Моделирование с использованием генераторов случайных чисел

Анализ сложности алгоритмов

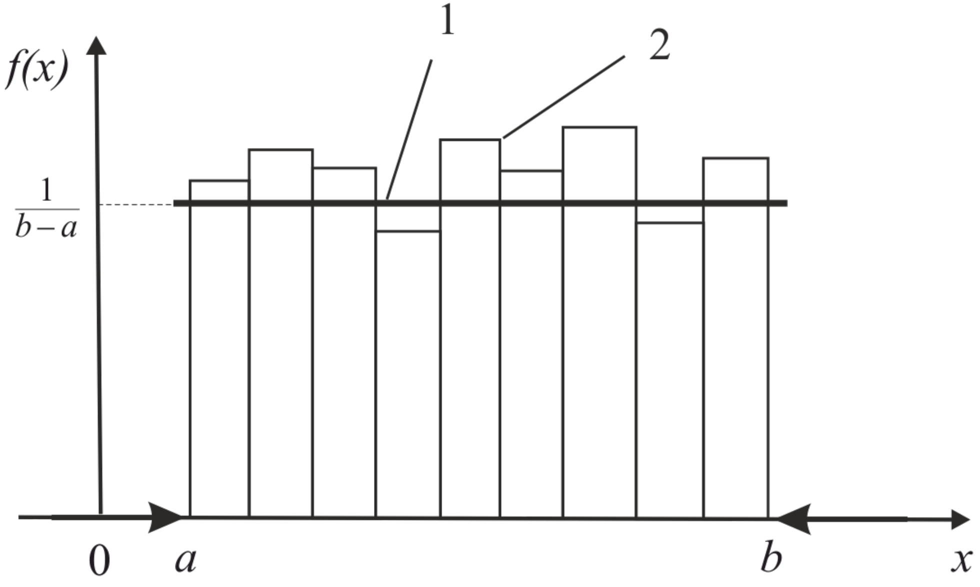
Логинов Сергей

НФИмд-01-22

Случайные числа

Главные свойства:

- Нельзя предсказать число до генерации
- Число не связано с другими числами последовательности и не зависит от них
- Числа распределены равномерно (или почти равномерно)



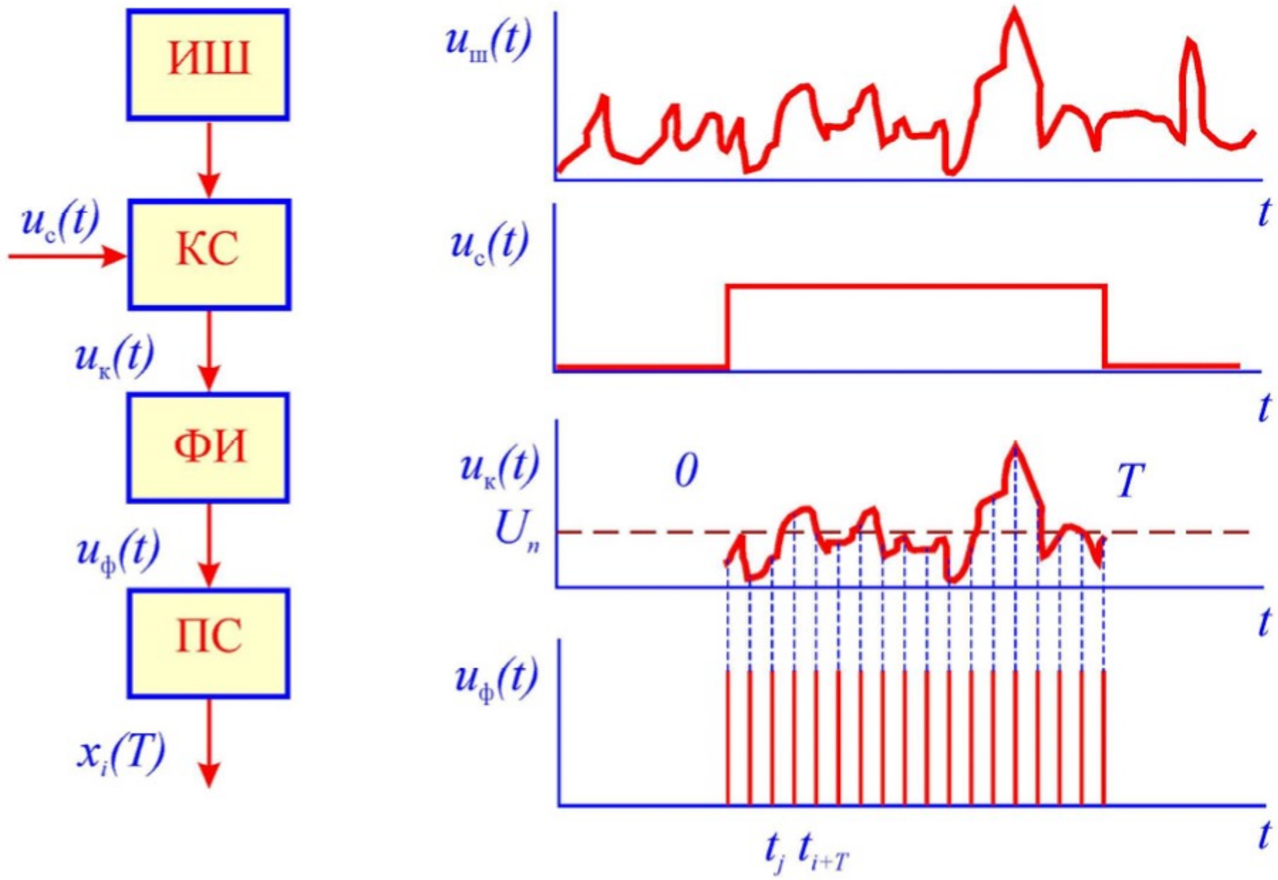
1- график функции плотности распределения вероятностей
2 - гистограмма

