

ACH2028 - Qualidade de Software

Aula 11 - Teste Baseado em Estados

Prof. Marcelo Medeiros Eler

marceloeler@usp.br

Objetivos

- Apresentar a técnica de teste baseada em estados
- Propor exercícios para exercitar os conceitos da técnica

Material utilizado

- Parte do material desta aula foi criado com base no livro Software Testing: From Theory to Practice, do Prof. Maurício Aniche da Delft University of Technology.
 - <https://sttp.site/>

Como encontrar todas as falhas de um software?

- Um conjunto de casos de teste com **qualidade ótima** é capaz de revelar todas as falhas de um software
- Para encontrar todas as falhas e garantir que o software está livre de defeitos é necessário **testá-lo em todas as condições** e com **todos os dados de entrada possíveis**

Como encontrar todas as falhas de um software?

- Entretanto, as condições, os cenários, os dados de entrada e suas combinações tendem a ser **infinitos ou muito grandes**
- Portanto, criar um conjunto de casos de teste ótimo é geralmente **impossível ou impraticável**

Uma pergunta importante neste cenário

- Como criar um conjunto de casos de teste que seja:
 - Finito e factível?
 - Capaz de revelar o maior número de falhas perceptíveis?
 - Capaz de revelar as falhas mais críticas ou relevantes do software?
- Esta pergunta esconde algumas outras:
 - Como seleccionar cenários/dados para criar bons casos de teste?
 - Quantos testes são necessários para testar um software ou parte dele?
 - Como saber se os testes criados são suficientes?

Técnicas e critérios de teste

- Um bom caso de teste é aquele que tem uma alta probabilidade de revelar uma falha ainda não descoberta
- Portanto, técnicas e critérios de teste foram propostos para orientar o testador na tarefa de derivar os casos de teste com base nos artefatos do projeto (especificação, código, modelos, etc):
 - Teste Adhoc
 - Teste Exploratório
 - Teste Funcional ou Caixa-Preta
 - Teste Estrutural ou Caixa-Branca
 - Teste Baseado em Erros

Modelos

Oferecem uma visão simplificada do artefato testado

Preservam as propriedades/características do artefato testado

Fornecem uma visão da estrutura e/ou comportamento do artefato testado

Podem ser analisados/verificados de forma sistemática/automática

Podem vir:

- da especificação (negócios/comportamento requerido)
- do código (desenvolvimento/comportamento implementado)

Modelos

Oferecem uma visão simplificada do artefato testado

Preservam as propriedades/características do artefato testado

Fornecem uma visão da estrutura e/ou comportamento do artefato testado

Podem ser analisados/verificados de forma sistemática/automática

Podem vir:

- **da especificação (negócios/comportamento requerido)**
- do código (desenvolvimento/comportamento implementado)

Teste baseado em modelos

Utiliza modelos para derivar casos de teste

Enquadra-se dentro da técnica funcional ou caixa-preta

Exemplos de modelos que podem ser utilizados:

- Tabelas de decisão (já abordadas nas aulas anteriores)
- Diagramas de atividades (UML)
- Diagramas de sequência (UML)
- Máquinas de Estados

Teste baseado em modelos

Utiliza modelos para derivar casos de teste

Enquadra-se dentro da técnica funcional ou caixa-preta

Exemplos de modelos que podem ser utilizados:

- Tabelas de decisão (já abordadas nas aulas anteriores)
- Diagramas de atividades (UML)
- Diagramas de sequência (UML)
- **Máquina de Estados**

Máquina de Estados Finitos

Uma MEF é um modelo que descreve o comportamento de um sistema por meio da descrição de seus estados e transições que existem entre eles

Dado um estado atual do sistema e uma entrada (estímulo/evento), é possível determinar qual estado o sistema vai assumir

Possíveis representações:

- Textual
- Tabular
- Gráfica

Máquina de Estados Finitos (MEF)

Representação textual

- Estados = {A, B, C}
- Saídas = {S1, S2}
- Entradas = {E1, E2, E3}
- Transição 1 (A, E1) = (B,S1)
- Transição 2 (B, E2) = (C,S2)
- Transição 3 (C, E3) = (A,S1)

Máquina de Estados Finitos (MEF)

Representação textual

- Estados = {Tela Login, Menu Principal, Emissão de Extrato}
- Saídas = {"Login realizado", "", "Volte sempre"}
- Entradas = {"login/senha", "botão extrato", "sair"}
- Transição 1 (Tela Login, login/senha) = (Menu Principal, "login realizado")
- Transição 2 (Menu principal, botão extrato) = (Emissão de Extrato, "")
- Transição 3 (Emissão de Extrato, sair) = (Tela Login, "volte sempre")

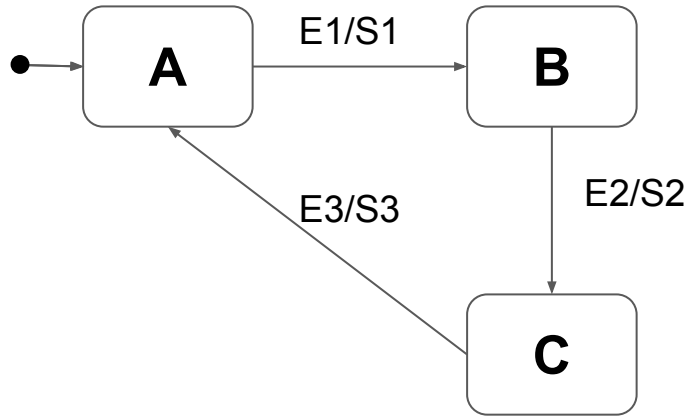
Máquina de Estados Finitos (MEF)

Representação tabular

	E1	E2	E3
A	B,S1		A,S1
B		C,S2	
C		D,S1	A,S1

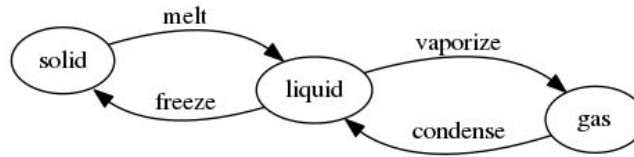
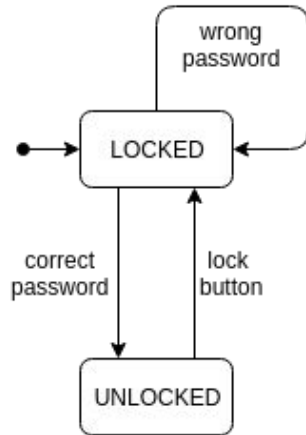
Máquina de Estados Finitos (MEF)

Representação gráfica



Máquina de Estados Finitos (MEF)

Representação gráfica



Máquina de Estados Finitos (MEF)

O comportamento da máquina pode ser detalhado utilizando-se condições

Uma mesma entrada/evento pode levar o sistema a assumir diferentes estados dependendo das condições

Nesses casos é preciso utilizar transições condicionais, que só são executadas se o evento acontecer e a condição for satisfeita (condições de guarda)

Máquina de Estados Finitos (MEF)

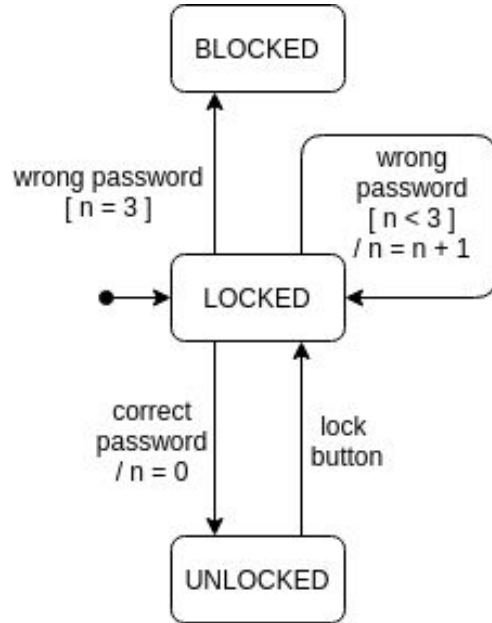
As condições associadas às transições geralmente dependem de um valor específico da máquina

Esses valores podem ser modificados por ações executadas quando uma específica transição acontece

Notação:

- evento [condição] / ação

Máquina de Estados Finitos (MEF)



Máquina de Estados Finitos (MEF)

Exemplo de uma Pilha (estrutura de dados)

Operações:

- push
- pop

Estados:

- Vazia
- Cheia
- Não-vazia (com itens)

Máquina de Estados Finitos (MEF)

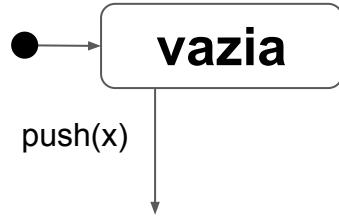
vazia

Máquina de Estados Finitos (MEF)

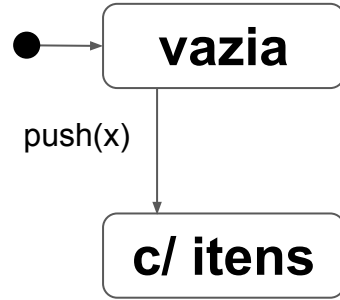
Estado inicial



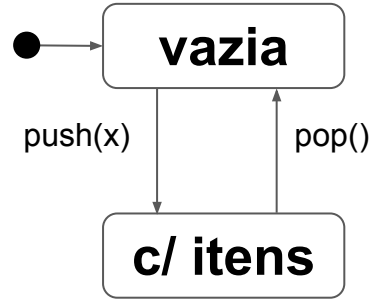
Máquina de Estados Finitos (MEF)



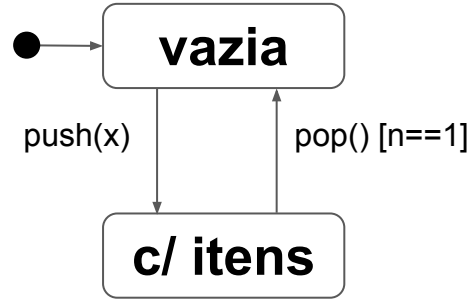
Máquina de Estados Finitos (MEF)



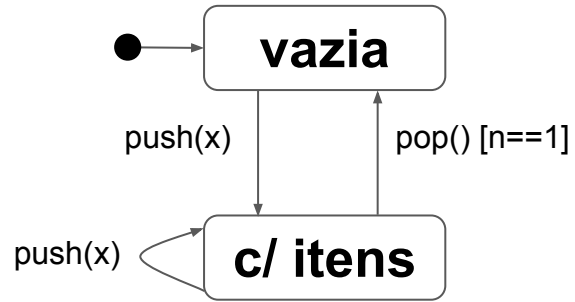
Máquina de Estados Finitos (MEF)



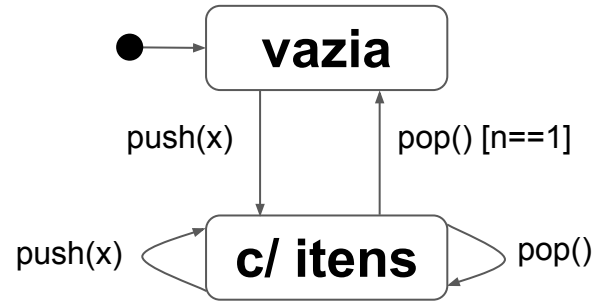
Máquina de Estados Finitos (MEF)



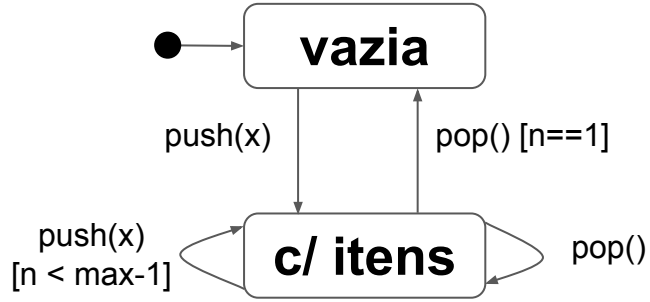
Máquina de Estados Finitos (MEF)



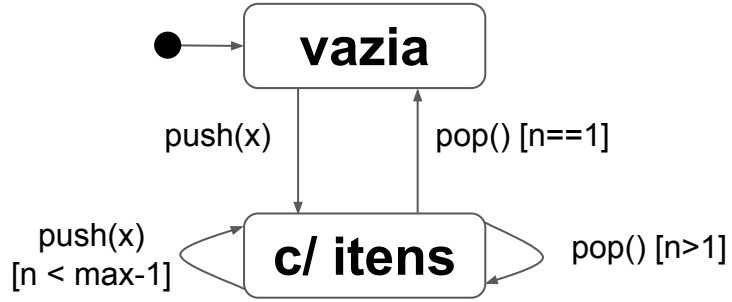
Máquina de Estados Finitos (MEF)



