

Diagrama de Máquina de Estados

Lucas Lima

Agenda

- Visão Geral
- Máquinas de estado, blocos e hierarquia
- Diagrama de máquina de estado
- Pseudoestados
- Estado
- Transição
- Roteamento com pseudoestados
- Máquinas de estado e chamadas a operações
- Estados compostos com múltiplas regiões ortogonais
- Pseudoestado de história
- Submáquinas de estado

Visão geral

Usadas para descrever o comportamento dependente de estado de um bloco, em termos dos seus estados e das transições entre eles.

Máquinas de estado, em geral, estão associadas a blocos e executam no contexto de uma instância de um bloco.

Geralmente, representa o ciclo de vida de um bloco.

Visão Geral

- Comportamento especificado por um conjunto de regiões, cada uma com seus próprios estados
 - Os estados em uma região são exclusivos: quando uma região está ativa, apenas um estado está ativo
 - Uma região tem um pseudo estado inicial, o local pelo qual a região começa a execução quando se torna ativa
 - Uma região tem um estado final que, quando ativo, significa que a região terminou
- Quando um estado é alcançado
 - uma ação de entrada (opcional) é executada (*entry behaviour*)
 - uma ação de saída (opcional) é executada (*exit behaviour*)

Visão Geral

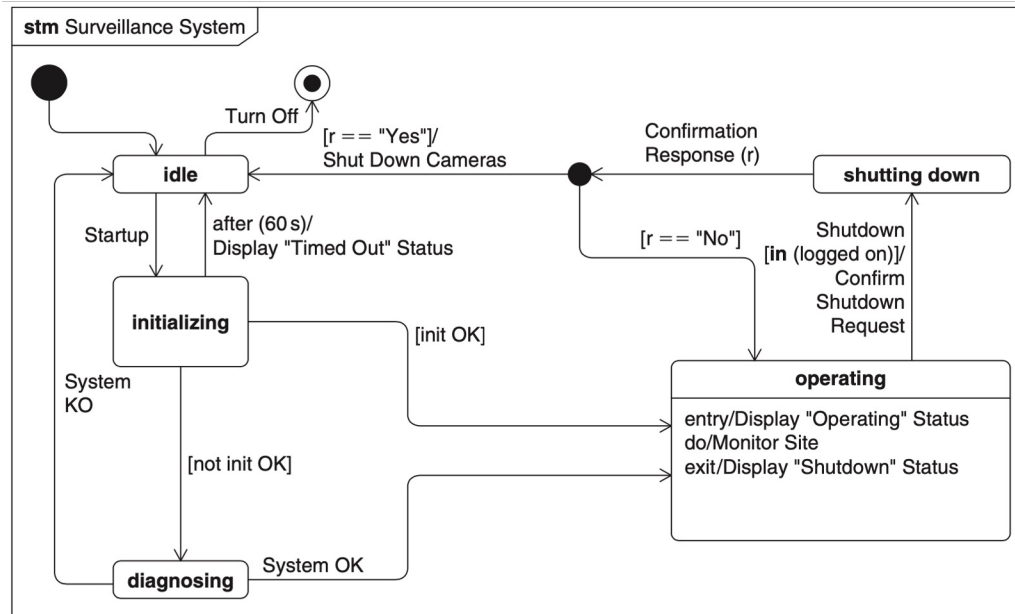
- Efetuado por transições que conectam um estado fonte a um estado alvo
- Transições são definidas por triggers, guardas, e efeitos
 - Um trigger indica que um evento pode causar uma transição
 - A guarda é avaliada a fim de testar se a transição é válida
 - O efeito é o comportamento executado uma vez que transição é disparada
 - Triggers pode estar baseados em eventos como
 - Expiração de um clock
 - Recepção de um sinal pelo objeto proprietário da máquina de estados
- Junção e escolha são pseudo-estados que permitem a transição composta (entre estados) com múltiplas guardas e efeitos

Máquinas de estado, blocos e hierarquia

- Máquinas de estados em blocos diferentes podem interagir por meio de sinais e invocando operações
- Hierarquia de estados surge quando um estado contém regiões
 - Estado com uma região: estado composto
 - Estado com mais de uma região: estado composto ortogonal
 - Um estado pode referenciar um outra máquina de estados (estado de submáquina)
- Máquinas de estado podem ser usadas em conjunto com outros comportamentos (atividades e interações)

Diagrama de máquina de estado

- Também conhecido como *state chart* ou diagrama de estados. Em SysML, **diagrama de máquina de estado**
- Cabeçalho: **stm**[State Machine] state machine name [diagram name]



Regiões

- Uma máquina de estados pode conter uma ou mais regiões
- Cada região é definida em termos de estados ou pseudo estados coletivamente nomeados como vértices e transições entre os vértices
- Uma região ativa tem apenas um estado ativo
 - Uma região nunca estará em um pseudo estado, este serve apenas para auxiliar a determinar o próximo estado ativo
- Uma máquina de estados com múltiplas regiões pode descrever comportamento concorrente

