
UML: Diagramma di Macchina a Stati

Laura Semini, Ingegneria del Software

Dipartimento di Informatica, Università di Pisa

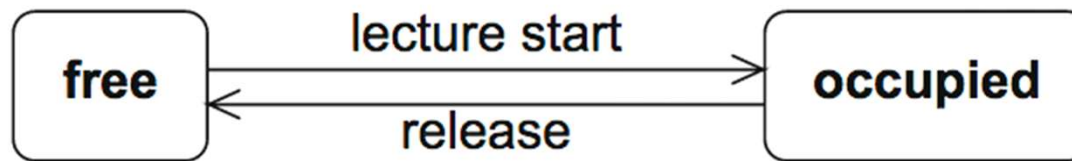


Macchina a Stati

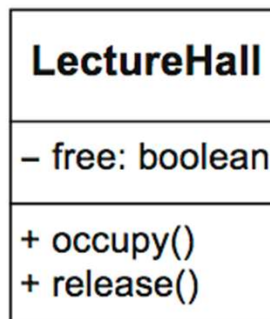
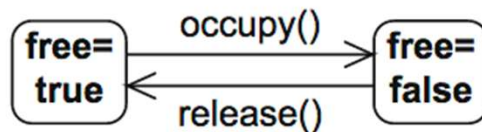
- Una macchina a stati è un grafo stati transizioni che descrive il comportamento delle istanze di una classe
 - in generale di un classificatore
- Occorre individuare gli stati significativi in cui si può trovare un oggetto durante la sua vita
- Inoltre dobbiamo descrivere come da ciascuno di questi stati l'oggetto può passare (transire) in un altro
- Le transizioni avvengono in risposta al verificarsi di un evento.
Esempio di eventi:
 - messaggi inviati da altri oggetti
 - eventi generati internamente

Importante il livello di dettaglio

Per modellare un'aula



verrà poi implementato come



```
class LectureHall {
    private boolean free;
    public void occupy() {
        free=false;
    }
    public void release() {
        free=true;
    }
}
```

Sintassi

Stato:

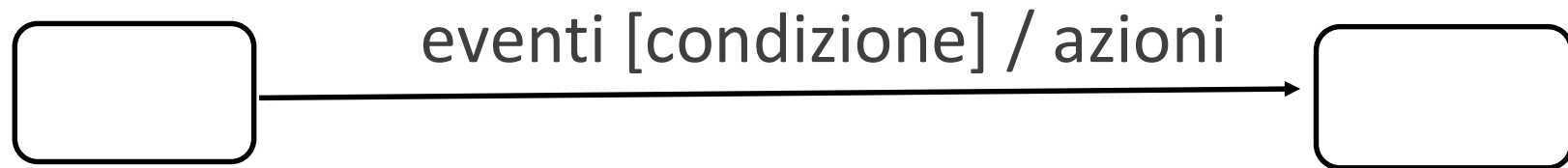


Stato iniziale: ●

Stato finale: ⊙

Transizione

- L'uscita da uno stato
 - definisce la 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



eventi ::= evento | evento , eventi (disgiunzione)

azioni ::= azione | azione , azioni (sequenza)

Tutti opzionali anche tranne l'evento

Solo nelle transizioni di completamento (più avanti) l'evento non serve.

Esempio: lampadina

Descriviamo la vita di una lampadina

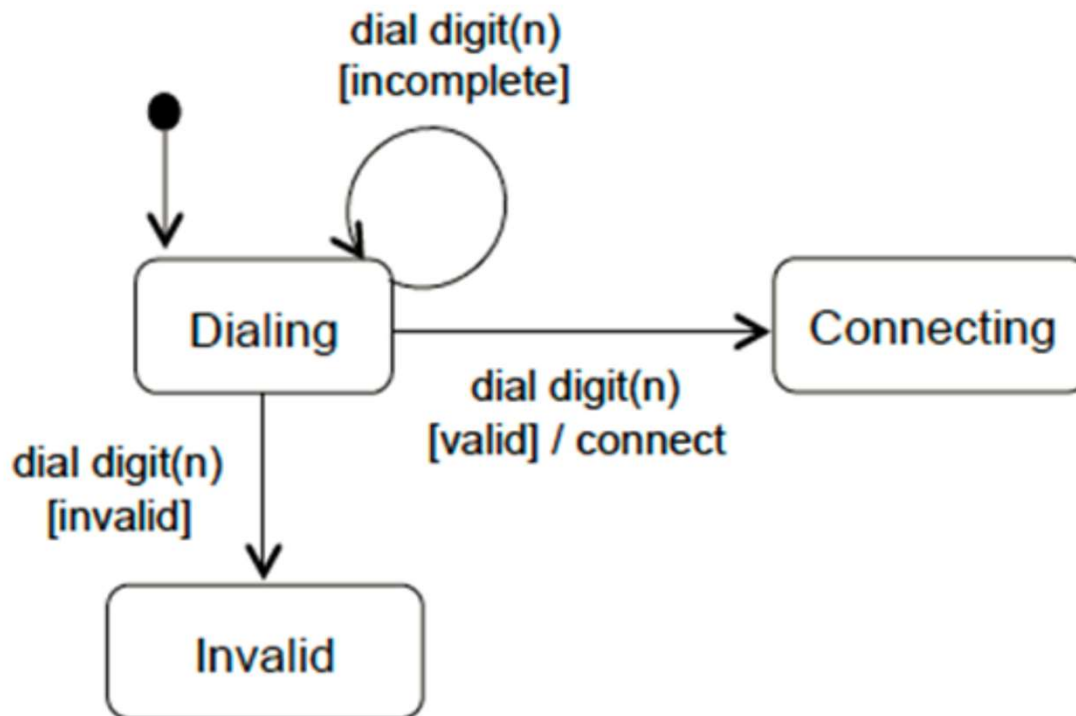


Lampadina
accesa : boolean = false
accendi() spegni()

OSSERVAZIONE: gli eventi (anche se non tutti) corrispondono alle operazioni della classe

Evento operazione

- Operazioni della classe telefono (dial digit(n))

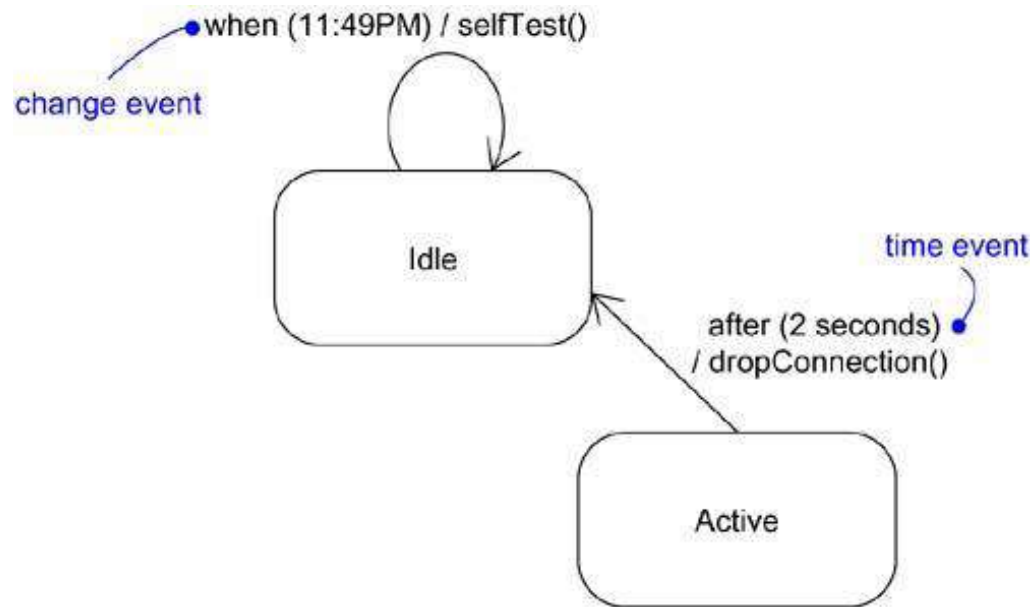


Evento

- Un evento occorre istantaneamente
- Gli eventi che occorrono quando l'oggetto è in uno stato per cui non è prevista alcuna transizione etichettata con quell'evento vengono ignorati
- È ammesso il non-determinismo: un evento può fare da trigger a più transizioni che escono dallo stesso stato :
 - Ne viene scelta una non-deterministicamente
- Livello di astrazione/dettaglio: Introducete un evento se ha degli effetti

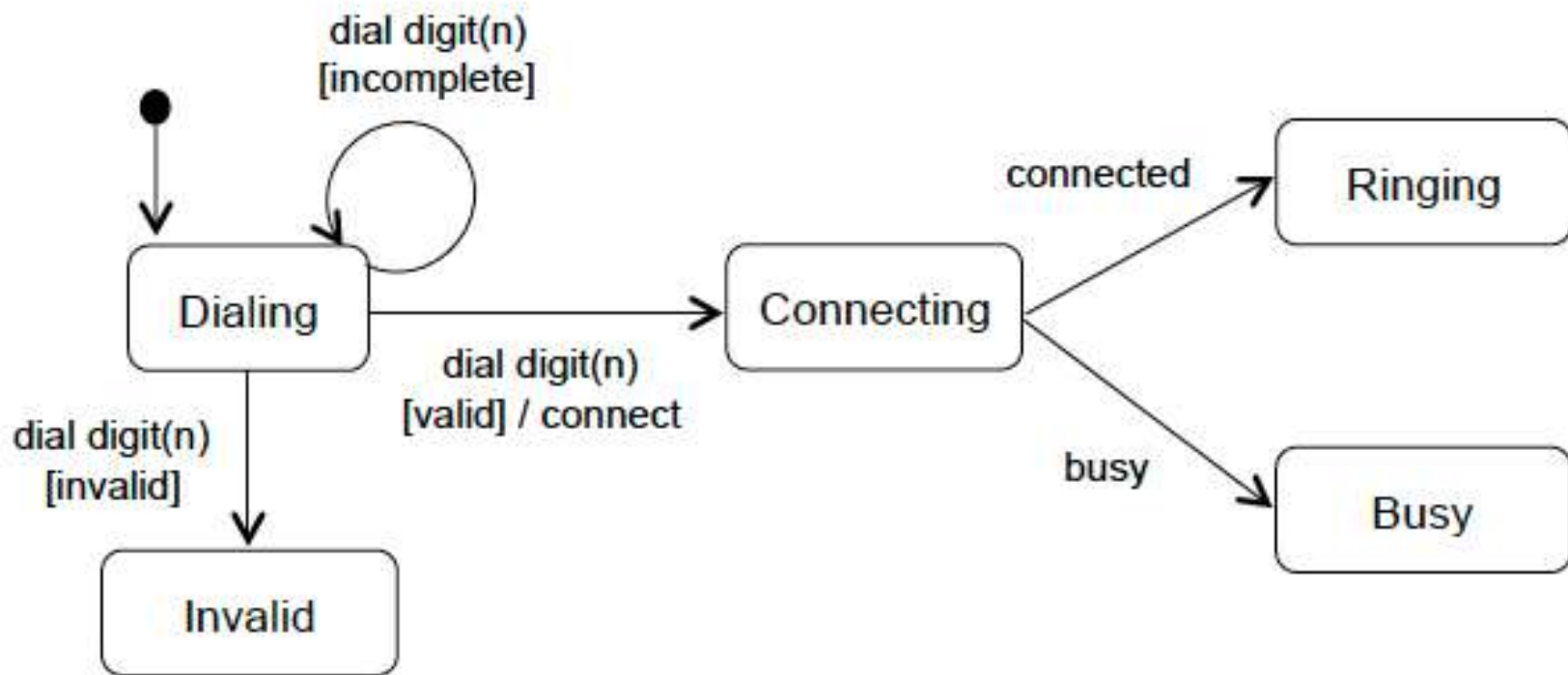
Tipi di evento

- Operazione o segnale **op(a:T)**
 - la transizione è abilitata quando l'oggetto (in quello stato) riceve una chiamata di metodo / un segnale con parametri (a) e tipo (T) (i parametri sono opzionali)
- Evento di variazione **when(exp)**
 - la transizione è abilitata appena l'espressione diventa vera
 - l'espressione può indicare un tempo assoluto o una condizione
- Evento temporale **after(time)**
 - la transizione è abilitata dopo che l'oggetto è stato fermo "time" in quello stato