Генераторы случайных чисел

	Плюсы	**Минусы**
Генератор истинных случайных чисел	По-настоящему случайные числа	Сложность эксплуатации, требование внешнего источника а также считывающего устройства
	Практически не задействуется вычислительный ресурс	Долгая и дорогая генерация
Генератор псевдослучайных чисел	Быстрая и недорогая генерация	Псевдослучайность и повторение последовательности в пределе
	Множество алгоритмов	Использование системных ресурсов

Генераторы истинных случайных чисел (ГИСЧ)

- Радиоактивный распад атомов
- Дробовой шум
- Тепловой шум
- Атмосферный шум



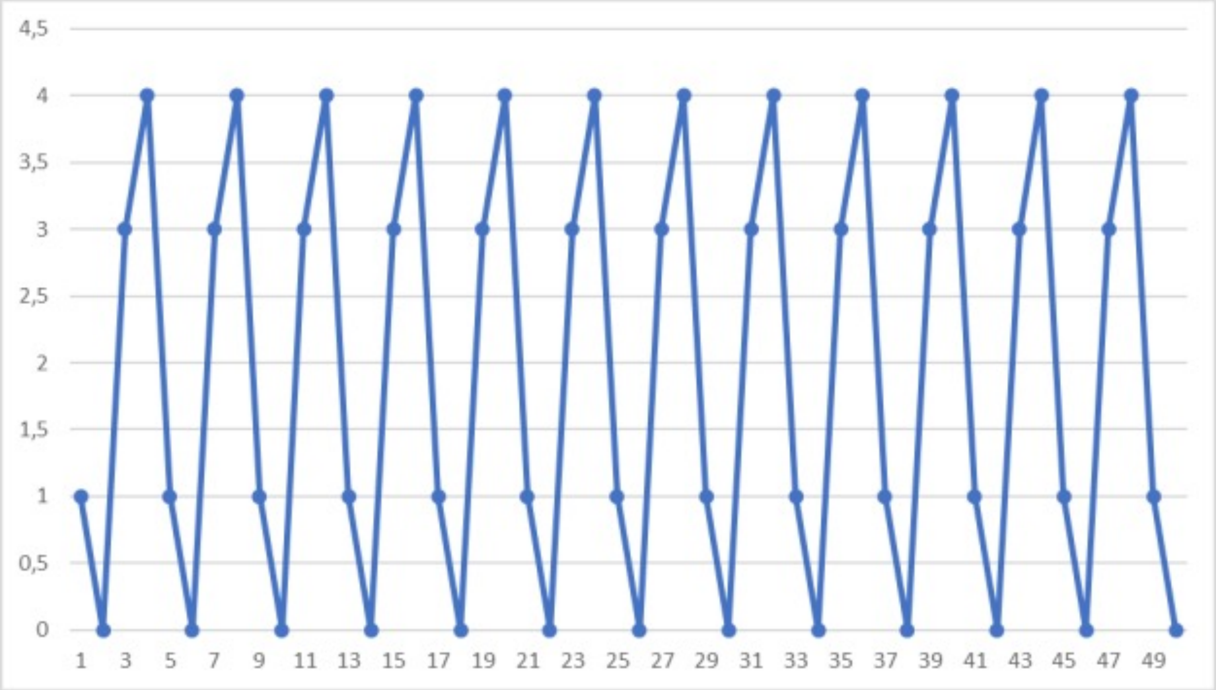
Генераторы псевдослучайных чисел (ГПСЧ)

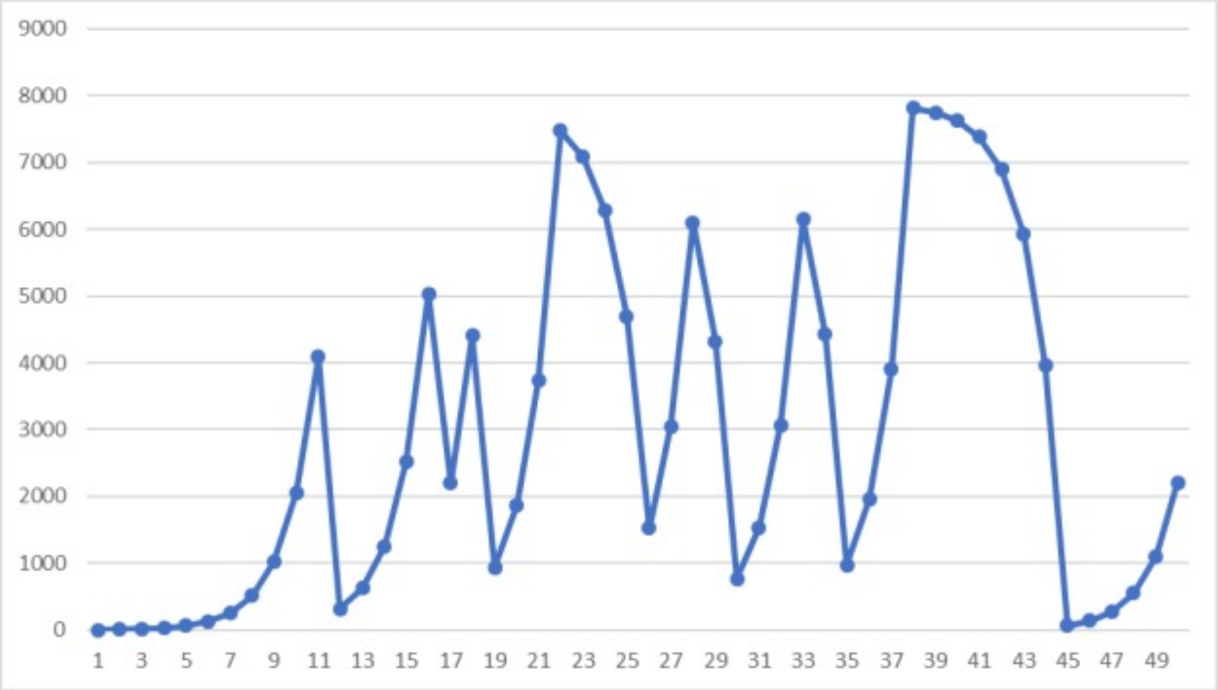
- Линейный конгруэнтный метод
- Метод перемешивания
- Метод квадратичных вычетов
- Blum Blum Shub
- ANSI X9.17
- PGP
- …