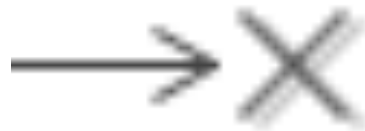
Pseudoestado inicial e final

- Estado simples
 - Sem regiões, sem estados aninhados
- Inicialização e finalização de uma região são descritos por pseudoestado inicial e estado final
- Pseudoestado inicial
 - Especifica o estado inicial de uma região
 - A transição de saída de um pseudo estado inicial pode incluir um efeito (por exemplo, definir valores iniciais usados pela máquina de estado)
- Quando o estado ativo é o **estado final**
 - A região termina e não acontecem mais transições
 - Um estado final não pode ter transições de saída



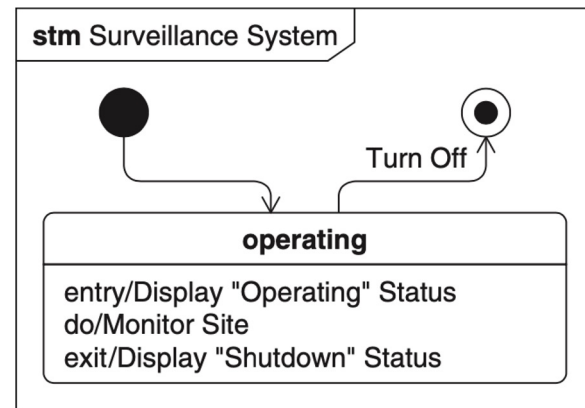
Pseudoestado de término

- Associado com o estado de uma máquina inteira
 - Se um pseudoestado de terminação é alcançado, o comportamento da máquina termina
 - Tem o mesmo efeito de alcançar os estados finais de todas as regiões de uma máquina de estados



Estado

- Representa a condição (situação) de um bloco
 - Como o bloco responde a eventos e quais comportamentos realiza
- A máquina de estado define todos os estados válidos e transições entre estes
- Cada estado pode ter comportamentos (atividades, ações) de entrada (*entry*) e saída (*exit*)
- Comportamento **do** é executado após o de entrada (*entry*)



Transições

- Especifica quando uma mudança de estado ocorre
- A máquina de estados executa de forma completa até uma transição ser disparada (trigger)
 - Não é capaz de consumir outro evento disparado até que complete o processamento do evento atual
- Composta por três elementos:
 - Trigger
 - Guarda
 - Efeito

Triggers

- Identificam os possíveis estímulos que levam uma transição a ocorrer
- Principais tipos
 - **Sinal:** evento que indica que uma nova mensagem assíncrona, correspondente a um sinal, chegou
 - Evento de tempo: indica que um intervalo de tempo passou desde que se entrou no estado atual (relativo) ou que um instante de tempo foi alcançado (absoluto)
 - Evento de mudança (change): indica que determinada condição foi satisfeita
 - Evento de chamada (call): indica que uma operação do bloco foi solicitada
- Eventos são consumidos pela máquina quando são apresentados, mesmo que não disparem transições

Guardas e efeitos de transições

- Um guarda contém uma expressão que deve ser avaliada como verdadeira para que a transição aconteça
 - Se uma guarda for avaliada como falsa o evento é consumido sem efeito
- O efeito de uma transição é um comportamento (geralmente atividade ou comportamento opaco)
 - No caso do trigger ser um sinal ou chamada, os argumentos podem ser utilizados no efeito (e.g., atualizar propriedades do bloco)
 - Pode conter comportamento complexo (e.g., enviar sinais ou fazer chamadas a operações de outros blocos)
- Quando um transição é disparada, a ação *exit* do estado atual (origem) é executada, o efeito da transição é executado e finalmente a ação *entry* do estado alvo é executada

Internal Transition vs Self-transition

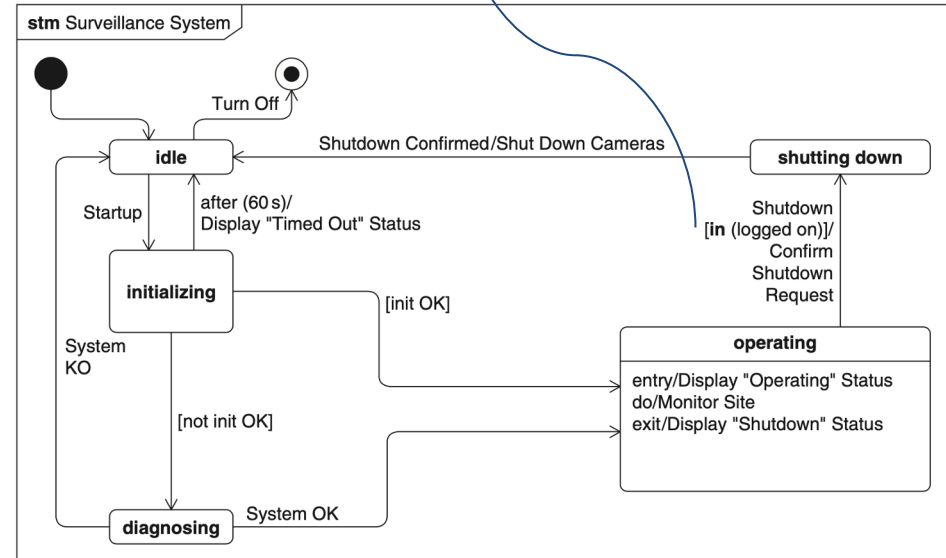
- Transições internas podem acontecer enquanto se está no estado que as possui, mas elas não disparam *entry* e *exit actions*
- Já transições recursivas (*self-transition*) são transições para o mesmo estado atual, só que elas disparam *entry* e *exit actions*

