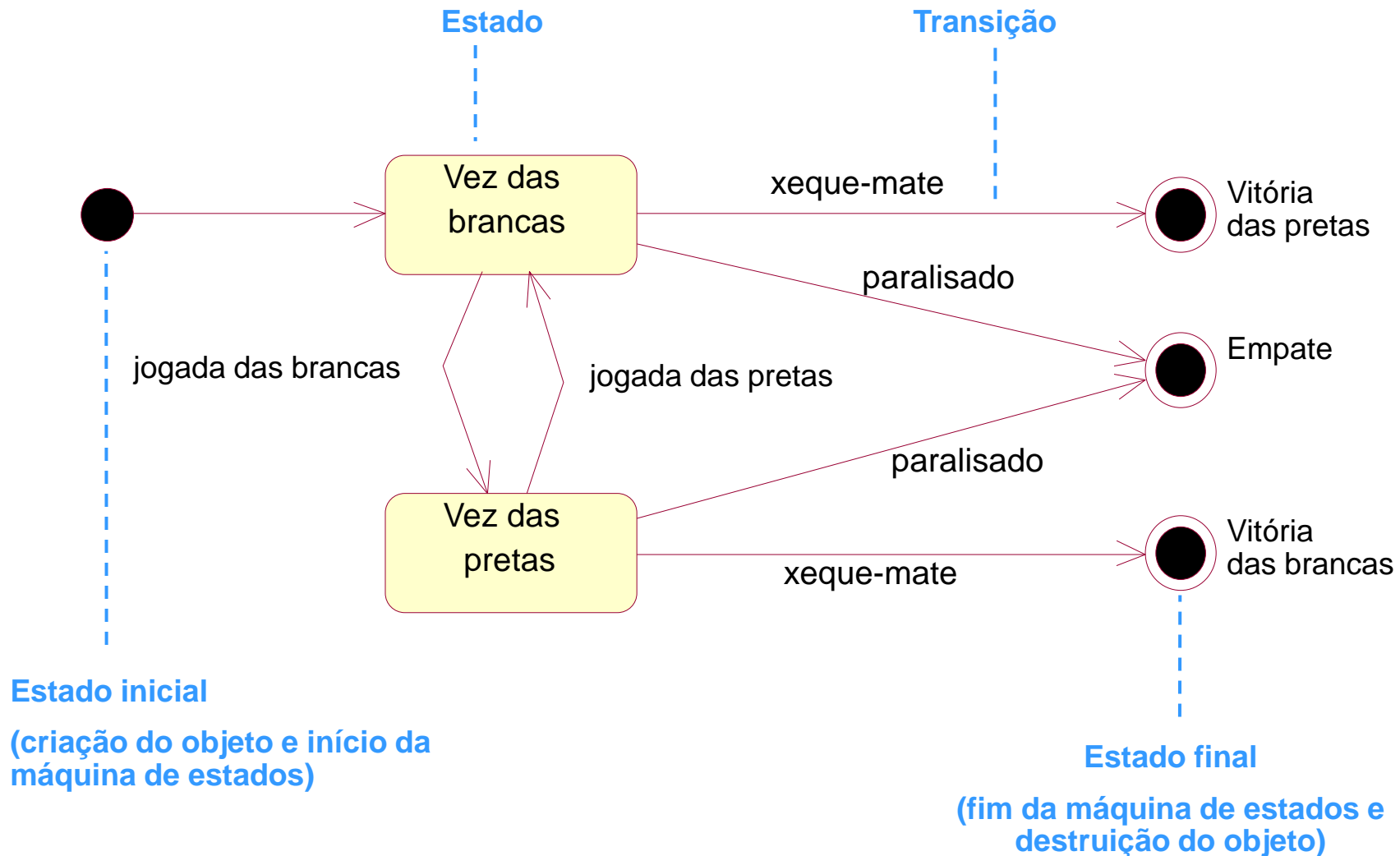


Diagramas de Estados

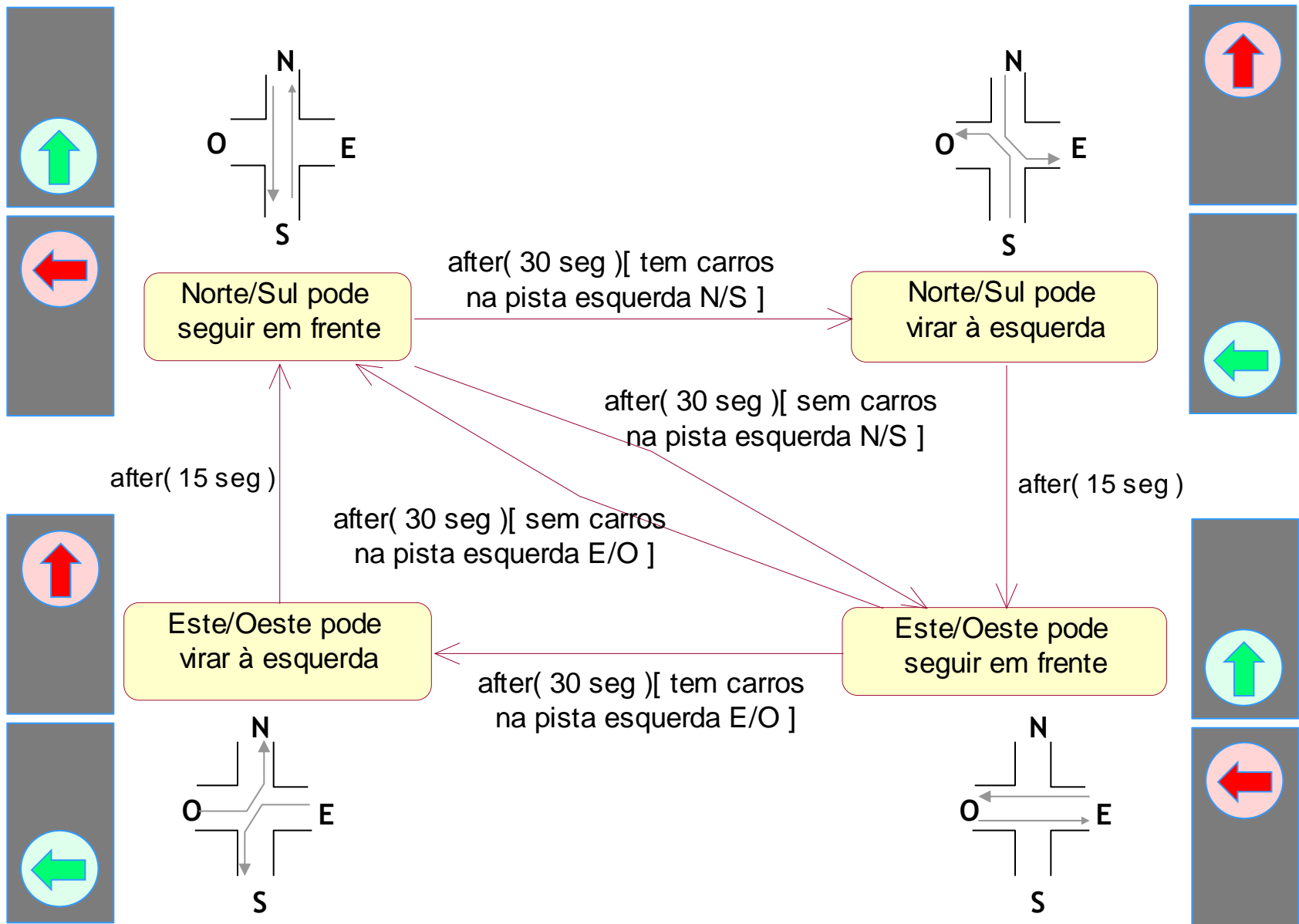
Exemplo: Jogo de xadrez



Objetivo

- Um diagrama de estados especifica **estados** (duradouros e em número finito) e **transições** entre estados (instantâneas) causadas por **eventos** (instantâneos)
- Pode também especificar as **ações** (instantâneas) e **atividades** (duradouras) realizadas em resposta a eventos ou durante a permanência em estados, respectivamente
- Usado normalmente para modelar o **ciclo de vida** dos objetos de uma classe (objeto visto como máquina de estados)
- Em geral, serve para modelar a dinâmica de um sistema ou objeto cujo estado evolui por saltos (transições instantâneas) em resposta a eventos, com um número finito de estados

Exemplo: Semáforo



Comportamento e ciclo de vida de objetos ou sistemas reativos (1)

- Um diagrama de estados é útil para modelar o comportamento e o ciclo de vida de um objeto ou sistema reativo
- **Objeto/sistema reativo**: funciona por resposta (reação) a estímulos (eventos)
 - Sistemas de interação com o usuário (com formulários, botões etc.) são tipicamente reativos
 - **Sistemas de tempo real** (ex: semáforo) podem ser vistos como sistemas reativos que reagem a **eventos temporais** (*timeout*, ...)
 - Sistemas reativos geralmente têm **memória** - um **estado** interno que acumula o efeito dos estímulos recebidos no passado e afeta a resposta a estímulos futuros

Comportamento e ciclo de vida de objetos ou sistemas reativos (2)

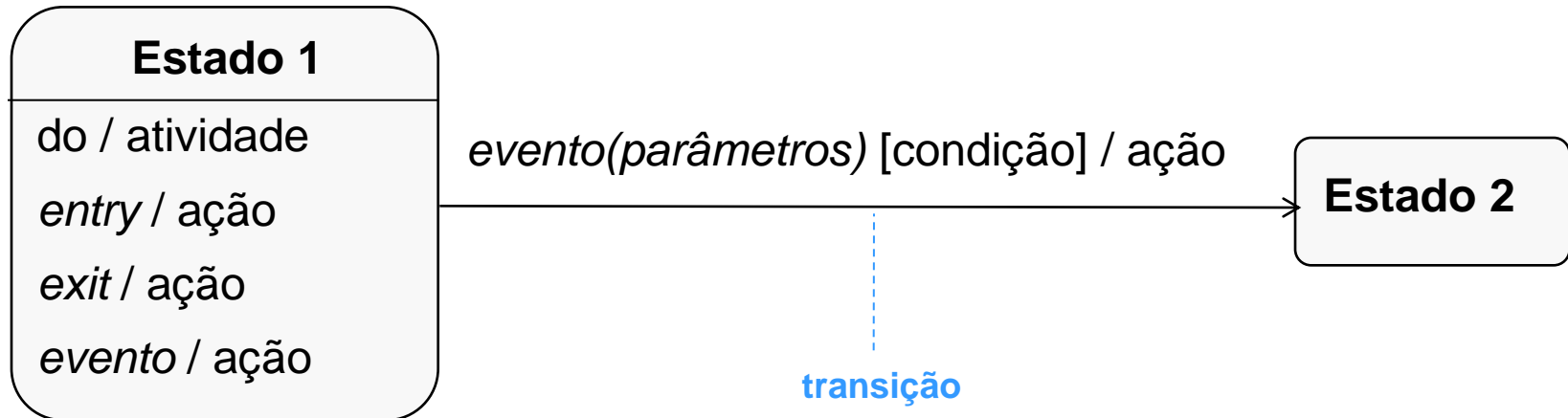
- **Ciclo de vida** de um objeto/sistema: as seqüências de estados por que pode passar durante a sua vida em resposta a eventos, em conjunto com as respostas a esses eventos, ou seja:
 - os **estados** possíveis (exemplo - estado civil: solteiro, casado, ...)
 - as **transições** de estado possíveis (exemplo: pode passar de solteiro para casado, mas não o contrário)
 - os **eventos** que causam essas transições (exemplo: o casamento implica a passagem ao estado de casado)
 - as **ações** do objeto em resposta a esses eventos (ex: despedida de solteiro)
- Objetos da mesma classe têm o mesmo ciclo de vida, pelo que basta construir um diagrama de estados por classe relevante
 - Interações entre objetos aparecem pouco explícitas, como trocas de mensagens

