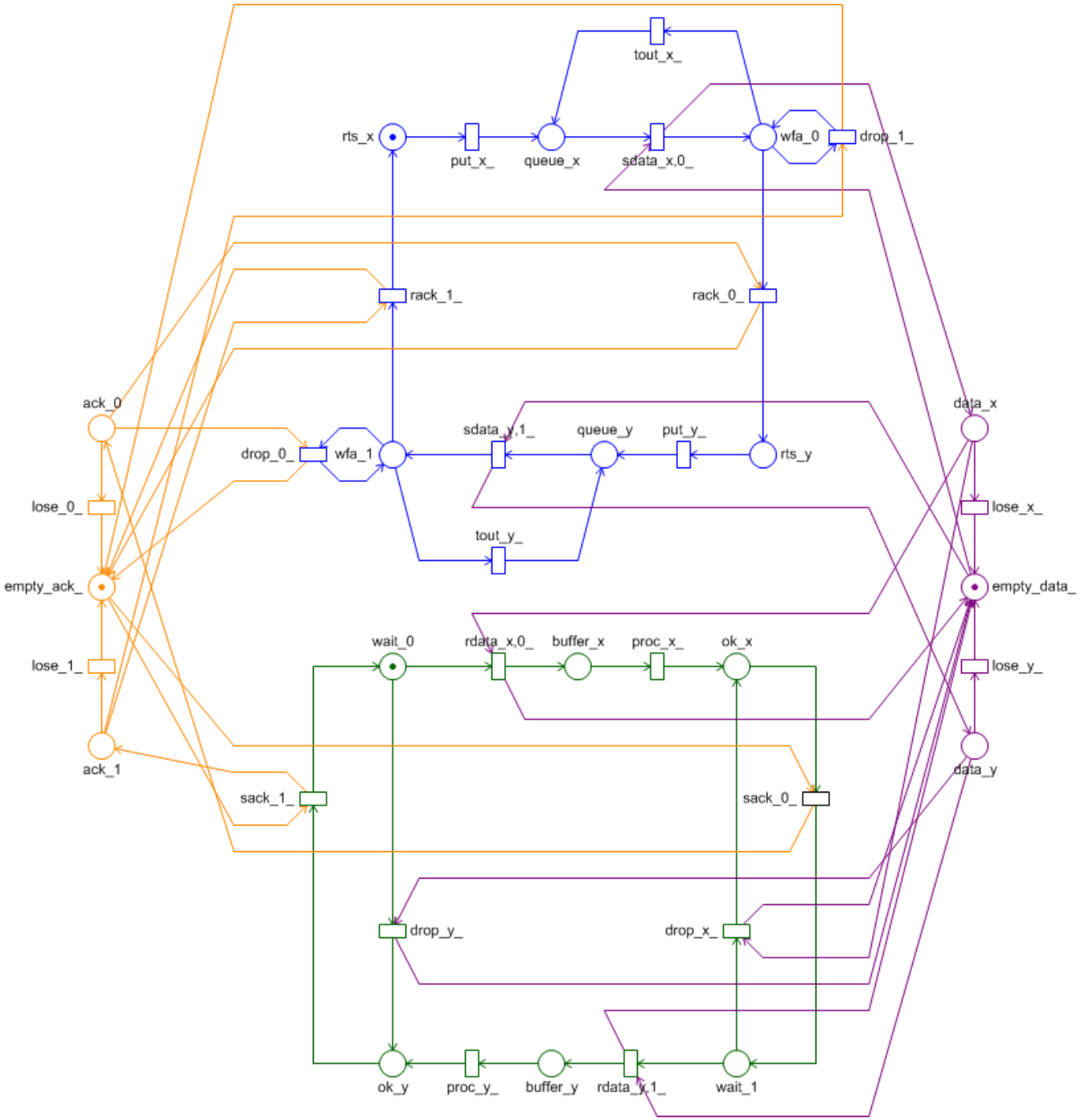
- sack()
- rack()
- drop()

Nyugta vesztése:

- lose()

