2009 - **2010**

Inginerie Software pentru Comunicatii (ISC / RST)

Titular curs: **Eduard-Cristian Popovici**

Suport curs: *http://discipline.elcom.pub.ro/isc/*

Moodle: *http://electronica07.curs.ncit.pub.ro/course/category.php?id=4*

Continut curs

1. Introducere in ingineria software

1.1. Necesitatea unei abordari sistematice a dezvoltarii software

1.2. Abordari si metodologii larg utilizate in ingineria software

2. Introducere in limbajul UML

2.1. Definirea, rolul si istoricul limbajului de modelare unificat (UML)

2.2. Tipuri de diagrame UML. Organizarea ierarhica a diagramelor

3. Diagrame UML statice

3.1. Diagrame UML de clase

3.2. Diagrame UML de obiecte

3.3. Diagrame UML de pachete

3.4. Diagrame UML de componente

3.5. Diagrame UML de structuri compozite

3.6. Diagrame UML de *deployment* (amplasare)

Continut curs

4. Diagrame UML dinamice

- 4.1. Diagramele UML de caz de utilizare
- 4.2. Diagrame UML de comunicatie si de robustete
- 4.3. Diagrame UML de secventa si de sumar al interactiunilor
- 4.4. Diagrame UML de masini de stari
- 4.5. Diagrame UML de activitati
- 4.6. Diagrame UML de timp

5. Introducere in procesul de dezvoltare Rational unificat (RUP)

- 5.1. Organizarea iterativa a proiectelor
- 5.2. Fazele si activitatile procesului RUP

6. Introducere in managementul si organizarea proceselor de dezvoltare

7. Elemente de reutilizabilitate a software-ului. Pattern-uri de proiectare

4. Diagrame UML dinamice

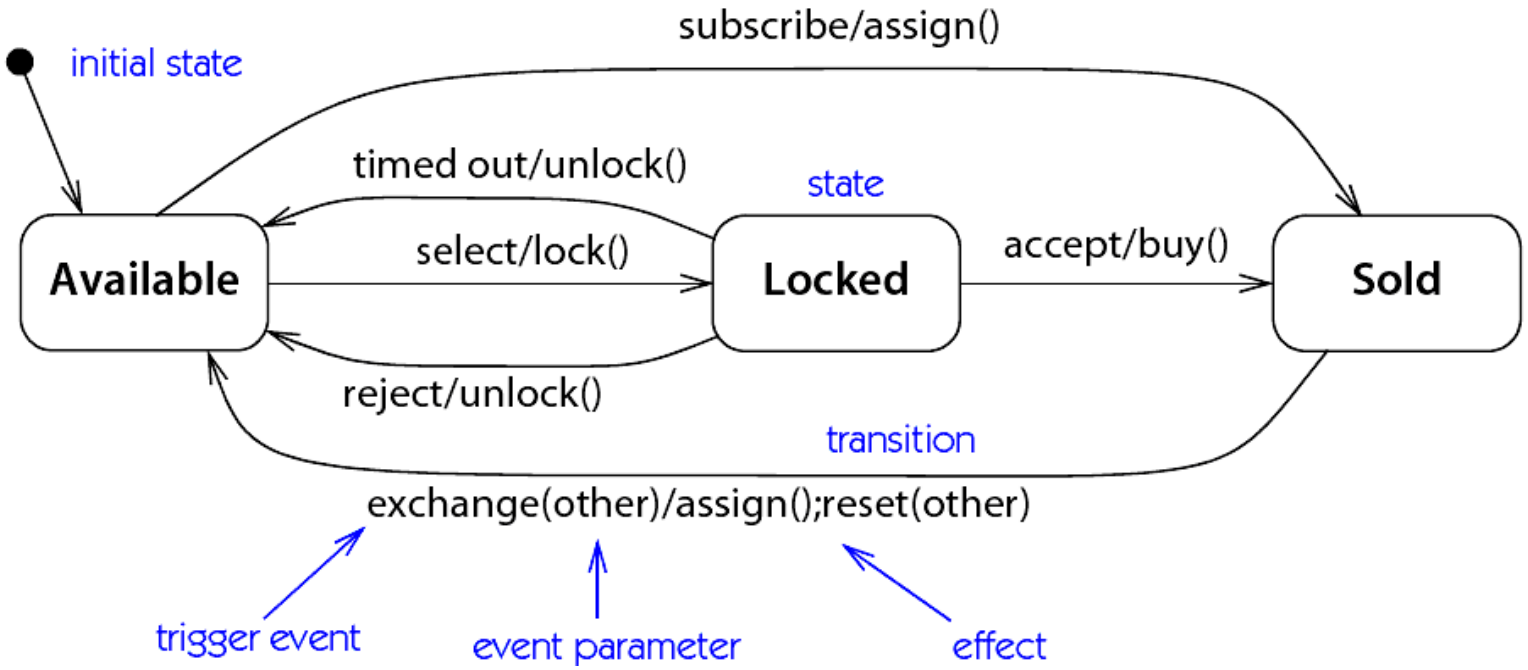
4.3. Diagrame UML de masini de stari

Diagramele masinilor de stari

- denumite anterior diagrame de ***stari-tranzitii***

- numite si **automate** (masini cu numar finit de stari), **diagrame FSM** (***Finite State Machines***), ***State Charts***, etc.

- prezinta **comportamentul unei clase în termeni de stari si de tranzitii intre stari**



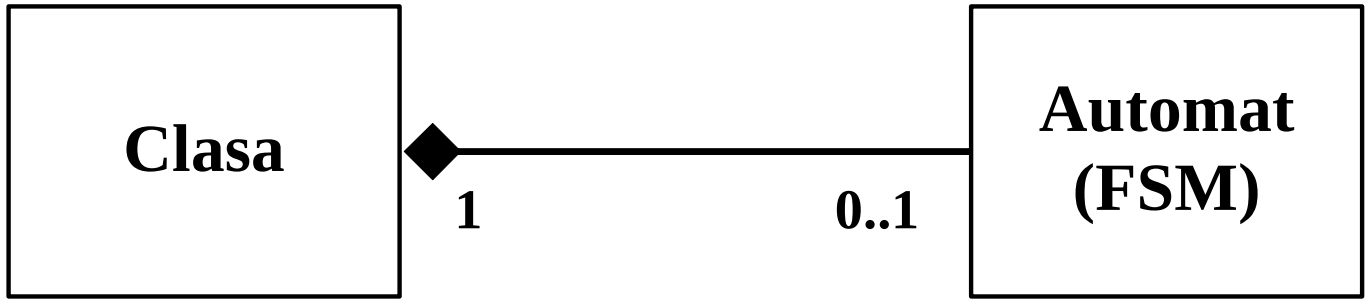
Diagramele masinilor de stari

Comportamentul obiectelor unei clase poate fi

- **descris** in mod **formal** in termeni de **stari**, de **tranzitii intre stari**, si de **evenimente** care declanseaza tranzitiile, prin intermediul **automatului asociat clasei** considerate

Automatul este o

- **abstractie a comportamentelor posibile** (asa cum diagramele de clase sunt abstractii ale structurii statice)



Diagramele masinilor de stari

Fiecare obiect

- urmeaza **comportamentul descris** in **automatul asociat clasei** sale si

- se gaseste **la un moment dat intr-o stare** care este **determinata de conditiile sale dinamice** (seria de stari anterioare, tranzitiile, evenimentele declansatoare, conditiile logice ale declansarii)

Automatele (FSM) si **scenariile** (MSC) sunt **complementare**

Scenariile (MSC) se reprezinta printr-o **colaborare** intre obiecte si prezinta

- **interactiunile** unui **ansamblu de obiecte** sau

- **interactiunile ansamblului componentelor** unui sistem

Automatele (FSM) prezinta

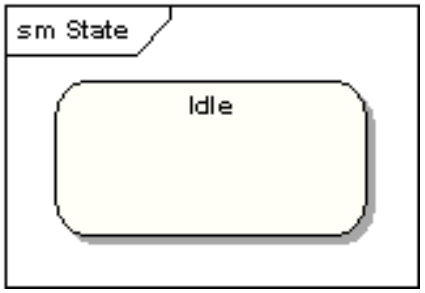
- **dinamica interna** a unui **obiect** fara a intra in detalii structurale sau

- **dinamica interna** a unui **sistem** fara a intra in detalii structurale

Diagramele masinilor de stari

Orice obiect

- se gaseste **intotdeauna intr-o stare data** pentru un anumit interval de timp
- **nu poate fi** intr-o **stare necunoscuta** sau **nedefinita**



Starea este

- caracterizata prin **durata** si **stabilitate**
- **reprezentata** in UML printr-un **dreptunghi** cu **colturile rotunjite**

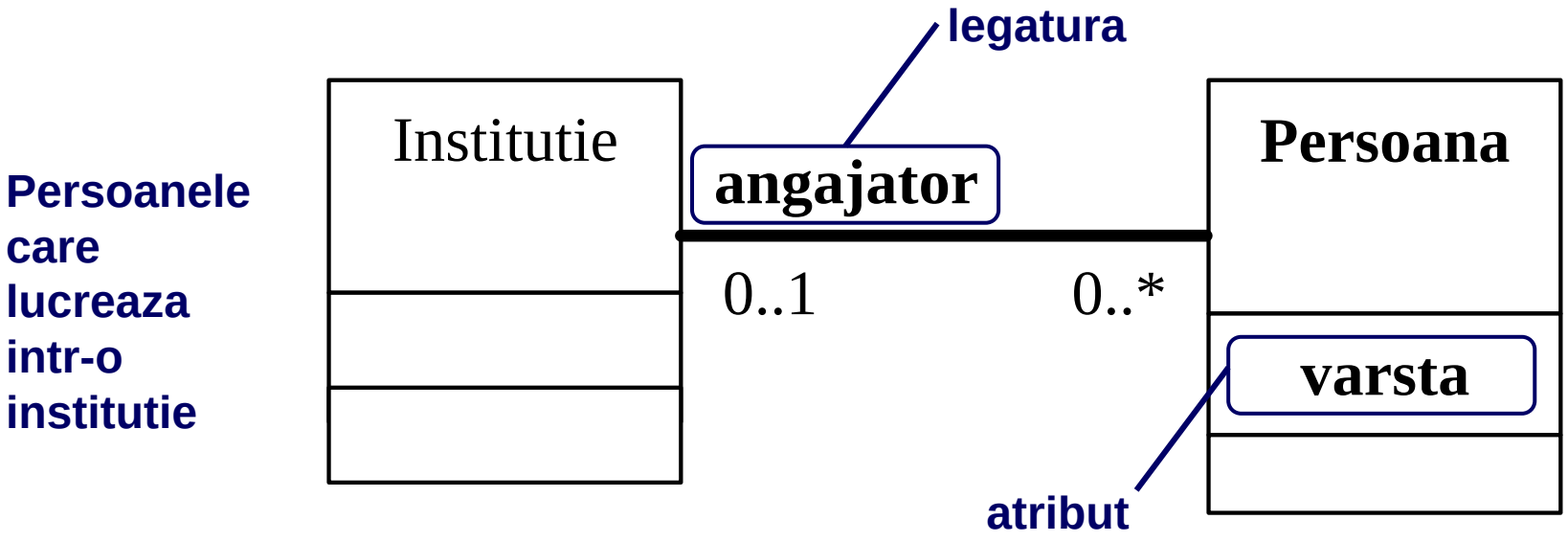
O stare

O alta stare

Diagramele masinilor de stari

Starea este

- **imaginea ansamblului curent**
  - al **valorilor atributelor** obiectului, si
  - al **prezentei sau absentei legaturilor** de la obiectul respectiv **catre alte obiecte**

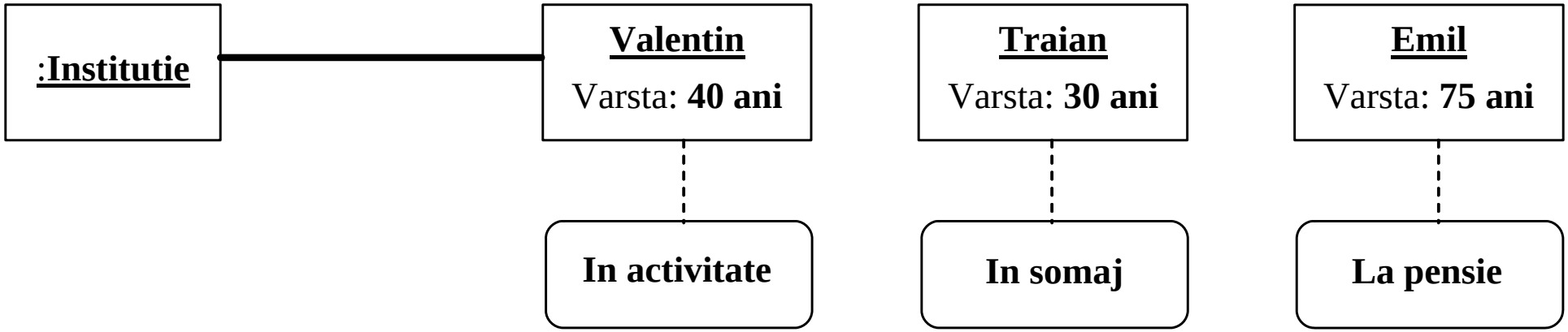


Diagramele masinilor de stari

Pentru a cunoaste situatia unei persoane in particular, trebuie **studiate**:

- **varsta** persoanei (care este **atribut** al clasei Persoana)
- prezenta **legaturii** (care este **tot atribut** al clasei Persoana) cu o Institutie

In diagrama urmatoare:



Traian (de **30 ani**) nu are **nicio** legatura cu o institutie **=>** este in starea **In somaj**

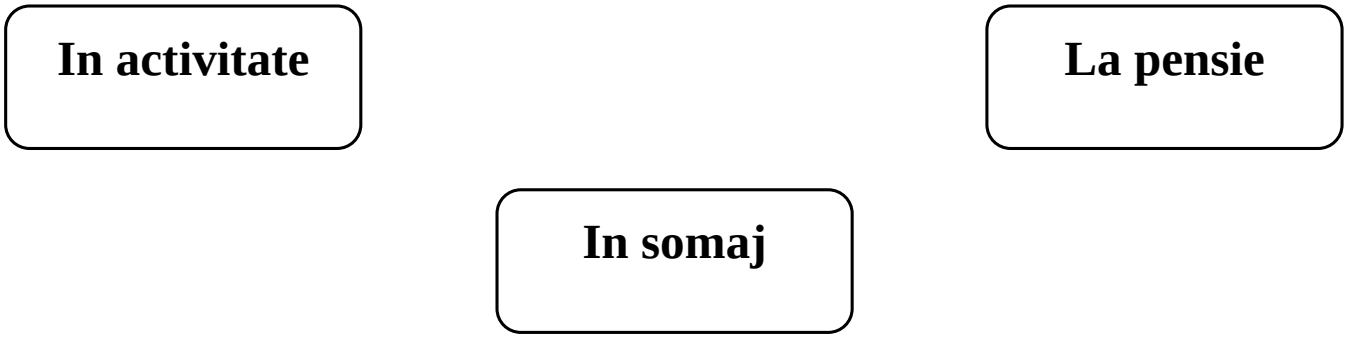
Valentin (de **40 ani**) are o **legatura** cu o institutie **=>** este in starea **In activitate**

Emil (de **75 ani**) nu are **nicio** legatura cu o institutie **=>** este in starea **La pensie**

Diagramele masinilor de stari

Persoanele nu au toate un loc de munca si

- se gasesc **la un moment dat** intr-**una dintre starile urmatoare**



Pentru a cunoaste situatia unei persoane in particular, trebuie **studiate**:

- **varsta** persoanei (care este **atribut** al clasei Persoana)
- prezenta **legaturii** (care este **tot atribut** al clasei Persoana) cu o Institutie

Diagramele masinilor de stari

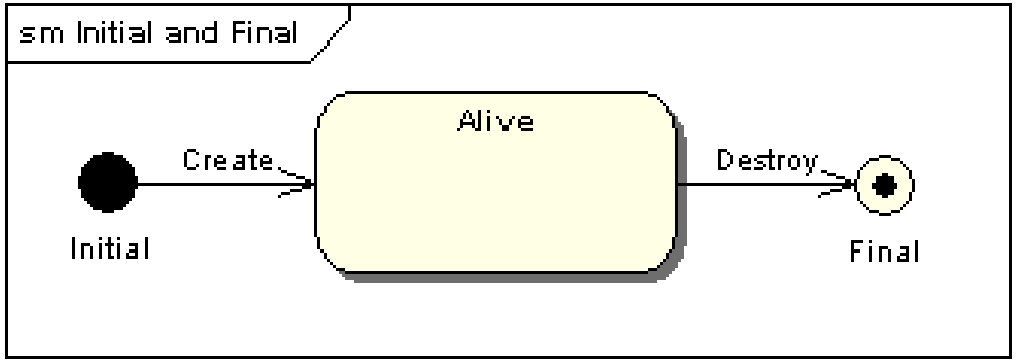
La un nivel ierarhic dat exista intotdeauna o **(pseudo)stare initiala** unica

In schimb este **posibila existenta mai multor (pseudo)stari finale**, in cazul sistemelor care nu se opresc niciodata

pseudo(stare)
initiala

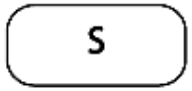
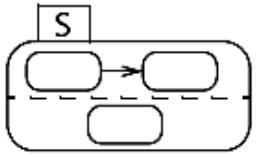
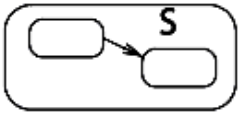
Stare obisnuita
(intermediara)

pseudo(stare)
finala



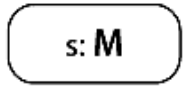
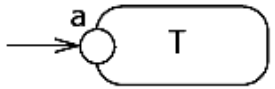
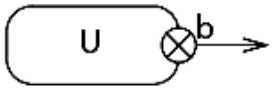
Diagramele masinilor de stari

Tipuri de stari cu **descrierea** si **modul lor de reprezentare**

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	

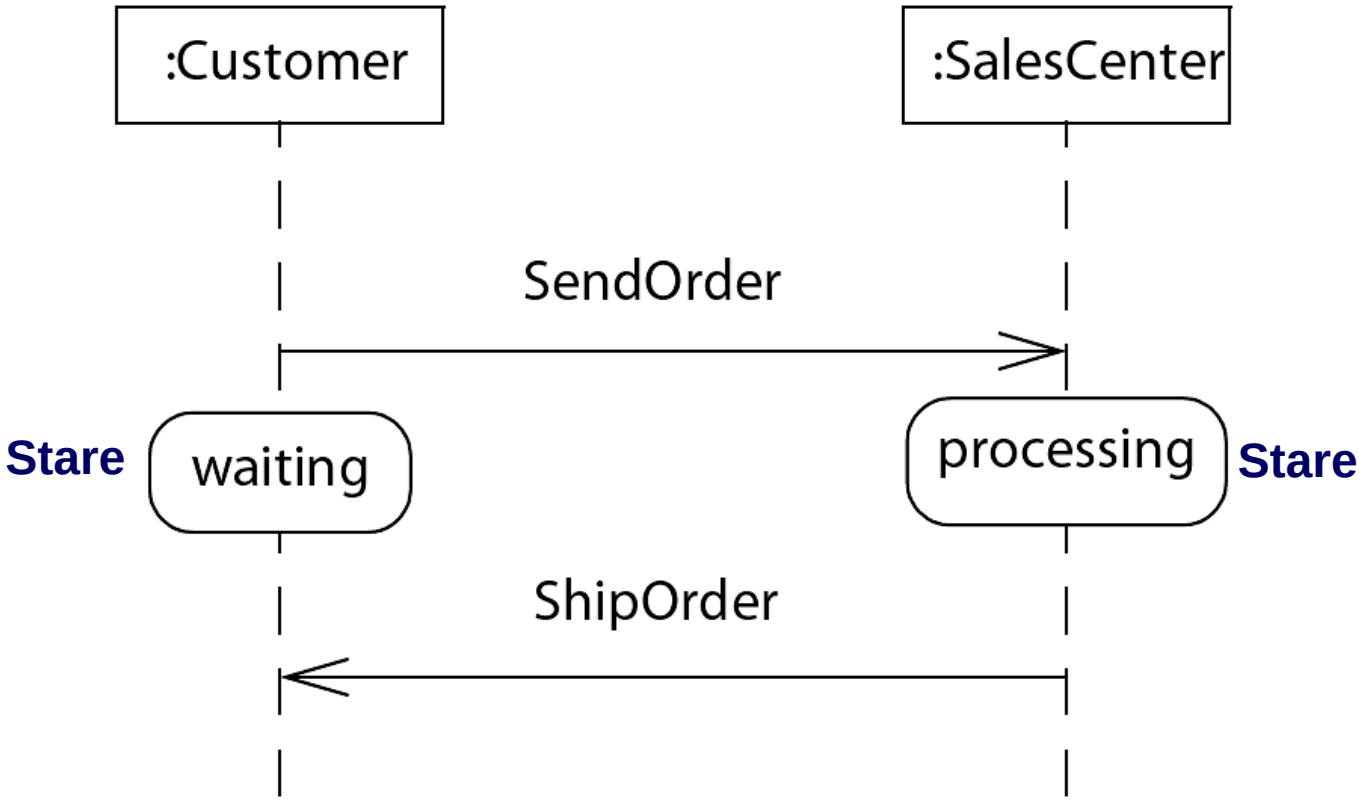
Diagramele masinilor de stari

Tipuri de stari cu **descrierea** si **modul lor de reprezentare**

State Kind	*Description*	*Notation*
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	
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	

