8. Sejtautomaták egy fajra

A tér-idő modellezés egyik hatékony eszköze a sejtautomaták. A sejtautomaták időben és térben egyaránt diszkrét folyamatokat írnak le, így viselkedésük számítógép segítségével könnyen szimulálható. A sejtautomaták fogalmát Neumann János [15] és Stanislaw M. Ulam definiálta először a XX. század közepén. A témáról részletesebben olvashatunk a [3], [16], [17] tanulmányokban.

A sejtek egy diszkrét valódi térben helyezkednek el (például a síkon egy négyzetrács négyzetei a sejtek), amelyeknek vannak szomszédai. A sejtek állapotát adott pillanatban az egyes fajok jelenléte definiálja. Lehetséges, hogy több faj egyszerre jelen van egy adott sejtben, de ki is zárhatják egymást. Amennyiben csupán az elfoglaltság tényét rögzítjük, foglaltsági modellről beszélünk. Minden egyes sejt állapota a diszkrét idő-ugrás alatt megváltozik a sejt és szomszédainak állapotától függően, amit az átmenetfüggvény ír le.

8.1. Alapfogalmak

Dolgozatunkban egyszerű elfoglaltsági modellekkel foglalkozunk. Az alábbiakban megadjuk az általunk használt sejtautomaták általános definícióját.

8.1.1. Definíció

A C_a sejtautomát (cellular automaton) a következő négyes definiálja: C_a=(S, N, P, f), ahol:

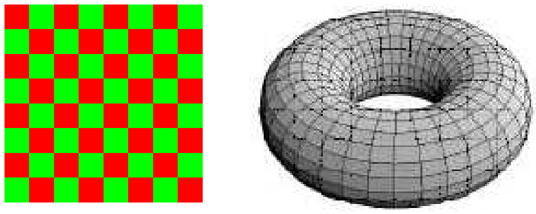
- S a sejttér
- N a szomszédsági reláció
- P az állapotfüggvény
- f a lokális átmenetfüggvény.

8.1.2. Definíció

Az S sejttér egy n-dimenziós euklideszi tér: $S = \mathbb{Z}^n$, ahol $n \in \mathbb{N}$.

8.1.1. Megjegyzés

A sejttér lényegében két féle lehet: véges illetve tórusz. Véges sejttér esetén van határ, ahol a sejtek másként viselkednek, míg tórusz sejttér esetén minden sejtnek azonos szomszédsága van. Egy másik szempont alapján a sejttér két leggyakrabban alkalmazott fajtája: négyzetháló illetve hatszögháló.



8.1.1. ábra

Véges illetve tórusz sejttér

Dolgozatunkban síkbeli négyzetrácsos sejtterekkel foglalkozunk, tóruszos topológiával.

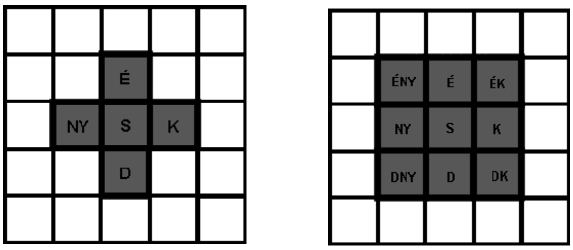
8.1.3. DEFINÍCIÓ

Az N szomszédsági reláció minden egyes sejtre megadja szomszédainak egy véges halmazát.

8.1.2. MEGJEGYZÉS

Két dimenziós, négyzethálós sejttér esetén a két leggyakrabban alkalmazott szomszédsági reláció:

- Neumann-féle szomszédság: minden sejtnek négy szomszédja van észak, kelet, dél, nyugat irányban
- Moore-féle szomszédság: minden sejtnek nyolc szomszédja van észak, kelet, dél, nyugat valamint északkelet, északnyugat, délkelet, délnyugat irányban.



8.1.2. ábra

Neumann- illetve Moore-féle szomszédság

Dolgozatunkban a Moore-féle szomszédségi relációt tekintjük. Megjegyezzük, hogy a Neumann- illetve Moore-féle szomszédság esetén az automaták gyakran különbözőképpen viselkednek.

8.1.4. DEFINÍCIÓ

A P állapotfüggvény minden sejt tetszőleges időpillanatbeli állapotát határozza meg.

8.1.5. DEFINÍCIÓ

Legyen X az állapotok halmaza, k a szomszédok száma, $s^{(t)}$ egy tetszőleges sejt állapota a t pillanatban, szomszédainak állapota legyenek: $s_1, s_2, \ldots, s_k$. Ekkor az f lokális átmenetfüggvény egy olyan $k+2$ változós függvény, melyre igaz:

$$f(t, s^{(t)}, s_1{}^{(t)}, s_2{}^{(t)}, \ldots, s_k{}^{(t)}) : \mathbb{N} \times X^{k+1} \to X, \quad s^{(t+1)} = f(t, s^{(t)}, s_1{}^{(t)}, s_2{}^{(t)}, \ldots, s_k{}^{(t)})$$

ahol $X^{k+1} = X \times X \times \ldots \times X$, Descartes-szorzat..

8.1.3. MEGJEGYZÉS

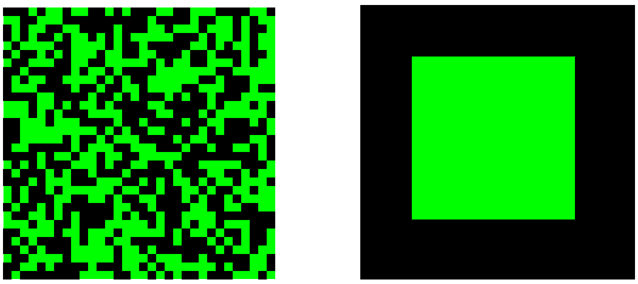
Foglaltsági modellek esetén az állapothalmaz {0, 1}, melyek segítségével egyértelműen meghatározható, hogy egy sejt üres vagy foglalt.

Ökológiailag fontos feltétel, hogy az átmenetfüggvény rendelkezzen azzal a tulajdonsággal, hogy ha egy sejt és annak mindegyik szomszédja a t időpillanatban üres, akkor a $t + 1$ időpillanatban is üres kell hogy legyen.

8.2. Az általunk vizsgált sejtautomata jellemzői

Sztochasztikus sejtautomata modelleket vizsgáltunk, azaz minden sejt a következő állapotát bizonyos valószínűséggel veszi fel.

