

# Felismerő P rendszerek

Problémák eldöntéséhez **felismerő P rendszereket** használunk:

- Rendelkezik egy kijelölt **bemeneti membránnal** és két kitüntetett objektummal: **yes** és **no**
- **Minden számításuk megáll** és az utolsó lépésben – és csak akkor – a **yes vagy no** szimbólumot küldi a környezetbe

Egyetlen felismerő P rendszer nem képes tetszőleges méretű bemenet kezelésére, a problémákat ilyen rendszerek **uniform családjával** oldjuk meg

- Egy  $\Pi = \{\Pi(n) \mid n \geq 1\}$  felismerő P rendszer család **polinomiálisan uniform** ha van olyan polinom idejű Turing-gép ami
- $1^n$ -ből kiszámolja  $\Pi(n)$  egy leírását és
- a problémák bemeneteit kódoló szavakból kiszámolja az ezeknek megfelelő objektum-multihalmazokat

# Problémák eldöntése P rendszerekkel

Egy  $P$  probléma  $f(n)$  időben eldönthető felismerő  $P$  rendszerek egy polinomiálisan uniform  $\Pi = \{\Pi(n) \mid n \geq 1\}$  családjával, ha  $P$  minden  $n$  méretű  $I$  bemenetére,

- $\Pi(n)$ -t elindítva az  $I$  elkódolásával a bemeneti membránjában,
- $\Pi(n)$  legfeljebb  $f(n)$  lépés alatt megáll és
- pontosan akkor küld a környezetbe *yes*-t, ha  $I$  pozitív bemenete  $P$ -nek

$\text{PMC}_{\mathcal{R}}$  jelöli azon problémák osztályát, melyek **polinom időben eldönthetők**  $\mathcal{R}$ -beli felismerő  $P$  rendszerek polinomiálisan uniform családjával

- Mi a továbbiakban a  $\text{PMC}_{\mathcal{AM}}$  osztályt vizsgáljuk, ahol  $\mathcal{AM}$  az **aktív membrános**  $P$  rendszerek osztályát jelöli

# A SAT eldöntése P rendszerekkel

**Tétel:**  $\text{SAT} \in \text{PMC}_{\mathcal{AM}}$

**Bizonyítás:**

- Legyen  $F = \{C_1, \dots, C_m\}$  egy  $n$  változót és  $m$  klózt tartalmazó KNF
- $n, m$ -nek megfeleltethető egy természetes szám, jelöljük ezt  $\langle n, m \rangle$ -mel
  - Vannak erre megfelelő,  $\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  típusú, injektív, polinom időben kiszámítható függvények
- Kell egy polinom idejű  $M$  Turing-gép, ami
  - az  $1^{\langle n, m \rangle}$  szóból megkonstruálja a  $\Pi(\langle n, m \rangle)$  membránrendszert, ami  $F$  kielégíthetőségét dönti el
  - és megkonstruálja a  $\text{cod}(F)$ -et, azaz  $F$  objektumokkal való elkódolását
- $M$ -et nem fogjuk explicit módon megkonstruálni

# A SAT eldöntése P rendszerekkel

- **$F$  elkódolása:**

$$\text{cod}(F) = \{p_{ij} \mid p_j \in C_i\} \cup \{\overline{p_{ij}} \mid \neg p_j \in C_i\}$$

- **Példa:** Legyen  $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}$

- Akkor  $\text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$

- A továbbiakban  $\Pi(\langle 2, 3 \rangle)$  működését vázoljuk  $\text{cod}(F)$ -en

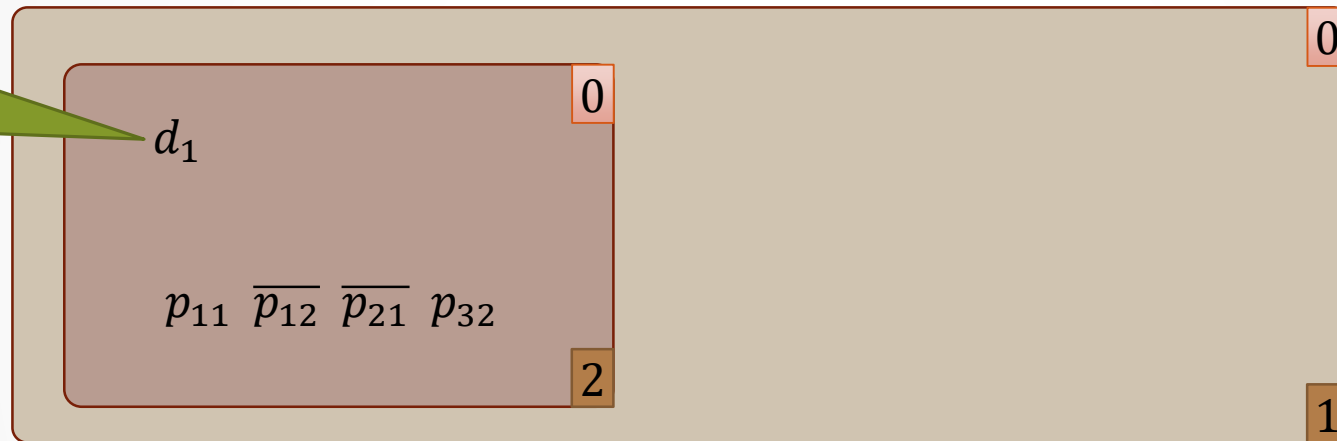
# A SAT eldöntése P rendszerekkel

- $\Pi(\langle 2,3 \rangle)$  a  $cod(F)$  multihalmazzal elindítva a következőt teszi:
  - membránosztással elkészít a  $p_1$  és  $p_2$  változó **összes interpretációjához** egy-egy membránt (tehát összesen 4 membránt)
  - közben ezekben a membránokban előállít olyan objektumokat, melyek a **megfelelő interpretációban igaz klózokat** reprezentálják
  - ezután minden membránban addig **számolja** az ezen klózokat reprezentáló objektumokat, amíg nem talál egy olyan klózt, ami nincs reprezentálva
  - ha talál olyan membránt, amiben **mind a három klóz reprezentálva van**, akkor **yes**-t küld a környezetben (hisz ekkor van olyan értékadás ami **kielégíti az összes klózt, azaz  $F$ -et**)
  - egyébként **no**-t küld a környezetbe

# A SAT megoldása - 1

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}$ ,  $cod(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$

$d_1$  szétosztja  
ezt a  
membránt

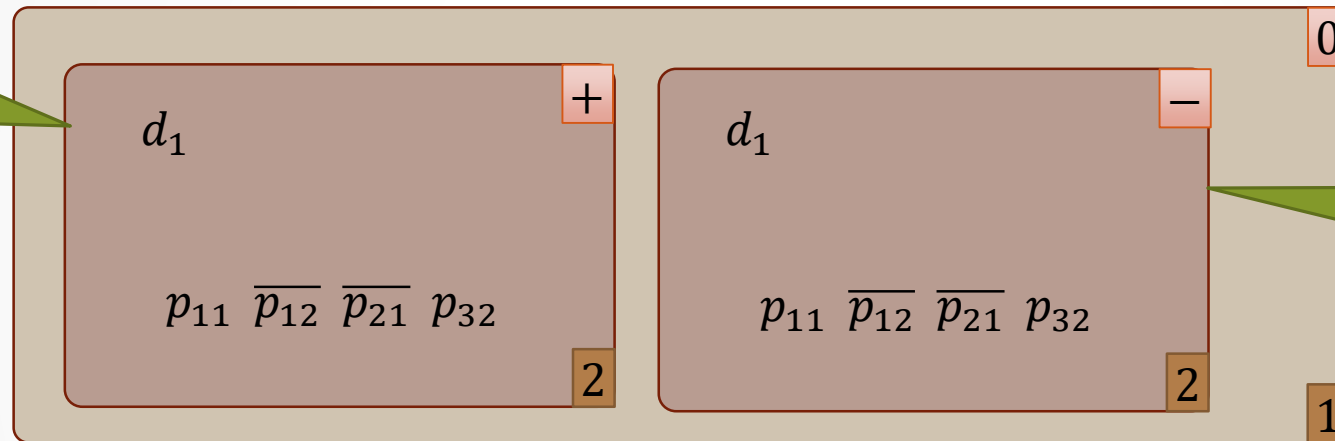


Felhasznált szabályok:  $[d_1]_2^0 \rightarrow [d_1]_2^+ [d_1]_2^-$

# A SAT megoldása - 1

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}$ ,  $cod(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$