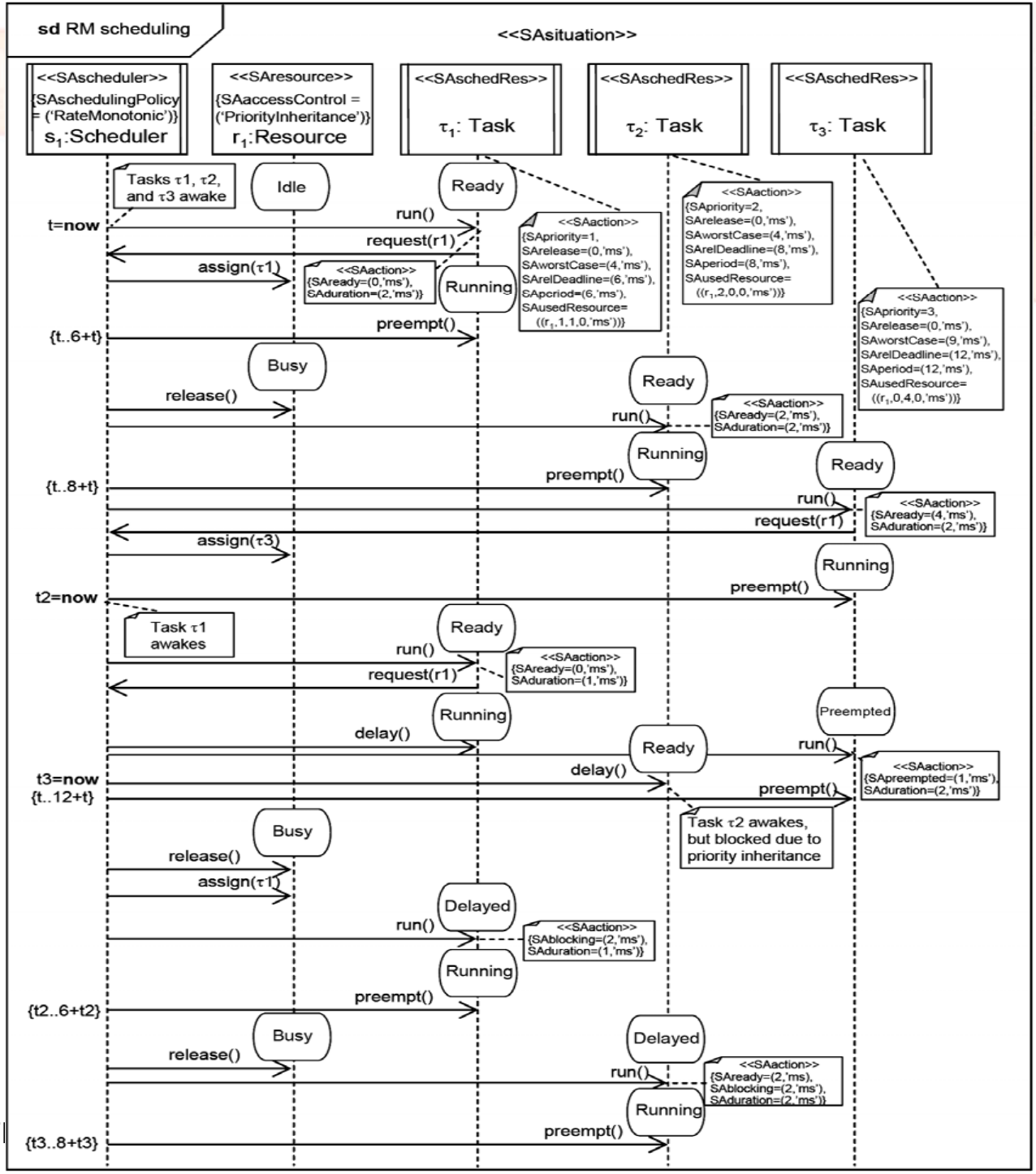
Diagramele masinilor de stari

Starile pot fi reprezentate si **in diagramele de secventa (MSC)**



Diagramele masinilor de stari

Starile pot fi reprezentate si **in diagramele de secventa (MSC)**



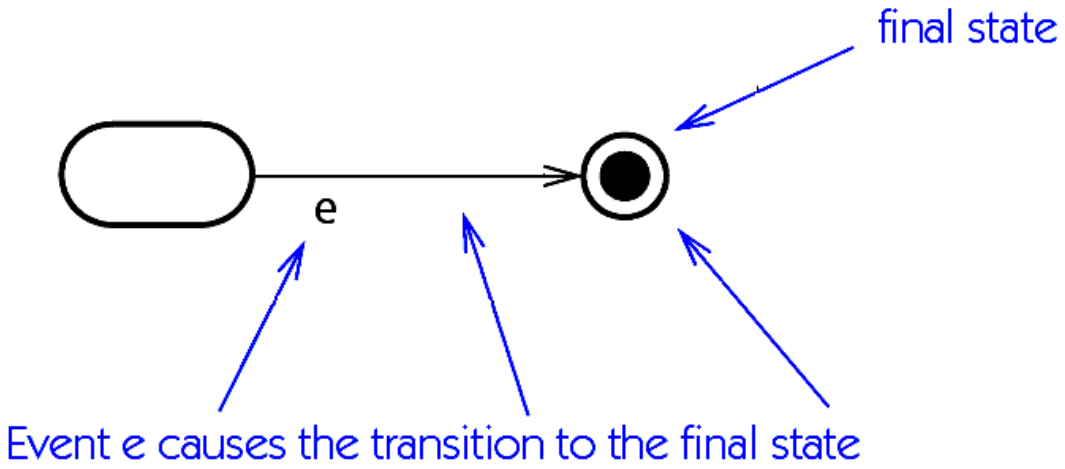
Diagramele masinilor de stari

Atunci cand **conditiile dinamice evolueaza**

- **obiectele** isi **schimba starea** urmand regulile descrise in **automatul** asociat clasei lor

Trecerea dintr-o stare in alta se efectueaza

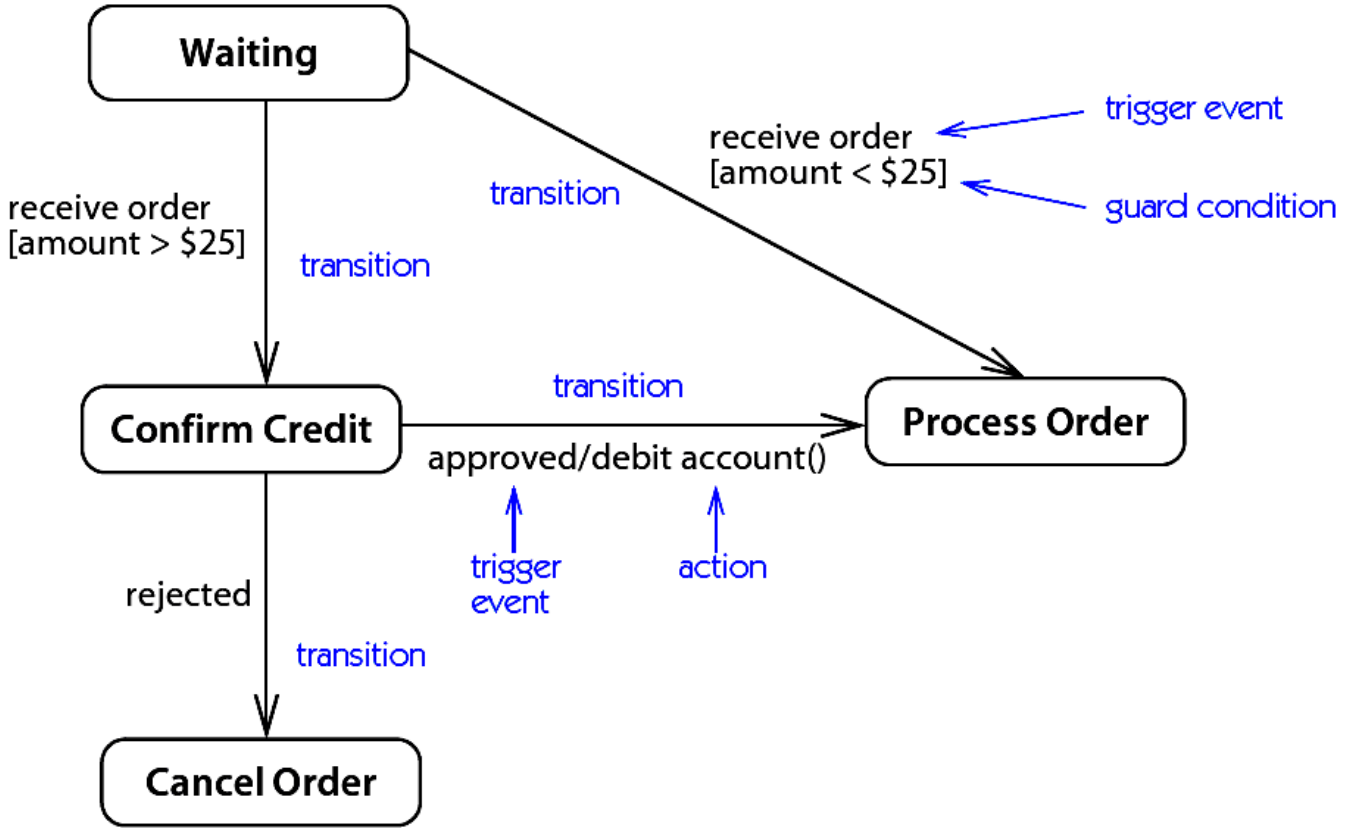
- atunci cand o **tranzitie** este **declansata** de catre un **eveniment** care apare in domeniul problemei



Diagramele masinilor de stari

Diagramele de masini de stari sunt **grafuri orientate** in sensul ca

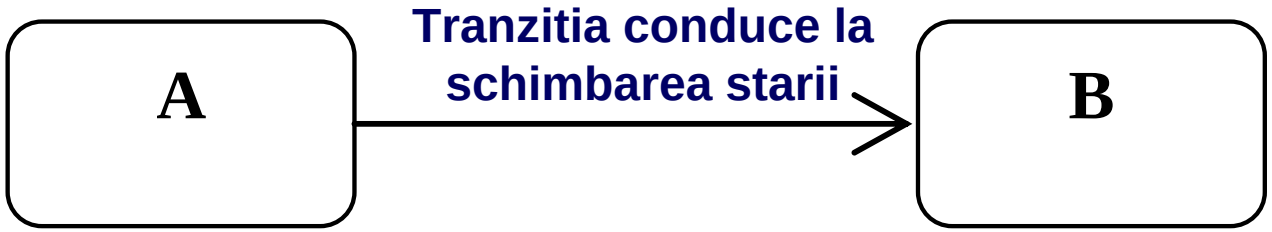
- starile sunt unite prin **conexiuni unidirectionale** (reprezentate sub forma de **sageti**) denumite **tranzitii**



Diagramele masinilor de stari

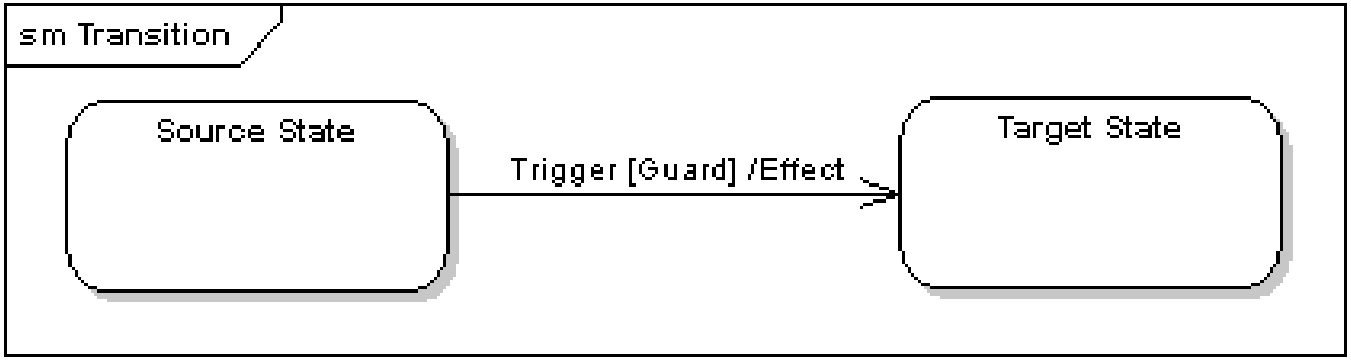
Trecerea dintr-o stare in alta (tranzitia) este **instantanee** (de **durata neglijabila** la scara de timp a executiei sistemului software) caci

- sistemul trebuie **intotdeauna** sa se afle intr-o **stare cunoscuta**



Sintaxa generala a unei tranzitii este urmatoarea

numeTranzitie: numeEveniment (listaParametri) [conditieLogica] / listaEfecte



Diagramele masinilor de stari

Sintaxa generala a unei tranzitii este urmatoarea

numeTranzitie: numeEveniment (listaParametri) [conditieLogica] / listaEfecte

Tranzitia poate fi descrisa prin:

- **numele tranzitiei** (optional) – urmat de ‘:’ (doua puncte)
- **numele evenimentului** (optional) care **declanseaza** tranzitia
- **lista** (optionala) de **parametri ai evenimentului** – in paranteze rotunde
- **conditia logica** (optionala) care poate fi **impusa pentru a se produce** tranzitia – in paranteze drepte
- **lista** (optionala) de **efecte ale tranzitiei** – dupa un ‘/’ (slash)

De exemplu

activareDistribuitorCafea: introducereBancnota **(** valoareBancnota **) [** bancnotaValida **] /** afisareMesajCerandSelectareaTipuluiDeCafea

Diagramele masinilor de stari

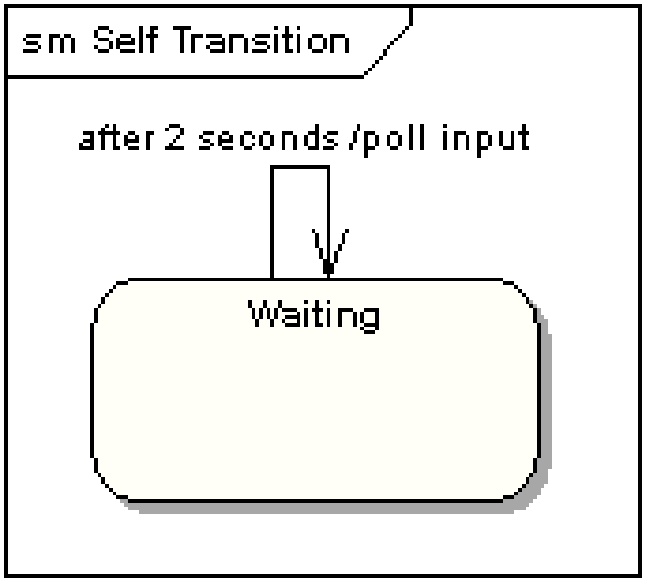
Tipuri de tranzitii cu **descrierea** si **sintaxa lor tipica**

Transition Kind	*Description*	*Syntax*
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

Diagramele masinilor de stari

Tranzitiile nu leaga neaparat stari distincte

- putand exista **trazitii reflexive** care pleaca dintr-o stare catre aceeasi stare



Diagramele masinilor de stari

Evenimentul

- corespunde **aparitiei** unei **situatii** date **in domeniul problemei**

- este prin natura lui **instantaneu**, fiind o informatie care trebuie sa fie tratata imediat (spre deosebire de **stare** care **are** o **durata**)

- serveste drept **declansator** al trecerii dintr-o stare in alta (tranzitiei)

Daca

- **tranzitiile** indica **traseele** in graful de stari

- **evenimentele** determina **care traseu va fi urmat**

Evenimentele, **tranzitiile** si **starile** sunt **indisociabile** in descrierea comportamentului dinamic

- **un obiect** aflat intr-o **stare data**, asteapta **aparitia unui eveniment** pentru a trece intr-o alta stare (a efectua o **tranzitie**)

Diagramele masinilor de stari

Tipuri de evenimente cu **descrierea** si **sintaxa lor tipica**

Event Type	*Description*	*Syntax*
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)

Diagramele masinilor de stari

Sintaxa generala a unui eveniment este urmatoarea

numeEveniment (numeParametru : tipParametru, ... **)**

Specificatia unui eveniment cuprinde:

- **numele evenimentului** care **declanseaza** tranzitia
- **lista** (optionala) de **parametri ai evenimentului** – in paranteze rotunde
  - **parametrii** fiind **perechi** nume : tip, **despartite prin virgula**, aflate in interiorul parantezei rotunde

De exemplu

introducereBancnota **(** valoareBancnota : int **)**

Diagramele masinilor de stari

Conditia logica (***guard condition***) este

- o **conditie booleana** care **valideaza** sau nu **declansarea** unei **tranzitii** la aparitia unui eveniment

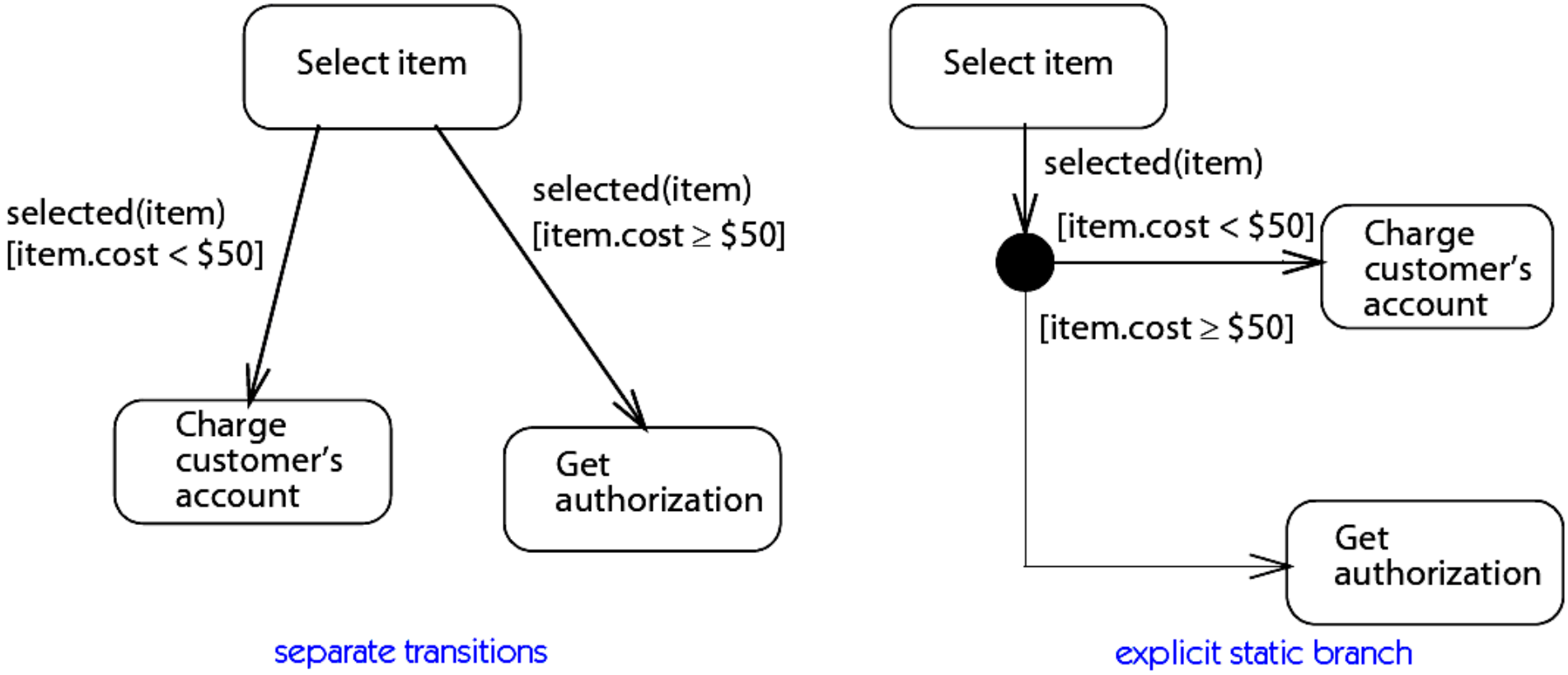
- permite mentinerea aspectului **determinist** al unei masini de stari (cand tranzitii diferite pot fi declansate in aceeasi stare de catre acelasi eveniment)

