CLOO

Ph. Dugerdil

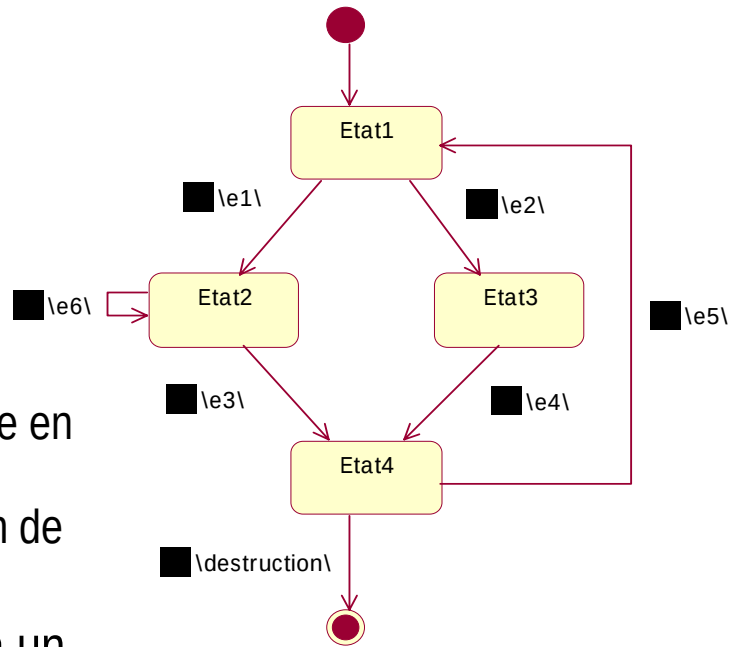
15.06.2004

Diagramme d’etats-transitions

StateCharts diagrams

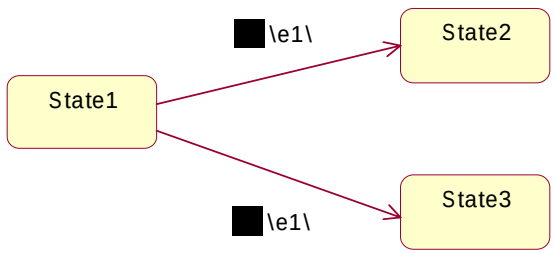
Graphe d'états-transitions

- Modèle formel: machines d'états finis
- Représentation: graphe orienté
  - Nœuds = états
  - Arcs = transitions
- Deux pseudo-états suplémentaires:
  - Début: pointe le premier état de la machine
  - Fin: indique que le système arrête de prendre en compte les évènements. (Fin de sa responsabilité, mise hors service, destruction de la machine,...).
- Transition: conditions de passage d'un état à un autre.
  - Saufs cas exeptionnels, toute transition possède un label indiquant les conditions de franchissement (événement, condition logique).



Machine déterministe

- Une machine d'états finis est dite:
  - Déterministe : étant donnés un état et un événement, l'état suivant est unique.
  - Non-déterministe : étant donnés un état et un événement, il y a plusieurs états suivants équiprobables.
    - Graphiquement:



Exercice – poste TV

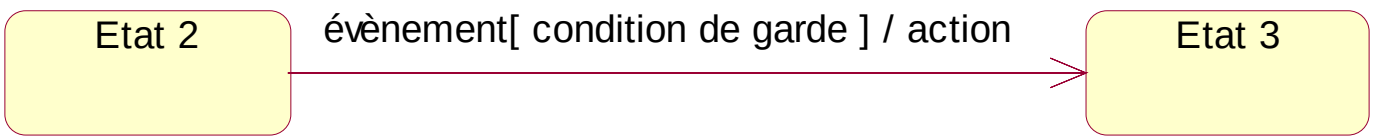
- Modèliser sous forme d'un graphe d'états l'enclenchement d'un poste de TV avec le bouton de secteur et la mise en marche sur la télécommande.
- 2 Evènements:
  - Bouton secteur appuyé
  - Bouton télécommande (0/1) appuyé.

Usage

- Un graphe d'état-transitions s'utilise pour modéliser le comportement des objets complexes.
- Un graphe d'état-transitions s'applique
  - A un objet d'une classe donnée
  - A un sous-système ou composant
- Un graphe d'états-transition ne décrit pas l'intéraction entre plusieurs objets mais le comportement d'un objet quand il communique avec son entourage.

Transitions UML

- Les transitions sont effectuées lors de l'arrivée d'un événement. La réaction dépend:
  - De l'état de départ
  - De l'événement
  - De conditions logiques supplémentaires éventuelles



Evènement

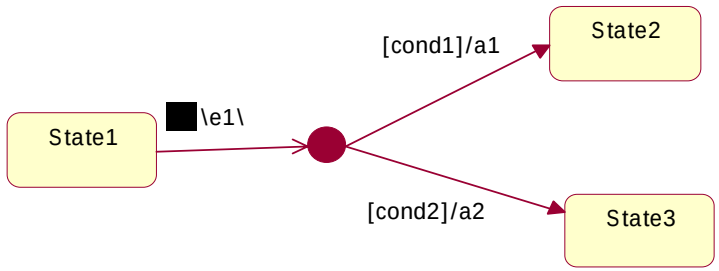
1. Réception d'un signal (communication asynchrone)
   - Syntaxe: nom du signal + paramètres

   ou réception d'un message (communication synchrone) envoyé par un autre objet
   - Syntaxe: signature du message

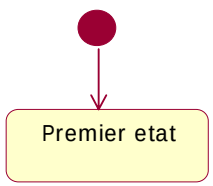
2. Passage du temps depuis l'entrée dans l'état
   - Syntaxe: **after**(expression temporelle d'une durée)
   - Exemple: after(10 sec.)