Relação com outros diagramas dinâmicos

- **Diagramas de estados** focam o fluxo (passagem) de controle de estado para estado num objeto, mostrando todas as seqüências possíveis de funcionamento de um único objeto
 - Bom para especificação
 - Comunicação entre objetos do mesmo sistema aparece de forma pouco explícita, através de eventos gerados (como ações) na máquina de estados de um objeto, que são testados (como eventos) na máquina de estados de outro objeto
- **Diagramas de interação** (seqüência ou comunicação) focam o fluxo (passagem) de controle de objeto para objeto numa seqüência particular de funcionamento de um sistema
 - Comunicação entre objetos aparece de forma explícita
 - Bom para ilustração mas fraco como especificação
- **Diagramas de atividades** focam o fluxo de controle de atividade para atividade em uma atividade de mais alto nível (operação, caso de uso etc.) de um objeto ou sistema

Notação e conceitos básicos

Notação básica



Seqüência de mudança de estado:

1. Ocorre o **evento** associado à **transição** e a **condição** de guarda é verdadeira
2. É interrompida a **atividade** associada ao estado de origem (**Estado 1**), se não tinha já terminado
3. É executada a **ação** à saída (**exit**) do estado de origem
4. É executada a **ação** associada à **transição**
5. É executada a **ação** à entrada (**entry**) do estado de destino
6. É iniciada a **atividade** associada ao estado de destino (**Estado 2**)

Estados

- Um **estado** é uma condição ou situação na vida de um objeto, durante a qual o objeto satisfaz alguma condição, realiza alguma atividade ou espera por algum evento
 - Exemplo (relativamente ao estado civil de uma pessoa): solteira, casada, ... **relação com atributos e ligações**
- Escolha dos estados e :
 - Foi dito anteriormente que o estado de um objeto é dado pelos valores de atributos e ligações com outros objetos que mantém num dado momento
 - Essa definição conduz, em geral, a demasiados estados
 - Ora, no diagrama de estados, interessa apenas distinguir estados que apresentam diferentes respostas a eventos
 - Assim, os estados que interessa considerar aqui correspondem, em geral, a conjuntos de valores de atributos e ligações (possivelmente expressos por condições), e ignoram-se atributos e ligações irrelevantes para o comportamento do objeto

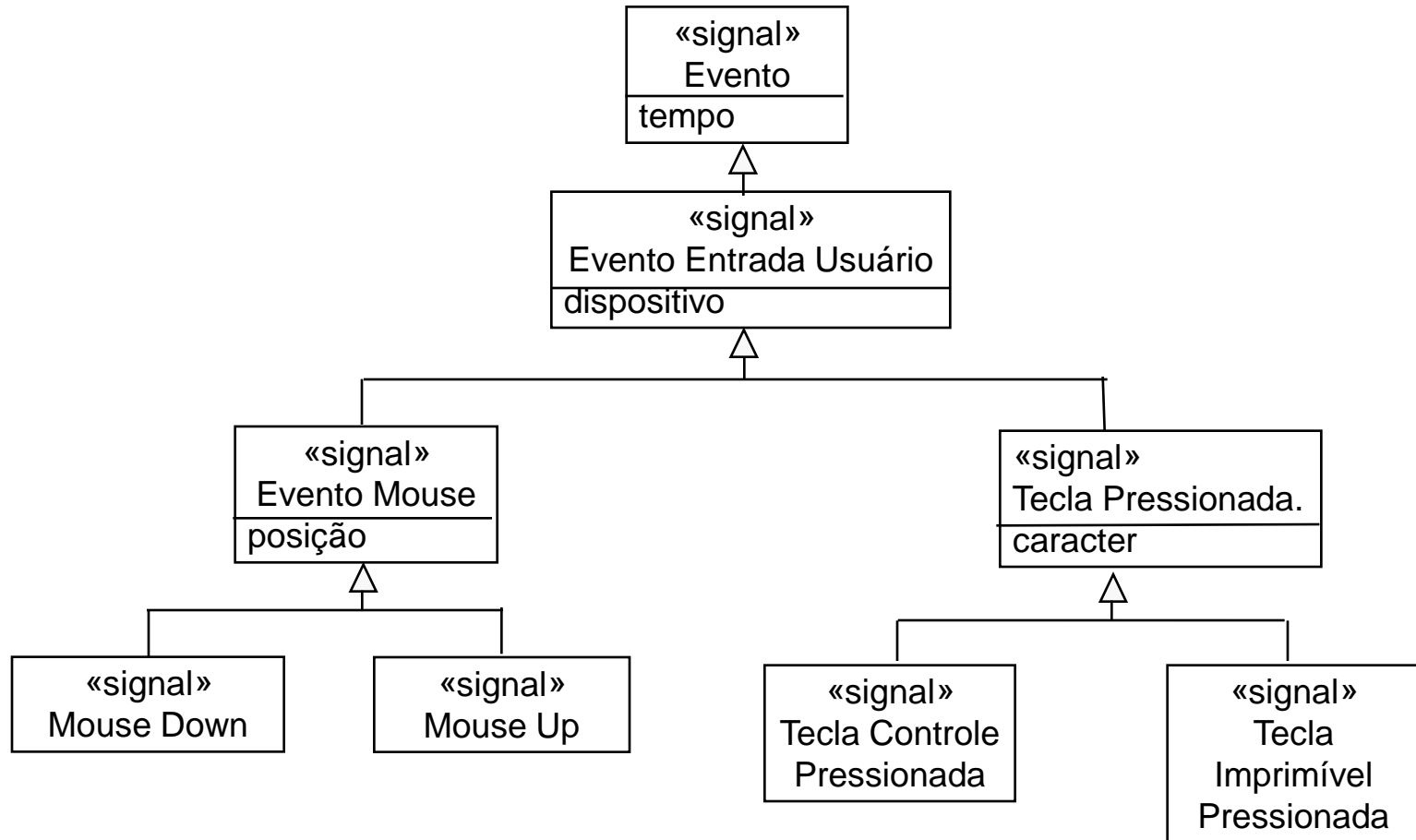
Eventos

- Um **evento** é uma ocorrência significativa que tem uma localização no tempo (instante de tempo do evento) e no espaço
- No contexto de uma máquina de estados, um evento pode ter como resposta uma **transição** (mudança de estado) e/ou uma **ação**
- Os eventos são **instantâneos**
 - O que interessa é que se lhe possa atribuir um instante de tempo de ocorrência
- Os eventos podem ser de vários tipos:
 - **Sinais** - eventos simbólicos sinalizados explicitamente
 - **Chamadas** - invocação de operações
 - **Eventos temporais** - passagem de tempo ou ocorrência de um data-hora
 - **Eventos de mudança** - uma condição tornar-se verdadeira
- Os eventos podem ter **parâmetros** [exemplo: dígito-marcado(5)]

Sinais (eventos simbólicos)

- Sinais são modelados por classes com estereótipo «**signal**»
 - Parâmetros (argumentos) do sinal são atributos da classe
 - Hierarquias de sinais são modeladas como hierarquias de generalização
- Envio de sinal representado por ação **send**
 - Receptor pode ser conhecido do emissor ou não
- Sinais em linguagens de programação
 - **Sinais assíncronos**, enviados imediatamente (kill, suspend, ...) ou de forma diferida (eventos em COM+) para processos concorrentes, usando primitivas do sistema operacional
 - **Sinais síncronos**, em que a geração do evento provoca a chamada de rotinas que tratam o evento (*event handlers*) no mesmo processo, usando primitivas das linguagens de programação
 - **Exceções**, representadas por objetos lançados com "throw" e apanhados com "catch", em que o lançamento da exceção provoca o retorno automático de vários níveis de chamadas e o salto para uma rotina de tratamento de exceções no mesmo processo

Exemplo de hierarquia de sinais



Chamadas

- Um chamada representa a invocação de uma operação
 - Para quem faz a chamada (ponto de vista do atuador), trata-se de uma ação
 - Para quem testa se a operação foi chamada (ponto de vista do observador), trata-se de um evento
- Um chamada é geralmente síncrona
 - Quando uma operação de um objeto invoca uma operação em outro objeto, o controle passa para o 2º objeto
- Nome do evento tem a sintaxe da invocação da operação
 - Exemplo: insert(record)
 - Diferença entre evento e ação por contexto

Eventos temporais

■ Evento de **tempo relativo**:

- Representa a passagem de um certo tempo desde um certo instante
- Notação: **after** (*período-de-tempo*)
- Por omissão, tempo conta desde a entrada no estado de origem da transição a que está associado o evento
- Também se usa a palavra chave **timeout** quando não se quer precisar o período de tempo
- Exemplo: **after (5 segundos)**

■ Evento de **tempo absoluto**:

- Representa a chegada a um certo instante de tempo (data e/ou hora)
- Notação: **when** (*instante-de-tempo*)
- Exemplo: **when (11:59PM)**

Eventos de mudança

- Um evento de mudança (*change event*) é um evento que representa o fato de uma condição se tornar verdadeira
- Notação: **when (*expressão booleana*)**
- A expressão booleana pode ser usada para marcar um tempo absoluto ou para o teste contínuo de uma expressão
- Exemplo: when (altitude < 1000)