Notação para transições

- Setas entre estados
 - *Self transition*
- Texto para um *trigger*
 - eventos *signal* e *call*: nome do sinal ou operação seguidos por atribuições entre parênteses
 - eventos de tempo: o termo *after* ou *at* seguido pelo tempo
 - eventos de mudança: o termo *when* é seguido pela condição

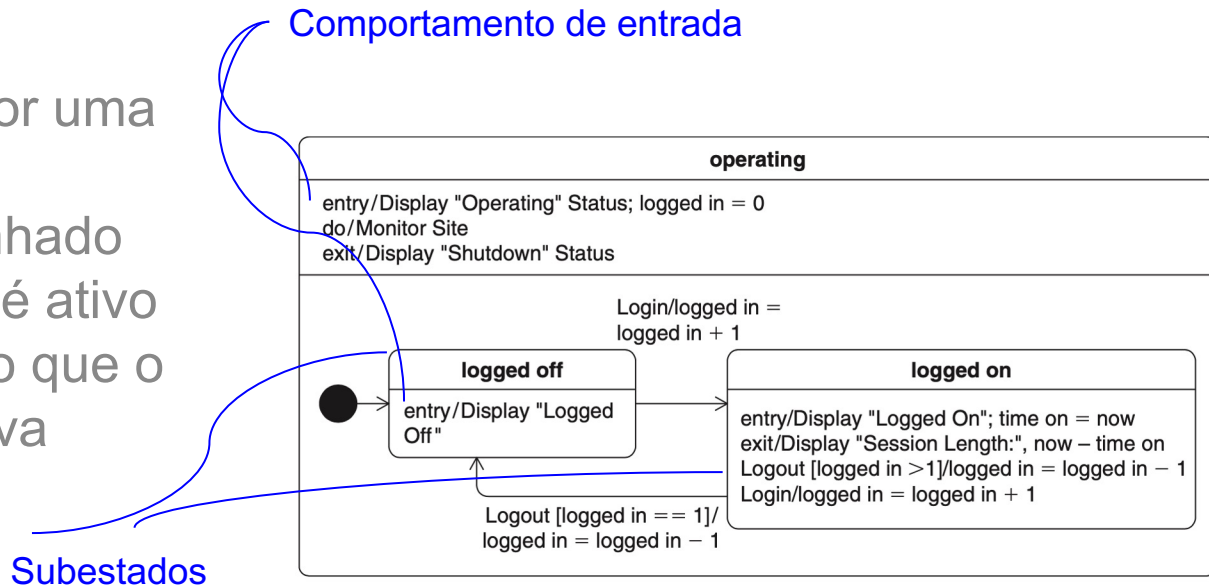
Notação para transição:
trigger[guard]/action

Teste de estado em guarda:
in (state x)
not in (state x)



Hierarquias de estados

- Estados podem ter regiões. Neste caso, os estados são **compostos** ou **hierárquicos**
- Estado composto por uma região
 - Um estado aninhado em uma região é ativo quando a região que o contém está ativa



