



MÔ HÌNH HOÁ PHẦN MỀM

TUẦN 4: STATE MACHINE DIAGRAM

GVLT: NGUYỄN THỊ MINH TUYỀN

cuu duong than cong . com

NỘI DUNG

1. Giới thiệu

2. Trạng thái

3. Chuyển tiếp

4. Loại sự kiện

5. Loại trạng thái

6. Các điểm vào/ra

NỘI DUNG

1. Giới thiệu

2. Trạng thái

[cuu duong than cong . com](http://cuuduongthancong.com)

3. Chuyển tiếp

4. Loại sự kiện

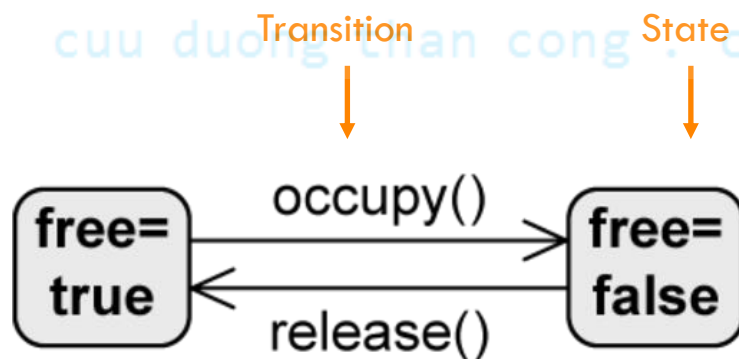
5. Loại trạng thái

cuu duong than cong . com

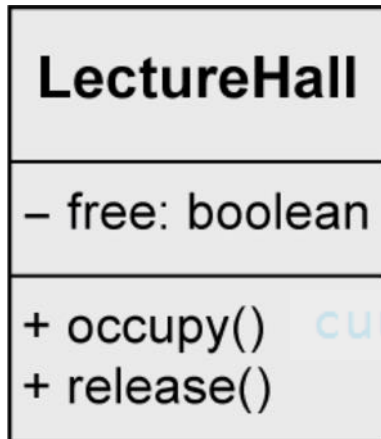
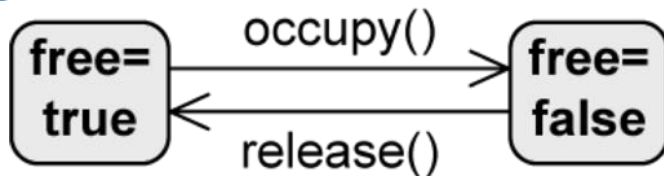
6. Các điểm vào/ra

GIỚI THIỆU

- Mỗi đối tượng có một số lượng hữu hạn các trạng thái suốt vòng đời của nó.
- Biểu đồ máy trạng thái (State machine diagram) được dùng để:
 - Mô hình các trạng thái có thể của một hệ thống hay đối tượng
 - Chỉ ra cách dịch chuyển trạng thái xảy ra như là kết quả của một sự kiện
 - Chỉ ra hành vi nào hệ thống hay đối tượng thể hiện ở mỗi trạng thái
- Ví dụ: mô tả ở mức cao của hành vi của một lecture hall

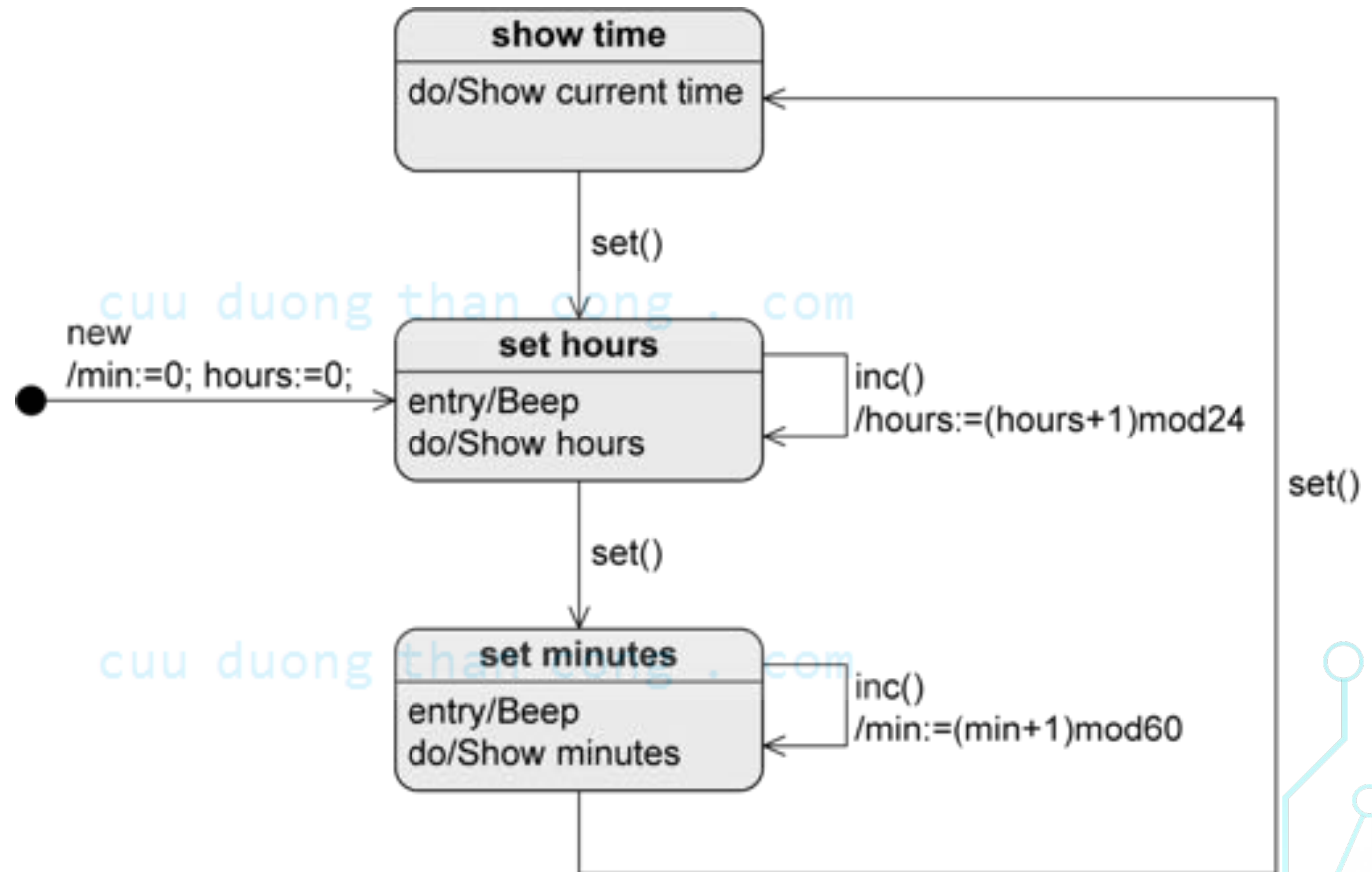
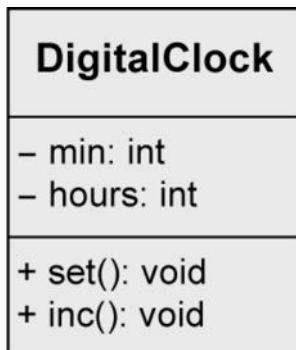


VÍ DỤ: LECTURE HALL WITH DETAILS



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

VÍ DỤ: DIGITAL CLOCK



NỘI DUNG

1. Giới thiệu

2. Trạng thái

cuuduongthancong.com

3. Chuyển tiếp

4. Loại sự kiện

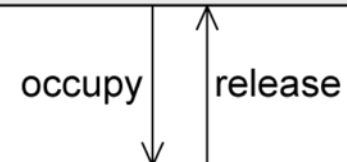
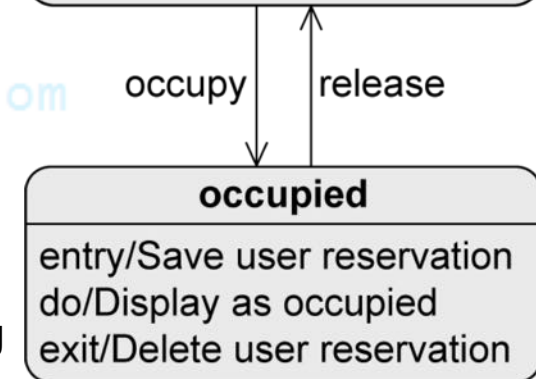
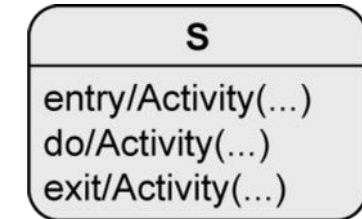
5. Loại trạng thái

cuuduongthancong.com

6. Composite state

TRẠNG THÁI

- Trạng thái = các node của máy trạng thái
- Khi một trạng thái đang hoạt động
 - Đối tượng đang ở trạng thái đó
 - Tất cả các hoạt động bên trong đặc tả trong trạng thái này có thể được thực thi
 - Một hoạt động gồm nhiều hành động
- entry / Activity(...)
 - Được thực thi khi đối tượng vào trạng thái
- exit / Activity(...)
 - Được thực thi khi đối tượng thoát khỏi trạng thái
- do / Activity(...)
 - Được thực thi trong khi đối tượng còn duy trì trạng thái này