Линейный конгруэнтный метод (ЛКМ)

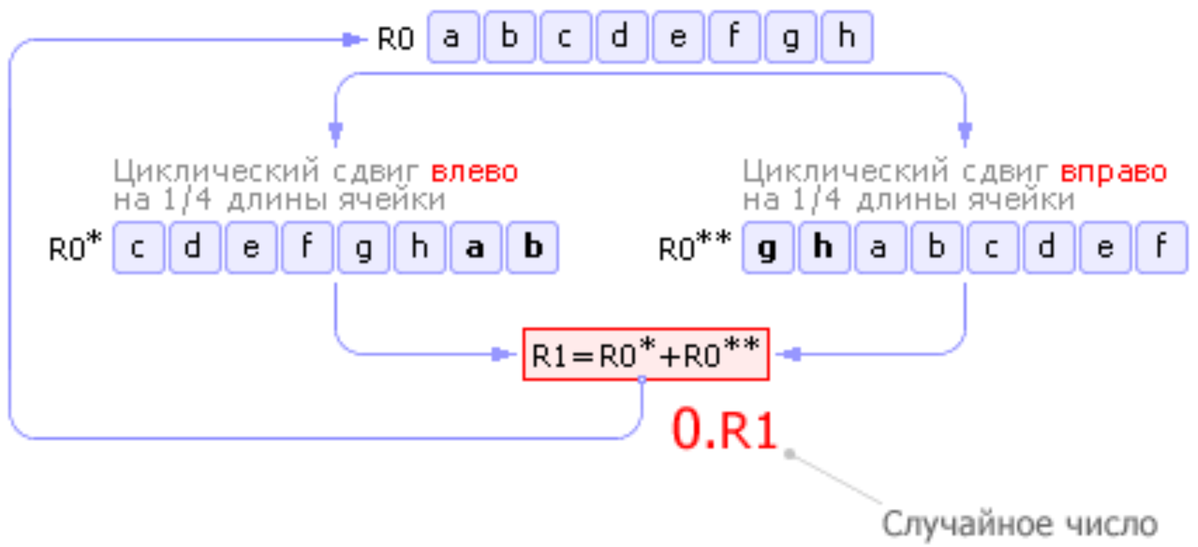
$$\boldsymbol{X_{n+1} = (aX_n + c)\ mod\ m,}$$

$$(0 < m < 2^{31} - 1), (0 \le a \le m), (0 \le c \le m)$$





Алгоритм перемешивания



$R = 8$ bit

$R_0{*} = 10010001_2 = 145_{10}$

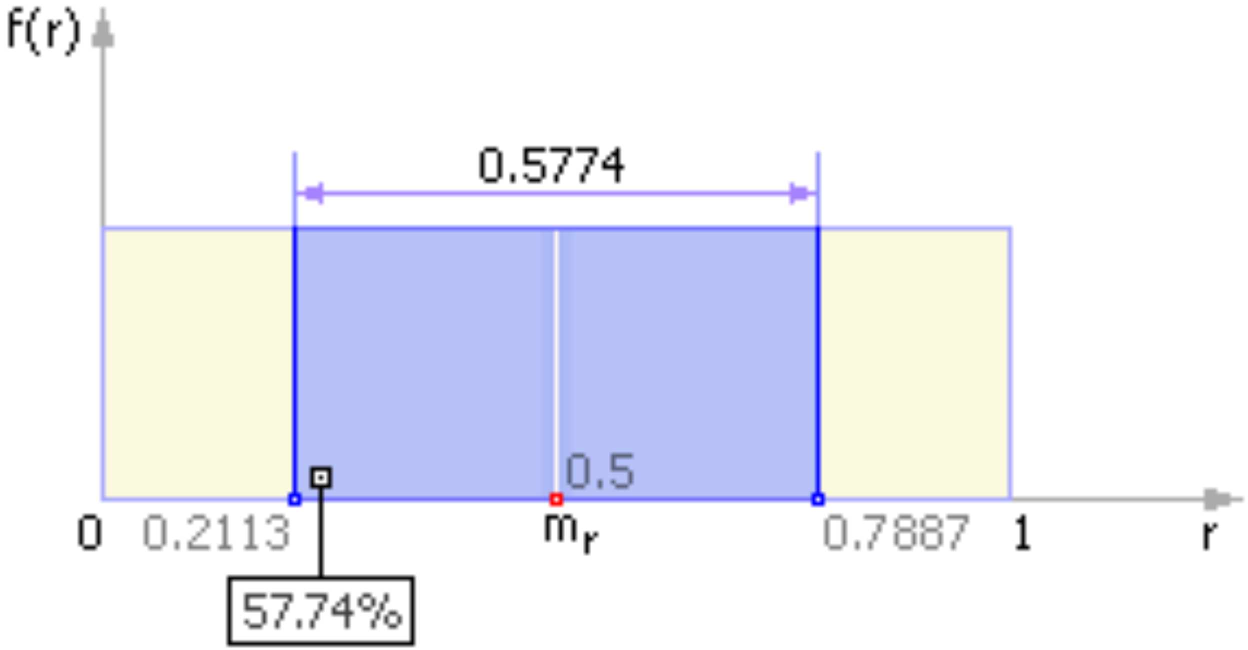
$R_0{**} = 10100001_2 = 161_{10}$

$R_0{*} + R_0{**} = 100110010_2 = 306_{10}$

R_1 (MSB/LSB) $= 00110010_2 = 50_{10}$

Проверка ГСЧ на равномерность

$m_r \approx 0{,}5,$
$D_r \approx 0{,}0833,$
$\sigma_r \approx 0{,}2887$



$$\chi^2_{\text{эксп}} = \sum_{i=1}^{k} \frac{(n_i - p_i * N)^2}{p_i * N} = \frac{1}{N} \sum_{i=1}^{k} \left(\frac{n_i^2}{p_i} \right) - N$$