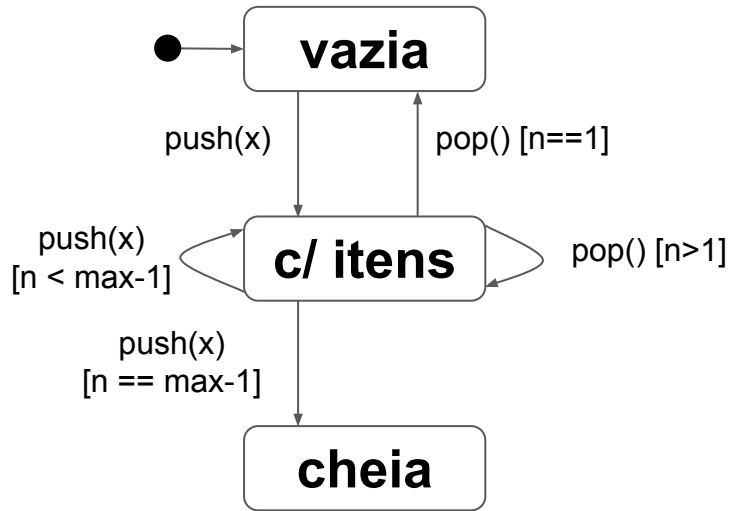
Máquina de Estados Finitos (MEF)



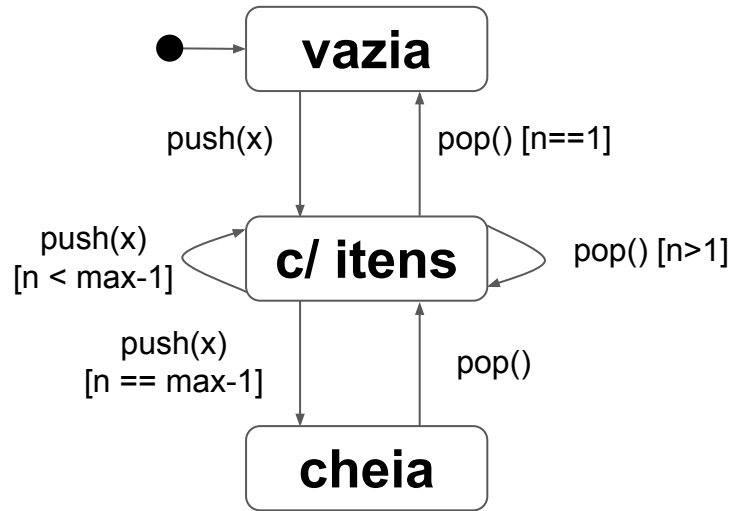
Máquina de Estados Finitos (MEF)



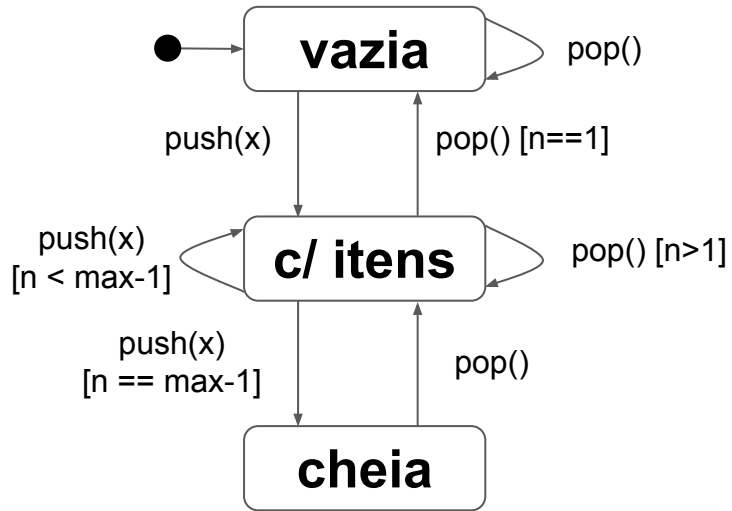
Máquina de Estados Finitos (MEF)



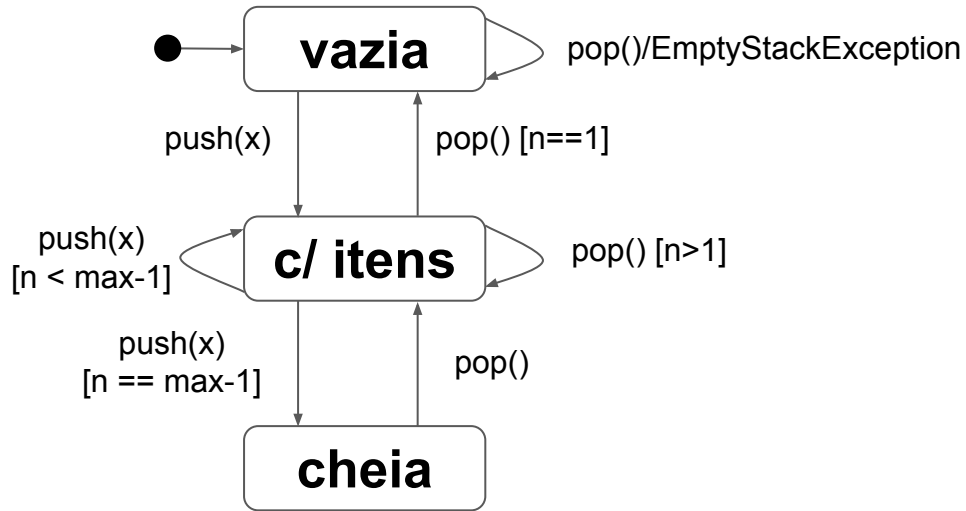
Máquina de Estados Finitos (MEF)



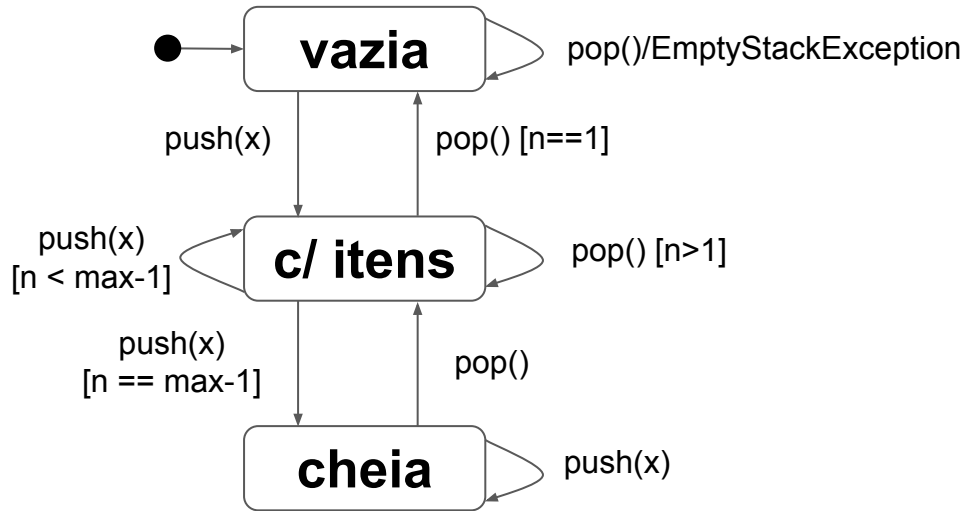
Máquina de Estados Finitos (MEF)



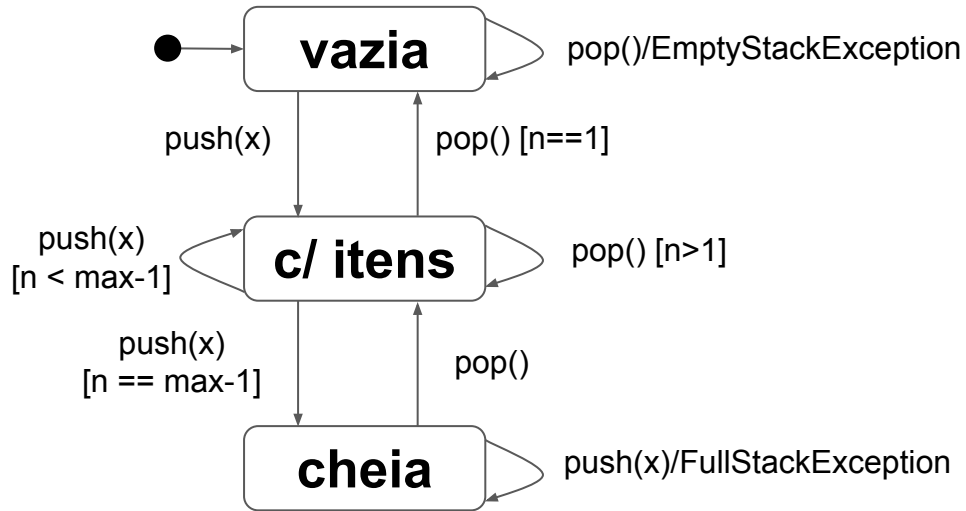
Máquina de Estados Finitos (MEF)



Máquina de Estados Finitos (MEF)



Máquina de Estados Finitos (MEF)



Teste baseado em MEFs

Após modelar o comportamento do sistema que se deseja testar, as MEFs são utilizadas para derivar casos de teste

Testes a serem considerados:

- Transições para estados incorretos
- Condições e ações
- Consistência do comportamento do sistema em determinado estado (deveria apresentar sempre o mesmo comportamento)

Teste baseado em MEFs

Crítérios a serem cobertos:

- Cobertura de estados: cada estado deve ser alcançado pelo menos uma vez
- Cobertura de transições: cada transição deve ser exercitada pelo menos uma vez

Caminhos na MEF não necessariamente configuram um objetivo de cobertura, mas são usados para derivar casos de teste

Teste baseado em MEFs

Para testar os estados:

- Leve o sistema a um determinado estado por meio das transições
- Verifique se ele se encontra naquele estado específico

Para testar uma transição, alguns passos podem ser necessários:

- Leve o sistema para o estado a partir do qual uma transição será testada
- Verifique se o sistema está naquele estado definido
- Acione o evento
- Se houver uma ação, verifique se a ação ocorreu (teste o efeito a da ação no sistema)
- Verifique se o sistema está no estado para o qual a transição deveria levá-lo

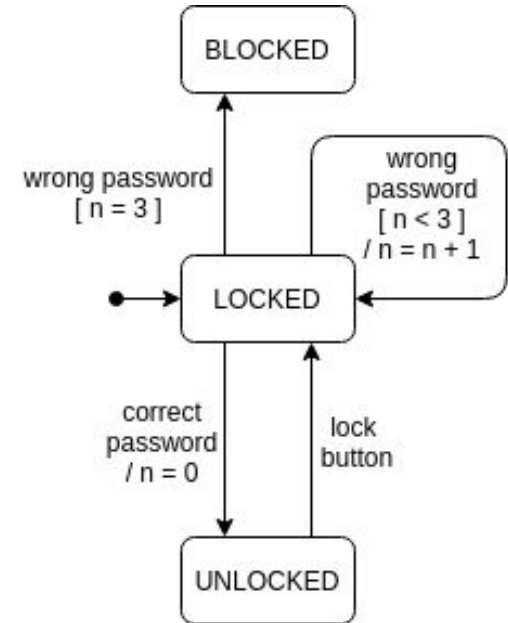
Teste baseado em MEFs

Cobertura de estados:

- Verificar se o celular está no estado LOCKED quando iniciado
- Inserir a senha correta e verificar se ele vai para UNLOCKED
- Inserir uma senha incorreta 4 vezes e verificar se vai para BLOCKED

Cobertura de transições:

- A cobertura de estados cobriu a maioria das transições
- Leve o celular ao estado UNLOCKED ao fornecer a senha correta
- Acione o botão “lock” para ir para o estado LOCKED



Teste baseado em MEFs

Caminhos e árvores de transição

Além da cobertura de estados e transições, é possível também testar combinações de transições (caminhos)

Testar todos os caminhos em uma máquina de estados pode ser impraticável principalmente quando existem laços