NỘI DUNG

1. Giới thiệu

2. Trạng thái

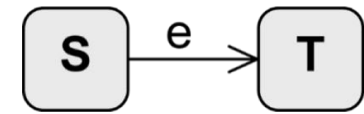
3. Chuyển tiếp

4. Loại sự kiện

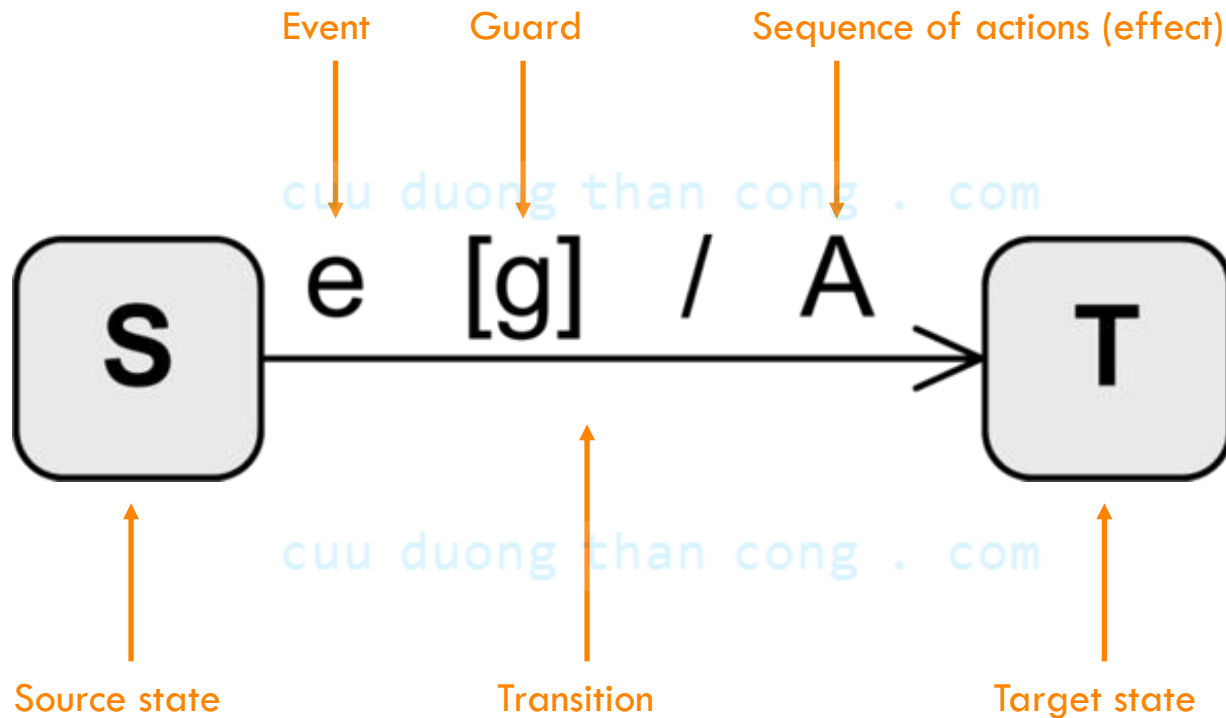
5. Loại trạng thái

6. Composite state

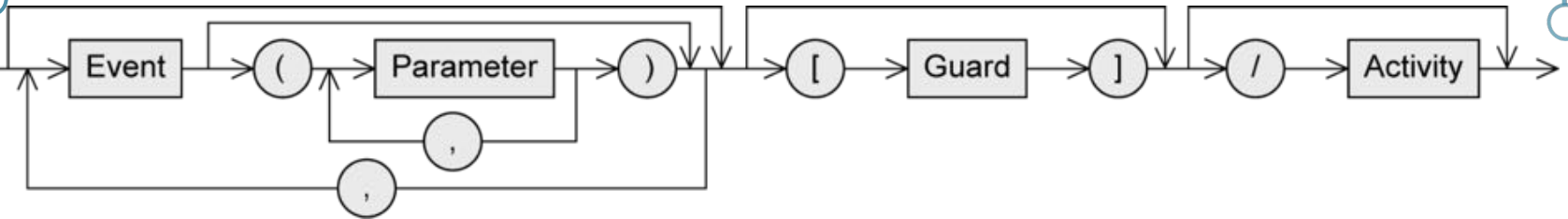
CHUYỂN TIẾP



- Chuyển từ trạng thái này sang trạng thái khác



CÚ PHÁP



- Event (trigger)

- Kích hoạt ngoại sinh
- Có thể kích hoạt quá trình chuyển đổi trạng thái

- Guard (condition)

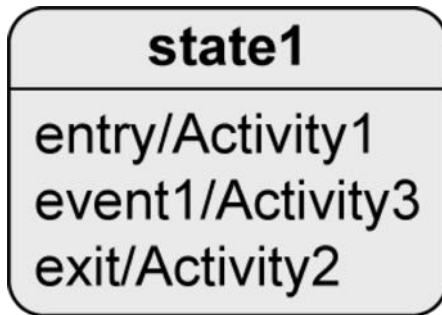
- Biểu thức Boolean
- Nếu event xảy ra, guard được kiểm tra
- Nếu guard đúng:
 - Tất cả các hoạt động của trạng thái hiện tại bị chấm dứt.
 - Mọi hoạt động thoát liên quan

được thực thi

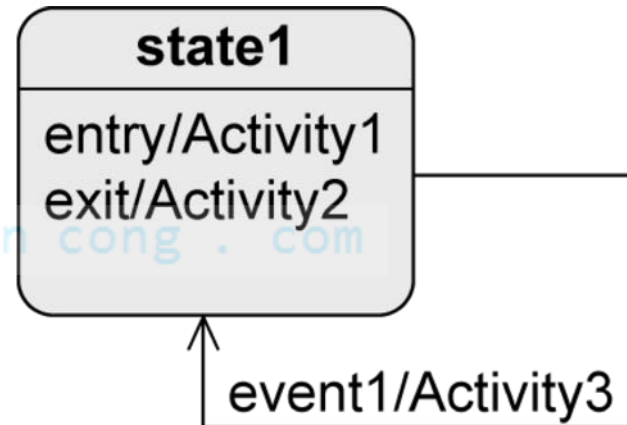
- Quá trình chuyển trạng thái diễn ra
- Nếu guard sai:
 - Không có quá trình chuyển trạng thái nào diễn ra, event bị huỷ bỏ
- Activity (effect)
 - Chuỗi các hành động được thực hiện trong quá trình chuyển đổi trạng thái

LOẠI CHUYỂN TIẾP [1]

Internal transition



External transition

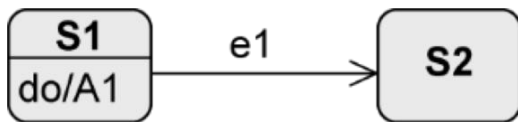


- Nếu **event1** xảy ra
 - Đối tượng duy trì trong **state1**
 - **Activity3** được thực hiện

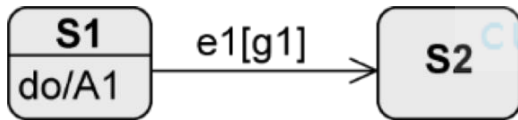
- Nếu **event1** xảy ra
 - Đối tượng rời khỏi **state1** và **Activity2** được thực hiện
 - **Activity3** được thực hiện
 - Đối tượng vào **state1** và **Activity1** được thực hiện

LOẠI CHUYỂN TIẾP [2]

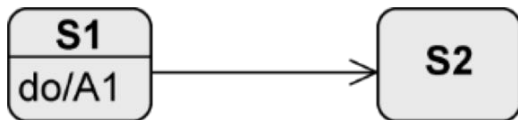
- Khi nào các chuyển tiếp sau diễn ra ?



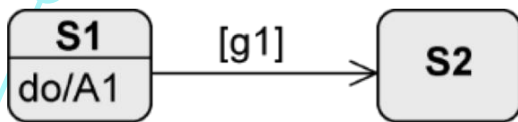
If **e1** occurs, **A1** is aborted and the object changes to **S2**



If **e1** occurs and **g1** evaluates to true, **A1** is aborted and the object changes to **S2**



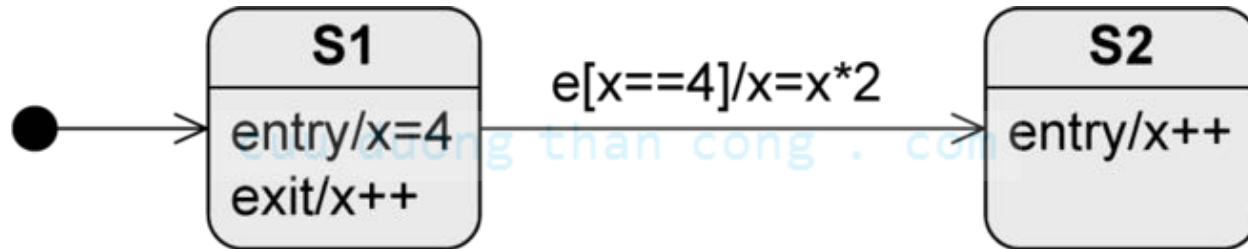
As soon as the execution of **A1** is finished, a completion event is generated that initiates the transition to **S2**



As soon as the execution of **A1** is finished, a completion event is generated; if **g1** evaluates to true, the transition takes place; If not, this transition can never happen

CHUỖI CÁC HOẠT ĐỘNG ĐƯỢC THỰC HIỆN

- Giả sử **S1** đang ở trạng thái hoạt động, giá trị của **x** sau khi **e** xảy ra là gì?



S1 becomes active, **x** is set to the value **4**

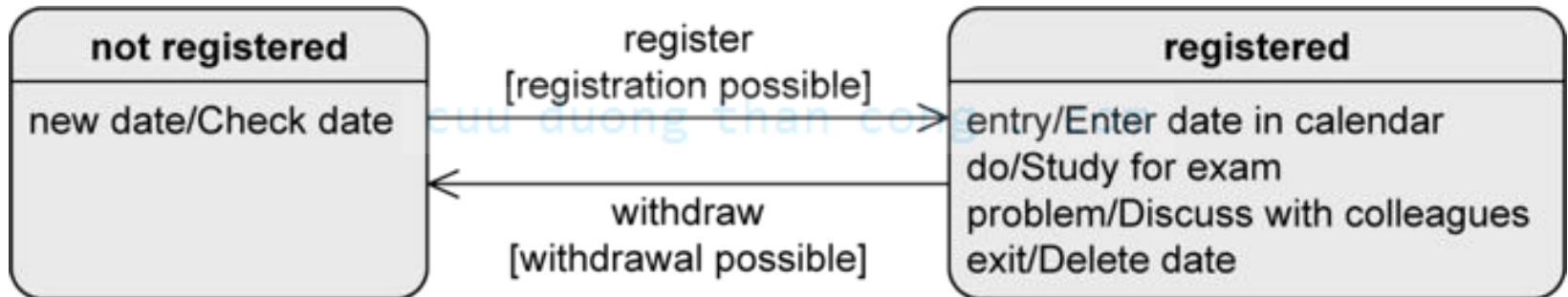
e occurs, the guard is checked and evaluates to true

S1 is left, **x** is set to **5**

The transition takes place, **x** is set to **10**

S2 is entered, **x** is set to **11**

VÍ DỤ: REGISTRATION STATUS OF AN EXAM



cuu duong than cong . com

NỘI DUNG

1. Giới thiệu

2. Trạng thái

3. Chuyển tiếp

4. Loại sự kiện

5. Loại trạng thái

6. Composite state

LOẠI SỰ KIỆN [1]

- **Signal event:** Bên gửi gửi tín hiệu cho bên nhận mà không cần đợi trả lời (giao tiếp không đồng bộ)
 - Ví dụ: `mousedown`, `sendSMS(message)`
- **Call event:** Triệu gọi thao tác
 - Ví dụ: `occupy(user, lectureHall)`, `register(exam)`
- **Time event:** Chuyển đổi trạng thái dựa trên thời gian
 - Tương đối: dựa trên thời gian xảy ra sự kiện trong trạng thái active
 - Ví dụ: `after(5 seconds)`
 - Tuyệt đối
 - Ví dụ: `when(time==16:00)`, `when(date==20150101)`

