



Polytech Paris-Sud  
Formation initiale 3<sup>e</sup> année  
Spécialité Informatique  
Année 2014-2015

# UML

## Diagrammes états-transitions

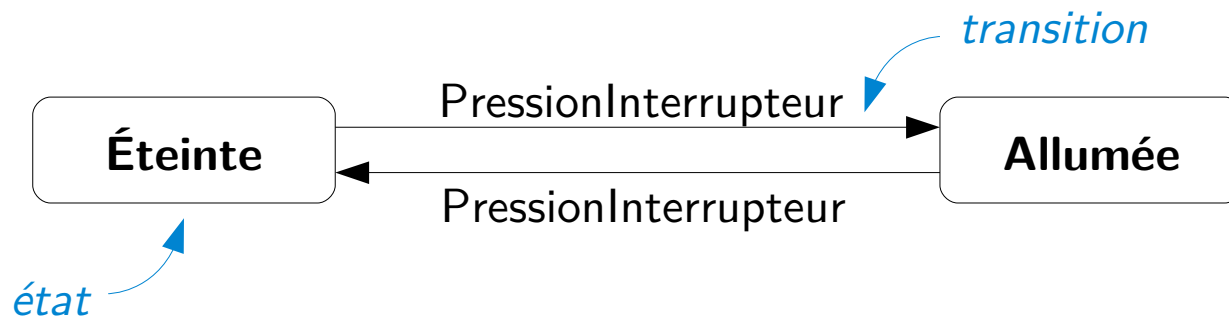
Delphine Longuet  
[delphine.longuet@lri.fr](mailto:delphine.longuet@lri.fr)

# Diagramme états-transitions

**Objectif** : Décrire le **comportement dynamique** d'une entité (logiciel, composant, objet...)

Comportement décrit par états + transitions entre les états

- **État** : abstraction d'un **moment de la vie d'une entité** pendant lequel elle satisfait un ensemble de conditions
- **Transition** : **changement d'état**



# Diagramme états-transitions

**Objectif** : Décrire le **comportement dynamique** d'une entité (logiciel, composant, objet...)

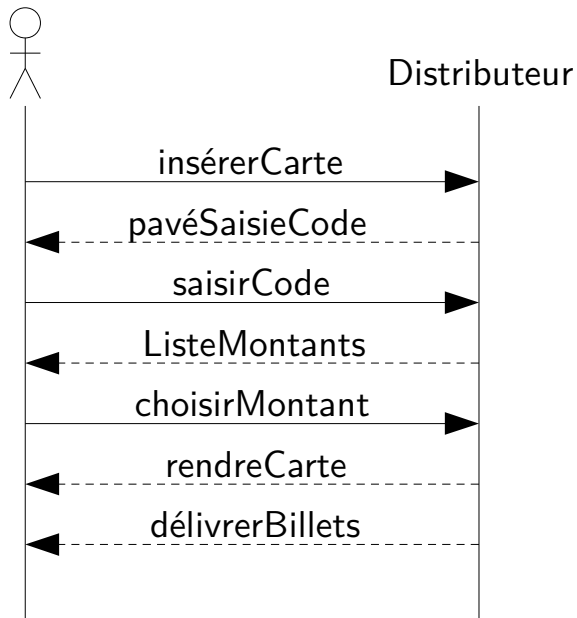
Comportement décrit par états + transitions entre les états

- **État** : abstraction d'un **moment de la vie d'une entité** pendant lequel elle satisfait un ensemble de conditions
- **Transition** : **changement** d'état

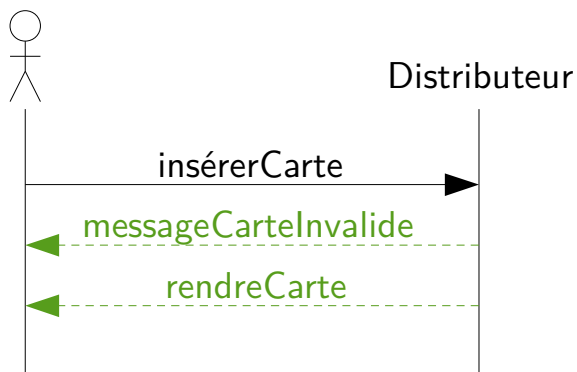
**Intérêt** :

- **Vue synthétique** de la dynamique de l'entité
- Regroupe un **ensemble de scénarios**