A dolgozatban a sejttér véges, két dimenziós és négyzethálós, melyet véges $n \times n$-es mátrixszal definiálunk. Az ökológiában is gyakrabban előforduló Moore-féle szomszédságot vesszük, így minden sejtnek nyolc szomszédja van. Meg kell adni egy kezdeti konfigurációt, a $t = 0$ időpillanatbeli állapotot: vizsgálunk véletlen (random) és blokkos keztetű véges $n \times n$-es mátrixot. Minden sejt állapota a {0, 1} halmaz elemeiből kerül ki: 0-val jelölve az üres, 1-gyel a foglalt sejtet. Az átmenetfüggvény valószínűségi függvény, melyet a rendszer minden egyes sejtjére alkalmazni kell. Az átmenetfüggvény kétváltozós. Az első változó egy adott sejt t időpillanatbeli állapota, a második az adott sejt t időpillanatbeli elfoglalt szomszédos sejtek száma. A modell egyszerű kierjesztésénél a szomszédok elhelyezkedését is figyelembe lehetne venni (pl. domborzat, szélirány stb.). Itt most ezzel nem foglalkozunk. Egy adott sejt minden egyes szomszédját egyenlő súllyal vesszük figyelembe.



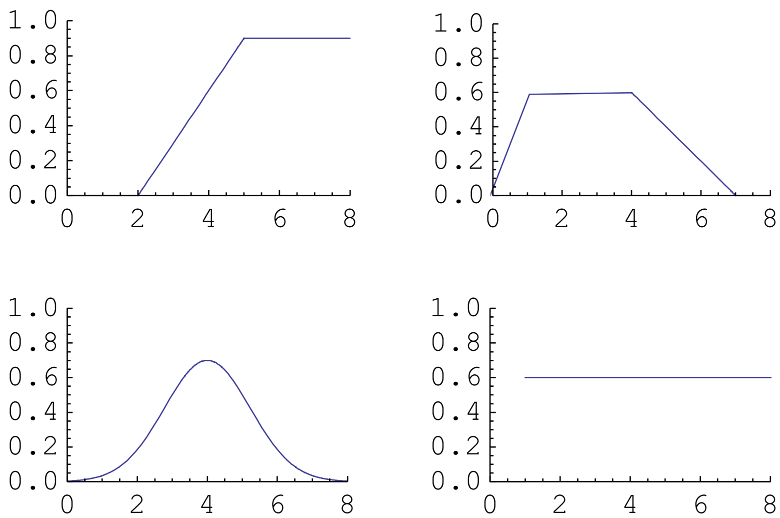
8.2.1. ábra

Random illetve blokkos kezdetű mátrix

8.3. Előzmények, kezdeti eredmények

A Levins-féle modell sejtautomata megfelelőjét vizsgáljuk. Itt e a lokális kihalási valószínűség, ekkora valószínűséggel pusztul ki minden egyes elfoglalt sejtből a faj: $e = P(1 \to 0)$; k a kolonizációs valószínűség, a faj ekkora valószínűséggel kolonizál üres sejteket: $k = P(0 \to 1)$. A Levins-féle klasszikus modell pontos analógiája szerint mindkét valószínűség konstans lenne. Ugyanakkor előfordulhat, hogy mindét valószínűség függ az adott sejt 1 állapotú szomszédainak számától. Így nem kolonizációs és kihalási rátáról beszélünk, hanem megfelelőbb kifejezés erre a kolonizációs és kihalási függvény.

A kolonizációs valószínűségfüggvényt Karsai János és Pestiné Rácz Éva Veronika vezette be. Olyan sztochasztikus sejtautomatákat vizsgáltak, ahol a kolonizációs valószínűség függ az elfoglalt szomszédos sejtek számától [18], [19], [20]. Már egy faj esetén is érdekes jelenségeket eredményezett a kolonizáció szomszédságfüggése (pl. foltosodás), de komoly következményekkel több faj versengése esetén járt, mint például a versengés kimenetelének kezdeti mintázattól való függése.



8.3.1. ábra

Példák kolonizációs függvényre

A fenti ábrák néhány jellemző kolonizációs függvényt mutatnak. Az első esetben 2 szomszéd még nem elegendő az üres folt elfoglalásához (küszöbérték), utána a valószínűség a szomszédok számával lineárisan nő, majd eléri a maximális 0,9 értéket. A második példában elfoglalt szomszéd nélkül nyilvánvalóan nincs kolonizáció, 1-4 szomszéd esetén azonos a betelepülés valószínűsége, és utána a túlzsúfoltság miatt romlik a kolonizáció esélye.

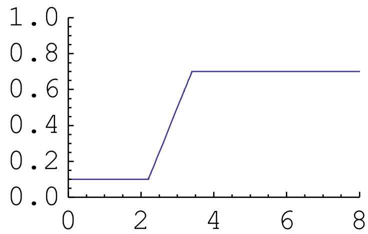
A szomszédságfüggő kolonizáció analógiájára szomszédságfüggő kihalást kezdtünk vizsgálni, mivel a kolonizációhoz hasonlóan a kihalás valószínűsége is függhet a szomszédoktól. Nyilván a szomszédok nemcsak a terjedést, hanem a túlélést is segíthetik vagy gátolhatják. Az eredményeink kezdetiek, ennek folytatása, kidolgozása, tanulmányozása és részletes elemzése reményeink szerint a jövőben meg fog valósulni.

Értelmezési okok miatt kihalási függvény helyett az életbenmaradási valószínűség-függvényt használjuk. Matematikailag ezeket hasonlóan kell kezeln
(kihalási függvény = 1− életbenmaradási függvény)

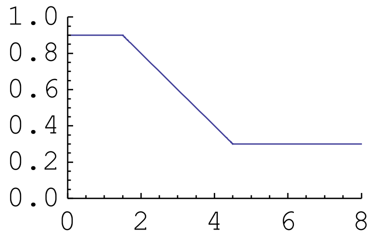
8.4. Az életbenmaradási függvény alakjai

A szomszédfüggő életbenmaradási függvény következő alakjait érdemes vizsgálni (minden egyéb ezen függvények kombinációjaként felírható):

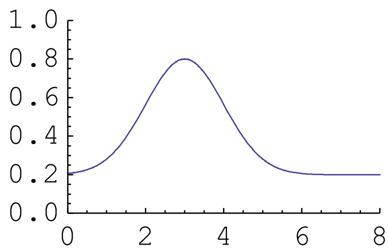
– Kevés szomszéd mellett a sejt valamilyen valószínűséggel életben marad (természetes belső tulajdonság), majd a szomszédok számának növekedésével lineárisan nő az életbenmaradási esélye, majd megáll a növekedés, hiszen a kihalás mindenképpen bekövetkezik.



