RW778: Implementation and Application of Automata, 2006
Week 5 Lecture 1

L. van Zijl

Department of Computer Science
University of Stellenbosch

2006

Random Number Generation with Cellular Automata

References:

1. Wolfram, Random Number Generation with Cellular Automata.
2. Law, Kelton: Simulation Modelling, chapter 7.
3. L'Ecuyer: Tausworthe Generators, Testing RNGs

Random Number Generation with Cellular Automata

- Pseudo-random number generators

Random Number Generation with Cellular Automata

- Pseudo-random number generators
- Behave as if uniformly distributed

Random Number Generation with Cellular Automata

- Pseudo-random number generators
- Behave as if uniformly distributed
- Algorithmic, therefore cycle

Random Number Generation with Cellular Automata

- Pseudo-random number generators
- Behave as if uniformly distributed
- Algorithmic, therefore cycle
- Main classes of algorithms: LCGs, Tausworthe

Random Number Generation with Cellular Automata

- Pseudo-random number generators
- Behave as if uniformly distributed
- Algorithmic, therefore cycle
- Main classes of algorithms: LCGs, Tausworthe
- LCG:

Random Number Generation with Cellular Automata

- Pseudo-random number generators
- Behave as if uniformly distributed
- Algorithmic, therefore cycle
- Main classes of algorithms: LCGs, Tausworthe
- LCG:
  - $X_i = (aX_{i-1} + b) \mod m$

Random Number Generation with Cellular Automata

- Pseudo-random number generators
- Behave as if uniformly distributed
- Algorithmic, therefore cycle
- Main classes of algorithms: LCGs, Tausworthe
- LCG:
  - $X_i = (aX_{i-1} + b) \mod m$
  - Choices of a, b, m, X_0 critical.

Random Number Generation with Cellular Automata

- Pseudo-random number generators
- Behave as if uniformly distributed
- Algorithmic, therefore cycle
- Main classes of algorithms: LCGs, Tausworthe
- LCG:
  - $X_i = (aX_{i-1} + b) \mod m$
  - Choices of a, b, m, X_0 critical.
- Tausworthe:

Random Number Generation with Cellular Automata

- Pseudo-random number generators
- Behave as if uniformly distributed
- Algorithmic, therefore cycle
- Main classes of algorithms: LCGs, Tausworthe
- LCG:
  - $X_i = (aX_{i-1} + b) \mod m$
  - Choices of a, b, m, X_0 critical.
- Tausworthe:
  - Stream of bits b_i in groups

Random Number Generation with Cellular Automata

- Pseudo-random number generators
- Behave as if uniformly distributed
- Algorithmic, therefore cycle
- Main classes of algorithms: LCGs, Tausworthe
- LCG:
  - $X_i = (aX_{i-1} + b) \mod m$
  - Choices of a, b, m, X_0 critical.
- Tausworthe:
  - Stream of bits b_i in groups
  - Function uses XOR of previous bits

Random Number Generation with Cellular Automata

- Pseudo-random number generators
- Behave as if uniformly distributed
- Algorithmic, therefore cycle
- Main classes of algorithms: LCGs, Tausworthe
- LCG:
  - $X_i = (aX_{i-1} + b) \mod m$
  - Choices of a, b, m, X_0 critical.
- Tausworthe:
  - Stream of bits b_i in groups
  - Function uses XOR of previous bits
- Good RNG: long cycle, no discernible patterns

Random Number Generation with Cellular Automata

- Pseudo-random number generators
- Behave as if uniformly distributed
- Algorithmic, therefore cycle
- Main classes of algorithms: LCGs, Tausworthe
- LCG:
  - $X_i = (aX_{i-1} + b) \mod m$
  - Choices of a, b, m, X_0 critical.
- Tausworthe:
  - Stream of bits b_i in groups
  - Function uses XOR of previous bits
- Good RNG: long cycle, no discernible patterns
- Testing: Theoretical, empirical (dotplot)

Random Number Generation with Cellular Automata

- Consider 1D CA, with given rules.

Random Number Generation with Cellular Automata

- Consider 1D CA, with given rules.
- Rewrite CA in matrix notation.

Random Number Generation with Cellular Automata

- Consider 1D CA, with given rules.
- Rewrite CA in matrix notation.
- Calculate characteristic polynomial $c(X)$.

Random Number Generation with Cellular Automata

- Consider 1D CA, with given rules.
- Rewrite CA in matrix notation.
- Calculate characteristic polynomial $c(X)$.
- Period of $c(X)$ determines length of cycle.

Random Number Generation with Cellular Automata

- Consider 1D CA, with given rules.
- Rewrite CA in matrix notation.
- Calculate characteristic polynomial $c(X)$.
- Period of $c(X)$ determines length of cycle.
- Connection to LFSRs – implies hardware implementation possible.

Random Number Generation with Cellular Automata

Good rules:

- $a_i^{'} = a_{i-1}$ XOR (a_i OR a_{i+1})

Random Number Generation with Cellular Automata

Good rules:

- $a_i' = a_{i-1}$ XOR (a_i OR a_{i+1})
- $a_i' = a_{i-1}$ XOR (a_i OR (NOT a_{i+1}))

Cellular Automata
Homework

Homework: Use your CA implementation to generate random numbers. Hand in binary (0-1) output, as well as output converted to integer format. Provide a dot-plot of the output (see Law and Kelton, pp 443–445). Test your output with the runs-up test.