

Petri-háló modellezés és analízis példák

dr. Bartha Tamás

dr. Majzik István

dr. Pataricza András

BME Méréstechnika és Információs Rendszerek Tanszék

Az alternáló bit protokoll

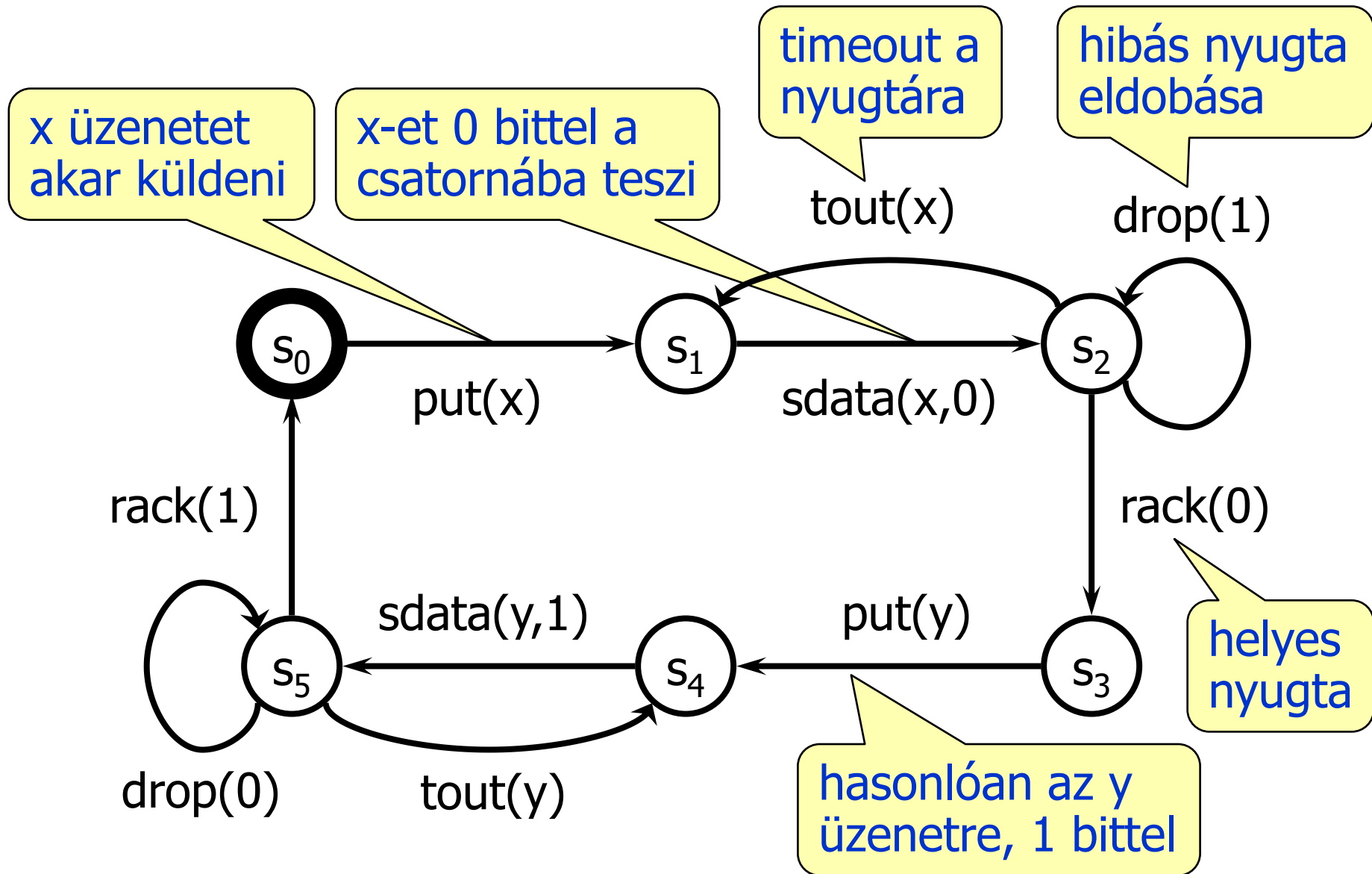
Az Alternating Bit Protocol

- Átviteli protokoll veszteséges csatornához
 - Üzenet **elveszhet** (véges számú alkalommal)
 - Üzenet tartalma nem változik (illetve az detektálható)
- Cél: a protokoll biztosítsa, hogy minden üzenet (véges számú próbálkozással) eljusson a vevőhöz
- Alapötlet:
 - A küldő az üzenetekhez egy **ellenőrző bitet** csatol
 - A vevő az üzenet megérkezését a vevőhöz küldött **nyugtával jelzi, ugyanazzal az ellenőrző bittel**
 - A küldő a következő üzenethez **negált ellenőrző bitet** csatol

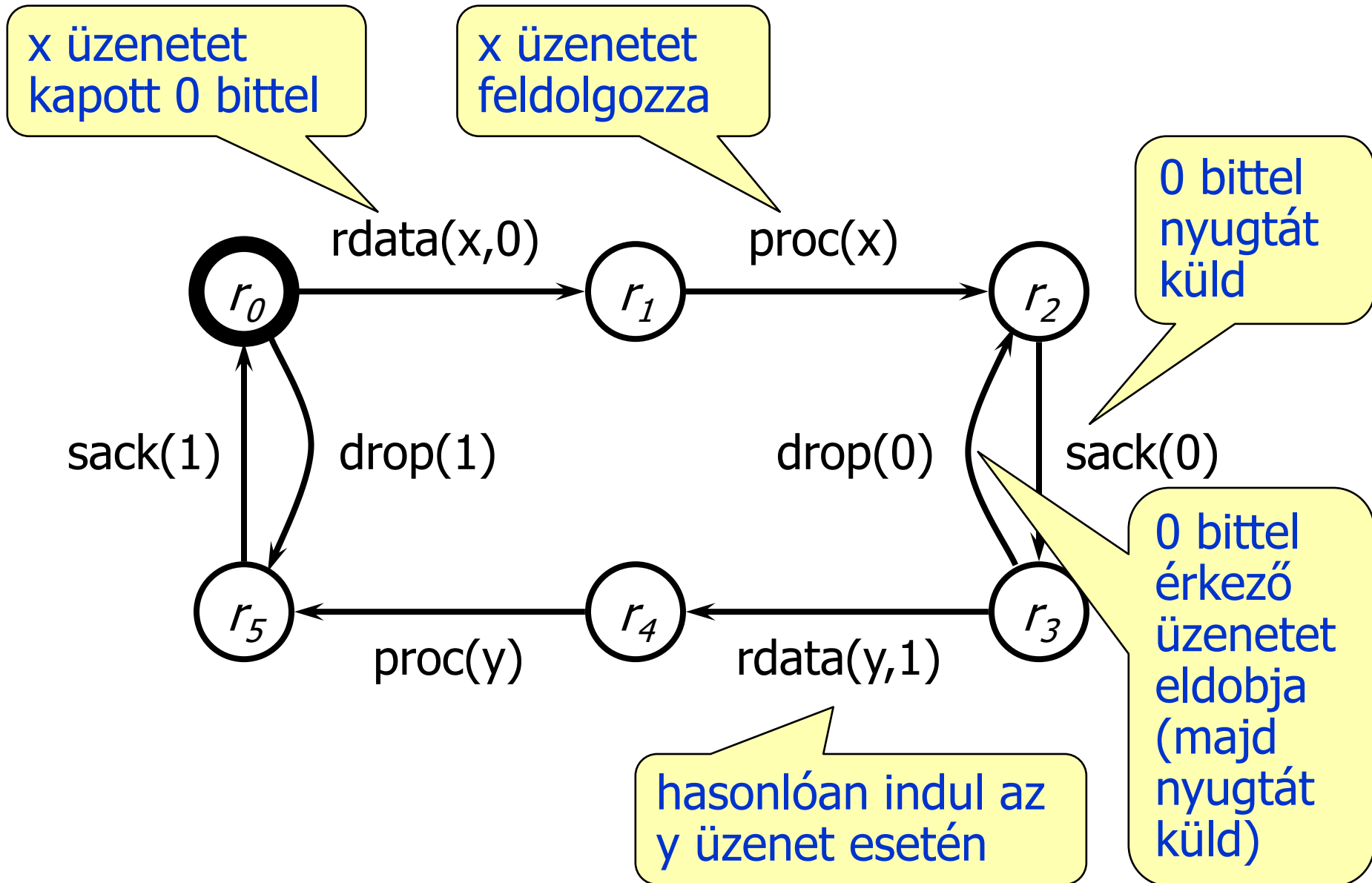
Az Alternating Bit Protocol üzenetvesztéssel

