

# 09. Comportamenti in UML

IS 2024-2025



**Laura Semini, Jacopo Soldani**

Corso di Laurea in Informatica

Dipartimento di Informatica, Università of Pisa

# DIAGRAMMA DI MACCHINA A STATI

Grafo **stati-transizioni** che descrive il **comportamento** delle istanze di una **classe**

- **stati** significativi in cui si può trovare un oggetto durante la sua vita
- come da ciascuno di questi stati l'oggetto può passare (**transire**) ad un altro

Le **transizioni** si **attivano** in risposta ad un **evento**, ad esempio

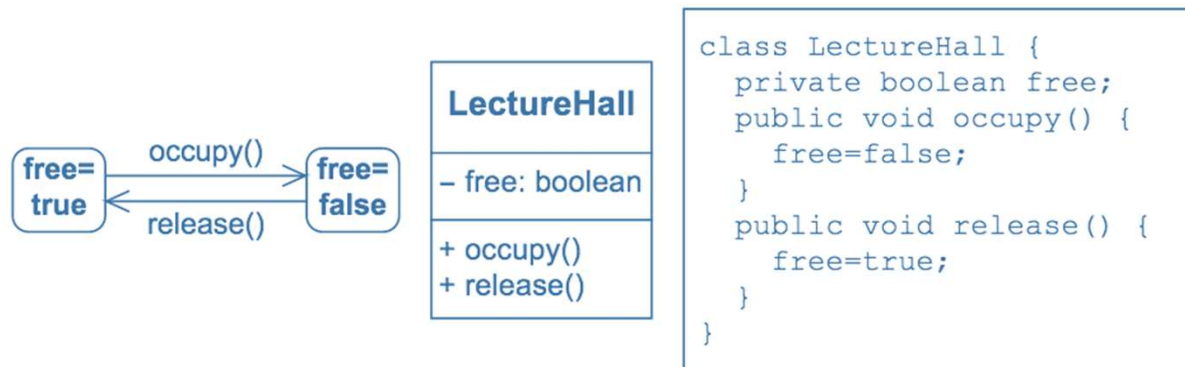
- messaggi inviati da altri oggetti
- eventi generati internamente

# LIVELLO DI DETTAGLIO

La modellazione di un'aula



verrà implementata come



# STATI E TRANSIZIONI



// Stato iniziale



// Stato S



// Stato finale

e le transizioni?



## STATI E TRANSIZIONI (CONT.)



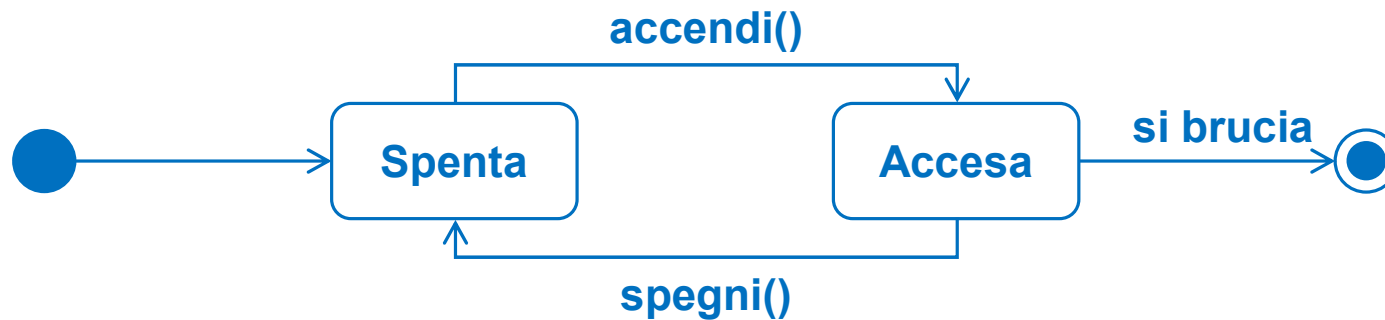
eventi ::= evento | evento, eventi (disgiunzione)  
azioni ::= azione | azione; azioni (sequenza)

La **transizione** uscente da uno stato

- definisce risposta dell'oggetto all'occorrenza di un **evento**,
- viene presa solo se la **condizione** è vera quando occorre l'evento,
- comporta l'esecuzione delle **azioni** specificate

NB: Condizione e azioni sono **opzionali**, mentre deve esserci **almeno un evento**

# UN ESEMPIO

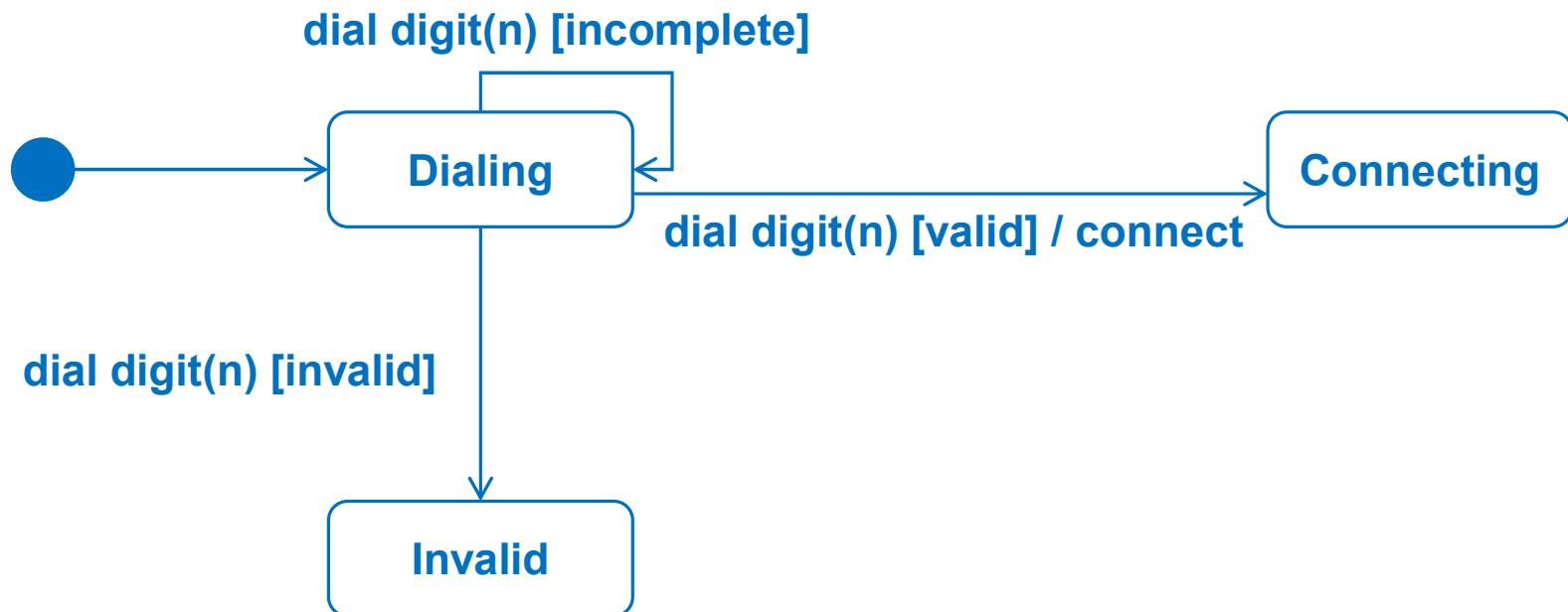


Lampadina
- accesa: boolean = false
+ accendi() + spegni()

Gli **eventi** (anche se non tutti) corrispondono alle **operazioni** offerte dalla classe

# UN ALTRO ESEMPIO

Classe Telefono, operazione **dial digit(n)**



# EVENTI

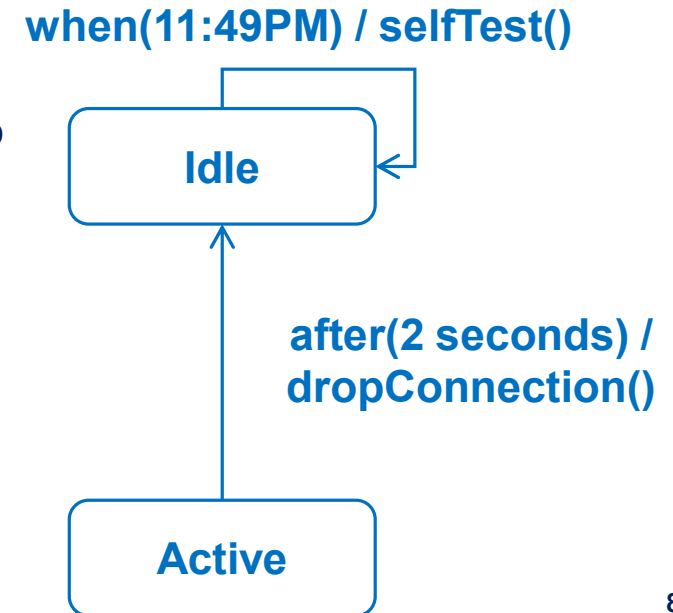
Un evento occorre **istantaneamente** e va introdotto solo se ha degli effetti

Se un evento occorre in uno **stato** che

- **non ha transizioni** per quell'evento, viene **ignorato**
- **ha più transizioni** per lo stesso evento, ne viene scelta una in modo **non-deterministico**

## Tipi di evento

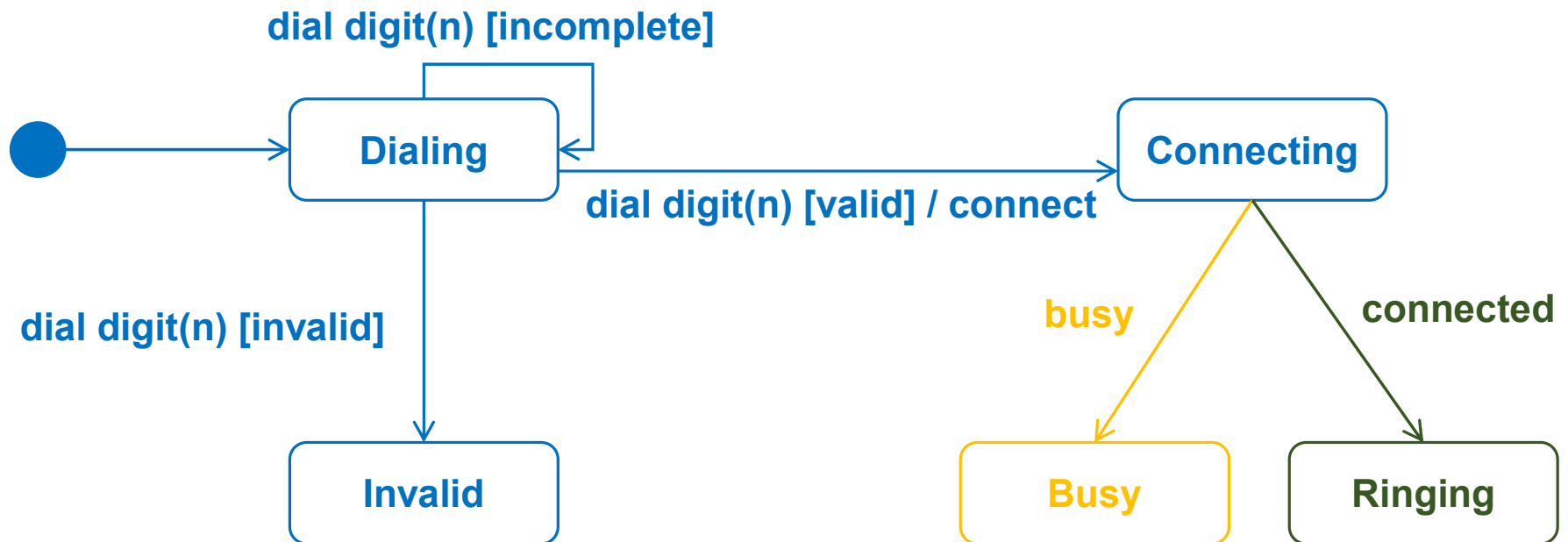
- **Operazione o segnale** → ricezione di una chiamata di metodo o un segnale con parametri (e tipi) compatibili, es. **op(a:T)**
- Evento di **variazione** «**when(exp)**» → l'espressione **exp** (tempo assoluto o condizione) diventa vera
- Evento **temporale** «**after(t)**» → l'oggetto è stato fermo per un tempo **t** nello stato di partenza



