

Automaty komórkowe w programie Mathematica



CELLULARAUTOMATON

ANNA ŚNIEG

Podstawowe komendy w programie Mathematica



- **CellularAutomaton[zasada, stan początkowy, t]** generuje listę przedstawiającą ewolucję automatu komórkowego za pomocą określonej reguły od stanu początkowego w t krokach.

CellularAutomaton - przykłady



- Zasada 30, stan początkowy $\{0,0,0,1,0,0,0\}$, liczba kroków – 5

```
In[14]:= CellularAutomaton[30, {0, 0, 0, 1, 0, 0, 0}, 5]
```

```
Out[14]= {{0, 0, 0, 1, 0, 0, 0}, {0, 0, 1, 1, 1, 0, 0}, {0, 1, 1, 0, 0, 1, 0}, {1, 1, 0, 1, 1, 1, 1}, {0, 0, 0, 1, 0, 0, 0}, {0, 0, 1, 1, 1, 0, 0}}
```

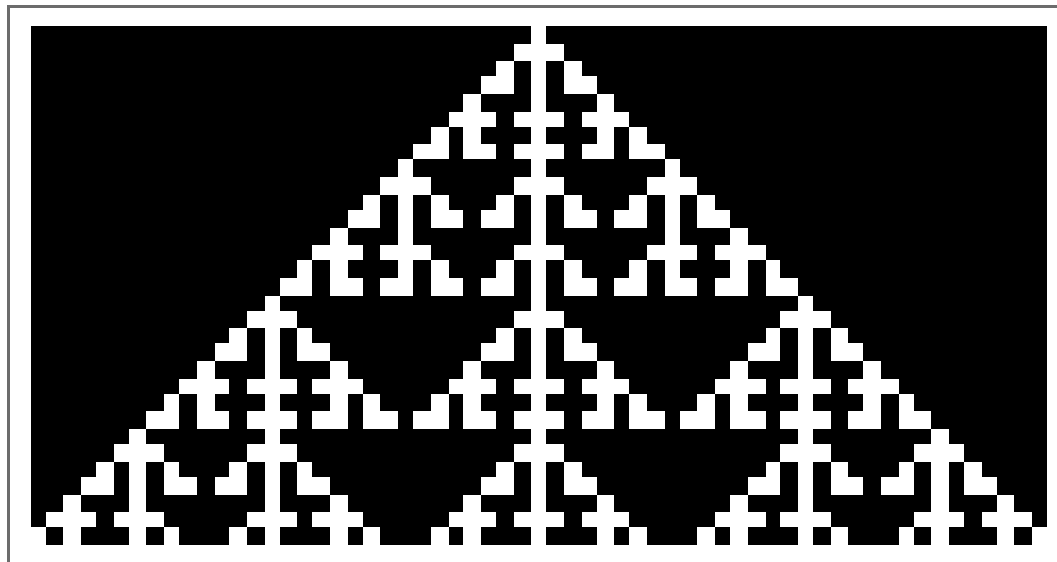
CellularAutomaton



- Zasada 150, stan początkowy- jedynka otoczona zerami tzn. 0001000, liczba kroków - 30

```
In[39]:= ArrayPlot[CellularAutomaton[150, {{0}, 1}, 30]]
```

Out[39]=



CellularAutomaton



- **CellularAutomaton[reguła, stan początkowy, {{x,t}}]** generuje listę przedstawiającą ewolucję automatu komórkowego za pomocą określonej reguły od stanu początkowego w t krokach, przy czym wyświetla tylko elementy od kroku x do kroku t.

CellularAutomaton - przykłady



```
In[58]:= CellularAutomaton[30, {0, 0, 0, 1, 0, 0, 0}, {{5}}]
```

```
Out[58]= { {0, 0, 1, 1, 1, 0, 0} }
```

```
In[59]:= CellularAutomaton[30, {0, 0, 0, 1, 0, 0, 0}, {{{5}}}]
```

```
Out[59]= {0, 0, 1, 1, 1, 0, 0}
```

```
In[80]:= CellularAutomaton[30, {0, 0, 0, 1, 0, 0, 0}, {{3, 5}}]
```

```
Out[80]= { {1, 1, 0, 1, 1, 1, 1}, {0, 0, 0, 1, 0, 0, 0}, {0, 0, 1, 1, 1, 0, 0} }
```