- Exemplo de celular: LOCKED -> UNLOCKED -> LOCKED -> UNLOCKED -> LOCKED -> -> UNLOCKED

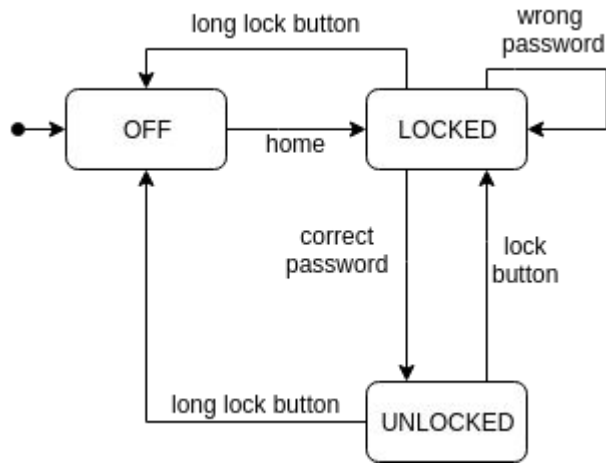
Caminhos e árvores de transição

Uma heurística comum é que cada laço deve ser executado pelo menos uma vez

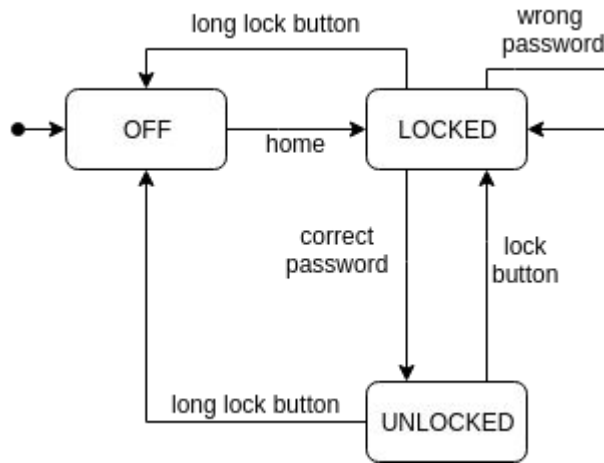
Para isso é possível criar caminhos por meio de árvores de transição seguindo esses procedimentos:

- O nó da árvore é o estado inicial da máquina
- Para cada nó N do nível mais baixo da árvore, faça o seguinte:
 - Se N corresponde a um estado que ainda não foi explorado, adicione a este nó filhos que correspondem aos estados para os quais N tem alguma transição
 - Se N corresponde a um estado que já foi explorado, não adicione nenhum filho a ele
- Para os estados que se repetem, adicione um número para diferenciá-los dos nós que já apareceram na árvore

Caminhos e árvores de transição

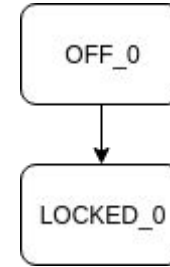
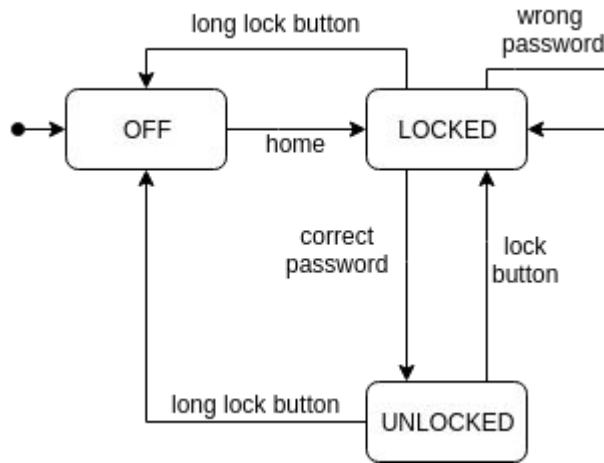


Caminhos e árvores de transição

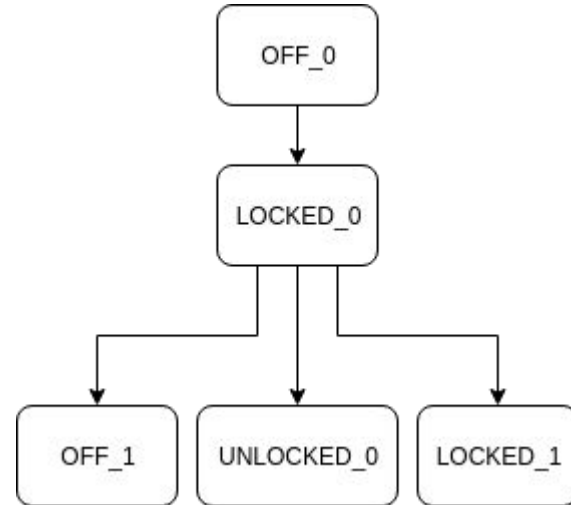
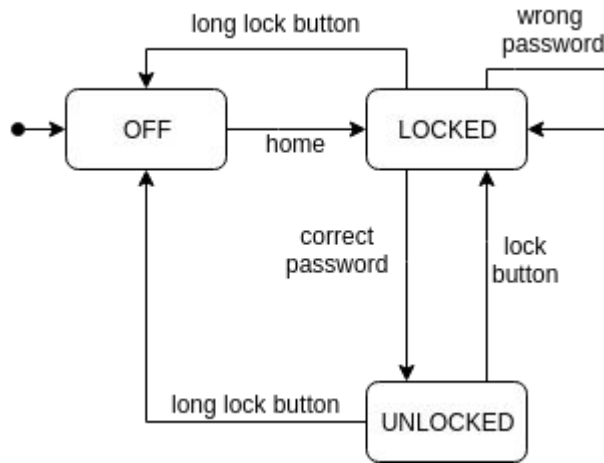


OFF_0

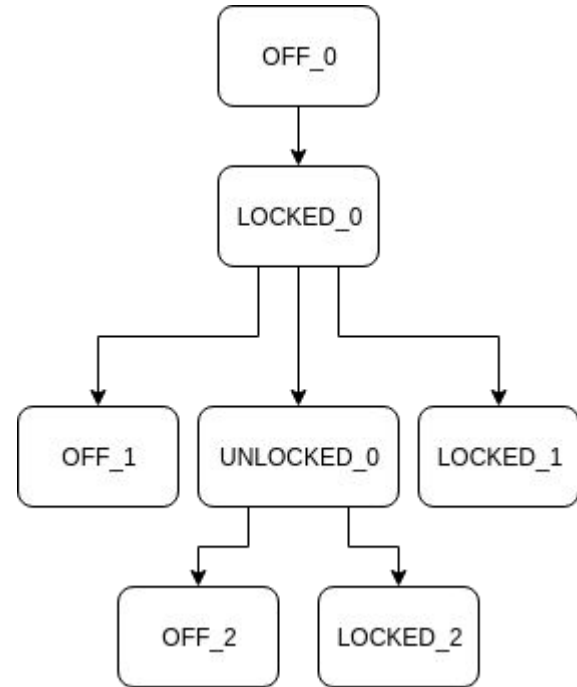
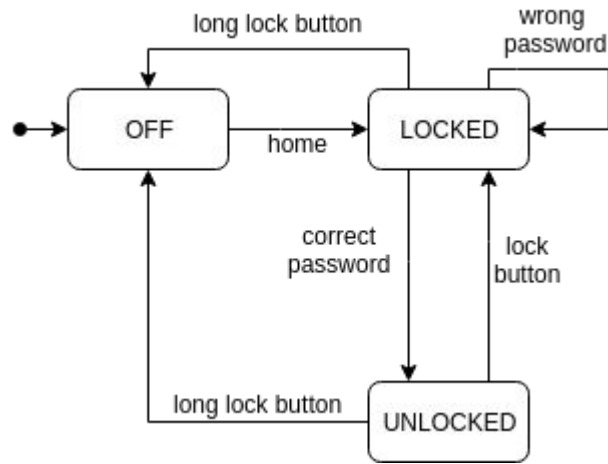
Teste baseado em MEFs



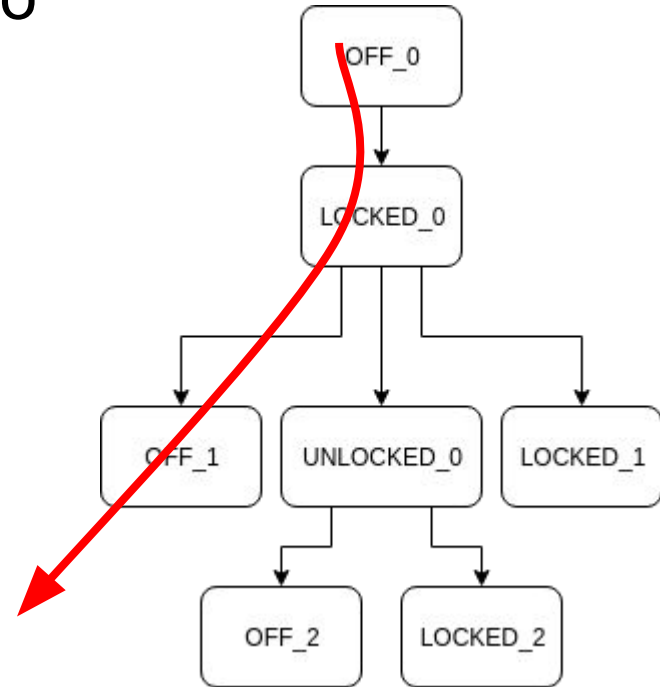
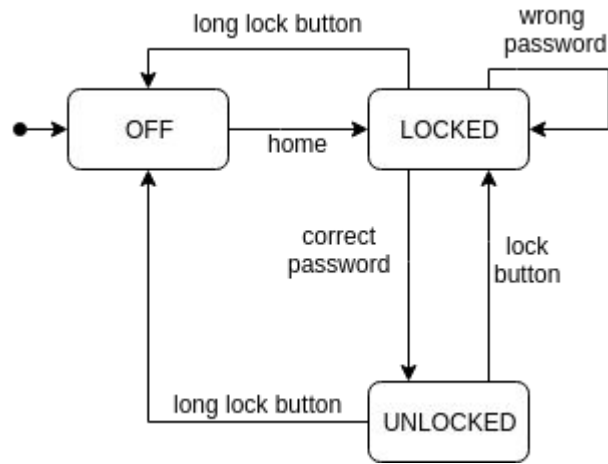
Caminhos e árvores de transição



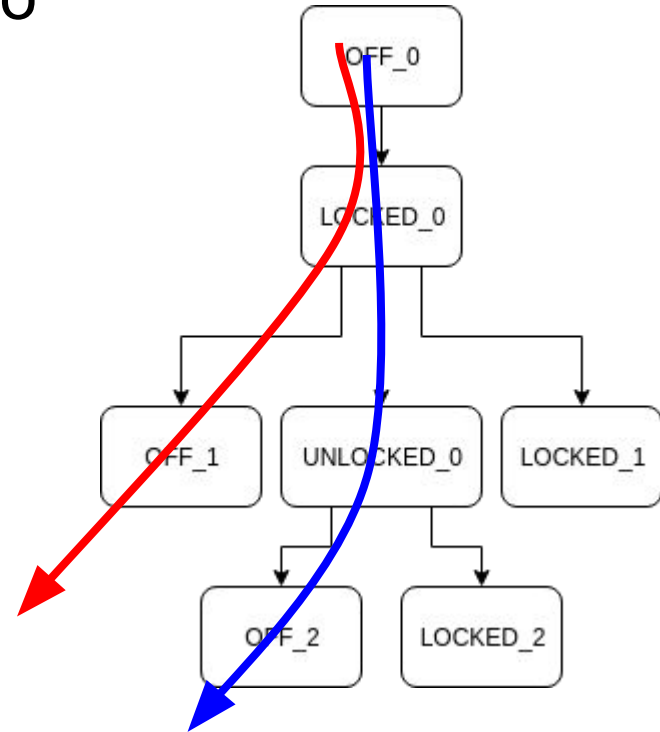
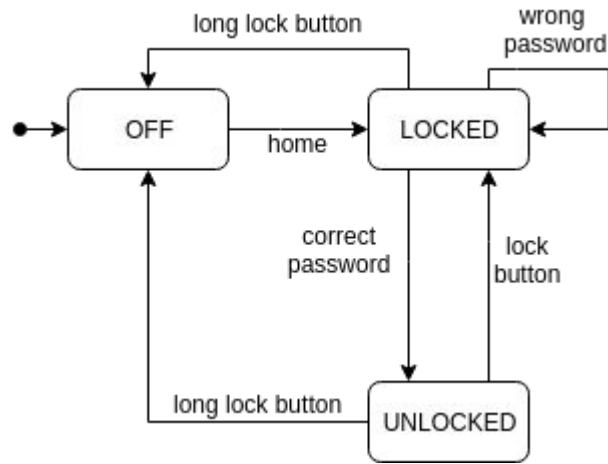
Caminhos e árvores de transição