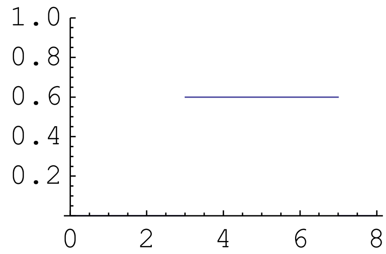
– Kevés szomszéd mellett nagyobb az életbenmaradási függvény értéke, majd monoton csökken egy bizonyos szomszédszámig, ahol elér egy minimumot. Azaz minél kevesebb a szomszéd, annál nagyobb a túlélés valószínűsége.



– Kevés szomszéd mellett kisebb az életbenmaradás valószínűsége, majd monoton nő egy bizonyos szomszédszámig, ahol monoton csökkenni kezd az életbenmaradási függvény egy szomszédszámig, ahol az akár konstanssá válhat. Azaz kevés illetve sok szomszéd esetén kisebb a túlélés valószínűsége, míg egy ideális szomszédszám esetén a legvalószínűbb, hogy életben marad. A túlzsúfoltság csökkenti a túlélés esélyét.



– Kevés szomszéd mellett nulla az életbenmaradási függvény értéke, majd egy szomszédszámtól kezdve egy másik szomszédszámig konstans az életbenmaradási függvény, majd újra nullává válik. Azaz kevés illetve sok szomszéd esetén nulla a túlélés valószínűsége, míg egy ideális szomszédszám-intervallum esetén az életben maradási függvény értéke pozitív konstans.



– Általános lépcsős függvények csupán matematikailag lehet érdekesek, ezért ezeket, mint életbenmaradási függvényt nem vizsgáljuk.

8.5. Néhány ökológiailag realisztikus példa

Karsai János és Pestiné Rácz Éva Veronika által írt programot felhasználva és módosítva néhány példát mutatunk be. Véletlen és blokkos kezdeti konfigurációkat tekintünk, különböző kolonizációs és életbenmaradási stratégiák mellett. Természetesen, az egyes függvényekben a paraméterek (küszöbök, maximális érték stb.) konkrét értéke döntő szerepet játszik a rendszer kimenetelét illetően. Az ezekre vonatkozó kísérletek további hosszú kutatásokat igényelnek. Az alábbiakban igyekeztünk a függvényeket realisztikus paraméterekkel megválasztani.

A sejttér mindegyik esetben 32 x32-es mátrix.

```
nn = 32;
```

■ *Mathematica programbetöltés. Program-definíciók*

■ *Változási valószínűség mátrixok*

Mtx1[k_] = {P(1 → 1), P(1 → 0)} - életbenmaradási mátrix
Mtx0[k_] = {P(0 → 1), P(0 → 0)} - elfoglalási mátrix
ahol k az adott cella szomszédainak aktuális súlyozott száma

```
Mtx1 = {EE, 1 - EE};
Mtx0 = {CC, (1 - CC)};

UpdRule :=
 {EE :> EE0[##],
  CC :> Col[##]
 }

UpdRule
 {EE :→ EE0[##1], CC :→ Col[##1]}

Mtx := Evaluate[({Mtx0 /. UpdRule, Mtx1 /. UpdRule})] &
```

▢ Sejtautomata

```
OneSpecies[{ProbMat_, neighbors_, weights_}, initconfig_, t_] :=
 Module[{livingNghbrs, Func},
  Attributes[Func] = Listable;
  Func[1, x_] :=
   Random[] /. a_ → Which[a ≤ ProbMat[Floor[x]]〚2, 1〛, 1, True, 0];
  Func[0, x_] := Random[] /.
    a_ → Which[a ≤ ProbMat[Floor[x]]〚1, 1〛, 1, True, 0];
  livingNghbrs[mat_] :=
   weights.Map[
     RotateRight[mat, #1] &, neighbors];
  NestList[Func[#1, livingNghbrs[#1]] &, initconfig, t]]
```

▢ Moore-féle szomszédság

```
neighbors = {{-1, -1}, {-1, 0},
   {-1, 1}, {0, -1}, {0, 1}, {1, -1}, {1, 0}, {1, 1}};
weights = {1, 1, 1, 1, 1, 1, 1, 1};

MatrixForm[neighbors];
```

▢ Kezdeti konfigurációk

```
Randinit[n_] := Table[Random[Integer, {0, 1}], {n}, {n}]

Saturated[n_, szazalek_Integer] :=
 (Table[Random[Integer, {0, 99}], {n}, {n}] /. Join[# → 1 & /@
     Range[0, szazalek - 1], # → 0 & /@ Range[szazalek, 100]])

Blockinit[n_, sortol_, sorig_, otol_, oig_] :=
 ReplacePart[Table[0, {n}, {n}], 1,
  Flatten[Table[{o, p}, {o, sortol, sorig}, {p, otol, oig}], 1]]
```

▢ Ábrázolások

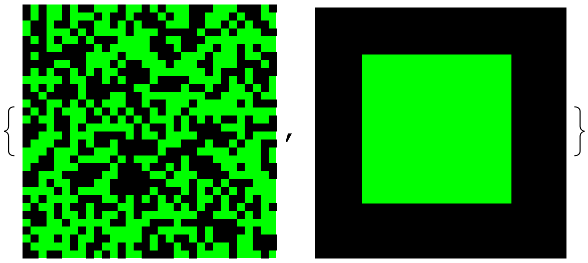
```
ShowLife[lista_] :=
 Map[(Show[Graphics[RasterArray[Reverse[lista[[#]] /.
         {1 → RGBColor[1, 0, 0], 0 → RGBColor[0, 0, 0]}]],
      AspectRatio → Automatic]]) &, Range[Length[lista]]]

ShowLife[lista_] := Map[(Show[Graphics[
      Raster[Reverse[lista[[#]] /. {1 → {0, 1, 0}, 0 → {0, 0, 0}}]],
      AspectRatio → Automatic]]) &, Range[Length[lista]]]
```

A kezdeti állapotok

```
InitRandom := Randinit[nn];
InitBlock = Blockinit[nn, 7, 25, 7, 25];
```

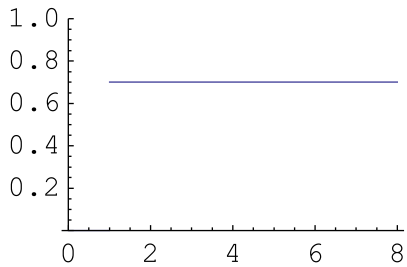
```
ShowLife[{InitRandom, InitBlock}]
```



