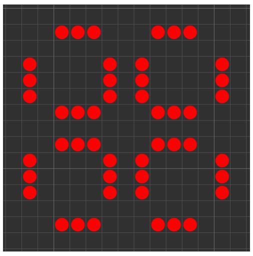
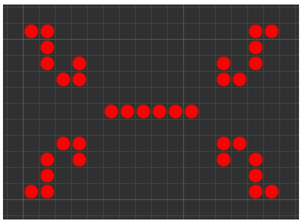
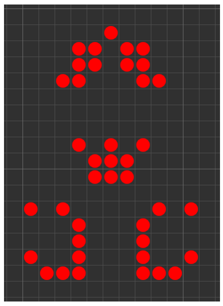
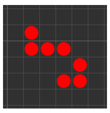
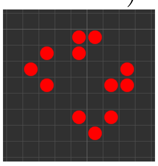
Számítási modellek - Második beadandó - Megoldások

2021.12.08.

Minden feladat 10 pontot ér. A megoldásokat a kolomax@inf.elte.hu címre kell elküldeni 2021.12.14. 23:59-ig.

1. Határozd meg az alábbi alakzatok kategóriáját a Conway-féle Élet Játékában (csendélet, oszcillátor, űrhajó, random). Az oszcillátorok és űrhajók esetében add meg a periódust is.

   a)  b)  c) 

   d)  e) 

   Megoldás: a) Oszcillátor, periódus=3; b) Oszcillátor, periódus=9; c) Űrhajó, periódus=12; d) Csendélet; e) Oszcillátor, periódus=4

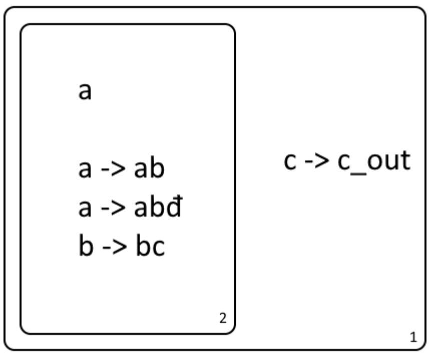
2. Legyen A egy két-dimenziós sejtautomata von Neumann szomszédsággal, ahol a sejtek állapotai a $Q = \{0, 1, 2, 3, 4\}$ halmaz elemei lehetnek, az állapotátmenet-függvény pedig az alábbi: $uj\_allapot := ((bal\_sz + jobb\_sz) * (felso\_sz + also\_sz) + aktualis\_allapot)\ (mod\ 5)$.
   Határozd meg a következő 10 generációt az alábbi konfigurációból kiindulva:

1	2	1
2	3	2
1	2	1

Megoldás:

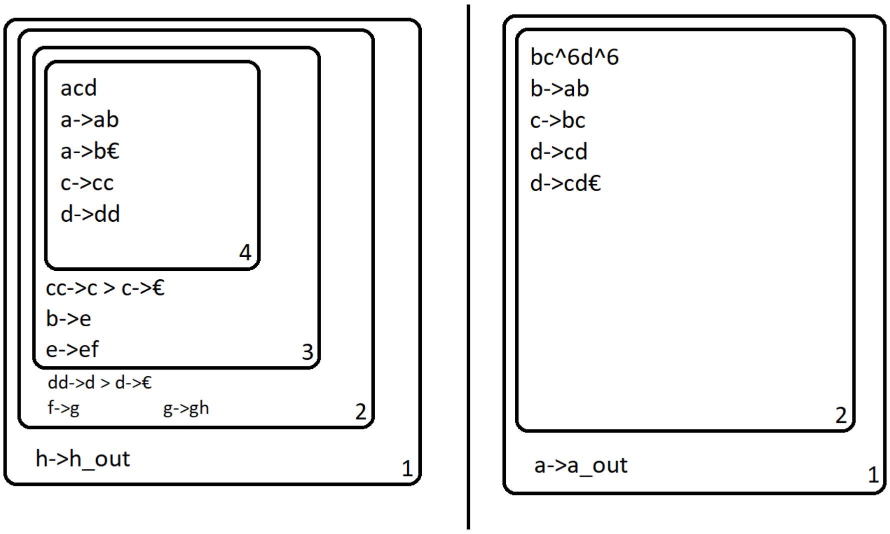
Gen 0			Gen 1			Gen 2			Gen 3			Gen 4			Gen 5		
1	2	1	0	3	0	4	3	4	3	3	3	2	4	2	3	2	3
2	3	2	3	4	3	3	0	3	3	1	3	4	2	4	2	1	2
1	2	1	0	3	0	4	3	4	3	3	3	2	4	2	3	2	3
			Gen 6			Gen 7			Gen 8			Gen 9			Gen 10		
			2	3	2	1	1	1	2	2	2	1	0	1	1	1	1
			3	2	3	1	3	1	2	2	2	0	3	0	1	3	1
			2	3	2	1	1	1	2	2	2	1	0	1	1	1	1