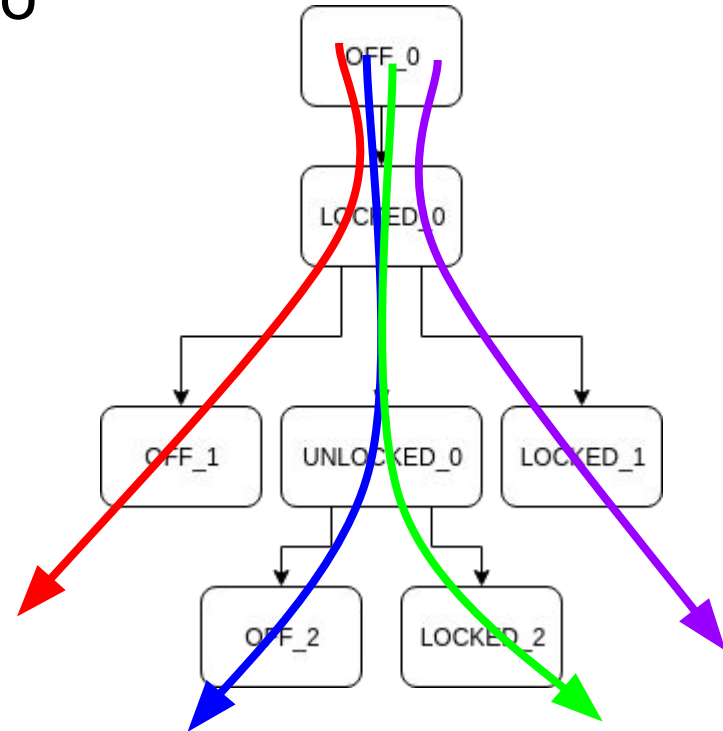
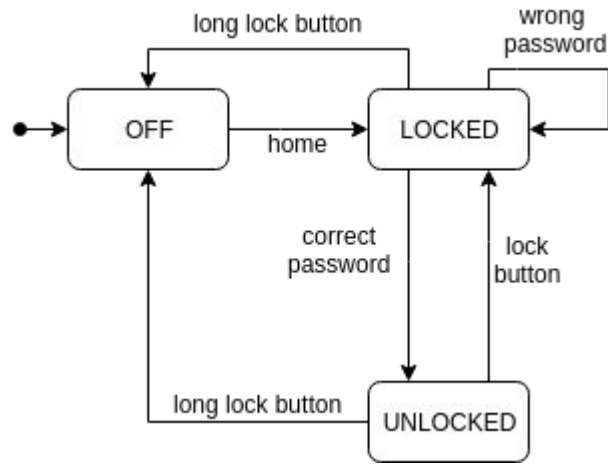
Caminhos e árvores de transição



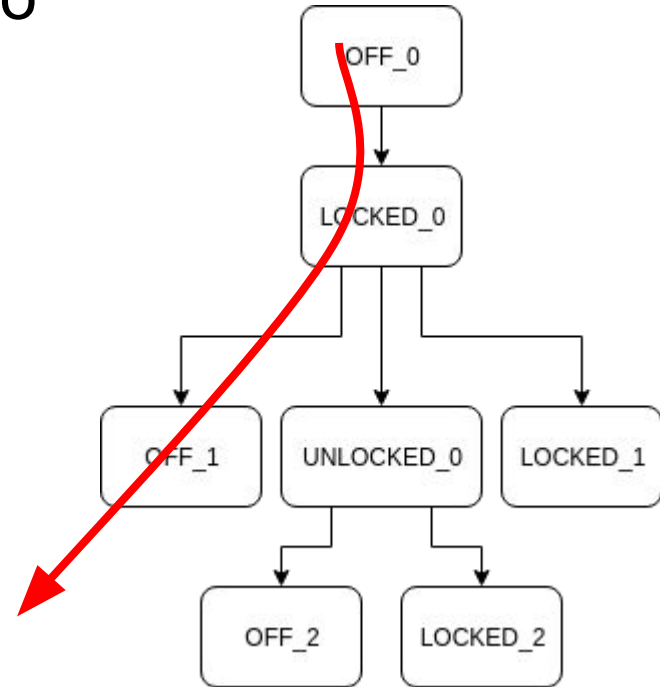
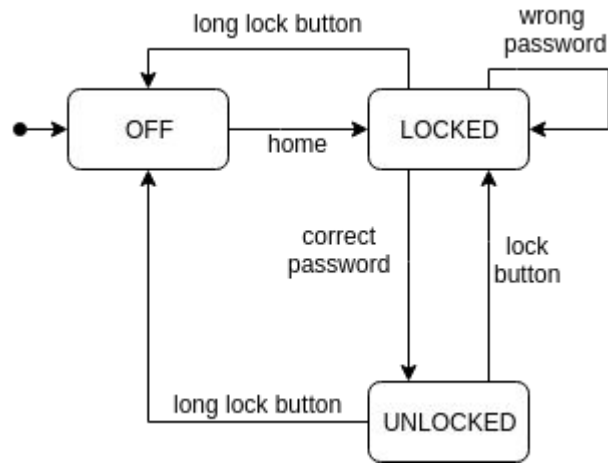
Caminhos e árvores de transição



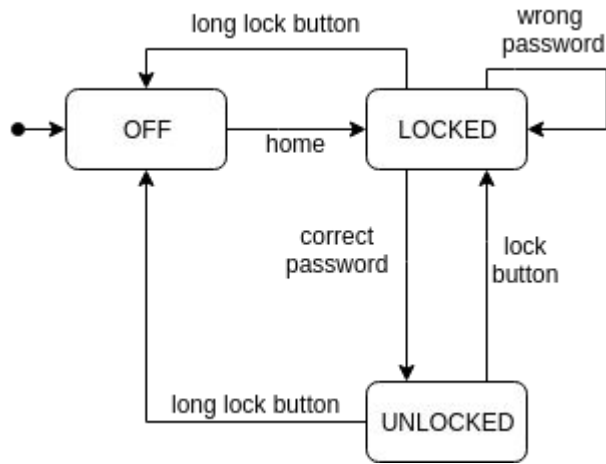
Caminhos e árvores de transição



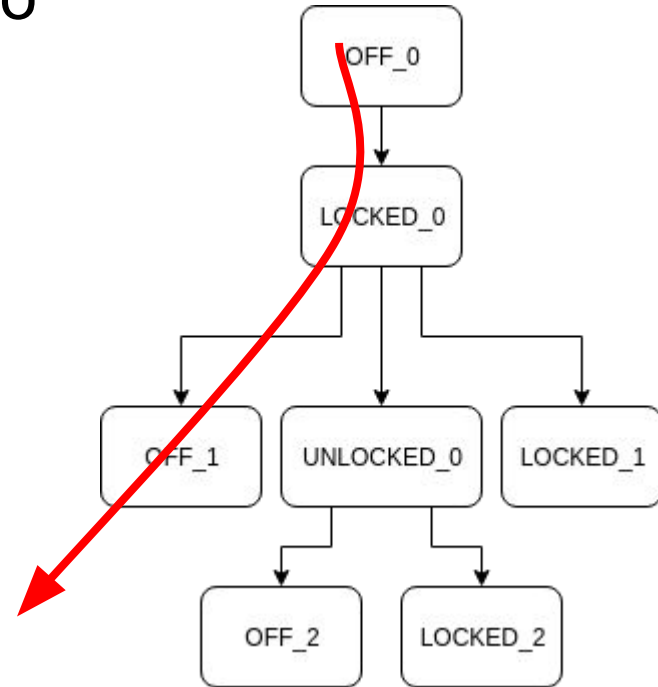
Caminhos e árvores de transição



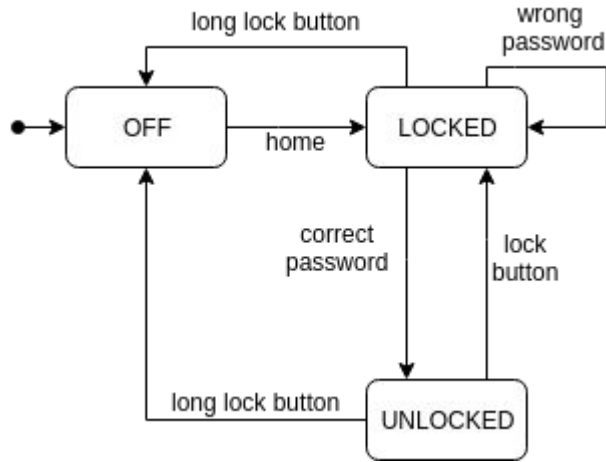
Caminhos e árvores de transição



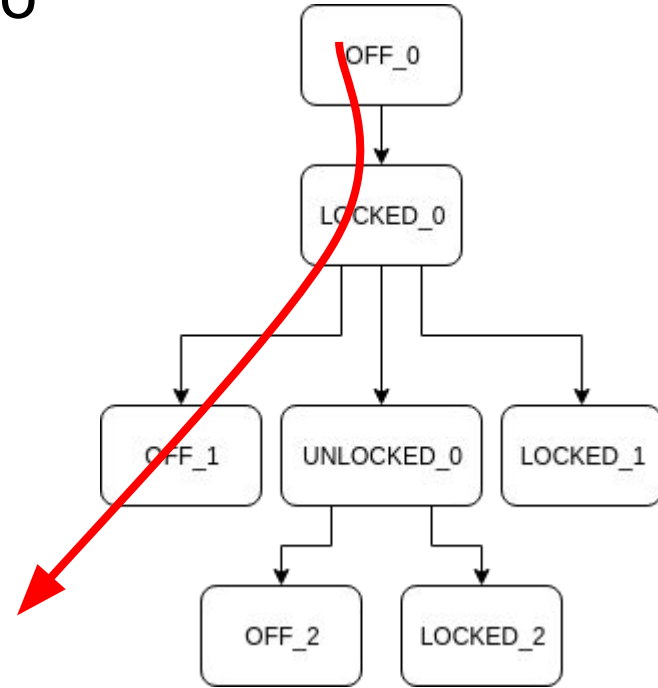
Verifique se o estado atual (inicial) é OFF



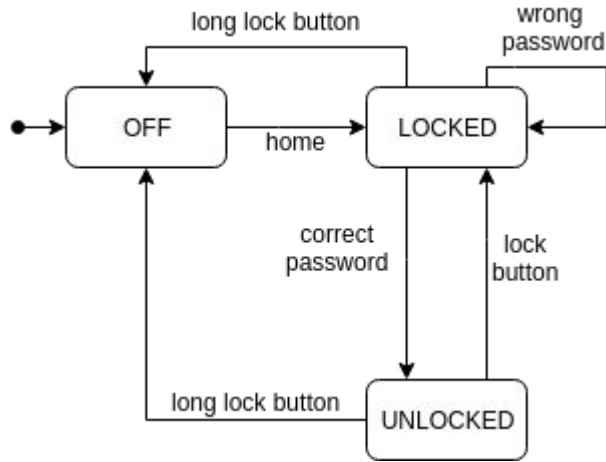
Caminhos e árvores de transição



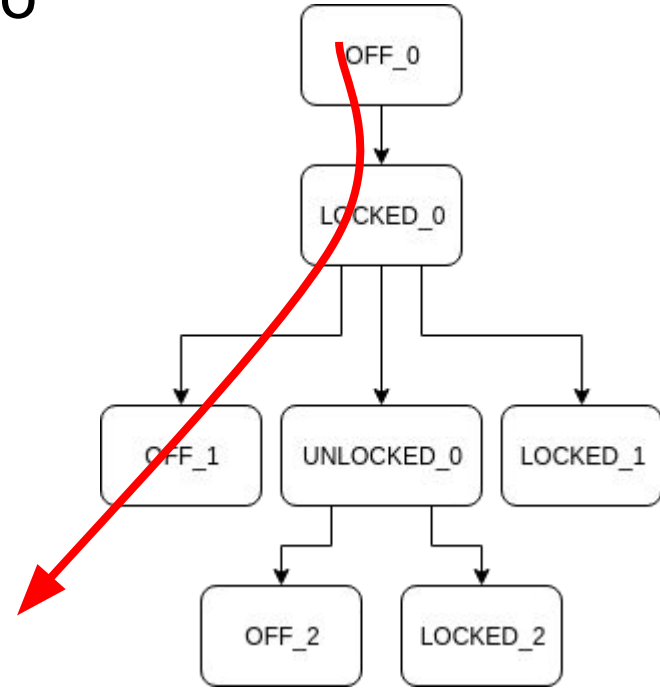
Verifique se o estado atual (inicial) é OFF
Aperte a tecla *home*