Atunci **cand are loc** un **eveniment**

- sunt **evaluate conditiile logice** (care trebuie sa fie **reciproc exclusive**) si
  - daca **tranzitia** este **validata** si
    - **tranzitia** (schimbarea starii) este **declansata**
  - altfel **starea** este **pastrata**

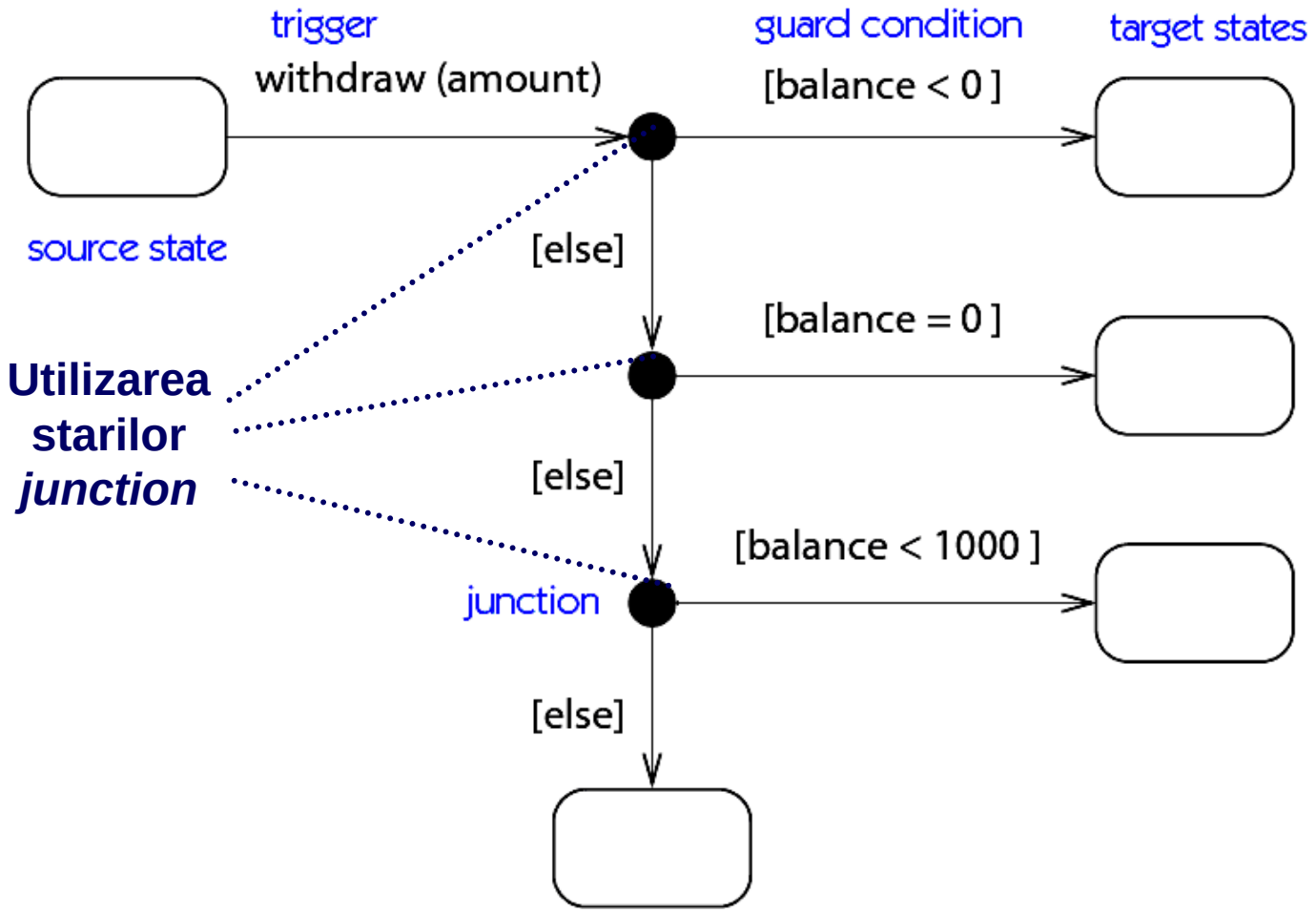
Diagramele masinilor de stari

Reprezentari echivalente ale conditiilor logice



Diagramele masinilor de stari

Reprezentari echivalente ale conditiilor logice (continuare)



Diagramele masinilor de stari

Legaturile dintre

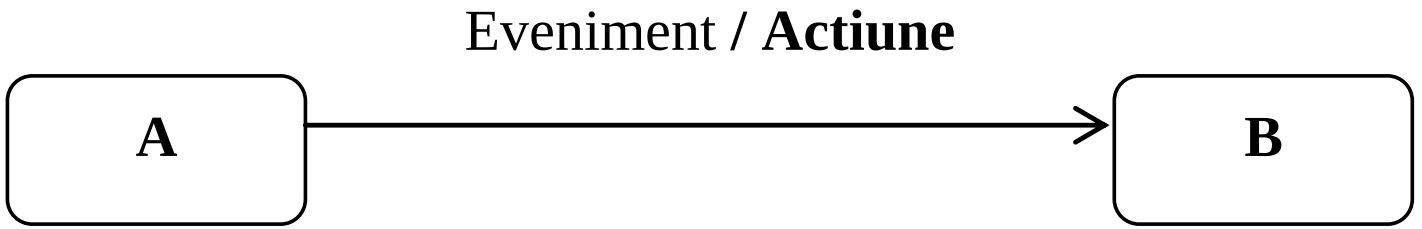
- **operatiile** din **specificatia clasei** si
- **evenimentele** care apar in **diagramele masinilor de stari**

sunt realizate prin intermediul

- **actiunilor** (operatii care au **durata neglijabila**, considerate **instantanee** si **atomice** – nu poate fi separata in parti) si al
- **activitatilor** (operatii care au **durata semnificativa**)

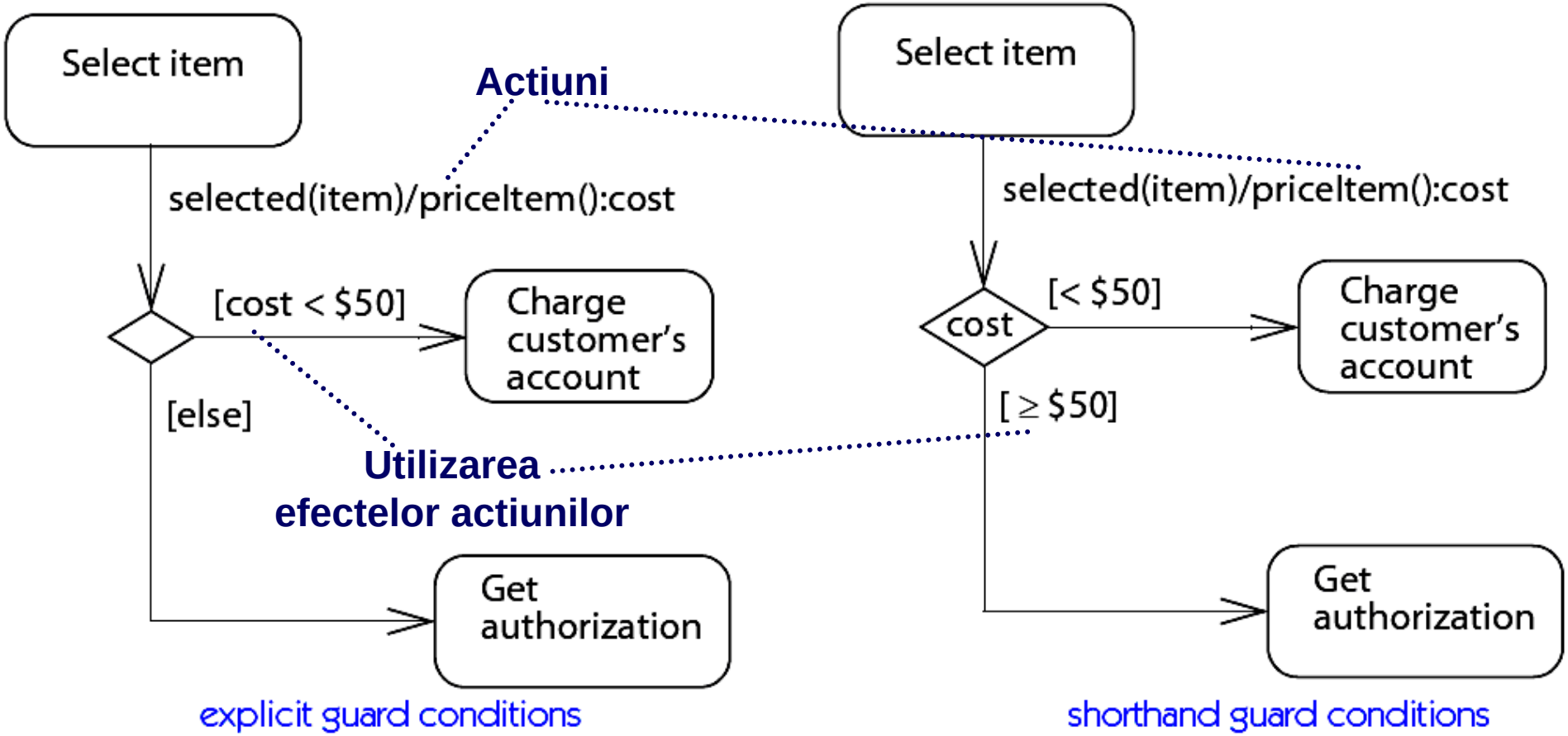
Tranzitia poate fi decorata cu

- o **actiune efect** care **se executa daca tranzitia este declansata**



Diagramele masinilor de stari

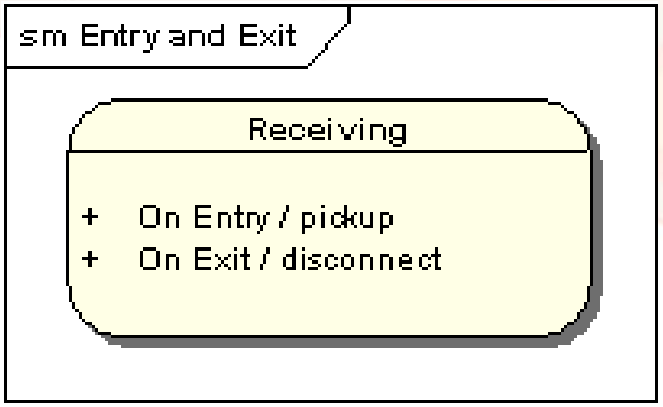
Reprezentari echivalente ale conditiilor logice (continuare)



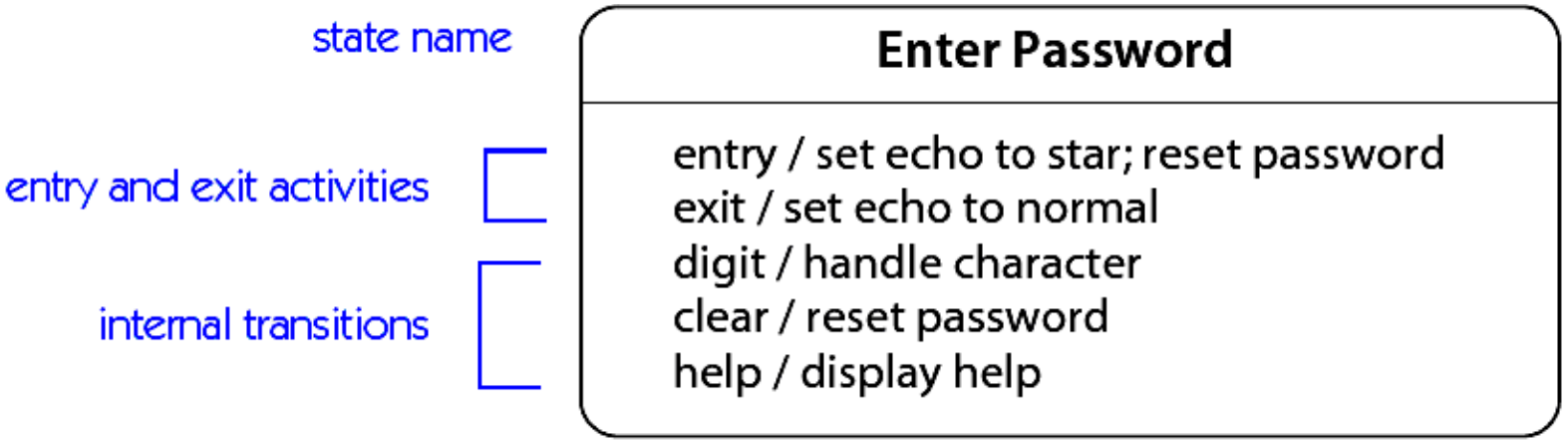
Diagramele masinilor de stari

Si **starile pot sa contina actiuni** care sunt **executate**

- **dupa intrarea in** sau **iesirea din** starea data
  - actiunile ***entry*** (***On Entry***) si ***exit*** (***On Exit***)



- la **aparitia unui eveniment pe durata pastrarii starii** respective a obiectului
  - actiunile **interne** corespunzatoare tranzitiilor interne



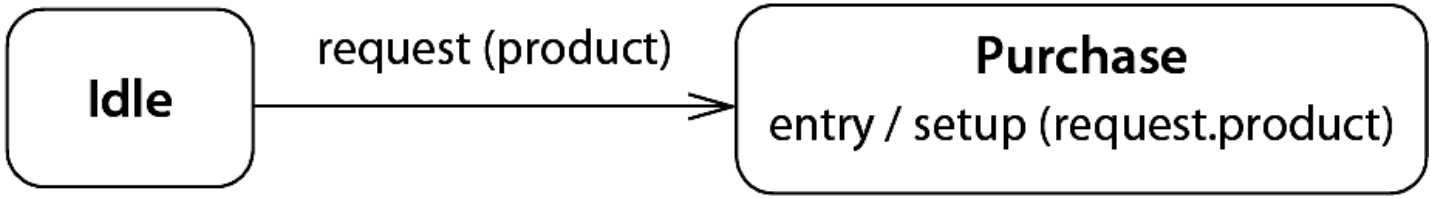
Diagramele masinilor de stari

O actiune are **acces la parametrii evenimentului**, precum **si la atributele obiectului**

Exemplu de sintaxa pentru

- **utilizarea unui parametru** (***product***) **al evenimentului** care declanseaza tranzactia (***request***)

- in momentul executiei **actiunii** numita ***setup*** de **tip intrare** (***entry***) **in starea *Purchase***



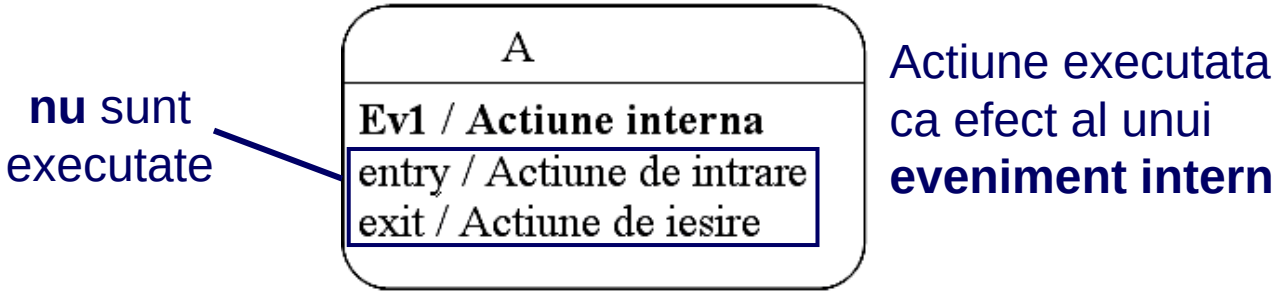
Actiunea executata la aparitia unui eveniment intern

- este executata atunci cand aparitia acestuia **nu conduce la o alta stare**

Diagramele masinilor de stari

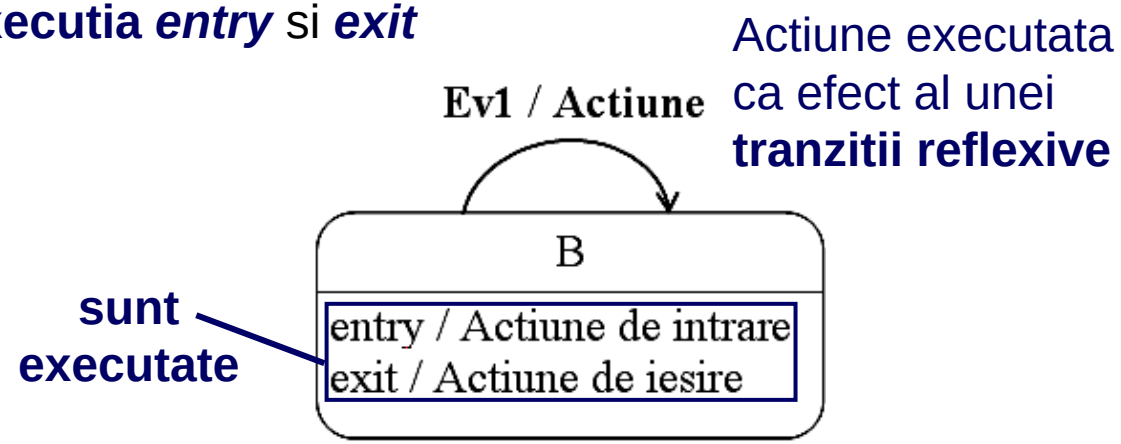
Evenimentul intern

- **nu antreneaza executia actiunilor *entry*** si ***exit***



Tranzitia reflexiva in schimb

- **antreneaza executia *entry*** si ***exit***

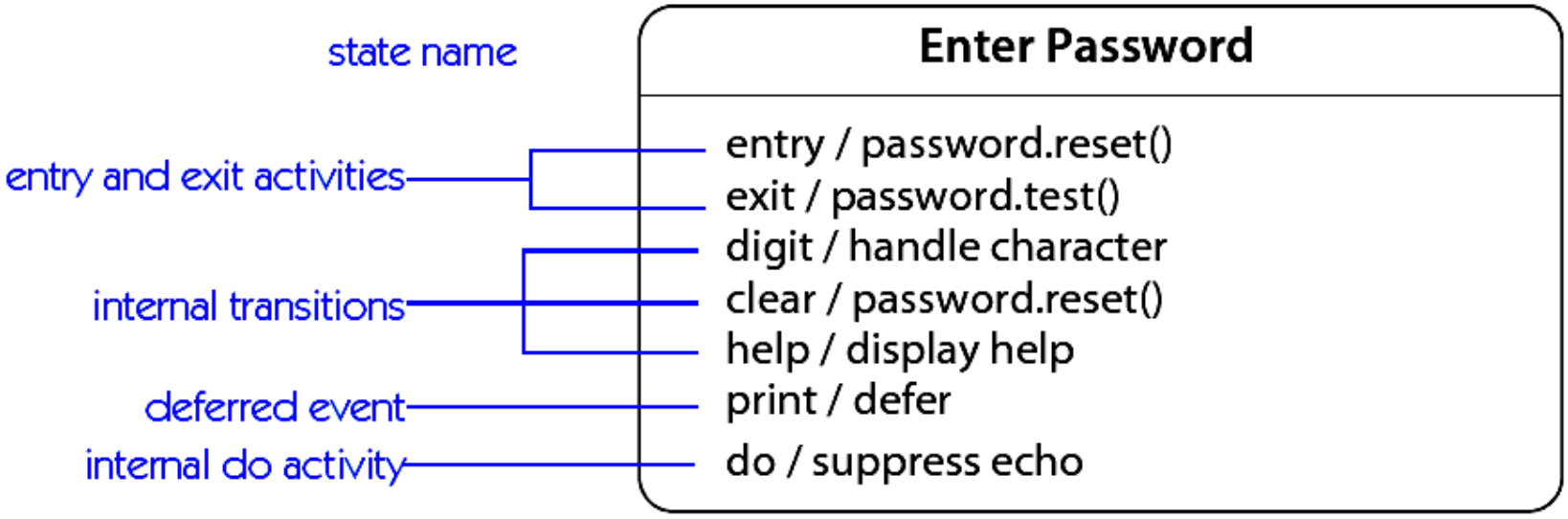


Diagramele masinilor de stari

Activitatea in UML este

- o **operatie** care are o **durata semnificativa** (care nu poate fi neglijata) si care este **executata cat timp obiectul este in starea data**