LOẠI SỰ KIỆN [2]

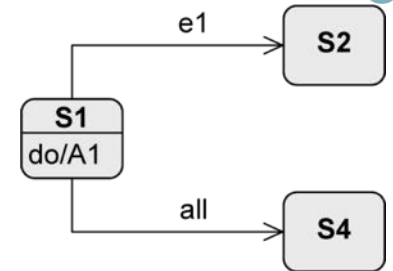
- **Any receive event:** đặc tả loại chuyển tiếp 'else', xảy ra khi bất kỳ sự kiện nào xảy ra mà không kích hoạt chuyển tiếp khác từ trạng thái hoạt động

- Từ khoá **a11**

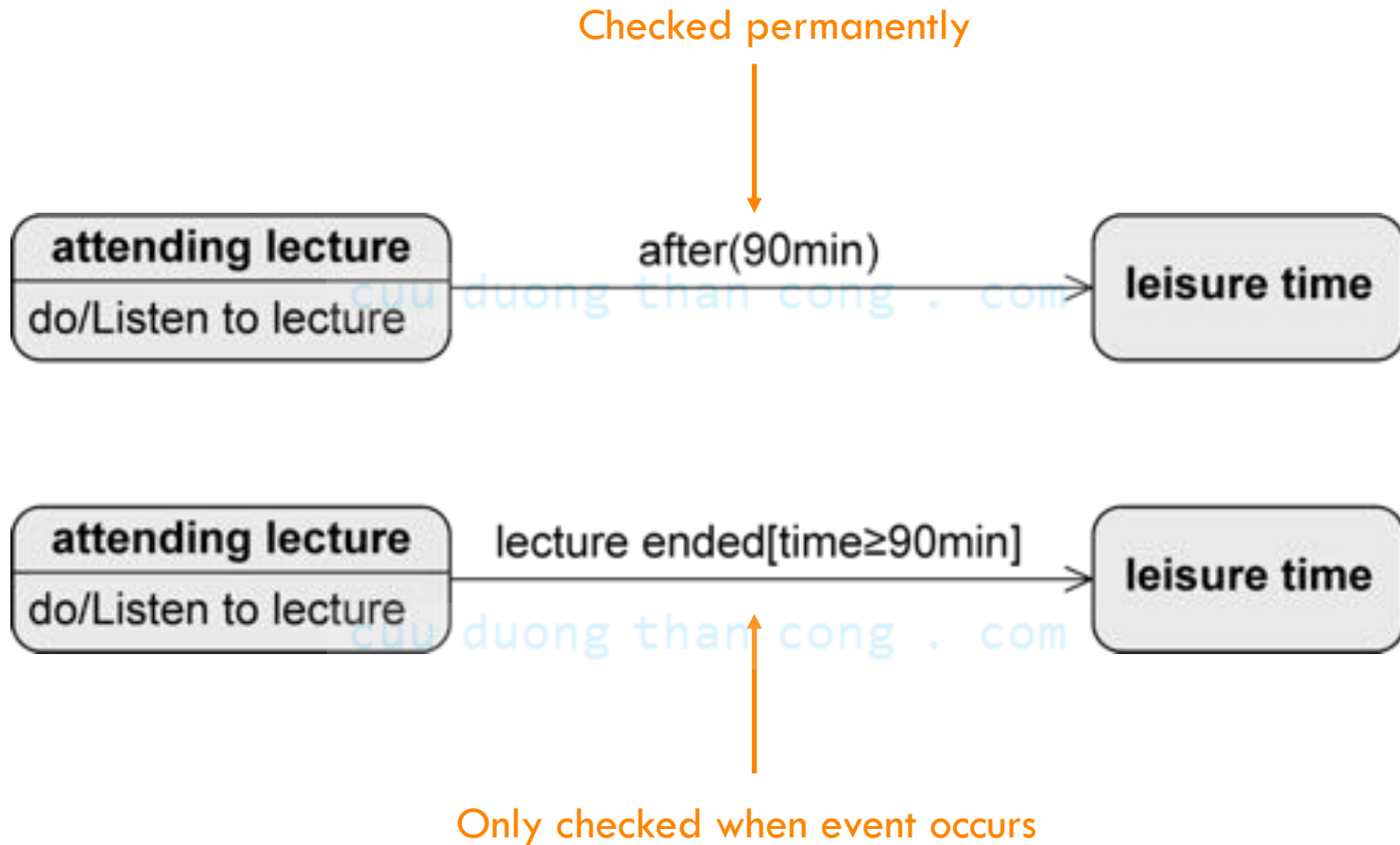
- **Completion event:** Được tạo tự động khi mọi thứ được thực hiện ở trạng thái hiện tại được hoàn thành.

- **Change event:** Kiểm tra lâu dài xem điều kiện có đúng không

- E.g., **when($x > y$), after(90min)**



CHANGE EVENT VS. GUARD



NỘI DUNG

1. Giới thiệu

2. Trạng thái

cuuduongthancong.com

3. Chuyển tiếp

4. Loại sự kiện

5. Loại trạng thái

cuuduongthancong.com

6. Composite state

INITIAL STATE

- “Bắt đầu” của một biểu đồ máy trạng thái
- Pseudostate
 - Tạm thời, tức là hệ thống không thể duy trì trong trạng thái đó
 - Là cấu trúc điều khiển hơn là một trạng thái thật
- Không có chuyển tiếp vào, thường có 1 cạnh ra
- Nếu cạnh ra > 1
 - Guard phải loại trừ lẫn nhau và bao phủ tất cả các trường hợp có thể xảy ra để đảm bảo duy nhất một trạng thái đích được đạt đến
- Nếu trạng thái đầu được kích hoạt, đối tượng sẽ lập tức chuyển sang trạng thái tiếp theo
 - Không có sự kiện nào được phép trên cạnh ra (ngoại trừ : **new()**)

FINAL STATE VÀ TERMINATE NODE

Final State



- Là trạng thái thật
- Đánh dấu kết thúc chuỗi trạng thái
- Đối tượng có thể duy trì trong trạng thái cuối mãi mãi

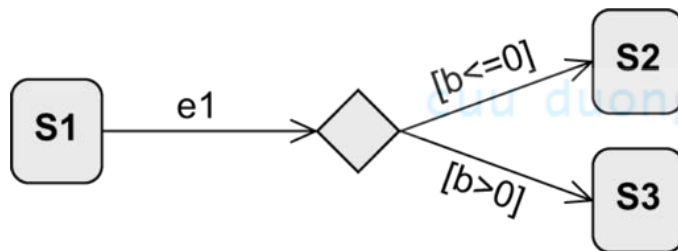
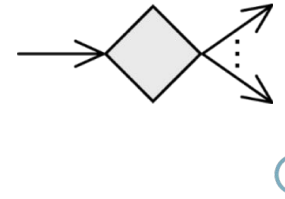
Terminate Node



- Pseudostate
- Kết thúc máy trạng thái
- Đối tượng được mô hình chấm dứt sự tồn tại (= is deleted)

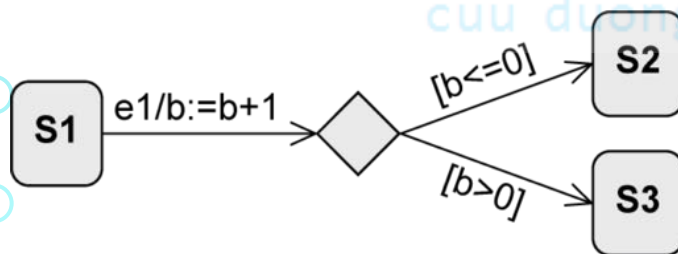
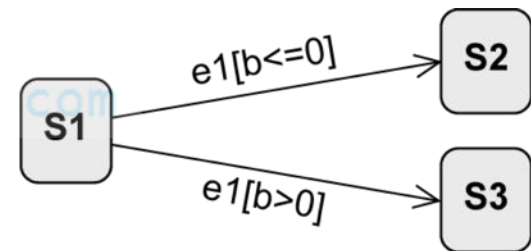
DECISION NODE

- Pseudostate
- Được dùng để mô hình các chuyển tiếp thay thế



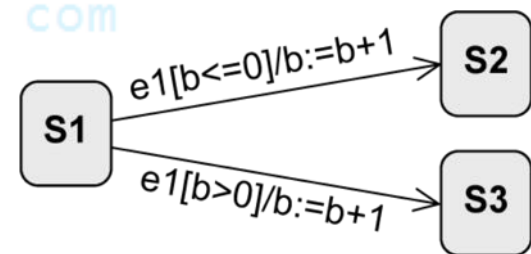
equivalent?