3. Adj meg egy membránrendszert ami a háromszögszámok halmazát generálja ($t_n = \sum_{i=1}^{n} i$).
   Megoldás:



   Magyarázat: A 2-es membránban (amíg fel nem oldódik) végig 1 darab a lesz jelen. Ez az 1 a minden lépés során visszatermeli magát és előállít 1 darab b-t ($a \to ab$). A b-k minden lépésben visszatermelik magukat és létrehoznak 1-1 darab c-t ($b \to bc$). Így a b-k száma minden lépés során eggyel nő, a c-k száma pedig annyival, amennyi b épp jelen van. Ily módon a membrán feloldásakor ($a \to ab\delta$) $0+1+2+\ldots+n$ darab c kerül az 1-es membránba, amik a következő lépés során mind kikerülnek a környezetbe ($c \to c_{out}$), és a számítás véget ér.

4. Adj meg egy membránrendszert ami a köbszámok halmazát generálja.
   Megoldás:



Magyarázat: A baloldali P rendszer az első n lépés során előállít n darab b valamint 2^n db c ill. d objektumot ($a \to ab, c \to cc, d \to dd$ szabályok), majd feloldódik ($a \to b\delta$ szabály az n-dik lépésben). A 3-as membrán az ezt követő $n+1$-dik lépésben oldódik fel (n lépésen keresztül felezgeti a c-k számát a $cc \to c$ szabállyal, majd a megmaradó 1 db c-re a $c \to \delta$ szabályt alkalmazza). Ezalatt az első lépésben az n db b átíródik fejenként 1, összesen n db e-re ($b \to e$). Az ezt követő n lépés mindegyikében keletkezik n darab f objektum, az e-k száma nem változik ($e \to ef$). Ezzel összesen n^2 db f jelenik meg a 2-es membránban, valamint az eddig tétlen 2^n db d. Hasonlóan a 3-as membránhoz, a d-k $n+1$ lépés alatt fogynak el és az utolsó feloldja a membránt, ezalatt az első lépésben létre jön n^2 db g, amik mind az n hátralévő lépés során létrehoznak n^2 darab h-t, összesen tehát n^3-öt. Utolsó lépésként ezek az objektumok kikerülnek a környezetbe.
A jobboldali P rendszer működése azon a tényen alapszik, hogy az egymást követő köbszámok $(0, 1, 8, 27, 64, \ldots)$ különbsége olyan sorozatot alkot $(1, 7, 19, 37, \ldots)$, ahol a különbségek 6 egymást követő többszörösei $(6, 12, 18, \ldots)$. Amíg a 2-es membrán fel nem oldódik a $d \to cd\delta$ szabály alkalmazásával, addig: a d-k száma állandó (6), a c-k száma 6-ról indulva hatosával növekszik ($d \to cd$), a b-k száma 1-ről indulva mindig annyival nő, ahány c van épp (6 többszörösei; $c \to bc$), az a-k száma pedig 0-ról indulva annyival nő, ahány b van a membránban ($b \to ab$). A feloldódás után az a-k egy lépésben kikerülnek a környezetbe.

5. Adj meg egy aktív membrános P rendszert, ami 2-hatvány inputokat fogad el (az input a^n valamelyik membránban elhelyezve, az output akkor és csak akkor lehet *yes*, ha $n = 2^k,\ k \in \mathbb{N}$).

Megoldás: $\Pi_5 = (O, H, \mu, w_1, w_2, w_3, i, R)$, ahol
$O = \{a, b, c, d, e, f, g, h, i, j\}$,
$H = \{1, 2, 3\}$,
$\mu = [_1[_2[_3]_3]_2]_1$,
$w_1 = \lambda$; $w_2 = b$; $w_3 = b$,
$i = 2$ (az input membrán címkéje; ide kerül be az elején a^n)
$R = \{$

