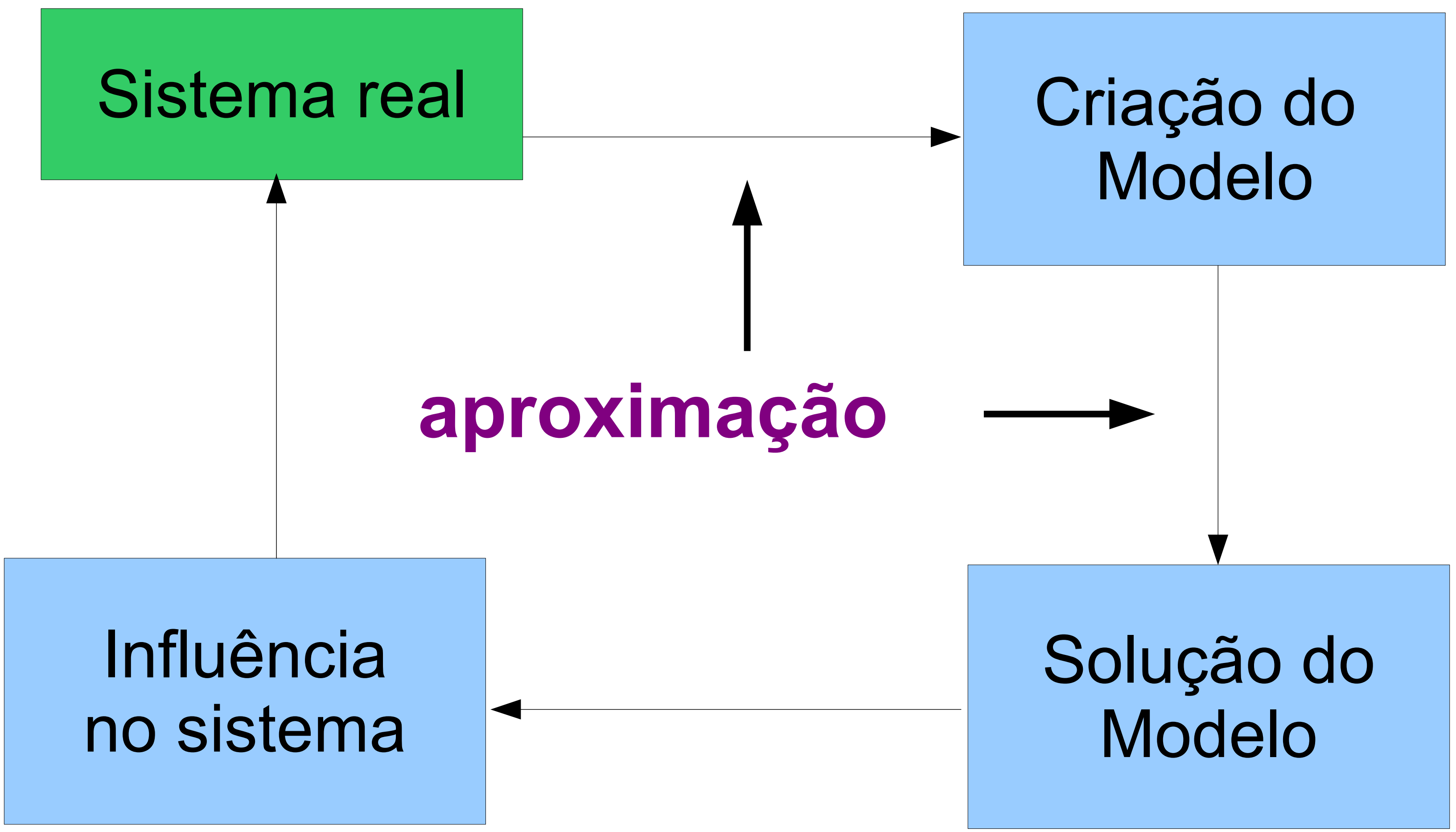
Avaliação e Desempenho
Aula 1 - Simulação

