Моделирование с использованием генераторов случайных чисел

Анализ сложности алгоритмов

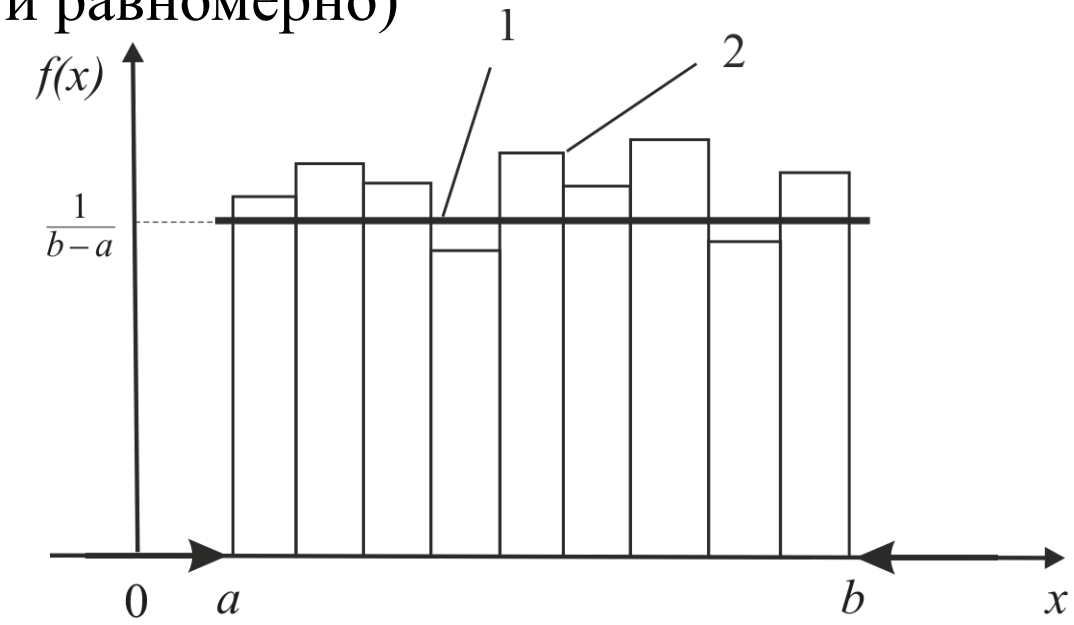
Логинов Сергей

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

Области применения:

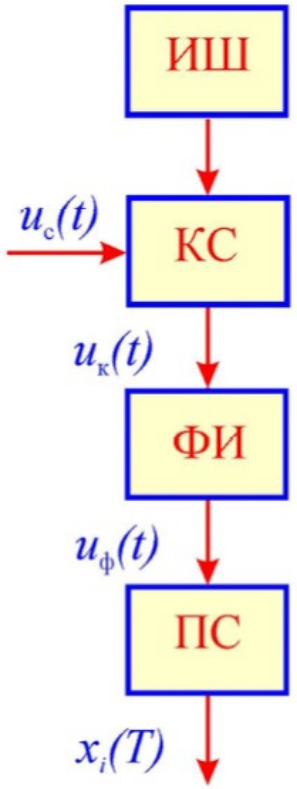
- Математическое и имитационное моделирование
- Математическая статистика
- Криптография
- Иные направления защиты информации
- Тестирование алгоритмов
- Сетевые протоколы

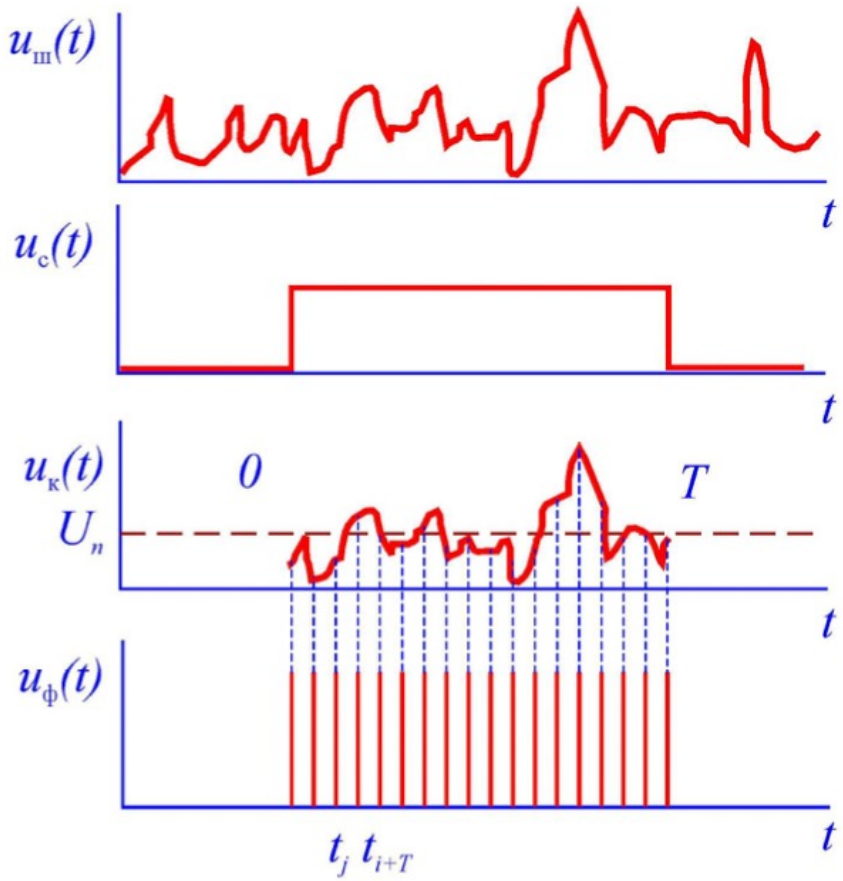
Генераторы случайных чисел

Характеристика	ГИСЧ	ГПСЧ
Отсутствие периодичности	Да	Нет
Непредсказуемость	Да	Условная
Независимость значений	Да	Условная
Уровень криптостойкости	Высокий	Условный
Скорость генерации	Низкая	Высокая
Воспроизводимость	Нет	Да
Простота генерации	Нет	Да
Стоимость генерации	Высокая	Низкая

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
- Аддитивные генераторы (последовательность Фибоначчи)
- Генераторы на базе клеточного автомата
- Генераторы, основанные на нечеткой логике
- Генераторы, основанные на обратной функции