- marcata prin cuvantul cheie ***do***

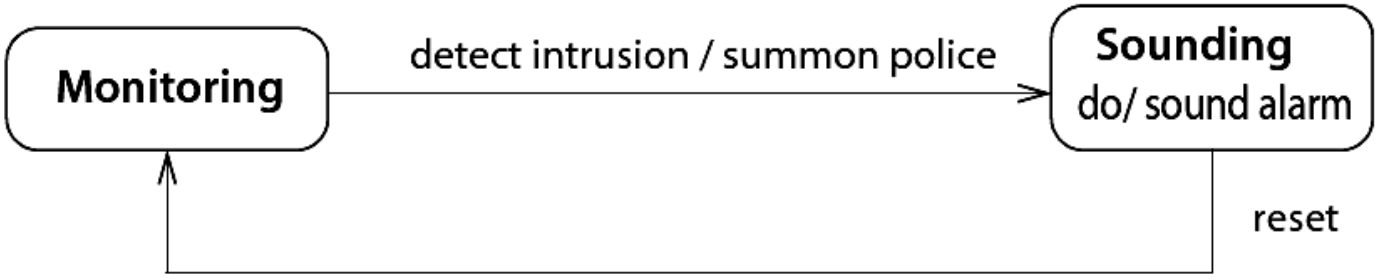


Diagramele masinilor de stari

Spre deosebire de actiuni, **activitatile**

- **pot fi intrerupte** in orice moment, **atunci cand este declansata** o **tranzitie de iesire din stare**

- de exemplu, evenimentul ***reset*** poate produce intreruperea activitatii ***sound alarm*** atunci cand automatul urmator se afla in starea ***Sounding***



Anumite **activitati** sunt **ciclice**, adica

- nu **se opresc** decat **atunci cand este declansata o tranzitie de iesire** din starea curenta

- de exemplu, activitatea ***alarm*** din automatul de mai sus

Diagramele masinilor de stari

Alte **activitati** sunt **secventiale**, adica

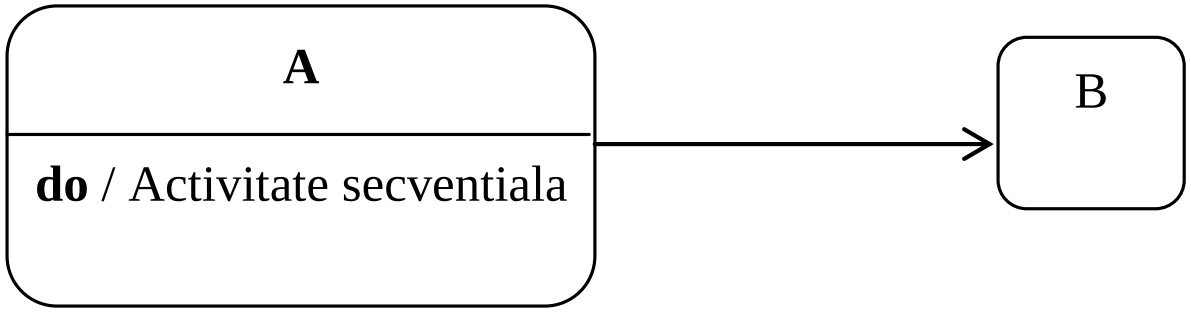
- **starea poate fi parasita si fara aparitia unui eveniment** care sa **declanseze o tranzitie de iesire**,

  - atunci **cand activitatea secventiala** ajunge la **finalul** sau

Acest tip de **tranzitie**, **care nu este declansata de catre un eveniment**

- se numeste **tranzitie automata**

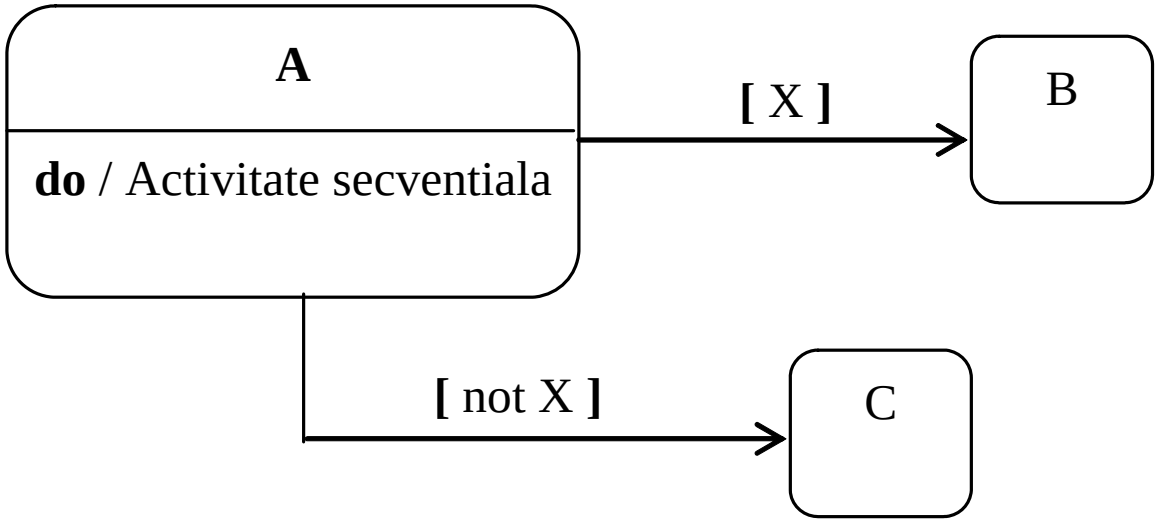
- si are asociat **pseudo-evenimentul *completion***



Diagramele masinilor de stari

O **tranzitie automata** (**care nu este declansata** de catre un eveniment)

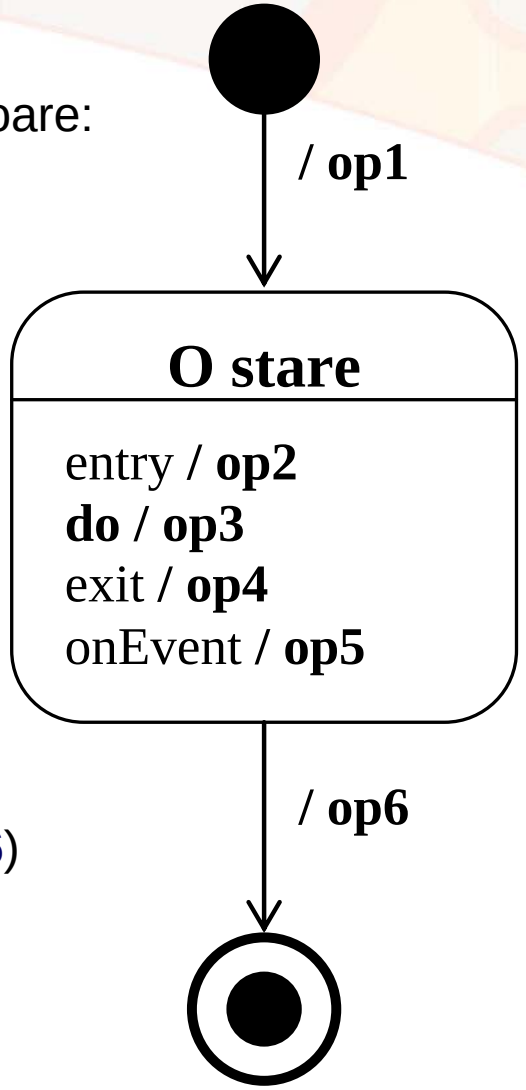
- poate fi insotita de **conditii logice**



Diagramele masinilor de stari

Pentru fiecare stare, exista **6 puncte pentru specificarea operatiilor** care trebuie sa fie executate, in ordinea urmatoare:

- **actiunea** asociata **tranzitiei de intrare** (**op1**)
- **actiunea** de intrare (***entry***) in stare (**op2**)
- **activitatea interna** (***do***) a starii (**op3**)
- **actiunea** de iesire (***exit***) din stare (**op4**)
- **actiunea** asociata **evenimentelor interne** (**op5**)
- **actiunea** asociata **tranzitiei de iesire** din stare (**op6**)



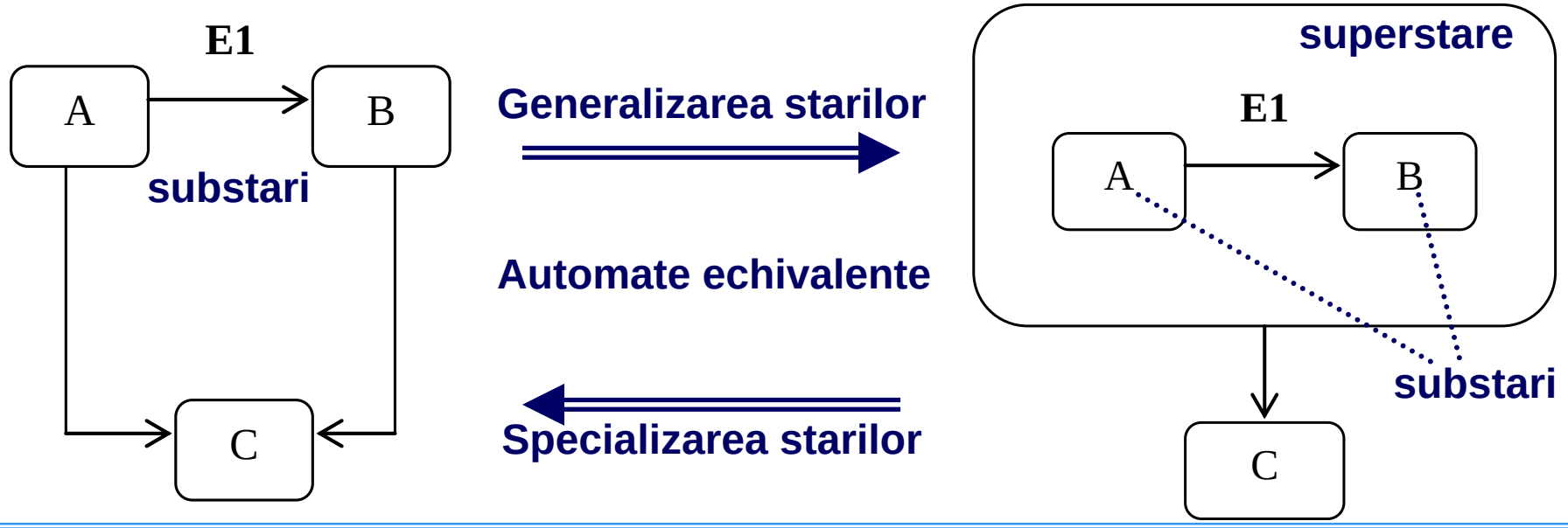
Diagramele masinilor de stari

Diagramele de masini de stari pot deveni destul de **dificil de citit**

- atunci cand **numarul de conexiuni intre stari devine ridicat**

Solutia pentru a rezolva aceasta situatie consta in utilizarea **principiului generalizarii / specializarii starilor**:

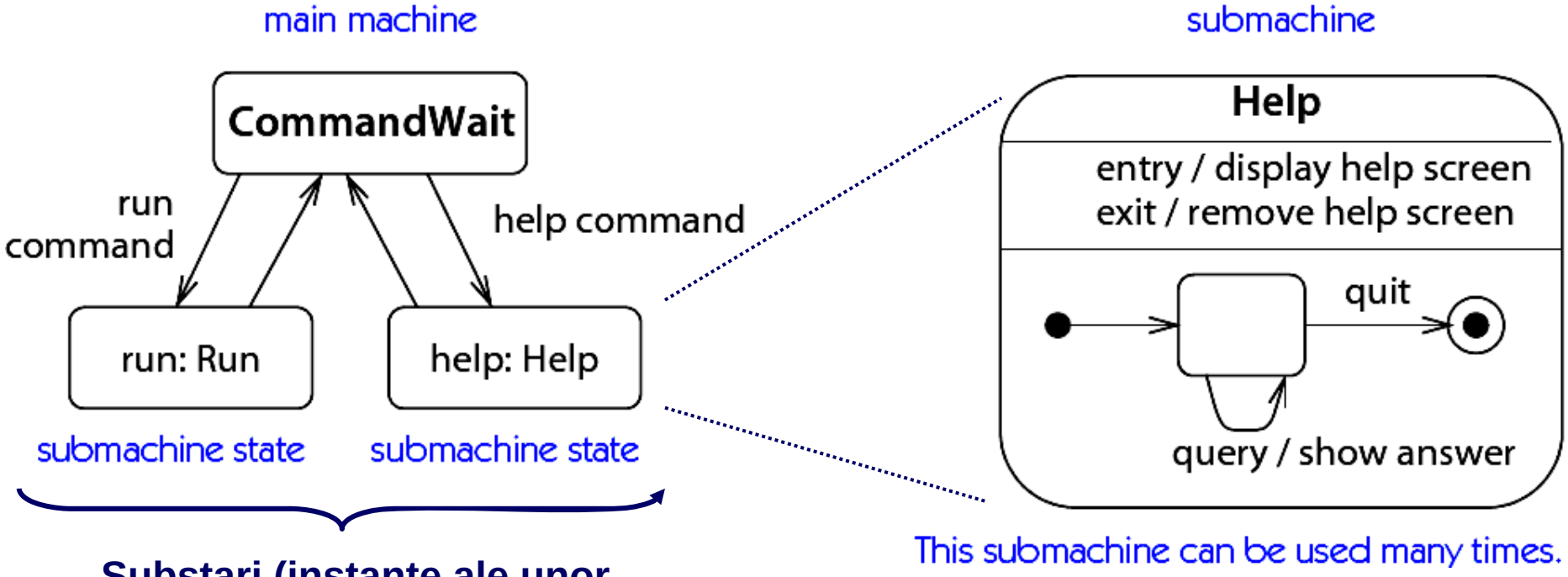
- starile **mai generale** poarta numele de **superstari**
- iar starile **mai specifice** sunt denumite **substari**



Diagramele masinilor de stari

O stare poate fi **descompusa**

- in mai multe **substari disjuncte / succesive**



Substari (instante ale unor submasini de stari)

Substarea (submasina de stari) este reutilizabila

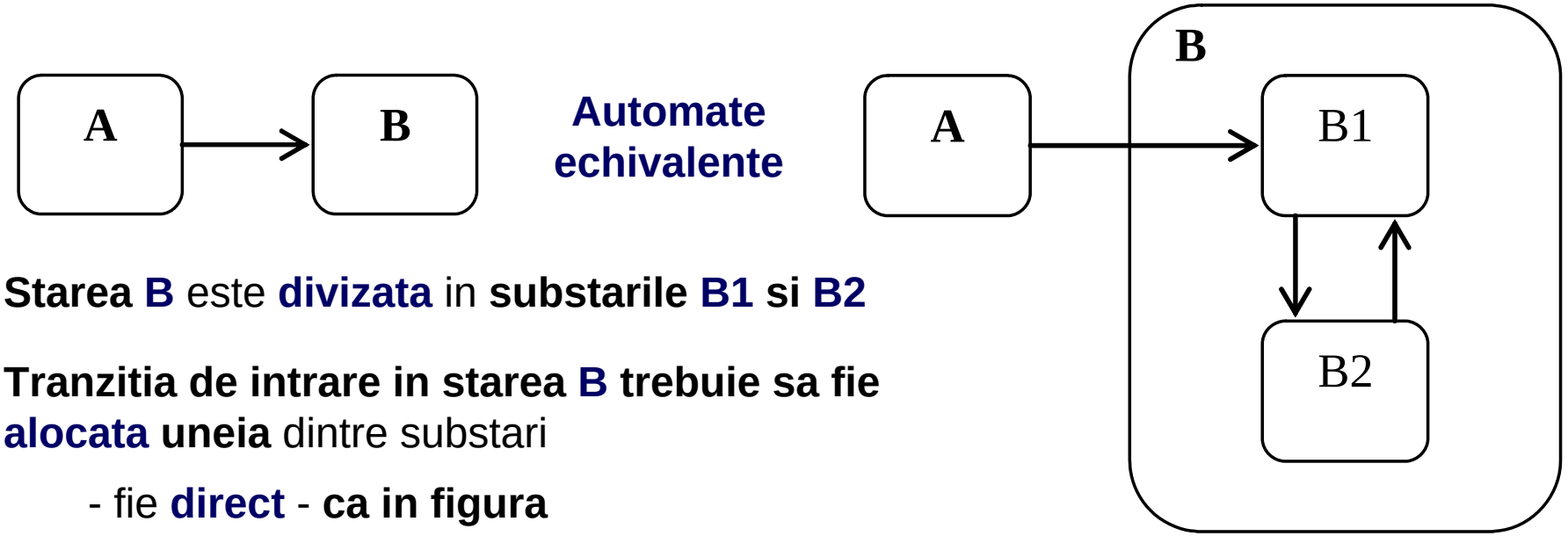
Diagramele masinilor de stari

Substarile mostenesc caracteristicile **superstarilor** lor

- in particular **variabilele de stare** si **tranzitiile externe**

Descompunerea in substari este denumita si descompunere **disjunctiva** (**XOR**)

- adica **obiectul** trebuie sa se afle **intr-o singura substare la un moment dat**



Starea B este **divizata** in **substarile B1 si B2**

Tranzitia de intrare in starea B trebuie sa fie alocata uneia dintre substari

- fie **direct** - **ca in figura**
- fie **indirect** - **prin intermediul unei stari initiale**

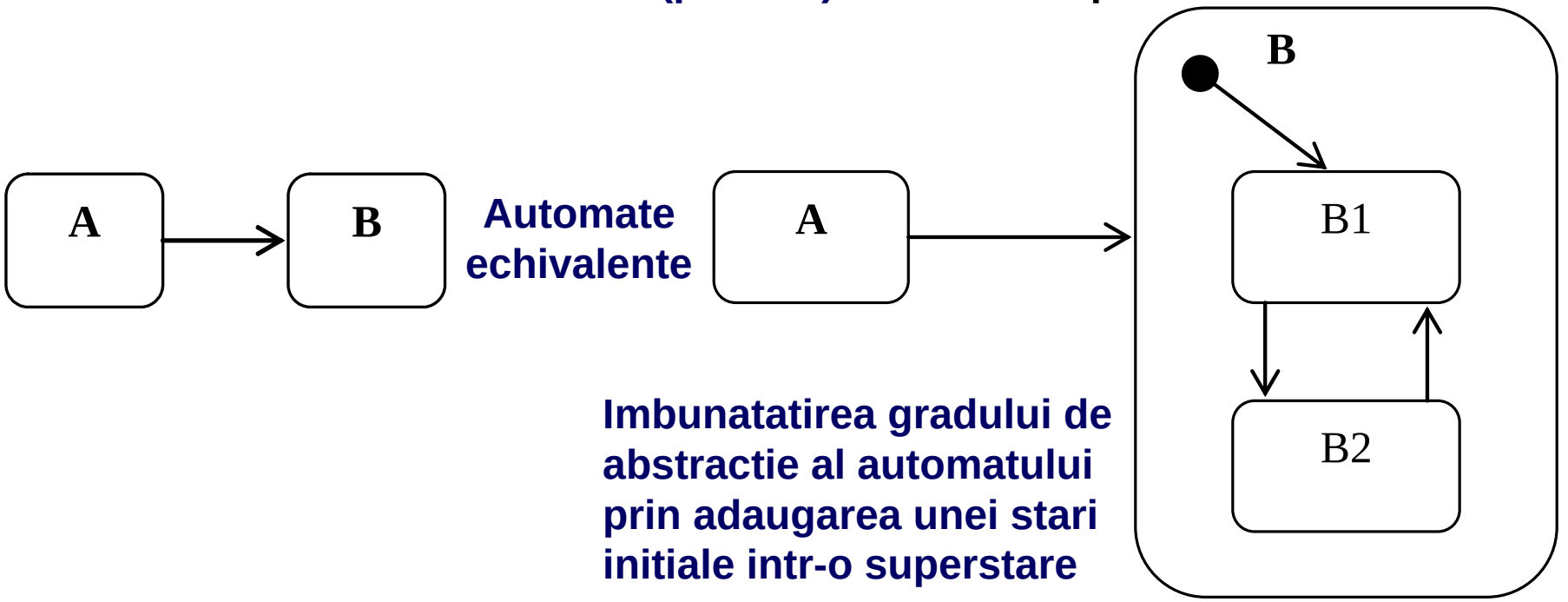
Diagramele masinilor de stari

Descompunerea in forma anterioara pierde din abstractizare

- ca orice **mecanism** scris in termenii superclasei **care are nevoie sa cunoasca detaliile subclaselor**