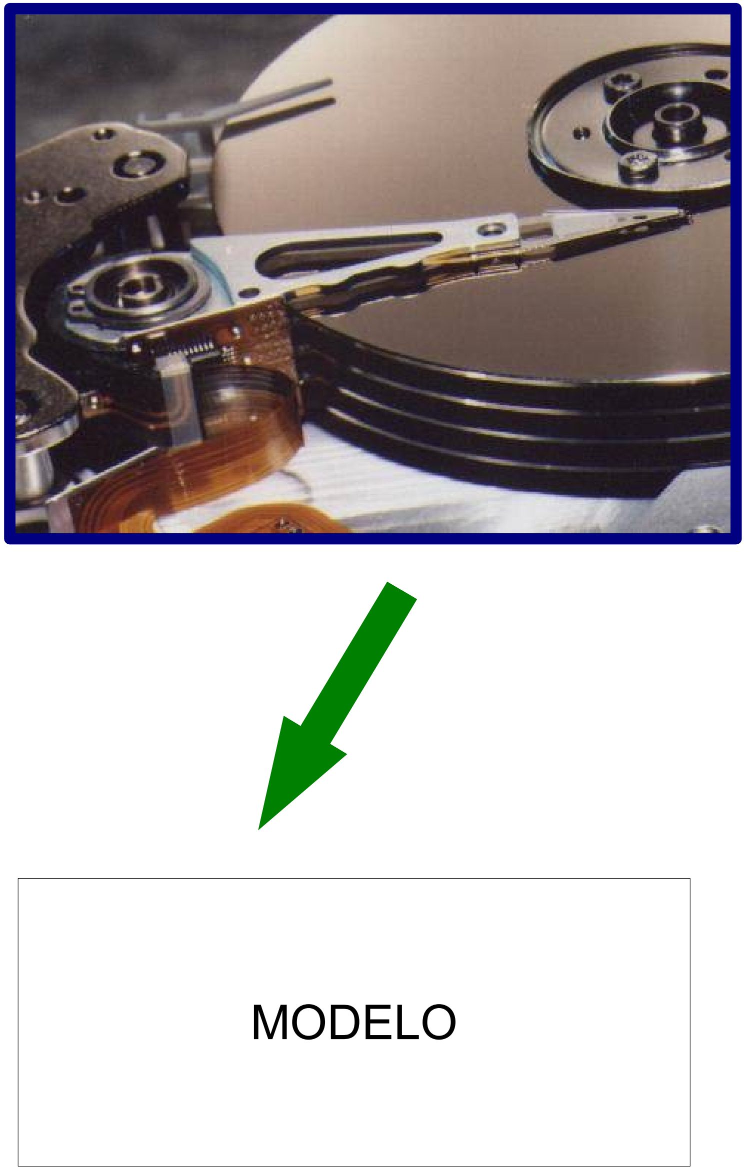
- Introdução à simulação
- Geração de números aleatórios
- Lei dos grandes números
- Geração de variáveis aleatórias

O Ciclo de Modelagem



Modelando um Sistema

- Abstração do sistema
- Simplificação necessária
- Representação matemática do sistema
- com aleatoriedade
- avaliação precisa do desempenho

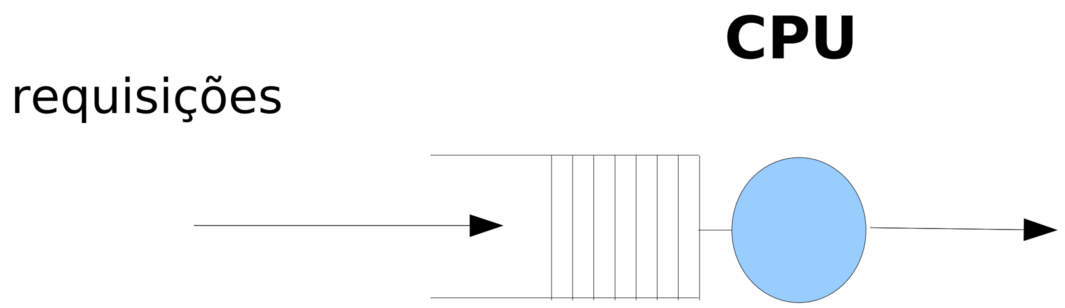


Cadeias de Markov

- Técnica de modelagem
  - abstração matemática do sistema
- muito utilizada
- **Propriedade de Markov**
  - estados futuros não depende do passado!
  - prob. de transição para um estado depende apenas do estado atual

Andrey Markov (1856-1922)

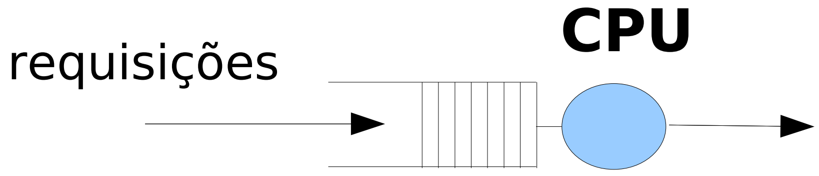
Fila Genérica



- Processo de chegada não é Poisson
- Tempo de serviço não é exponencial
- Como avaliar o desempenho desta fila?
  - ex. qual o tamanho médio da fila?