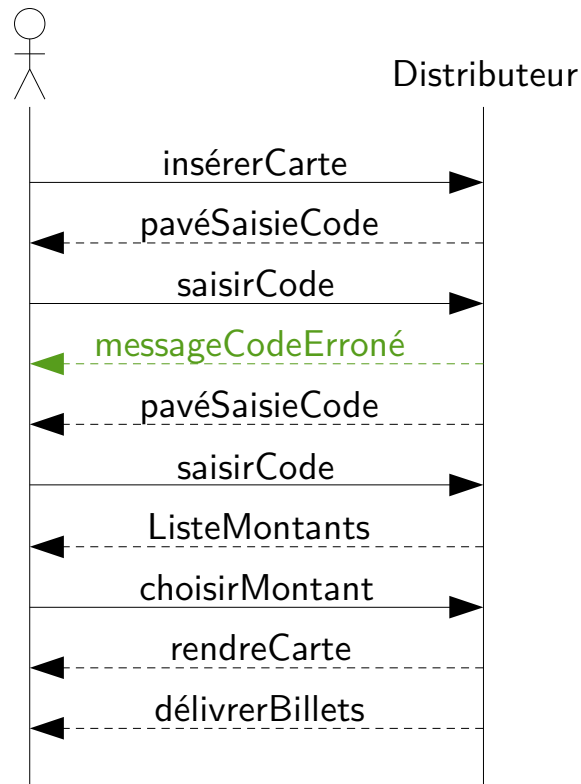
# Exemple : distributeur automatique



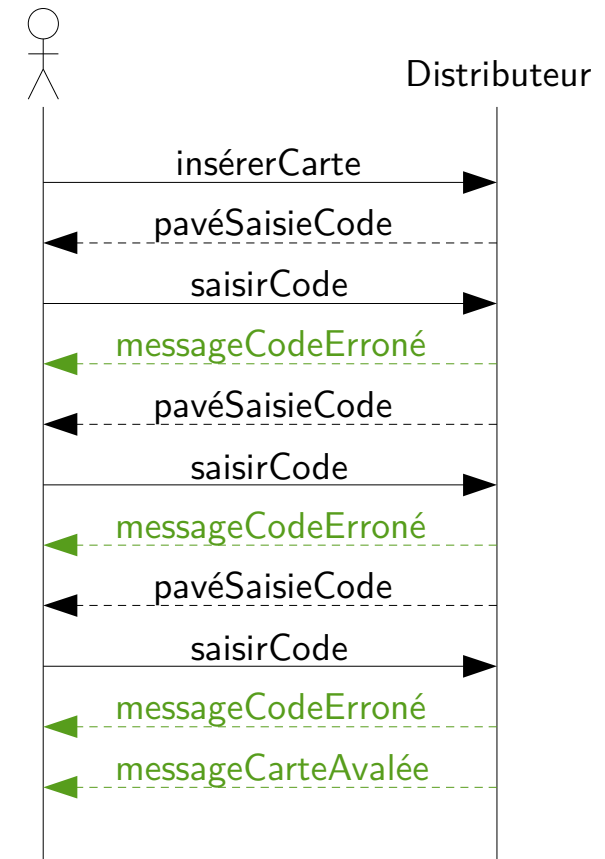
Scénario principal



Carte invalide

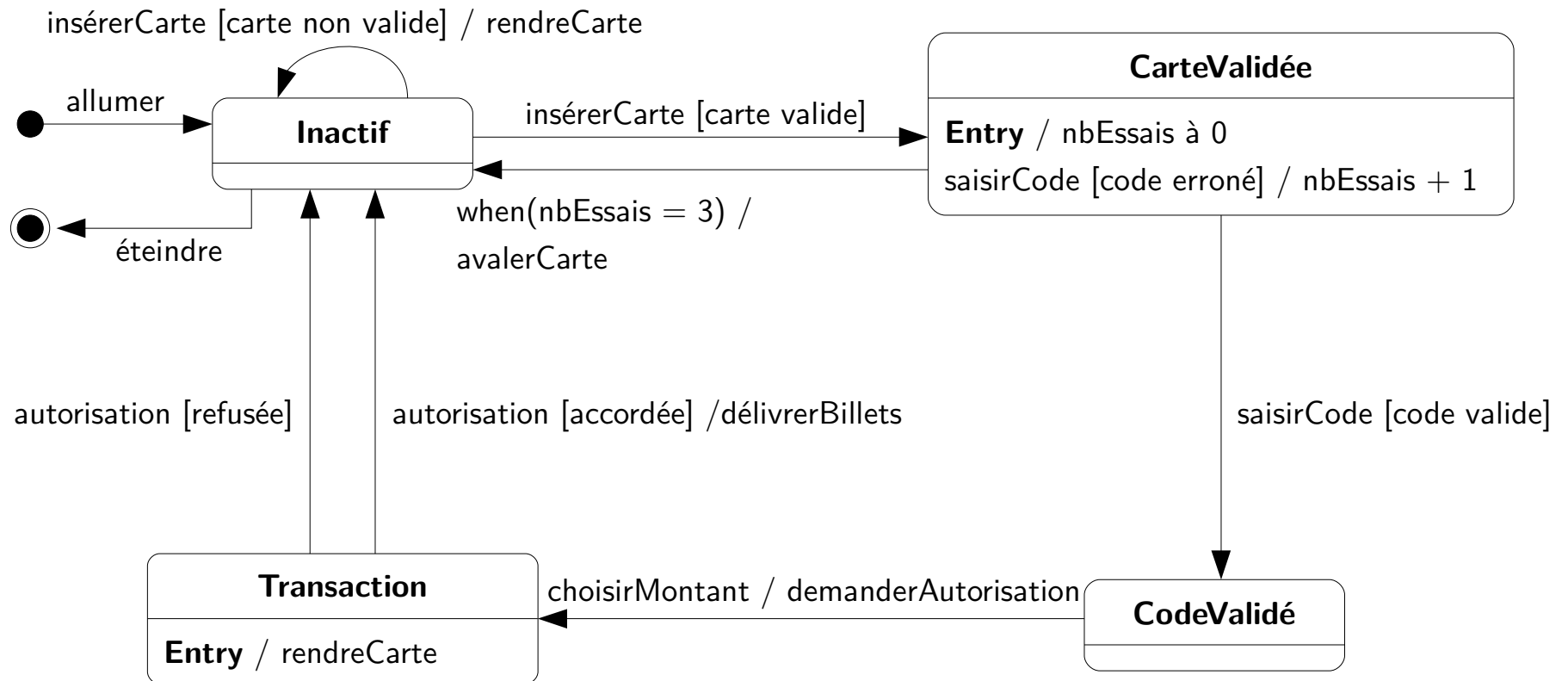


Une erreur de code



Trois erreurs de code

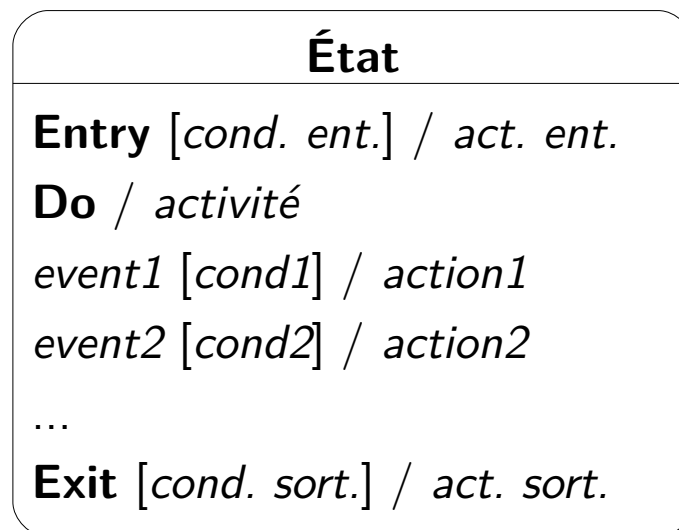
# Diagramme états-transitions correspondant



# État

## Types d'états

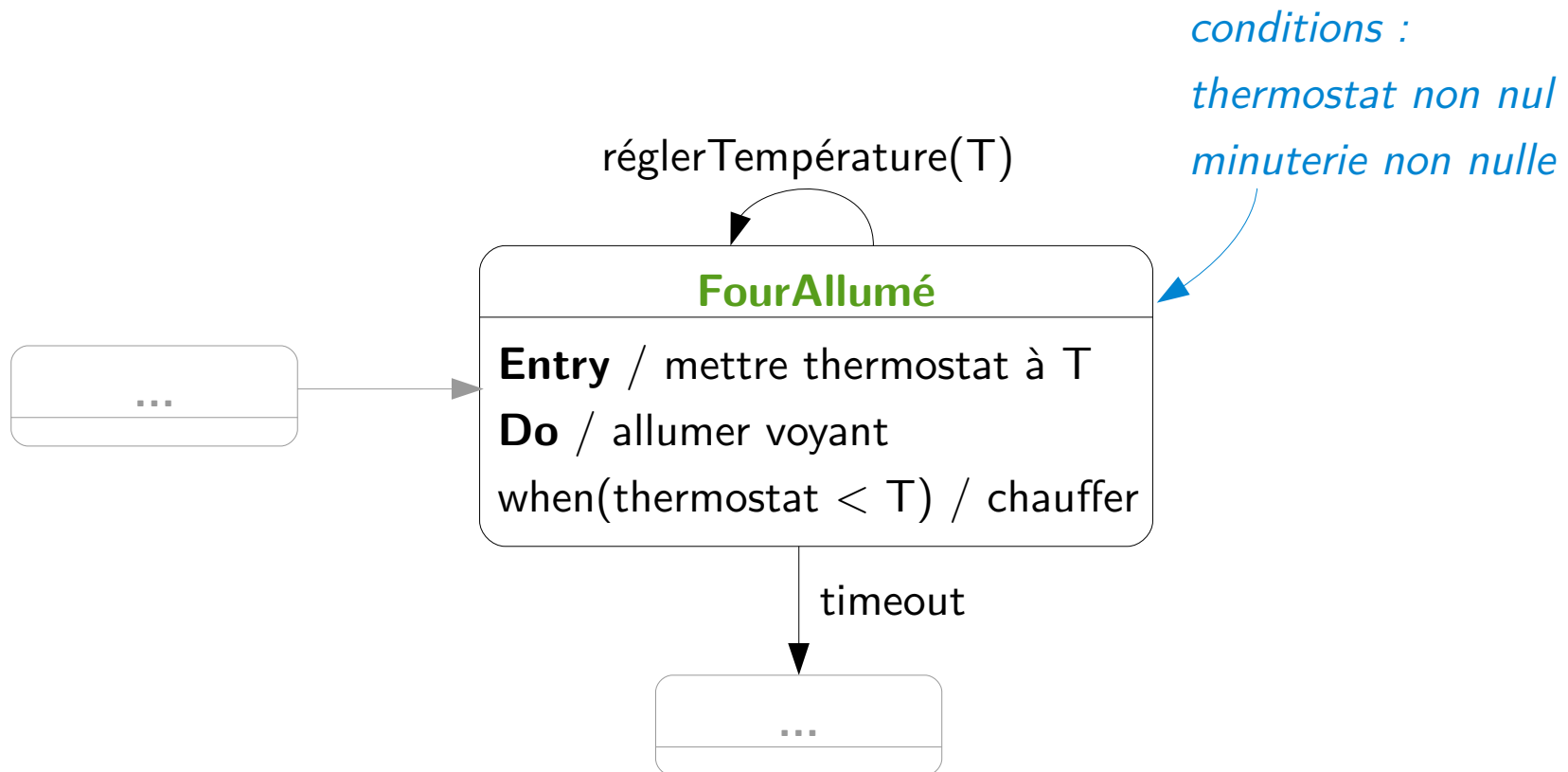
- **État initial** ●  
Initialisation du système, exécution du constructeur de l'objet
- **État final** ●  
Fin de vie du système, destruction de l'objet
- **États intermédiaires** : étapes de la vie du système, de l'objet