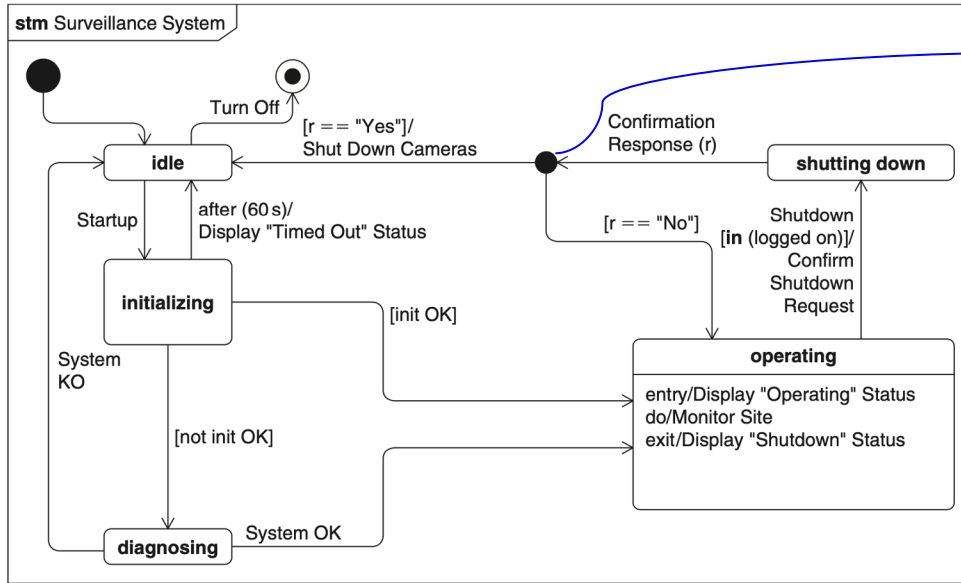
Exercício

- Crie um máquina de estado para o veículo com as seguintes características:
 - Deve ter dois estados: ligado e desligado
 - O estado ligado deve ter subestados para indicar o veículo parado, o veículo em movimento para frente, o veículo dando ré. Ele também deve ter entry (checa status), do (fornece potência) e exit (desliga acessórios) actions.
 - Crie eventos e comportamentos que achar interessante, mas sem efeitos de transição muito detalhados

Roteamento de transições e pseudoestados

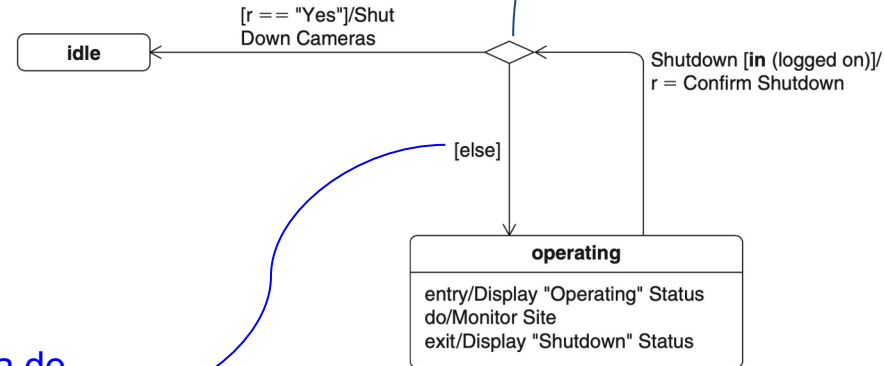
- Pseudoestados que permitem a transição composta entre estados:
junction e *choice*
- Junção (*junction*)
 - Permite a especificação de mais de um caminho de transição entre estados. Apenas um caminho é tomado como resposta a um evento
 - A transição selecionada será uma daquelas com guarda avaliada como verdadeira
- Escolha (*choice*)
 - Permite múltiplas transições de entrada e saída
 - As guardas das transições de saída são avaliadas apenas quando o pseudoestado (*choice*) é alcançado
 - Há ao menos uma transição de saída, do contrário a máquina é inválida

Roteamento de transições e pseudoestados



junction

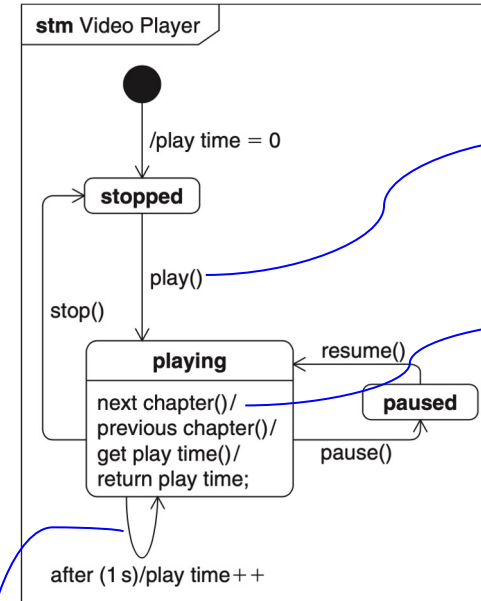
choice



else é uma forma de assegurar a validade da máquina

Máquina de estados e chamadas a operações

- Máquinas de estado podem responder a chamadas a operações no bloco por eventos de chamada (*call events*)
 - Manipulação síncrona
 - O bloco que chama bloqueia e aguarda a resposta
 - Manipulação assíncrona
 - Comportamento similar a receber um sinal
- A máquina executa todas as ações disparadas pelo evento de chamada



Call events

Disparam transições internas ao estado *playing*

Self transition

| Video Player |
|---|
| <i>values</i> play time : Seconds |
| <i>operations</i> get play time() : Seconds next chapter() pause() play() previous chapter() resume() stop() |