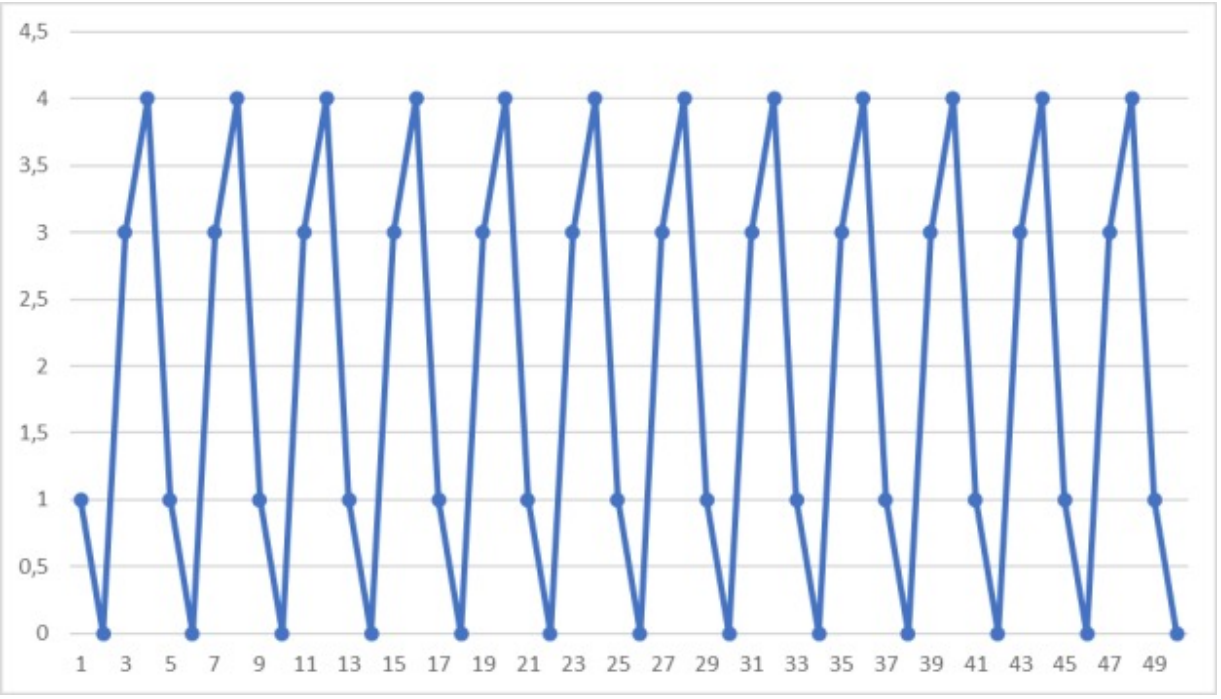
Свойства псевдослучайных последовательностей:

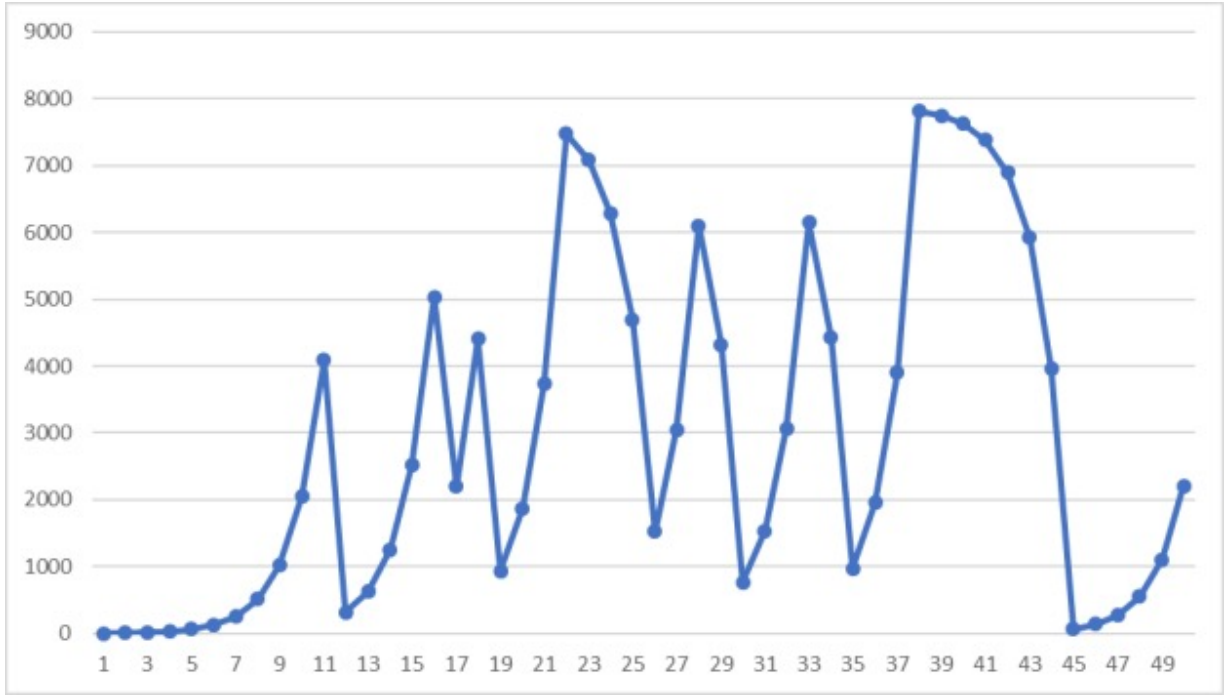
- Непредсказуемость
- Неотличимость статистических свойств от истинно-случайных последовательностей
- Большой период
- Возможность эффективной программной реализации

Линейный конгруэнтный метод (ЛКМ)