=

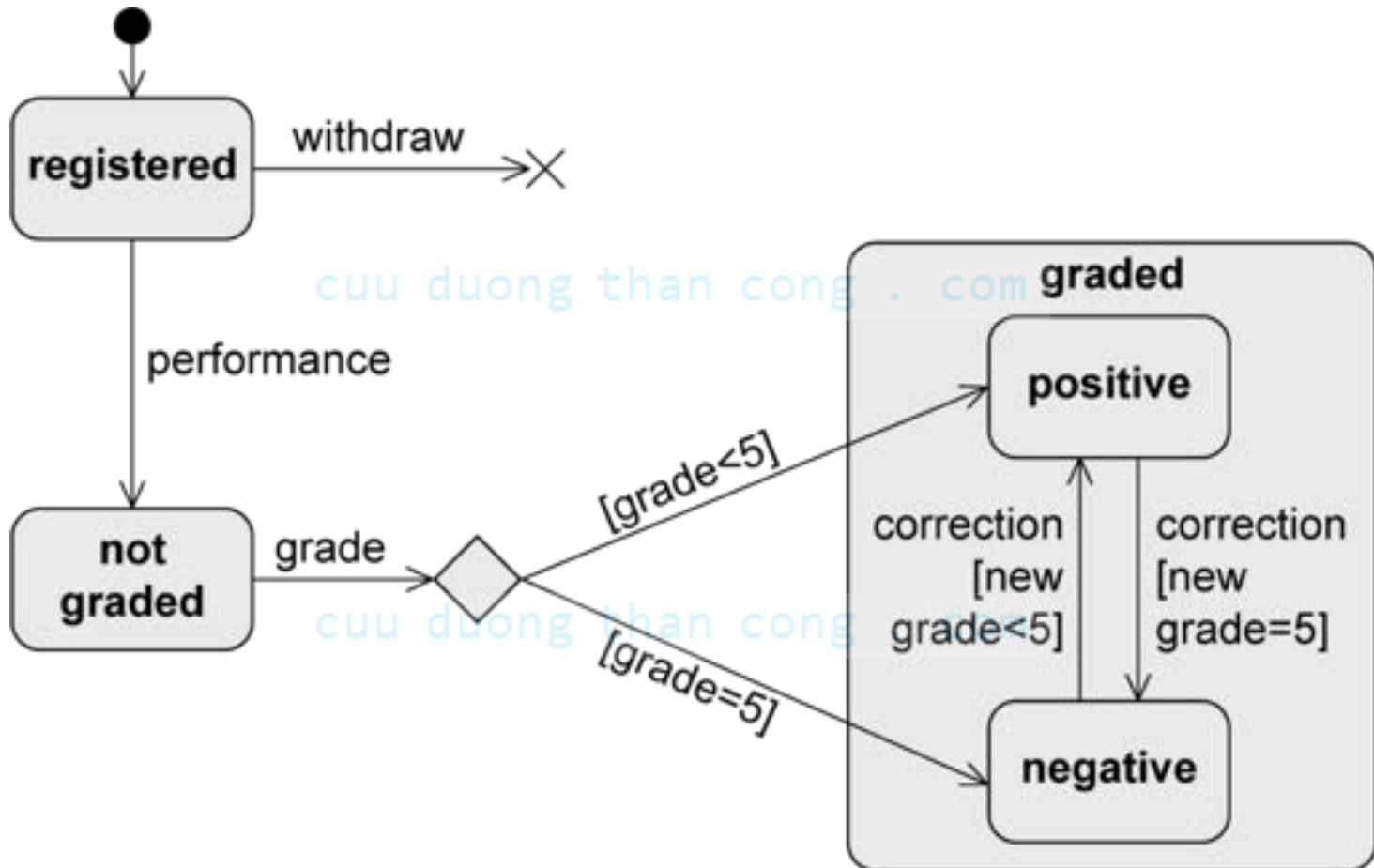


equivalent?

≠



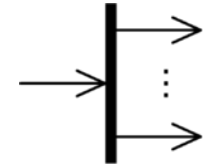
EXAMPLE: DECISION NODE



PARALLELIZATION VÀ SYNCHRONIZATION NODE

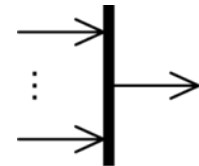
Parallelization node

- Pseudostate
- Tách luồng điều khiển thành nhiều luồng đồng thời
- 1 cạnh vào
- >1 cạnh ra



Synchronization node

- Pseudostate
- Hợp nhất nhiều luồng đồng thời
- >1 cạnh vào
- 1 cạnh ra



NỘI DUNG

1. Giới thiệu

2. Trạng thái

[cuu duong than cong . com](https://fb.com/taiieuudentuoc)

3. Chuyển tiếp

4. Loại sự kiện

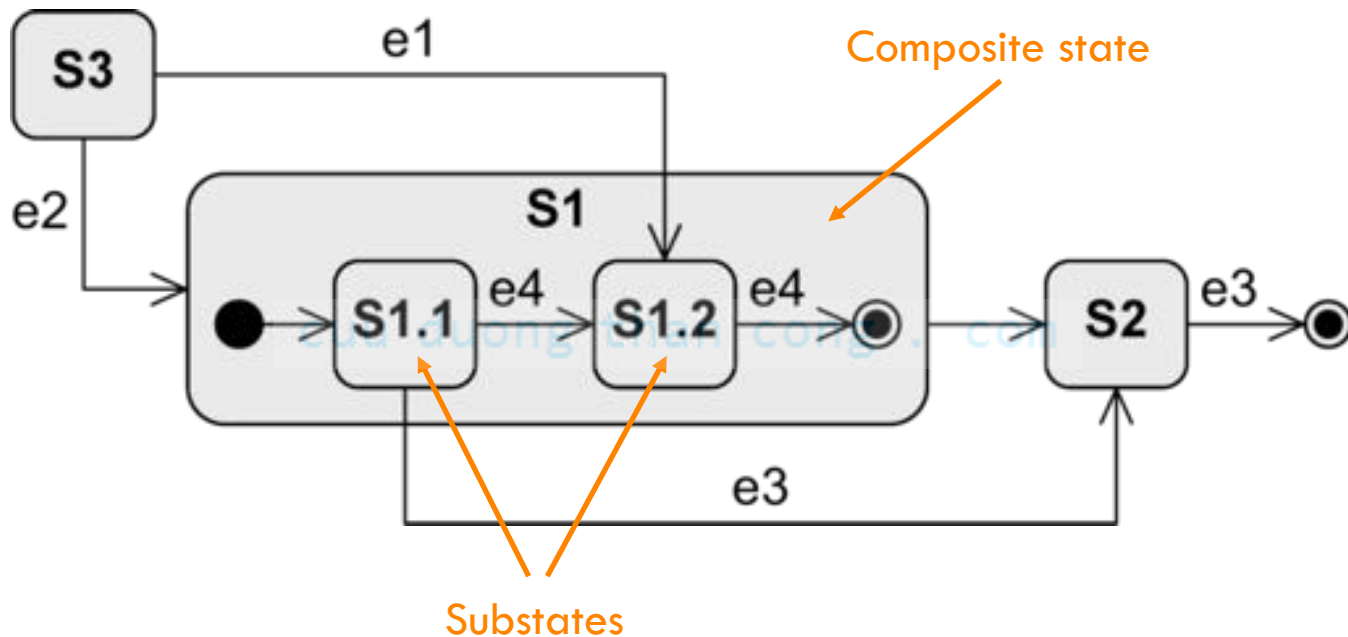
5. Loại trạng thái

[cuu duong than cong . com](https://fb.com/taiieuudentuoc)

6. Composite state

COMPOSITE STATE

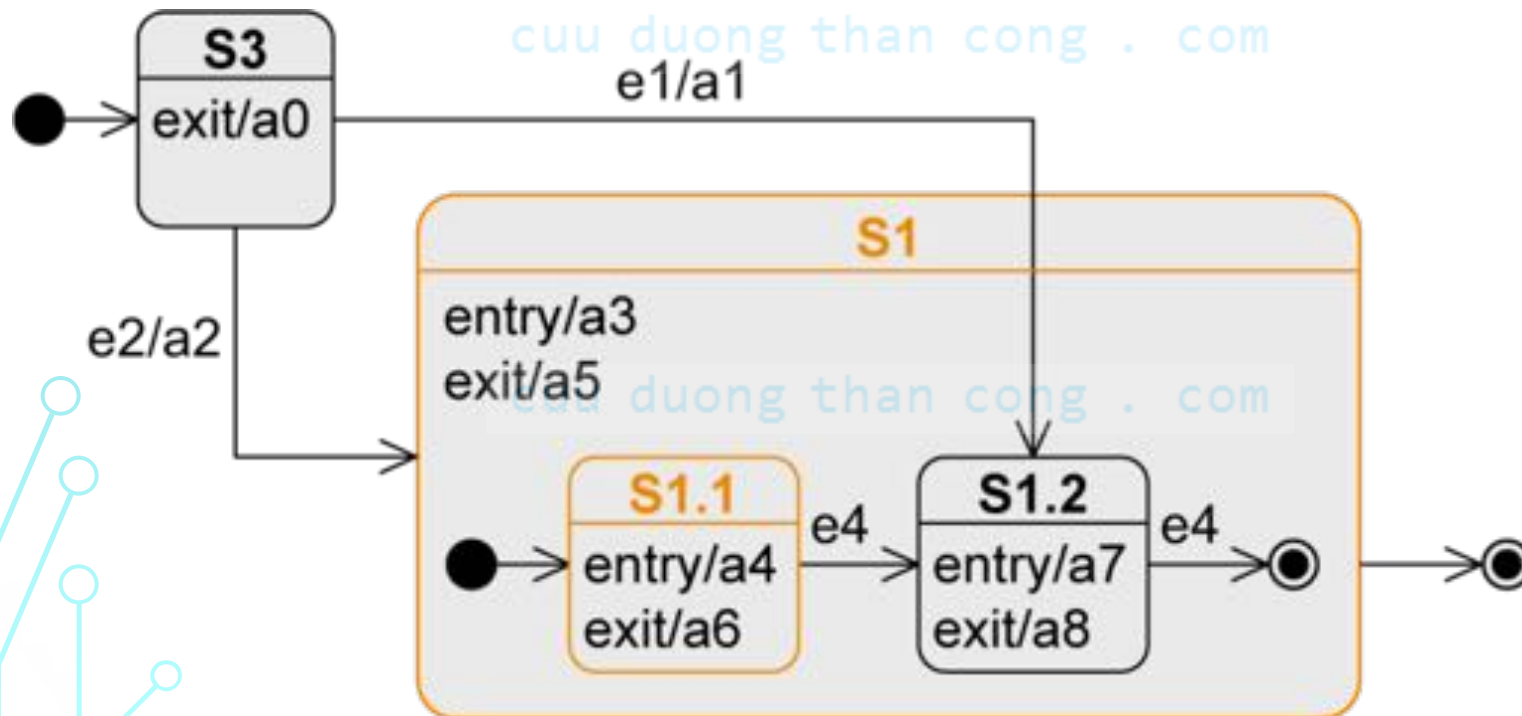
- Từ đồng nghĩa: complex state, nested state
- Chứa các trạng thái khác – “substates”
 - Chỉ một trong các substate được kích hoạt tại một thời điểm
- Độ sâu của các substate là tùy ý



VÀO MỘT COMPOSITE STATE (1/2)