Ez a membrán felel meg a  $p_1 = igaz$ -nak



Ez meg a  $p_1 = hamis$ -nak

Felhasznált szabályok:  $[d_1]_2^0 \rightarrow [d_1]_2^+ [d_1]_2^-$

# A SAT megoldása - 2

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$

Kimegy a régióból, semlegesre állítja a polarizációt

Itt ez a literál igaz, ezért evolválódik azzá a klózzá amit kielégít

Eltűnik, mert nem elégít ki egy klózt sem

Ez is eltűnik

Kimegy a régióból, semlegesre állítja a polarizációt

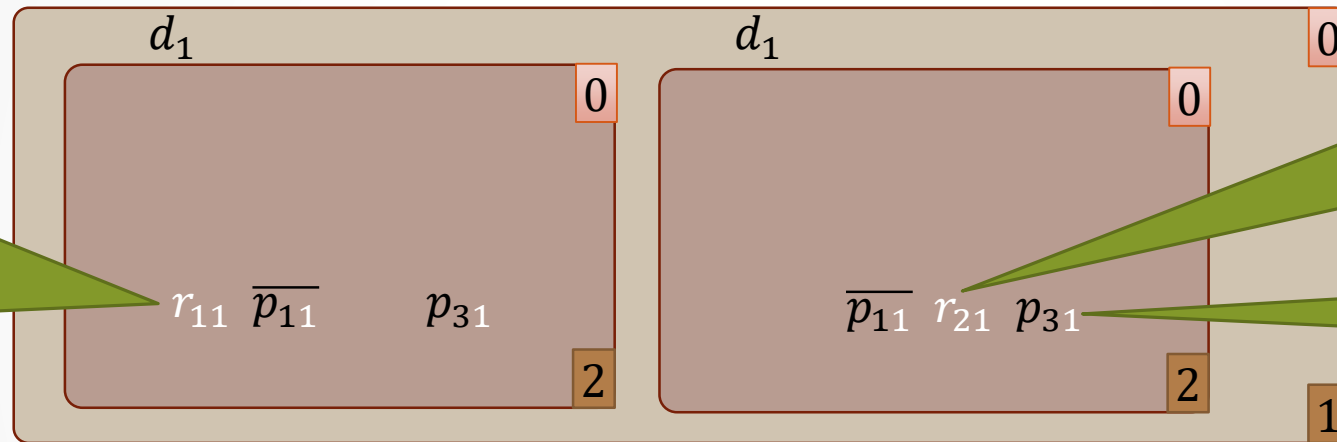
Itt ez a literál hamis, ezért evolválódik azzá a klózzá amit kielégít

Felhasznált szabályok:  $[d_1]_2^+ \rightarrow [\ ]_2^0 d_1$ ,  $[d_1]_2^- \rightarrow [\ ]_2^0 d_1$   
 $[p_{11} \rightarrow r_{11}]_2^+$ ,  $[\overline{p_{21}} \rightarrow \varepsilon]_2^+$ ,  $[p_{11} \rightarrow \varepsilon]_2^-$ ,  $[\overline{p_{21}} \rightarrow r_{21}]_2^-$ ,  
 $[\overline{p_{12}} \rightarrow \overline{p_{11}}]_2^+$ ,  $[p_{32} \rightarrow p_{31}]_2^+$ ,  $[\overline{p_{12}} \rightarrow \overline{p_{11}}]_2^-$ ,  $[p_{32} \rightarrow p_{31}]_2^-$

# A SAT megoldása - 2

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}$ ,  $cod(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$

Azt jelzi, hogy az 1-es klózt kielégíti a  $p_1 = igaz$  (a második index egy számláló)



Azt jelzi, hogy a 2-es klózt kielégíti a  $p_1 = hamis$

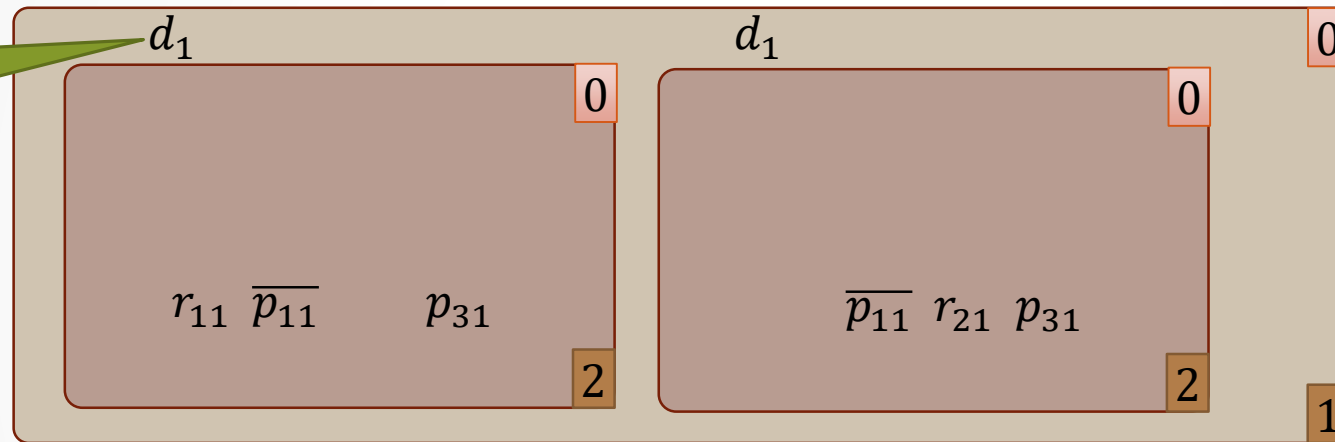
$p_2$ -ből lett az új  $p_1$ ; ugyanígy a másik membránban is

Felhasznált szabályok:  $[d_1]_2^+ \rightarrow [ ]_2^0 d_1$ ,  $[d_1]_2^- \rightarrow [ ]_2^0 d_1$   
 $[p_{11} \rightarrow r_{11}]_2^+$ ,  $[\overline{p_{21}} \rightarrow \varepsilon]_2^+$ ,  $[p_{11} \rightarrow \varepsilon]_2^-$ ,  $[\overline{p_{21}} \rightarrow r_{21}]_2^-$ ,  
 $[\overline{p_{12}} \rightarrow \overline{p_{11}}]_2^+$ ,  $[p_{32} \rightarrow p_{31}]_2^+$ ,  $[\overline{p_{12}} \rightarrow \overline{p_{11}}]_2^-$ ,  $[p_{32} \rightarrow p_{31}]_2^-$

# A SAT megoldása - 3

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}$ ,  $cod(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$

Visszamennek a membránokba

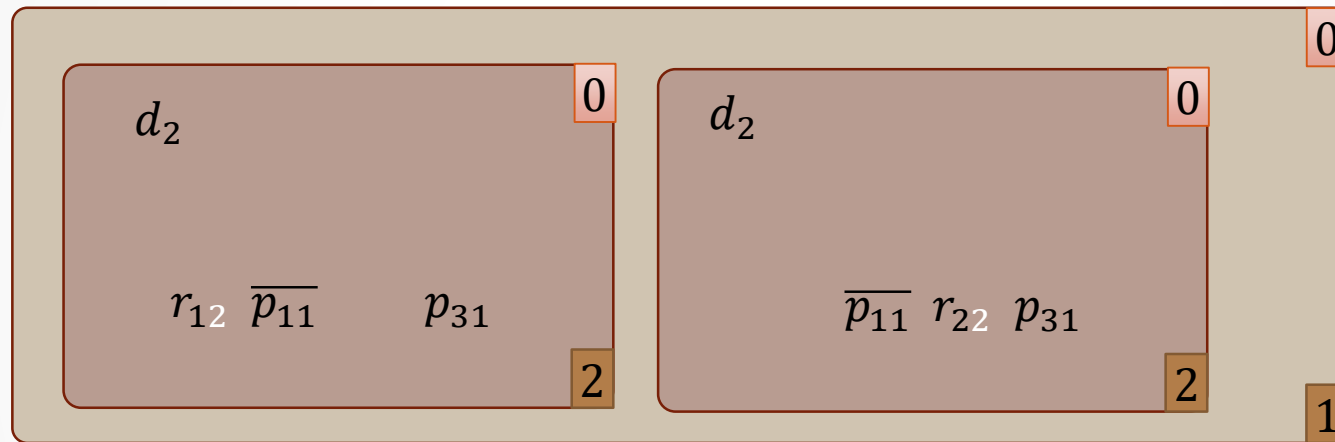


Felhasznált szabályok:  $d_1[ ]_2^0 \rightarrow [d_2]_2^0$ ,  
 $[r_{11} \rightarrow r_{12}]_2^0, [r_{21} \rightarrow r_{22}]_2^0$

Növelik az  $r$ -ek második indexét

# A SAT megoldása - 3

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}$ ,  $cod(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$



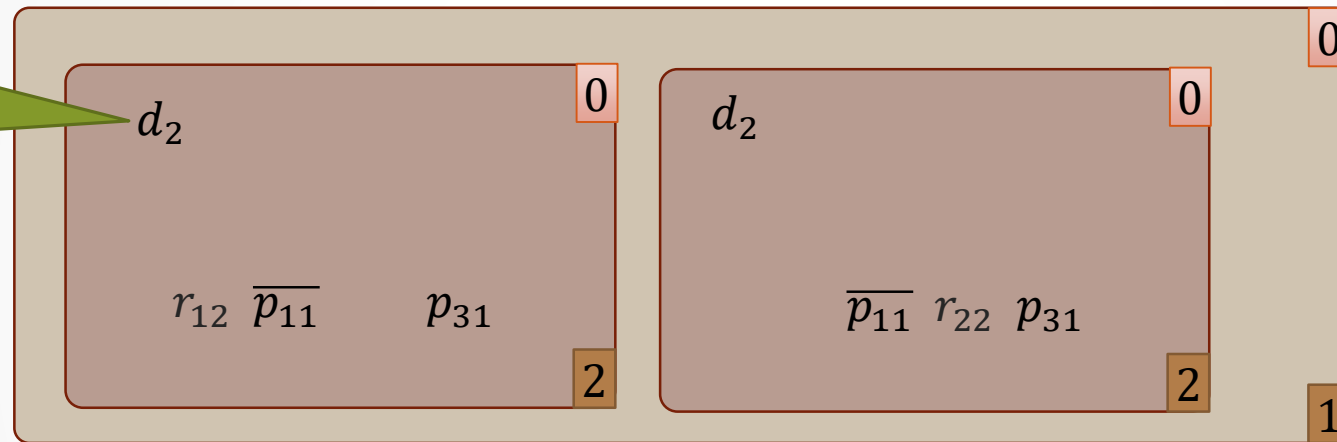
A  $d_1$ -ek bementek a régiókba, közben  $d_2$ -re változtak; az  $r$ -ek második indexe növelve