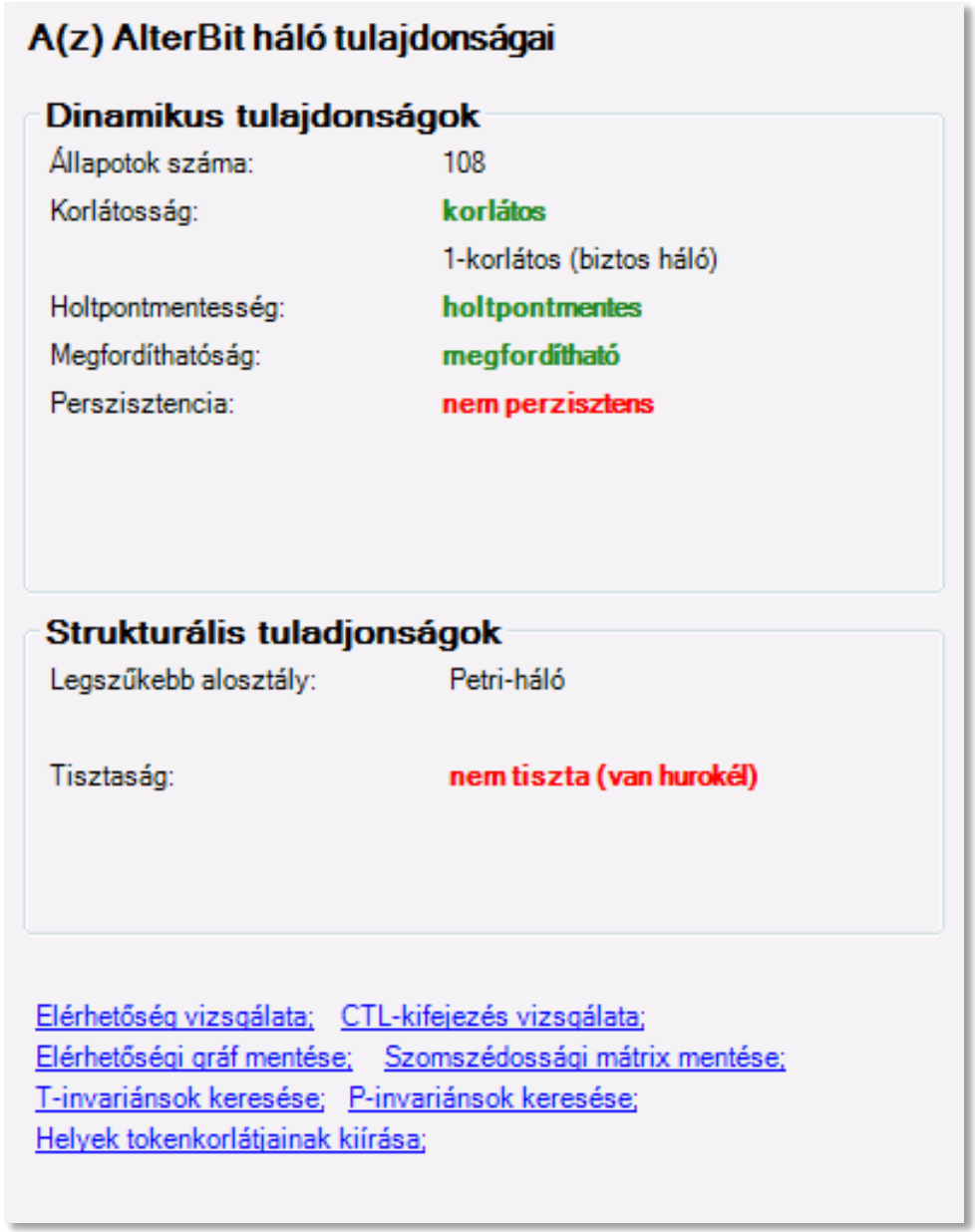


Teljes modell

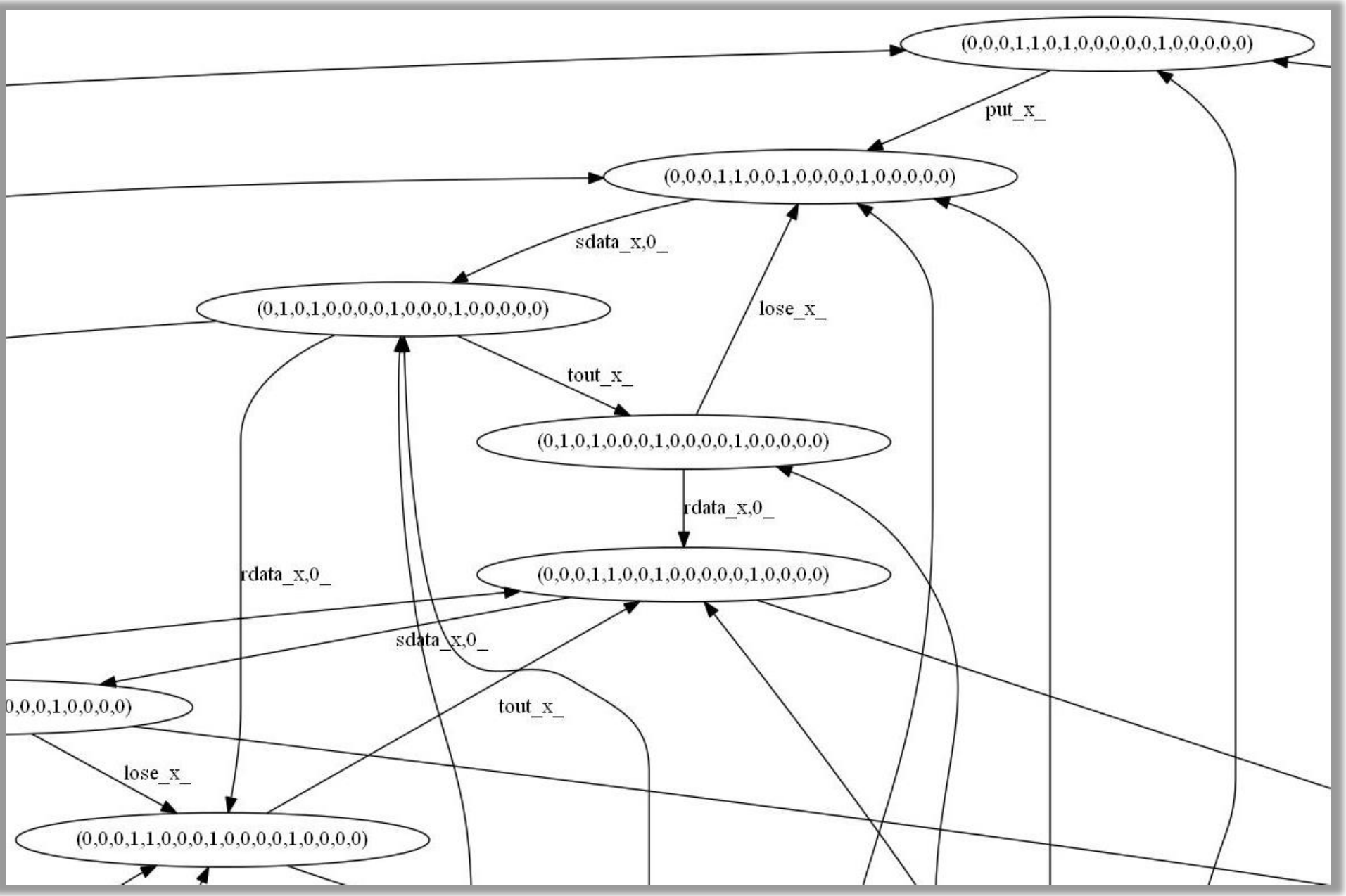