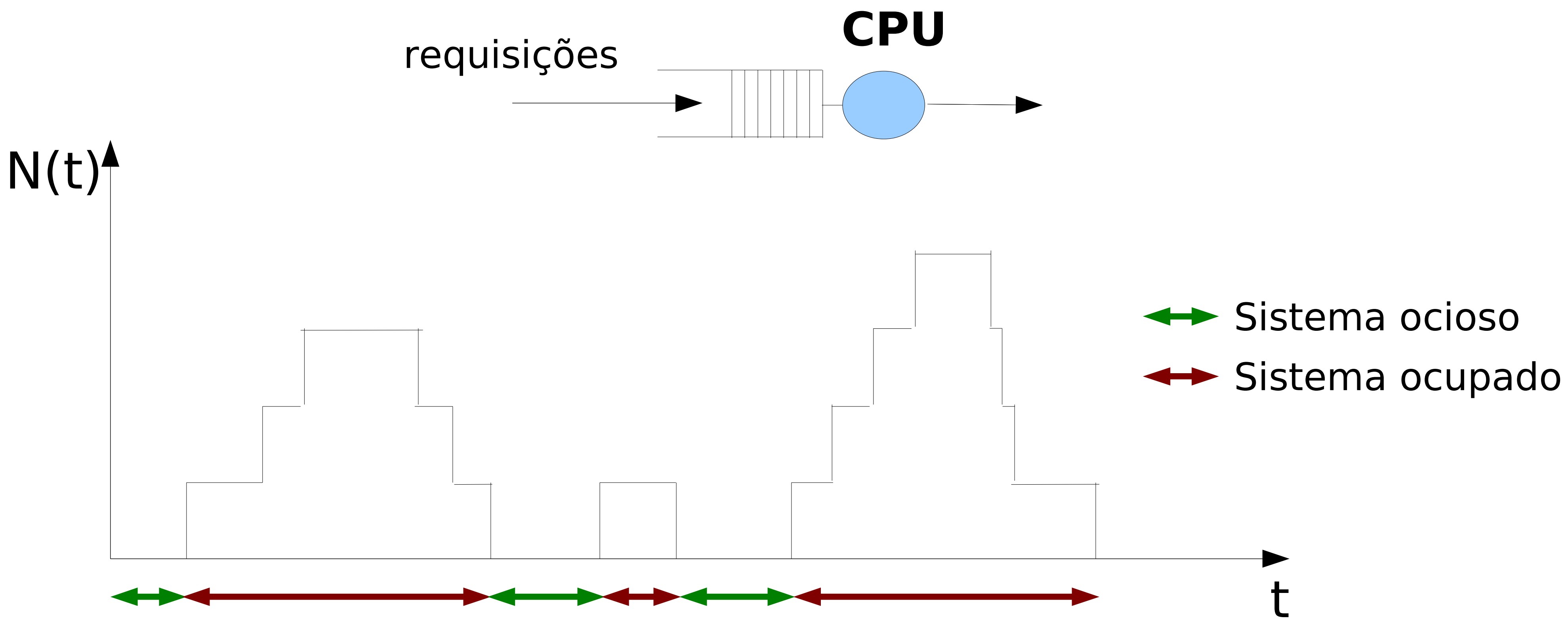
Simulação

Simulando uma Fila Genérica



- O que é uma simulação?
  - programa que **imita** comportamento do sistema
- Como simular este sistema?
  - programa deve gerar eventos de chegadas e serviço, manter estado da fila, calcular medidas de interesse

Simulando uma Fila Genérica



- Imitar evolução do sistema: **N(t)**
- Obter medidas de interesse
  - Utilização, **E[N]**, etc.

Simulação

Vantagens

- Pode lidar com modelos mais complexos
- Melhor representação do sistema real

Desvantagens

- Difícil interpretar e verificar resultados
- Longo tempo de execução

- Eterno debate entre simulação e avaliação matemática

Caracterizando um Simulador

- Determinístico ou Estocástico
  - Modelo contém eventos aleatórios?
- Estático ou Dinâmico
  - Evolução do tempo influi no sistema?
- Contínuo ou Discreto
  - Estado do sistema evolui continuamente ou em pontos discretos no tempo?