Felhasznált szabályok:  $d_1[ ]_2^0 \rightarrow [d_2]_2^0$ ,  
 $[r_{11} \rightarrow r_{12}]_2^0, [r_{21} \rightarrow r_{22}]_2^0$

# A SAT megoldása - 4

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}$ ,  $cod(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$

A  $d_2$ -k  
szétosztják a  
membránokat



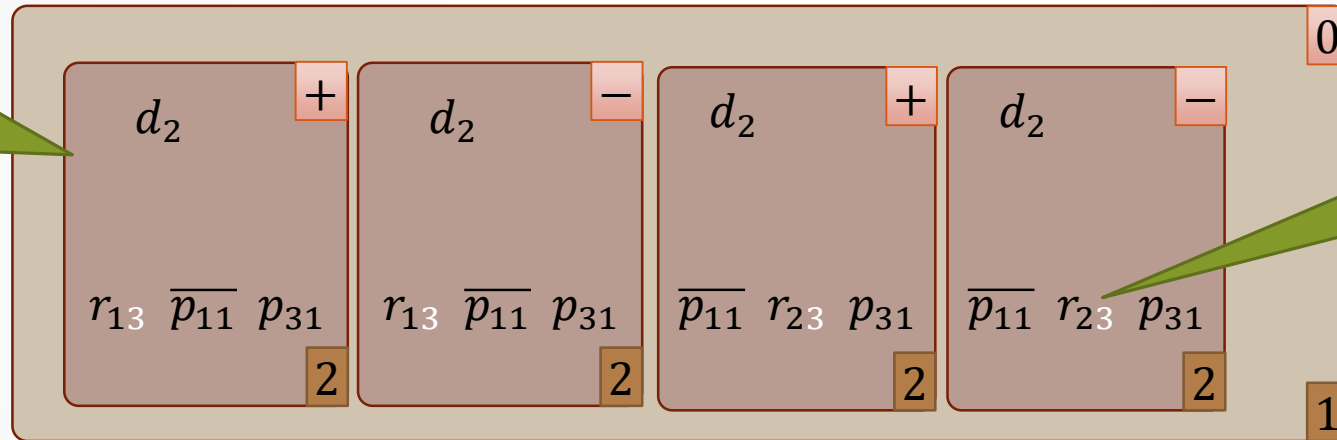
Felhasznált szabályok:  $[d_1]_2^0 \rightarrow [d_1]_2^+ [d_1]_2^-$ ,  
 $[r_{12} \rightarrow r_{13}]_2^0, [r_{22} \rightarrow r_{23}]_2^0$

Az  $r$ -ek indexe  
növekszik

# A SAT megoldása - 4

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$

Ez a membrán felel meg a  $p_1 = \text{igaz}, p_2 = \text{igaz}$ -nak



Az  $r$ -ek második indexe 1-gyel növelve

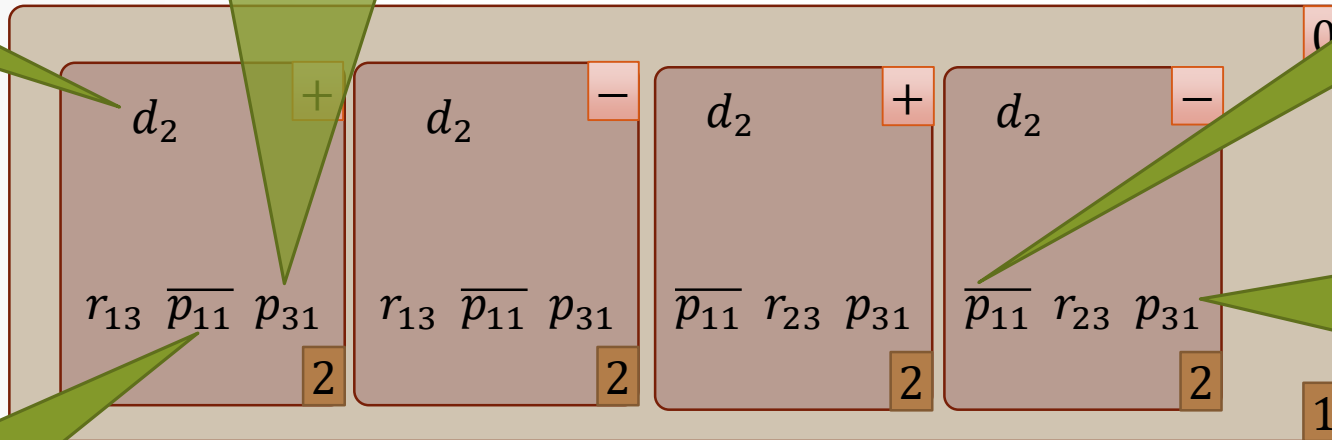
Felhasznált szabályok:  $[d_1]_2^0 \rightarrow [d_1]_2^+ [d_1]_2^-$ ,  
 $[r_{12} \rightarrow r_{13}]_2^0, [r_{22} \rightarrow r_{23}]_2^0$

# A SAT megoldása - 5

Megjegyezzük a klózt amit kielégít;  
hasonlóan a harmadik 2-es membránban is

$$F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$$

Kimennek a régiókból



Megjegyezzük a klózt amit kielégít;  
hasonlóan a második 2-esben is

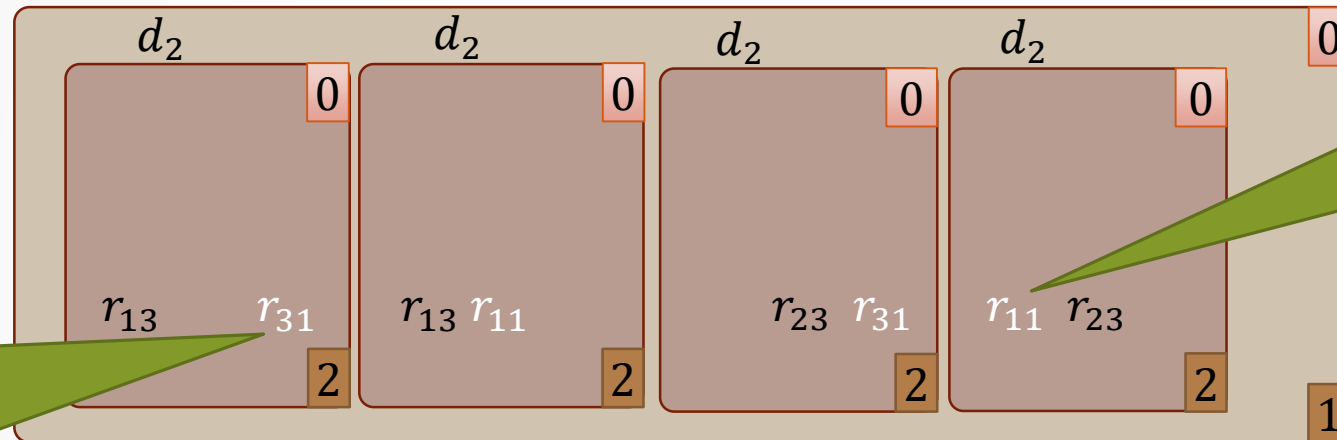
Eltűnik, mert nem elégít ki egy klózt sem;  
hasonlóan eltűnik a második 2-es membránból is

Eltűnik, mert nem elégít ki egy klózt sem;  
hasonlóan eltűnik a harmadik 2-es membránból is

Felhasznált szabályok:  $[d_2]_2^+ \rightarrow [ ]_2^0 d_2$ ,  $[d_2]_2^- \rightarrow [ ]_2^0 d_2$   
 $[p_{31} \rightarrow r_{31}]_2^+$ ,  $[\overline{p_{11}} \rightarrow \varepsilon]_2^+$ ,  $[p_{31} \rightarrow \varepsilon]_2^-$ ,  $[\overline{p_{11}} \rightarrow r_{11}]_2^-$

# A SAT megoldása - 5

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$



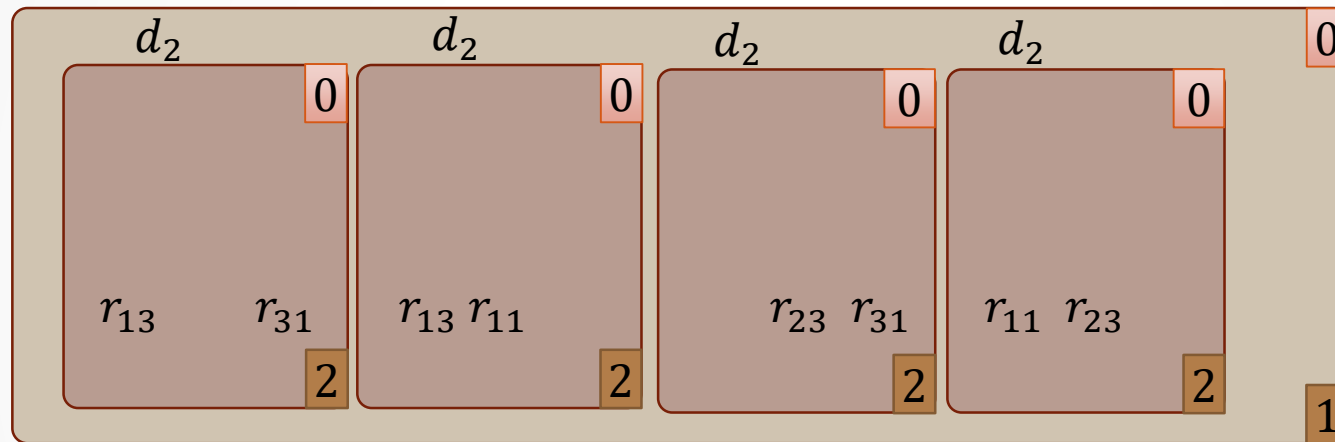
Azt jelzi, hogy az 3-as klózt kielégíti a  $p_2 = igaz$ ; hasonlóan igaz a 3-ik 2-es membránra