A példa modell analízise

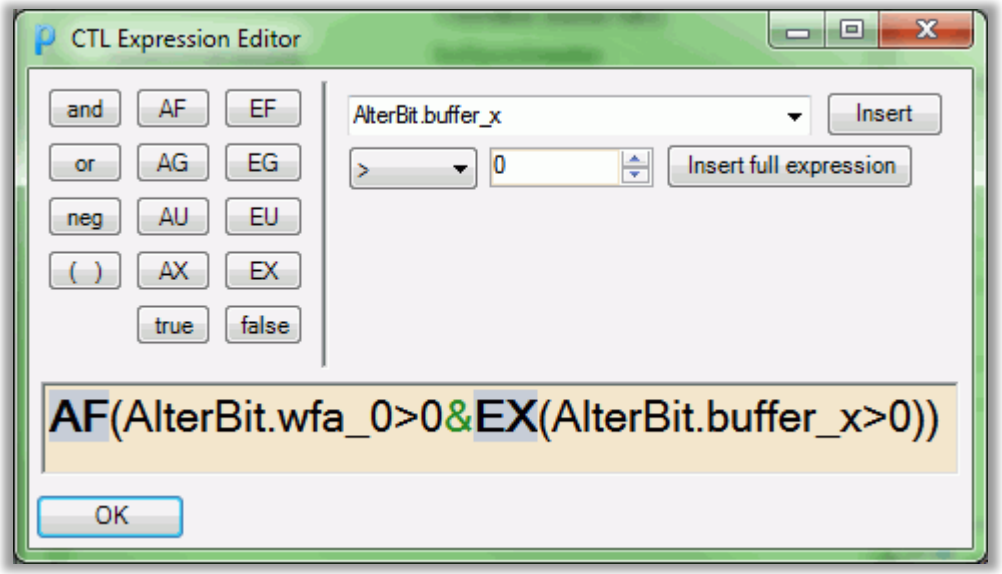
PetriDotNet: A modell dinamikus tulajdonságai

