

DevTitans

Máquinas de Estados Finitos

Desenvolvimento de Software

- A maioria dos programadores preferem iniciar a codificação **sem qualquer planejamento formal prévio**
- O argumento é que qualquer planejamento formal **desperdiça muito tempo** sem melhoria significativa da qualidade do produto final
- Seriam bem-vindos métodos de projeto **simples de usar**, que ajude a criar **código melhor**, além de gerar a **documentação**

Máquina de estados finitos

- Uma máquina de estados finitos (também chamado de **autômato finito determinístico**) é uma **abstração matemática** que pode ser usada para projetar algoritmos
- É uma máquina abstrata, baseada em **estados**, que está em somente **um** dos estados por vez
- Uma **transição** indica uma mudança de estado e é descrita por uma **ação** (ou evento) que precisa ser satisfeita
- A **ação** é a descrição de uma atividade que precisar ser realizada para que a transição ocorra

Autômatos com saídas

Um autômato com saída é definido pela sêxtupla $(Q, \Sigma, \delta, q_0, \Delta, \sigma)$ em que:

Q é um conjunto finito de estados

Σ alfabeto de entrada

$\delta: Q \times \Sigma \rightarrow Q$ é a função de transição

q_0 é o estado inicial

Δ é o alfabeto de saída

$\sigma: Q \rightarrow \Delta$ é a função de saída

São AFDs que não tem estados de aceitação. São máquinas **transdutoras** (*transforma uma sequência de símbolos de entrada em uma sequência de símbolos de saída*). O resultado é o que a máquina vai produzir na saída. Há dois tipos de autômatos com saída: **Moore** e de **Mealy**. A diferença é só na função de saída (σ).