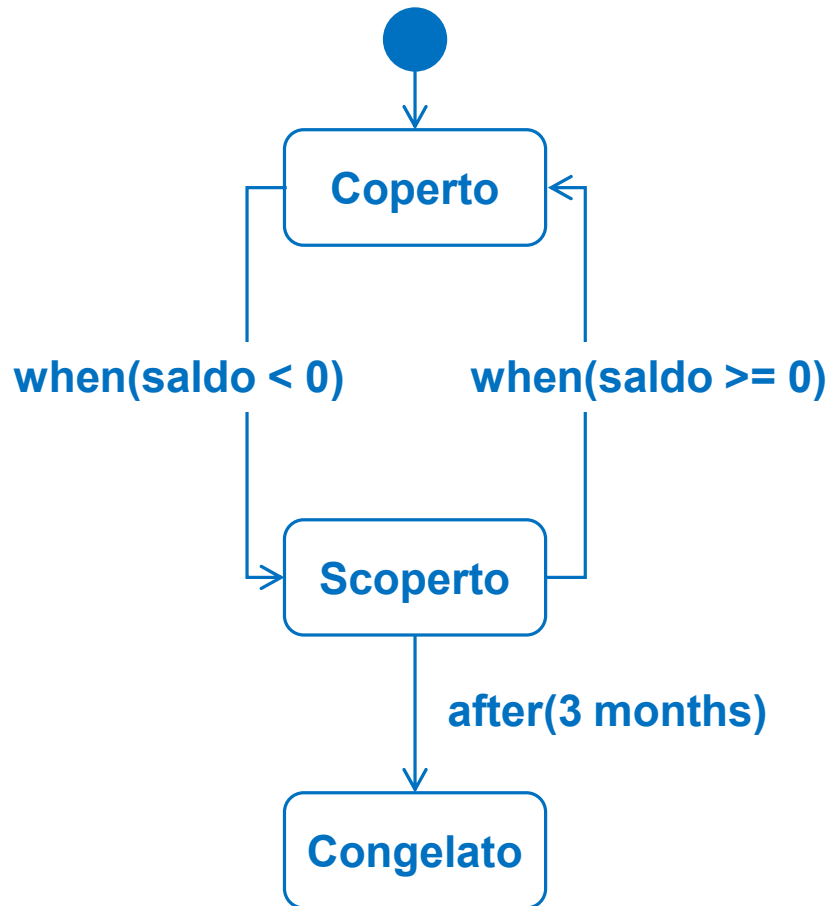


# ESEMPIO: SEGNALI

Due possibili segnali: **busy** e **connected**



# EVENTI DI VARIAZIONE E TEMPORALI



## Eventi di **variazione**

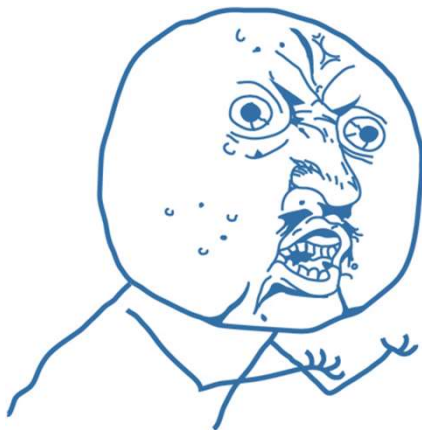
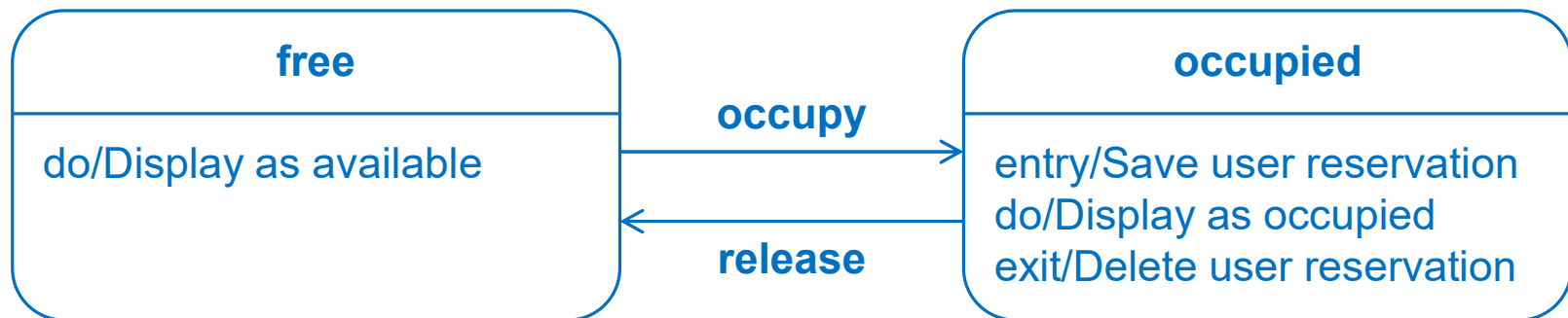
- Un evento occorre istantaneamente
- Una condizione non è istantanea
- Diventa «istantanea» nel momento in cui è vera

## Eventi **temporali**

- Transizione dopo un dato intervallo di tempo
- Nell'esempio, dopo che l'oggetto è stato *Scoperto* per 3 mesi, diventa *Congelato*

# UN (NUOVO) ESEMPIO

Macchina a stati che rappresenta l'occupazione di un'aula



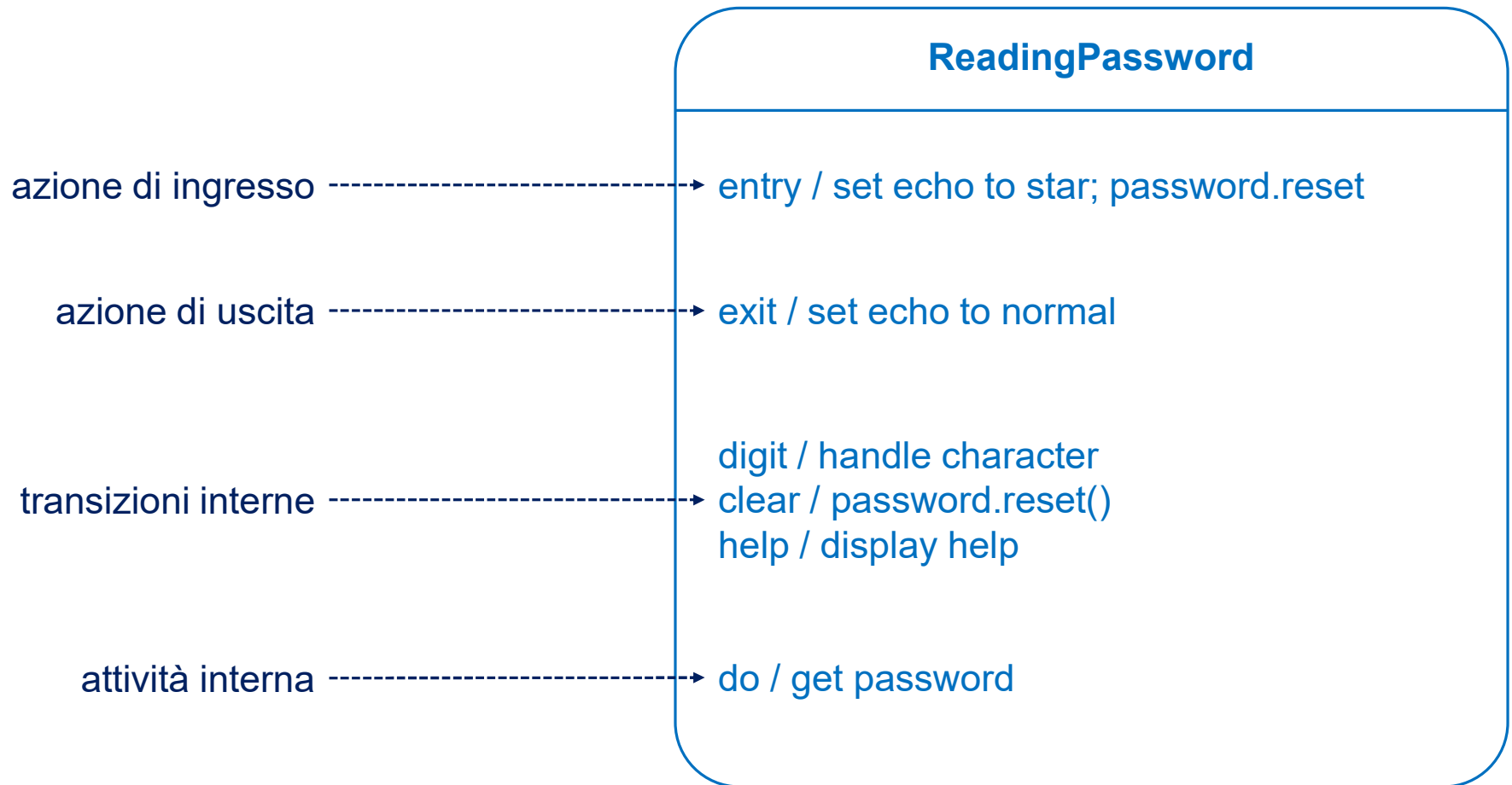
do?  
entry?  
exit?