Este preferabila limitarea legaturilor intre niveluri ierarhice ale unui automat

- **definind sistematic cate o (pseudo)stare initiala pentru fiecare nivel**

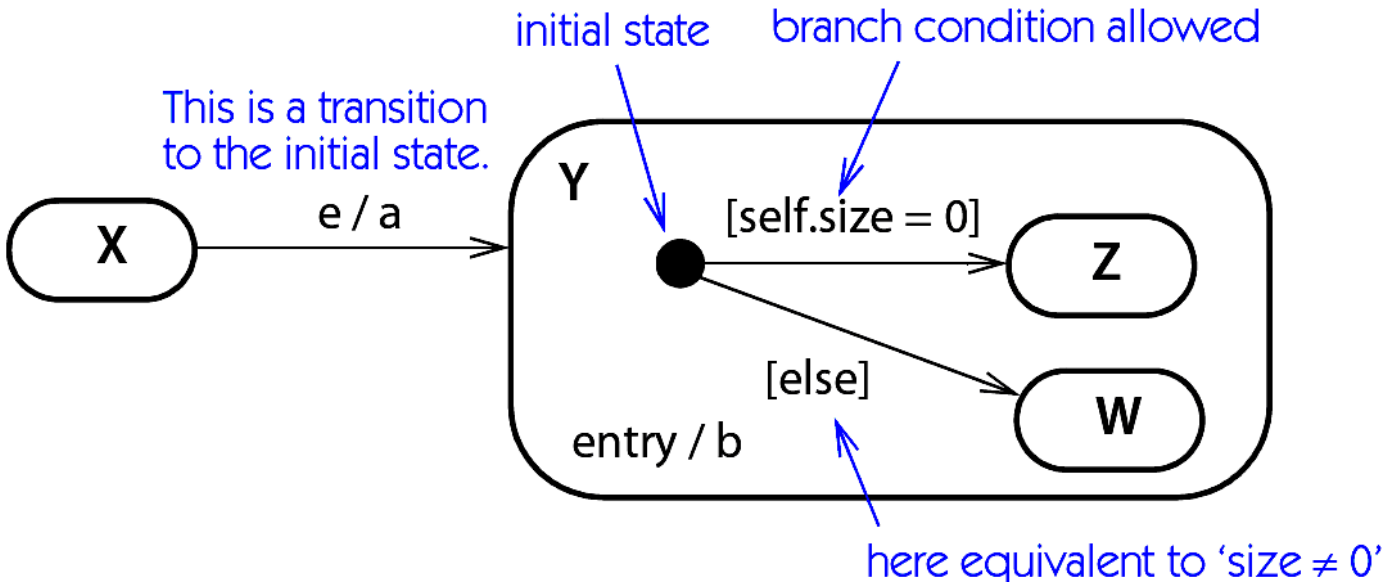


Imbunatatirea gradului de abstractie al automatului prin adaugarea unei stari initiale intr-o superstare

Diagramele masinilor de stari

Exemplu de

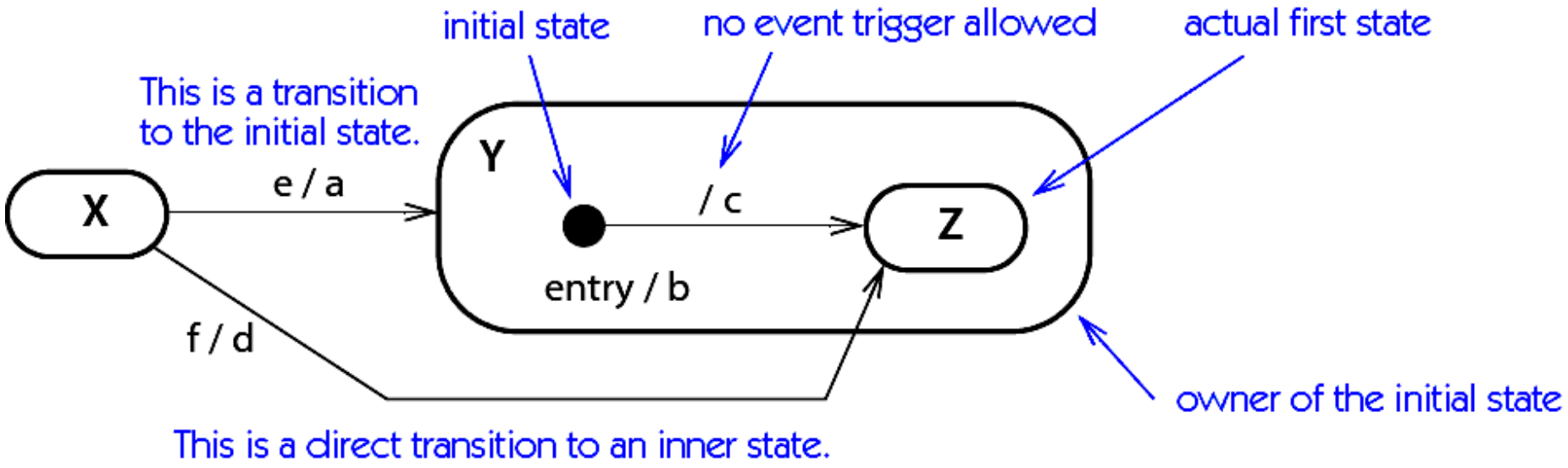
- **definire a unei (pseudo)substari initiale**
- urmata de **tranzitii automate conditionate**
- si de **utilizare a cuvantului cheie *else*** in conditia logica



Diagramele masinilor de stari

Exemplu de

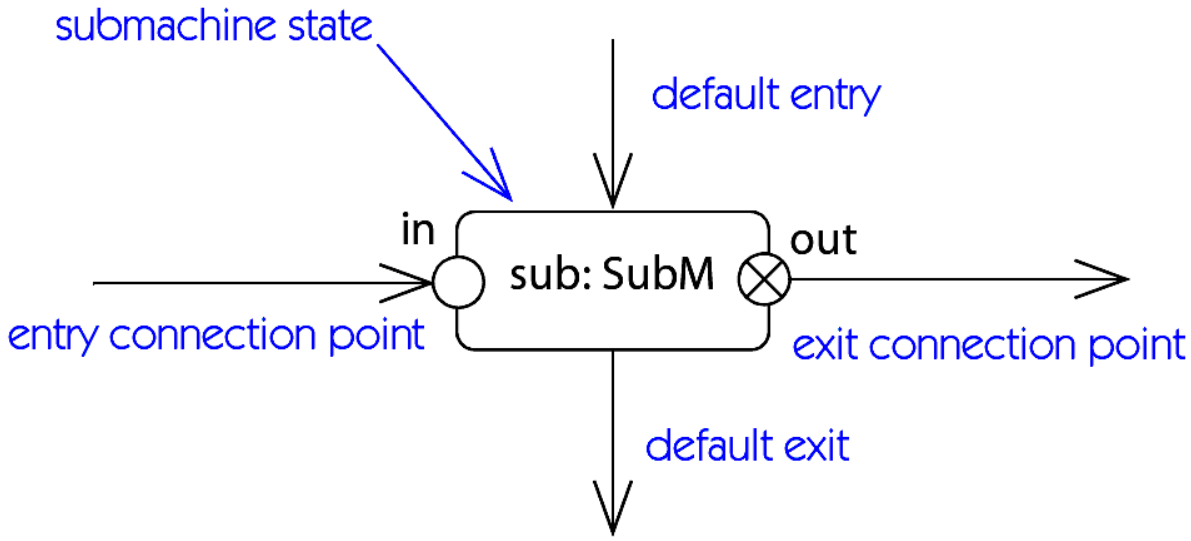
- **definire a unei (pseudo)substari initiale**
- urmata de **tranzitie automata**
- dar si de **tranzitie directa intr-o stare interna**



Diagramele masinilor de stari

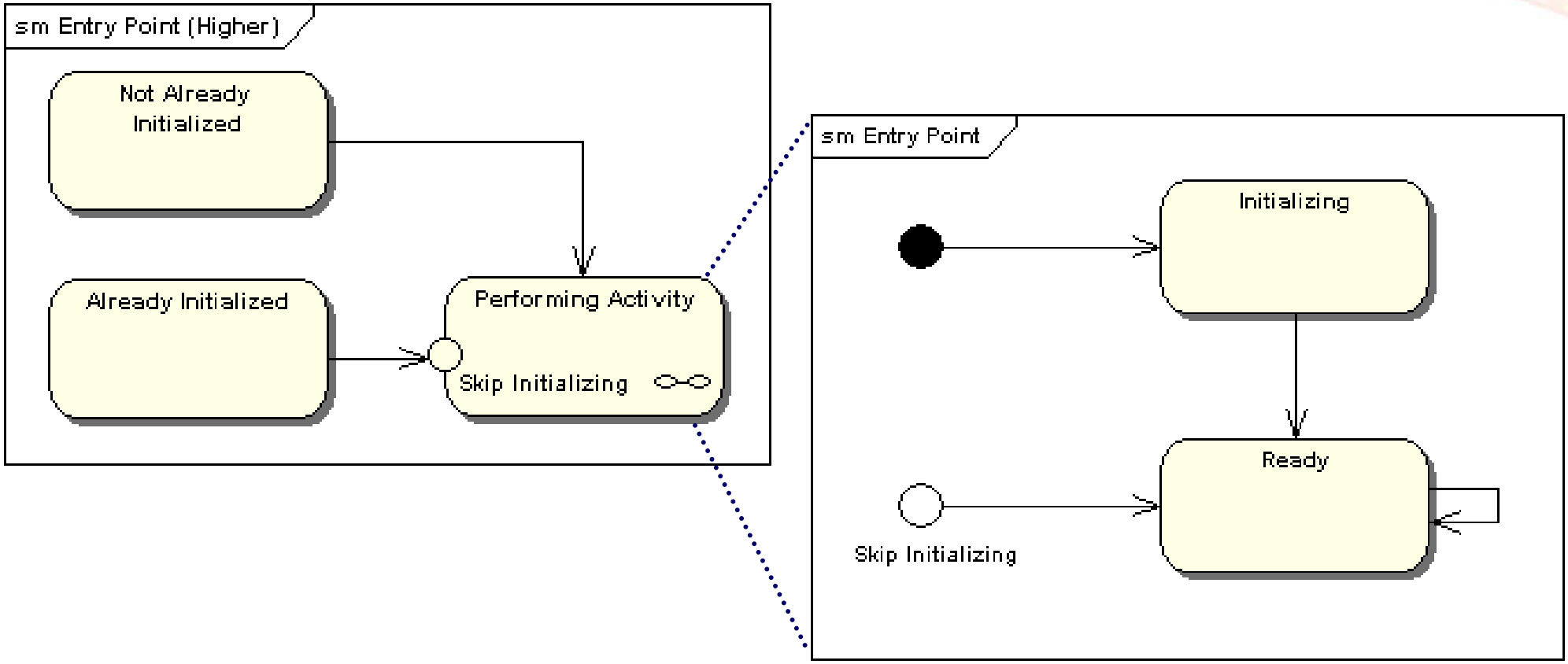
Punctele de conectare permit

- **marcarea explicita** in diagrama masinii de stari **a intrarii/iesirii intr-o substare (submasina de stari)**



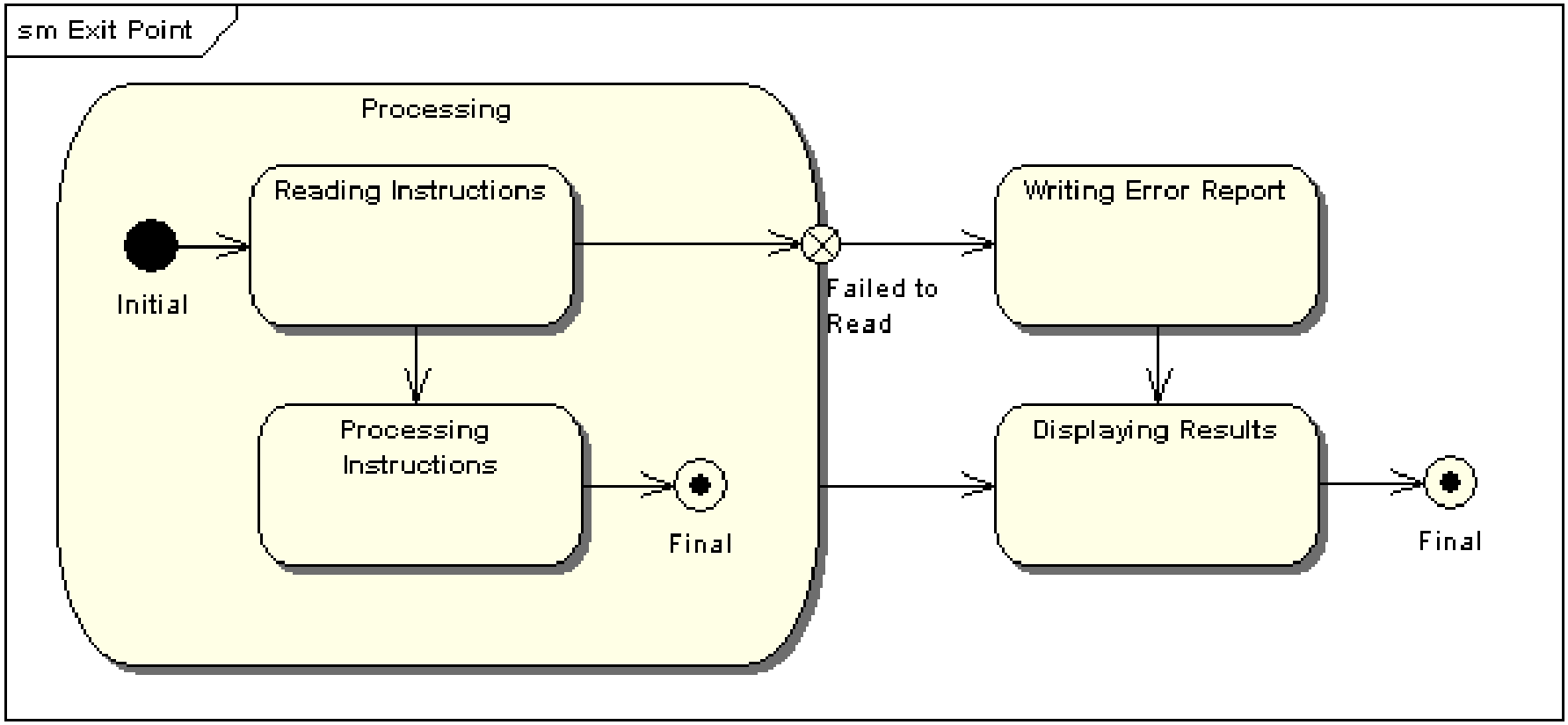
Diagramele masinilor de stari

Exemplu de **utilizare a punctelor de conectare** pentru a **marca explicit intrarile distincte intr-o substare (submasina de stari)**



Diagramele masinilor de stari

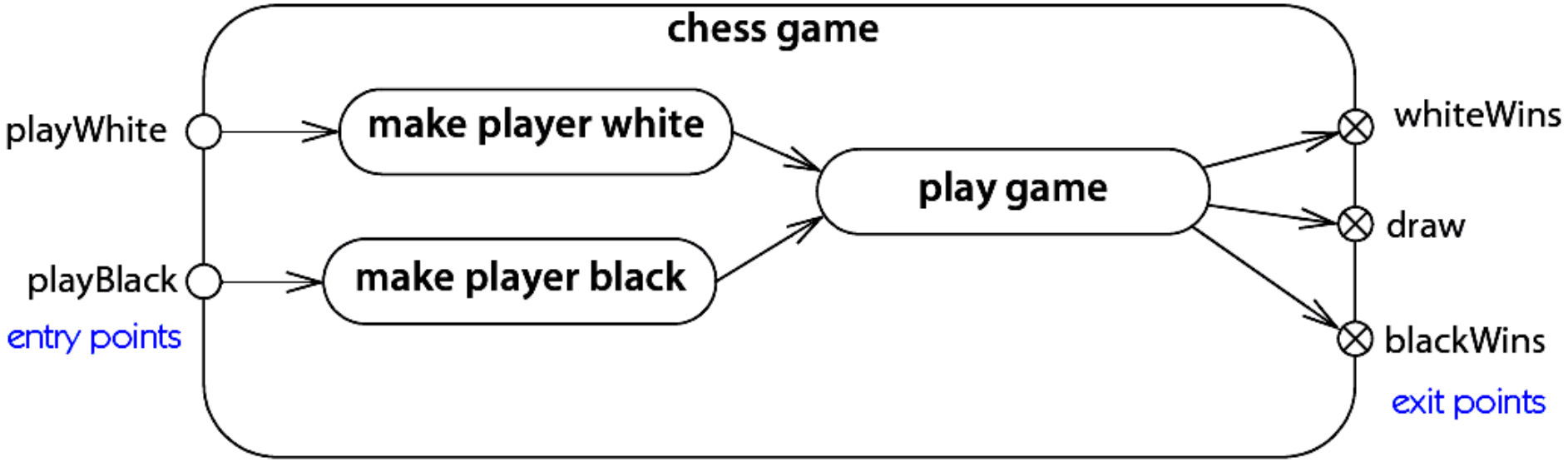
Exemplu de **utilizare a punctelor de conectare** pentru a **marca explicit iesirile distincte intr-o substare (submasina de stari)**



Diagramele masinilor de stari

Exemplu de

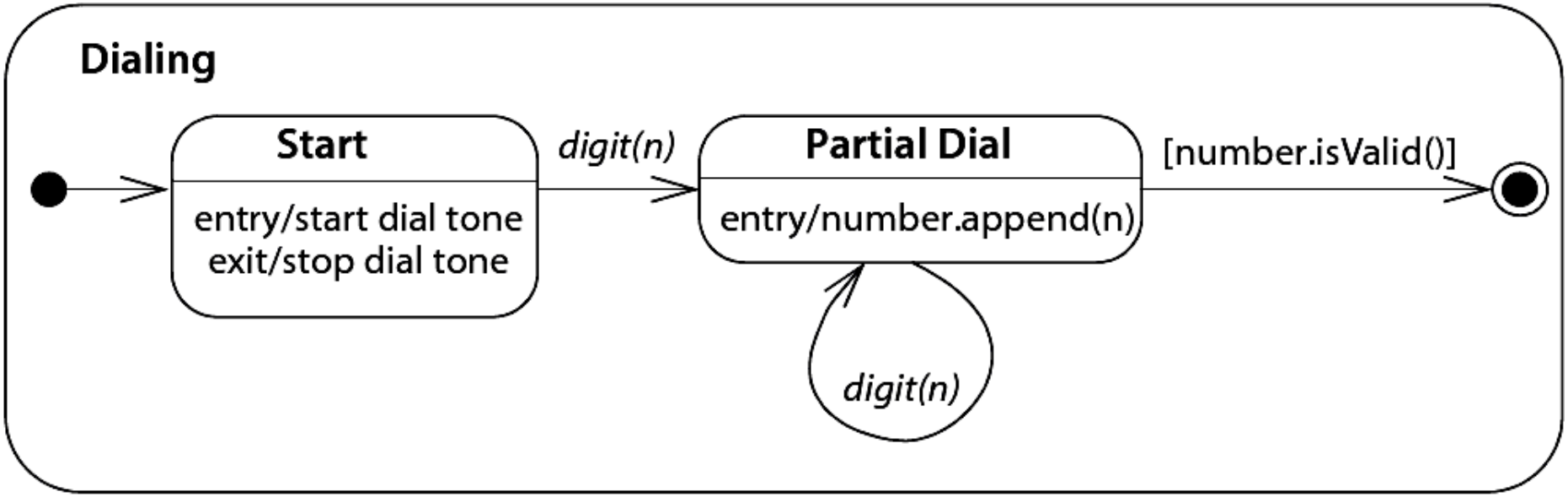
- **utilizare a punctelor de conectare** pentru a **marca explicit**
  - substarile **asociate intrariilor in/iesirilor dintr-o submasina de stari**