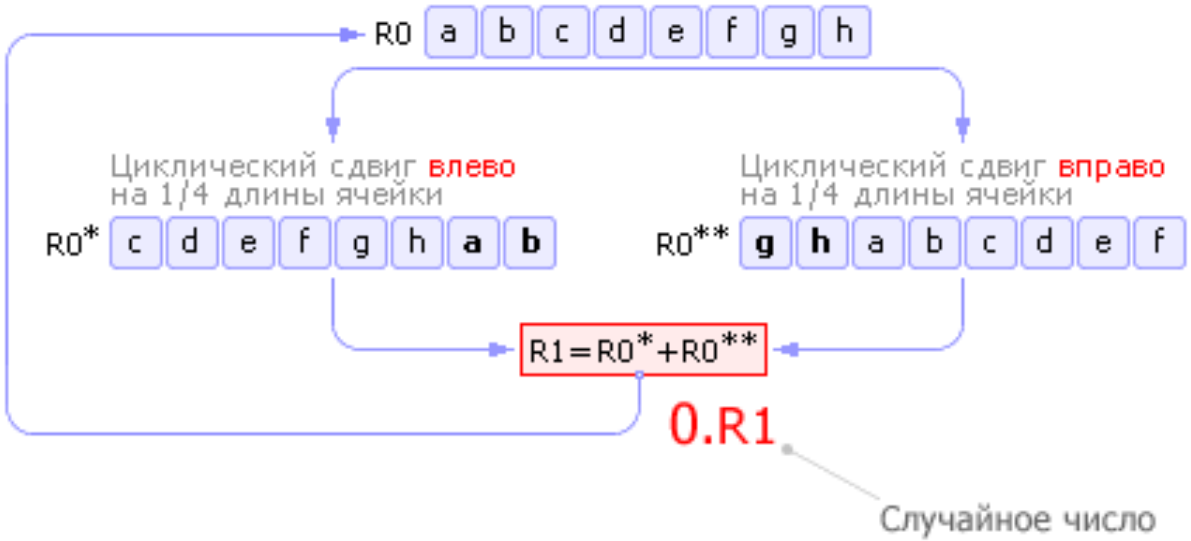
$$\boldsymbol{X_{n+1} = (aX_n + c)\ mod\ m,}$$

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





Алгоритм перемешивания



$R = 8$ bit

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

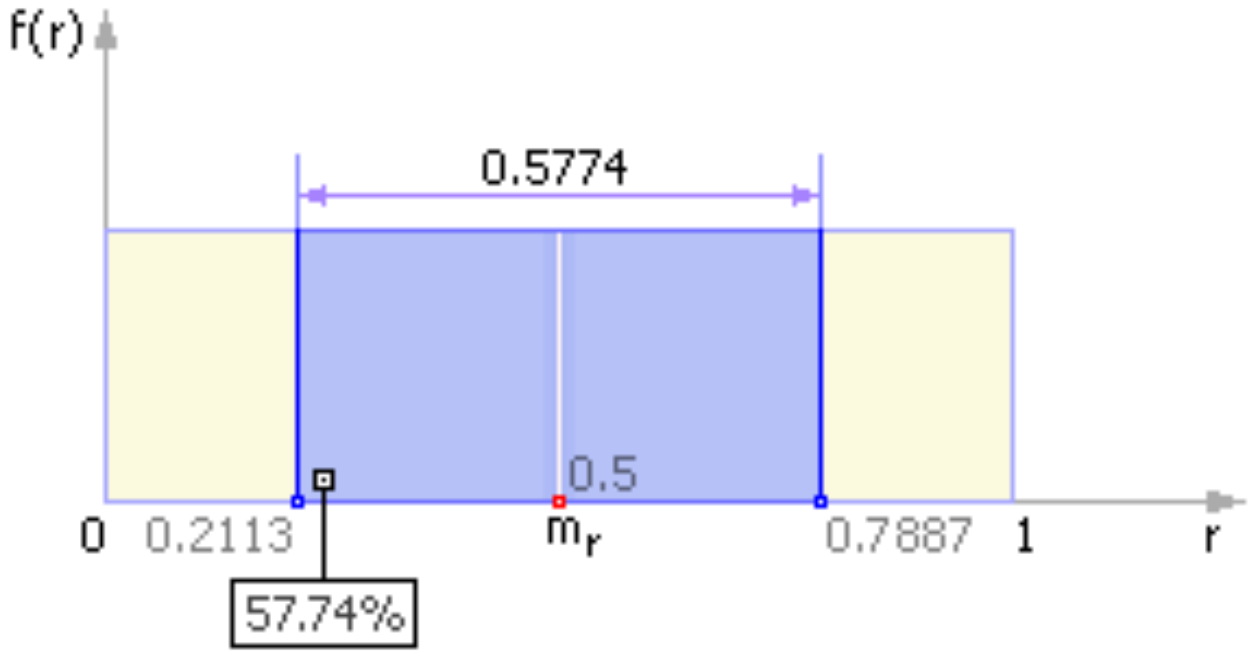
$R_0^{**} = 10100001_2 = 161_{10}$

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

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. $\boldsymbol{\chi^2_{эксп}}$

Пакет статистических тестов NIST STS

1. Частотный тест
2. Частотный тест внутри блока
3. Проверка накопленных сумм
4. Проверка серий
5. Проверка максимальной длины серии в блоке
6. Проверка ранга двоичной матрицы
7. Спектральный тест на основе дискретного преобразования Фурье
8. Проверка перекрывающихся шаблонов
9. Универсальный тест Маурера
10. Энтропийный тест
11. Проверка случайных отклонений
12. Проверка случайных отклонений (вариантный)
13. Тест на подпоследовательности
14. Проверка неперекрывающихся шаблонов
15. Проверка линейной сложности

Выводы о прохождении теста

$$\left[(1-\ \alpha)-3\sqrt{\frac{\alpha(1-\alpha)}{m}},\qquad (1-\alpha)+3\sqrt{\frac{\alpha(1-\alpha)}{m}}\right], m-\text{объем выборки}$$

$$\chi^2=\frac{\sum_{i=1}^{k}\left(v_i-{}^{m}/_{k}\right)^2}{{}^{m}/_{k}}, v_i-\text{количество } p\ value \text{ на } i-\text{ом интервале}$$