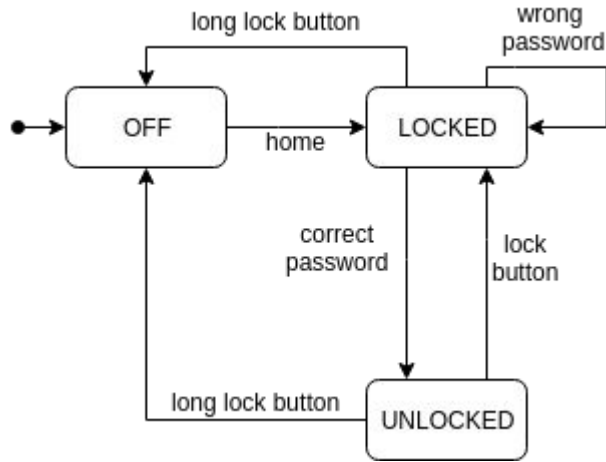
Caminhos e árvores de transição



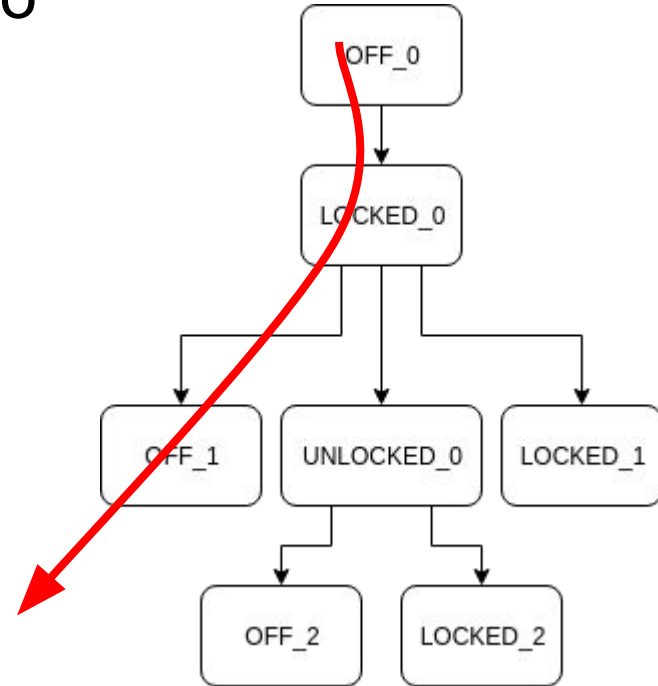
Verifique se o estado atual (inicial) é OFF
Aperte a tecla *home*
Verifique se o estado atual é LOCKED



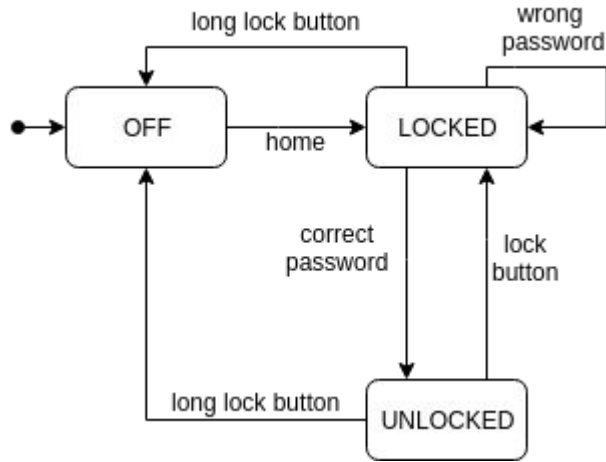
Caminhos e árvores de transição



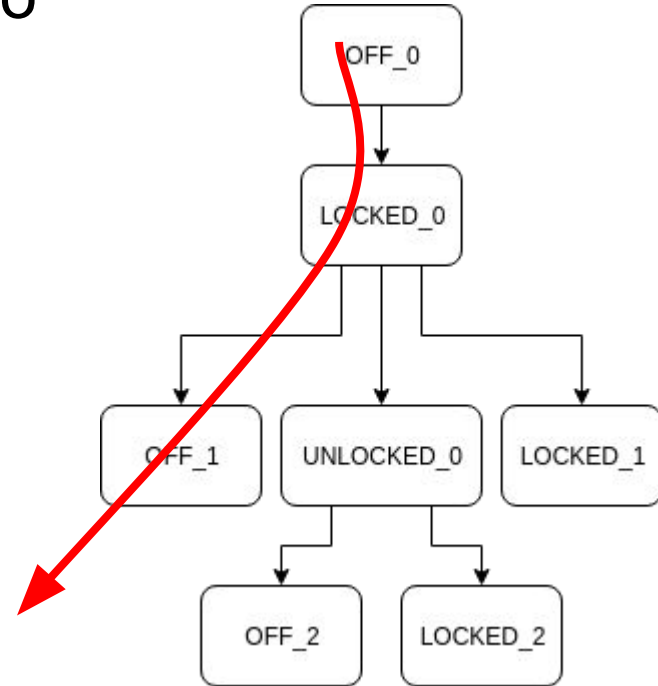
Verifique se o estado atual (inicial) é OFF
Aperte a tecla *home*
Verifique se o estado atual é LOCKED
Aperte a tecla *long lock button*



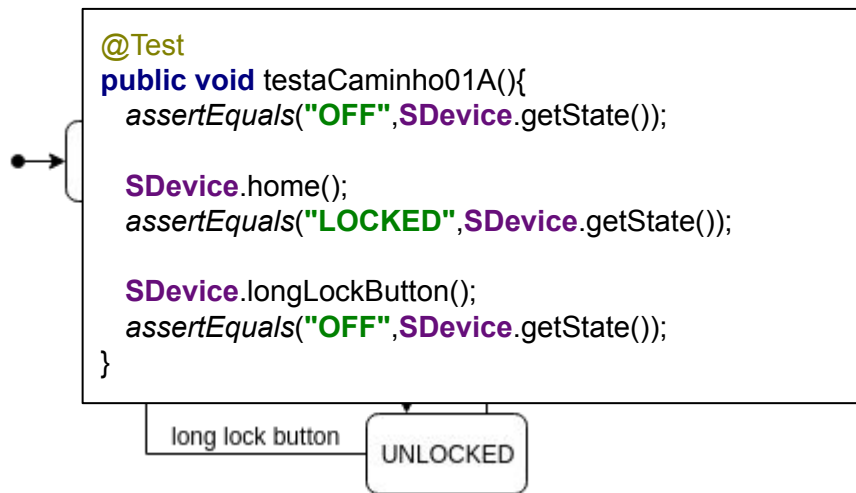
Caminhos e árvores de transição



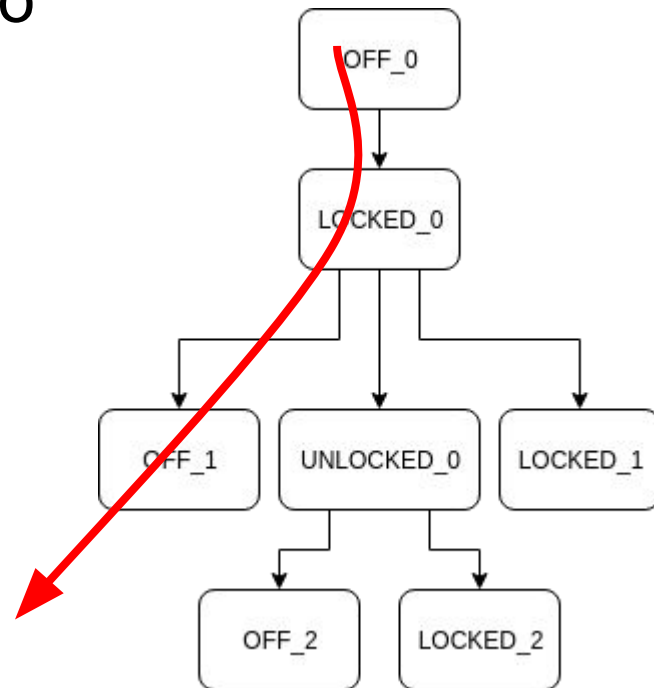
Verifique se o estado atual (inicial) é OFF
Aperte a tecla *home*
Verifique se o estado atual é LOCKED
Aperte a tecla *long lock button*
Verifique se o estado atual é o OFF



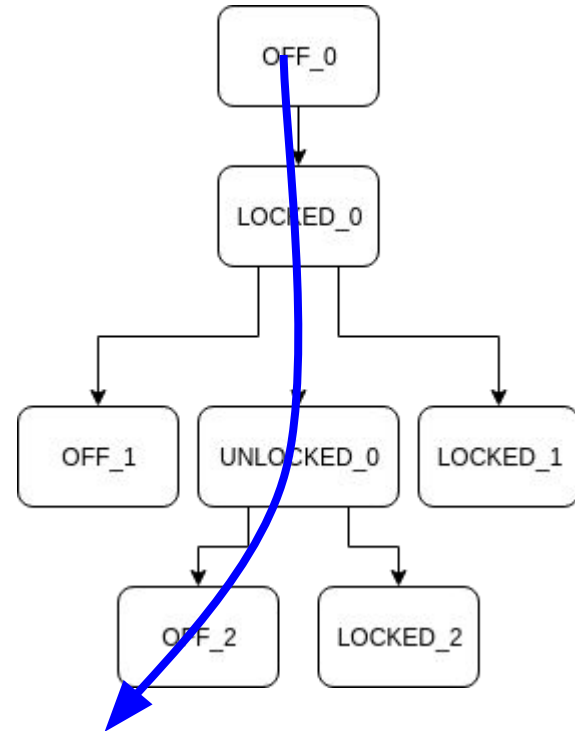
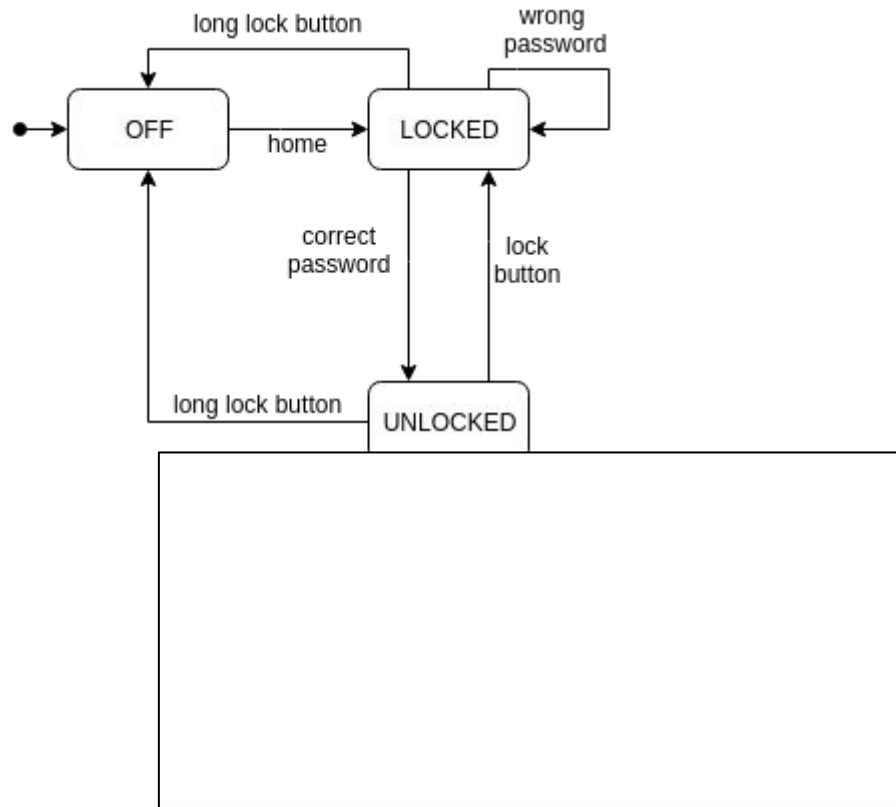
Caminhos e árvores de transição