# ENTRY, EXIT, DO (E TRANSIZIONI INTERNE)

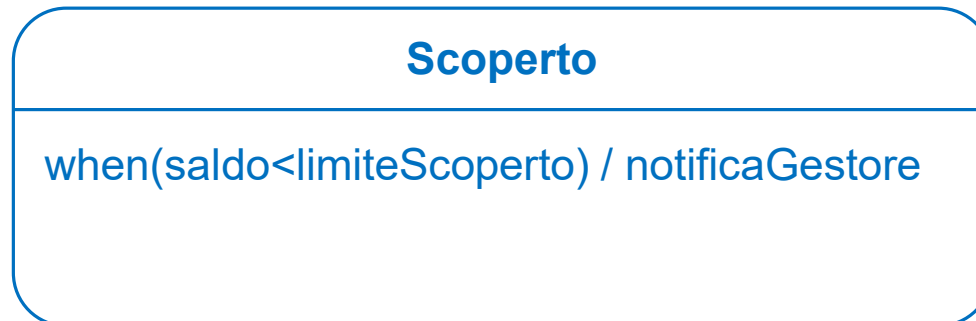
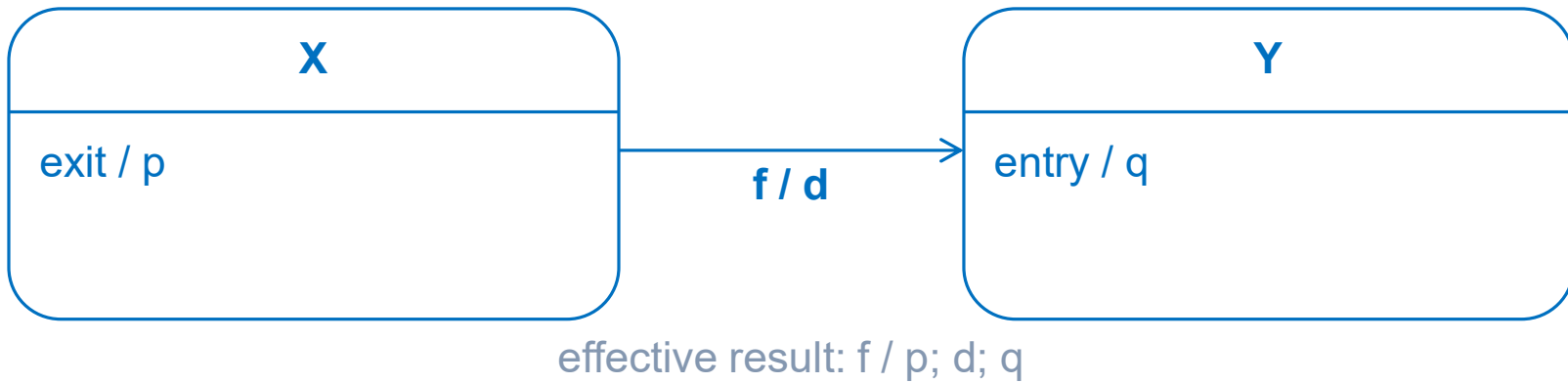
- **entry** → azione di **entrata**, eseguita all'ingresso in uno stato
- **do** → azione **interna**, eseguita in modo **continuato** mentre l'oggetto si trova in quello stato
  - non necessita di eventi scatenanti
  - a differenza di tutte le altre azioni (che sono atomiche)
    - consuma del tempo
    - può essere interrotta (quando un evento fa uscire dallo stato)
- **exit** → azione di **uscita**, eseguita all'ingresso in uno stato

Ma ci sono anche le **transizioni interne**, eseguite in risposta ad un **evento**

# ESEMPIO



## ALTRI ESEMPI



# STATO COMPOSITO (SEQUENZIALE)

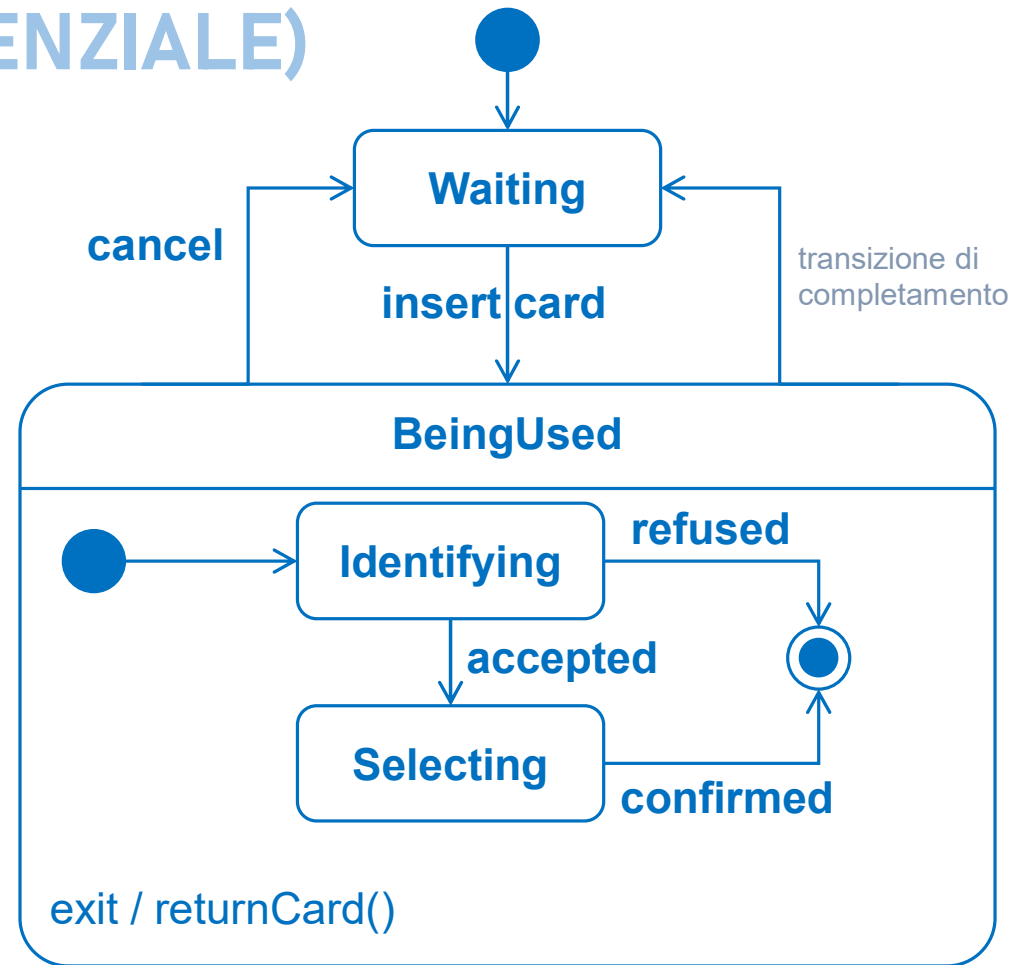
**Include** un'ulteriore **macchina a stati**

**Ingresso:** transizione che arriva nello stato

- se arriva sul **bordo**, si prosegue dallo **stato iniziale** della macchina interna
- se arriva su uno **stato interno**, si prosegue da tale stato (non nell'esempio)

**Uscita:** transizione che esce dallo stato

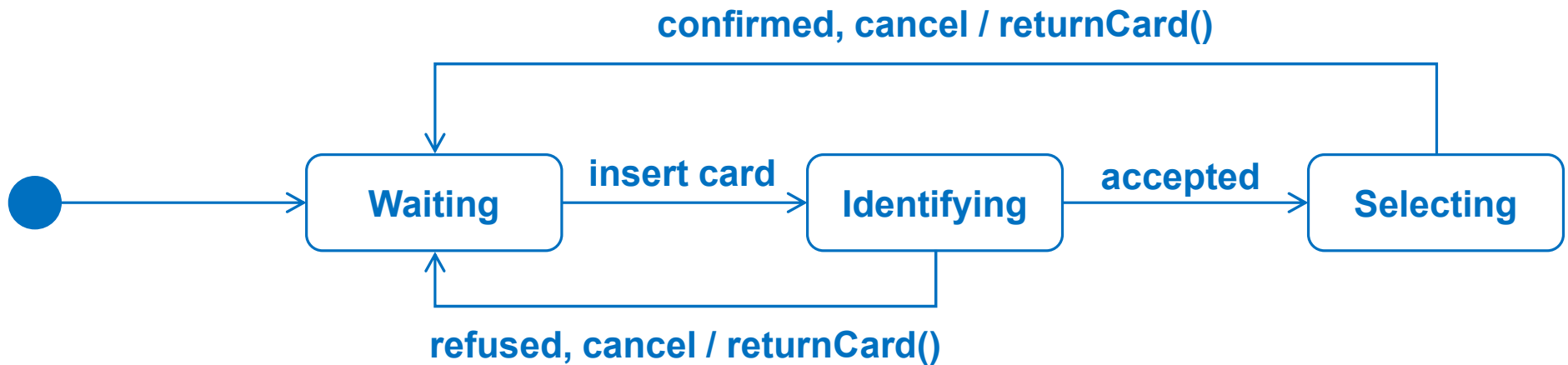
- una transizione (etichettata) che parte dal **bordo** è attivabile da **qualsiasi stato** interno
- dallo **stato finale** della macchina interna, si prosegue nella **transizione di completamento** (che non ha evento)
- transizioni uscenti da stati interni possono «**attraversare il bordo**» (non nell'esempio)