Estado compostos com múltiplas regiões ortogonais

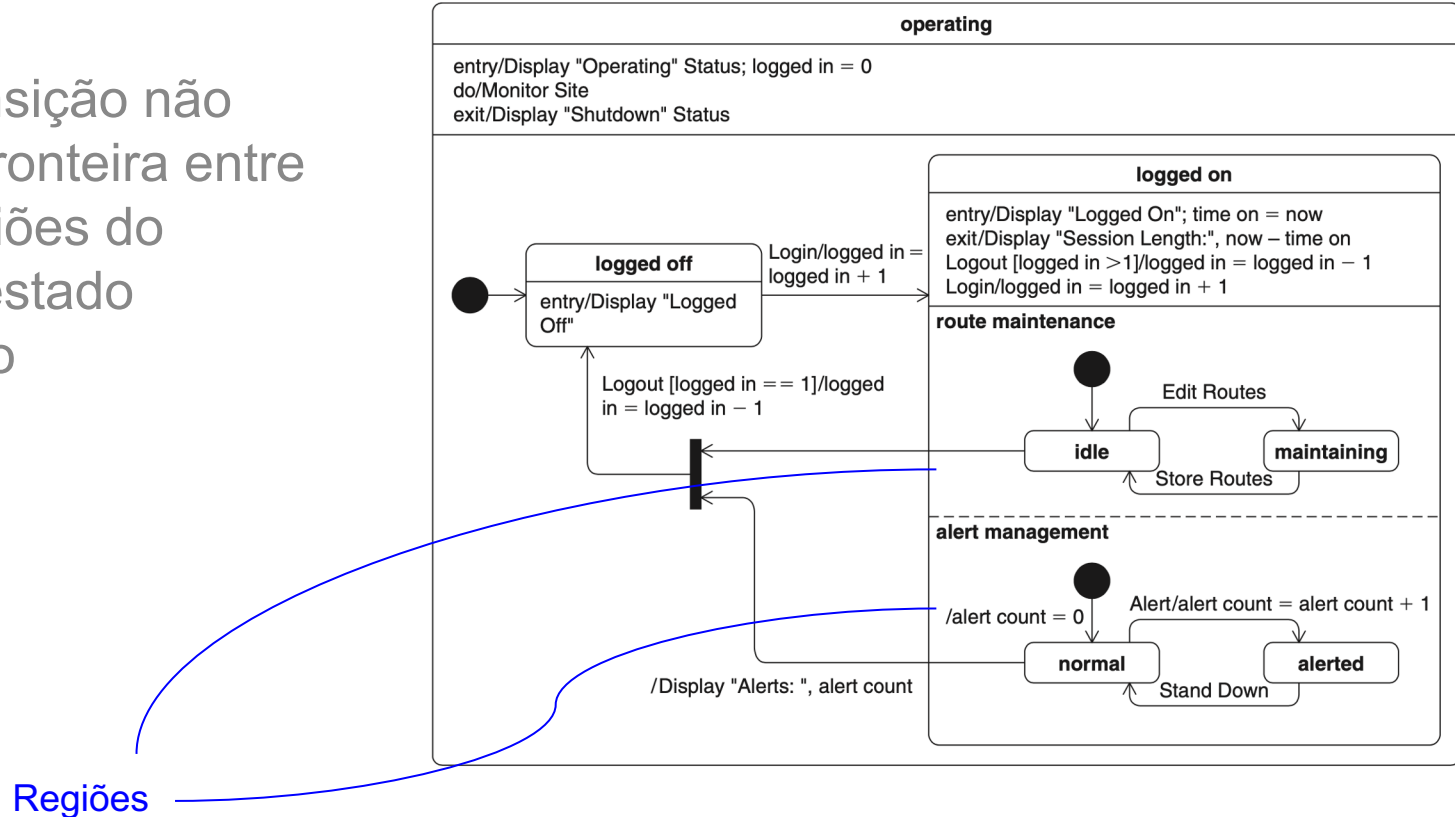
- Estados podem conter múltiplas regiões ortogonais que podem conter subestados
 - Chamado de estado composto ortogonal
 - Quando ativo, cada região tem seu próprio estado ativo independente dos outros; qualquer evento de entrada é analisado independentemente em cada região
 - Uma transição que termina em um estado composto dispara transições do pseudo estado inicial de cada região
 - Um evento de terminação para o estado composto ocorrerá quando todas as regiões estiverem em seus estados finais

Estado compostos com múltiplas regiões ortogonais

- Pseudo estado de bifurcação (*fork*)
 - Uma transição de entrada e tantas transições de saída quantas forem as regiões ortogonais
 - Todas as transições de saída são parte da transição composta
 - Podem não ter triggers ou guardas, mas podem ter efeito
- Pseudo estado de junção (*join*)
 - Coordena transições de saída de um estado composto ortogonal
 - Possui múltiplas transições de entrada e uma de saída
- Sintaxe: barras verticais ou horizontais em que transições começam ou terminam