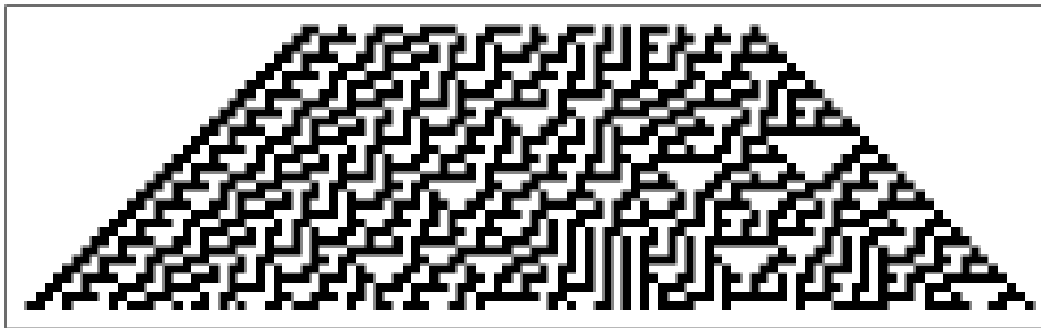
CellularAutomaton – przykłady



- Reguła 30, stan początkowy $\{1,0,0,0,0,0,0,0,0,1\}$, krok od 20 do 50.

```
In[33]:= ArrayPlot[CellularAutomaton[30, {SparseArray[{1 → 1, 10 → 1}], 0}, {{20, 50}} ]]
```

Out[33]=



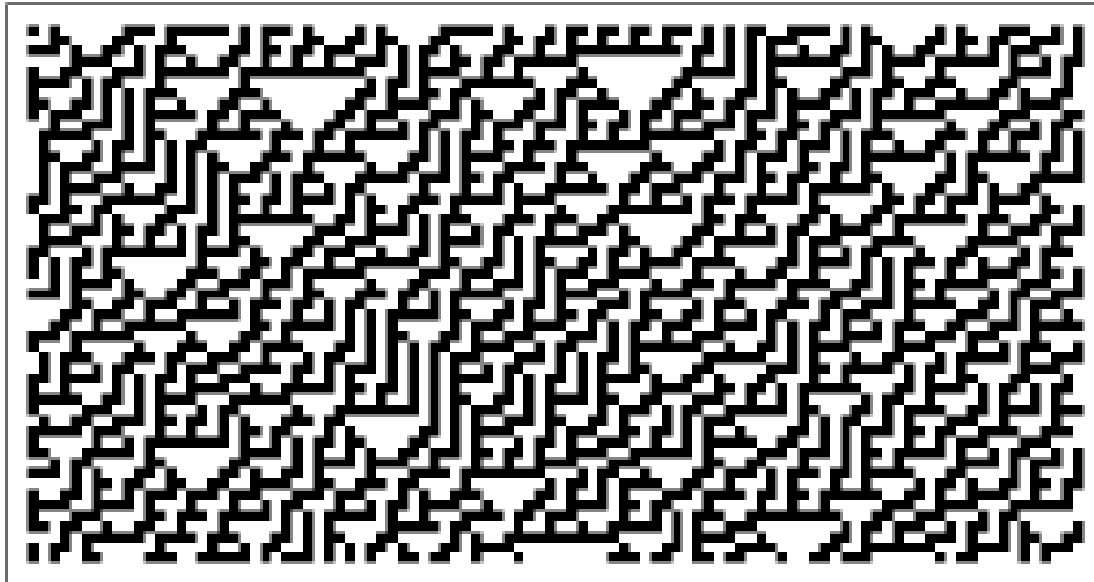
CellularAutomaton – sposoby zapisu stanu początkowego



- Za pomocą komendy **Randominteger[n,k]** możemy wyznaczyć losowe warunki początkowe

```
In[4]:= ArrayPlot[CellularAutomaton[30, RandomInteger[1, 100], 50]]
```

```
Out[4]=
```