# État

## Caractéristiques d'un état :

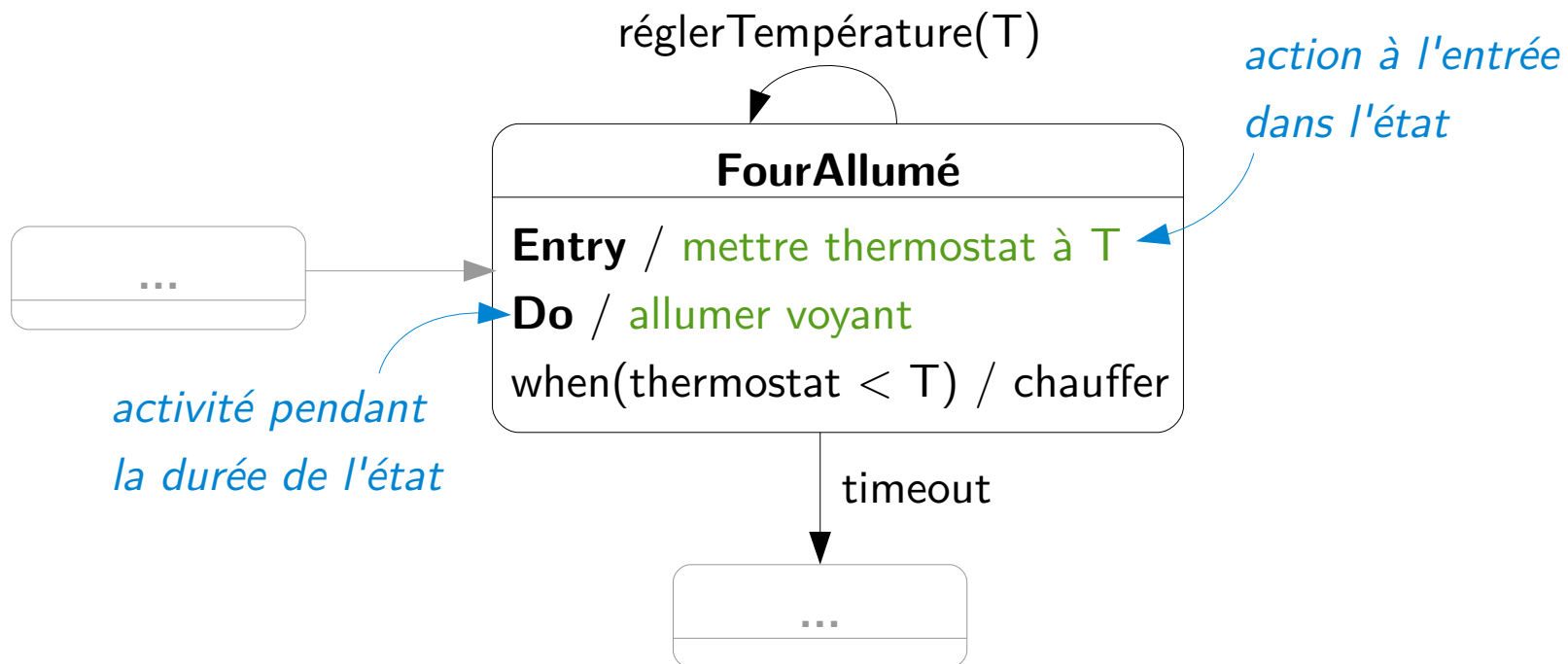
- Conditions vérifiées
- Actions réalisées
- Événements attendus



# État

## Caractéristiques d'un état :

- Conditions vérifiées
- **Actions réalisées**
- Événements attendus

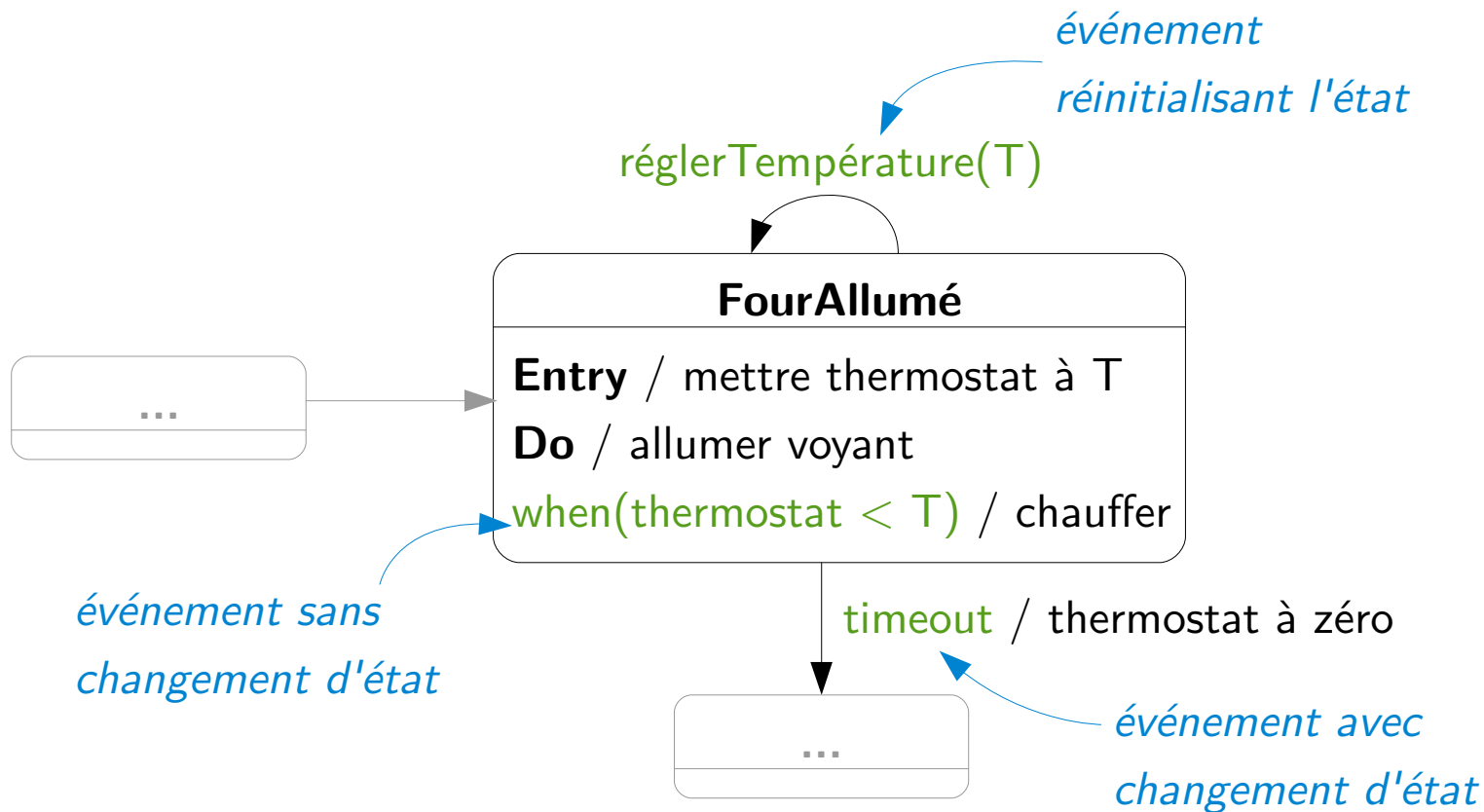




# État

## Caractéristiques d'un état :

- Conditions vérifiées
- Actions réalisées
- Événements attendus



# Événement

**Événement** : Fait **instantané** venant de **l'extérieur** du système et survenant à un **instant donné**

**Types d'événements** :

- **Signal** : réception d'un message asynchrone
- **Appel d'une opération** (synchrone) : liée aux cas d'utilisation, opération du diagramme de classes...
- **Satisfaction d'une condition** booléenne : **when**(*cond*), évaluée continuellement jusqu'à ce qu'elle soit vraie
- **Temps**
  - Date relative : **when**(date = *date*)
  - Date absolue : **after**(*durée*)

# Action

**Action** : Réaction du système à un événement

**Caractéristiques** : atomique, instantanée, non interruptible

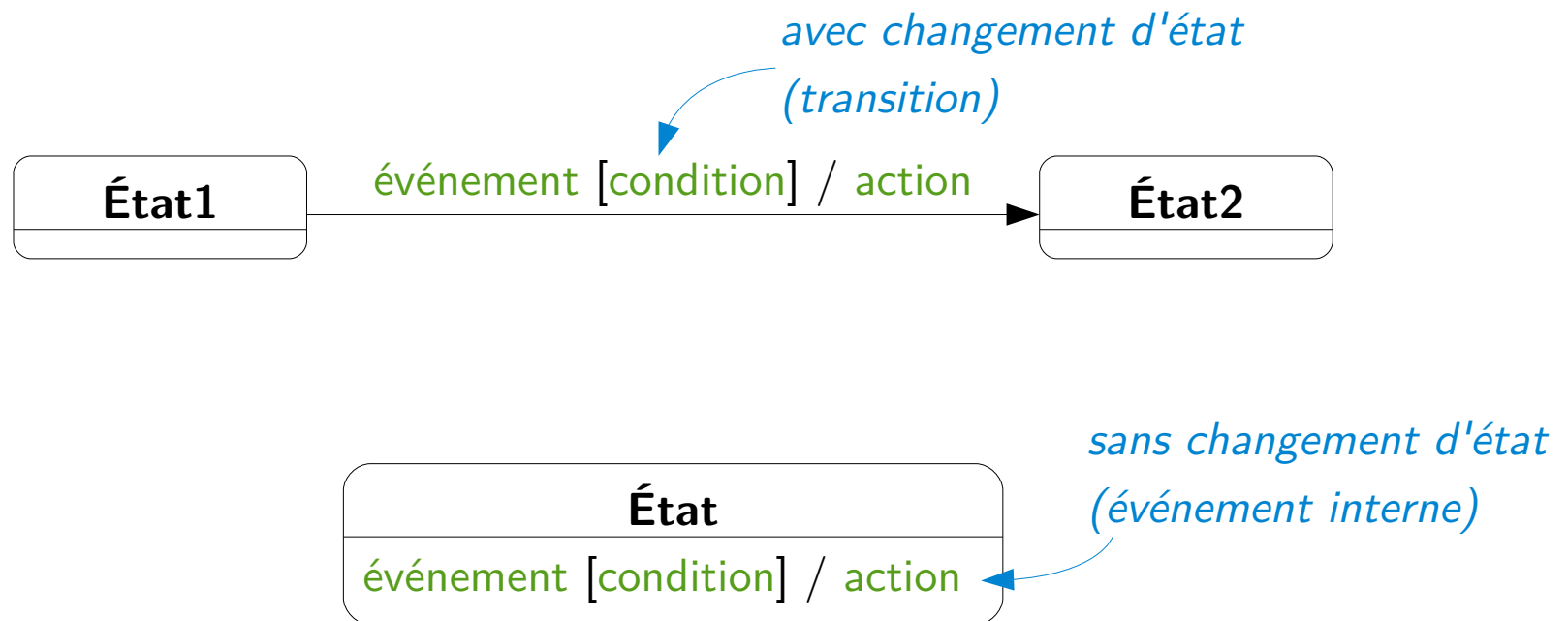
**Exemples d'actions** (syntaxe laissée libre) :