3. Réalisation d'une conditions booléenne
   - Syntaxe: **when** (condition booléenne)
   - Exemple: when(compteur = 30)

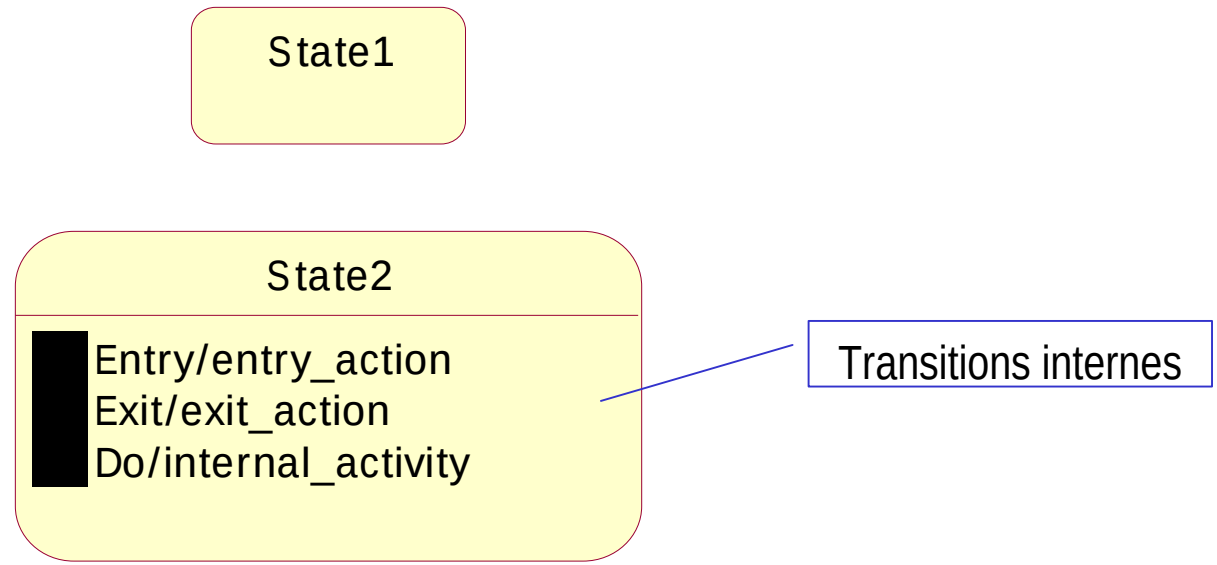
Notations



Peut être remplacé par



Etat UML: notation



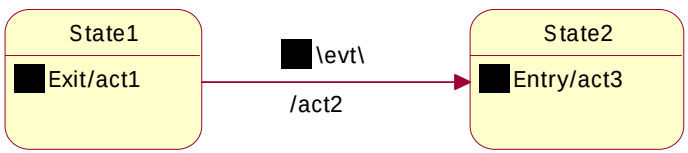
Expression des autres transitions internes :
event-name '(' *comma-separated-parameter-list* ')' '[' *guard-condition*']' '/' *action-expression*

Transitions propres et internes

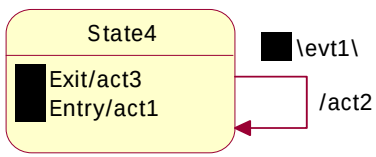
- Transitions propres
  - représentent une sortie de l'état puis une ré-entrée
    - Les actions de sortie et d'entrée sont réexécutées
    - L'activité interne est interrompue.
- Transitions internes
  - ne provoquent pas de sortie de l'état
    - Pas d'exécution des actions de sortie et d'entrée
    - Pas d'interruption de l'activité interne
- Activité
  - Opération ou tâche s'exécutant sur une certaine durée
    - Activité exécutée tant que l'objet est dans l'état
    - Activité exécutée jusqu'à sa terminaison (calcul par exemple). Possibilité d'événement « fin de calcul »

Ordre des actions

- Bien que les actions, ainsi que les transitions soient instantanées, il y a un ordre implicite.
  - Exit action de l'état source
  - Action de la transition
  - Entry action de l'état cible



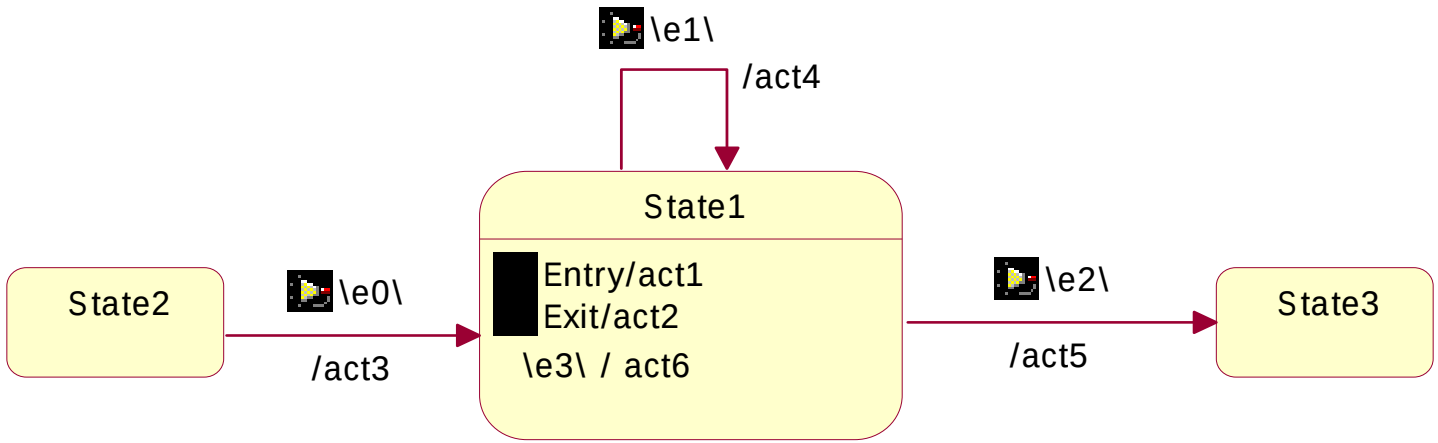
Si evt est lancé dans State1, l'ordre des actions est : act1, act2, act3



Si evt1 est lancé dans State4,l'ordre des actions est : act3, act2, act1

NB: si la transition propre était interne, on n'aurait eu qu'une exécution de act2

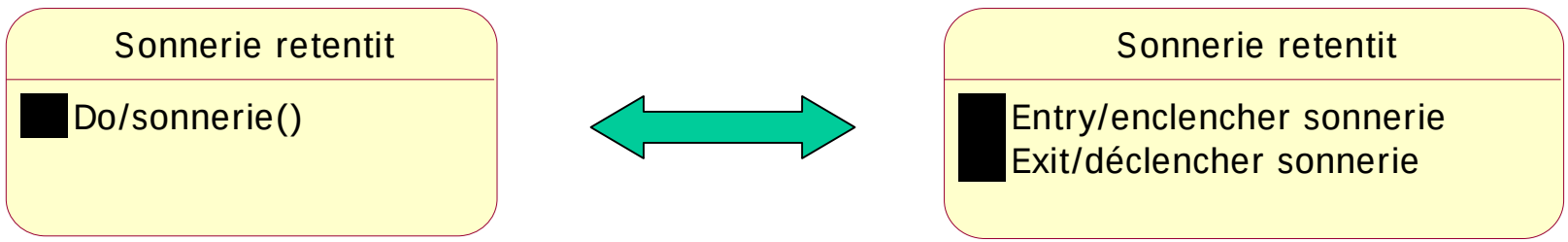
Exemple



Quelles sont les actions exécutées lors de la réception des évènements suivants (à partir de State2): e0, e1, e3, e2 ?

