Course 08/1

Random number generators (2/2)

- Portable algorithms: Schrage's algorithm
- Non-portable algorithms
- Shift-register generator

Portable algorithms: Schrage's algorithm

Motivation

Development of a computer-independent scheme for generating random numbers by avoiding overflow upon multiplication. The method is based on an approximate factorization.

Premises

Suppose we have a "modulo" generator with given a and m :

$m = a \cdot q + r$ where $q = [\, m / a \,]$ and $r = m \bmod a$.

integer part

Extra condition required for this algorithm to work: $r < q$

Suppose the i-th random number is x_i. Then $z = x_i$ and $0 < z \leq m - 1$.

For determining the next random number, we need to calculate $x_{i+1} = (a \cdot z) \bmod m$.

z can always be written as: $z = [\, z / q \,]\ q + (z \bmod q)$.

Portable algorithms: Schrage's algorithm

$z = [\, z / q \,] \cdot q \;+\; (\, z \bmod q \,)$ $\qquad\qquad$ $x_{i+1} = (\, a \cdot z \,) \bmod m$

Demonstration

$$x_{i+1} = (\, a \cdot z \,) \bmod m \;=\; \{\, [\, z / q \,] \cdot a \cdot q \;+\; a \cdot (\, z \bmod q \,) \,\} \bmod m \qquad m = a \cdot q + r$$

$$= \{\, [\, z / q \,] \cdot (\, m - r \,) + \; a \cdot (\, z \bmod q \,) \,\} \bmod m \qquad \text{because of mod } m$$

$$= \{\, a \cdot (\, z \bmod q \,) - r \;\; [\, z / q \,] \,\} \bmod m$$

1 2

Next, we show that both terms in the curly brackets belong to the interval: $(\, 0 \,,\, m - 1 \,]$.

Portable algorithms: Schrage's algorithm

We show that both terms belong to the interval $(0 , m - 1]$.

1 $$a \cdot (z \bmod q) \leq a \cdot (q - 1) \leq m - r - a < m$$

because of mod q

$aq = m - r$

2 $$r [z / q] \leq r \cdot a < q \cdot a = m - r$$

$m = a \cdot q + r$ for $z < m$, $[z / q] \leq a$

Schrage's condition

x_{i+1} can now be calculated without overflow: $x_{i+1} = (a \cdot z) \bmod m = \{ 1 - 2 \} \bmod m$

if 1 ≥ 2 : $x_{i+1} = 1 - 2$

if 1 < 2 : $x_{i+1} = 1 - 2 + m$

Non-portable algorithms

Some computers have a built-in capacity of dealing with overflow by performing modulo 2^{32}.

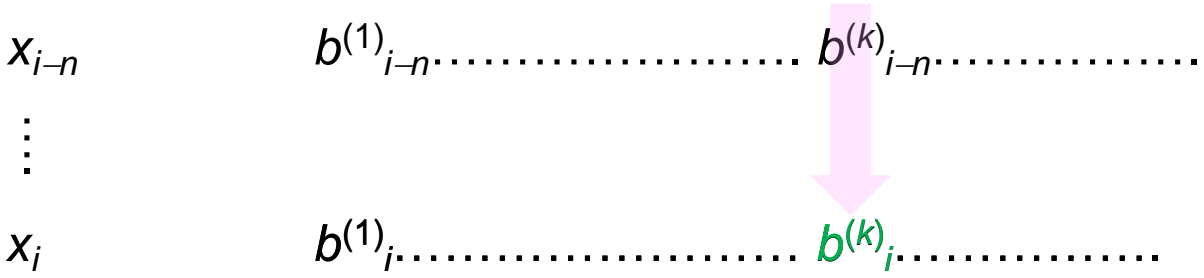
Advantage: very fast !

Example: $x_{n+1} = 69069 \cdot x_n$

Shift-register generator

In the computer, a number is memorized as a string of 0 and 1:

1	0	0	1	1	0	1	0	0	0	1	0	1	1	0



$$b^{(k)}_i = (c_1\, b^{(k)}_{i-1} + c_2\, b^{(k)}_{i-2} + \ldots + c_n\, b^{(k)}_{i-n}) \bmod 2 \qquad \text{with } c_i = 0/1$$

Maximum cycle: $2^{32} - 1$ (for special combinations of the c_i)

Start: n random numbers have to fill the register before starting

Simple version: $b^{(k)}_i = (b^{(k)}_{i-p} + b^{(k)}_{i-q}) \bmod 2$

with magic pairs (p,q) for optimal cycles: (98,27), (521,32), (250,103)

Course 08/1

Random number generators (2/2)

- Portable algorithms: Schrage's algorithm
- Non-portable algorithms
- Shift-register generator