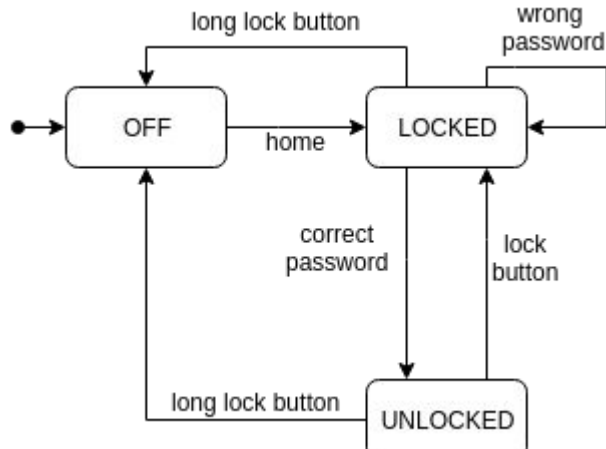
Verifique se o estado atual (inicial) é OFF
Aperte a tecla *home*
Verifique se o estado atual é LOCKED
Aperte a tecla *long lock button*
Verifique se o estado atual é o OFF



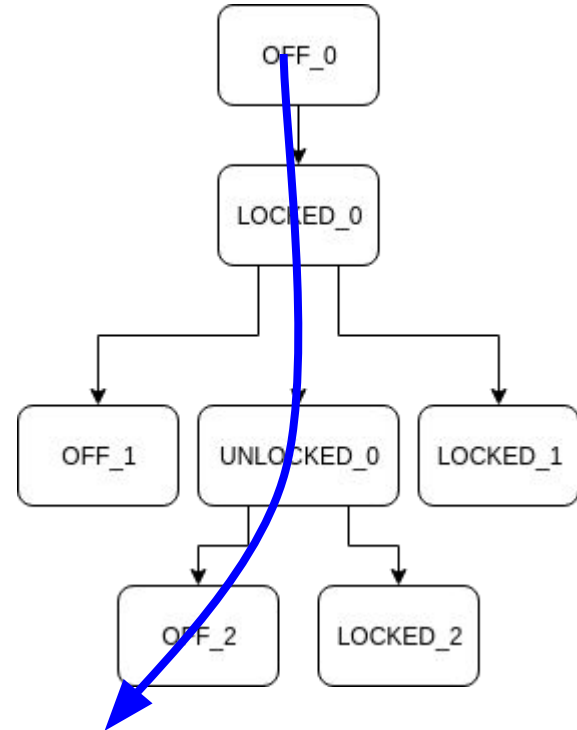
Caminhos e árvores de transição



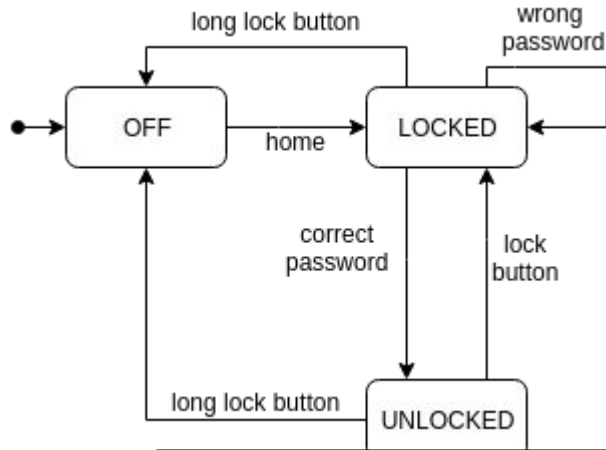
Caminhos e árvores de transição



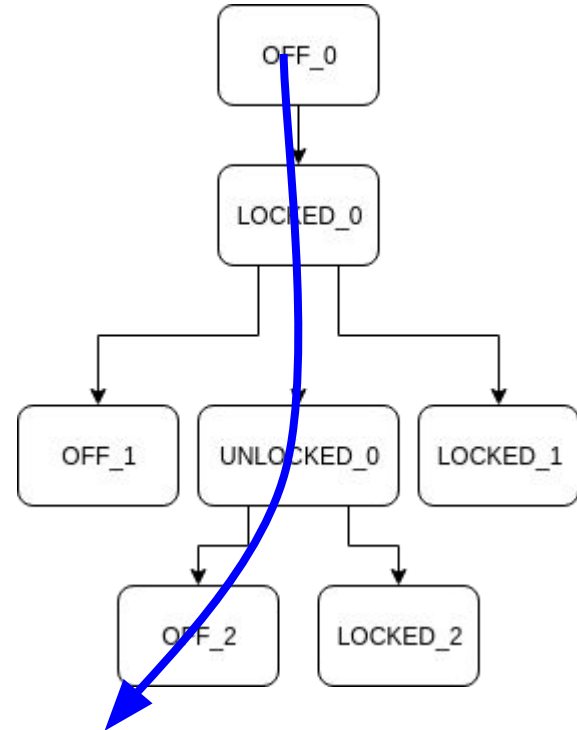
Verifique se o estado atual (inicial) é OFF



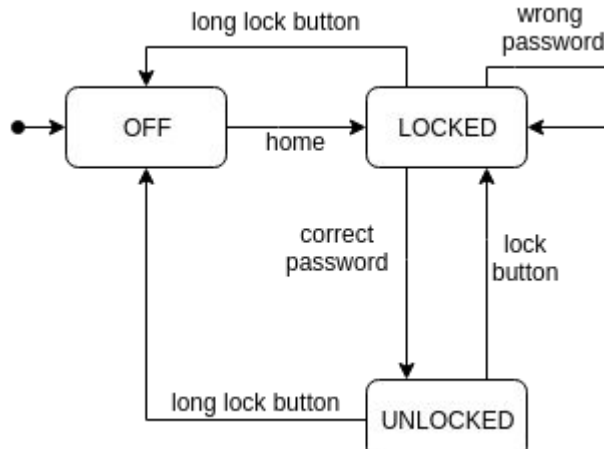
Caminhos e árvores de transição



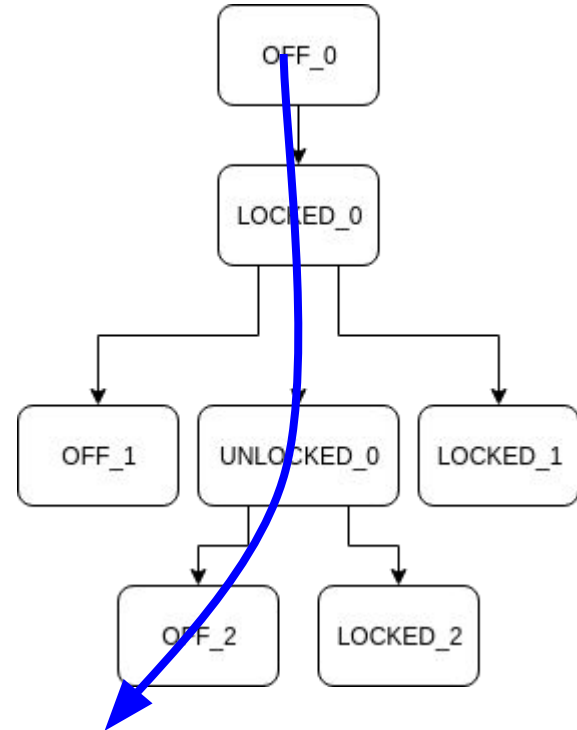
Verifique se o estado atual (inicial) é OFF
Aperte a tecla *home*



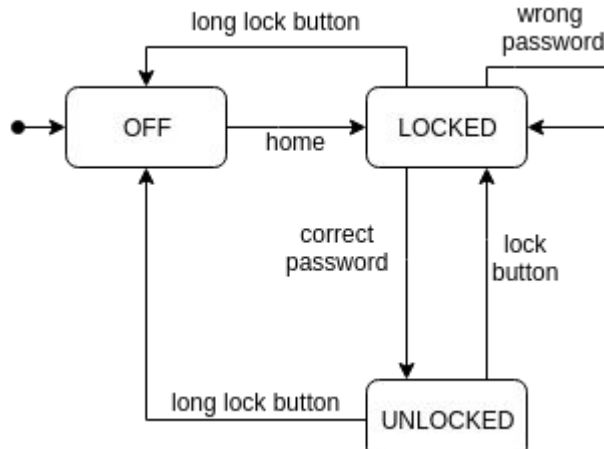
Caminhos e árvores de transição



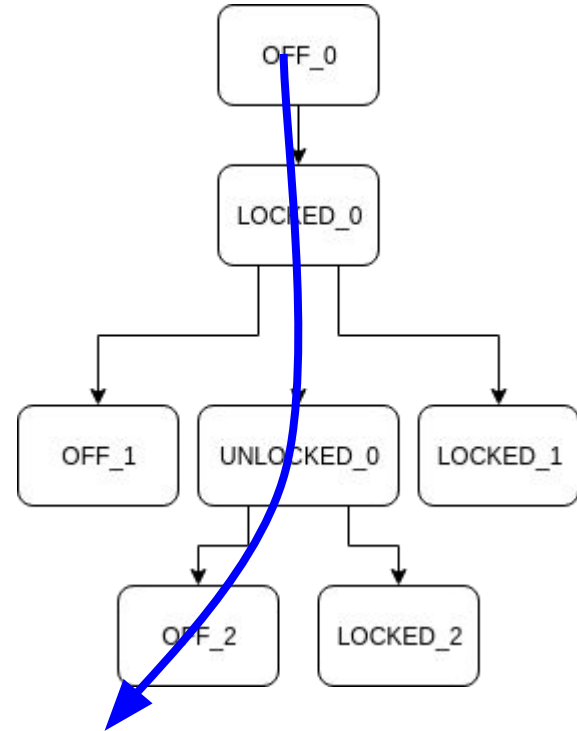
Verifique se o estado atual (inicial) é OFF
Aperte a tecla *home*
Verifique se o estado atual é LOCKED



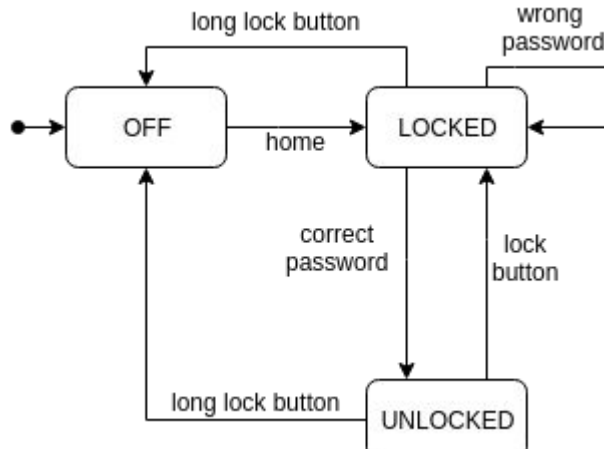
Caminhos e árvores de transição



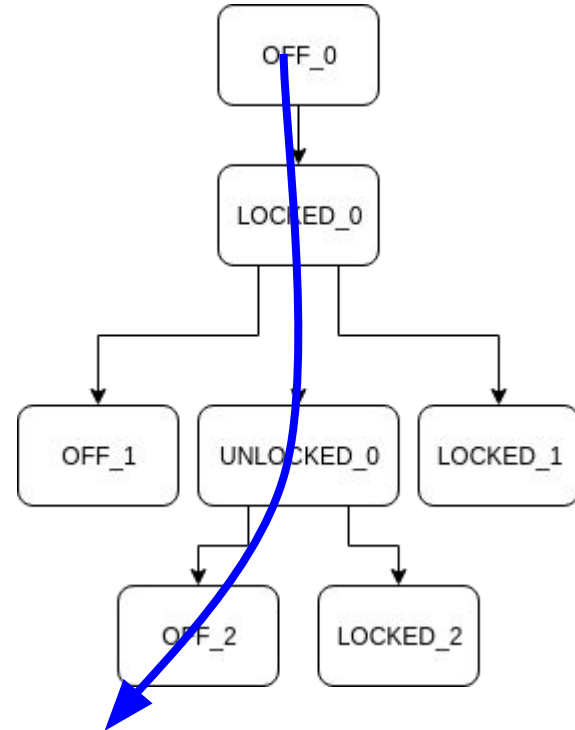
Verifique se o estado atual (inicial) é OFF
Aperte a tecla *home*
Verifique se o estado atual é LOCKED
Insira uma senha correta