Diagramele masinilor de stari

Exemplu de

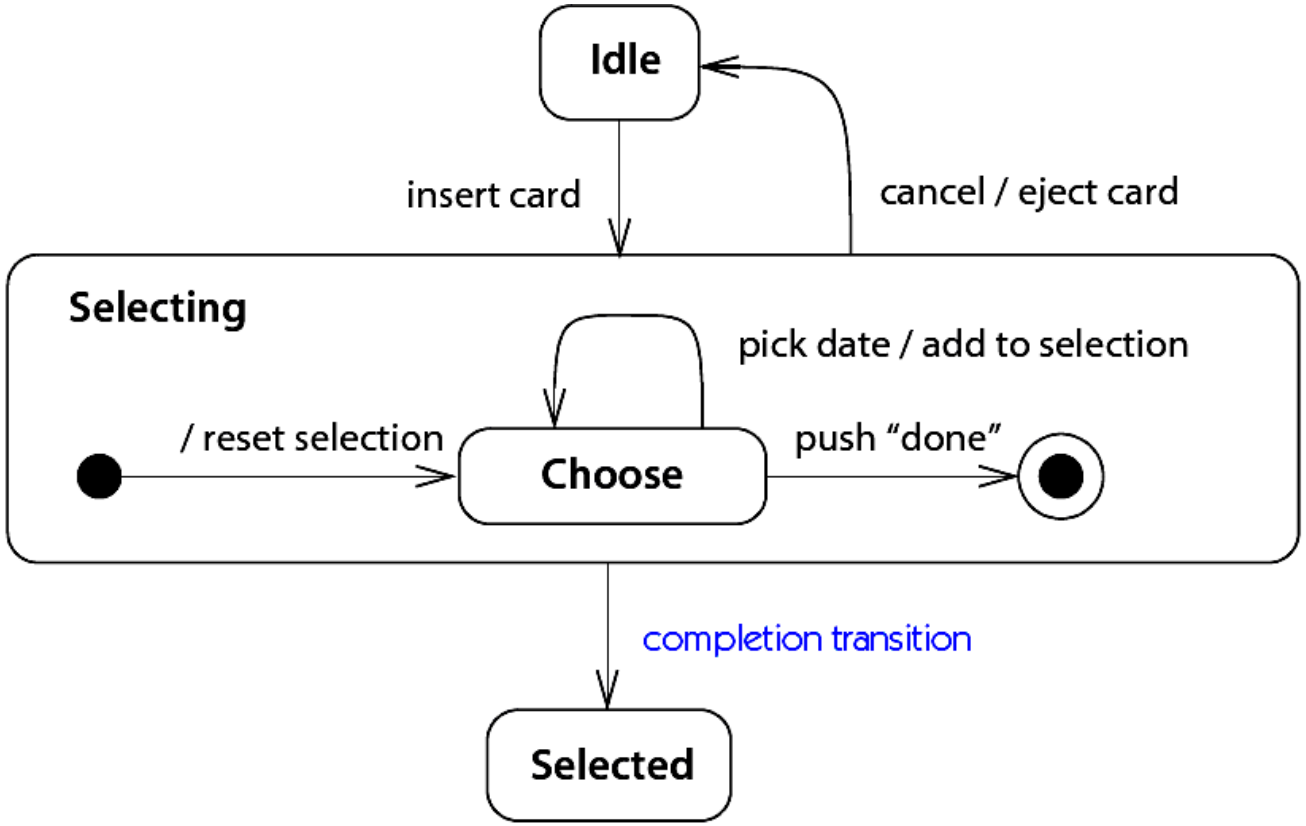
- **definire a** unei **pseudostari finale in interiorul** unei **submasini de stari**



Diagramele masinilor de stari

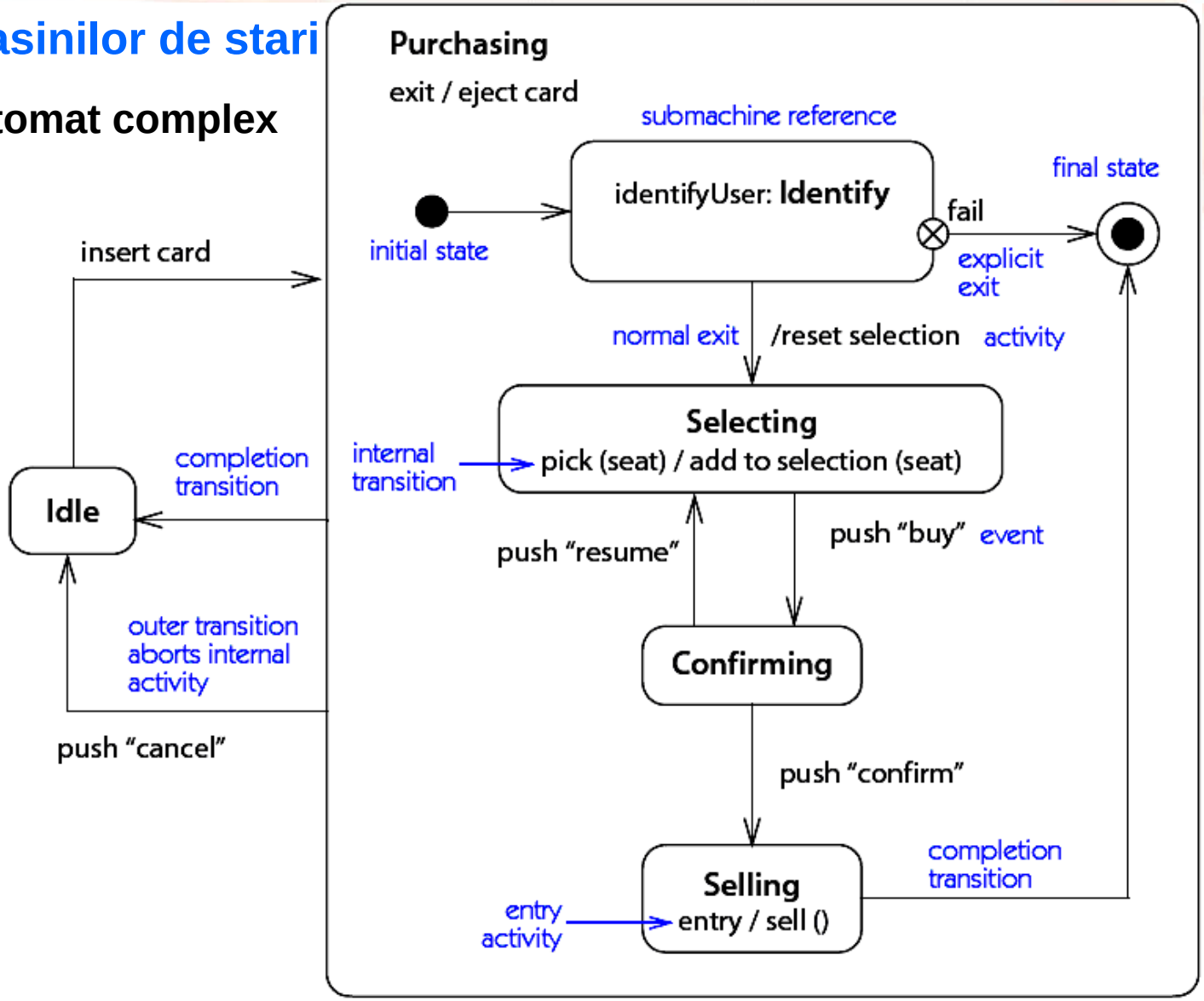
Exemplu de

- **tranzitie automata (*completion*) la iesirea dintr-o submasina de stari**



Diagramele masinilor de stari

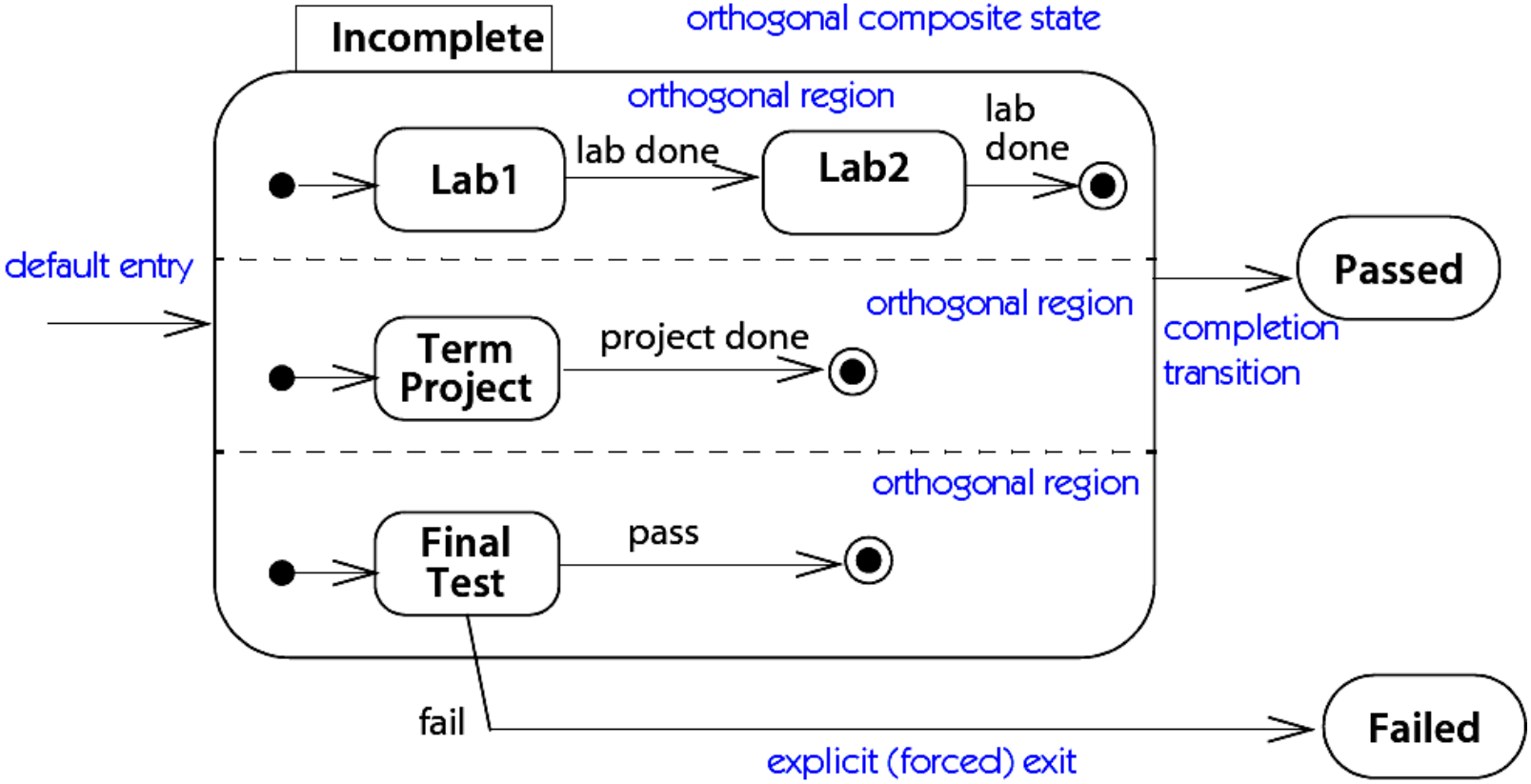
Exemplu de automat complex



Diagramele masinilor de stari

O stare poate fi **descompusa** si

- in mai multe **regiuni ortogonale / concurente**



Diagramele masinilor de stari

Descompunerea in regiuni ortogonale este denumita si descompunere **conjunctiva** (**AND, ortogonala**)

- adica **obiectul** trebuie sa se afle **simultan in cate o stare din fiecare dintre automatele din regiunile ortogonale**

- conjunctia starilor reprezentand o **forma de paralelism intre automate**

Automatele din **regiunile ortogonale**

- sunt in general **independente intre ele**

Agregarea starilor este **operatia inversa** prin care

- pornind **de la automate independente** (care devin regiuni ortogonale)

  - se formeaza **stari compuse conjunctiv** (**ortogonal**)

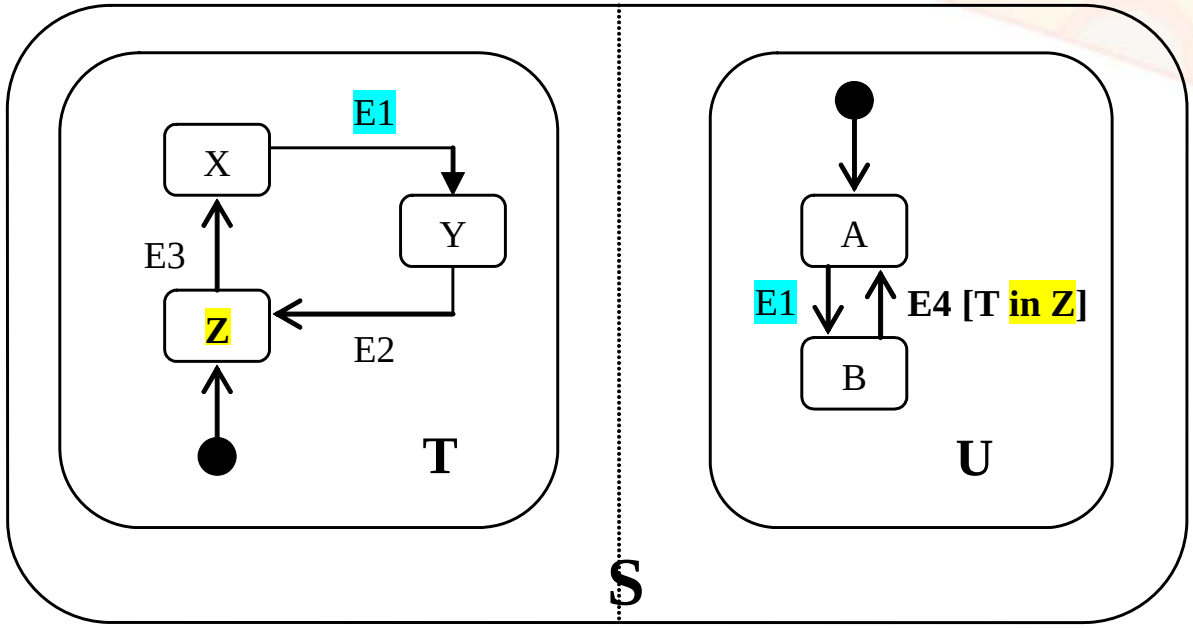
Diagramele masinilor de stari

Exemplu care ilustreaza diferite **aspecte ale notiunii de agregare a starilor**

Starea S e formata din doua **automate ortogonale T si U**

- **T** este compus din substarile **X, Y si Z**
- **U** este compus din substarile **A si B**

S este **produsul cartezian** al starilor T si U



Tranzitia de intrare in starea S implica

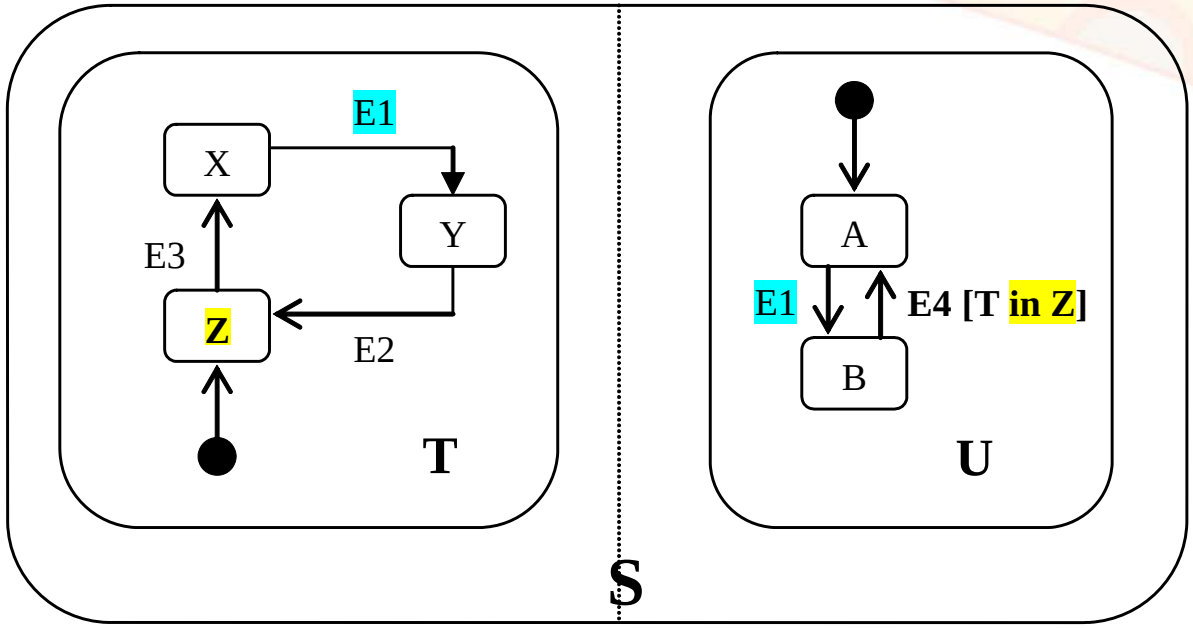
- **activarea simultana** a automatelor **T si U**, adica obiectul este plasat in **starea compusa (Z, A)**

Diagramele masinilor de stari

Exemplu care ilustreaza diferite **aspecte ale notiunii de agregare a starilor**

Cand are loc un eveniment **E3**

- automatele **T si U** pot evolua **independent**
- ceea ce conduce obiectul **din starea (Z, A) in starea (X, A)**
- adica U ramane in substarea A



Automatele **T si U pot evolua** si **simultan**

- ceea ce este cazul atunci **cand** evenimentul **E1**
  - conduce obiectul **din starea compusa (X, A) in starea compusa (Y, B)**

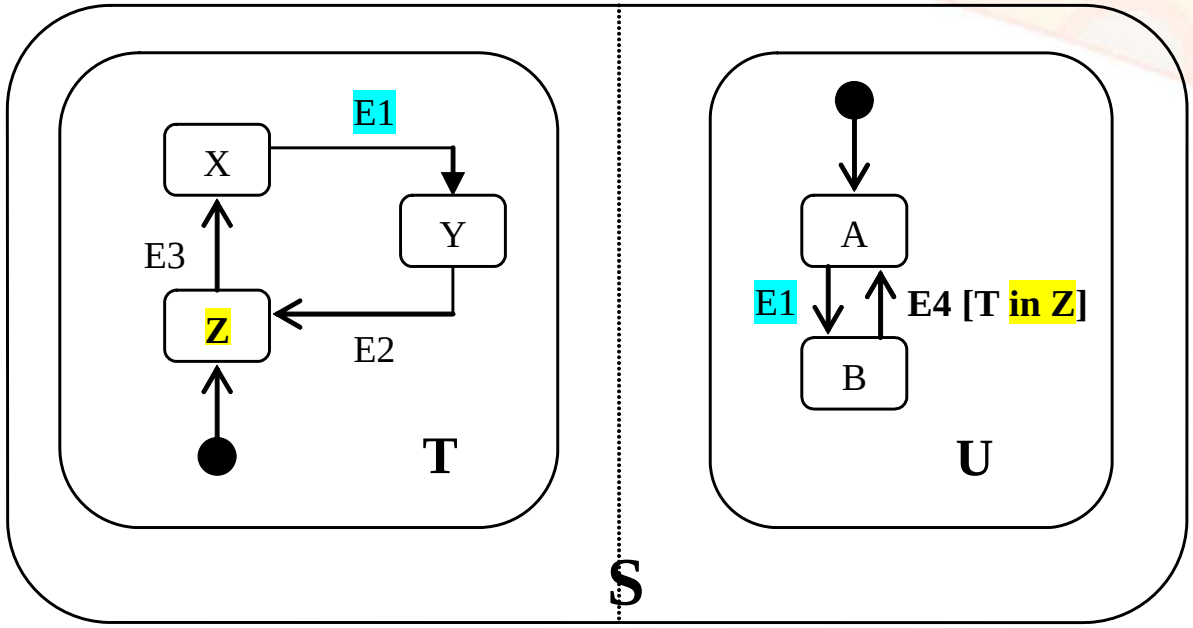
Diagramele masinilor de stari

Exemplu care ilustreaza diferite **aspecte ale notiunii de agregare a starilor**

Adaugarea conditiilor pe tranzitii

- cum este conditia logica **[in Z]** plasata **pe tranzitia de la B la A**

- permite **introducerea relatiilor de dependenta** (temporara si izolata) **intre componentele agregatului**