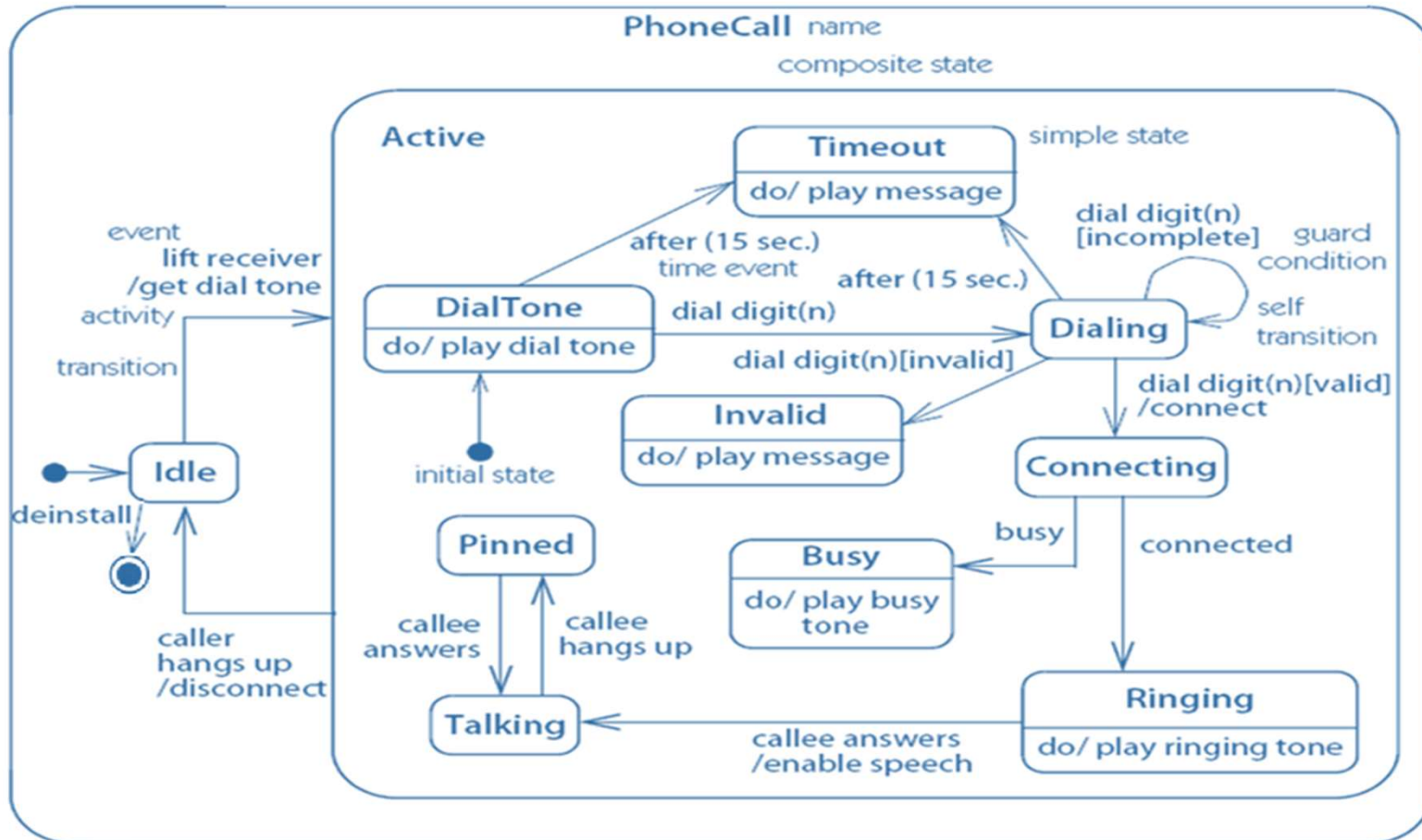
Fino a qui martedì 15



## ESEMPIO (SENZA LO STATO COMPOSITO)

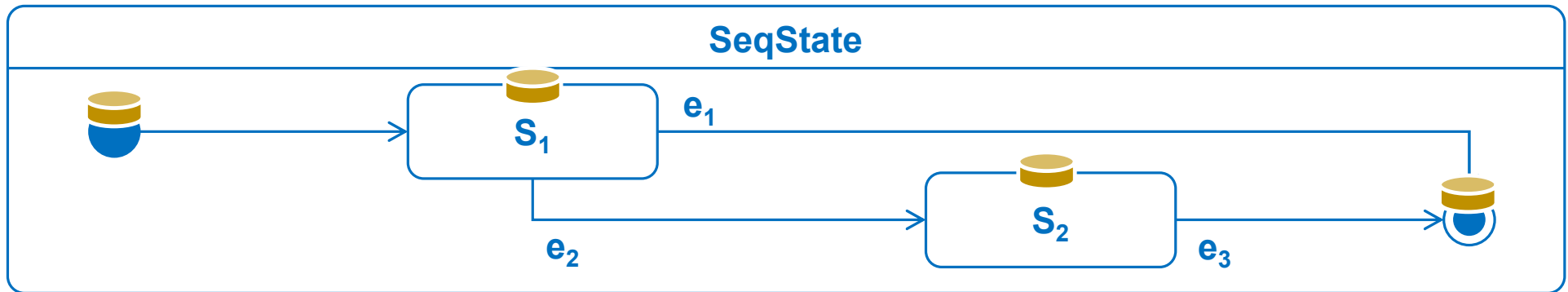


# ESEMPIO: STATO COMPOSITO (SENZA STATO FINALE)

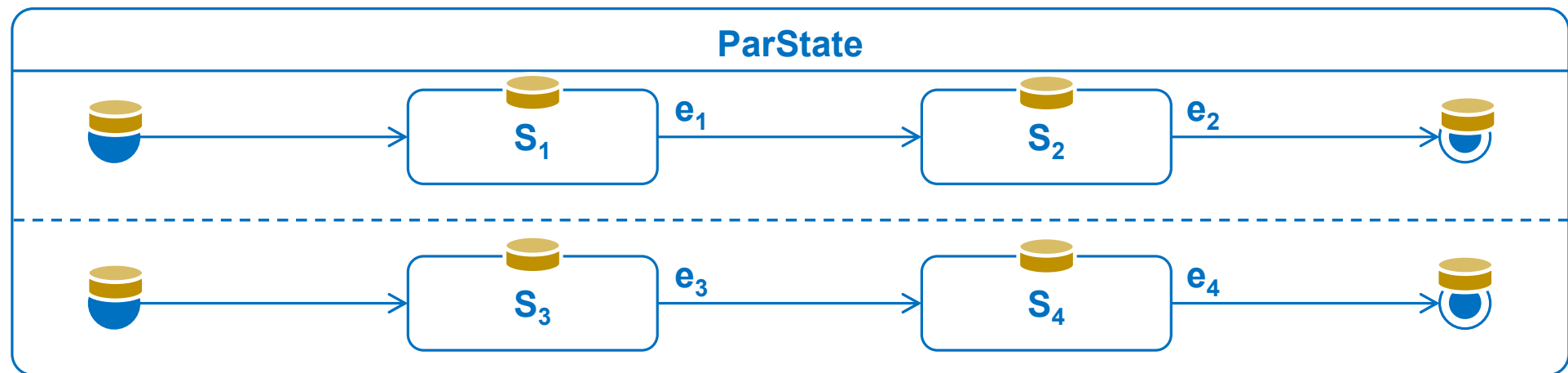


# STATI COMPOSITI SEQUENZIALI VS PARALLELI

**sequenziale**  $\Rightarrow$  solo un sottostato attivo in ogni istante



**parallelo**  $\Rightarrow$  più sottostati attivi contemporaneamente, uno per **regione**

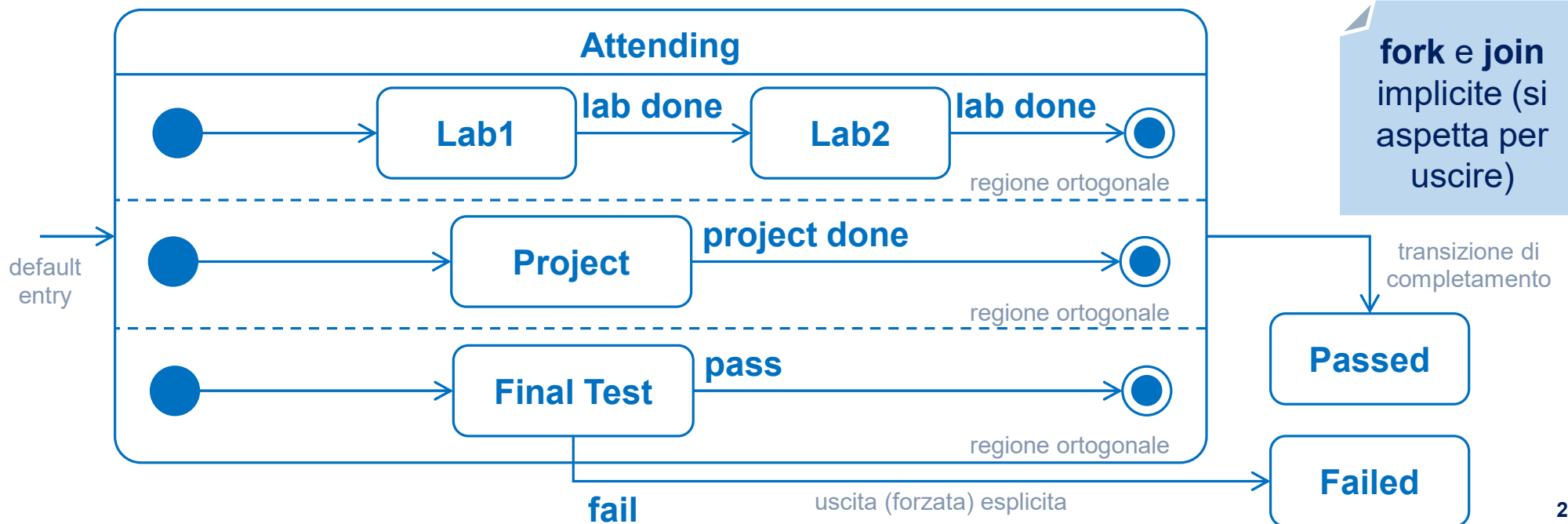


# STATO COMPOSITO (PARALLELO)

**Ingresso:** transizione (**default entry**) che arriva sul bordo e prosegue in **tutti i sottostati iniziali**

**Uscita:** transizione che esce dallo stato

- transizione di **completamento** attivata se si raggiungono **tutti i sottostati finali**
- se parte da un **sottostato**, è attivabile solo lì e **fa uscire da tutti i sottostati**
- se parte dal **bordo**, è attivabile sempre e **fa uscire da tutti i sottostati** (non nell'esempio)

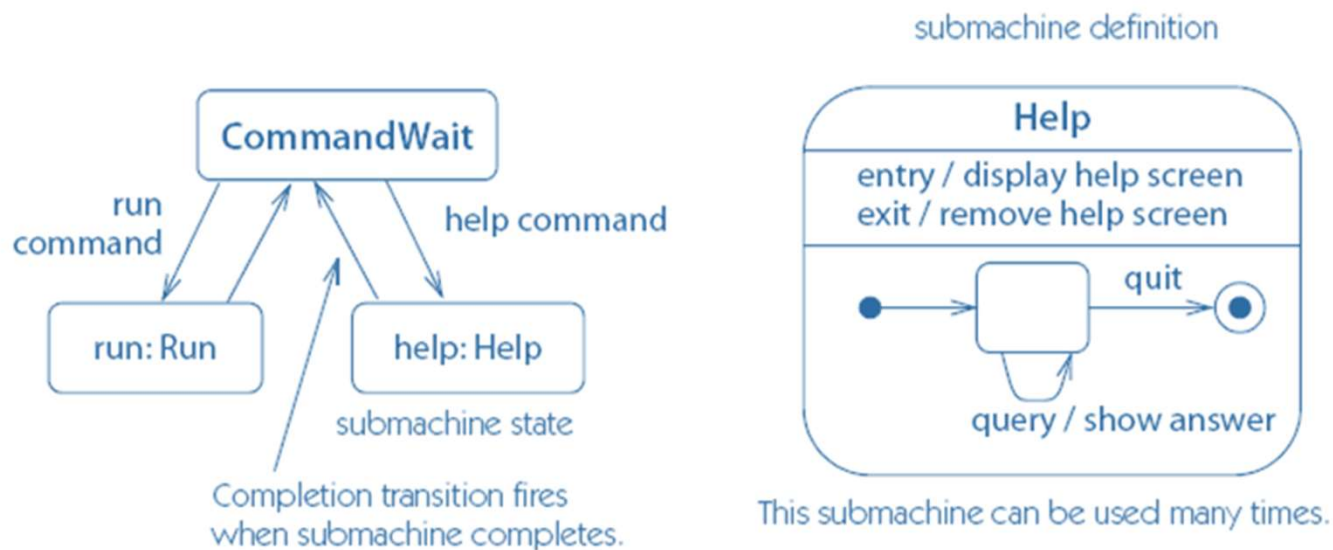


# SOTTOMACCHINE

Una **sottomacchina** consente di descrivere uno stato composito in un **diagramma a parte**