Activité ou actions

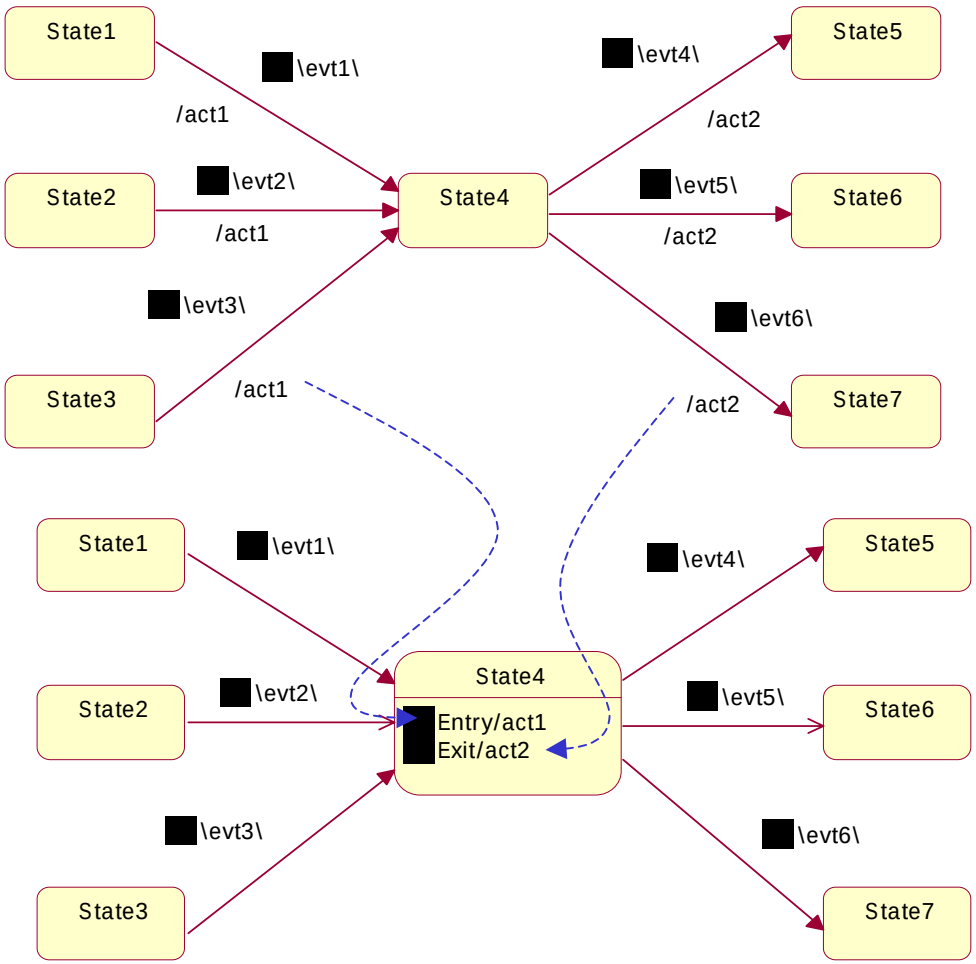
- Quand l'activité peut être enclenchée et arrêtée par une action, on remplace parfois cette activité par une paire d'actions.



Exercice - compteur

- Modéliser par un graphe d'états un objet représentant un compteur
  - Au départ, le compteur vaut 0
  - La méthode add() incrémente le compteur
  - Quand le compteur atteint 20, l'objet fait retentir une sonnerie en exécutant la méthode sonnerie().
  - L'exécution de add() ne modifie rien dès que la sonnerie retentit
  - La métode reset() fait repasser le compteur à 0.

Placement des actions

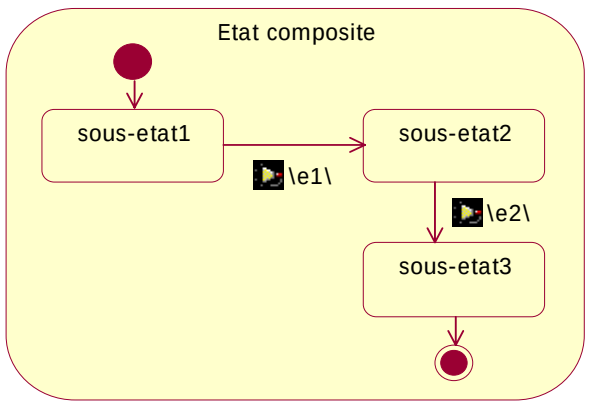


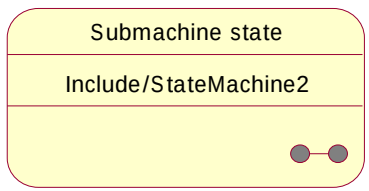
Exercice – numéro de téléphone

- Etablir le diagramme d'état d'un objet représentant la validation d'un numéro de téléphone national ou international:
  - Un numéro national doit être 0xn suivi de 7 chiffres, avec x différent de 0 et n = chiffre de 0 à 9.
  - Un numéro international doit être 00 xn ym suivi de 7 chiffres, avec x,y différents de 0 et n,m = chiffres de 0 à 9.
  - Si l'utilisateur n'entre pas un chiffre pendant une durée donnée, le système se met en timeout.
  - L'événement est constitué de l'unique méthode digit(n) avec n = chiffre de 0 à 9.

Etats composites

- Représentent une machine d'état « interne » à un état
  - Utilisation: description des machines d'états complexes sous forme hiérarchiques
  - Si la sous-machine est trop volumineuse: etat de sous-machine (submachine state) qui référence le diagramme de la sous-machine (Include)