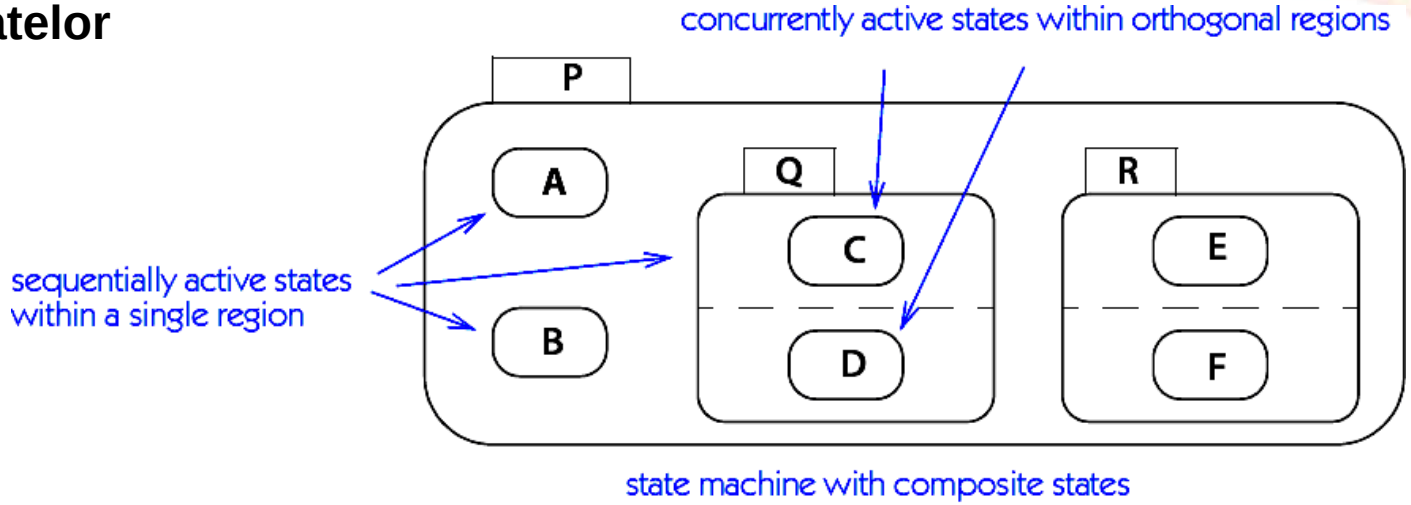
Astfel, **cand are loc** evenimentul **E4**

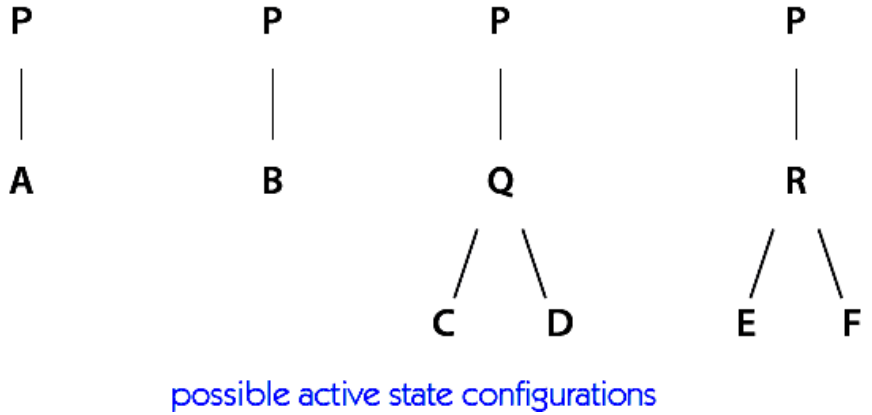
- **tranzitia din B in A** este **validata**

  - **doar daca** obiectul **este in acel moment si in starea Z**

Diagramele masinilor de stari

Agregarea si **generalizarea** starilor permit **simplificarea reprezentarii automatelor**





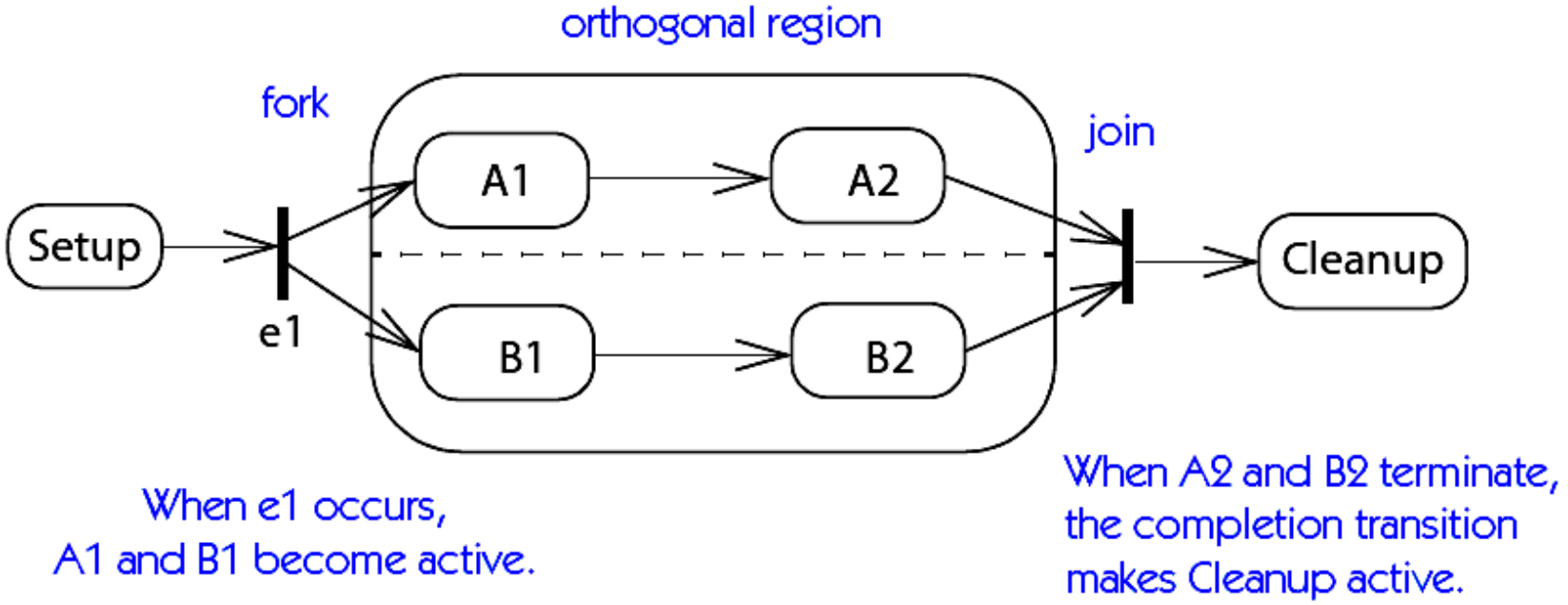
Diagramele masinilor de stari

Reprezentarea starilor agregate poate utiliza si

- **bare de sincronizare**

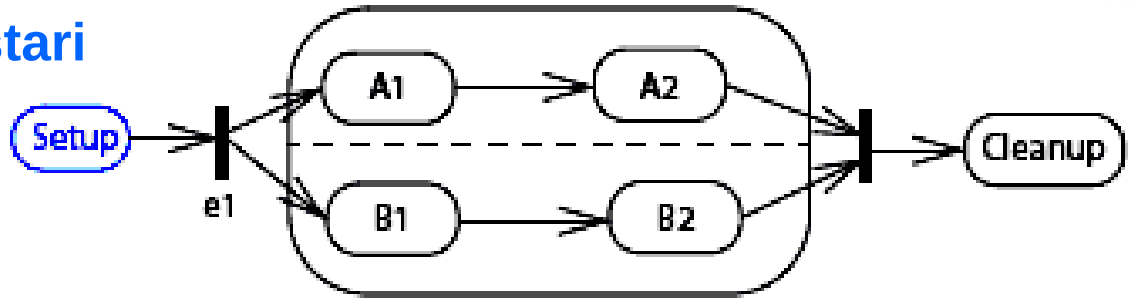
  - **tranzitiile care pleaca** dintr-o bara (***fork***) sunt **declansate simultan**

  - o bara **poate fi depasita doar** atunci **cand toate tranzitiile** care **intra in bara** (***join***) au fost declansate

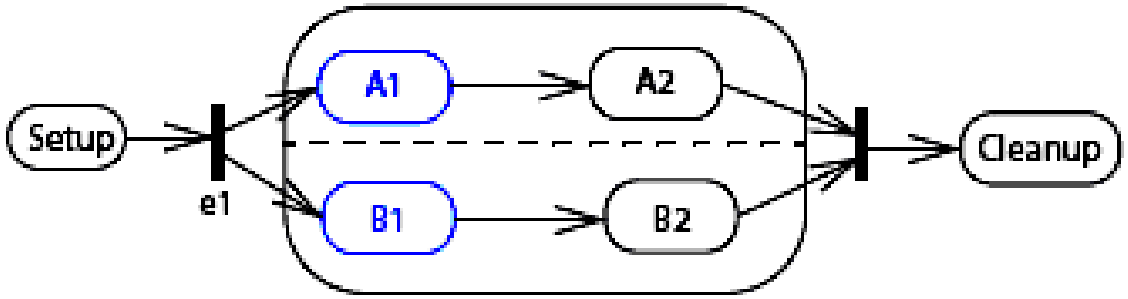


Diagramele masinilor de stari

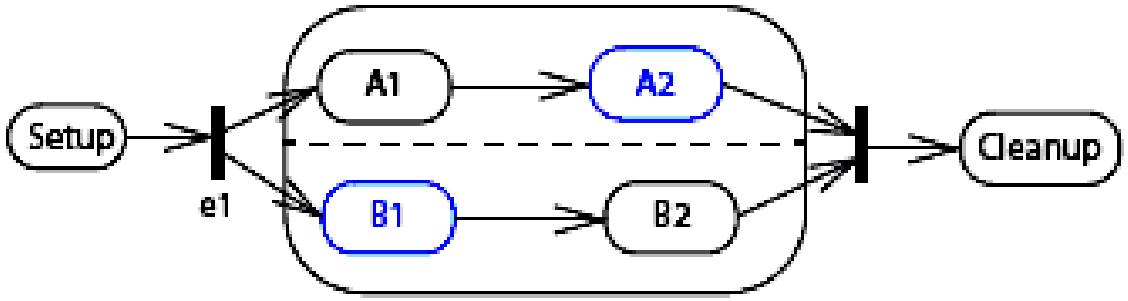
Istoricul starilor active
intr-o **masina de stari**
cu **regiuni ortogonale**



e1 occurs



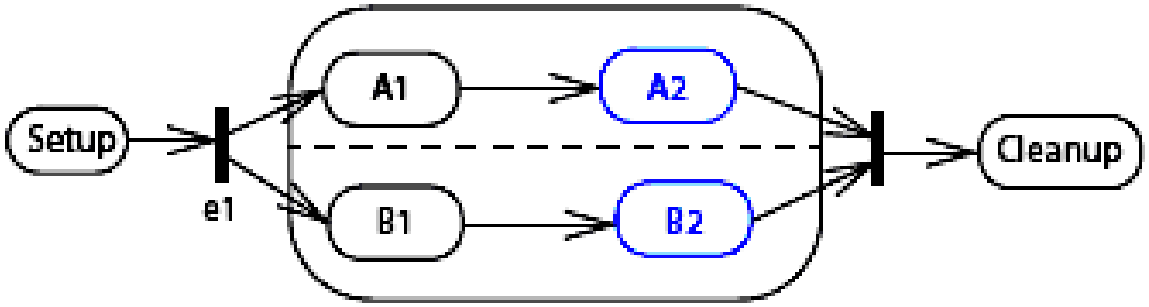
A1 completes



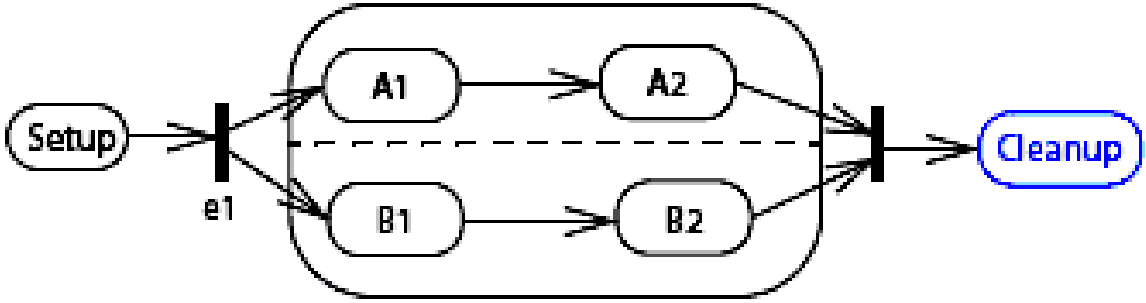
Diagramele masinilor de stari

Istoricul starilor active
intr-o **masina de stari**
cu **regiuni ortogonale**

B1 completes



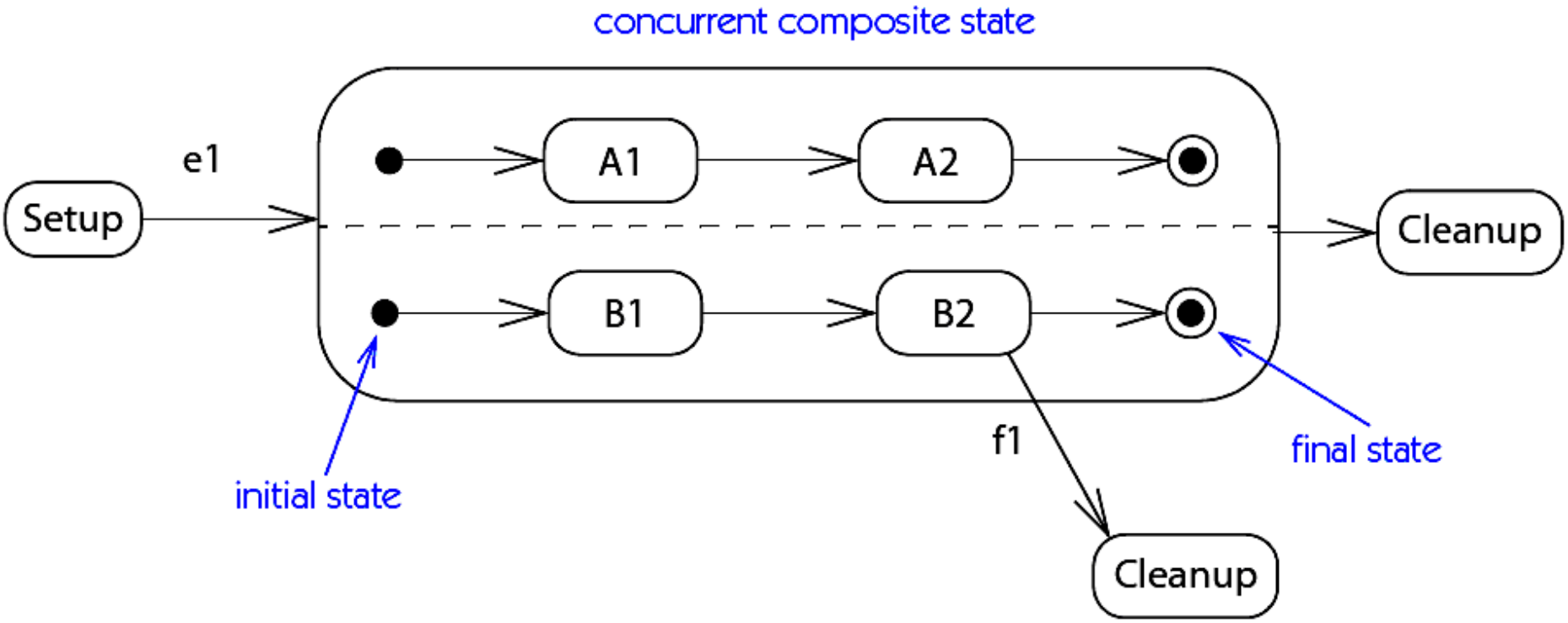
A2 and B2 complete



(Active states shown in blue)

Diagramele masinilor de stari

Reprezentarea acelorasi stari ortogonale utilizand notatile clasice



Diagramele masinilor de stari

Obiectele comunica schimband **mesaje** (apeluri functii, intreruperi *hardware*, etc.)

- care **declanseaza operatii** pentru tratarea lor

Expedierea (trimiterea) unui mesaj intre doua obiecte sunt reprezentate **in FSM**

- ca **trimitere a unui eveniment** intre **automatele claselor obiectelor** respective

Sintaxa unei expedieri de eveniment este:

^Tinta.TrimitereEveniment (Argumente)

unde **Tinta** reprezinta **clasa obiectelor destinatie** a evenimentului

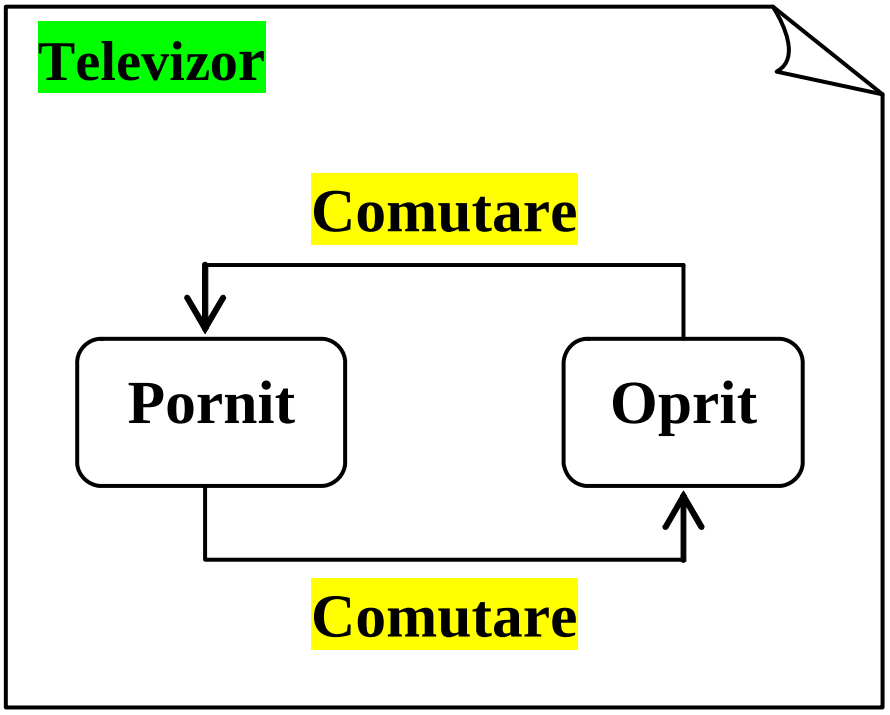
Sintaxa completa a unei tranzitii este:

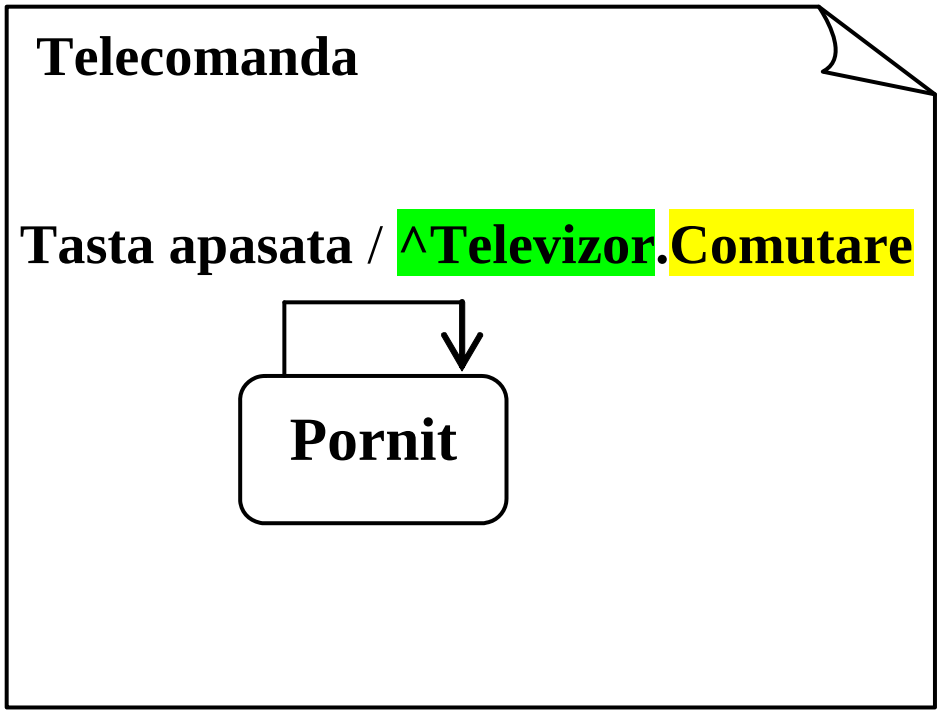
Tranzitie: **PrimireEveniment** (Parametri) [Conditii] / **^Tinta.TrimitereEveniment** (Argumente)