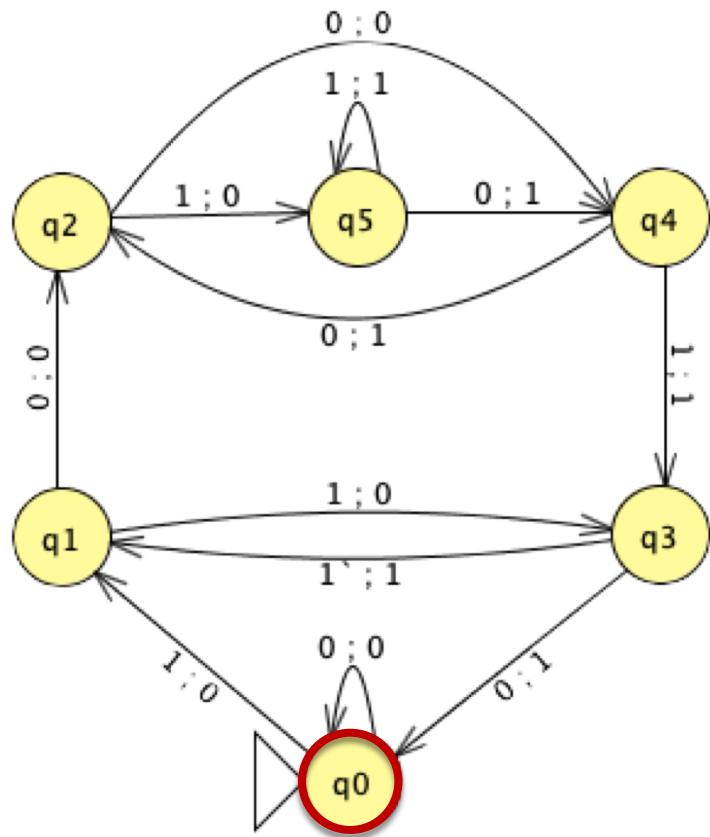
Máquina de Mealy

Exemplo: determina o quociente da divisão de um número binário por seis

$$\omega = 1000$$

Máquina de Mealy: Exemplo

Determina o quociente da divisão de um número binário por seis

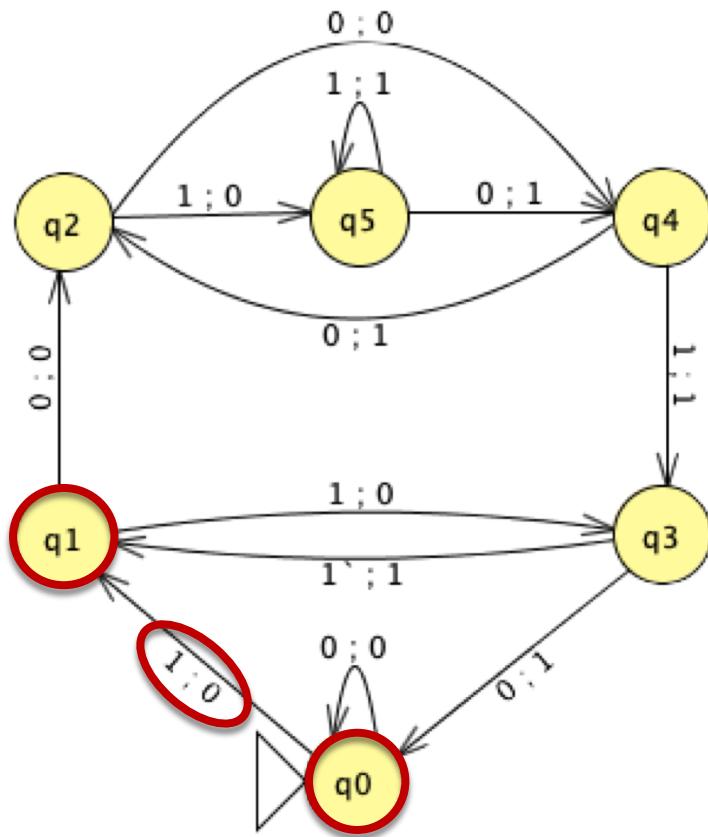


$$\omega = 1000$$

q_0

Máquina de Mealy: Exemplo

Determina o quociente da divisão de um número binário por seis

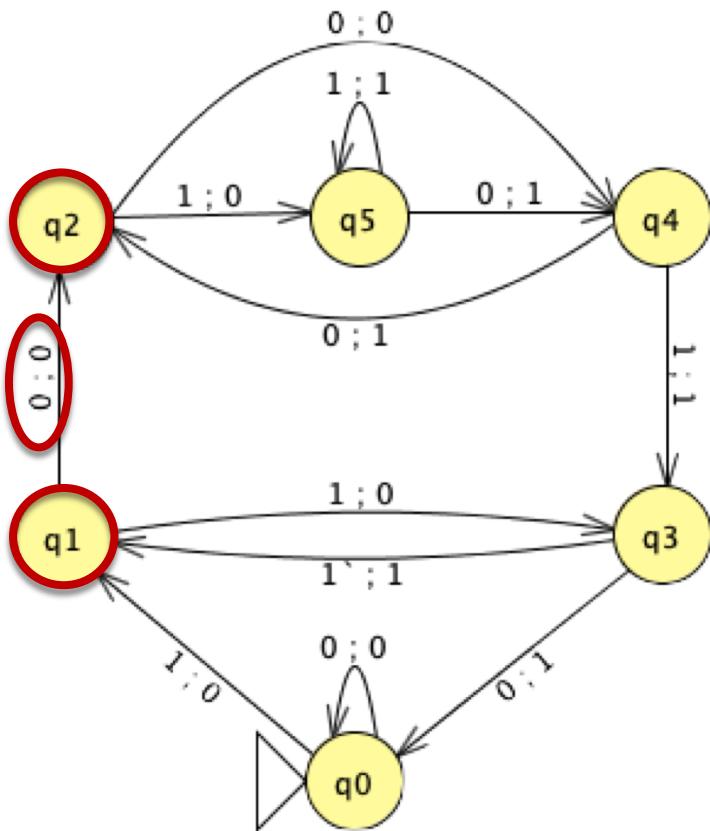


$$\omega = 1000$$

$$q_0 \xrightarrow[0]{1} q_1$$