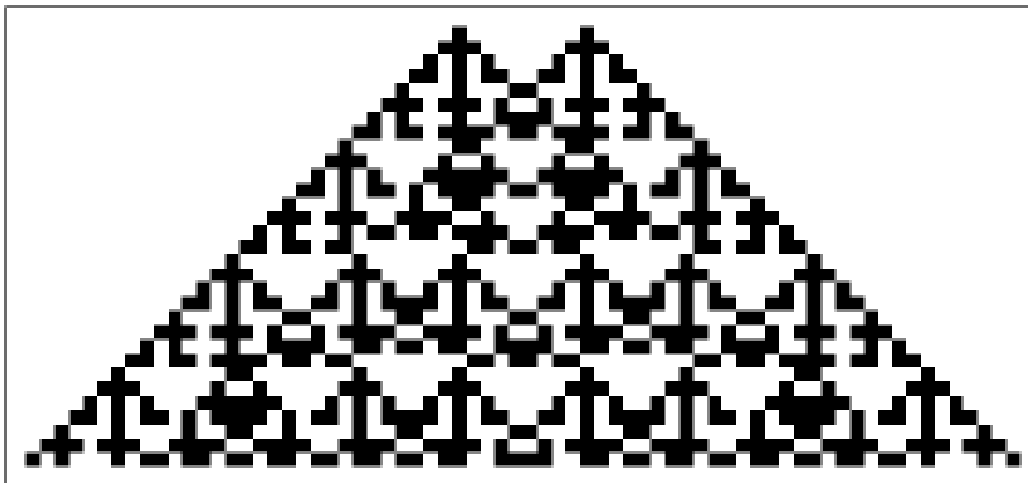
CellularAutomaton – sposoby zapisu stanu początkowego



- Za pomocą komendy **SparseArray** [{pozycja_1 -> wartość_1, pozycja_2 -> wartość_2,...}, n] możemy wyznaczyć stan początkowy, w którym ustalona wartość pojawi się w określonej pozycji, a długość wektora jest równa n.

```
In[14]:= ArrayPlot[CellularAutomaton[150, {SparseArray[{1 -> 1, 10 -> 1}], 0}, 30]]
```

Out[14]=

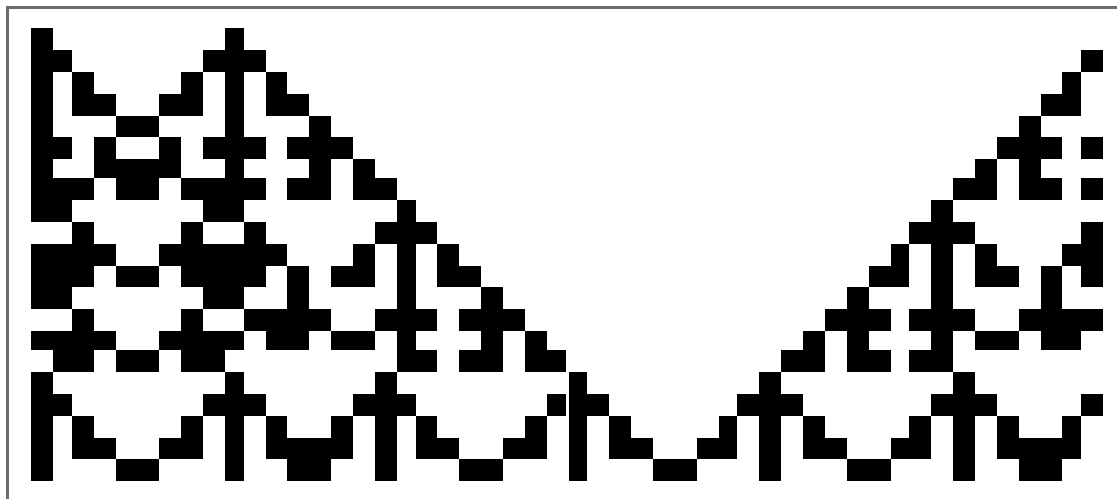


CellularAutomaton – sposób zapisu stanu początkowego



```
In[26]:= ArrayPlot[CellularAutomaton[150, SparseArray[{1 → 1, 10 → 1}, 50], 20]]
```