Azt jelzi, hogy az 1-es klózt kielégíti a  $p_2 = hamis$ ; hasonlóan igaz a 2-ik 2-es membránra

Felhasznált szabályok:  $[d_2]_2^+ \rightarrow [ ]_2^0 d_2, [d_2]_2^- \rightarrow [ ]_2^0 d_2$   
 $[p_{31} \rightarrow r_{31}]_2^+, [\overline{p_{11}} \rightarrow \varepsilon]_2^+, [p_{31} \rightarrow \varepsilon]_2^-, [\overline{p_{11}} \rightarrow r_{11}]_2^-$

# A SAT megoldása - 6

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$



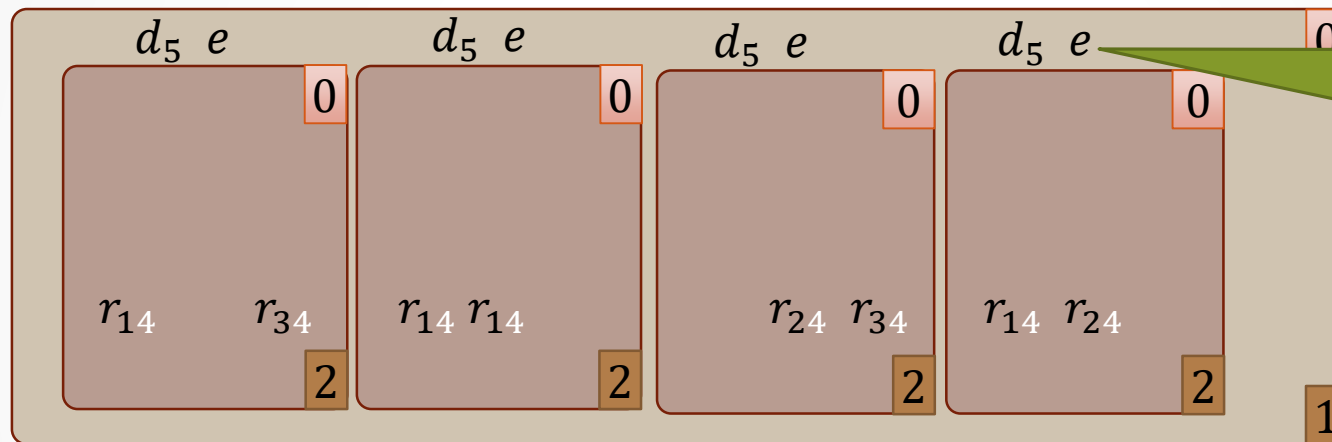
Felhasznált szabályok:

$[d_2 \rightarrow \dots \rightarrow d_4]_1^0, [d_4 \rightarrow d_5 \text{ e}]_1^0, [r_{ij} \rightarrow \dots r_{i4}]_2^0 \ (i = 1,2)$

Elszámol  $d_5$ -ig, ezalatt minden  $r$ -nek 4 lesz a második indexe

# A SAT megoldása - 6

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$



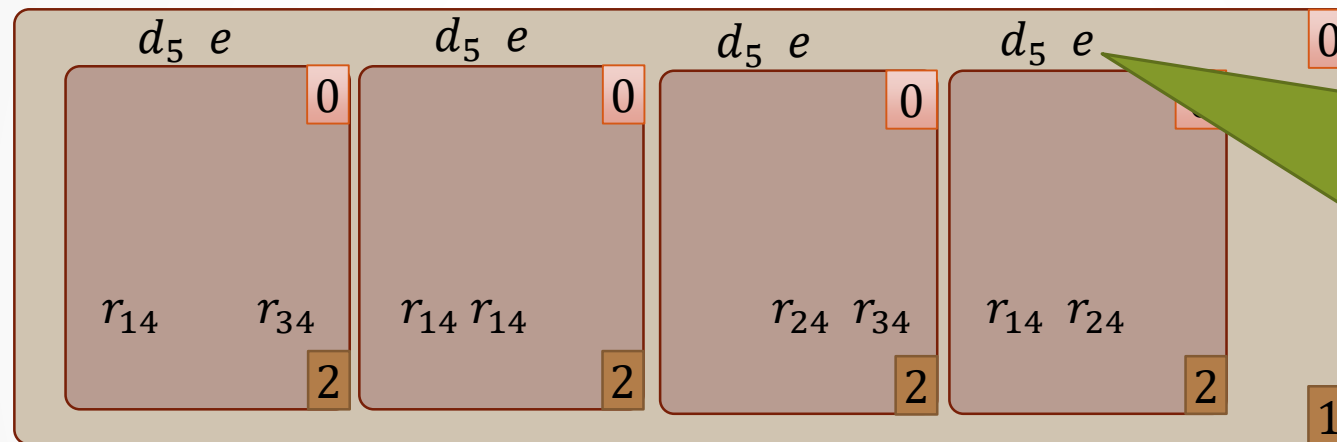
Megjelenik annyi  $e$  amennyi 2-es membrán van

Felhasznált szabályok:

$$[d_2 \rightarrow \dots \rightarrow d_4]_1^0, [d_4 \rightarrow d_5 e]_1^0, [r_{ij} \rightarrow \dots r_{i4}]_2^0 \quad (i = 1, 2)$$

# A SAT megoldása - 7

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$

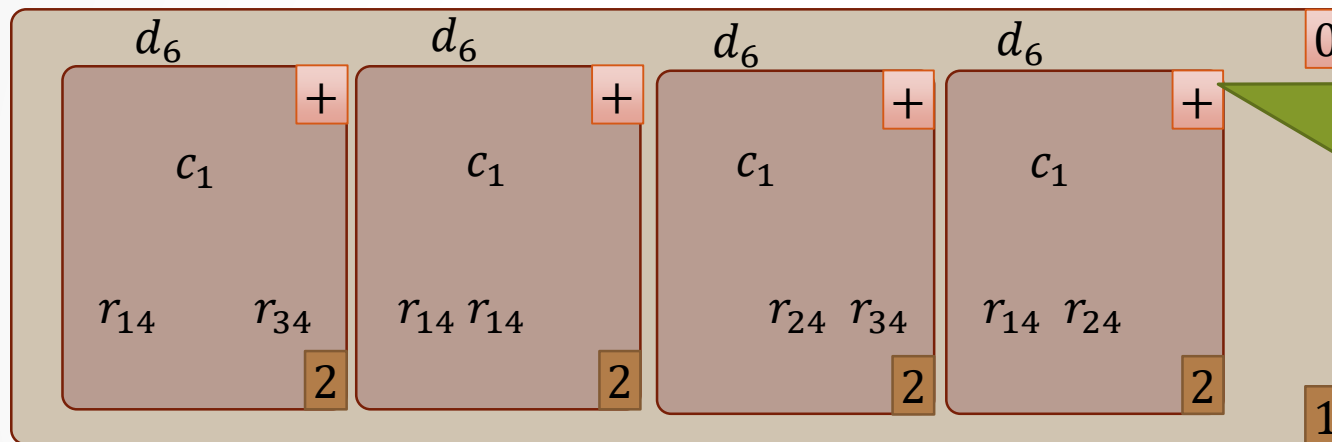


Minden 2-es membránba bemegy egy  $e$  és  $c_1$  lesz belőle; a  $c$ -k indexe fogja számolni, hogy mennyi klóz (azaz különböző  $r$  van a membránban)

Felhasznált szabályok:  $[d_5 \rightarrow d_6]_1^0, e[ ]_2^0 \rightarrow [c_1]_2^+$

# A SAT megoldása - 7

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$

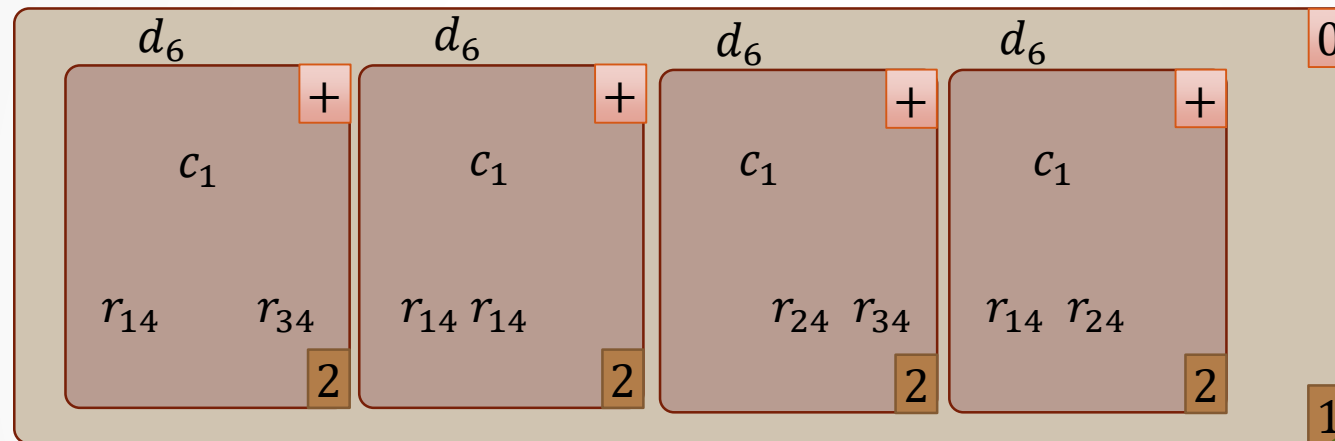


A 2-es membránok címkéje pozitívvá vált; ez teszi majd lehetővé azt, hogy az 1-es első indexű  $r$ -ek kimenjenek