Máquina de Mealy: Exemplo

Determina o quociente da divisão de um número binário por seis

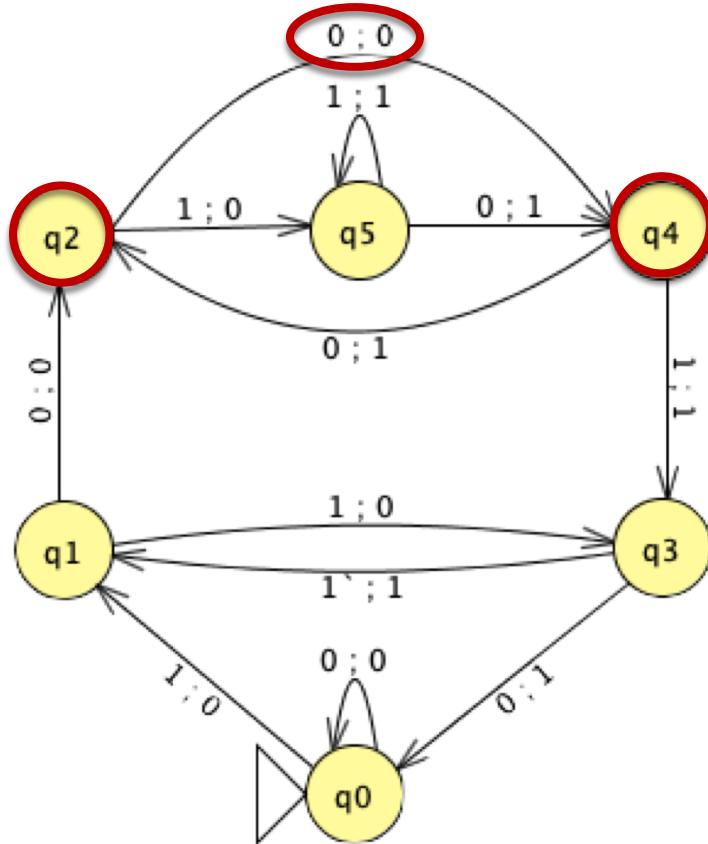


$$\omega = 1000$$

$$q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2$$

Máquina de Mealy: Exemplo

Determina o quociente da divisão de um número binário por seis

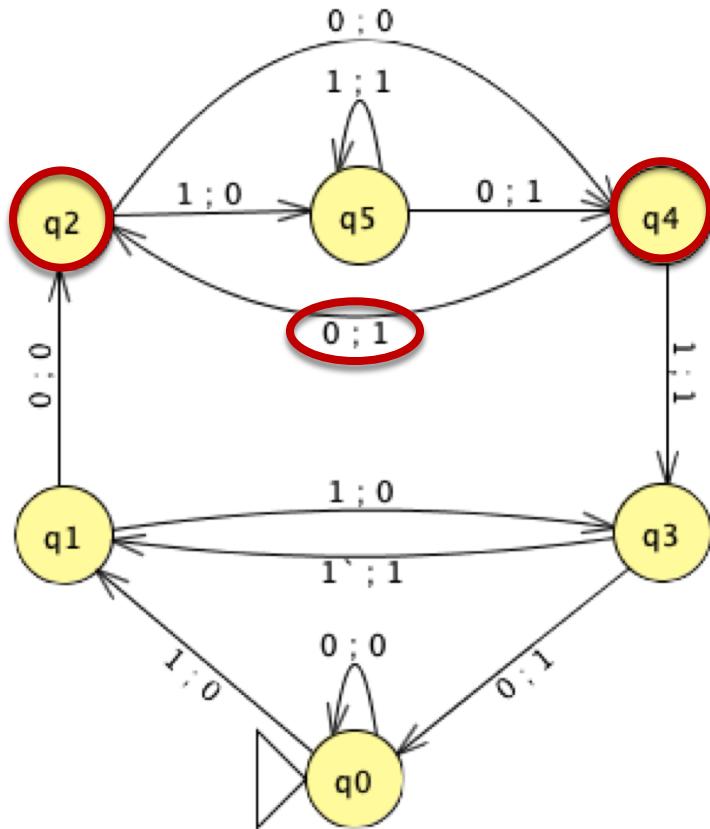


$$\omega = 1000$$

$$q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{0} q_4$$

Máquina de Mealy: Exemplo

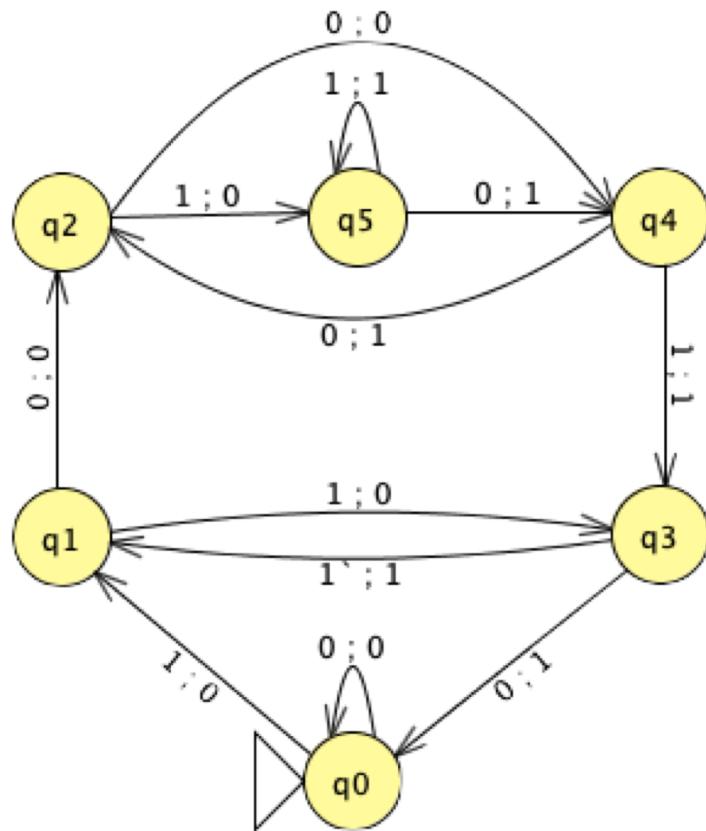