Запуск тестового скрипта на тестовых данных

```
(base) lalogin@MacBook-Air-Sergej sts-2.1.2 % ./assess 100000
          G E N E R A T O R    S E L E C T I O N
          ----------------------------------

    [0] Input File                [1] Linear Congruential
    [2] Quadratic Congruential I  [3] Quadratic Congruential II
    [4] Cubic Congruential        [5] XOR
    [6] Modular Exponentiation    [7] Blum-Blum-Shub
    [8] Micali-Schnorr            [9] G Using SHA-1

   Enter Choice: 0


             User Prescribed Input File: data/data.pi

             S T A T I S T I C A L   T E S T S
             ---------------------------------

    [01] Frequency                     [02] Block Frequency
    [03] Cumulative Sums               [04] Runs
    [05] Longest Run of Ones           [06] Rank
    [07] Discrete Fourier Transform    [08] Nonperiodic Template Matchings
    [09] Overlapping Template Matchings [10] Universal Statistical
    [11] Approximate Entropy           [12] Random Excursions
    [13] Random Excursions Variant     [14] Serial
    [15] Linear Complexity

         INSTRUCTIONS
            Enter 0 if you DO NOT want to apply all of the
            statistical tests to each sequence and 1 if you DO.

   Enter Choice: 1
```

```
          P a r a m e t e r   A d j u s t m e n t s
          -----------------------------------------

    [1] Block Frequency Test - block length(M):         128
    [2] NonOverlapping Template Test - block length(m): 9
    [3] Overlapping Template Test - block length(m):    9
    [4] Approximate Entropy Test - block length(m):     10
    [5] Serial Test - block length(m):                  16
    [6] Linear Complexity Test - block length(M):       500

   Select Test (0 to continue): 0

   How many bitstreams? 10

   Input File Format:
    [0] ASCII - A sequence of ASCII 0's and 1's
    [1] Binary - Each byte in data file contains 8 bits of data

   Select input mode:  0

     Statistical Testing In Progress.........

     Statistical Testing Complete!!!!!!!!!!!!

(base) lalogin@MacBook-Air-Sergej sts-2.1.2 %
```

Начальные данные и результаты

data.pi

```
10100000110000011101011011
00001101111011001011000000
00110011010011111111000011
11010101010000000011011000
11110011111011010101111001
10101110010010000001001100
01110000101000100011110011
10101011001111010011110000
00101111001011101010101100
11110000101001110111011100
11001000011010111110001100
00001100010001011001010010
11100111001010101101011100
01010101110110010111100111
10000100011011100110010011
00010110001111010101111000
01100110111101111110000110
01010101000000001000110110
10011011100101100101010111
01010111011010100000100000
00011010000101101000110011
00011110000101010101101000
10110011001100011001111011
11010011101010111000000000
10000110111000111101111000
11011010101000000010110001
10011000011101011111101000
00010011001100101011111011
11100110010111111001100011
00011110101111110110101000
```

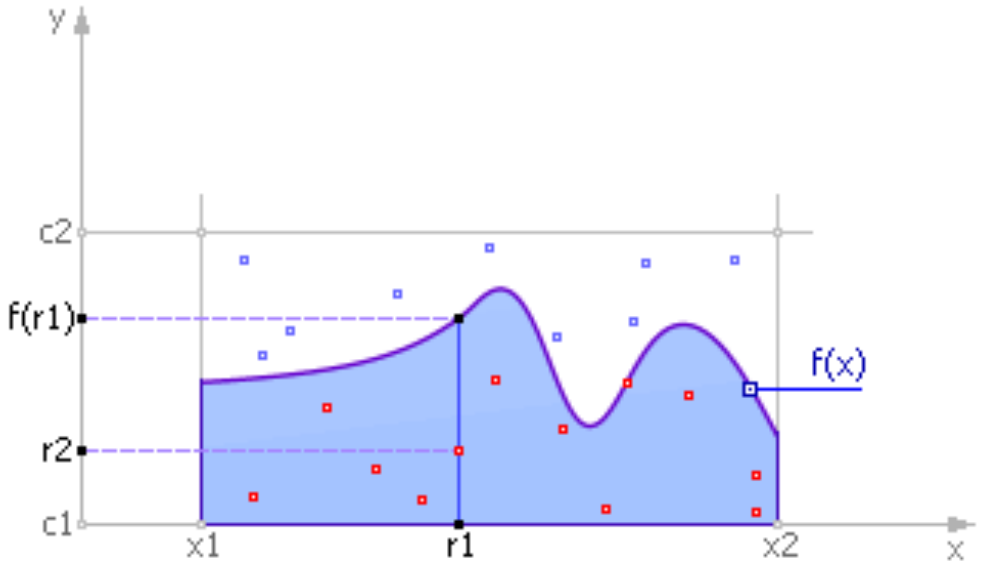
```
------------------------------------------------------------------------------
RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES
------------------------------------------------------------------------------
   generator is <data/data.pi>
------------------------------------------------------------------------------
```

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
1	1	3	0	0	2	1	0	1	1	0.534146	10/10	Frequency
1	2	1	0	2	2	1	0	1	0	0.739918	10/10	BlockFrequency
1	1	1	2	1	0	0	2	1	1	0.911413	10/10	CumulativeSums
1	2	0	1	1	1	1	2	1	0	0.911413	10/10	CumulativeSums
0	4	1	1	0	2	0	1	0	1	0.122325	10/10	Runs
0	1	0	4	1	0	1	1	1	1	0.213309	10/10	LongestRun
1	1	0	1	1	1	2	1	0	2	0.911413	10/10	Rank
2	1	0	0	2	1	1	1	2	0	0.739918	10/10	FFT
1	2	1	0	2	2	1	1	0	0	0.739918	10/10	NonOverlappingTemplate
1	3	0	0	3	3	0	0	0	0	0.035174	10/10	NonOverlappingTemplate
3	1	1	0	0	1	0	3	0	1	0.213309	10/10	NonOverlappingTemplate
2	0	1	2	1	0	1	2	1	0	0.739918	10/10	NonOverlappingTemplate
0	2	1	1	0	2	1	1	2	0	0.739918	10/10	NonOverlappingTemplate
1	1	1	1	0	1	0	2	2	1	0.911413	10/10	NonOverlappingTemplate
0	2	1	1	1	2	1	0	1	1	0.911413	10/10	NonOverlappingTemplate
2	1	1	1	1	0	1	3	0	0	0.534146	10/10	NonOverlappingTemplate
1	0	1	1	1	2	1	0	2	1	0.911413	9/10	NonOverlappingTemplate
0	0	2	1	0	1	1	4	1	0	0.122325	10/10	NonOverlappingTemplate
0	2	0	2	0	1	2	1	1	1	0.739918	10/10	NonOverlappingTemplate
2	1	1	1	1	1	0	0	2	1	0.911413	10/10	NonOverlappingTemplate
1	1	3	1	0	1	1	1	1	0	0.739918	10/10	NonOverlappingTemplate
4	0	1	0	0	1	1	1	0	2	0.122325	9/10	NonOverlappingTemplate
1	2	1	0	1	2	1	0	0	2	0.739918	10/10	NonOverlappingTemplate
3	0	1	1	1	1	0	2	1	0	0.534146	10/10	NonOverlappingTemplate
2	1	0	0	2	1	1	2	0	1	0.739918	9/10	NonOverlappingTemplate
0	0	1	3	1	2	0	1	1	1	0.534146	10/10	NonOverlappingTemplate

