
Aula 05:

Computação Evolutiva e Conexionista – Programação Evolucionária

Prof. Hugo Puertas de Araújo
hugo.puertas@ufabc.edu.br
Sala: 509.2 (5º andar / Torre 2)

■ Agenda

■ Programação Evolucionária

- ❖ Visão Geral
- ❖ Definições dos parâmetros da Programação Evolucionária / Evolutiva (PE)

Computação Evolutiva e Conexionista

■ Visão geral

- A Programação Evolutiva (PE) foi proposta por Fogel, Owens e Walsh em meados da década de 60
- “Artificial Intelligence Through Simulated Evolution”
- Proposta original:
 - ❖ Predição de comportamento usando máquinas de estados finitos.
 - ❖ Foco no comportamento e não na otimização

■ Visão geral

- Na PE, cada indivíduo gera um único descendente através de mutação (não é usado recombinação).
- A melhor metade da população ascendente e a melhor metade da população descendente são reunidas para formar a nova geração (estratégia $\mu+\mu$).
 - ❖ Que tipo de seleção é esta?

Exercício

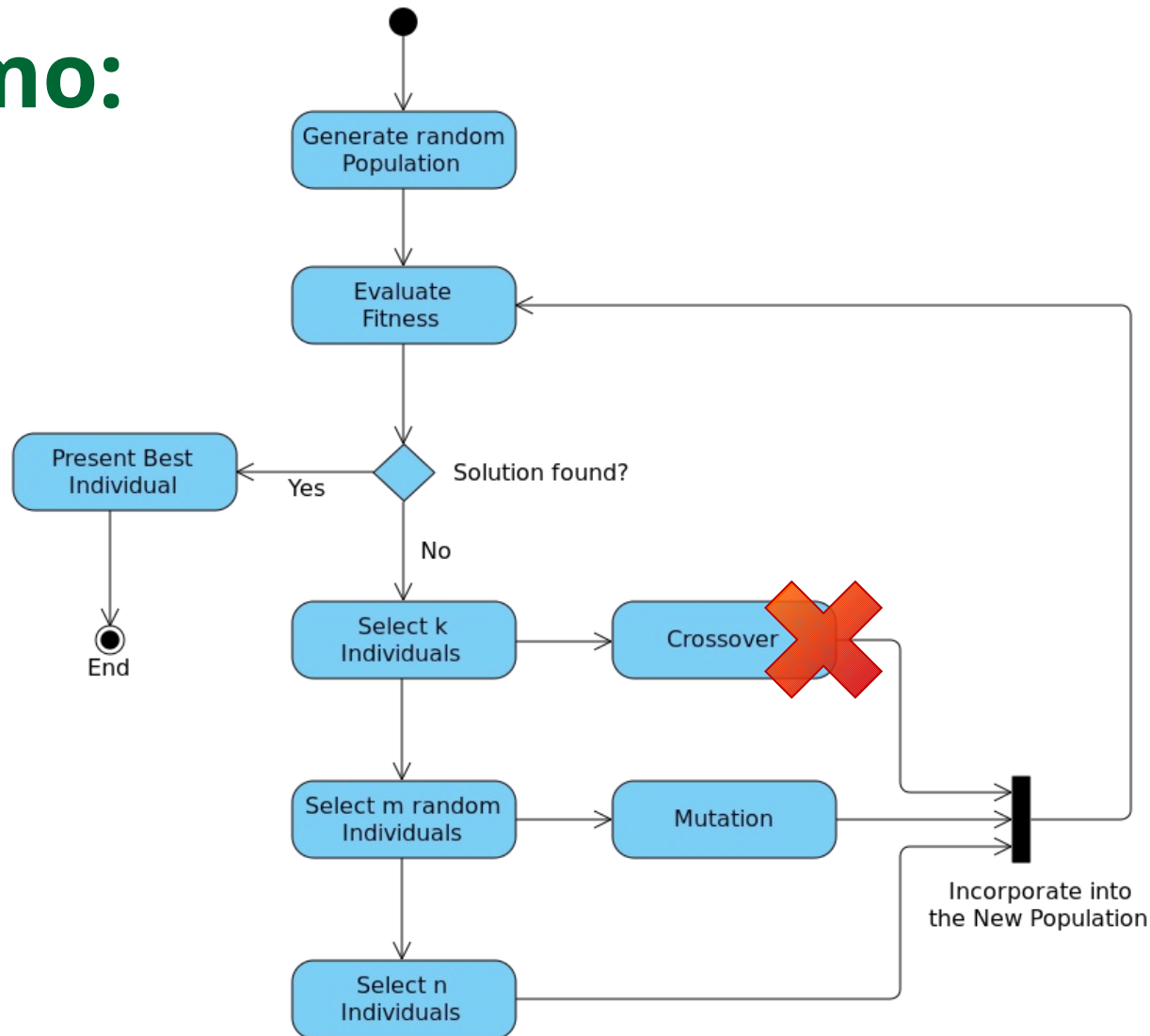
■ Visão geral

- Na PE, cada indivíduo gera um único descendente através de mutação (não é usado recombinação).
- A melhor metade da população ascendente e a melhor metade da população descendente são reunidas para formar a nova geração (estratégia $\mu+\mu$).
 - ❖ Elitista

■ Visão geral

- Diferentemente dos AGs, a PE enfatiza desenvolvimento de modelos comportamentais
 - ❖ Modelar o comportamento afim de prever o que pode acontecer (PREDIÇÃO).
 - ❖ Capturar a interação do sistema com seu ambiente.

O algoritmo:



■ Máquinas de Estados Finitos – FSM

- Uma maneira comum de se prever uma ação consiste na análise de ações passadas.
- No contexto de uma máquina de estados finitos, cada ação pode ser representada por um símbolo.
 - ❖ Dado uma sequência de símbolos, deve-se prever qual será o próximo símbolo.