➔ **Simulação Discreta de Eventos**

- estocástica, dinâmica e discreta

Primeiro Passo para Simulação

- Como gerar números aleatórios?
  - seu computador possui gerador de números aleatórios?
- Computador é fundamentalmente determinístico
- Gerar números ***pseudo-aleatórios***
  - que *pareçam* aleatórios
  - algoritmos determinísticos

Gerador de Números Pseudo-Aleatórios

- Algoritmo que gera uma sequência de números inteiros U_1, U_2, ..., que pareçam ser:
  - *Uniformemente distribuídos* no intervalo [0, M-1] (para algum M fixo)
  - Estatisticamente *independentes*
- "pareçam ser": sequência deve ter propriedades *relevantes* de uma sequência de v.a. uniforme

Método Congruente

- Dois parâmetros *a* e *M* (números inteiros)
- Dado x_0 (semente)

$$x_i = a\,x_{i-1} \bmod M$$

$$U_i = \frac{x_i}{M}$$

- x_0 determina a sequência inteira
- *M* determina quantos diferentes números podem ser gerados

Exemplo

- Supor *a=3* e *M=11*

$$x_i = 3x_{i-1} \bmod 11$$

- Quantos números diferentes podemos gerar?

- Supor $x_0 = 4$ (semente)

$$x_1 = 3x_0 \bmod 11 \qquad x_2 = 3x_1 \bmod 11$$

$$= 12 \bmod 11 \qquad = 3 \bmod 11$$

$$= 1 \qquad = 3$$

Exemplo

- Supor *a=3* e *M=11*

$$x_i = 3x_{i-1} \bmod 11$$

- Gerador entra em loop depois de 5 amostras

- E se x_0 for diferente?

$$x_0 = 4$$
$$x_1 = 1$$
$$x_2 = 3$$
$$x_3 = 9$$
$$x_4 = 5$$
$$x_5 = 4$$
$$x_6 = 1$$
$$\vdots$$

Propriedades do Gerador

- Todos números são gerados?
- Qual o tamanho de uma sequência?
- Escolher *a* e *M* tal que para qualquer x_0
  - Sequência pareça aleatória
  - Longo ciclo antes de repetição
  - Cálculo seja eficiente
- Sugestão (para máquina de 32 bits)

$$a = 7^5$$

Método Congruente Linear

- Três parâmetros *a*, *c* e *M* (números inteiros)
- Dado x_0 (semente)

$$x_i = (a x_{i-1} + c) \bmod M$$

$$U_i = \frac{x_i}{M}$$

diferença do outro método

- x_0 determina a sequência inteira
- Bastante utilizado

Exemplo

- Supor *a = 3, M = 11, c = 2*
- Semente $x_o = 4$

$$x_i = (3x_{i-1} + 2) \bmod 11$$

$$x_1 = (3x_0 + 2) \bmod 11$$
$$= 14 \bmod 11$$
$$= 3$$

$$x_0 = 4$$
$$x_1 = 3$$
$$x_2 = 0$$
$$x_3 = 2$$
$$x_4 = 8$$
$$x_5 = 4$$
$$x_6 = 3$$
$$\vdots$$

Geradores na Prática

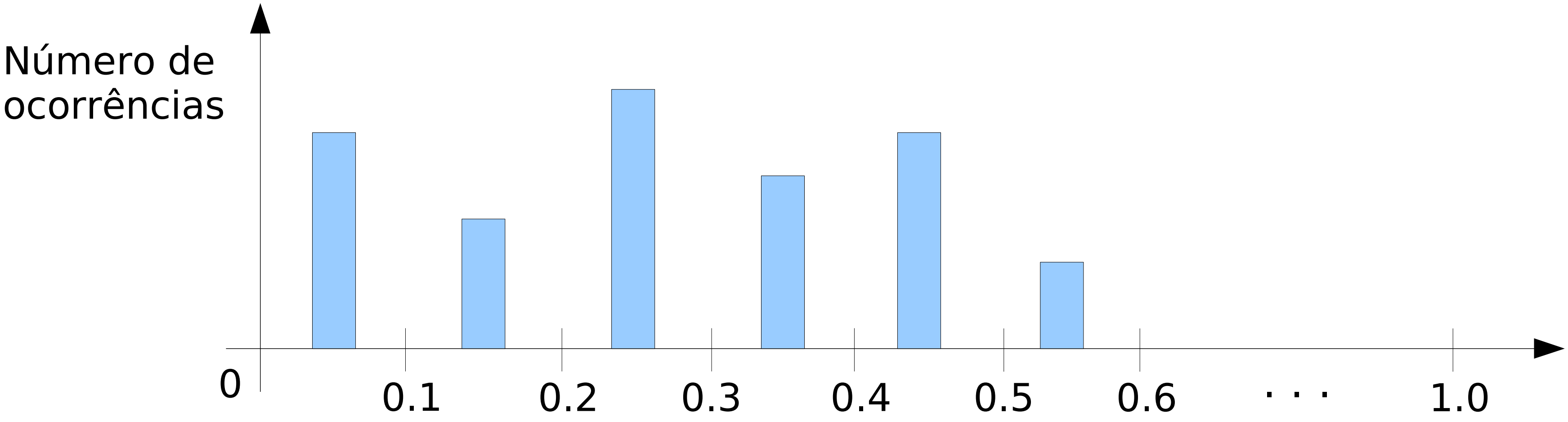
- Linguagens de programação oferecem geradores
- C ou C++ (gnu)
  - *lrand48()* - retorna *long* $[0, 2^{31}]$
  - *drand48()* - retorna *float* [0, 1)
  - método Congruente Linear
  - usuário define semente (x_o)