Transições

- Uma **transição** é uma relação entre dois estados indicando que um objeto no 1º estado realizará uma certa **ação** (opcional) e passará ao 2º estado *quando* um **evento** especificado ocorrer se uma **condição** especificada (opcional) for satisfeita
- Duas transições a sair do mesmo estado devem ter eventos diferentes ou condições mutuamente exclusivas, para que o diagrama de estados seja determinístico
- Se, num dado estado, ocorrer um evento que não corresponde a nenhuma transição, nenhuma transição é disparada e o evento é ignorado
- **Transição automática:** transição sem evento
 - Tem como evento implícito o fim da atividade associada ao estado de origem
 - Se tiver uma condição, fica à espera que a condição seja satisfeita

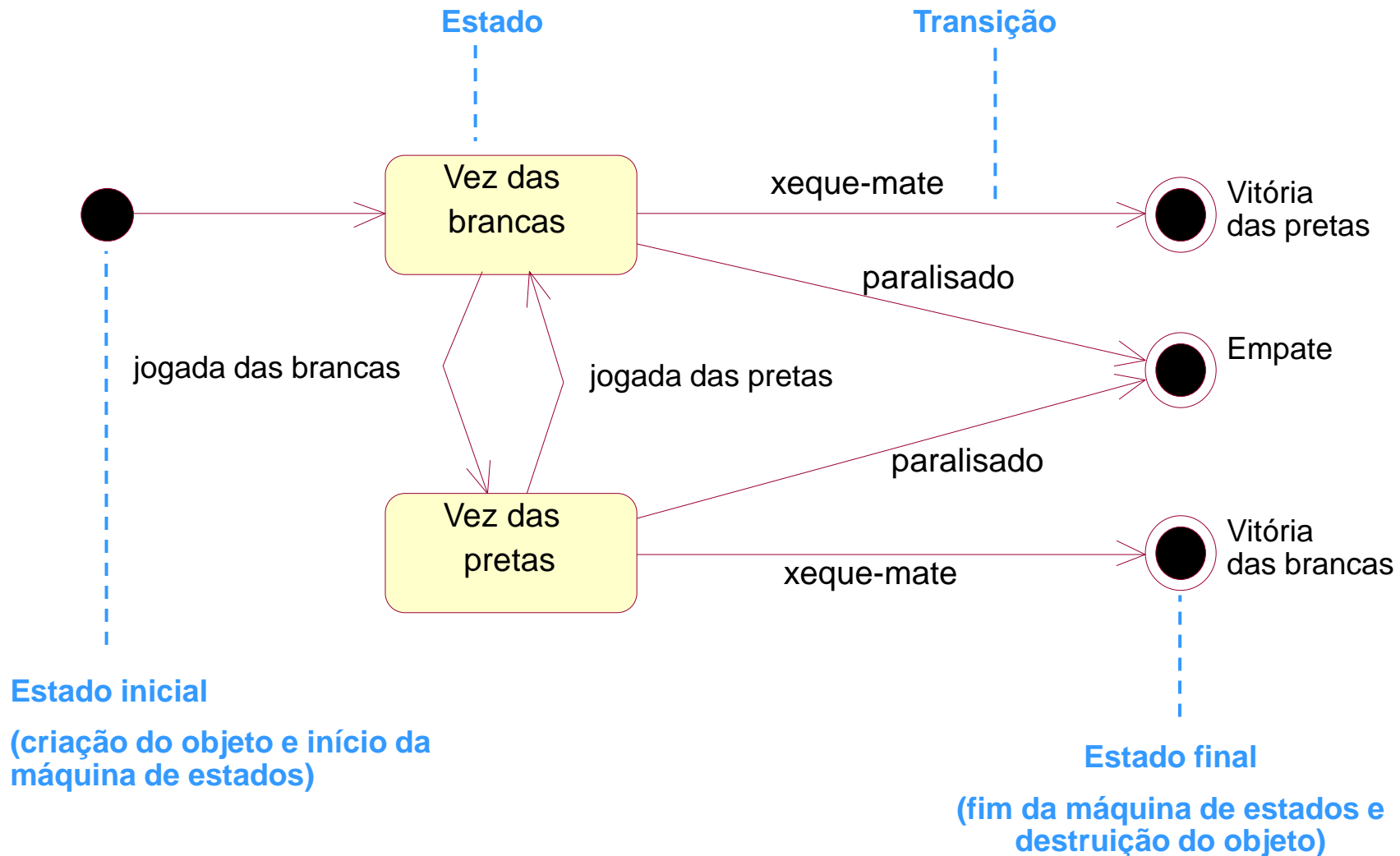
Atividades

- Uma **atividade** é uma execução contínua (*ongoing execution*) não atômica numa máquina de estados
 - Tem duração
 - Pode ser interrompida
 - Pode ter fim (termina por si só) ou não (só termina se for interrompida)
- É associada a estados
 - Inicia-se ao entrar no estado
 - Termina por si só ou é interrompida na saída do estado (causada por um evento)
- Exemplo (em um telefone): do / dá sinal de discagem

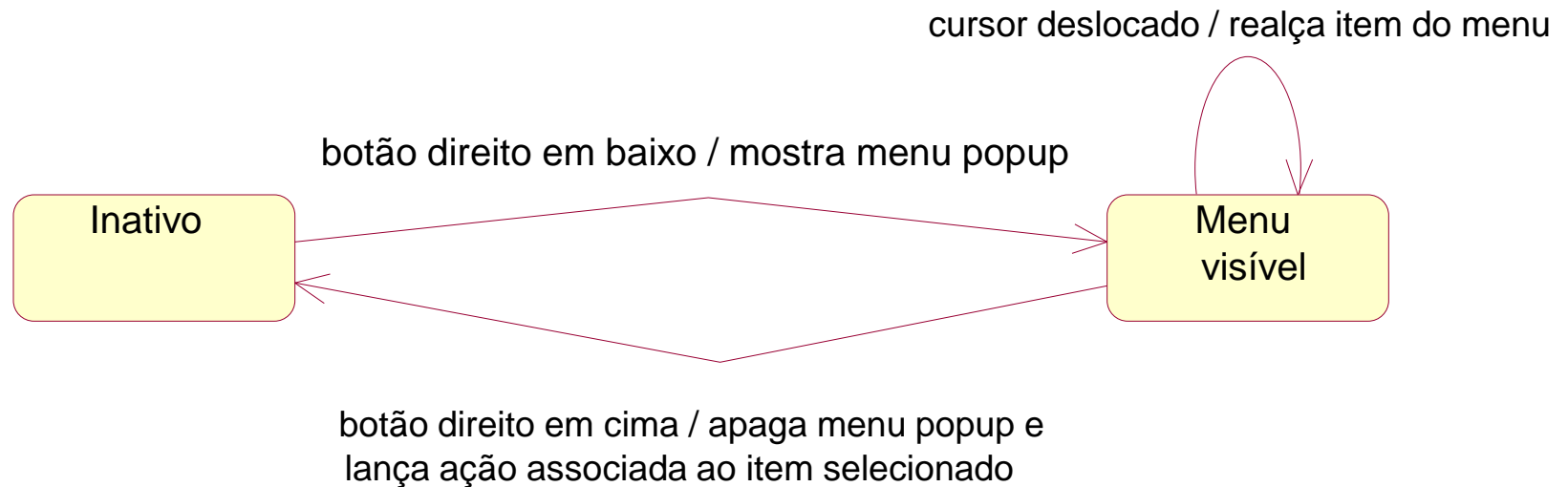
Ações

- Uma **ação** é uma computação atômica executável, que resulta numa mudança de estado ou no retorno de um valor
 - A mudança de estado pode ser em outro objeto
 - Não tem duração (pelo menos no modelo) e não pode ser interrompida
 - Ocorre em resposta a um evento
 - Exemplo (em um telefone): descansa o fone / pára sinal de discagem
 - Ações são associadas a transições (mais comum) ou estados
- Ação à entrada num estado: **entry/ação**
 - equivale a associar a ação a cada transição que entra no estado
- Ação à saída de um estado: **exit/ação**
 - equivale a associar a ação a cada transição que sai do estado
- Ação em resposta a evento interno a um estado: **event/ação**
 - difere de auto-transição, porque não são executadas ações à saída e entrada e não é interrompida a atividade associada ao estado