- A küldő minden üzenethez egy ellenőrző bitet csatol
- A vevő nyugtáz minden üzenetet azzal, hogy visszaküldi a kapott ellenőrző bitet
 - Ha a kapott ellenőrző bit nem negáltja az előzőleg kapott üzenet ellenőrző bitjének, akkor is visszaküldi, de az így kapott üzenetet nem dolgozza fel (eldobja)
- Ha a küldő azt látja, hogy a nyugtában visszakapta az előzőleg küldött ellenőrző bitet, akkor a következő üzenetküldéshez negálja azt
 - Ha nem azt az ellenőrző bitet kapta vissza, akkor eldobja a nyugtát (így timeout következhet be a jó nyugtára várás során)
 - Ha nem jön a nyugta (timeout), akkor újraküldi az üzenetet

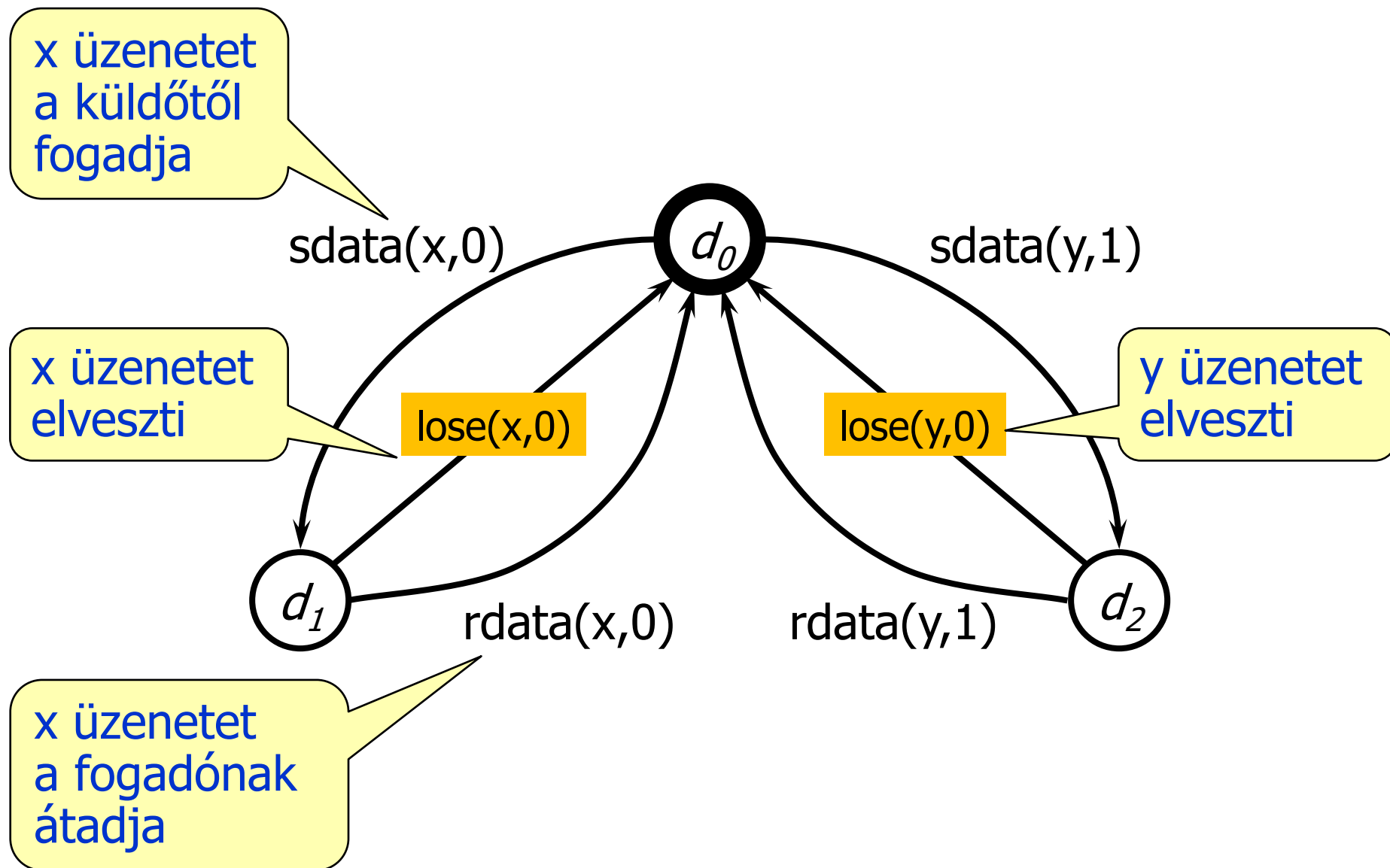
Küldő folyamat állapotgráfja



Fogadó folyamat állapotgráfja



Adat csatorna állapotgráfja

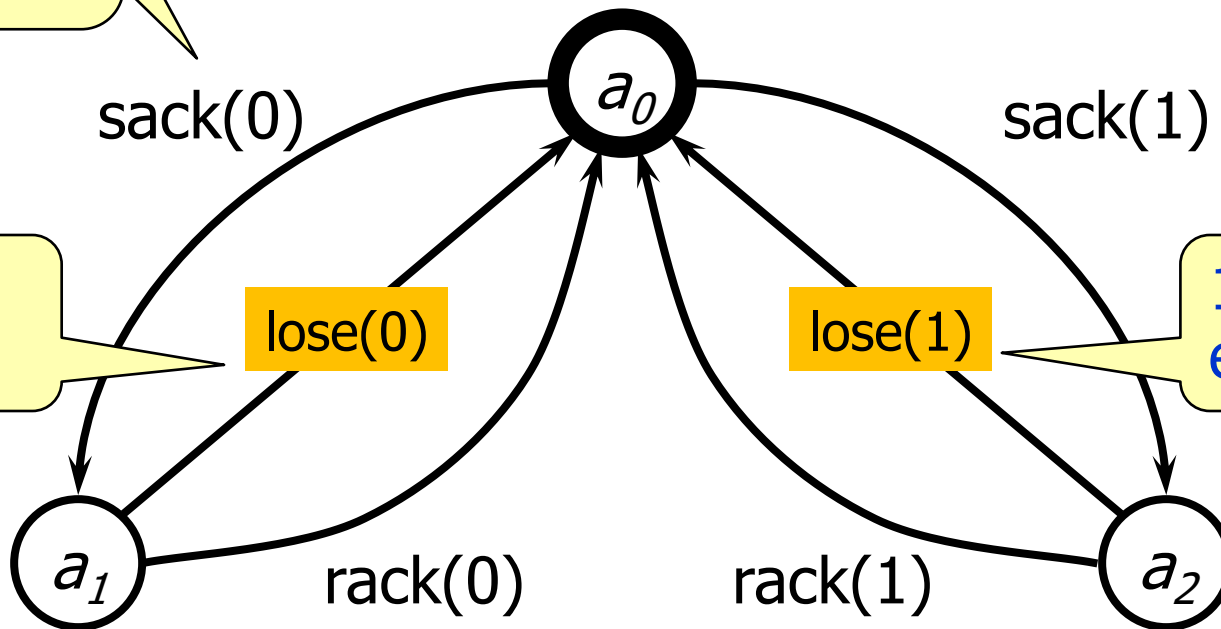


Nyugtázó csatorna állapotgráfja

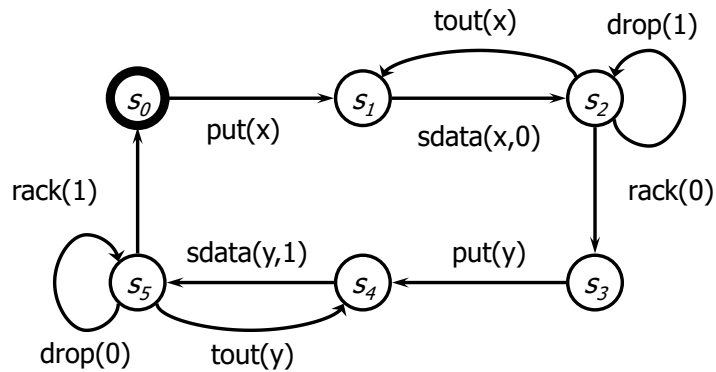
0 nyugtát
a fogadótól
fogadja

0 nyugtát
elveszti

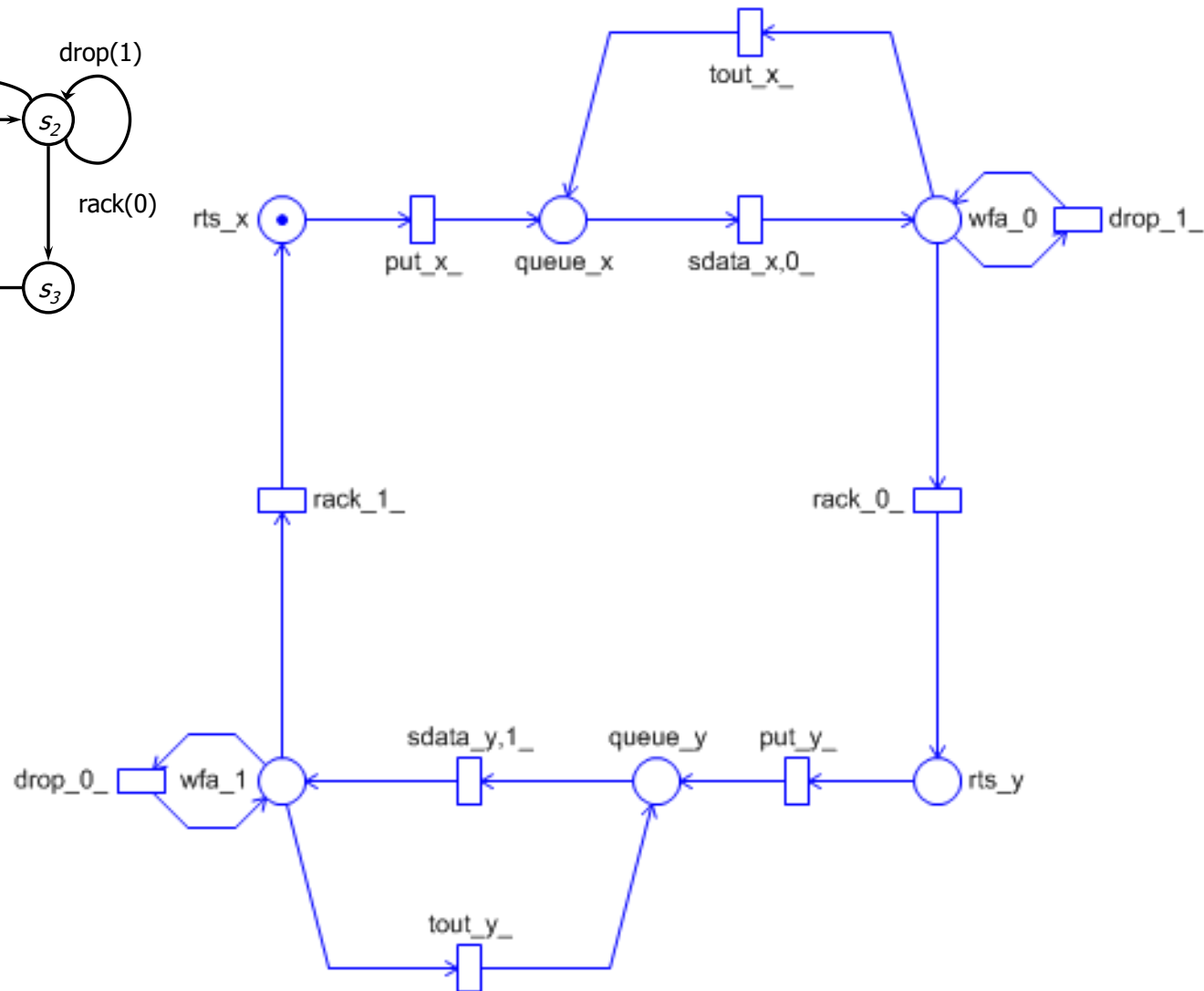
0 nyugtát
a küldőnek
átadja



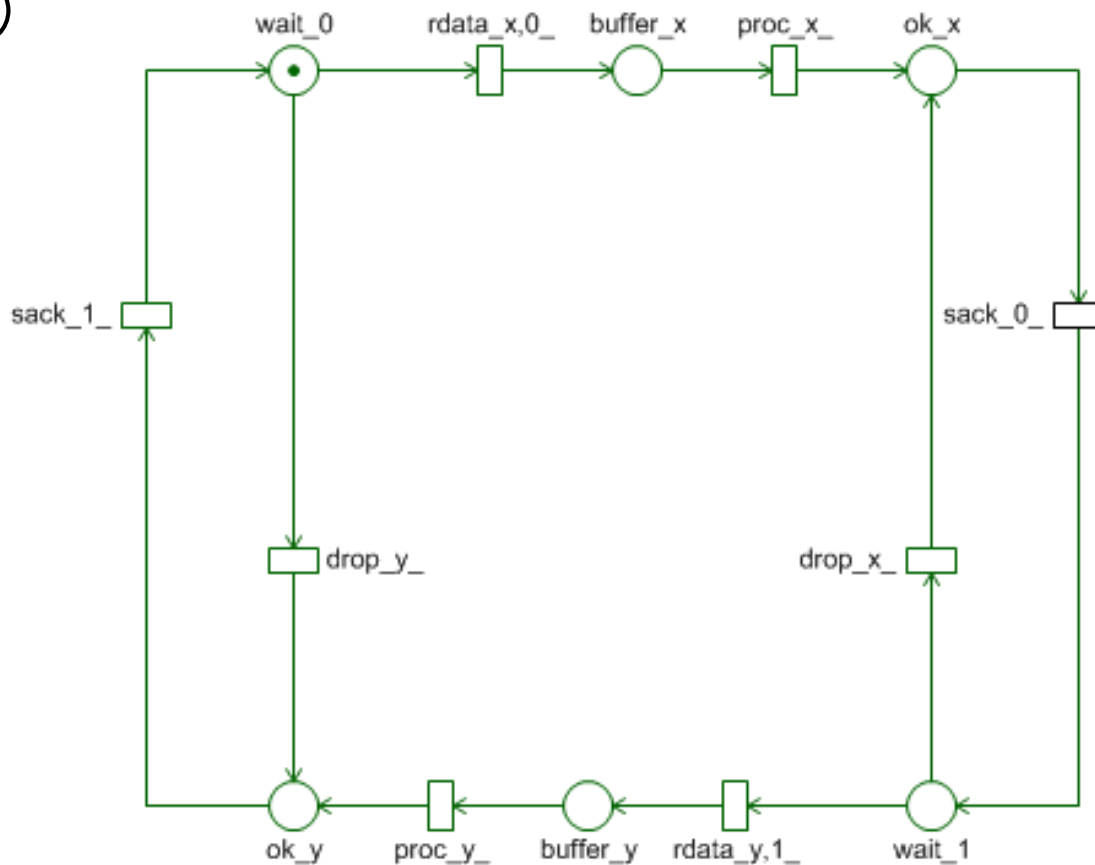
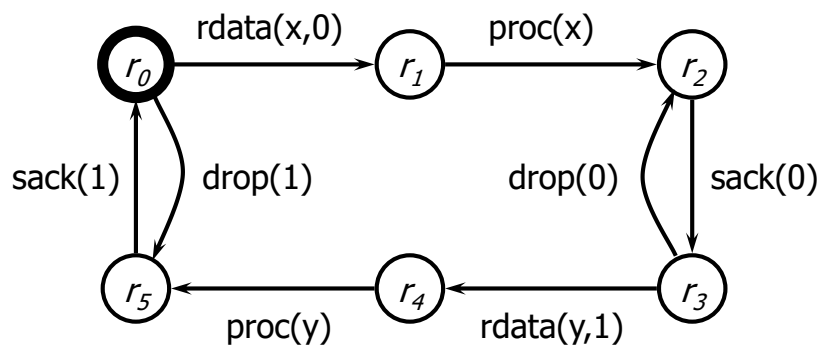
Küldő folyamat Petri-háló modellje