Determina o quociente da divisão de um número binário por seis



$$\omega = 1000$$
$$q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{0} q_4 \xrightarrow{1} q_2$$

Máquina de Mealy: Exemplo

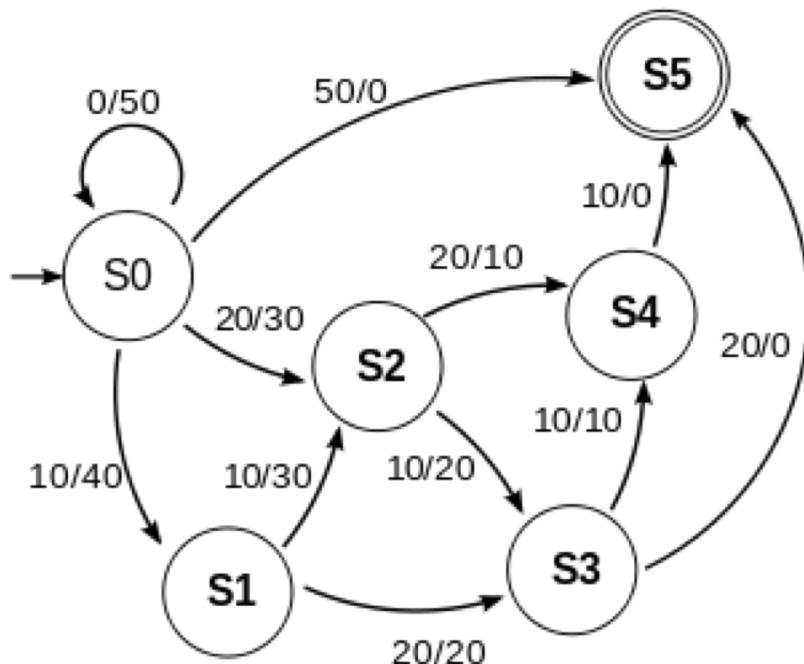
Determina o quociente da divisão de um número binário por seis



$$\omega = 1000$$
$$q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{0} q_4 \xrightarrow{0} q_2$$

Lê o número binário “1000” (8)
e gerou na saída o número
binário “0001” (1)