Felhasznált szabályok:  $[d_5 \rightarrow d_6]_1^0, e[ ]_2^0 \rightarrow [c_1]_2^+$

# A SAT megoldása - 8

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$



Felhasznált szabályok:  $[d_6 \rightarrow d_7]_1^0, [r_{1,2n}]_2^+ \rightarrow [ ]_2^- r_{1,2n}$ ,

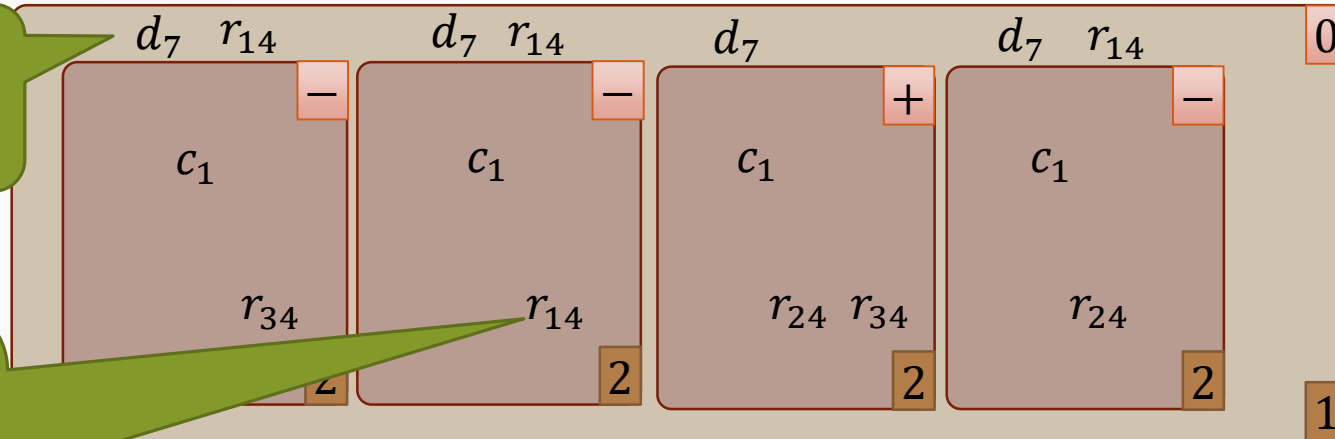
Kiküldi az 1-es klóznak megfelelő (1-es első indexű)  $r$ -et és negatívra állítja a polarizációt

# A SAT megoldása - 8

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}$ ,  $cod(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$

A  $d$ -k indexe közben  
nőtt 1-gyel

Csak egy  $r$  mehet  
ki; ha van még egy  
1-es első indexű  $r$   
az bent marad, és  
később  
haszontalanná  
válk

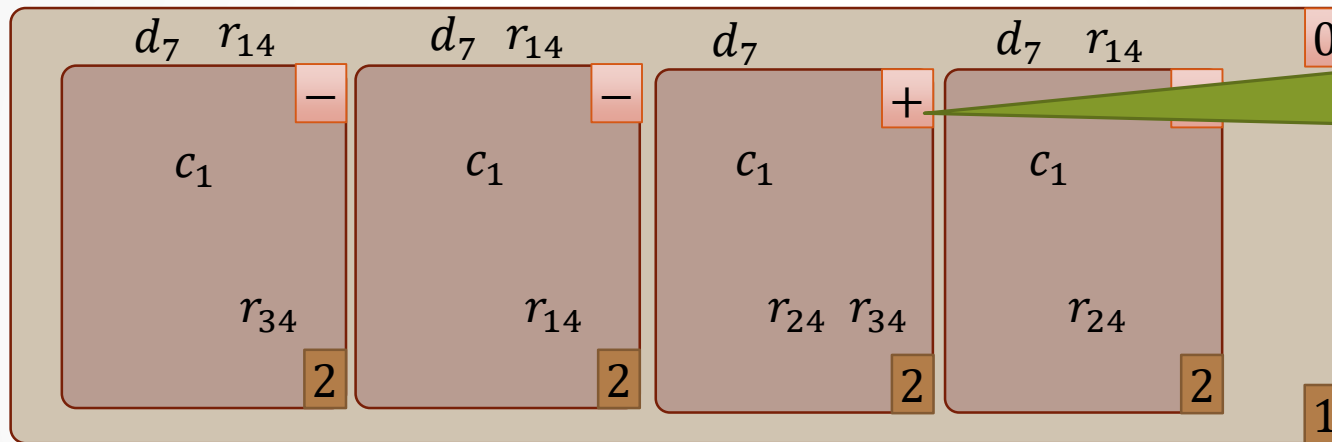


Felhasznált szabályok:  $[d_6 \rightarrow d_7]_1^0, [r_{1,2n}]_2^+ \rightarrow [ ]_2^- r_{1,2n}$

# A SAT megoldása - 9

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$

Beküld egy  $r$ -et a negatív 2-esekbe és pozitívrá állítja a polarizációt; közben a negatív 2-esekben a  $c$  indexe nő egygel



Itt nem fog nőni a  $c$  indexe a pozitív polaritás miatt

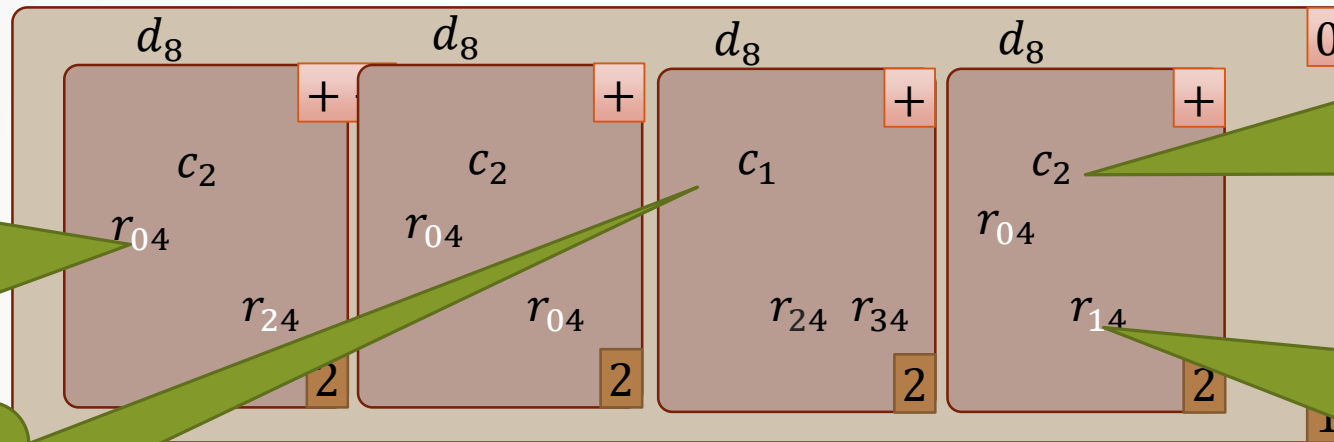
Felhasznált szabályok:  $[d_7 \rightarrow d_8]_1^0$ ,  
 $r_{14}[\ ]_2^- \rightarrow [r_{04}]_2^+, [c_1 \rightarrow c_2]_2^-, [r_{i4} \rightarrow r_{i-1,4}]_2^- \ (i = 1, \dots, 3)$

# A SAT megoldása - 9

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$

A visszaérkező  $r$ -ek első indexe 0 lesz, rájuk nincs tovább szükség

Itt már soha nem lesz 1-es indexű  $r$ , tehát ebben a membránban a számítás leáll



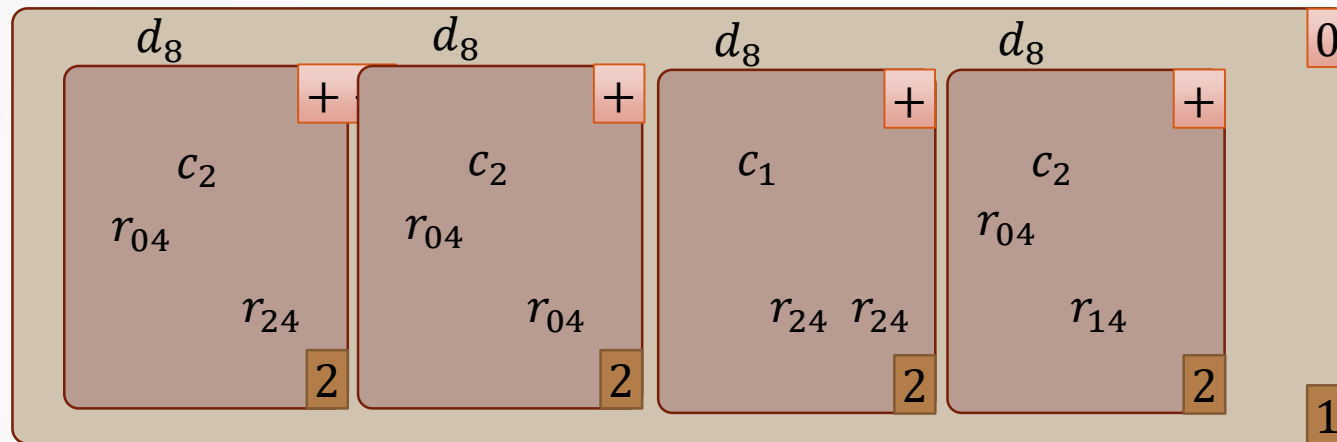
Azokban a 2-esekben ahol (egy lépésig) negatív volt a polarizáció  $c$  indexe tudott nőni egyet (első, második és harmadik 2-es membrán)

