28 de agosto de 2018
Universidade Federal de Santa Catarina

INE5425 – Modelagem e Simulação
Relatório Desenvolvimento de Software 1

Alunos:
Jacyara Bosse (13200650)
Telmo Henrique Valverde da Silva (13202970)

Tema: Diferentes implementações e avaliações de métodos para geração de números pseudo-aleatórios

Dicionário de termos:

- Semente / *seed*: O valor de entrada em um algoritmo de geração de números aleatórios.
- Período: Número de iterações de um algoritmo de geração de números pseudo aleatórios na qual o conjunto ainda pode ser considerado aleatório. Executar o algoritmo além do seu período máximo implica na repetição dos mesmos números na mesma ordem.
- *Big Crush[1]*: A maior bateria de testes de aleatoriedade disponível no programa *TestU01*, totalizando 160 testes.

1) Gerador Linear Congruencial

O método do gerador linear congruencial*[2;3]* (*linear congruential generator / LCG*) foi divulgado inicialmente pelo Prof. D. H. Lehmer em 1951, onde seus estudos apontavam que restos de potências sucessivas de um número possuem boas características de aleatoriedade.

O método começa com a *seed* de *X,* sendo 0 <= *X* < m. A variável “a” é o multiplicador, “c” é o incremento (sendo 0 < a < m e 0 <= c < m) e “m” é o módulo (sendo m > 0). A cada iteração a seguinte fórmula é utilizada, resultando em um número aleatório:

$$X_{n+1} = (aX_n + c) \mod m$$

Apesar do algoritmo ser capaz de passar em testes formais de aleatoriedade, a sua funcionalidade está diretamente atrelada à escolha apropriado para o valor das suas variáveis, por exemplo a utilização de a = 1 e c = 1 torna o algoritmo um simples contador do módulo m. A escolha de valores apropriados para as variáveis “a”, “c” e “m” também aumenta a confiabilidade da geração aleatória e o tamanho do período, por isso optamos por utilizar na nossa implementação os mesmos valores utilizados pelo popular compilador de código fonte aberto GCC. Sendo m = “2147483648”, a = “1103515245” e c = “12345”.

2) Registrador de deslocamento com feedback linear

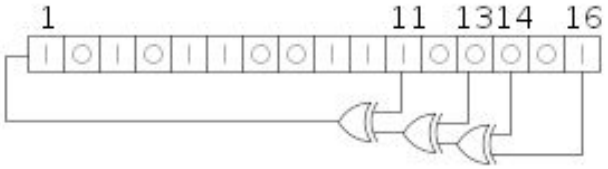
Os registradores de deslocamento com feedback linear*[4]* (*linear-feedback shift register / LFSR*) são registradores de deslocamento nos quais o bit de entrada é dado por uma função linear do estado anterior do registrador. O valor inicial de um LFSR é chamado de *seed* e a

geração de números através desse método é sempre determinada pelo último estado do registrador.

Apesar de o número de estados possíveis ser finito, a utilização de uma boa função linear é capaz de produzir uma sequência de números que parece aleatória e tem um ciclo bastante longo. Na nossa implementação foi utilizada a seguinte função, que é capaz de gerar 65.535 números diferentes, partindo de qualquer *seed* (0x7f2a, no nosso caso) diferente de 0:

$$x^{16} + x^{14} + x^{13} + x^{11} + 1$$

Os índices indicam em quais bits será aplicada a operação XOR que resultará no valor do bit a ser colocado na primeira posição do registrador após o deslocamento do conteúdo dele para a direita. O resultado dessa operação será o número aleatório gerado, assim como a *seed* para a geração do próximo número aleatório.



3) Método Middle-Square

O método *Middle-Square[5]* foi proposto por John Von Neumann, e utiliza como *seed* um número de quatro dígitos, este número é então elevado ao quadrado, produzindo um número de 8 dígitos e deste são extraídos os quatro algarismos do centro que irão formar a semente da próxima iteração, onde é repetido os passos descritos.

Uma das desvantagens e o motivo por não ser considerado um bom algoritmo é que a sequência de números gerada tende a se repetir após poucas iterações (curto período), um outro caso é quando o meio do quadrado coincide com uma sequência de zeros, parando a execução do algoritmo.