Máquina de Mealy



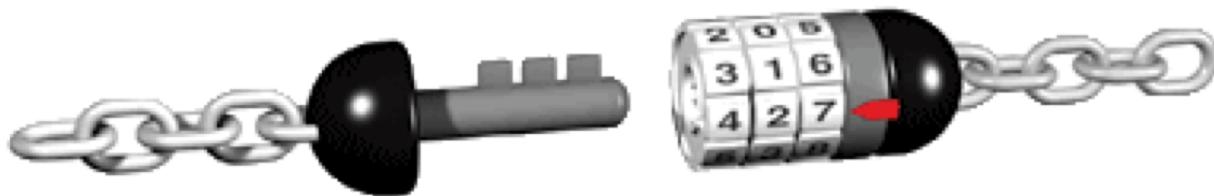
Uma **máquina Mealy** é uma máquina de estado finito cujos valores de saída são determinados pelas entradas que mudam os estados

Qual seria a saída para as seguintes entradas:

- (i) 50; (ii) 20, 10, 20; e (iii) 10,10,10,20?

O que faz esta máquina de estados?

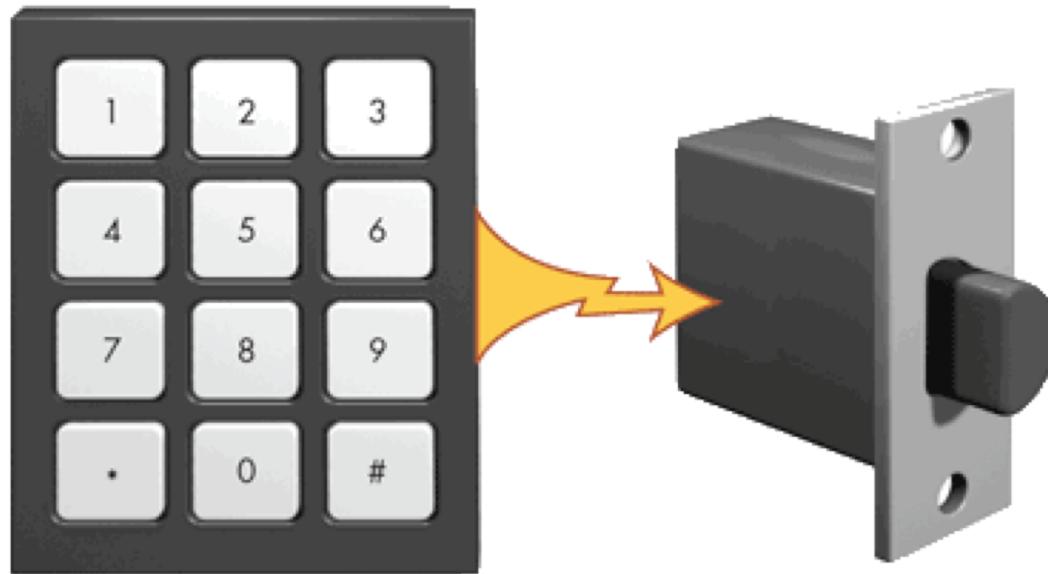
Dispositivo combinacional



Este cadeado abrirá quando os três dígitos forem colocados de acordo com a senha numérica. Não importa a ordem de colocação dos dígitos. Este dispositivo não precisa de memória.

Consequentemente, **não é uma MEF**, mas somente um dispositivo combinacional

Cadeado com teclado numérico



O cadeado acima possui um teclado (dispositivo de entrada) e uma trava (dispositivo de saída). Este é **um bom exemplo de uma MEF**, que são também chamadas de máquinas sequenciais. Ele tem a habilidade de lembrar-se da ordem de entrada dos números.

