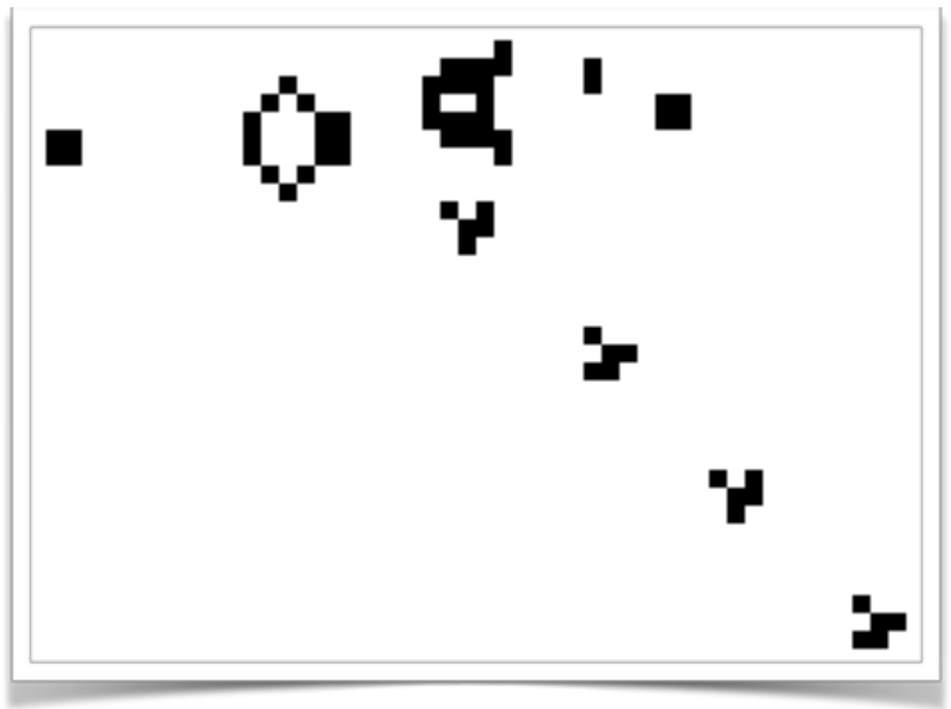
Pseudo Random Number Generator with Cellular Automation (Rule 30)

11510281 吴静柔

2018.12.27

Background

- Pseudo Random Number Generator (PRNG)
  - Generating a sequence of numbers whose properties approximate the properties of sequences of random numbers
- Cellular Automation (CA)
  - A discrete model consist of a regular grid of *cells*, each of a finite number of *states*



Rule 30

- One-dimensional binary rule
  - Every cell spontaneously changes state based on its current state and the state of its two neighbors
- Display aperiodic, chaotic behavior