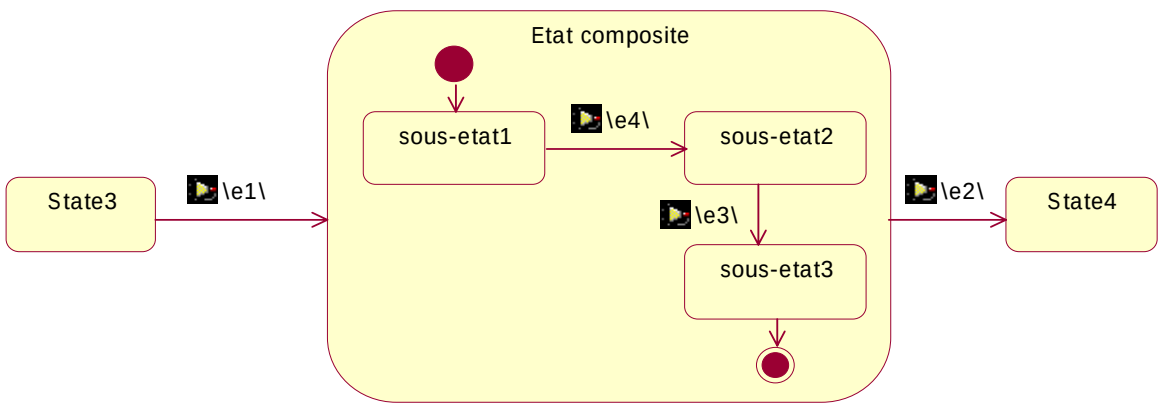




Transition de/vers un état composite

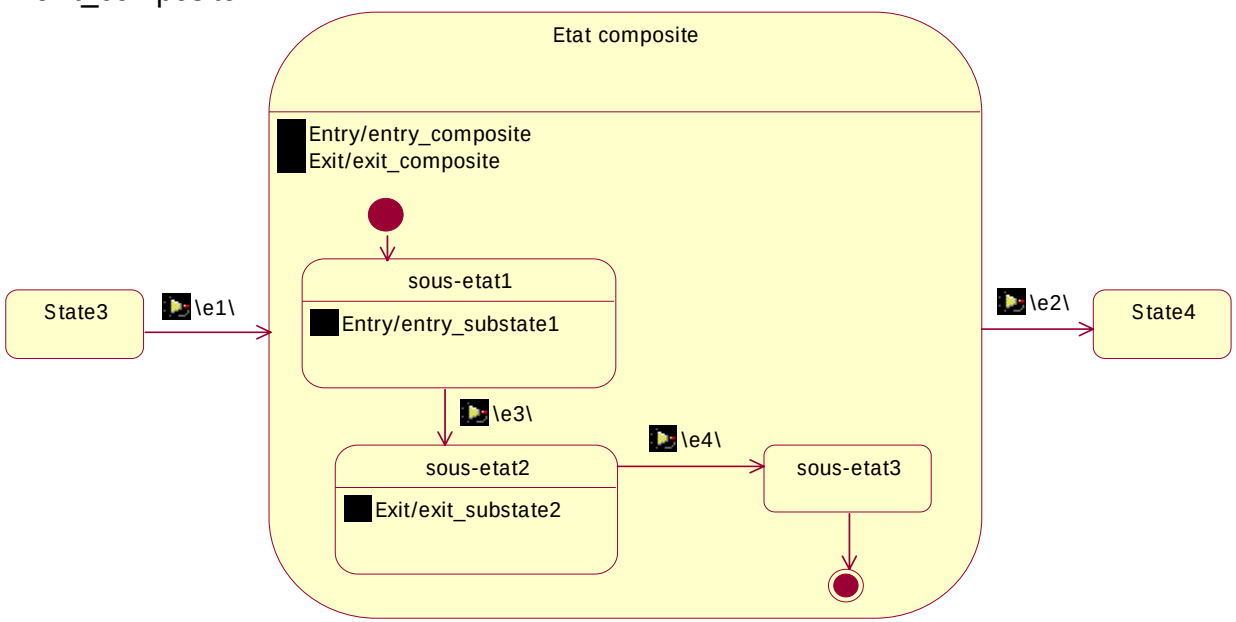
- Transaction entrante: transition vers le pseudo-état initial de la sous-machine
  - ou le dernier état actif (voir le point « historique »).
- Transaction sortante: force la sortie de la sous-machine. Equivalent à la transition depuis l'état courant de la sous-machine.

- Exemple:
  - L'événement e1 dans l'état State3 est une transition vers le sous-état1
  - L'événement e2 dans l'état composite, force la transition vers l'état State4, quelque soit le sous-état courant



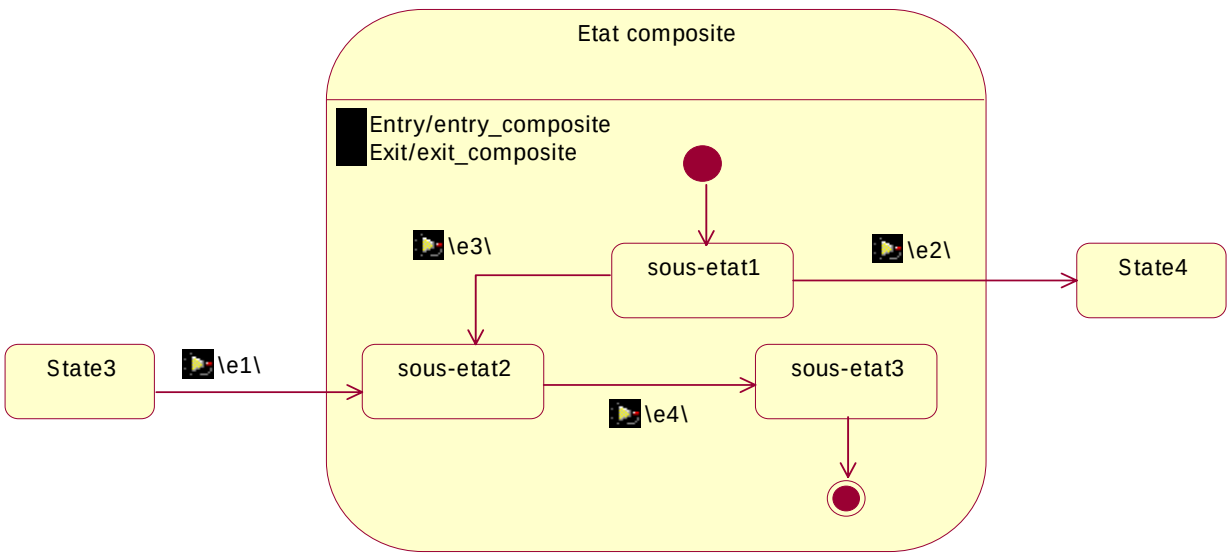
Etat composite et actions

- Lors des transitions de/vers une sous-machine d'état
  - Les actions d'entrée sont exécutées de l'état le plus général vers le plus détaillé (haut en bas)
  - Les actions de sortie sont exécutées de l'état le plus détaillé vers l'état le plus général (de bas en haut)
- Exemple:
  - En entrée dans l'état composite, on exécute entry_composite puis entry_substate1
  - En sortie de l'état composite, si on se trouve dans le sous-etat2, on exécute exit_substate2 puis exit_composite