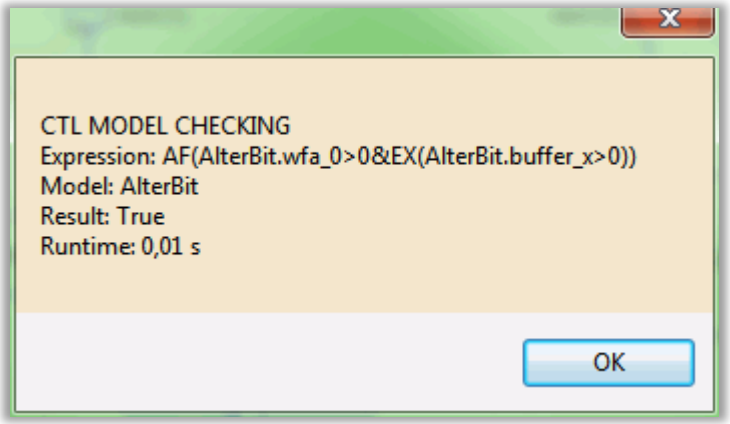


PetriDotNet: Elérhetőségi gráf kirajzolása (GraphViz)



PetriDotNet: CTL modellellenőrzés





AF(AlterBit.wfa_0>0 & **EX**(AlterBit.buffer_x>0)) ⇒ True

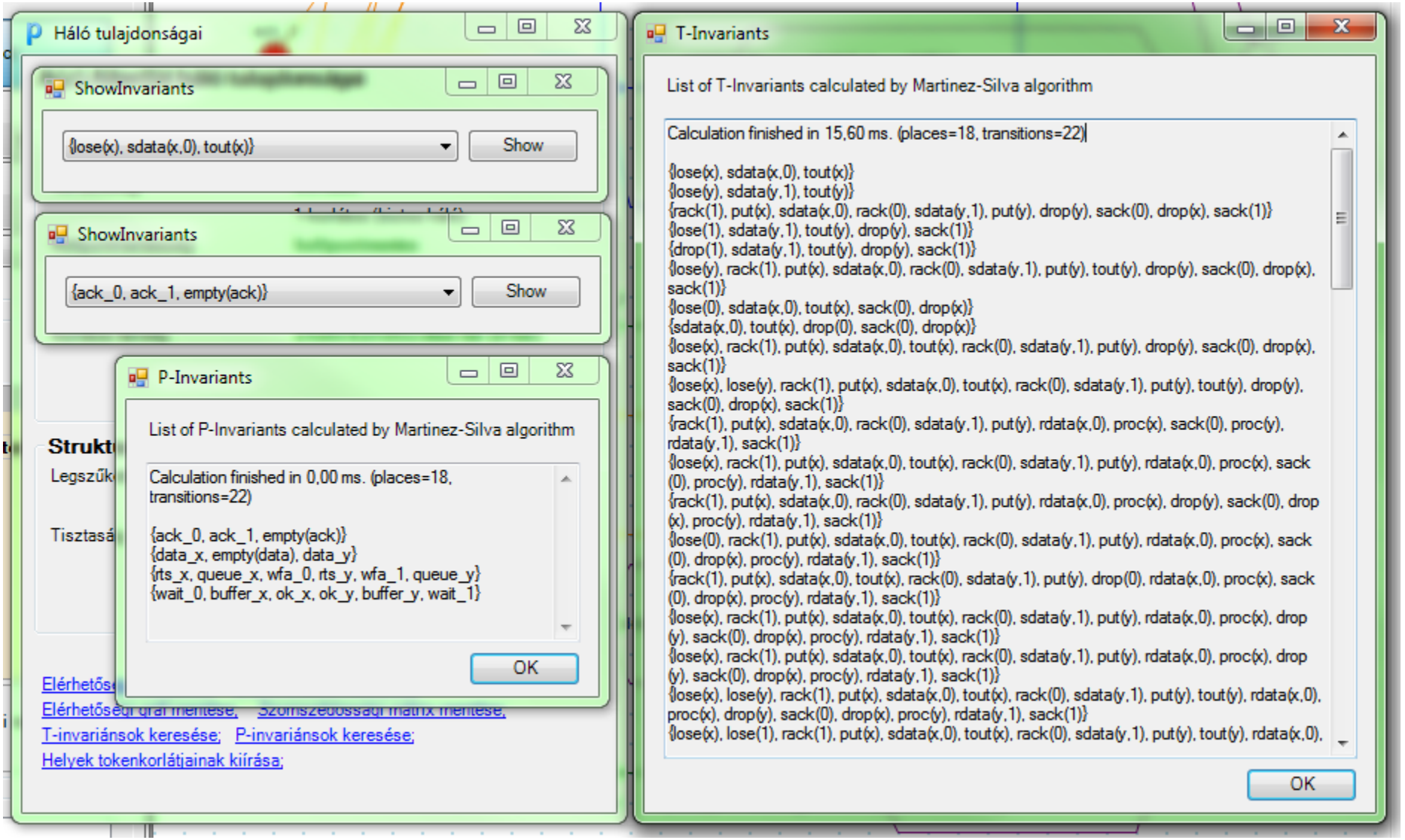