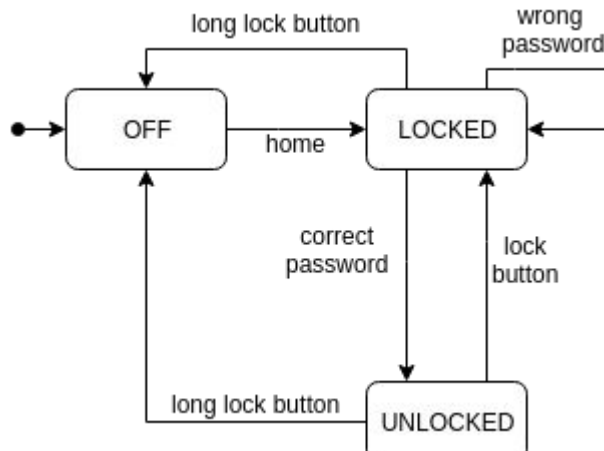
Caminhos e árvores de transição



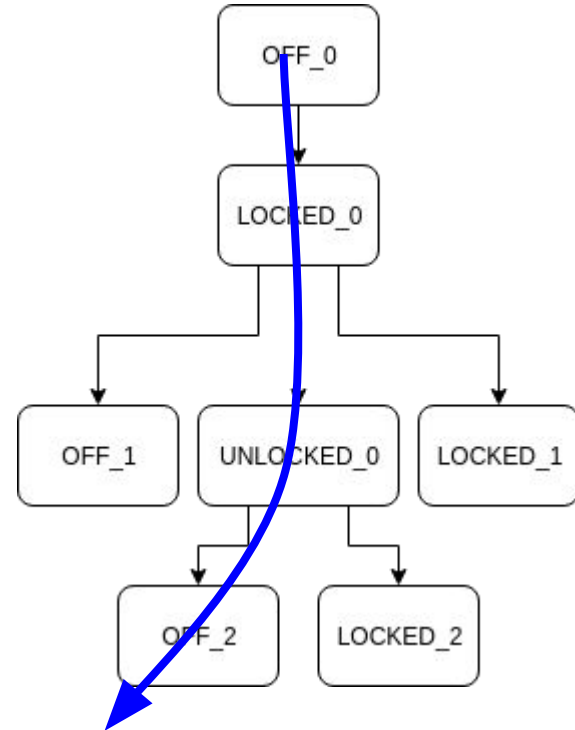
Verifique se o estado atual (inicial) é OFF
Aperte a tecla *home*
Verifique se o estado atual é LOCKED
Insira uma senha correta
Verifique se o estado atual é UNLOCKED



Caminhos e árvores de transição



Verifique se o estado atual (inicial) é OFF
Aperte a tecla *home*
Verifique se o estado atual é LOCKED
Insira uma senha correta
Verifique se o estado atual é UNLOCKED
Aperte a tecla *long lock button*



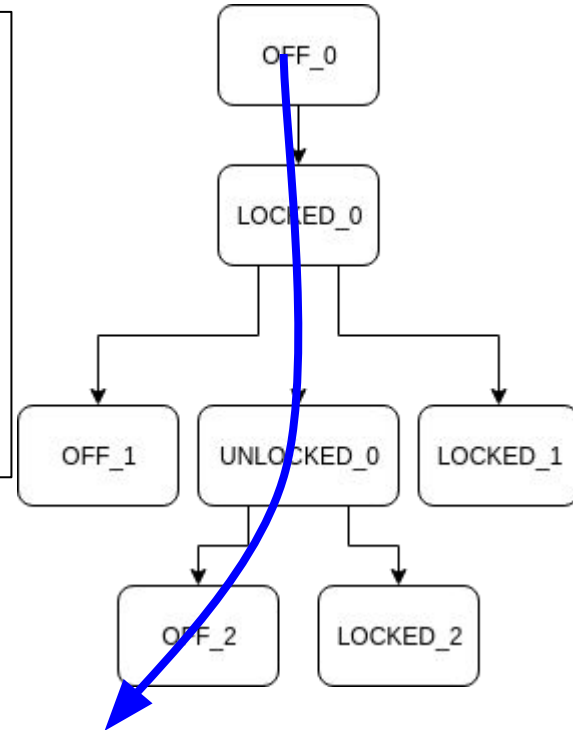
Caminhos e árvores de transição

```
public void testaCaminho02A(){  
    assertEquals("OFF", SDevice.getState());  
  
    SDevice.home();  
    assertEquals("LOCKED", SDevice.getState());  
  
    SDevice.login("login", "password");  
    assertEquals("UNLOCKED", SDevice.getState());  
  
    SDevice.longLockButton();  
    assertEquals("OFF", SDevice.getState());  
}
```

long lock button

UNLOCKED

Verifique se o estado atual (inicial) é OFF
Aperte a tecla *home*
Verifique se o estado atual é LOCKED
Insira uma senha correta
Verifique se o estado atual é UNLOCKED
Aperte a tecla *long lock button*
Verifique se o estado atual é o OFF



Caminhos secretos e tabelas de transição

Um caminho secreto é um caminho que não deveria existir na máquina de estados

Esses caminhos mostram comportamentos inesperados do sistema (uma falha)

Uma das formas de encontrar caminhos secretos é utilizando tabelas de transição

Caminhos secretos e tabelas de transição

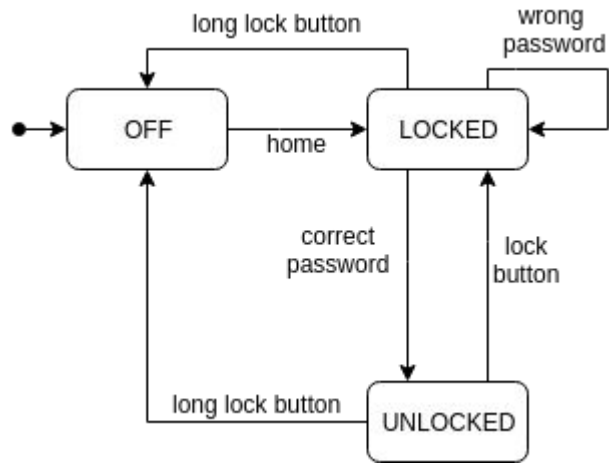
Uma tabela de transição contém cada transição da máquina de estados

Uma transição é obtida pelo estado da qual ela está saindo, o evento que a causa, e o estado para o qual ela vai

Este tipo de tabela pode ser construída da seguinte forma:

- Liste todos os estados nas linhas
- Liste todos os eventos nas colunas
- Para cada transição na máquina de estados, coloque o estado de destino na célula correspondente da tabela (estado x evento)

Caminhos secretos e tabelas de transição



Caminhos secretos e tabelas de transição

	home	wrong password	correct password	lock button	long lock button
OFF					
LOCKED					
UNLOCKED					

Caminhos secretos e tabelas de transição

	home	wrong password	correct password	lock button	long lock button
OFF	LOCKED				
LOCKED					
UNLOCKED					

Caminhos secretos e tabelas de transição

	home	wrong password	correct password	lock button	long lock button
OFF	LOCKED				
LOCKED		LOCKED			
UNLOCKED					

Caminhos secretos e tabelas de transição

	home	wrong password	correct password	lock button	long lock button
OFF	LOCKED				
LOCKED		LOCKED	UNLOCKED		
UNLOCKED					

Caminhos secretos e tabelas de transição

	home	wrong password	correct password	lock button	long lock button
OFF	LOCKED				
LOCKED		LOCKED	UNLOCKED		
UNLOCKED				LOCKED	

Caminhos secretos e tabelas de transição

	home	wrong password	correct password	lock button	long lock button
OFF	LOCKED				
LOCKED		LOCKED	UNLOCKED		OFF
UNLOCKED				LOCKED	

Caminhos secretos e tabelas de transição

	home	wrong password	correct password	lock button	long lock button
OFF	LOCKED				
LOCKED		LOCKED	UNLOCKED		OFF
UNLOCKED				LOCKED	OFF

Caminhos secretos e tabelas de transição

É preciso decidir qual é o comportamento esperado para as células vazias (ignorar evento, lançar exceção, etc)

	home	wrong password	correct password	lock button	long lock button
OFF	LOCKED				
LOCKED		LOCKED	UNLOCKED		OFF
UNLOCKED				LOCKED	OFF

Caminhos secretos e tabelas de transição

O número de possíveis caminhos secretos é igual ao número de células vazias da tabela de transição

Para testar todos os possíveis caminhos secretos, deve-se criar pelo menos um caso de teste para cada célula vazia na tabela de transição

- Leve o sistema para o estado em que existe uma célula vazia na tabela (pode-se usar a árvore de transição para achar um caminho adequado)
- Acione o evento indicado pela coluna correspondente da célula vazia
- Verifique se o sistema está no mesmo estado de antes de acionar o evento

Caminhos secretos e tabelas de transição

	home	wrong password	correct password	lock button	long lock button
OFF	LOCKED				
LOCKED		LOCKED	UNLOCKED		OFF
UNLOCKED				LOCKED	OFF

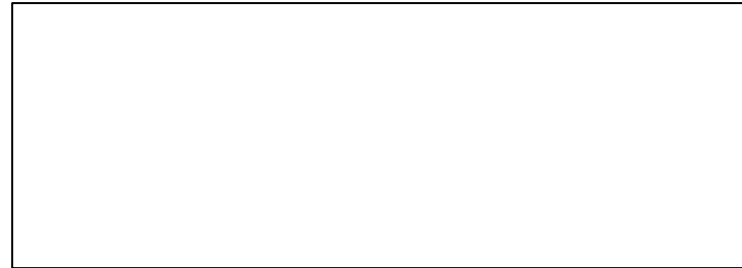
Caminhos secretos e tabelas de transição

	home	wrong password	correct password	lock button	long lock button
OFF	LOCKED				
LOCKED		LOCKED	UNLOCKED		OFF
UNLOCKED				LOCKED	OFF



Caminhos secretos e tabelas de transição

	home	wrong password	correct password	lock button	long lock button
OFF	LOCKED				
LOCKED		LOCKED	UNLOCKED		OFF
UNLOCKED				LOCKED	OFF



Caminhos secretos e tabelas de transição

	home	wrong password	correct password	lock button	long lock button
OFF	LOCKED				
LOCKED		LOCKED	UNLOCKED		OFF
UNLOCKED				LOCKED	OFF



Verifique se o estado atual (inicial) é OFF

Caminhos secretos e tabelas de transição

	home	wrong password	correct password	lock button	long lock button
OFF	LOCKED				
LOCKED		LOCKED	UNLOCKED		OFF
UNLOCKED				LOCKED	OFF



Verifique se o estado atual (inicial) é OFF
Aperte a tecla *home*

Caminhos secretos e tabelas de transição

	home	wrong password	correct password	lock button	long lock button
OFF	LOCKED				
LOCKED		LOCKED	UNLOCKED		OFF
UNLOCKED				LOCKED	OFF



Verifique se o estado atual (inicial) é OFF
Aperte a tecla *home*
Verifique se o estado atual é LOCKED

Caminhos secretos e tabelas de transição

	home	wrong password	correct password	lock button	long lock button
OFF	LOCKED				
LOCKED		LOCKED	UNLOCKED		OFF
UNLOCKED				LOCKED	OFF



Verifique se o estado atual (inicial) é OFF
Aperte a tecla *home*
Verifique se o estado atual é LOCKED
Aperte o botão *lock button*

Caminhos secretos e tabelas de transição

	home	wrong password	correct password	lock button	long lock button
OFF	LOCKED				
LOCKED		LOCKED	UNLOCKED		OFF
UNLOCKED				LOCKED	OFF



Verifique se o estado atual (inicial) é OFF
Aperte a tecla *home*
Verifique se o estado atual é LOCKED
Aperte o botão *lock button*
Verifique se o estado atual é LOCKED

Super estados

As máquinas de estado que modelam sistemas muito grandes e complexos tendem a ser difíceis de entender e gerenciar

Nesses casos, é possível agrupar estados em super estados para simplificar o modelo ou um nível de abstração dele

Os super estados são, eles mesmos, máquinas de estados que são utilizadas como estados em uma máquina de estados mais complexa

Super estados

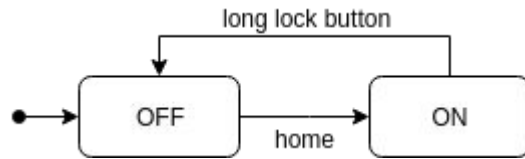
Os super estados possuem estados, transições e um estado inicial