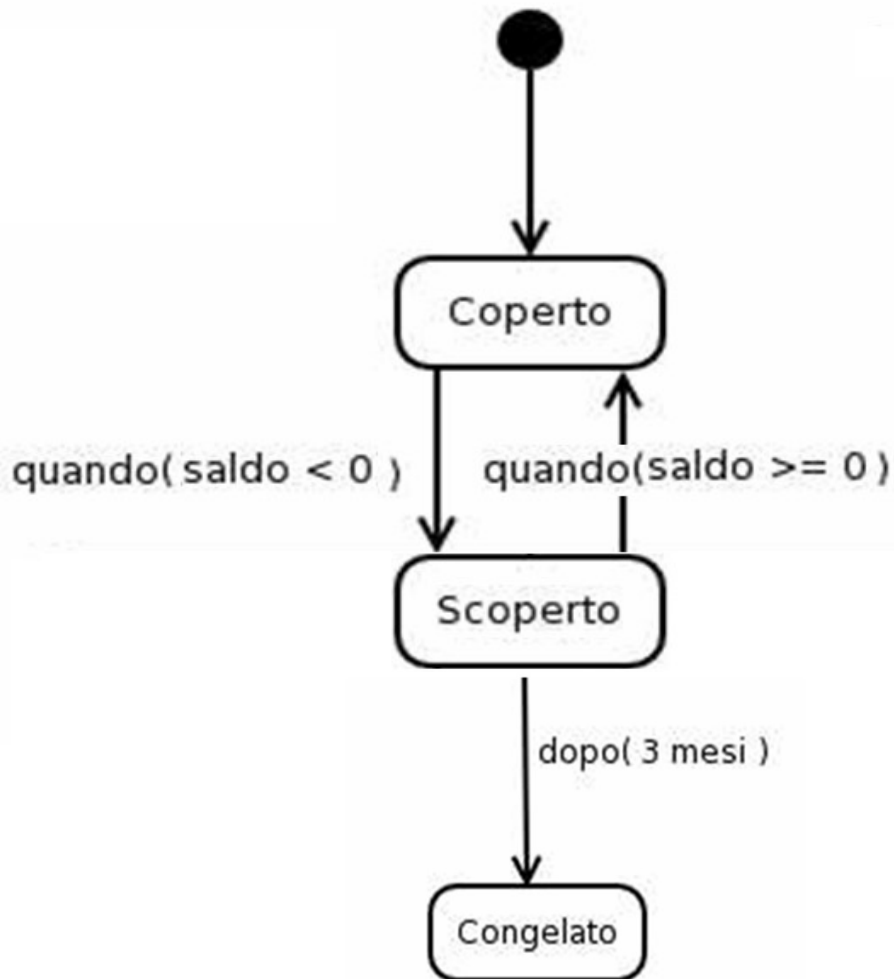


Segnali

segnali (busy e connected)



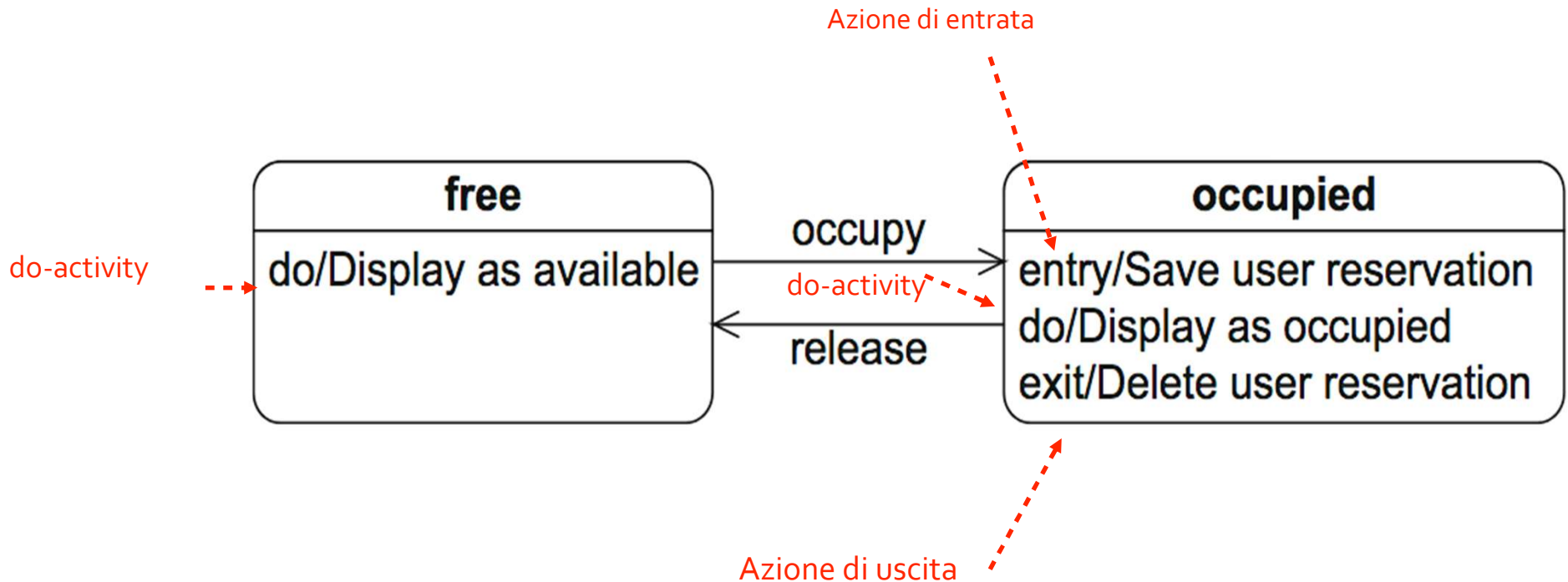
Eventi di variazione ed eventi temporali



- **Eventi di variazione, a cosa servono?**
 - Un evento occorre in modo istantaneo
 - una condizione non è istantanea
 - è istantaneo il momento in cui diventa vera
- **Evento temporale**
 - Dopo che l'oggetto è stato 3 mesi fermo nello stato Scoperto, transisce nello stato Congelato

Entry, exit, transizioni e attività interne

- Ritorniamo sull'esempio dell'aula:

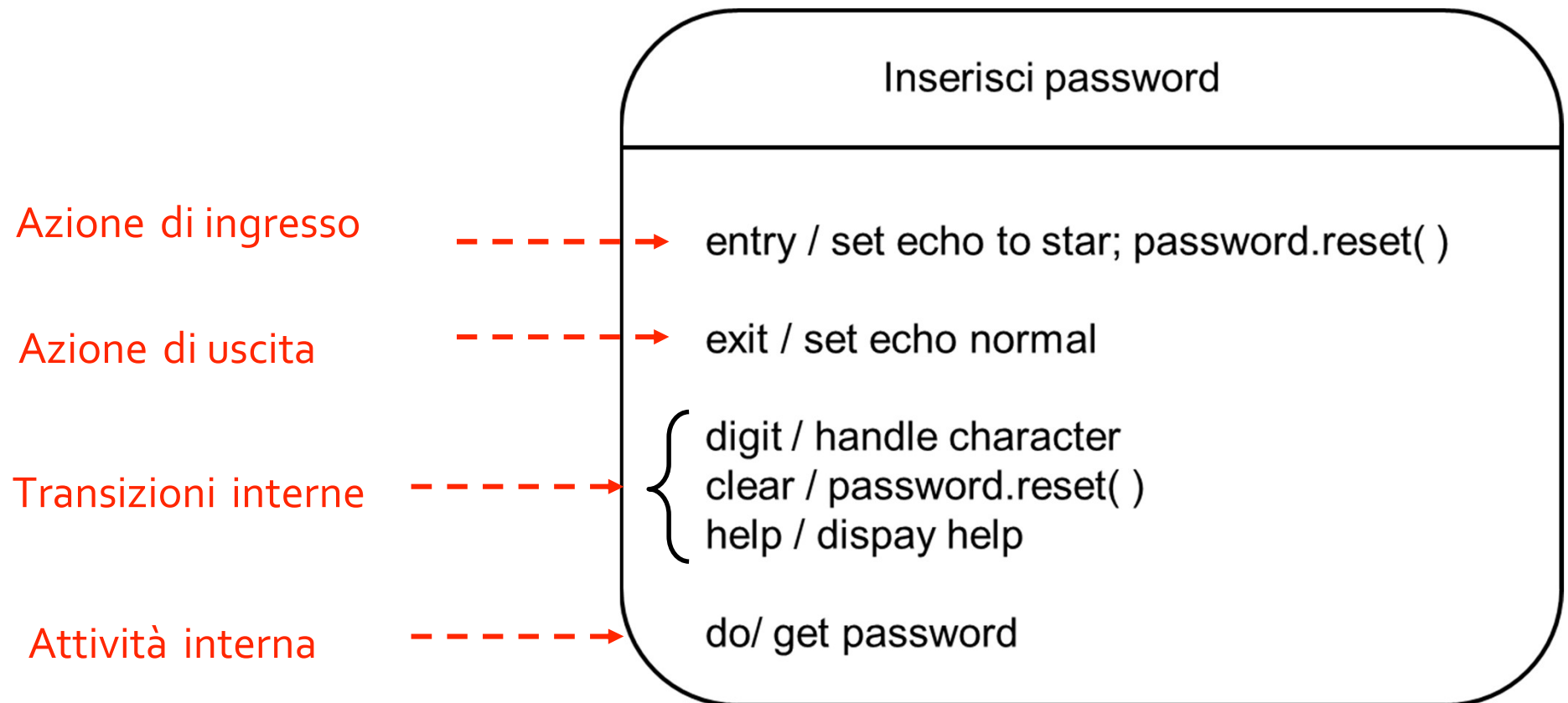


Entry, exit, transizioni e attività interne

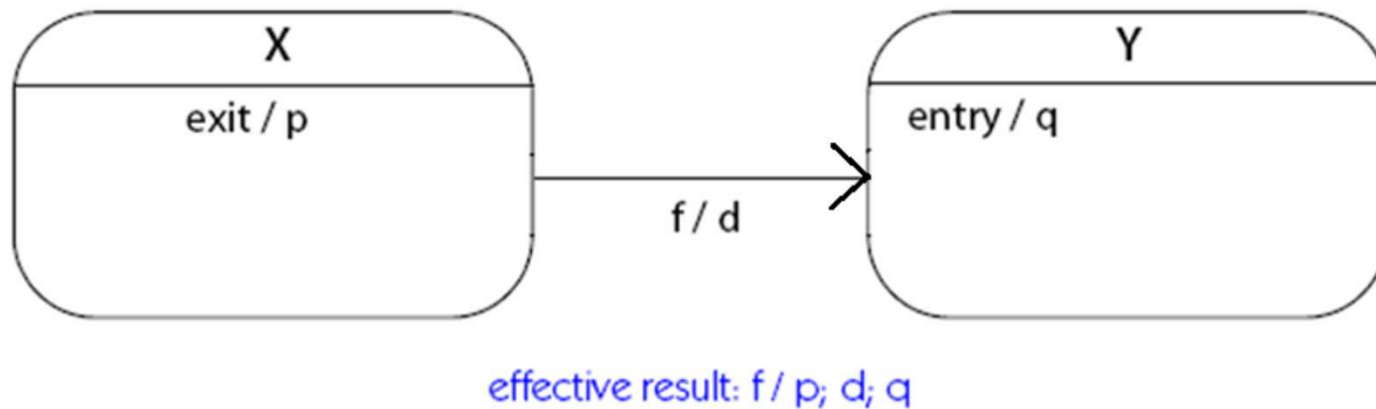
- **Azione di entrata:** eseguita all'ingresso in uno stato
- **Azione di uscita:** eseguita all'uscita di uno stato
- **Transizione interna:** risposta ad un evento