Проверка ГСЧ на независимость

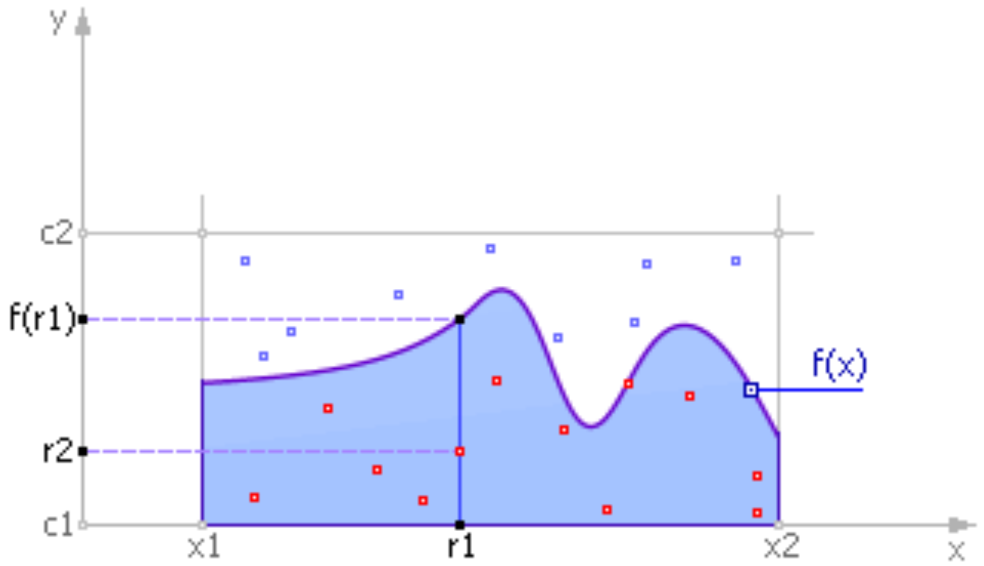
Проверка частоты появления цифры:

1. x_1 = 0. 2463389991, x_2 = 0.5467766618.
2. X = [2,4,6,3,3,8,9,9,9,1,5,4,6,7,7,6,6,6,1,8]
3. $p_{i\ теор} = 0.1,\ i \in [0, 9]$
4. $p_{i\ эксп}$ считается по частоте
5. $\boldsymbol{\chi}^{\mathbf{2}}_{эксп}$

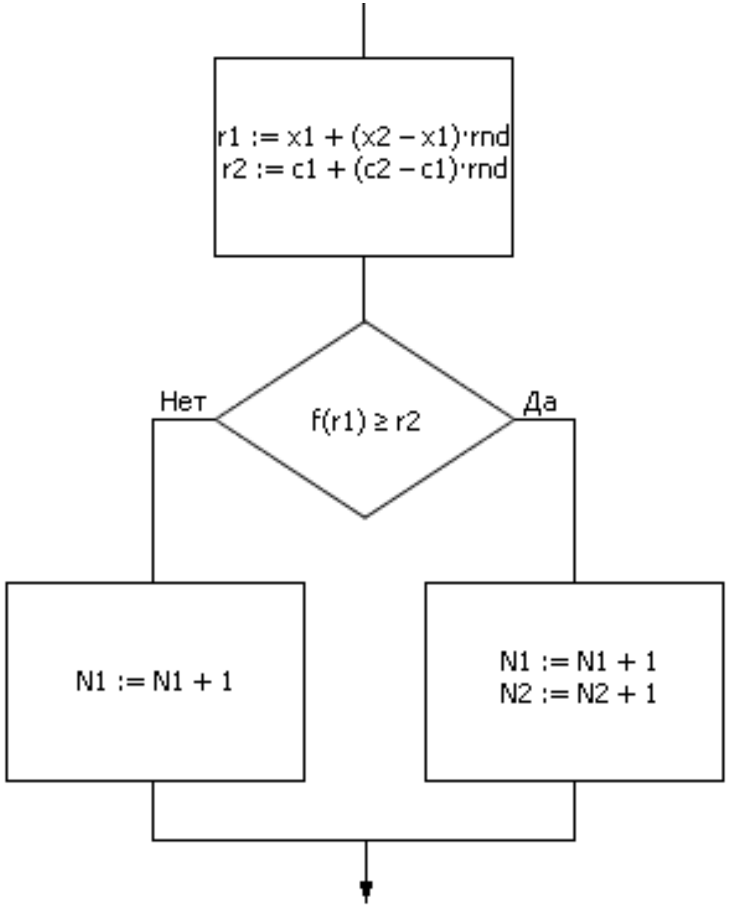
ГСЧ в моделировании

- Метод Монте-Карло
- Имитация случайных событий
- Моделирование полной группы несовместных событий
- Моделирование случайных величин
- Моделирование нормального распределения
- Моделирование потоков случайных событий
- Моделирование марковских процессов