Kolonizációs függvények

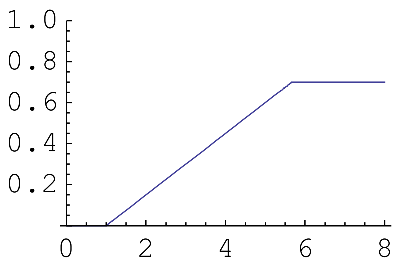
Konstans kolonizáció

```
KonstKolon[k_] := 0.7 UnitStep[(k - 1)]
Plot[KonstKolon[k], {k, 0, 8}, PlotRange → {0, 1}]
```



Lineáris kolonizáció küszöbértékkel

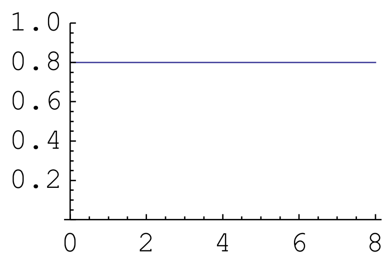
```
LinearKolon[k_] := Min[Max[0, 0.15 (k - 1)], 0.7]
Plot[LinearKolon[k], {k, 0, 8}, PlotRange → {0, 1}]
```



Életbenmaradási függvények

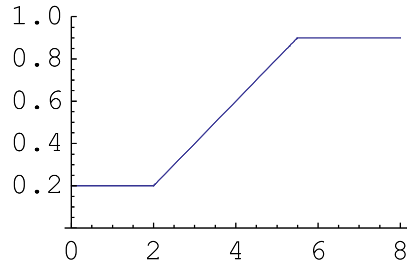
Konstans életbenmaradási valószínűség

```
KonstLife[k_] := 0.8 UnitStep[(k)]
Plot[KonstLife[k], {k, 0, 8}, PlotRange → {0, 1}]
```



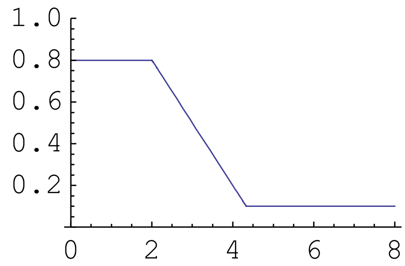
◻ Növekvő életbenmaradási valószínűség

```
IncrLife[k_] := 0.2 + Min[Max[0, 0.2 (k - 2)], 0.7]
Plot[IncrLife[k], {k, 0, 8}, PlotRange → {0, 1}]
```



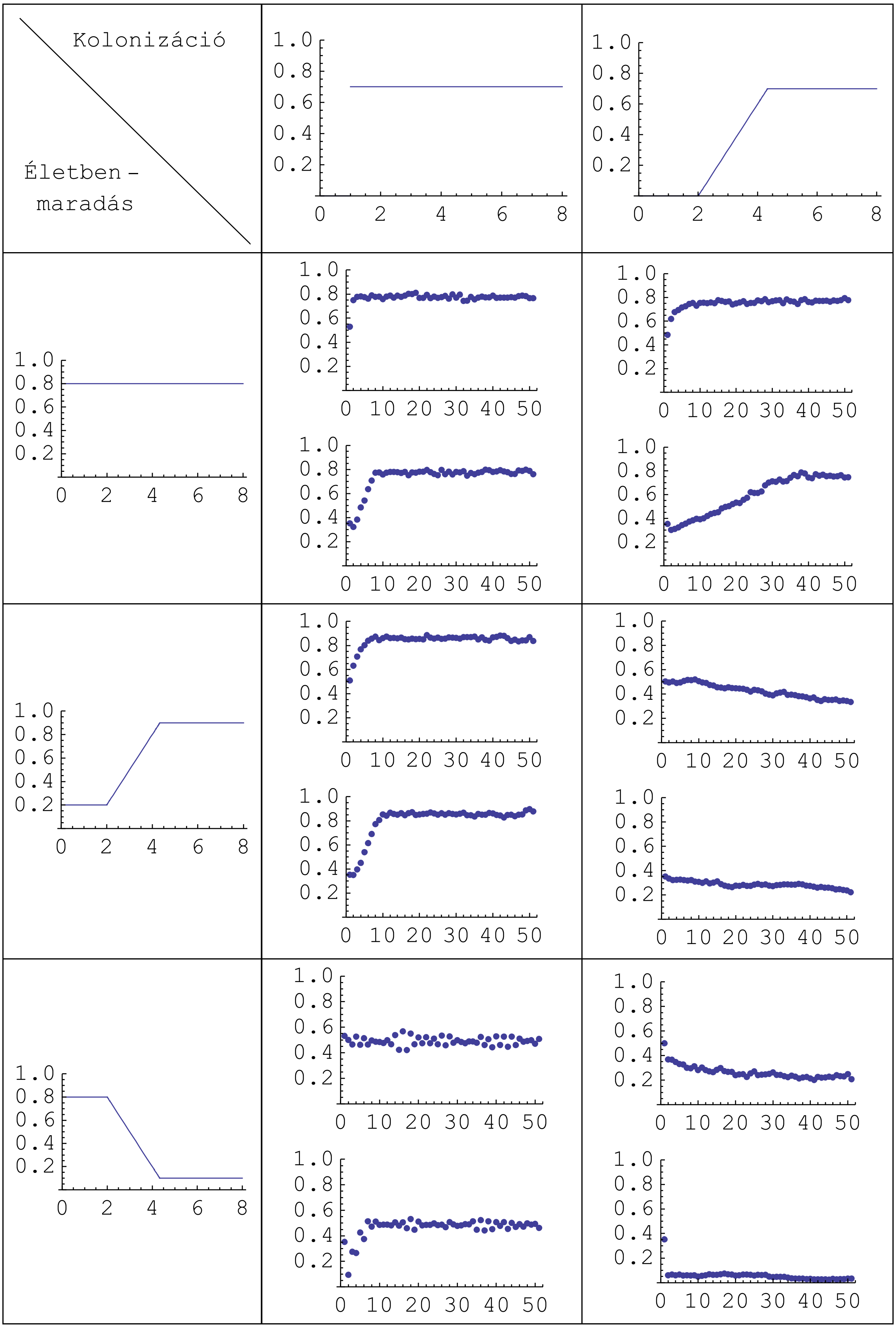
◻ Csökkenő életbenmaradási valószínűség

```
DecrLife[k_] := 0.8 - Min[Max[0, 0.3 (k - 2)], 0.7]
Plot[DecrLife[k], {k, 0, 8}, PlotRange → {0, 1}]
```



8.5.1. Összegzés

Az alábbi táblázatban összefoglaltuk az egyes példákra kapott eredményeket: az átlagtér fejlődést mutatjuk az idő függvényében a véletlen (felső grafikon) és blokkos (alsó grafikon) kezdeti formációból kiindulva, a megadott kolonizációs (első sor) és életben maradási (első oszlop) függvények esetén.