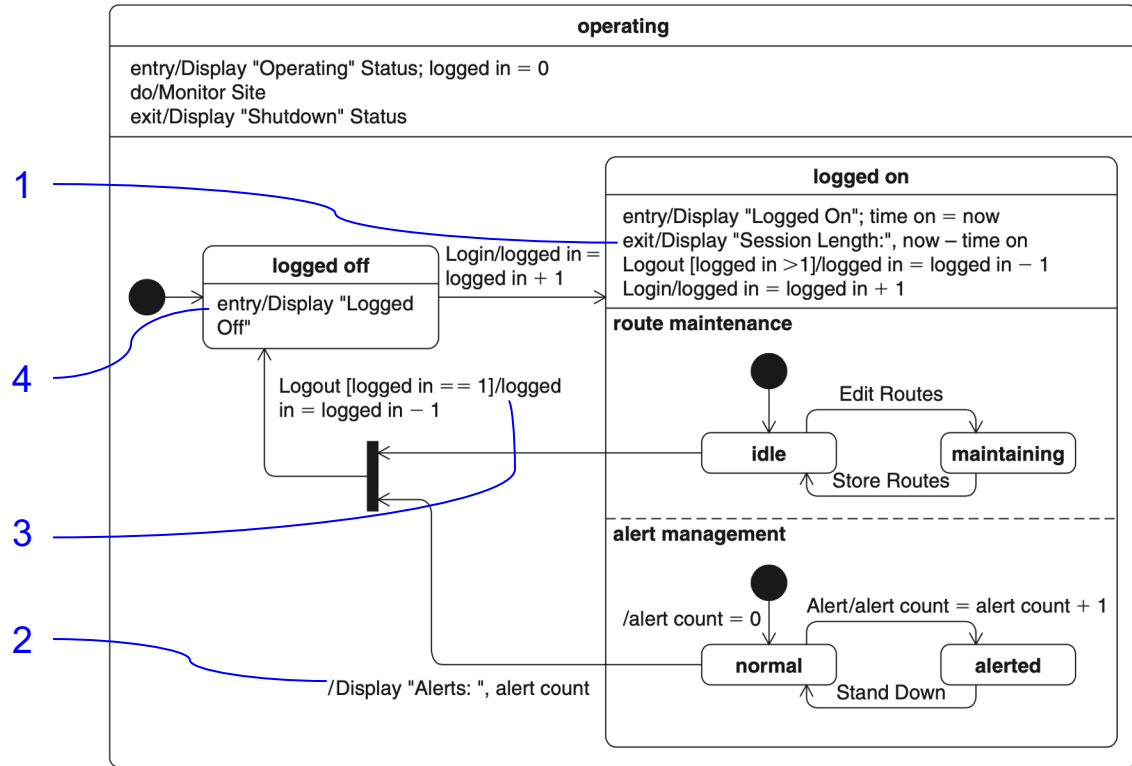
Estado compostos com múltiplas regiões ortogonais

- Uma transição não cruza a fronteira entre duas regiões do mesmo estado composto



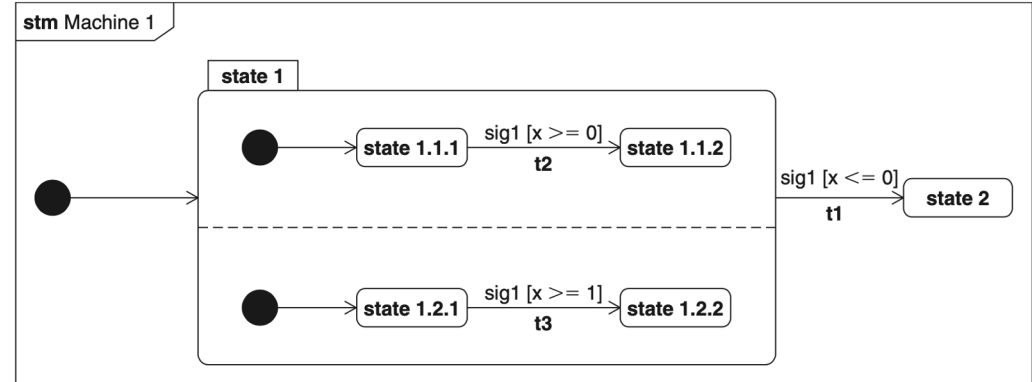
Estado compostos com múltiplas regiões ortogonais

- Ordem de execução disparada por um sinal *Logout* válido
 - Transição de saída do join dos subestados *idle* e *normal*



