

# UPS com planta supervisionada - apresentação de acompanhamento

Aluno: João Guilherme de Oliveira Jr.



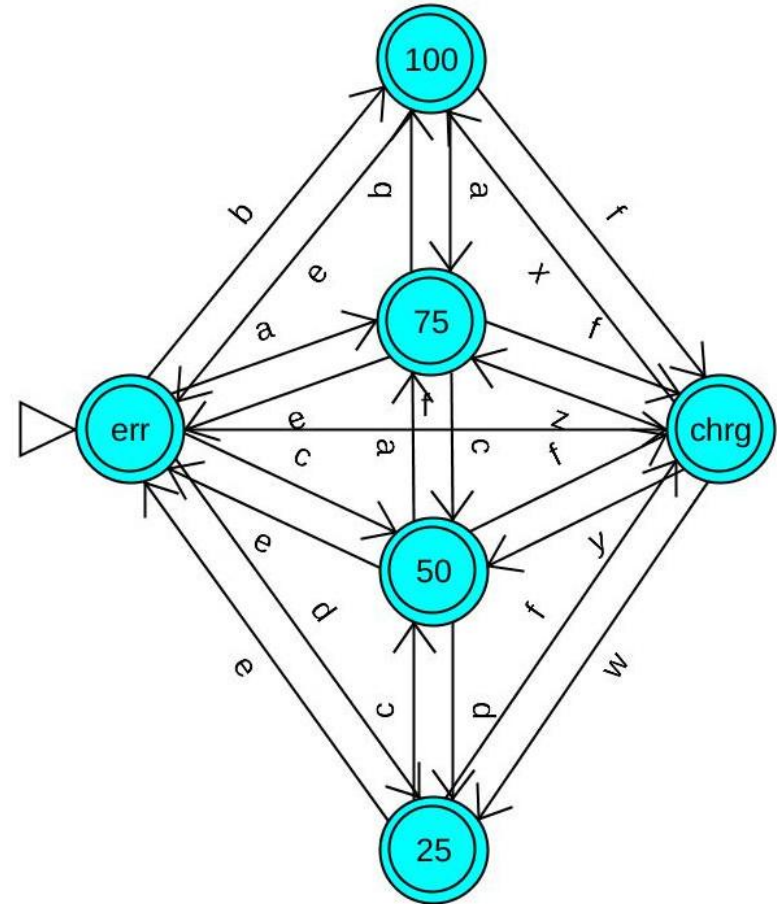
# Autômato da bateria

a, b, c, d - eventos que são criados a partir da combinação dos sinais provenientes dos sensores da bateria.

e - evento de erro que é composto a partir dos sinais dos sensores da bateria.

f - evento não controlado que ocorre quando há falta de energia elétrica.

w, x, y, z - eventos compostos pelo eventos a, b, c, d e f.



# Autômato da bateria

type bat\_states = BERR | B100 | B75 |  
B50 | B25 | BCHR

node bateria(ac, cp, l1, l2, v:bool) returns  
(estado\_bat:bat\_states)

$e = \text{not } a \ \& \ \text{not } b \ \& \ \text{not } c \ \& \ \text{not } d;$

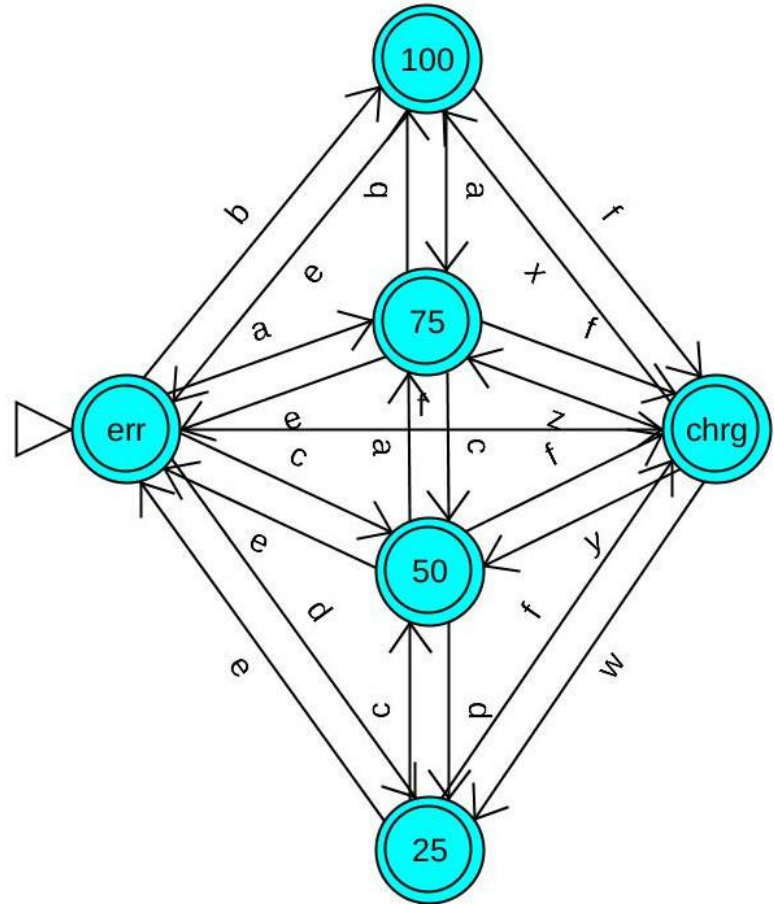
$f = ac$

$z = \text{not } f \ \& \ a$

$x = \text{not } f \ \& \ b$

$y = \text{not } f \ \& \ c$

$w = \text{not } f \ \& \ d$



# Novo estado da bateria.

...