Kolonizáció
Életben –
maradás

A táblázatból sejthető, hogy az egymással összhangban levő hatások (pl. növő kolonizáció és növő életbenmaradás) esetén hasonló a rendszer kimenetele mindkét kezdeti konfigurációból kiindulva. Ugyanakkor az ellentétes hatások (ami ugyan kevéssé valószerű) esetén a kimenet különbözhet. Ezeket a sejtéseket további szimulációsorozatokkal kell alátámasztanunk. Hasonlóan, újabb kutatásokat igényel az egyensúlyi állapot is. Példáink sejtetik, hogy ahol az egyensúlyi állapot kezdeti konfigurációtól függetlenül kialakul, akkor az ugyanaz minden esetben, ami a Levins-féle átlagtér modellek használatának jogosultságát igazolja.

Az explicit térbeli fejlődésre vonatkozó speciális tulajdonságokat az egyes példáknál elemezzük.

8.5.2. Példa: KonstLife, KonstKolon

```
Clear[EE0, Col]
EE0[k_] := KonstLife[k]
Col[k_] := KonstKolon[k]
```

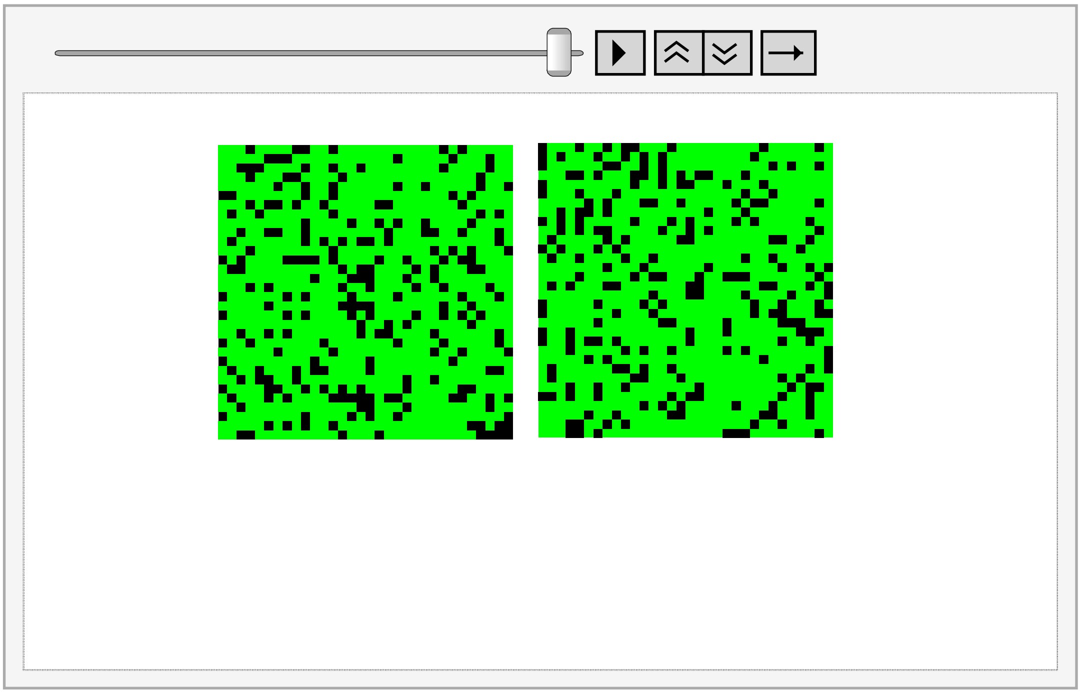
Az átmeneti valószínűség mátrixok: $\{\{P(0 \to 1), P(0 \to 0)\}, \{P(1 \to 1), P(1 \to 0)\}, \ldots\}$

```
Table[Mtx[i], {i, 0, 8}]
{{{0, 1}, {0.8, 0.2}}, {{0.7, 0.3}, {0.8, 0.2}}, {{0.7, 0.3}, {0.8, 0.2}},
 {{0.7, 0.3}, {0.8, 0.2}}, {{0.7, 0.3}, {0.8, 0.2}}, {{0.7, 0.3}, {0.8, 0.2}},
 {{0.7, 0.3}, {0.8, 0.2}}, {{0.7, 0.3}, {0.8, 0.2}}, {{0.7, 0.3}, {0.8, 0.2}}}
```

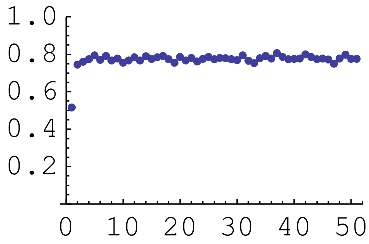
Futtatás

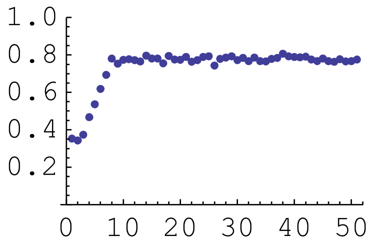
```
SolRandom = OneSpecies[{Mtx, neighbors, weights}, InitRandom, 50];
SolBlock = OneSpecies[{Mtx, neighbors, weights}, InitBlock, 50];
```

Ábrázolás



Vegyük észre hogy a környezetfüggetlen hatások keveredést eredményeznek, a két fejlődés között hosszú távon nincs különbség.