- affectation
- envoi d'un signal
- appel d'une opération
- création ou destruction d'un objet

# Action déclenchée par un événement

événement [condition] / action

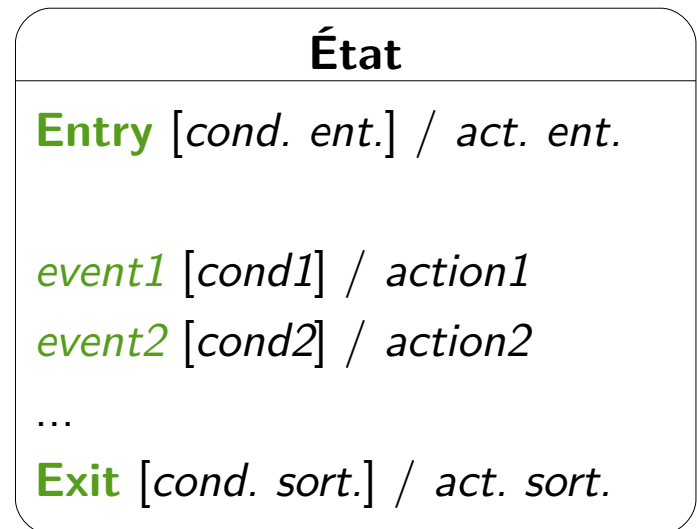
Lorsque l'événement se produit, si la condition est vérifiée, alors l'action est effectuée



# Dynamique d'un état

## Événements internes à l'état :

- Événement à l'entrée : **Entry**
- Événement à la sortie : **Exit**
- Événement sans changement d'état : *event*



# Dynamique d'un état

## Événements internes à l'état :

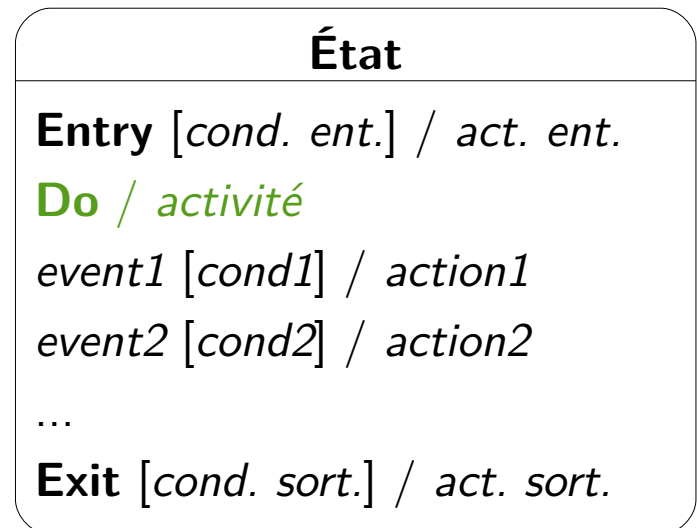
- Événement à l'entrée : **Entry**
- Événement à la sortie : **Exit**
- Événement sans changement d'état : *event*
- **Activité** : **Do**

## **Activité ≠ action**

- Pas instantanée
- Peut être interrompue

## Effet des événements internes :

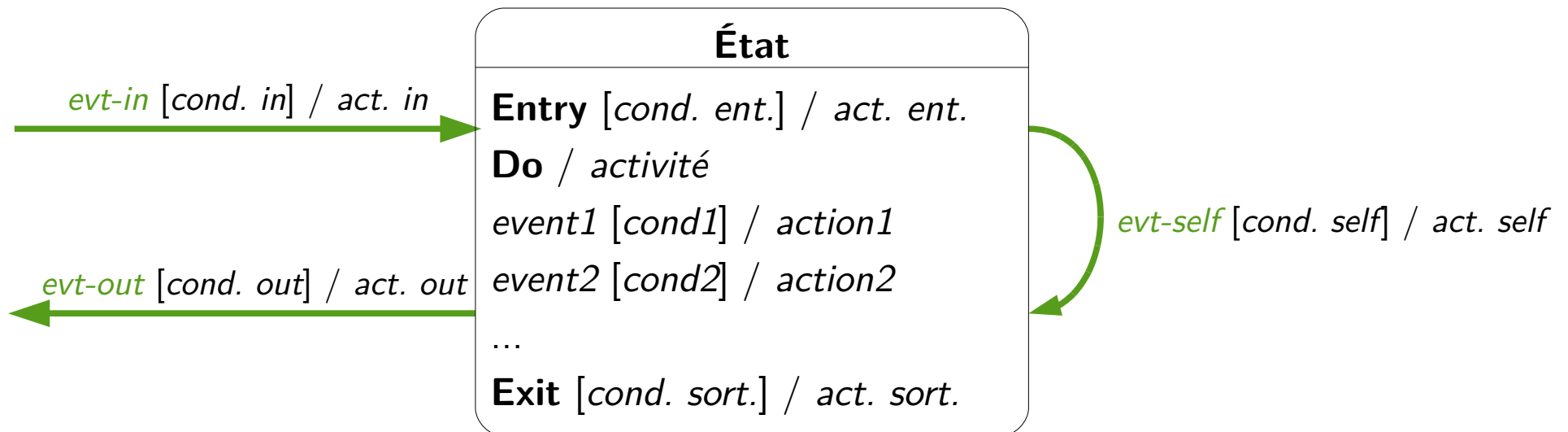
Interruption de l'activité **avec sauvegarde du contexte**