Ordem de disparo das transições em hierarquias de estados aninhados

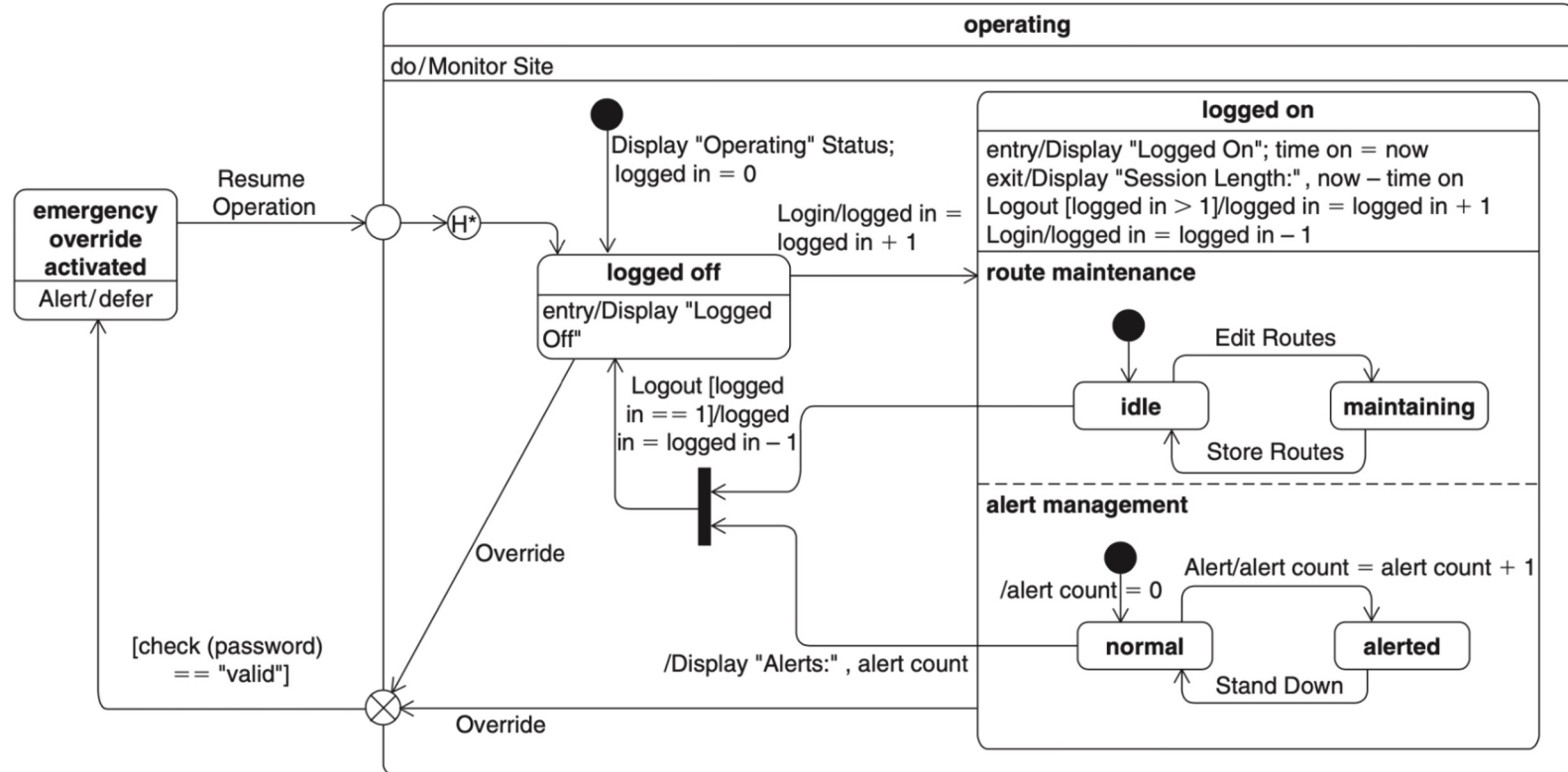
- Um mesmo evento pode disparar transições em vários níveis em uma hierarquia de estados, apenas uma das transições pode ser tomada em um instante (com exceção de regiões concorrentes)
 - Prioridade para os estados mais internos
- Considere x com valor variando de um -1 a 1
- Ordem dos disparos
 - $x = -1$: $t1$
 - $x = 0$: $t2$
 - $x = 1$: $t2$ e $t3$ podem ser disparados



Pseudo estado de história

- Manipulação de um evento de exceção interrompe o estado atual, responde ao evento e retorna ao estado do momento da interrupção
- Pseudo estado **history**
 - *deep*: registra os estados de todas as regiões na hierarquia de estado abaixo, incluindo a região a que pertence o pseudo estado
 - *shallow*: registra apenas o estado de em mais alto nível da região a que pertence
- Descrito pela "H" dentro de um círculo