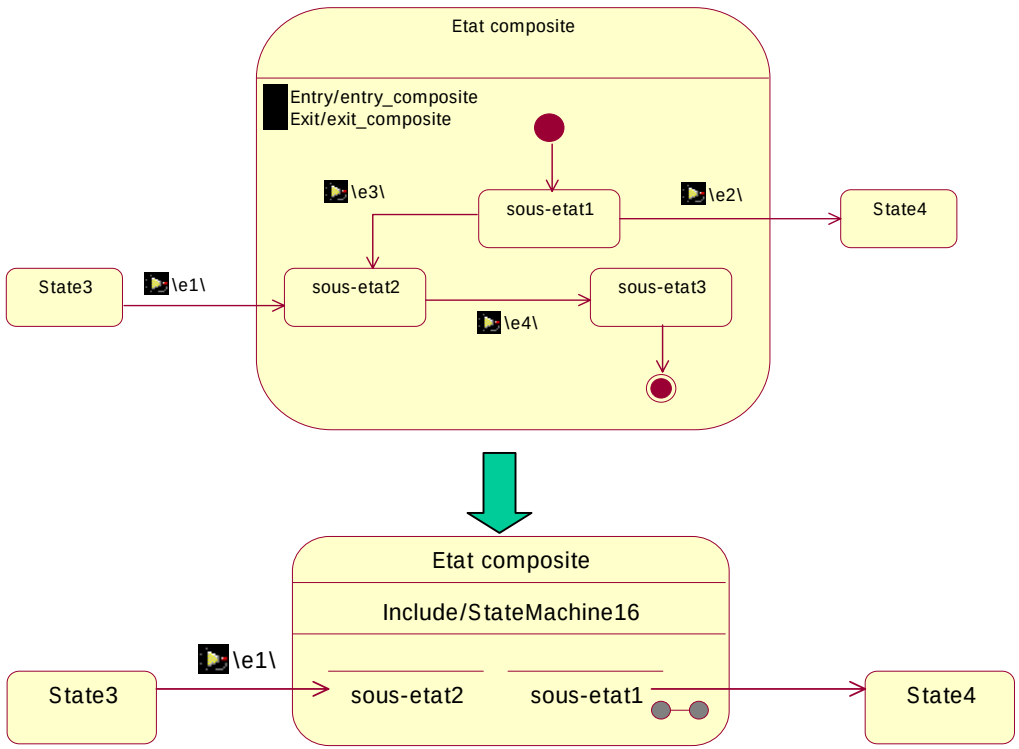
Transition de/vers un sous-etat

- Les transitions peuvent pointer un sous-état spécifique, ou sortir depuis un sous-stat spécifique.
- Exemple
  - Dans State3 si e1 est reçu, on passe au sous-etat 2 de l'état composite
  - Dans sous-etat1 si e2 est reçu, on passe à l'état State4.