A többi  $r$  első indexe a negatív 2-esekben csökken egyet, így majd ezeket is ki tudja küldeni a rendszer (kivéve ha 0-ra csökken, mint a második membránban)

Felhasznált szabályok:  $[d_7 \rightarrow d_8]_1^0$ ,  
 $r_{14}[\ ]_2^- \rightarrow [r_{04}]_2^+, [c_1 \rightarrow c_2]_2^-, [r_{i4} \rightarrow r_{i-1,4}]_2^- \ (i = 1, \dots, 3)$

# A SAT megoldása - 10

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}$ ,  $cod(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$

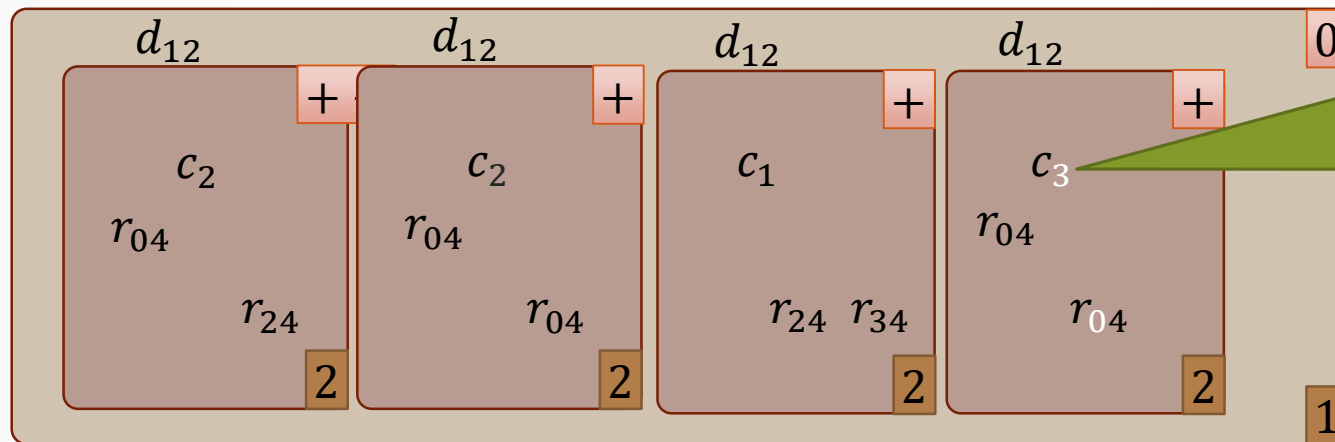


A 8 és 9 lépésben leírtak megismétlődnek még 2-szer az aktuális  $r$ -eken; a  $c$ -k számolják, hányszor ment ki  $r$

Felhasznált szabályok:  $[d_8 \rightarrow \dots \rightarrow d_{12}]_1^0$ , plusz a 8-9. lépés  $r_{ij}$ -ket és  $c_i$ -ket manipuláló megfelelő szabályai

# A SAT megoldása - 10

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$

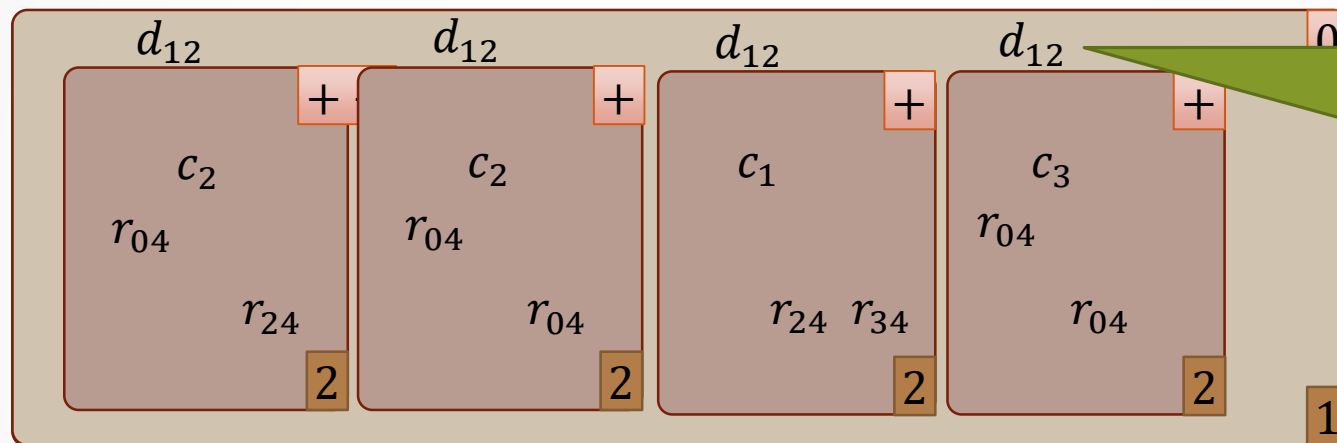


Ekkor a  $c$ -k indexe mínusz 1 az a szám ami megmutatja, mennyi különböző klóznak megfelelő  $r$  tudott kimenni egyes membránokból

Felhasznált szabályok:  $[d_8 \rightarrow \dots \rightarrow d_{12}]_1^0$ , plusz a 8-9. lépés  $r_{ij}$ -ket és  $c_i$ -ket manipuláló megfelelő szabályai

# A SAT megoldása - 11

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}$ ,  $cod(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$

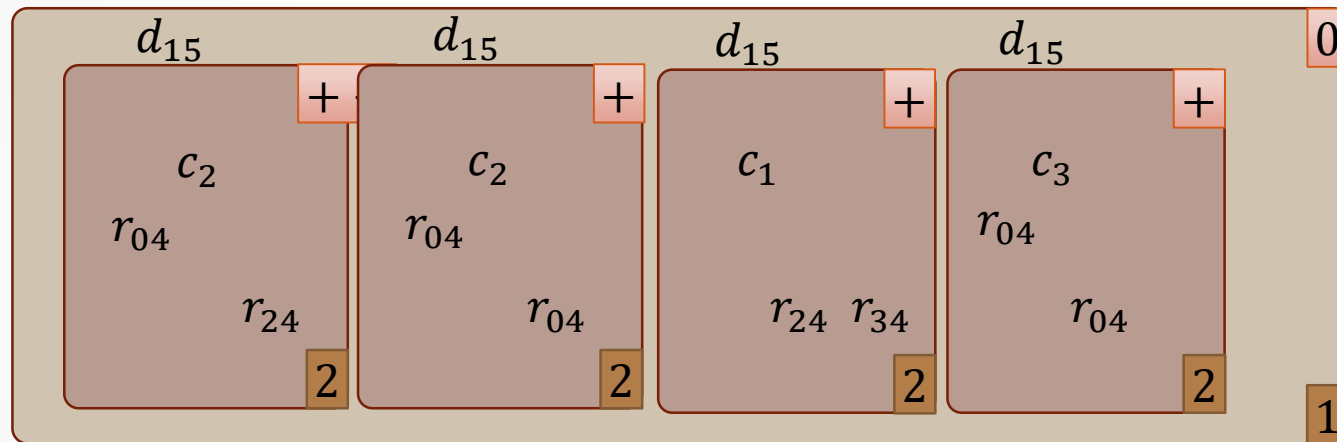


A  $d$ -k még számolnak pár lépésig azért, mert ha lenne olyan 2-es amiben összegyűlt az összes klóz, akkor időt kell adni annak, hogy ez kiderüljön

Felhasznált szabályok:  $[d_{12} \rightarrow \dots \rightarrow d_{15}]_1^0$

# A SAT megoldása - 11

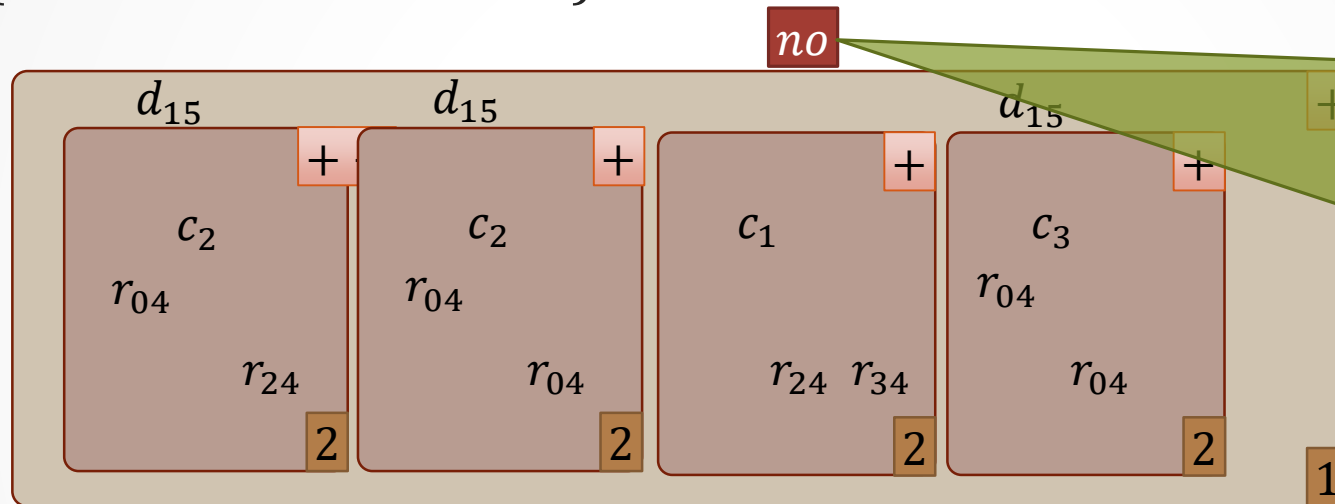
- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}$ ,  $cod(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$



