

matriz de estado - auto 1 +

potencia de estado

estado critico	$\lambda = 1$	$\lambda = 1$	$\lambda = 1$
potencia de estado	$\text{pot} = 2$	$\text{pot} = 1$	potencia de estado
	$\text{pot} = 2$	$\text{pot} = 1$	

< estado (1) estado critico >

estado critico $\lambda = 1$ \Rightarrow $\text{pot} = 2$ \Rightarrow $\text{pot} = 1$ \Rightarrow $\text{pot} = 1$

estado (1) $\lambda = 1$ \Rightarrow $\text{pot} = 2$ \Rightarrow $\text{pot} = 1$ \Rightarrow $\text{pot} = 1$

estado critico $\lambda = 1$ \Rightarrow $\text{pot} = 2$ \Rightarrow $\text{pot} = 1$ \Rightarrow $\text{pot} = 1$

estado critico $\lambda = 1$ \Rightarrow $\text{pot} = 2$ \Rightarrow $\text{pot} = 1$ \Rightarrow $\text{pot} = 1$

qual tipo a cond. de $\lambda = 1$ \Rightarrow $\text{pot} = 2$ \Rightarrow $\text{pot} = 1$ \Rightarrow $\text{pot} = 1$

- Selec. jatos para o polinomio da estado critico

quin. local | reducao matricial \Rightarrow $\text{pot} = 2$ \Rightarrow $\text{pot} = 1$ \Rightarrow $\text{pot} = 1$

lin. de derivada \Rightarrow $\text{pot} = 2$ \Rightarrow $\text{pot} = 1$ \Rightarrow $\text{pot} = 1$

sem estado derivada \Rightarrow $\text{pot} = 2$ \Rightarrow $\text{pot} = 1$ \Rightarrow $\text{pot} = 1$

estado critico $\lambda = 1$ \Rightarrow $\text{pot} = 2$ \Rightarrow $\text{pot} = 1$ \Rightarrow $\text{pot} = 1$

$\langle (1 - \text{pot}) \text{pot} \rangle$ \Rightarrow $\text{pot} = 2$ \Rightarrow $\text{pot} = 1$ \Rightarrow $\text{pot} = 1$

- Se - beak (algebra de Pitagoras)

adica: o passo da derivada quando a das que vem antes no estado critico

Uma unica derivada para derivar o ultimo o estado

relacao de derivada mais simples

$\langle (1 - \text{pot}) \text{pot} \rangle$	$\langle (1 - \text{pot}) \text{pot} \rangle$
$\text{pot} = 2$	$\text{pot} = 1$

potencia de estado	$\text{pot} = 2$	$\text{pot} = 1$	$\text{pot} = 1$
potencia de estado	$\text{pot} = 2$	$\text{pot} = 1$	$\text{pot} = 1$

→ 1ª aula: basicamente a mesma

	1	2
questão de entrada	in1 = true; ①	in2 = true;
	while (in2) skip;	while (in1) skip;

→ pode dar deadlock! se ambos a linha ① de 2 ^{no} entram ^{no} mesmo tempo de deadlock
 → isso não ocorre, porque se 2 entrar primeiro, 1 não entra

→ uma variável auxiliar para a caso entre in1 e in2 não concordarem
 last: variável que indica o último que entrou

```

while (in1 & false; in2 & false; ) {
  // ...
}
while (true) {
  in1 = true;
  last = 1;
  < await ('in1 ou last == 1'); >
  // código crítico;
  in1 = false;
  // código não crítico;
}

```

→ in1 = true; → se não o último que quis entrar
 last = 1; → se não o último que quis entrar
 < await ('in1 ou last == 1'); > → o último que entrar o last deve ser o último que quiser

```

while (true) {
  in2 = true;
  last = 2;
  < await ('in2 ou last == 2'); >
  // código crítico;
  in2 = false;
  // código não crítico;
}

```

→ in2 = true; → se não o último que quis entrar
 last = 2; → se não o último que quis entrar
 < await ('in2 ou last == 2'); > → o último que entrar o last deve ser o último que quiser

→ isso não dá um deadlock porque um de 1 e 2 sempre entra primeiro e o outro sempre entra depois

```

< await ('in1 ou last == 1'); >
while (in1 ou last == 1) skip;

```

→ isso não dá um deadlock porque um de 1 e 2 sempre entra primeiro e o outro sempre entra depois

→ 2^a tentativa:

$(a=0, 1), (a=1, 2), \dots, (a=0, 1), (a=1, 2), \dots$

	1	2
procedura de intrare	mark1 = true;	mark2 = true; $\{ (a,b) \}$ difer
	while (mark2) ship;	while (mark1) ship;

↓

procedura de iesire:

$\{ (a,b) \}$ difer $\{ (a,b) \}$ difer

→ intrare: dactila se daua confida

$(1, \dots, 1) \dots (1, 1) \dots (1, 1) \dots$

→ iesire: dactila se daua confida

mark1 = true;

	1	mark1 = true;
procedura de intrare	mark1 = true;	mark2 = true;
	while (mark2) {	while (mark1) {
pe a si va pe b	mark1 = false;	mark2 = false;
a si b este egala	ship;	ship;
cu dactila	mark1 = true;	mark2 = true;

↓

atras de recursivitate: se unu dea dactila - nu toate

Algoritm de iesire: + marcare intrare true = 1

pe a si b parca intrare =

→ marcare intrare true

iesire a parca si atras

	1	2
procedura de intrare	mark1 = true;	mark2 = true;
	while (mark2) {	while (mark1) {
	if (true == 1) {	if (true == 1) {
	mark1 = false;	mark2 = false;
	while (true == 1) ship;	while (true == 1) ship;
	mark1 = true;	mark2 = true;

	1	2
palabra de salida	turn = 2; word = folo;	turn = 1; word = folo;

→ arrow & labels & extra

• Algoritmo Maza - Crucel

word = 0; word = 0;

	1	2
palabra de salida	< if (word = 0) word = 1; else word = 1; > while (word = 1 + word = 2) stop;	< if (word = 1) word = 0; else word = 0; > while (word = 2 - word = 1) stop;
palabra de salida	word = 0;	word = 0;

¿1 que es word = 0

¿1 o 2 que es word en un tiempo?

algun momento o if antes ya se el i anterior

de compare en 2 e el prob execution o if primero (o word de 1° to 0/1)

en terminales en if 5 paises: if 1

if 2

do 2

while 2 o e when

do 1

while 1

en terminales de if e recursion.