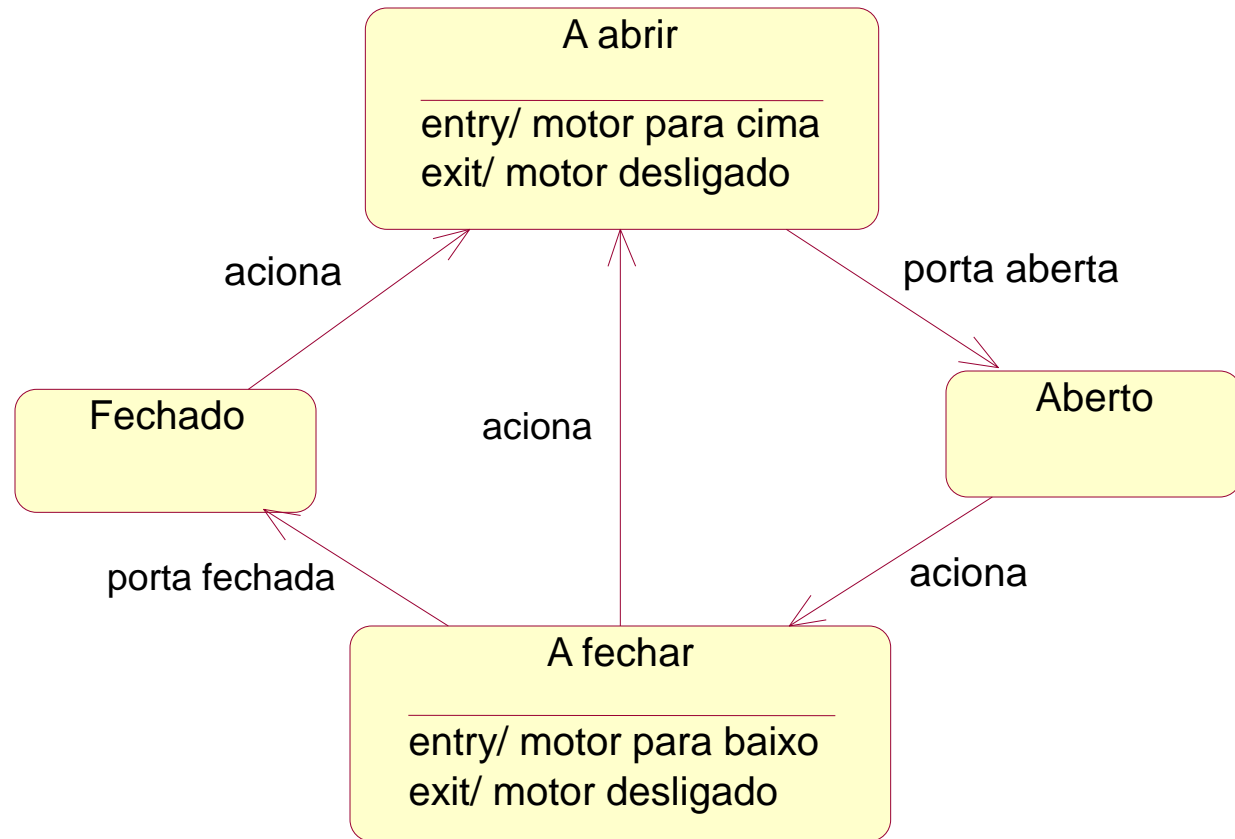
Exemplo: Jogo de xadrez



Exemplo: Menu *popup*

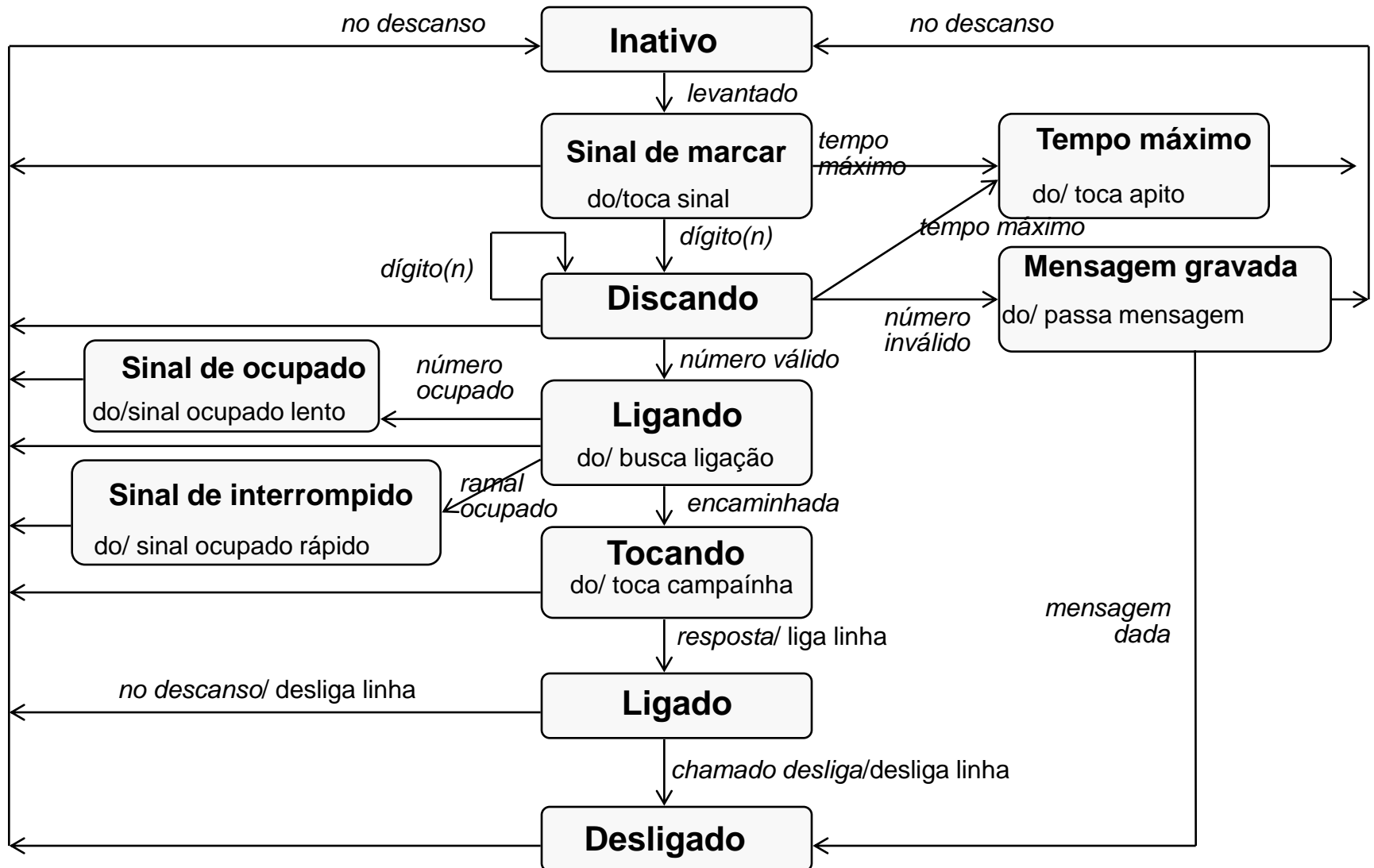


Exemplo: Porta com motor



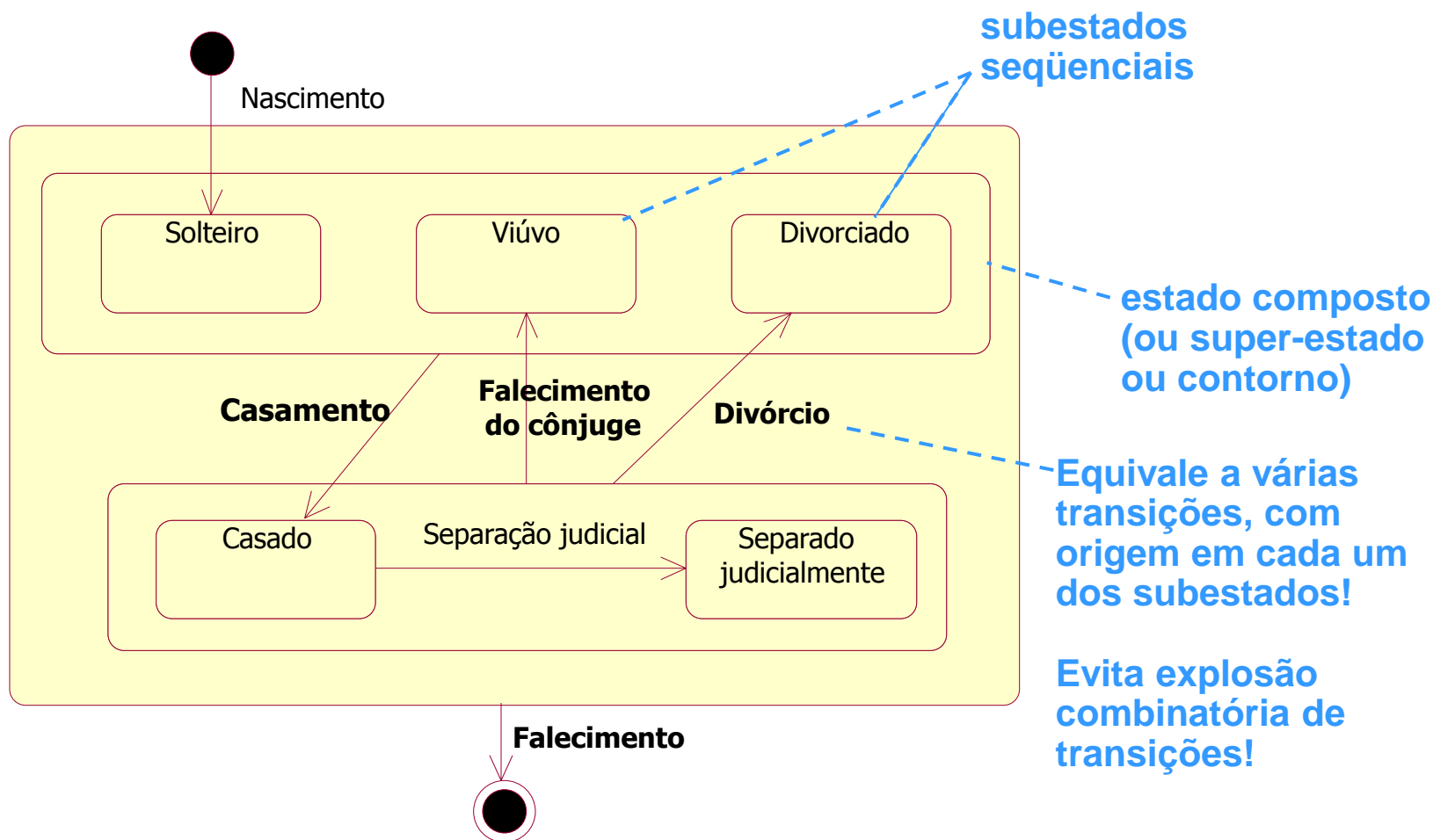
Como seria este diagrama com as ações associadas às transições?

Exemplo: Telefone (fazer chamadas)