Felhasznált szabályok:  $[d_{12} \rightarrow \dots \rightarrow d_{15}]_1^0$

# A SAT megoldása - 12

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}, \{p_2\}\}$ ,  $cod(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}, p_{32}\}$



Felhasznált szabályok:  $[d_{15}]_1^0 \rightarrow [ ]_1^+ no$

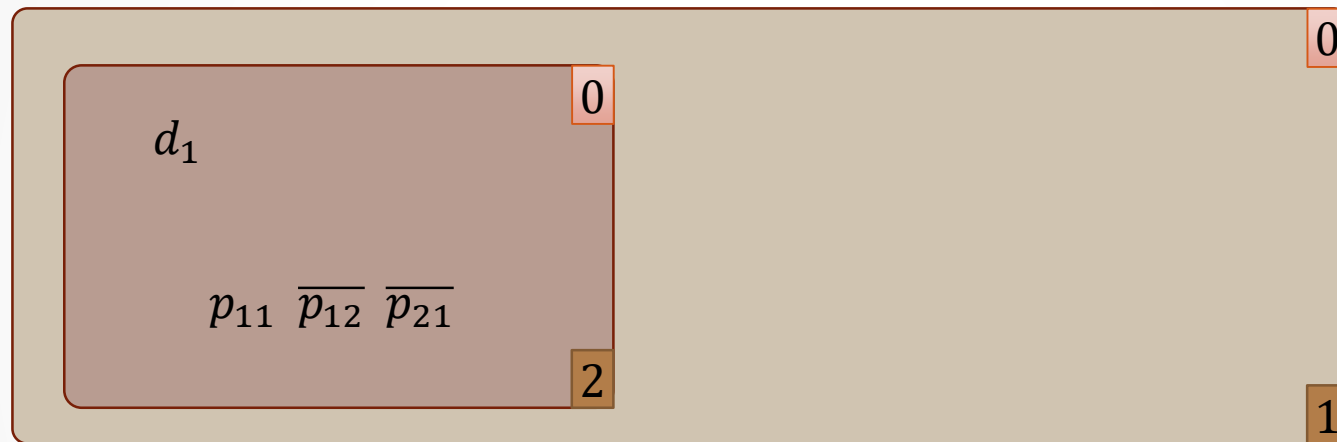
Mivel egyik 2-esben sem gyűlt össze az összes klóznak megfelelő  $r$ , azaz egyik változóértékkadás sem elégíti ki az összes klózt, a rendszer kiküldi a környezetbe a  $no$ -t; azaz a rendszer válasza arra, hogy  $F$  kielégíthető-e az, hogy nem az.

# A SAT megoldása – Másik példa

- Legyen most  $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}\}$
- $cod(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}\}$
- Most  $n = 2$  (változók száma) és  $m = 2$  (klózik száma)
- Most a  $\Pi(2,2)$  működését mutatjuk be  $cod(F)$ -el elindítva

# A SAT megoldása - 1

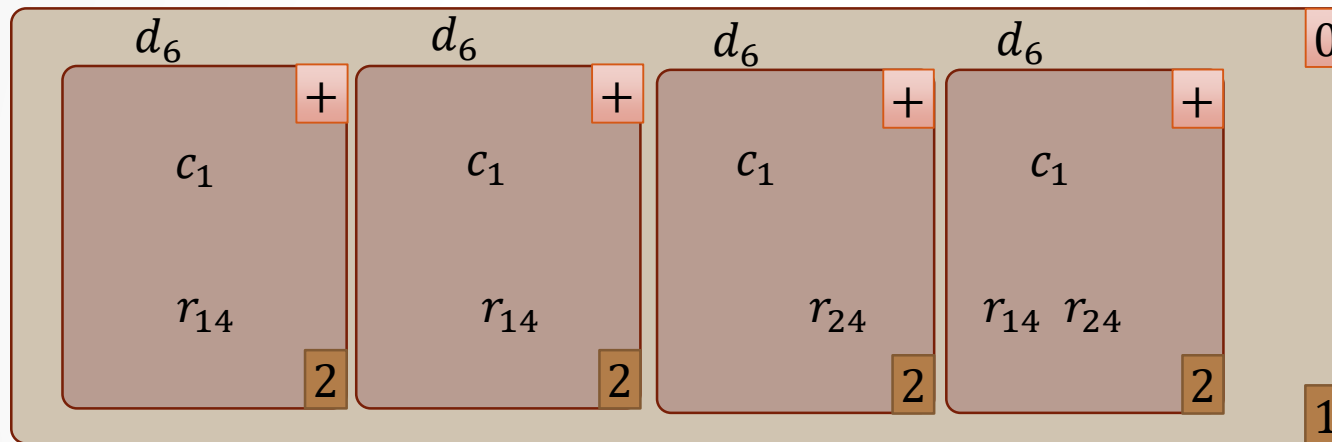
- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}\}$



Az előző példához hasonlóan folyik a számítás a következő konfigurációig

# A SAT megoldása – 2

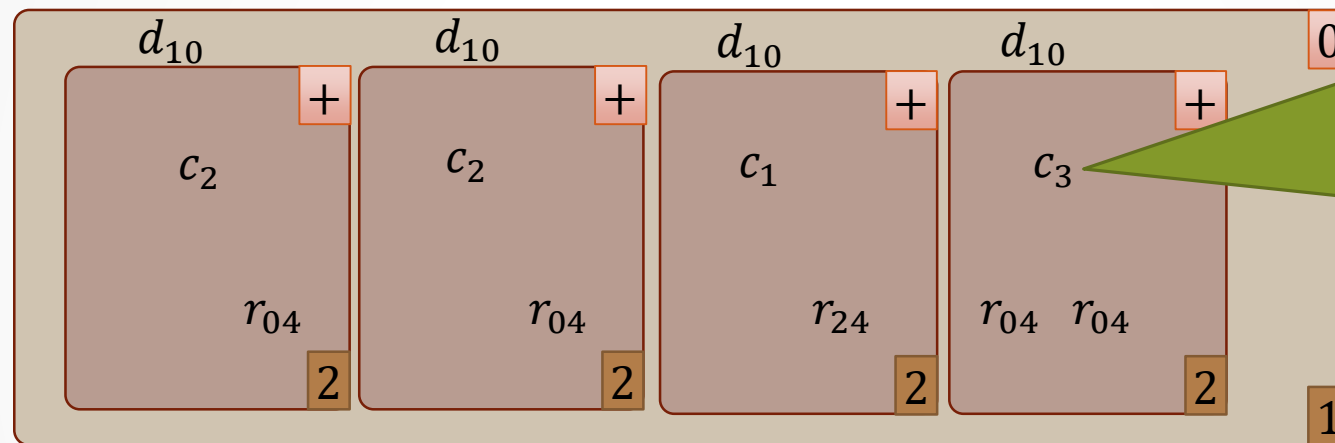
- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}\}$



Innen is az előző példához hasonlóan folyik a számítás a következő konfigurációig

# A SAT megoldása – 3

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}\}$

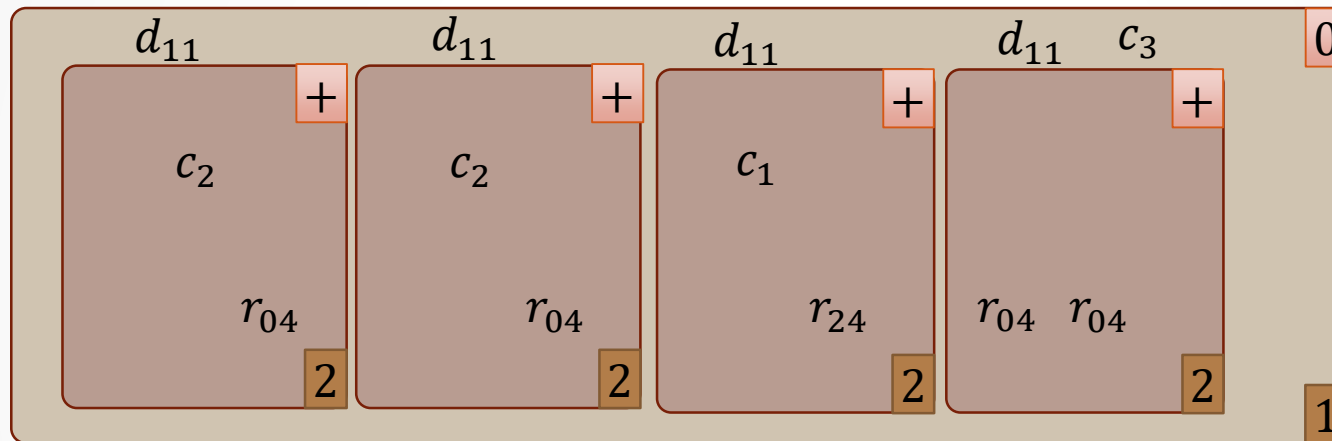


Felhasznált szabályok:  $[d_{10} \rightarrow d_{11}]_1^0, [c_3]_2^+ \rightarrow [ ]_2^+ c_3$