- Chuyển đến ranh giới
 - Initial node của composite state được kích hoạt

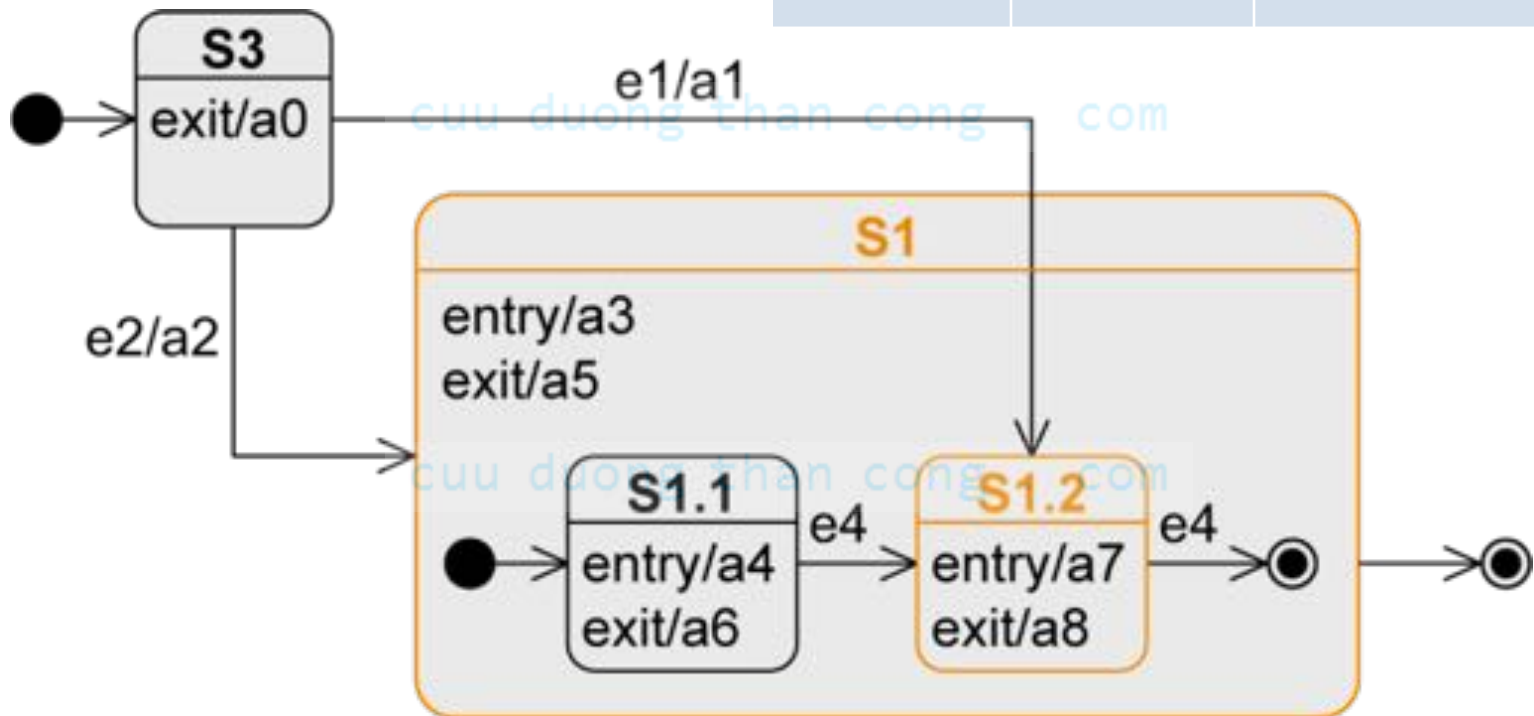
Event	State	Executed Activities
"Beginning"	S3	
e2	S1/S1.1	a0-a2-a3-a4



VÀO MỘT COMPOSITE STATE (2/2)

- Chuyển đến một substate
 - Substate được kích hoạt

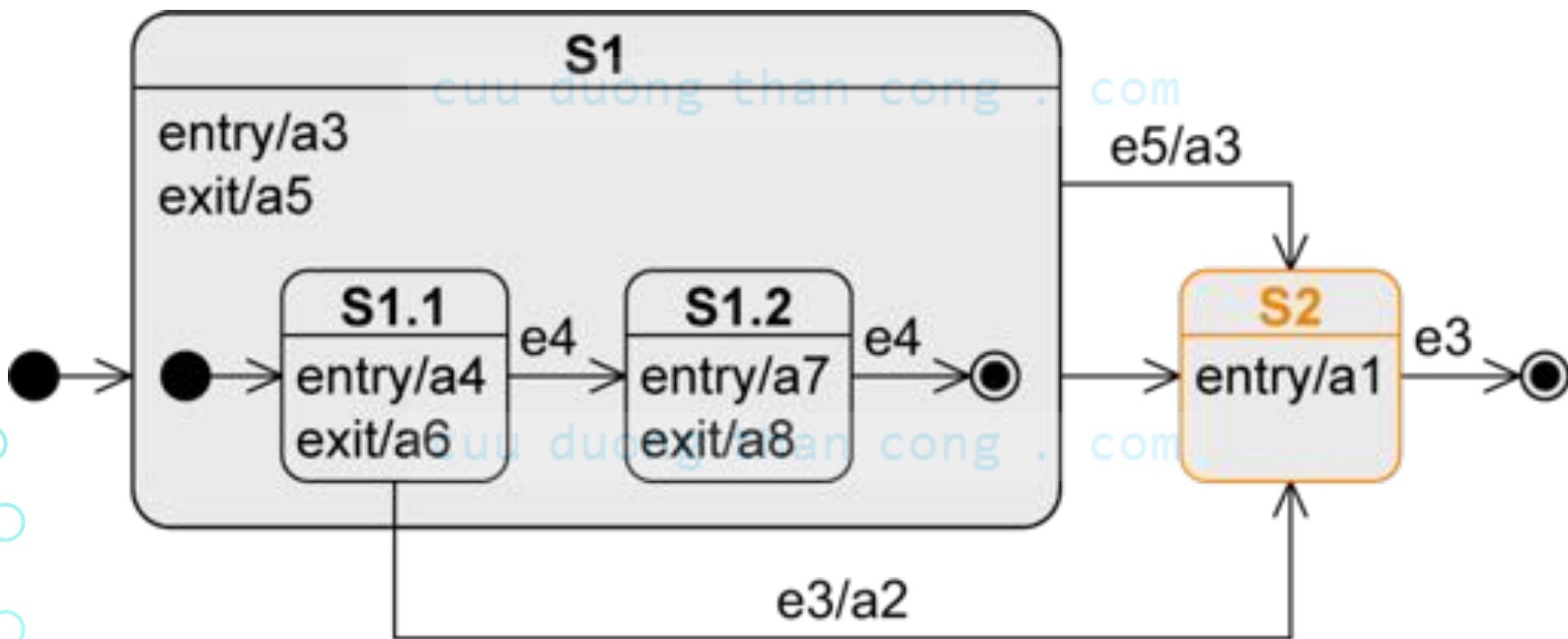
Event	State	Executed Activities
"Beginning"	S3	
e1	S1/S1.2	a0-a1-a3-a7



RA KHỎI MỘT COMPOSITE STATE (1/3)

- Chuyển đi từ một substate

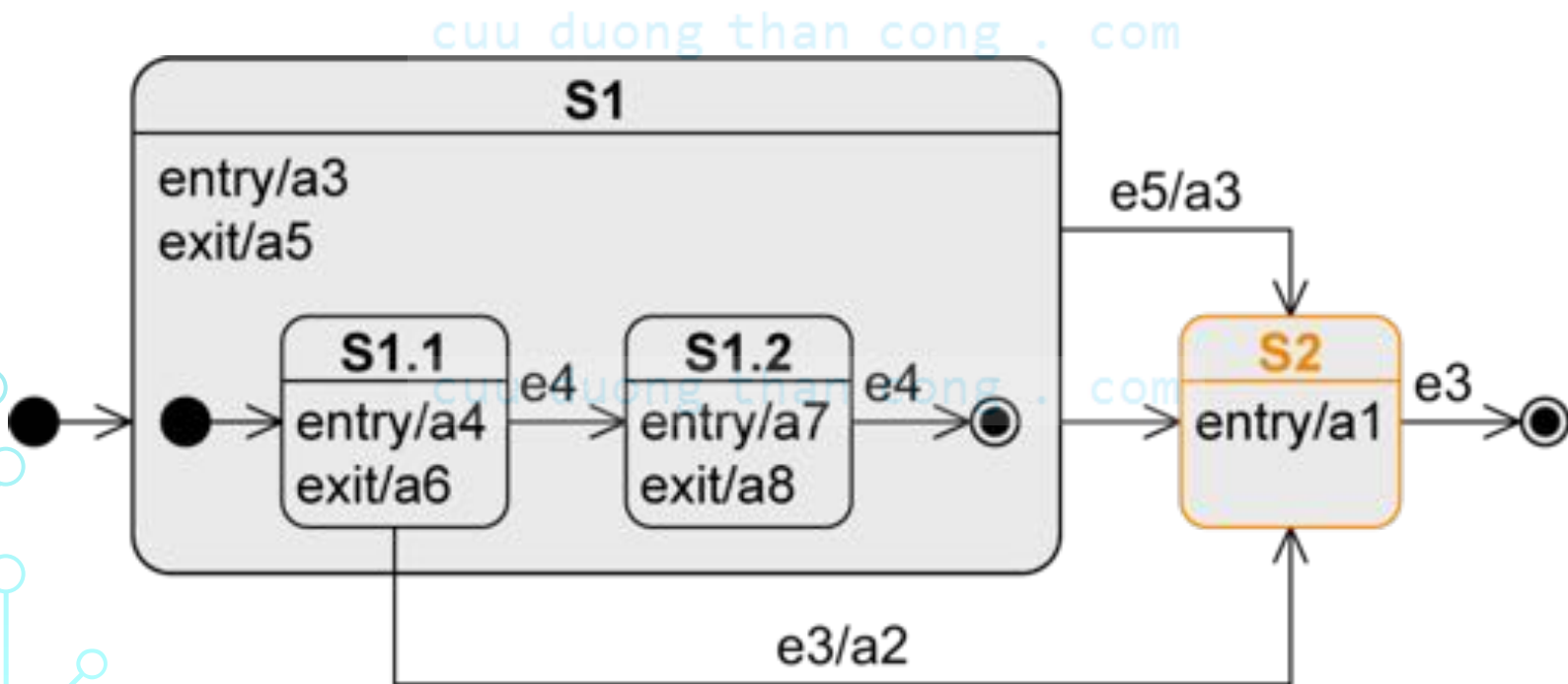
Event	State	Executed Activities
"Beginning"	S1/S1.1	a3-a4
e3	S2	a6-a5-a2-a1



RA KHỎI MỘT COMPOSITE STATE (2/3)

