8.1 Random Number Generators

True RNG → slow. Expensive

Pseudo RNG → Deterministic Algorithms

↳ Deterministic sequence of numbers that approximately random

Goal: Create uniform distribution between 0 and 1.

$\sim U[0,1)$

reality: $\{0, \dots, m-1\}$

$x \sim \{0, \dots, m-1\}$

$u = \frac{x}{m-1} \sim U[0,1]$

$= \frac{x}{m} \simeq U[0,1)$

Definition

Pseudo Random Number Generator

Struct (S, μ, f, U, g)

S = finite set of states $(m = |S|)$

μ = a distribution on S for the initial state (seed)

~~f = function $S \to S$~~

$f: S \to S$, state transition function

U = output space, $[0,1)$, ~~or~~ perhaps $\{0, \dots, m-1\}$

$g: S \to U$, output function

How do we use PRNG?

Select $x_0 \sim \mu$, $u_0 = g(x_0)$

1. For $n \geq 1$

$x_n = f(x_{n-1})$

$u_n = g(x_n)$

~~2. Stop at 1 goto step~~

$u_0, u_1, \dots$ are PR numbers

PRNG are finite state machines → they WILL repeat in finite time

Period = m

$\exists p \leq 6$

$\exists n \geq k \quad s_{p+n} = s_n$

Three families of PRNGs

1. Linear congruential generators (LCG)
2. Linear Feedback shift registers (LFSR)
3. Cellular Automata (CA)

LCG

Goto for a loong time.

$$x_{n+1} = (a x_n + b) \bmod m$$

$$u_n = \frac{x_n}{m}$$

a: multiplier, b: increment, m: modulus

DEFINITION
MODULO FUNCTIONS

$r = x \bmod m$

$x = km + r$

r: remainder

~~Theorem: Hull-Dobell theorem~~

LCG with a, c, m, x_0 has period m iff:

1. c is relatively prime to m ("coprime")
2. $b = a - 1$ is a multiple of p for every prime p in m.
3. b is a multiple of 4 if m is a multiple of 4.

MCG's exist. → multiplicative / mixed congruential generators

$c = 0$

$x_{n+1} = a x_n \bmod m$ → max period $m-1$

→ RANDU from IBM

LFSR

- Fast!
- XOR's
- shift gates

Steps:
1. $x = x \text{ XOR } (x << 13)$
2. $x = x \text{ XOR } (x >> 17)$
3. $x = x \text{ XOR } (x << 15)$

MERSENNE TWISTER

Uses rand no. in bitshift → Python RAND!

Why XOR and shifts?
- XOR'ing different bit patterns you get complex mixing
- Creates an avalanche effect

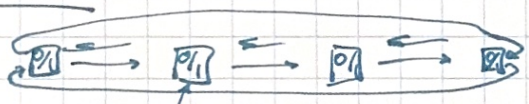
Adv.
- very fast
- tiny state
- simple to implement
- good enough

Dis.
- Fails some rigorous tests (Bigcrush, testU01)
- Not suitable for crypto

PCG: Permuted Congruential Generator (Convolutional?)

combines LCG step plus LFSR step

CA: Cellular Automata

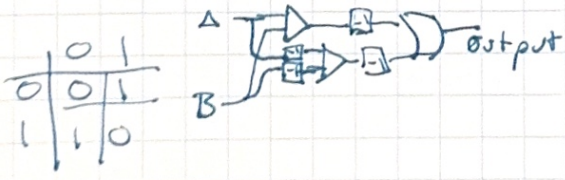


Rule 30

111	110	101	100	011	010	001	000
0	0	0	1	1	1	1	0

$$u = \sum_{i=1}^{m-1} b_i 2^{-i}$$

- Good for multiple threads
- Parallel process!
- chaotic



	0	1
0	0	1
1	1	0