Out[26]=



CellularAutomaton



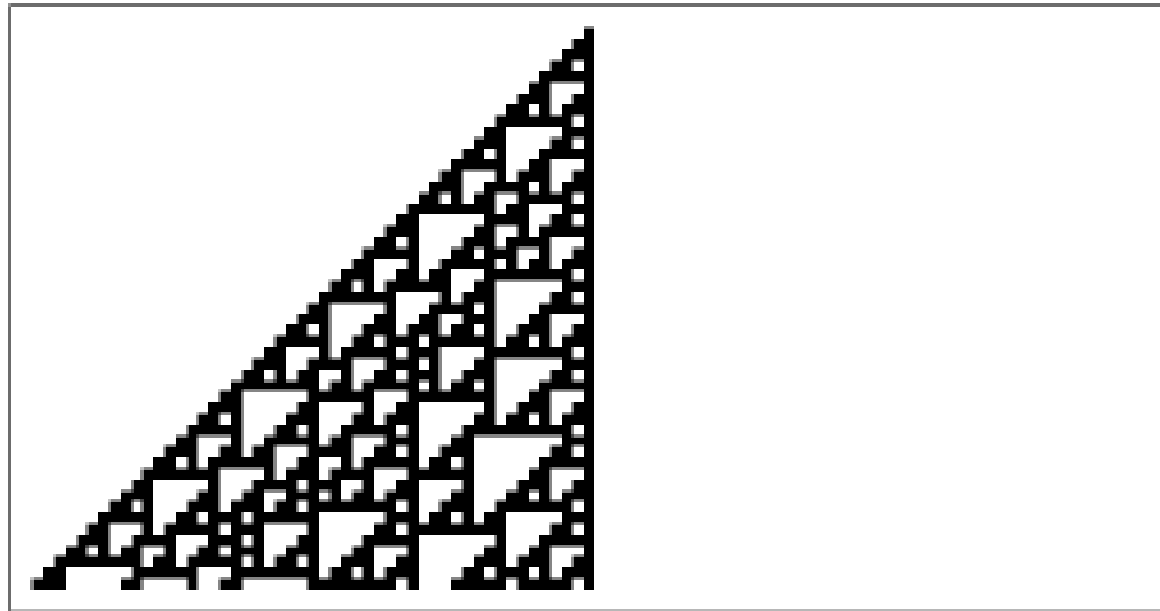
- **CellularAutomaton[reguła, stan początkowy, {t, all}]** generuje listę przedstawiającą ewolucję automatu komórkowego za pomocą określonej reguły od stanu początkowego obejmując na każdym kroku wszystkie komórki, nawet te nieznaczące.

CellularAutomaton - przykład



```
In[34]:= ArrayPlot[CellularAutomaton[110, {{1}, 0}, {50, All}]]
```