Cadeado com teclado numérico

:: Tabela de Transição de Estados

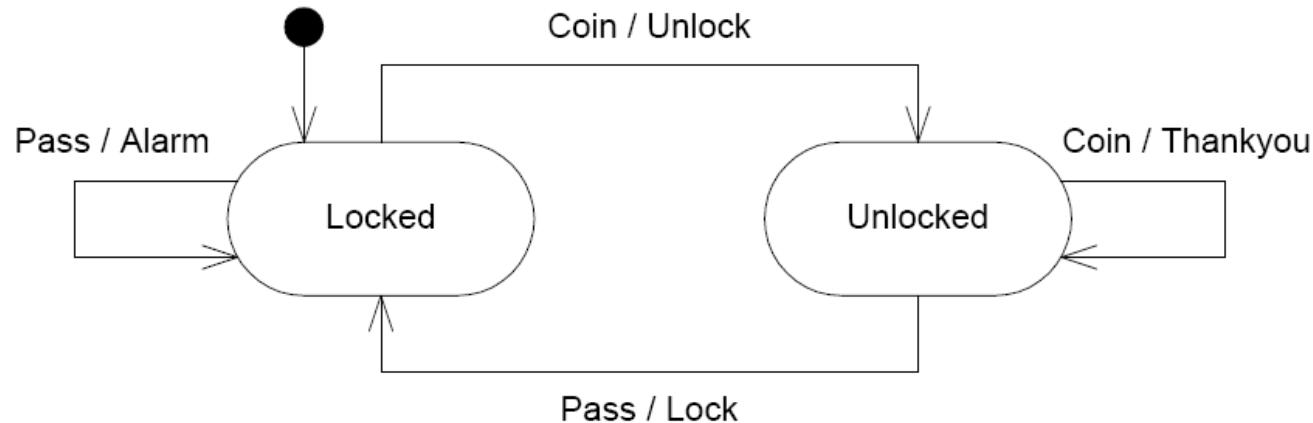
Assuma que a combinação para desbloquear o cadeado é 4-9-1

ENTRADA	A (início)	B (primeiro)	C (prim&seg)
4	B, lock	A, lock	A, lock
9	A, lock	C, lock	A, lock
1	A, lock	A, lock	A, unlock
default	A, lock	A, lock	A, lock

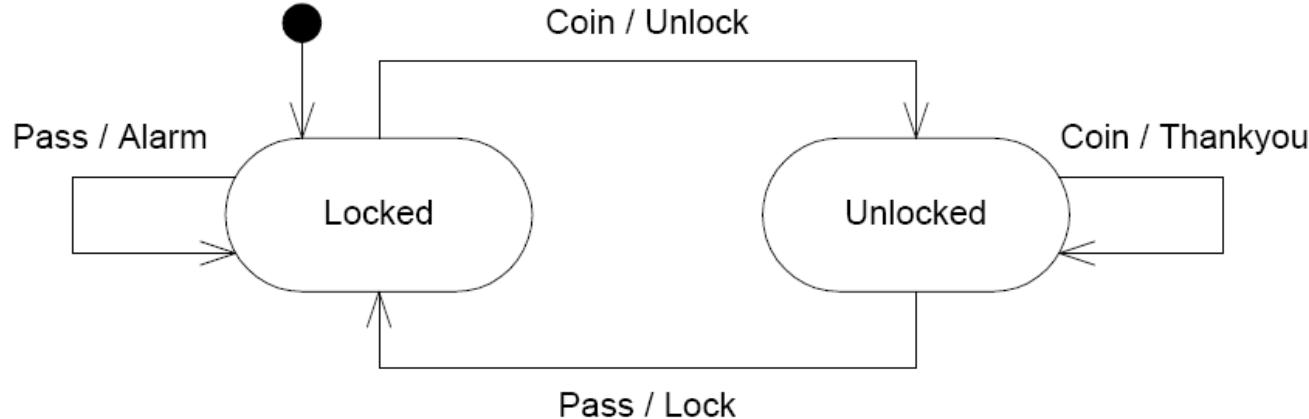
Para uma dada combinação de entrada e estado atual, cada célula mostra o próximo estado e ações/saídas

Por exemplo, se estiver no estado “A” e leu o valor “4”, vai para o estado “B” e mantém o cadeado trancado

Codificação de Máquina de Mealy



Codificação de Máquina de Mealy



$$Q = \{\text{Locked}, \text{Unlocked}\}$$

$$\Sigma = \{\text{Pass}, \text{Coin}\}$$

$$\delta(\text{Locked}, \text{Pass})=\text{Locked}; \quad \delta(\text{Locked}, \text{Coin})=\text{Unlocked};$$