8.5.3. Példa: KonstLife, LinearKolon

```
Clear[EE0, Col]
EE0[k_] := KonstLife[k]
Col[k_] := LinearKolon[k]
```

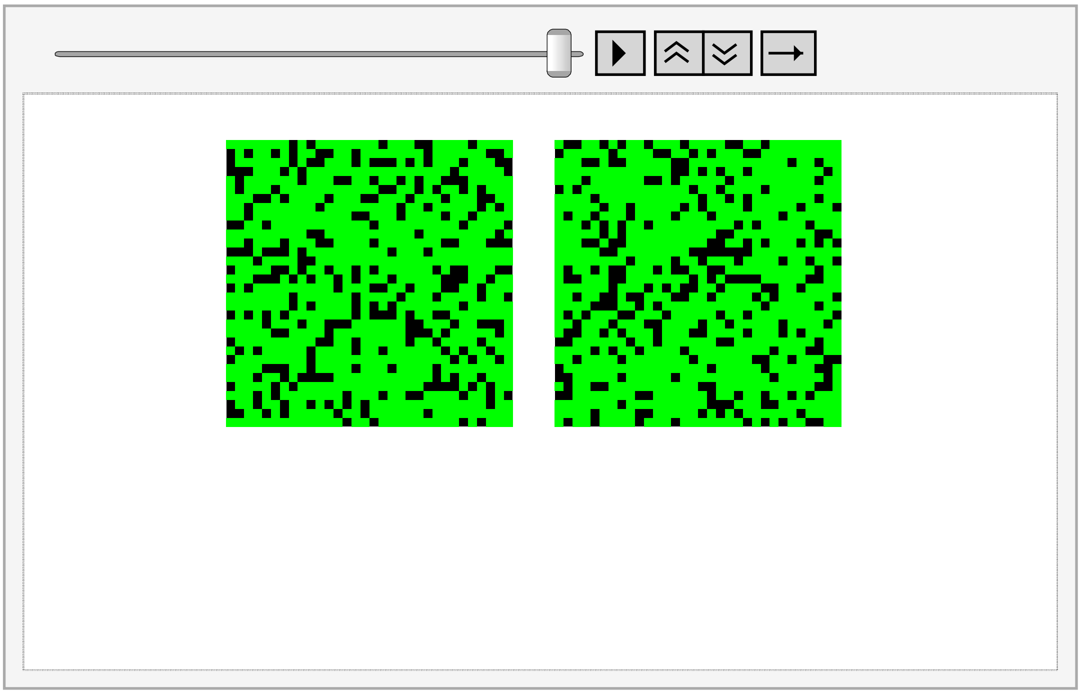
Az átmeneti valószínűség mátrixok: $\{\{P(0 \to 1),\ P(0 \to 0)\},\{P(1 \to 1),\ P(1 \to 0)\}, ...\}$

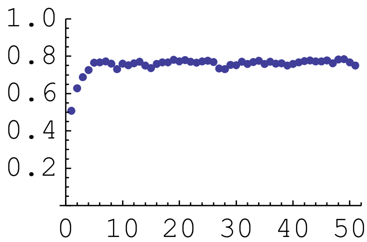
```
Table[Mtx[i], {i, 0, 8}]
{{{0, 1}, {0.8, 0.2}}, {{0, 1}, {0.8, 0.2}}, {{0.15, 0.85}, {0.8, 0.2}},
 {{0.3, 0.7}, {0.8, 0.2}}, {{0.45, 0.55}, {0.8, 0.2}}, {{0.6, 0.4}, {0.8, 0.2}},
 {{0.7, 0.3}, {0.8, 0.2}}, {{0.7, 0.3}, {0.8, 0.2}}, {{0.7, 0.3}, {0.8, 0.2}}}
```

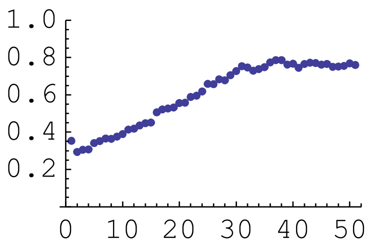
Futtatás

```
SolRandom = OneSpecies[{Mtx, neighbors, weights}, InitRandom, 50];
SolBlock = OneSpecies[{Mtx, neighbors, weights}, InitBlock, 50];
```

Ábrázolás







8.5.4. Példa: IncrLife, KonstKolon

```
Clear[EE0, Col]
EE0[k_] := IncrLife[k]
Col[k_] := KonstKolon[k]
```

Az átmeneti valószínűség mátrixok: $\{\{P(0 \to 1),\ P(0 \to 0)\},\{P(1 \to 1),\ P(1 \to 0)\},\ \ldots\}$

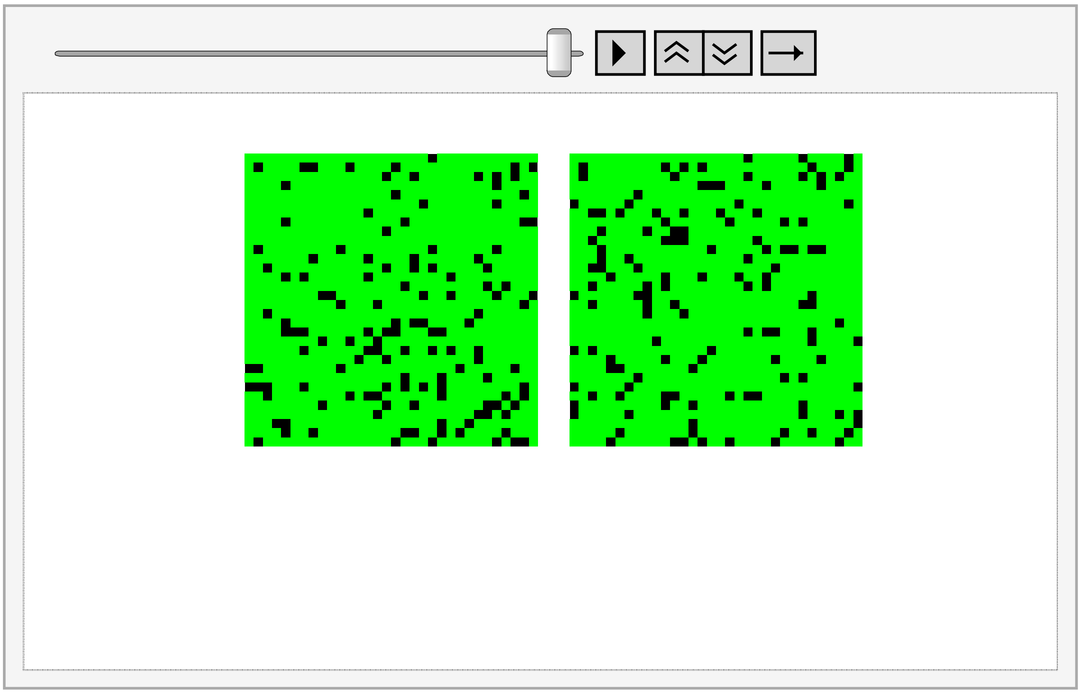
```
Table[Mtx[i], {i, 0, 8}]
```

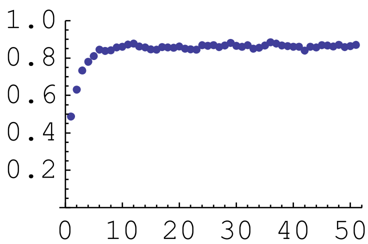
```
{{{0, 1}, {0.2, 0.8}}, {{0.7, 0.3}, {0.2, 0.8}}, {{0.7, 0.3}, {0.2, 0.8}},
 {{0.7, 0.3}, {0.4, 0.6}}, {{0.7, 0.3}, {0.6, 0.4}}, {{0.7, 0.3}, {0.8, 0.2}},
 {{0.7, 0.3}, {0.9, 0.1}}, {{0.7, 0.3}, {0.9, 0.1}}, {{0.7, 0.3}, {0.9, 0.1}}}
```