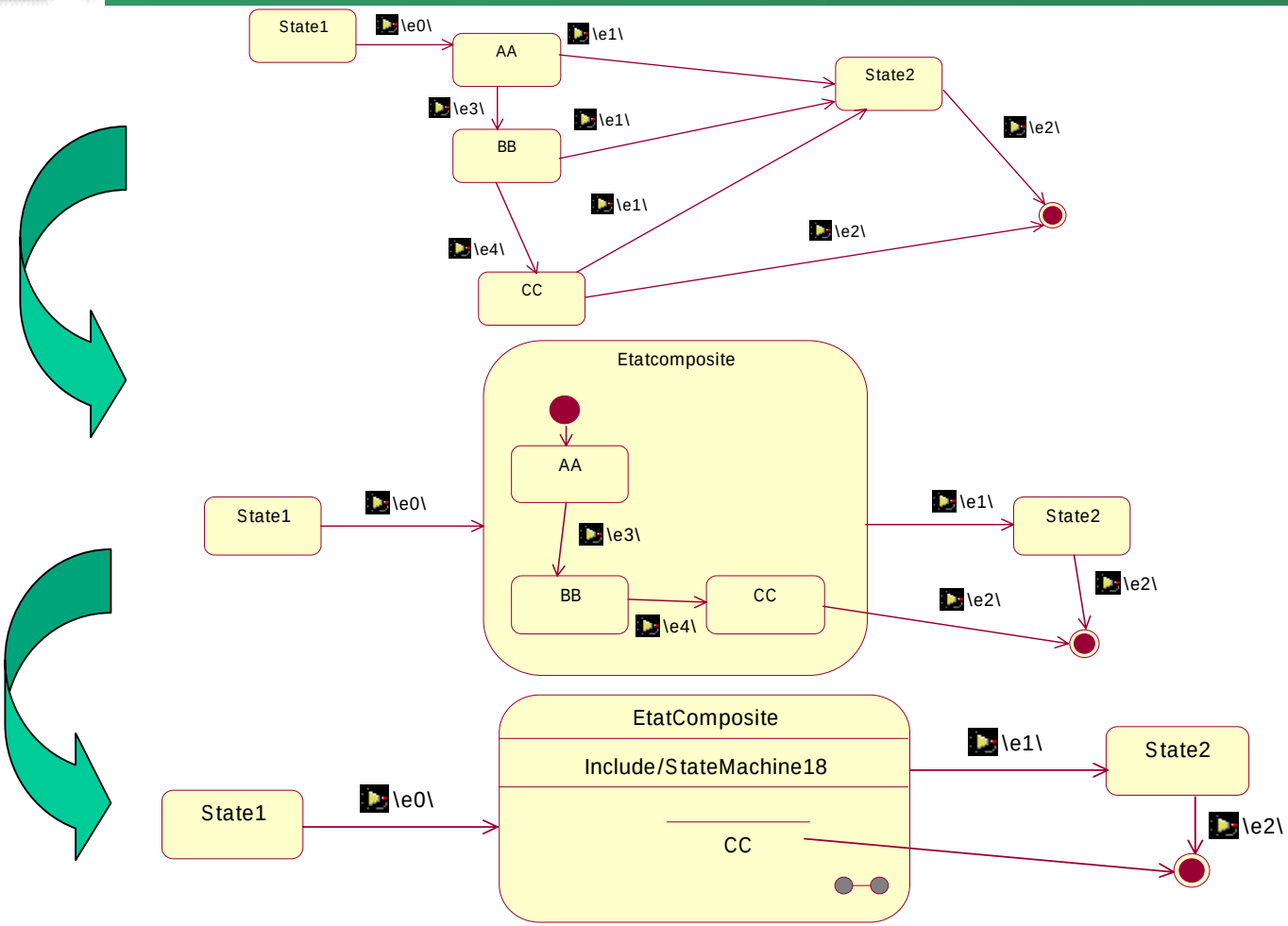


Proxy de sous-etat

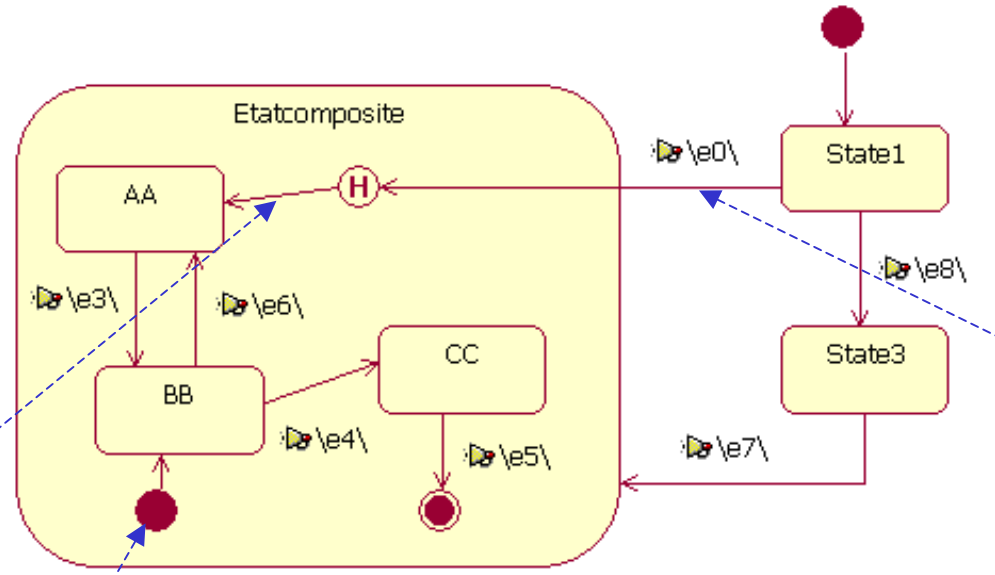
- Lorsqu'une transition va de/vers un sous-état mais qu'on ne veut pas représenter la sous-machine, il est possible d'indiquer un « proxy » pour le sous-etat.



Transformation et factorisation d'etats



Pseudo-etat « history » H



Pointe l'état de défaut s'il ny avait pas d'état actif au préalable
Obligatoire

On transite vers le dernier sous-etat actif

Etat de défaut si on fait une transitions d'entrée « stantard » (e7)

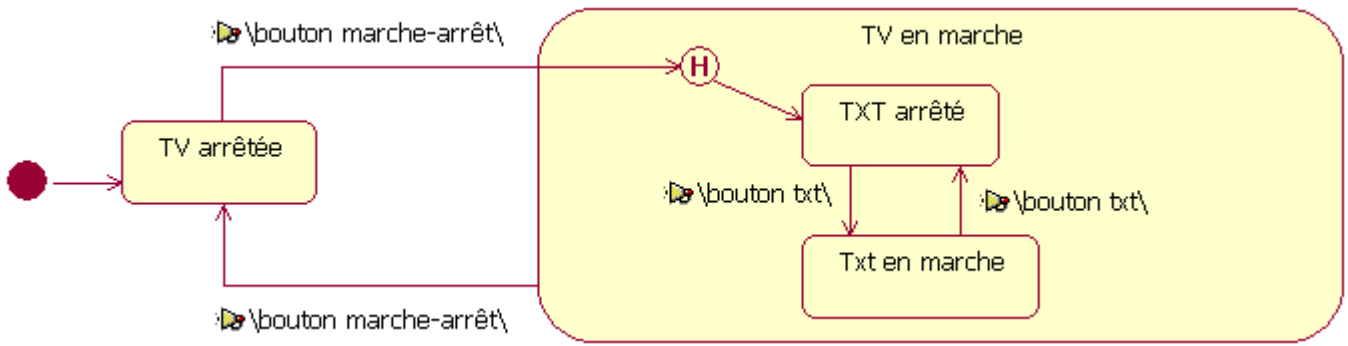
H « shallow history » (1 niveau)

H* « deep history » (tous niveaux, récursivement)

Exemple

- Simuler par une machine d'état l'affichage de sous-titres télétexte sur un téléviseur. La mise en marche de la TV sélectionne le mode TXT le plus récent.
- Evènements:
  - Bouton marche/arrêt sur la télécommande
  - Bouton txt sur la télécommande.

Solution



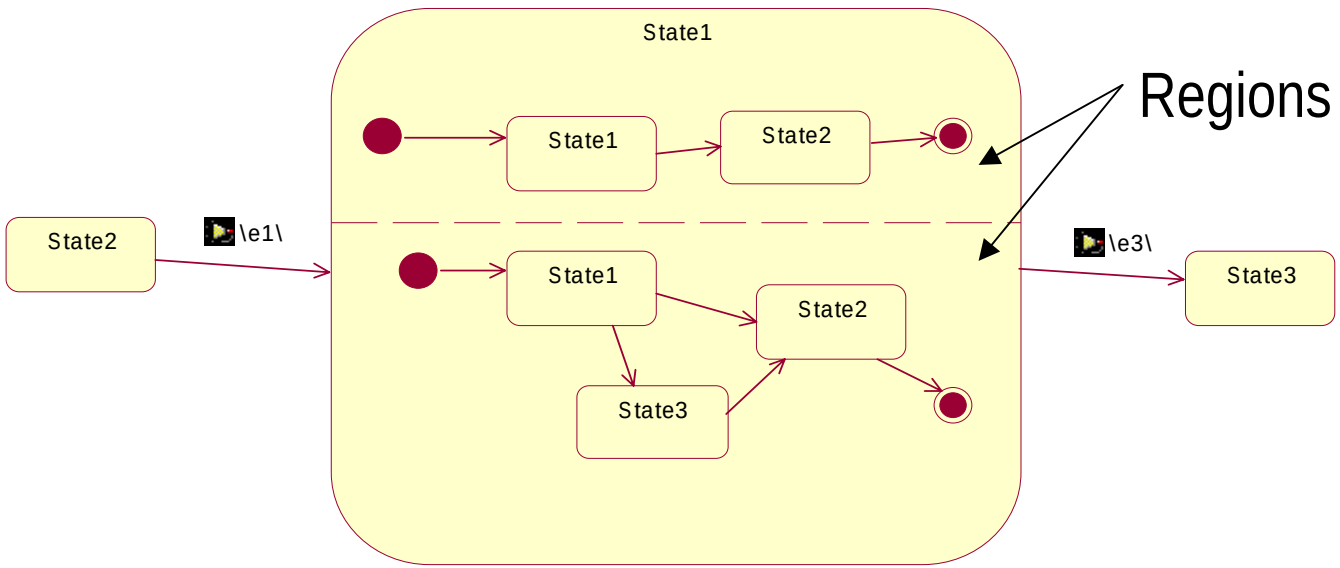
Exercice – radio-reveil

- Modèliser un radio réveil simple avec les fonctions suivantes:
- Si l'alarme est en marche et la radio éteinte, alors la radio s'allume au moment où l'heure de l'alarme est atteinte
- Si on appuie sur marche/arrêt alors que la radio est marche, elle s'arrête.
- Si on appuie sur marche/arrêt alors que la radio est eteinte, elle se met en marche.
- Si l'alarme est déclenchée, alors la radio ne s'allume plus avec l'heure présélectionnée.
- La mise en marche et arrêt de l'alarme ne peut se faire que lorsque la radio est éteinte