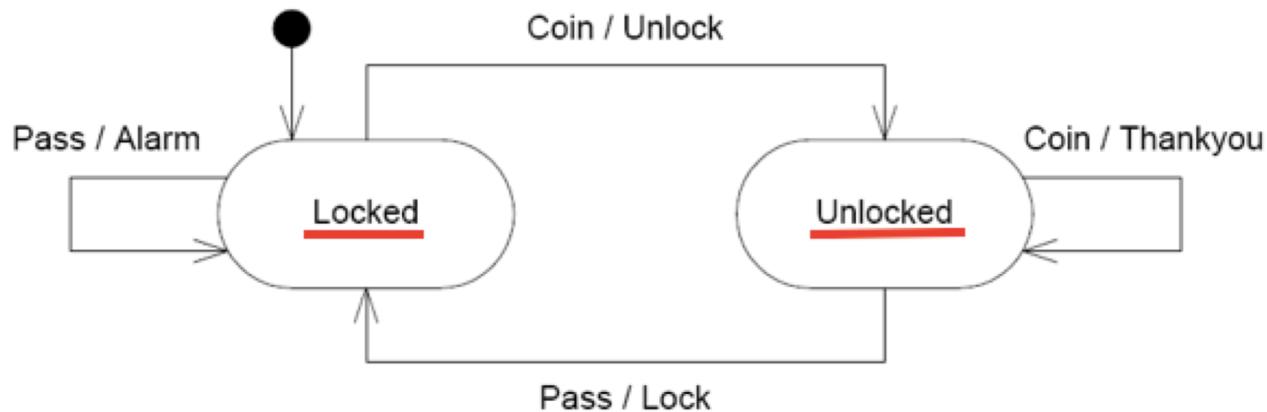
$$\delta(\text{Unlocked}, \text{Pass})=\text{Locked}; \quad \delta(\text{Unlocked}, \text{Coin})=\text{Unlocked};$$

$$q_0 = \text{Locked}$$

$$\Delta = \{\text{Lock}, \text{Unlock}, \text{Alarm}, \text{Thankyou}\}$$

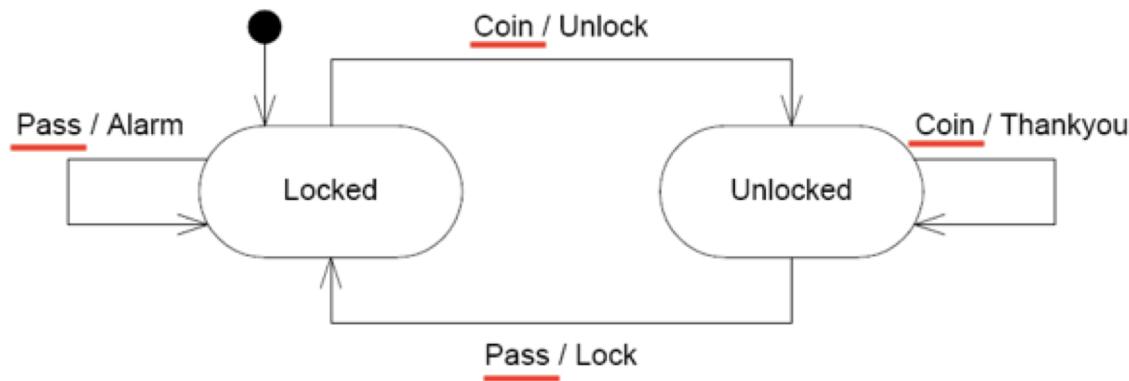
$$\sigma(\text{Locked}, \text{Pass})=\text{Alarm}; \quad \sigma(\text{Locked}, \text{Coin})=\text{Unlock};$$

$$\sigma(\text{Unlocked}, \text{Pass})=\text{Lock}; \quad \sigma(\text{Unlocked}, \text{Coin})=\text{Thankyou};$$



Entrada	Locked	Unlocked
Pass	Alarm(), Locked	Lock(), Locked
Coin	Unlock(), Unlocked	Thankyou(), Unlocked

```
enum State {Locked, Unlocked};
```