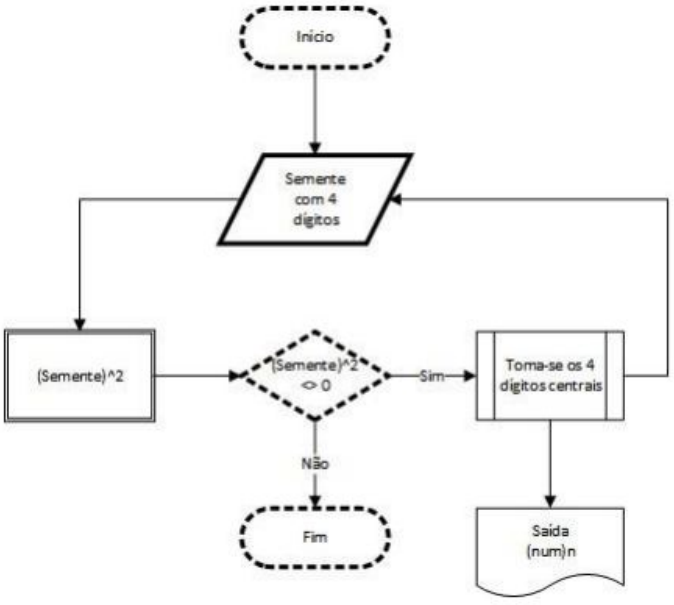


Figura 1* Método do meio do quadrado

4) Multiply-with-Carry

O método *Multiply-with-Carry[6]* proposto por George Marsaglia, gera números inteiros aleatórios baseados em um conjunto inicial de duas sementes, o **X** e o **C**. E sendo o “a” o multiplicador, o “b” a base e o “r” geralmente representado por 1, o que torna esse método similar ao linear congruencial no sentido de utilizar o resultado do anterior para fazer seu cálculo. Em comparação dos dois métodos em questão de eficiência, o linear congruencial gera até até 2^32 números e o MWC até um pouco menos de 2^63 com o mesmo período.

$$x_n = (ax_{n-r} + c_{n-1}) \bmod b,\ c_n = \left\lfloor \frac{ax_{n-r} + c_{n-1}}{b} \right\rfloor$$

5) XORshift

Assim como o *Multiply-with-carry*, o *XORShift[7]* é um método proposto por George Marsaglia, que fornece o próximo número fazendo um deslocamento e depois um XOR com o número atual. Ele é considerado um tipo específico de método de registrador de deslocamento com feedback linear. No trabalho foi implementada a versão de 32 bits, que não passa no conjunto de testes *Big Crush*.

```c
int xorshift(int x) { // XORshift
  uint32_t seed = 548787455;
  int i = 0;

  for (int i = 1; i <= x; i++) {
    uint32_t temp = seed;
    temp ^= temp << 13;
    temp ^= temp >> 17;
    temp ^= temp << 5;
    seed = temp;

    printf("%u (%u)\n", temp, i);
  }

  return 0;
}
```

Referências:

[1] https://en.wikipedia.org/wiki/TestU01 (acesso em 28/08/2018)
[2] https://en.wikipedia.org/wiki/Linear_congruential_generator (acesso em 15/08/2018)
[3] http://www.dca.fee.unicamp.br/~rafael/ia369/var_aleat.pdf (acesso em 28/08/2018)
[4] https://en.wikipedia.org/wiki/Linear-feedback_shift_register (acesso em 28/08/2018)
[5] https://en.wikipedia.org/wiki/Middle-square_method (acesso em 28/08/2018)

[6] https://en.wikipedia.org/wiki/Multiply-with-carry (acesso em 28/08/2018)
[7] https://en.wikipedia.org/wiki/Xorshift (acesso em 28/08/2018)

[Figura 1] DA SILVA JÚNIOR, João Ferreira; DE OLIVEIRA MENDONÇA, Sérgio Francisco Tavares; DE CARVALHO JÚNIOR, Edson Alves. Simulação: Pseudoaleatoriedade, um estudo sobre o método do meio do quadrado. **Revista da Escola Regional de Informática**, v. 2, n. 2, p. 123-127, 2013.