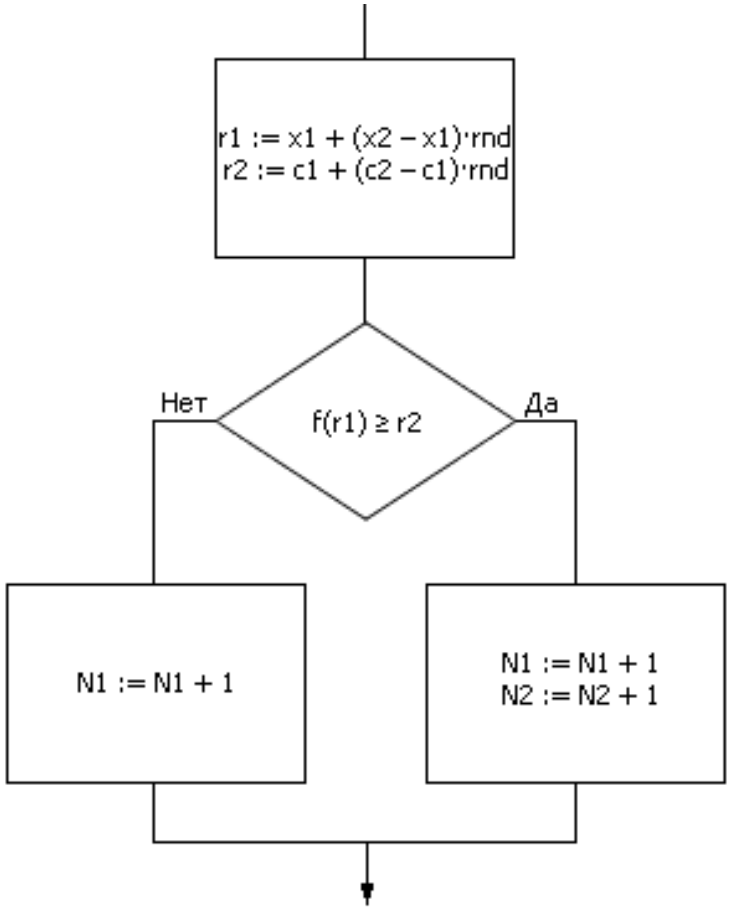
ГСЧ в моделировании

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

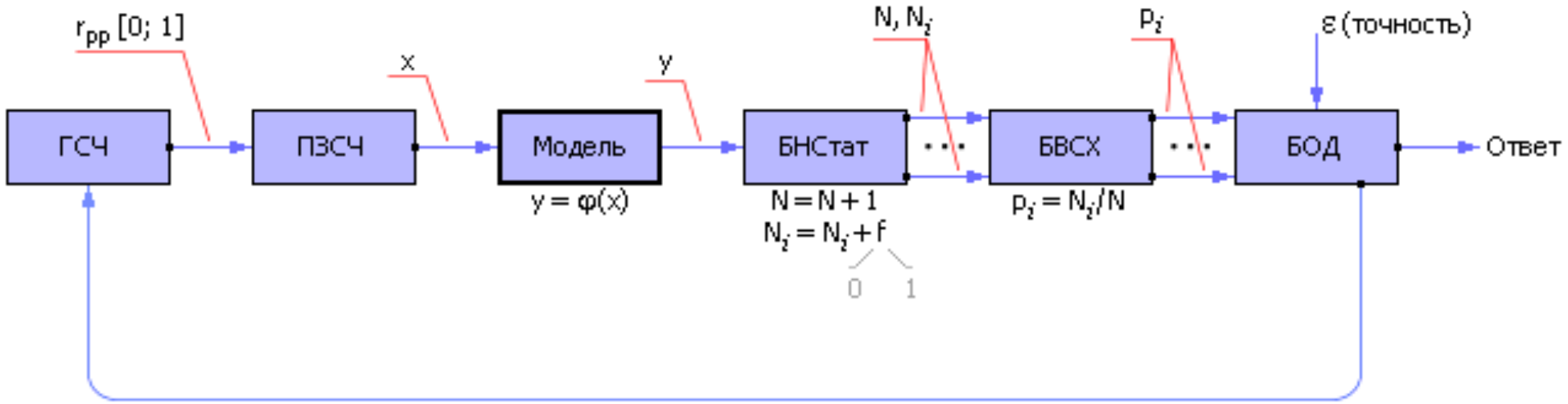
ГСЧ в методе Монте-Карло



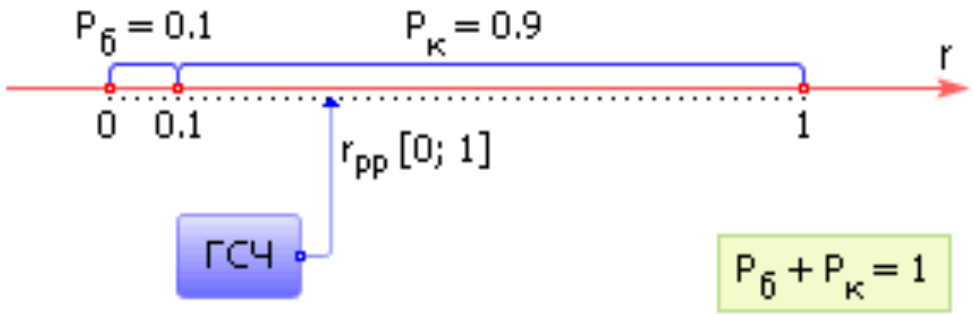
$$\frac{N_2}{N_1} = \frac{y}{(x_2 - x_1)(c_2 - c_1)}$$