$$M = 2^{48} \quad a = 25214903917 \quad c = 11$$

- *Matlab* utiliza múltiplos geradores
  - usuário pode escolher

Tarefa de Casa

- Fazer um programa para gerar números pseudo-aleatórios
- Escolher parâmetros método congruente linear
- Gerar 1000 números entre [0, 1] (dividir por *M*)
- Gerar gráfico com histograma



- Repetir usando gerador da linguagem
- comparar resultados (aumentar amostras)

Lei dos Grandes Números

- Você já deve conhecer (intuitivamente)

- Experimento: jogar *n* vezes dado com seis faces
- D : resultado de uma jogada
- $N_1(n)$: número de vezes que resultado é 1
- $F_1(n)$: *fração* de vezes que o resultado é 1 →

$$F_1(n) = \frac{N_1(n)}{n}$$

Lei dos Grandes Números

$$F_1(n) = \frac{N_1(n)}{n}$$

- Realizar experimento (jogar o dado)
  - quanto vale $F_1(10)$?
  - e $F_1(100)$?
  - e $F_1(1000)$?
- $F_1(n)$ *converge* para $P[D = 1]$

prob. do resultado ser igual a 1

  - quando *n* tende ao infinito!
- $P[D = 1] = 1/6$

Lei dos Grandes Números

- Frequência relativa do resultado de um experimento aleatório *converge* para sua probabilidade
- Resultado fundamental em probabilidade e estatística
- Atribui significado físico a um conceito abstrato (probabilidade)
  - números (quando muitos) convergem

Lei dos Grandes Números

- Generalização: média de uma sequência de v.a. (iid) *converge* para valor esperado

Variável aleatória: X

Valor esperado: $E[X]$

Sequência (iid): $X_1, X_2, \ldots, X_n$ $\longrightarrow$ $E[X] = E[X_i]$ para todo *i*

Média amostral: $\overline{X_n} = \dfrac{\sum_{i=1}^{n} X_i}{n}$

Convergência: $\overline{X_n} \rightarrow E[X]$ quando *n* tende ao infinito

Exemplo

- Experimento: jogar um dado
- *X* v.a. que indica o resultado
- X_i v.a. que indica o resultado do *i*-ésimo experimento
- Quanto vale *E[X]*? → $E[X]=3.5$

3, 1, 2, 5, 6, 3, 2, 1, 4, 4, 2, 3, 5, 6, 1, 2, 2, 5, 3, 1.

Sequência de resultados de experimentos repetidos

Média: $\overline{X}_n = \frac{\sum_{i=1}^{n} X_i}{n} = \frac{61}{20} = 3.05$

Lei dos Grandes Números

- Seja X_1, X_2, ..., X_n uma sequencia de v.a. iid com valor esperado $\mu < \infty$