■ Máquinas de Estados Finitos – FSM

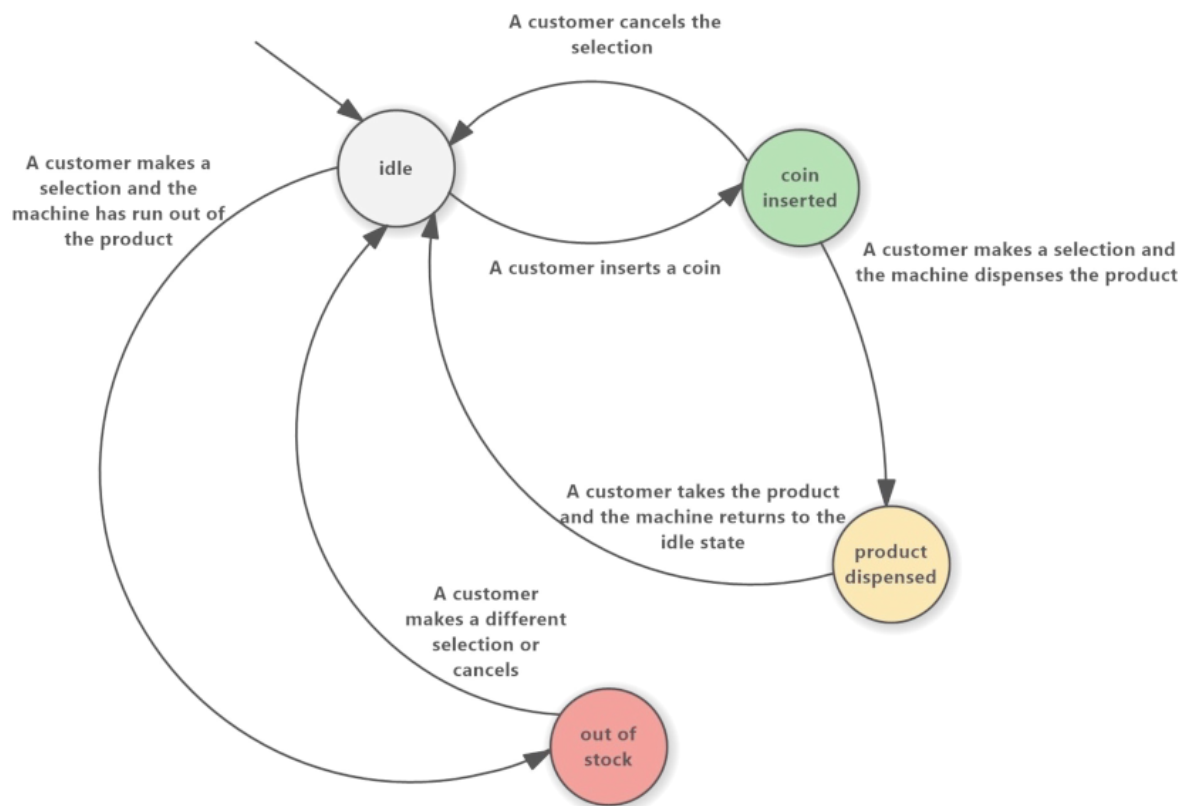
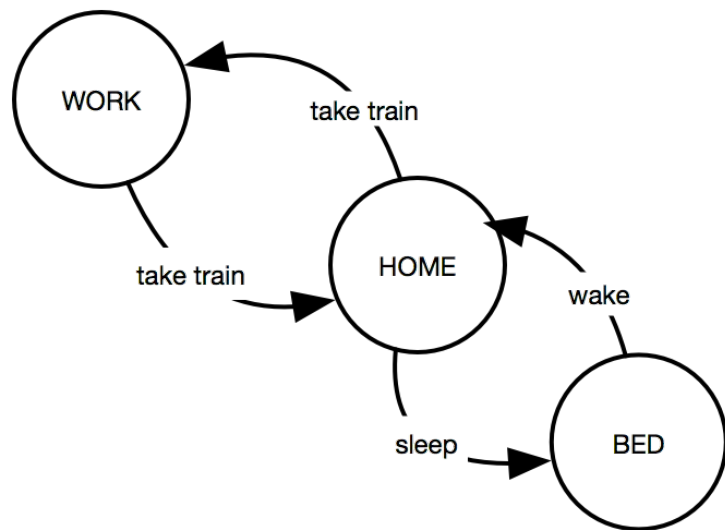
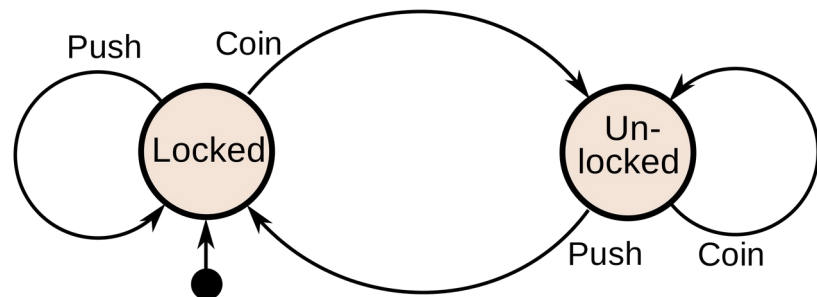
- Assim como nos AGs, os símbolos devem pertencer a um alfabeto finito.
- Máquina de Estados Finitos:
 - ❖ Analisar a sequência de símbolos
 - ❖ Gerar uma saída que otimize uma dada função de fitness, a qual envolve a previsão do próximo símbolo da sequência
 - Mercado financeiro, previsão do tempo, etc...