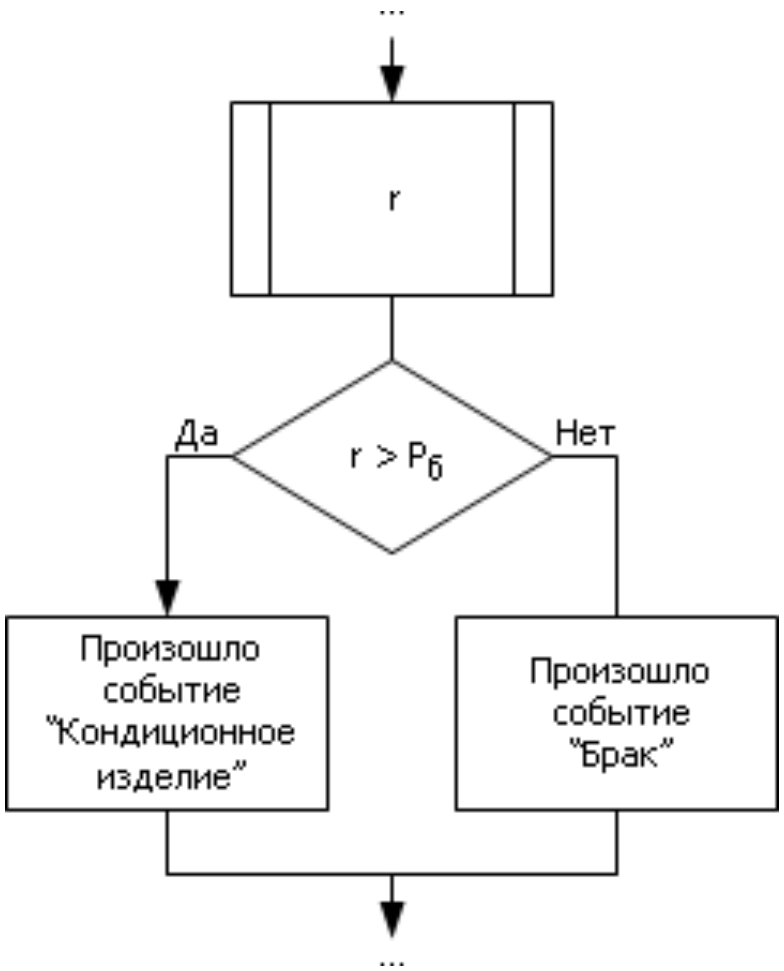


Использование метода Монте-Карло для исследования систем со случайными параметрами