Zárójelek
helyett ' ' van
a modellben



Fogadó folyamat Petri-háló modellje



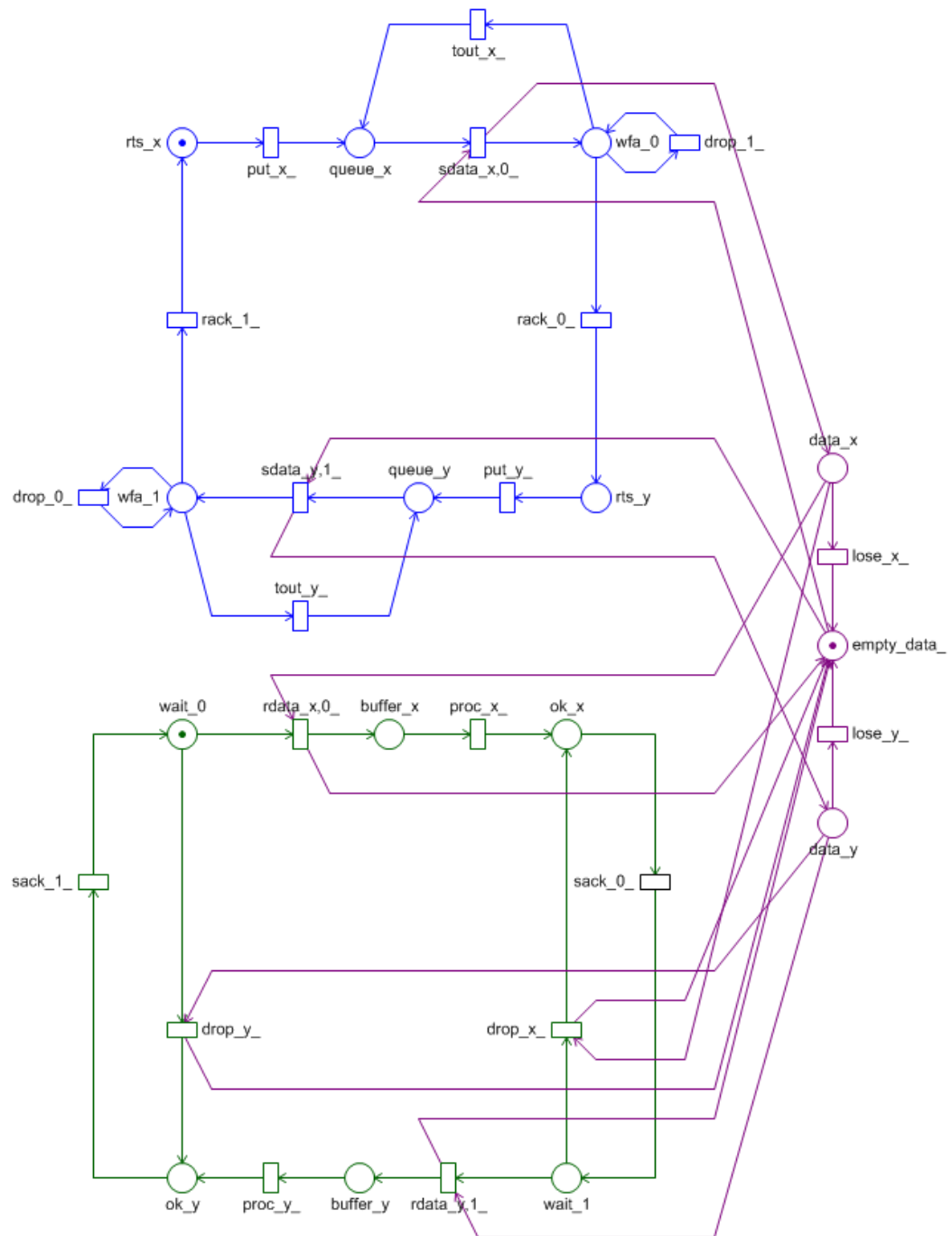
Adat csatorna és adatátvitel

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

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

Üzenetvesztés:

- lose()



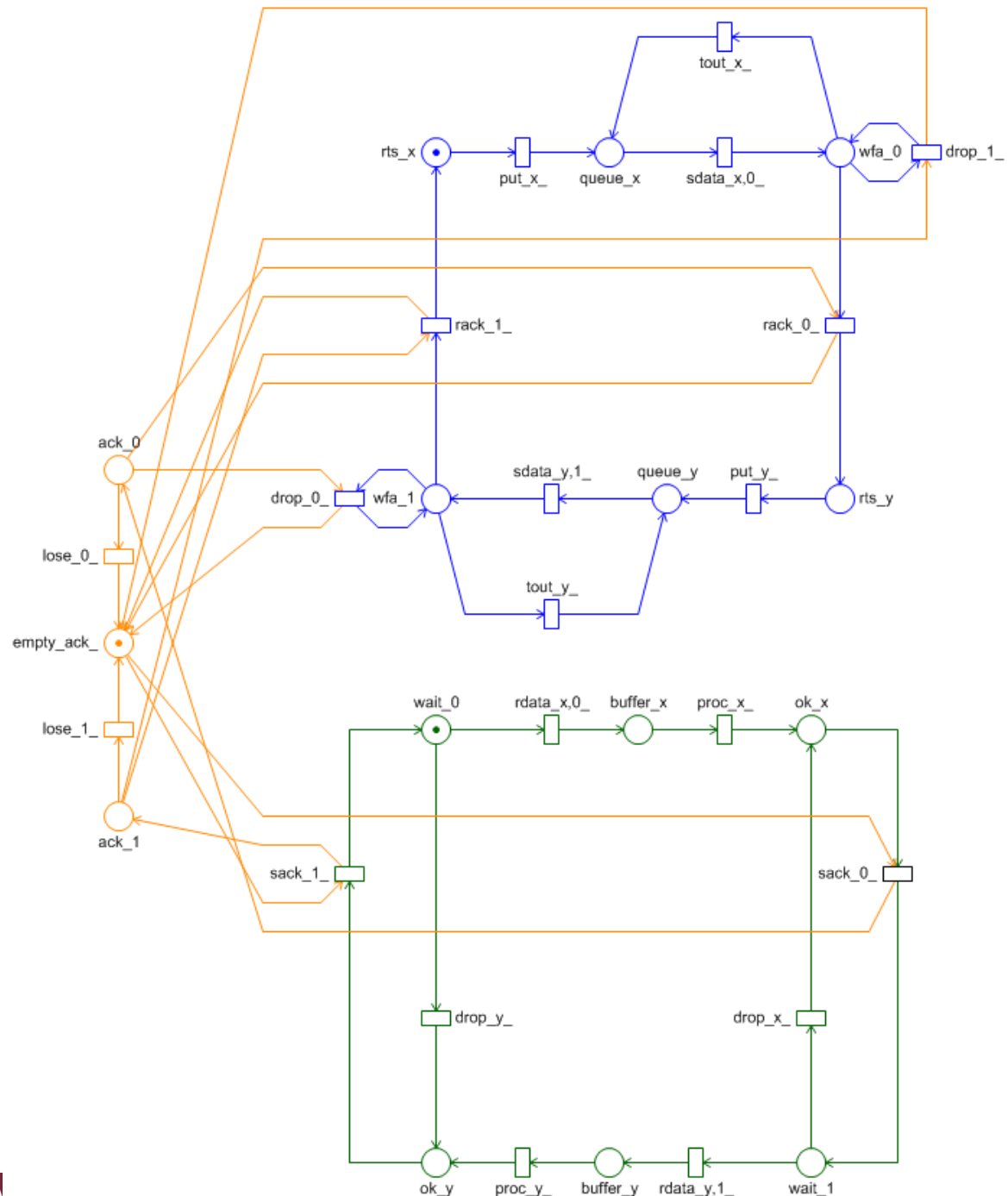
Nyugtázó csatorna és nyugtázás

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

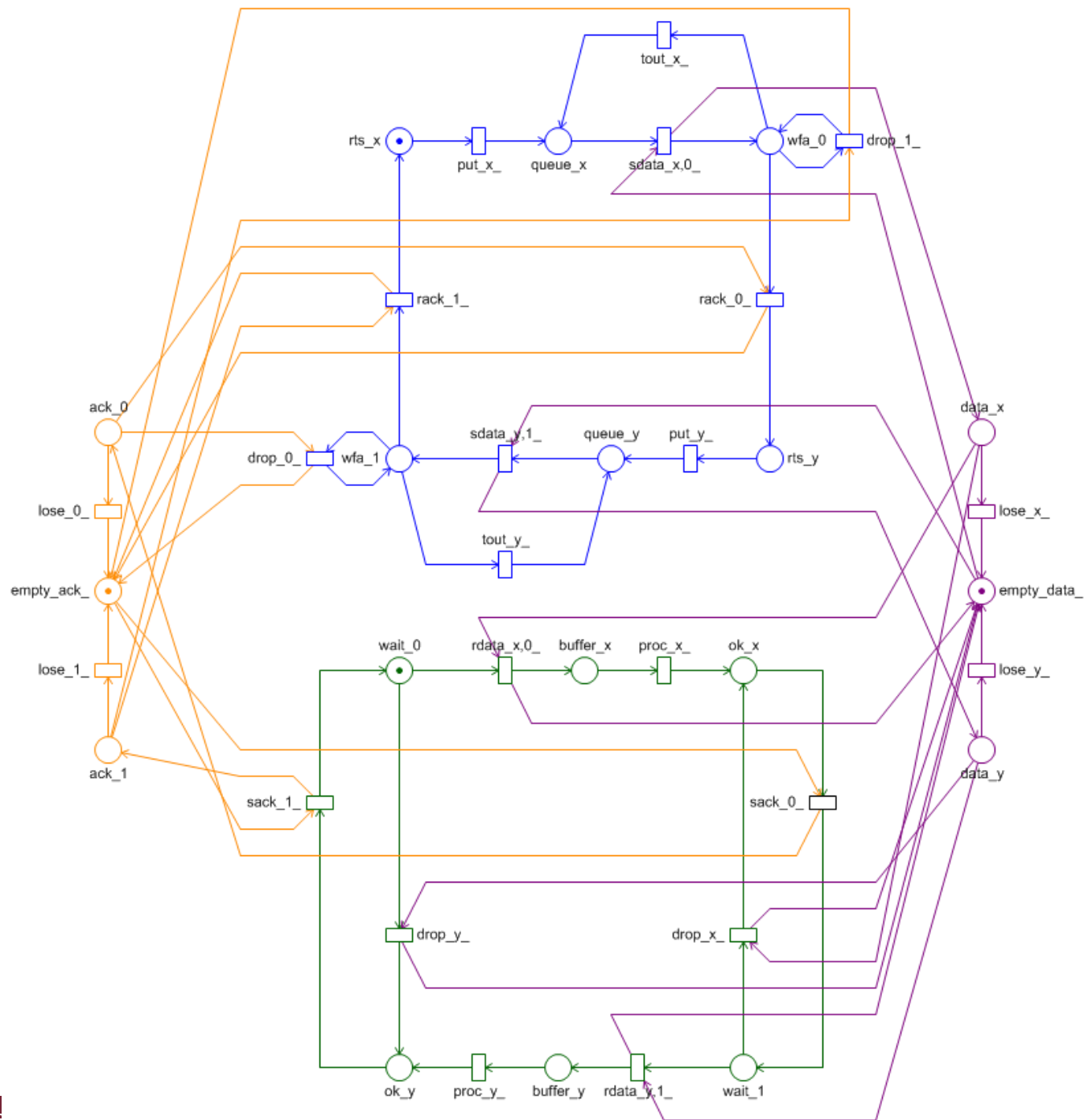
- sack()
- rack()
- drop()