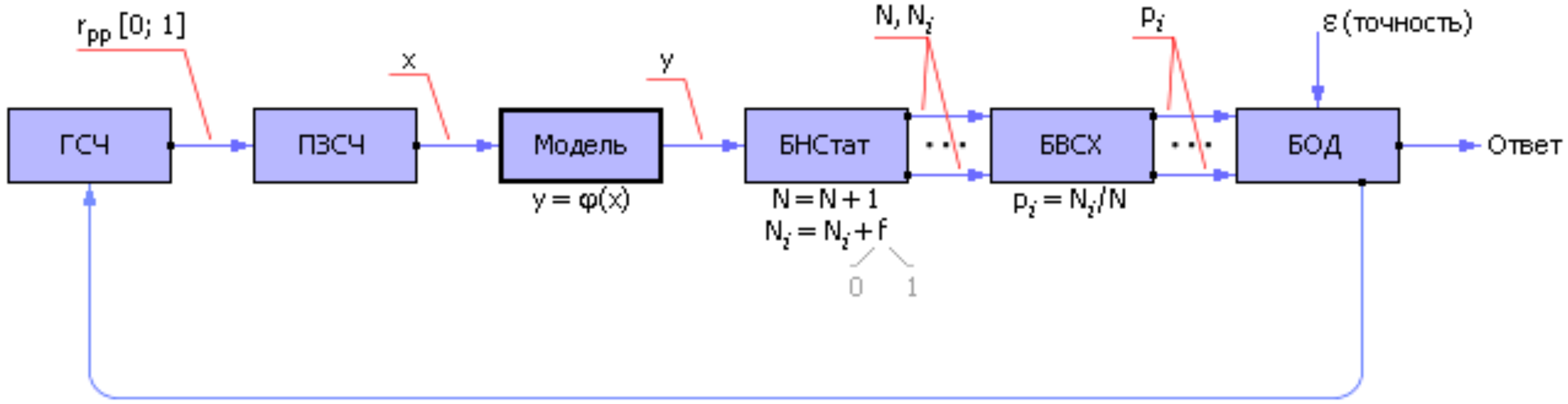
ГСЧ в методе Монте-Карло



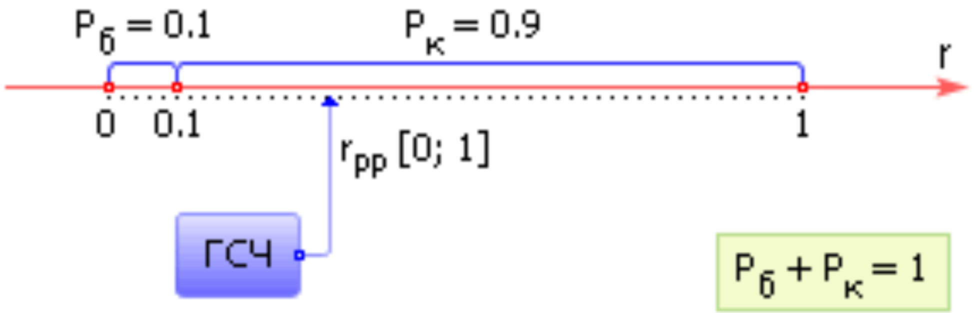
$$\frac{N_2}{N_1} = \frac{y}{(x_2 - x_1)(c_2 - c_1)}$$

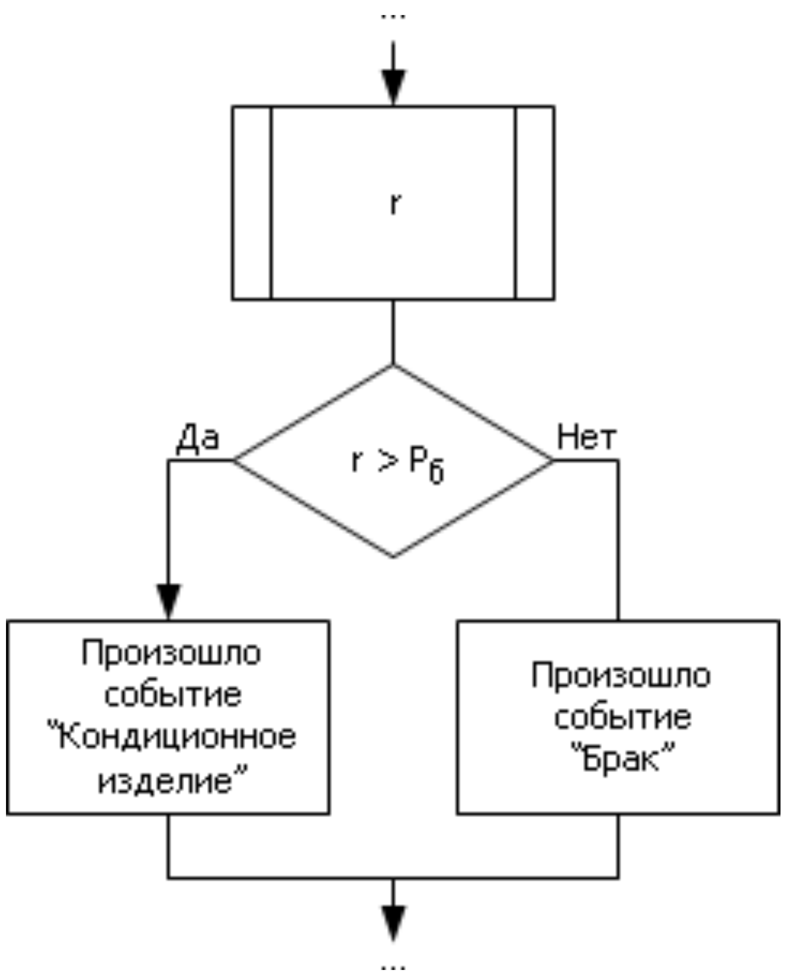


Использование метода Монте-Карло для исследования систем со случайными параметрами