1. $[b \to c]_3^0$
2. $a[\,]_3^0 \to [a]_3^+$
3. $[c]_3^0 \to no''$
4. $[c \to b]_3^+$
5. $[a]_3^+ \to [\,]_3^+ a$
6. $[d]_3^+ \to [\,]_3^+ d$
7. $e[\,]_3^+ \to [b]_3^0$
8. $[b]_3^+ \to [d]_3^+ [d]_3^+$
9. $[b]_2^0 \to [\,]_2^+ b$
10. $[no'' \to no']_2^+$
11. $[a]_2^+ \to [\,]_2^- a$
12. $j[\,]_2^+ \to [yes']_2^+$
13. $[no']_2^+ \to no'$
14. $[yes']_2^+ \to yes'$
15. $[d \to e]_2^-$
16. $a[\,]_2^- \to [a]_2^-$
17. $j[\,]_2^- \to [b]_2^0$
18. $[no']_2^- \to no'$
19. $[b \to g]_1^0$
20. $[g \to h]_1^0$
21. $[h \to i]_1^0$
22. $[i \to j]_1^0$
23. $[no']_1^0 \to [\,]_1^- no$
24. $[yes']_1^0 \to [\,]_1^+ yes$

Magyarázat: A rendszer ellenőrzi, hogy az a objektumok száma hogyan viszonyul a 3-as címkéjű membránok számához. Ha több a van, akkor megduplázza a 3-as membránok számát és újra ellenőriz, ha kevesebb, akkor *no* választ állít elő, ha megegyezik a kettő, akkor *yes* választ állít elő.
Tegyük fel, hogy a rendszer konfigurációja az alábbi:
$[_1\,[_2 a^n b\,[_3 b]_3^0[_3 b]_3^0 \ldots [_3 b]_3^0]_2^0]_1^0$,
ahol a 3-as címkéjű membránokból 2^k db van ($k \in \mathbb{N}$). Ekkor három eset lehetséges: $n > 2^k$, $n = 2^k$, $n < 2^k$.

- I A következő 6 lépésben az alábbiak történnek: 1.) 1-es szabály átírja minden 3-as címkéjű membránban a b-t c-re, 2-es szabállyal minden 3-as címkéjű membránba bemegy 1-1 a ami $+$-ra állítja a membrán polaritását, valamint a 9-es szabállyal a b kimegy a 2-es címkéjű membránból és $-$-ra állítja a polaritást. 2.) A 4-es szabály visszaírja a c-ket b-kre, az 5-ös szabály visszaküldi az a-kat a 2-es membránba, a 11-es szabály kiküld egy a-t a 2-es membránból és negatívra állítja a polaritást, a 19-es szabály pedig átírja a b-t az 1-es membránban g-re. 3.) A 8-as szabály kettéosztja a 3-as membránokat és mindegyikben lecseréli b-t d-re, a 16-os szabály visszahozza az a-t a 2-es membránba, a 20-as pedig átírja g-t h-ra az 1-es membránban. 4.) A 6-os szabály kiküldi a d-ket a 3-as membránokból, a 21-es pedig átírja h-t i-re az 1-es membránban. 5.) A 15-ös szabály átírja a d-ket e-kre a 2-es membránban, a 22-es pedig átírja i-t j-re az 1-es membránban. 6.) A 7-es szabállyal minden 3-as címkéjű membránba bemegy egy e objektum b-vé alakulva és 0-ra állítva a polaritást, a 17-es szabállyal pedig az j belép a 2-es membránba, b-vé alakul és 0-ra állítja a polaritást. Ezzel megdupláztuk a 3-as címkéjű membránok számát, mindegyikben van egy b, a 2-es membránban egy b van és az n db a, más objektum nincs a rendszerben és minden membrán polaritása 0 (tehát k értékét megnöveltük eggyel).
- II Annyi az eltérés a fenti esettől, hogy nincs plusz a ami negatívra állítsa a 2-es membrán polaritását, így az ötödik lépésben a 15-ös szabály nem hajtódik végre, valamint a hatodik lépésben a 17-es szabály helyett a 12-es fog végrehajtódni, így a yes' objektum jelenik meg a 2-es membránban, majd a 14-es szabály feloldja a membránt, ezt követően a 24-es szabály kiküldi a helyes választ jelképező *yes* objektumot a környezetbe (az utolsó lépésben).
- III Az első lépés után lesz olyan 3-as címkéjű membrán, aminek nem jutott a így a polaritása megmarad 0-nak. A 3-as, 10-es, 13-as,

18-as és 23-as szabályok garantálják hogy a rendszer működése leálljon, a helyes *no* válasz objektum létre jöjjön és az utolsó lépésben kerüljön ki a környezetbe.