Nyugta vesztese:

- lose()



A teljes modell



PetriDotNet: A modell dinamikus tulajdonságai

A(z) AlterBit háló tulajdonságai

Dinamikus tulajdonságok

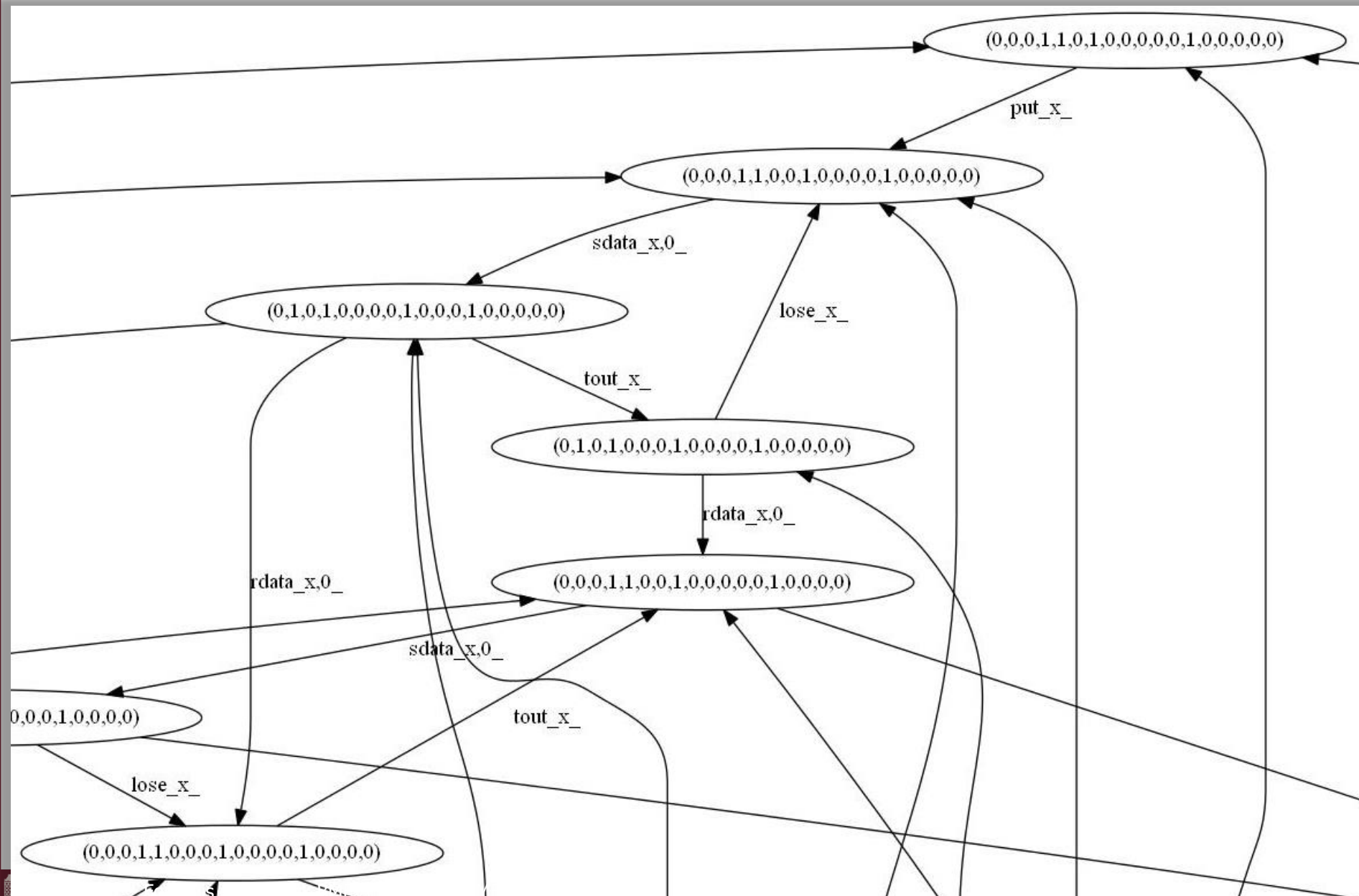
Állapotok száma:	108
Korlátosság:	korlátos 1-korlátos (biztos háló)
Holtpontmentesség:	holtpontmentes
Megfordíthatóság:	megfordítható
Perszisztencia:	nem perzisztens

Strukturális tulajdonságok

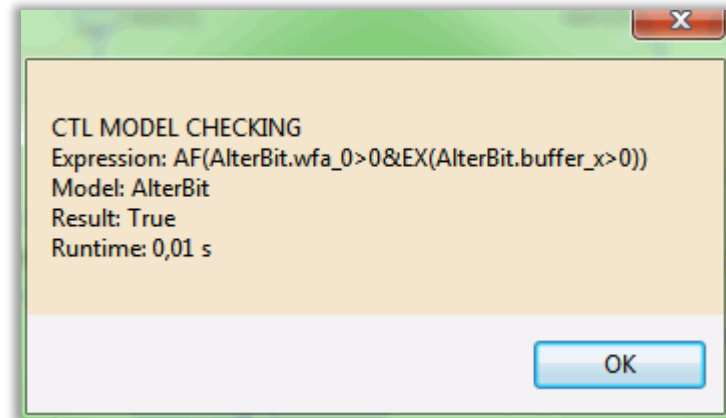
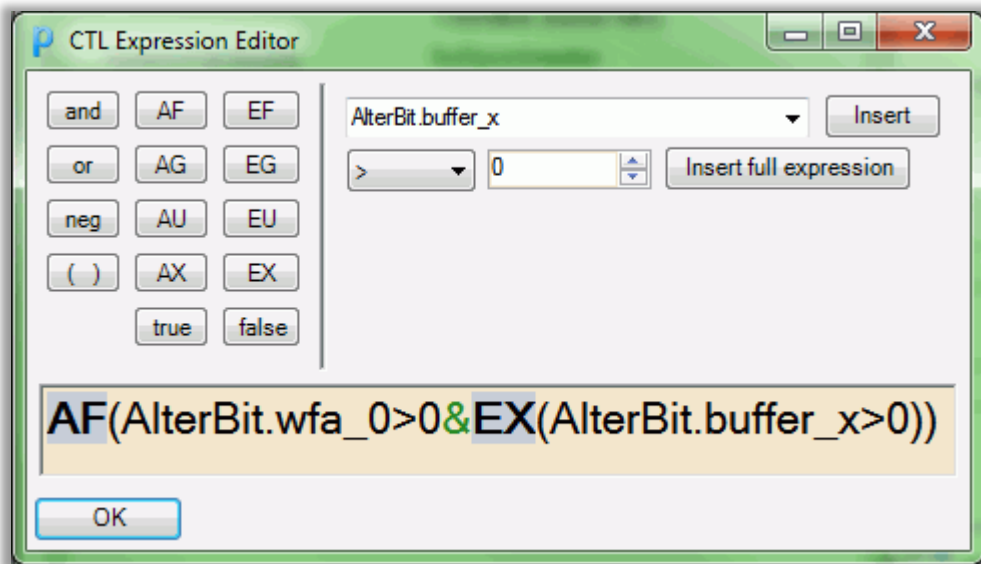
Legszűkebb alosztály:	Petri-háló
Tisztaság:	nem tiszta (van hurokél)

[Elérhetőség vizsgálata:](#) [CTL-kifejezés vizsgálata:](#)
[Elérhetőségi gráf mentése:](#) [Szomszédossági mátrix mentése:](#)
[T-invariánsok keresése:](#) [P-invariánsok keresése:](#)
[Helyek tokenkorlátjainak kiírása:](#)

PetriDotNet: Elérhetőségi gráf részlete (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

The screenshot displays the PetriDotNet application with several windows open for invariant analysis.