state Carregando do

estado\_bat = BCHR;

unless not ac & b then Cheio

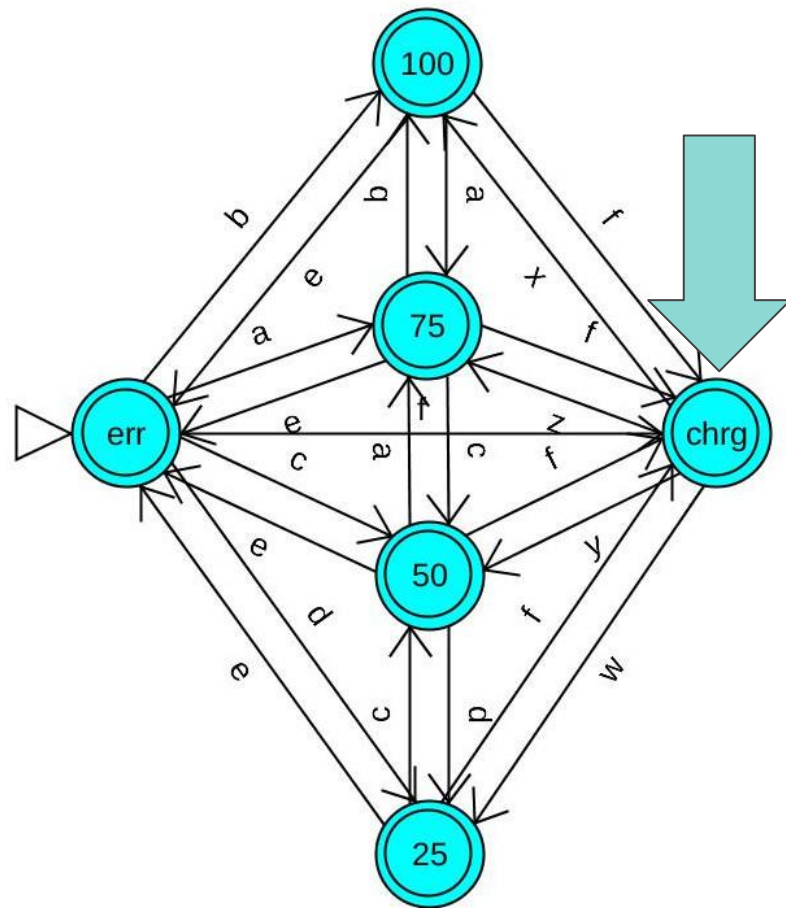
| not ac & a then Limiar1

| not ac & c then Limiar2

| not ac & d then Baixo

end

...





## **Políticas adicionais**

- 1 - Quando voltar a energia todos os equipamentos devem ficar ligados;**
- 2 - O controlador só permite que a planta opere se a bateria não estiver com problemas;
- 3 - Se a bateria estiver com problema, o led indicador de troca de bateria deve permanecer ligado.
- 4 - Não manter o modem ligado se não houver conectividade com internet (falha do provedor), desligando-o em intervalos de 15 minutos.

# Implementação da primeira regra.



...

| BERR do

erro = true;

r\_dia = dia & not pabx\_atuador & not modem\_atuador & not serv\_atuador;

r\_noite = not dia & not pabx\_atuador & not modem\_atuador & not serv\_atuador;

| BCHR do

erro = false;

r\_dia = dia & pabx\_atuador & (modem\_estado > 0 ) & (serv\_estado > 0 );

r\_noite = not dia & pabx\_atuador & (modem\_estado > 0 ) & (serv\_estado > 0 );

...



# Resolução do problema do controlador sintetizado - “Caminhada aleatória”

```
node computador(c_atx, off_atx, ts, cr1, cr3, cr4, ba, b1, b3, b4, s_atx, s1, s3, s4:bool)  
returns (atuador:bool; estado: int;)
```

```
var a, r1, r3, r4: bool;
```

```
let
```

```
a = (ba & s_atx) or not c_atx;
```

```
r1 = (b1 & s1) or not cr1;
```

```
r3 = (b3 & s3) or not cr3;
```

```
r4 = (b4 & s4) or not cr4;
```

Automaton

...



# Obrigado!

Código e arquivos atualizados em:

[https://github.com/joaogojunior/ufrpe\\_tmc.git](https://github.com/joaogojunior/ufrpe_tmc.git)