- Seja $\overline{X_n} = \dfrac{\sum_{i=1}^{n} X_i}{n}$ a média da sequência

- Lei *fraca* dos grandes números
  - para qualquer $\epsilon > 0$

$$\lim_{n \to \infty} P[|\overline{X_n} - \mu| < \epsilon] = 1$$

convergência em probabilidade

Gerando Outras Distribuições

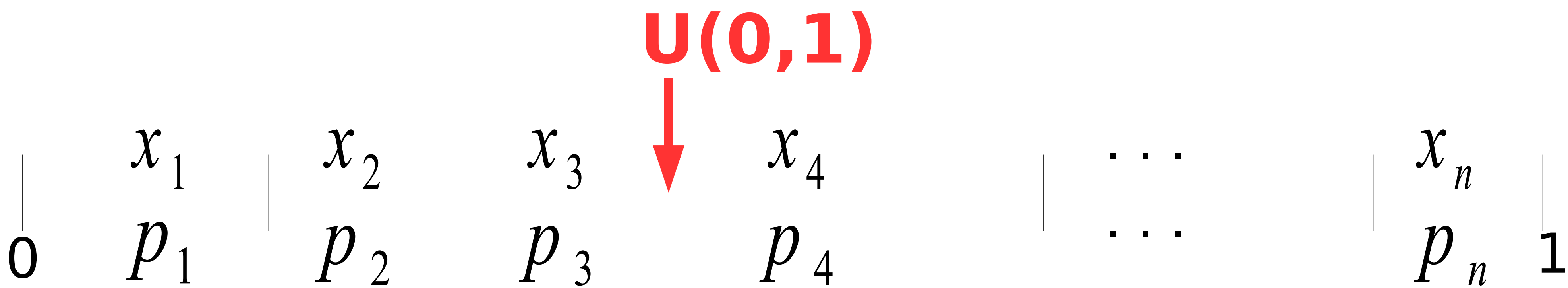
- Como gerar v.a. de outras distribuições?

  Utilizar a uniforme!

- V.a. discreta *X* (*n* valores diferentes)

$$P[X=x_j]=p_j \qquad \sum_{j=1}^{n} p_j = 1$$

- Como gerar valores de *X*?



Gerando V.A. Discreta

- V.a. discreta *X*

$$P[X=x_j]=p_j \qquad \sum_{j=1}^{n} p_j=1$$

- Se $0<a<b<1$ e *U* é uniforme (0,1) , então

$$P[a\leqslant U<b]=b-a$$

definição da
v.a. uniforme!

- Assim, temos

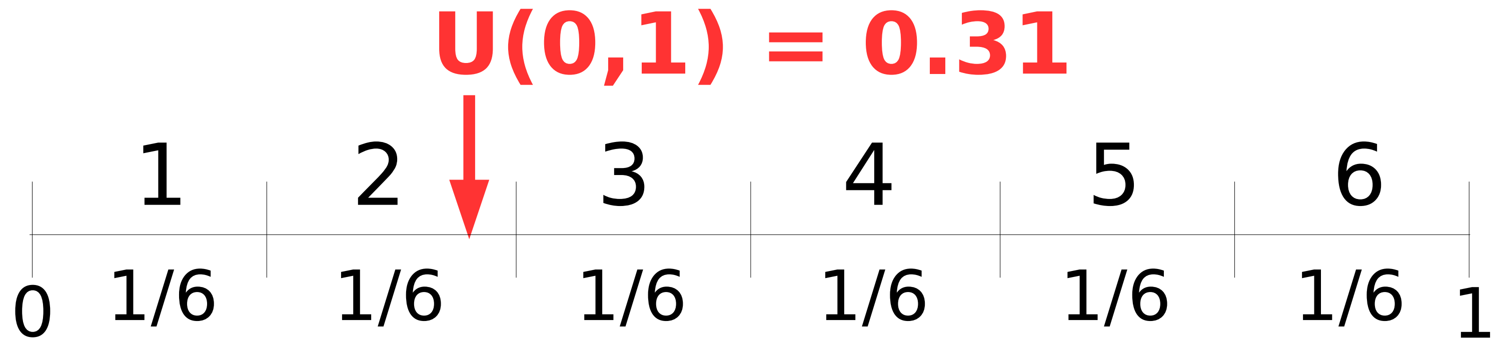
$$P[X=x_j]=P[\sum_{i=1}^{j-1} p_i\leqslant U<\sum_{i=1}^{j} p_i]=p_j$$

Jogando um Dado

- Como gerar resultado de um dado?
- Qual a função probabilidade do dado?