# Dynamique d'un état

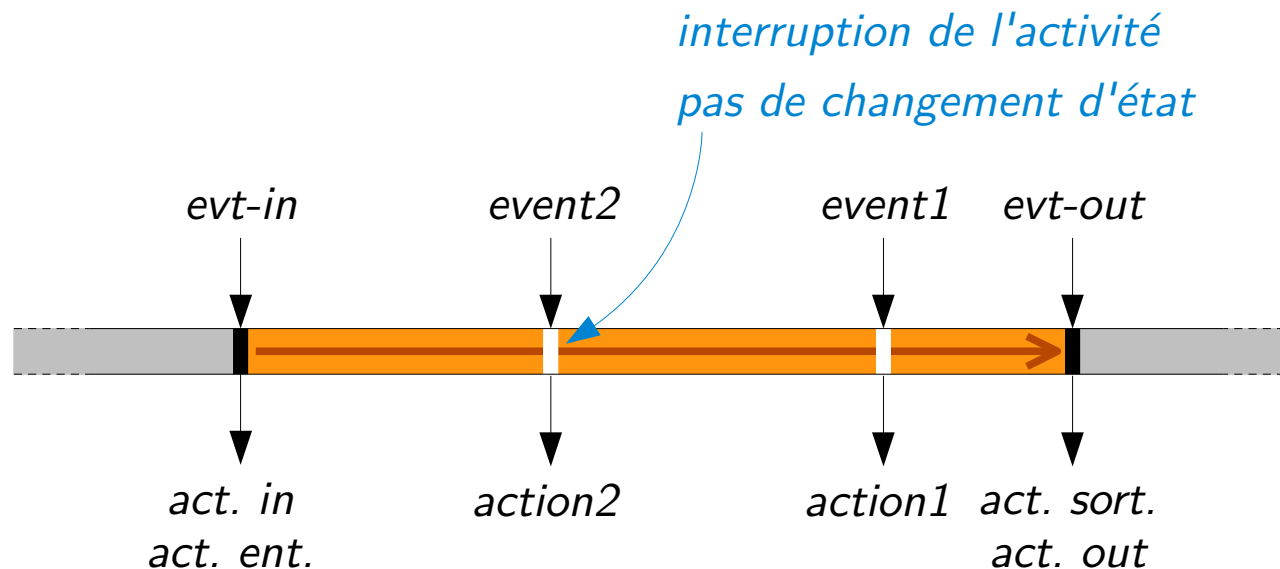
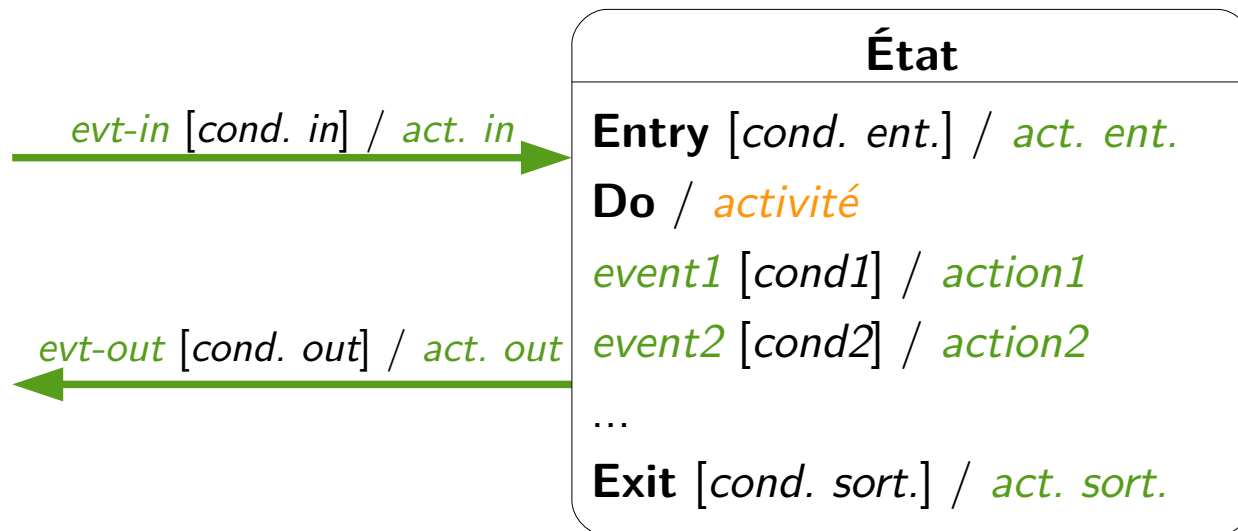
## Événements externes à l'état : transitions

- Transition vers l'état : *evt-in*
- Transition depuis l'état : *evt-out*
- Transition depuis l'état vers lui-même : *evt-self*



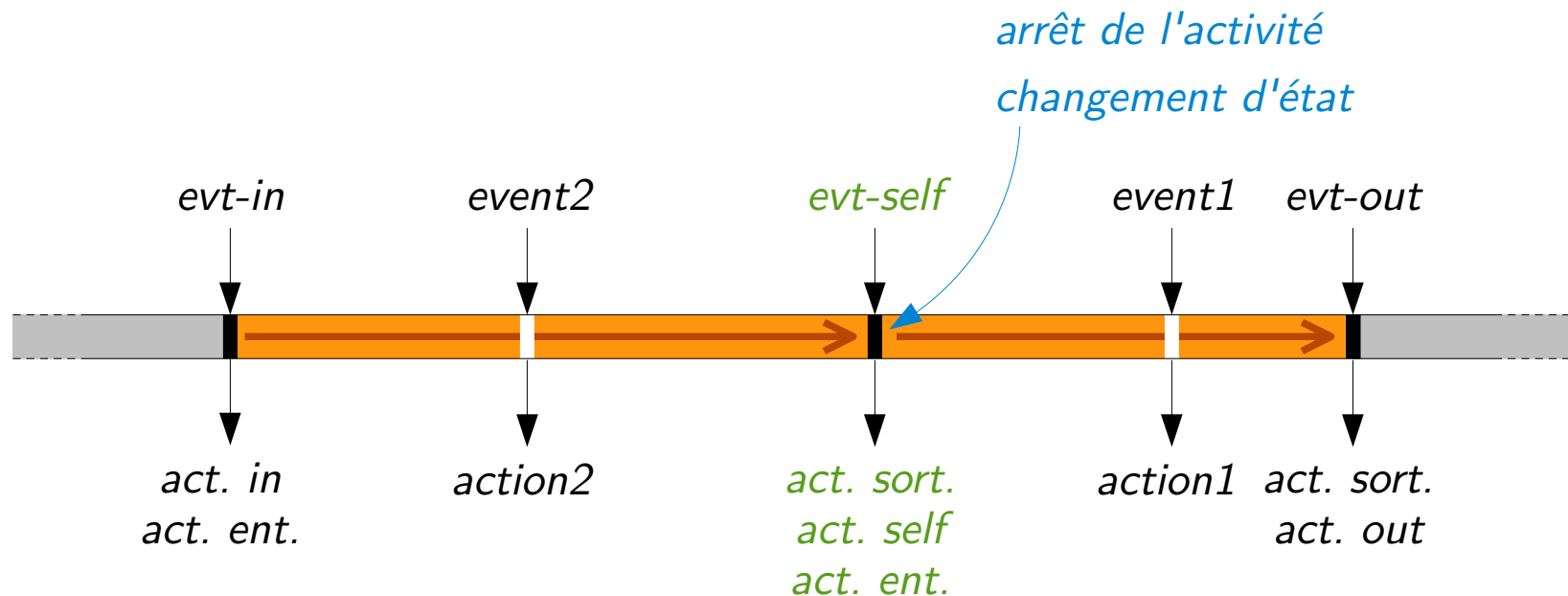
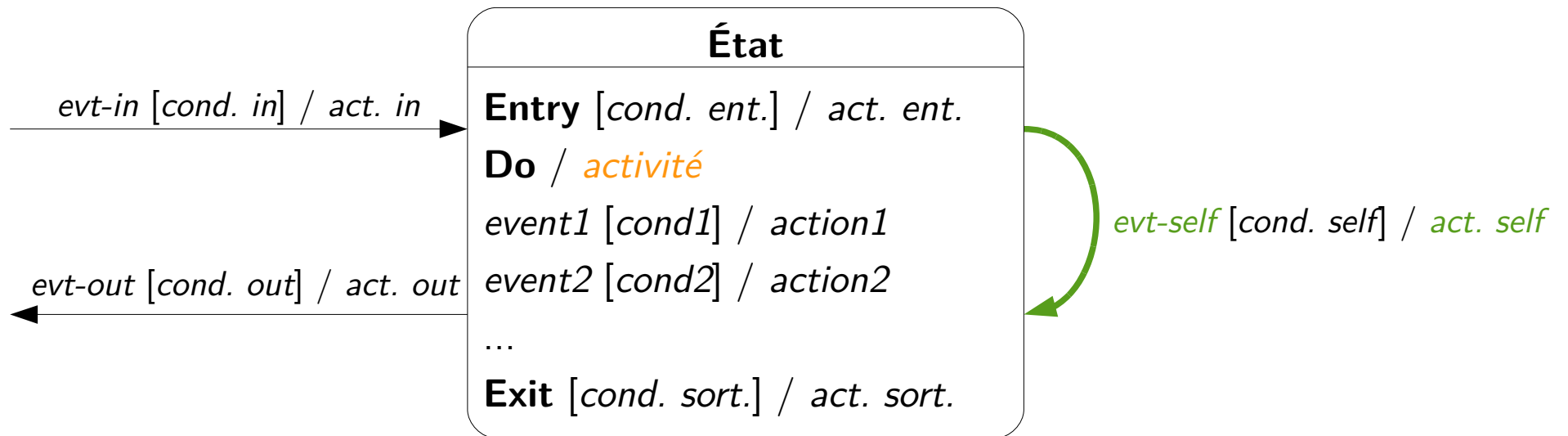
**Effet de *evt-self*** : Réinitialisation de l'état, interruption de l'activité sans sauvegarde du contexte