■ Máquinas de Estados Finitos – FSM

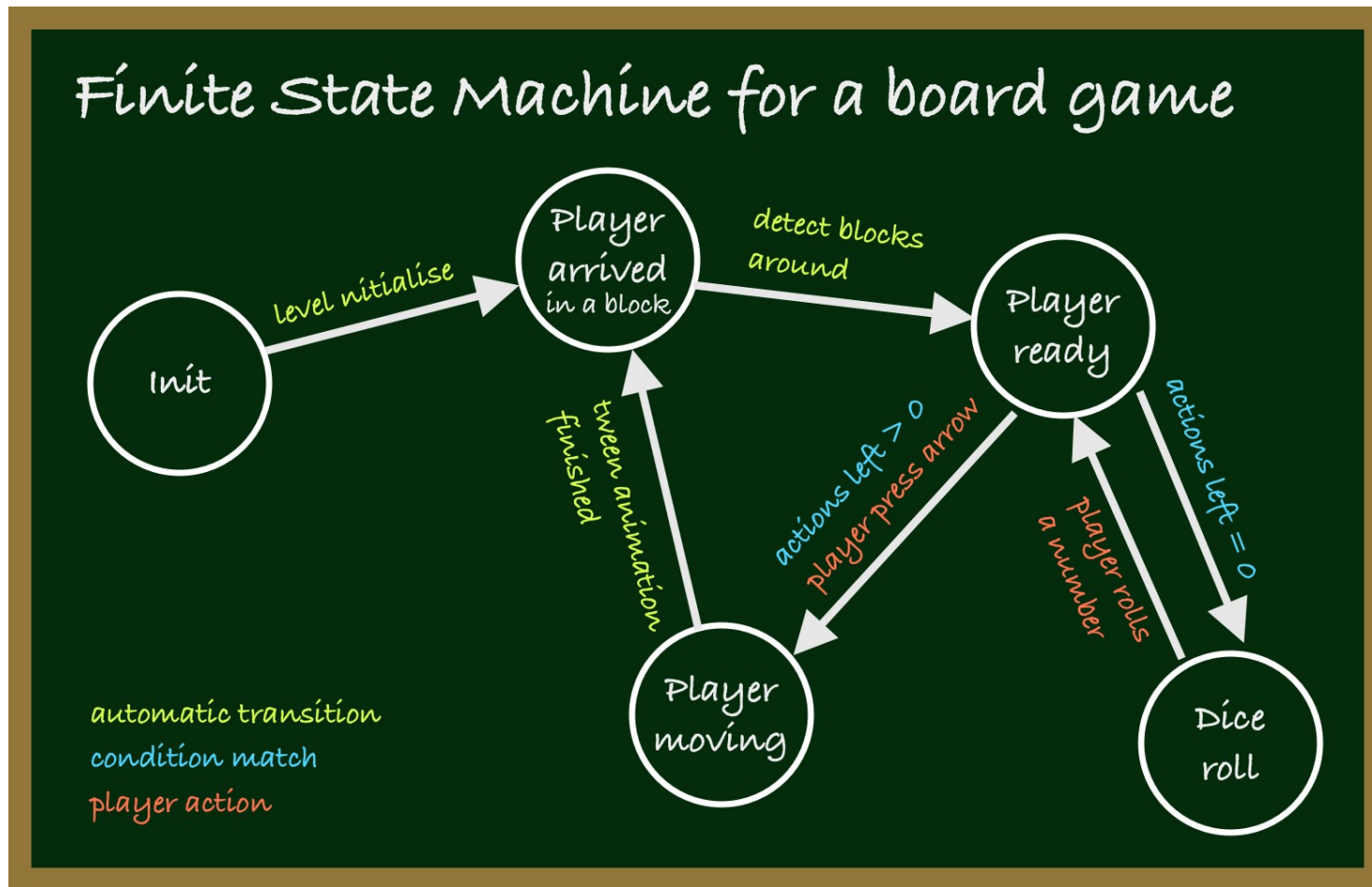
■ Podem ser vistas como transdutores:

- ❖ Quando estimulado por um alfabeto finito de símbolos, responde com um outro alfabeto finito de símbolos e possui um número finito de estados.