- **Attività interna (Do-activity):**
 - eseguita in modo continuato mentre l'oggetto si trova in quello stato
 - senza necessità di un evento scatenante
 - al contrario di tutte le altre azioni che sono atomiche:
 - consuma del tempo
 - può essere interrotta (quando un evento fa uscire dallo stato)

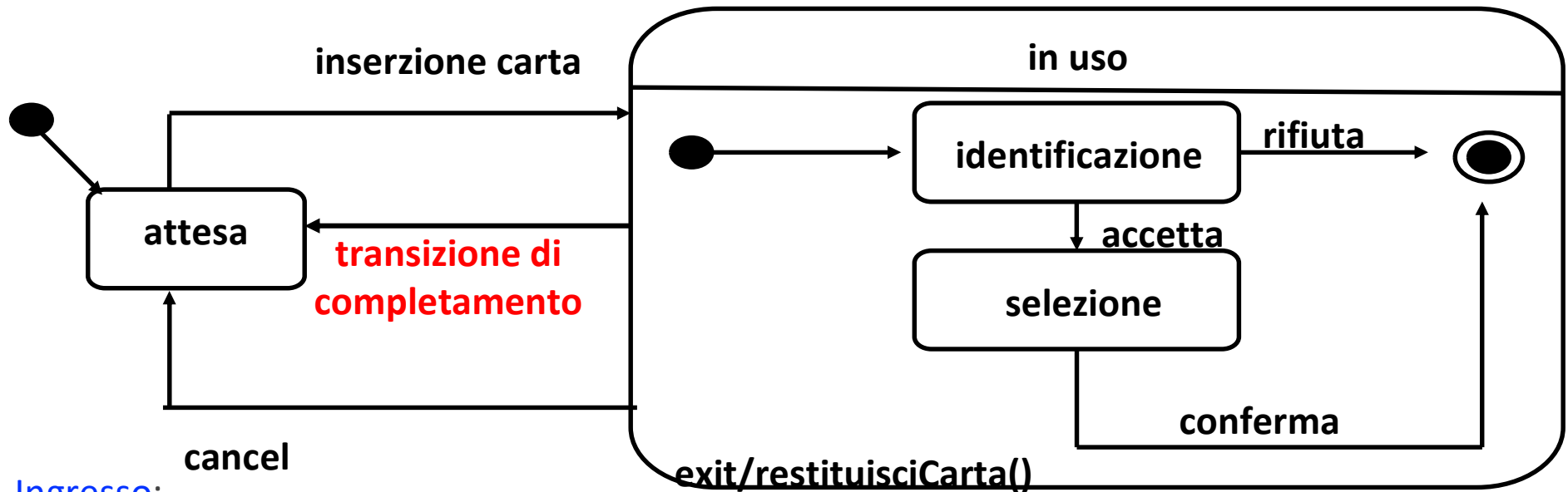
Entry, exit, transizioni e attività interne



Entry, exit, transizioni e attività interne: esempi



(Esempio di) stato composito (sequenziale)



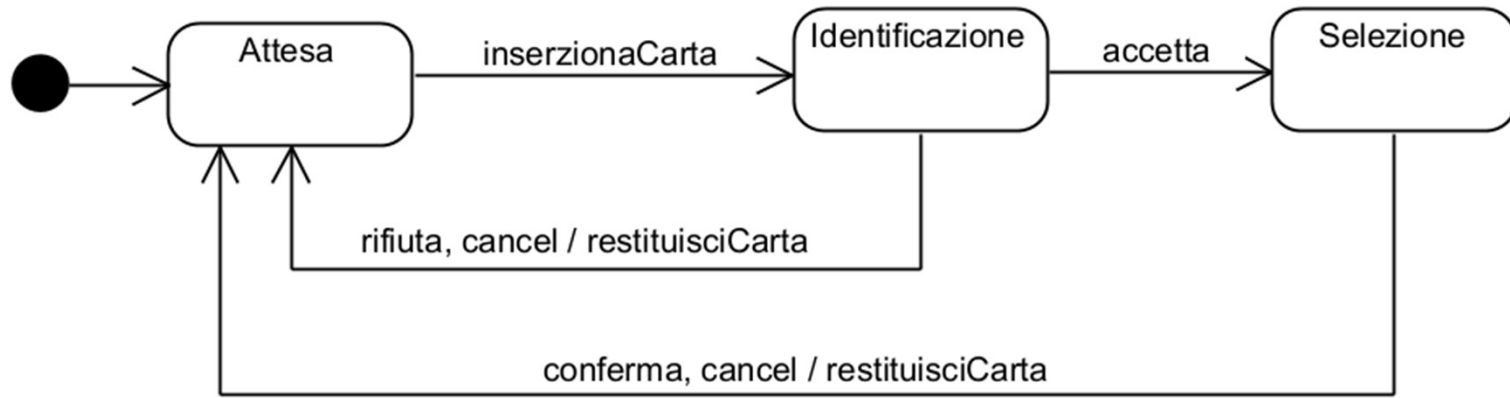
- **Ingresso:**

- Una transizione (inserzione carta) che arriva sul bordo prosegue nello stato iniziale dello stato composito
- Una transizione di ingresso può anche avere come target uno stato interno (non nell'esempio)

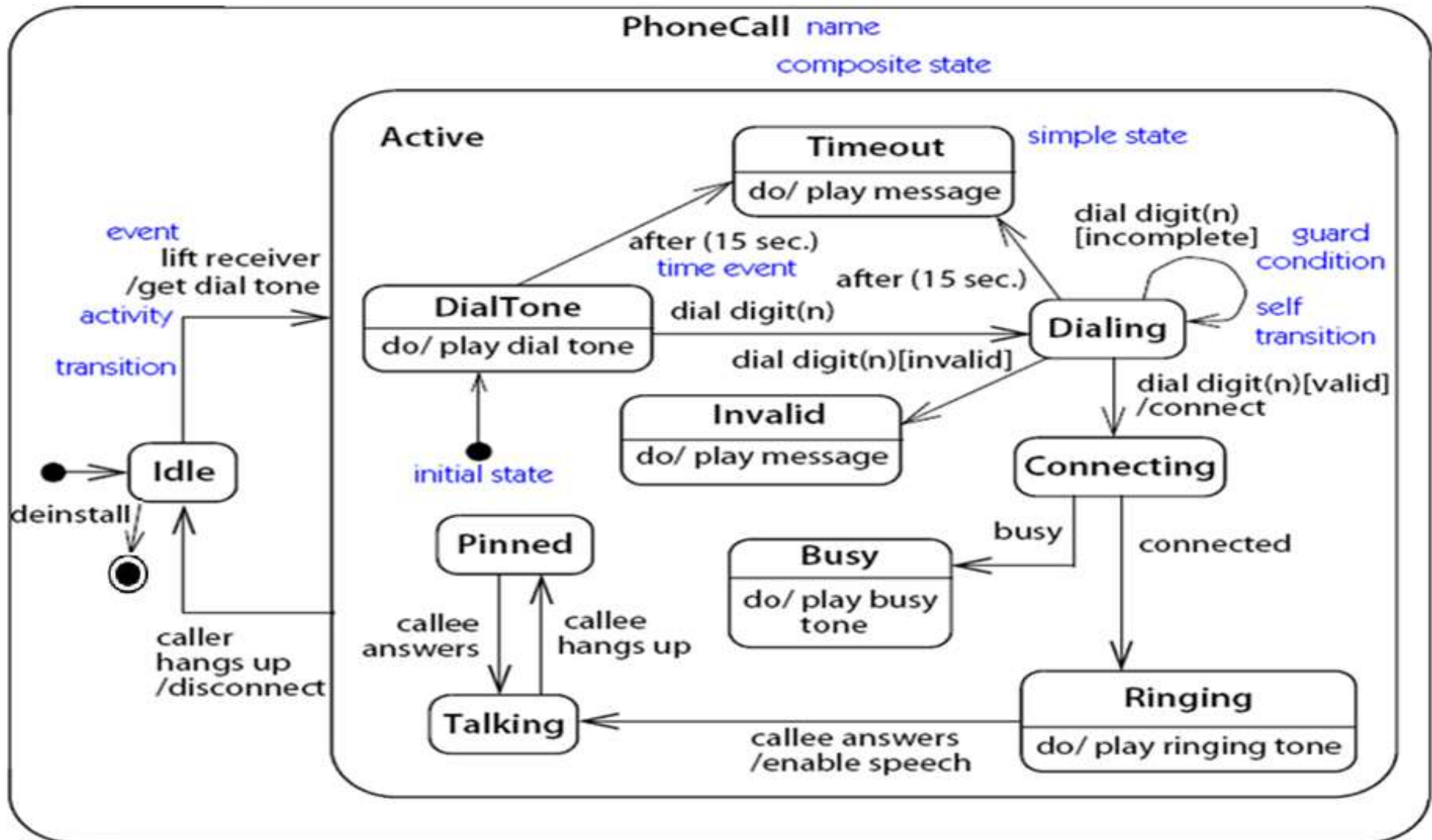
- **Uscita**

- Una transizione (cancel) che parte dal bordo si intende possibile **da un qualsiasi stato interno**
- Dallo stato finale (dopo le exit) si prosegue nella **transizione di completamento** (non ha evento)
- Possono esserci transizioni che "bucano" il bordo e si intendono possibili **dal solo stato interno** a cui sono collegate (non nell'esempio)