Egyedül ebben a 2-es membránban jön ki a  $c_3$ , ami azt jelenti, hogy az ennek megfelelő értékadás ( $p_1 = \text{hamis}$  és  $p_2 = \text{hamis}$ ) kielégíti az összes klózt

# A SAT megoldása – 3

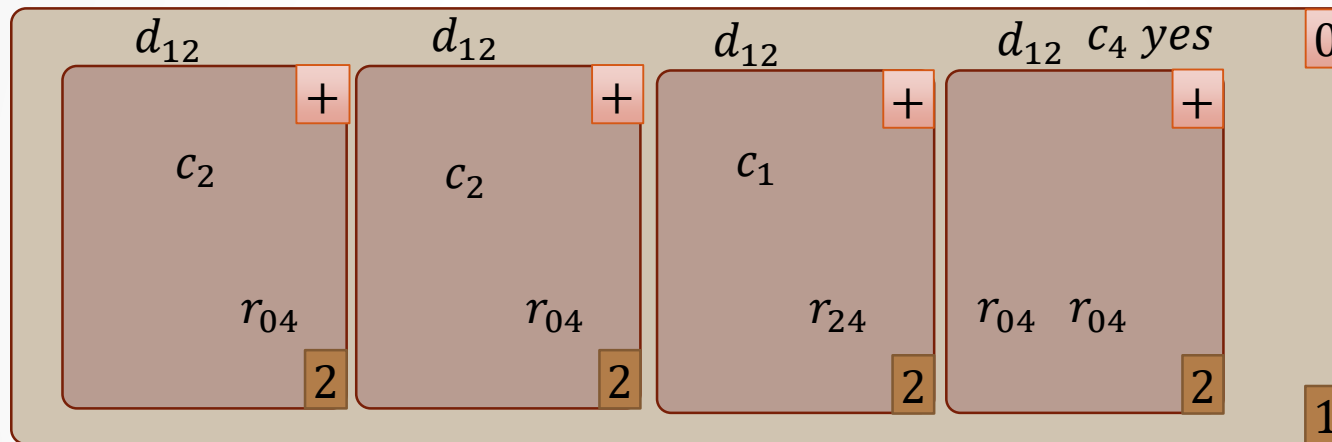
- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}\}$



Felhasznált szabályok:  $[d_{11} \rightarrow d_{12}]_1^0, [c_3 \rightarrow c_4 \text{ yes}]_2^+$

# A SAT megoldása – 4

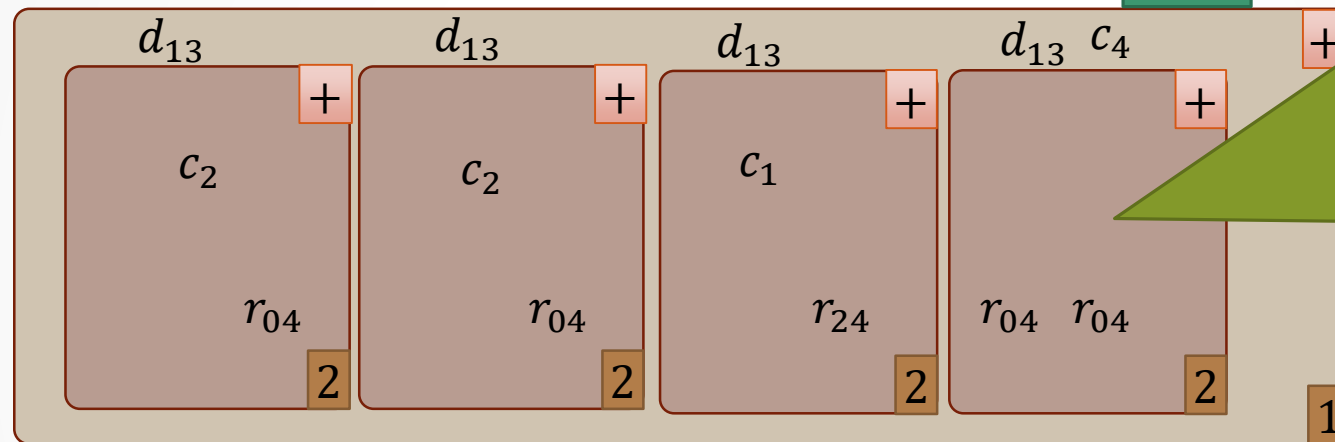
- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}\}$



Felhasznált szabályok:  $[d_{12} \rightarrow d_{13}]_1^0, [yes]_1^0 \rightarrow [ ]_1^+ yes$

# A SAT megoldása – 5

- $F = \{\{p_1, \neg p_2\}, \{\neg p_1\}\}, \text{cod}(F) = \{p_{11}, \overline{p_{12}}, \overline{p_{21}}\}$



Felhasznált szabályok:  $[d_{12} \rightarrow d_{13}]_1^0, [yes]_1^0 \rightarrow [ ]_1^+ yes$

- Mivel az utolsó 2-esben össze-gyűlt az összes klóznak megfelelő  $r$ , azaz a  $p_1 = hamis, p_2 = hamis$  kielégíti az összes klózt, a rendszer kiküldi a környezetbe a *yes*-t
- azaz a rendszer válasza arra, hogy  $F$  kielégíthető-e az, hogy igen
- Közben az 1-es membrán töltése pozitív lesz, így a benne lévő  $d$ -k nem tudnak *no*-vá válni a környezetben

# $\Pi(\langle n, m \rangle)$ futási ideje, uniformitás

- Futási idő:
  - Az  $n$  változó **összes interpretációjának** elkészítése  $O(n)$  lépés
  - Ezalatt minden membránban elkészülnek azon objektumok, melyek az adott membránhoz tartozó interpretáció által kielégített klózatokat reprezentálják
  - Annak eldöntése, hogy van-e olyan membrán melyben **mind az  $m$  klóz** reprezentálva van  $O(m)$  lépés
  - A rendszer futási ideje:  $O(n + m)$
- Uniformitás:
  - $\Pi(\langle n, m \rangle)$  minden komponense valamint valamint  $\text{cod}(F)$  is polinom méretű
  - Ezért megadható olyan polinom idejű  $M$  Turing-gép, ami ezeket megkonstruálja

# Aktív membrános P rendszerek – Számítási erő

- Egy  $L$  eldöntési probléma **komplementere** az az  $\bar{L}$  probléma melyre a következő teljesül
  - Az  $\bar{L}$  tetszőleges  $B$  bemenetére:
  - $B$  az  $\bar{L}$  **pozitív** bemenete  $\Leftrightarrow B$  az  $L$  **negatív** bemenete
- Például a  **$\overline{\text{SAT}}$**  a következő probléma:
  - adott egy  $\varphi$  konjunktív normálforma
  - Döntsük el, hogy  $\varphi$  **kielégíthetetlen-e**
- Ezt a problémát **UNSAT**-nak is szokás nevezni
  - **nem ismert** UNSAT-ot **polinom időben megoldó nemdeterminisztikus** Turing-gép
  - Az a **sejtés**, hogy **nem ugyanaz** a bonyolultsága mint a SAT-nak
- **coNP**: az NP-beli problémák komplementereinek osztálya
- Ami NP-teljes, az **coNP-teljes** is
  - UNSAT coNP-teljes

# Aktív membrános P rendszerek – NP-teljesség

Emlékeztető: Aktív membrános P rendszerekkel polinom időben eldönthető problémák osztálya

- **Tétel:**  $NP \cup coNP \subseteq \mathbf{PMC}_{\mathcal{AM}}$
- **Bizonyítás:**
  - SAT NP-teljes, ezért minden  $L$  NP-beli probléma visszavezethető SAT-ra polinom időben
  - Azaz egy tetszőleges NP-beli  $L$  nyelv egy tetszőleges  $B$  bemenetéhez polinom időben megadható egy  $F$  formula úgy, hogy
    - $B$  pozitív bemenet  $\Leftrightarrow F$  kielégíthető
  - $F$  kielégíthetősége pedig eldönthető polinom időben aktív membrános P rendszerrel
    - Polinom időben megkonstruáljuk  $\Pi(\langle n, m \rangle)$ -et, ahol  $n$  az  $F$ -ben szereplő változók,  $m$  pedig a klózik száma
    - $\Pi(\langle n, m \rangle)$  futási ideje pedig  $n + m$ ,
  - Azaz,  $L \in \mathbf{PMC}_{\mathcal{AM}}$ , vagyis  $NP \subseteq \mathbf{PMC}_{\mathcal{AM}}$
  - Továbbá, ha  $L \in \mathbf{PMC}_{\mathcal{AM}}$ , akkor  $\bar{L} \in \mathbf{PMC}_{\mathcal{AM}}$  (egyszerűen cseréljük fel az  $L$ -et eldöntő P rendszerben a *yes* és a *no* objektumokat)
  - Vagyis  $coNP \subseteq \mathbf{PMC}_{\mathcal{AM}}$  is teljesül