AG(**AF**(AlterBit.queue_y>0)) ⇒ False

AF(**EG**(AlterBit.queue_x=0)) ⇒ True

EF(AlterBit.wfa_0>0 & AlterBit.data_x=0) ⇒ True

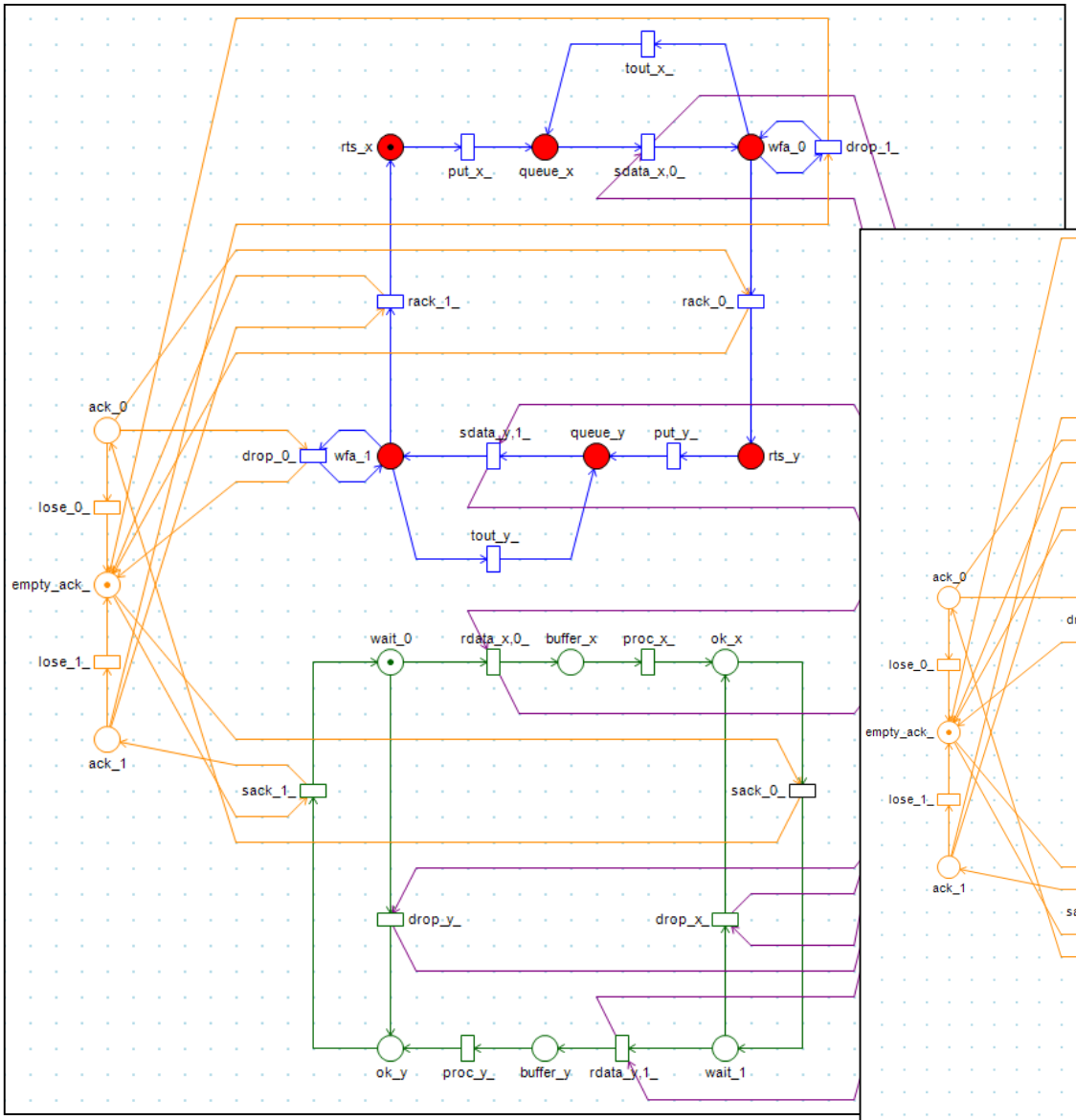
AF(AlterBit.queue_x>0 & **AX**(AlterBit.wfa_0>0 & AlterBit.data_x>0)) ⇒ True