Senza stato composito

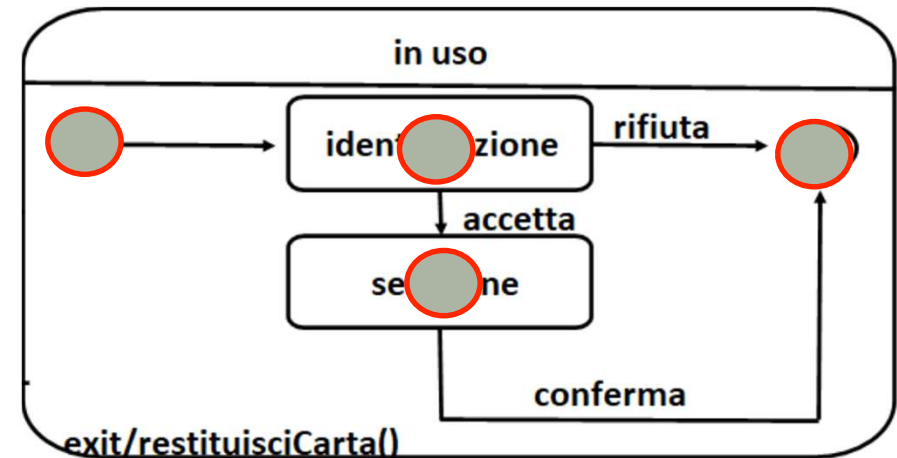


Esempio: stato composito sequenziale (senza stato finale)

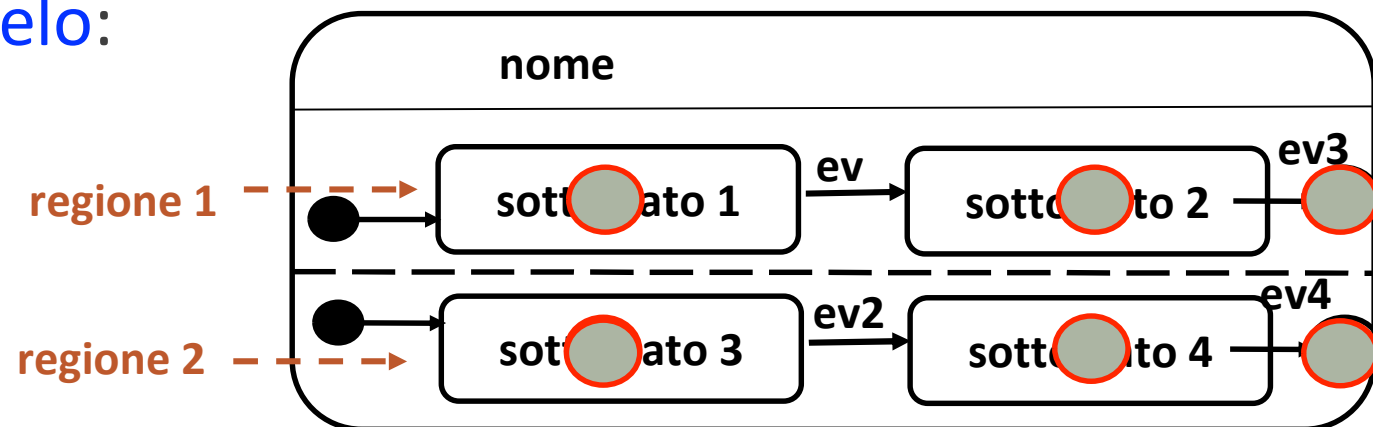


Stati compositi: sequenziali e paralleli

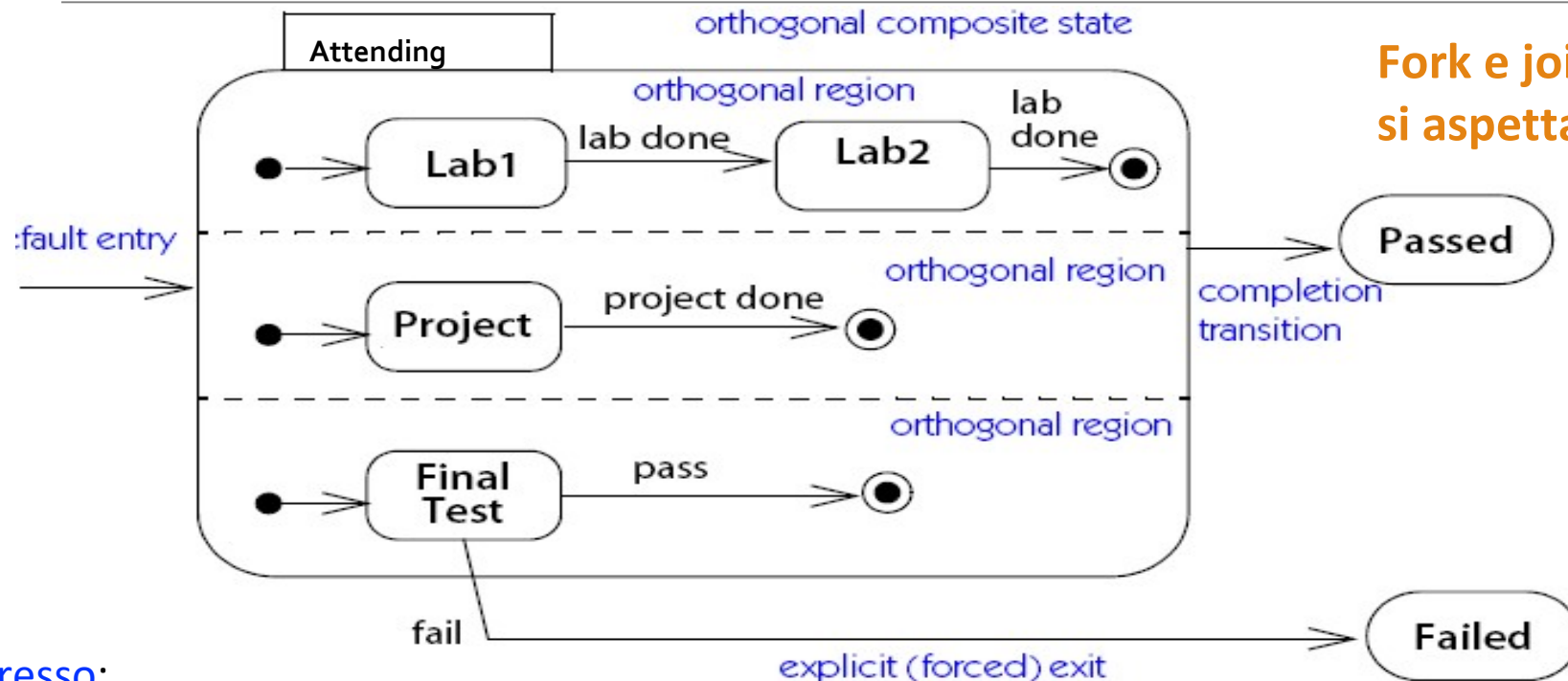
- Composito **sequenziale**:
- Un solo sottostato attivo in ogni istante



- Composito **parallelo**:
- sottostati attivi contemporaneamente
- uno per regione



(Esempio di) Stato composito parallelo



Fork e join implicite:
si aspetta per uscire.

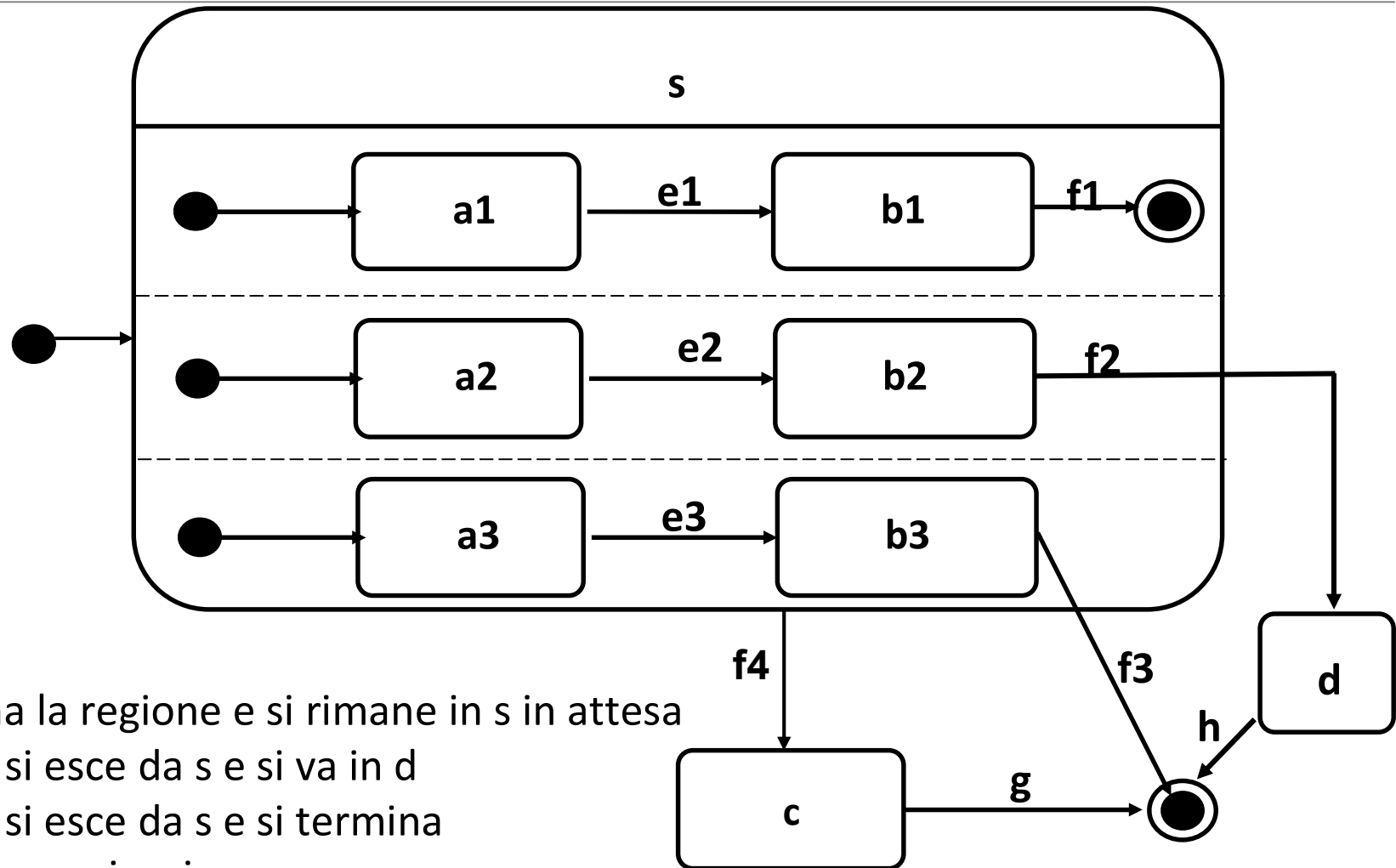
- Ingresso:

- Una transizione (default entry) che arriva sul bordo prosegue **in tutti gli stati iniziali**

- Uscita

- Una transizione che "buca" il bordo (fail) si intende possibile **solo** se l'evento avviene quando la macchina si trova in **nello stato interno** a cui è collegata: **fa uscire da tutti i sottostati**
- Una volta raggiunti **tutti gli stati finali** si prosegue nella transizione di completamento
- Una transizione che parte dal bordo si intende possibile **da un qualsiasi stato interno**, **fa uscire da tutti i sottostati** (non nell'esempio)