# Dynamique d'un état





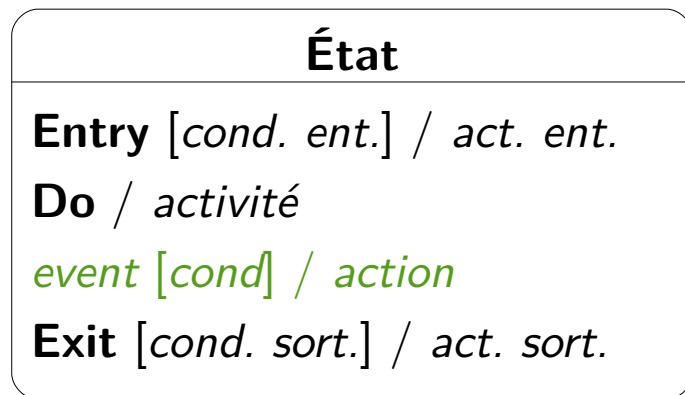
# Dynamique d'un état



# Dynamique d'un état

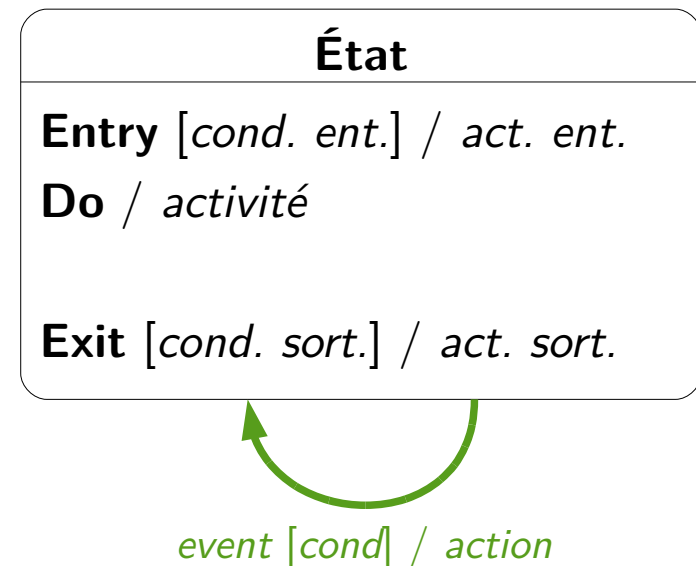
Ne pas confondre

Sauvegarde du contexte  
de l'activité pendant *action*

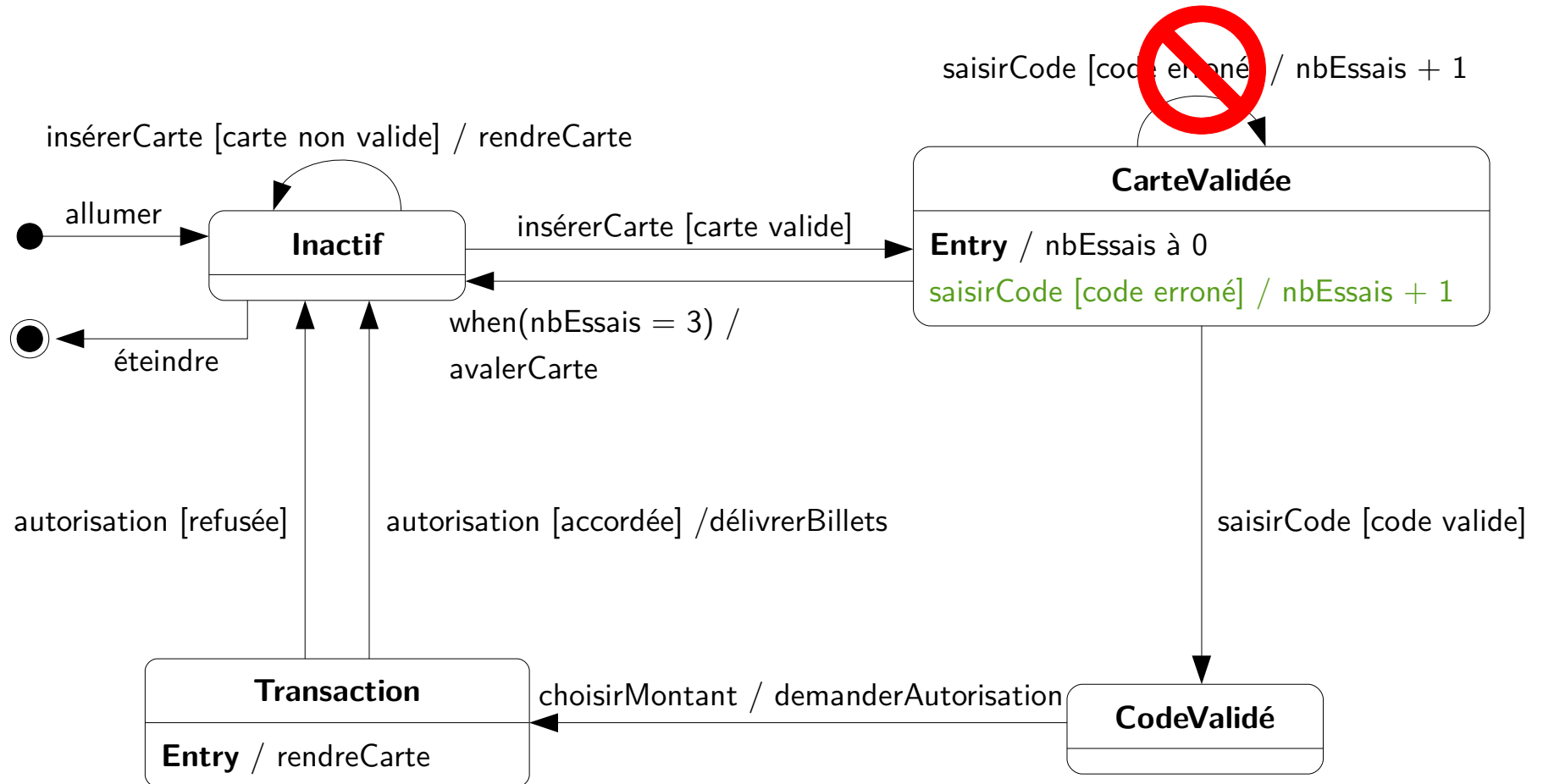


≠

Perte du contexte  
de l'activité pendant *action*



# Exemple – Distributeur automatique

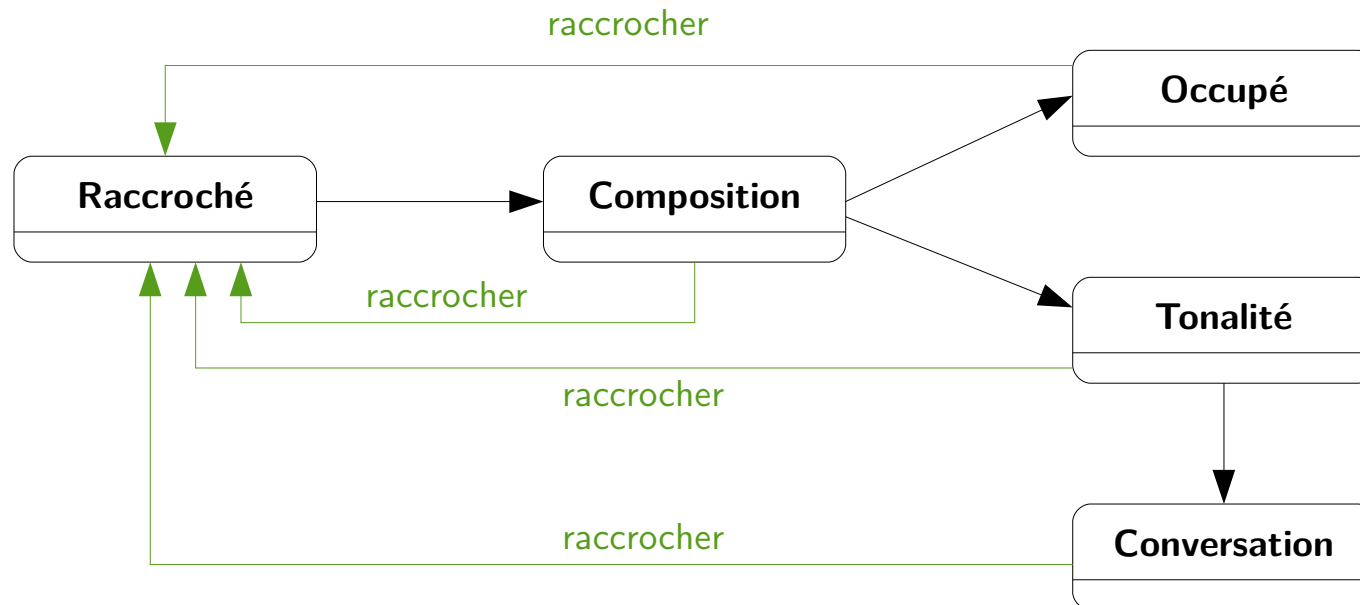


# États composites

État composite : État regroupant un ensemble d'états

Objectifs :

- Hiérarchiser les états
- Structurer les comportements complexes
- Factoriser les actions