Uma transição que vai para o super estado, vai para seu estado inicial

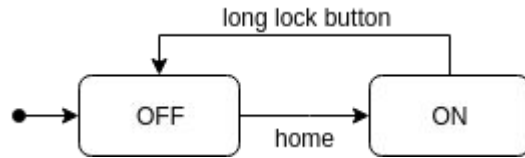
Uma transição que sai de um super estado significa que, para qualquer de seus estados, aquela transição existe dado o evento associado a ele

No modelo, um super estado pode ser mostrado como um simples estado, ou com os estados que os compõem

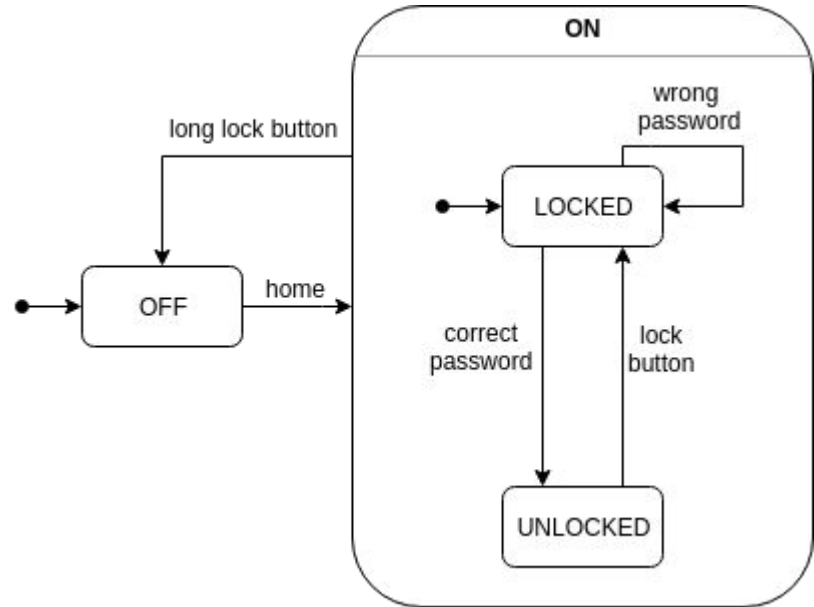
Super estados



Super estados



ou



Regiões

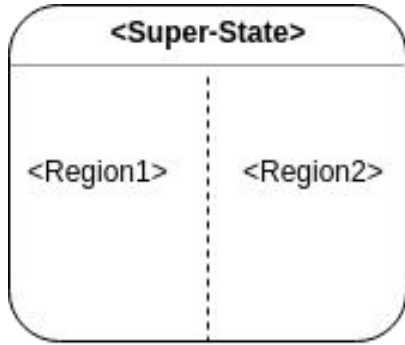
O recurso de representação de regiões é usado quando a modelagem requer que o sistema possa assumir dois estados ao mesmo tempo

Um super estado pode ser dividido em duas regiões ortogonais

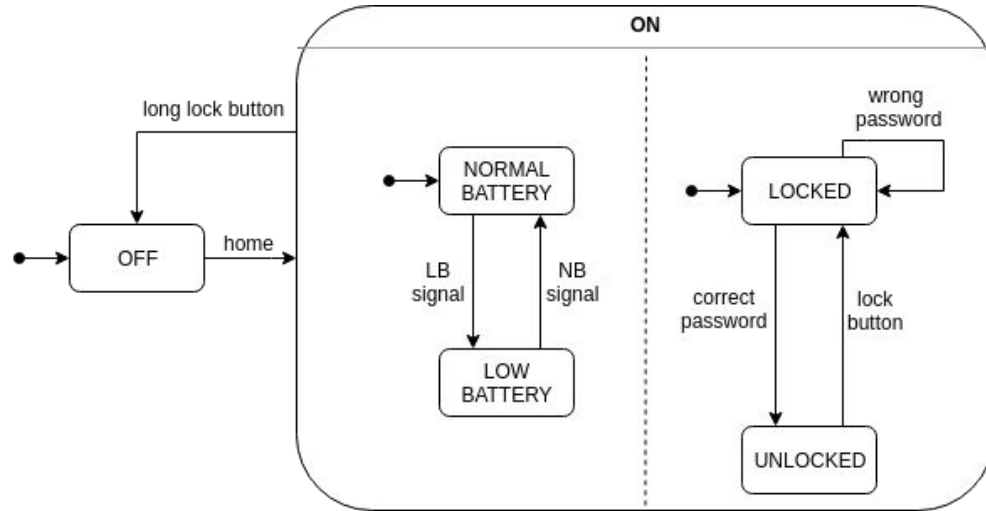
Quando as regiões são ortogonais significa que

- cada região contém uma máquina de estado
- as máquinas de cada região são independentes entre si
- quando o sistema entra no super estado, ele entra nos estados iniciais de cada região

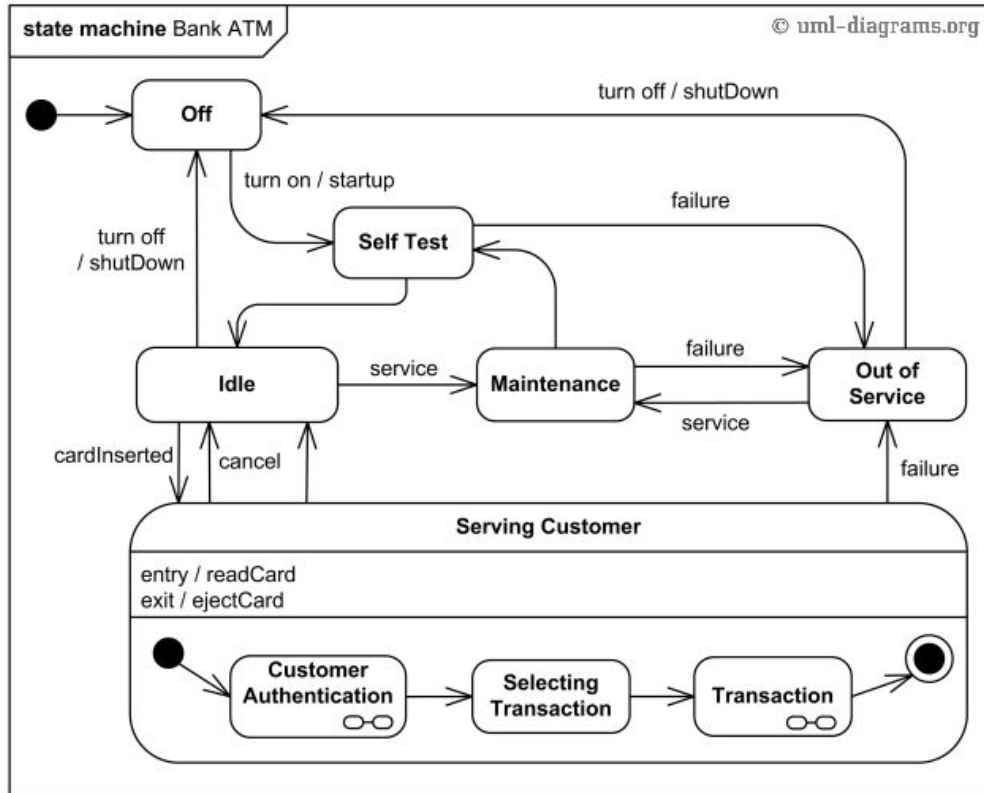
Super estados e regiões



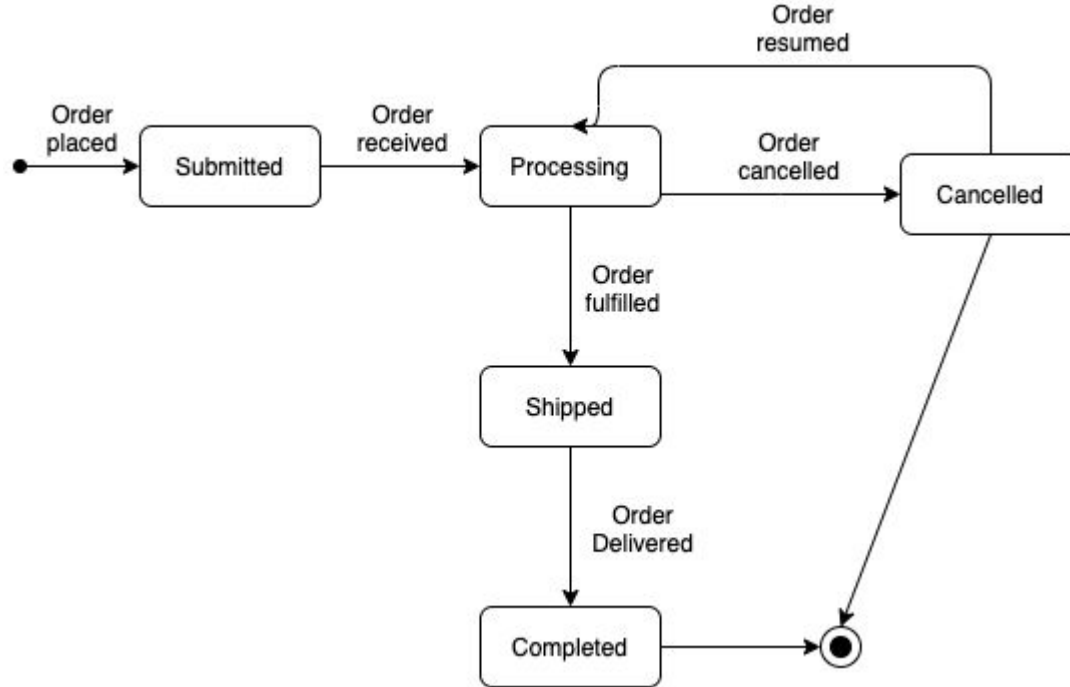
Super estados e regiões



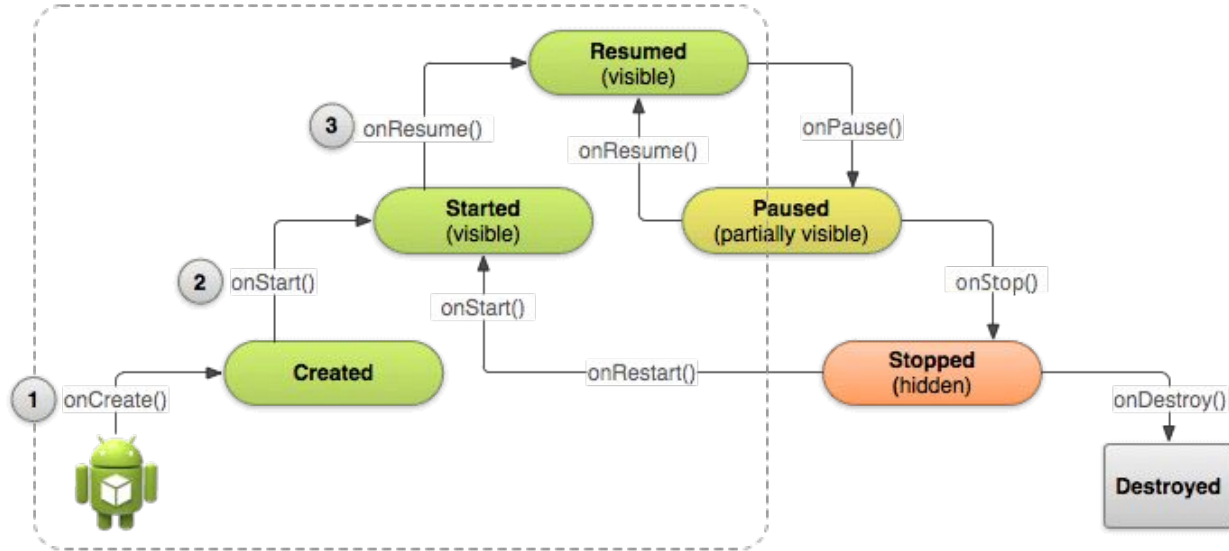
Exemplos de máquinas de estados finito



Exemplos de máquinas de estados finito



Exemplos de máquinas de estados finito



Na prática

Teste de unidade

- Classes
- Getters
- Métodos/operações

Na prática

Teste de sistema

- API
- Interface web
- Interface mobile

ACH2028 - Qualidade de Software

Aula 11 - Teste Baseado em Estados

Prof. Marcelo Medeiros Eler

marceloeler@usp.br