Háló tulajdonságai (Network Properties) window:

- ShowInvariants** (top): Input field contains `{lose(x), sdata(x,0), tout(x)}`, with a **Show** button.
- ShowInvariants** (bottom): Input field contains `{ack_0, ack_1, empty(ack)}`, with a **Show** button.

P-Invariants window:

List of P-Invariants calculated by Martinez-Silva algorithm

Calculation finished in 0,00 ms. (places=18, transitions=22)

```
{ack_0, ack_1, empty(ack)}  
{data_x, empty(data), data_y}  
{rts_x, queue_x, wfa_0, rts_y, wfa_1, queue_y}  
{wait_0, buffer_x, ok_x, ok_y, buffer_y, wait_1}
```

T-Invariants window:

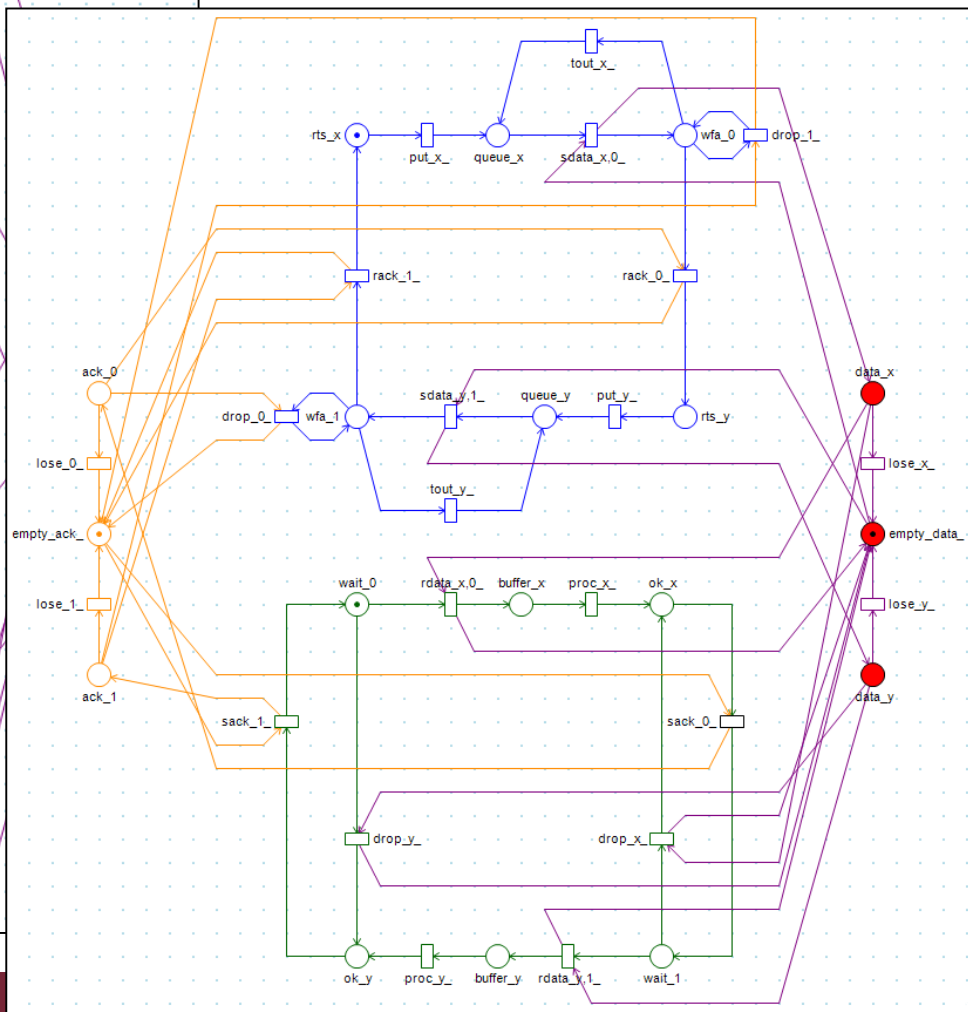
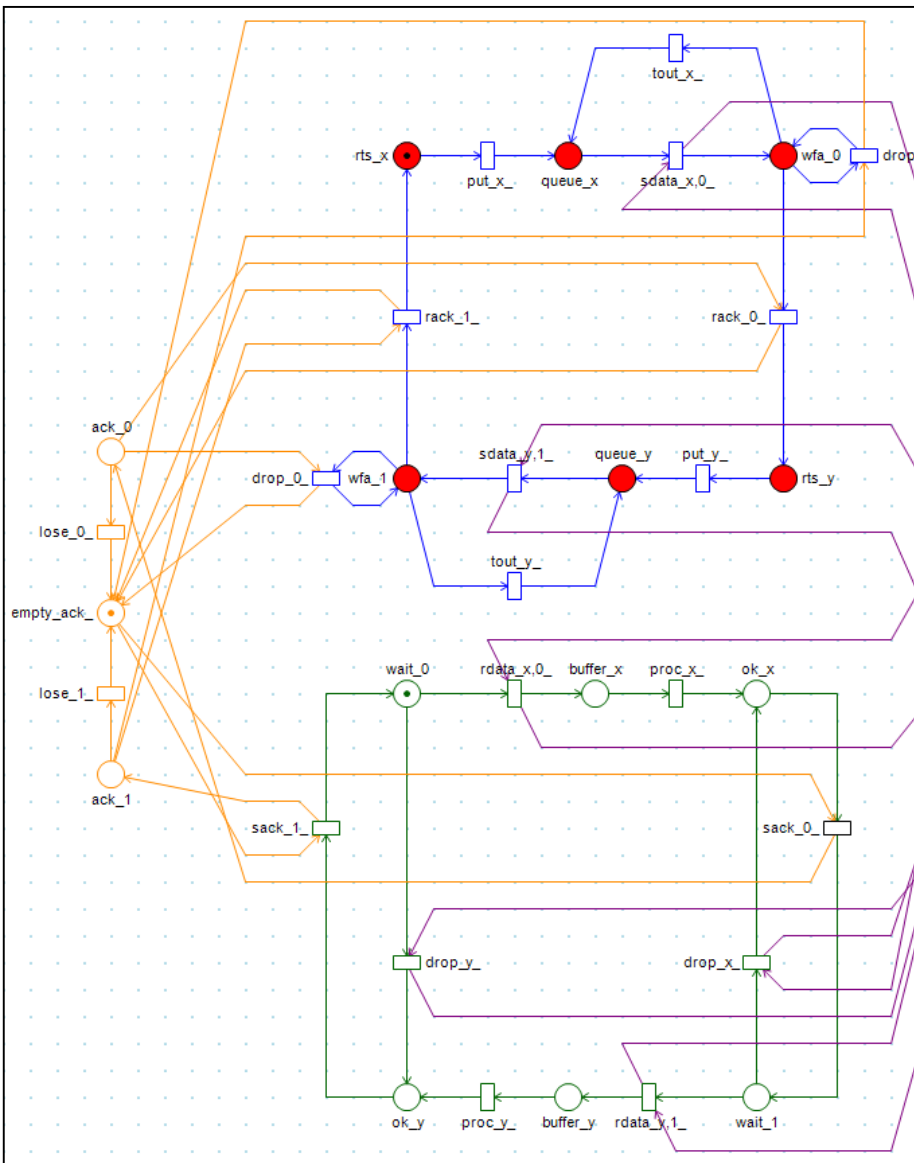
List of T-Invariants calculated by Martinez-Silva algorithm