- Evènements
  - Radio marche-arrêt
  - Alarme marche-arrêt

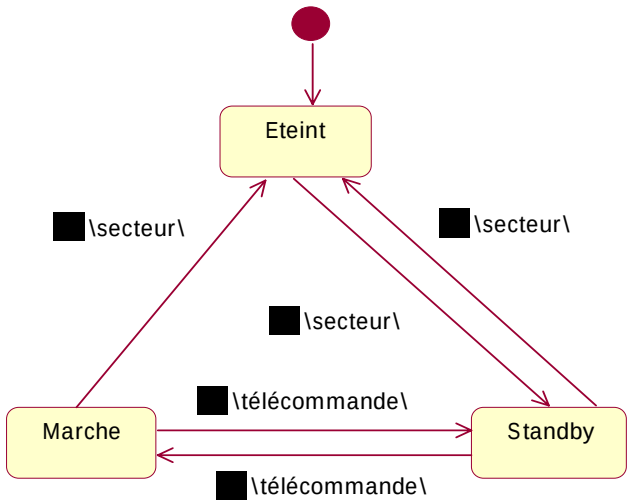
Etats composites à sous-machines concurrentes

- Un état composite peut comprendre plusieurs sous-machines concurrentes. Chaque machine est définie dans une « région ».
- Une transition vers l'état composite (exple: e1) représente une transition vers l'état initial de chaque sous-machine concurrente.
- Une transition sortant de l'état composite (exple: e3) représente un arrêt des sous-machiens concurrentes et une sortie de l'état composite.