- Chuyển ra khỏi composite state

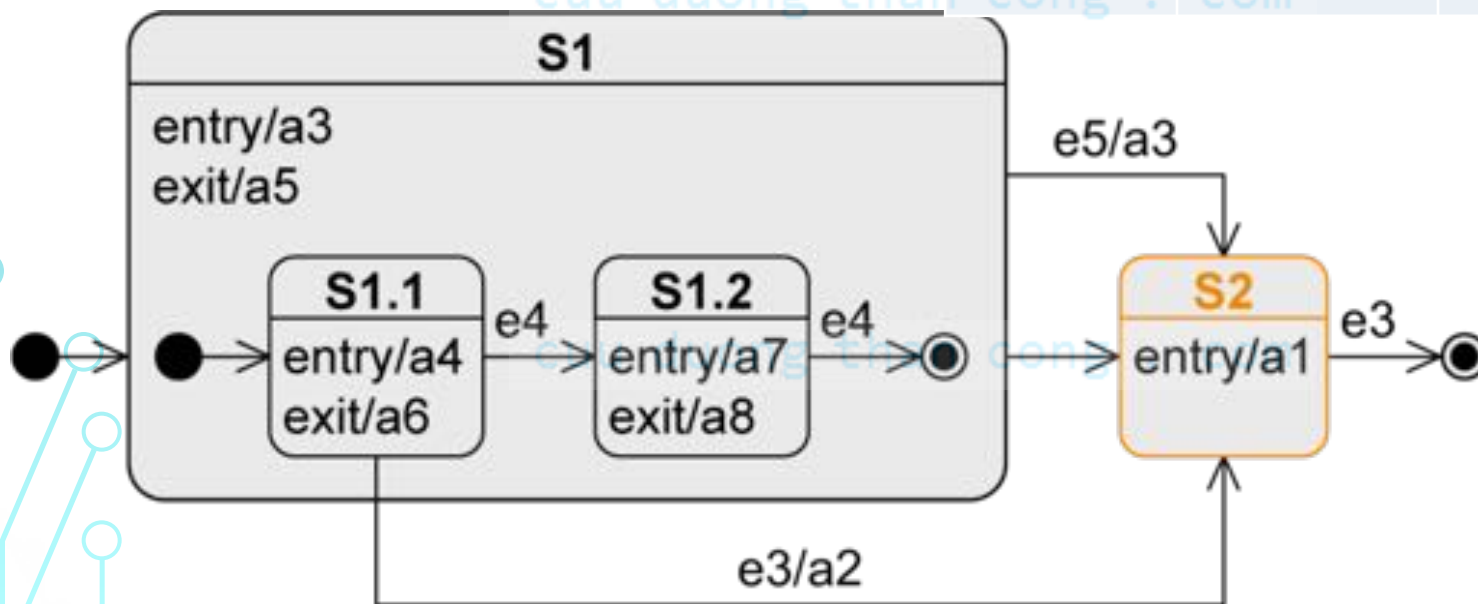
Event	State	Executed Activities
"Beginning"	S1/S1.1	a3-a4
e5	S2	a6-a5-a3-a1



RA KHỎI MỘT COMPOSITE STATE (3/3)

- Chuyển đổi hoàn toàn từ composite state

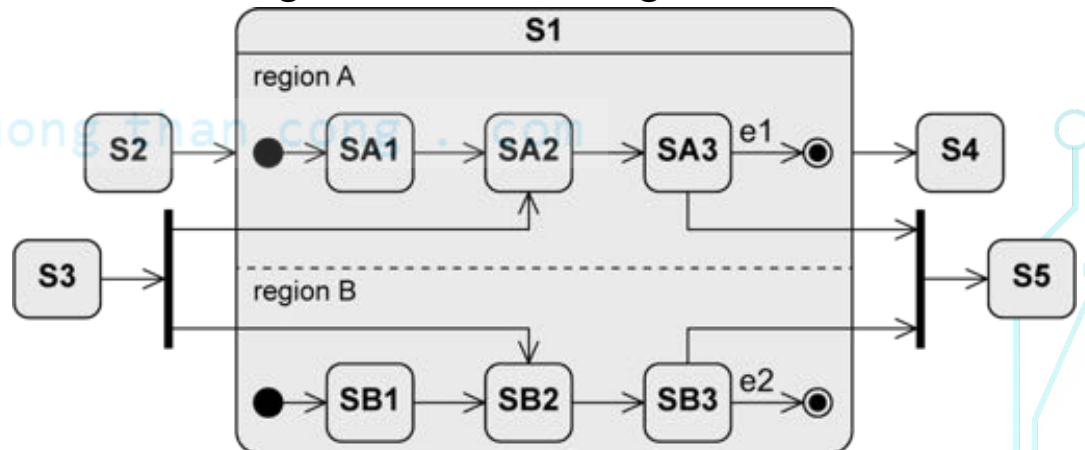
Event	State	Executed Activities
"Beginning"	S1/S1.1	a3-a4
e4	S1/S1.2	a6-a7
e4	S2	a8-a5-a1



ORTHOGONAL STATE

- Composite state được phân thành hai hoặc nhiều vùng tách biệt nhau bằng đường nét đứt
- Một trạng thái của mỗi vùng luôn luôn được kích hoạt tại thời điểm nào đó, tức là các substate đồng thời
- Entry: chuyển đến ranh giới của orthogonal state kích hoạt tất cả các initial state của tất cả các vùng
- Exit: final state phải được đạt đến trong tất cả các vùng để kích hoạt sự kiện hoàn thành

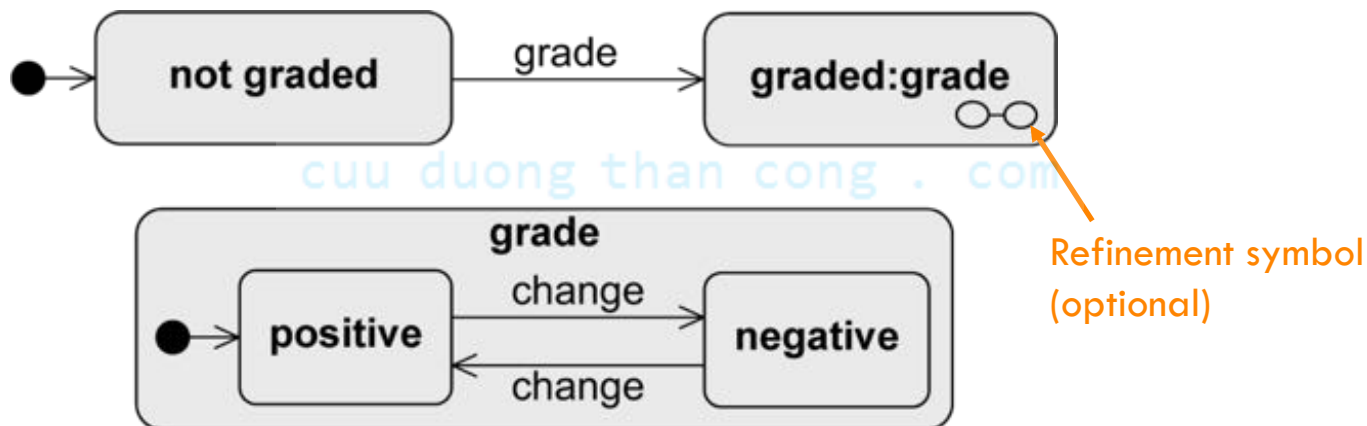
Using parallelization and synchronization node to enter different substates



SUBMACHINE STATE (SMS)

S:SMS

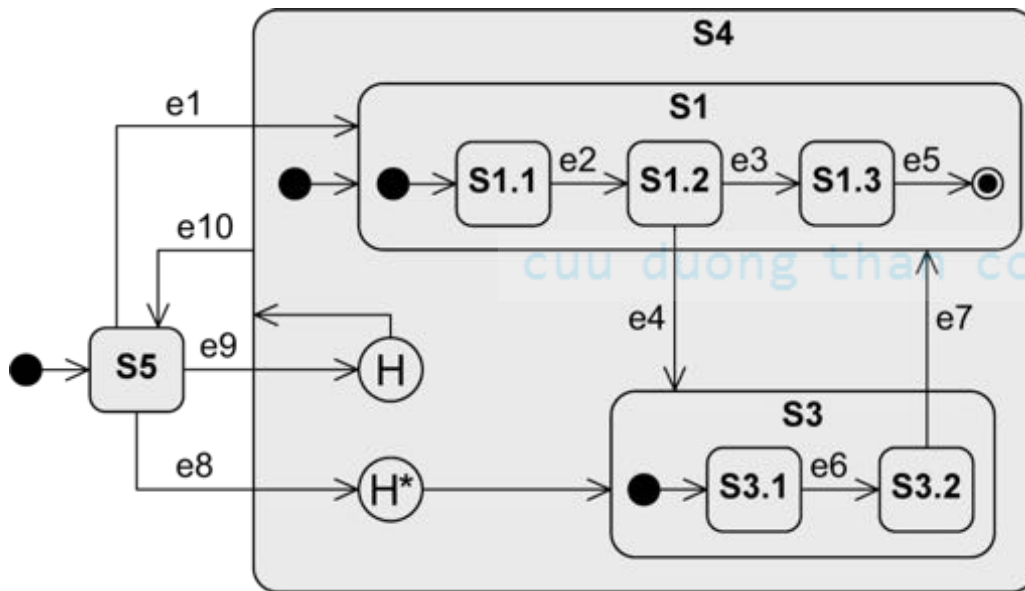
- Để tái sử dụng các phần của biểu đồ máy trạng thái trong các biểu đồ máy trạng thái khác
- Ký hiệu: **state:submachineState**
- Ngay khi submachine state được kích hoạt, hành vi của submachine được thực thi
 - Tương ứng với việc gọi một chương trình con trong ngôn ngữ lập trình



HISTORY STATE

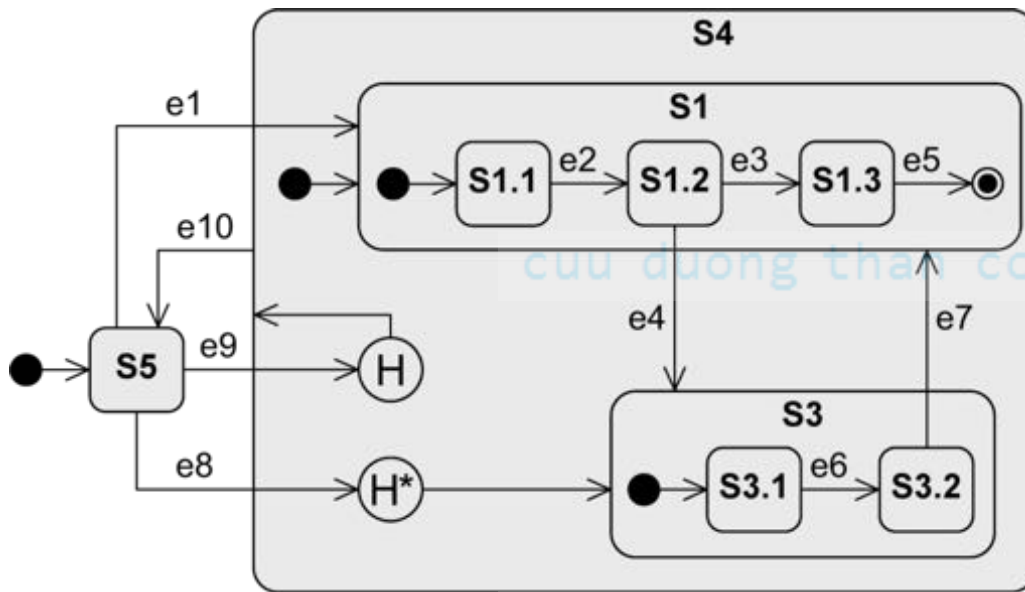
- Được sử dụng khi, sau khi một chuyển tiếp bên ngoài dẫn đến việc thoát khỏi composite state, hệ thống trở lại cùng substate đã được kích hoạt trước khi việc chuyển tiếp xảy ra.
- Kích hoạt substate "cũ" và tất cả các entry activity được tiến hành tuần tự từ bên ngoài đến bên trong composite state
- Chính xác một cạnh ra của history state chỉ đến một substate đã được sử dụng nếu
 - composite state không bao giờ được kích hoạt trước đó
 - composite state được thoát ra thông qua final state
- Shallow history state khôi phục trạng thái trên cùng level với composite state (H)
- Deep history state khôi phục trạng thái kích hoạt cuối cùng active substate toàn bộ chiều sâu của substate (H*)