Stato composito parallelo: esempio



f1: si termina la regione e si rimane in s in attesa

f2: se in b2, si esce da s e si va in d

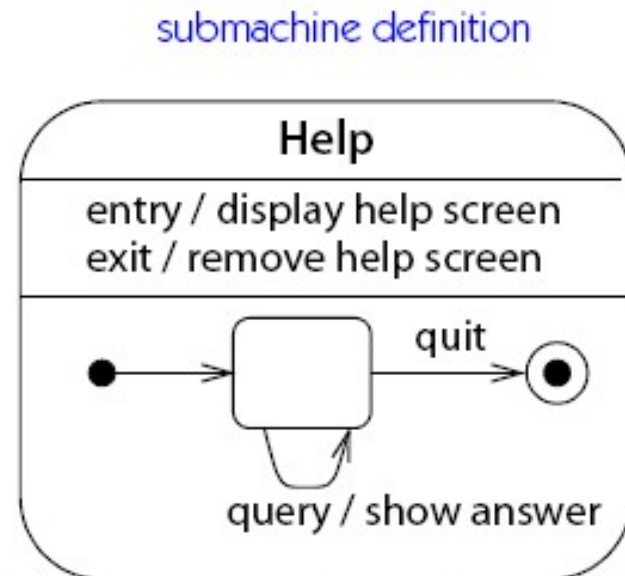
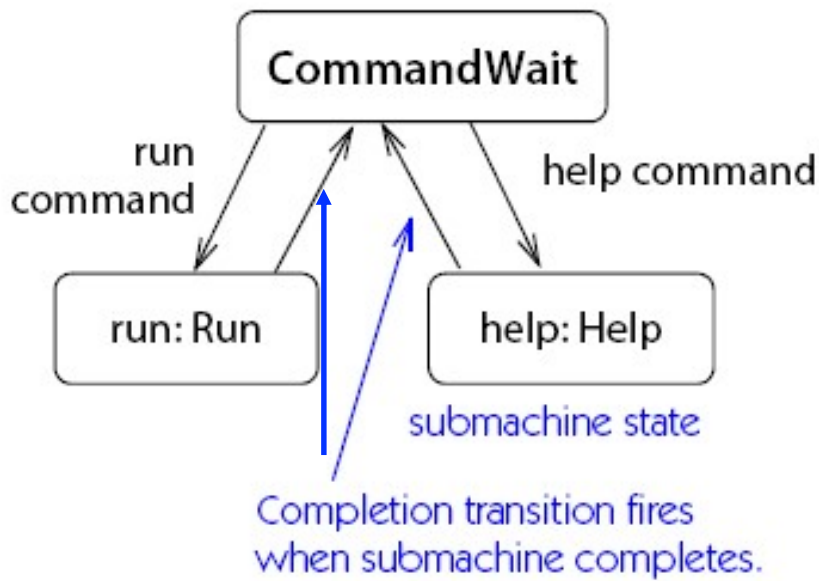
f3: se in b3, si esce da s e si termina

f4: si esce da s e si va in c

Uscendo da s si esce anche da tutti i sottostati

Sottomacchine

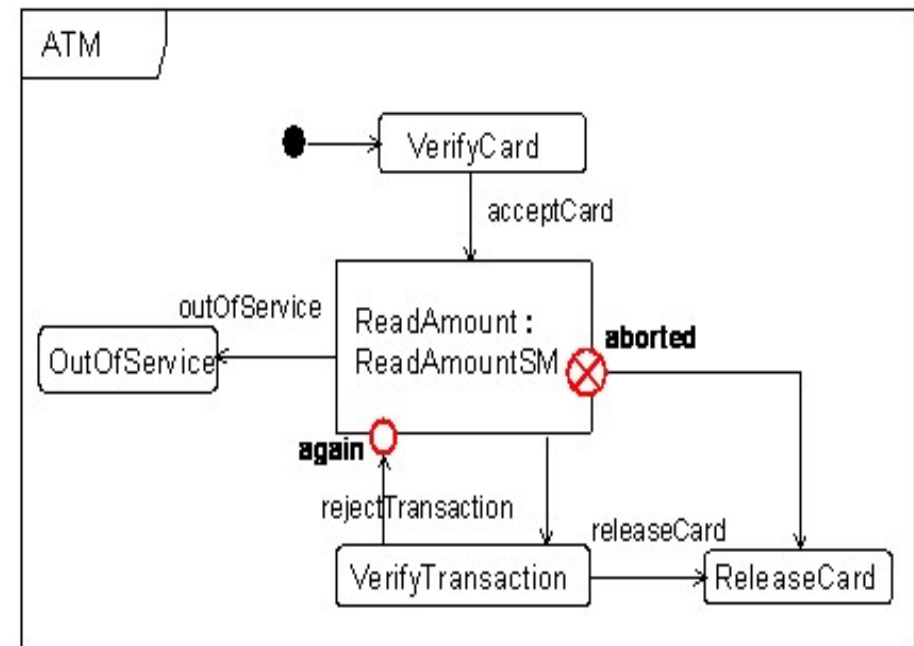
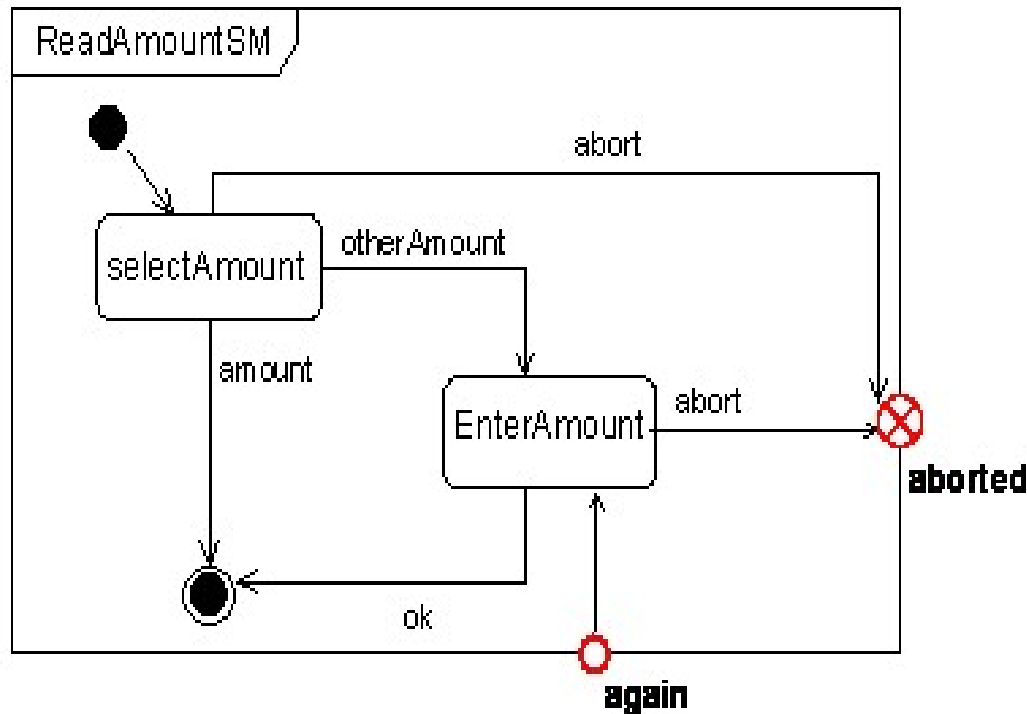
- Si usa quando si vuole descrivere uno stato composito in un diagramma a parte, per leggibilità o per definirlo e riusarlo in più contesti.
- La sottomacchina ha un nome (tipo), le istanze di uso si indicano con nomeIstanza:Tipo



This submachine can be used many times.

Sottomacchine: entry and exit points

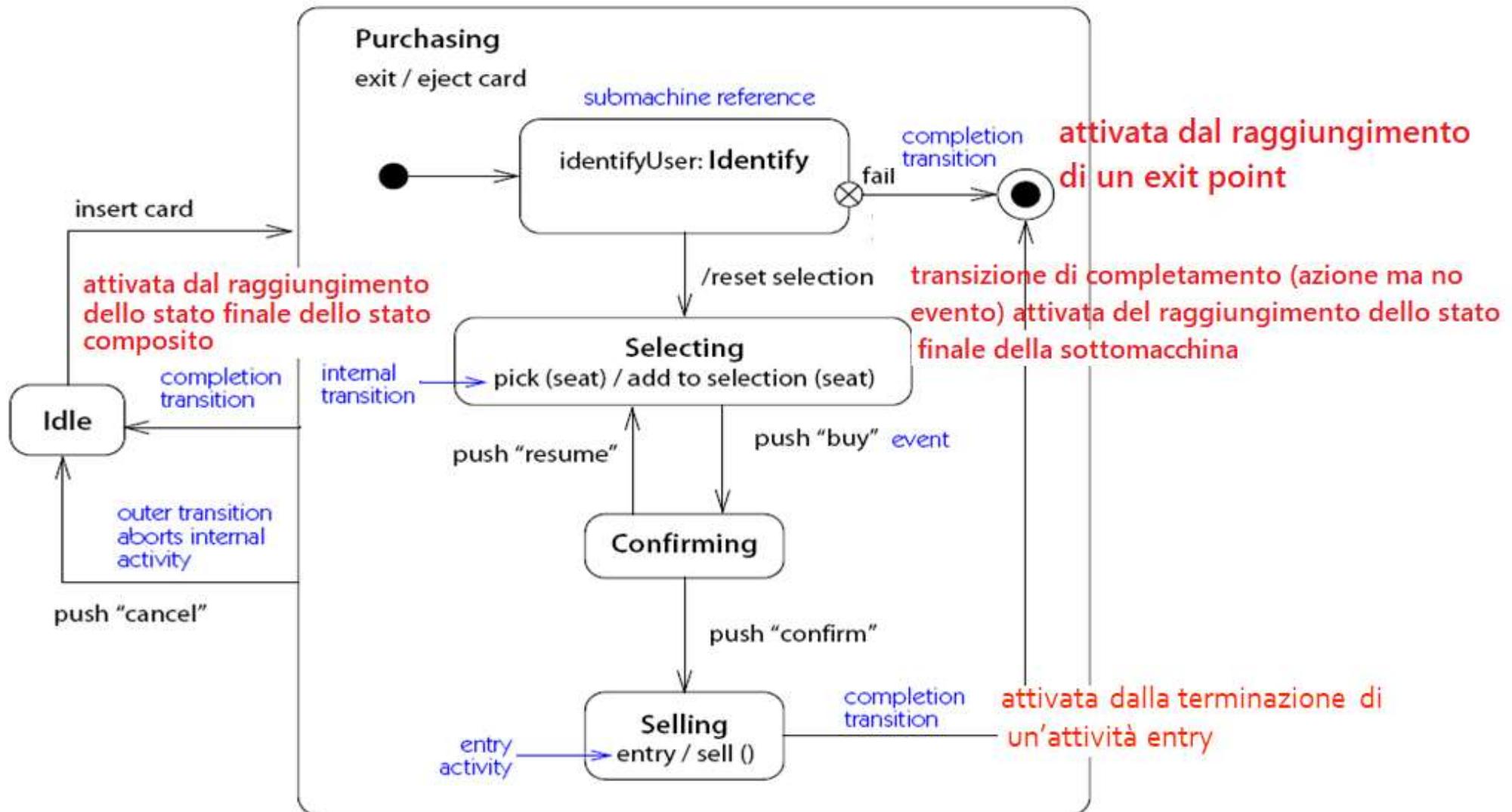
Una sottomacchina può definire entry and exit points
servono per collegare le transizioni della macchina principale