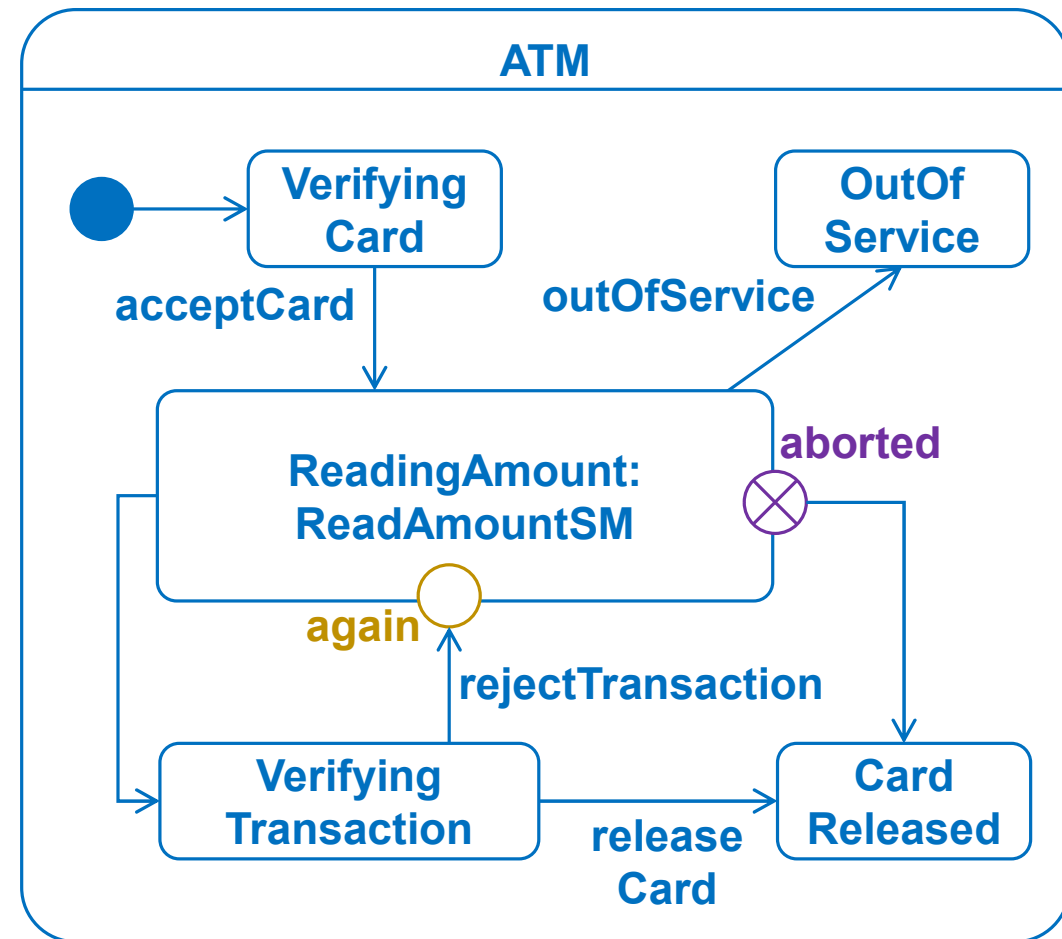
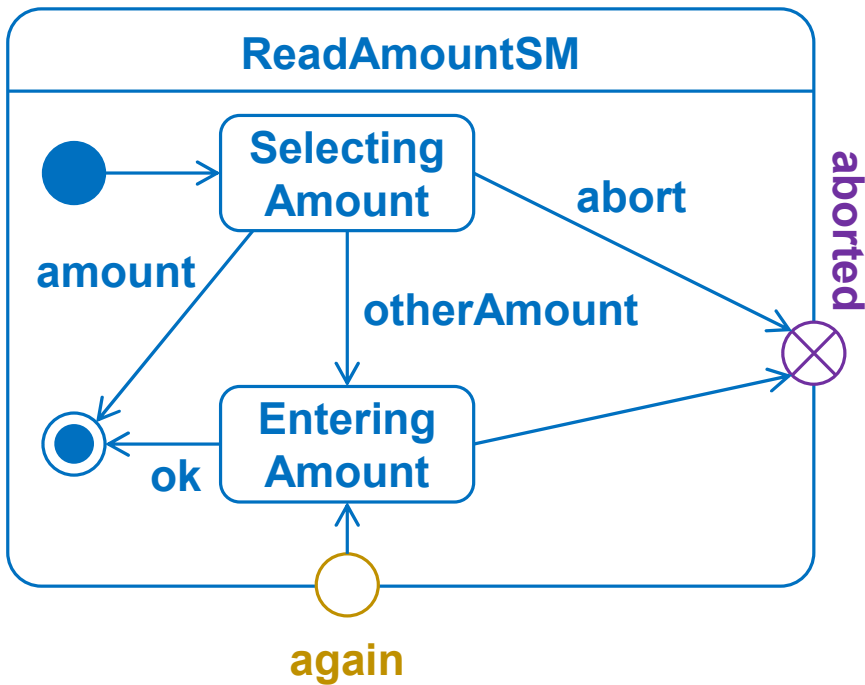
- migliore leggibilità
- favorisce riuso in altri contesti

La sottomacchina ha un **nome (tipo)**  $\Rightarrow$  le istanze di uso si indicano con **nomeIstanza:Tipo**



# SOTTOMACCHINE: PUNTI DI INGRESSO/USCITA

Definire **entry/exit** point per collegare le sottomacchine alle transizioni della macchina principale

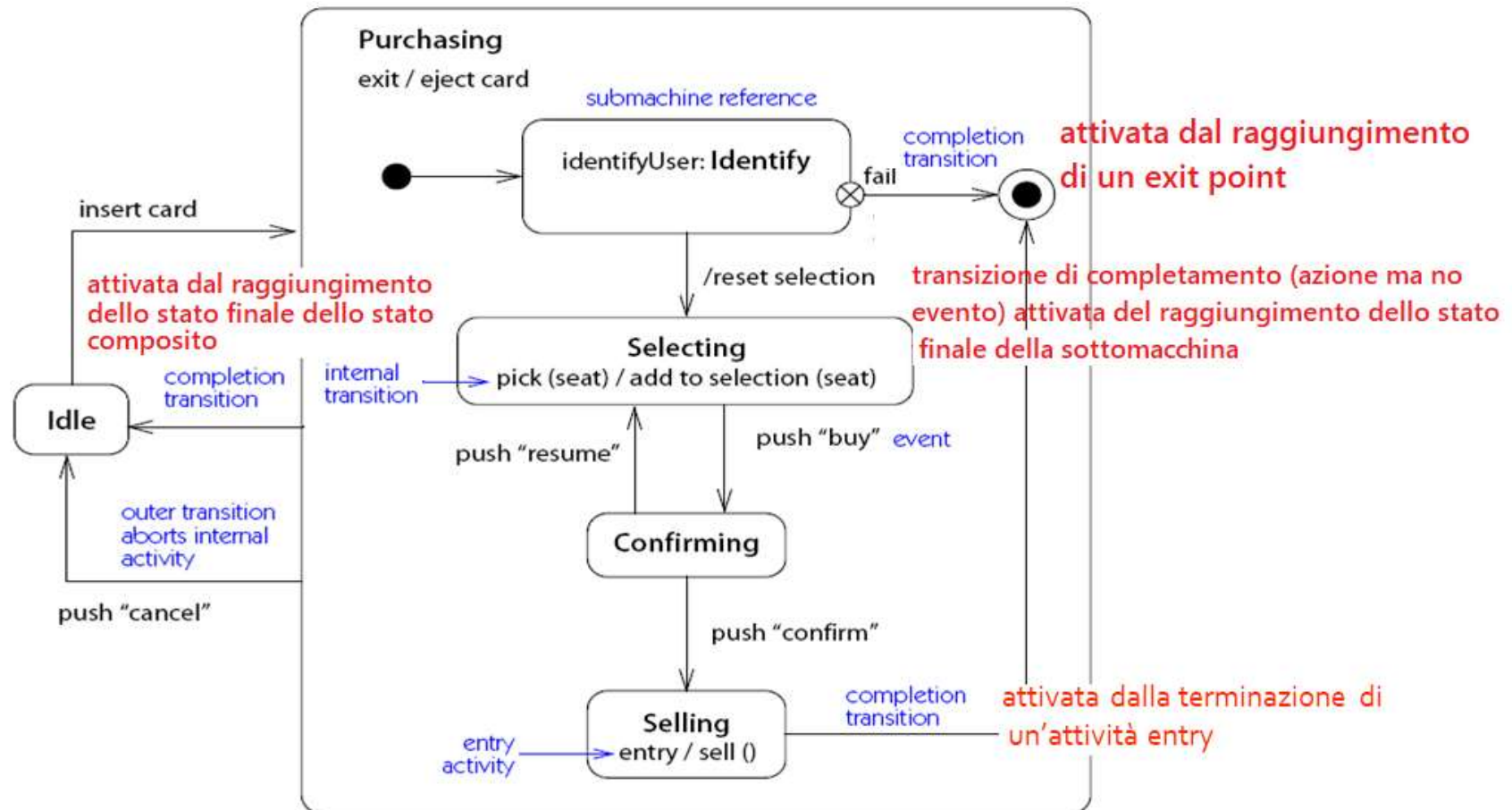


# TRANSIZIONI DI COMPLETAMENTO: QUANDO?

Le transizioni di completamento (senza evento) scattano

- quando si raggiunge la **terminazione** di un'attività composita, aka
  - **stato finale** in un stato composito sequenziale
  - **stati finali** di tutte le regioni ortogonali di un stato composito parallelo
  - un **exit point**
- alla terminazione delle attività **entry** e/o **do**  
(NB: l'attività **exit** viene eseguita quando scatta la transizione di completamento)

# ESEMPIO





# TRANSIZIONI DI COMPLETAMENTO: QUANDO? (CONT.)

Le transizioni di completamento (senza evento) scattano

- quando si raggiunge la **terminazione** di un'attività composta, aka
  - **stato finale** in un stato composito sequenziale
  - **stati finali** di tutte le regioni ortogonali di un stato composito parallelo
  - un **exit point**
- alla terminazione delle attività **entry** e/o **do**  
(NB: l'attività **exit** viene eseguita quando scatta la transizione di completamento)
- quando si raggiunge uno pseudo-stato **giunzione**

pseudo-stato?

giunzione?