Calculation finished in 15,60 ms. (places=18, transitions=22)

```
{lose(x), sdata(x,0), tout(x)}  
{lose(y), sdata(y,1), tout(y)}  
{rack(1), put(x), sdata(x,0), rack(0), sdata(y,1), put(y), drop(y), sack(0), drop(x), sack(1)}  
{lose(1), sdata(y,1), tout(y), drop(y), sack(1)}  
{drop(1), sdata(y,1), tout(y), drop(y), sack(1)}  
{lose(y), rack(1), put(x), sdata(x,0), rack(0), sdata(y,1), put(y), tout(y), drop(y), sack(0), drop(x), sack(1)}  
{lose(0), sdata(x,0), tout(x), sack(0), drop(x)}  
{sdata(x,0), tout(x), drop(0), sack(0), drop(x)}  
{lose(x), rack(1), put(x), sdata(x,0), tout(x), rack(0), sdata(y,1), put(y), drop(y), sack(0), drop(x), sack(1)}  
{lose(x), lose(y), rack(1), put(x), sdata(x,0), tout(x), rack(0), sdata(y,1), put(y), tout(y), drop(y), sack(0), drop(x), sack(1)}  
{rack(1), put(x), sdata(x,0), rack(0), sdata(y,1), put(y), rdata(x,0), proc(x), sack(0), proc(y), rdata(y,1), sack(1)}  
{lose(x), rack(1), put(x), sdata(x,0), tout(x), rack(0), sdata(y,1), put(y), rdata(x,0), proc(x), sack(0), proc(y), rdata(y,1), sack(1)}  
{rack(1), put(x), sdata(x,0), rack(0), sdata(y,1), put(y), rdata(x,0), proc(x), drop(y), sack(0), drop(x), proc(y), rdata(y,1), sack(1)}  
{lose(0), rack(1), put(x), sdata(x,0), tout(x), rack(0), sdata(y,1), put(y), rdata(x,0), proc(x), sack(0), drop(x), proc(y), rdata(y,1), sack(1)}  
{rack(1), put(x), sdata(x,0), tout(x), rack(0), sdata(y,1), put(y), drop(0), rdata(x,0), proc(x), sack(0), drop(x), proc(y), rdata(y,1), sack(1)}  
{lose(x), rack(1), put(x), sdata(x,0), tout(x), rack(0), sdata(y,1), put(y), rdata(x,0), proc(x), drop(y), sack(0), drop(x), proc(y), rdata(y,1), sack(1)}  
{lose(x), rack(1), put(x), sdata(x,0), tout(x), rack(0), sdata(y,1), put(y), rdata(x,0), proc(x), drop(y), sack(0), drop(x), proc(y), rdata(y,1), sack(1)}  
{lose(x), lose(y), rack(1), put(x), sdata(x,0), tout(x), rack(0), sdata(y,1), put(y), tout(y), rdata(x,0), proc(x), drop(y), sack(0), drop(x), proc(y), rdata(y,1), sack(1)}  
{lose(x), lose(1), rack(1), put(x), sdata(x,0), tout(x), rack(0), sdata(y,1), put(y), tout(y), rdata(x,0),
```

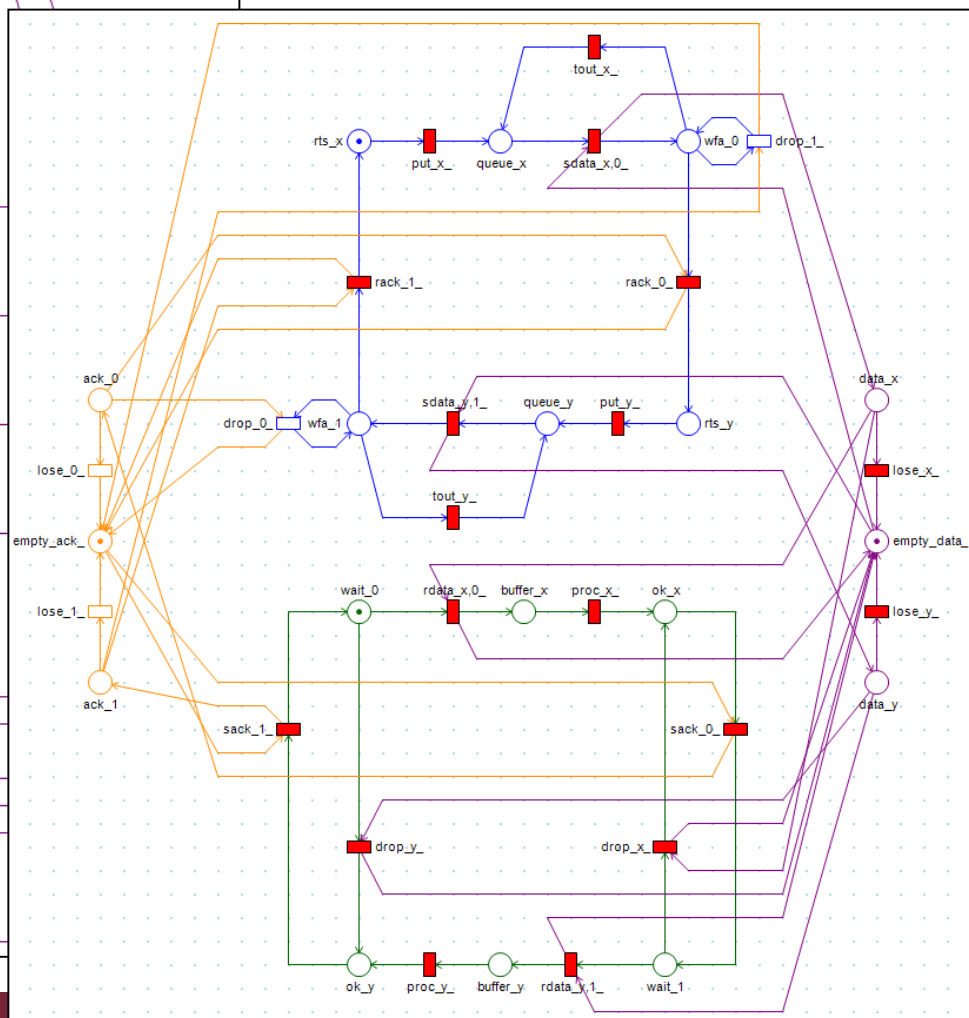
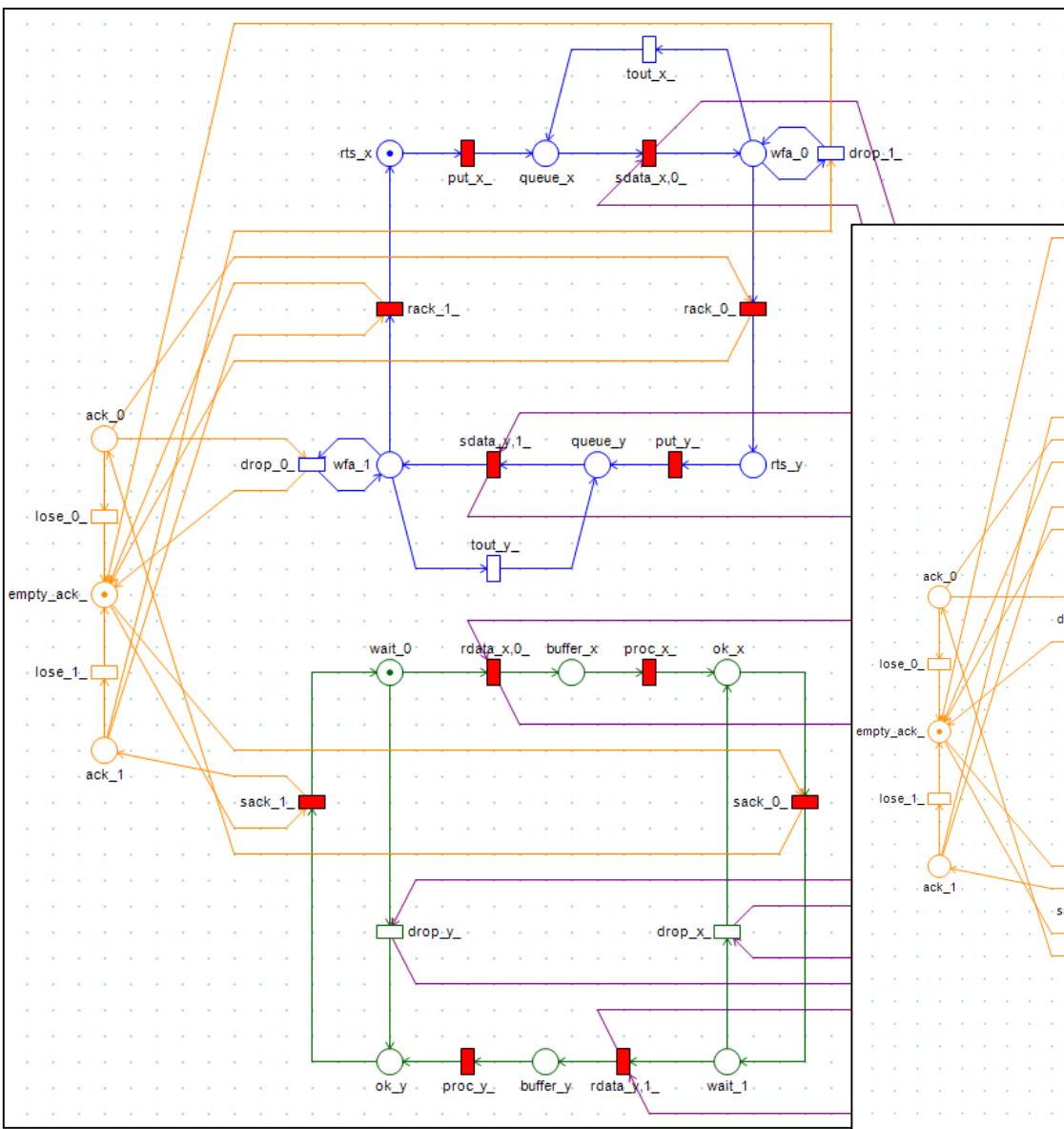
PetriDotNet: P-invariánsok (példák)