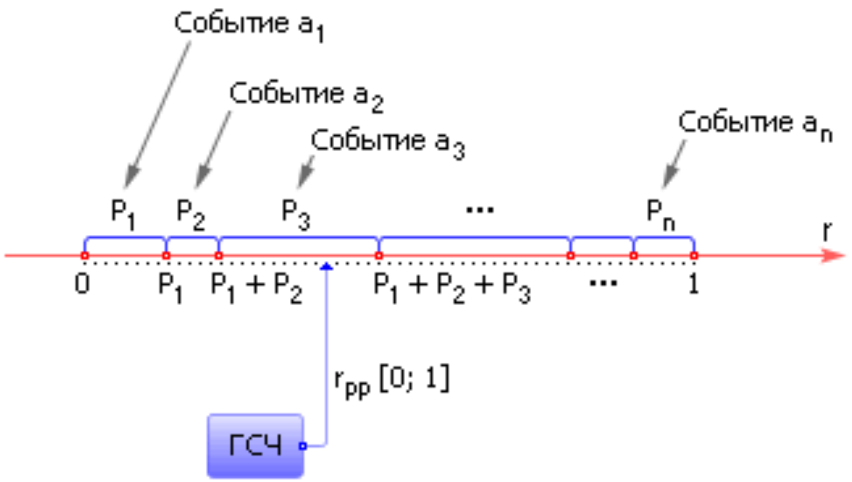


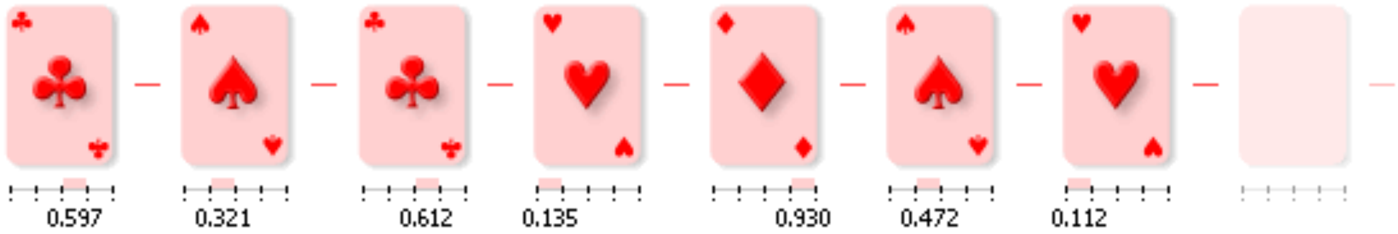
Имитация случайных событий

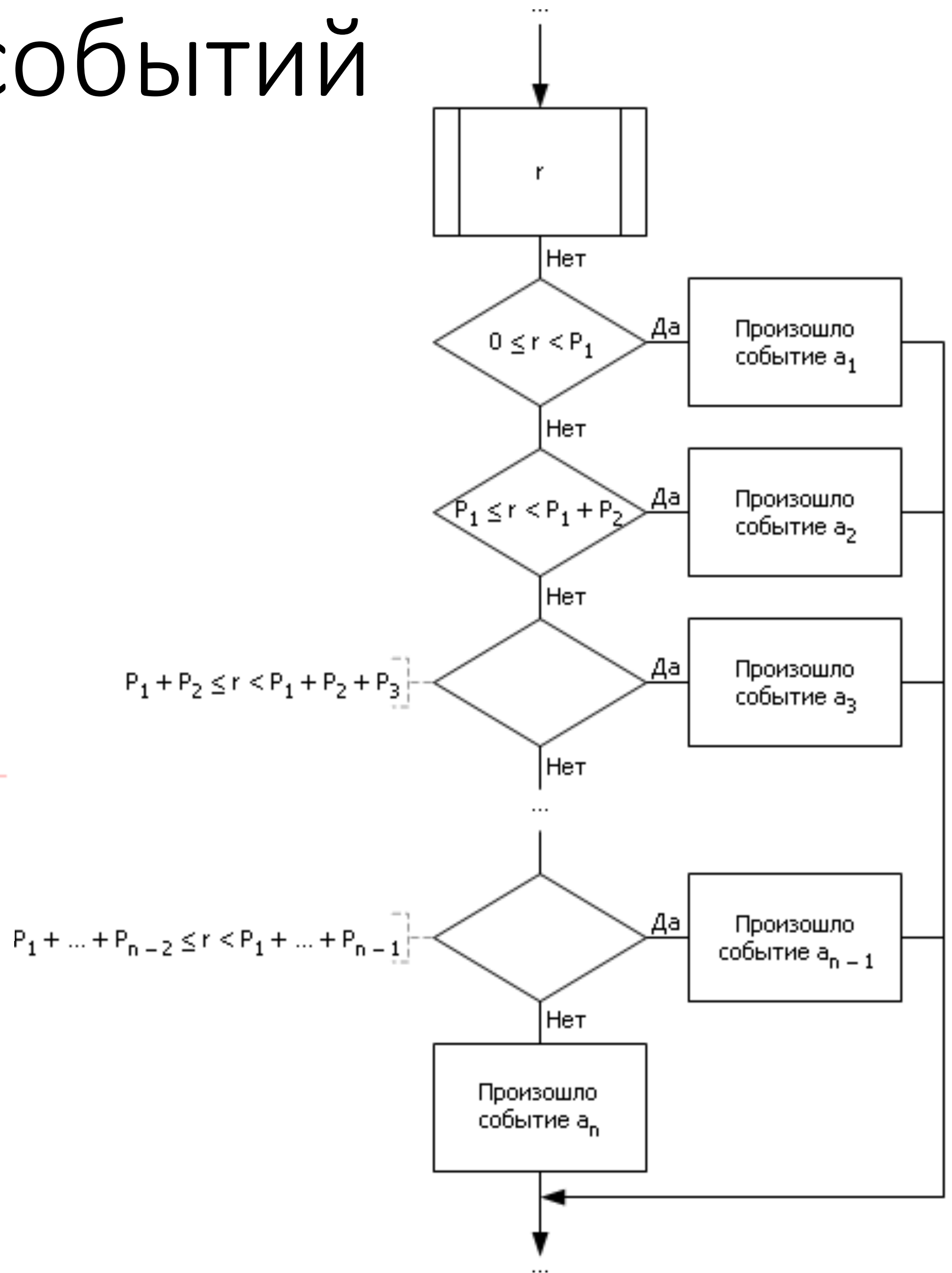




Моделирование полной группы несовместных событий

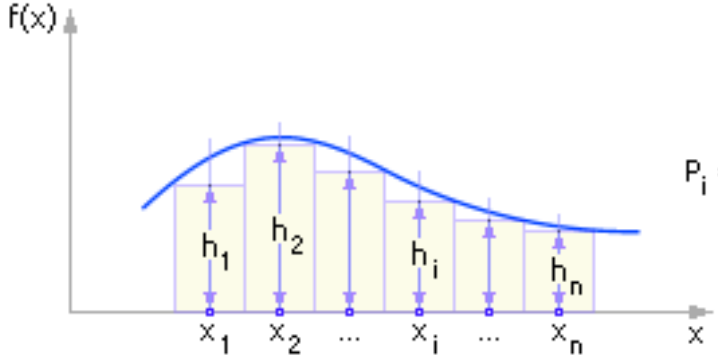






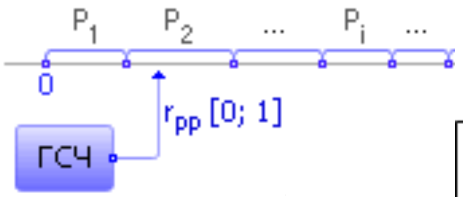
Моделирование случайных величин

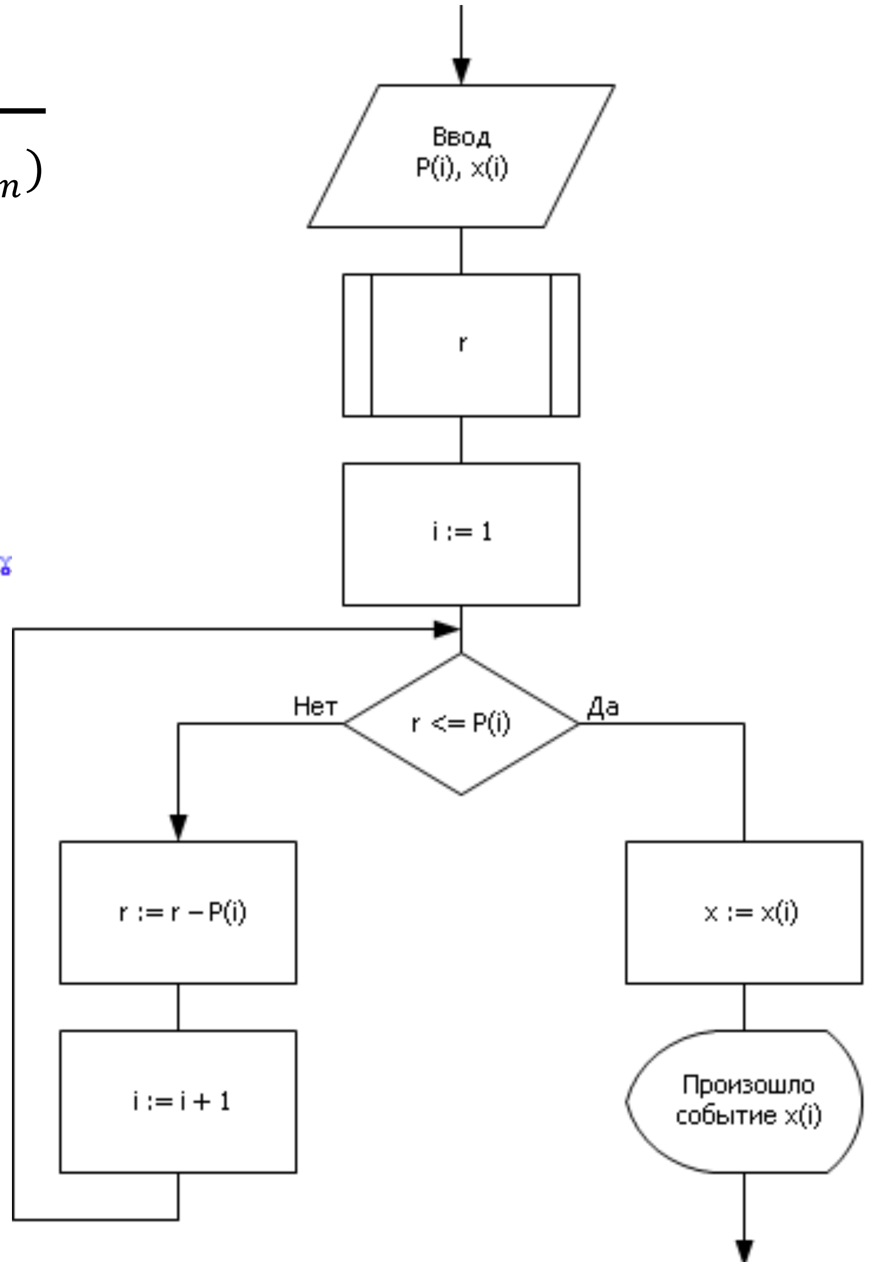
$$P_i = \frac{h_i}{(h_1 + h_2 + \dots + h_i + \dots + h_n)}$$



$$P_i = \frac{h_i}{h_1 + h_2 + \dots + h_i + \dots + h_n}$$

$$\sum_{i=1}^{n} P_i = 1$$





Моделирование нормального распределения