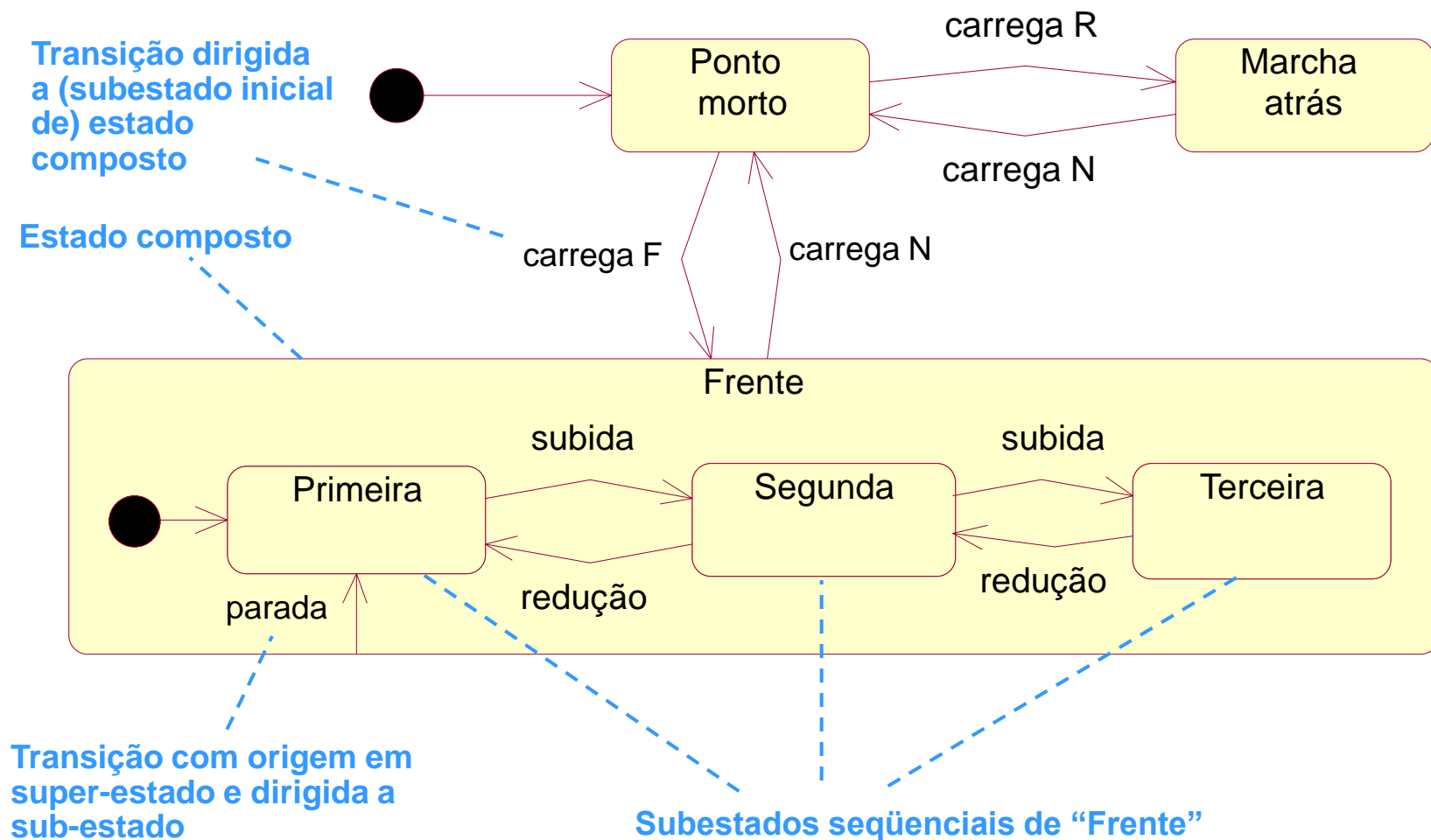


Subestados seqüenciais

Exemplo: Estado civil



Exemplo: Transmissão automática



Subestados seqüenciais

- Vários estados de um diagrama (e transições entre esses estados) podem ser agrupados num único **estado composto**
- Os primeiros estados passam a chamar-se **subestados seqüenciais**
 - **Subestados** – por estarem encaixados dentro de outro
 - **Seqüenciais** – por não ser possível estar em dois estados simultaneamente
- Estado composto pode ser visto simplesmente como um **contorno**
- Estados e transições agrupados no estado composto formam um **diagrama de estados encaixado**, podendo ter estado inicial e final
- Também se chama a isto **composição “ou”**
 - Estar no estado composto é estar no 1º subestado ou...ou no n-ésimo subestado
- Também se chama a isto **generalização de estados**
 - O estado composto é também chamado um **superestado**
 - Estar num subestado é estar também no respectivo superestado

Transições com estados compostos

- Podem-se definir transições com origem no estado composto
 - Equivale a repetir a mesma transição com origem em cada um dos subestados (exceto ●). Por outras palavras, os subestados **herdam** a transição.
 - **Simplifica-se** o diagrama porque se desenha 1 transição em vez de n transições (tantas quantos os subestados), evitando-se a explosão combinatória de transições!
 - A transição pode ser dirigida a um subestado ou a um estado externo
- Podem-se definir transições com destino ao estado composto
 - Equivale a definir a mesma transição com destino ao respectivo estado inicial (ou melhor, ao estado apontado por ●), que tem de estar definido
- Vantagem comum: abstração de detalhes do estado composto
 - O estado composto pode até ser detalhado separadamente
- No entanto, também se podem definir transições que atravessam o estado composto, com origem ou destino em subestados
 - O estado inicial é apenas um estado inicial por omissão

Propriedades de estados compostos

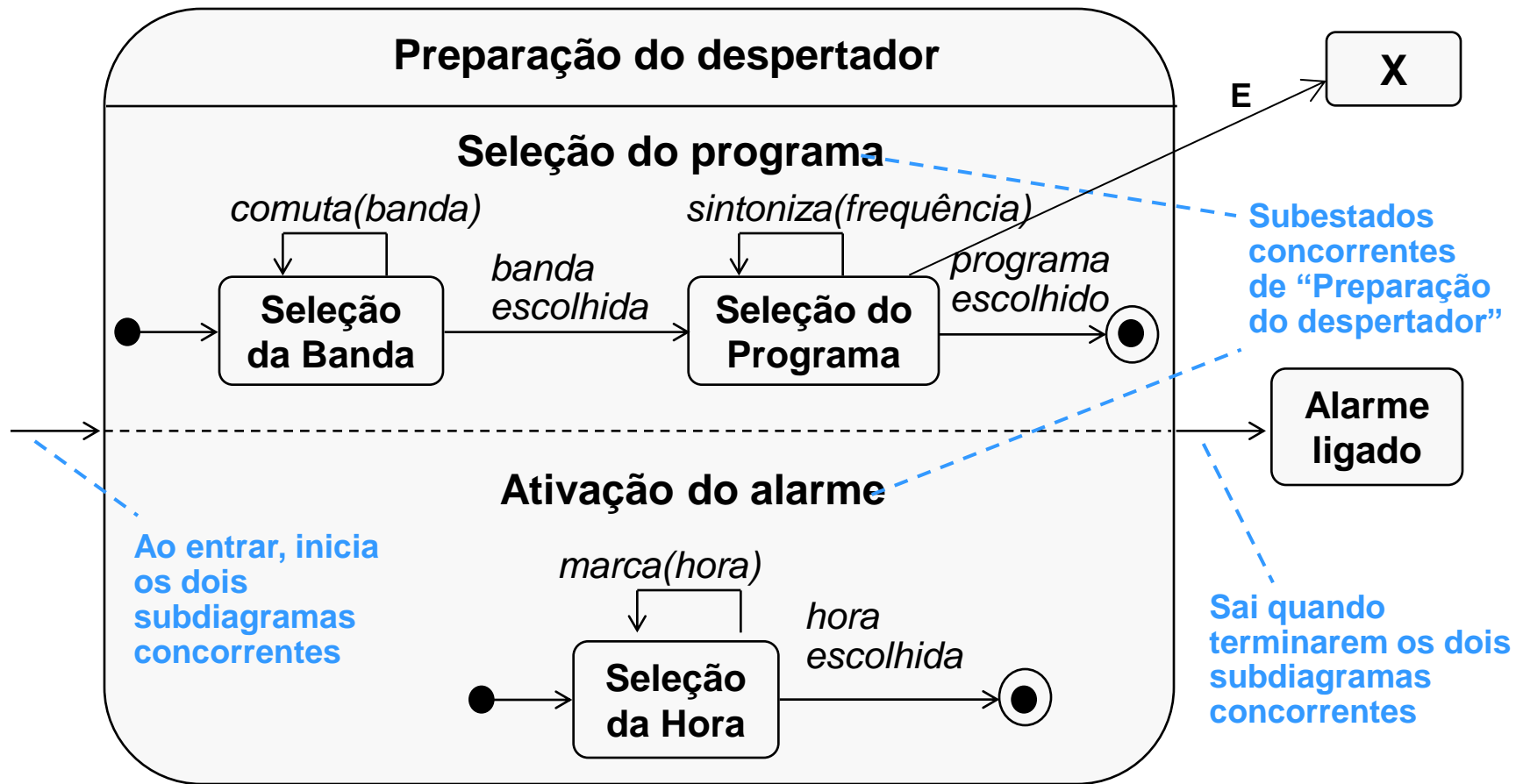
- Um estado composto pode ter todas as propriedades dos estados simples, mas normalmente só tem o nome, que mesmo assim é opcional
- Um estado composto pode ter ações à entrada e à saída
 - Ao entrar no estado composto, executa primeira as ações à entrada no estado composto, e depois as ações à entrada no subestado de destino
 - Ao sair do estado composto, executa primeira as ações à saída do subestado de origem, e depois as ações à saída do estado composto
- A atividade do estado composto é detalhada pelos subestados e transições entre eles, isto é, pelo diagrama de estados encaixado
- Em alternativa, pode-se dar um nome à atividade do estado composto (com “do/...”), e detalha-se essa atividade separadamente (com o mesmo diagrama de estados encaixado)
 - As ações à entrada e à saída são representadas no 1º diagrama
 - Uma atividade pode ser detalhada por um diagrama de estados ou atividade