Transizioni di completamento: tutti i casi

- Senza evento, scattano al raggiungimento:
 1. Della terminazione di un'attività composta, i.e. al raggiungimento
 - Dello stato finale in un stato composto sequenziale
 - Degli stati finali di tutte le regioni ortogonali di un stato composto parallelo
 - Di un exit point
 2. Alla terminazione di entry e/o di do activity (la exit activity viene eseguita quando scatta la transizione di completamento)
 3. Di uno pseudo-stato giunzione (lo vedremo in un attimo)

Transizioni di completamento: esempio



Altri tipi di stato (pseudostati)

Giunzione



Storia

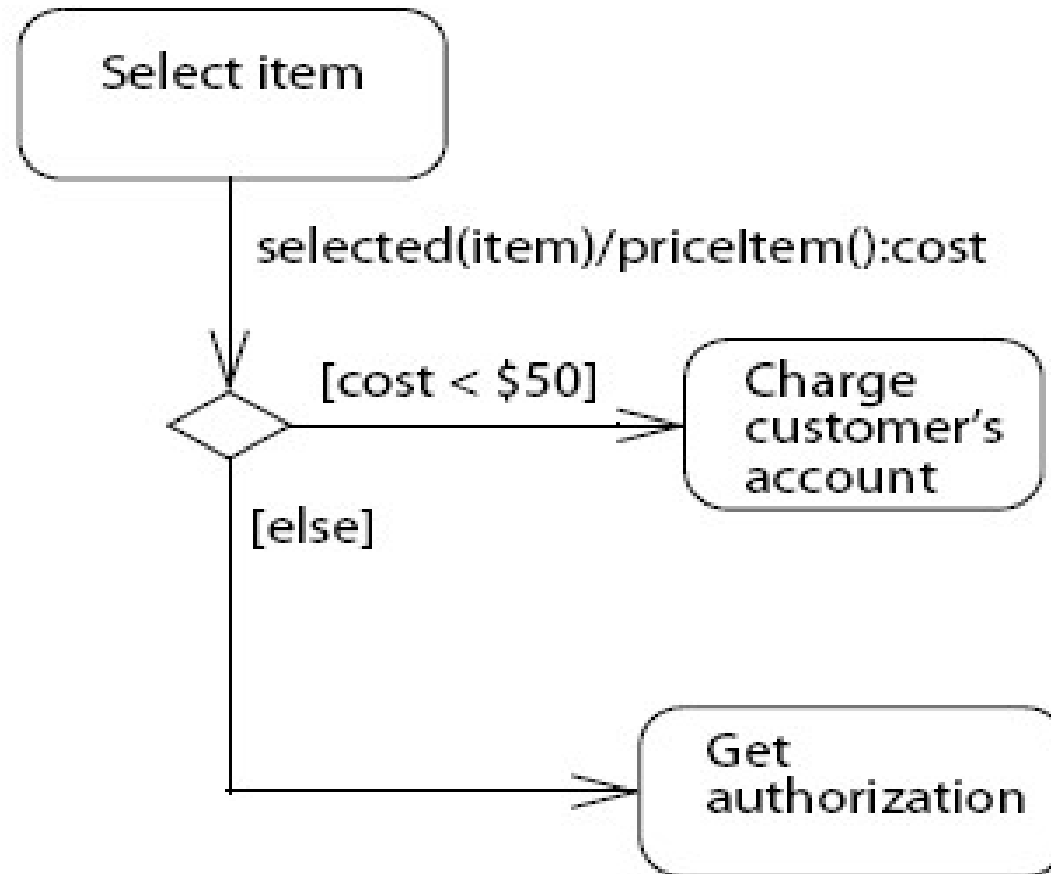


Decisione

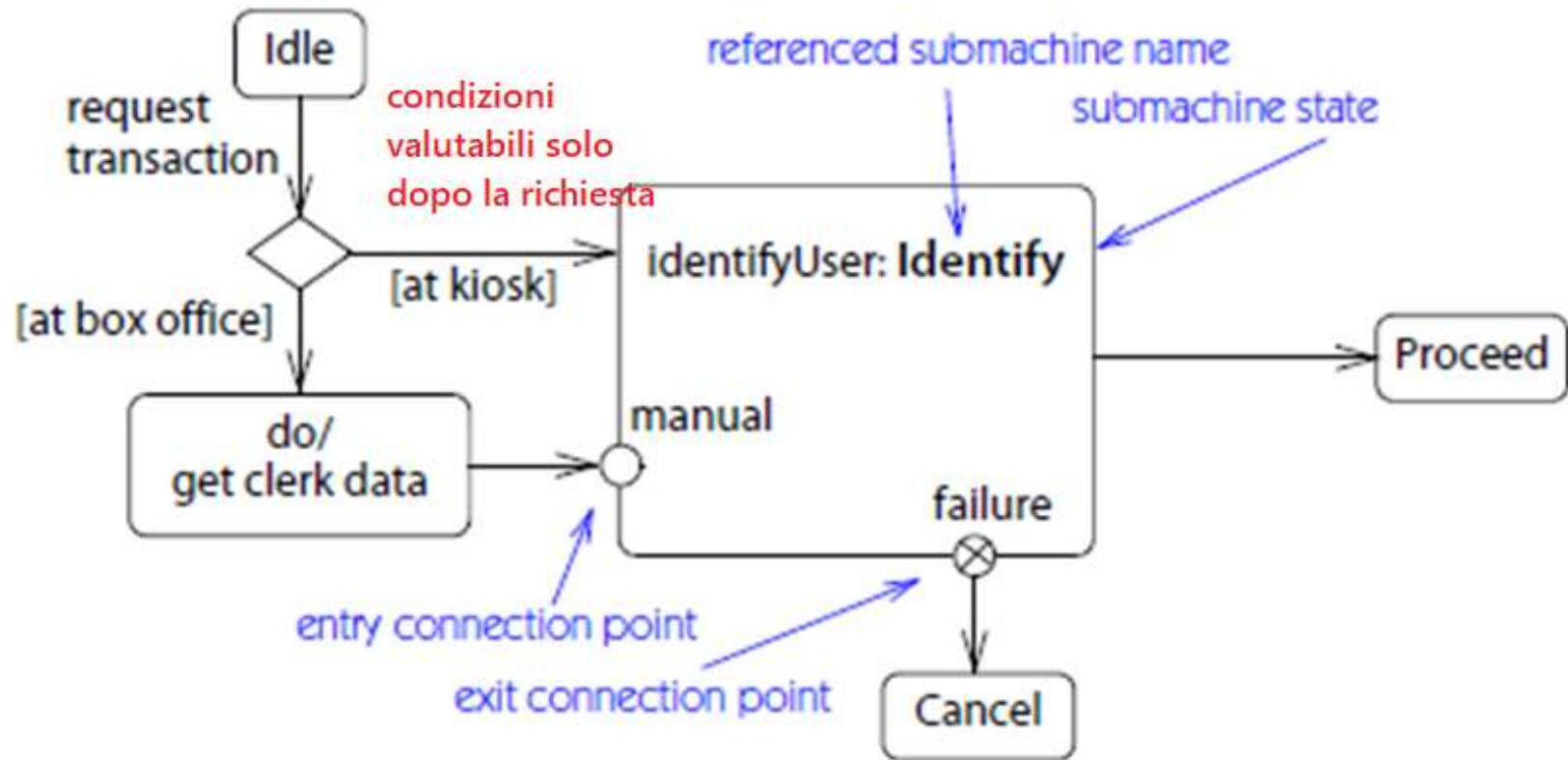


Esempio di choice

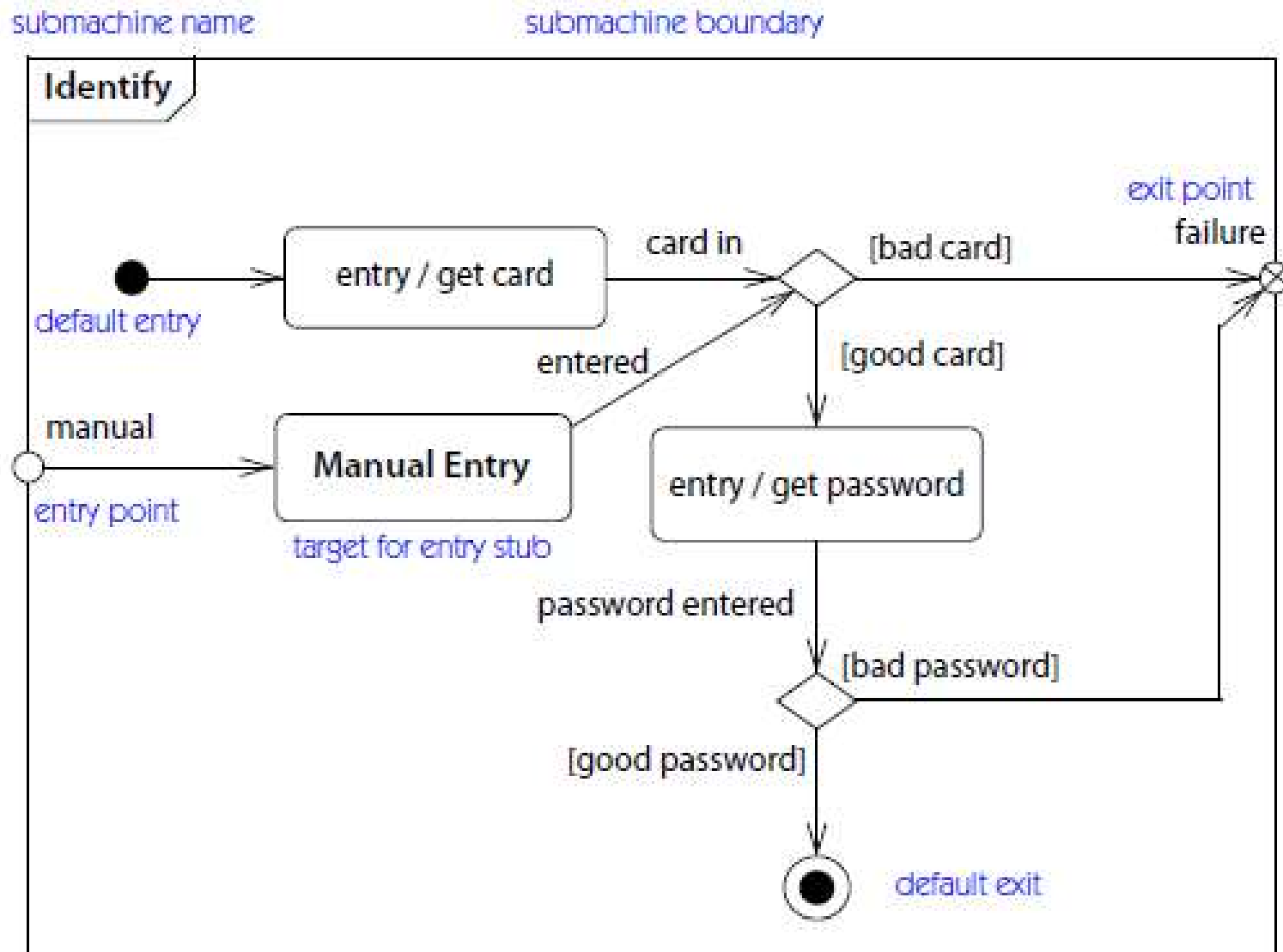
- Condizioni valutate **dinamicamente**
- Come per la choice dei diagrammi di attività:
 - La disgiunzione delle guardie deve valere "true"
 - È ammesso il non-determinismo