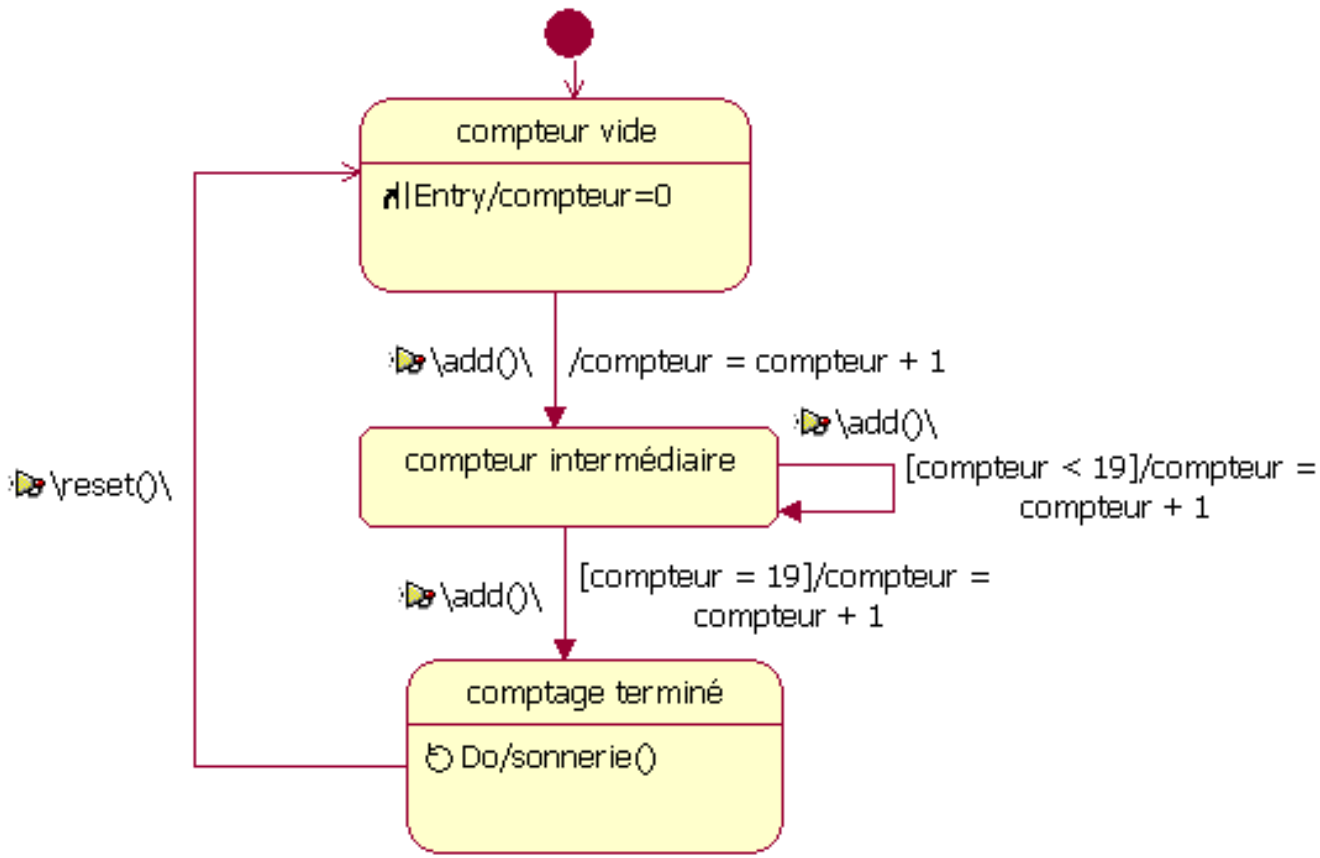
State graphs - solutions

Poste TV

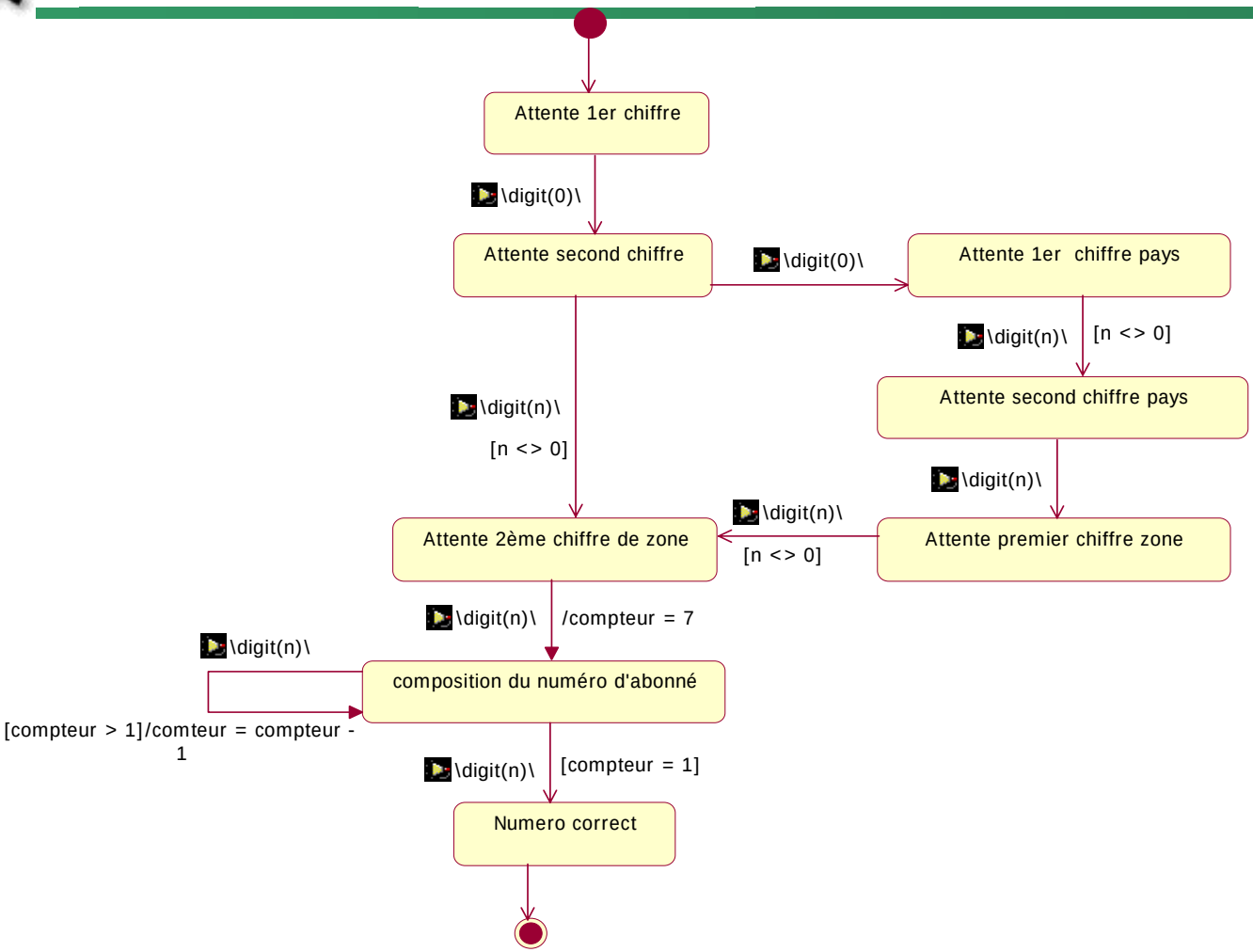


NB: Secteur = bouton secteur appuyé
Télécommande = bouton télécommande appuyé

Compteur

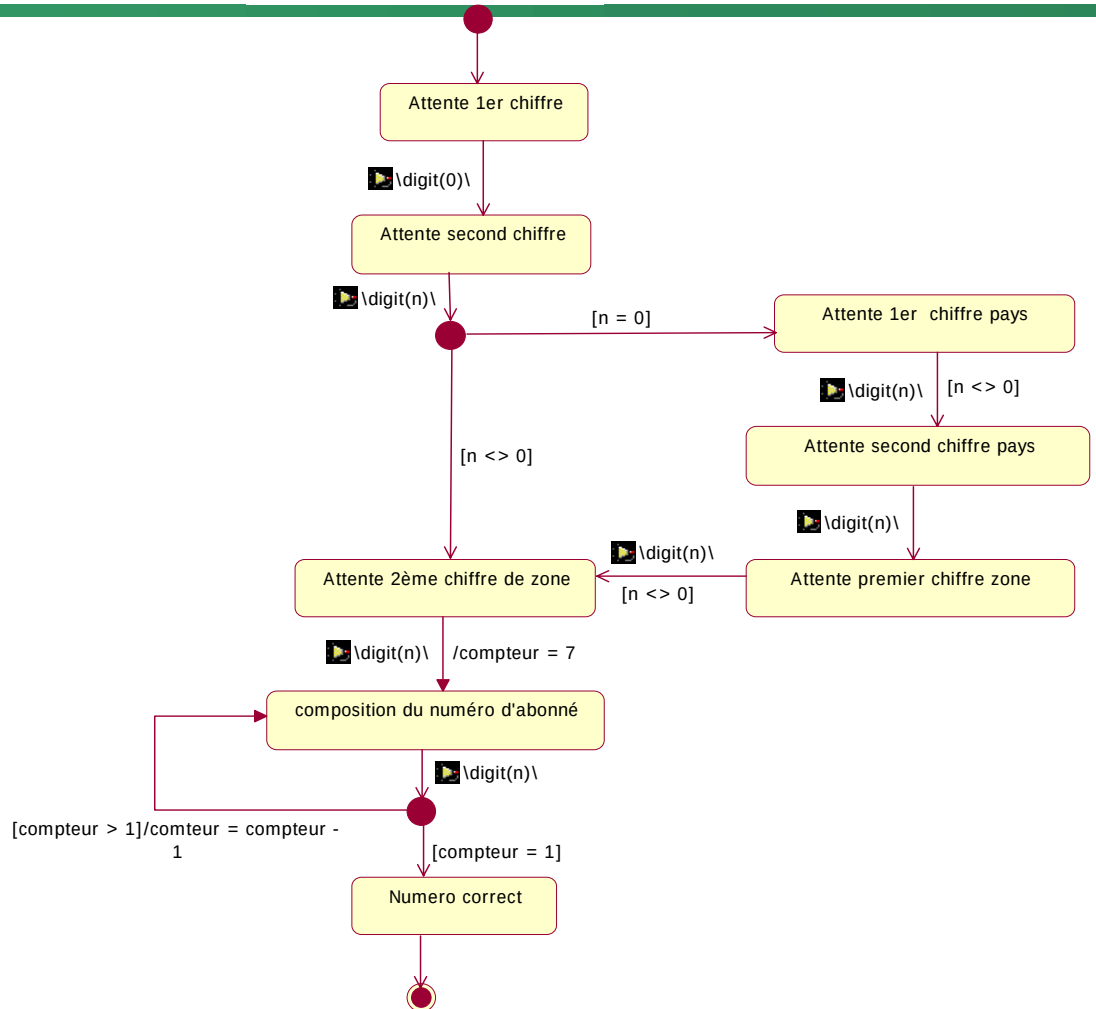


Validation numéro téléphone I



Validation numéro téléphone

représentation alternative



Radio-reveil