Получить последовательность *X* вида $Norm(m_X, \sigma_X)$

1. Генерация n случайных чисел r_i, образующих ряд *S*, где $m_S = \frac{n}{2}, \sigma_S = \sqrt{\frac{n}{12}}$
2. *z*-стандартизация: $z_i = \frac{s_i - m_S}{\sigma_S}$
3. Сдвиг и масштабирование до требуемого распределения: $x_i = z_i * \sigma_x + m_x$

Моделирование потоков случайных событий

$$P_m = \frac{a^m e^{-a}}{m!}$$

$$a = \int_{t_0}^{t_0+\Delta} \lambda(t)dt$$

$$\xrightarrow{\lambda(t) = const}$$

$$a = \lambda t$$

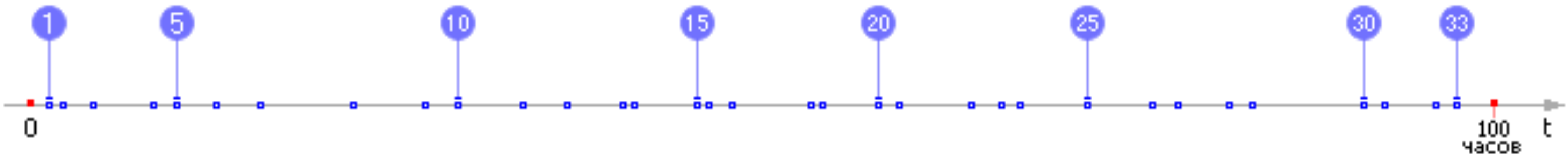
$$P_m = \frac{(\lambda\tau)^m e^{-\lambda\tau}}{m!}$$