Subestados concorrentes

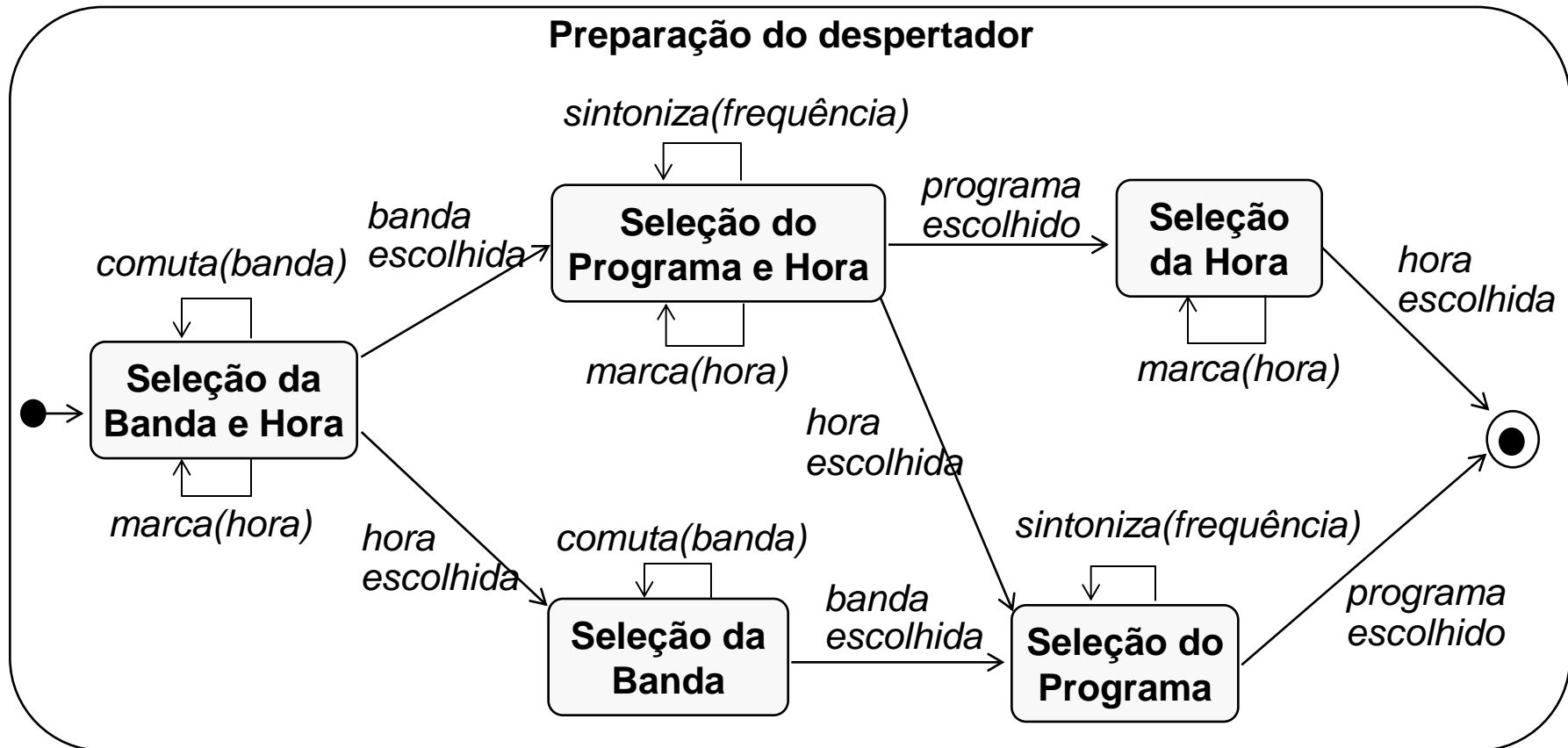
Subestados concorrentes

- Um estado (**estado composto**) pode ser dividido em duas ou mais regiões concorrentes, separadas por linhas com traço interrompido, representando **subestados concorrentes** (que, por sua vez, têm normalmente subestados seqüenciais)
- Subestados concorrentes correspondem a aspectos do objeto (grupos de atributos e ligações ou sub-objetos) que evoluem de forma mais ou menos independente
- Dependências entre regiões podem ser expressas através de condições de guarda (num componente testar o estado de outro)
- Também se chama a isto **composição “e”**:
 - Estar no estado composto é estar no 1º e ... e no nº subestado concorrente
- Notar que, a um nível mais global, os objetos (e portanto os respectivos diagramas de estados) são concorrentes entre si

Exemplo: Concorrência num objeto



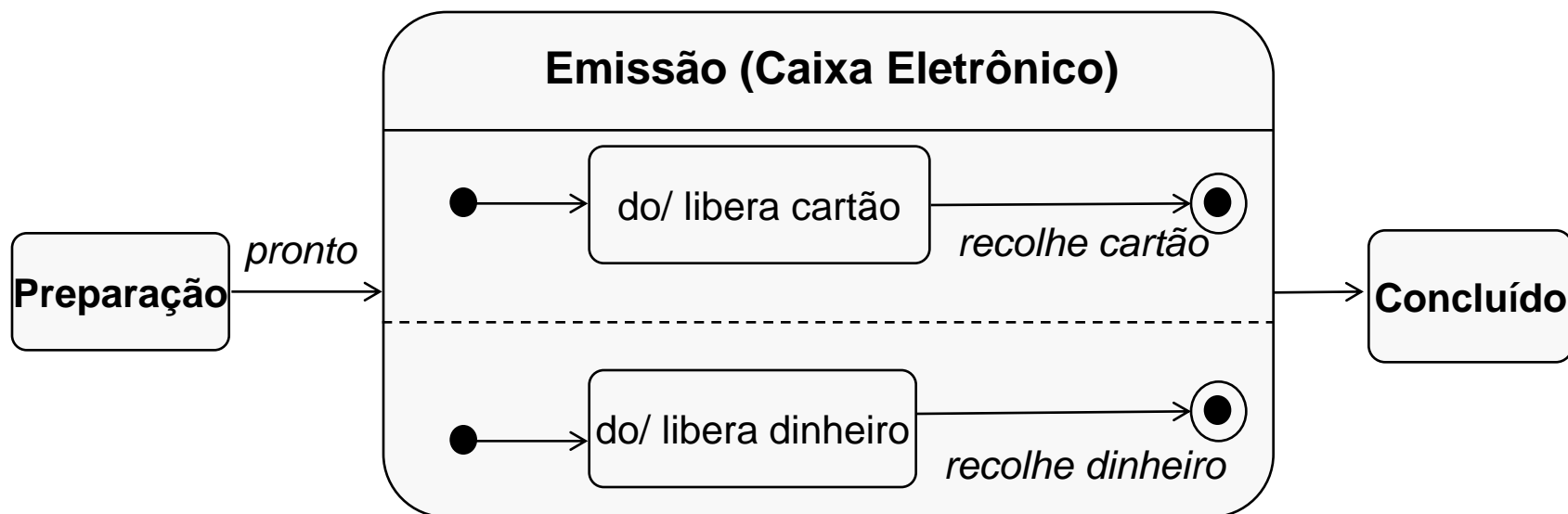
O mesmo exemplo sem subestados concorrentes



➡ Tem-se (quase) o produto cartesiano dos estados do diagrama anterior!

Sincronização: separação e fusão

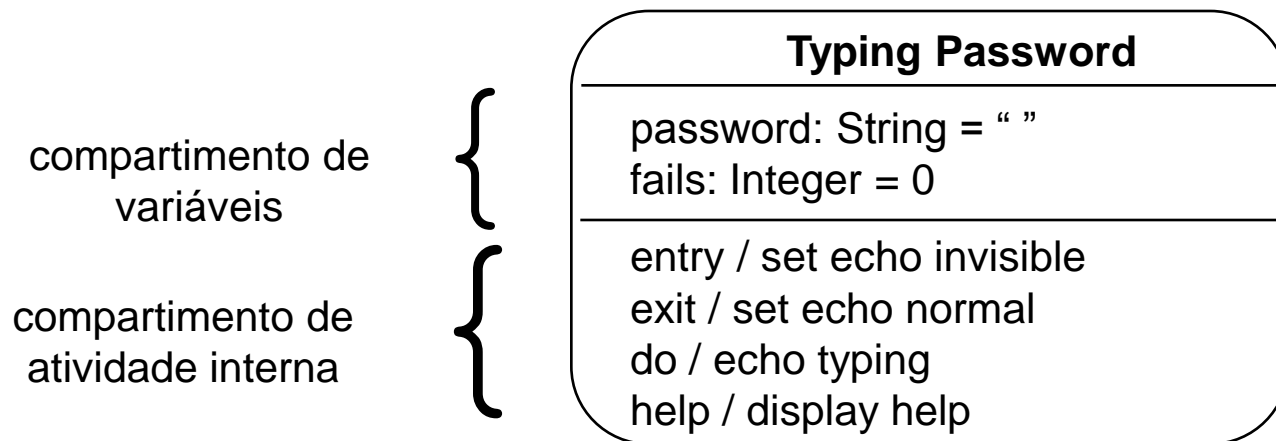
- **Separação** - quando um objeto passa a fazer várias atividades concorrentemente (ordem irrelevante)
 - transição para estado com subdiagramas concorrentes ativa cada um deles
- **Fusão** - quando as atividades concorrentes têm que terminar antes de passar ao estado seguinte
 - subdiagramas que não estejam na fusão são automaticamente terminados



Conceitos avançados

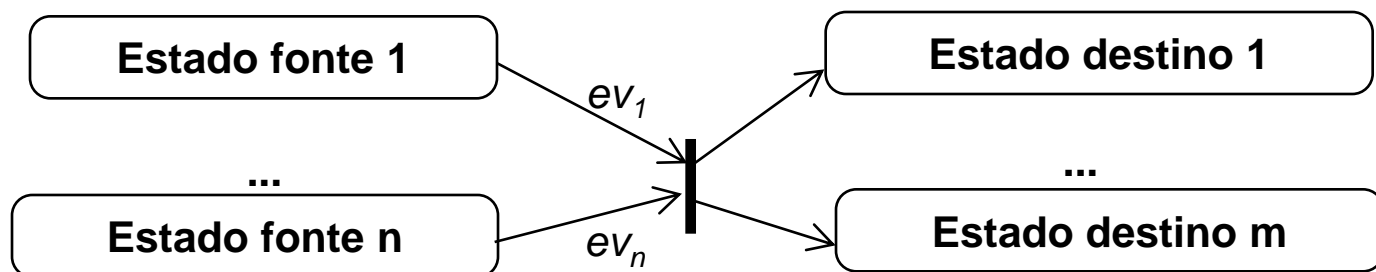
Variáveis num estado

- Um estado pode ter um compartimento de variáveis
- Estas variáveis são atributos da classe dona do diagrama de estados, distinguidos porque são usadas ou afetadas por ações no diagrama de estados
- Todos os compartimentos (nome, variáveis e atividade interna) são opcionais



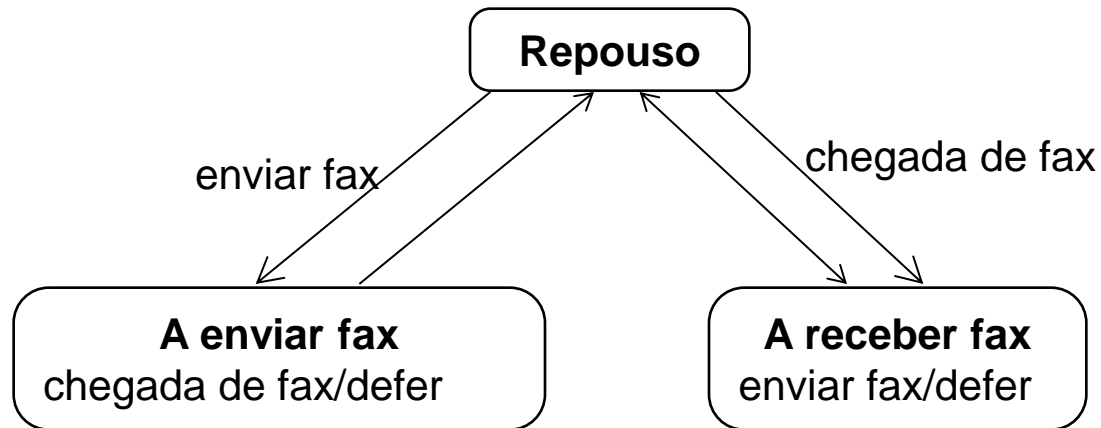
Transições complexas

- Uma **transição complexa** é uma transição com múltiplos estados fonte (normalmente de subdiagramas concorrentes) ou múltiplos estados de destino (normalmente de subdiagramas concorrentes)
- Representa uma **separação** e/ou **fusão** de controle em/de “fios” de controle concorrentes
- Aplicável mesmo sem subdiagramas concorrentes
- Desenha-se com uma barra forte (**barra de sincronização**):