VÍ DỤ: HISTORY STATE (1/4)



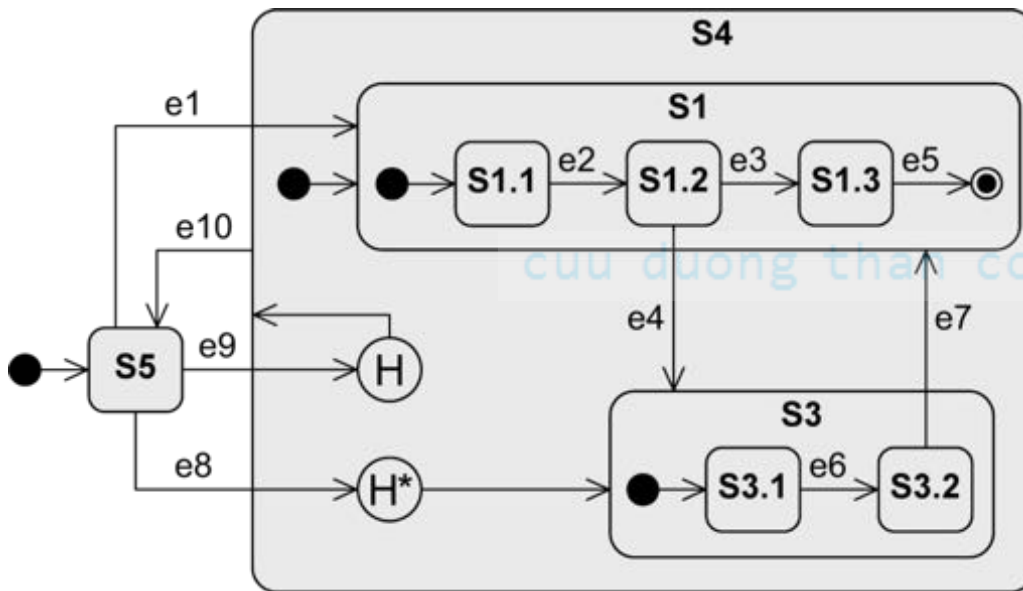
Event	State
"Beginning"	S5
e1	S4/S1/S1.1
e2	S1.2
e10	S5
e9	(H→) S1/S1.1

VÍ DỤ: HISTORY STATE (2/4)



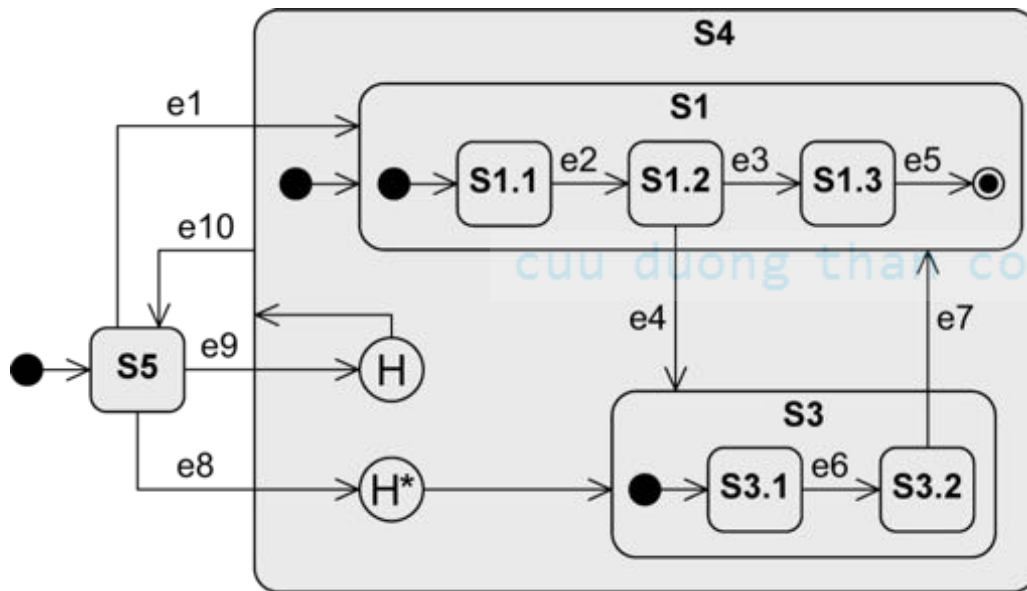
Event	State
"Beginning"	S5
e1	S4/S1/S1.1
e2	S1.2
e10	S5
e8	(H*→) S1.2

VÍ DỤ: HISTORY STATE (3/4)



Event	State
"Beginning"	S5
e9	(H→) S1/S1.1

VÍ DỤ: HISTORY STATE (4/4)



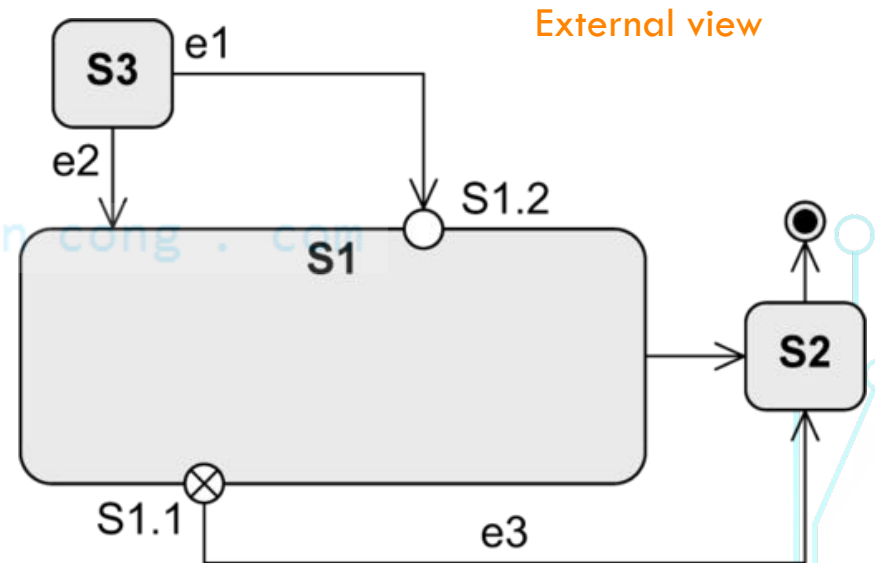
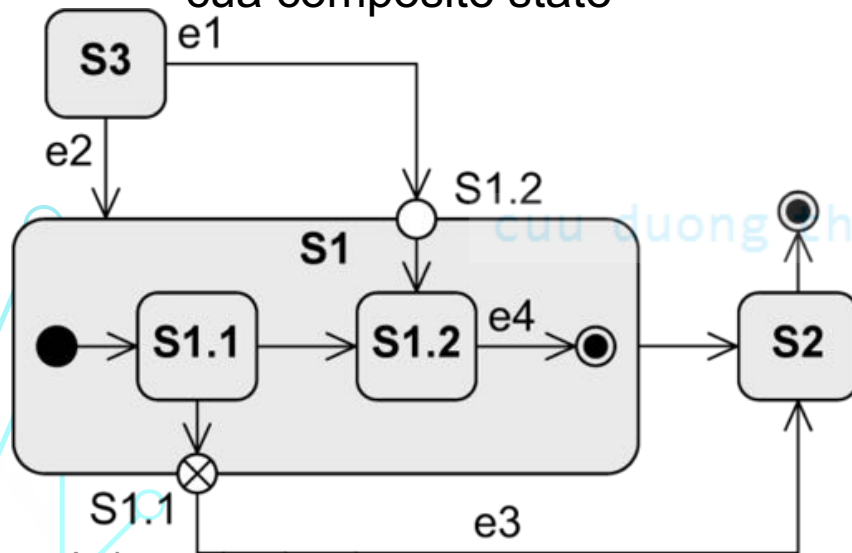
Event	State
"Beginning"	S5
e8	(H* →) S3/S3.1