$$P_0 = \frac{(\lambda\tau)^0 e^{-\lambda\tau}}{0!} = e^{-\lambda\tau}$$

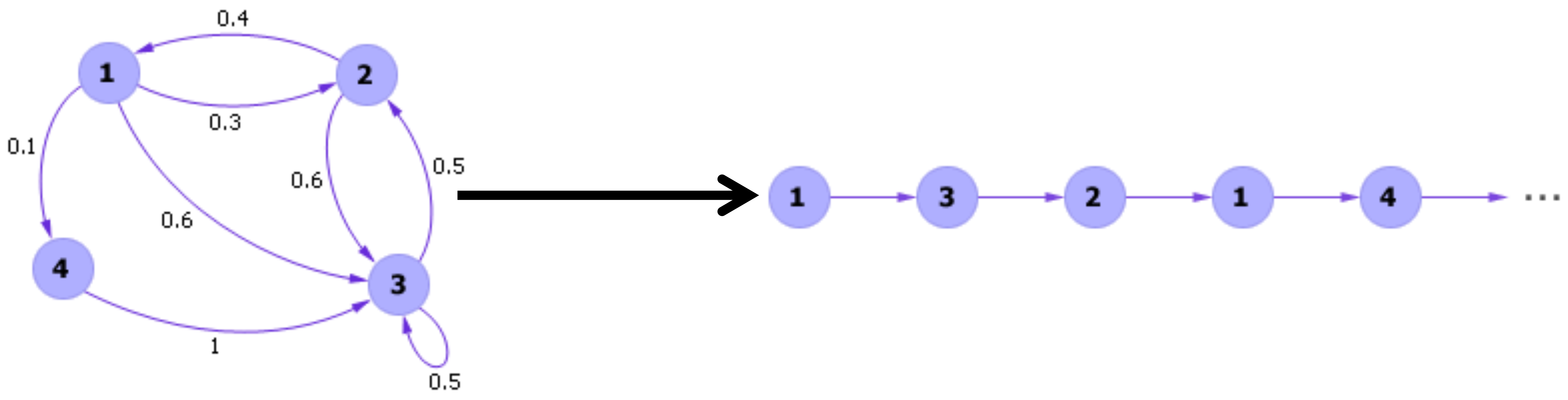
$$P_{m>0} = 1 - P_0 = 1 - e^{-\lambda\tau}$$

Алгоритм моделирования потока случайных событий

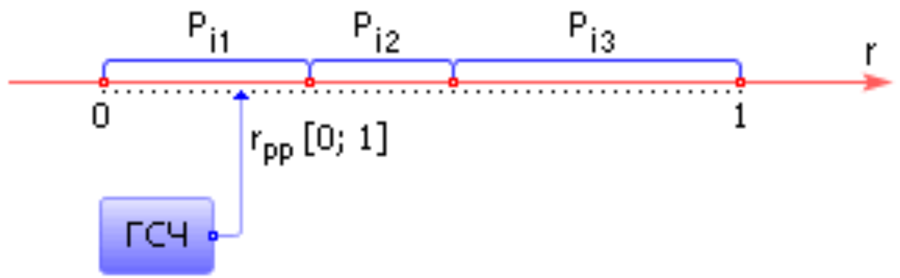
1. $t = 0, N = 0$
2. Получить r из ГСЧ
3. $\tau = -\frac{1}{\lambda}\ln(r)$
4. $t = t + \tau$
5. $N = N + 1$
6. $t \leq T?$
7. Да – возврат к шагу 2, нет – конец



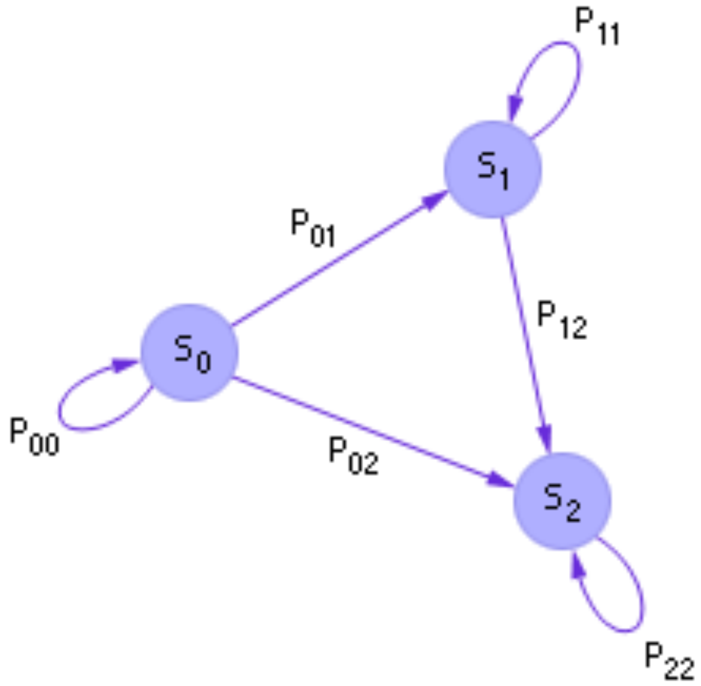
Моделирование марковских процессов



Интервалы $P_{i1}, P_{i2}, P_{i3}, \dots$ $(P_{i1} + P_{i2} + P_{i3} + \dots = 1)$



Пример моделирования марковского процесса



	S_0	S_1	S_2
S_0	0.45	0.4	0.15
S_1	0	0.45	0.55
S_2	0	0	1

Вектор начальных состояний

$P_0 = (1, 0, 0)$

```
import numpy as np
```

```
np.random.uniform(size=6)
```

```
array([0.26767933, 0.4905282 , 0.34289642, 0.78617414, 0.92882727,
       0.00942918])
```

Последовательность переходов:

1. $r = 0.27, S_0$
2. $r = 0.49, S_1$
3. $r = 0.34, S_1$
4. $r = 0.78, S_2$