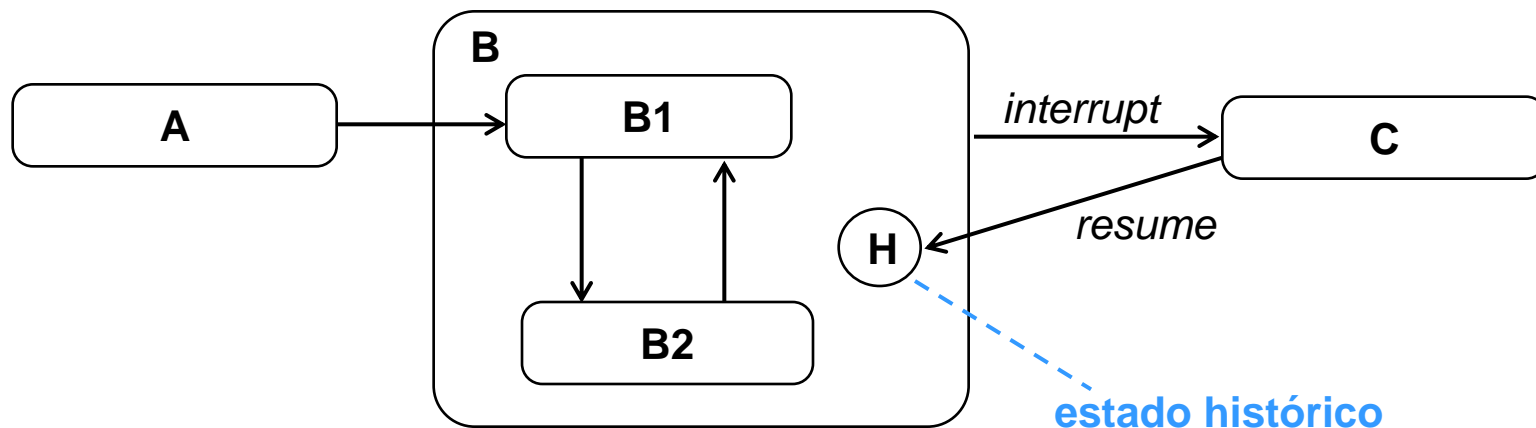
Eventos deferidos

- Uma ação possível em resposta a um evento é “**defer**”, que significa guardar o evento até chegar a um estado capaz de o “consumir”
- É aplicável num par “evento/ação” interno a um estado
- Exemplo (máquina de fax):



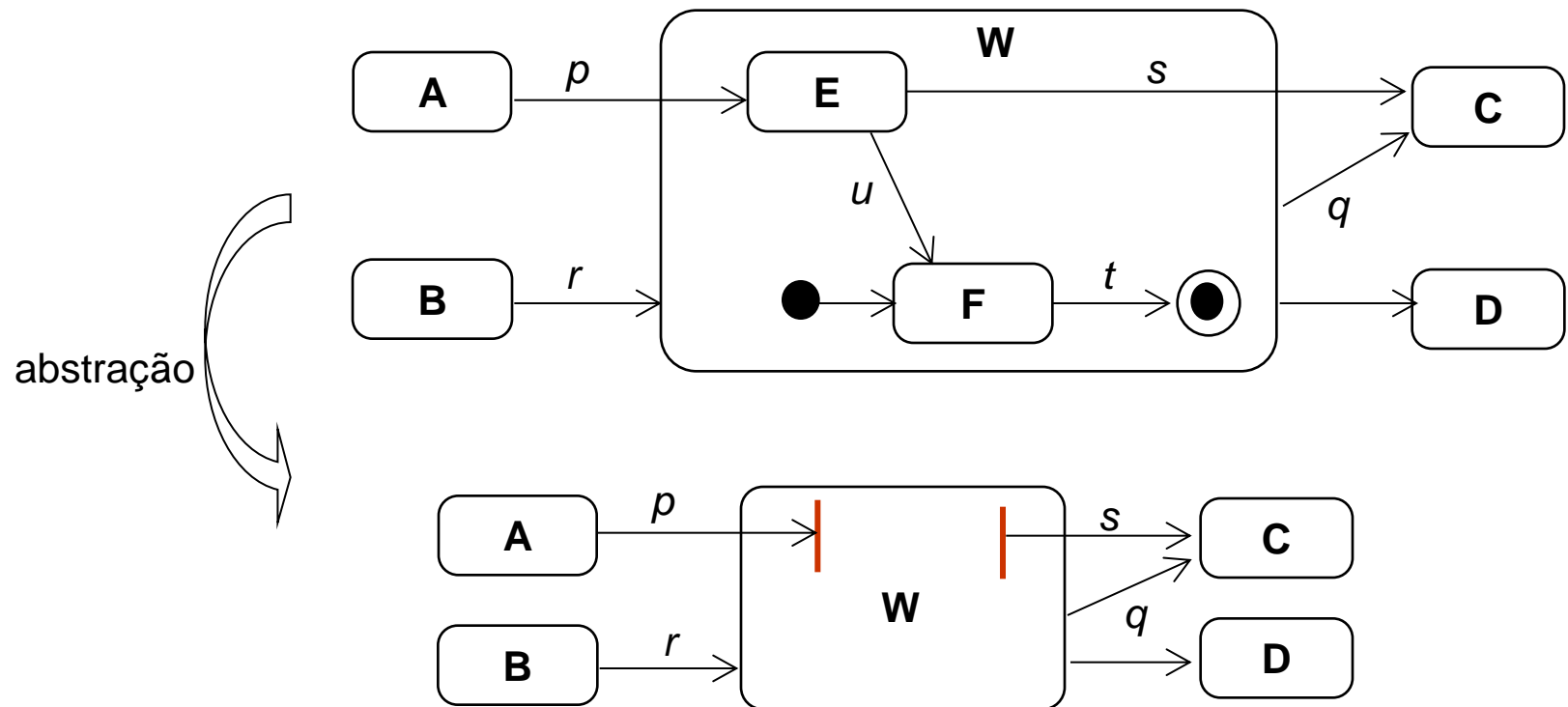
Estados históricos (*history states*)

- **Estado histórico** (dentro de um estado composto) – refere-se ao último subestado em que se encontrava o estado composto
- Útil para reentrar num estado composto no subestado em que se encontrava anteriormente
- Não é aplicável na primeira entrada



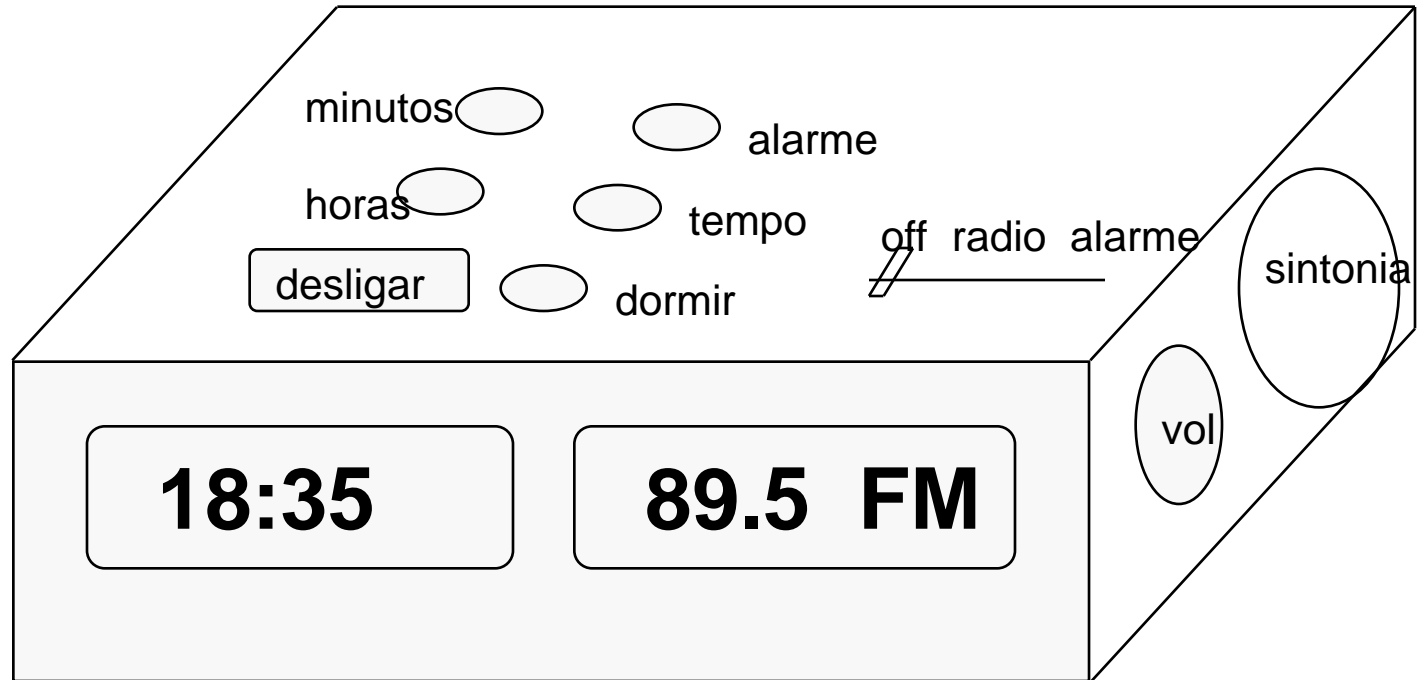
Resíduos de subestados (*stubs*)

- Aplicáveis quando se escondem os detalhes de um estado composto, mas há transições que o atravessam