- ❖ Alfabetos de entrada e saída não são necessariamente idênticos.

Máquinas de Estados Finitos – FSM



Máquinas de Estados Finitos – FSM



Máquinas de Estados Finitos – FSM

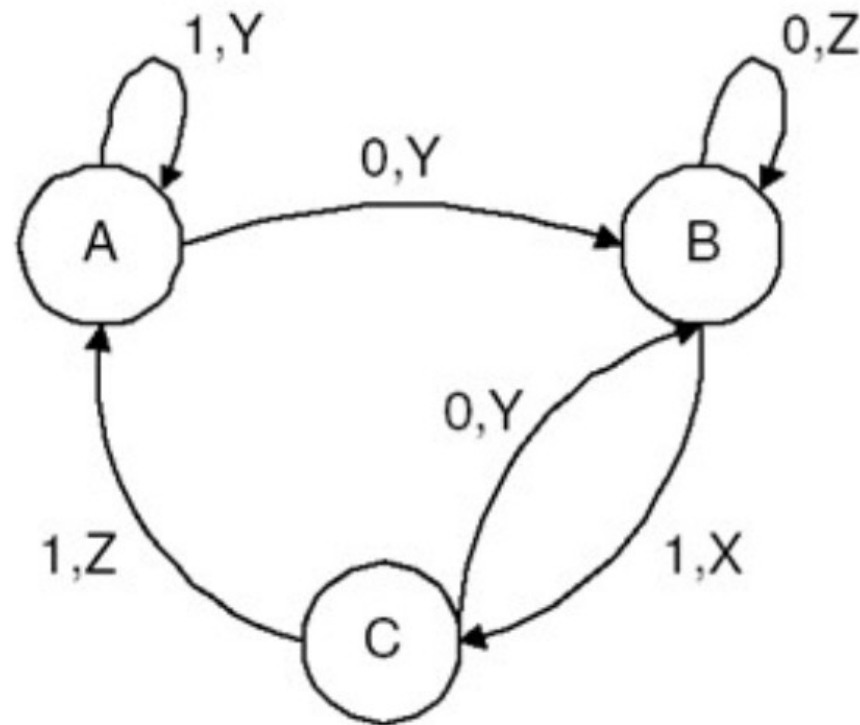
Estados Entradas	1	0
A	Y,A	Y,B
B	X,C	Z,B
C	Z,A	Y,B