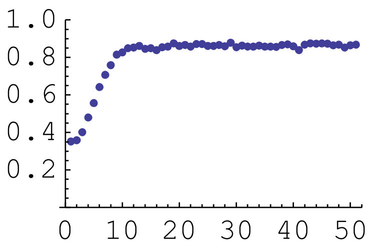
Futtatás

```
SolRandom = OneSpecies[{Mtx, neighbors, weights}, InitRandom, 50];
SolBlock = OneSpecies[{Mtx, neighbors, weights}, InitBlock, 50];
```

Ábrázolás







8.5.5. Példa: IncrLife, LinearKolon

```
Clear[EE0, Col]
EE0[k_] := IncrLife[k]
Col[k_] := LinearKolon[k]
```

Az átmeneti valószínűség mátrixok: $\{\{P(0 \to 1),\ P(0 \to 0)\},\{P(1 \to 1),\ P(1 \to 0)\},\ \ldots\}$

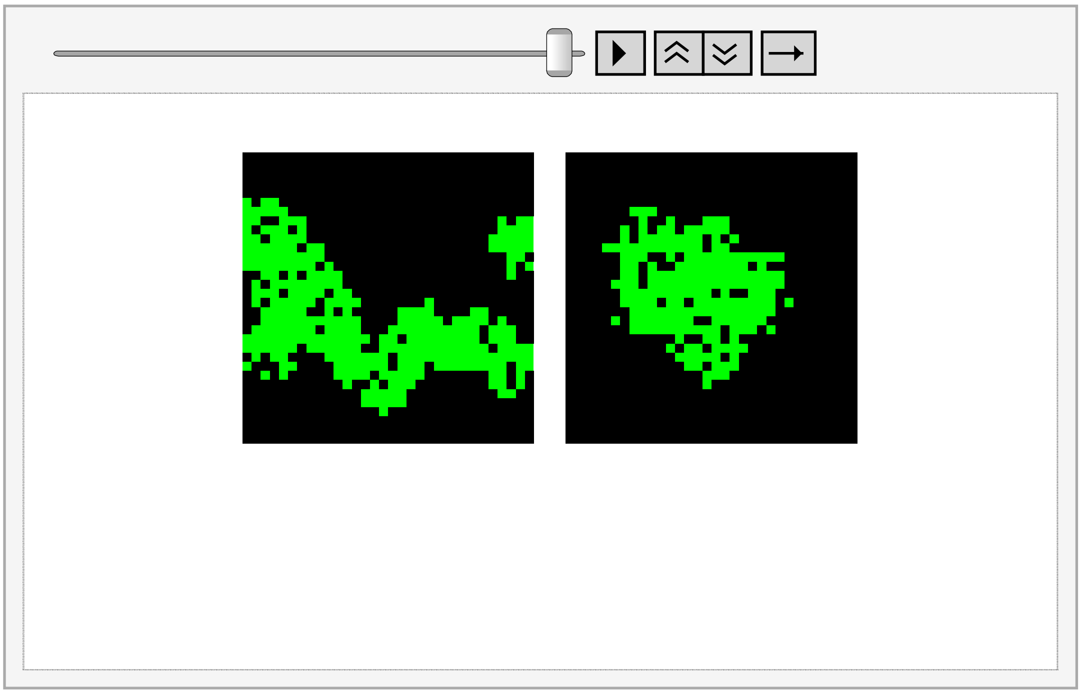
```
Table[Mtx[i], {i, 0, 8}]
```

```
{{{0, 1}, {0.2, 0.8}}, {{0, 1}, {0.2, 0.8}}, {{0.15, 0.85}, {0.2, 0.8}},
 {{0.3, 0.7}, {0.4, 0.6}}, {{0.45, 0.55}, {0.6, 0.4}}, {{0.6, 0.4}, {0.8, 0.2}},
 {{0.7, 0.3}, {0.9, 0.1}}, {{0.7, 0.3}, {0.9, 0.1}}, {{0.7, 0.3}, {0.9, 0.1}}}
```

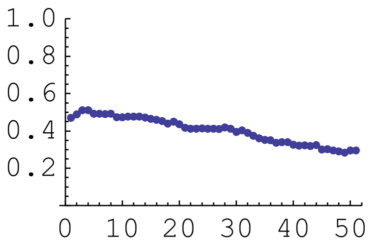
Futtatás

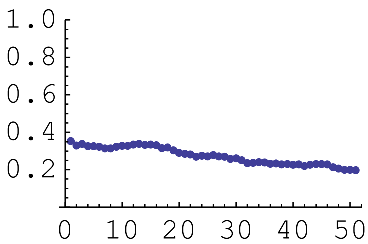
```
SolRandom = OneSpecies[{Mtx, neighbors, weights}, InitRandom, 50];
SolBlock = OneSpecies[{Mtx, neighbors, weights}, InitBlock, 50];
```

Ábrázolás



Jól látható ebben a modellben a foltosodás jelensége.