# PSEUDO-STATI



**giunzione**



**history**



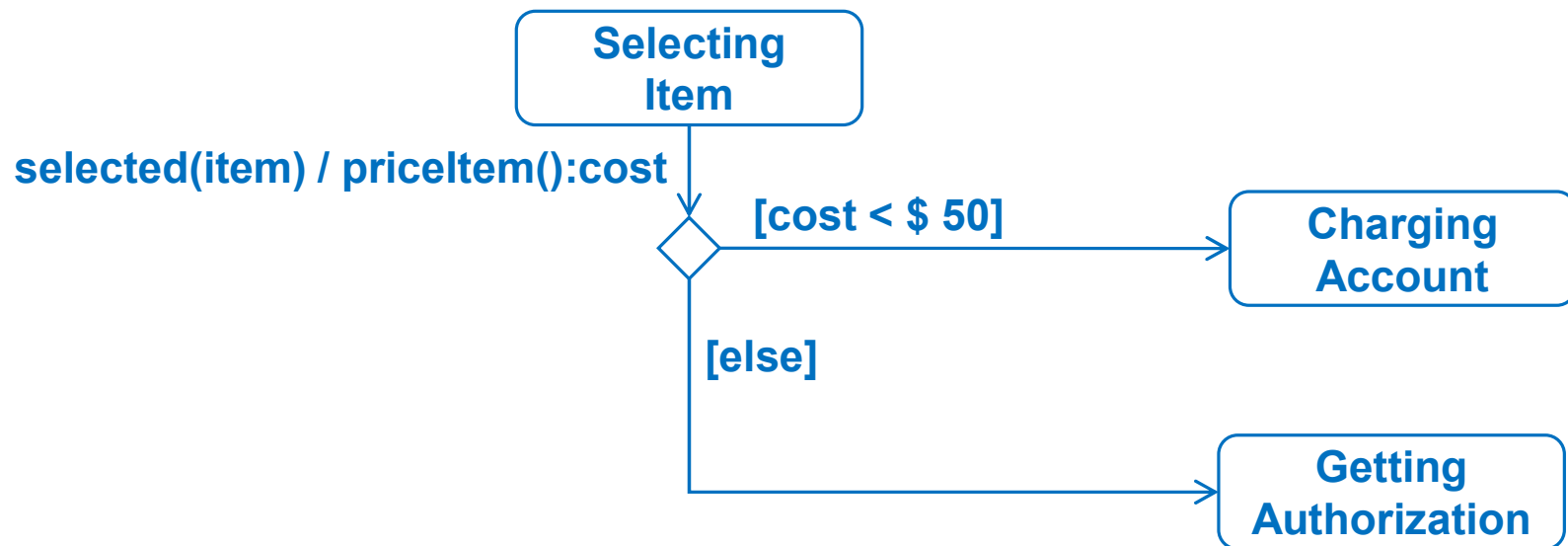
**scelta**

# SCELTA

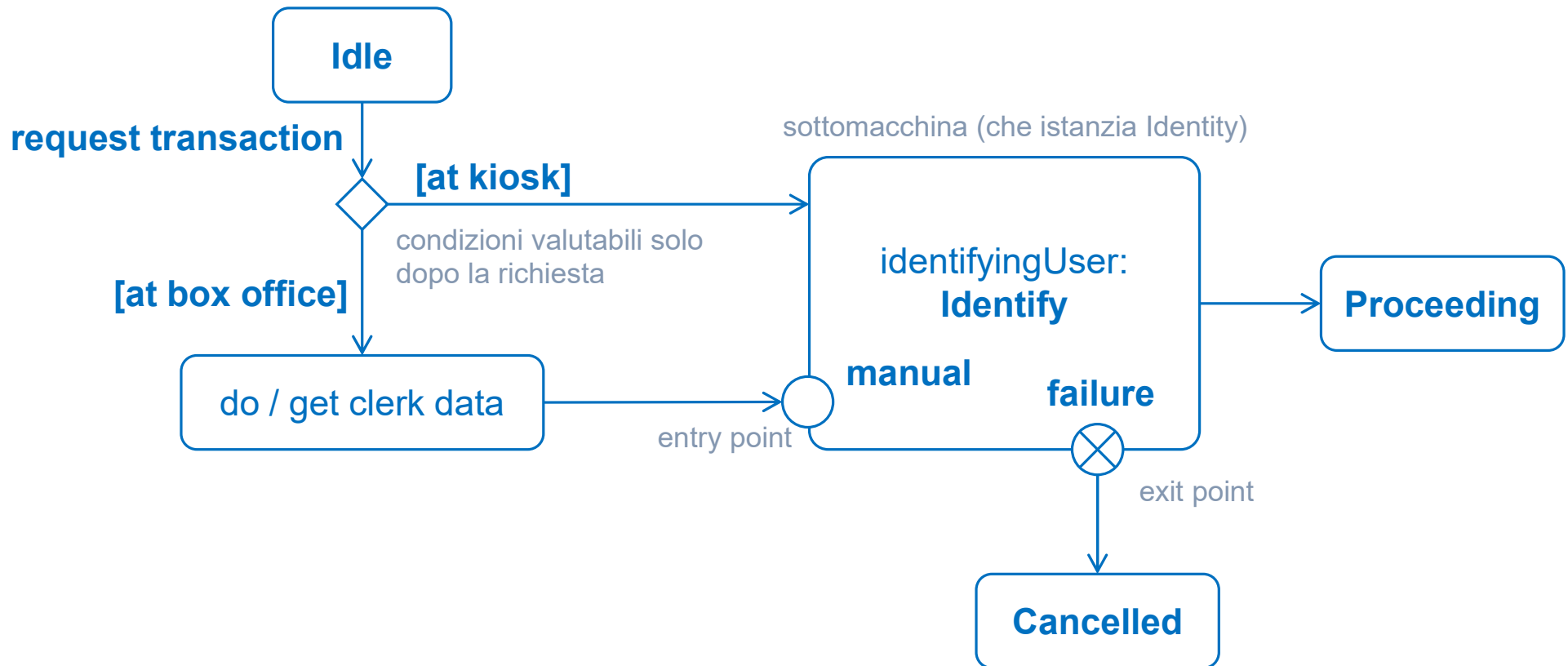
Scelta basata su **condizioni** valutate **dinamicamente**

Come nei diagrammi di attività

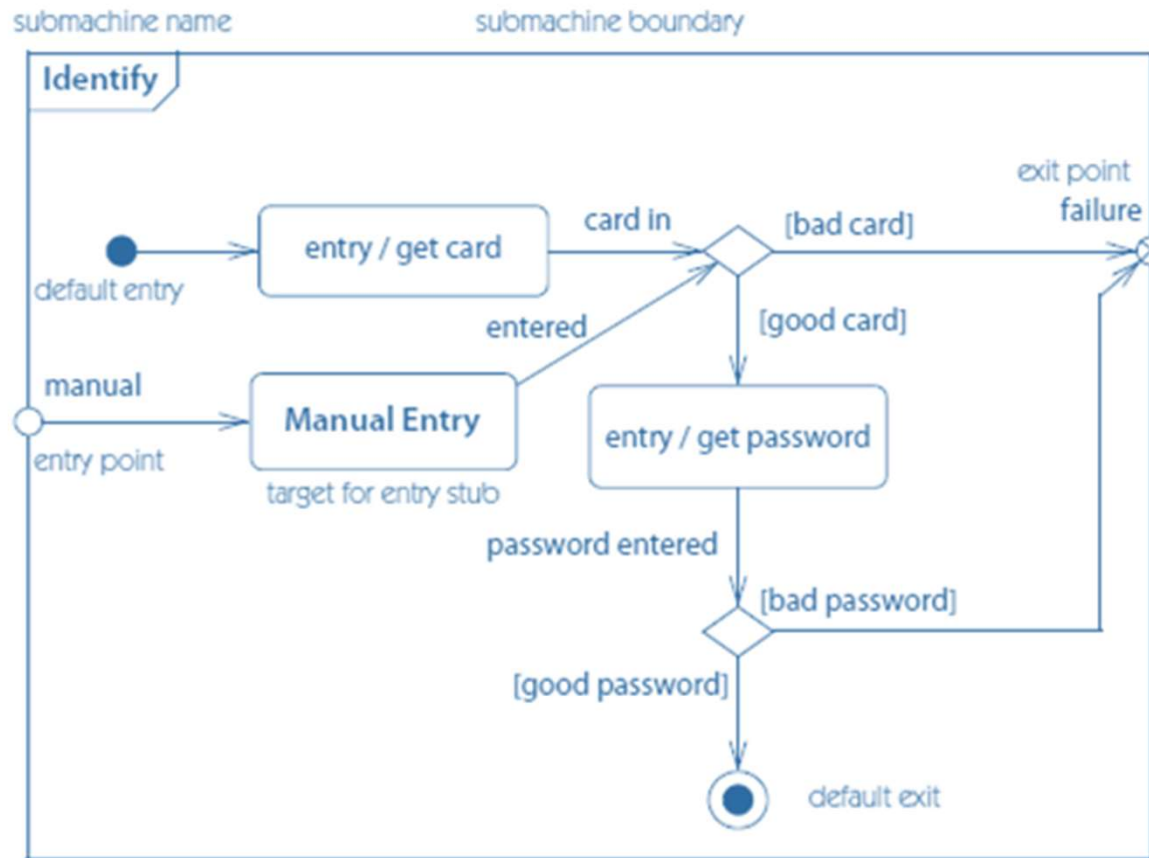
- La **disgiunzione** delle guardie deve essere sempre vera
- È ammesso il **non-determinismo**



# ESEMPIO DI CHOICE, CON SOTTOMACCHINA



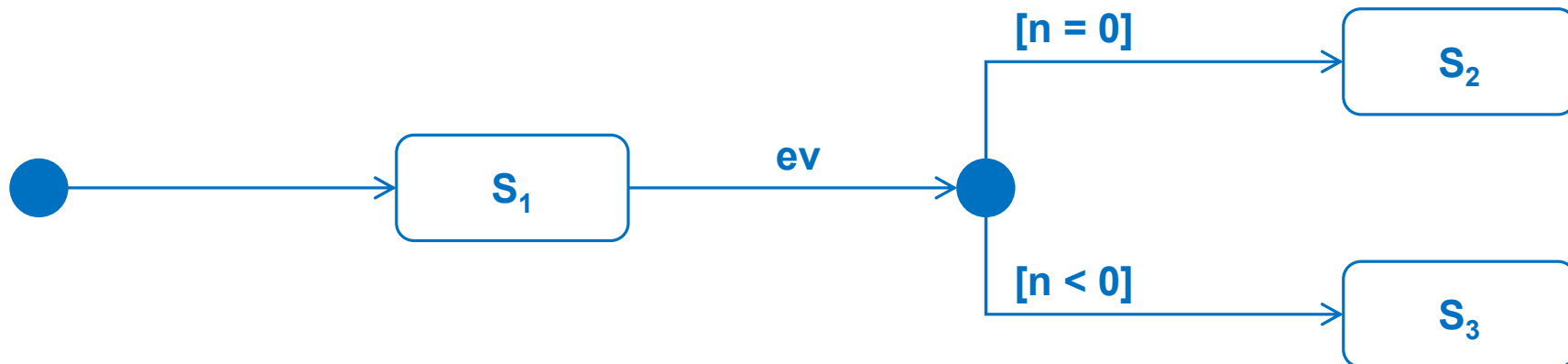
# ESEMPIO DI CHOICE, IN SOTTOMACCHINA



# GIUNZIONE

Pseudo-stato da cui escono e/o entrano due o **più transizioni**

- eventuali **condizioni** sono valutabili **staticamente** e
- **prima degli eventi** che attivano le transizioni di ingresso (nell'esempio, prima dell'evento **ev**)