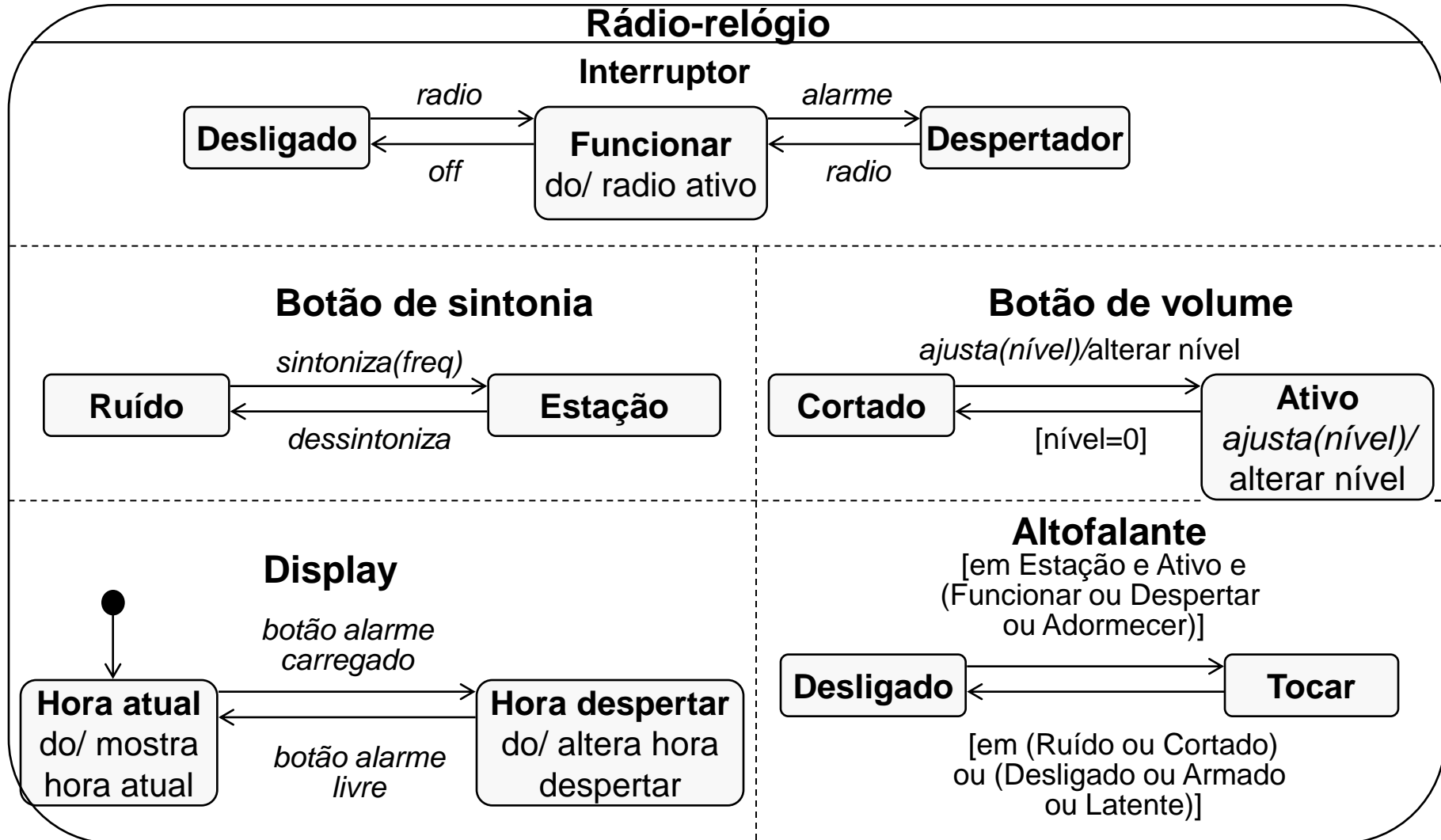
Exemplos e exercícios

Exemplo: Rádio-relógio (a rever)

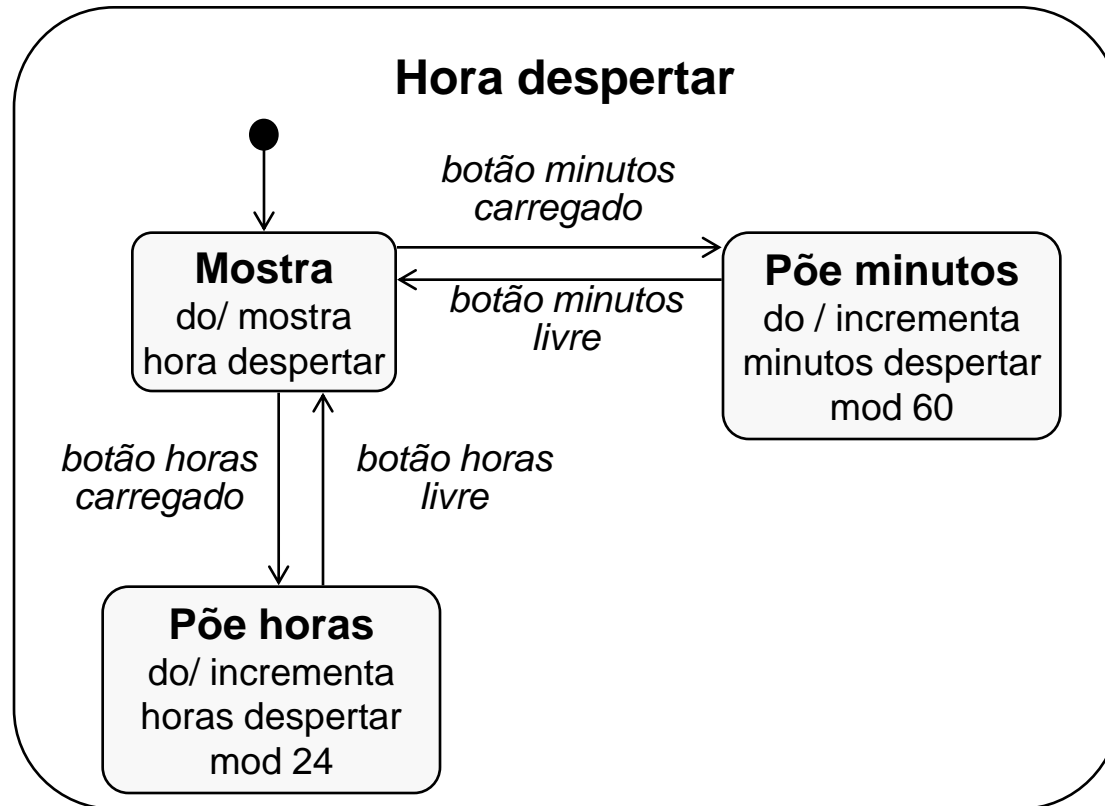


- obter o diagrama de estados
 - o botão *dormir* faz tocar durante uma hora (*desligar* cala-o logo)
 - ao chegar o instante do alarme, começa a tocar durante 1 hora (*desligar* só desliga durante 10 minutos)

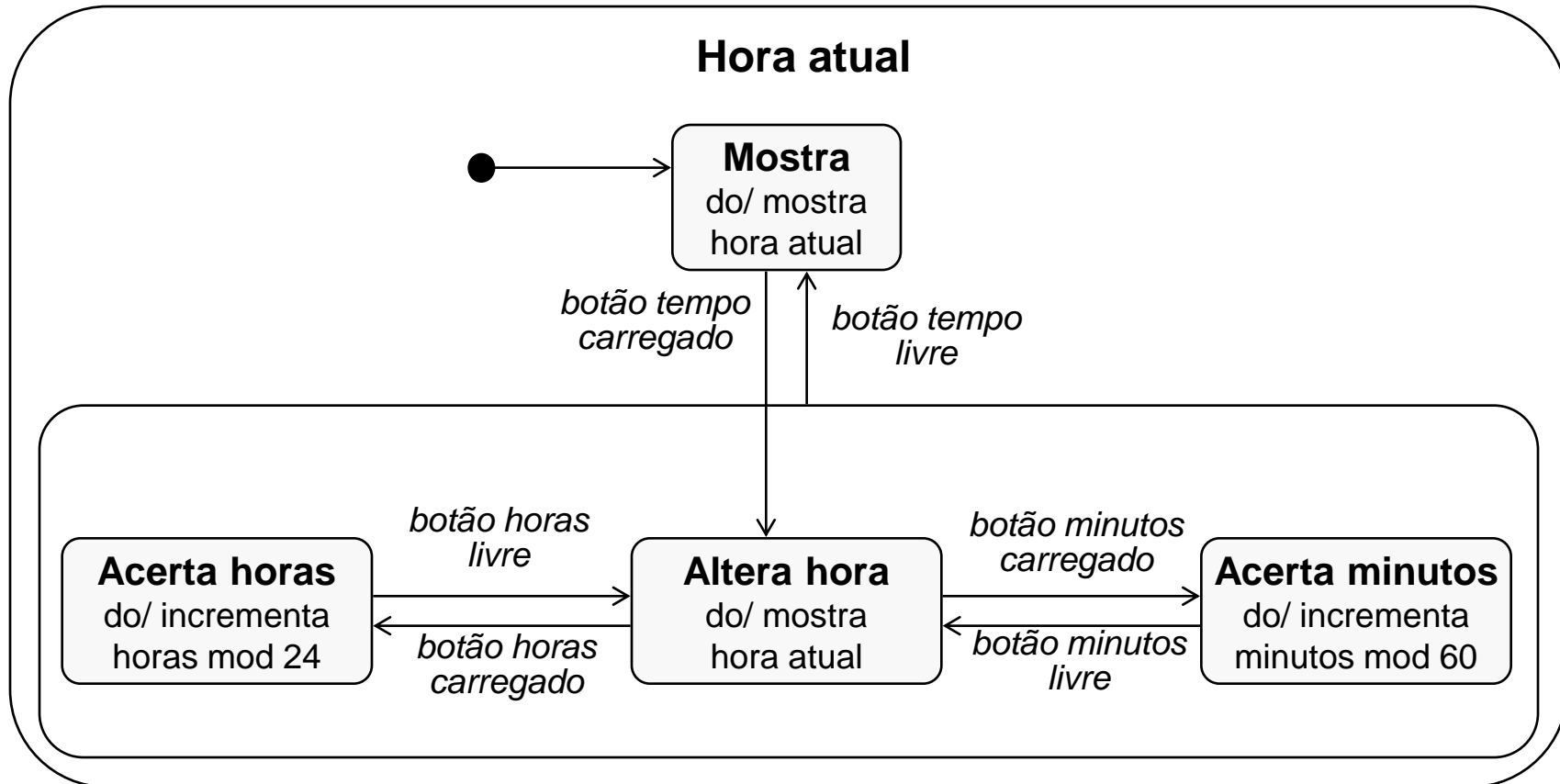
Diagrama principal



- Subdiagrama *Hora despertar*



Subdiagrama *Hora atual*



Subdiagrama *Despertador*