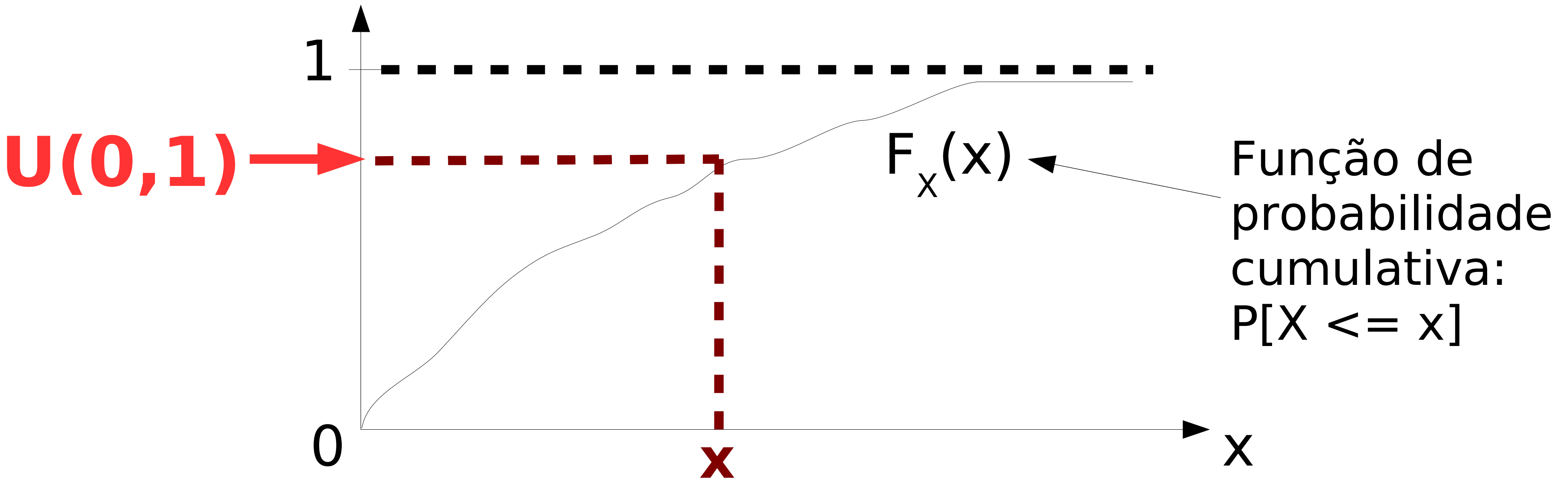
$$P[D=i]=1/6$$ para i = 1, 2, 3, 4, 5, 6

D é a v.a. que indica resultado do dado

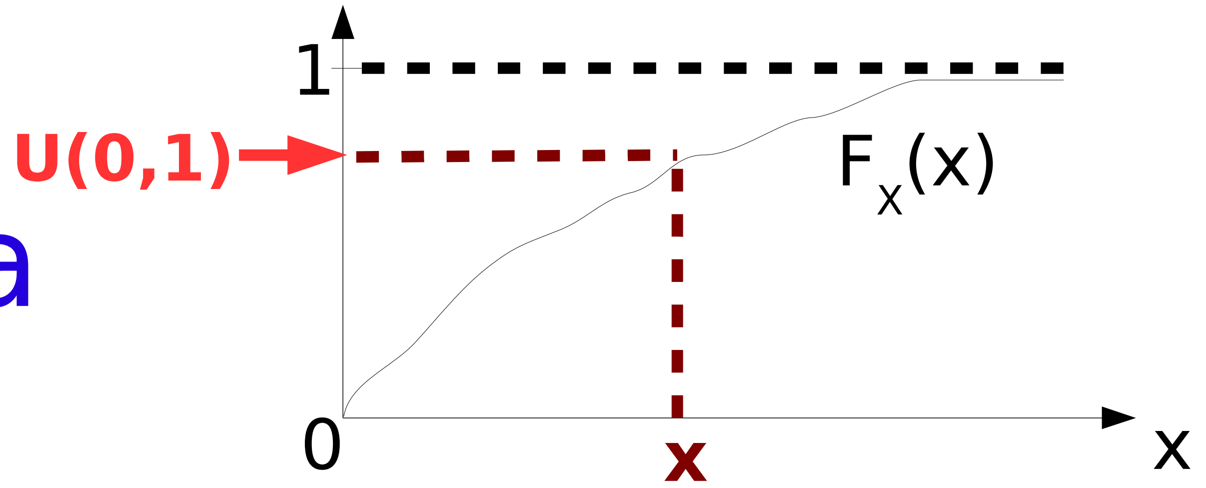


Gerando V.A. Contínua

- Mesma idéia: utilizar uniforme(0,1)
- Método da Transformada Inversa
  - generalização da técnica que acabamos de ver



Método da Transformada Inversa



- Seja *U* uniforme (0, 1) e *X* uma v.a. com função distribuição cumulativa *F*. *X* é dado por:

$$X = F^{-1}(U)$$

- onde F^{-1} é a inversa de *F* (valor de x para o qual *F*(x) = u)

Método da Transformada Inversa (Prova)