Alfabeto de entrada de dois símbolos:

$I = \{1, 0\}$

Alfabeto de saída de três símbolos:

$O = \{X, Y, Z\}$

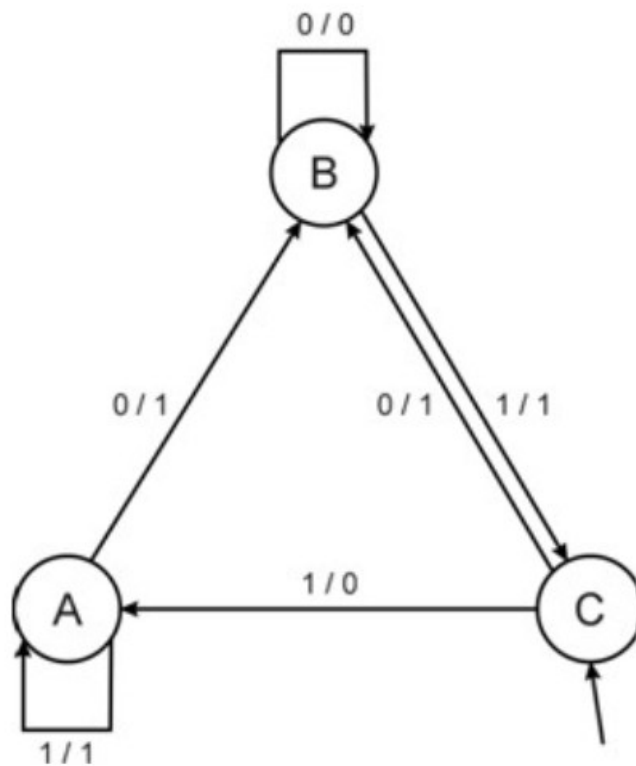


Máquina de três estados
 $S = \{A, B, C\}$

PE – Um exemplo

Tarefa: Prever a próxima entrada

- Medida da Qualidade:
 - Número de previsões corretas
- Estado Inicial: C
- Sequência de Entrada
 - 011101
- Sequência de Saída
 - 110111
- Qualidade: 3 de 5