Diagramele masinilor de stari

Exemplul urmator arata fragmente ale

- **automatului** unui **televizor** si
- **automatului telecomenzii sale**





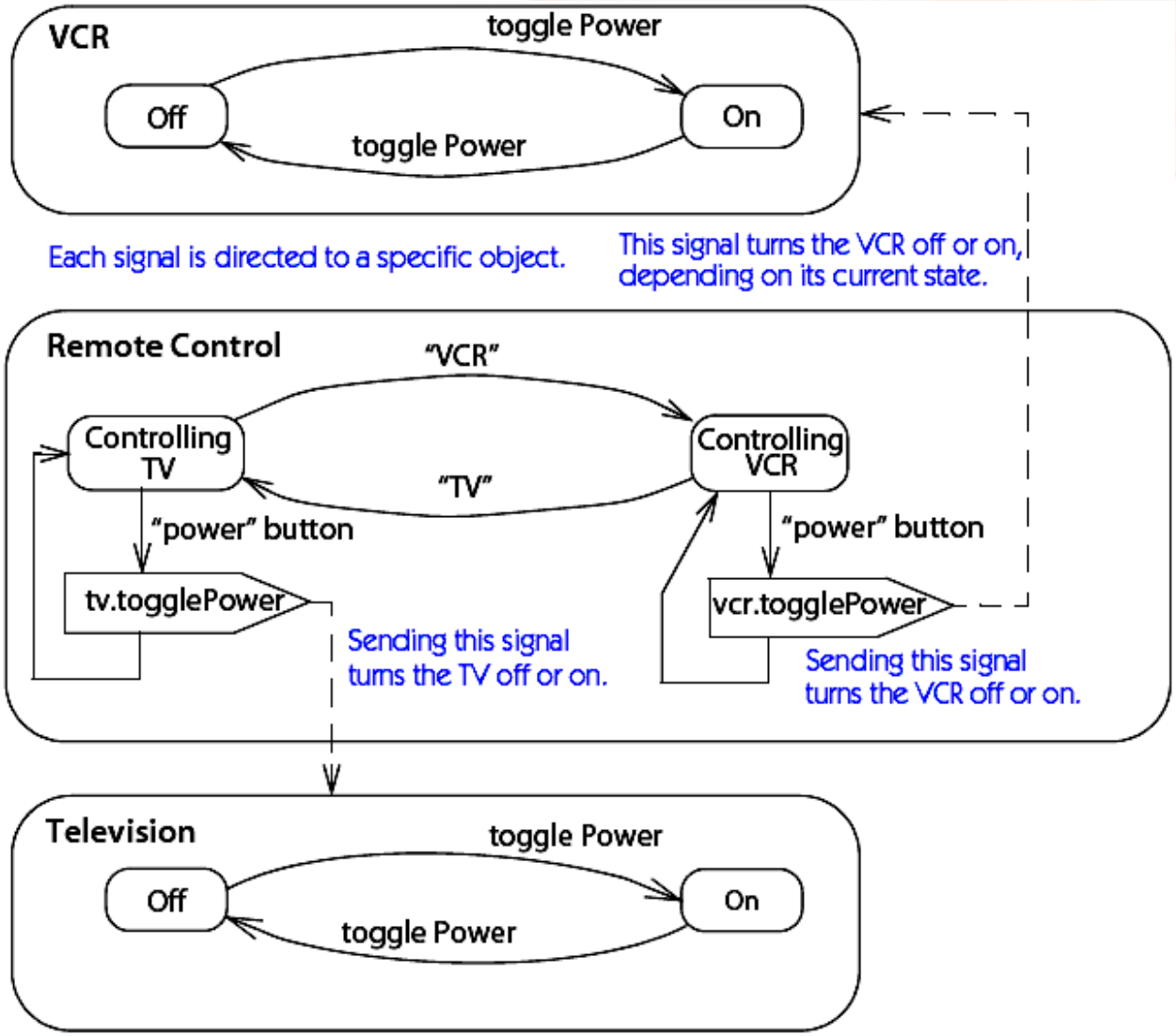
Diagramele masinilor de stari

Exemplu de **comunicatie intre automate**

- **automatului** unui *video player* **(VCR)**

- **automatului** unei **telecomenzi “universale”**

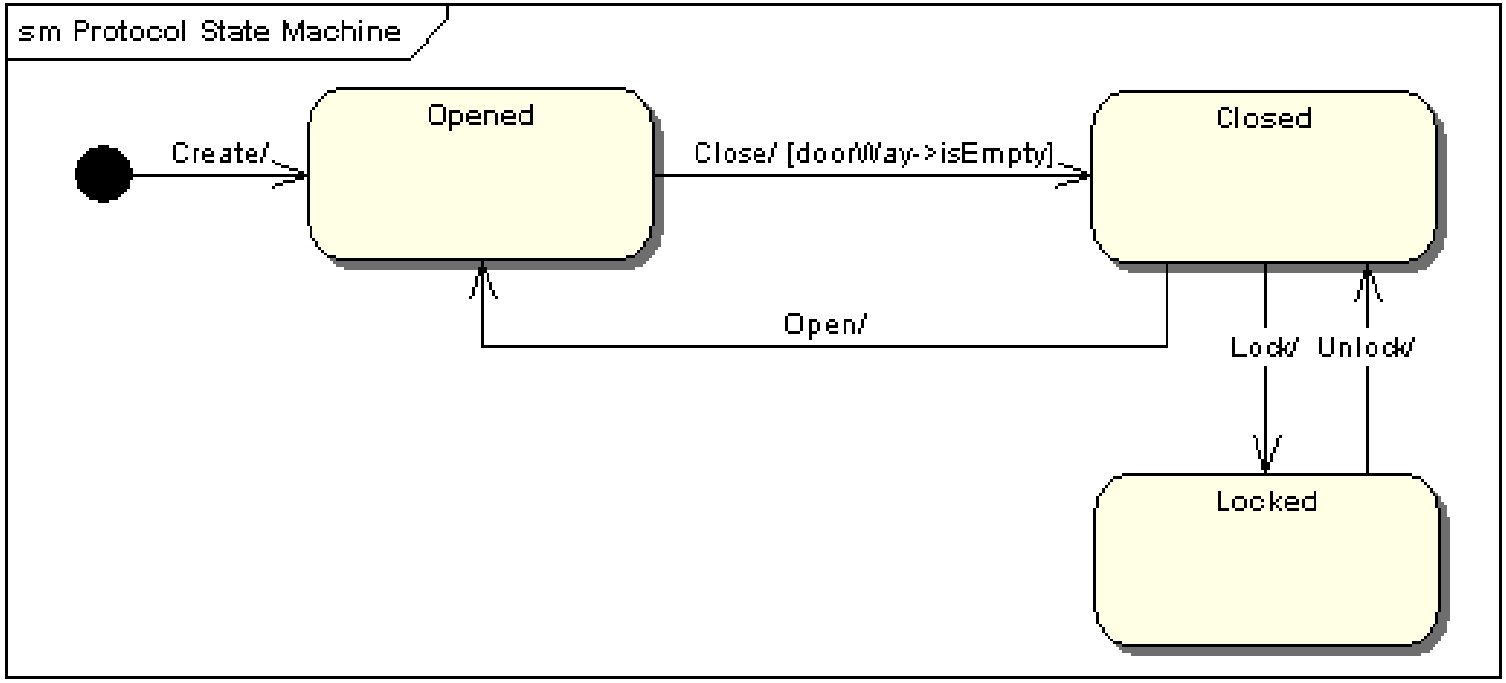
- **automatului** unui **televizor**



Diagramele masinilor de stari

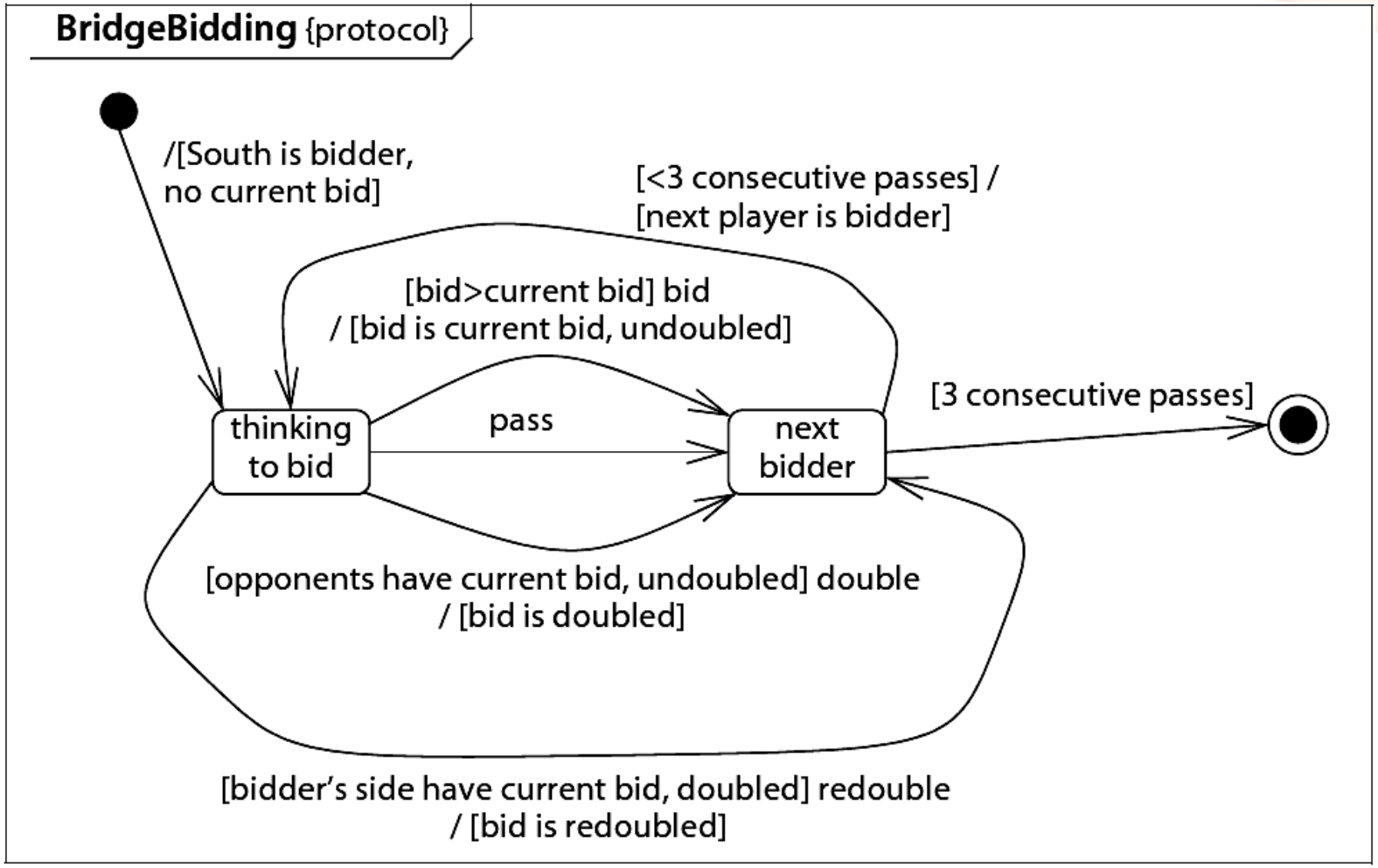
In UML 2

- **masina de stari obisnuita** este numita **comportamentala** si
  - exprima **comportamentul** unui **obiect / sistem**
- **masina de stari protocol**
  - specifica **secventa legala de apeluri si semnale primite de un obiect**



Diagramele masinilor de stari

Exemplu de masina de stari protocol



Diagramele masinilor de stari

Tranzitiile pot avea

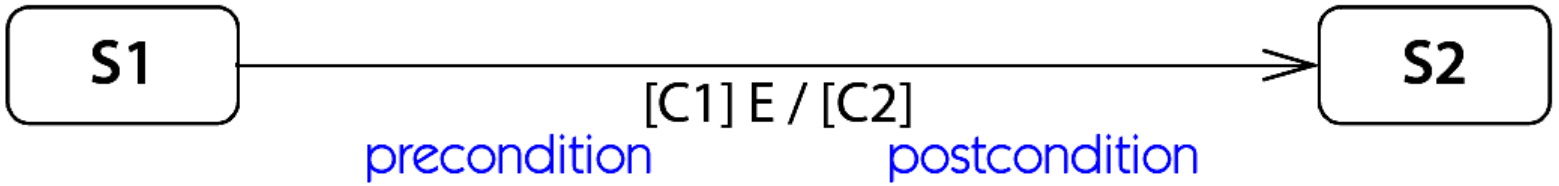
- **pre-conditii** = conditii logice **obisnuite**, care **valideaza tranzitia inaintea aparitiei** evenimentului

- **post-conditii** = conditii logice care **valideaza tranzitia dupa aparitia evenimentului**

Sintaxa pre- si post-conditiilor

[preConditie] eveniment / [postConditie]

Exemplu



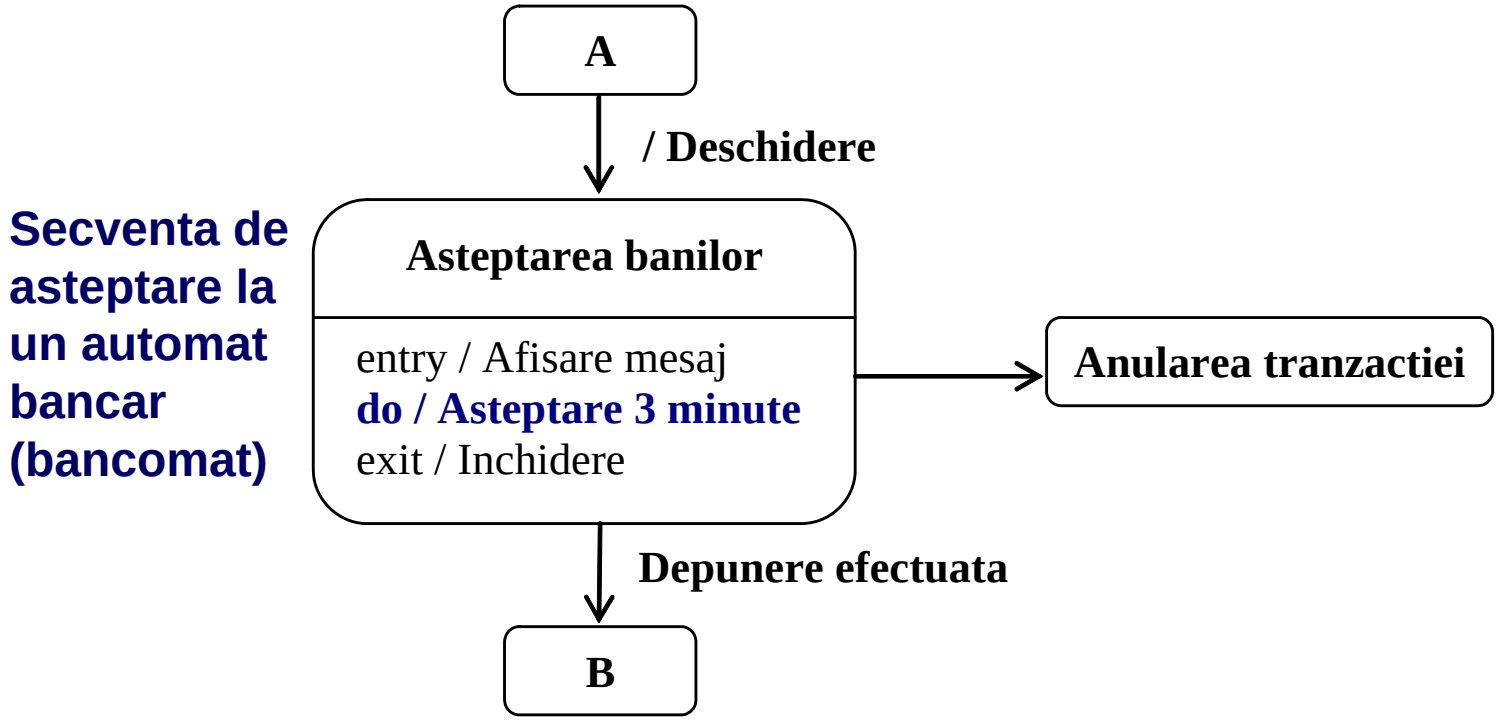
Diagramele masinilor de stari

Asteptarea este definitie o **activitate** care **dureaza un anumit timp** care

- care este **intrerupta** atunci **cand evenimentul asteptat are loc**

  - eveniment care **declanseaza o tranzitie de iesire** din **starea care gazduieste activitatea de asteptare** respectiva

Secventa de asteptare la un automat bancar (bancomat)

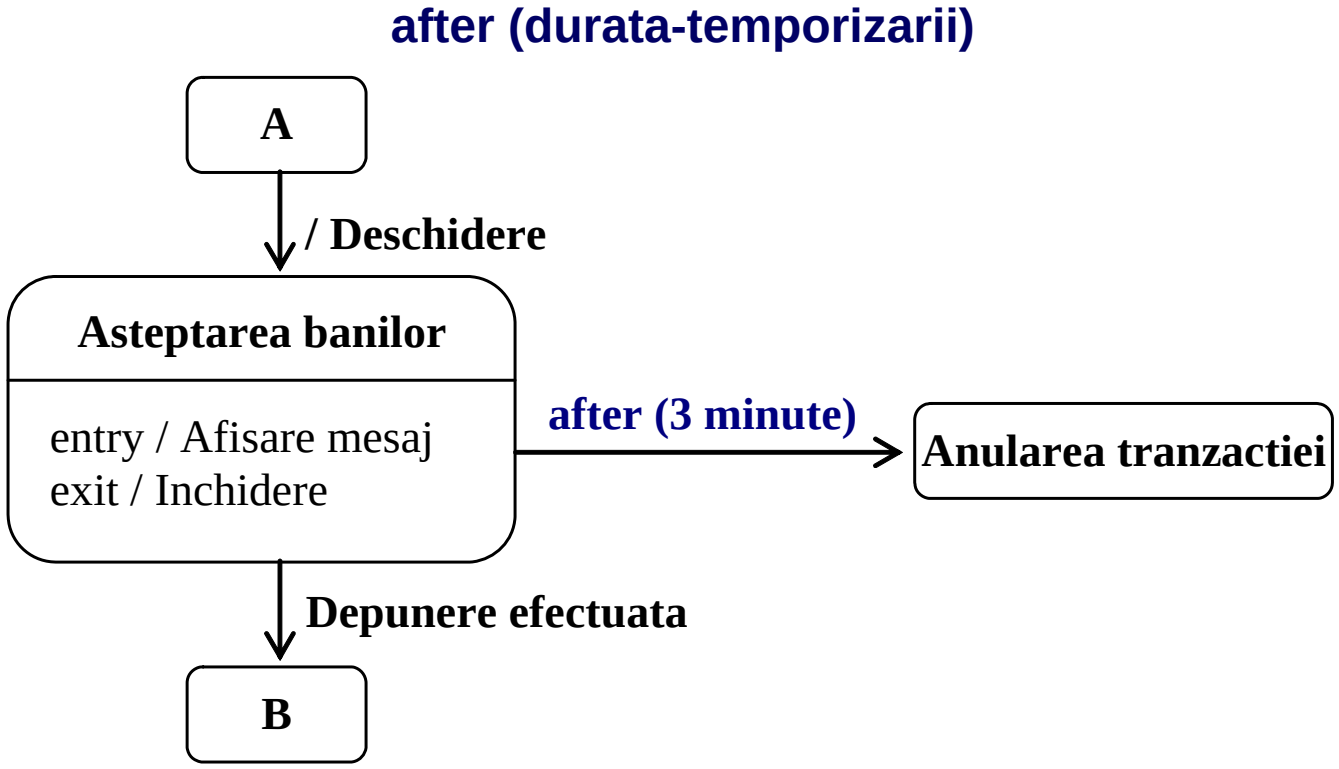


Diagramele masinilor de stari

Asteptarea (temporizarea) poate fi reprezentata prin intermediul unei notatii compacte

- **atasata direct tranzactiei declansate dupa intarzierea** de asteptare

Sintaxa unui eveniment de temporizare este:



Diagramele masinilor de stari

Exemplu complex