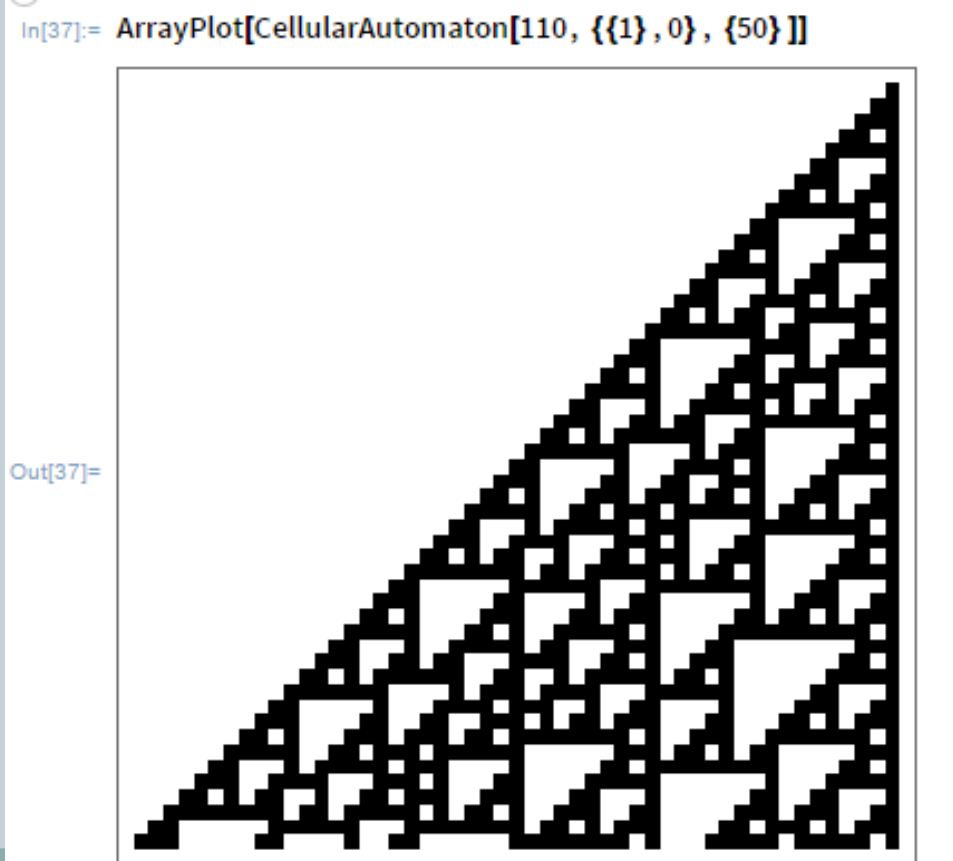
Out[34]=



CellularAutomaton - przykład



- Domyślnie CellularAutomaton usuwa wszystkie struktury nieobjęte wzorem.



Trójkąt Sierpińskiego



- Za pomocą reguły 90 możemy otrzymać trójkąt Sierpińskiego

```
In[55]:= CellularAutomaton[90, {{1}, 0}, 10] // Grid
```

```
Out[55]=  
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0  
0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0  
0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0  
0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0  
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0  
0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0  
1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
```

CellularAutomaton



- Wewnątrz komendy CellularAutomaton wartości mogą być przedstawiane za pomocą symboli.

```
In[53]:= CellularAutomaton[{{a, _, b} → a, {b, _, a} → b, {x_, _, x_} → a}, {{b}, a}, 4]
```

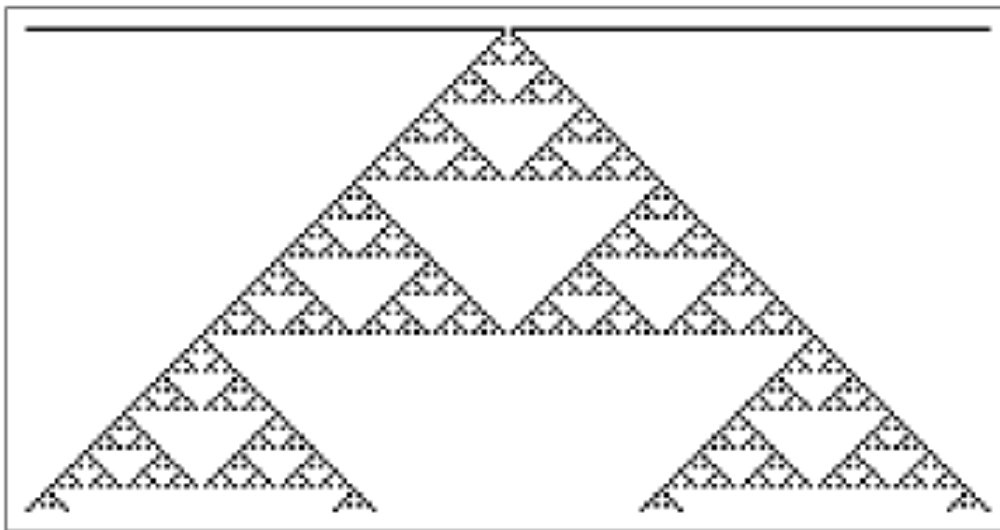
```
Out[53]= {{b, a, a, a, a}, {a, b, a, a, a}, {a, a, b, a, a}, {a, a, a, b, a}, {a, a, a, a, b}}
```

Trójkąt Sierpińskiego



```
In[49]:= ArrayPlot[CellularAutomaton[{{1, _, 0} → 1, {0, _, 1} → 1, {0, _, 0} → 0, {1, _, 1} → 0}, {{0}, 1}, 100]]
```

Out[49]=



Bibliografia



- <http://reference.wolfram.com/language/>