CÁC ĐIỂM VÀO/RA

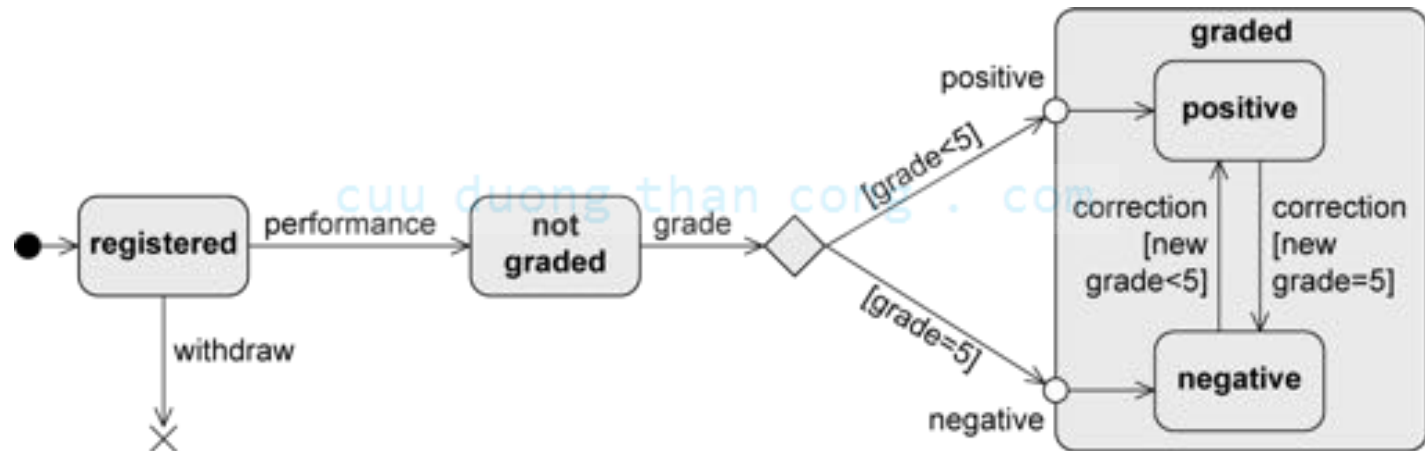
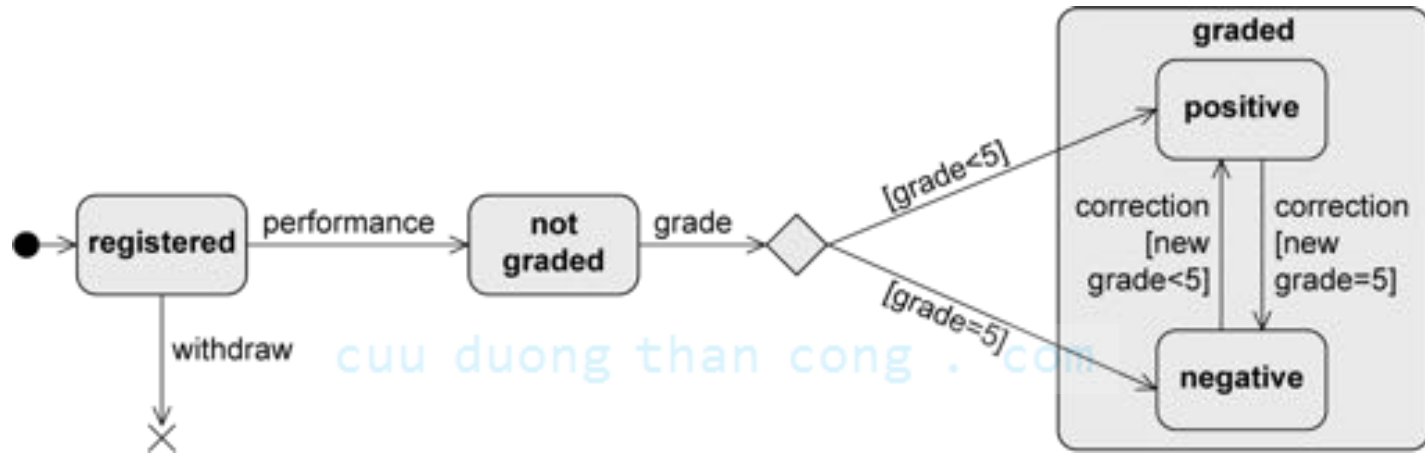
- Entry and Exit Points

- Cơ chế đóng gói

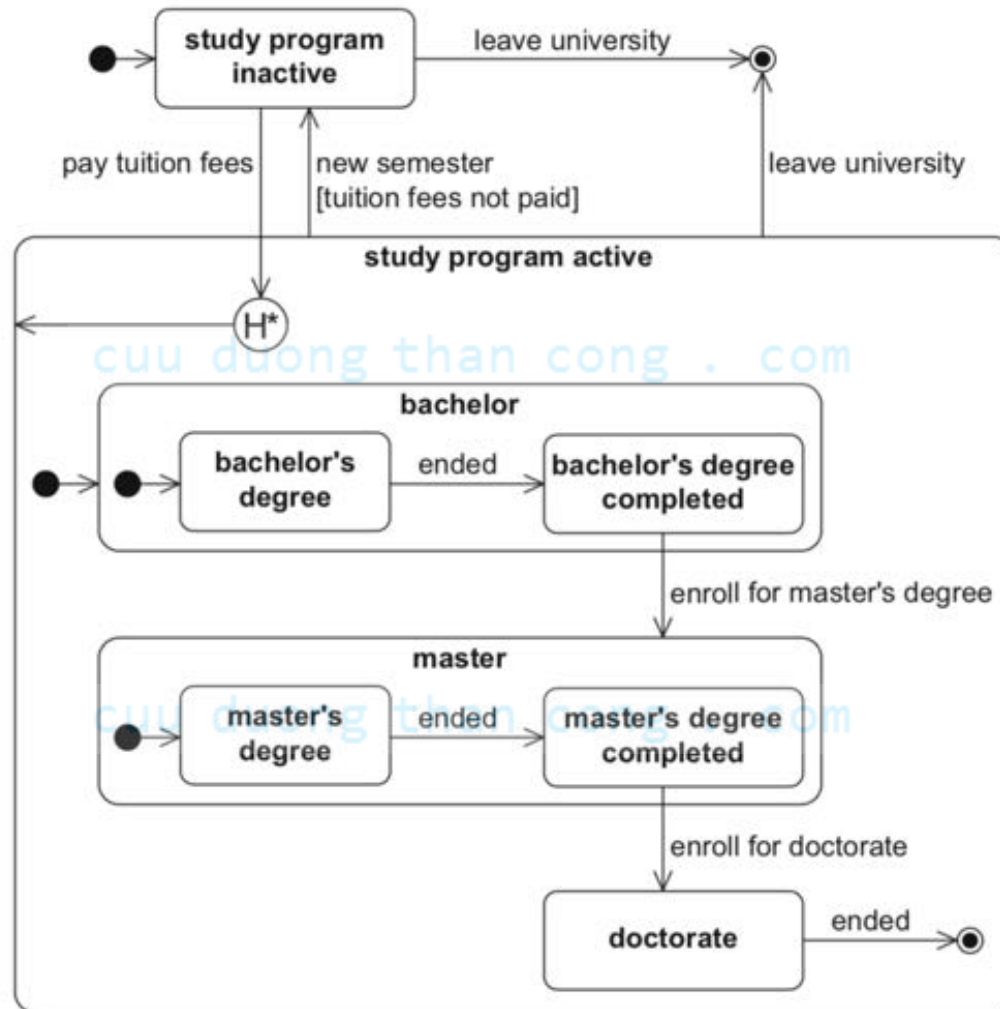
- Một composite state sẽ được vào hoặc thoát ra thông qua một trạng thái hơn là các initial state và final state
- Quá trình chuyển tiếp bên ngoài mà không cần biết cấu trúc bên trong của composite state



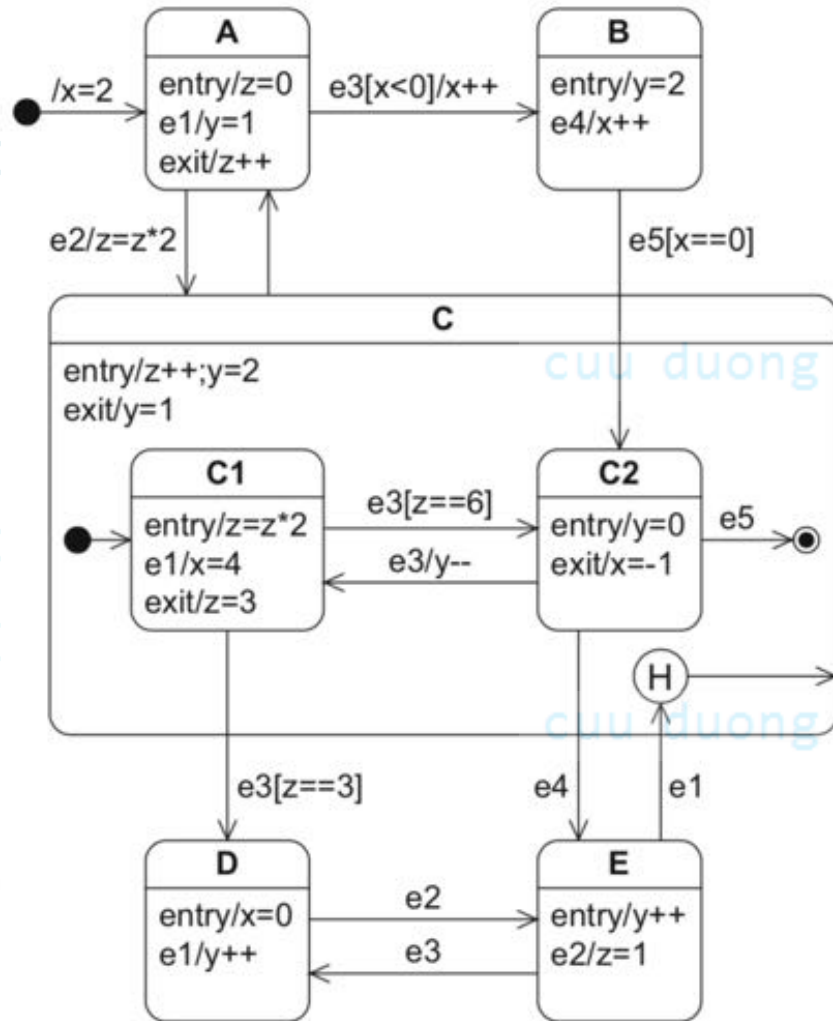
VÍ DỤ



VÍ DỤ MINH HOẠ [1]

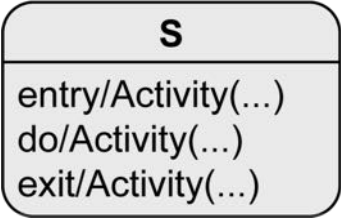
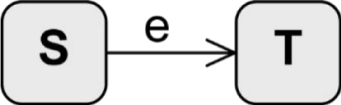





VÍ DỤ MINH HOẠ [2]

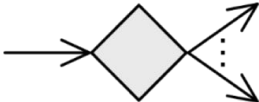
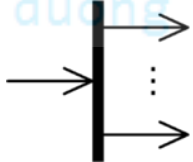
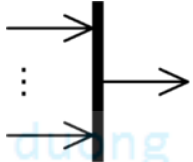
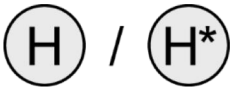


Event	State entered	x	y	z
Start	A	2		0
e2	C1		2	6
e1	C1	4		
e3	C2		0	3
e4	E	-1	2	
e1	C2		0	4
e5	A	-1	1	0

NOTATION ELEMENTS (1/2)

Name	Notation	Description
State		Description of a specific “time span” in which an object finds itself during its “life cycle”. Within a state, activities can be executed by the object.
Transition		State transition e from a source state S to a target state T
Initial state		Start of a state machine diagram
Final state		End of a state machine diagram
Terminate node		Termination of an object’s state machine diagram

NOTATION ELEMENTS (2/2)

Name	Syntax	Beschreibung
Decision node		Node from which multiple alternative transitions can origin
Parallelization node		Splitting of a transition into multiple parallel transitions
Synchronization node		Merging of multiple parallel transitions into one transition
Shallow / deep history state		"Return address" to a substate or a nested substate of a composite state

Câu hỏi?

cuu duong than cong . com

cuu duong than cong . com