$$S = \{A, B, C\} \quad I = \{0, 1\} \quad O = \{0, 1\}$$

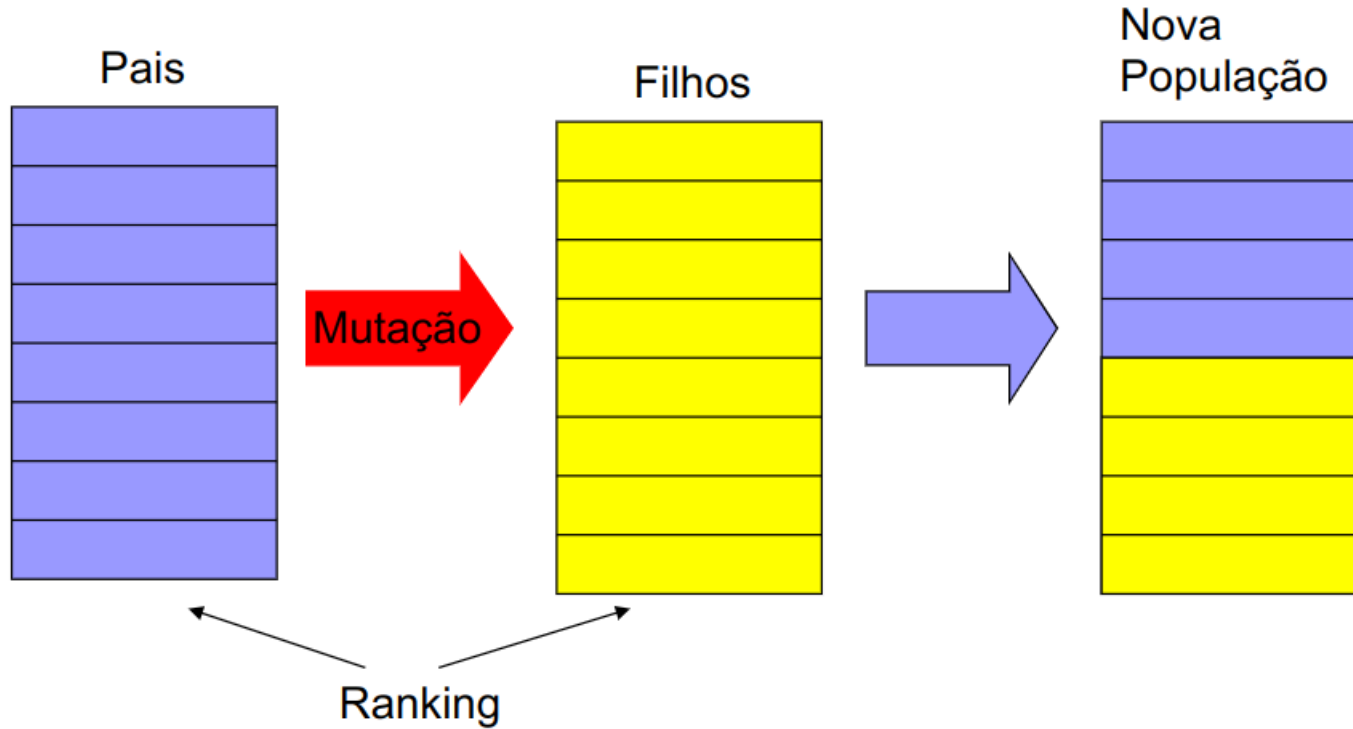
Mutação

- Cinco tipos de mutação podem ocorrer em uma máquina de estado finitos:
 - ❖ O estado inicial pode mudar.
 - ❖ O estado inicial pode ser eliminado.
 - ❖ Um estado pode ser adicionado.
 - ❖ Uma transição entre estados pode ser mudada.
 - ❖ O símbolo de saída para um determinado estado e o símbolo de entrada podem ser mudados.

■ Seleção

- Uma vez que cada pai gera um filho após a mutação, a população dobra de tamanho a cada geração.
- Após o cálculo da fitness, conserva-se a melhor metade dos pais ($\mu/2$) e a melhor metade dos filhos ($\mu/2$).
 - ❖ População de tamanho constante.
 - ❖ Elitismo presente por conservar também os melhores pais

■ Seleção



■ Critério de parada

- Fazer a predição utilizando o melhor indivíduo da população.
 - ❖ Isso pode ocorrer a qualquer instante
 - ❖ Se a fitness for satisfatória (Lei da Suficiência) o algoritmo pode ser terminado.
 - ❖ Fixar um número de gerações.

Tamanho de cromossomo

- Diferentemente de outros paradigmas evolutivos, na PE a mutação pode mudar o tamanho do indivíduo.
 - ❖ Estados podem ser adicionados ou eliminados, de acordo com as regras vistas anteriormente.
 - ❖ Isso pode causar alguns espaços na tabela
 - Mutações neutras.