Pseudo estado de história

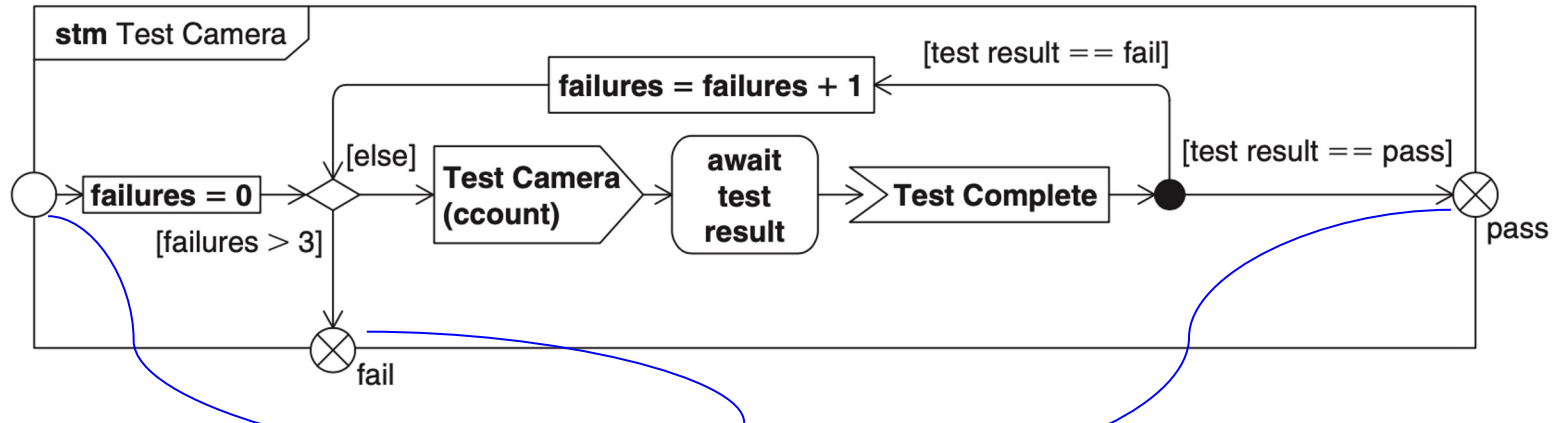


Sub-máquinas de estado

- Reuso acontece por meio de um estado chamado *submachine state*
 - Uma transição terminando em um estado submáquina dará início à máquina referenciada
 - Eventos de completção disparam transições que tem como fonte a submáquina, quando a máquina referenciada termina
 - Dois subestados adicionais
 - **entry** e **exit** permitem à máquina definir pontos de entrada e saída adicionais que podem ser acessado a partir da submáquina
 - descritos como círculos que sobrepõem a fronteira da máquina de estados ou estado composto
 - entrada: círculo vazio
 - saída: círculo vazio contendo X

Reuso de máquinas de estados

- Máquina de estado com uma região
 - Pseudo estados de entrada e saída são similares a *junctions*
 - As guardas de saída são avaliadas antes da transição composta ser disparada; a única transição de saída será tomada



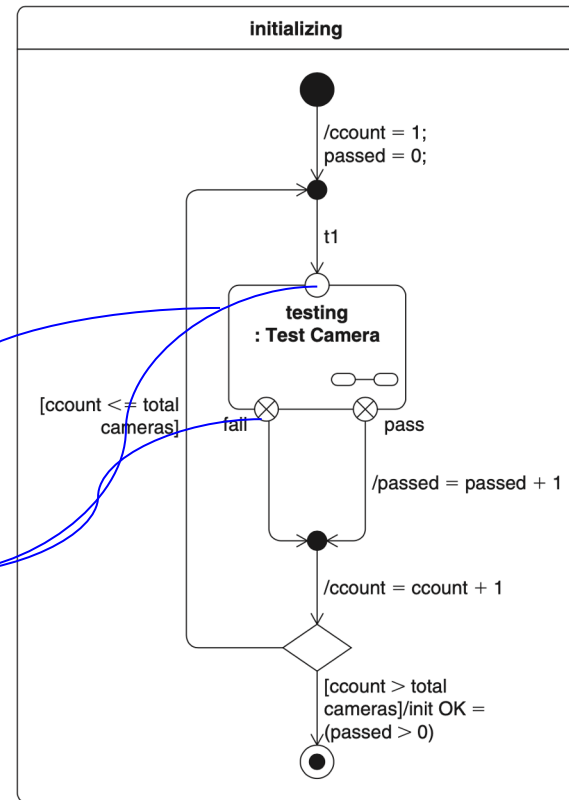
Pontos de conexão

Submáquina

- Um estado submáquina contém uma referência para uma outra máquina de estados que é executada como parte
- Pontos de conexão

Submáquina

Pontos de conexão



Exercício

- Atualize a máquina de estado do veículo com as seguintes características:
 - Escolha algum estado dentro do estado “ligado” e o detalhe em uma outra máquina de estado e use uma submáquina no lugar dele
 - Crie uma região ortogonal dentro do estado “ligado” que detalha algum comportamento de interação com o GPS ou multimedia do veículo

Resumo

- Máquinas de estado descrevem comportamento reusável dependente de estado de um bloco
- Cada máquina contém ao menos uma região que pode conter subestados e transições. Cada região possui um único estado ativo
- Um estado é uma abstração de uma situação no ciclo de vida de um bloco
- Transições descrevem mudanças de estado válidas e sob quais circunstâncias as mudanças acontecerão
- Uma guarda expressa quaisquer restrições adicionais que precisam ser satisfeitas se a transição vai ser disparada
- Estados podem ser compostos por estados aninhados em uma ou mais regiões