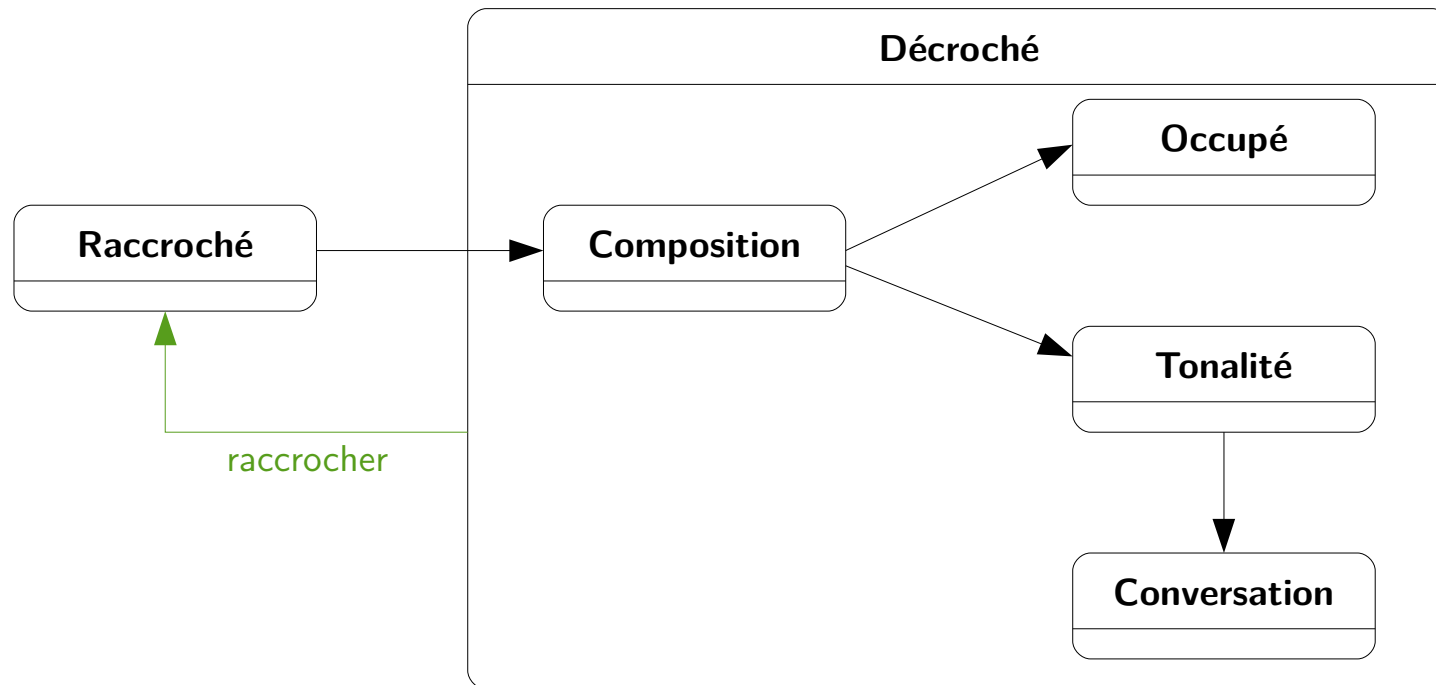


# États composites

État composite : État regroupant un ensemble d'états

Objectifs :

- Hiérarchiser les états
- Structurer les comportements complexes
- Factoriser les actions

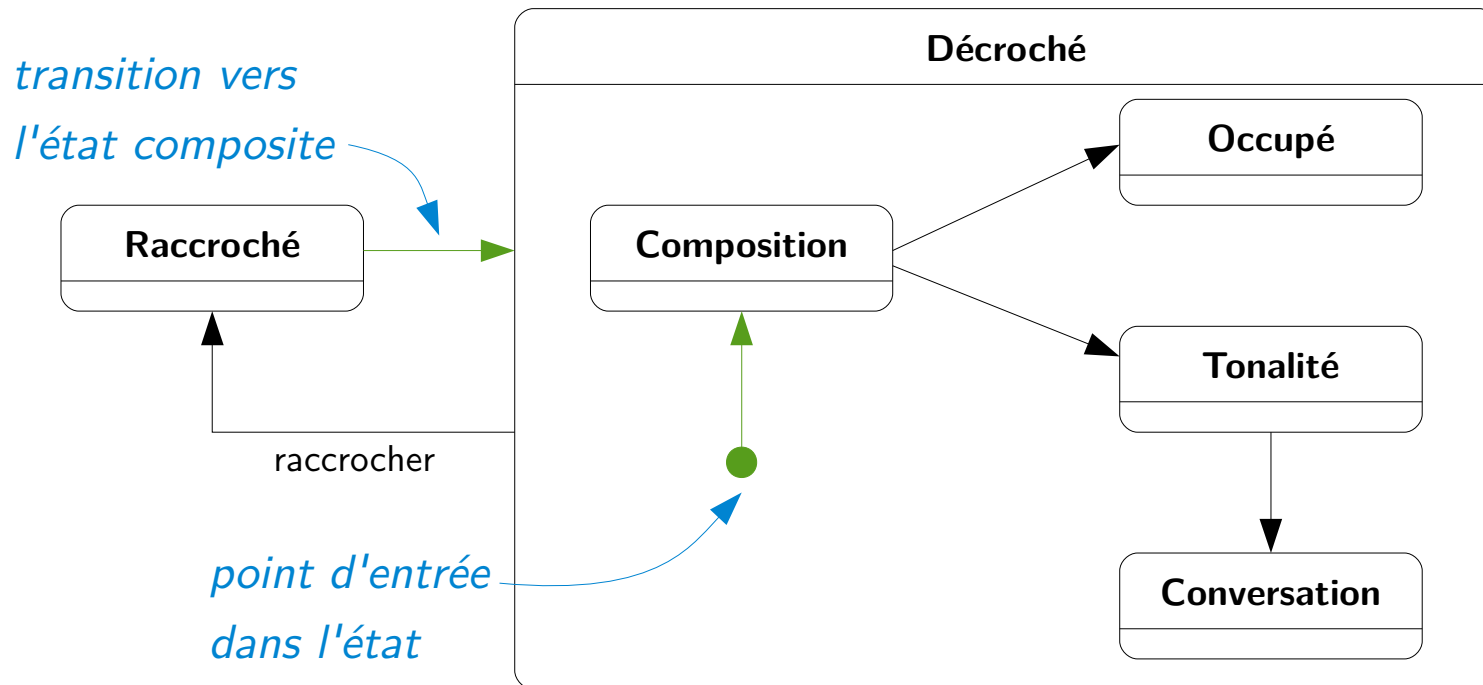


# États composites

État composite : État regroupant un ensemble d'états

Objectifs :

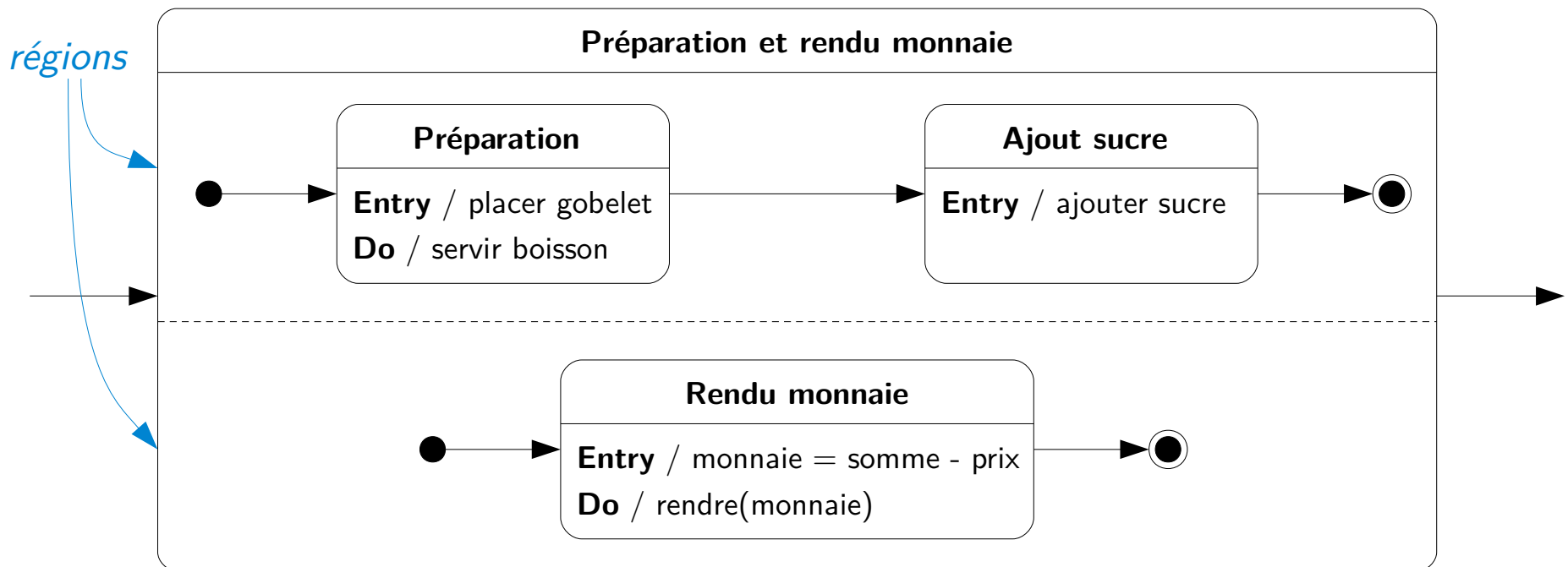
- Hiérarchiser les états
- Structurer les comportements complexes
- Factoriser les actions