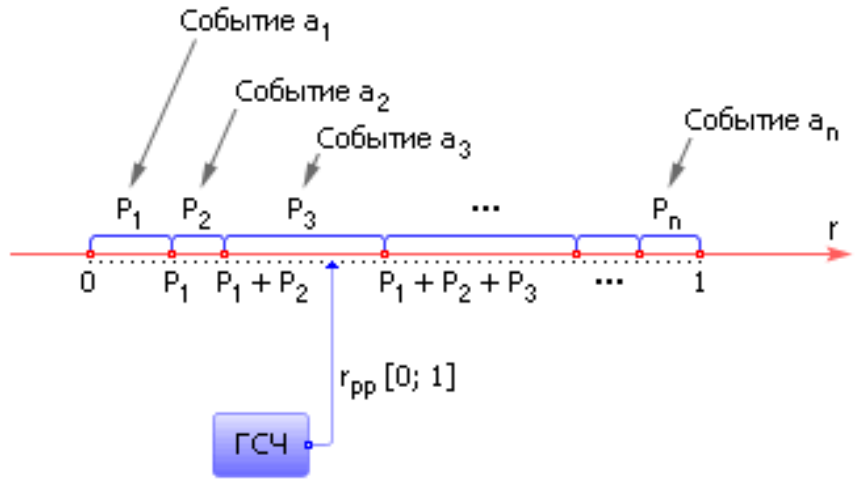


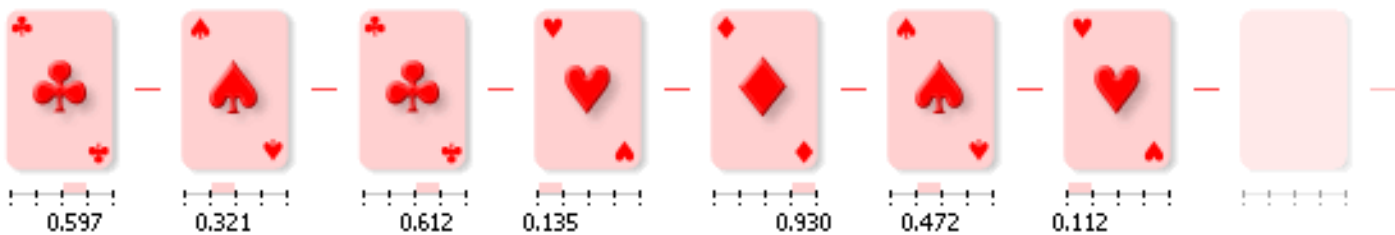
Имитация случайных событий

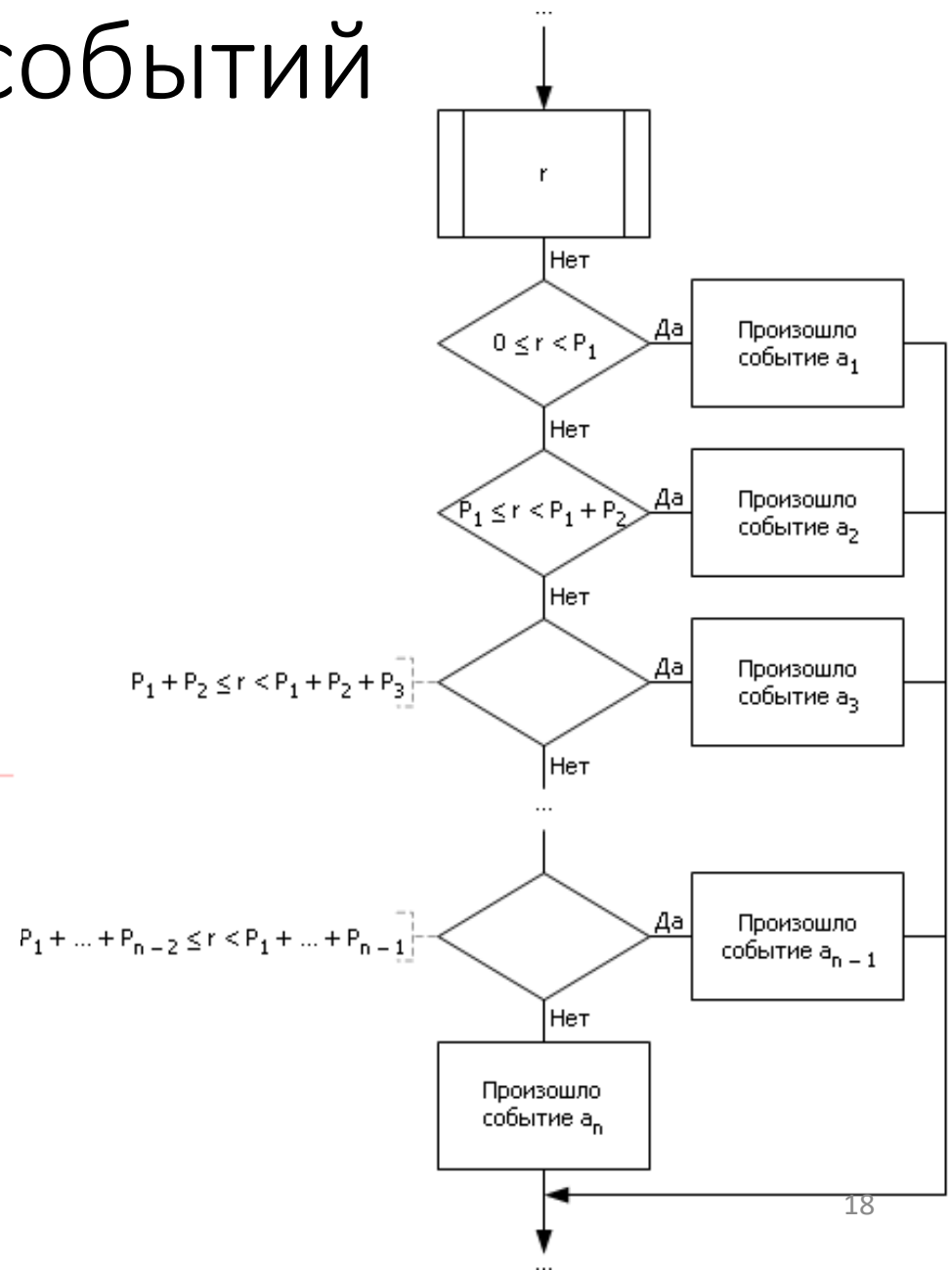




Моделирование полной группы несовместных событий

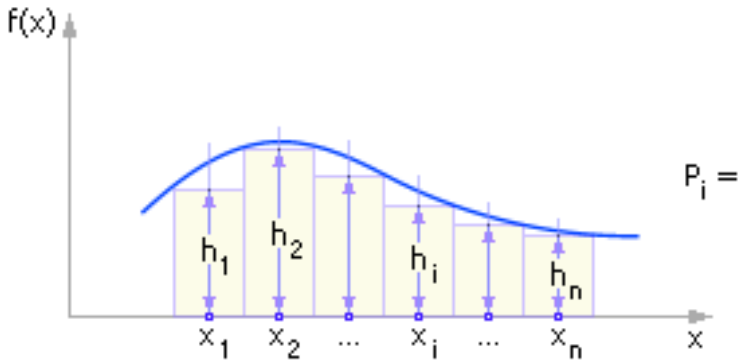






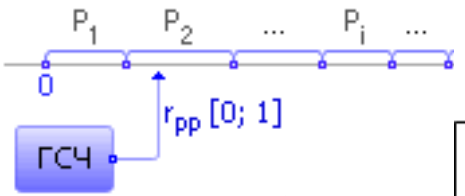
Моделирование случайных величин

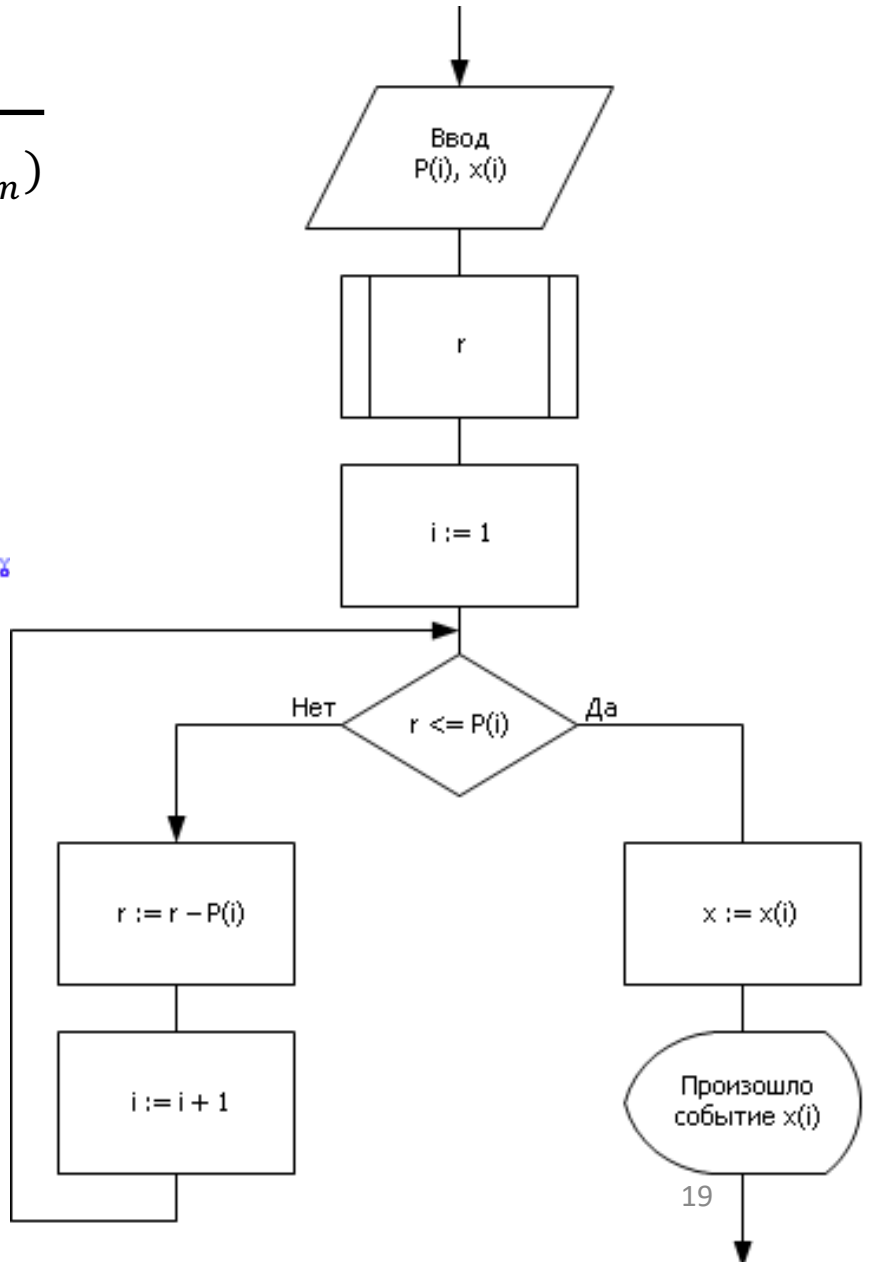
$$P_i = \frac{h_i}{(h_1 + h_2 + \ldots + h_i + \ldots + h_n)}$$