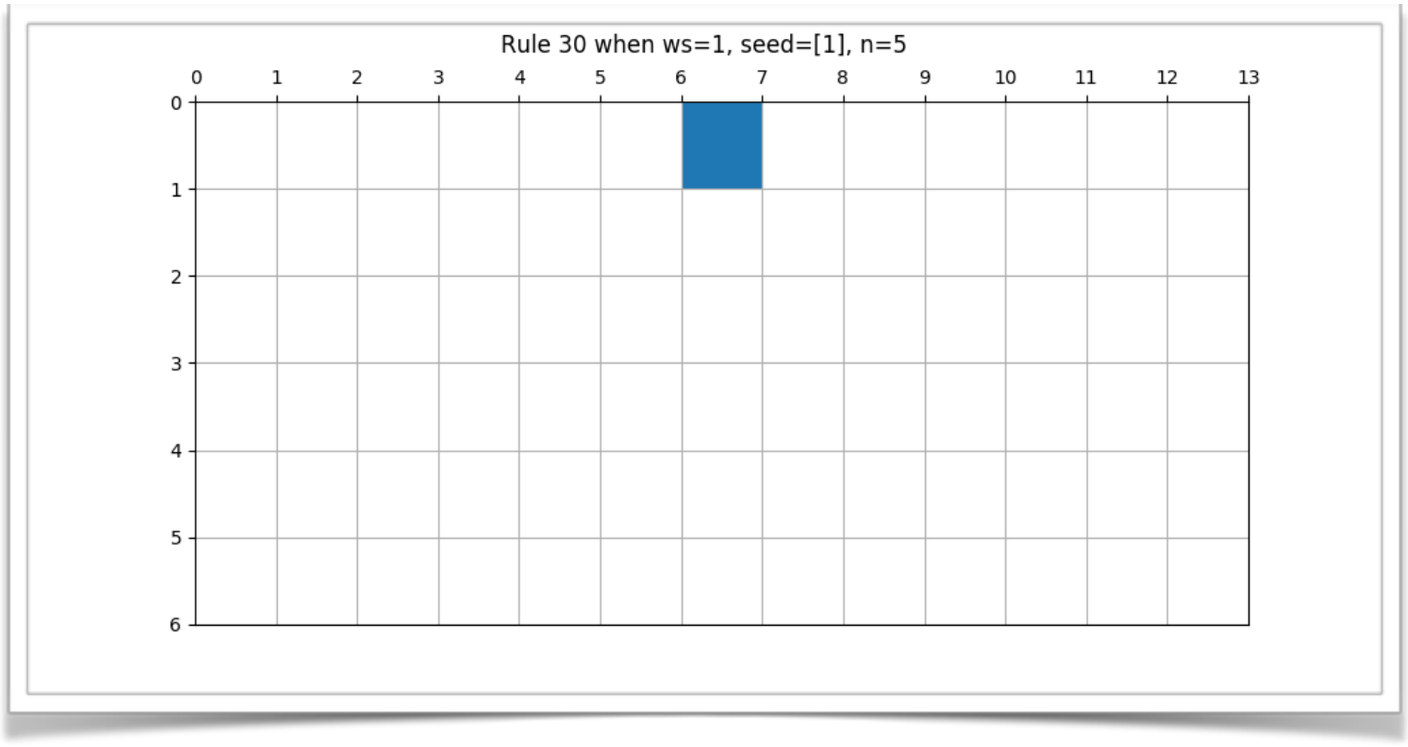


Fig 1. One example with seed = [1]

Table 1 The new state of the cell

Current pattern	110	110	101	100	011	010	001	000
New state of center cell	0	0	0	1	1	1	1	0

PRNG with Rule 30

- Use the center column
- Parameters
  - Window size : # of binary bits
- Seed: Initial 01 string
- n: iteration times