# États composites

**État orthogonal** : État composite dans lequel **plusieurs états sont actifs simultanément** (concurrency/parallélisme)

État actif global = un état actif par région

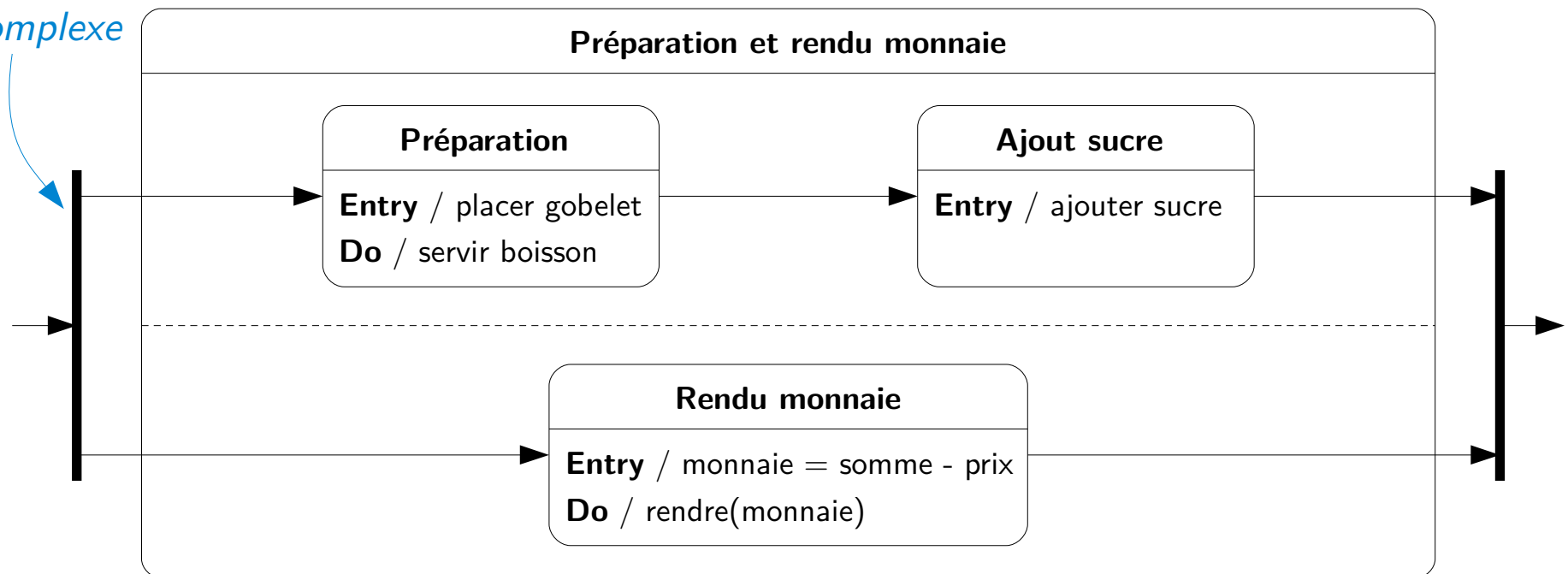


# États composites

**État orthogonal** : État composite dans lequel **plusieurs états sont actifs simultanément** (concurrency/parallélisme)

État actif global = un état actif par région

*transition complexe*



Représentation équivalente



# Utilisation des diagrammes états-transitions

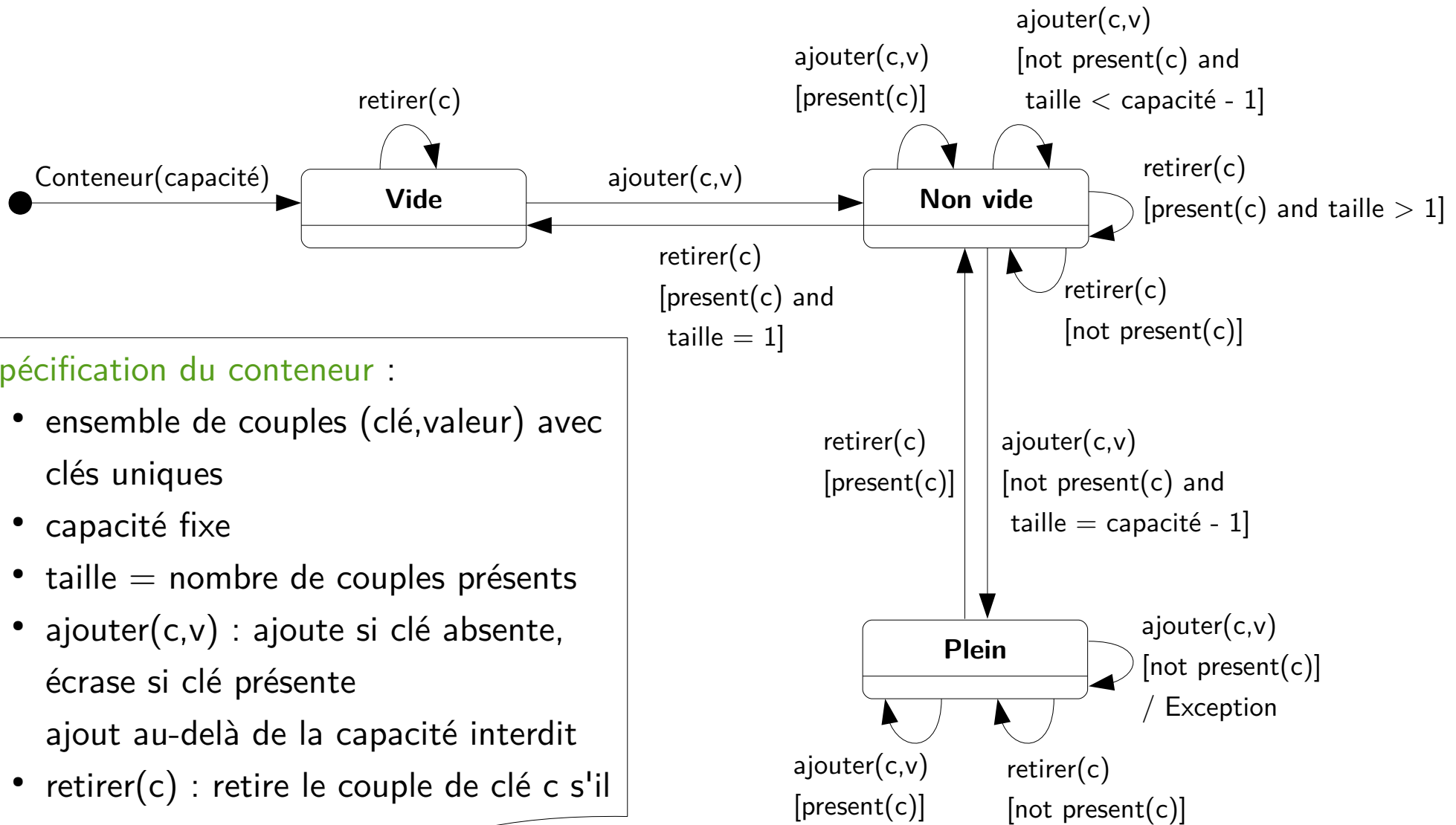
## En phase d'analyse :

- Description de la **dynamique du système** vu de l'extérieur
- Synthèse des scénarios liés aux **cas d'utilisation**
- Événements = **action des acteurs**

## En phase de conception :

- Description de la **dynamique d'un objet** particulier
- Événements = **appels d'opérations**

# Diagramme états-transitions d'un objet



## Spécification du conteneur :

- ensemble de couples (clé,valeur) avec clés uniques
- capacité fixe
- taille = nombre de couples présents
- `ajouter(c,v)` : ajoute si clé absente, écrase si clé présente  
ajout au-delà de la capacité interdit
- `retirer(c)` : retire le couple de clé `c` s'il existe

# Résumé rapide

# Utilisation d'UML en projet logiciel

## Analyse des besoins et spécification

- Diagramme des cas d'utilisation (+ description textuelle)
- Diagrammes de séquence de scénarios d'utilisation (scénarios génériques ou instanciés)
- Diagramme états-transitions du système global

## Conception

- Diagramme de classes
- (Diagrammes de séquence des cas d'utilisation détaillés par rapport au diagramme de classes)
- Diagrammes états-transitions d'objets