8.5.6. Példa: DecrLife, KonstKolon

```
Clear[EE0, Col]
EE0[k_] := DecrLife[k]
Col[k_] := KonstKolon[k]
```

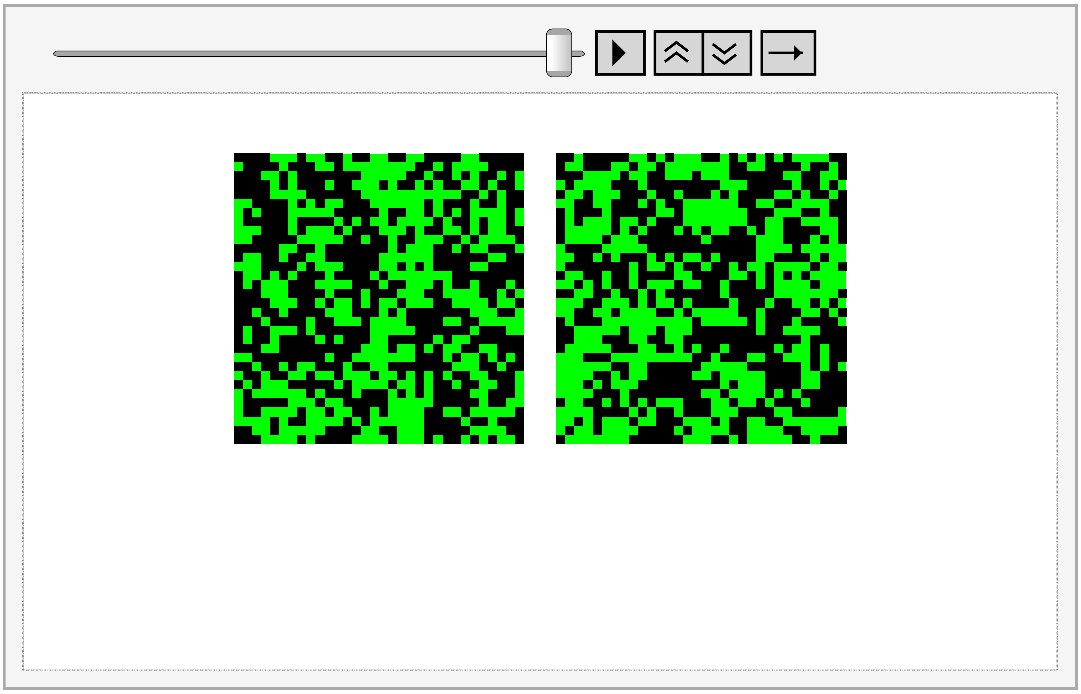
Az átmeneti valószínűség mátrixok: $\{\{P(0 \to 1), P(0 \to 0)\}, \{P(1 \to 1), P(1 \to 0)\}, \ldots\}$

```
Table[Mtx[i], {i, 0, 8}]
{{{0, 1}, {0.8, 0.2}}, {{0.7, 0.3}, {0.8, 0.2}}, {{0.7, 0.3}, {0.8, 0.2}},
 {{0.7, 0.3}, {0.5, 0.5}}, {{0.7, 0.3}, {0.2, 0.8}}, {{0.7, 0.3}, {0.1, 0.9}},
 {{0.7, 0.3}, {0.1, 0.9}}, {{0.7, 0.3}, {0.1, 0.9}}, {{0.7, 0.3}, {0.1, 0.9}}}
```

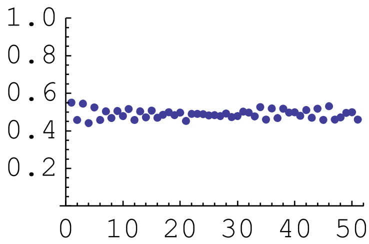
Futtatás

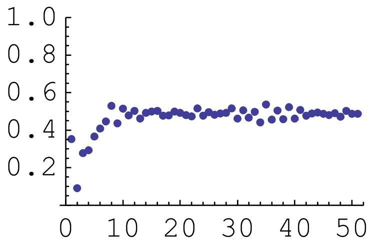
```
SolRandom = OneSpecies[{Mtx, neighbors, weights}, InitRandom, 50];
SolBlock = OneSpecies[{Mtx, neighbors, weights}, InitBlock, 50];
```

Ábrázolás



Ez egy fura modell. A sok szomszéd majdnem kihalást eredményez, de utána beáll az egyensúly.





8.5.7. Példa: DecrLife, LinearKolon

```
Clear[EE0, Col]
EE0[k_] := DecrLife[k]
Col[k_] := LinearKolon[k]
```

Az átmeneti valószínűség mátrixok: $\{\{P(0 \to 1),\ P(0 \to 0)\},\{P(1 \to 1),\ P(1 \to 0)\},\ ...\}$

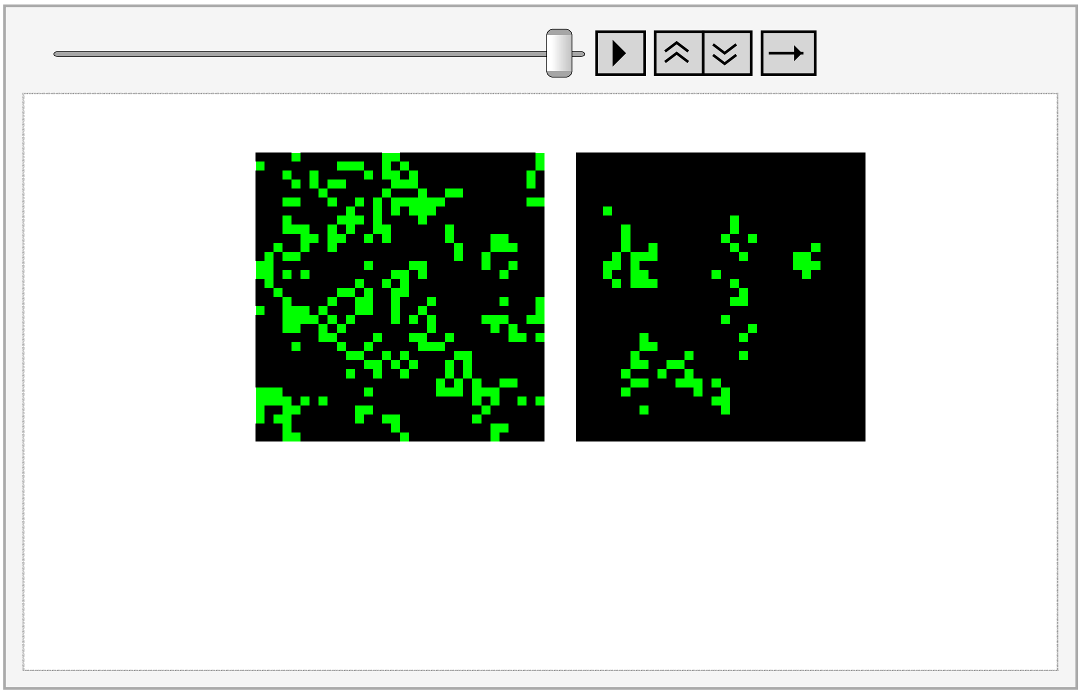
```
Table[Mtx[i], {i, 0, 8}]
```

```
{{{0, 1}, {0.8, 0.2}}, {{0, 1}, {0.8, 0.2}}, {{0.15, 0.85}, {0.8, 0.2}},
 {{0.3, 0.7}, {0.5, 0.5}}, {{0.45, 0.55}, {0.2, 0.8}}, {{0.6, 0.4}, {0.1, 0.9}},
 {{0.7, 0.3}, {0.1, 0.9}}, {{0.7, 0.3}, {0.1, 0.9}}, {{0.7, 0.3}, {0.1, 0.9}}}
```

Futtatás

```
SolRandom = OneSpecies[{Mtx, neighbors, weights}, InitRandom, 50];
SolBlock = OneSpecies[{Mtx, neighbors, weights}, InitBlock, 50];
```

Ábrázolás



Az ellentétes kolonizációs és túlélési stratégiák miatt az egyensúly egy alacsony szinten áll be.