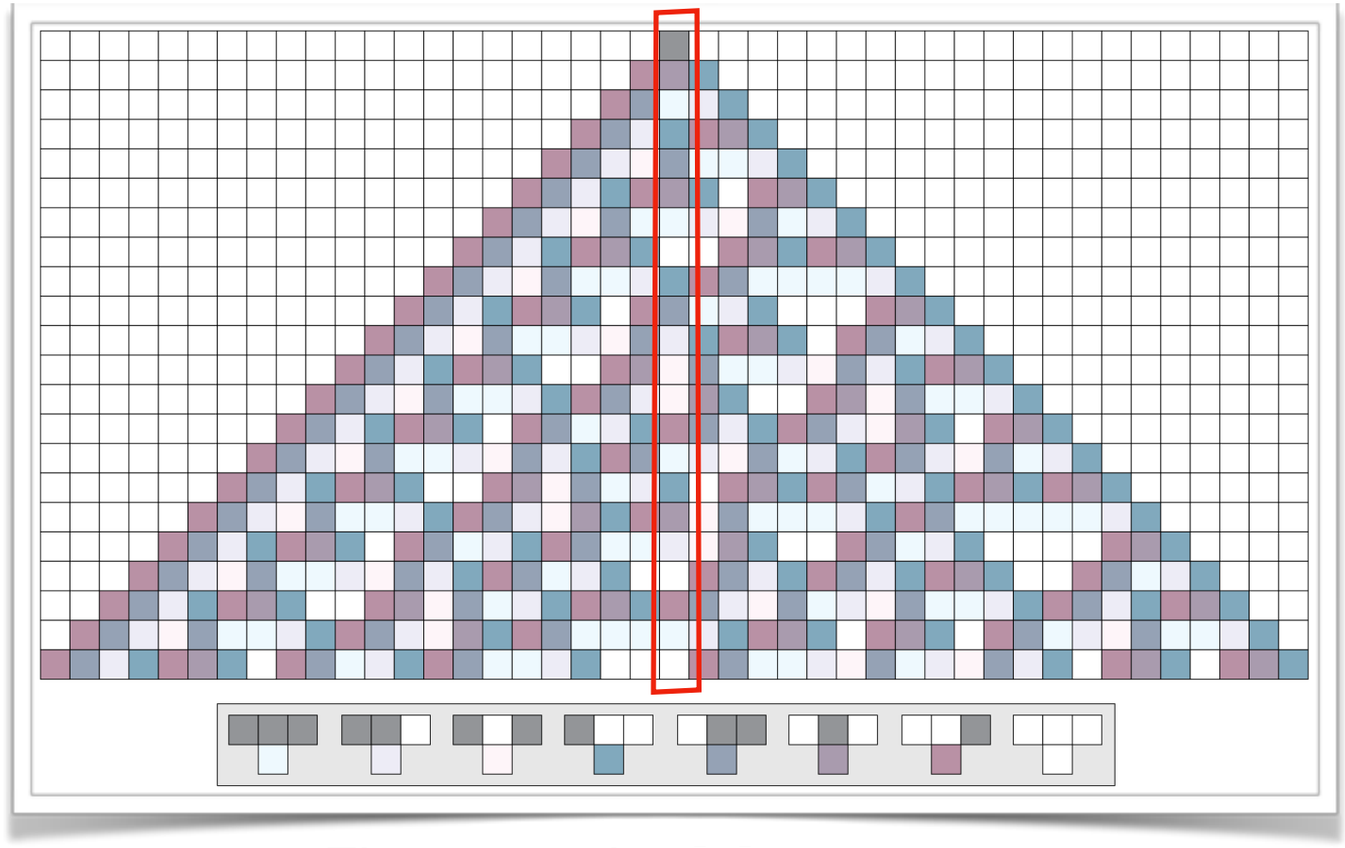


Fig 2. seed = [1], n = 21

Table 2 PRNG with different window size

Window size = 1	[1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0]
Window size = 2	[3, 2, 1, 3, 3, 2, 0, 1, 3, 2, 0, 0, 1, 2, 1, 3, 2, 0, 1, 2]

NIST Randomness Test

Ws = 1, n=100, seed = [1]	p-value (Result)	Memo	Recommendation
Frequency Test	0.6892 (Pass)		n >= 100
Frequency Test within a Block	0.8013 (Pass)	M = 11	n >= 100
Runs Test	0.4138 (Pass)		n >= 100
Longest Run Of Ones			n > 128
Binary Matrix Rank Test	0.4171 (Pass)	M=Q=3	M = Q = 32
Discrete Fourier Transform	0.7456 (Pass)		n>=1000
Non-overlapping Template	0.3841 (Pass)	M=10, B=001	m=9/10
Overlapping Template Matching	0.7362 (Pass)	M=10, K=5, B=[1,1]	n>=10^6
Maurer's“Universal	0.9291 (Pass)	M=2, Q=4	n>=387,840
Linear Complexity Test	0.8088 (Pass)	M=13, K=6	n>=10^6
Serial Test	2/2 p-value>0.01(Pass)		$m < \lfloor \log_2 n \rfloor - 2$
Approximate Entropy Test	0.8843 (Pass)	m=2	$m < \lfloor \log_2 n \rfloor - 2$
Cumulative Sums (Cusum) Test	0.6292 (Pass)	Forward	n >= 100
Random Excursions Test	4/8 p-value <0.01	Further test needed	n>=10^6
Random Excursions Variant Test	2/18 p-value <0.01	Further test needed	n>=10^6

Resources

- My codes:
  - Github: https://github.com/wjr0102/PRNG_with_rule30.git
    - Code of rule 30, some other PRNGs and NIST randomness test (python 2.7)
- NIST randomness test official website
  - Document and download: https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software
- The software using rule 30 to generate random number
  - Wolfram Mathematica: http://www.wolfram.com/mathematica/

References

- Wolfram, S. . (1986). Random sequence generation by cellular automata. *Advances in Applied Mathematics, 7*(2), 123-169.

- Spencer, J. . (2013). Cellular automata in cryptographic random generators. *Computer Science*.

- Gage, D.., Laub, E.., Mcgarry, G.. Cellular Automate: Is Rule 30 Random? (https://www.cs.indiana.edu/~dgerman/2005midwestNKSconference/dgelbm.pdf)

- NIST SP 800-22 Guidance (https://csrc.nist.gov/Projects/Random-Bit-Generation/Documentation-and-Software/Guide-to-the-Statistical-Tests)

Thanks