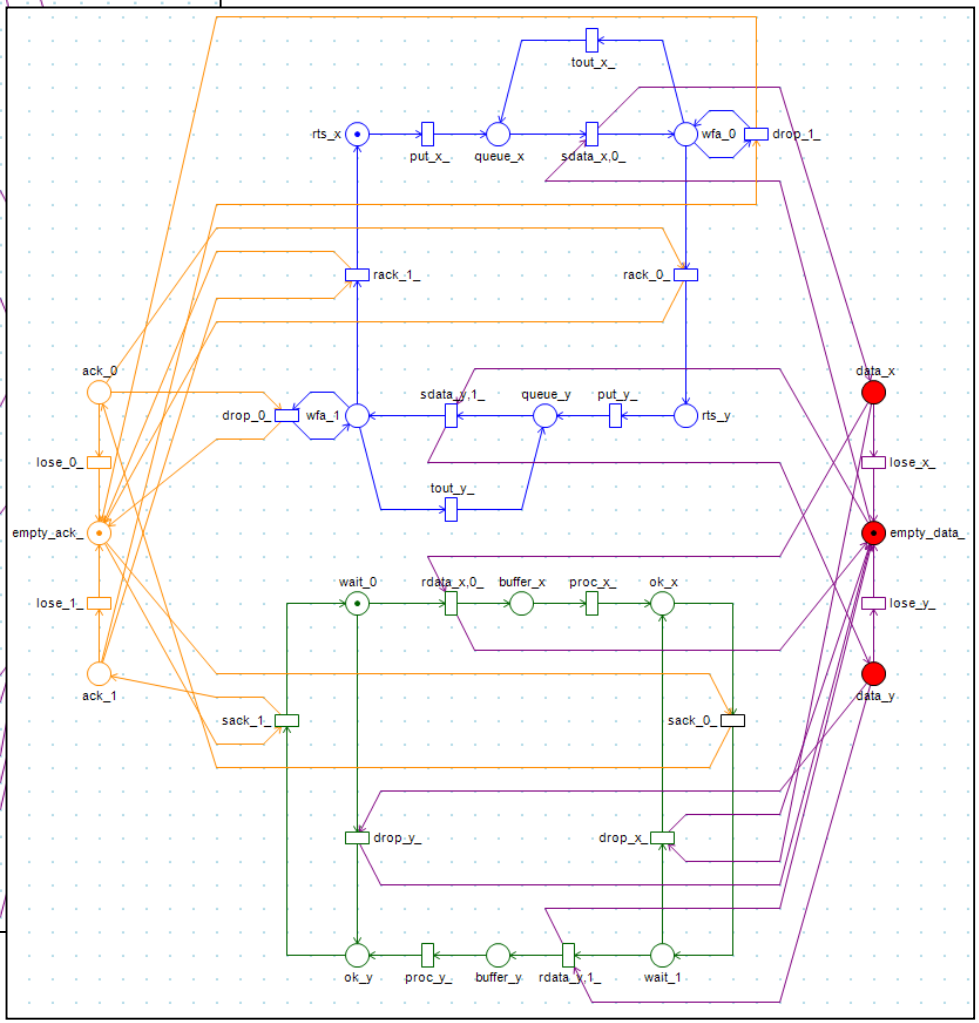
PetriDotNet: Invariáns analízis



PetriDotNet: P-invariánsok (példák)

Komponensek állapotgépei





PetriDotNet: T-invariánsok (példák)

Ciklikus működések (itt: hibátlan ill. adatvesztés)