Esempio di choice (con invocazione di sotto-macchina)

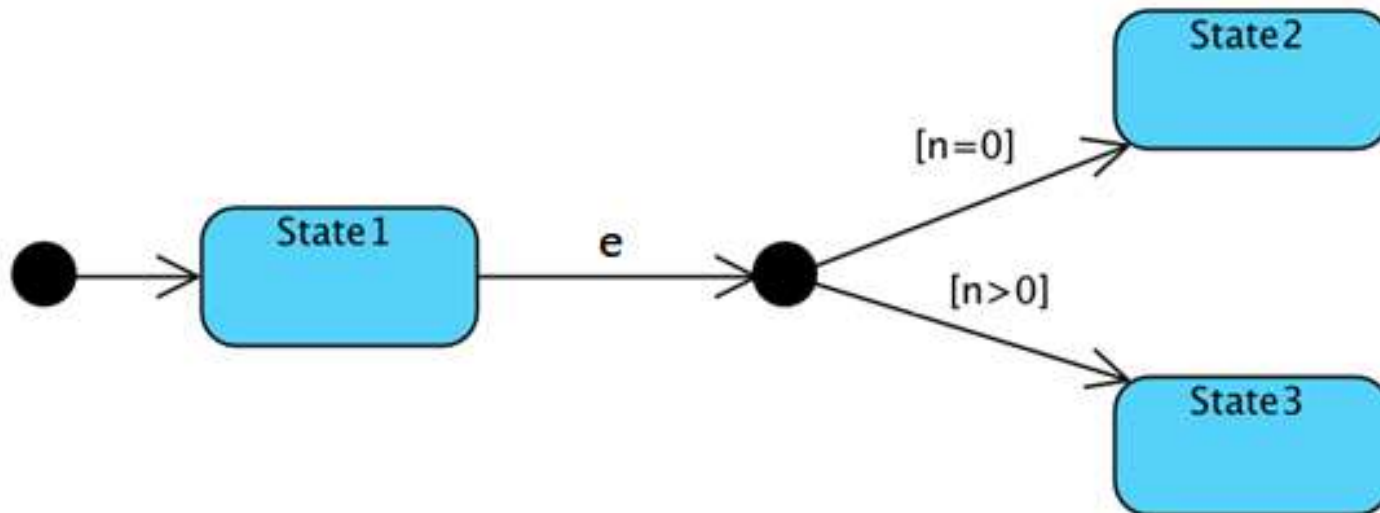


Esempio di choice (in una sotto-macchina)



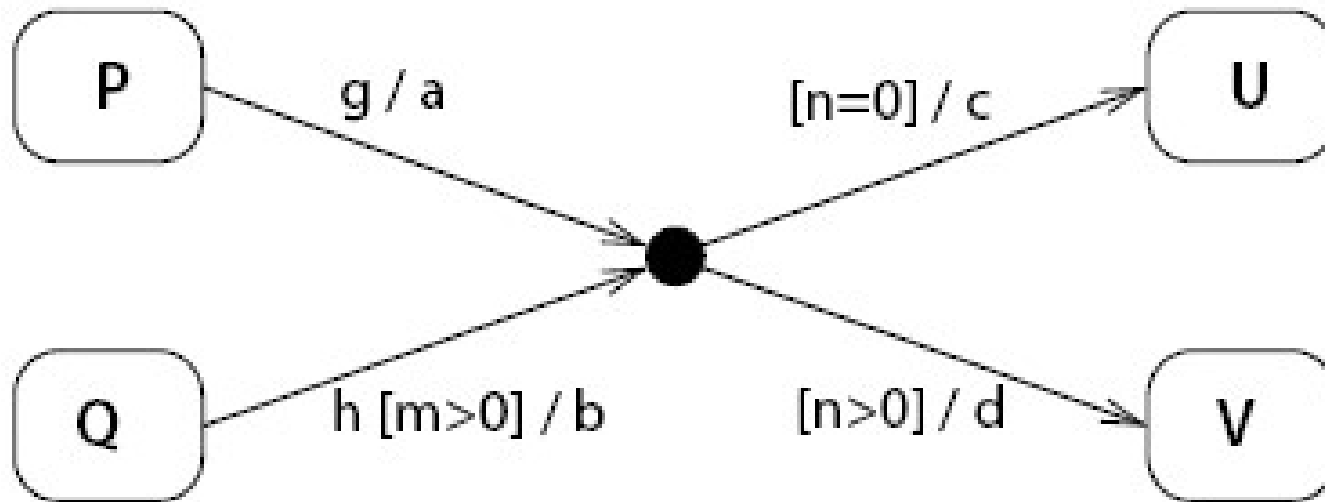
Giunzione

- Uno pseudo-stato da cui escono e/o entrano due o più transizioni
- Eventuali condizioni sono valutabili in modo statico
 - prima dell'evento e



- Se $n < 0$ l'evento e viene ignorato e si rimane nello stato 1

Giunzione, un altro esempio



Equivalent transitions:

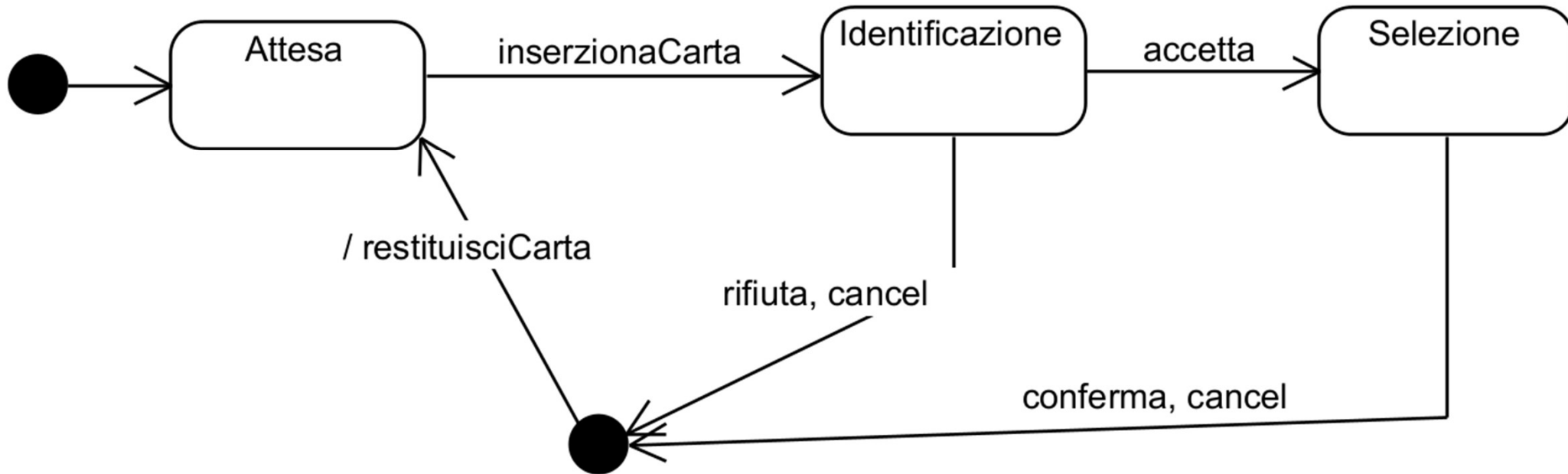
$P \xrightarrow{g} [n=0] / a, c \xrightarrow{c} U$

$P \xrightarrow{g} [n>0] / a, d \xrightarrow{d} V$

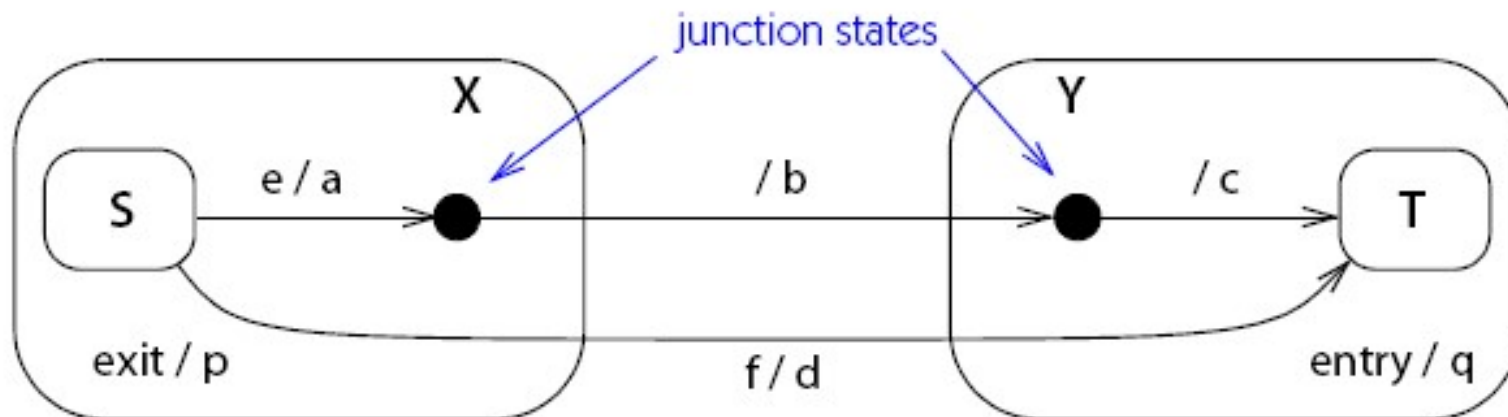
$Q \xrightarrow{h} [m>0 \text{ and } n=0] / b, c \xrightarrow{c} U$

$Q \xrightarrow{h} [m>0 \text{ and } n>0] / b, d \xrightarrow{d} V$

L'esempio visto prima con pseudostato giunzione



Giunzione, con entry e exit actions



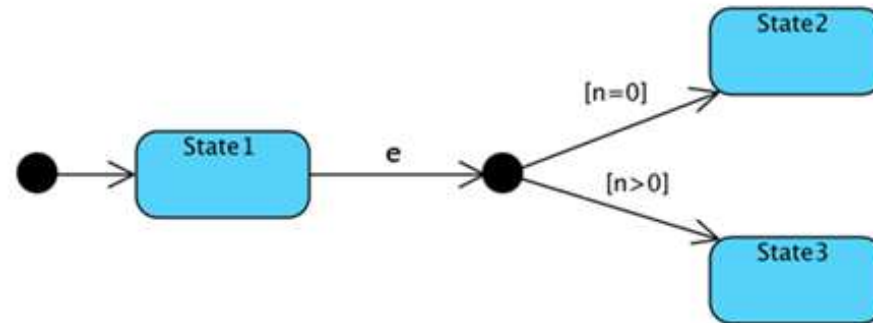
effective result

$e/a; p; b; q; c$

$f/p; d; q$

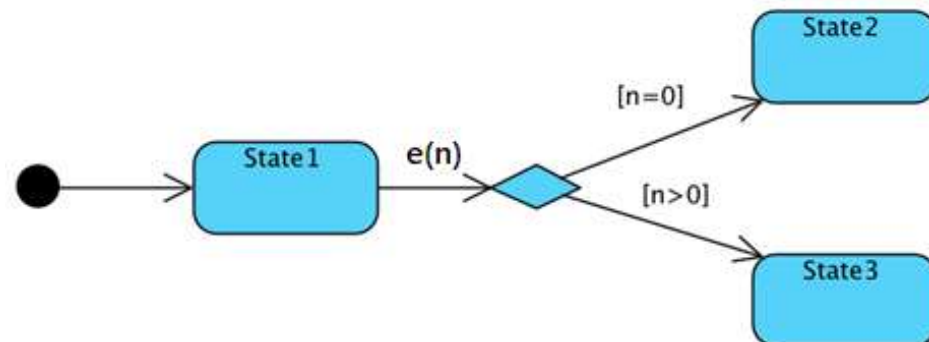
Giunzione vs choice

■ Giunzione (statica)