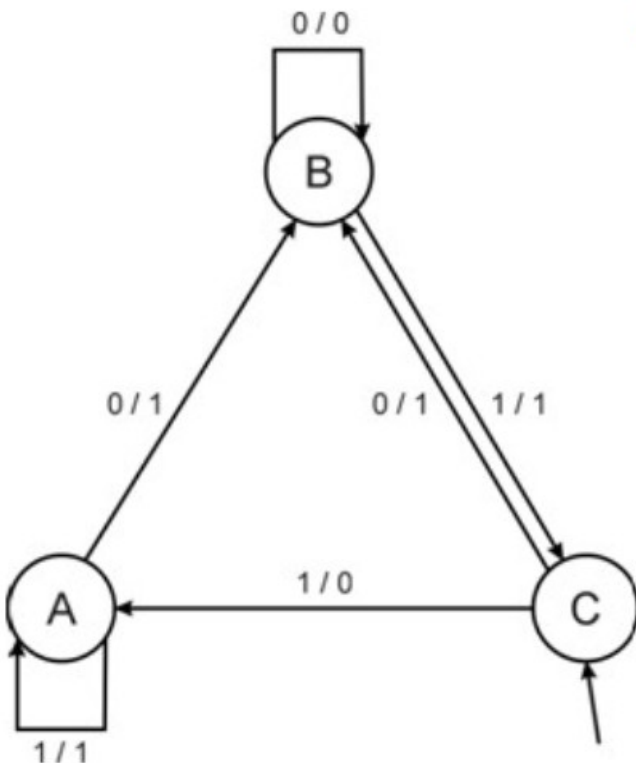
Tamanho de cromossomo

- A mutação ainda pode criar uma transição que não seja possível, pois um estado pode ter sido eliminado durante a mutação.
- Esses problemas devem ser identificados e corrigidos durante a implementação
- Menos frequentes em máquina com bastante estados.

Tamanho de cromossomo

- Embora PE possa ter indivíduos de tamanho variável, é possível evoluir uma máquina de estado finitos com PE onde os indivíduos tem tamanho fixo.
 - ❖ Definir um número máximo de estados.
- Para exemplificar, vamos considerar a máquina de predição apresentada anteriormente, a qual pode ter no máximo 4 estados (próximo slide):

Tamanho de cromossomo



Cada estado pode ser representado por 7 bits

Bit No.	Representação
0	1 ativo; 0 não ativo
1	símbolo de entrada
2	símbolo de entrada
3	símbolo de saída
4	símbolo de saída
5	estado de saída
6	estado de saída

A	B	C	D
1 1 0 1 1 A B	1 0 1 0 1 B C	1 1 0 0 1 A B	0 0 0 0 0 D A

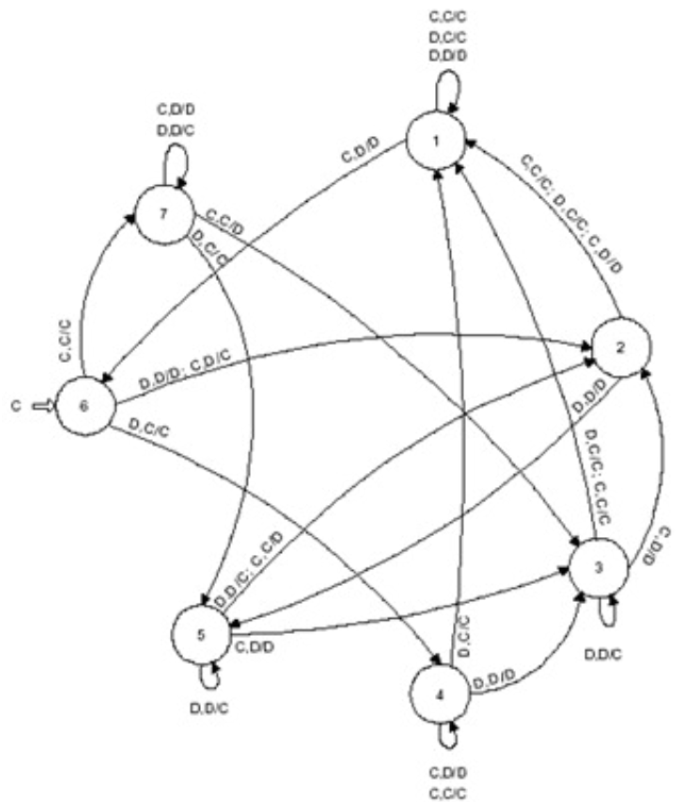
Tamanho de cromossomo

- Como visto, cada estado pode ser representado por uma string de 7 bits.













A	B	C	D
1 1 0 1 1 A B	1 0 1 0 1 B C	1 1 0 0 1 A B	0 0 0 0 0 D A

- Sendo assim, cada indivíduo possui 28 bits
 - Cada um representa uma máquina completa.

Dilema do prisioneiro



Prisoners' dilemma

Prisoners' dilemma		prisoner B			
		confess 	remain silent 		
prisoner A	confess 	 5 years	 5 years	 0 year	 20 years
	remain silent 	 20 years	 0 year	 1 year	 1 year

© 2010 Encyclopædia Britannica, Inc.