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

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





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

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

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

$$\lambda(t) = const$$

$$a = \lambda t$$

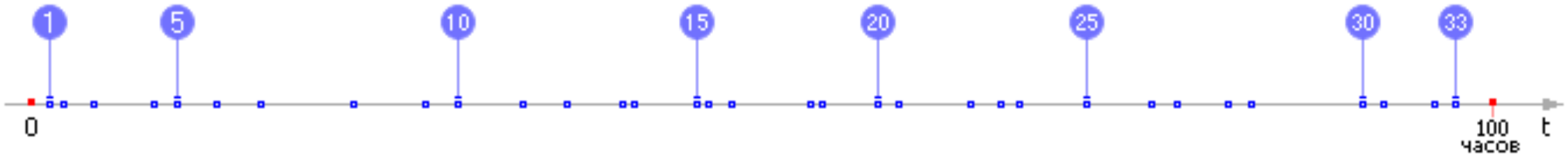
$$P_m = \frac{(\lambda\tau)^m e^{-\lambda\tau}}{m!}$$

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

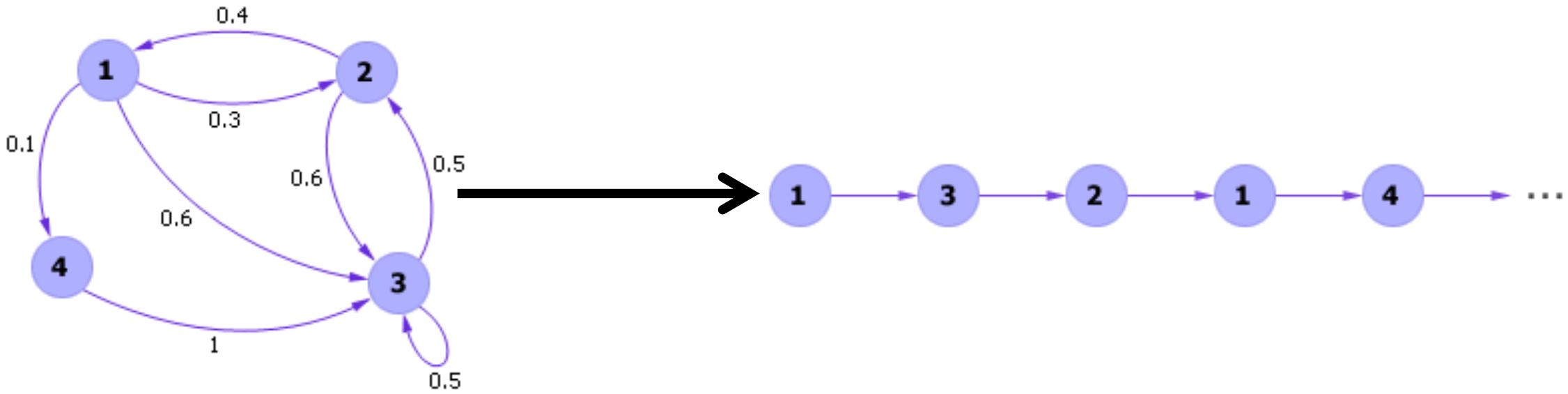
$$P_{m>0} = 1 - P_0 = 1 - e^{-\lambda\tau}$$

Алгоритм моделирования потока случайных событий

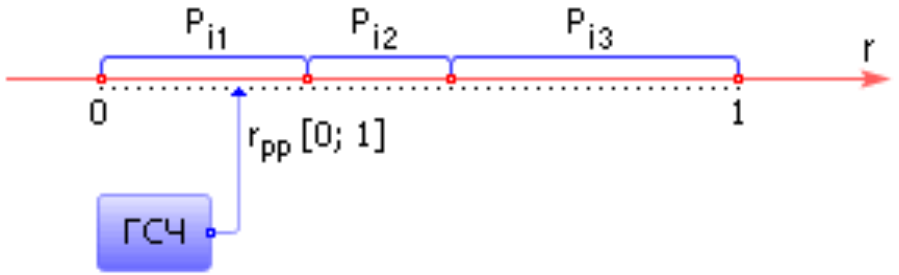
1. $t = 0, N = 0$
2. Получить r из ГСЧ
3. $\tau = -\frac{1}{\lambda}\ln(r)$
4. $t = t + \tau$
5. $N = N + 1$
6. $t \leq T$?
7. Да – возврат к шагу 2, нет – конец



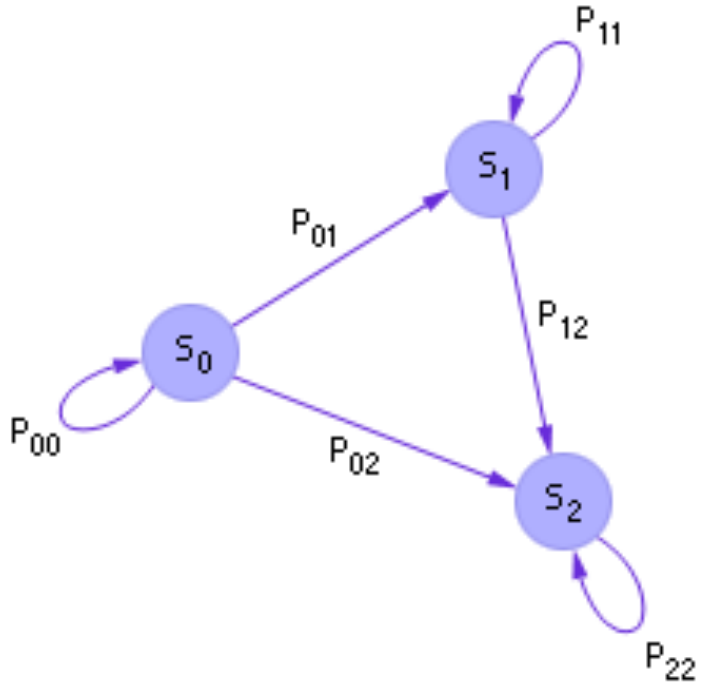
Моделирование марковских процессов



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



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