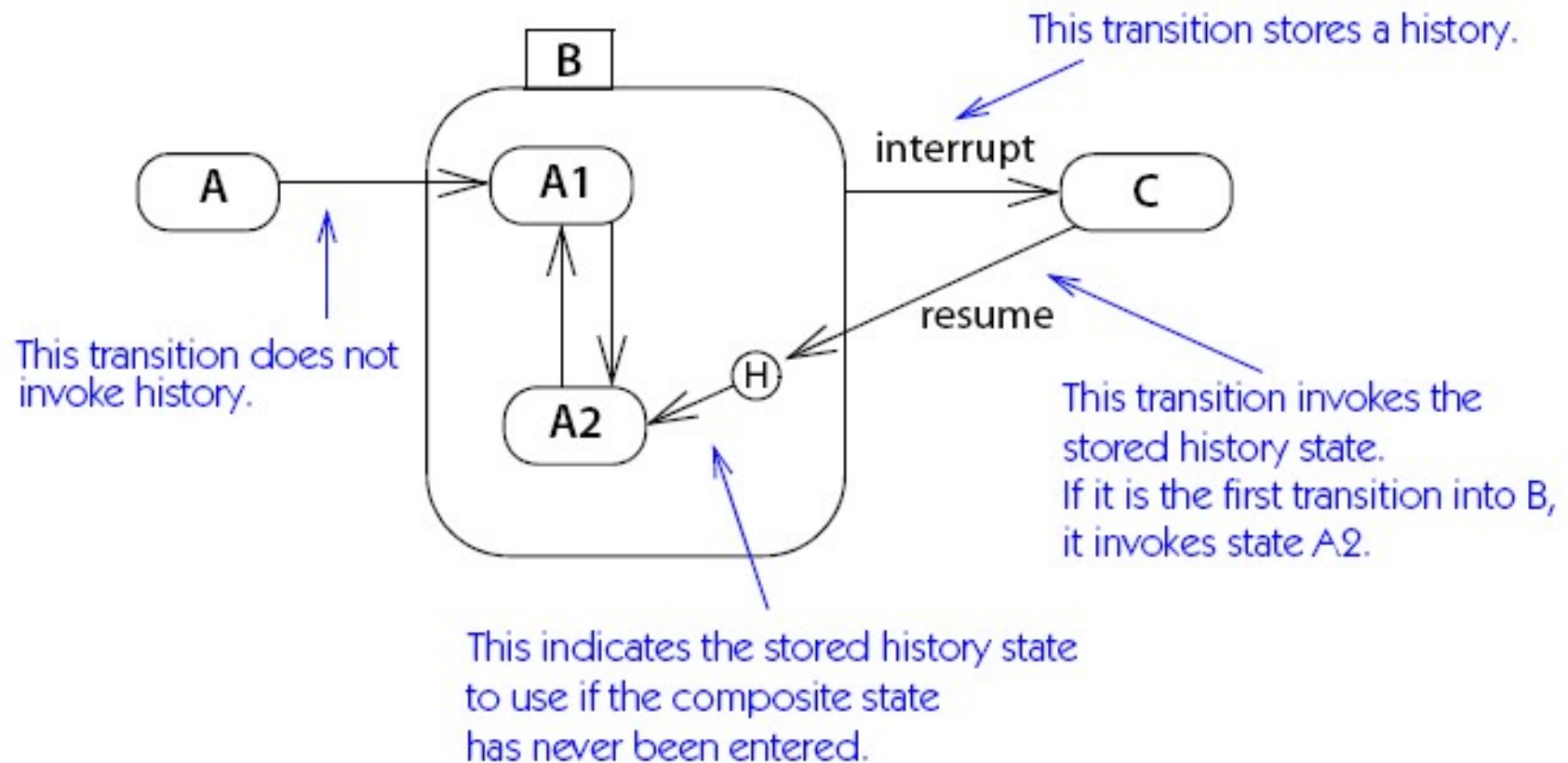
Le guardie sono valutate prima di uscire da State1. Se $n < 0$, l'evento e viene ignorato e nessuna transizione viene presa

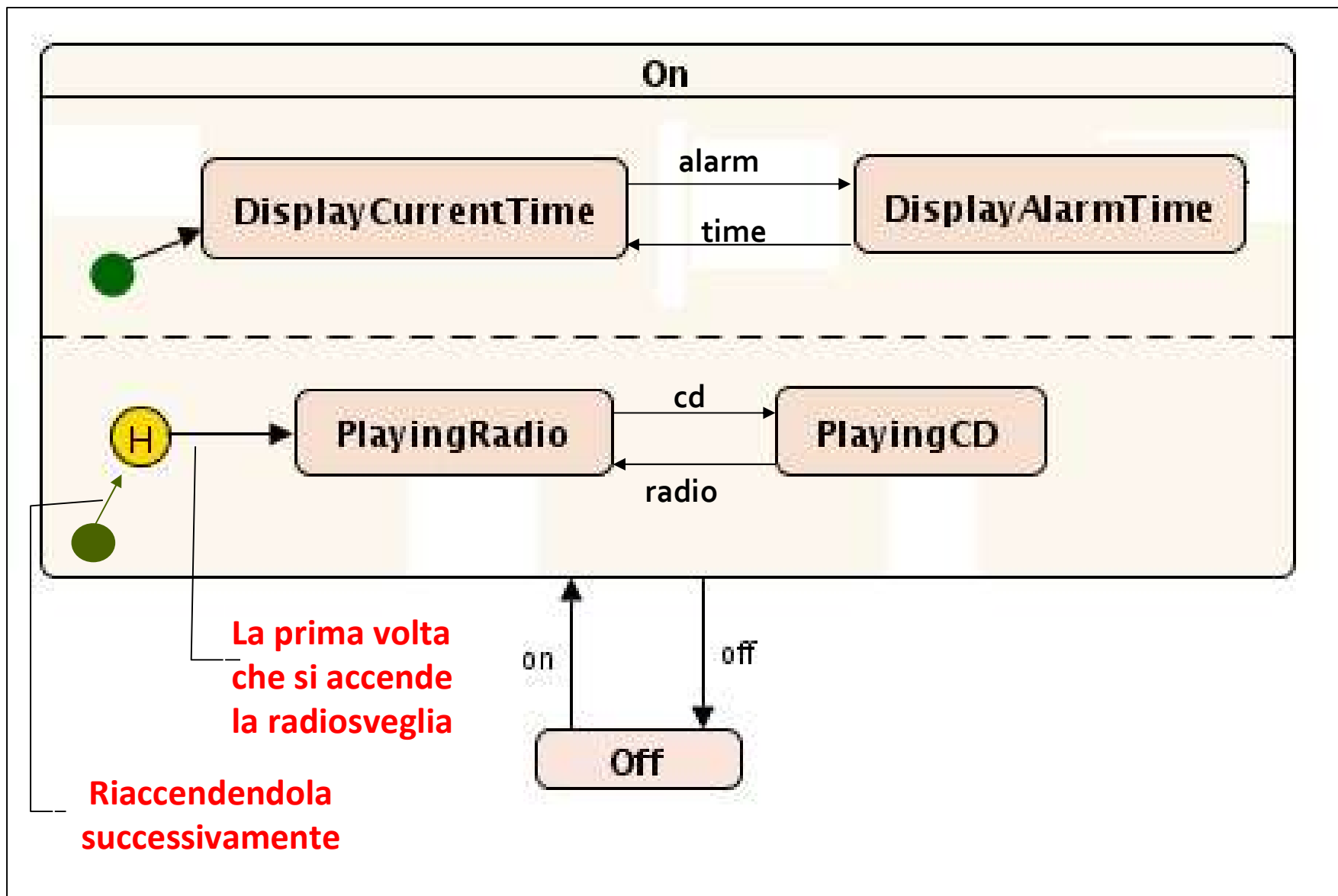
■ Choice (dinamica)



Le guardie sono valutate dopo $e(n)$. In questo esempio occorre avere garanzia che n sia maggiore o uguale a zero

Stato history





Tips and tricks



Macchina a stati o attività?

- Come scegliere il diagramma più appropriato per descrivere il modello dinamico
- Ricordiamo:
 - Il diagramma di macchina a stati parla dell'evoluzione nel tempo delle istanze di un classificatore
 - il diagramma di attività parla di un'agenda di azioni da fare.

Se il focus è

- mettere in ordine un insieme di azioni da fare → attività
- mostrare l'evoluzione di un oggetto in risposta a eventi → stati

Descrivere il modello dinamico: nomi degli stati e delle azioni

- Nomi degli stati:
 - aggettivi: attivo,
 - participi passati: accesa, spenta, pinned
 - gerundi: dialing, connecting
 - Altri: inAttesa
- Nomi delle azioni:
 - verbi all'indicativo, imperativo o infinito: crea, inviare
 - sostantivi che indicano un'azione: interrogazione DB
- Non è una regola e spesso nella pratica si disattende (eccezioni anche negli esempi visti), ma seguire la questa prassi è aiuta a costruire i diagrammi correttamente
- Inoltre: Errore comune nei compiti confondere stati e azioni

Homework

La Piscina: descrivere con un diagramma di macchina a stati che descrive 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 quali ad esempio arrivo fuori orario

Esercizi consigliati

- Dare un diagramma di macchina a stati che modelli
 - L'evoluzione nel tempo della classe Utente (o Situazione Utente) di Myair
 - L'evoluzione nel tempo del semaforo (ex. Semafori)

Syllabus

- UML@Classroom:
 - Cap 5 (tranne deep History)

Appendice

Da UML Reference Manual


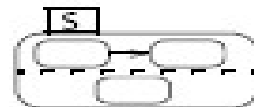







Table 7-1: *Kinds of Events*



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

Table 7-2: Kinds of Transitions and Implicit Effects

<i>Transition Kind</i>	<i>Description</i>	<i>Syntax</i>
entry transition	The specification of an entry activity that is executed when a state is entered	entry/ activity
exit transition	The specification of an exit activity that is executed when a state is exited	exit/ activity
external transition	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
internal transition	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

Table 7-3: Kinds of States

State Kind	Description	Notation
simple state	A state with no substructure	
orthogonal state	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.	
nonorthogonal state	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	
initial state	A pseudostate that indicates the starting state when the enclosing state is invoked	
final state	A special state whose activation indicates the enclosing state has completed activity	
terminate	A special state whose activation terminates execution of the object owning the state machine	
junction	A pseudostate that chains transition segments into a single run-to-completion transition	
choice	A pseudostate that performs a dynamic branch within a single run-to-completion transition	
history state	A pseudostate whose activation restores the previously active state within a composite state	

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