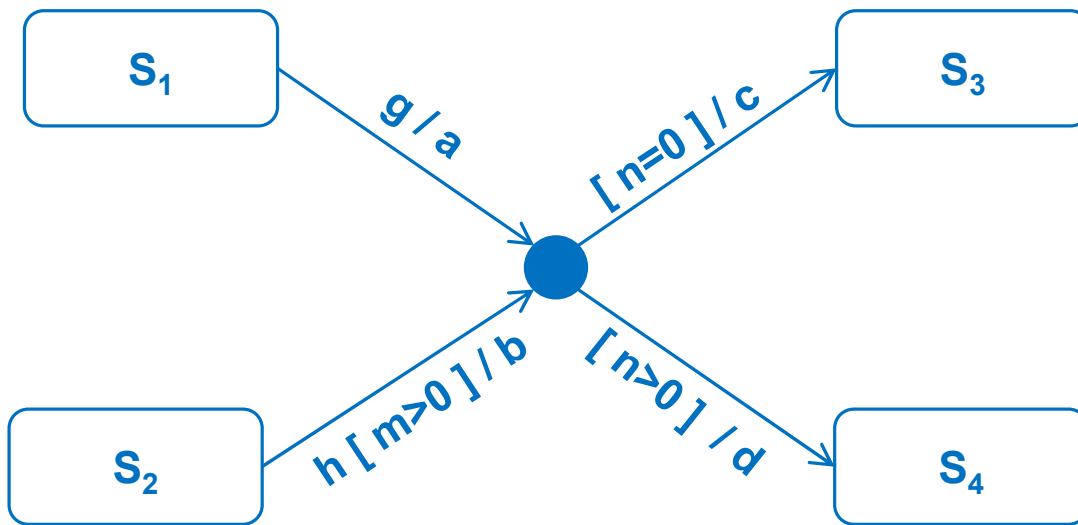


Consentono di **modellare più transizioni** che condividono lo **stesso evento** di attivazione

Nell'esempio:

- **ev [  $n=0$  ]** e **ev [  $n>0$  ]**
- se  **$n<0$** , **e** viene ignorato e si rimane in  $S_1$

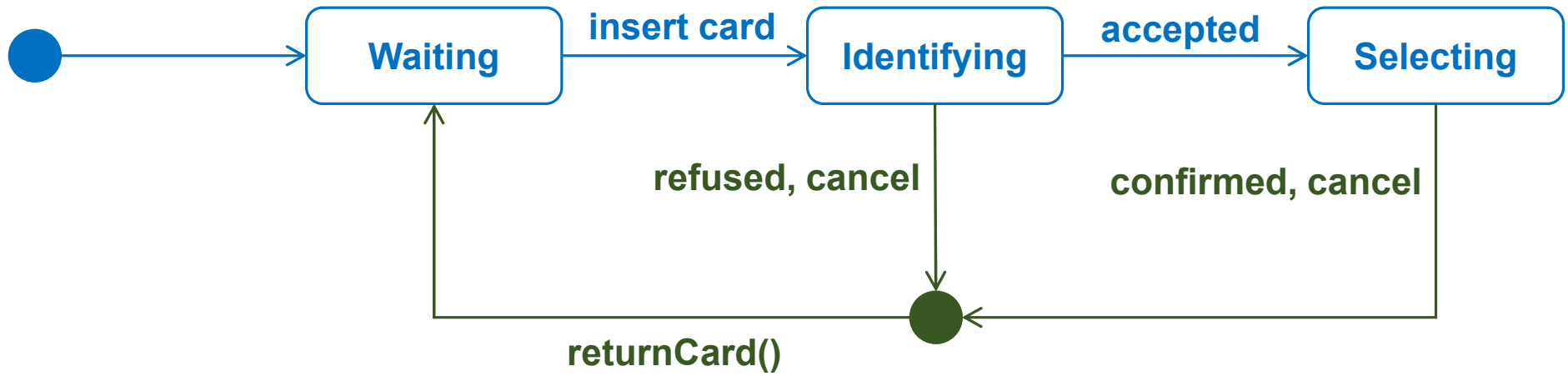
# ESEMPIO, PIÙ COMPLESSO



Transizioni possibili:

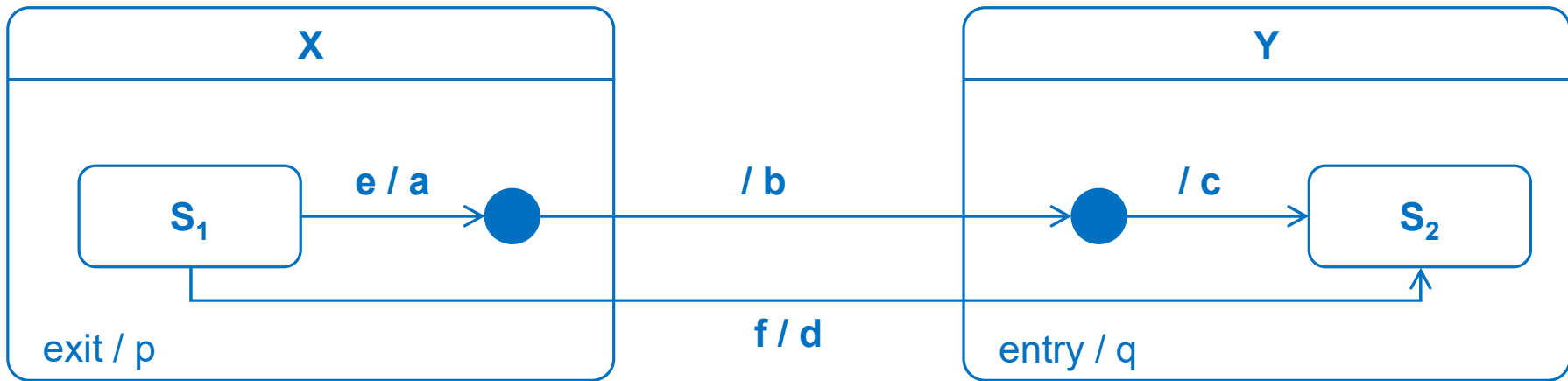
- $S_1 - g [n=0] / a; c \rightarrow S_3$
- $S_1 - g [n>0] / a; d \rightarrow S_4$
- $S_2 - h [m>0 \text{ and } n=0] / b; c \rightarrow S_4$
- $S_2 - h [m>0 \text{ and } n>0] / b; d \rightarrow S_4$

## E L'ESEMPIO DI PRIMA?





# ESEMPIO, CON SOTTOMACCHINE



Transizioni possibili:

- $S_1 - e / a; p; b; q; c \rightarrow S_2$
- $S_1 - f / p; d; q \rightarrow S_2$

# GIUNZIONE O SCELTA?

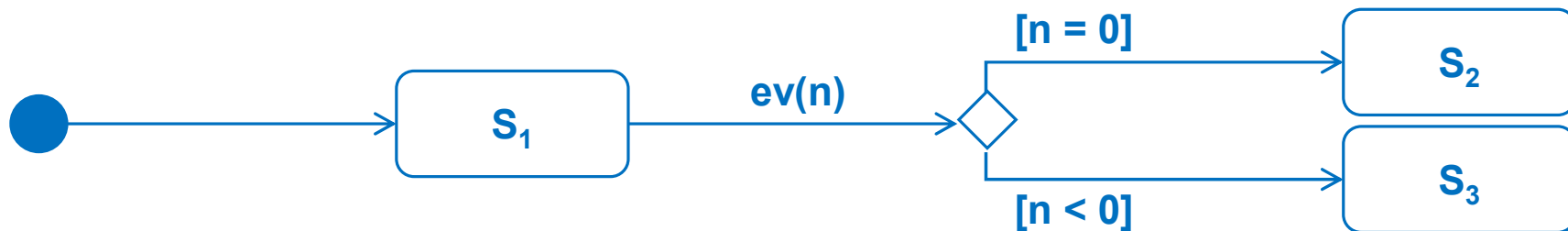
Giunzione:

- statica → guardie **valutate prima di uscire** dallo stato di partenza //  $S_1$
- nell'esempio, se  $n < 0$ , **ev** viene ignorato e non si prende nessuna transizione

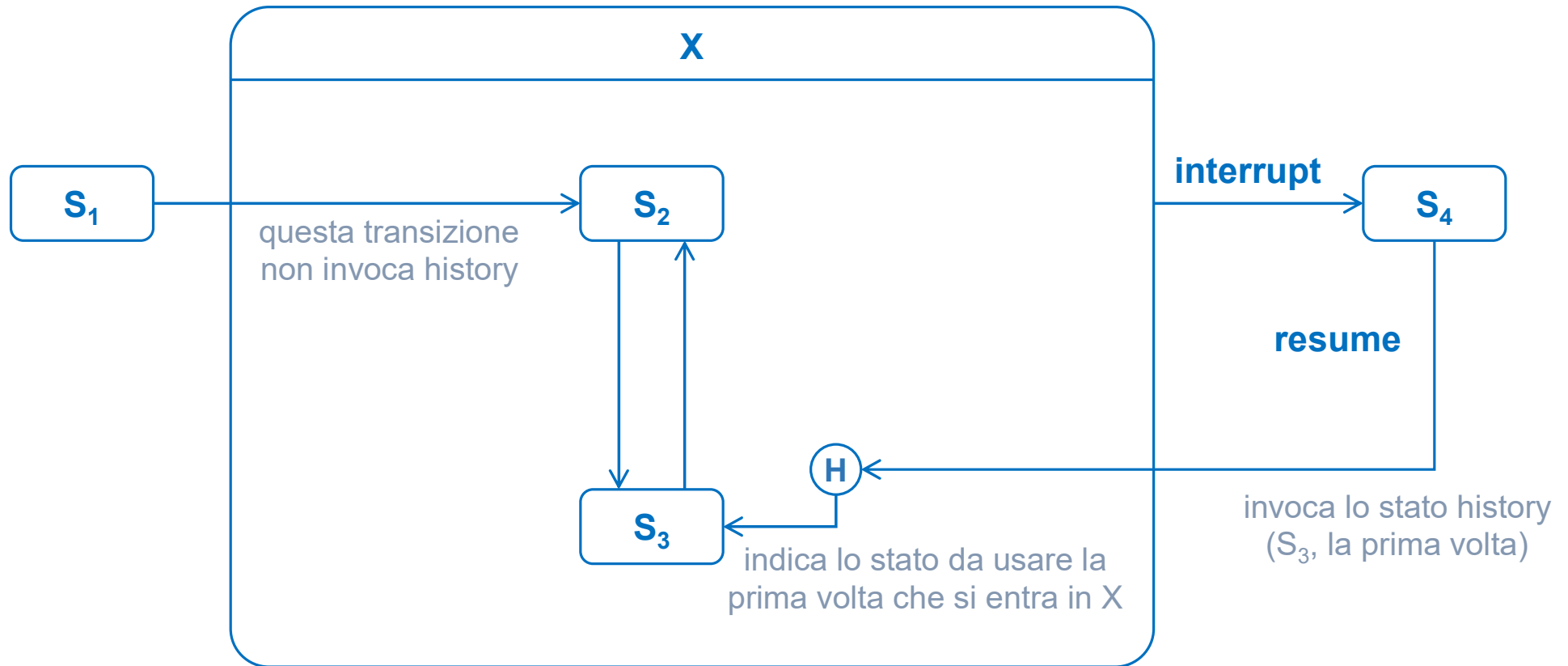


Scelta:

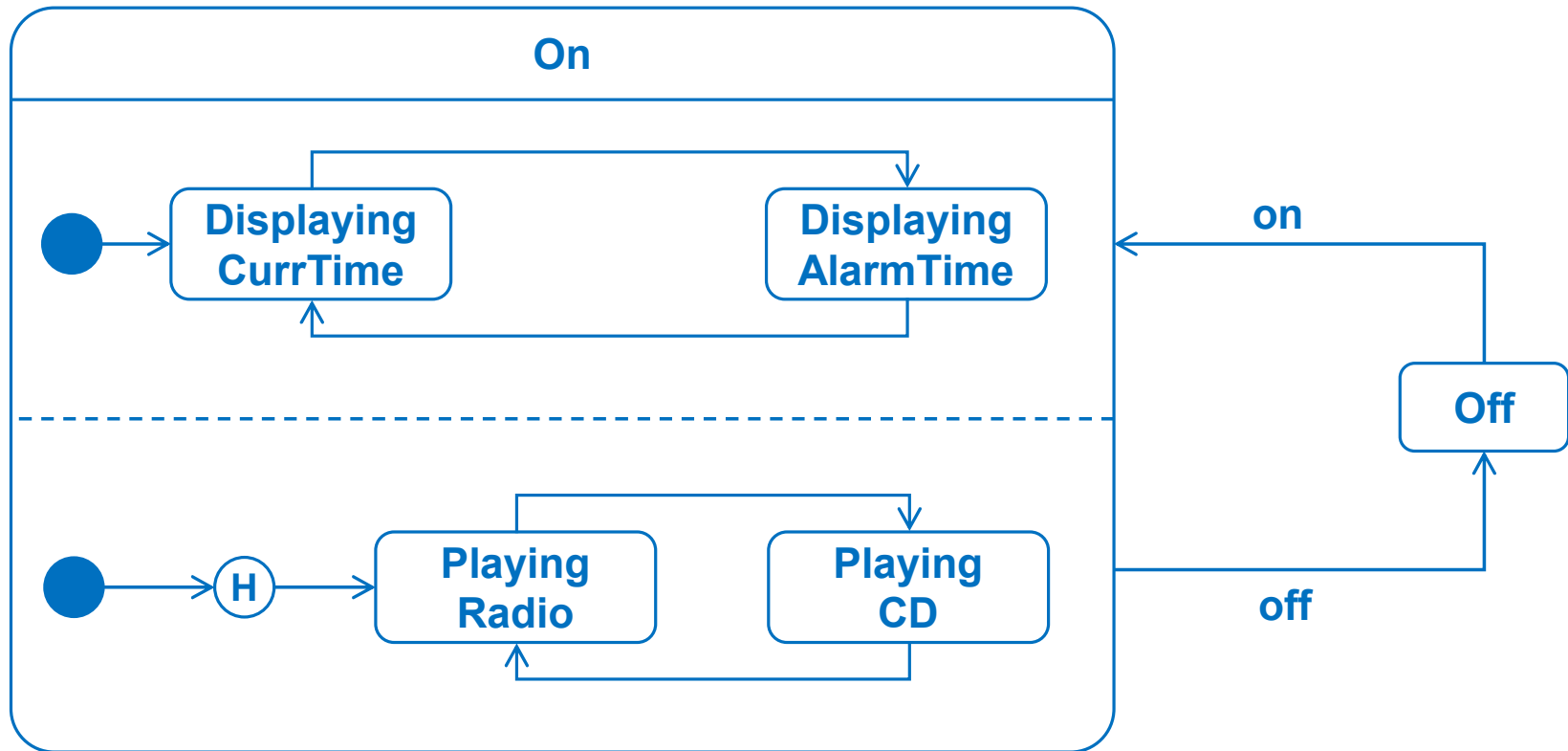
- dinamica → guardie **valutate quando occorre l'evento** // **ev(n)**
- nell'esempio, si assume che valga sempre  $n \geq 0$  (altrimenti il diagramma non è valido)



# HISTORY



# ESEMPIO

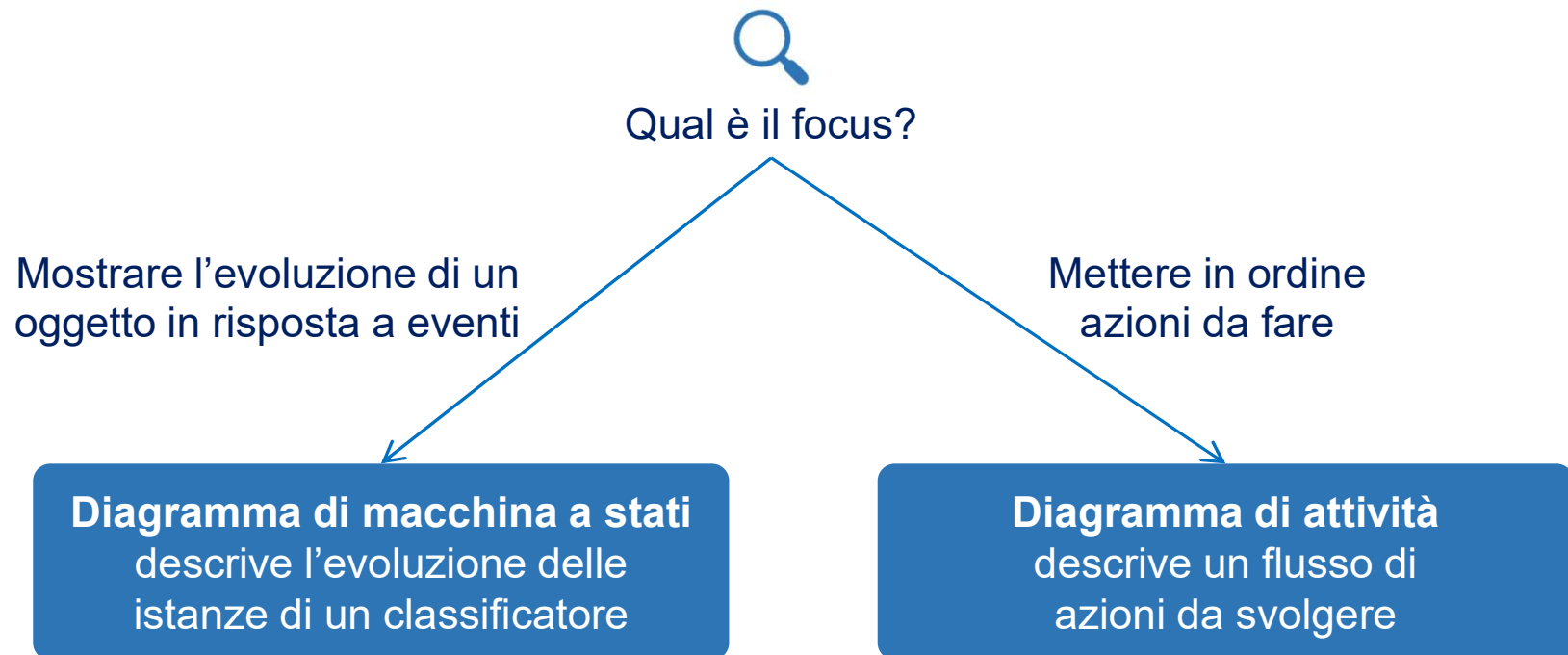


Accensione	Ora	Riproduzione
Prima	Corrente	Radio
Successive	Corrente	Ultima*

\*dipende da H

# FAQ: ATTIVITÀ O STATI?

Come scegliere il diagramma più appropriato per descrivere il modello dinamico?



# FAQ: QUALI SONO I NOMI APPROPRIATI?

I nomi degli stati dovrebbero indicare la **permanenza** in tali stati (per un certo lasso di tempo)

- aggettivi // active, idle
- participi passati // opened, closed, pinned
- gerundi // dialing, connecting

I nomi delle azioni dovrebbero indicare l'**esecuzione** di tale azione

- verbi all'indicativo, imperativo o infinito // create, delete
- sostantivi che indicano un'azione // DBConnection

Si tratta di una **convenzione** a volte disattesa (eccezioni anche negli esempi visti), ma

- aiuta a costruire diagrammi corretti e
- a non confondere stati e azioni nei compiti (errore comune)





# HOMEWORK

## Caso di riferimento: La Piscina

Descrivere con un diagramma di macchina a stati gli stati di un utente che vuole fare nuoto libero:

- partendo dallo stato in cui non è prenotato
- fino a quando è uscito

Trattare anche i casi di errore (ad esempio, l'arrivo fuori orario)



# ESERCIZI CONSIGLIATI

Fornire diagrammi di macchina a stati che modellino

- l'evoluzione della classe Utente (o Situazione Utente) di Myair
- l'evoluzione del semaforo (ex. Semafori)

# RIFERIMENTI

## Contenuti

- **Capitolo 5** di "UML@Classroom" (M. Seidl et al., 2015)  
*// escludendo *deep history**

## Approfondimenti

- **Sezione 13.4** di "Software Engineering" (G. C. Kung, 2023)
- **UML Reference Manual** (vedi appendice)

## APPENDICE (DA UML REFERENCE MANUAL)

Event Type	Description	Syntax
<b>call event</b>	Receipt of an explicit synchronous call request by an object	op(a:T)
<b>change event</b>	A change in value of a Boolean expression	<b>when</b> (exp)
<b>signal event</b>	Receipt of an explicit, named, asynchronous communication among objects	sname(a:T)
<b>time event</b>	The arrival of an absolute time or the passage of a relative amount of time	<b>after</b> (time)

## APPENDICE (DA UML REFERENCE MANUAL)

Transition Kind	Description	Syntax
<b>entry transition</b>	The specification of an entry activity that is executed when a state is entered	<b>entry</b> / activity
<b>exit transition</b>	The specification of an exit activity that is executed when a state is exited	<b>exit</b> / activity
<b>external transition</b>	A response to an event that causes a change of state or a self-transition, together with a specified effect. It may also cause the execution of exit and/or entry activities for states that are exited or entered	e(a:T)[guard]/activity
<b>Internal transition</b>	A response to an event that causes the execution of an effect but does not cause a change of state or execution of exit or entry activities	e(a:T)[guard]/activity

## APPENDICE (DA UML REFERENCE MANUAL)

State Kind	Description	Notation
<b>simple state</b>	A state with no substructure	(see manual)
<b>orthogonal state</b>	A state that is divided into two or more regions. One direct substate from each region is concurrently active when the composite state is active	(see manual)
<b>nonorthogonal state</b>	A composite state that contains one or more direct substates, exactly one of which is active at one time when the composite state is active	(see manual)
<b>initial state</b>	A pseudostate that indicates the starting state when the enclosing state is invoked	(see manual)
<b>final state</b>	A special state whose activations indicates the enclosing state has completed activity	(see manual)
<b>terminate</b>	A special state whose activation terminates execution of the object owning the state machine	(see manual)
<b>junction</b>	A pseudostate that chains transitions segments into a single run-to-completion transition	(see manual)
<b>choice</b>	A pseudostate that performs a dynamic branch within a single run-to-completion transition	(see manual)
<b>history state</b>	A pseudostate whose activation restores the previously active state within a composite state	(see manual)

## APPENDICE (DA UML REFERENCE MANUAL)

State Kind	Description	Notation
<b>submachine state</b>	A state that references a state machine definition, which conceptually replaces the submachine state	(see manual)
<b>entry point</b>	A externally visible pseudostate within a state machine that identifies an internal state as a target	(see manual)
<b>exit point</b>	A externally visible pseudostate within a state machine that identifies an internal state as a source	(see manual)