Komponensek
állapotgépei



PetriDotNet: T-invariánsok (példák)






Ciklikus működések (itt:
hibátlan ill. adatvesztés)



Sarkvidéki drón (demo)

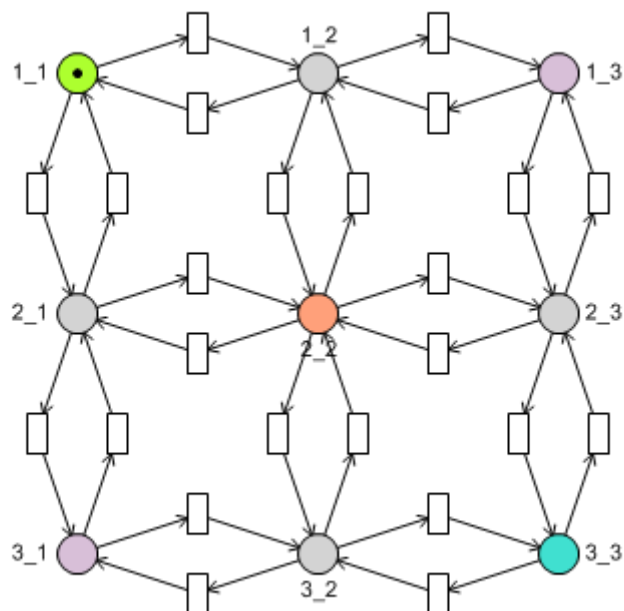
A feladat

- A drón a sarkvidéken végez küldetések
- A drón mozgása
 - A terület cellákra van osztva, ezeken mozog
 - Átlósan nem mozoghat
- A drón töltése
 - Kezdetben a max. 4 egység
 - Szomszédos cellák közötti mozgás egy egységet fogyaszt
 - Az (1,3) és (3,1) cellában töltőállomás van 4-4 egységnyi töltéssel: ezekből feltölthető
- A drón missziója
 - Az (1,1) bázisról indulva felvenni a (3,3) cellában a segélycsomagot és a (2,2) cellában lévő balesethez vinni

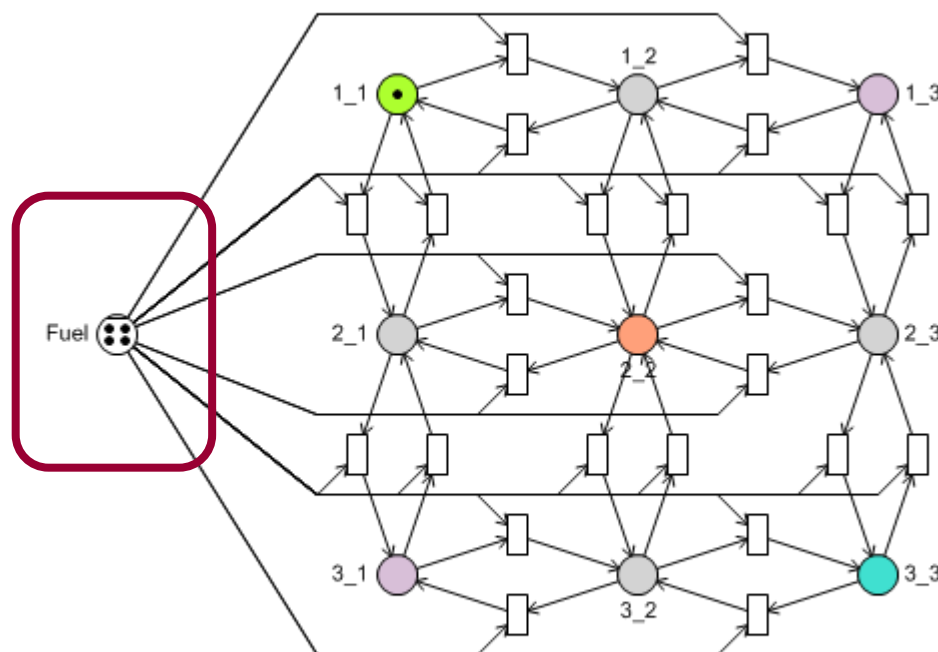
(1,1) 	(1,2)	(1,3) 
(2,1)	(2,2) 	(2,3)
(3,1) 	(3,2)	(3,3) 

Modellezés

1. lépés: cellák és mozgások (honnan hová)

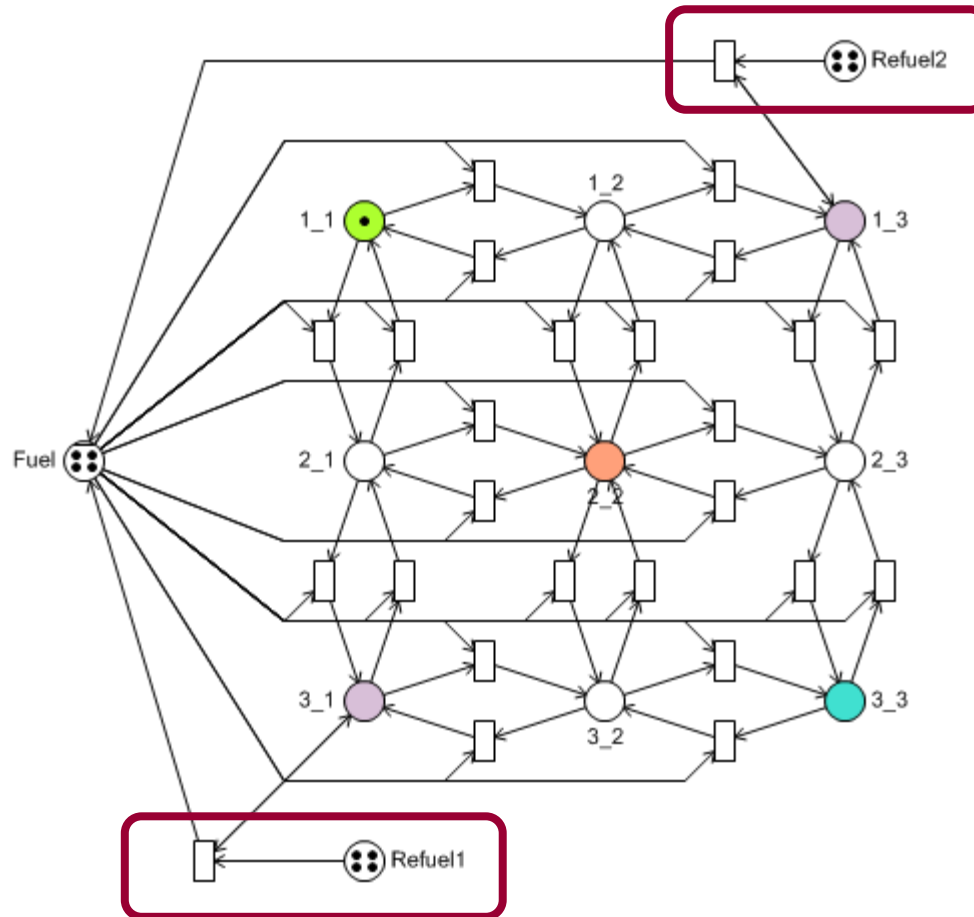


2. lépés: mozgás energiaigénye



Még hiányzik: A drón feltöltési lehetősége

A teljes modell



- PN helyek: Cellák, töltöttség, töltőállomások
- PN tranzíciók: Drón mozgása (honnan-hová, töltést igényel), drón utántöltése (töltőállomáson)

A modell analízise

- Szimuláció: Mozgás, töltés kifogyhat (holtpont)
- A modellezés helyességének ellenőrzése
 - Nőhet-e a tokenek száma (ld. drón, üzemanyag) a modellben?
 - Korlátos-e a modell? Melyek az egyes helyek tokenszám korlátai?
 - A drón mindig csak egy cellában tartózkodhat-e?
 - P-invariáns 1-es súlyokkal lefedi az összes cellát:
 $\{1 \times 1_1, 1 \times 1_2, 1 \times 1_3, 1 \times 2_1, 1 \times 2_2, 1 \times 2_3, 1 \times 3_1, 1 \times 3_2, 1 \times 3_3\}$
Kezdőállapotban 1 token volt, így a tokenek összege a cellák helyein mindig 1
- A viselkedés ellenőrzése
 - Leállhat-e a drón a jégmezőn (kifogyhat-e a töltése)?
 - Van-e holtpont?
 - Eljuthat-e egyáltalán a drón a segélycsomagért a (3,3) cellába?
 - CTL modellellenőrzés: $EF(Net1.3_3 > 0)$
 - Elvégezheti-e a drón a missziót, majd a bázisra visszatérhet-e?
Bázis (1,1) – segélycsomag (3,3) – baleset helye (2,2) – bázis (1,1)
 - CTL: $EF(Net1.3_3 > 0 \ \& \ (EF(Net1.2_2 > 0 \ \& \ EF(Net1.1_1 > 0))))$