- Seja X uma v.a. com função de prob. cumulativa F_X
- Seja U uma v.a. Uniforme (0,1)
- Assuma que X é obtido através de $F^{-1}(U)$
- Provar que $F_X(x) = F(x)$

$$F_X(x) = P[X \leq x]$$

$$= P[F^{-1}(U) \leq x] \quad \longleftarrow \text{definição}$$

$$= P[F(F^{-1}(U)) \leq F(x)] \quad \longleftarrow \text{Equivalência de eventos e monotonicidade de F}$$

$$= P[U \leq F(x)]$$

$$= F(x) \quad \longleftarrow \text{definição uniforme}$$

Gerando uma Exponencial

- Seja X v.a. exponencial com parâmetro λ

$$F_X(x) = 1 - e^{-\lambda x}$$

- Pelo método, temos que $x = F_X^{-1}(u)$
- Então

$$F_X(x) = u$$

$$1 - e^{-\lambda x} = u$$

$$e^{-\lambda x} = 1 - u$$

$$-\lambda x = \log(1-u)$$

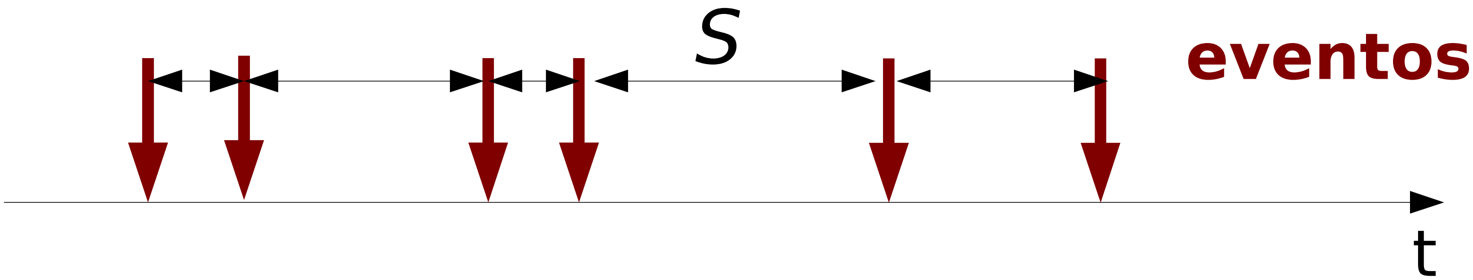
$$x = -\frac{1}{\lambda}\log(1-u)$$

(1-U) também é uniforme(0,1)

$$x = -\frac{1}{\lambda}\log(u)$$

Gerando um Processo de Poisson

- Como gerar eventos de acordo com o processo de Poisson?



- Seja *N(t)* uma v.a. Poisson com parâmetro λ

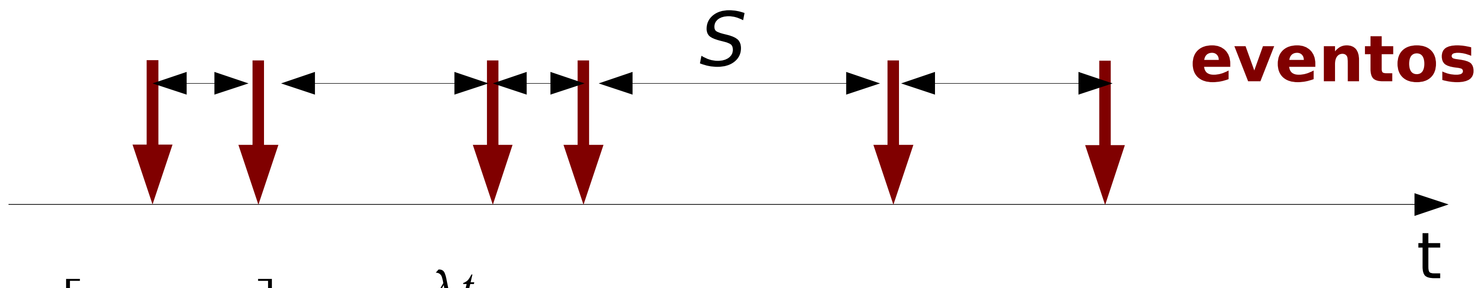
$$P\left[N\left(t_o\right)=k\right]=\frac{\left(\lambda t_o\right)^k}{k!}e^{-\lambda t_o}$$

- Seja *S* o tempo entre chegadas
- Qual a distribuição do tempo entre chegadas?

$$P\left[S>t\right]=?$$

Exponencial!

Gerando um Processo de Poisson



$$P[S>t]=e^{-\lambda t}$$

- Gerar sequência de exponenciais com parâmetro λ

- Quantos eventos em um intervalo t_o?
- Gerar exponencias em sequência até que soma dos tempos seja maior que t_o