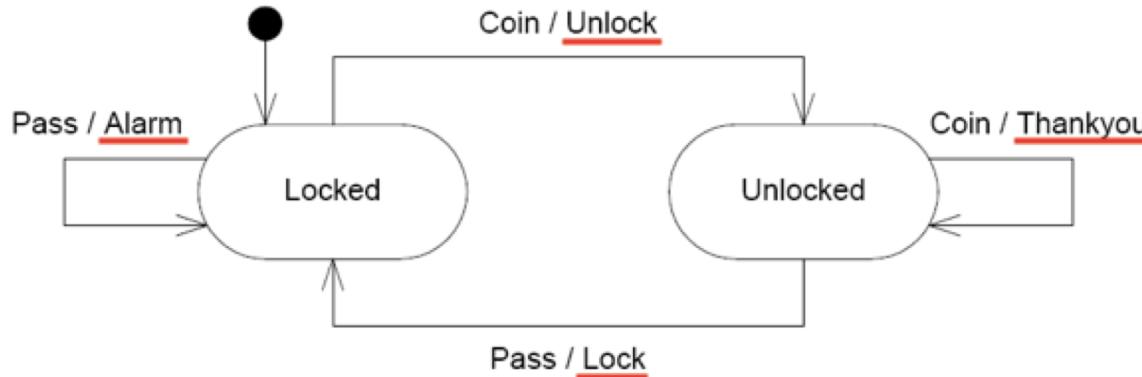


Entrada	Locked	Unlocked
Coin	Unlock(), Locked	Thankyou(), Unlocked
Pass	Alarm(), Locked	Lock(), Locked

```

enum State {Locked, Unlocked};
enum Event {Pass, Coin};

```



Entrada	Locked	Unlocked
Pass	Alarm(), Locked	Lock(), Locked
Coin	Unlock(), Unlocked	Thankyou(), Unlocked

```

enum State {Locked, Unlocked};
enum Event {Pass, Coin};

```

```

void Unlock();
void Lock();
void Thankyou();
void Alarm();

```

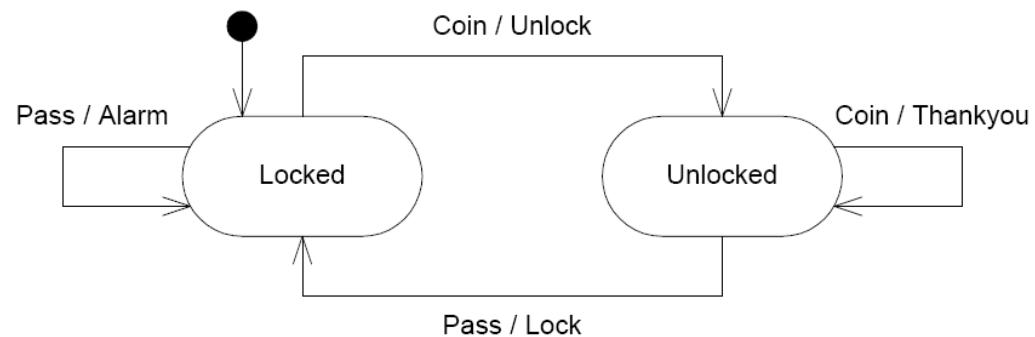
```
enum state {Locked, Unlocked};  
enum event {Pass, Coin};  
  
void Unlock(); void Lock();  
void Thankyou(); void Alarm();  
  
state s = Locked;  
do forever {  
    read_event(e);  
    switch(s) {  
        case Locked:  
            . . .  
            break;  
        case Unlocked:  
            . . .  
            break;  
    } // switch (s)  
} // do forever
```

```

switch(s) {
    case Locked:
        switch(e) {
            case Coin:
                s = Unlocked;
                Unlock();
                break;
            case Pass:
                Alarm();
                break;
        } // switch (e)
        break;
    case Unlocked:
        switch(e) {
            case Coin:
                Thankyou();
                break;
            case Pass:
                s = Locked;
                Lock();
                break;
        } // switch (e)
        break;
} // switch (s)

```

Entrada	Locked	Unlocked
Pass	Alarm(), Locked	Lock(), Locked
Coin	Unlock(), Unlocked	Thankyou(), Unlocked



Referências

- Wagner, F (2006). Modeling Software with Finite State Machines: A Practical Approach. Auerbach Publications, 2006, ISBN 0-8493-8086-3
- John Hopcroft e Jeffrey Ullman (2006). Introduction to Automata Theory, Languages, and Computation. Pearson. 3 edition. ISBN: 978-0321455369
- <https://www.embedded.com/using-finite-state-machines-to-design-software/>

**SÓ EXISTEM 10 TIPOS
DE PESSOAS NO MUNDO.**



@cursos.cae

**AS QUE SABEM BINÁRIOS
E AS QUE NÃO SABEM.**